



universität
wien

DIPLOMARBEIT

Titel der Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten
über Biologie, Raum und Zeit

angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer.nat.)

Verfasser:	Stephan Hordynski
Matrikel-Nummer:	9227169
Studienrichtung (lt. Studienblatt):	A 444 Ökologie
Betreuer:	Ao. Univ.-Prof. Dr. Gert Michael Steiner Ass.-Prof. Mag. Dr. Karl Reiter

Wien, am 1. Juli 2008

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

*Habe nun, ach! Philosophie,
Juristerei und Medizin,
Und leider auch Theologie!
Durchaus studiert, mit heißem Bemühn.
Da steh ich nun, ich armer Tor!
Und bin so klug als wie zuvor;*

Goethe, Faust I

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

Inhaltsverzeichnis

Inhaltsverzeichnis.....	5
1 Einleitung.....	9
1.1 Problemstellung	12
1.2 Inhalt der Arbeit.....	13
1.2.1 Inhalte der einzelnen Kapitel.....	15
2 Vegetationsökologie.....	16
3 Methoden - Definitionen	18
3.1 Ökologische Informatik.....	18
3.2 Verteilung.....	18
3.2.1 Verteilte Systeme	18
3.2.2 Verteilte Anwendungen	19
3.2.3 Middleware	21
3.2.4 Architekturmodelle	22
3.2.4.1 Peer-to-Peer Architektur	23
3.2.4.2 Client/Server Architektur	23
3.2.4.2.1 n-Tier Architektur	24
3.2.4.2.2 3-Tier Architektur	25
3.3 Datenhaltung	27
3.3.1 Relationale Datenbanken	31
3.3.2 Relationale Datenbankmanagement Systeme (RDBMS).....	33
3.3.1.1 Oracle	34
3.3.1.2 MySQL	34
3.4 Anwendungslogik - Applikationsserver	35
3.4.1 Apache HTTP Server	35
3.4.2 PHP.....	35
3.4.3 Javascript	36
3.4.4 XML	37
3.4.5 Webservices	38
3.5 Analyse und Entwurf von Software	38
4 Konzeption.....	41

4.1 Allgemeine Systemübersicht.....	41
4.2 Beschreibung der Ausgangssituation (IST – Analyse).....	42
4.2.1 Organisation, Arbeitsabläufe	43
4.3 Beschreibung der gewünschten Situation (SOLL – Analyse).....	44
4.3.1 Organisation, Arbeitsabläufe	45
4.3.1.1 Grobe Ablaufarchitektur.....	46
4.3.2 Kernfunktionen	46
4.3.2.1 Weitere Vorgangsweise	50
4.3.2.2 Analyse Vegetationsaufnahmen	50
4.3.2.3 Analyse Botanische Namen.....	57
4.3.2.4 Analyse Administration	62
5 Entwurf.....	65
5.1 Allgemeine Datenbankstruktur	65
5.2 Internetkonzept (allgemein).....	65
5.2.1 Einleitung	65
5.3 Vegetationsaufnahme – Modul AUFNAHME	66
5.3.1 Datenmodell	66
5.3.2 Konzept Internetanwendung.....	70
5.3.2.1 Öffentlicher Bereich.....	71
5.3.2.2 Administrationsbereich	72
5.4 Botanische Namen – Modul BOTNAME.....	78
5.4.1 Datenmodell	78
5.4.2.1 Erweiterung des Datenmodells BOTNAME.....	80
5.4.3 Konzept Internetanwendung.....	81
5.4.3.1 Öffentlicher Bereich.....	81
5.4.3.2 Administrationsbereich	82
5.5 Administration – Modul ADMINISTRATION.....	83
5.4.1 Datenmodell	83
5.4.3.1 Öffentlicher Bereich.....	84
5.3.2.2 Administrationsbereich	84
6 Umsetzung.....	84
6.1 iFlora Öffentlicher Bereich (iFloraPublic).....	85

6.1.1 Allgemeine Informationen.....	86
6.1.1.1 Verzeichnisstruktur.....	87
6.1.1.2 Libraries.....	88
6.1.2 Startseite.....	88
6.1.2.1 Codierung.....	89
6.1.3 Auswahl	89
6.1.3.1 Codierung.....	102
6.1.4 Objektansicht.....	104
6.1.4.1 Codierung.....	109
6.1.5 Auswertungen.....	110
6.1.6 Downloads.....	110
6.2 iFlora Administration (iFloraAdmin).....	110
6.2.1 Allgemeine Informationen.....	113
6.2.1.1 Verzeichnisstruktur.....	114
6.2.1.2 Libraries.....	115
6.2.2 Startseite.....	115
6.2.2.1 Codierung.....	117
6.2.3 Systemverwaltung.....	117
6.2.3.1 Codierung.....	121
6.2.4 Administration.....	122
6.2.4.1 Codierung.....	126
6.2.5 Datenbearbeitung	127
6.2.5.1 Codierung.....	129
6.2.6 Auswertungen.....	131
6.2.7 Schnittstelle.....	131
6.2.7.1 Codierung.....	136
7 Zusammenfassung und Ausblick.....	137
7.1 Auswertungen.....	137
7.2 Modul GIS.....	137
7.3 Modul Botanische Namen.....	138
7.4 Modul Pflanzensoziologie	138
7.5 Modul Dokumente	139

7.6 Modul Sitemap	139
7.7 Modul Webservices.....	139
8 Literatur.....	140
9 ANHANG	141
9.1 Datenbankschema	141
9.1.1 CREATE SCHEMA - iFlora	141
9.2 Datenbankstruktur	141
9.2.1 CREATE STRUCTURE – Modul AUFNAHME	141
9.2.2 CREATE STRUCTURE – Modul BOTNAME	149
9.2.3 CREATE STRUCTURE – Modul METADATEN	152
9.3 Sourcecode.....	155
9.3.1 iFlora Administration	155
9.3.1.1 Programmbibliotheken (Libraries)	155
9.3.1.2 Controller für einzelnen Seiten	190
9.3.2 iFlora Öffentlicher Bereich.....	268
9.3.2.1 Programmbibliotheken (Libraries)	268
9.3.2.2 Controller für einzelnen Seiten	293
Lebenslauf	322

1 Einleitung

Eigentlich ist diese Diplomarbeit keine biologische Arbeit im klassischen Sinn! Vielmehr geht es um die schrittweise Entwicklung einer EDV-Infrastruktur für vegetationsökologische Daten und die punktuelle Lösung von Problemstellungen auf diesem Gebiet mit den Mitteln der Informatik. Darüber hinaus sollen die Daten mit dieser Infrastruktur einer interessierten Öffentlichkeit über das Internet zur Verfügung gestellt werden. Diese EDV-Infrastruktur ist eine Internetanwendung und wird **iFlora** genannt.

Wie in allen Teilgebieten der Biologie kommt es auch in der Ökologie zu einer immer umfangreicheren Datenfülle, die eine Entwicklung von neuen Methoden für eine leistungsstarke, computerunterstützte Bearbeitung erfordern (Ökologische Informatik – ecological informatics).

Dazu bedarf es neuer Softwaretechniken und deren Anwendungen auf Problemstellungen in der Ökologie, beziehungsweise ihrem Teilgebiet der Vegetationsökologie. Durch diese Techniken könnte es zu neuen Sichtweisen auf das jeweilige Fachgebiet kommen - und neue Sichtweisen sind die Basis neuer wissenschaftlicher Erkenntnisse (nach SPENGER 2000 und RECKNAGEL 2003).

Die Vegetationsökologie¹ beschäftigt sich grob gesagt mit der Auffindung und der Beschreibung von Vegetationsmustern, der Verknüpfung und kausalen Erklärung dieser Muster durch biotischen und abiotischen Faktoren. Hier spielen z.B. Konkurrenz, als Beispiel eines biotischen Faktors, und das Klima, als Beispiel für einen abiotischen Faktor eine Rolle. Die Basis für das Muster bildet der Artenpool, der sich aus der geographischen Lage ergibt, d.h. es kann sich das Vegetationsmuster nur aus den Arten, die im geographischen Gebiet verfügbar sind, zusammensetzen.

Wesentliches Element der Vegetationsökologie ist die Vegetationsaufnahme. Bei einer Vegetationsaufnahme werden auf ausgewählten Flächen alle Pflanzen aufgenommen, bestimmt und deren Artmächtigkeit bewertet. Diese Bewertung richtet sich meist nach der Bedeckung der Oberfläche (z.B. % der Aufnahmefläche) bzw. nach der Häufigkeit der Pflanze auf der Aufnahmefläche (z.B. Anzahl [selten]).

¹ Lt. Vorlesung „Vegetationsökologie“ von O. Univ.-Prof. Mag. Dr. Georg Grabherr

Oft kommt man bei der Klärung einer wissenschaftlichen Fragestellung mit der Auswertung einer relativ kleinen Anzahl Aufnahmen von einem ausgewählten Pflanzenbestand aus. Oft will man diese Aufnahmen aber mit anderen Aufnahmen, von entfernten Orten oder aus einer anderen Zeit (diese können z.T. jahrhundertealt sein), vergleichen und daraus neue Erkenntnisse gewinnen.

Ein besonderes Problem in diesem Zusammenhang ist die sehr dynamische Änderung der wissenschaftlichen Namen der Pflanzen. Die wissenschaftlichen Namen ändern sich aufgrund von neuen Erkenntnissen über die Verwandtschaftsverhältnisse der Pflanzen zum einem und der Entdeckungen von unbekannten, noch älteren Erstbeschreibungen der Pflanze zum anderem, ständig. So bekommen die Pflanzen regelmäßig neue wissenschaftliche Namen und die alten Namen werden zu Synonymen.

Hinzu kommt, dass man mit den wissenschaftlichen Namen der Pflanze noch nicht weiß, was darunter gemeint ist! Erst in den Florenwerken bekommen diese wissenschaftlichen Namen eine Bedeutung bzw. eine Interpretation². Das heißt in den Florenwerken ist definiert, welche reale Pflanze mit einem wissenschaftlichen Namen gemeint ist. Leider werden auch diese Florenwerke ständig aktualisiert, den neuen Erkenntnissen angepasst und damit verändert. Die Florenwerke verhalten sich dabei fast so dynamisch wie die wissenschaftlichen Namen. Dazu kommt, dass sich Florenwerke nur auf ein bestimmtes Land beziehen und dort gültig sind (z.B. „Exkursionsflora von Österreich“ bezieht sich nur auf das Gebiet von Österreich, Liechtenstein und Südtirol, die „Exkursionsflora von Deutschland“ bezieht sich nur auf das Gebiet von Deutschland usw., die einzige Ausnahme ist die „Flora Europaea“, die europaweit gültig ist). Will man z.B. Aufnahmen, für die Florenwerke der jeweiligen Länder verwendet wurden, europaweit vergleichbar machen, so muss man die Arten der verschiedenen Florenwerke der einzelnen Länder miteinander in Beziehung setzen!

Die besondere Problematik besteht darin, die verschiedenen wissenschaftlichen Namen und ihre verschiedene Bedeutung in ein System zu bringen, damit man auch

² Lt. mündlicher Besprechung mit Dr. Wolfgang WILLNER

z.B. zeitlich sehr unterschiedliche Datenreihen und/oder verschiedene Länder mit Hilfe des Computers vergleichen kann.

Ein Ziel ist die Entwicklung einer Metadatenstruktur (Repository), die diese verschiedenen Bedeutungen (aus den verschiedenen Florenwerken) und die verschiedenen alten botanischen Bezeichnungen (Synonyme) mit den jeweils aktuellen wissenschaftlichen Namen in Beziehung zu setzen und damit vergleichbar zu machen (ADLER ET AL. 1994, S 32 ff). Damit kann man eine Vegetationsaufnahme, dessen Artenliste sich nach dem Florenwerk X hält mit einer anderen Vegetationsaufnahme, dessen Basis für die Artenliste das Florenwerk Y ist, vergleichen! In diesen Metadaten (Metadaten sind Daten über Daten) muss die Information enthalten sein, mit welcher Art aus Florenwerk X die Art aus Florenwerk Y vergleichbar ist! Aus zeitlichen Gründen kann diese Metadatenstruktur in dieser Arbeit nur kurz beschrieben werden.

Bisher wurde nur die Problematik der zeitlichen Integration beschrieben. Die räumliche verhält sich dagegen relativ einfach. Jeder Fundort eines bestimmten Bestandes, der über eine Vegetationsaufnahme erfasst wurde, ist über seine geographischen Koordinaten genau bestimmbar. Mittels GPS ist das heute kein großes Problem mehr. Zusätzlich muss man für einige Spezialfälle auch die genaue absolute Höhe über dem Meeresspiegel bestimmen. Das gehört aber nicht zu dieser Diplomarbeit!

Ein weiteres Problem ist die Integration über die Biologie. Dies könnten z.B. ökophysiologische Eigenschaften sein. So kann man das Muster einer Vegetation durch ökophysiologische Typen beschreiben und modellieren. Wie man diese Ideen in diese Software einbauen könnte, erfordert allerdings noch weitere Überlegungen die nicht mehr Teil dieser Diplomarbeit sind!

Auch Überlegungen, wie man die Daten aus der Fernerkundung in dieses System einbauen und mit den anderen Strukturen vergleichbar machen könnte, sprengen den Umfang dieser Arbeit.

Aus der oben beschriebenen Themenstellung hat sich langsam der Titel der Diplomarbeit entwickelt:

"Computerunterstützte Integration Vegetationsökologischer Daten über
Biologie, Raum und Zeit"

1.1 Problemstellung

Aus dem Beispiel der österreichischen Moordatenbank (enthält vegetationsökologische Daten der Moore Österreichs) soll eine Software für die Verwaltung, Bearbeitung und Auswertung von vegetationsökologischen Daten entwickelt werden, die möglichst viele Eigenschaften integriert, d.h. die Stammdaten des Ortes der Aufnahme wie Ortsname, Klimadaten, geographische Daten usw. und die Aufnahmedaten wie die wissenschaftlichen Namen der Pflanzen und ihre vegetationsökologische Bewertung.

Darüber hinaus sollen diese Daten für die interessierte Öffentlichkeit im Internet zur Verfügung gestellt werden. In einem geschützten Bereich sollen die Mitarbeiter der einzelnen vegetationsökologischen Projekte die Möglichkeit haben, ihre Daten zu bearbeiten und auszuwerten.

Im Moment gibt es für die meisten Projekte eine eigene EDV-Insellösung (meist mit MS Access). Diese Insellösungen sollen mit der Zeit alle in die neue EDV-Lösung übertragen werden. Dadurch wird eine zentrale Datenhaltung erreicht. Das hat einige Vorteile:

- 1.) zentrale Datenhaltung („große“ Relationale Datenbanksystem mit allen Vorteilen)
- 2.) zentrale Tools zur Datenbearbeitung (z.B. Integration des Statistikprogramms „R“ in die Anwendung, usw.)
- 3.) zentrale Datensicherung („Backupsystem“)

Allerdings gibt es durch eine zentrale Datenhaltung auch einige Nachteile:

- 1.) Das System ist abhängig von der Funktionalität eines Netzwerkes.
- 2.) Bei großen Datenmengen kann es zu Performanceproblemen kommen.
- 3.) Gibt es ein Problem mit der Datenbank, der Internetanwendung oder dem Netzwerk, dann stehen alle Projektmitarbeiter.

1.2 Inhalt der Arbeit

Die Arbeit soll die schrittweise Entwicklung von der Aufgabenstellung über die Analyse der Problemstellung, zum Design bis zur fertigen Software zeigen. Diese schrittweise Entwicklung hält sich an die Methoden der Objektorientierten Analyse (**OOA**) und des Objektorientierten Designs (**OOD**).

Die jeweiligen Ergebnisse der Schrittweisen Entwicklung werden in Diagrammen nach den UML2 Standard dargestellt.

Der Umfang des Themas ist für eine einheitliche Bearbeitung zu groß. Deshalb wird es in handhabbare Module aufgeteilt.

Aus den gewünschten Aufgaben und der sich daraus ergebenden Problemstellung kann man die Software in 3 Hauptkomponenten aufteilen.

- a) **Botanische Namen:** In diesem Modul werden die wissenschaftlichen Namen der Pflanzen gespeichert und verwaltet. Im Laufe der Zeit werden diese Namen immer wieder neuen wissenschaftlichen Erkenntnissen angepasst und verändert. Diese Änderungen werden erfasst und die verschiedenen Bezeichnungen miteinander vernetzt. Die wissenschaftlichen Namen sagen noch nichts über die Interpretation der Pflanzen aus. Die Interpretation³ der wissenschaftlichen Namen mit den realen Pflanzen steht in einzelnen Florenwerken. Die Vernetzung der wissenschaftlichen Namen mit der Interpretation dieser Namen in den einzelnen Florenwerken wird auch in dieser Komponente verwaltet. Auch die Vernetzung der wissenschaftlichen Interpretationen mit den Änderungen der wissenschaftlichen Namen müssen in diesem Modul verwaltet werden.
- b) **Aufnahme:** In dieser Komponente werden die einzelnen Aufnahmen verwaltet und mit den einzelnen wissenschaftlichen Namen der Komponente „Botanische Namen“ verbunden.
- c) **Administration:** In dieser Komponente werden die Benutzer für den geschützten Bereich angelegt, verwaltet und die Rechte für diese Benutzer vergeben. Außerdem können hier zusätzliche Felder für die verschiedenen Aufnahmen angelegt werden. Es lassen sich 3 Bereiche unterscheiden:

³ Lt. mündlicher Besprechung mit Dr. Wolfgang WILLNER

- a. **Benutzerverwaltung:** Es werden die Benutzer angelegt und die Datenbankrechte für eine einfache Grundfunktionalität (SELECT – Leserechte auf alle Datentabellen außerhalb der Administration) vergeben.
- b. **Rechteverwaltung:** In diesem Bereich werden die jeweiligen Schreib- und Löschrechte sowohl für die Datengrundstruktur als auch für die benutzerdefinierten (meist projektspezifischen) Datenstrukturen vergeben.
- c. **Datenstrukturverwaltung:** Hier werden neue Datenstrukturen angelegt und verwaltet. Diese neuen, meist projektspezifischen Datenstrukturen werden den einzelnen Projekten zugeordnet und es werden diese Datenstrukturen über Metastrukturen in die Datengrundstruktur eingebunden.

Das Berechtigungskonzept sieht die Vergabe von Rechten bis auf Projekt – Eigentümer – Ebene vor. Es werden die Daten eingeteilt in Produktionsdaten (Daten der verschiedenen Aufnahmen, wissenschaftlichen Namen, usw.) und Administrationsdaten. Dadurch muss das datenbankspezifische Berechtigungssystem erweitert werden. In der Grundfunktionalität hat jeder User Leserecht (SELECT) auf alle Produktionsdaten. Für den geschützten Bereich soll es verschiedene Benutzergruppen geben.

- a) **Projektmitarbeiter**, die nur innerhalb ihres Projektes arbeiten dürfen, Daten eingeben und nur diese Daten bearbeiten dürfen.
- b) **Projektleiter** (dabei handelt es sich um den IT-Projektleiter, der muss nicht mit dem Leiter des Gesamtprojektes übereinstimmen), die innerhalb ihres Projektes alle Daten bearbeiten und diese Daten auch abschließen dürfen. Dieser Abschluss schützt die Daten vor weiterer Veränderung. Die Projektleiter können diese Daten nur dann wieder freigeben, wenn es keine abhängigen Daten in einem anderen Projekt gibt!
- c) **Administratoren**, sind Benutzer, die im Prinzip alles dürfen, was vom Datenbankmanagementsystem (DBMS) erlaubt ist d.h. sie dürfen z.B. keine Daten löschen, von denen andere Daten abhängen (Erhaltung der Datenkonsistenz), usw..

1.2.1 Inhalte der einzelnen Kapitel

Dieses Kapitel gibt eine kurze Beschreibung der Kapitel dieser Arbeit.

- **Einleitung** - Gibt einen kurzen Überblick über die vorliegende Arbeit.
- **Vegetationsökologie** - Soll eine kurze Einführung in die Vegetationsökologie geben, mit Schwerpunkt auf die Probleme, die in dieser Arbeit besprochen werden.
- **Methoden - Definitionen** - Geben einen Überblick über die verwendeten Konzepte, Technologien und Begriffe. Da diese Themen nicht Teil des Biologiestudiums sind, ist dieses Kapitel etwas umfangreicher ausgefallen, da diese Themen die Grundlage der ganzen Arbeit bilden.
- **Konzeption** - Dieses Kapitel beschreibt die Aufgaben der zu erstellenden Software aus Sicht der Problemstellung. Das heißt der Inhalt dieses Kapitels ist nur eine Aufgabenbeschreibung. Es handelt sich noch nicht um die Vorgangweise der technischen Umsetzung.
- **Entwurf** - Dieses Kapitel beschreibt das Konzept der technischen Umsetzung der Problemstellung für eine Software.
- **Umsetzung** - Dieses Kapitel stellt die Software vor, die aus der Aufgabenstellungen des Kapitels Konzeption und aus dem darauf aufbauenden Konzept der technischen Umsetzung, Kapitel Entwurf, entstanden ist. Es werden in diesem Kapitel die einzelnen Formulare beschrieben und es gibt einen Link auf den Sourcecode, der im Anhang enthalten ist. Dieses Kapitel ist die Dokumentation der Software.
- **Literatur** - Eine Zusammenstellung der verwendeten Literatur.
- **Anhang** - hier werden alle Element der Anwendung abgespeichert. Es besteht aus den SQL-Skripten zur Erzeugung der Datenbankstruktur und den Sourcecode-Dateien der Internetanwendung (d.h. der einzelnen Formulare, die im Internet Browser zur Verfügung stehen).

2 Vegetationsökologie

Viele Wissenschaften beruhen darauf, dass in der Natur verschiedene „**Muster**“ erkennbar sind und der Mensch versucht, zuerst die Muster zu ordnen und zu beschreiben und sich danach die Frage stellt, warum diese „**Muster**“ zustande kommen⁴.

In der Vegetationsökologie beschäftigt man sich mit den verschiedenen Mustern der Vegetation. So fällt auf, dass in bestimmten Gebieten bestimmte Pflanzenarten oft vorkommen und in anderen Gebieten völlig andere Arten die Vegetation dominieren. Eine Aufgabe der Vegetationsökologie ist es, diese „**Muster**“ zu erkennen, zu beschreiben und in ein künstliches hierarchisches System zu bringen (Synsystematik). Eine weitere Aufgabe der Vegetationsökologie ist es, gemeinsam mit anderen Bereichen der Biologie, die naturwissenschaftlichen Ursachen dieser Muster zu erklären und anhand bestimmter abiotischer Eigenschaften (Klima, Boden u.a.) eines Gebietes vorhersagen über die Vegetation dieses Gebietes zu treffen.

Der Begriff **Vegetationsökologie** setzt sich aus den Begriffen „**Vegetation**“ und „**Ökologie**“ zusammen.

Die **Vegetationskunde** beschäftigt sich mit der Pflanzendecke, die sich aus Pflanzengesellschaften zusammensetzt. Diese Pflanzengesellschaften können natürlich, naturnah oder mehr oder minder von Menschen beeinflusst sein. Eine Pflanzengesellschaft besteht aus einer Gruppe von ähnlichen Artenkombinationen (Glavac nennt diese Artenkombinationen „Pflanzengemeinschaften“, dieser Begriff wird jedoch wegen der Verwechslungsgefahr mit dem englischen Begriff „plant community“, dessen deutsche Übersetzung auch „Pflanzengesellschaft“ ist, nicht mehr verwendet) und kann auch als eine „Gesamtheit von strukturähnlichen Vegetationsausschnitten“ bezeichnet werden. Die Pflanzengesellschaft ist die Grundeinheit der Vegetationskunde. Eine wichtige Voraussetzung für die vegetationskundliche Arbeit ist die Florenkenntnis. Unter **Flora** eines Gebietes

⁴ Lt. Vorlesung „Grundkonzepte und Paradigmen der Geographie“ von Univ.-Prof. Dr. Peter WEICHHART

versteht man die Gesamtheit der in ihm vorkommenden Pflanzenarten (GLAVAC 1996, S16).

Die **Ökologie** ist die Wissenschaft von den Wechselwirkungen und Wechselbeziehungen der Lebewesen untereinander und zu ihrer unbelebten Umwelt. Der Organisationshöhe entsprechend wird sie folgendermaßen aufgeteilt (GLAVAC 1996, S17):

- **Autökologie** – ist die Wissenschaft von den Beziehungen eines Individuums oder einer Art zu ihrer Umwelt.
- **Demökologie** – ist die Wissenschaft von den Wechselbeziehungen einer Population zu ihrer Umwelt (wird auch Populationsökologie genannt).
- **Synökologie** – ist die Wissenschaft von den Organismengemeinschaften und ihren Beziehungen zur Umwelt.
- **Holökologie** – ist die Wissenschaft, die sich mit den Beziehungen der Biosphäre oder ihren Ausschnitten zu den anderen Sphären (Lithosphären, Hydrosphären, Pedosphären, Atmosphären) beschäftigt.

Die **Vegetationsökologie** ist der Teil der **Synökologie**, der sich mit der Vegetationsdecke und deren Beziehungen zur Umwelt beschäftigt.

3 Methoden - Definitionen

3.1 Ökologische Informatik

Ökologische Informatik (Ökoinformatik, ecological informatics) ist ein interdisziplinäres Konzept für die Anwendung moderner Computertechnik zur Erforschung von Prinzipien der Informationsverarbeitung in Ökosystemen auf verschiedenen Komplexitätsebenen – von den Genen bis hin zu Nahrungsnetzen – und zur Entscheidungsfindung bei der Lösung ökologischer Problemstellungen, z.B. in Zusammenhang mit Artenmannigfaltigkeit (Biodiversität), Nachhaltigkeit und Klimawandel (RECKNAGEL, 2003).

3.2 Verteilung

Viele komplexe Aufgaben lassen sich schneller und besser erledigen, wenn man ihre Durchführung auf speziell dafür entwickelten und optimierten Systemen ausführt. Eine solche Verteilung gibt es auch im Bereich der EDV und diese Verteilung ist die Basis für das Softwareprojekt, das in dieser Arbeit entwickelt wird.

Ein weiterer Vorteil der Verteilung ist die gemeinsame Nutzung von teuren Geräte (z.B. Drucker, Scanner, große Datenspeicher, usw.). Diese werden durch die Mehrfachnutzung effizienter genutzt. Die Funktionalität des Gerätes wird auf viele Nutzer verteilt.

Die Vorteile der Verteilung lassen sich in einen Satz zusammenfassen:
Sie ermöglicht die gemeinsame Nutzung von Ressourcen wie Hardwarekomponenten, Daten, Informationen und Funktionalität (HAMMERSCHALL 2005, S 30).

3.2.1 Verteilte Systeme

In einem „Verteiltem System“ (siehe Abbildung 1) befinden sich verschiedene Geräte und Programme für bestimmte Funktionen auf verschiedenen Computer, die über ein

Netz miteinander verbunden sind und miteinander Daten austauschen können. Dadurch können diese gemeinsamen Geräte bzw. Funktionen von allen anderen Computern genutzt werden. Daraus ergibt sich folgende Definition:

Ein Verteiltes System ist ein System, in dem sich Hardware- und Softwarekomponenten auf vernetzten Computern befinden und miteinander über den Austausch von Nachrichten kommunizieren (HAMMERSCHALL 2005, S 30).

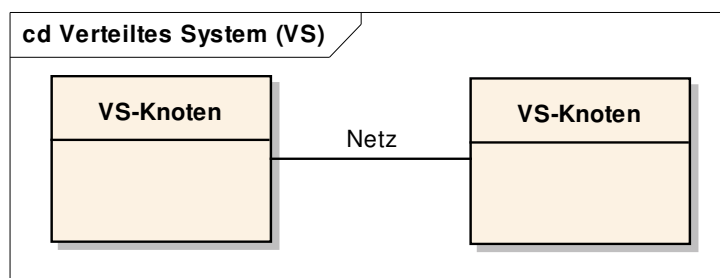


Abbildung 1 – Verteiltes System

Das heute bekannteste Verteilte System ist das Internet. Millionen von Rechnern sind weltweit damit vernetzt. Die Basis, die Vernetzung ermöglicht, ist eine einheitliche Sprache. Diese Sprache sind die Protokolle des Internet Protokollstacks, auch TCP/IP Protokollstack genannt (HAMMERSCHALL 2005, S 16).

Ein Protokoll definiert die Sprache und die Regeln der Kommunikation. Es gibt eine Vielzahl von Protokollen, der TCP/IP Protokollstack wird davon am häufigsten verwendet. Er umfasst eine Reihe von Protokollen, die in ihrer Gesamtheit dafür sorgen, dass Nachrichten zuverlässig zwischen Rechnern übertragen werden. Er ist in gewisser Weise der Kleber, der das Internet zusammenhält (HAMMERSCHALL 2005, S 16).

3.2.2 Verteilte Anwendungen

Bei einer „Verteilten Anwendung“ sind verschiedene Funktionen auf verschiedene Computer in einem Netz verteilt. Ein solches Computernetz nennt man Verteiltes System. Eine Anwendung nutzt diese verschiedenen Funktionen um bestimmte Aufgaben zu erledigen, sie verbindet sie zu einer in sich geschlossen Anwendung, die sie den Anwendern zur Verfügung stellt. Daraus ergibt sich folgende Definition:

Eine Verteilte Anwendung nutzt ein Verteiltes System, um Anwendern eine in sich geschlossene Funktionalität zur Verfügung zu stellen. Sie bietet in irgendeiner Form einen fachlichen Mehrwert (HAMMERSCHALL 2005, S 30).

Kennzeichen einer Verteilten Anwendung (siehe Abbildung 2) ist die Verteilung der Anwendungslogik auf mehrere voneinander weitgehend unabhängigen Anwendungskomponenten. Jede Anwendungskomponente kann auf einem separaten Knoten (Rechner) eines Verteilten Systems liegen (HAMMERSCHALL 2005).

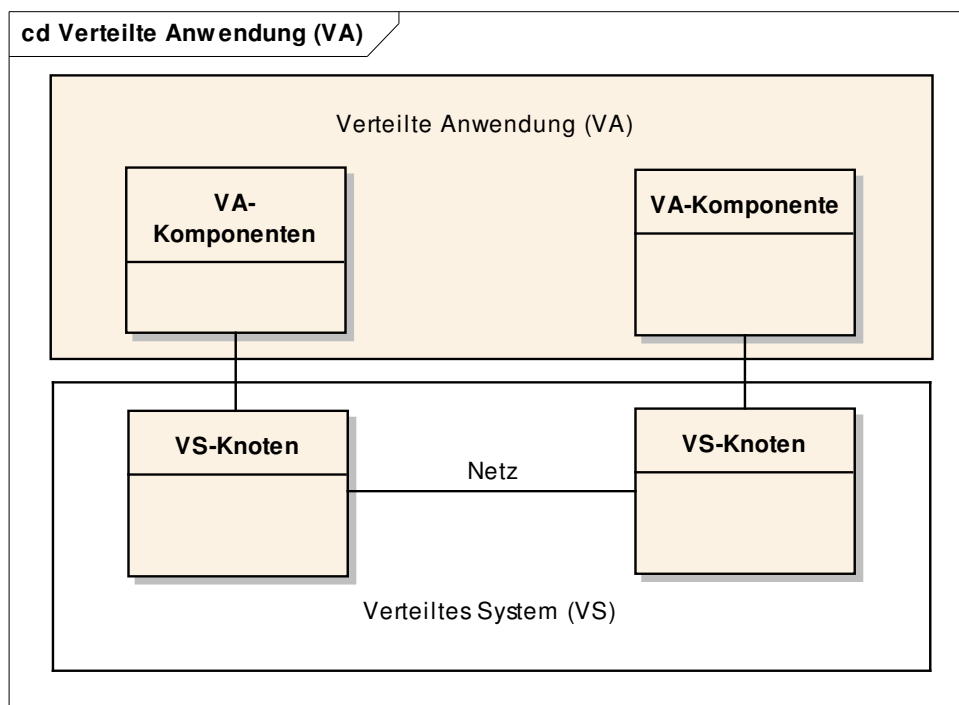


Abbildung 2 – Verteilte Anwendung

Nur in ihrer Gesamtheit erfüllen die Anwendungskomponenten die Aufgaben der Anwendung. Das verteilte System nutzen sie als Kommunikationsinfrastruktur. Da eine Anwendung ohne System nicht existieren kann, wird häufig von Systemen gesprochen, auch wenn letztlich nur die fachliche Anwendung gemeint ist (HAMMERSCHALL 2005, S 17).

3.2.3 Middleware

Verteilte Anwendungen nutzen ein Verteiltes System als Kommunikationsinfrastruktur für ihre Verteilten Komponenten. Das Verteilte System bietet jedoch nur rudimentäre Kommunikationsdienste, wie Verbindungsaufbau und Verbindungsabbau und die Übertragung von Datenpaketen im Byteformat. Einige grundlegende Zusicherungen zu Übertragungssicherheit und Fehlerbehandlung werden ebenfalls vom verteilten System garantiert. Komplexere Aufgaben müssen jedoch auf höherer Ebene bearbeitet werden, in der Anwendung selbst oder auf der Ebene einer zusätzlichen Softwareschicht, der Middleware (HAMMERSCHALL 2005, S 18). Daraus ergibt sich folgende Definition:

Middleware ist eine nichtfachliche Softwarekomponente, die von der Verteilten Anwendung als Programmierschnittstelle zum Verteilten System genutzt wird. Aufgabe der Middleware ist die Abstraktion der Netzwerkprogrammierung sowie das Verbergen von Verteilungsaspekten vor der Anwendung (Transparenz).

Setzt eine Verteilte Anwendung (siehe Abbildung 3) direkt auf einem Verteilten System auf, so spricht man auch von Netzwerkprogrammierung. In diesem Fall implementiert die Anwendung selbst den Zugriff auf den Protokollstack des verteilten Systems (HAMMERSCHALL 2005, S 18f).

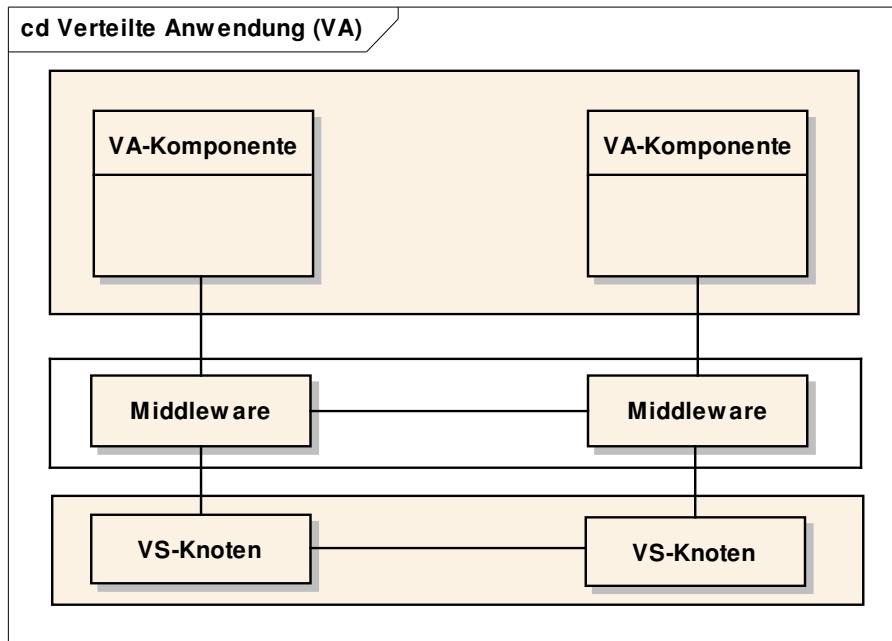


Abbildung 3 – Verteilte Anwendung

Um den Entwickler die mühselige Arbeit der Netzwerkprogrammierung zu ersparen, wird eine zusätzliche Softwareschicht, die Middleware, zwischen dem Verteilten System und der Verteilten Anwendung geschoben. Die Middleware ist eine intelligente Programmierschnittstelle, die auf dem Verteilten System aufsetzt. Ihre Aufgabe ist es, alle Aspekte der Netzwerkprogrammierung so weit wie möglich und sinnvoll vor der verteilten Anwendung zu verbergen sowie die Anwendung selbst geeignet in ihren Abläufen zu unterstützen (HAMMERSCHALL 2005, S 19).

3.2.4 Architekturmodelle

Ein Architekturmodell beschreibt die Rollen der Komponenten innerhalb einer verteilten Anwendung sowie die Beziehungen zwischen ihnen. Die Rolle einer Anwendungskomponente ist im wesentlichen durch die Eigenschaften der Prozesse, in dem sie läuft, festgelegt (HAMMERSCHALL 2005, S 22). Daraus ergibt sich folgende Definition:

Die Komponenten einer Verteilten Anwendung laufen in unabhängigen Prozessräumen. Architekturmodelle beschreiben Interaktionsmuster und Rollen der Prozesse, die als Vorgaben für den Architekturentwurf dienen (HAMMERSCHALL 2005, S 30).

Heute finden vor allem 2 Architekturmodelle ihren Einsatz:

Peer – to – Peer Architekturmodell

Client – Server – Architekturmodell

3.2.4.1 Peer-to-Peer Architektur

Das Peer-to-Peer-Architekturmodell (siehe Abbildung 4) arbeitet mit gleichberechtigten Prozessen. Peer-Prozesse laufen lokal und tauschen nur bei Bedarf Informationen aus (HAMMERSCHALL 2005, S 24).

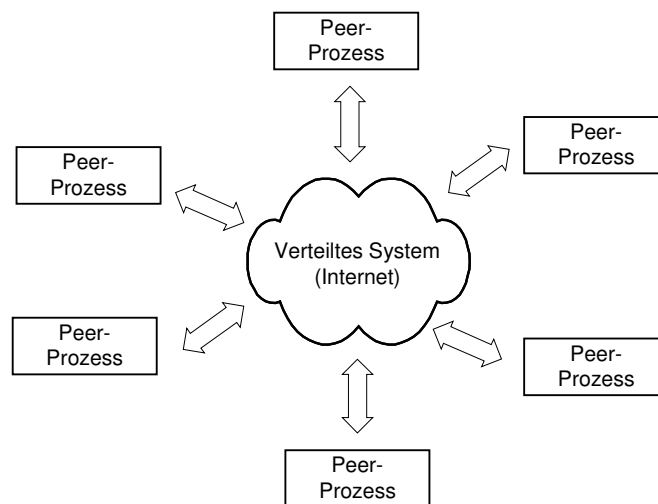


Abbildung 4 - Peer-to-Peer Architektur

Das Peer-to-Peer-Modell hat den Vorteil, dass es keinen zentralen Prozess benötigt. In der Realität treten zum Teil auch Mischformen des Client-Server und Peer-to-Peer-Modells auf.

3.2.4.2 Client/Server Architektur

Verteilte Anwendungen implementieren heute in vielen Fällen das Client-Server Architekturmodell (siehe Abbildung 5). Das Modell ist gekennzeichnet durch einen kurzlebigen Clientprozess, der mit einem langlebigen Serverprozess kommuniziert. Der Clientprozess läuft eine begrenzte Zeit und erfüllt eine festgelegte Aufgabe. Nach Beendigung der Aufgabe wird der Clientprozess beendet. Ein Serverprozess hingegen läuft über einen längeren Zeitraum. Das bedeutet, er wird einmal gestartet und ist aktiv, bis der Rechner abgestellt wird oder der Prozess explizit vom Administrator

beendet wird. Solange der Serverprozess läuft, steht er für Anfragen von einem oder mehreren Clientprozessen zur Verfügung. In der Praxis gibt es mehrere Variationen des Client-Server-Modells wie z.B. die verschiedenen n-Tier Architekturmodelle (HAMMERSCHALL 2005, S 23).

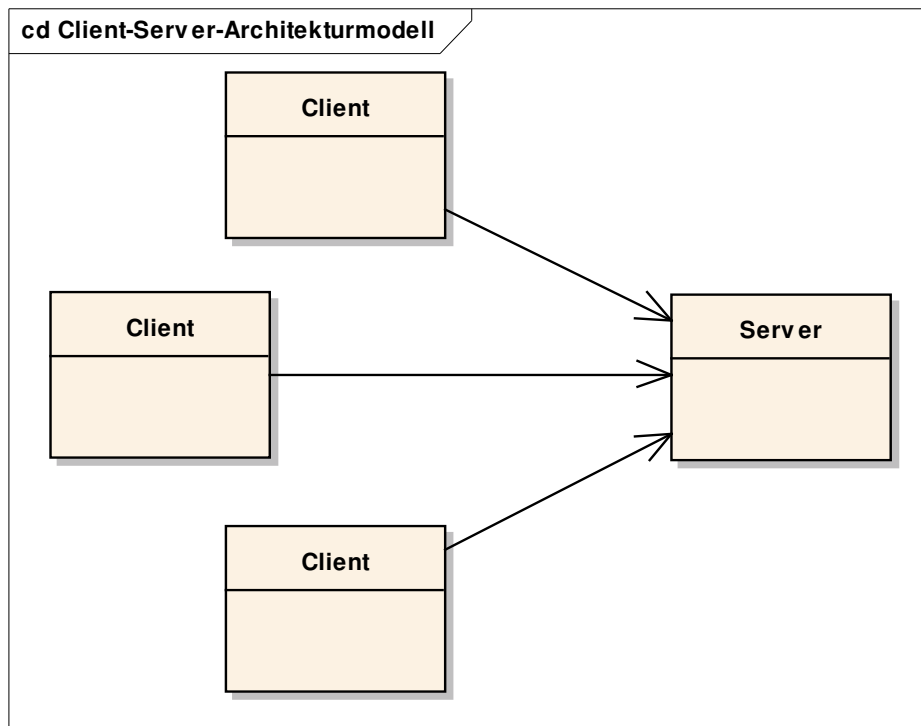


Abbildung 5 – Client - Server - Architekturmodell

3.2.4.2.1 n-Tier Architektur

Der Entwicklung einer Verteilten Anwendung bringt eine Reihe von Herausforderungen mit sich. Eine zentrale, nicht verteilte Anwendung ist sicher und performant. Beim Entwurf der Verteilten Anwendung müssen die Komponenten so auf die verschiedenen Knoten verteilt werden, dass es zu keinen oder zumindest möglichst geringen Verlusten bei der Performance kommt und auch keine Sicherheitsrisiken entstehen (HAMMERSCHALL 2005, S 24). Dazu sind folgende Fragen zu beantworten (HAMMERSCHALL 2005, S 24):

- a) Welche und wie viele Anwendungskomponenten innerhalb der Anwendung wird es geben?
- b) Wie werden die Aufgaben der Anwendung geeignet auf die Anwendungskomponenten verteilt?

c) Wie werden die Anwendungskomponenten auf die Knoten des verteilten Systems verteilt?

n-Tier Architekturen sind eine Ergänzung zum Client – Server Architekturmodell und beschreiben Modelle zur Verteilung einer Anwendung auf den Knoten des verteilten Systems (HAMMERSCHALL 2005, S 26).

Grundlage einer n-Tier Architektur ist die Schicht oder Stufe [engl.: Tier]. Im Client – Server Architekturmodell bezeichnet ein „Tier“ einen eigenständigen Prozessraum innerhalb einer Anwendung. Tier – Grenzen treten generell an Rechnergrenzen auf, sind jedoch auch lokal an Grenzen von Prozessräumen möglich. Es gibt verschiedene n-Tier Architekturtypen, die sich erheblich auf die Eigenschaften einer Anwendung auswirken (HAMMERSCHALL 2005, S 26f). Daraus ergibt sich folgende Zusammenfassung:

n-Tier Architekturen sind eine Verfeinerung des Client-Server-Modells. Sie beschreiben, wie die Komponenten einer Verteilten Anwendung auf Tiers (Schichten bzw. Stufen) im Verteilten System verteilt werden und welche Aufgabe eine Komponente innerhalb einer verteilten Anwendung übernimmt. Für Internetanwendungen verwendet man vor allem 3- und 4-Tier-Architekturen. Der Vorteil dieser Architekturen ist die Möglichkeit zur zentralen und geschützten Verwaltung von Middle-Tier und Server-Tier im Intranet (HAMMERSCHALL 2005, S 30).

3.2.4.2.2 3-Tier Architektur

Bei dieser Architektur werden die Aufgaben einer Verteilten Anwendung auf 3 Tiers (Ebenen) verteilt. Die Präsentationskomponente liegt auf dem Client – Tier, die Anwendungslogik liegt auf dem Middle – Tier (diese Schicht wird auf Grund ihrer Position Middle – Tier bezeichnet und hat mit der Middleware nichts zu tun) und die Datenhaltung auf dem Server – Tier (HAMMERSCHALL 2005, S 26).

Hier eine Zusammenfassung der 3 Tiers (siehe auch Abbildung 6) und ihrer Aufgaben (HAMMERSCHALL 2005, S 26):

- a.) **Client-Tier** => **Präsentationsschicht**: Auf den jeweiligen Benutzerrechner erfolgt nur die Präsentation => es wird HTML-Code passiv im Internet-

Browser angezeigt. Einzige Ausnahme ist das Abprüfen von Dateneingabe auf bestimmte Wertebereiche z.B. kann man eine String-Feld auf 25 Zeichen definieren, dann wird das mittels Javascript überprüft! Das ist der BenutzerPC mit Internet-Browser.

- b.) **Middle-Tier** => **Anwendungslogik** – Schicht: In dieser Schicht ist nach der Theorie die eigentliche Funktionalität der Anwendung implementiert. Hier werden mittels Programmcode nach den Angaben der Benutzer die eigentlichen HTML-Seiten erstellt, die Daten werden vom Server der Datenhaltung geholt und bei Dateneingabe des Users werden die Daten von der Präsentationsschicht gesammelt, noch einmal überprüft und dann die Daten auf der Datenhaltungsschicht geändert. Das ist der Internetserver mit Apache und PHP.
- c.) **Server-Tier** => **Datenhaltungsschicht**: Hier befindet sich die Datenhaltung und die damit verbundenen Funktionalitäten. Das ist ein Server mit einer Datenbank z.B. Oracle10g.

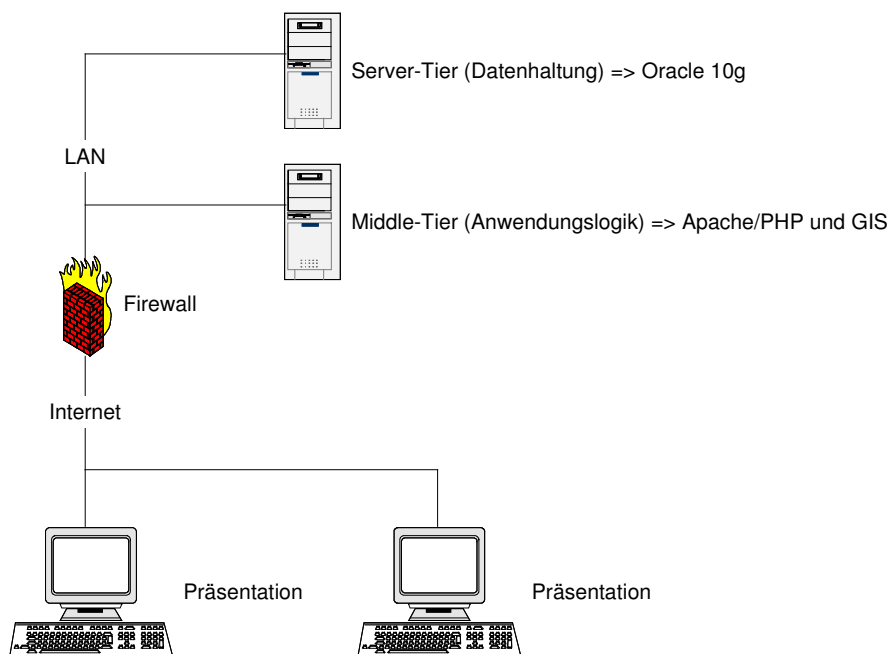


Abbildung 6 - 3 - Tier Architektur

3.3 Datenhaltung

Die in einem Programm erzeugten Daten bleiben nur so lange erhalten, solange auch das Programm läuft. Mit dem Ende des jeweiligen Programms gehen die Daten wieder „verloren“, wenn man nicht besondere Vorkehrungen getroffen hat. Diese Vorkehrungen bestehen darin, dass man vor dem Ende des Programms die Daten, die man später wieder braucht, in einer Datei speichert (siehe Abbildung 7). Der Inhalt dieser Dateien steht so lange dauerhaft (**persistent**) für Verfügung, bis er explizit gelöscht oder überschrieben wird. Diese Dateien befinden sich auf externen Speichermedien (BALZERT 1996, S 670).

Die Inhalte dieser Dateien können nicht direkt vom Programm verarbeitet werden, sondern nur indirekt mittels Funktionen des Betriebssystems. Daher müssen die Inhalte von Programmvariablen in bzw. von Dateien explizit kopiert werden. Der Aufbau der abgespeicherten Daten und die damit verbundene Semantik ist nur im entsprechenden Programm definiert und können daher auch nur von diesem Programm genutzt werden d.h. sie sind für andere Anwendungen nicht zugänglich (BALZERT 1996, S 670).

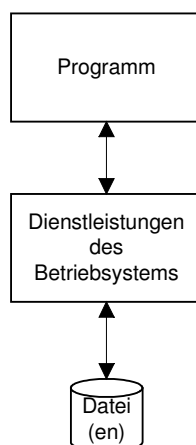


Abbildung 7 - Datenhaltung - Dateien

Es zeigt sich aber, dass oft verschiedene Programme die gleichen Daten brauchen. Daher ist es nicht sinnvoll, dass verschiedene Programme identische Daten für sich verwalten. Dies führt zu Doppelerfassungen von Daten und redundanter

Speicherung. Redundanz⁵ ist die mehrfache Speicherung ein und derselben Information. Nach längerer Zeit erhält man inkonsistente Datenbestände. Die Inkonsistenz⁶ von Daten ist die Widersprüchlichkeit zwischen diesen Daten (z.B. wenn ein Bundesland in verschiedenen Anwendungen abgespeichert wird und durch einen Tippfehler verschieden geschrieben wurde). Ziel ist es daher, die Daten eines Unternehmens oder eines großen Anwendungsbereichs **integriert** zu verwalten. Die integrierte Verwaltung macht es sinnvoll, dass die Datenbeschreibungen unabhängig von den Programmen gespeichert werden und nicht jedes Programm für sich eigene (proprietäre) Datenelemente und Datenstrukturen festlegt. Dies wird in einem Datenbanksystem (DBS) umgesetzt. Im IT-Bereich versteht man unter „proprietär“⁷ IT-Strukturen, die sich nicht an einem allgemein anerkannten Standard halten, also sozusagen „hauseigene“ Entwicklungen sind. Ein Datenbanksystem sorgt für (BALZERT 1996, S 670f):

- a) dauerhafte (persistente),
- b) zuverlässige und
- c) unabhängige Verwaltung sowie die
- d) komfortable,
- e) flexible und
- f) geschützte Verwendung
- g) großer,
- h) integrierter und
- i) mehrfachbenutzbarer Datenbanken.

Ein **Datenbanksystem (DBS)** (siehe Abbildung 8) besteht aus einer oder mehreren Datenbanken (**DB**), einem Data Dictionary (**DD**) und einem Datenbankmanagementsystem (**DBMS**).

⁵ <http://de.wikipedia.org/wiki/Redundanz>

⁶ <http://de.wikipedia.org/wiki/Inkonsistenz>

⁷ http://de.wikipedia.org/wiki/Propriet%C3%A4re_Software

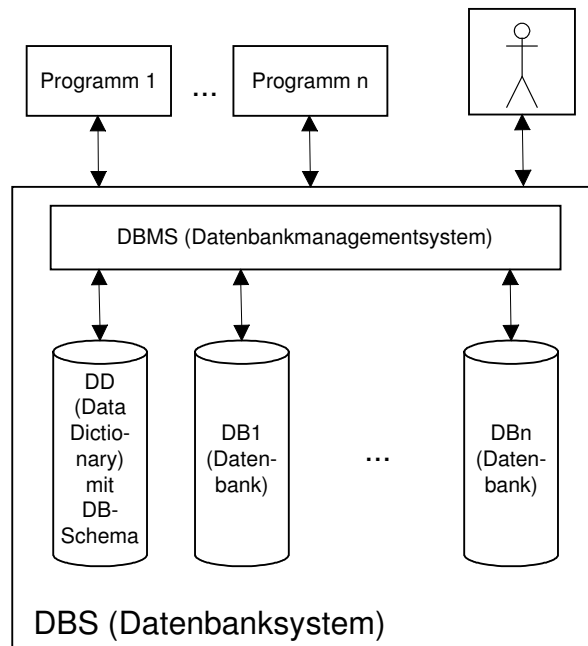


Abbildung 8 - Datenhaltung DBMS

Definitionen (BALZERT 1996, S 671f):

Die **Datenbank (DB)** enthält die Gesamtheit der Daten eines Anwendungsbereichs.

Im **Daten Dictionary (DD)** wird das Datenbankschema gespeichert, das den Aufbau der Daten in der Datenbank beschreibt.

Das **Datenbankmanagementsystem (DBMS)** verwaltet und kontrolliert zentral die Datenbestände der Datenbank(en). Die Informationen über die einzelnen Datenbankschemas und deren Datenbestände erhält das DBMS aus dem DD.

Zuverlässige Verwaltung bedeutet, dass das DBMS über Mechanismen verfügt, um die Konsistenz, die Integrität und die Unversehrtheit der Daten sicherzustellen. Im Fehlerfall muss ein Wiederanlauf (**recovery**) des Datenbanksystems möglich sein.

Unabhängige Verwaltung bedeutet, dass die Programme, die ein Datenbanksystem benutzen, und das Datenbanksystem selbst weitgehend unabhängig voneinander sind. Strukturelle und andere interne Änderungen, die für den jeweils anderen nicht von Bedeutung sind, wirken sich auch nicht auf ihn aus (**Datenunabhängigkeit**). Das heißt, dass die Daten in den Datenbanksystemen einheitlich beschrieben werden können, unabhängig von den jeweiligen Anwendungsprogrammen. Die Beschreibung erfolgt im Datenbankschema.

Komfortable Verwendung bedeutet, dass über eine höhere, abstrakte Schnittstelle mit der Datenbank kommuniziert wird. Beispielsweise muss man sich nicht um die Speicherdetails einzelner Datenelemente kümmern.

Flexible Verwendung bedeutet, dass man mit Hilfe spezieller Abfragesprachen *ad hoc*, d.h. ohne eigentliche Programmierung, auf die Daten zugreifen kann.

Geschützte Verwendung bedeutet, dass Daten vor unberechtigt Zugriff geschützt werden können (**Datenschutz**).

Große Datenbank bedeutet, dass sie nicht vollständig im Arbeitsspeicher (RAM) gehalten werden kann.

Integrierte Datenbank bedeutet, dass alle Daten redundanzarm gespeichert werden, selbst wenn sie von verschiedenen Anwendungen verwendet werden. Das hat zur Folge, dass nicht jedes Anwendungsprogramm alle Daten benötigt, sondern nur bestimmte Ausschnitte. Es muss daher möglich sein, Sichten (**views**) auf Teile der Datenbank zu definieren, die für das jeweilige Anwendungsprogramm relevant sind. Es kommt zu einer zweistufigen Arbeitsweise. Im ersten Schritt werden die benötigten Daten ausgewählt (Aufgabe des DBMS's) und im zweiten Schritt erfolgt die eigentliche Bearbeitung (Aufgabe des Anwendungsprogramms).

Mehrfachbenutzbare Datenbank bedeutet, dass die Daten „gemeinsam“ von mehreren Programmen u. U. sogar gleichzeitig, verwendet werden können. Der parallele Zugriff mehrerer Programme oder Benutzer auf denselben Datenbestand muss daher koordiniert werden.

Damit ein Datenbanksystem (DBS) die beschriebenen Leistungen erbringen kann und damit die Anwendungsprogramme nicht allein für die Semantik der Daten zuständig sind, müssen die Inhalte einer Datenbank beschrieben werden. Im sogenannten **Datenmodell** wird festgelegt (BALZERT 1996, S 672),

- a.) durch welche Eigenschaften Datenelemente charakterisiert werden können,
- b.) wie die Struktur der Datenelemente aussehen kann,
- c.) welche Konsistenzbedingungen einzuhalten sind und
- d.) welche Operationen zum Speichern, Auffinden, Ändern und Löschen von Datenelementen erlaubt sind.

Syntax (behandelt die Muster und Regeln, nach denen Wörter bzw. Konstruktionen zu größeren funktionellen Einheiten zusammengestellt werden) und **Semantik** (behandelt den Sinn und die Bedeutung von Wörtern bzw. von logischen Konstrukten) eines Datenmodells werden durch eine Definitionssprache (DL) und eine Manipulationssprache (ML) festgelegt (BALZERT 1996, S 672).

Das (Datenbank) **Schema** beschreibt eine konkrete Datenbank, d.h. das Datenmodell wird auf einen speziellen Einsatzfall angewandt (BALZERT 1996, S 672).

Das heute in der Praxis am meisten verwendete Datenmodell für Datenbanksysteme ist das **relationale Datenmodell** (BALZERT 1996, S 672).

3.3.1 Relationale Datenbanken

Relationale Datenbanksysteme speichern Daten (inkl. ihrer Beziehungen) in Tabellenform. Die Tabelle ist eine Entitätsmenge. Eine Zeile einer Tabelle (ein Datensatz) repräsentiert eine Entität.

Eine Entität (entity) ist ein individuelles und identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt und wird durch Eigenschaften beschrieben. Für Entitäten wird synonym oft der Begriff Objekt verwendet (BALZERT 1996, S 138).

Die Spalten der Tabelle repräsentieren die Eigenschaften (Attribute) der Entitäten.

Die Definition der Datenstruktur durch Tabellen bezeichnet man als **logisches Schema**. Die formale Definition des logischen Schemas geschieht durch die Datendefinitionssprache (*data definition language*, DDL), die das DBMS zur Verfügung stellt (BALZERT 1996, S 674f).

In einem relationalen Datenbanksystem (siehe Abbildung 9) werden die Datenstrukturen durch drei Schemata beschrieben (BALZERT 1996, S 700):

- a) Das **logische Schema** wird durch die Tabellenstruktur festgelegt.
- b) Die **externen Schemata** beschreiben externe Sichten (*views*) von Benutzern oder Anwendungsprogrammen, basierend auf dem logischen Schema.
- c) Durch ein **internes Schema** wird die physische Organisation der Tabellen auf externen Speichern beschrieben.

Die Definition der Struktur der Daten auf drei verschiedenen Abstraktionsebenen (externe Schemata, logisches Schema, internes Schema) führt zu einer Dreischichten-Architektur eines relationalen Datenbanksystems (BALZERT 1996, S 700).

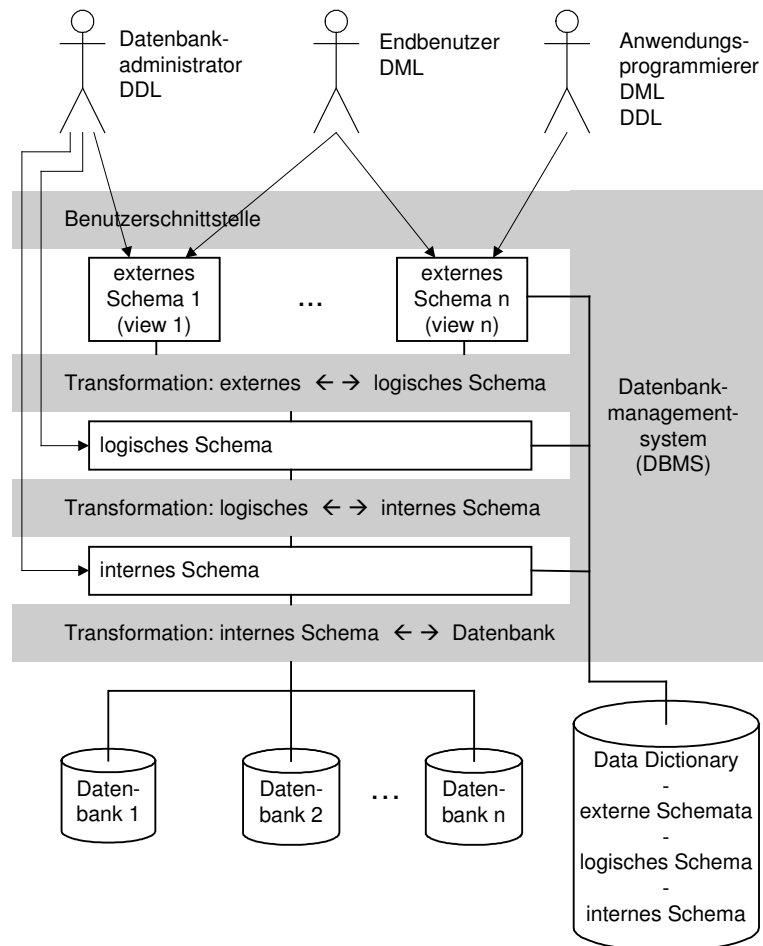


Abbildung 9 - 3 Schichten Architektur RDBMS

Jeder Schicht wird ein eigenes Schema zugeordnet. Die interne Schicht (auch physische Schicht oder Speicherschicht) enthält Informationen, wie Daten gespeichert werden, deren Speicherorganisation und die Zugriffspfade. Die formale Beschreibung erfolgt im **internen Schema**. Die Transformation „internes Schema \leftrightarrow Datenbank“ muss z.B. die Abbildung des logischen Adressraumes in den physischen unter Benutzung von Betriebssystemfunktionen gewährleisten (BALZERT 1996, S 701).

Die logische Gesamtsicht aller Daten in der Datenbank und ihrer Beziehungen untereinander wird im **logischen Schema** beschrieben. Dazu wird ein Datenmodell verwendet, das von der internen Schicht abstrahiert. Es ist frei von Datenstruktur-

oder Zugriffsaspekten d.h. das wird von der internen Schicht abgeschirmt und betrifft die logische Schicht nicht mehr (BALZERT 1996, S 701).

Die externe Schicht (auch Benutzerschicht) umfasst alle individuellen Sichten (views) der einzelnen Benutzer, Benutzergruppen und Anwendungsprogrammen auf die Datenbank. Diese Schichten werden jeweils in eigenen **externen Schemata** beschrieben. Jedes Schema enthält den Ausschnitt der logischen Gesamtsicht, den der Benutzer oder das Anwendungsprogramm sehen darf (BALZERT 1996, S 701).

Bevor man Daten eingeben, bearbeiten, auswerten und verwalten kann, muss die Datenbank eingerichtet werden, d.h. die einzelnen Schemata müssen beschrieben werden. Dafür stellt das DBMS eine Datendefinitionssprache (*data definition language* - **DDL**) zur Verfügung, mit der alle logischen und internen Schemas definiert werden. Die Daten der eingerichteten Datenbank können von den Benutzern eingetragen, geändert und gelöscht werden. Außerdem können die Benutzer Anfragen (*queries*) an die Datenbank stellen. Für diese Aufgaben stellt das DBMS eine Datenmanipulationssprache (*data manipulation language* - **DML**) zur Verfügung. Diese DML kann als eigenständige Dialog-Sprache implementiert sein, so dass der Benutzer direkt im Dialog mit der Datenbank arbeiten kann (BALZERT 1996, S 701).

Als Standard für die DDL und DML hat sich bei relationalen Datenbanken **SQL** (**Structured Query Language**) etabliert.

3.3.2 Relationale Datenbankmanagement Systeme (RDBMS)

Jede (IT) Anwendung besteht aus verschiedenen Systemeigenschaften und Werkzeuge, damit sie ihre Aufgaben erfüllen kann. Dies wird mittels Systemprogrammen und Werkzeugprogrammen (z.B.: SQL) umgesetzt.

Als Zusammenfassung der einleitenden Kapiteln kann gesagt werden, ein Datenbanksystem⁸ (**DBS**) besteht aus 2 Teilen:

- Verwaltungssoftware, genannt Datenbankmanagementsystem (**DBMS**)
- Menge der zu verwaltenden Daten, der eigentlichen Datenbank (**DB**)

Die Aufgabe eines Datenbankmanagement Systems (**DBMS**) ist die Verwaltung und Auswertung der Daten von der eigentlichen Datenbank (**DB**). Handelt es sich dabei

⁸ <http://de.wikipedia.org/wiki/DBMS>

um ein relationales Datenmodell, nennt man das **DBMS** auch „Relationale Datenbank System“ (**RDBMS**).

Zusammenfassend kann man sagen, ein **RDBMS** stellt alle Funktionen zur Verfügung, die man für die Arbeit mit einer Relationalen Datenbank braucht.

3.3.1.1 Oracle

Oracle⁹ ist das am meisten verwendete und erfolgreichste kommerzielle „Relationale Datenbankmanagement System“. Diese Datenbank ist ein vollständiges und ausgereiftes „Relationale Datenbanksystem“, mit dem auch sehr große Datenmengen verwaltet werden können und auf die sehr komplexe Abfragen durchgeführt werden können. Zusätzlich zu den **SQL**-Tools gibt es die Möglichkeit einer prozeduralen Abfragesprache (PL/SQL, d.h. **P**rocedural **L**anguage/**SQL**). PL/SQL ist eine Programmiersprache mit speziell erweiterten Eigenschaften für die Auswertung und Bearbeitung von großen Datenmengen, die in einem relationalen Datenmodell abgespeichert sind. Die Eigenschaften und Möglichkeiten von Oracle zu beschreiben, würden den Rahmen dieser Diplomarbeit sprengen.

Es ist vorgesehen, dass die Produktionsversion, die in dieser Diplomarbeit beschriebenen Software (iFlora), mit einer Oracle Datenbank zusammenarbeitet.

3.3.1.2 MySQL

MySQL¹⁰ ist das erfolgreichste „Relationale Datenbankmanagement System“ im Open Source Bereich. Diese Anwendung hatte über lange Jahre eine sehr eingeschränkte Funktionalität, so waren lange nicht alle Standardbefehle des SQL99 Standards implementiert. Auch eine Foreign Key – Primary Key Beziehung war nur bedingt möglich. Es gab auch lange keine prozedurale Abfragesprache innerhalb von MySQL. Deshalb hat man MySQL lange Zeit nicht als „echte, relationale Datenbank“ angesehen. In den letzten Jahren hat man versucht, diese Nachteile zu beheben. Damit wurde MySQL zu einer echten Alternative für, auch große, Standardanwendungen, d.h. für Anwendungen, die nicht eine komplexe Auswertung brauchen.

⁹ <http://de.wikipedia.org/wiki/Oracle>

¹⁰ <http://de.wikipedia.org/wiki/Mysql>

Ein MySQL Server wurde für die Entwicklung des Prototypen, der in dieser Diplomarbeit beschriebenen Software (iFlora), verwendet.

3.4 Anwendungslogik - Applikationsserver

Die Software, die in dieser Diplomarbeit besprochen wird, besteht aus einem Datenteil, der in einer Relationalen Datenbank realisiert wird und einem Anwendungsteil, der im Applikationsserver angelegt wird. Der Applikationsserver besteht aus einem Apache HTTP Server mit einem PHP Modul (siehe folgende Kapitel).

3.4.1 Apache HTTP Server

Der Apache HTTP Server¹¹ ist ein Webserver. Dieser stellt Funktionen bereit, mit denen man HTML Seiten im Internet zur Verfügung stellen kann. Diese HTML Seiten können statische HTML Seiten sein oder zur Laufzeit dynamisch zusammengestellte HTML Seiten. Ein Webserver stellt Module zur Verfügung, die verschiedene Funktionen realisieren. Es gibt z.B. ein Modul, mit dem man die Kommunikation zwischen den Benutzer, der eine HTML-Seite aufruft und den Apache http Server verschlüsseln kann. Ein weiteres Modul von Apache bietet die Möglichkeit, mittels serverseitiger Skriptsprachen Webseiten dynamisch zu erstellen. Häufig verwendete Skriptsprachen sind PHP, Perl oder Ruby.

Ein Apache HTTP Server mit dem Modul für PHP wurde für die Entwicklung des Prototypen, der in dieser Diplomarbeit beschriebenen Software (iFlora), verwendet und wird für den Betrieb der fertigen Software verwendet werden.

3.4.2 PHP

PHP¹² ist eine Skriptsprache, mit der man auf einen Webserver (HTTP Server) dynamische Webseiten programmieren kann. Sie wird z.B. über ein eigenes Modul in den Apache HTTP Server eingebunden werden (siehe auch Kapitel Apache HTTP Server).

¹¹ http://de.wikipedia.org/wiki/Apache_HTTP_Server

¹² <http://de.wikipedia.org/wiki/Php>

Bei PHP handelt es sich um eine vollständige Programmiersprache, dessen Code nicht kompiliert d.h. in einen maschinenlesbaren Code übersetzt, abgespeichert und bei Bedarf ausgeführt wird, sondern das Programm wird als Skript abgespeichert (d.h. in PHP) und bei jedem Aufruf in maschinenlesbaren Code übersetzt.

Der Code von PHP kann in eine HTML Seite integriert oder als eigenständige Datei abgespeichert werden. Bei beiden Versionen müssen bei den Standardeinstellungen des Apache HTTP Servers die Dateierweiterungen der Dateien mit PHP-Code von „html“ in „php“ geändert werden! Der jeweilige Webserver erkennt dann, dass er vor der Weiterleitung der Webseite den PHP Code ausführen muss, d.h. der Webserver übergibt den Skriptcode seinem PHP Modul. Dieses PHP Modul verwandelt den Skriptcode, ähnlich wie bei Java, in einen Bytecode, der dann über eine einfache Übersetzung (in einen maschinenlesbaren Code) vom jeweiligen Prozessor des Servers ausgeführt werden kann. Mit dieser Methode wird eine schnelle Abarbeitung des Skriptcodes erreicht.

3.4.3 Javascript

Javascript¹³ ist eine Skriptsprache, mit der man im Webbrowser des jeweiligen Benutzers eine HTML-Seite dynamisieren kann. Das heißt, im Gegensatz zu PHP, wo der Code beim Aufrufen der HTML-Seite im Webserver (z.B.: „Apache HTTP Server“) ausgeführt wird, wird der Code von Javascript beim Aufruf der HTML-Seite erst im Webbrowser beim jeweiligen Benutzer (z.B.: „Microsoft Internet Explorer“) ausgeführt. Die Sprache Javascript ähnelt Java, daher kommt auch der Name. In der Anwendung iFlora wird z.B. der Menübaum mittels Javascript realisiert (siehe Abbildung 10).

¹³ <http://de.wikipedia.org/wiki/JavaScript>

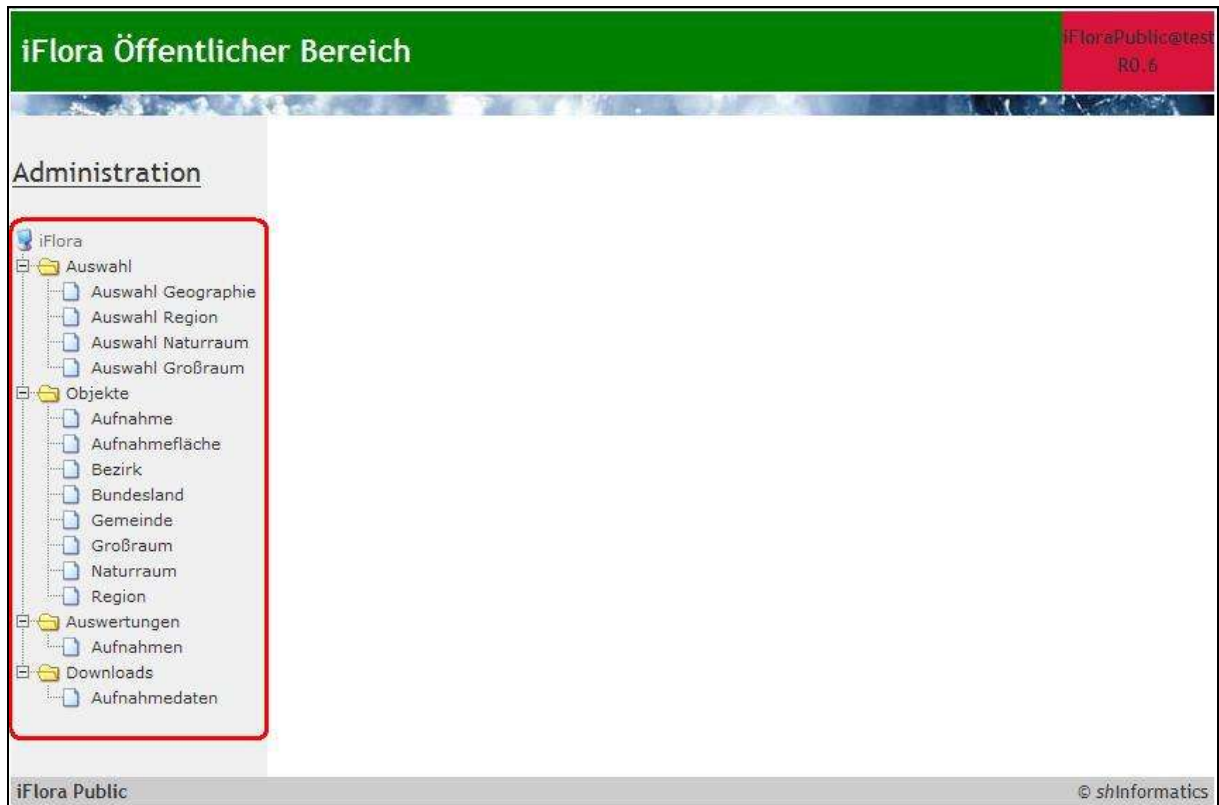


Abbildung 10 – iFlora Menü

3.4.4 XML

Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungssprache“), abgekürzt XML¹⁴, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form einer Textdateien. XML ist das typische Format für den Datenaustausch mittels Webservice (siehe nächstes Kapitel).

Eine Auszeichnungssprache¹⁵ (engl. „Markup Language“, abgekürzt ML) dient zur Beschreibung der Daten und teilweise des Verfahrens, das zur Bearbeitung dieser Daten nötig ist. Es werden die Eigenschaften, Zugehörigkeiten und Verfahren von bestimmten Wörtern, Sätzen und Abschnitten (Elementen) eines Textes oder einer Datenmenge beschrieben bzw. zugeteilt, meist indem sie mit Tags (z.B.: „<element_1>“) markiert werden. Wie eine Programmiersprache besitzen auch alle Auszeichnungssprachen eine Syntax (formalen Beziehungen zwischen Zeichen), eine (formale) Grammatik (systematischen Sprachbeschreibung) und eine Semantik

¹⁴ <http://de.wikipedia.org/wiki/XML>

¹⁵ <http://de.wikipedia.org/wiki/Auszeichnungssprache>

(Bedeutungslehre). Allerdings unterscheidet sich eine Programmiersprache von einer reinen Auszeichnungssprache durch die Elemente von Iteration (Wiederholungen) und Auswahl (abhängig von einem Zustand wird eine Auswahl getroffen).

3.4.5 Webservices

Ein „Webservice“¹⁶ ist eine Softwareanwendung, die über das Internet Funktionen für andere Softwareanwendungen bereit stellt. Die Funktionen eines Webservice können von anderen Anwendungen (egal ob Internetanwendung oder Clientprogramm wie z.B.: Microsoft Word) aufgerufen und genutzt werden (solange ihnen eine Internetverbindung zur Verfügung steht). Das heißt, im Code einer Anwendung kann eine Funktion eines Webservice wie jede andere Funktion aufgerufen werden. Der Datenaustausch wird über XML realisiert. So können Funktionen eines Webservers (z.B.: „Apache HTTP Server“) von Programmen anderer Webserver genutzt und in eigene Anwendungen (HTML-Seiten) eingebunden werden.

3.5 Analyse und Entwurf von Software

»If you wait for a complete and perfect concept to germinate in your mind, you are likely to wait forever.« DeMarco

Es gibt eine Aufgabe, die mit einem EDV Programm erledigt werden soll. Am Anfang beschreibt man die fachliche Aufgabe. Aufgrund dieser Beschreibung analysierte man z.B. einen typischen Datenfluss dieser Aufgabe und beschreibt ihn anhand einer Grafik. Damit jeder unter einem Begriff und einem Diagramm das gleiche versteht, verwendet man ein standardisiertes Diagramm. Dieses standardisierte Diagramm kann jeder, nach dem er sich in den Standard eingearbeitet hat, eindeutig verstehen und interpretieren. In den nächsten Schritten teilt man komplexere Aufgaben in kleine, überschaubare Einheiten, Module genannt. Diese Module beschreibt man dann von den verschiedensten Seiten unter den verschiedensten Aspekten. Der verwendete Standard beschreibt eine Reihe von verschiedenen Möglichkeiten, unter welchen Seiten und Aspekten man eine Aufgabe fachliche untersuchen bzw.

¹⁶ <http://de.wikipedia.org/wiki/Webservice>

analysieren und beschreiben. Er stellt dafür standardisierte Verfahren und, als Beschreibungssprache, standardisierte Diagramme zur Verfügung. Abhängig von der gestellten Aufgabe entscheidet der IT-Analyst (derjenige, der die fachliche **ANALYSE** durchführt) welche Aspekte für die Beschreibung der fachlichen Aufgabe notwendig sind und welche Verfahren er vom Standard verwendet. Das oben beschriebene Verfahren wird „Objektorientierte Analyse“ (**OOA**) genannt. Die dahinterliegenden Standards der Verfahren und Diagramme werden in der **Unified Modeling Language (UML)** definiert (BALZERT 1999, S 17ff):

- **Klassendiagramm** (class diagram)
- **Geschäftsprozessdiagramm** (use case diagram)
- **Aktivitätsdiagramm** (activity diagram)
- **Sequenzdiagramm** (sequence diagram)
- **Zustandsdiagramm** (statechart diagram)

Die Ergebnisse der Objektorientierten Analyse (**OOA**) sind eine möglichst vollständige fachliche Beschreibung (bzw. das Modell) einer Aufgabe mit standardisierten Verfahren und Werkzeugen (diese Werkzeuge erzeugen die standardisierten Diagramme). Das heißt, Ziel der Objektorientierten Analyse (**OOA**) ist es, ein konsistentes, vollständiges und eindeutiges Modell (BALZERT 1999, S 8) der gewünschten (fachlichen) Aufgaben zu erstellen. Es soll die essentielle Struktur und Semantik der Aufgabe, aber noch keine technische Lösung beschreiben (BALZERT 1999, S 9). Dieses Modell soll (da die Beschreibungen im Modell manchmal etwas abstrakt sind, ist das manchmal schwierig!) die Aufgaben sowohl für die jeweiligen Projektmitarbeiter verständlich darstellen, damit diese das Ergebnis auf mögliche Fehler überprüfen können, als auch den Informatiker bzw. Analysten als Basis für die weitere Vorgangsweise dienen, der kein Fachwissen von dem dahinterliegenden Projekt hat.

Auf Basis dieser Beschreibung (dem Ergebnis der Objektorientierten Analyse (**OOA**)), erfolgt das „Objektorientierte Design“ (**OOD**). Bei diesem Prozess wird die standardisierte fachliche Beschreibung einer Aufgabe in ein technisch umsetzbares Konzept (**ENTWURF**) umgewandelt, d.h. in ein Konzept, das dem Programmierer genau sagt, welche informationstechnischen Strukturen er für die Lösung dieser Aufgaben entwickeln muss. Für diese Aufgabe gibt es wieder standardisierten

Verfahren und Diagramme, die oft den gleichen Namen wie in der Objektorientierten Analyse haben, aber verschieden verwendet werden. Der Standard, der die Methoden des Objektorientierten Designs (**OOD**) beschreibt ist die bereits oben erwähnte **Unified Modeling Language (UML)**. Die **UML** ist eine standardisierte visuelle Spezifikationssprache für die Objektorientierte Modellierung¹⁷. Es stellt eine Reihe von Verfahren und Diagrammen zur Umsetzung von fachlichen Problemstellungen in ein technisch umsetzbares Konzept bereit:

- **Klassendiagramm** (class diagram)
- **Geschäftsprozessdiagramm** (use case diagram)
- **Aktivitätsdiagramm** (activity diagram)
- **Sequenzdiagramm** (sequence diagram)
- **Zustandsdiagramm** (statechart diagram)

Beim **OOD** werden die gleichen Diagramme wie bei der **OOA** verwendet, es unterscheiden sich nur die Inhalte.

Ziel des Objektorientierten Designs (**OOD**) ist die technische Lösung der Aufgaben, ist die Beschreibung der technischen Umsetzung des Modells aus der Objektorientierten Analyse (**OOA**).

Kurz: Die Analyse einer Aufgabenstellung und deren Umsetzung in ein anwendbares Programm (Software) verfolgt in einem 2 Stufenprozess und wird nach der **Unified Modeling Language (UML)** beschrieben:

- Fachliche Analyse der Aufgabenstellung gemeinsam mit den Fachleuten – Ergebnisse sind die standardisierten Diagramme und Beschreibungen der **Objektorientierten Analyse (OOA)**.
- Umsetzung der Ergebnisse der fachlichen Analyse in ein technisch umsetzbares Konzept (ENTWURF) – die Ergebnisse sind die standardisierten Diagramme und Beschreibungen des **Objektorientierten Designs (OOD)**.

¹⁷ http://en.wikipedia.org/wiki/Unified_Modeling_Language

4 Konzeption

Eine Konzeption¹⁸ ist eine umfassende Zusammenstellung der Ziele und daraus abgeleiteten Maßnahmen zur Umsetzung eines größeren Vorhabens. Das Vorhaben dieser Arbeit ist die Erstellung der Internetanwendung iFlora. Dieses Kapitel beinhaltet die dazu notwendigen Informationen und Begründungszusammenhänge.

4.1 Allgemeine Systemübersicht

Angestrebt wird eine Internetlösung mit zentraler Datenhaltung (siehe Abbildung 11):

- Die Anwendung (Business Logik) befindet sich auf einem Internetserver (Applikationsserver).
- Diese Anwendung greift auf die Daten zu, die auf einem Datenbankserver liegen.
- Es gibt verschiedene Anwendergruppen (Gruppen von Benutzern mit ähnlichen Rechten), die alle über das Internet auf die Daten zugreifen können.
 - Administrator – verwaltet die gesamte Anwendung, d.h. er legt neue Benutzer an und verwaltet diese, sorgt für eine zentrale Datensicherung, optimiert die Performance, usw.
 - Projektleiter in den Fachabteilungen – diese legen neue Projekte an, verwalten diese, schützen die Daten gegen irrtümliche Änderungen durch Setzen eines Status und arbeiten auch innerhalb der Projekte mit.
 - Projektmitarbeiter – bearbeiten die Projektdaten.
 - Öffentlicher Benutzer – dieser kann auf die veröffentlichten Daten im Internet zugreifen.

¹⁸ <http://de.wikipedia.org/wiki/Konzeption>

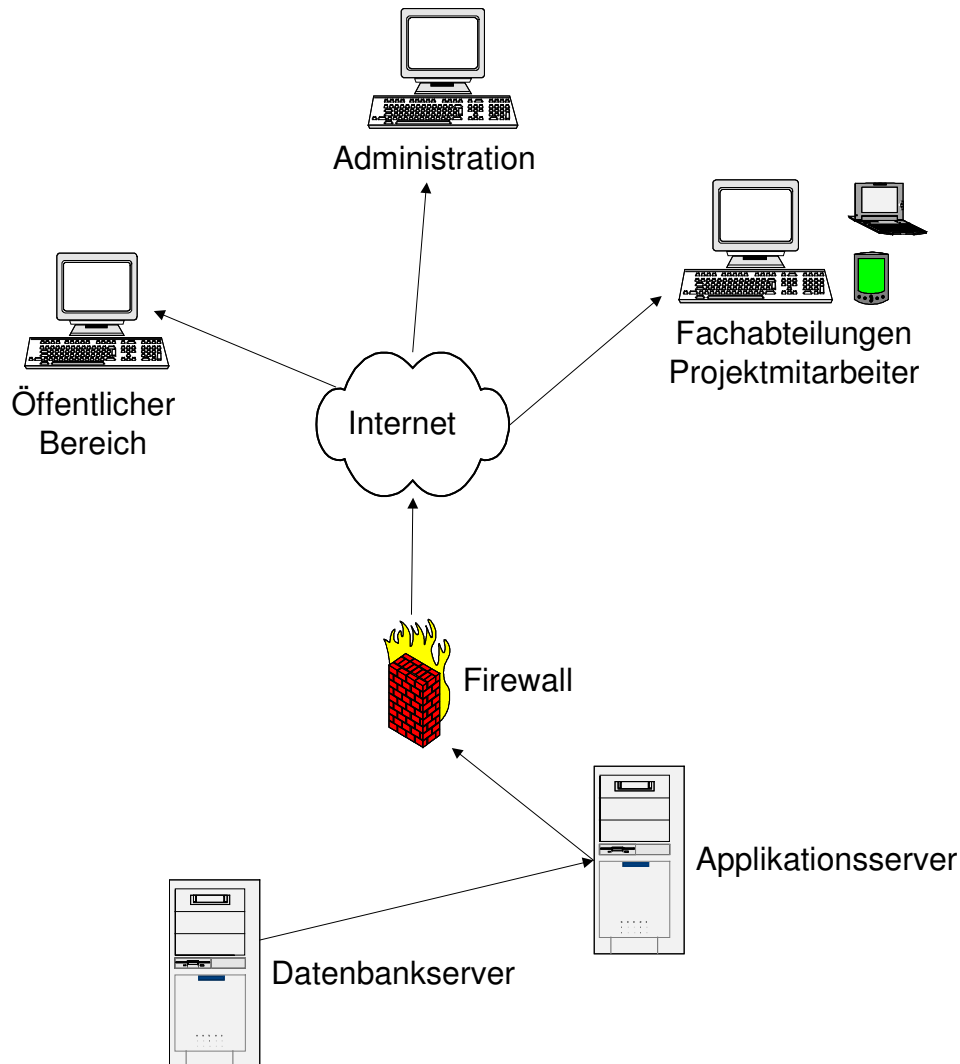


Abbildung 11 – allgemeine Systemübersicht iFlora

4.2 Beschreibung der Ausgangssituation (IST – Analyse)

Derzeit ist (fast) jedes Projekt im Department eine Insellösung (siehe Abbildung 12). Jede Projektgruppe legt ihre eigenen Projekte an. Meist handelt es sich um eine MS Excel oder MS Access Lösung. Es gibt keine zentrale Datenhaltung, Datensicherung und keine zentrale Optimierung der Performance.

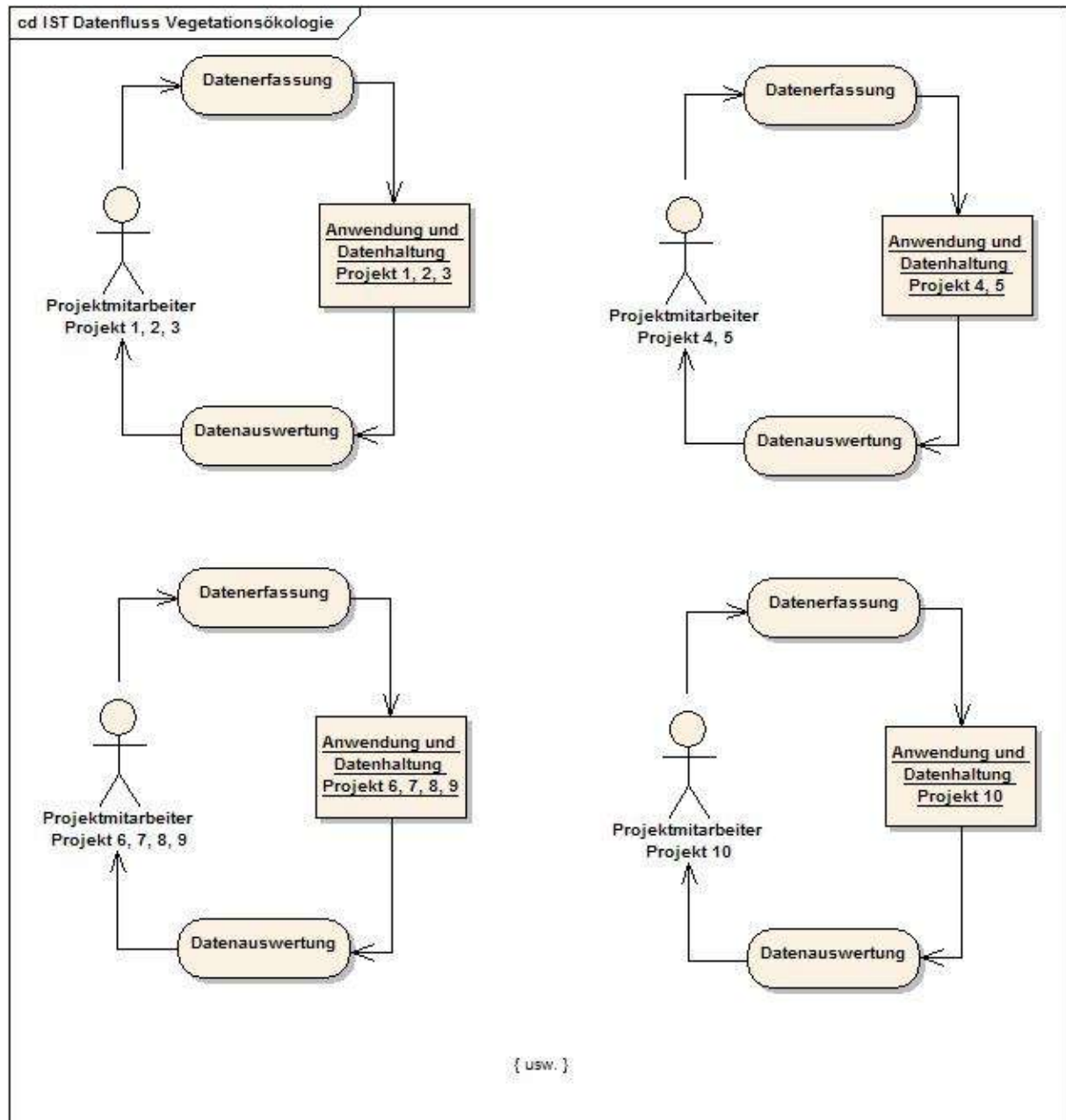


Abbildung 12 – IST Datenfluss Vegetationsökologie

4.2.1 Organisation, Arbeitsabläufe

Jede Projektgruppe entwickelt für ihre Projekte eigene Anwendungen (meist MS Access) oder legt die Daten ihres Projektes nur in Form von Excel Tabellen ab. Die Datenhaltung erfolgt getrennt in ihrem jeweiligen Verzeichnis am Abteilungsserver (durch die zentral Sicherung dieser Verzeichnisse ist eine minimale Datensicherung gewährleistet).

4.3 Beschreibung der gewünschten Situation (SOLL – Analyse)

Meist ist die Projektstruktur bei vegetationsökologischen Projekten sehr ähnlich. Es werden in den Projekten Vegetationsaufnahmen durchgeführt und diese Daten werden abgespeichert. Zu einem späteren Zeitpunkt werden die Daten der verschiedenen Vegetationsaufnahmen ausgewertet. Dies wird meist mit einem Statistikprogramm durchgeführt (SPSS, TWINSpan (**T**wo-way **I**ndicator **S**pecies **A**nalysis), CANOCO, S-PLUS, R: Modul vegan, usw.).

Es soll für alle diese Projekte eine gemeinsame Anwendung geschaffen werden um die (meisten) Insellösungen zu ersetzen (siehe Abbildung 13). Diese Anwendung bekommt den Namen „**iFlora**“. Die Anwendung soll so generisch sein, dass damit verschiedene vegetationsökologische Aufgaben durchgeführt werden können. Für den Erfolg der Anwendung ist es unerlässlich, dass es eine Schnittstelle zu den oben erwähnten Statistikprogrammen gibt (Datenexport als Datei im ASCII- und Excelformat).

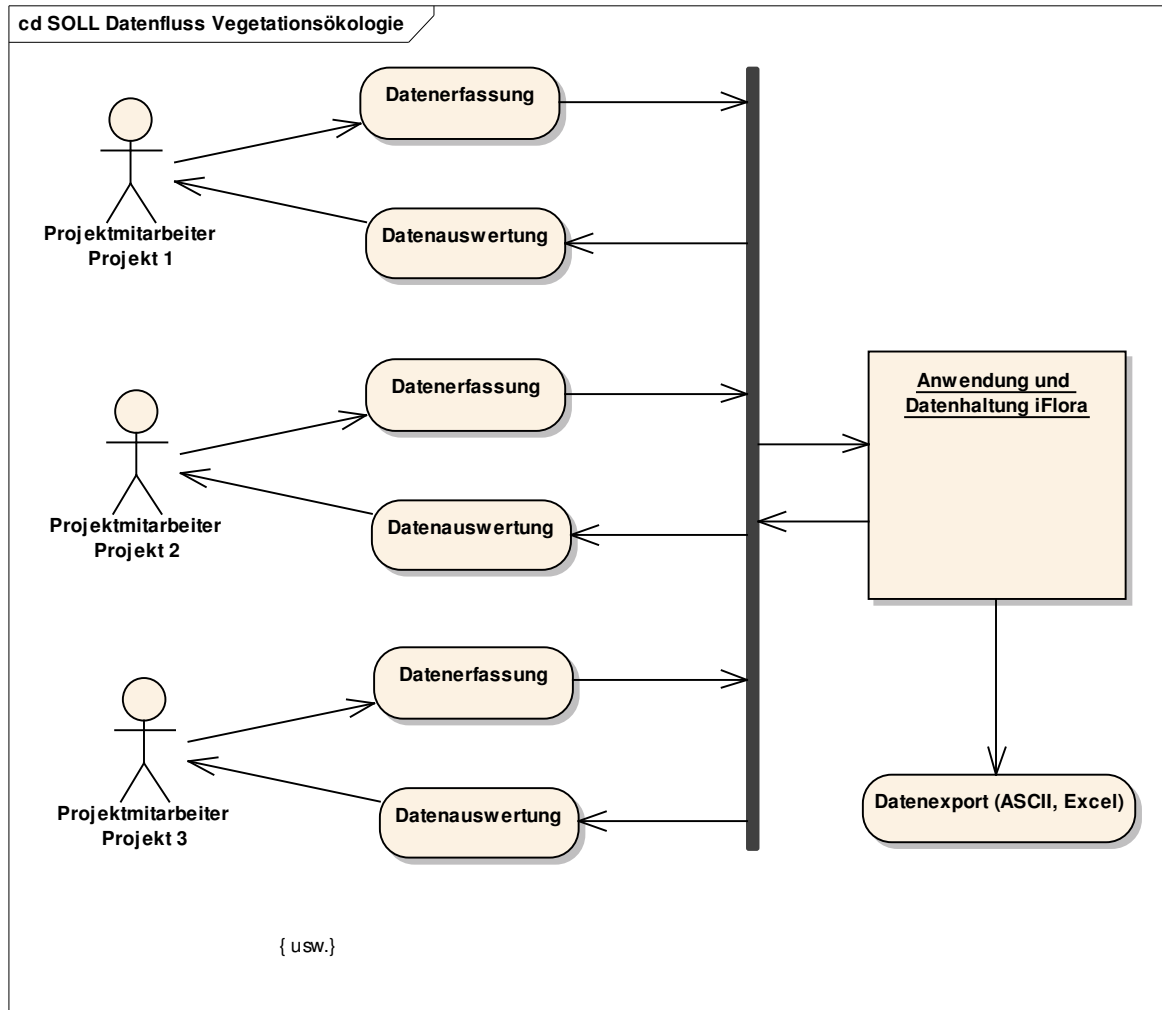


Abbildung 13 – SOLL Datenfluss Vegetationsökologie

4.3.1 Organisation, Arbeitsabläufe

Im Idealfall konfiguriert sich jede Projektgruppe (d.h. der Projektleiter) ihr Projekt in der Anwendung **iFlora**. Die Datenhaltung erfolgt in den zentralen Tabellen der Datenbank (Oracle) von **iFlora**. Alle Formulare, Auswertungen und Exportfunktionen werden für alle Projekte gemeinsam entwickelt und angeboten. Durch die zentrale Datenhaltung kann eine zentrale Datensicherung durchgeführt werden. Jede Projektgruppe kann über die Exportfunktionen noch zusätzlich eine lokale Datensicherung (BACKUP) durchführen. Auch die Daten der Anwendung (Skripten mit dem Quellcode der Business Logik und den verschiedenen Formulare und Funktionen) können zentral gesichert werden. Es ist allerdings **NICHT** vorgesehen, dass die einzelnen Projektgruppen eine Sicherung der Anwendungslogik lokal durchführen können.

4.3.1.1 Grobe Ablaufarchitektur

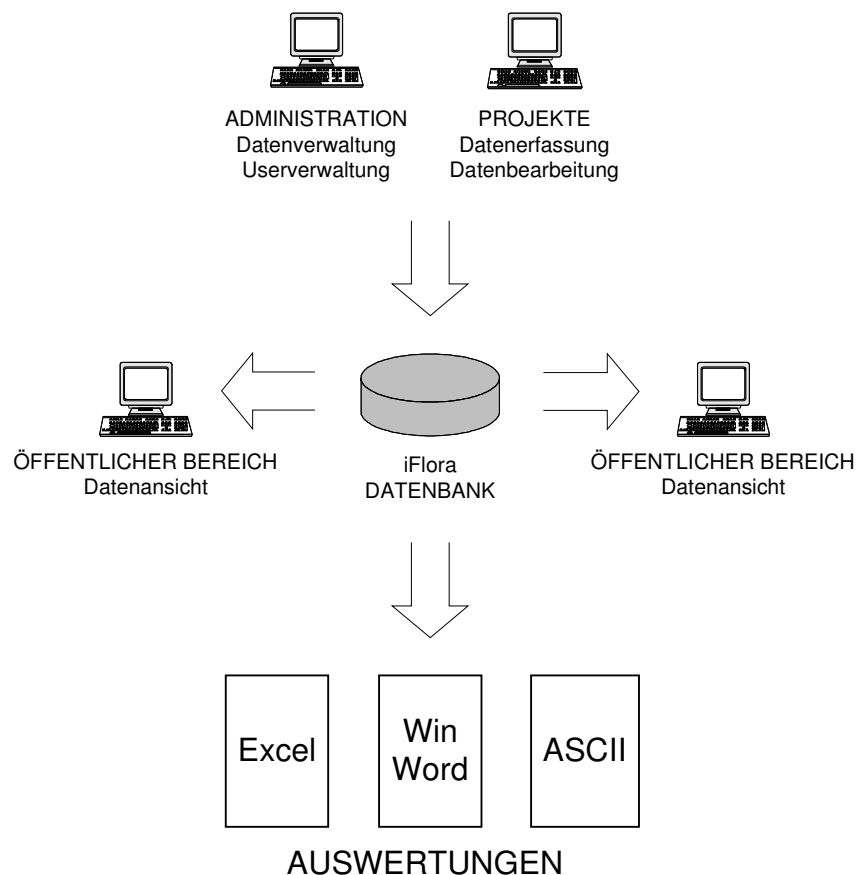


Abbildung 14 - Grobe Ablaufarchitektur

In diesem Projekt werden die wissenschaftlichen Daten über PROJEKTE erfasst und bearbeiten. Die ADMINISTRATION verwaltet Strukturdaten, Metadaten und Userdaten. Im ÖFFENTLICHEN BEREICH können alle (freigegebenen) Projektdaten von allen Internetusern eingesehen werden. Projektmitarbeiter und, wenn auch eingeschränkt, Internetuser können die vorhandenen wissenschaftlichen Daten AUSWERTEN (siehe Abbildung 14).

4.3.2 Kernfunktionen

Bevor man eine Anwendung und ihre Datenstruktur entwickeln kann, muss man den Bedarf ermitteln. Eine Möglichkeit, den Bedarf zu ermitteln ist, sich am Anfang einen typischen Datenfluss anzuschauen. Anhand der Analyse des Datenflusses teilt man die Aufgaben der Anwendung in kleine überschaubare Einheiten, Module genannt. Diese Module können dann näher beschrieben werden. Manchmal sind diese Module

noch immer zu groß und man muss sie in noch kleinere Einheiten auftrennen. Die Aufgaben dieser Module werden dann in der Objektorientierten Analyse (**OOA**) mit den verschiedensten Methoden, wie, Klassendiagramm, Geschäftsprozessdiagramm, Aktivitätsdiagramm, Sequenzdiagramm beschrieben. Die Ergebnisse der **OOA** sind die fachlichen Beschreibungen der Aufgaben mit standardisierten Werkzeugen. Diese Werkzeuge erzeugen standardisierte Diagramme, welche die benötigten Aufgaben sowohl für die jeweiligen Projektmitarbeiter verständlich darstellen (sollen) als auch für den Informatiker, der kein Fachwissen von dem dahinterliegenden Projekt hat. Auf Basis dieser Beschreibung, die gemeinsam mit den zukünftigen Anwendern erarbeitet wurde, erfolgt der Entwurf der Anwendung, auch Objektorientiertes Design (**OOD**) genannt. Bei diesem Prozess wird die standardisierte fachliche Beschreibung einer Aufgabe in ein technisch umsetzbares Konzept umgewandelt. Das heißt in ein Konzept, das dem Programmierer genau sagt, welche informationstechnischen Strukturen er für die Lösung der gestellten Aufgaben entwickeln muss und wie diese Strukturen zusammenarbeiten sollen.

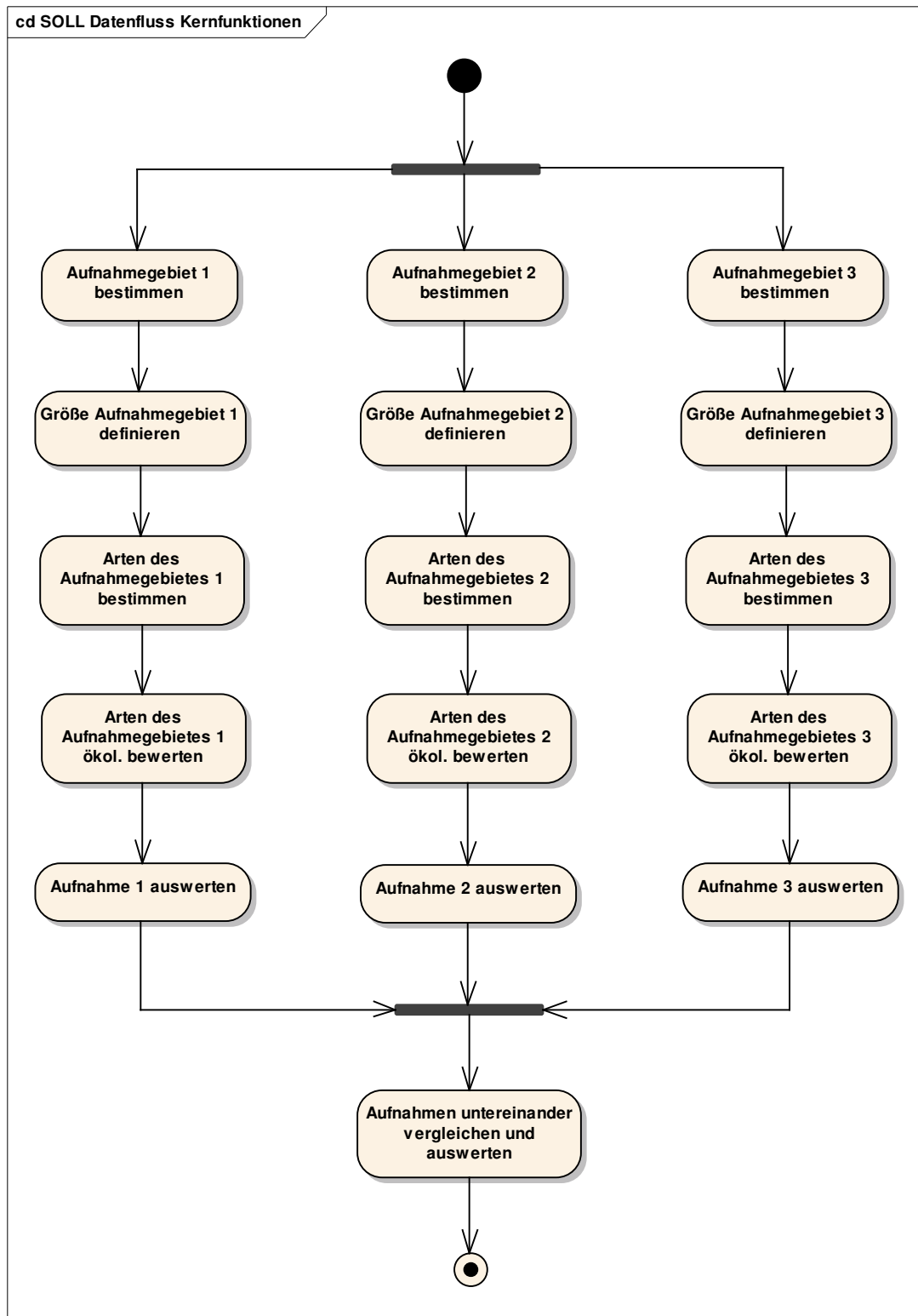


Abbildung 15 – SOLL Datenfluss Kernfunktionen

Diese Anwendung soll die Daten von Vegetationsaufnahmen von verschiedenen Aufnahmeflächen (und Projekten) erfassen, speichern und bearbeiten (siehe Abbildung 15). Sie soll ein Werkzeug sein für die Auswertung und den Vergleich der

verschiedenen Aufnahmen untereinander, d.h. von Aufnahmen verschiedener Projekte, die von verschiedenen Aufnahmeflächen kommen und zu verschiedenen Zeitpunkten und mit verschiedenen Artensystemen aufgenommen wurden.

Eine Vegetationsaufnahme besteht aus einer Liste aller Arten, die man auf einer definierten Fläche findet und der Artmächtigkeit der einzelnen Arten dieser Artenliste. Zusätzlich wird noch die jeweilige Fläche der Aufnahme näher beschrieben (GPS-Koordinaten, Inklination (Hangneigung), Exposition (Hangausrichtung), usw.). Diese Daten werden im weiteren Verlauf Stammdaten genannt. Die Fläche der Aufnahme wird auch Aufnahmefläche genannt.

Mit den Vegetationsaufnahmen wird der Pflanzenbestand eines Gebietes untersucht. Dazu kann eine Aufnahme mit einer großen Fläche des Pflanzenbestandes verwendet werden oder mehrere Aufnahmen mit jeweils einer kleinen Fläche. Statistisch gesehen ist die letzte Variante besser.

Da die verschiedenen Aufnahmen untereinander verglichen werden sollen, empfiehlt es sich, die Arten aus einer zentralen Artenliste zu nehmen, d.h. alle Aufnahmen eines bestimmten Zeitraumes (oder auch eines bestimmten Zeitraumes innerhalb eines bestimmten Projektes) verwenden die Arten einer standardisierten Artenliste (die auf einem bestimmten Florenwerk basiert).

Um die Aufnahmen von verschiedenen Zeitpunkten und evt. aus verschiedenen Projekten miteinander zu vergleichen, müssen die verschiedenen Standardartenlisten miteinander in Beziehung gesetzt werden, d.h. es muss definiert werden, wie die einzelnen Arten der verschiedenen Standardartenlisten miteinander zusammenhängen, oder einfach gesagt, die Art X aus der Standardliste 1 ist mit der Art Y aus der Standardliste 2 identisch.

Daraus ergeben sich die 2 Fachmodule:

- **Vegetationsaufnahme** – kurz: **AUFNAHME** [Vorsilbe: **AFN**]
- **Verwaltung der Arten** – kurz: **BOTANISCHE NAMEN** [Vorsilbe: **BOT**]

Dazu kommen noch allgemeine Verwaltungsaufgaben (Benutzerverwaltung, Rechteverwaltung, Projektverwaltung usw.):

- **Verwaltungsaufgaben** – kurz: **ADMINISTRATION** [Vorsilbe: **ADMIN**]

4.3.2.1 Weitere Vorgangsweise

Zuerst werden die Strukturen der einzelnen Fachmodule in ihren derzeitigen Zustand ermittelt. Daraus werden dann schrittweise die Strukturen immer mehr abstrahiert um eine möglichst generische Datenstruktur für die Umsetzung in ein Relationales Datenmodell zu erhalten. Zusätzlich entwickelt man aus dieser Analyse auch den Ablauf der verschiedenen Arbeitsschritte, die gemeinsam dann auch die Realisierung der Oberfläche (Formulare) definieren.

4.3.2.2 Analyse Vegetationsaufnahmen

Um die Datenstruktur der Anwendung zu ermitteln, muss man zuerst den Bedarf kennen. Dabei muss man die Vorgangsweise in der Vegetationsökologie analysieren. Am Anfang steht immer das erkennen von bestimmten, sich wiederholenden Mustern in der Vegetation. In der Vegetationsökologie¹⁹ werden diese verschiedenen Muster der Vegetation beschrieben und mit verschiedenen Faktoren in Zusammenhang gebracht, d.h. es wird untersucht unter welchen physikalischen oder chemischen Faktoren, wie Klima, Bodenverhältnissen, usw. – kurz: **abiotischen Faktoren** – aber auch unter welchen biologischen Faktoren, wie Weidedruck durch Herbivoren, Bestäuber, usw. – kurz: **biotischen Faktoren** – sich ein bestimmtes Vegetationsmuster ausbildet. Welche Pflanzenarten dominieren einen bestimmten Standort, d.h. einen Ort mit bestimmten biotischen und abiotischen Faktoren, welche Arten sind besonders standorttreu, ohne den Bestand zu dominieren, welche sind selten oder fehlen völlig und warum. Dazu werden auf einer bestimmten Fläche alle Arten bestimmt, deren Artmächtigkeit bewertet und statistisch ausgewertet. Gleiche oder ähnliche Vegetationsmuster werden zu Pflanzengesellschaften zusammengefasst, die wieder in eine Systematik gestellt werden. Zusammenfassend kann man sagen, eine Vegetationsaufnahme besteht aus der Auflistung aller Arten einer bestimmten Fläche mit der Angabe der Artmächtigkeit zu jeder dieser Arten. Über eine oder mehrere Vegetationsaufnahmen kann der Pflanzenbestand eines Gebietes (kurz Bestand) untersucht werden.

¹⁹ Lt. Vorlesung „Vegetationsökologie“ von O. Univ.-Prof. Mag. Dr. Georg Grabherr

Zu einer Bestandsfläche werden auch verschiedene Daten erfasst, wie Name, geographische Position, klimatische, physikalische und chemische Eigenschaften, Bodenverhältnisse usw.. Diese Daten kann man unter den Namen „**Stammdaten eines Bestandes**“ zusammenfassen.

Die Datenstruktur einer Aufnahme kann man in „Artenliste“ und „Arten“ unterteilen. Die Arten und ihre Interpretationen sind eine sehr komplexe Thematik die in einem eigenen Modul (**BOTNAME**) behandelt werden.

Die Analyse der Daten eines Bestandes und seiner Aufnahmen ergibt anhand des Beispiels der Aufnahmen der österreichischen Moore Datenbank folgende Datenstruktur (siehe Abbildung 16):

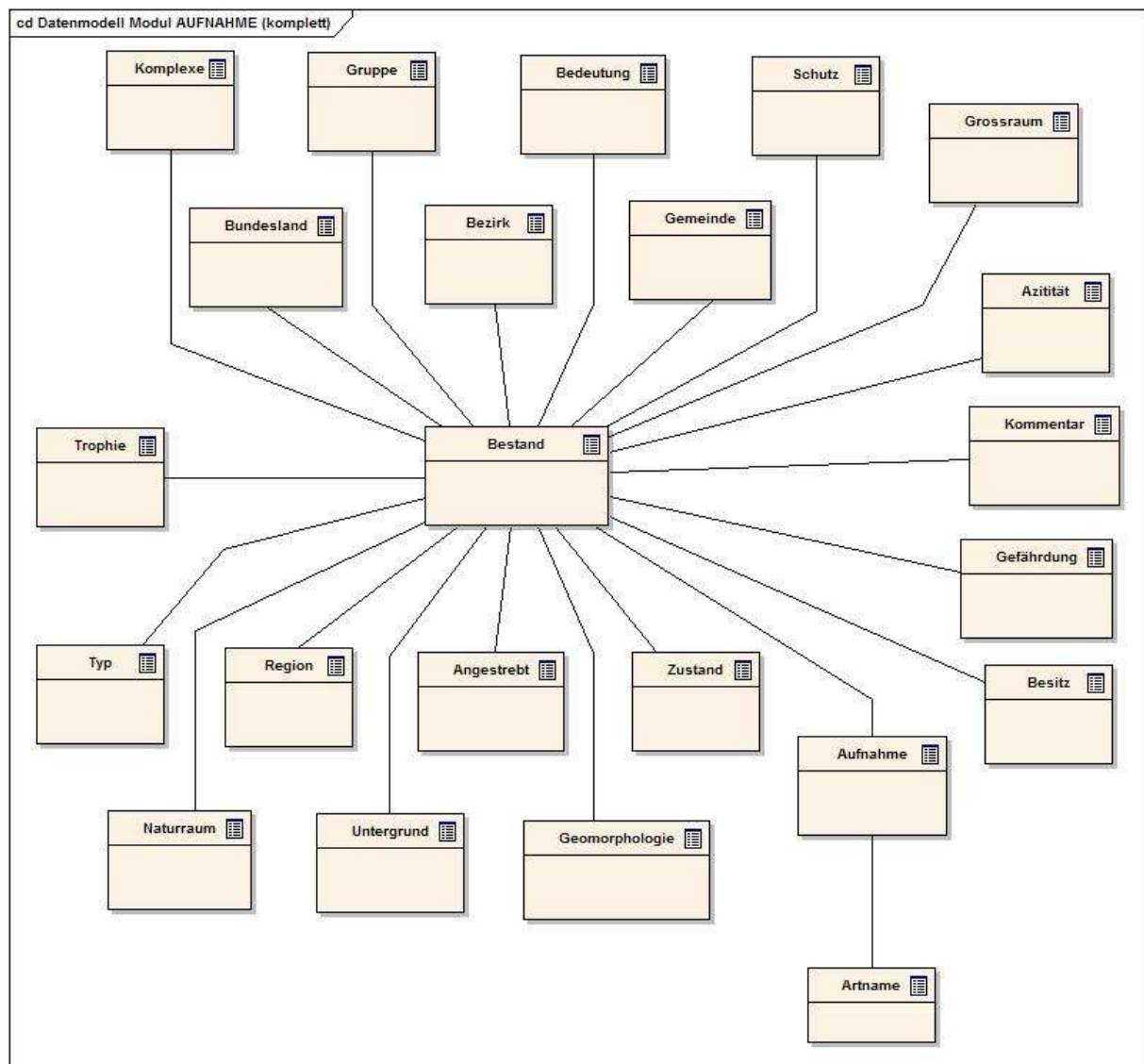


Abbildung 16 – Datenmodell Modul Aufnahme (fachlich)

Das Diagramm oben beschreibt das fachliche Datenmodell für Aufnahme­daten, d.h. der Stammdaten eines Bestandes und dessen Aufnahmen mit den Artenlisten.

Aus diesem fachlichen Datenmodell kann man sich wiederholende Teile zusammenfassen. Es ergeben sich aus meiner Sicht folgende Grundelemente (siehe Abbildung 17):

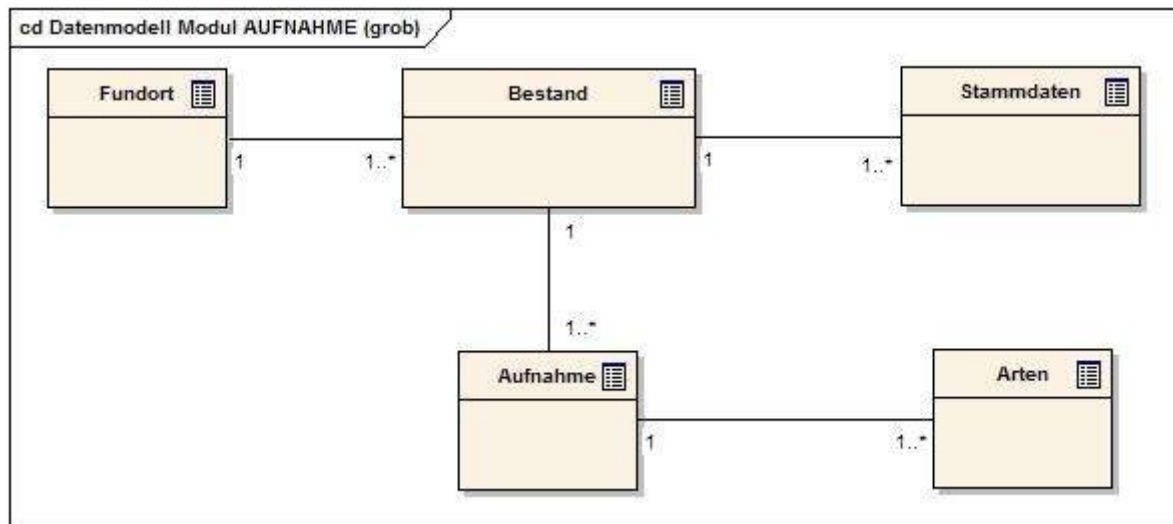


Abbildung 17 – Datenmodell Modul Aufnahme (grob)

Alle Elemente aus dem „Datenmodell Modul AUFNAHME (komplett)“ können in die Datenstruktur „Datenmodell Modul AUFNAHME (grob)“ übertragen werden!

Die Tabelle AUFNAHME ist die Haupttabelle. Sie enthält die allgemeinen Daten der Aufnahme (Name, eindeutige Nummer). Jeder Datensatz der Tabelle AUFNAHME verweist auf genau einen Datensatz aus der Tabelle BESTAND, d.h. jede Aufnahme wird genau einem Bestand zugeordnet, es können sich aber in einem Bestand mehrere Aufnahmen befinden (z.B.: wenn die Vegetation über einen längeren Zeitraum beobachtet werden soll oder wenn in einem Bestand mehrere Aufnahmen genommen werden). Jeder Datensatz der Tabelle BESTAND verweist auf einen oder mehreren Datensätzen aus der Tabelle STAMMDATEN, d.h. zu jedem BESTAND sind die Eigenschaften, wie z.B. die biotischen und abiotische Eigenschaften der Fläche, die meteorologischen Daten, Metadaten, usw. gespeichert. Im Prinzip gehört auch der Fundort zu den Stammdaten, es werden aber die Gemeinde, der politische Bezirk und das jeweilige Bundesland wegen der besseren Abfragemöglichkeiten und der besseren Wiederverwendbarkeit in einer eigenen Tabelle FUNDORT gespeichert, wobei nur die Gemeinde, in der der Bestand liegt, gespeichert wird. An dieser

Gemeinde sind im System bereits der Bezirk und das jeweilige Bundesland verknüpft. Der genaue Ort (man könnte dafür z.B. das Feld GPS-Koordinaten definieren) wird in der Tabelle AUFNSTAMMDATEN abgespeichert. Bei den Stammdaten ist keine Zeitreihe der Daten möglich, d.h. es ist z.B. nicht möglich (bzw. sinnvoll) die Jahresdurchschnittstemperatur eines BESTANDES über mehrere Jahre zu erfassen, als Stammdaten abzuspeichern und statistisch auszuwerten. Bei Bedarf kann diese Funktion über eine zusätzliche Tabelle, in der man zu den Stammdaten noch die Daten einer Zeitreihe abspeichern kann, realisieren. Zu jeder AUFNAHME eines BESTANDES gehört die Liste aller gefundenen Arten mit ihren wissenschaftlichen Namen und ihrer Artmächtigkeit. Diese Daten werden in den Tabellen AUFNAHME und ARTEN abgespeichert. Jeder Datensatz der Tabelle BESTAND kann einem oder mehrere Datensätze aus der Tabelle AUFNAHME enthalten. Bei einer Aufnahme werden auf einer definierten Fläche alle Arten bestimmt und bewertet (Artmächtigkeit). Diese Liste von Arten mit ihrer Artmächtigkeit wird in der Tabelle AUFNAHME gespeichert. Deshalb verweist jeder Datensatz der Tabelle AUFNAHME genau auf einen Datensatz der Tabelle ARTEN (d.h. für jeden Art der Artenliste einer Aufnahme gibt es einen eigenen Datensatz in der Tabelle AUFNAHME).

Jede Aufnahme befindet sich in einer Fläche, die Bestand genannt wird. Jeder Bestand hat einige Stammdaten, die sich oft wiederholen. Diese Stammdaten werden Aufnahmestammdaten genannt. Hier kann es zu einer Dateninkonsistenz kommen, wenn z.B. ein Bundesland durch einen Tippfehler in einen Aufnahmestammdatensatz anders geschrieben ist, als in einem anderen. Dadurch kann es maschinell nicht mehr richtig zugeordnet werden und daher werden diese Daten in einer entsprechenden Auswertung bzw. Suche nicht oder falsch ausgewertet bzw. angezeigt.

Deshalb macht es Sinn, diese Daten zusammenzufassen d.h. gleiche Werte in eine eigene Tabelle auszulagern und in der Tabelle AUFNSTAMMDATEN bei den einzelnen Datensätze statt den Wert nur mehr eine „Verknüpfung“ auf den Datensatz in der neuen Tabelle abzuspeichern (Normalisierung). Dadurch werden falsche Werte vermieden.

Unter Normalisierung²⁰ eines relationalen Datenbankschemas versteht man die schrittweise erfolgende Zerlegung von Relationen (in der Datenbank: Tabellen), in mehrere Relationen. Durch diese Zerlegung sollen mehrfach verwendete Daten in einer eigenen Relation (Tabelle) zusammengefasst werden. Dadurch werden Redundanzen der Daten vermeiden, die einen erhöhten Speicherplatz benötigen, das Durchsuchen und Analysieren der Daten verlängern und bei der Änderung von Daten zu Inkonsistenzen führen können.

Eine konsequente Durchführung der Normalisierung ergibt folgendes Datenmodell (siehe Abbildung 18):

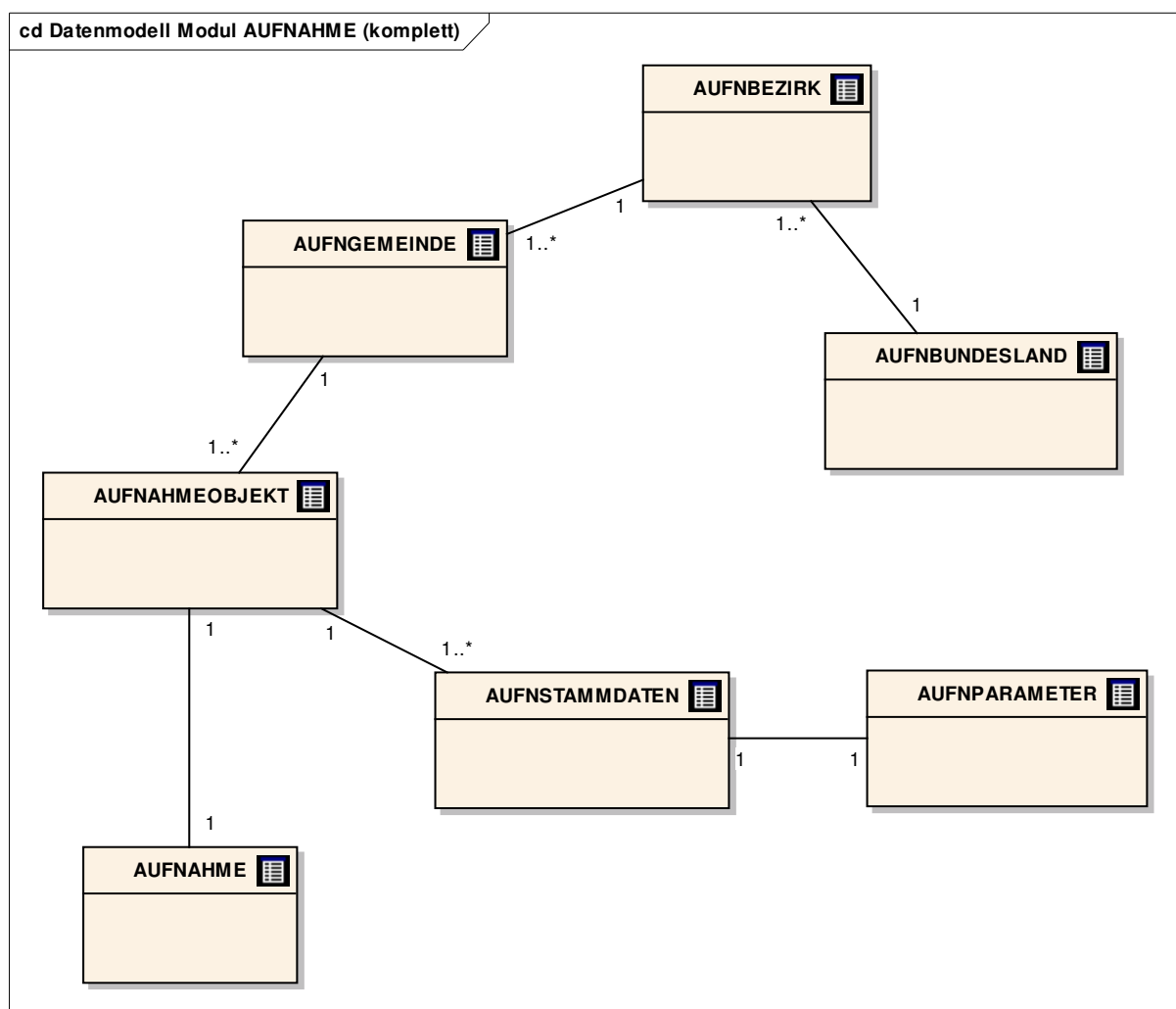


Abbildung 18 – Datenmodell Modul Aufnahme (komplett)

²⁰ [http://de.wikipedia.org/wiki/Normalisierung_\(Datenbank\)](http://de.wikipedia.org/wiki/Normalisierung_(Datenbank))

In diesem Datenmodell wird der Bestand in der Tabelle AUFNAHMEOBJEKT gespeichert, dem eine oder mehrere Aufnahmen in der Tabelle AUFNAHME zugeordnet werden.

Beim Arbeiten im Modul AUFNAHME sind verschiedene Tätigkeiten möglich, d.h. Projekt anlegen, die einzelnen AUFNAHMEOBJEKTE anlegen, die Arten der AUFNAHME mit ihrer Artmächtigkeit eingeben, die Daten kontrollieren und gegen irrtümliche Veränderung sperren, usw.. Nicht jeder darf alles machen, es gibt verschiedene Rollen, die bestimmte Tätigkeiten durchführen dürfen und denen bestimmte Tätigkeiten verboten sind. Im USE-CASE Diagramm (siehe Abbildung 19) sind die verschiedenen Tätigkeiten den verschiedenen Rollen zugeordnet.

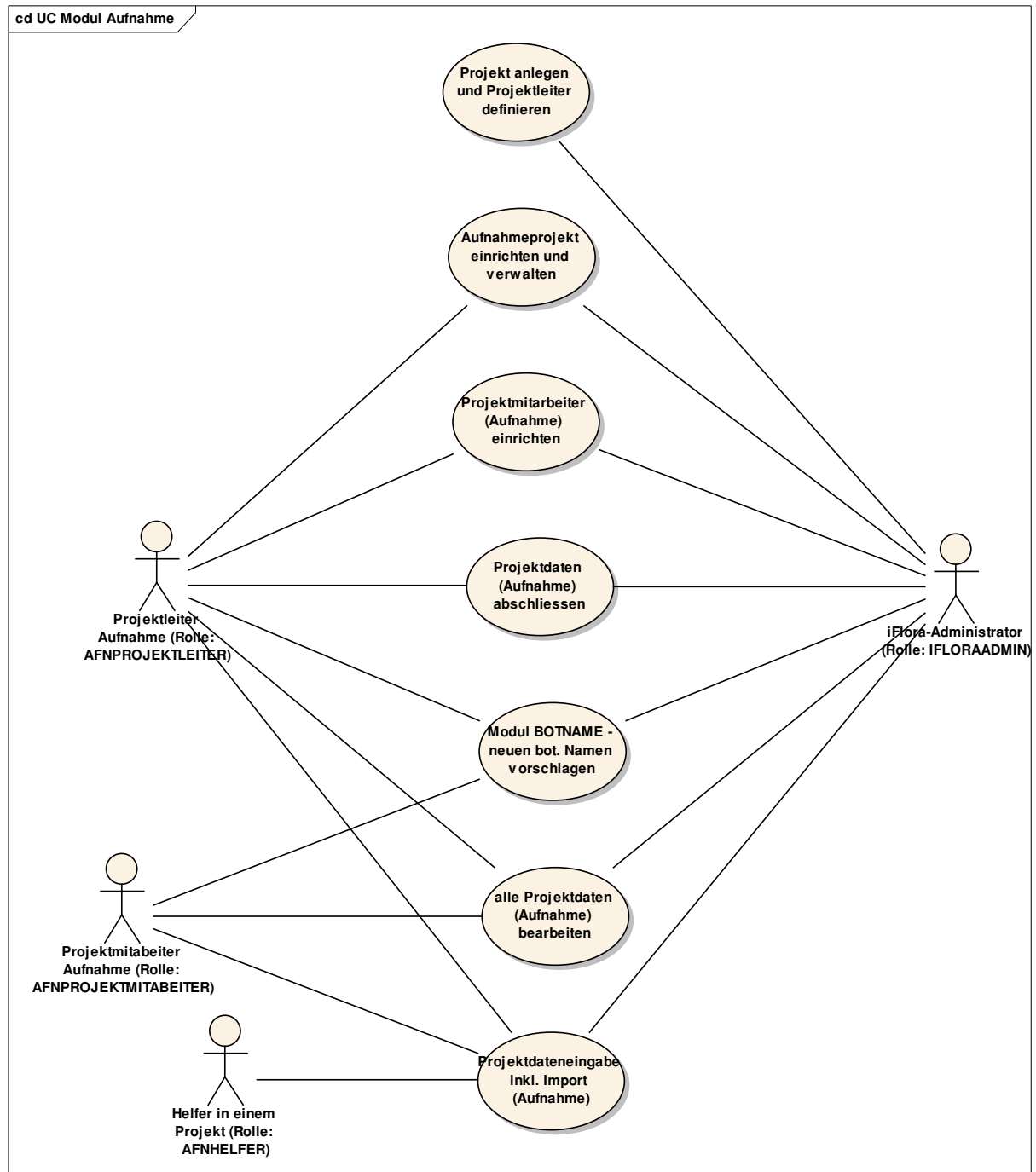


Abbildung 19 – Use Case Diagramm Modul Aufnahme

In jeden Modul gibt es definierte Abläufe, die für das Modul **AUFNAHME** sind im Diagramm (siehe Abbildung 20) beschrieben:

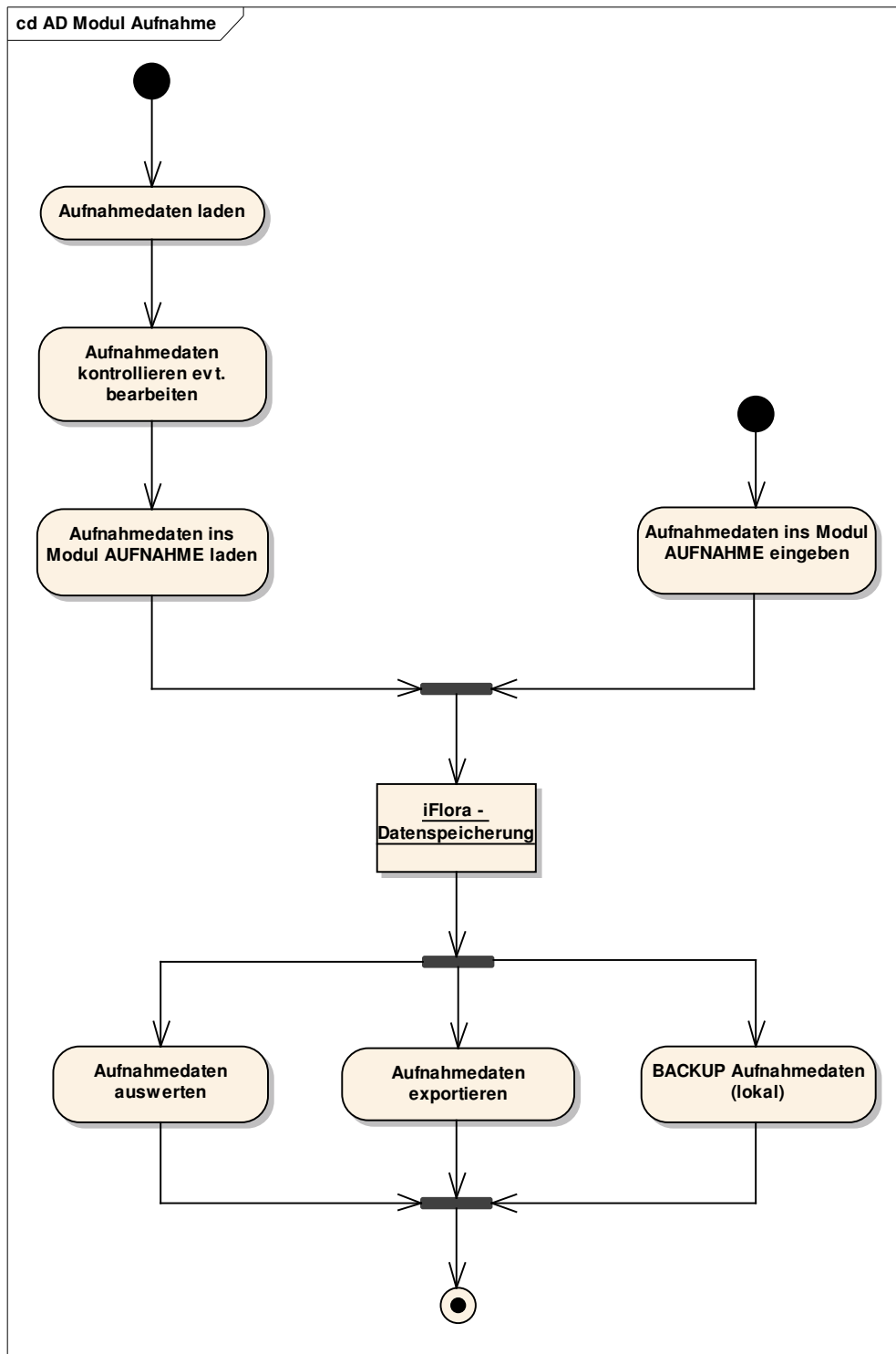


Abbildung 20 – Activity Diagramm Modul Aufnahme

4.3.2.3 Analyse Botanische Namen

Dieses Modul dient zur Verwaltung der einzelnen Arten, die bei den Vegetationsaufnahmen verwendet werden. Der Name einer Art besteht immer aus zwei Wörtern. Dem Gattungsnamen am Anfang (immer mit großen

Anfangsbuchstaben), zu dem die Art gehört und dem Art-Beinamen (immer mit kleinen Anfangsbuchstaben) (ADLER 1994, S 33). Zusätzlich gehört zum vollständigen wissenschaftlichen Name jeder Art auch die Angabe des nomenklatorischen Autors, das ist der Name (beziehungsweise Abkürzung) jenes Botanikers, der die erste Benennung und Beschreibung vorgenommen hat (ADLER 1994, S 33). Die Kenntnis des Autors und des Zitats der Erstbeschreibung sagt noch nichts über die aktuelle Bedeutung des Taxons²¹ (eine in der Biologie eine als systematische Einheit erkannte Gruppe von Lebewesen, z.B. eine Art) aus. Die ursprüngliche Bedeutung kann sich durch neue Erkenntnisse verändern. Der Artnamen sagt nichts über die über den taxonomischen Umfang (die Abgrenzung) der betreffenden Sippe aus, weil bei taxonomischen Änderungen die alten Namen nicht verschwinden dürfen, sondern nur eine andere Bedeutung erlangen. Ein bestimmter wissenschaftlicher Name wird daher nicht selten in verschiedenen Büchern in verschiedenen Sinn gedeutet. Um die Angabe des Pflanzentaxons exakter zu machen, empfiehlt es sich, neben den nomenklatorischen Autor noch jenes Florenwerk (Bestimmungsbuch) anzuführen, mit dem die Bestimmung durchgeführt wurde. Die Angabe des Florenwerkes muss immer mit der Auflagennummer und dem Erscheinungsjahr erfolgen, da sich auch die Bedeutung einer Art zwischen verschiedenen Auflagen ändern kann (ADLER 1994, S 33 f). Viele Arten haben mehr als einen lateinischen Namen. Eine Art darf aber nur einen einzigen korrekten Namen haben, der durch internationale Nomenklaturregeln festgelegt wird. Außerdem muss jeder korrekte wissenschaftliche Name einer Art mit einem Herbarexemplar (den nomenklatorischen Typus) verknüpft sein. Die anderen Namen einer Art heißen Synonyme. Sie entstehen meist, weil durch neue Erkenntnisse sich der korrekte Namen einer Art ändert und viele noch mit den alten, bekannten Namen weiterarbeiten (ADLER 1994, S 34).

Zusätzlich können auch Unterarten bei einer Vegetationsaufnahme verwendet werden. Eine Unterart besteht aus einem Gattungsnamen, dem Art-Beinamen, dem Kennzeichen „SSP“ und dem Namen der Unterart. Die Unterart ist der Art untergeordnet, daher sind bei der Angabe einer Art alle Unterarten eingeschlossen (ADLER 1994, S 26). Es kann sinnvoll sein, bei einer Vegetationsaufnahme die Art

²¹ <http://de.wikipedia.org/wiki/Taxon>

näher einzuschränken und mit einer Unterart zu arbeiten. Deshalb wird dieses Taxon im Modul **AUFNAHME** zur Auswahl angeboten und daher müssen sie im Datenmodell des Moduls (**BOTNAME**) berücksichtigt werden.

Mehrere Kleinarten (sehr ähnliche Arten, die nur mit großen Schwierigkeiten bestimmt bzw. unterschieden werden können) können zu Artengruppen zusammengefasst werden. Diese Artengruppen werden Aggregat genannt und ist nach der jeweils häufigsten Art dieser Gruppe genannt (Gattungsname und Art-Beinamen) und durch den Zusatz „AGG“ als Aggregat gekennzeichnet (ADLER 1994, S 26). In der Anwendung müssen die einzelnen Kleinarten, die durch das Aggregat zusammengefasst sind, gespeichert werden.

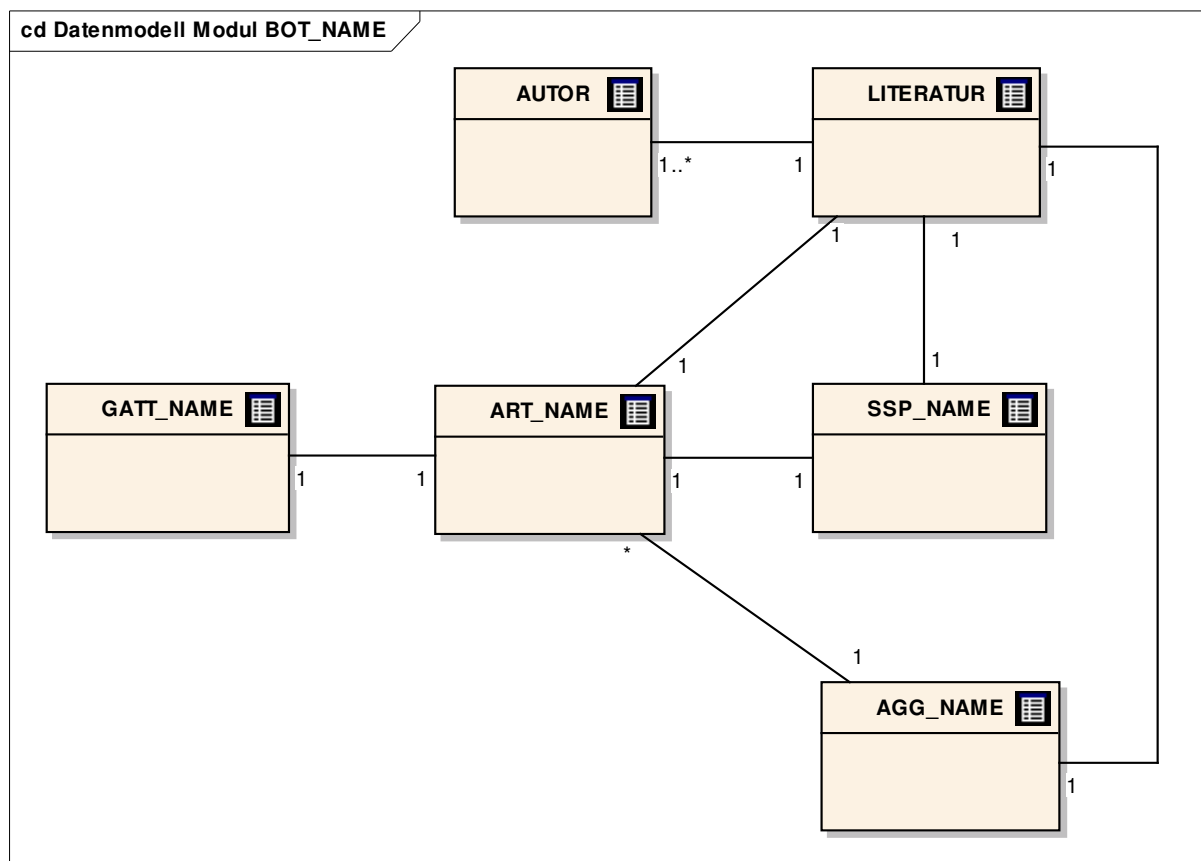


Abbildung 21 – Datenmodell Modul Botanische Namen

Die Tabellen des Artnamens (GATT_NAME und ART_NAME) sind der zentrale Teil des Moduls (siehe Abbildung 21) und definiert eine Art, zusätzlich mit der Tabelle Unterart (SSP_NAME) eine Unterart oder ein Aggregat (mit Hilfe der Tabelle AGG_NAME, in der die Verknüpfungen aller Kleinarten des Aggregats enthalten sind).

Ein Datensatz in der Tabelle ART_NAME enthält den Art-Beinamen und die Verknüpfung zum Gattungsnamen, der in der Tabelle GATT_NAME abgespeichert ist. Über die Tabelle AGG_NAME kann eine Art zu einem Aggregat erweitert werden. Dabei befinden sich in der Tabelle AGG_NAME die Verknüpfungen zu allen Kleinarten, die zum jeweiligen Aggregat gehören.

Wichtig ist in diesem Zusammenhang, dass die Verknüpfung mit den Gattungsnamen (über die eindeutige ID des Gattungsnamens) und dem Feld mit den Art-Beinamen in der Tabelle **ART_NAME** nicht eindeutig sein darf, da in dieser Tabelle zur Art noch Unterarten und Aggregate gespeichert werden können (jeweils ein eigener Datensatz, da die Tabelle **ART_NAME** die zentrale Tabelle für die Verknüpfung zu den anderen Modulen ist).

Zusätzlich gibt es im Modul **BOTNAME** noch verschiedene Tätigkeiten (einen wissenschaftlichen Namen eingeben, einen wissenschaftlichen Namen abschließen und diesen damit für die Benutzung durch die anderen Module freigeben, usw.). Nicht jeder darf alles machen, es gibt verschiedene Rollen, die bestimmte Tätigkeiten durchführen dürfen und denen bestimmte Tätigkeiten verboten sind. Im USE-CASE Diagramm (siehe Abbildung 22) sind die verschiedenen Tätigkeiten den verschiedenen Rollen zugeordnet.

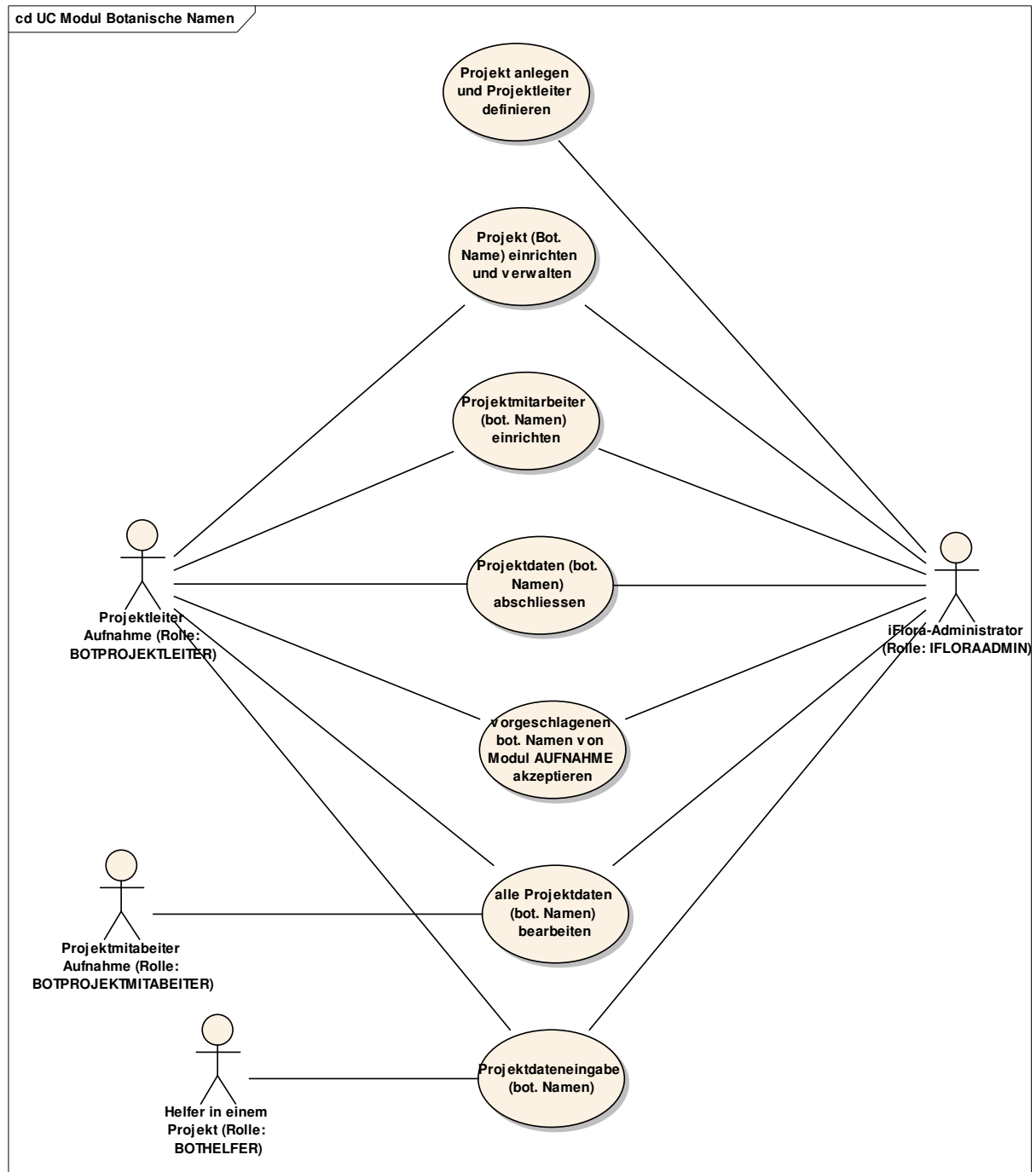


Abbildung 22 – Use Case Diagramm Modul Botanische Namen

In jeden Modul gibt es definierte Abläufe, die für das Modul **BOTNAME** sind im Diagramm (siehe Abbildung 23) beschrieben.

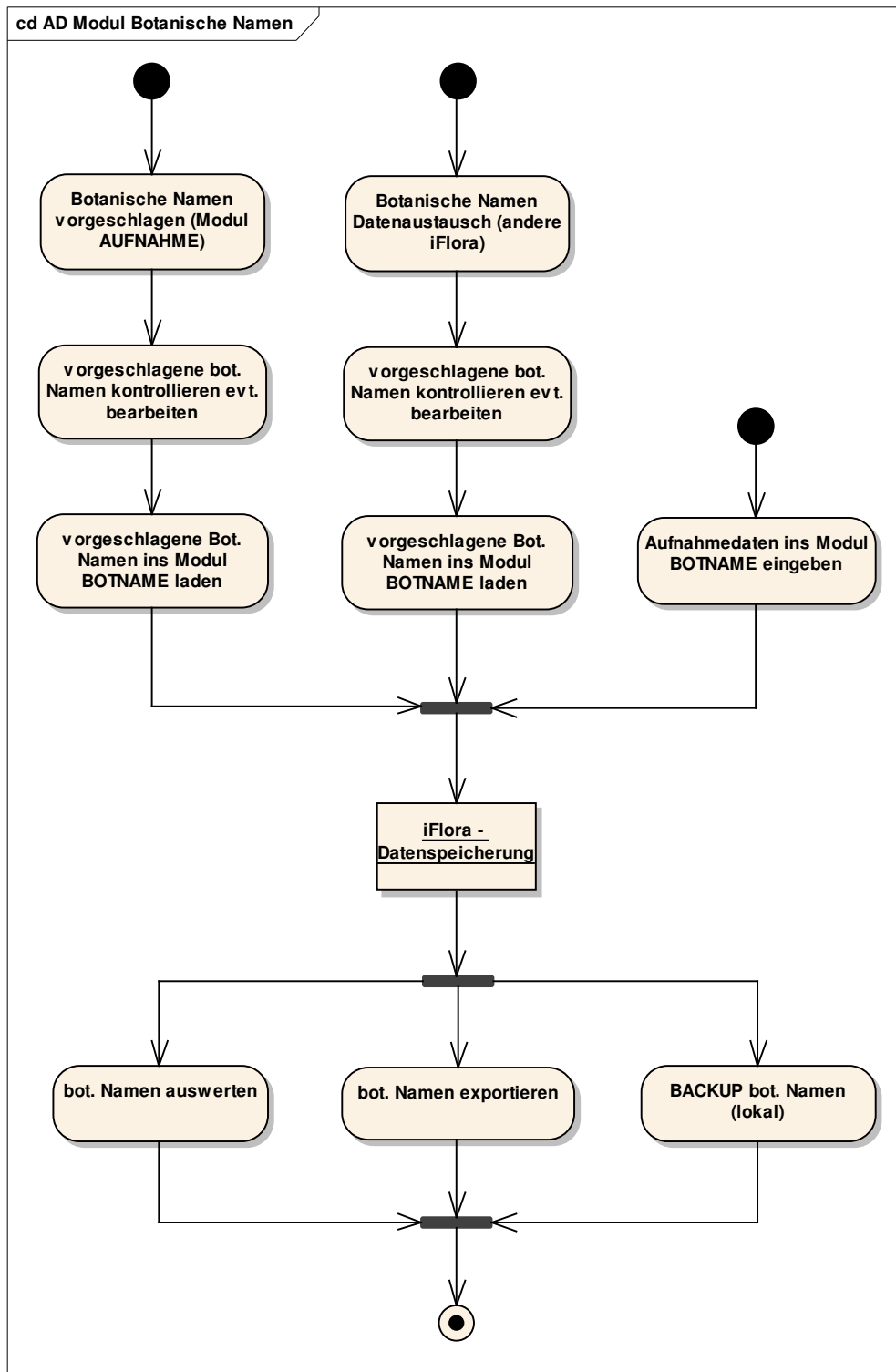


Abbildung 23 – Activity Diagramm Modul Botanische Namen

4.3.2.4 Analyse Administration

Für allgemeine Verwaltungsaufgaben muss auch eine Datenstruktur geschaffen werden. Die einzelnen Benutzer der Anwendung werden nicht direkt in der

Datenbank angelegt, sondern sie sind nur der Anwendung bekannt und der Zugriff auf die Datenbank erfolgt über einen zentralen Anwendungsbenutzer. Aus diesem Grund müssen die einzelnen Benutzer mit ihren Benutzernamen und Passwort in der Anwendung registriert werden. Zusätzlich müssen die Rechte dieser Benutzer gespeichert werden. Diese Daten werden im Modul ADMINISTRATION verwaltet und gespeichert (siehe Abbildung 24).

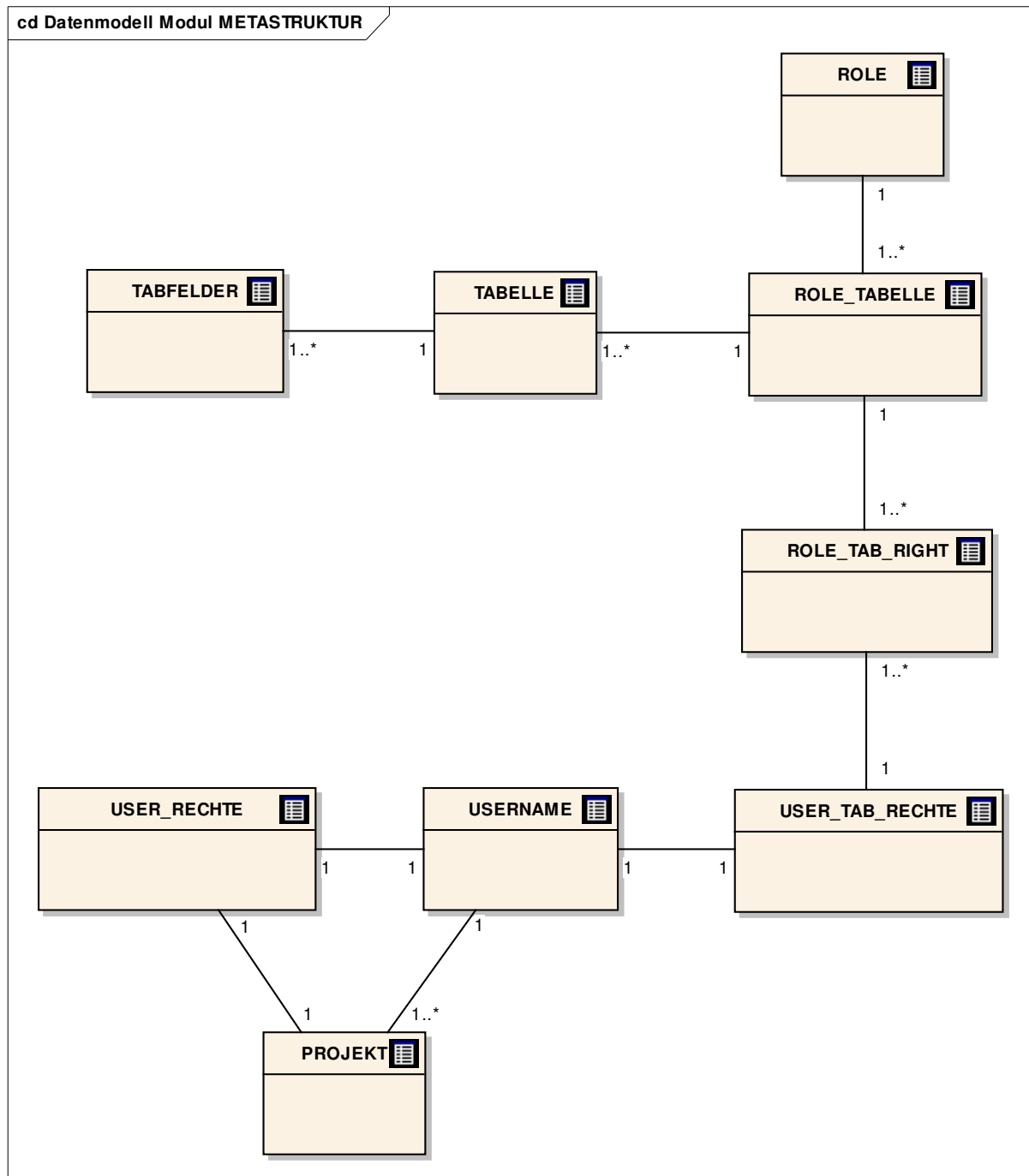


Abbildung 24 – Datenmodell Modul Administration

Die Administration soll die Rechte der einzelnen Benutzer innerhalb des Administrationsbereichs der Anwendung iFlora definieren. Jeder Benutzer muss sich mit Namen und Passwort beim System identifizieren. Zusätzlich werden jedem Benutzer Rechte zugeordnet und er wird für ein Projekt freigeschaltet.

Es gibt Überlegungen, den Administratoren die Möglichkeit zu geben, eigene Tabelle zu generieren. Dazu können für die einzelnen Tabellen Rechte für die Benutzer vergeben werden, wobei die Rechte pro Tabelle zu Rollen zusammengefasst und dann den Benutzern zugewiesen werden.

Diese Funktionalität kann aus Zeitgründen im Rahmen dieser Arbeit nicht umgesetzt werden.

5 Entwurf

5.1 Allgemeine Datenbankstruktur

Es gibt ein SCHEMA, das mit dem User SH\$IFLORA verknüpft ist. Dieses Schema enthält die Datenstruktur der Module **AUFNAHME (AFN)**, **BOTANISCHE NAMEN (BOT)** und **ADMINISTRATION (ADM)**. Vor allem das Modul BOTANISCHE NAMEN ist ausgelegt auf einen massiven Datenaustausch mit anderen Institutionen.

5.2 Internetkonzept (allgemein)

Die Daten sollen über das Internet zugänglich sein. Es soll eine Anwendung mit einer Internet Benutzerschnittstelle entwickelt werden.

Dabei gibt es Daten (Projektdaten) die einer breiten Öffentlichkeit „lesend“ zur Verfügung gestellt werden und einen Bereich in dem die Daten (Projektdaten, Administrationsdaten, Metadaten) bearbeitet werden können (http Verbindung zwischen dem Internetbrowser des Interessenten und den Apache Server von iFlora). Dafür müssen sich die Projektmitarbeiter über einen Benutzernamen und ein Passwort gegenüber dem System identifizieren und es wird eine gesicherte Verbindung (https) zwischen den Client (Internetbrowser des Projektmitarbeiters) und den Server (Apache Server von iFlora) eingerichtet.

5.2.1 Einleitung

Durch die Aufgabe, die Daten einer breiten Öffentlichkeit zur Verfügung zu stellen und den Projektmitarbeitern zu ermöglichen, ihre Daten über das Internet zu bearbeiten, teilt sich die Anwendung in zwei Bereiche:

- **„iFlora Öffentlicher Bereich“** (kurz: **„iFloraPublic“**) – ist der Bereich, der die Projektdaten lesend einer breiten Öffentlichkeit zur Verfügung stellt.
- **„iFlora Administration“** (kurz: **„iFloraAdmin“**) – ist der Bereich zur Bearbeitung der Daten (Projektdaten, Administrationsdaten, Metadaten).

Aufgrund der Datenstruktur der Projektdaten (Module **AUFNAHME (AFN)** und **BOTANISCHE NAMEN (BOT)**) ist im Bereich **iFloraPublic** die Grobstruktur der Internetanwendung vorgeben:

- Ansicht der Vegetationsaufnahmen
- Ansicht der Botanischen Namen

Zusätzlich soll die Daten dynamisch zusammengestellt werden und sie sollen Downgeloaded werden können, d.h. es kommen noch folgende Punkte dazu:

- Auswertungen
- Download

Bei der Grobstruktur des Bereichs **iFloraAdmin** müssen alle Datenstrukturen berücksichtigt werden (Module **AUFNAHME (AFN)**, **BOTANISCHE NAMEN (BOT)** und **ADMINISTRATION (ADM)**):

- Verwaltung der Administrationsdaten
- Verwaltung der Metadaten
- Verwaltung der Vegetationsaufnahmen
- Verwaltung der Botanischen Namen

Zusätzlich soll die Daten dynamisch zusammengestellt werden und sie sollen Downgeloaded bzw. die Daten eines Projektes sollen lokal gesichert werden können, d.h. es kommen noch folgende Punkte dazu:

- Auswertungen
- Download

5.3 Vegetationsaufnahme – Modul AUFNAHME

5.3.1 Datenmodell

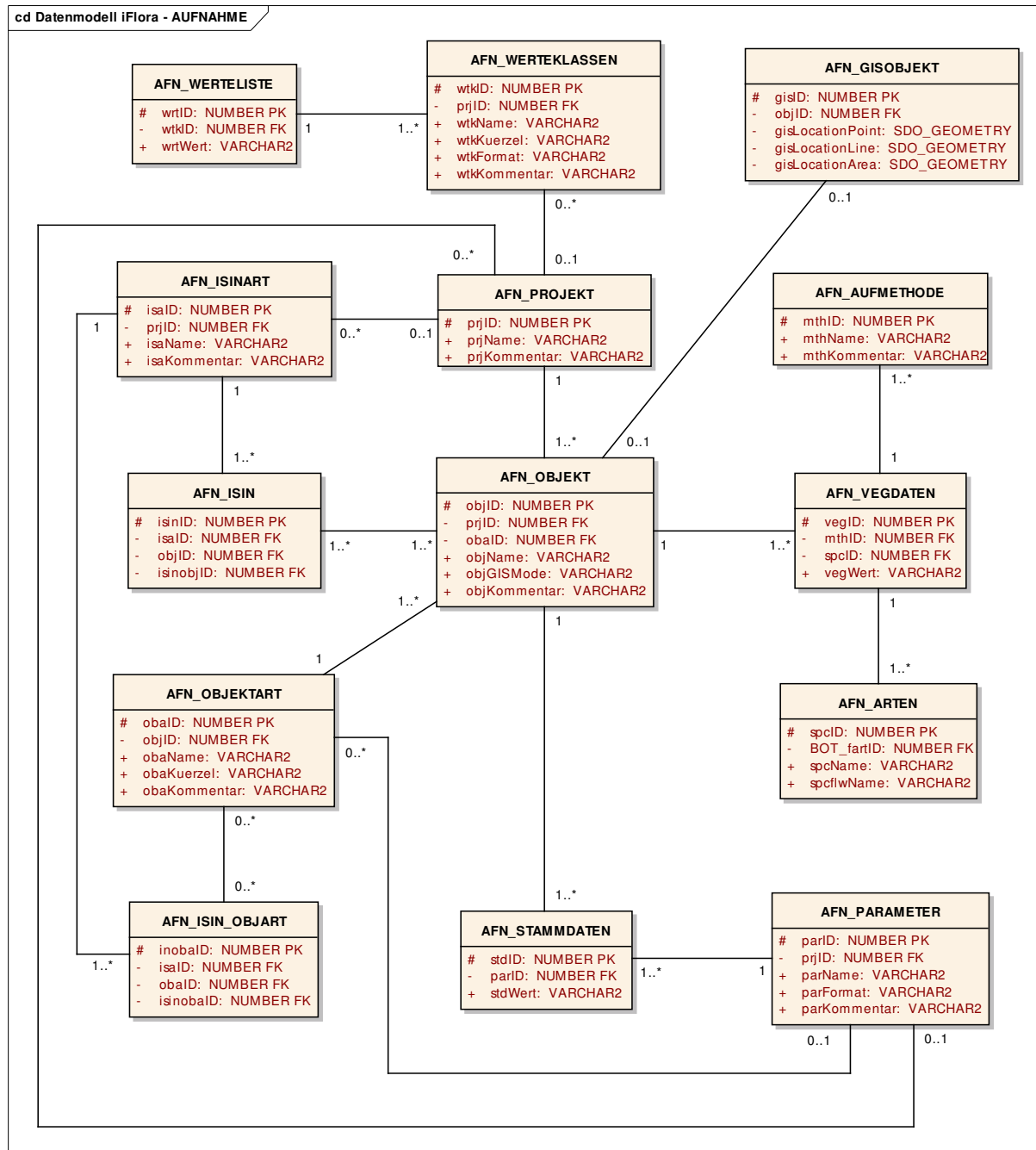


Abbildung 25 – Datenmodell iFlora AUFNAHME

Die Umsetzung des Moduls „Aufnahme“ (IST Modell) entwickelt sich aus dem Kapitel „Analyse Vegetationsaufnahmen“ (SOLL Modell).

Die Tabelle **AFN_OBJEKT** ist die zentrale Tabelle (siehe Abbildung 25). Hier werden alle Objekte des Moduls „**AUFNAHME**“ abgespeichert. Ein Objekt²² ist allgemein etwas Unspezifiziertes, eine Sache oder ein Gegenstand, worauf ein Subjekt seine

²² <http://de.wikipedia.org/wiki/Objekt>

beobachtende und praktisch-verändernde Aktivität richtet. Dazu gehören in diesem Modul z.B. der Bestand, die Aufnahmen, die Bundesländer, die Bezirke, die Gemeinden usw.

Jedes Objekt gehört zu einer bestimmten Objektart. Diese Objektarten werden in der Tabelle **AFN_OBJEKTART** gespeichert.

Bestimmte Objekte können näher beschrieben werden. Diese Daten werden Stammdaten (eines Objektes) genannt und in der Tabelle **AFN_STAMMDATEN** gespeichert (dabei werden die Werte unter ihrem jeweiligen Parameter (siehe unten) abgespeichert).

Die Stammdaten (eines Objektes) sind die verschiedenen Eigenschaften bzw. nähere Erklärungen zu einem Objekt. Jeder Eigenschaft bzw. Erklärung wird ein Parameter zugeordnet. Hier wird definiert, um welche Eigenschaft bzw. Erklärung es sich handelt und in welcher Form diese Daten abgespeichert werden (z.B.: „Jahresdurchschnittstemperatur“ als Zahl, „Aufnahmedatum“ als Datum, „Fundort“ als Text, usw. => Parametername/Parameterformat). Diese Daten, d.h. die Definition um welche Art von Eigenschaft es sich handelt (Parametername) und in welcher Form diese Daten abgespeichert werden (Parameterformat), werden in der Tabelle **AFN_PARAMETER** abgespeichert. Jeder Parameter kann einer Objektart zugeordnet werden, dann gilt dieser Parameter nur für Objekte dieser Objektart und ist für die Objekte der anderen Objektarten nicht sichtbar.

Beim Parameterformat kann auch definiert werden, dass die Werte eines Parameter nicht manuell frei wählbar eingegeben werden dürfen, sondern nur aus einer festgelegten Liste von Werten ausgewählt werden darf. Diese Liste wird Werteliste genannt. Jede Werteliste (d.h. Gruppe von verschiedenen Werten) bekommt einen Namen, der Wertelistenklasse genannt wird. Die Namen der Wertelistenklassen werden in der Tabelle **AFN_WERTEKLASSE** gespeichert, die dazugehörenden Werte der Wertelisten in der Tabelle **AFN_WERTELISTE**.

Verschiedene Objekte können miteinander in Beziehung gesetzt werden. So werden einem Bestand die einzelnen Aufnahmen zugeordnet. Dem Bestand die geographische Lage über die Objekte Gemeinde, Bezirk und Bundesland. Diese Daten werden in der Tabelle **ANF_ISIN** gespeichert.

Jede Beziehung von Objekten wird unter einem bestimmten Thema abgespeichert. So bekommen sie erst einen Sinn und können ausgewertet werden. Diese Metadaten werden in der Tabelle **AFN_ISINART** gespeichert.

Es können nur Objekte verschiedener Objektarten miteinander in Beziehung gesetzt werden. Diese Metadaten werden in der Tabelle **AFN_ISIN_OBJART** gespeichert.

Jedes Objekt kann GIS-Daten enthalten. GIS-Daten sind die Koordinaten oder Polygone eines „Geographischen Informations Systems“ und definieren die geographische Position eines Punktes oder einer Fläche (innerhalb dieses Systems). Die GIS-Daten eines Objektes werden in der Tabelle **AFN_GISOBJEKT** gespeichert.

Zu jeder Aufnahme (d.h. Objekt von der Objektart AUFNAHME) gibt es Vegetationsdaten. Die Vegetationsdaten besteht aus einer Artenliste (das sind alle Arten, die auf der definierten Fläche der Aufnahme gefunden wurden) und deren Artmächtigkeit (es gibt verschiedene Methoden der Bewertung der Artmächtigkeit von Arten einer Aufnahme, im Prototypen, der in dieser Arbeit beschrieben wird, wurde nur die Bewertung nach Braun-Blanquet eingefügt). Die Vegetationsdaten werden in der Tabelle **AFN_VEGDATEN** gespeichert. Innerhalb dieser Tabelle werden die Arten aus der Tabelle **AFN_ARTEN** verwendet, d.h. es werden nicht bei jedem Datensatz der Artenliste die Arten direkt eingetragen, sondern immer nur eine Verknüpfung (Referenz) auf den Namen der Art in der Tabelle **AFN_ARTEN**. (Diese Tabelle ist nur der Platzhalter für die Verknüpfung mit den Arten des Moduls **BOTNAME** (Tabelle **ART_NAME**), die in dieser Arbeit aus Zeitgründen nicht mehr umgesetzt werden konnte.) Die Methoden der Aufnahme werden in der Tabelle **AFN_AUFMETHODE** abgespeichert.

Die Daten des Moduls **AUFNAHME** müssen einem Projekt zugeordnet werden. Damit können verschieden Projektgruppen diese Infrastruktur gemeinsam verwenden, ohne sich gegenseitig zu stören. Außerdem können damit die Benutzerrechte auf Projektebene verteilt werden und damit können die Mitarbeiter eines Projektes nicht irrtümlich die Daten eines anderen Projektes verändern. Ein zusätzlicher Vorteil ist, das man die gesamten Daten eines Projektes zu jedem Zeitpunkt sichern kann. Die Projekte werden in der Tabelle **AFN_PROJEKT** gespeichert und man kann jedes Objekt, jeden Parameter, jede Werteklasse und jede Objektart-Beziehung einem Projekt zuordnen. Mit dieser Zuordnung sind diese

Definitionen nur für das jeweilige Projekt sichtbar. Gibt es keine Zuordnungen, dann sind sie für alle Projekte sichtbar.

Zusätzlich bekommen alle fachlichen Daten (Objekte, Stammdaten, Vegetationsdaten) der einzelnen Projekte eine Verknüpfung mit dem jeweiligen Projekt. Dadurch ist erst die Zuordnung aller fachlichen Daten zu einem Projekt möglich. Diese Funktionalität wird beim Export der Daten (Menüpunkt Download) ausgenutzt.

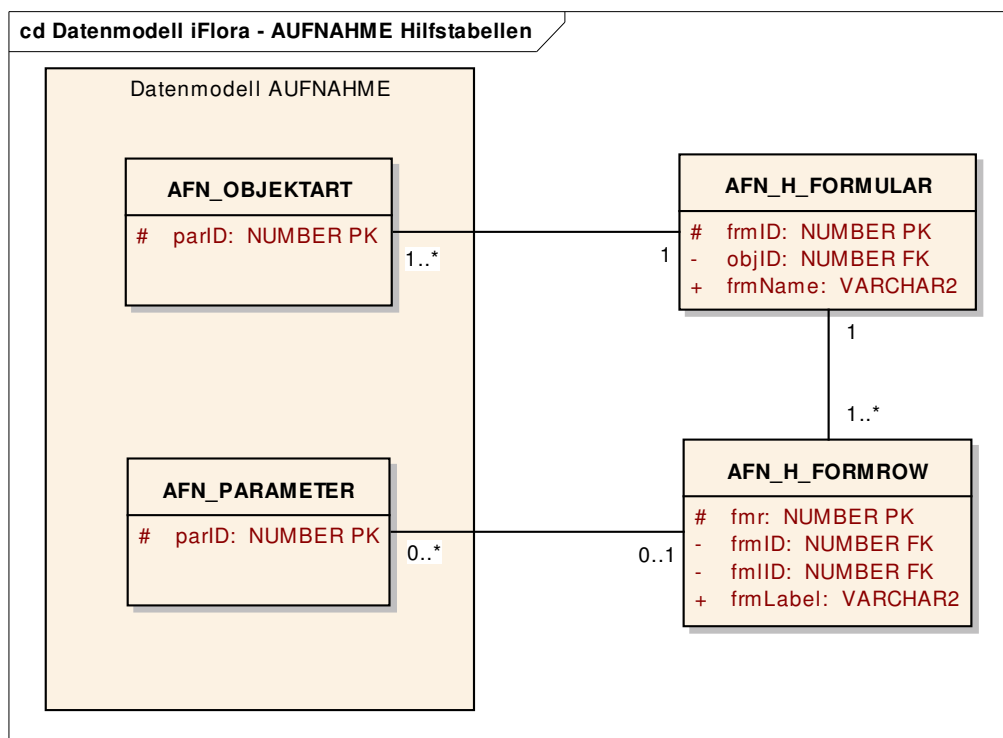


Abbildung 26 – Datenmodell iFlora Aufnahme Hilfstabellen

Die Formulare sollen dynamisch erstellt werden (siehe Abbildung 26). Dazu wird ein Formular definiert (Tabelle AFN_H_FORMULAR) und zu jedem Formular werden die Zeilen definiert, wobei eine Zeile aus einer Überschrift oder aus den Daten eines Parameters bestehen kann (Tabelle AFN_H_FORMROW).

Bei den Formularen handelt es sich um Hilfsdaten (bzw. Metadaten) der Anwendung, deshalb enthält der Tabellename die Zeichenkombination „_H_“.

5.3.2 Konzept Internetanwendung

Durch die Aufgabe, die Daten einer breiten Öffentlichkeit zur Verfügung zu stellen und den Projektmitarbeitern zu ermöglichen, ihre Daten über das Internet zu

bearbeiten, kommt es zu einer Aufteilung der Anwendung in die Bereiche **iFloraPublic** und **iFloraAdmin**. In beiden Bereichen sollen die Daten der **AUFNAHME** angezeigt werden.

5.3.2.1 Öffentlicher Bereich

Eine Aufnahme gehört immer zu einem Pflanzenbestand (im weiteren Verlauf kurz Bestand genannt) und dieser befindet sich immer in einem geographischen Bezug (Gemeinde, Bezirk, Bundesland). Ein Bestand mit seinen Aufnahmen muss immer einem Projekt zugeordnet sein. Zusätzlich gibt es für eine Aufnahme (über den Bestand) noch andere Beziehungen (Region, Naturraum, Großraum)

Zuerst sollen die Bestände mit ihren Aufnahmen nach diesen Beziehungen ausgewählt werden können. Dazu wird es für jede dieser Beziehungen ein eigenes Formular geben:

- Geographie
- Region
- Naturraum
- Großraum

Wenn man einen Bestand ausgewählt hat, gibt es entweder die Möglichkeit, sich dessen Stammdaten anzusehen oder weiter zu gehen und sich die Aufnahmen des Bestandes in einem eigenen Formular anzeigen zu lassen. Hier kann man sich dann entweder die Stammdaten der Aufnahme oder die Vegetationsdaten in einem Formular anzeigen lassen. Diese Auswahl soll unter dem Menüpunkt **AUSWAHL** aufgerufen werden können.

In der Aufnahme gibt es Objekte, die in der Tabelle **AFN_OBJEKT** gespeichert werden. Jedes Objekt gehört zu einer Objektart (Tabelle **AFN_OBJEKTART**). Es soll für jede Objektart einen eigenen Untermenüpunkt geben unter dem sich ein Formular öffnen, in dem alle Objekte der Objektart alphabetisch angeordnet sind. Gleichzeitig wird ein Link zum Öffnen des Stammdatenformulars angezeigt. Alle diese Untermenüs sollen im Menü **OBJEKT** zusammengefasst werden.

Zusätzlich soll die Daten dynamisch zusammengestellt werden und sie sollen Downgeloaded werden können, d.h. es kommen noch folgende Punkte dazu:

- Auswertungen

- Download

Die Auswertung soll den Benutzer die Möglichkeit geben, sich die Daten der **AUFNAHME** schrittweise zusammenzustellen. Es sind aber nur Daten sichtbar, die von den jeweiligen Projektadministratoren freigegeben wurden (d.h. die den Status „FREIGABE“ haben). Im ersten Schritt sollen die Benutzer eine Objektart auswählen können. Im zweiten Schritt kann man eine oder mehrere Objekte der ausgewählten Objektart auswählen. Jedes Objekt kann Stammdaten enthalten. Eine Objektart (**AUFNAHME**) kann zusätzlich noch Vegetationsdaten besitzen. Der nächste Schritt wird nur angezeigt, wenn es sich beim Objekt um eine Aufnahme handelt. Hier gibt es eine Auswahl, ob man nach den Vegetationsdaten oder nach den Stammdaten des Objektes suchen will.

In der Stammdatensuche kann man sich im nächsten Schritt die Stammdatenparameter zusammenstellen, über die man Information abrufen will. Danach gibt man das Ausgabeformat an, in dem man das Ergebnis der Abfrage angezeigt haben will. Es gibt die Möglichkeit sich die Daten in HTML oder Excel ausgeben zu lassen.

Hat man sich bei den Aufnahmen für die Vegetationsdaten entschieden, kann man in diesem Schritt die Arten auswählen, die man bei der Abfrage angezeigt haben will. Nach Auswahl des Ausgabeformates wird das Ergebnis angezeigt.

Über den Punkt DOWNLOAD kann man die gesamten, freigegebenen Daten eines Projektes downloaden.

5.3.2.2 Administrationsbereich

5.3.2.2.1 Systemverwaltung

Im Administrationsbereich der **AUFNAHME** können die Administratoren die Benutzer- und Metadaten administrieren. Es gibt hier folgende Möglichkeiten:

- **Benutzerverwaltung** – hier können neue Benutzer angelegt, vorhandene Benutzer verwaltet und alte Benutzer deaktiviert werden. Zusätzlich können hier die Benutzerrechte verwaltet und die Benutzer bestimmten Projekten zugeordnet werden. Diese Funktionalität kann nur von Administratoren und Projektadministratoren (nur für Benutzer seines Projektes) ausgeführt werden.

So macht es Sinn, wenn Projektadministratoren für Benutzer, die ihrem Projekt zugeordnet sind, z.B. neue Passwörter vergeben können.

- **Projektverwaltung** – hier können Projekte angelegt und verwaltet werden. Auch diese Funktionalitäten können nur Administratoren durchführen.
- **Objektartverwaltung** – jedes Objekt muss einer Objektart zugeordnet sein. Es können beliebige Objektarten frei definiert werden. Damit es nicht zu Doppelgleisigkeiten kommt, sollte diese Funktionalität allerdings nur von den Administratoren, in Abstimmung mit den Projektadministratoren ausgeübt werden.
- **IS-IN-Artverwaltung** – Objekte einer Objektart können Objekten einer anderen Objektart, unter einem bestimmten Thema, zugeordnet werden. Eine Objektbeziehung muss immer einem Thema zugeordnet sein. Dieses Thema wird unter diesem Punkt verwaltet. Damit es nicht zu Doppelgleisigkeiten bei den Themen kommt, sollte diese Funktionalität nur von den Administratoren, in Abstimmung mit den Projektadministratoren ausgeübt werden.
- **IS-IN-Objektartverwaltung** – Bei jeder Objektbeziehung müssen die Objektarten, aus denen das Thema besteht, vordefiniert werden. Es ist zwar möglich, diese Objektarten frei zu definieren, aber diese Definition ist dann fix. Es ist nicht möglich, unter einem bestimmten Thema Objekte beliebiger Objektarten miteinander zu verknüpfen. Damit es nicht zu Doppelgleisigkeiten bei den Themen kommt, sollte diese Funktionalität nur von den Administratoren, in Abstimmung mit den Projektadministratoren ausgeübt werden.

Diese Punkte werden unter dem Menü **SYSTEMVERWALTUNG** zusammengefasst.

5.3.2.2.2 Administration

Zusätzlich haben die Administratoren und Projektadministratoren noch die Möglichkeit, sich ihre Daten- und Arbeitsumgebung einzurichten.

- **Parameterverwaltung** – Jedes Objekt kann bestimmte Eigenschaft (z.B.: Klima, Besitz, Gefährdung, usw.) haben. Jede Eigenschaft wird als Parameter definiert und die Form der eingegebenen Werte der Eigenschaft werden als Parameterformat definiert. Es können beliebige Eigenschaften als Parameter definiert werden, solange es sich um eine Zeichenfolge von max. 255 Zeichen

handelt. In Zukunft ist geplant, auch Dateien (pdf, jpg, usw.) als Parameterinhalte zuzulassen. Zusätzlich sollen zu jedem Parameter Hilfetexte erstellt werden, die in den Stammdatenformularen sowohl im Öffentlichen Bereich (iFloraPublic) als auch im Administrationsbereich (iFloraAdmin) über einen Hilfebutton („?“) aufgerufen werden kann. Die Parameter können von den Administratoren und den Projektadministratoren verwaltet werden.

- **Wertelistenverwaltung** – Es kann vorkommen, dass man bei bestimmten Eigenschaften keine manuelle Eingabe zulassen will, sondern nur eine Auswahl aus einer vorgegebenen Liste von möglichen Einträgen. Diese Listen von Einträgen werden in der Wertelistenverwaltung verwaltet. Jeder Werteliste wird ein eindeutiger Name zugeordnet und die Liste mit Werten gefüllt. Es können beliebige Wertelisten und Einträge in Wertelisten angelegt werden (max. Länge von 255 Zeichen). Die Wertelisten können von den Administratoren und den Projektadministratoren verwaltet werden.
- **Formularverwaltung** – die einzelnen Eingabeformulare müssen für die verschiedenen Objektarten definiert werden (damit das System weiß, welches Formular bei einem Objekt aufgerufen werden soll). Ein Formular besteht aus einem eindeutigen Formularnamen und aus Formularzeilen. Eine Formularzeile kann eine Überschrift sein, die verschieden formatiert werden kann. Es gibt Haupt- und Unterüberschriften und Formularzeilen mit einem normalen Text. Dazu gibt es noch Formularzeilen, die einem Parameter zugeordnet werden. In dieser Formularzeile wird der Inhalt des Parameters des aufgerufenen Objektes angezeigt. Wie dieser Inhalt angezeigt wird, ergibt sich durch das Parameterformat. Die Formulare können von den Administratoren und Projektadministratoren verwaltet werden.

Diese Punkte werden unter dem Menü **ADMINISTRATION** zusammengefasst.

5.3.2.2.3 Datenbearbeitung

Die Hauptaufgabe der Anwendung, die in dieser Arbeit beschrieben wird, ist die Verwaltung von „Vegetationsökologischen Daten“. Dazu müssen Objektarten angelegt werden können. Es müssen die dazugehörenden Objekte verwaltet werden. Weiters müssen die Daten (Stammdaten und bei den Objekten der Objektart

AUFNAHME auch die Vegetationsdaten) der einzelnen Objekte eingegeben, verändert und gelöscht werden können. Dazu gibt es folgende Möglichkeiten:

- **Objektverwaltung** – Hier können die Objekte der verschiedenen Objektarten verwaltet werden (eingeben, bearbeiten, löschen, Status setzen). Im Prinzip können die Objekte von allen registrierten Benutzern des Administrationsbereichs (Administratoren, Projektadministratoren, Mitarbeiter eines Projektes) angelegt werden. Dabei gibt es allerdings bestimmte Objekte, die nur für bestimmte Projekte gültig sind. Um diese Objekte verwalten zu können, muss der Benutzer dem jeweiligen Projekt zugeordnet sein. Die Mitarbeiter eines Projektes können nur ihre eigenen Objekte bearbeiten und löschen, solange sie den Status bearbeiten haben.
- **<Objektart>** – Unter diesen Punkt gibt es für jede mögliche Objektart ein Untermenü für die Verwaltung der Objekte einer Objektart. Der Name des Untermenüs ist der Name der jeweiligen Objektart. Zuerst wird ein Formular zur Suche eines Objektes der jeweiligen Objektart aufgerufen, das dann das Stammdatenformular und, bei den Objekten von der Objektart **AUFNAHME**, das Vegetationsdatenformular aufruft. In diesen Formularen können die Stamm- bzw. Vegetationsdaten verwaltet werden (eingeben, bearbeiten, löschen). Dieser Punkt wird vom System automatisch erstellt. Es werden für alle definierten Objektarten Untermenüs angeboten. Unter jedem Untermenü wird automatisch, beim Aufruf, die Suchseite generiert und die gefundenen Objekte sind automatisch der Link zum Stammdatenformular des Objektes. In diesem Stammdatenformular können die Daten des Objektes auch bearbeitet werden. Sollte für die Objektart kein Formular definiert worden sein, kommt die Meldung, es gibt keine Stammdaten zu dieser Objektart. Die Daten eines Objektes eines Projektes können von allen registrierten Benutzern des Administrationsbereichs, die dem Projekt zugeordnet sind, zu der das Objekt gehört, verwaltet werden. Die Mitarbeiter eines Projektes können nur die Daten verwalten, die von ihnen eingegeben wurden, solange sie den Status „**BEARBEITEN**“ haben. Die Projektadministratoren können die einzelnen Objekte ihres Projektes bearbeiten und, in der Objektverwaltung, mittels des Status „**FREIGABE**“ für die Öffentlichkeit freigeben und damit gegen weitere

Bearbeitung sperren. Dieser Status kann allerdings von den Projektadministratoren jederzeit wieder auf „**BEARBEITEN**“ zurückgesetzt werden, damit evt. Fehler von den Mitarbeitern des Projektes ausgebessert werden können.

- **Objektbeziehungen** – Verschiedene Objekte einer Objektart können mit Objekten einer anderen Objektart verknüpft werden. Diese Verknüpfungen gehören immer zu einem bestimmten **THEMA**. Diese Definitionen, d.h. welche Objekte einer Objektart mit einem Objekt einer anderen Objektart unter einem bestimmten **THEMA** verknüpft werden können, wird von den Administratoren in der Systemverwaltung unter den Punkten **IS-IN-Artverwaltung** und **IS-IN-Objektartverwaltung** definiert. Die Objektbeziehungen eines Projektes können von allen registrierten Benutzern des Administrationsbereichs, die dem Projekt zugeordnet sind, zu der das Objekt gehört, verwaltet werden. Die Mitarbeiter eines Projektes können nur die Objektbeziehungen verwalten, die von ihnen eingegeben wurden, solange sie den Status „**BEARBEITEN**“ haben. Die Projektadministratoren können die einzelnen Objektbeziehungen ihres Projektes bearbeiten und, in der Objektverwaltung, mittels des Status „**FREIGABE**“ für die Öffentlichkeit freigeben und damit gegen weitere Bearbeitung sperren. Dieser Status kann allerdings von den Projektadministratoren jederzeit wieder auf „**BEARBEITEN**“ zurückgesetzt werden, damit evt. Fehler von den Mitarbeitern des Projektes ausgebessert werden können.

Diese Punkte werden unter dem Menü **DATENBEARBEITUNG** zusammengefasst.

5.3.2.2.4 Auswertungen

Die Auswertung soll den registrierten Benutzer des Administrationsbereichs die Möglichkeit geben, sich die Daten des **BESTANDES** und dessen **AUFNAHMEN** schrittweise zusammenzustellen. Dabei stehen den Benutzer alle Daten (d.h. mit den Status „**FREIGABE**“ als auch „**BEARBEITEN**“) der Projekte zur Verfügung, in denen sie registriert sind. Im ersten Schritt sollen die Benutzer eine Objektart auswählen können. Im zweiten Schritt kann man eine oder mehrere Objekte der ausgewählten Objektart auswählen. Jedes Objekt kann Stammdaten enthalten. Eine Objektart (**AUFNAHME**) kann zusätzlich noch Vegetationsdaten besitzen. Der nächste Schritt

wird nur angezeigt, wenn es sich beim Objekt um eine Aufnahme handelt. Hier gibt es eine Auswahl, ob man nach den Vegetationsdaten oder nach den Stammdaten des Objektes suchen will.

In der Stammdatensuche kann man sich im nächsten Schritt die Stammdatenparameter zusammenstellen, über die man Information abrufen will. Danach gibt man das Ausgabeformat an in dem man das Ergebnis der Abfrage angezeigt haben will. Es gibt die Möglichkeit sich die Daten in HTML oder Excel ausgeben lassen.

Dieser Punkt kann unter dem Menü **AUSWERTUNG** aufgerufen werden.

5.3.2.2.4 Schnittstelle

Dieser Punkt dient den registrierten Benutzer des Administrationsbereichs für den IMPORT und EXPORT aller Daten (d.h. mit den Status „**FREIGABE**“ als auch „**BEARBEITEN**“) der Projekte im Bereich der **AUFNAHME**.

- **Download** – Über diesen Punkt kann man die gesamten Daten (d.h. Daten mit den Status „**FREIGABE**“ als auch „**BEARBEITEN**“) eines Projektes downloaden. Diese Funktion kann für ein vollständiges lokales Backup eines Projektes verwendet werden.
- **ORI-Upload** – Über diesen Punkt können die Vegetationsdaten aus einer Datei mit einem definierten Vegetationsdatenformat (ORI-Datei) in ein Projekt aufgeladen und über einen Zwischenschritt in den Vegetationsdatentabellen gespeichert werden (unter den Status „**BEARBEITEN**“).

Diese Punkte werden unter dem Menü **SCHNITTSTELLE** zusammengefasst.

5.3.2.2.4 Webservice

Bei einem Webservice handelt es sich um eine Maschinenabfrage – andere Anwendungen sollen die Möglichkeit haben, die Daten Moduls **AUFNAHME** automatisch abzufragen. Nur registrierten Anwendungen soll es möglich sein, alle Daten eines Projektes über das Webservice automatisch abzufragen. Nicht registrierte Anwendungen sollen nur Zugang zu den freigegeben Daten (Daten mit den Status „FREIGEgeben“) haben. Mehrere Anwendungen von iFlora, die von verschiedenen Institutionen verwendet werden, sollen ihre Daten untereinander mit dem Webservice austauschen können.

5.4 Botanische Namen – Modul BOTNAME

5.4.1 Datenmodell

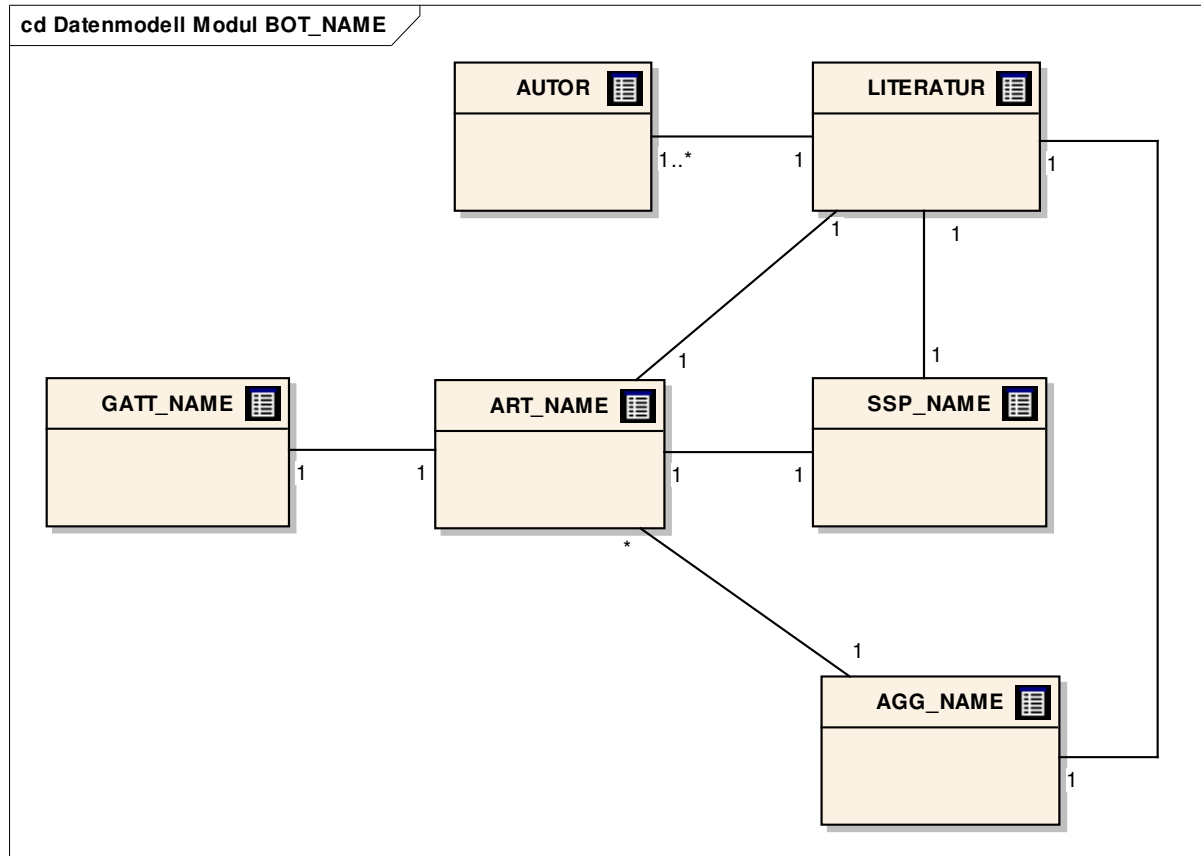


Abbildung 27 – Datenmodell iFlora Botanische Namen

Die Umsetzung des Moduls „Analyse Botanische Namen“ (IST Modell) entwickelt sich aus dem Kapitel „Analyse Botanische Namen“ (SOLL Modell).

Die Tabelle **ART_NAME** ist die zentrale Tabelle (siehe Abbildung 27). Hier werden die botanischen Namen, d.h. die Arten (**ART_NAME**), die Unterarten (**SSP_NAME**) oder die Aggregate (**AGG_NAME**) des Moduls „**BOTNAME**“ abgespeichert. Ein Artnamen besteht immer aus einem Gattungsnamen, der in der Tabelle **GATT_NAME** abgespeichert wird, und einem Art-Beinamen, der in der Tabelle **ART_NAME** gespeichert wird. Zusätzlich, wie oben erwähnt, ist diese Tabelle die zentrale Verbindung zu den anderen Modulen, weil jeder Teil der botanischen Namen immer auch einen Artnamen enthält, der in dieser Tabelle abgespeichert wird. Beim Prototyp, der dieser Arbeit zu Grunde liegt, ist das die Verbindung zu den

Vegetationsdaten im Modul **AUFNAHME**. Handelt es sich um eine Unterart, dann besteht der Name aus Gattungsname, Artnamen und, erweitert durch den Begriff „ssp.“ der Unterartname, letzterer wird in der Tabelle **SSP_NAME** gespeichert. In Zukunft ist geplant, dass in der Tabelle **SSP_NAME** noch die Verknüpfung mit der übergeordneten Art (Datensatz der Tabelle **ART_NAME**) abgespeichert wird. Ein Aggregat ist ein Hilfskonstrukt, das eine Gruppe von Kleinarten zusammenfasst um damit besser umgehen zu können. Er besteht aus der Gattung, den Art-Beinamen und der Erweiterung „agg.“. Intern ist die Tabelle **AGG_NAME** nur eine Beziehungstabelle, zu allen Kleinarten, die im Aggregat enthalten sind.

Jede Art, Unterart oder Aggregat wird irgendwann wissenschaftlich in der Literatur beschrieben. Dieses Literaturzitat wird in der Tabelle **LITERATUR** mit Jahreszahl und Verknüpfung auf die Autoren, die in der Untertabelle **AUTOR** verwaltet werden, gespeichert.

Da der Inhalt dieser Arbeit nur der Prototyp einer Internetanwendung ist, ist dieser Teil der Arbeit nicht zur Gänze im Prototyp umgesetzt. Statt des gesamten Moduls **BOTNAME** wird im Prototyp nur die Tabelle **AFN_ARTEN** im Modul **AUFNAHME** umgesetzt. Damit kann die Grundfunktionalität gezeigt werden.

5.4.2.1 Erweiterung des Datenmodells BOTNAME

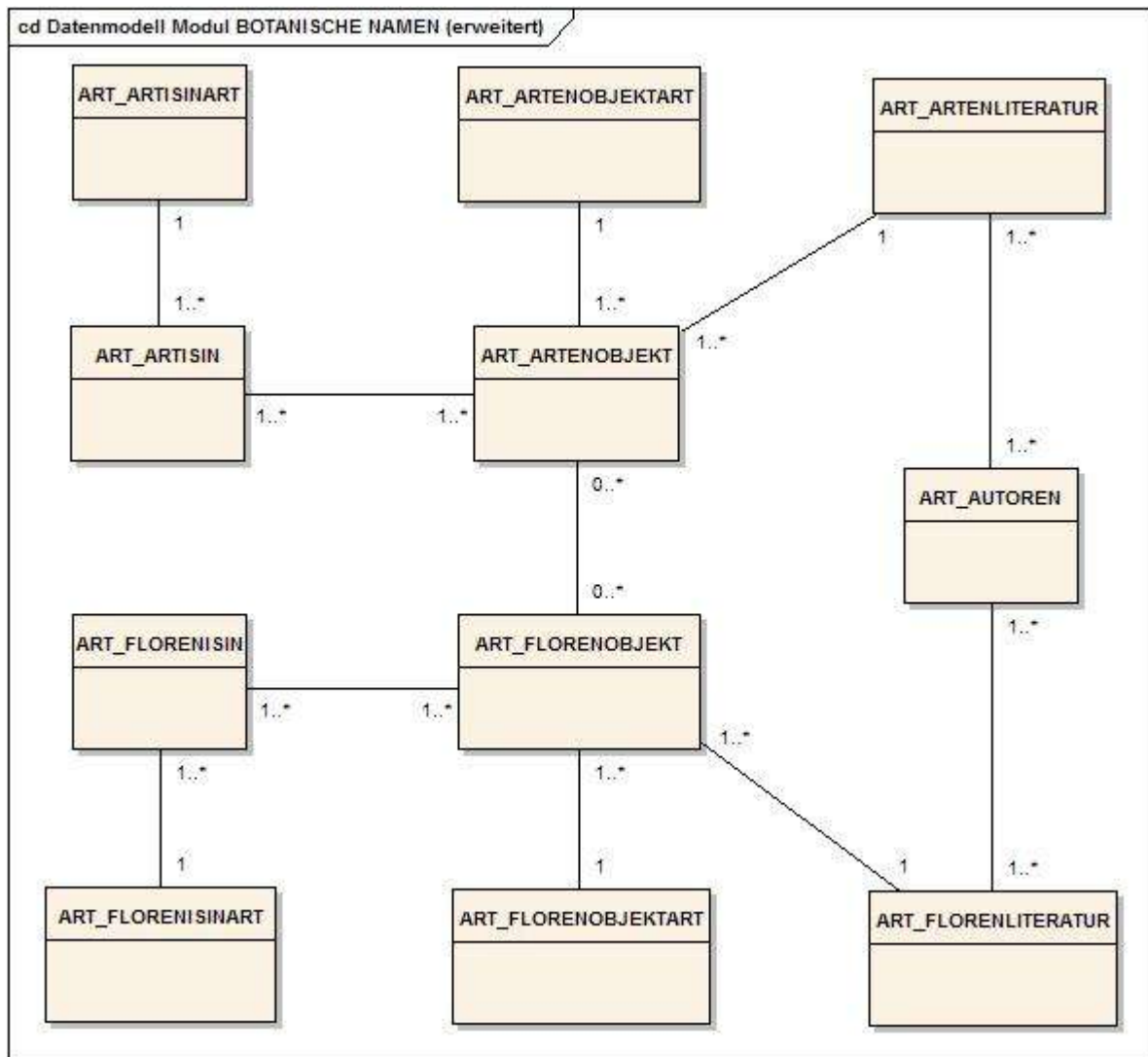


Abbildung 28 - Datenmodell Modul BOTANISCHE NAMEN (erweitert)

Im Datenmodell im vorigen Kapitel sind die Aufgaben die zeitliche Änderung der wissenschaftlichen Namen und deren Interpretation in den Florenwerken, die sich auch mit der Zeit ändern können, nicht berücksichtigt. Dieses Kapitel versucht einen ersten Ansatz für einen Lösungsversuch für diese Erweiterungen.

Die zentralen Tabellen sind **ART_FLORENOBJEKT** und **ART_ARTENOBJEKT** (siehe Abbildung 28). In beiden Tabellen sind die Teile der wissenschaftlichen Pflanzennamen als Objekt definiert. So ist der Gattungsname ein Objekt (in den Tabellen **ART_FLORENOBJEKT** bzw. **ART_ARTENOBJEKT**) der Objektart **GATTUNGSNAMEN** (in den Tabellen **ART_ARTENOBJEKTART** bzw. **ART_FLORENOBJEKTART**). Auch der Art-Beiname wird so definiert (Objektart:

ARTBEINAME), genauso wie die Unterart und das Aggregat. Der Vorteil dieses Konzeptes ist, das in den Objekttabellen auch die Familien, Ordnungen usw. definiert werden können. Die Verknüpfung der verschiedenen Objekte zu einer sinnvollen Entität wird über die IS-IN-Tabellen (**ART_ARTISIN** bzw. **ART_FLORENISIN**) realisiert. So kann der Name einer Gattung mit einem Art-Beinamen zu einer Art zusammengefasst werden. Über die IS-IN-Objektart Tabellen (**ART_ARTISINART** bzw. **ART_FLORENISINART**) ist definiert, das es sich bei der Objektkombination um eine Art handelt. Das Problem ist, das bei diesem Konzept immer nur 2 Objekte miteinander verknüpft werden können. Ein Aggregat kann so nicht umgesetzt werden, d.h. die IS-IN-Tabellen müssen eine Verknüpfung mit Objektclustern zulassen. Das könnte man realisieren, indem man auch eine Verknüpfung mit einer Objektkombination zulässt. Über die Literaturtabellen (**ART_ARTENLITERATUR** bzw. **ART_FLORENLITERATUR**) werden die Objekte mit einem Literaturzitat oder einer Erstbeschreibung und den jeweiligen Typusexemplar verknüpft. In diesem Zusammenhang ist zu erwähnen, das diese Verknüpfung v.a. auf ein Objektcluster, der z.B. eine Art repräsentiert, sinnvoll ist und in diesem Konzept noch nicht berücksichtigt ist.

Dieses Konzept soll nur ein Diskussionsvorschlag für weitere Überlegungen sein, die nicht mehr Teil dieser Arbeit sein können, da sie den Zeitrahmen sprengen würden.

5.4.3 Konzept Internetanwendung

Durch die Aufgabe, die Daten einer breiten Öffentlichkeit zur Verfügung zu stellen und den Projektmitarbeitern zu ermöglichen, ihre Daten über das Internet zu bearbeiten, kommt es zu einer Aufteilung der Anwendung in die Bereiche **iFloraPublic** und **iFloraAdmin**. In beiden Bereichen sollen die Daten der **BOTNAME** angezeigt werden.

Da der Inhalt dieser Arbeit nur der Prototyp einer Internetanwendung ist, ist dieses Kapitel nicht im Prototyp umgesetzt.

5.4.3.1 Öffentlicher Bereich

Im öffentlichen Bereich (iFloraPublic) werden die Arten mit ihren Literaturhinweisen angezeigt werden. Es gibt dazu eine Suchmaske, in der man nach Gattung, Artnamen

oder einer Kombination von beiden suchen kann und dann alle dazugehörenden Arten, Unterarten und Aggregate zurückbekommt. Zusätzlich kann man über eine Checkbox auswählen, ob man zu den Aggregaten alle dazugehörigen Arten ausgegeben haben will.

5.4.3.2 Administrationsbereich

5.4.3.2.1 Systemverwaltung

Im Administrationsbereich des Moduls **BOTNAME** sollen die Administratoren, wie beim Modul **AUFNAHME**, die Benutzer- und Metadaten administrieren können. Im Rahmen dieser Arbeit wurden aus Zeitgründen aber keine weiteren Überlegungen dazu angestellt.

5.4.3.2.2 Artenverwaltung

In das Formular der Artenverwaltung können neue Arten, Unterarten und Aggregate eingegeben und mit einem Literaturhinweis verknüpft werden. Sollte noch kein Literaturdatensatz angelegt sein, kommt man bei der Eingabe in der Artenverwaltung direkt auf das Formular der Literaturverwaltung und kann das neue Literaturzitat eingeben.

5.4.3.2.3 Literaturverwaltung

Im Formulare der Literaturverwaltung können die Literaturhinweise mit dem Erscheinungsjahr und Autoren eingegeben werden.

5.4.2.2.4 Webservice

Bei einem Webservice handelt es sich um eine Maschinenabfrage – andere Anwendungen sollen die Möglichkeit haben, die Daten des Moduls **BOTNAME** automatisch abzufragen. Registrierte Anwendungen sollen alle Daten eines Projektes über das Webservice automatisch abfragen können. Nicht registrierte Anwendungen sollen nur Zugriff auf freigegebene Daten haben. Mehrere Anwendungen von iFlora, die von verschiedenen Institutionen verwendet werden, soll ihre Daten untereinander mit den Webservice austauschen können. So könnte man die Verwaltung der Botanischen Namen (Modul **BOTNAME**) einem botanischen Institut überlassen und dann von dort immer die aktuelle Datenversion abfragen und in die eigene Anwendung von iFlora übernehmen.

5.5 Administration – Modul ADMINISTRATION

5.4.1 Datenmodell

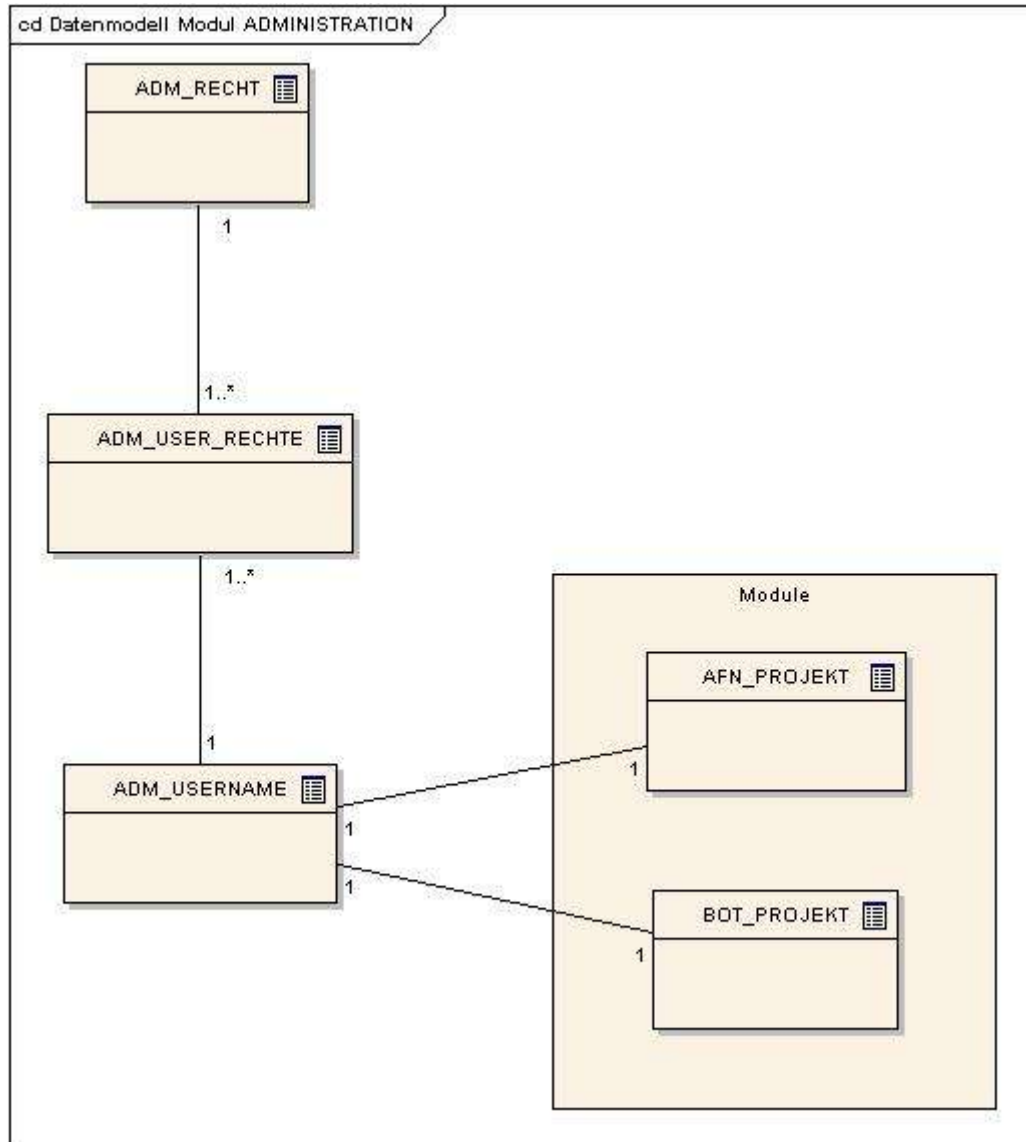


Abbildung 29 – Datenmodell iFlora Administration

Die Umsetzung des Moduls „Administration“ (IST Modell) entwickelt sich aus dem Kapitel „Analyse Administration“ (SOLL Modell).

Die Tabelle **ADM_USERNAME** ist die zentrale Tabelle (siehe Abbildung 29). Hier werden die Namen und Passwörter der einzelnen Benutzer gespeichert. Zusätzlich werden hier die Benutzer bei den verschiedenen Projekte registriert. Die Registrierung erfolgt über eine Zuweisung eines Projektes aus den verschiedenen Modulen (im Prototyp kann einem Benutzer nur ein Projekt aus dem Modul

AUFNAHME zugeordnet werden). In jedem Modul gibt es dazu eine Tabelle zur Verwaltung der Projekte – so verwaltet die Tabelle **AFN_PROJEKT** die Projekte des Moduls AUFNAHME und die Tabelle **BOT_PROJEKT** die Projekte des Moduls BOTNAME.

Jeden Benutzer werden Rechte zugewiesen, dies wird über die Tabelle **ADM_USER_RECHTE** umgesetzt. Die möglichen Rechte sind in der Tabelle **ADM_RECHT** definiert.

Im Prototyp ist nur eine vereinfachte Benutzerrechteverwaltung implementiert.

5.4.3.1 Öffentlicher Bereich

Im öffentlichen Bereich (iFloraPublic) werden vom Modul keine Daten angezeigt.

5.3.2.2 Administrationsbereich

5.3.2.2.1 Systemverwaltung

Das Modul **ADMINISTRATION** hat keinen eigenen Benutzerbereich, alle Daten werden bei den anderen Modulen verwaltet. Hier können die Administratoren die Benutzer- und Metadaten für das Modul ADMINISTRATION administrieren. Es gibt hier folgende Möglichkeiten, die bei den einzelnen Modulen integriert sind (siehe v.a. das Modul **AUFNAHME**):

- **Benutzerverwaltung** – hier können neue Benutzer angelegt, vorhandene Benutzer verwaltet und alte Benutzer deaktiviert werden. Zusätzlich können hier die Benutzerrechte verwaltet und die Benutzer bestimmten Projekten zugeordnet werden. Diese Funktionalität kann nur von Administratoren und Projektadministratoren (nur für Benutzer seines Projektes) ausgeführt werden, so macht es Sinn, wenn die Projektadministratoren für Benutzer, die ihrem Projekt zugeordnet sind, neue Passwörter vergeben können.
- **Projektverwaltung** – hier können Projekte angelegt und verwaltet werden. Auch diese Funktionalitäten können nur Administratoren durchführen.

6 Umsetzung

Es gibt für die Anwendung iFlora einen öffentlichen Teil (iFloraPublic) und einen Teil für die Administration der Daten (iFloraAdmin). Der Öffentliche Teil ist für alle

Benutzer des Internets frei zugänglich. Die Daten können von allen Benutzern (eingeschränkt) ausgewertet werden. Im Teil für die Administration, können, von registrierten Benutzern, die Daten eingegeben, bearbeitet und (ohne Einschränkung) ausgewertet werden.

Im Rahmen der Diplomarbeit kann, aus zeitlichen Gründen, nur ein Prototyp, d.h. eine Beispielanwendung, dessen Funktionen nicht vollständig ausprogrammiert sind, umgesetzt werden.

6.1 iFlora Öffentlicher Bereich (iFloraPublic)

Jede Seite des öffentlichen Bereichs hat eine Hauptseite mit folgenden Komponenten (siehe Abbildung 30):

- Kopfzeile: Die Kopfzeile ist die oberste Zeile im Browserfenster. Es enthält die weiße Überschrift „iFlora Öffentlicher Bereich“ auf grünem Hintergrund. Dazu befindet sich in der rechten Ecke eine Information, in welcher Version (R0.6) man sich gerade befindet und mit welcher Datenbank (der Prototyp ist durch „test“ gekennzeichnet) man verbunden ist.
- Menüleiste: Die Menüleiste befindet sich im linken Bereich des Browserfensters. Hier befindet sich ein Link für den Aufruf der Administrationsversion von iFlora und der Auswahlbaum zu Aufrufen der verschiedenen Funktionen der Anwendung iFloraPublic. Für den Auswahlbaum wurde die Anwendung „dTree 2.05“ verwendet (<http://www.destroydrop.com/javascripts/tree/>).
- Statuszeile: Die Statuszeile ist die unterste Zeile im Browserfenster. Sie enthält links den Namen der Anwendung und den Namen des Benutzers. Da es im Öffentlichen Bereich keine registrierten Benutzer gibt, steht hier immer „Public“. Weiters befindet sich rechts ein Copyright Hinweis.
- Rahmen für die dynamischen Webseiten: Über den Auswahlbaum können verschiedene Funktionen der Anwendung iFlora aufgerufen werden. Die Seiten dieser Funktionen öffnen sich in diesem Rahmen.

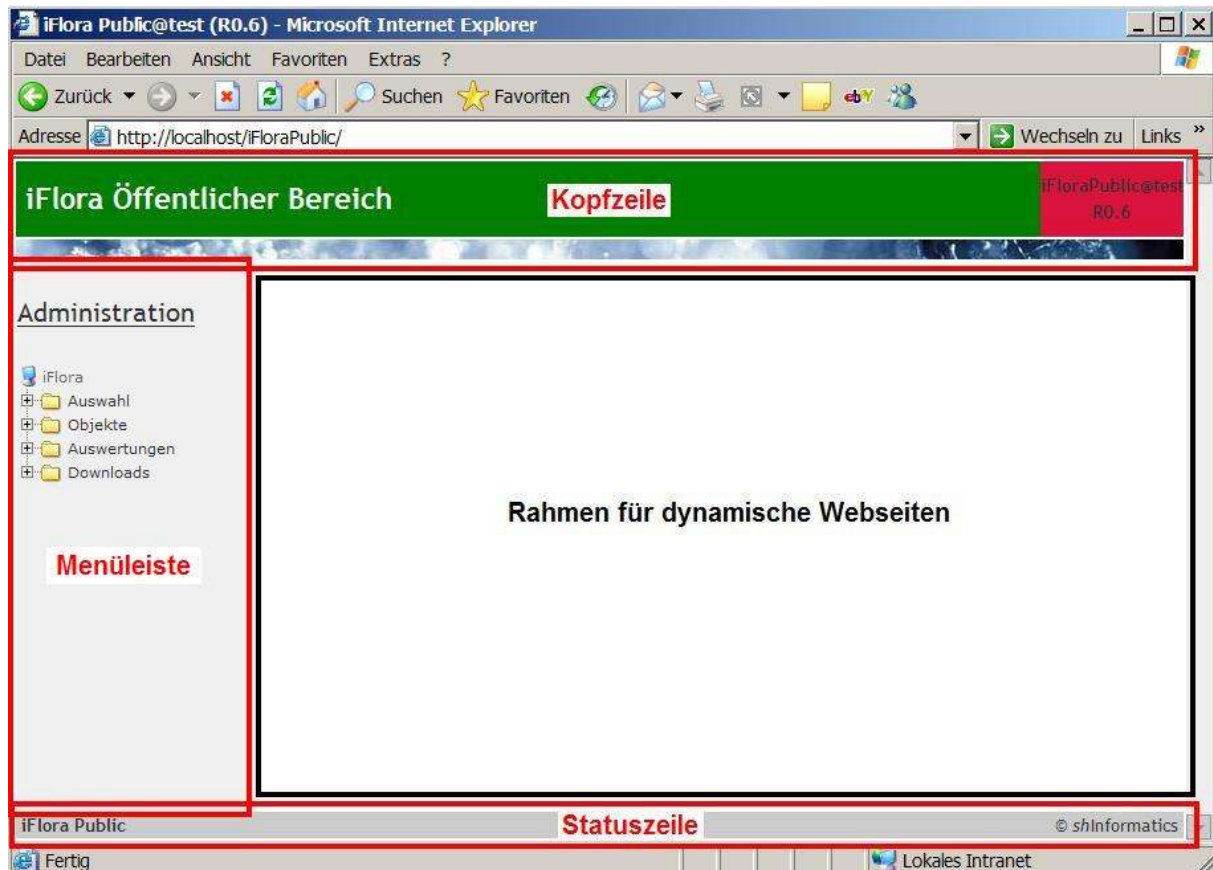


Abbildung 30 – iFlora Öffentlicher Bereich - Startseite mit einzelnen Elementen

6.1.1 Allgemeine Informationen

Die Dateien der Anwendung befinden sich in einer Verzeichnisstruktur (siehe 6.1.1.1 Verzeichnisstruktur).

In den meisten Anwendungen befinden sich Programmteile, die nicht einem bestimmten Formular zugeordnet werden, sondern die von mehreren Formularen verwendet werden (siehe 6.1.1.2 Libraries).

Für den Prototyp wurde das System XAMPP in der Version „XAMPP (basic package) 1.4.15“ (<http://www.apachefriends.org/de/xampp.html>) verwendet. Dieses System enthält u.a. folgende Komponenten, die für die Prototypen iFloraPublic und iFloraAdmin verwendet werden:

- Apache 2.0.54
- OpenSSL 0.9.8
- MySQL 4.1.13
- PHP 5.0.4 + PEAR

- eAccelerator 0.9.3 (PHP5)
- SQLite 2.8.15
- phpMyAdmin 2.6.3-pl1
- ADODB 4.65

6.1.1.1 Verzeichnisstruktur

Die verschiedenen Dateien der Anwendung iFloraPublic befinden sich in den folgenden Verzeichnissen (siehe Abbildung 31).

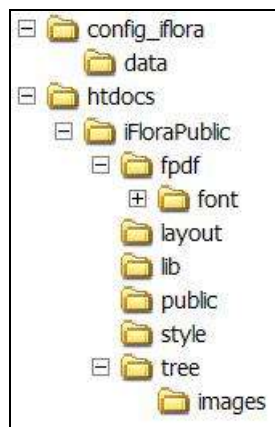


Abbildung 31 – iFlora Öffentlicher Bereich - Verzeichnisstruktur der Anwendung

- „/config_iflora/“: In diesem Verzeichnis befinden sich heikle Konfigurationsdaten.
- „/htdocs/“: Alle Unterverzeichnisse in diesem Verzeichnis können im (unverschlüsselten) Internet-Pfad des Servers (<http://...>) als (gültiger) Link eingegeben werden.
- „/htdocs/iFloraPublic/Public/“: Hier befinden sich alle Sourcecode-Dateien der verschiedenen Formulare (Funktionen von iFloraPublic). Diese Formulare werden über den **Auswahlbaum** der „**Menüleiste**“ aufgerufen und im „**Rahmen für dynamische Webseiten**“ angezeigt.
- „/htdocs/iFloraPublic/fpdf/“: Bei verschiedenen Formularen gibt es die Möglichkeit, die Daten in Form einer PDF-Datei downzuloaden. Die dafür notwendigen Tools werden von der Anwendung FPDF bereitgestellt. Es wird die Version FPDF 1.53 (<http://www.fpdf.org/>) verwendet. Die Dateien dieser Anwendung befinden sich alle in diesem Verzeichnis.

- „/htdocs/iFloraPublic/layout/“: Für das Layout der Startseite werden einige Grafiken gebraucht, diese befinden sich in diesem Verzeichnis.
- „/htdocs/iFloraPublic/lib/“: Einige Programmbereiche werden von mehreren Formularen gemeinsam genutzt. Diese Sourcecode-Datei werden in diesem Verzeichnis gespeichert.
- „/htdocs/iFloraPublic/“: Dieses Verzeichnis enthält alle Dateien der Anwendung **iFloraPublic** und wird über den Pfad <http://localhost/iFloraPublic> aufgerufen. Über die Datei „index.php“, die automatisch beim Aufruf dieses Verzeichnisses ausgeführt wird, meldet sich Anwendung.
- „/htdocs/iFloraPublic/style/“: Der Style der verschiedenen Seiten wird über CSS-Dateien definiert. Diese CSS-Dateien befinden sich in diesem Verzeichnis.
- „/htdocs/iFloraPublic/tree/“: Für den Auswahlbaum in der Menüleiste wird die Anwendung „dTree 2.05“ verwendet. Alle Dateien dieser Anwendung befinden sich in diesem Verzeichnis.

6.1.1.2 Libraries

Für die Anwendung iFloraPublic werden bestimmte Teile des Programms für mehrere Funktionen verwendet und in Bibliotheken (Libraries) zusammengefasst.

- „DBManager.php“: Diese Bibliothek enthält alle die allgemeinen Programmteile der Anwendung iFloraPublic für die Abfragen der Datenbank.
- „LAYOUTManager.php“: Diese Bibliothek enthält die allgemeinen Layout-Teile (z.B. HTML-Header, HTML-Footer, usw.) der verschiedenen dynamischen Webseiten.
- „VIEWManager.php“: Enthält verschiedene Funktionen zum Aufbau der Datenbereich der dynamischen Webseiten.

6.1.2 Startseite

Nach der Eingabe des Links der Anwendung „iFlora Öffentlicher Bereich“ im Internet Browser (beim Prototyp: <http://localhost/iFloraPublic/>) öffnet sich die Startseite, wie unten (Abbildung 32) dargestellt.

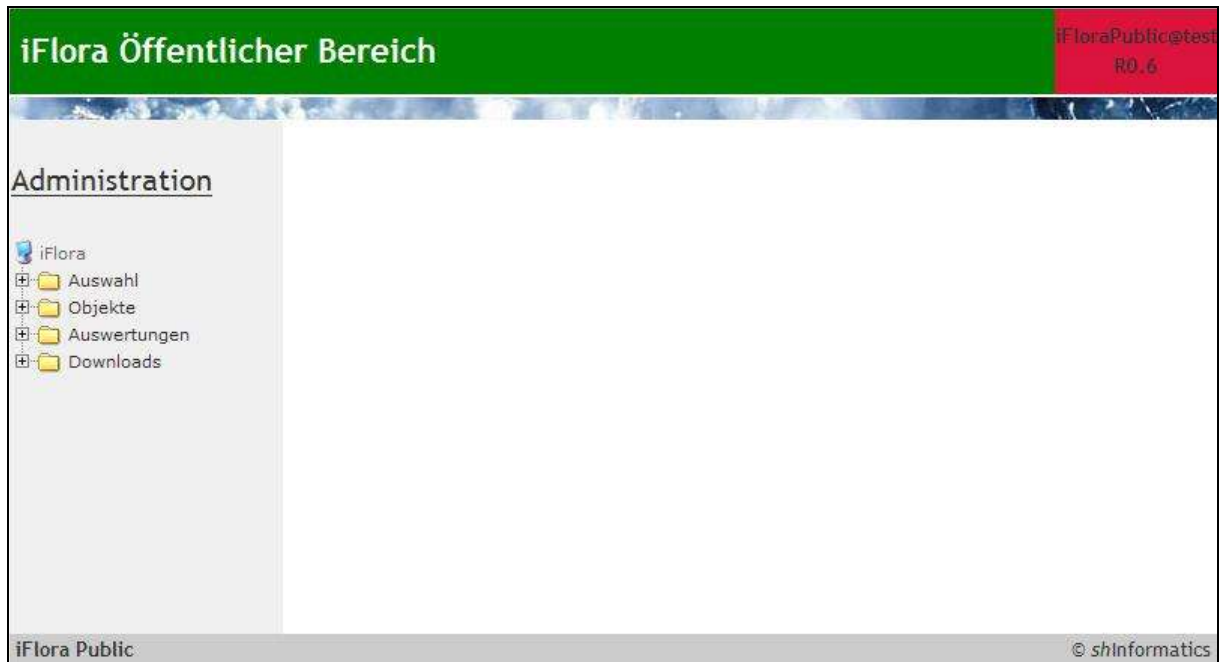


Abbildung 32 – iFlora Öffentlichlicher Bereich - Startseite

Über die Menüleiste auf der linken Seite kann man die verschiedenen Funktionen der Anwendung „iFloraPublic“ aufrufen. Sie enthält außerdem einen Link auf die Administrationsversion von iFlora.

6.1.2.1 Codierung

Diese Startseite ist in der Datei „/iFloraPublic/index.php“ codiert (Sourcecode siehe „9.3.2.2.1 Startseite – Rahmen mit Menü“). Beim Start wird im „Rahmen für dynamische Webseiten“ eine leere HTML-Datei angezeigt („/iFloraPublic/work.html“ – Sourcecode siehe „9.3.2.2.2 Startseite – Work (im Rahmen)“).

6.1.3 Auswahl

Über den Menüpunkt „Auswahl“ kann man einen Bestand und innerhalb eines Bestandes einer der Aufnahmen über folgenden Weg auswählen (siehe Abbildung 33):

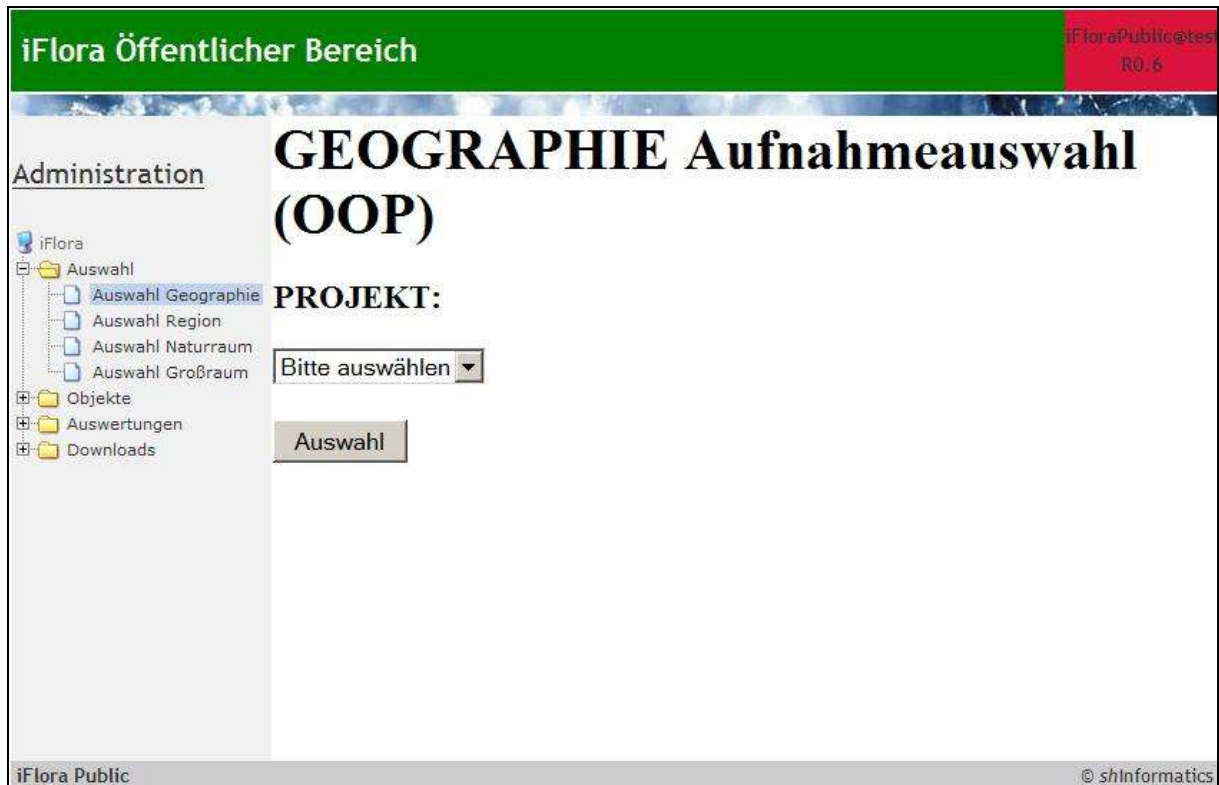


Abbildung 33 – iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Geographie

- **Menüpunkt:** Auswahl Geographie – enthält folgende Auswahl (das wurde über eine Abfolge von verschiedenen Formularen realisiert):
 - Projekt

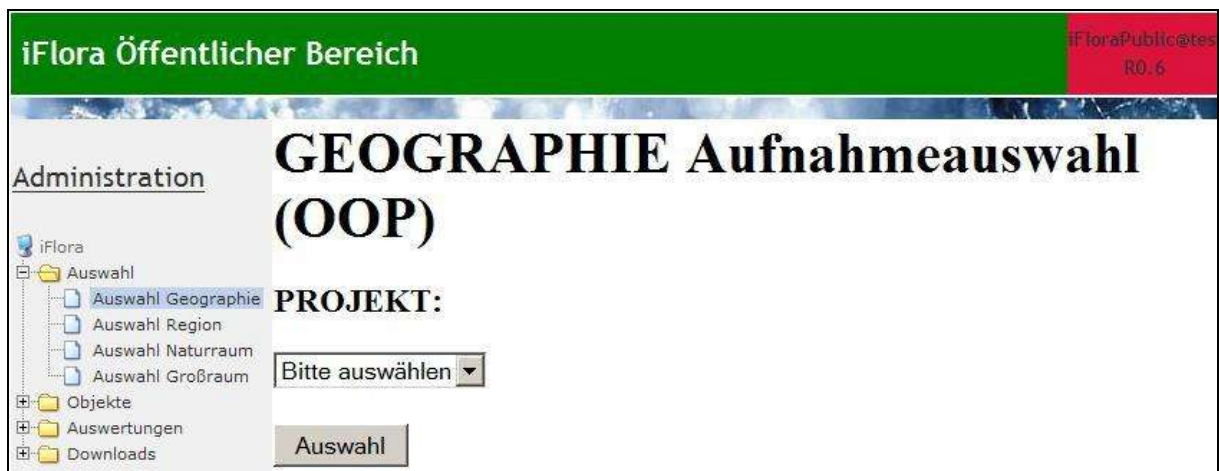


Abbildung 34 – iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Geographie - Projektauswahl

- Bundesland (siehe Grafik unten)

iFlora Öffentlichlicher Bereich

iFloraPublic@tes
R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

GEOGRAPHIE Aufnahmeauswahl (OOP)

PROJEKT: MOOR

BUNDESLAND:

Bitte auswählen ▼

Auswahl

Abbildung 35 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Geographie - Bundeslandauswahl

- Bezirk

iFlora Öffentlichlicher Bereich

iFloraPublic@test
R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

GEOGRAPHIE Aufnahmeauswahl (OOP)

PROJEKT: MOOR

BUNDESLAND: Kärnten

BEZIRK:

Bitte auswählen ▼

Auswahl

Abbildung 36 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Geographie - Bezirksauswahl

- Gemeinde

iFlora Öffentlicher Bereich iFloraPublic@test
R0.6

GEOGRAPHIE Aufnahmeauswahl (OOP)

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

PROJEKT: MOOR
BUNDESLAND: Kärnten
BEZIRK: Sankt Veit an der Glan
GEMEINDE:

Bitte auswählen ▼

Auswahl

Abbildung 37 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Geographie - Gemeindeauswahl

- Bestand

iFlora Öffentlicher Bereich iFloraPublic@test
R0.6

GEOGRAPHIE Aufnahmeauswahl (OOP)

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

PROJEKT: MOOR
BUNDESLAND: Kärnten
BEZIRK: Sankt Veit an der Glan
GEMEINDE: Liebenfels

BESTAND:

Bitte auswählen ▼

Auswahl

Abbildung 38 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Geographie - Bestandsauswahl

- Aufnahme

iFlora Öffentlichlicher Bereich iFloraPublic@test
R0.6

GEOGRAPHIE Aufnahmeauswahl (OOP)

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

PROJEKT: MOOR
BUNDESLAND: Kärnten
BEZIRK: Sankt Veit an der Glan
GEMEINDE: Liebenfels
BESTAND: Freundsamer Moos E

AUFNAHME:

Bitte auswählen ▼

Auswahl

[Bestand Stammdaten](#)

iFlora Public © shinformatics

Abbildung 39 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Geographie - Aufnahmeauswahl

- **Menüpunkt:** Auswahl Region – enthält folgende Auswahl (auch über eine Abfolge von verschiedenen Formularen realisiert):
 - Projekt

iFlora Öffentlichlicher Bereich iFloraPublic@test
R0.6

REGION Aufnahmeauswahl (OOP)

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

PROJEKT:

Bitte auswählen ▼

Auswahl

iFlora Public © shinformatics

Abbildung 40 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Region - Projektauswahl

- Region

iFlora Öffentlicher Bereich iFloraPublic@test R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

REGION Aufnahmeauswahl (OOP)

PROJEKT: MOOR

REGION:

Bitte auswählen

Auswahl

Abbildung 41 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Region - Regionsauswahl

- Bestand

iFlora Öffentlicher Bereich iFloraPublic@test R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

REGION Aufnahmeauswahl (OOP)

PROJEKT: MOOR

REGION: Hohe Tauern

BESTAND:

Bitte auswählen

Auswahl

Abbildung 42 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Region - Bestandsauswahl

- Aufnahme

iFlora Öffentlicher Bereich iFloraPublic@test R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region**
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

REGION Aufnahmeauswahl (OOP)

PROJEKT: MOOR
REGION: Hohe Tauern
BESTAND: Fuscher Rotmoos

AUFNAHME:

Bitte auswählen ▼

Auswahl

[Bestand Stammdaten](#)

Abbildung 43 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Region - Aufnahmeauswahl

- **Menüpunkt: Auswahl Naturraum** – enthält folgende Auswahl (auch über eine Abfolge von verschiedenen Formularen realisiert):
 - Projekt

iFlora Öffentlicher Bereich iFloraPublic@test R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum**
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

NATURRAUM Aufnahmeauswahl (OOP)

PROJEKT:

Bitte auswählen ▼

Auswahl

[Bestand Stammdaten](#)

Abbildung 44 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Naturraum - Projektauswahl

- Naturraum



Abbildung 45 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Naturraum - Naturraumauswahl

- Bestand



Abbildung 46 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Naturraum - Bestandauswahl

- Aufnahme

iFlora Öffentlichlicher Bereich iFloraPublic@test
R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum**
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

NATURRAUM Aufnahmeauswahl (OOP)

PROJEKT: MOOR
NATURRAUM: Hochschwab
BESTAND: Moor b.Grünen See

AUFNAHME:

Bitte auswählen ▼

[Bestand Stammdaten](#)

Abbildung 47 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Naturraum - Aufnahmeauswahl

- **Menüpunkt:** Auswahl Großraum – enthält folgende Auswahl (auch über eine Abfolge von verschiedenen Formularen realisiert):
 - Projekt

iFlora Öffentlichlicher Bereich iFloraPublic@test
R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum**
 - Objekte
 - Auswertungen
 - Downloads

GROSSRAUM Aufnahmeauswahl (OOP)

PROJEKT:

Bitte auswählen ▼

Abbildung 48 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Großraum - Projektauswahl

- Großraum

iFlora Öffentlicher Bereich iFloraPublic@test
R0.6

GROSSRAUM Aufnahmeauswahl (OOP)

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

PROJEKT: MOOR

GROSSRAUM:

Bitte auswählen

Auswahl

Abbildung 49 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Großraum - Großraumauswahl

- Bestand

iFlora Öffentlicher Bereich iFloraPublic@test
R0.6

GROSSRAUM Aufnahmeauswahl (OOP)

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

PROJEKT: MOOR

GROSSRAUM: Nordalpen

BESTAND:

Bitte auswählen

Auswahl

Abbildung 50 - iFlora Öffentlicher Bereich – Aufnahmeauswahl Großraum - Bestandauswahl

- Aufnahme



Abbildung 51 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Großraum - Aufnahmeauswahl

- **Ansicht:** „Aufnahme mit Stammdatenlink“ – dieser Punkt ist für alle Punkte des Menüs „Auswahl“ gleich!



Abbildung 52 - iFlora Öffentlichlicher Bereich – Aufnahmeauswahl Großraum - Stammdatenlink

- **Ansicht:** „Bestand Stammdaten“ – dieser Punkt ist für alle Punkte des Menüs „Auswahl“ gleich!

iFlora Öffentlichlicher Bereich iFloraPublic@test
R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

Meteorologie

S112 - Jahresniederschlag	über 2000	?
S113 - Niederschlag über einen Tag	über 130	?
S114 - durchschnittliche Jännertemperatur	-2 bis -6	?
S115 - durchschnittliche Julitemperatur	10 bis 15	?
S121 - Klimatyp	k. a.	?

Moor

M100 - Moortyp	Quellmoor	?
M101 - pH Wert	kalkreich	?
M102 - Trophie	mesotroph	?

Aufnahmen

- AF: Bärenmoos 1
- AF: Bärenmoos 2
- AF: Bärenmoos 3
- AF: Bärenmoos 4

DOWNLOAD: Excel PDF Text

iFlora Public © shInformatics

Abbildung 53 - iFlora Öffentlichlicher Bereich – Stammdatenformular Bestand

Eine Besonderheit ist im Stammdatenformular der Objekte von der Objektart „BESTAND“ ist die Liste der Objekte der Objektart „AUFNAHME“, die dem Bestand zugeordnet wurden.

- **Ansicht: „Bestand Stammdaten“** – es gibt Stammdaten, die einem Bestand zugeordnet werden und Stammdaten, die innerhalb eines Bestandes jeder Aufnahme zugeordnet werden. Für die Stammdaten einer Aufnahme wird dieses Formular verwendet. Sie wird über die Aufnahme-Links („AF: ...“) im Formular „Bestand Stammdaten“ aufgerufen (diese Vorgehensweise ist für alle Punkte des Menüs „Auswahl“ gleich – siehe Abbildungen 53 und 54).

iFlora Öffentlicher Bereich iFloraPublic@test
R0.6

Administration

- iFlora
 - Auswahl
 - Auswahl Geographie
 - Auswahl Region
 - Auswahl Naturraum
 - Auswahl Großraum
 - Objekte
 - Auswertungen
 - Downloads

AF: Bärenmoos 1

Stammdaten

Diverses

S117 - Assoziation	Caricetum paniculatae	?
S118 - Rasse	k.a.	?
S119 - Subassoziation	caricetosum davallianae	?
S120 - Variante	Variante von Cratoneuron filicinum	?

Moor

M103 - Facies	k.a.	?
M104 - Phase	k.a.	?

Bestand

Bärenmoos

DOWNLOAD: [Excel](#) [PDF](#) [Text](#)

VERWEIS (Link): [Vegetationsdaten](#)

iFlora Public © shInformatics

Abbildung 54 - iFlora Öffentlicher Bereich – Stammdatenformular Aufnahme

- **Download:** „Stammdaten - Excel“ – Der Inhalt des Formulars Stammdaten wird über den Link „Excel“ in einer Datei im Excelformat zum Download angeboten.
- **Download:** „Stammdaten - PDF“ – Der Inhalt des Formulars Stammdaten wird über den Link „PDF“ in einer Datei im PDF-Format zum Download angeboten.
- **Download:** „Stammdaten - Text“ – Der Inhalt des Formulars Stammdaten wird über den Link „Text“ in einer Datei im ASCII-Format zum Download angeboten.
- **Ansicht:** „Vegetationsdaten der Aufnahme“ – hier werden die Daten der Vegetationsdaten einer Aufnahme angezeigt (siehe Abbildung 55). Diese Seite wird aus dem Formular „Aufnahme Stammdaten“ über den Button „Vegetationsdaten“ aufgerufen (diese Vorgehensweise ist für alle Punkte des Menüs „Auswahl“ gleich).

AF: Bärenmoos 1 Aufnahme Vegetationsdaten	
Viola biflora L.	2
Valeriana dioica L.	+
Trollius europaeus L.	+
Ranunculus aconitifolius L.	+
Philonotis calcarea (B. & S.) Schimp.	1
Eriophorum latifolium Hoppe	+
Equisetum telmateia Ehrh.	+
Equisetum palustre L.	+
Juncus alpinoarticulatus Chaix	+
Gymnadenia conopsea (L.) R. Br.	+
Filipendula ulmaria (L.) Maxim.	1
Carex paniculata L.	4
Carex panicea L.	+

Abbildung 55 - iFlora Öffentlichlicher Bereich – Vegetationsdatenformular

- **Download:** „Aufnahmedaten - Excel“ – Der Inhalt des Formulars Vegetationsaufnahme wird über den Link „Excel“ in einer Datei im Excelformat zum Download angeboten.
- **Download:** „Aufnahmedaten - PDF“ – Der Inhalt des Formulars Vegetationsaufnahme wird über den Link „PDF“ in einer Datei im PDF-Format zum Download angeboten.
- **Download:** „Aufnahmedaten - Text“ – Der Inhalt des Formulars Vegetationsaufnahme wird über den Link „Text“ in einer Datei im ASCII-Format zum Download angeboten.
- **Download:** „Aufnahmedaten - ORI-Datei“ – Der Inhalt des Formulars Vegetationsaufnahme wird über den Link „ORI-Datei“ in einer ASCII-Datei im ORI-Format zum Download angeboten.

6.1.3.1 Codierung

- **Menüpunkt:** Auswahl Geographie – der Sourcecode befindet sich in der Datei „/iFloraPublic/public/OOPSelGeographie.php“ (siehe Kapitel 9.3.2.2.3 Controller – „Auswahl Geographie“).

- **Menüpunkt:** Auswahl Region – der Sourcecode befindet sich in der Datei „iFloraPublic/public/OOPSelRegion.php“ (siehe Kapitel 9.3.2.2.4 Controller – „Auswahl Region“).
- **Menüpunkt:** Auswahl Naturraum – der Sourcecode befindet sich in der Datei „iFloraPublic/public/OOPSelNaturraum.php“ (siehe Kapitel 9.3.2.2.5 Controller – „Auswahl Naturraum“).
- **Menüpunkt:** Auswahl Großraum – der Sourcecode befindet sich in der Datei „iFloraPublic/public/OOPSelGrossraum.php“ (siehe Kapitel 9.3.2.2.6 Controller – „Auswahl Großraum“).
- **Ansicht:** „Aufnahmefläche mit Stammdatenlink“ – der Sourcecode ist befindet sich in den einzelnen Dateien der obengenannten Menüpunkte, unterscheidet sich aber nicht im Aussehen und Funktion!
- **Ansicht:** „Aufnahme Stammdaten“ – der Sourcecode befindet sich in der Datei „iFloraPublic/public/OOPFormStammdata.php“ (siehe Kapitel 9.3.2.2.8 Controller – „Aufnahme Stammdaten“).
- **Download:** „Stammdaten - Excel“ – der Download von Stammdaten (Vegetationsaufnahme) im Excel-Format wird in der Datei „iFloraPublic/public/OOPExcelStammdata.php“ codiert (siehe Kapitel 9.3.2.2.11 Controller – „Download Stammdaten Excel“).
- **Download:** „Stammdaten - PDF“ – der Download von Stammdaten (Vegetationsaufnahme) im PDF-Format wird in der Datei „iFloraPublic/public/OOPpdfStammdata.php“ codiert (siehe Kapitel 9.3.2.2.12 Controller – „Download Stammdaten PDF“).
- **Download:** „Stammdaten - Text“ – der Download von Stammdaten (Vegetationsaufnahme) im ASCII-Format wird in der Datei „iFloraPublic/public/OOPASCIISTammdata.php“ codiert (siehe Kapitel 9.3.2.2.10 Controller – „Download Stammdaten ASCII“).
- **Ansicht:** „Vegetationsdaten der Aufnahme“ – diese Ansicht ist in der Datei „iFloraPublic/public/OOPFormAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.9 Controller – „Vegetationsdat. der Aufnahmefl.“).
- **Download:** „Aufnahmedaten - Excel“ – der Download von Aufnahmedaten (Vegetationsaufnahme) im Excel-Format wird in der Datei

„/iFloraPublic/public/OOPExcelAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.15 Controller – „Download Aufn. Excel“).

- **Download:** „Aufnahmedaten - PDF“ – der Download von Aufnahmedaten (Vegetationsaufnahme) im PDF-Format wird in der Datei „/iFloraPublic/public/OOPpdfAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.16 Controller – „Download Aufn. PDF“).
- **Download:** „Aufnahmedaten - Text“ – der Download von Aufnahmedaten (Vegetationsaufnahme) im ASCII-Format wird in der Datei „/iFloraPublic/public/OOPASCIIAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.13 Controller – „Download Aufn. ASCII“).
- **Download:** „Aufnahmedaten - ORI-Datei“ – der Download von Aufnahmedaten (Vegetationsaufnahme) im ORI-Format (ist eine spezielle Form des Textformats) wird in der Datei „/iFloraPublic/public/OOPASCIIAufnahmedataORI.php“ codiert (siehe Kapitel 9.3.2.2.14 Controller – „Download Aufn. ORI-Datei“).

6.1.4 Objektansicht

Über den Menüpunkt „Objekte“ kann man die Daten aller Objekte abrufen. Es werden in diesem Menü automatisch alle möglichen Arten von Objekten angezeigt (siehe Abbildung 56).

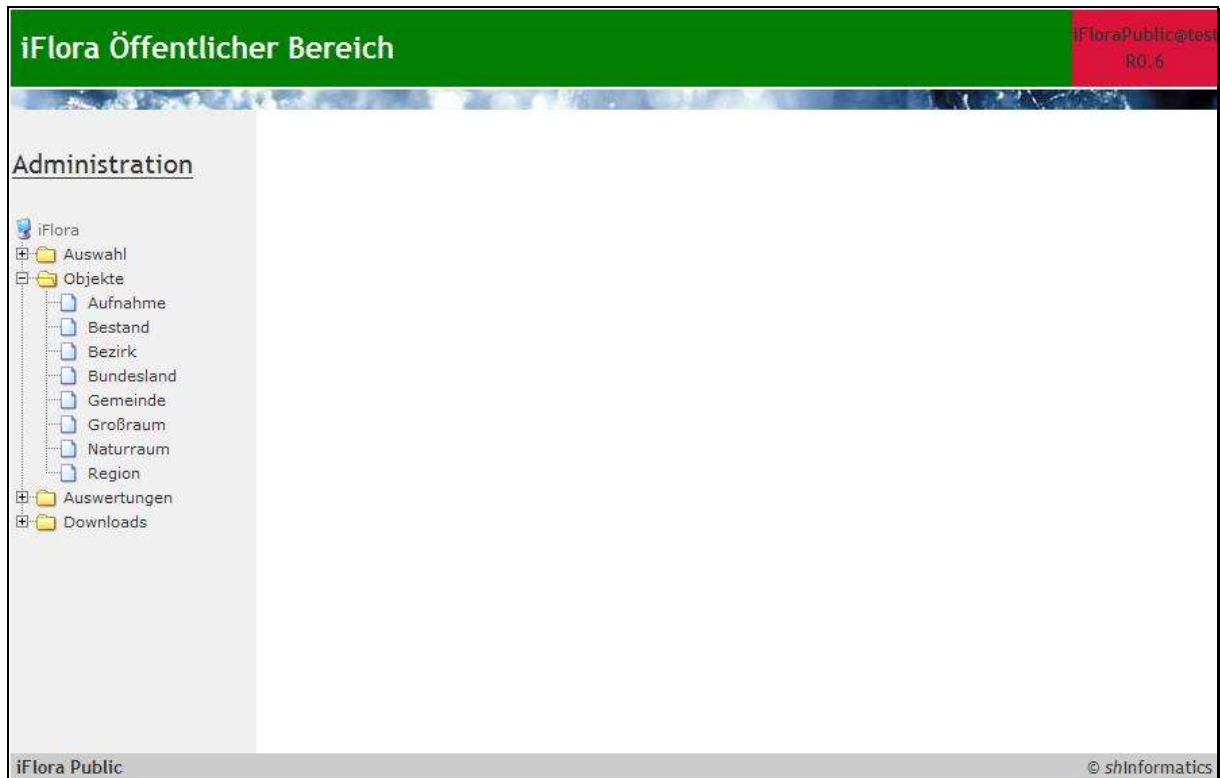


Abbildung 56 - iFlora Öffentlichlicher Bereich – Objekte

Über diese Auswahl kommt man, bei jedem Objekt, zuerst zu der Projektauswahl (da jedes Objekt jeder Objektart immer einem Projekt zugeordnet ist (siehe Abbildung 57)).



Abbildung 57 - iFlora Öffentlicher Bereich – Objekte Bestand

Nach der Projektauswahl kommt man zur Objektauswahl (siehe Abbildung 58 unten – am Beispiel des Bestandes).



Abbildung 58 - iFlora Öffentlicher Bereich – Objekte Bestandauswahl

Jedes Objekt in der Liste ist ein Link auf das Stammdatenformular des Objektes (siehe Abbildung 59 unten).

iFlora Öffentlichlicher Bereich

Administration

Angern NW

Stammdaten

Allgemeines

S100 - Österreichische Kartenummer: 111

S101 - Aufnahmedatum: k.a.

S104 - Bedeutung: regional

S105 - Besitz: Privat

Standort

O100 - Standort - Gemeinde: Hohenems

O100 - Standort - Bezirk: Dornbirn

O100 - Standort - Bundesland: Vorarlberg

O101 - Region: Nordalpine Täler u. Senken

O102 - Naturraum: Vorarlberger Rheintal

O103 - Großraum: Tal- u. Beckenlandschaften

Abmessungen

S102 - Fläche: 14014244.000

iFlora Public © shInformatics

Abbildung 59 - iFlora Öffentlichlicher Bereich – Objekte Stammdatenformular

Besitzt das Objekt Aufnahmen (d.h. es gibt in der Anwendung nur eine Objektart, die Aufnahmen besitzt: BESTAND) dann werden in der Stammdatenseite des Objektes Links auf diese Aufnahmen angezeigt (siehe Abbildung 60 unten).

iFlora Öffentlichlicher Bereich

Administration

Angern NW

Bestand

Meteorologie

S111 - Geomorphologie: Talboden

S116 - Untergrund: Alluvionen

S112 - Jahresniederschlag: 1000-1500

S113 - Niederschlag über einen Tag: 130-150

S114 - durchschnittliche Jännertemperatur: über -1

S115 - durchschnittliche Julitemperatur: 17 bis 19

S121 - Klimatyp: k.a.

Moor

M100 - Moortyp: Überflutungsmoor

M101 - pH Wert: subneutral

M102 - Trophie: mesotroph

Aufnahmen

[AF: Angern NW](#)

DOWNLOAD: Excel PDF Text

iFlora Public © shInformatics

Abbildung 60 - iFlora Öffentlichlicher Bereich – Objekte Bestand – Link zu den Aufnahmen

Besitzt das Objekt noch Vegetationsdaten (d.h. es gibt in der Anwendung nur eine Objektart, die Vegetationsdaten besitzt: AUFNAHME), dann wird in der Stammdatenseite des Objektes ein Link auf diese Vegetationsdaten angezeigt (siehe Abbildung 61 unten).

The screenshot shows the 'iFlora Öffentliches Bereich' web application. On the left is an 'Administration' sidebar with a tree view containing folders like 'Auswahl', 'Objekte', 'Auswertungen', and 'Downloads'. Under 'Objekte', there are sub-items: 'Aufnahme', 'Bestand' (highlighted), 'Bezirk', 'Bundesland', 'Gemeinde', 'Großraum', 'Naturraum', and 'Region'. The main content area displays the 'Stammdaten' (Basic Data) form for 'AF: Angern NW'. This form is divided into sections: 'Diverses' with fields for 'S117 - Assoziation', 'S118 - Rasse', 'S119 - Subassoziation', and 'S120 - Variante'; 'Moor' with fields for 'M103 - Facies' and 'M104 - Phase'; and 'Bestand' with a value of 'Angern NW'. Below the form, there are links for 'DOWNLOAD:' (Excel, PDF, Text) and 'VERWEIS (Link):' (Vegetationsdaten). The footer shows 'iFlora Public' and '© shInformatics'.

Abbildung 61 - iFlora Öffentliches Bereich – Objekte Bestand Stammdatenformular

Über den Link Vegetationsdaten kommt man auf die Seite mit den Vegetationsdaten des Objektes (siehe Abbildung 62).

The screenshot shows the iFlora Öffentlichlicher Bereich interface. On the left is an 'Administration' sidebar with a tree view containing folders like 'Auswahl', 'Objekte', 'Aufnahme', 'Bestand', 'Bezirk', 'Bundesland', 'Gemeinde', 'Großraum', 'Naturraum', 'Region', 'Auswertungen', and 'Downloads'. The 'Objekte' folder is expanded, showing 'Aufnahme' and 'Bestand'. The main area displays 'AF: Angern NW' with a sub-header 'Aufnahme' and a table titled 'Vegetationsdaten'. The table lists various plant species and their corresponding values.

AF: Angern NW	
Aufnahme	
Vegetationsdaten	
Vicia sepium L.	+
Rhinanthus glacialis Personnat	1
Serratula tinctoria L.	1
Sanguisorba officinalis L.	+
Salix aurita L.	+
Molinia arundinacea Schrank	4
Lysimachia vulgaris L.	+
Phleum pratense agg.	+
Phragmites australis (Cav.) Steud.	2
Equisetum palustre L.	+
Iris sibirica L.	3
Inula salicina L.	+
Gladiolus palustris Gaudin	+

iFlora Public © shInformatix

Abbildung 62 - iFlora Öffentlichlicher Bereich – Objekte Vegetationsdatenformular

Aus den oben genannten Gründen sind die Objektarten „BESTAND“ und „AUFNAHME“ zentrale Elemente dieser Anwendung. Für verschiedene Aufgaben können beliebig viele Objektarten geschaffen werden und die Objekte dieser Objektarten miteinander in Beziehung gesetzt werden.

Für diese Eigenschaften gibt es in diesem Prototyp und der zugrunde liegenden Arbeit, noch keine Überlegungen.

6.1.4.1 Codierung

Menüpunkt: Auswahl Objekt – der Sourcecode befindet sich in der Datei

„/iFloraPublic/public/OOPSelObjekt.php“ (siehe Kapitel 9.3.2.2.7 Controller – „<Objekt>“). Die diesem Sourcecode sind die Projektauswahl und die Objektliste mit den Link auf die jeweiligen Stammdatenformulare umgesetzt.

Formular: Stammdatenformular – der Sourcecode befindet sich in der Datei

„/iFloraPublic/public/OOPFormStammdata.php“ (siehe Kapitel 9.3.2.2.8 Controller – „Aufnahme Stammdaten“).

Formular: Vegetationsaufnahme – der Sourcecode befindet sich in der Datei

„/iFloraPublic/public/OOPFormAufnahmedata.php“ (siehe Kapitel 9.3.2.2.9 Controller – „Vegetationsdat. der Aufnahme“).

6.1.5 Auswertungen

In diesem Punkt ist geplant, dass die Benutzer sich über mehrere Schritte eine komplexe Abfrage zusammenstellen, sich die Daten anzeigen lassen und, nach einer Kontrolle der Daten, diese in verschiedenen Formaten (Text, Excel) Downloaden kann (z.B.: für eine „Statistische Analyse“ mit SPSS).

Für diese Eigenschaften gibt es in diesem Prototyp und der zugrunde liegenden Arbeit, noch keine Überlegungen.

6.1.6 Downloads

Unter diesem Menüpunkt sollen die Benutzer und v.a. die Projektmitarbeiter den gesamten Datenstand eines Projektes abrufen können. Damit soll ein zusätzliches dezentrales Backup-System implementiert werden. Zusätzlich soll damit eine Weitergabe aller Daten eines Projektes gewährleistet werden.

Für diese Eigenschaften gibt es in diesem Prototyp und der zugrunde liegenden Arbeit, noch keine Überlegungen.

6.2 iFlora Administration (iFloraAdmin)

Die iFlora Administration öffnet sich zuerst mit einer Login-Seite (siehe Abbildung 63).

Die Daten von iFlora dürfen nur registrierte Benutzer bearbeiten, die sich mit ihrem Benutzernamen und Passwort auf der Login-Seite der Anwendung iFloraAdmin anmelden müssen. Deshalb wird der gesamte Datentransfer dieser Anwendung zwischen den Formularen im Browser des Benutzers (Client) und dem Webserver mit der Anwendung iFloraAdmin verschlüsselt (HTTPS).

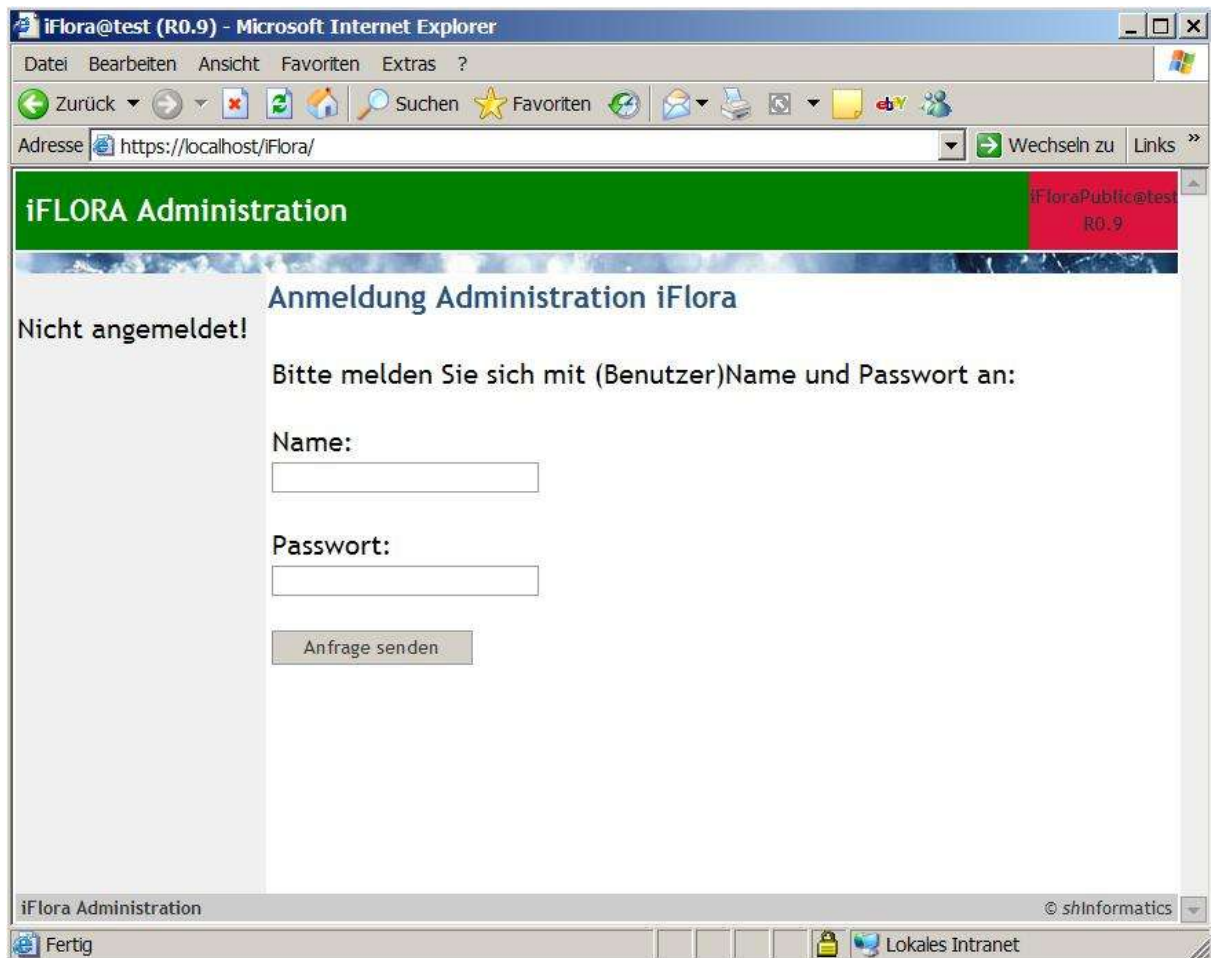


Abbildung 63 – iFlora Administration - Login

Wenn bei der Anmeldung ein Fehler auftritt (z.B. ein Tippfehler bei der Eingabe des Benutzernamens oder des Passwortes), wird dies mit einer Fehlermeldung angezeigt und der Benutzer hat die Möglichkeit für einen weiteren Versuch (siehe Abbildung 64).

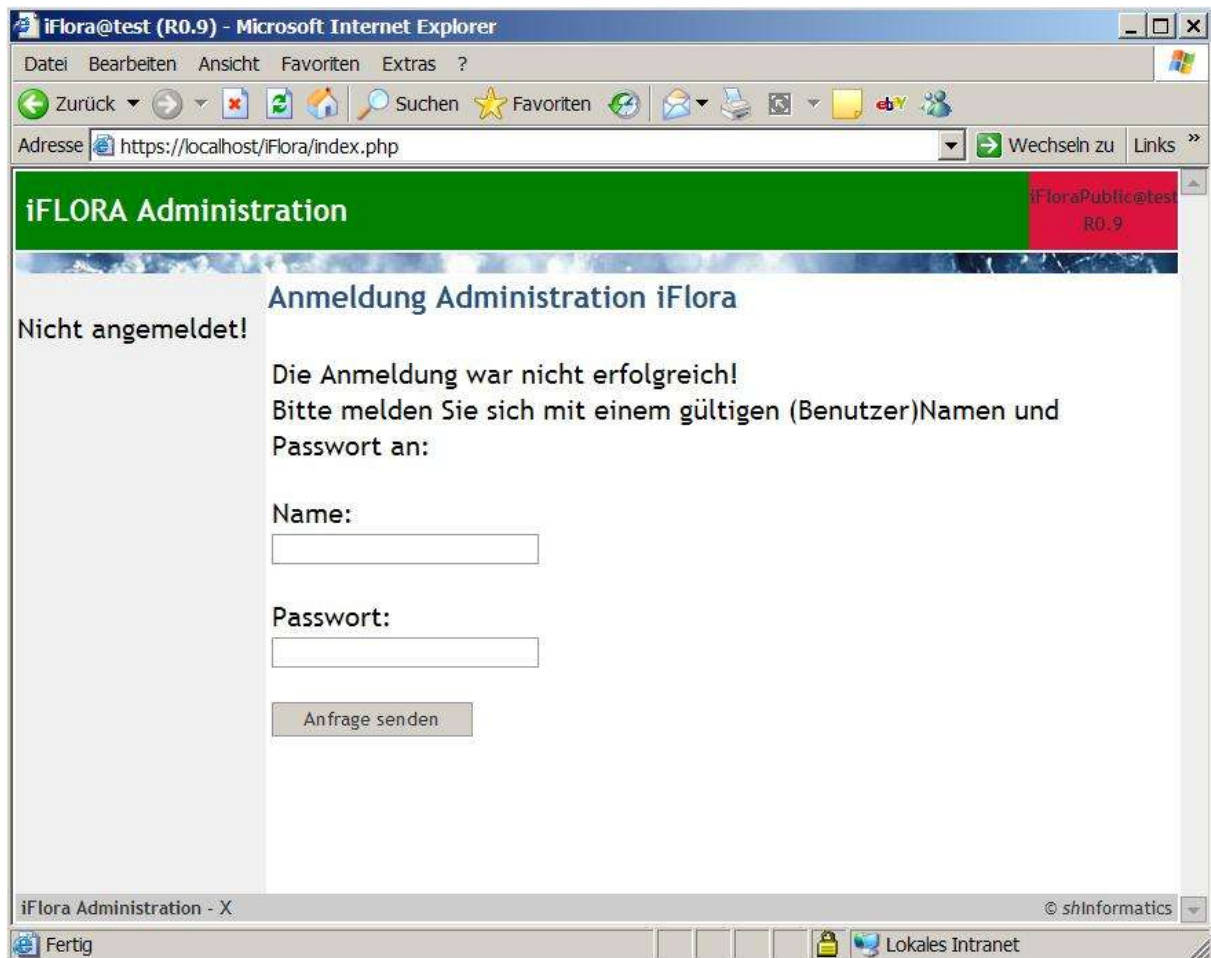


Abbildung 64 - iFlora Administration – Login Fehlermeldung

Nach der erfolgreichen Anmeldung hat jede Seite der Administration, wie der öffentliche Bereich, eine Hauptseite mit folgenden Komponenten (siehe auch Abbildung 65):

- Kopfzeile: Die Kopfzeile die ist oberste Zeile im Browserfenster. Es enthält die weiße Überschrift „iFlora Administration“ auf grünen Hintergrund. Dazu in der rechten Ecke eine Information, in welcher Version (R0.9) man sich gerade befindet und mit welcher Datenbank („test“) man verbunden ist.
- Menüleiste: Die Menüleiste befindet sich im linken Bereich des Browserfensters. Hier befindet sich der Auswahlbaum, um die verschiedenen Funktionen der Anwendung auszuwählen und aufzurufen. Für diese Funktionalität wird die Anwendung „dTree 2.05“ verwendet (<http://www.destroydrop.com/javascripts/tree/>).

- Statuszeile: Die Statuszeile ist die unterste Zeile im Browserfenster. Sie enthält links den Namen der Anwendung und den Benutzernamen (ADMIN). Weiters befindet sich rechts ein Copyright Hinweis.
- Rahmen für die dynamischen Webseiten: Über den Auswahlbaum können verschiedene Funktionen der Anwendung iFlora aufgerufen werden. Die Seiten dieser Funktionen öffnen sich in diesem Rahmen.

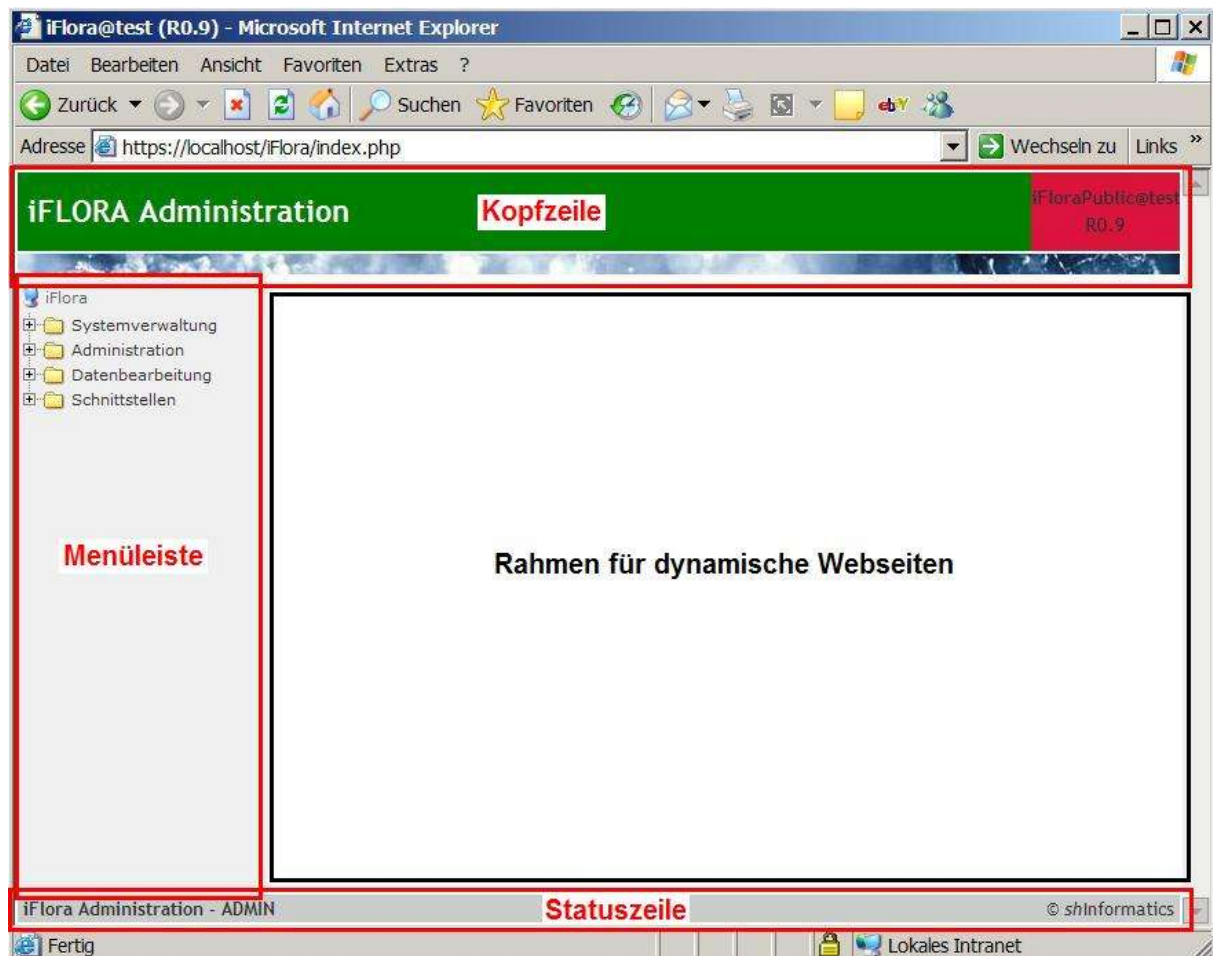


Abbildung 65 - iFlora Administration - Startseite mit einzelnen Elementen

6.2.1 Allgemeine Informationen

Die Dateien der Anwendung befinden sich in einer Verzeichnisstruktur (siehe 6.1.1.1 Verzeichnisstruktur).

In den meisten Anwendungen befinden sich Programmteile, die nicht einem bestimmten Formular zugeordnet werden, sondern die von mehreren Formularen gemeinsam verwendet werden (siehe 6.1.1.2 Libraries).

6.2.1.1 Verzeichnisstruktur

Die verschiedenen Dateien der Anwendung iFloraAdmin befinden sich in den folgenden Verzeichnissen (siehe Abbildung 66 unten).



Abbildung 66 - iFlora Administration – Verzeichnisstruktur der Anwendung

- „/config_iflora/“: In diesem Verzeichnis befinden sich heikle Konfigurationsdaten.
- „/httpsdocs/“: Alle Unterverzeichnisse in diesem Verzeichnis sind im verschlüsselten Internet-Pfad des Servers (https://...) aufrufbar.
- „/httpsdocs/iFlora/“: Dieses Verzeichnis enthält alle Dateien der Anwendung **iFloraAdmin** und wird über den Pfad <https://localhost/iFlora> aufgerufen. Über die Datei „index.php“, die automatisch beim Aufruf dieses Verzeichnisses ausgeführt wird, meldet sich die Anwendung.
- „/httpsdocs/iFlora/admin/“: Hier befinden sich alle Sourcecode-Dateien der verschiedenen Formulare (Funktionen von iFloraAdmin). Diese Formulare werden über den **Auswahlbaum** der „**Menüleiste**“ aufgerufen und im „**Rahmen für dynamische Webseiten**“ angezeigt.
- „/httpsdocs/iFlora/fpdf/“: Bei verschiedenen Formularen gibt es die Möglichkeit, die Daten in Form einer PDF-Datei downzuloaden. Die dafür notwendigen Tools werden von der Anwendung FPDF bereitgestellt. Es wird die Version FPDF 1.53 (<http://www.fpdf.org/>) verwendet. Die Dateien dieser Anwendung befinden sich alle in diesem Verzeichnis.
- „/httpsdocs/iFlora/layout/“: Für das Layout der Startseite werden einige Grafiken gebraucht, diese befinden sich in diesem Verzeichnis.

- „/httpsdocs/iFlora/lib/“: Einige Programmbereiche werden von mehreren Formularen benutzt. Diese Sourcecode-Dateien werden in diesem Verzeichnis gespeichert.
- „/httpsdocs/iFlora/style/“: Der Style der verschiedenen Seiten wird über CSS-Dateien definiert. Diese CSS-Dateien befinden sich in diesem Verzeichnis.

„/httpsdocs/iFlora/tree/“: Für den Auswahlbaum in der Menüleiste wird die Anwendung „dTree 2.05“ verwendet. Alle Dateien dieser Anwendung befinden sich in diesem Verzeichnis.

6.2.1.2 Libraries

In der Anwendung iFloraAdmin werden bestimmte Teile des Programms für mehrere Funktionen und Formularen gemeinsam verwendet und in Bibliotheken (Libraries) zusammengefasst.

- „DBManager.php“: Diese Bibliothek enthält alle allgemeinen Teile der Anwendung zum Abfragen der Datenbank.
- „LAYOUTManager.php“: Diese Bibliothek enthält das Layout der verschiedenen dynamischen Webseiten (z.B. HTML-Header, HTML-Footer).
- „VIEWManager.php“: Enthält verschiedene Funktionen zum Aufbau der Datenbereich der dynamischen Webseiten.

6.2.2 Startseite

Nach der Eingabe des Links der Anwendung „iFlora Administration“ im Internet Browser (beim Prototyp: <https://localhost/iFlora/>) öffnet sich eine Seite, wie unten (Abbildung 67) dargestellt.



Abbildung 67 - iFlora Administration - Startseite

Über die Menüleiste auf der linken Seite kann man die verschiedenen Funktionen der Anwendung „iFloraAdmin“ aufrufen. Sie zeigt nur die Menüpunkte auf, die für die Rollen (näheres siehe unten) des jeweiligen Benutzers erlaubt sind.

Bei Funktionen gibt es verschiedene Möglichkeiten, wie diese Funktionen genutzt werden können, d.h. ob sie vollständig genutzt werden kann oder nur mit Einschränkungen. Diese Möglichkeiten werden Rechte genannt. Damit ein Benutzer diese Funktionen in einer bestimmten weise nutzen kann, muss er die Rechte dafür besitzen. Es ist sehr schwierig und aufwendig, für jeden User die Rechte, die er bei den verschiedenen Funktionen braucht, zu verwalten. Daher werden typische Benutzer ermittelt und die Rechte, die dieser Benutzer bei den verschiedenen Funktionen braucht, werden zu Rollen zusammengefasst.

Es gibt folgende Rollen, d.h. typische Benutzergruppen:

- **Administratoren** – die Benutzer mit dieser Rolle dürfen alle Funktionen ausführen bzw. alle Seiten aufrufen.
- **Projektleiter** – die Benutzer mit dieser Rolle sind innerhalb ihrer Projekte die Administratoren, sie haben aber keine Rechte in der Systemverwaltung.

- **Projektmitarbeiter** – diese Benutzer haben keinerlei Administrationsrechte, sondern dürfen nur innerhalb der ihnen zugeordneten Projekte die Daten eingeben, verwalten und, eingeschränkt, auch löschen.

6.2.2.1 Codierung

Die Startseite ist in der Datei „iFlora/index.php“ codiert (Sourcecode siehe „9.3.1.2.1 Startseite – Rahmen mit Menü“). Beim Start wird im „Rahmen für dynamische Webseiten“ eine leere HTML-Datei angezeigt („iFlora/work.html“ – Sourcecode siehe „9.3.1.2.2 Startseite – Work (im Rahmen)“).

6.2.3 Systemverwaltung

Über den Menüpunkt „Systemverwaltung“ können alle Systemvariablen verwaltet werden. Die Seiten der Systemverwaltung stehen nur Benutzern mit der Rolle **ADMINISTRATOR** zur Verfügung! Es gibt folgende Funktionen in der Systemverwaltung (siehe Abbildung 68):

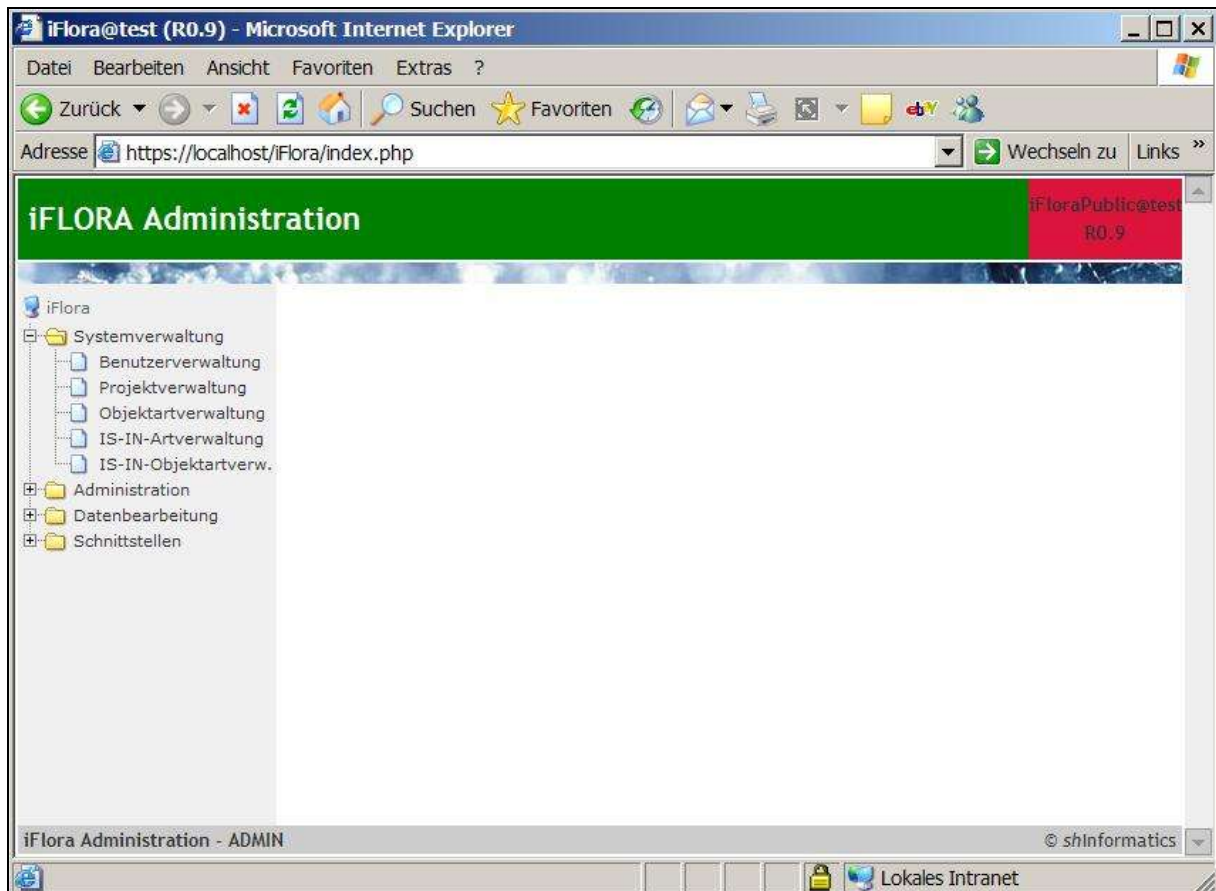


Abbildung 68 - iFlora Administration - Systemverwaltung

- **Menüpunkt: Benutzerverwaltung** – hier kann man die Benutzer des Systems anlegen, verwalten und deaktivieren. Über die Schnellsuche in der obersten Frameworkzeile kann man nach einem bestimmten Benutzernamen oder Familiennamen eines Benutzers suchen (siehe Abbildung 69):



Abbildung 69 - iFlora Administration - Benutzerverwaltung

- **Menüpunkt: Projektverwaltung** – hier kann man die Projekte des Systems anlegen und verwalten. Über die Schnellsuche in der obersten Frameworkzeile kann man nach einem bestimmten Projekt suchen (siehe Abbildung 70):



Abbildung 70 - iFlora Administration - Projektverwaltung

- **Menüpunkt: Objektartverwaltung** – hier kann man neue Objektarten anlegen und verwalten. Über die Schnellsuche in der obersten Frameworkzeile kann man nach einem bestimmten Objekt suchen (siehe Abbildung 71):

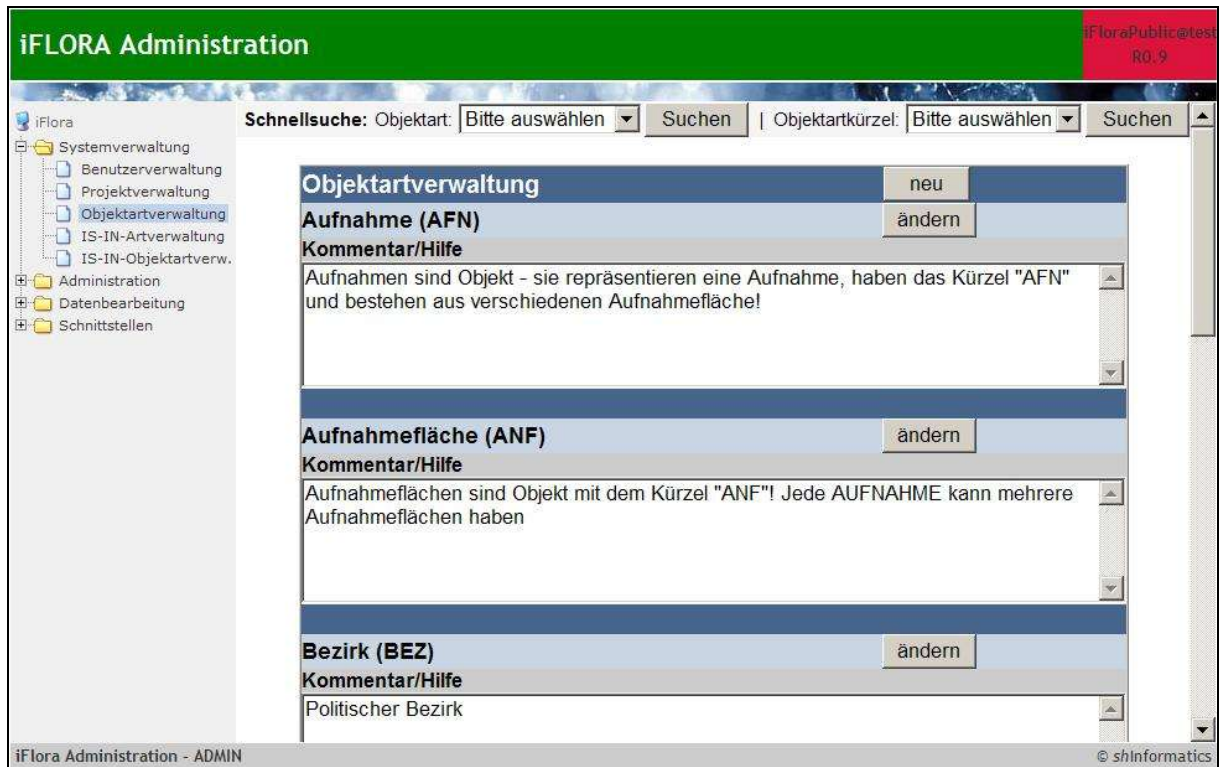


Abbildung 71 - iFlora Administration - Objektartverwaltung

- **Menüpunkt: IS-IN-Artverwaltung** – einzelne Objekte (verschiedener Objektarten) können miteinander in Beziehungen gesetzt werden. In diesem System wird jeder dieser Objektbeziehungen (IS-IN-Arten) ein Name vergeben – dieser Name wird auch Pfad oder Thema genannt. In diesem Formular können diese IS-IN-Arten verwaltet werden. Im Menüpunkt IS-IN-Objektartverwaltung werden zu diesen Themen die Objektarten, für die man Objekte in Beziehung setzen will, zugeordnet (siehe nächster Punkt). Über die Schnellsuche in der obersten Frameworkzeile kann man nach einer bestimmten IS-IN-Art bzw. nach einem bestimmten IS-IN-Artkürzel suchen (siehe Abbildung 72):

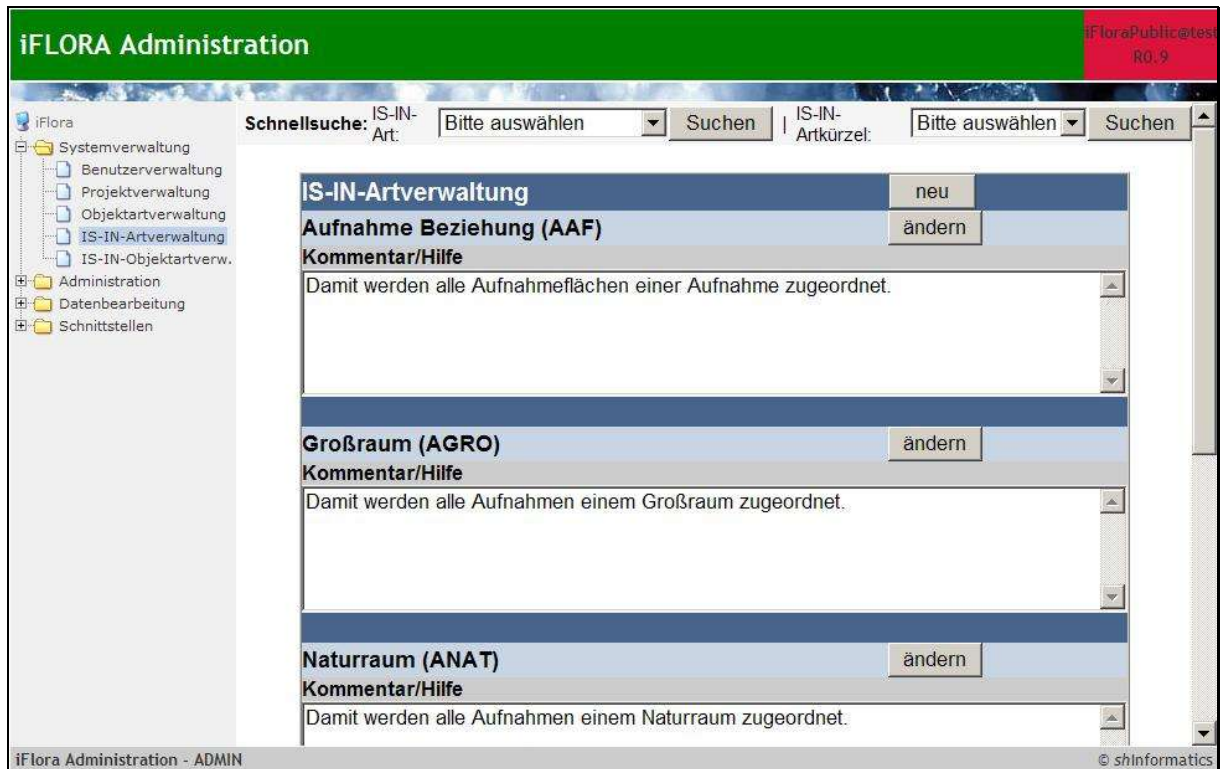


Abbildung 72 - iFlora Administration – IS-IN-Artverwaltung

- **Menüpunkt: IS-IN-Objektartverwaltung** – in diesem Formular werden zu den Themen der IS-IN-Artverwaltung die Objektarten, für die man Objekte in Beziehung setzen will, verwaltet. Über die Schnellsuche in der obersten Frameworkzeile kann man nach einem bestimmten Objekt suchen (siehe Abbildung 73):

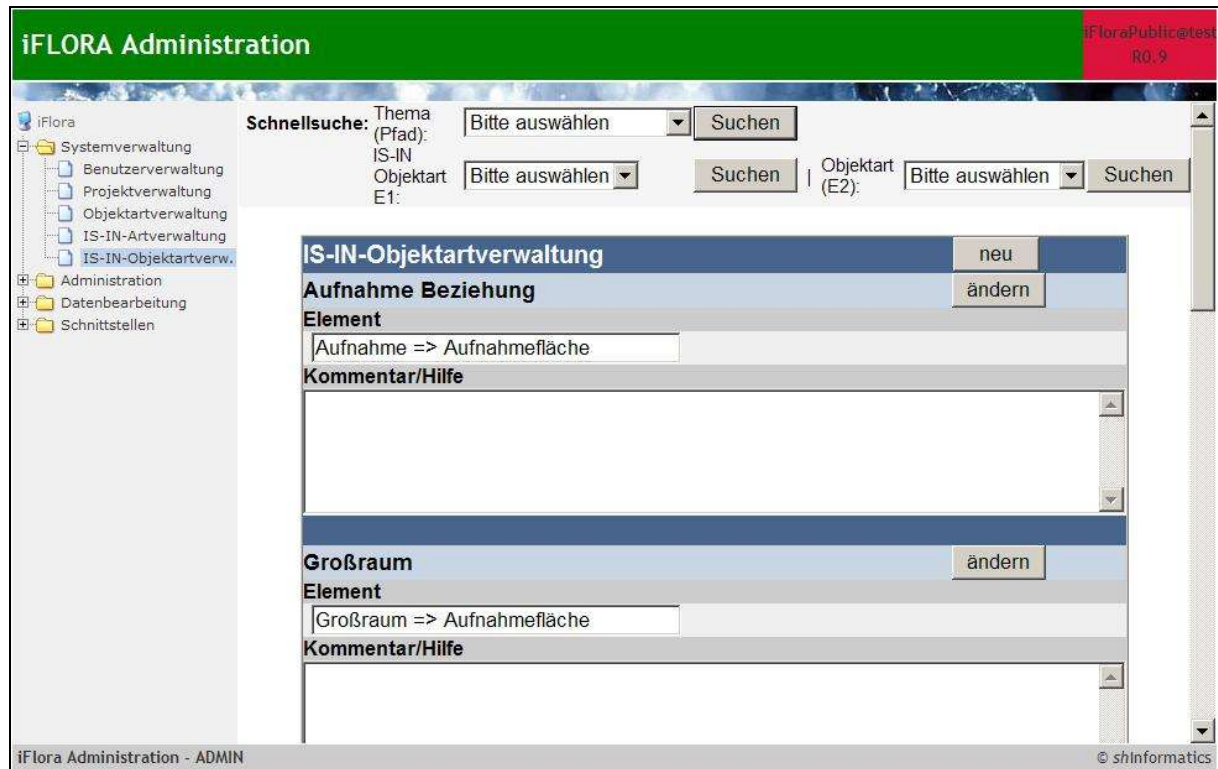


Abbildung 73 - iFLora Administration – IS-IN-Objektartverwaltung

6.2.3.1 Codierung

- **Menüpunkt: Benutzerverwaltung** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormBenutzer.php“ (siehe Kapitel 9.3.2.2.3 Controller – „Benutzerverwaltung“).
- **Menüpunkt: Projektverwaltung** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormProjekt.php“ (siehe Kapitel 9.3.2.2.4 Controller – „Projektverwaltung“).
- **Menüpunkt: Objektartverwaltung** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormObjektart.php“ (siehe Kapitel 9.3.2.2.5 Controller – „Objektartverwaltung“).
- **Menüpunkt: IS-IN-Artverwaltung** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormISINArt.php“ (siehe Kapitel 9.3.2.2.6 Controller – „IS-IN-Artverwaltung“).
- **Menüpunkt: IS-IN-Objektartverwaltung** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormISINOBA.php“ (siehe Kapitel 9.3.2.2.7 Controller – „IS-IN-Objektartverwaltung“).

6.2.4 Administration

Über den Menüpunkt „Administration“ können verschiedene Eigenschaften verwaltet werden. Dieser Menüpunkt steht den Benutzern mit der Rolle **ADMINISTRATOR** oder der Rolle **PROJEKTLLEITER** zur Verfügung! Es gibt folgende Funktionen in der Administration (siehe Abbildung 74):

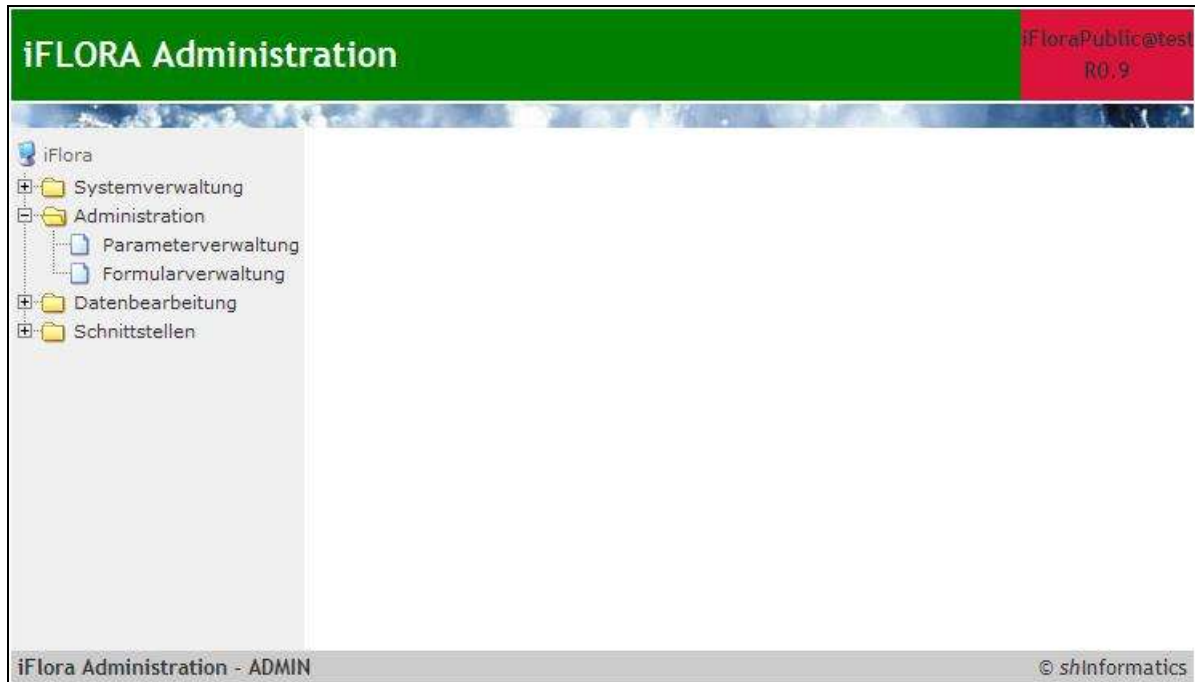


Abbildung 74 - iFlora Administration - Administration

- **Menüpunkt: Parameterverwaltung** – hier kann man Parameter, die man in den verschiedenen Projekten braucht, anlegen (Button „NEU“) und verwalten (Button „ÄNDERN“). Über die Schnellsuche in der obersten Frameworkzeile kann man nach einer bestimmten Parameterklasse oder nach einem bestimmten Parameter suchen (siehe Abbildung 75 unten):

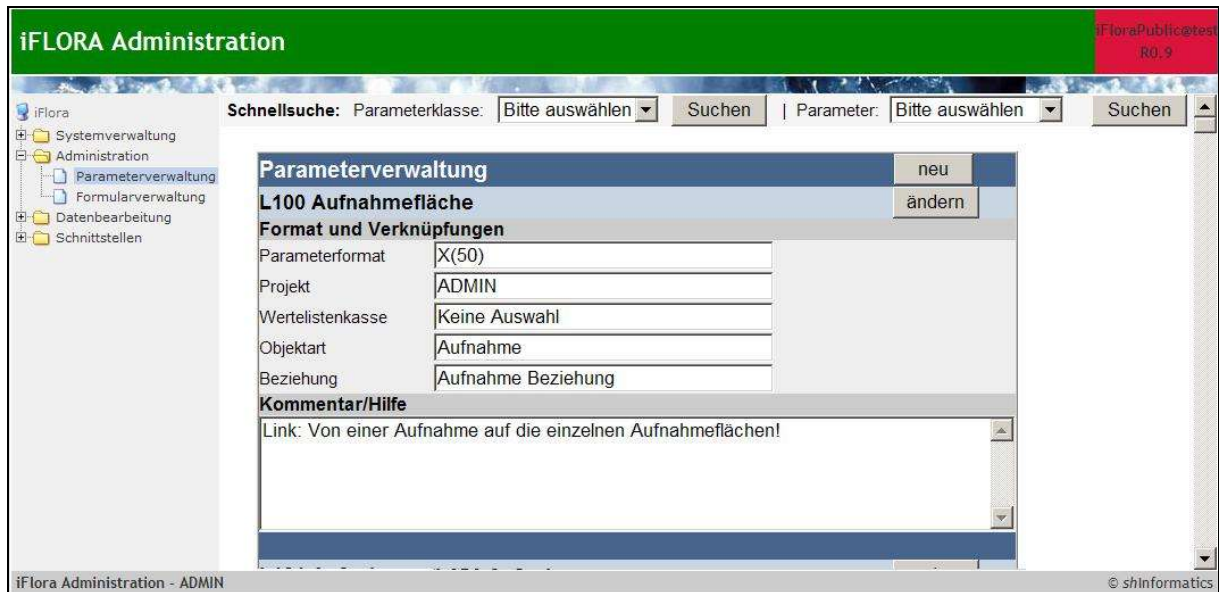


Abbildung 75 - iFlora Administration - Parameterverwaltung

- **Menüpunkt:** Formularverwaltung – Die Formulare bestehen aus der Verwaltung des Formulars und dessen Formularzeilen.
 - Formularverwaltung - Formular – In diesem Formular wird der Name eingegeben und der Dateninhalt, über die Objektart bestimmt. Man kann hier die Formulare eines Projektes anlegen (Button „NEU“) und verwalten (Button „ÄNDERN“). Über den Link „Formularzeilen“ kommt man in das Formular zur Verwaltung der Zeilen des jeweiligen Formulars. Über die Schnellsuche in der obersten Frameworkzeile kann man ein oder mehrere Formulare über ein bestimmtes Projekt oder einen Formularnamen suchen (siehe Abbildung 76):

Abbildung 76 - iFlora Administration – Formularverwaltung 1

- Formularverwaltung - Formularzeile – In diesem Formular (aufgerufen über den Link „Formularzeile“ in der Formularverwaltung – Formular) kann man Formularzeilen des jeweiligen Formulars anlegen (Button „NEU“) und verwalten (Button „ÄNDERN“). Über die Schnellsuche in der obersten Frameworkzeile kann man ein oder mehrere Formulare nach einem bestimmten Projekt oder Formularnamen suchen (siehe Abbildung 77):

iFLORA Administration iFloraPublic@test
R0.9

Schnellsuche: Formularzeile: Bitte auswählen Suchen

Formularzeilenverwaltung neu

MOOR - Aufnahme Stammdaten

Zeile:	ändern
Zeile: 1	ändern
Position	10
Überschrift	Allgemeines
Layoutebene	3
Zeile: 2	ändern
Position	20
Parameter	S100-Österreichische Kartennummer
Layoutebene	4
Zeile: 3	ändern
Position	30
Parameter	S101-Aufnahmedatum

iFlora Administration - ADMIN © shinformatics

Abbildung 77 - iFlora Administration – Formularverwaltung 2

Im Gegensatz zu den anderen Formularen ändert sich die Ansicht im Editmodus einer Formularzeile, da sich der Formularzeileninhalt entweder aus einer Überschrift oder aus einem Parameterinhalt zusammensetzt. Das wird über ein Auswahl-Pflichtfeld mit Radiobutton und Texteingabe bzw. Parameterauswahl realisiert (siehe Abbildung 78):

Abbildung 78 - iFlora Administration – Formularverwaltung 3

6.2.4.1 Codierung

- **Menüpunkt:** Parameterverwaltung – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormParameter.php“ (siehe Kapitel 9.3.2.2.9 Controller – „Parameterverwaltung“).
- **Menüpunkt:** Formularverwaltung – dieser Menüpunkt besteht aus den 2 Teilen Formular und Formularzeile.
 - Formular - der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormFormular.php“ (siehe Kapitel 9.3.2.2.10 Controller – „Formular“).
 - Formularzeile - der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormFormRow.php“ (siehe Kapitel 9.3.2.2.11 Controller – „Formularzeile“).

6.2.5 Datenbearbeitung

Über den Menüpunkt „Datenbearbeitung“ können die Stamm- und, so weit vorhanden (nur bei Aufnahmen), die Vegetationsdaten der verschiedenen Objekte (verschiedener Objektarten) verwaltet werden, d.h. die einzelnen Objekte können ausgewählt werden und die verschiedenen Daten können bearbeitet werden. Dieser Menüpunkt steht den Benutzern mit der Rolle **ADMINISTRATOR**, der Rolle **PROJEKTLLEITER** und der Rolle **MITARBEITER** zur Verfügung! Es gibt folgende Funktionen in der Datenbearbeitung (siehe Abbildung 79):

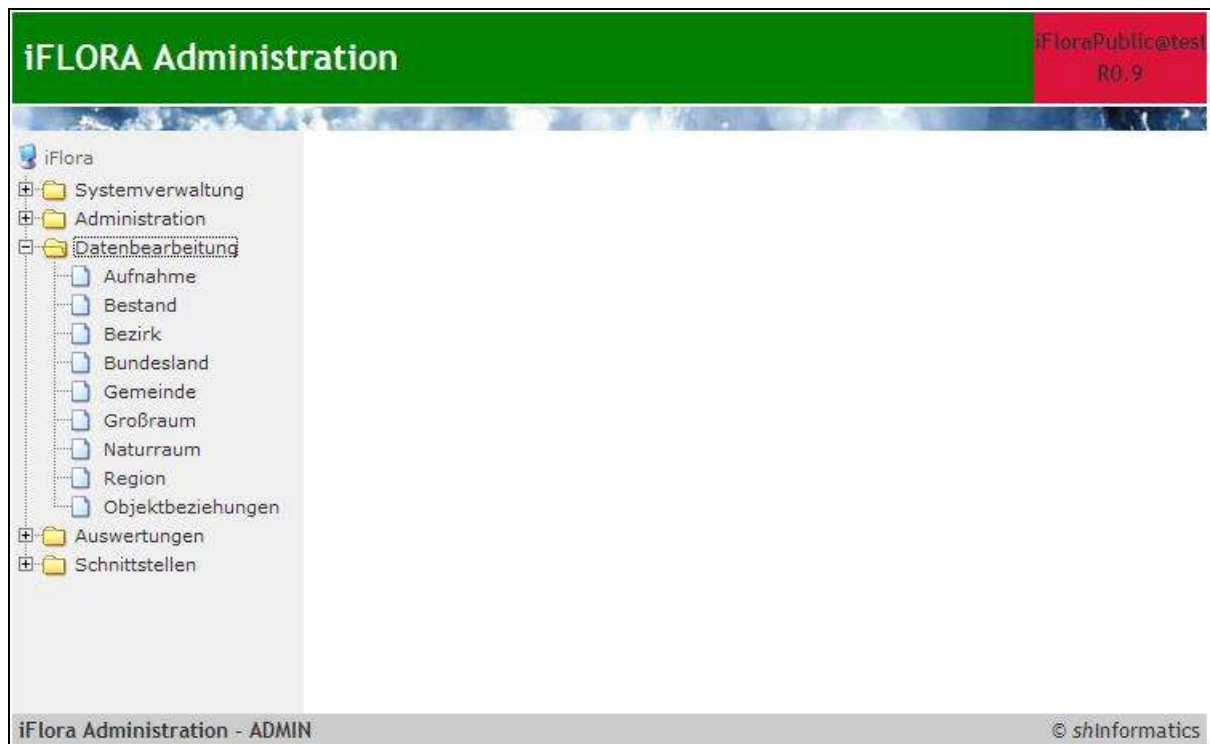


Abbildung 79 - iFlora Administration – Datenbearbeitung

Im Menü „Datenbearbeitung“ werden immer alle definierten Objektarten angezeigt (das Menü wird bei jedem Aufruf dynamisch erstellt). Beim Aufruf einer Objektart wird zuerst eine Projektauswahlseite angezeigt. Durch Auswahl eines Projektes und klicken auf den Button „Auswahl“ gelangt man auf die Objektseite. Hier werden alle Objekte der Objektarten, die dem ausgewählten Projekt zugeordnet sind, angezeigt. Jeder Objektname ist auch gleichzeitig eine Link auf das Stammdatenformular. Wurde zu einer Objektart kein Stammdatenformular definiert, dann erscheint beim Aufruf dieses Links die Meldung „Zu diesem Objekt sind keine Daten vorhanden!“.

Gibt es zu einem Objekt für dessen Objektart ein Stammdatenformular definiert wurde, keine Stammdaten, dann wird eine leere Stammdatenseite angezeigt.

Für diesen Prototypen wurden die Formulare des Menüs „Datenbearbeitung“ nur READONLY ausgeführt, d.h. es gibt keine Möglichkeit die Daten (Stamm- und Vegetationsdaten) zu bearbeiten. Daher ist die gesamte Funktionalität des Prototypen mit den Formularen unter dem Menüpunkt „Objekte“ der Anwendung iFloraPublic (mit der Ausnahme des Menüpunktes „Objektbeziehungen“ – siehe Abbildung 79 oben) vergleichbar.

Es gibt derzeit folgende Objektarten und daher folgende Untermenüpunkte im Menü „Datenbearbeitung“:

- **Menüpunkt: Aufnahme**
- **Menüpunkt: Bestand**
- **Menüpunkt: Bezirk**
- **Menüpunkt: Bundesland**
- **Menüpunkt: Gemeinde**
- **Menüpunkt: Großraum**
- **Menüpunkt: Naturraum**
- **Menüpunkt: Region**
- **Menüpunkt: Objektbeziehungen** – hier kann man die einzelnen Objektbeziehungen zwischen den verschiedenen Objekten verwalten.

Einzelne Objekte können mit anderen Objekten in Beziehung stehen. So gehört eine Gemeinde (Objekt **Gemeinde**) immer zu einem Bezirk (Objekt **Bezirk**) und dieser immer zu einem Bundesland (Objekt **Bundesland**). Das wird unter den Beziehungen zwischen Objekten (verschiedener Objektarten) verstanden. Eine Aufnahme (Objekt **Aufnahme**) ist immer mit einem Bestand (Objekt **Bestand**) verbunden und der Bestand ist mit einer Gemeinde verbunden. Damit ist die Aufnahme aus den oben genannten Beziehungen auch einem Bundesland zugeordnet.

Eine Beziehung hat immer ein Thema (auch Pfad genannt) unter dem 2 Objekte miteinander verknüpft werden.

Über die Schnellsuche in der obersten Frameworkzeile kann man nach bestimmten Objekten suchen und nach einem Thema oder Pfad, mit dem die verschiedenen Objekte verbunden sind (siehe Abbildung 80):

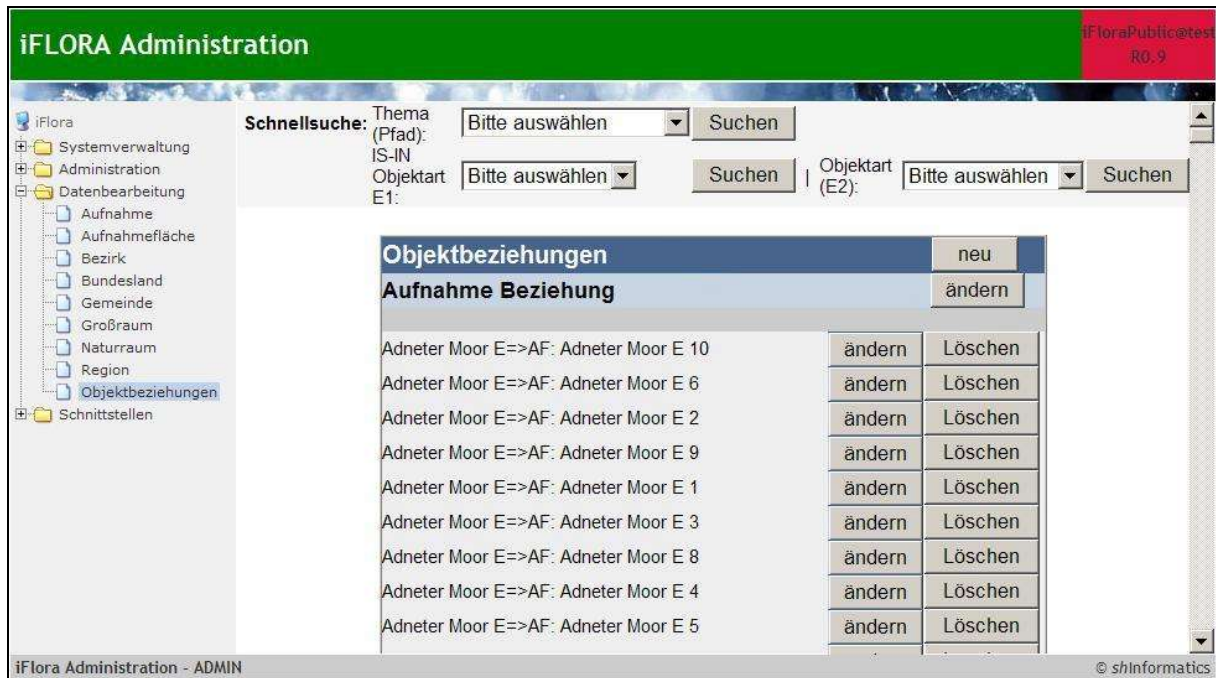


Abbildung 80 - iFlora Administration – Objektbeziehungen

6.2.5.1 Codierung

Untermenüpunkt: Datenbearbeitung – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINSelObjekt.php“ (siehe Kapitel 9.3.1.2.12 Controller – „<Objekt>“). In diesem Sourcecode sind die Projektauswahl und die Objektliste mit den Link auf die jeweiligen Stammdatenformulare umgesetzt.

- **Formular: Stammdatenformular** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormStammdata.php“ (siehe Kapitel 9.3.1.2.14 Controller – „Aufnahme Stammdaten“).
- **Formular: Vegetationsaufnahme** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormAufnahmedata.php“ (siehe Kapitel 9.3.1.2.15 Controller – „Veg. der Aufnahmefl.“).
- **Formular: Objektbeziehungen** – der Sourcecode befindet sich in der Datei „iFlora/admin/ADMINFormObjektBez.php“ (siehe Kapitel 9.3.1.2.13 Controller – „Objektbeziehungen“).

- **Download: „Stammdaten - Excel“** – der Download von Stammdaten (Vegetationsaufnahme) im Excel-Format wird in der Datei „iFloraPublic/public/OOPExcelStammdata.php“ codiert (siehe Kapitel 9.3.2.2.11 Controller – „Download Stammdaten Excel“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.
- **Download: „Stammdaten - PDF“** – der Download von Stammdaten (Vegetationsaufnahme) im PDF-Format wird in der Datei „iFloraPublic/public/OOPpdfStammdata.php“ codiert (siehe Kapitel 9.3.2.2.12 Controller – „Download Stammdaten PDF“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.
- **Download: „Stammdaten - Text“** – der Download von Stammdaten (Vegetationsaufnahme) im ASCII-Format wird in der Datei „iFloraPublic/public/OOPASCIISTammdata.php“ codiert (siehe Kapitel 9.3.2.2.10 Controller – „Download Stammdaten ASCII“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.
- **Ansicht: „Vegetationsdaten der Aufnahme“** – diese Ansicht ist in der Datei „iFloraPublic/public/OOPFormAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.9 Controller – „Vegetationsdat. der Aufnahme“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.
- **Download: „Aufnahmedaten - Excel“** – der Download von Aufnahmedaten (Vegetationsaufnahme) im Excel-Format wird in der Datei „iFloraPublic/public/OOPExcelAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.15 Controller – „Download Aufn. Excel“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.
- **Download: „Aufnahmedaten - PDF“** – der Download von Aufnahmedaten (Vegetationsaufnahme) im PDF-Format wird in der Datei „iFloraPublic/public/OOPpdfAufnahmedata.php“ codiert (siehe Kapitel 9.3.2.2.16 Controller – „Download Aufn. PFD“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.
- **Download: „Aufnahmedaten - Text“** – der Download von Aufnahmedaten (Vegetationsaufnahme) im ASCII-Format wird in der Datei „iFloraPublic/public/OOPASCIIAufnahmedata.php“ codiert (siehe Kapitel

9.3.2.2.13 Controller – „Download Aufn. ASCII“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.

- **Download:** „Aufnahmedaten - ORI-Datei“ – der Download von Aufnahmedaten (Vegetationsaufnahme) im ORI-Format (ist eine spezielle Form des Textformats) wird in der Datei „/iFloraPublic/public/OOPASCIIAufnahmedataORI.php“ codiert (siehe Kapitel 9.3.2.2.14 Controller – „Download Aufn. ORI-Datei“) – wird im Prototyp gemeinsam mit iFloraPublic verwendet.

6.2.6 Auswertungen

In diesem Punkt (siehe Abbildung 81) ist geplant, dass die Benutzer sich über mehrere Schritte eine komplexe Abfrage zusammenstellen, sich die Daten anzeigen lassen und, nach einer Kontrolle der Daten, diese in verschiedenen Formaten (Text, Excel) Downloaden können (z.B.: für eine statistische Analyse mit SPSS).

Für diese Eigenschaften gibt es in diesem Prototyp und der zugrunde liegenden Arbeit, noch keine Überlegungen.



Abbildung 81 - iFlora Administration – Auswertungen

6.2.7 Schnittstelle

Unter diesem Menüpunkt können die Benutzer Daten importieren (UPLOAD) bzw. auch exportieren (DOWNLOAD) (siehe Abbildung 82 unten).

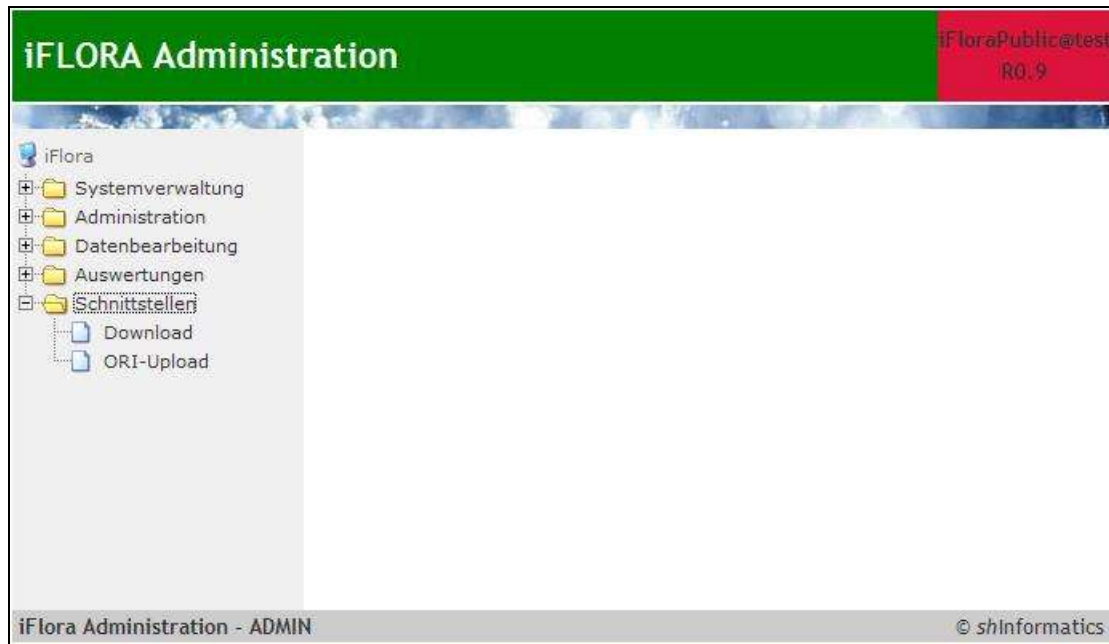


Abbildung 82 - iFlora Administration – Schnittstellen

- **Menüpunkt: Download** – Mittels der DOWNLOAD Funktion können die Projektmitarbeiter den gesamten Datenstand eines Projektes abrufen und lokal, d.h. auf dem PC des jeweiligen Benutzers abspeichern. Damit soll ein zusätzliches dezentrales Backup-System implementiert werden. Zusätzlich soll damit eine Weitergabe aller Daten eines Projektes gewährleistet werden. Für diese Eigenschaften gibt es in diesem Prototyp und der zugrunde liegenden Arbeit, noch keine Überlegungen.
- **Menüpunkt: ORI-Upload** – Über diesen Punkt wurde die UPLOAD Funktion realisiert. Die Benutzer können Aufnahmedaten im Format einer ORI-Datei in die Anwendung laden. Diese Funktion wurde für den Prototypen ausprogrammiert und getestet. Beim Import der Daten aus einer ORI-Datei gibt es folgende Schritte:
 - **Daten laden** – Beim Aufruf aus dem Menüpunkt „ORI-Upload“ öffnet sich das Formular „UPLOAD ORI-Datei“. Hier gibt es die Funktion, eine Datei auf dem lokalen PC zu suchen. Über den Button „Datei übertragen“ wird der Import gestartet (siehe Abbildung 83).

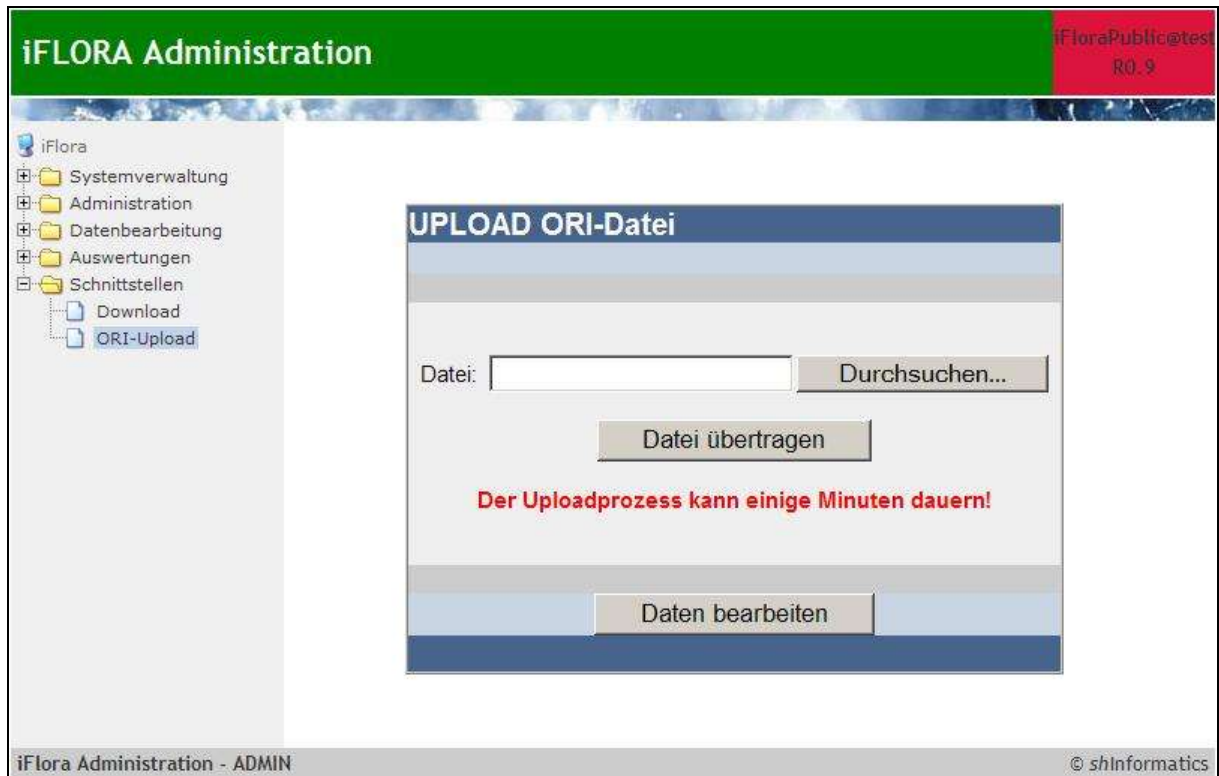


Abbildung 83 - iFlora Administration – ORI-Schnittstelle 1

- Daten prüfen – Beim Importieren der Daten werden diese überprüft. Sind die Daten nicht in Ordnung (z.B.: weil die übergebenen Aufnahmeflächen bereits vorhanden sind) wird eine Fehlermeldung ausgegeben (siehe Abbildung 84).

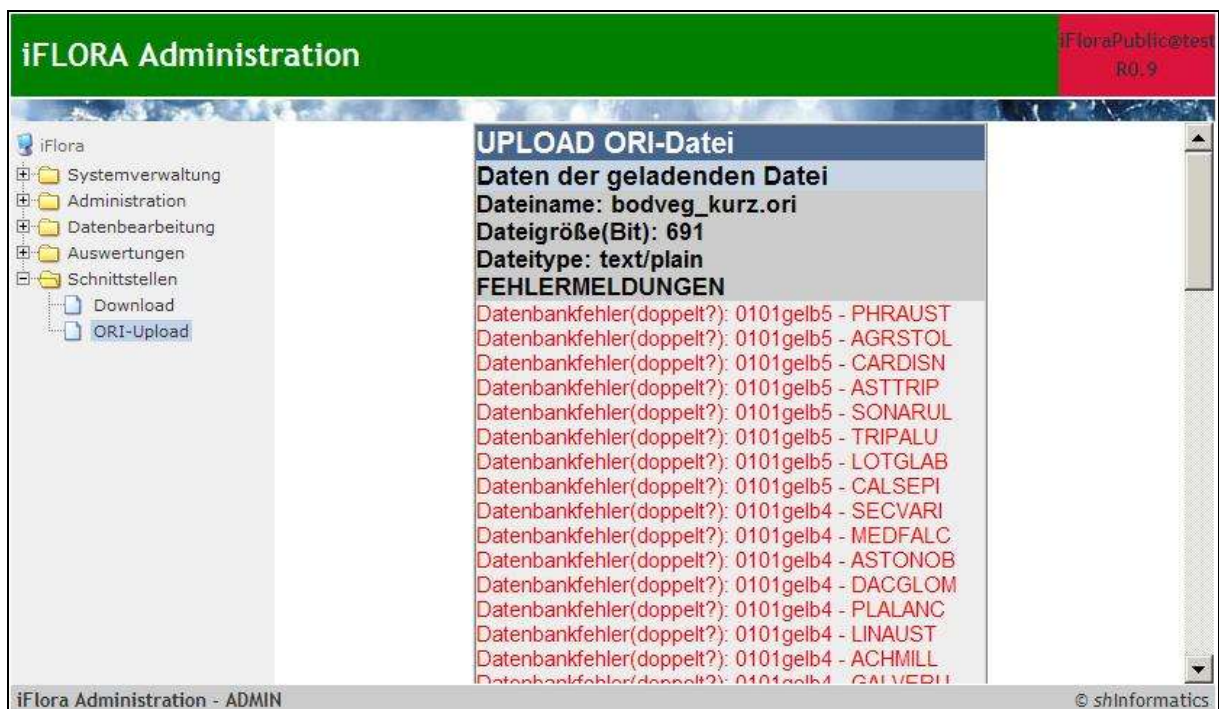


Abbildung 84 - iFlora Administration – ORI-Schnittstelle 2

- Datenansicht – Die importierten Daten werden in eine Temporäre Tabelle geladen und in einem Formular („UPLOAD-ORI Datei – Daten der geladenen Datei“) angezeigt. Es kann auch über den Button „Daten bearbeiten“ aus dem Formular „UPLOAD ORI-Datei“ (Startseite) auf diese Seite gegangen werden. Sind die Daten nicht in Ordnung (z.B.: weil man einen Tippfehler festgestellt hat) gibt es in jeder Datenzeile einen Button „Ändern“. (siehe Abbildung 85).

iFLORA Administration iFloraPublic@test
R0.9

UPLOAD ORI-Datei

Daten der geladenen Datei
 Dateiname: bodveg_kurz.ori
 Dateigröße(Bit): 691
 Dateitype: text/plain

0101gelb5

PHRAUST[Phragmites australis (Cav.) Trin. ex Steudk.]	2	ändern
AGRSTOL[Agrostis stolonifera L.]	3	ändern
CARDISN[Carex distans L.]	1	ändern
ASTTRIP[Aster tripolium L.]	3	ändern
SONARUL[Sonchus arvensis ssp. uliginosus (MB.) Nyman]	1	ändern
TRIPALU[Triglochin palustre L.]	1	ändern
LOTGLAB[Lotus glaber Mill.]	+	ändern
CALSEPI[Calystegia sepium (L.) R.Br.]	+	ändern

0101gelb4

SECVARI[Securigera varia (L.) Lassen]	2	ändern
MEDFALC[Medicago falcata L.]	3	ändern
ASTONOB[Astragalus onobrychis L.]	2	ändern
DACGLOM[Dactylis glomerata L.]	+	ändern
PLALANC[Plantago lanceolata L.]	1	ändern

iFlora Administration - ADMIN © shInformatics

Abbildung 85 - iFlora Administration – ORI-Schnittstelle 3

- Daten bearbeiten – Über den Button „Ändern“ in der Datenseite („UPLOAD-ORI Datei – Daten der geladenen Datei“ - siehe Abbildung 85) öffnet sich ein EDIT-Fenster mit der gewählten Datenzeilen (siehe Abbildung 86). Die Einträge können verändert werden und über den Button „Speichern“ in der Temporären Tabelle gespeichert werden.

Über den Button „Löschen“ kann die gesamte Datenzeile gelöscht werden.

The screenshot displays the 'iFLORA Administration' web application. On the left is a sidebar menu with options: iFlora, Systemverwaltung, Administration, Datenbearbeitung, Auswertungen, Schnittstellen, Download, and ORI-Upload. The main content area is titled 'UPLOAD ORI-Daten' and 'Datensatz Bearbeiten'. It features three input fields: 'Artcode' with the value 'VERAGRE', 'Art' with 'Veronica agrestis L.', and 'Artmächtigkeit' with a '+' sign. Below these fields are two buttons: 'speichern' and 'löschen'. A 'zurück' button is located in the top right corner of the main area. The top header bar is green with 'iFLORA Administration' and red with 'iFloraPublic@test R0.9'. The bottom status bar shows 'iFlora Administration - ADMIN' and '© shInformatics'.

Abbildung 86 - iFlora Administration – ORI-Schnittstelle 4

- Daten importieren – Sind die geladenen Daten in Ordnung, können sie nach Auswahl des Projektes und der Methode (dzt. ist nur die Methode nach „Braun-Blanquet“ möglich) über den Button „Speichern“ in die verschiedenen Tabellen (Aufnahme, Vegetationsdaten) geladen werden (siehe Abbildung 87). Dabei werden die Daten gleichzeitig aus der Temporären Tabelle gelöscht!

iFLORA Administration iFloraPublic@test
R0.9

Schnittstellen

Code	Name	Status	Ändern
CARFLAC	[Carex flacca Schreb.]	+	ändern
CARLIPA	[Carex liparocarpos Gaudin]	+	ändern
LISOVAT	[Listera ovata (L.)R.Br.]	-	ändern
COLAUTU	[Colchicum autumnale L.]	-	ändern
ONOSPPIA	[Ononis spinosa L.]	+	ändern
CENJAAN	[Centaurea jacea ssp. angustifolia Greml]	+	ändern
EUPCYPA	[Euphorbia cyparissias L.]	+	ändern
ROSCANI	[Rosa canina L.]	-	ändern
CERSEMI	[Cerastium semidecandrum L.]	+	ändern
VERAGRE	[Veronica agrestis L.]	+	ändern

Projekt:

Methode:

Der Uploadprozess kann einige Minuten dauern!

iFlora Administration - ADMIN © shInformatics

Abbildung 87 - iFlora Administration – ORI-Schnittstelle 5

6.2.7.1 Codierung

Untermenüpunkt: ORI-Upload – der Sourcecode befindet sich in der Datei „/iFlora/admin/ADMINUploadORI.php“ (siehe Kapitel 9.3.1.2.16 Controller – „ORI-Upload“).

7 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Konzept einer Software für die zentrale Erfassung und Auswertung von vegetationsökologischen Daten erarbeitet. Mit diesem Konzept wurde versucht eine einheitliche Basis für die weiteren Arbeiten in der Abteilung Vegetationsökologie zu schaffen.

Es gibt noch einige Konzepte, die im Rahmen dieses Thema interessant wären:

7.1 Auswertungen

Im Rahmen dieser Arbeit konnte aus Zeitgründen nicht definiert werden, wie die Auswertungen der Daten der verschiedenen Module im Detail ausgeführt werden soll. Der Benutzer soll schrittweise die Daten auswerten können. Zusätzlich sollen die Auswertungen auch die geografischen Daten einbeziehen und einige Auswahlsschritte sollen auch über dynamische Karten erfolgen können.

7.2 Modul GIS

Zweck dieses Moduls ist die Verwaltung von geographischen Daten der verschiedenen Objekte von **iFlora**. Dieses Modul ist kein Ersatz für eine GIS-Anwendung, sondern dient vor allem der zentralen Datenspeicherung der GIS-Daten aller Projekte, die in der Software **iFlora** verwaltet werden und der Darstellung dieser Daten im Internet. Die Daten werden für die Bearbeitung in einem GIS-Programm als Shapefile exportiert, in einem GIS Programm (z.B.: ArcInfo der Firma ESRI) bearbeitet und das Ergebnis wieder ins GIS Modul der Software iFlora geladen. Zusätzlich könnte man noch eine direkte Verbindung zwischen den Daten in der Oracle Datenbank von iFlora und dem jeweiligen GIS-Programm andenken (z.B.: ArcSDE der Firma ESRI, seit 2007 in den ArcGIS Server integriert). Vorteil ist die zentrale Datenhaltung aller Daten eines Projektes und daraus folgende zentrale Datensicherung aller Projektdaten. Im Modul **AUFNAHME** gibt es bereits eine Tabelle (**AFN_GISOBJEKT**), die alle GIS-Koordinaten der verschiedenen Objekte dieses Moduls speichert.

Diese GIS-Daten können für die Darstellung der Objekte in dynamischen Karten im Internet genutzt werden. Für die Darstellung der Projektdaten inkl. der GIS Daten im Internet wurde der UMN MapServer (<http://ms.gis.umn.edu/>) ausgewählt.

7.3 Modul Botanische Namen

Das Modul **BOTNAME** soll die Möglichkeit bieten, die verschiedenen wissenschaftlichen Pflanzennamen zu verwalten (inkl. der Literaturzitate) und ihre historischen Veränderungen zu dokumentieren.

Diese wissenschaftlichen Namen der Pflanzen sind die zentrale Komponente für das Modul **AUFNAHME**. Über die Beziehungen zwischen den verschiedenen Arten sollen Aufnahmen, die zeitlich lange auseinanderliegen und dessen Artenlisten auf verschiedenen Floranwerken basieren, vergleichbar gemacht werden.

Zusätzlich können bei den einzelnen Arten noch Stammdaten definiert werden. Hier können z.B. die ökologischen Zeigerwerte, die Lebens- oder Wuchsform, ein Kommentar usw. der jeweiligen Art abgespeichert werden (ELLENBERG 1996, S1020).

Über eine eigene Infrastruktur sollen die nomenklatorischen Typen (Herbarexemplare) der jeweiligen Arten verwaltet werden können (ADLER 1994, S 34).

Für alle diese Aufgaben muss das vorgestellte erweiterte Konzept (Kapitel 5.4.2.1 „Erweiterung des Datenmodells BOTNAME“) weiterentwickelt werden.

7.4 Modul Pflanzensoziologie

In der Natur kommen bestimmte Vegetationsmuster vor, die auch Pflanzengesellschaften genannt werden. Die Erfassung, Typisierung und Klassifizierung von Pflanzengesellschaften ist die Aufgabe der Pflanzensoziologie. Dieses Modul soll die Möglichkeit bieten, die verschiedenen Pflanzengesellschaften zu verwalten (inkl. der Literaturzitate) und ihre historischen Veränderungen zu dokumentieren. Weiters soll es die Möglichkeit geben, die verschiedenen historischen Pflanzengesellschaften miteinander in Beziehung zu setzen. Zusätzlich soll die Möglichkeit bestehen, den verschiedenen Aufnahmen bzw. Aufnahmeflächen im Modul **AUFNAHME** (d.h. einem Objekt vom Typ Aufnahme oder Aufnahmefläche) eine Pflanzengesellschaft zuzuordnen. Zusätzlich kann man z.B.: über Stammdaten

die Charakterarten, usw. als Stammdaten anlegen, bei Bedarf können diese Daten auch mit den Daten anderer Module vernetzt werden (z.B.: Charakterarten mit dem Modul **BOTNAME**).

7.5 Modul Dokumente

Es soll die Möglichkeit geben, zu den Objekten der verschiedenen Module Dokumente zu speichern und diese miteinander zu vernetzen. Dazu soll das zentrale Modul **DOKUMENT** entwickelt werden. Diese Dokumente können Kommentare als auch Literaturartikel sein. Ein Dokument besteht aus einer Zusammenfassung und einem Inhalt, der wahlweise aus einem Text, einem Dokument (pdf Datei) oder einem Link auf ein Dokument einer elektronischen Zeitschrift bestehen kann. Diese Dokumente können jedem Objekt aller Module zugeordnet und untereinander vernetzt werden.

7.6 Modul Sitemap

Die aktuellen Daten der Anwendung von iFlora sollen über eine dynamische Sitemap den Benutzern zur Verfügung gestellt werden. Zusätzlich können über das „Sitemaps Protocol“ (<http://www.sitemaps.org/>) die verfügbaren HTML-Seiten eines Webserverns auch einer Suchmaschine (z.B.: Google) zur Verfügung gestellt werden. Diese Daten werden den Suchmaschinen (lt. „Sitemaps Protocol“) im XML-Format angeboten. Das ist bei dynamischen Dateninhalten besonders wichtig, da ja die Inhalte nicht als statische HTML Seiten mit fixen Links zur Verfügung stehen, sondern bei jedem Abruf dynamisch zusammengestellt werden.

7.7 Modul Webservices

Dieses Modul stellt die Funktionen der Webservices zentral zur Verfügung. Bei einem Webservice handelt es sich um eine Maschinenabfrage – andere Anwendungen können bestimmte Daten von der Anwendung iFlora abrufen. Dieser Abruf ist betriebs-, system-, programmiersprachen- und hardwareunabhängig. Das heißt die Daten können jeder Anwendung zur Verfügung gestellt werden, wobei es bestimmte Daten nur für registrierte Anwendungen gibt, andere Daten dagegen für alle Anwendungen zugänglich sein sollen. Es soll dabei die Möglichkeit geben, die Daten der verschiedenen Module von **iFlora** automatisch abzufragen.

8 Literatur

- ADLER, W., Oswald K., Fischer R., 1994, „Exkursionsflora von Österreich“, Stuttgart.
- BALZERT, Heide, 1999, „Lehrbuch der Objektmodellierung“, Heidelberg.
- BALZERT, Helmut, 1996, „Lehrbuch der Software-Technik – Band 1“, Heidelberg.
- ELLENBERG, Heinz, 1996⁵, „Vegetation Mitteleuropas mit den Alpen“, Stuttgart.
- GLAVAC, Vjekoslav, 1996, „Vegetationsökologie“, Jena.
- HAMMERSCHALL, Ulrike, 2005, „Verteilte Systeme und Anwendungen“, München.
- HITZ, M., Kappel G., Kapsammer E., Retschitzegger W., 2005, „UML@Work“, Heidelberg.
- RECKNAGEL, Friedrich (Hsg.), 2003, „Ecological Informatics“, Berlin.
- SPENGER, Silvia J., 2000, „Bioinformatics in the Information Age“, Science Vol 287 p 1221f.
- TOWNSEND C. R., Harper J. L., Begon M. E., 2003, „Ökologie“, Berlin.

9 ANHANG

9.1 Datenbankschema

9.1.1 CREATE SCHEMA - iFlora

```
REM *****
REM ***** SCHEMA: iFLORA *****
REM *****

REM *****
REM ** USER: SH$IFLORA
REM *****
create user SH$IFLORA identified by abc
default tablespace SH_FLORA
temporary tablespace TEMP;

grant CONNECT, RESOURCE to SH$IFLORA;

REM *****ENDE*****
```

9.2 Datenbankstruktur

9.2.1 CREATE STRUCTURE – Modul AUFNAHME

```
REM *****
REM ***** MODUL: AUFNAHME *****
REM *****

REM *****
REM *****TABLE's*****
REM *****

REM *****
REM ** TABLE(1): SH$IFLORA.AFN_PROJEKT
REM *****
create table SH$IFLORA.AFN_PROJEKT(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_PRJID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    PRJNAME            VARCHAR2(30) NOT NULL,
    PRJNR              VARCHAR2(15),
    PRJKOMMENTAR       VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_PROJEKT
REM *****
create sequence SH$IFLORA.SEQ_AFN_PROJEKT;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_PROJ_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_PROJEKT_NAME
on SH$IFLORA.AFN_PROJEKT(PRJNAME);
```

```

REM *****
REM ** TABLE(2): SH$IFLORA.AFN_PARAMETER
REM *****
create table SH$IFLORA.AFN_PARAMETER(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_PARID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER             NUMBER(13,3) NOT NULL,
    DS_STATUS           VARCHAR2(3) NOT NULL,
    PRJID              NUMBER(13,3) CONSTRAINT FK_AFN_PAR_PRJID REFERENCES
                        SH$IFLORA.AFN_PROJEKT(TID),
    WLKID              NUMBER(13,3) CONSTRAINT FK_AFN_PAR_WLKID REFERENCES
                        SH$IFLORA.AFN_WERTEKLASSEN(TID),
    ISAIID              NUMBER(13,3) CONSTRAINT FK_AFN_PAR_ISAIID REFERENCES
                        SH$IFLORA.AFN_ISINART(TID),
    LINKOBAID           NUMBER(13,3) CONSTRAINT FK_AFN_PAR_LINKOBAID REFERENCES
                        SH$IFLORA.AFN_OBJEKTART(TID),
    PARNAME             VARCHAR2(30) NOT NULL,
    PARNR              VARCHAR2(15),
    PARFORMAT           VARCHAR2(10) NOT NULL,
    PARKOMMENTAR        VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_PARAMETER
REM *****
create sequence SH$IFLORA.SEQ_AFN_PARAMETER;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_PARNAME
REM *****
create unique index SH$IFLORA.IND_AFN_PARNAME
    on SH$IFLORA.AFN_PARAMETER(PARNAME);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_PARNR
REM *****
create unique index SH$IFLORA.IND_AFN_PARNR
    on SH$IFLORA.AFN_PARAMETER(PARNR);

REM *****
REM ** TABLE(3): SH$IFLORA.AFN_OBJEKT
REM *****
create table SH$IFLORA.AFN_OBJEKT(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_OBJID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER             NUMBER(13,3) NOT NULL,
    DS_STATUS           VARCHAR2(3) NOT NULL,
    PRJID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_OBJ_PROJ REFERENCES
                        SH$IFLORA.AFN_PROJEKT(TID),
    OBJNAME            VARCHAR2(30) NOT NULL,
    OBJART              VARCHAR2(10),
    OBJGISMOMODE        VARCHAR2(10),
    OBJKOMMENTAR        VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_OBJEKT
REM *****
create sequence SH$IFLORA.SEQ_AFN_OBJEKT;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_OBJ_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_OBJEKT_NAME
    on SH$IFLORA.AFN_OBJEKT(OBJNAME);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_OBJ_PROJID
REM *****
create index SH$IFLORA.IND_OBJ_PROJID
    on SH$IFLORA.AFN_OBJEKT(PROJID);

```

```
REM *****
REM ** TABLE(4): SH$IFLORA.AFN_GISOBJEKT
REM *****
create table SH$IFLORA.AFN_GISOBJEKT(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_GISID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    OBJID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_GIS_OBJ REFERENCES
                        SH$IFLORA.AFN_OBJEKT(TID),
    GISLOCPOINT        SDO_GEOMETRY,
    GISLOCLINE         SDO_GEOMETRY,
    GISLOCAREA         SDO_GEOMETRY);

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_GISOBJEKT
REM *****
create sequence SH$IFLORA.SEQ_AFN_GISOBJEKT;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_GIS_OBJID
REM *****
create index SH$IFLORA.IND_GIS_OBJID
  on SH$IFLORA.AFN_GISOBJEKT(OBJID);

REM *****
REM ** TABLE(5): SH$IFLORA.AFN_STAMMDATEN
REM *****
create table SH$IFLORA.AFN_STAMMDATEN(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_STDID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    OBJID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_STD_OBJID REFERENCES
                        SH$IFLORA.AFN_OBJEKT(TID),
    PARID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_STD_PARID REFERENCES
                        SH$IFLORA.AFN_PARAMETER(TID),
    STDWERT            VARCHAR2(255) NOT NULL);

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_STAMMDATEN
REM *****
create sequence SH$IFLORA.SEQ_AFN_STAMMDATEN;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_STD_OBJ
REM *****
create index SH$IFLORA.IND_AFN_STD_OBJ
  on SH$IFLORA.AFN_STAMMDATEN(OBJID);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_STD_PAR
REM *****
create index SH$IFLORA.IND_AFN_STD_PAR
  on SH$IFLORA.AFN_STAMMDATEN(PARID);

REM *****
REM ** TABLE(6): SH$IFLORA.AFN_ISINART
REM *****
create table SH$IFLORA.AFN_ISINART(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_ISAID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    PRJID             NUMBER(13,3) CONSTRAINT FK_AFN_ISA_PROJ REFERENCES
                        SH$IFLORA.AFN_PROJEKT(TID),
    ISANAME            VARCHAR2(30) NOT NULL,
```

```

        ISAKUERZEL      VARCHAR(4) NOT NULL,
        ISASTRUKTUR     VARCHAR(150) NOT NULL,
        ISAKOMMENTAR    VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_ISINART
REM *****
create sequence SH$IFLORA.SEQ_AFN_ISINART;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISA_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_ISA_NAME
on SH$IFLORA.AFN_ISINART(ISANAME);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISA_KUERZEL
REM *****
create unique index SH$IFLORA.IND_AFN_ISA_KUERZEL
on SH$IFLORA.AFN_ISINART(ISAKUERZEL);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISA_PROJID
REM *****
create index SH$IFLORA.IND_AFN_ISA_PROJID
on SH$IFLORA.AFN_ISINART(PROJID);

REM *****
REM ** TABLE(7): SH$IFLORA.AFN_ISIN
REM *****
create table SH$IFLORA.AFN_ISIN(
        TID              NUMBER(13,3) CONSTRAINT PK_AFN_ISINID PRIMARY KEY,
        DS_DATE          DATE NOT NULL,
        DS_USER           NUMBER(13,3) NOT NULL,
        DS_STATUS         VARCHAR2(3) NOT NULL,
        ISAIID            NUMBER(13,3) CONSTRAINT FK_AFN_ISIN_ISA REFERENCES SH$IFLORA.AFN_ISINART(TID),
        OBJID             NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_ISIN_OBJID REFERENCES
                        SH$IFLORA.AFN_OBJEKT(TID),
        ISINOBJID         NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_ISIN_ISINOBJID REFERENCES
                        SH$IFLORA.AFN_OBJEKT(TID));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_ISIN
REM *****
create sequence SH$IFLORA.SEQ_AFN_ISIN;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISIN_OBJID
REM *****
create index SH$IFLORA.IND_AFN_ISIN_OBJID
on SH$IFLORA.AFN_ISIN(OBJID);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISIN_ISINOBJID
REM *****
create index SH$IFLORA.IND_AFN_ISIN_ISINOBJID
on SH$IFLORA.AFN_ISIN(ISINOBJID);

REM *****
REM ** TABLE(8): SH$IFLORA.AFN_OBJEKTART
REM *****
create table SH$IFLORA.AFN_ISINART(
        TID              NUMBER(13,3) CONSTRAINT PK_AFN_OBAID PRIMARY KEY,
        DS_DATE          DATE NOT NULL,
        DS_USER           NUMBER(13,3) NOT NULL,
        DS_STATUS         VARCHAR2(3) NOT NULL,
        PRJID            NUMBER(13,3) CONSTRAINT FK_AFN_ISA_PROJ REFERENCES
                        SH$IFLORA.AFN_PROJEKT(TID),
        OBANAME           VARCHAR2(30) NOT NULL,

```



```

        OBAKUERZEL      VARCHAR2(10),
        OBAKOMMENTAR VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_OBJEKTART
REM *****
create sequence SH$IFLORA.SEQ_AFN_OBJEKTART;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISA_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_OBA_NAME
  on SH$IFLORA.AFN_OBJEKTART(OBANAME);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISA_PROJID
REM *****
create index SH$IFLORA.IND_AFN_OBA_PROJID
  on SH$IFLORA.AFN_OBJEKTART(PROJID);

REM *****
REM ** TABLE(9): SH$IFLORA.AFN_ISINOBJART
REM *****
create table SH$IFLORA.AFN_ISINOBJART(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_INOBAID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    ISAID              NUMBER(13,3) CONSTRAINT FK_AFN_ISINOBA_ISA REFERENCES
                        SH$IFLORA.AFN_ISINART(TID),
    OBAID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_OBA_OBA REFERENCES
                        SH$IFLORA.AFN_OBJEKTART(TID),
    ISINOBAID          NUMBER(13,3) NOT NULL CONSTRAINT FK_ARN_ISINOBA_OBA REFERENCES
                        SH$IFLORA.AFN_OBJEKTART(TID),
    ISINOBAKOMMENTAR   VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_ISINOBJEKTART
REM *****
create sequence SH$IFLORA.SEQ_AFN_ISINOBJART;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_INOBA_OBAID
REM *****
create index SH$IFLORA.IND_AFN_ISINOBA_OBAID
  on SH$IFLORA.AFN_ISINOBJEKTART(OBAID);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_ISINOBA_ISINOBAID
REM *****
create index SH$IFLORA.IND_AFN_ISINOBA_ISINOBAID
  on SH$IFLORA.AFN_ISINOBJEKTART(ISINOBAID);

REM *****
REM ** TABLE(10): SH$IFLORA.AFN_AUFMETHODE
REM *****
create table SH$IFLORA.AFN_AUFMETHODE(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_MTHID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    MTHNAME            VARCHAR2(30) NOT NULL,
    MTHKUERZEL         VARCHAR2(10),
    MTHKOMMENTAR       VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_AUFMETHODE
REM *****
create sequence SH$IFLORA.SEQ_AFN_AUFMETHODE;

```

```

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_MTH_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_MTH_NAME
on SH$IFLORA.AFN_AUFMETHODE(MTHNAME);

REM *****
REM ** TABLE(11): SH$IFLORA.AFN_ARTEN
REM *****
create table SH$IFLORA.AFN_ARTEN(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_SPCID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    BOT_FARID          NUMBER(13,3) NOT NULL,
    BOT_FLWID          NUMBER(13,3) NOT NULL,
    SPCNAME            VARCHAR2(100) NOT NULL,
    SPCFLWNAME         VARCHAR2(255),
    SPCKOMMENTAR       VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_ARTEN
REM *****
create sequence SH$IFLORA.SEQ_AFN_ARTEN;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_MTH_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_MTH_NAME
on SH$IFLORA.AFN_ARTEN(SPCNAME);

REM *****
REM ** TABLE(12): SH$IFLORA.AFN_VEGDATEN
REM *****
create table SH$IFLORA.AFN_VEGDATEN(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_VEGID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    OBJID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_VEG_OBJ REFERENCES
                        SH$IFLORA.AFN_OBJEKT(TID),
    MTHID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_VEG_MTH REFERENCES
                        SH$IFLORA.AFN_AUFMETHODE(TID),
    SPCID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_VEG_SPC REFERENCES
                        SH$IFLORA.AFN_ARTEN(TID),
    VEGWERT            VARCHAR2(255) NOT NULL);

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_VEGDATEN
REM *****
create sequence SH$IFLORA.SEQ_AFN_VEGDATEN;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_VEG_MTH
REM *****
create index SH$IFLORA.IND_AFN_VEG_MTH
on SH$IFLORA.AFN_VEGDATEN (MTHID);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_VEG_OBJ
REM *****
create index SH$IFLORA.IND_AFN_VEG_OBJ
on SH$IFLORA.AFN_VEGDATEN (OBJID);

```

```

REM *****
REM ** TABLE(13): SH$IFLORA.AFN_TEMP_VEGDAT
REM *****
create table SH$IFLORA.AFN_TEMP_VEGDAT (
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_TVGLID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    TVGAFNNR          VARCHAR2(10),
    TVGSCHICHT         VARCHAR2(1),
    TVGARTCODE         VARCHAR2(10),
    TVGVEGWERT         VARCHAR2(1));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_TEMP_VEGDAT
REM *****
create sequence SH$IFLORA.SEQ_AFN_TEMP_VEGDAT;

REM *****
REM ** TABLE(14): SH$IFLORA.AFN_WERTEKLASSEN
REM *****
create table SH$IFLORA.AFN_WERTEKLASSEN(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_WTKID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    WTKNAME            VARCHAR2(30) NOT NULL,
    WTKBEZ             VARCHAR2(100),
    WTKFORMAT          VARCHAR2(10) NOT NULL,
    WTKKOMMENTAR       VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_WERTEKLASSEN
REM *****
create sequence SH$IFLORA.SEQ_AFN_WERTEKLASSEN;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_WTK_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_WTK_NAME
    on SH$IFLORA.AFN_WERTEKLASSEN(WTKNAME);

REM *****
REM ** TABLE(15): SH$IFLORA.AFN_WERTELISTE
REM *****
create table SH$IFLORA.AFN_WERTELISTE(
    TID                NUMBER(13,3) CONSTRAINT PK_AFN_WRTID PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    WTKID              NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_WRT_WTKID REFERENCES
        SH$IFLORA.AFN_WERTEKLASSEN(TID),
    WRTWERT            VARCHAR2(255) NOT NULL,
    WRTPRIORITAET      NUMBER(4));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_WERTELISTE
REM *****
create sequence SH$IFLORA.SEQ_AFN_WERTELISTEN;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_WRT_WTK
REM *****
create index SH$IFLORA.IND_AFN_WRT_WTK
    on SH$IFLORA.AFN_WERTELISTE(WTKID);

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
REM *****
REM ** TABLE(2a): SH$IFLORA.AFN_PARAMETER
REM *****
alter table SH$IFLORA.AFN_PARAMETER(
    WLKID          NUMBER(13,3) CONSTRAINT FK_AFN_PAR_WTKID REFERENCES
                    SH$IFLORA.AFN_WERTEKLASSEN(TID),
    ISAIID         NUMBER(13,3) CONSTRAINT FK_AFN_PAR_ISAIID REFERENCES
                    SH$IFLORA.AFN_ISINART(TID));

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_PAR_WTK
REM *****
create index SH$IFLORA.IND_AFN_PAR_WTK
on SH$IFLORA.AFN_PARAMETER (WTKID);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_PAR_ISA
REM *****
create index SH$IFLORA.IND_AFN_PAR_ISA
on SH$IFLORA.AFN_PARAMETER (ISAIID);

REM *****
REM *****HILFs TABLE's*****
REM *****

REM *****
REM ** TABLE(16): SH$IFLORA.AFN_H_FORMULAR
REM *****
create table SH$IFLORA.AFN_H_FORMULAR (
    TID            NUMBER(13,3) CONSTRAINT PK_AFN_H_FRMID PRIMARY KEY,
    DS_DATE        DATE NOT NULL,
    DS_USER        NUMBER(13,3) NOT NULL,
    DS_STATUS      VARCHAR2(3) NOT NULL,
    PRJID          NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_H_FRM_PRJID REFERENCES
                    SH$IFLORA.AFN_PROJEKT(TID),
    OBAID          NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_H_FRM_OBAID REFERENCES
                    SH$IFLORA.AFN_OBJEKTART(TID),
    FRMNAME        VARCHAR2(30) NOT NULL);

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_H_FORMULAR
REM *****
create sequence SH$IFLORA.SEQ_AFN_H_FORMULAR T;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_H_FRM_NAME
REM *****
create unique index SH$IFLORA.IND_AFN_H_FRM_NAME
on SH$IFLORA.AFN_H_FORMULAR(FRMNAME);

REM *****
REM ** TABLE(17): SH$IFLORA.AFN_H_FORMROW
REM *****
create table SH$IFLORA.AFN_H_FORMROW (
    TID            NUMBER(13,3) CONSTRAINT PK_AFN_H_FMRID PRIMARY KEY,
    DS_DATE        DATE NOT NULL,
    DS_USER        NUMBER(13,3) NOT NULL,
    DS_STATUS      VARCHAR2(3) NOT NULL,
    FRMID          NUMBER(13,3) NOT NULL CONSTRAINT FK_AFN_H_FMR_FRM REFERENCES
                    SH$IFLORA.AFN_H_FORMULAR(TID),
    PARID          NUMBER(13,3) CONSTRAINT FK_AFN_H_FMR_PARID REFERENCES
                    SH$IFLORA.AFN_PARAMETER(TID),
    FMRLABEL       VARCHAR2(130),
    FMRSTYLE       NUMBER(1),
    FMRPOSITION    NUMBER(4));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AFN_H_FORMROW
REM *****
```

```

create sequence SH$IFLORA.SEQ_AFN_H_FORMROW;

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_H_FMR_FRM
REM *****
create index SH$IFLORA.IND_AFN_H_FMR_FRM
  on SH$IFLORA.AFN_H_FORMROW (FRMID);

REM *****
REM ** INDEX: SH$IFLORA.IND_AFN_H_FMR_PAR
REM *****
create index SH$IFLORA.IND_AFN_H_FMR_PAR
  on SH$IFLORA.AFN_H_FORMROW (PARID);

REM *****ENDE*****

```

9.2.2 CREATE STRUCTURE – Modul BOTNAME

```

REM *****
REM *****  MODUL: BOTNAME *****
REM *****

REM *****
REM *****TABLE's*****
REM *****

REM *****
REM ** TABLE: SH$IFLORA.BOTGATTNAME
REM *****
create table SH$IFLORA.BOTGATTNAME(
  TID                NUMBER(13,3) CONSTRAINT PK_GATT_NAME PRIMARY KEY,
  DS_DATE            DATE NOT NULL,
  DS_USER            NUMBER(13,3) NOT NULL,
  DS_STATUS          VARCHAR2(3) NOT NULL,
  DS_PROJ_ID         NUMBER(13,3),
  GATT_NAME          VARCHAR2(30) NOT NULL,
  KOMMENTAR         VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_GATT_NAME
REM *****
create sequence SH$IFLORA.SEQ_GATT_NAME;

REM *****
REM ** INDEX: SH$IFLORA.IND_GATT_NAME
REM *****
create unique index SH$IFLORA.IND_GATT_NAME
  on SH$IFLORA.GATT_NAME(GATT_NAME);

REM *****
REM ** TABLE: SH$IFLORA.ART_NAME
REM *****
create table SH$IFLORA.ART_NAME(
  TID                NUMBER(13,3) CONSTRAINT PK_ART_NAME PRIMARY KEY,
  DS_DATE            DATE NOT NULL,
  DS_USER            NUMBER(13,3) NOT NULL,
  DS_STATUS          VARCHAR2(3) NOT NULL,
  DS_PROJ_ID         NUMBER(13,3),
  GATT_ID            NUMBER(13,3) NOT NULL CONSTRAINT FK_ART_GATT_NAME REFERENCES
    SH$IFLORA.GATT_NAME(TID),
  ART_NAME           VARCHAR2(30) NOT NULL,
  PSEUDO             VARCHAR2(1),
  KOMMENTAR         VARCHAR2(4000));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_ART_NAME
REM *****
create sequence SH$IFLORA.SEQ_ART_NAME;

```

```
REM *****
REM ** INDEX: SH$IFLORA.INT_ART_NAME
REM *****
create unique index SH$IFLORA.IND_ART_NAME
  on SH$IFLORA.ART_NAME(ART_NAME,GATT_ID);

REM *****
REM ** TABLE: SH$IFLORA.SSP_NAME
REM *****
create table SH$IFLORA.SSP_NAME(
  TID          NUMBER(13,3) CONSTRAINT PK_SPP_NAME PRIMARY KEY,
  DS_DATE      DATE NOT NULL,
  DS_USER      NUMBER(13,3) NOT NULL,
  DS_STATUS    VARCHAR2(3) NOT NULL,
  DS_PROJ_ID   NUMBER(13,3),
  ART_ID       NUMBER(13,3) NOT NULL CONSTRAINT FK_SPP_ART_NAME REFERENCES
  SH$IFLORA.ART_NAME(TID),
  NAME         VARCHAR2(30) NOT NULL);

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_SSP_NAME
REM *****
create sequence SH$IFLORA.SEQ_SSP_NAME;

REM *****
REM ** INDEX: SH$IFLORA.INT_SSP_NAME
REM *****
create unique index SH$IFLORA.IND_SSP_NAME
  on SH$IFLORA.SSP_NAME(NAME, ART_ID);

REM *****
REM ** TABLE: SH$IFLORA.AGG_NAME
REM *****
create table SH$IFLORA.AGG_NAME(
  TID          NUMBER(13,3) CONSTRAINT PK_AGG_NAME PRIMARY KEY,
  DS_DATE      DATE NOT NULL,
  DS_USER      NUMBER(13,3) NOT NULL,
  DS_STATUS    VARCHAR2(3) NOT NULL,
  DS_PROJ_ID   NUMBER(13,3),
  ART_ID       NUMBER(13,3) NOT NULL CONSTRAINT FK_AGG_ART_NAME REFERENCES
  SH$IFLORA.ART_NAME(TID),
  ART_ID_AGG   NUMBER(13,3) NOT NULL CONSTRAINT FK_AGG_ART_AGG REFERENCES
  SH$IFLORA.ART_NAME(TID),
  KOMMENTAR    VARCHAR2(400));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AGG_NAME
REM *****
create sequence SH$IFLORA.SEQ_AGG_NAME;

REM *****
REM ** TABLE: SH$IFLORA.LITERATUR
REM *****
create table SH$IFLORA.LITERATUR(
  TID          NUMBER(13,3) CONSTRAINT PK_LITERATUR PRIMARY KEY,
  DS_DATE      DATE NOT NULL,
  DS_USER      NUMBER(13,3) NOT NULL,
  DS_STATUS    VARCHAR2(3) NOT NULL,
  DS_PROJ_ID   NUMBER(13,3),
  NAME         VARCHAR2(100) NOT NULL,
  JAHR         DATE NOT NULL,
  AUFLAGE      NUMBER(4),
  ISBN         VARCHAR2(20),
  KOMMENTAR    VARCHAR2(400));
```

```

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_LITERATUR
REM *****
create sequence SH$IFLORA.SEQ_LITERATUR;

```

```

REM *****
REM ** TABLE: SH$IFLORA.AUTOR
REM *****
create table SH$IFLORA.AUTOR(
    TID                NUMBER(13,3) CONSTRAINT PK_AUTOR PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    DS_PROJ_ID         NUMBER(13,3),
    NAME               VARCHAR2(50) NOT NULL,
    VORNAME            VARCHAR2(50),
    KOMMENTAR          VARCHAR2(400));

```

```

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AUTOR
REM *****
create sequence SH$IFLORA.SEQ_AUTOR;

```

```

REM *****
REM ** TABLE: SH$IFLORA.AUTOR_LITERATUR_BEZ
REM *****
create table SH$IFLORA.AUTOR_LITERATUR_BEZ(
    TID                NUMBER(13,3) CONSTRAINT PK_AUTOR_LIT_BEZ PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    DS_PROJ_ID         NUMBER(13,3),
    AUTOR_ID           NUMBER(13,3) NOT NULL CONSTRAINT FK_AUTOR_LIT REFERENCES
    SH$IFLORA.AUTOR(TID),
    LITERATUR_ID       NUMBER(13,3) NOT NULL CONSTRAINT FK_LIT_AUT_LIT REFERENCES
    SH$IFLORA.LITERATUR(TID));

```

```

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_AUTOR_LITERATUR_BEZ
REM *****
create sequence SH$IFLORA.SEQ_AUTOR_LITERATUR_BEZ;

```

```

REM *****
REM ** TABLE: SH$IFLORA.FLORA_ART_BEZ
REM *****
create table SH$IFLORA.FLORA_ART_BEZ(
    TID                NUMBER(13,3) CONSTRAINT PK_FLORA_ART_BEZ PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    DS_STATUS          VARCHAR2(3) NOT NULL,
    DS_PROJ_ID         NUMBER(13,3),
    ART_ID             NUMBER(13,3) CONSTRAINT FK_ART_FLORA_ART_BEZ REFERENCES
    SH$IFLORA.ART_NAME(TID),
    SSP_ID             NUMBER(13,3) CONSTRAINT FK_SSP_FLORA_ART_BEZ REFERENCES
    SH$IFLORA.SSP_NAME(TID),
    AGG_ID             NUMBER(13,3) CONSTRAINT FK_AGG_FLORA_ART_BEZ REFERENCES
    SH$IFLORA.AGG_NAME(TID),
    LITERATUR_ID       NUMBER(13,3) NOT NULL CONSTRAINT FK_LIT_FLORA_ART_BEZ REFERENCES
    SH$IFLORA.LITERATUR(TID));

```

REM evt. fehlt noch Verknüpfung zu HYBRIDEN, evt. sogar GATT_ID => eigene Felder!

```

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_FLORA_ART_BEZ
REM *****
create sequence SH$IFLORA.SEQ_FLORA_ART_BEZ;

```

REM *****ENDE*****

9.2.3 CREATE STRUCTURE – Modul METADATEN

REM *****
REM ***** MODUL: AUFNAHME *****
REM *****

REM *****
REM *****TABLE's*****
REM *****

REM *****
REM ** TABLE: SH\$IFLORA. ADM_RECHT
REM *****

```
create table ADM_RECHT(
    TID                FLOAT,
    DS_DATE            DATE NOT NULL,
    DS_USER            FLOAT NOT NULL,
    NAME               VARCHAR(30),
    KUERZEL            VARCHAR(5),
    KOMMENTAR          VARCHAR(255),
    PRIORITAET         INT(29),
    PRIMARY KEY (TID));
```

REM *****
REM ** TABLE: SH\$IFLORA. ADM_USER_RECHTE
REM *****

```
create table ADM_USER_RECHTE(
    TID                FLOAT,
    DS_DATE            DATE NOT NULL,
    DS_USER            FLOAT NOT NULL,
    USERID             FLOAT REFERENCES ADM_USERNAME(TID),
    RECHTEID           FLOAT REFERENCES ADM_RECHT(TID),
    PROJEKTID          FLOAT REFERENCES AFN_PROJEKT(TID),
    PRIMARY KEY (TID));
```

REM *****
REM ** TABLE: SH\$IFLORA. ADM_USERNAME
REM *****

```
create table ADM_USERNAME(
    TID                FLOAT,
    DS_DATE            DATE NOT NULL,
    DS_USER            FLOAT NOT NULL,
    NAME               VARCHAR(40) NOT NULL,
    VORNAME            VARCHAR(40),
    USERNAME           VARCHAR(40) NOT NULL,
    E_MAIL             VARCHAR(40),
    STATUS             VARCHAR(10),
    PWD                VARCHAR(100),
    PRIMARY KEY (TID));
```

REM *****
REM ** TABLE: SH\$IFLORA.DS_STATUS => Tabellenstatus(abgeschlossen,...)!
REM *****

```
create table SH$IFLORA.DS_STATUS (
    TID                NUMBER(13,3) CONSTRAINT PK_STATUS PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    NAME               VARCHAR2(40) NOT NULL,
    KUERZEL            VARCHAR2(3) NOT NULL,
    KOMMENTAR          VARCHAR2(4000));
```

REM *****
REM ** SEQUENCE: SH\$IFLORA.SEQ_DS_STATUS
REM *****
create sequence SH\$IFLORA.SEQ_DS_STATUS;


```

REM *****
REM ** INDEX: SH$IFLORA.IND_SEQ_DS_STATUS
REM *****
create unique index SH$IFLORA.IND_DS_STATUS
on SH$IFLORA.DS_STATUS(KUERZEL);

REM *****
REM ** TABLE: SH$IFLORA.PROJEKT
REM *****
create table SH$IFLORA.PROJEKT (
    TID                NUMBER(13,3) CONSTRAINT PK_PROJEKT PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    NAME               VARCHAR2(40) NOT NULL,
    KOMMENTAR          VARCHAR2(4000));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_PROJEKT
REM *****
create sequence SH$IFLORA.SEQ_PROJEKT;

REM *****
REM ** TABLE: SH$IFLORA.TABELLE
REM *****
create table SH$IFLORA.TABELLE(
    TID                NUMBER(13,3) CONSTRAINT PK_TABELLE PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    NAME               VARCHAR2(40) NOT NULL,
    PROJEKT_ID         NUMBER(13,3),
    KOMMENTAR          VARCHAR2(4000)
    STATUS             VARCHAR2(8));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_TABELLE
REM *****
create sequence SH$IFLORA.SEQ_TABELLE;

REM *****
REM ** TABLE: SH$IFLORA.TABFELDER => neue Tabellenfelder!
REM *****
create table SH$IFLORA.TABFELDER (
    TID                NUMBER(13,3) CONSTRAINT PK_TABFELDER PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    TABELLE_ID         NUMBER(13,3) NOT NULL CONSTRAINT FK_TABELLE REFERENCES
    SH$IFLORA.SH_TABELLE(TID),
    NAME               VARCHAR2(30) NOT NULL,
    FELDTYP            VARCHAR2(30) NOT NULL,
    LAENGE             NUMBER(5));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_TABFELDER
REM *****
create sequence SH$IFLORA.SEQ_TABFELDER;

REM *****
REM ** TABLE: SH$IFLORA.ROLE
REM *****
create table SH$IFLORA.ROLE (
    TID                NUMBER(13,3) CONSTRAINT PK_ROLE PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    NAME               VARCHAR2(40) NOT NULL,
    KOMMENTAR          VARCHAR2(4000)
    STATUS             VARCHAR2(8));

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_ROLE
REM *****

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
create sequence SH$IFLORA.SEQ_ROLE;
```

```
REM *****
```

```
REM ** TABLE: SH$IFLORA.ROLE_TABELLE
```

```
REM *****
```

```
create table SH$IFLORA.ROLE_TABELLE (
```

```
    TID                NUMBER(13,3) CONSTRAINT PK_ROLE_TABELLE PRIMARY KEY,
```

```
    DS_DATE            DATE NOT NULL,
```

```
    DS_USER            NUMBER(13,3) NOT NULL,
```

```
    ROLE_ID            NUMBER(13,3) NOT NULL CONSTRAINT FK_ROLE_TAB_ROLE REFERENCES  
    SH$IFLORA.ROLE(TID),
```

```
    TABELLEN_ID        NUMBER(13,3) NOT NULL CONSTRAINT FK_ROLE_TAB_TAB REFERENCES  
    SH$IFLORA.TABELLE(TID));
```

```
REM *****
```

```
REM ** SEQUENCE: SH$IFLORA.SEQ_ROLE_TABELLE
```

```
REM *****
```

```
create sequence SH$IFLORA.SEQ_ROLE_TABELLE;
```

```
REM *****
```

```
REM ** TABLE: SH$IFLORA.ROLE_TAB_RIGHT
```

```
REM *****
```

```
create table SH$IFLORA.ROLE_TAB_RIGHT (
```

```
    TID                NUMBER(13,3) CONSTRAINT PK_ROLE_TAB_RIGHT PRIMARY KEY,
```

```
    DS_DATE            DATE NOT NULL,
```

```
    DS_USER            NUMBER(13,3) NOT NULL,
```

```
    ROLE_TAB_ID        NUMBER(13,3) NOT NULL CONSTRAINT FK_ROLE_RIGHT_TAB REFERENCES  
    SH$IFLORA.ROLE_TABELLE(TID),
```

```
    RIGHT              VARCHAR2(1) NOT NULL);
```

```
REM *****
```

```
REM ** Welche Tabellen Rechte können vergeben werden:
```

```
REM ** =====
```

```
REM **
```

```
REM **      S => SELECT
```

```
REM **      I => INSERT
```

```
REM **      U => UPDATE
```

```
REM **      D => DELETE
```

```
REM **      E => EXECUTE
```

```
REM **      evt. CHECK programmieren?
```

```
REM **
```

```
REM *****
```

```
REM *****
```

```
REM ** SEQUENCE: SH$IFLORA.SEQ_ROLE_TABELLE
```

```
REM *****
```

```
create sequence SH$IFLORA.SEQ_ROLE_TAB_RIGHT;
```

```
REM *****
```

```
REM ** TABLE: SH$IFLORA.USER_RECHTE => vordefinierte Rollen (PROJ_LEITER, ADMIN, MITARBEITER)
```

```
REM *****
```

```
create table SH$IFLORA.USER_RECHTE (
```

```
    TID                NUMBER(13,3) CONSTRAINT PK_USER_RECHTE PRIMARY KEY,
```

```
    DS_DATE            DATE NOT NULL,
```

```
    DS_USER            NUMBER(13,3) NOT NULL,
```

```
    NAME               VARCHAR2(40) NOT NULL,
```

```
    ROLE_TAB_ID        NUMBER(13,3) NOT NULL CONSTRAINT FK_RECHTE_USER REFERENCES  
    SH$IFLORA.ROLE_TABELLE(TID),
```

```
    KOMMENTAR          VARCHAR2(4000));
```

```
REM *****
```

```
REM ** Welche USER-Rechte können vergeben werden:
```

```
REM ** =====
```

```
REM **
```

```
REM **      A => ADMINISTRATOR
```

```
REM **      P => PROJEKTLLEITER
```

```
REM **      U => USER
```

```
REM **      => auf das jeweilige Projekt bezogen!
```

```
REM **
```

```
REM *****
```

```
REM *****
```

```

REM ** SEQUENCE: SH$IFLORA.SEQ_USER_RECHTE
REM *****
create sequence SH$IFLORA.SEQ_USER_RECHTE;

REM *****
REM ** TABLE: SH$IFLORA.USERNAME
REM *****
create table SH$IFLORA.USERNAME (
    TID                NUMBER(13,3) CONSTRAINT PK_USER PRIMARY KEY,
    DS_DATE            DATE NOT NULL,
    DS_USER            NUMBER(13,3) NOT NULL,
    NAME               VARCHAR2(40) NOT NULL,
    VORNAME            VARCHAR2(40),
    E_MAIL             VARCHAR2(40),
    USER_RECHT_ID      NUMBER(13,3),
    PROJEKT_ID         NUMBER(13,3),
    STATUS             VARCHAR2(3);

REM *****
REM ** SEQUENCE: SH$IFLORA.SEQ_USERNAME
REM *****
create sequence SH$IFLORA.SEQ_USERNAME;

REM *****ENDE*****

```

9.3 Sourcecode

9.3.1 iFlora Administration

9.3.1.1 Programmbibliotheken (Libraries)

9.3.1.1.1 DBManager.php

```

<?php

// This script is called by iFlora CONTROLLER to get Data
// for DHTML DB elements.
//
// history
// 2005-09-24 1.0 SH  begin DBManager and Method GETSelectData
// 2005-09-25 1.1 SH  begin Method GETInputData
//
// description
// GETSelectData => GET Data for a SELECT form
// GETInputData => GET Data for INPUT form

class DBManager
{
    private $_dbms;
    private $_dbhost;
    private $_sqluser;
    private $_sqlpwd;
    private $_sqldb;
    private $_sqlschema;

    /*******

function __construct($dbms,$host,$user,$pwd,$db,$schema)
{
    $this->_dbms = $dbms;
    $this->_dbhost = $host;
    $this->_sqluser = $user;
    $this->_sqlpwd = $pwd;

```

```

$this->_sqldb = $db;
$this->_sqlschema = $schema;
}

//*****

function GETSelectDatum($table,$field,$orderby,$where)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $query = "SELECT DISTINCT $field "
        . "FROM $table "
        . "$where "
        . "$orderby ";

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $datum = $result->fields[0];
    }
    return $datum;
}

//*****

function GETSelectSQLDatum($query)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $datum = $result->fields[0];
    }
    return $datum;
}

//*****

function GETSelectSQLSelect($query)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        while (!$result->EOF)
        {

```

```

        $arrDaten[] = $result->fields[0];
        $result->MoveNext();
    }
}
return $arrDaten;
}

//*****

function GETUpdateSQLDatum($query)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $result = $db->Execute($query);
}

//*****

function GETUpdateSQLDatumErr($query)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        $STATUS = "ERROR";
    } else {
        $STATUS = "OK";
    }
    return $STATUS;
}

//*****

function GETSelectSQL($query,$fields,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // $arrfield = preg_split("/,/",$fields);
    $arrfield = explode(',',$fields);

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($arrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash[$arrfield[$i]] = $result->fields[$i];
            }
            $arrDaten[] = $hash;
        }
    }
}

```

```

        $result->MoveNext();
    }
}

//*****

function GETSelectStammdata($frmid,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // fields
    $fields = "LABEL,POSITION,FORMAT,PARID,FORMLABEL,STYLE";
    $arrfield = explode(',',$fields);

    // Querys
    $SQLquery = "SELECT concat(p.PARNR, ' - ', p.PARNAME) AS LABEL, "
                ."r.FMRPOSITION AS POSITION, "
                ."p.PARFORMAT AS FORMAT, "
                ."p.TID AS PARID, "
                ."r.FMRLABEL AS FORMLABEL, "
                ."r.FMRSTYLE AS STYLE "
                ."FROM   afn_h_formrow r, afn_parameter p "
                ."WHERE  r.FRMID = $frmid "
                ."AND    p.TID = r.PARID "
                ."ORDER BY r.FMRPOSITION ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($SQLquery);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($arrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            {
                for ($i=0;$i<$count;$i++)
                {
                    $hash[$arrfield[$i]] = $result->fields[$i];
                }
                $arrDaten[] = $hash;
                $result->MoveNext();
            }
        }
    }
}

//*****

function GETObjektStammdata($obaid,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // fields
    $fields = "LABEL,POSITION,FORMAT,PARID,FORMLABEL,STYLE,FRMID";
    $arrfield = explode(',',$fields);

    // Querys
    $SQLquery = "SELECT concat(p.PARNR, ' - ', p.PARNAME) AS LABEL, "
                ."r.FMRPOSITION AS POSITION, "
                ."p.PARFORMAT AS FORMAT, "
                ."p.TID AS PARID, "
                ."r.FMRLABEL AS FORMLABEL, "
                ."r.FMRSTYLE AS STYLE, "
                ."f.TID AS FRMID "
                ."FROM   afn_h_formular f, afn_h_formrow r, afn_parameter p "

```

```

        ."WHERE f.OBAID = $obaid "
        ."AND    r.FRMID = f.TID "
        ."AND    p.TID = r.PARID "
        ."ORDER BY r.FMRPOSITION ";

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

// Datenbearbeitung
$result = $db->Execute($SQLquery);

if ($result==" " || $result==" " || $result === false)
{
    $status = "ERROR or EMPTY";
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hash[$arrfield[$i]] = $result->fields[$i];
        }
        $arrDaten[] = $hash;
        $result->MoveNext();
    }
}

}

//*****

function GETSelectAufnahmedata($objid,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // fields
    $fields = "ART,VEGWERT";
    $arrfield = explode(',',$fields);

    // Querys
    $SQLquery = "SELECT a.SPCNAME AS ART, v.VEGWERT AS VEGWERT "
        ."FROM    afn_vegdaten v, "
        ."        afn_arten a "
        ."WHERE a.TID = v.SPCID "
        ."AND    v.OBJID = $objid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    // Datenbearbeitung
    $result = $db->Execute($SQLquery);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($arrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash[$arrfield[$i]] = $result->fields[$i];
            }
            $arrDaten[] = $hash;
            $result->MoveNext();
        }
    }
}
}

```

```

/*****

function GETSelectParamdata($objid,$parid,&$arrParam)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $PARquery = "SELECT DISTINCT TID, STDWERT "
                ."FROM   afn_stammdaten "
                ."WHERE  objID = $objid "
                ."AND    parID = $parid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    // Datenbearbeitung
    $result = $db->Execute($PARquery);

    // fields
    $fieldes = "TID,STDWERT";
    $sarrfields = explode(',',$fieldes);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<2;$i++)
            {
                $hash[$sarrfields[$i]] = $result->fields[$i];
            }
            $sarrParam[] = $hash;
            $result->MoveNext();
        }
    }
}

/*****

function GETSelectObjektdata($objid,$parid,&$arrParam)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $ISINquery = "SELECT DISTINCT o.tid AS TID, o.objname AS NAME "
                ."FROM   afn_isin i, afn_parameter p, afn_objekt o "
                ."WHERE  p.tid = $parid "
                ."AND    i.isaid = p.isaid "
                ."AND    i.objid = $objid "
                ."AND    o.tid = i.isinobjid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    // Datenbearbeitung
    $result = $db->Execute($ISINquery);

    // fields
    $fieldes = "TID,NAME";
    $sarrfields = explode(',',$fieldes);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<2;$i++)

```



```

        {
            $hash[$sarrfields[$i]] = $result->fields[$i];
        }
        $sarrParam[] = $hash;
        $result->MoveNext();
    }

}

}

//*****

function GETSelectISINObjektdata($objid,$parid,&$sarrParam)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $ISINquery = "SELECT DISTINCT o.tid AS TID, o.objname AS NAME "
        ."FROM      afn_isin i, afn_parameter p, afn_objekt o "
        ."WHERE      p.tid = $parid "
        ."AND         i.isaid = p.isaid "
        ."AND         i.isinobjid = $objid "
        ."AND         o.tid = i.objid "
        ."ORDER BY o.objname ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($ISINquery);

    // fields
    $fieldes = "TID,NAME";
    $sarrfields = explode(',',$fieldes);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            {
                for ($i=0;$i<2;$i++)
                {
                    $hash[$sarrfields[$i]] = $result->fields[$i];
                }
                $sarrParam[] = $hash;
                $result->MoveNext();
            }
        }
    }

}

//*****

function GETisintartPOL($objid,$parid,&$sarrPOL)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // Abfrage
    $query = "SELECT DISTINCT ogem.objname AS Gemeinde, ogem.tid AS GemeindeID, "
        ."obez.objname AS Bezirk, obez.tid AS BezirkID, "
        ."obl.objname AS Bundesland, obl.tid AS BundeslandID "
        ."FROM afn_isin igem, "
        ."afn_isin ibez, "
        ."afn_isin ibl, "
        ."afn_parameter p, "
        ."afn_objekt ogem, "
        ."afn_objekt obez, "
        ."afn_objekt obl "
        ."WHERE p.tid = $parid "
        ."AND igem.isaid = p.isaid "

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
        ."AND igem.objjid = $objjid "
        ."AND ogem.tid = igem.isinobjjid "
        ."AND ibez.isaid = p.isaid "
        ."AND ibez.objjid = igem.isinobjjid "
        ."AND obez.tid = ibez.isinobjjid "
        ."AND ibl.isaid = p.isaid "
        ."AND ibl.objjid = ibez.isinobjjid "
        ."AND obl.tid = ibl.isinobjjid ";

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

// Datenbearbeitung
$result = $db->Execute($query);

// fields
$fields = "Gemeinde,GemeindeID,Bezirk,BezirkID,Bundesland,BundeslandID";
$arrfield = explode(',',$fields);

if ($result==" " || $result==" " || $result === false)
{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hashPOL[$arrfield[$i]] = $result->fields[$i];
        }
        $arrPOL[] = $hashPOL;
        $result->MoveNext();
    }
}

//*****

function GETisinOBJART($objjid,$parid,&$arrISINOBA)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // Abfrage
    $query = "SELECT DISTINCT ioa.isaid AS ISAID, "
        ."ioa.obaid AS ISOBAID, "
        ."ioa.isinobaid AS ISINOBAID, "
        ."o.obaid AS OBAID, "
        ."oa.obaname AS OBANAME "
        ."FROM "
        ."afn_isinobjart ioa, "
        ."afn_parameter p, "
        ."afn_objekt o, "
        ."afn_objektart oa "
        ."WHERE "
        ."p.tid = $parid "
        ."AND "
        ."oa.tid = o.obaid "
        ."AND "
        ."ioa.isaid = p.isaid "
        ."AND "
        ."o.tid = $objjid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($query);

    // fields
    $fields = "ISAID,ISOBAID,ISINOBAID,OBAID,OBANAME";
    $arrfield = explode(',',$fields);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    }
}
```

```

} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hashOBA[$arrfield[$i]] = $result->fields[$i];
        }
        $arrISINOBA[] = $hashOBA;
        $result->MoveNext();
    }
}

}

}

//*****

function GETSelectData($table,$fields,$orderby,$where,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // $arrfield = preg_split("/,/",$fields);
    $arrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ~)!
    if ($this->_dbms=="mysql")
    {
        $table = "~".strtolower($table). "~";
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $query = "SELECT DISTINCT $fields "
        . "FROM $table "
        . "$where "
        . "$orderby ";

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($arrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash[$arrfield[$i]] = $result->fields[$i];
            }
            $arrDaten[] = $hash;
            $result->MoveNext();
        }
    }
}

}

//*****

function GETSelDistArray($tables,$fields,$orderby,$where,&$arrDaten,&$arrLabel)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $arrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ~)!
    //if ($this->_dbms=="mysql")
    //{
    // $table = "~".strtolower($table). "~";

```

```
//}

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

$query = "SELECT DISTINCT $fields "
        . "FROM $tables "
        . "$where "
        . "$orderby ";

$result = $db->Execute($query);

if ($result==" " || $result==" " || $result === false)
{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0; $i<$count; $i++)
        {
            $label = $arrLabel[$arrfield[$i]];
            $arrDaten[$label] = $result->fields[$i];
        }
        $result->MoveNext();
    }
}

}

/*****

function GETInputData($table,$select,$fields,$orderby,$where,$params,$labels, $types,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $arrfield = explode(',',$fields);
    $paramfield = explode(',',$params);
    $labelfield = explode(',',$labels);
    $typfield = explode(',',$types);

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = ``.strtolower($table).``;
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $queryCOUNT = "SELECT count(TID) "
        . "FROM $table "
        . "$where ";

    $query = "SELECT DISTINCT $select "
        . "FROM $table "
        . "$where "
        . "$orderby ";

    $resultCOUNT = $db->Execute($queryCOUNT);

    if ($resultCOUNT==" " || $resultCOUNT==" " || $resultCOUNT === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $pruef = $resultCOUNT->fields[0];
        if ($pruef>0)
        {
            $result = $db->Execute($query);

            if ($result==" " || $result==" " || $result === false)

```

```

{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($sarrfield);
    $j = 0;
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hash["value"] = $result->fields[$i];
            $hash["label"] = $labelfield[$i];
            $hash["param"] = $paramfield[$i];
            $hash["typ"] = $typfield[$i];
            $harr[] = $hash;
            $hash = NULL;
        }

        $sarrDaten[] = $harr;
        $harr = NULL;

        $result->MoveNext();
    }
} else {
    $count = count($sarrfield);
    for ($i=0;$i<$count;$i++)
    {
        $hash["value"] = "";
        $hash["label"] = $labelfield[$i];
        $hash["param"] = $paramfield[$i];
        $hash["typ"] = $typfield[$i];
        $harr[] = $hash;
        $hash = NULL;
    }

    $sarrDaten[] = $harr;
    $harr = NULL;
}
}

}

/*****

function GETOutputData($table,$fields,$orderby,$where,&$sarrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $sarrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = ``.strtolower($table).``;
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $queryCOUNT = "SELECT count(TID) "
                    . "FROM $table "
                    . "$where ";

    $query = "SELECT DISTINCT $fields "
            . "FROM $table "
            . "$where "
            . "$orderby ";

    $resultCOUNT = $db->Execute($queryCOUNT);

    if ($resultCOUNT==" " || $resultCOUNT==" " || $resultCOUNT === false)

```

```

{
    print_r($TEXT['ADODB-nottable']);
} else {
    $pruef = $resultCOUNT->fields[0];
    if ($pruef>0)
    {
        $result = $db->Execute($query);

        if ($result==" " || $result==" " || $result === false)
        {
            print_r($TEXT['ADODB-nottable']);
        } else {
            $count = count($arrfield);
            // Auswahlliste zusammenstellen
            while (!$result->EOF)
            {
                for ($i=0;$i<$count;$i++)
                {
                    $harr[] = $result->fields[$i];
                }
                $arrDaten[] = $harr;
                $harr = NULL;
                $result->MoveNext();
            }
        }
    }
}
}
}

//*****

function INSERTStandard($table,$sequence,&$arrINSERT)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    if ($this->_dbms=="oci8")
    {
        $sequence = "$sequence.NEXTVAL";
        $saktdate = "SYSDATE";
    } elseif ($this->_dbms=="postgres") {
        $sequence = "NEXTVAL('$sequence')";
        $saktdate = "CURRENT_TIMESTAMP";
    } elseif ($this->_dbms=="mysql") {
        $sequence = "";
        $saktdate = "CURRENT_TIMESTAMP";
        $table = "`".strtolower($table)."`";
    }

    $user = 0;

    $query = "INSERT INTO ".$this->_sqlschema.$table." VALUES ($sequence, "
                                     ."$saktdate, "
                                     ."$user, ";

    // INSERTquery zusammenstellen
    foreach ($arrINSERT as $Daten)
    {
        if ($Daten["typ"]=="T" || $Daten["typ"]=="D")
        {
            $query = $query."".$Daten["value"].", ";
        } else {
            $query = $query.$Daten["value"].", ";
        }
    }

    $len = strlen($query);
    $query = substr($query,0,$len-2);
    $query = $query.");";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

```

```

$db->Execute($query);

}

//*****

function UPDATEStandard($table,$tid,&$arrUPDATE)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // UPDATEquery zusammenstellen
    $query = "UPDATE ".$this->_sqlschema.$table." SET ";

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = "`".strtolower($table)."`";
    }

    foreach ($arrUPDATE as $Daten)
    {
        if ($Daten["typ"]=="T" || $Daten["typ"]=="D")
        {
            $field = $Daten["field"]." = ".$Daten["value"].", ";
        } else {
            $field = $Daten["field"]." = ".$Daten["value"].", ";
        }
        $query = $query.$field;
    }

    $len = strlen($query);
    $query = substr($query,0,$len-2);
    $query = $query." WHERE TID = $tid";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $db->Execute($query);
}

//*****

function DELETEStandard($table,$tid)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // UPDATEquery zusammenstellen
    $query = "DELETE FROM ".$this->_sqlschema.$table." WHERE TID = ".$tid;

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = "`".strtolower($table)."`";
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $db->Execute($query);
}

//*****

public function GETIfnull($text,$ifnull)
{
    if ($this->_dbms=="oci8")
    {
        $text = "NVL($text,$ifnull)";
    }
}

```

```

} elseif ($this->_dbms=="postgres") {
    $text = "ifnull($text,$ifnull)";
} elseif ($this->_dbms=="mysql") {
    $text = "ifnull($text,$ifnull)";
}
return $text;
}

//*****

function GET_Userstatus($sql_user)
{
    // INCLUDE
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // SQL-Abfrage zusammenstellen
    if ($this->_dbms=="oci8")
    {
        $query = "SELECT STATUS, "
            . "      (SYSDATE - DS_DATE), "
            . "      NVL(DAUER,0) "
            . "FROM USERNAME "
            . "WHERE NAME = '$sql_user'";
        $queryPRUEF = "SELECT count(TID) "
            . "FROM USERNAME ";
    } elseif ($this->_dbms=="postgres") {
        $query = "SELECT STATUS, "
            . "      (CURRENT_DATE - DS_DATE), "
            . "      IFNULL(DAUER,0) "
            . "FROM USERNAME "
            . "WHERE NAME = '$sql_user'";
        $queryPRUEF = "SELECT count(TID) "
            . "FROM USERNAME ";
    } elseif ($this->_dbms=="mysql") {
        $query = "SELECT STATUS, "
            . "      (UNIX_TIMESTAMP() - UNIX_TIMESTAMP(DS_DATE))/86400, "
            . "      ifnull(DAUER,0) "
            . "FROM `username` "
            . "WHERE NAME = '$sql_user'";
        $queryPRUEF = "SELECT count(TID) "
            . "FROM `username` ";
    }
}

try
{
    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // SQL Abfrage ausführen
    $result = $db->Execute($query);

    // Auswertung
    if ($result==" " || $result==" " || $result === false)
    {
        $user_aktiv = "N-DB";
    } else {
        $fkt_typ = $result->fields[0];
        $fkt_diff = $result->fields[1];
        $fkt_dauer = $result->fields[2];

        if ($fkt_typ!="")
        {
            if ($fkt_typ == "ADMIN" || $fkt_typ == "USER")
            {
                /*****
                * USER Typen:                                *
                * =====                                *
                * ADMIN => Administrator                    *
                * USER  => Permanent (normaler User)        *
                *****/
            }
        }
    }
}

```



```

* TEST => Temporär (Testuser)      *
*****/

    $user_aktiv = 'Y';
} else {
    if ($fkt_diff > $fkt_dauer)
    {
        $user_aktiv = 'N-TIMEOUT';
    } else {
        $user_aktiv = 'Y';
    }
}
} else {
    $user_aktiv = 'N-NOTVALID';
}
}
} catch (Exception $exception) {
    $user_aktiv = "N_DB";
}

return $user_aktiv.";".$fkt_typ;
}

*****/

function GETCreateUser($sql_user,$pwd)
{
    // INCLUDE
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // DBMS abhängiges USER anlegen!
    if ($this->_dbms=="oci8")
    {
        $queryCREATE = "CREATE USER ".$sql_user." IDENTIFIED BY abc";
        $queryGRANT = "GRANT connect, resource TO ".$sql_user." ";
        $db->Execute($queryCREATE);
        $db->Execute($queryGRANT);
    } elseif ($this->_dbms=="postgres") {
        $queryCREATE = "CREATE USER $sql_user IDENTIFIED BY $pwd";
        $db->Execute($queryCREATE);
    } elseif ($this->_dbms=="mysql") {
        $queryCREATE = "GRANT SELECT ON ".$this->_sqldb.".projekt TO '".$sql_user."' IDENTIFIED BY '".$pwd.'";";
        $db->Execute($queryCREATE);
    }
}

*****/

function GETCreateStandardTable($tablename)
{
    // INCLUDE
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // DBMS abhängiges USER anlegen!
    if ($this->_dbms=="oci8")
    {
        $queryCREATE = "CREATE TABLE ".$this->_sqlschema.$tablename." ( "
            ."TID      NUMBER(13,3) CONSTRAINT PK_".$tablename." PRIMARY KEY, "
            ."DS_DATE   DATE NOT NULL, "
            ."DS_USER   NUMBER(13,3) NOT NULL)";
        $querySEQUENCE = "CREATE SEQUENCE ".$this->_sqlschema."SEQ_".$tablename;
    }
}

```

```

$querySYNONYM = "CREATE PUBLIC SYNONYM ".$tablename." FOR ".$this->$_sqlschema.$tablename;
$db->Execute($queryCREATE);
$db->Execute($querySEQUENCE);
$db->Execute($querySYNONYM);
} elseif ($this->_dbms=="postgres") {
    $queryCREATE = "CREATE TABLE ".$this->$_sqlschema.$tablename." ( "
        ."TID      NUMBER(13,3) CONSTRAINT PK_". $tablename." PRIMARY KEY, "
        ."DS_DATE   DATE NOT NULL, "
        ."DS_USER   NUMBER(13,3) NOT NULL)";
    $querySEQUENCE = "CREATE SEQUENCE ".$this->$_sqlschema."SEQ_". $tablename;
    $db->Execute($queryCREATE);
    $db->Execute($querySEQUENCE);
} elseif ($this->_dbms=="mysql") {
    $queryCREATE = "CREATE TABLE ".$this->$_sqlschema.$tablename." ( "
        ."TID      INT(13) NOT NULL AUTO_INCREMENT, "
        ."DS_DATE   DATETIME NOT NULL, "
        ."DS_USER   INT(13) NOT NULL, "
        ."PRIMARY KEY(TID))";
    $db->Execute($queryCREATE);
}
}
}
?>

```

9.3.1.1.2 ViewManager.php

<?php

```

// This script is called by iFlora CONTROLLER to print
// DHTML DB elements.
//
// history
// 2005-09-24 1.0 SH  begin GETSelect
// 2005-09-25 1.1 SH  split GETSelect in GETSelect and
//                  GETSelectForm
//                  begin GETInputForm
// 2005-09-27 1.2 SH  private Arrays
// 2005-09-28 1.3 SH  private Arrays to slow!!!
//                  => references
//
// description
// GETSelect => produce a SELECT
// GETSelectForm => produce a SELECT with a form
// GETOutput => produce a list of data

```

```

class VIEWManager
{
    /**
     *
     */
    // Übergabe von Arrays zu langsam!!! => deshalb REFERENZ!!!

    public function GETSelect($tid,$parameter,&$arrSQL)
    {
        // Auswahlliste BEGIN
        echo ' <p>
            <select name="'. $parameter.'" size="1">
            <option value="0">Bitte auswählen</option>';

        // Auswahlliste zusammenstellen
        foreach ($arrSQL as $Daten)
        {
            foreach ($Daten as $key => $value)
            {
                if ($key=="TID")
                {
                    $hTID = $value;
                } elseif ($key=="NAME") {
                    $hNAME = $value;
                }
            }
            if ($hTID==$tid)
            {

```

```

        print 'option selected value="'. $hTID. '">'. $hNAME. '</option>';
    } else {
        print 'option value="'. $hTID. '">'. $hNAME. '</option>';
    }
}
// Auswahlliste END
echo ' </select>';

// ENDE Auswahlliste
}

//*****

public function GETSelectForm($tid,$param,$file,$lbSelect,&$arrSQL)
{
    // Formular BEGIN
    echo '<form method="GET" action="'. $file. '">
        <p>';

    self::GETSelect($tid,$param,$arrSQL);

    // Formular END
    echo ' </p>
        <p>
            <INPUT TYPE="SUBMIT" VALUE="'. $lbSelect. '">
        </p>
    </form>';
}

//*****

public function GETSelFormHidden($tid,$param,$file,$lbSelect,&$arrSQL,&$arrHIDDEN)
{
    // Formular BEGIN
    echo '<form method="GET" action="'. $file. '">
        <p>';

    self::GETSelect($tid,$param,$arrSQL);

    // Formular END
    echo ' </p>
        <p>';

    if (is_array($arrHIDDEN))
    {
        foreach ($arrHIDDEN as $key => $value)
        {
            echo '<input type="hidden" name="'. $key. '" value="'. $value. '">';
        }
    }

    echo ' <INPUT TYPE="SUBMIT" VALUE="'. $lbSelect. '">
        </p>
    </form>';
}

//*****

public function GETInputForm($file,$act,$button,$buttonDELETE,$buttonNEW,$tid,$param,$formlink,$buttonLINK,
$namlink,&$arrInputSQL)
{
    // Form BEGIN
    echo '<form method="GET" action="'. $file. '">
        <table>';

    self::GETInput($arrInputSQL);

    echo ' <tr>
        <td>';

    if ($act=="UPDATE")
    {
        print '<input type="hidden" name="'. $param. '" value="'. $tid. '">';
    }
}

```

```
echo '  
    <input type="hidden" name="upd" value="'. $act.' ">  
        <input type="SUBMIT" value="'. $button.'">  
    </td>  
<td>  
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</td>  
</tr>  
</table>  
</form>;'  
  
if ($act=="UPDATE")  
{  
    echo '<table>  
        <tr>  
            <td>  
                <form form method="GET" action="'. $file.'">  
                    <input type="SUBMIT" value="'. $buttonNEW.'">  
                </form>  
            </td>  
            <td>  
                <form name="FormDELETE" method="GET" action="'. $file.'">  
                    <input type="hidden" name="'. $param.'" value="'. $tid.'">  
                    <input type="hidden" name="upd" value="DELETE">  
                    <input type="SUBMIT" value="'. $buttonDELETE.'">  
                </form>  
            </td>;'  
    if ($formlink != "")  
    {  
        echo ' <td>  
            <form name="FormLINK" method="GET" action="'. $formlink.'">  
                <input type="hidden" name="parenttid" value="'. $tid.'">  
                <input type="hidden" name="parentnam" value="'. $namlink.'">  
                <input type="SUBMIT" value="'. $buttonLINK.'">  
            </form>  
        </td>;'  
    }  
    echo ' </tr>  
    </table>;'  
}  
'  
'  
'  
  
//*****  
  
public function GETOutputList($updel,$lupdate,$ldelete,&$header,&$arrOutputSQL)  
{  
    // Liste BEGIN  
    echo '<table>;'  
    $rownr = 1;  
  
    // Ueberschrift (Header)  
    echo '<tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</td>;'  
    foreach ($header as $lbl)  
    {  
        echo '<td bgcolor="#C0C0C0"><b>' . $lbl . '</b></td>  
             <td bgcolor="#C0C0C0">&nbsp;&nbsp;&nbsp;&nbsp;&~</td>;'  
    }  
    if ($updel == "U" || $updel == "UD")  
    {  
        echo ' <td bgcolor="#C0C0C0">&nbsp;&nbsp;&~</td>  
              <td bgcolor="#C0C0C0">&nbsp;&nbsp;&~</td>;'  
    }  
    if ($updel == "D" || $updel == "UD")  
    {  
        echo ' <td bgcolor="#C0C0C0">&nbsp;&nbsp;&~</td>  
              <td bgcolor="#C0C0C0">&nbsp;&nbsp;&~</td>;'  
    }  
    echo '</tr>;'  
  
    // Datenliste  
    foreach ($arrOutputSQL as $Daten)  
    {  
        $rowmod = $rownr%2;  
        if ($rowmod == 0)  
        {  
            $farbe = '#EEEEEE';
```

```

    } else {
        $farbe = '#FFFFFF';
    }
    echo '<tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    $tid = $Daten[0];
    $maxcount = count($Daten);
    for ($i=1;$i<$maxcount;$i++)
    {
        echo '<td bgcolor="'. $farbe.'">'. $Daten[$i]. '</td><td bgcolor="'. $farbe.'">&nbsp;&nbsp;&nbsp;</td>';
        //print "$i / $farbe / $Daten[$i] <br />";
    }
    if ($supdel == "U" || $supdel == "UD")
    {
        echo ' <td bgcolor="'. $farbe.'">
            <form name="update" method="GET" action="tabfieldadmin.php">
                <input type="hidden" name="tid" value="'. $tid.'">
                <input type="hidden" name="upd" value="SELFORUPD">
                <input type="SUBMIT" value="'. $lblupdate.'">
            </form>
        </td>
        <td bgcolor="'. $farbe.'">
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
        </td>';
    }
    if ($supdel == "UD" || $supdel == "D")
    {
        echo ' <td bgcolor="'. $farbe.'">
            <form name="delete" method="GET" action="tabfieldadmin.php">
                <input type="hidden" name="tid" value="'. $tid.'">
                <input type="hidden" name="upd" value="DELETE">
                <input type="SUBMIT" value="'. $lbldelete.'">
            </form>
        </td>
        <td bgcolor="'. $farbe.'">
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
        </td>';
    }
}

echo '</tr>';
$rownr++;
}
echo '</table>';
}

/**
 *
 */

public function GETLabel($label,$format)
{
    print "<$format>$label</$format>";
}

/**
 *
 */

public function GETRow()
{
    print "<p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</p>";
}

/**
 *
 */

public function GETDataRow($data)
{
    print "&nbsp;$data&nbsp;&nbsp;&~<br />";
}

/**
 *
 */

public function GETDataLink($file,$data)
{
    echo '<a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$file.'">'. $data.'</a><br />';
}

/**
 *
 */

public function GETDataRowBold($data)

```

```
{
    print "&nbsp;<br /><b>$data&nbsp;</b><br />";
}

//*****

public function GETTextfield($text)
{
    print "<p>$text</p>";
}

//*****

public function GET1DTable(&$arrHISTDaten)
{
    if (is_array($arrHISTDaten))
    {
        echo '<table>';

        foreach ($arrHISTDaten as $key => $value)
        {
            echo '<tr>'
                <td>
                    <b>'. $key.'</b>
                </td>
                <td>
                    '. $value.'
                </td>
            </tr>;'
        }

        echo '</table>';
    }
}

//*****

public function GET2DTable(&$arrLISTDaten)
{
    if (is_array($arrLISTDaten))
    {
        echo '<table id="gridtable">';
        $colnum = 1;

        foreach ($arrLISTDaten as $key => $value)
        {
            $col = $colnum%2;
            if ($col == 0)
            {
                echo '<tr>';
            }
            echo ' <td>
                <b>'. $key.'</b>
            </td>
            <td>
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
            </td>
            <td>
                '. $value.'
            </td>
            <td>
                ~~~~~~
            </td>';
            if ($col == 1)
            {
                echo '</tr>';
            }
            $colnum = $colnum + 1;
        }
        echo '</table>';
    }
}

//*****
```

```
public function GETLink_old($file,$lbSelect,&$arrHIDDEN)
{
    // Formular BEGIN
    echo '<form method="GET" action="'. $file.'">'
        <p>;

if (is_array($arrHIDDEN))
{
    foreach ($arrHIDDEN as $key => $value)
    {
        echo '<input type="hidden" name="'.$key.'" value="'. $value.'">';
    }
}

echo '   <INPUT TYPE="SUBMIT" VALUE="'. $lbSelect.'">'
      </p>
</form>';

}


//*****

public function GETLink($file,$lbSelect)
{
    // Formular BEGIN
    echo '<p>'
         <a href="http://$_SERVER['SERVER_NAME']/iFloraPublic/public/'. $file.'">. $lbSelect.</a>
       </p>;

}


//*****Formular*****

public function GETUpload($label,$button)
{
    echo '<table>'
        <tr><td colspan="2">&nbsp;&nbsp;&nbsp;</td></tr>
        <tr><td colspan="2">&nbsp;&nbsp;&nbsp;</td></tr>
    </table>
    <table border="1" cellspacing="0" cellpadding="0" align="center" >
        <tr>
            <td colspan="2">
                <table border="0" cellspacing="0" cellpadding="0" align="center">
                    <tr id="frmLevel1"><td colspan="3">'. $label.' </td></tr>
                    <tr id="frmLevel2"><td colspan="3">&nbsp;&nbsp;&nbsp;</td></tr>
                        <form enctype="multipart/form-data" action=". $PHP_SELF." method=post>
                            <input type="hidden" name="MAX_FILE_SIZE" value="500000">
                            <tr id="frmLevel3"><td colspan="3">&nbsp;&nbsp;&nbsp;</td></tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&nbsp;</td></tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&nbsp;</td></tr>
                            <tr id="frmLevel4">
                                <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Datei:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
                                <td colspan="2"><input name="ori_file" type="file">&nbsp;&nbsp;&nbsp;</td>
                            </tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&nbsp;</td></tr>
                            <tr id="frmLevel4">
                                <td colspan="3" align="center"><input type="submit" name="send_ori" value="'. $button.'"></td>
                            </tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&nbsp;</td></tr>
                            <tr id="frmLevelError4"><td colspan="3" align="center">&nbsp;&nbsp;&Der Uploadprozess kann einige Minuten dauern!&nbsp;&nbsp;&</td>
                            </tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&~</td></tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&~</td></tr>
                            <tr id="frmLevel4"><td colspan="3">&nbsp;&nbsp;&~</td></tr>
                            <tr id="frmLevel3"><td colspan="3">&nbsp;&nbsp;&~</td></tr>
                        </from>
                        <form>
                            <tr id="frmLevel2"><td colspan="3" align="center">
                                <input type="submit" name="work" value="Daten bearbeiten"></td>
                                <input type="hidden" name="show" value="true">
                            </tr>
                            <tr id="frmLevel1"><td colspan="3">&nbsp;&nbsp;&~</td></tr>
                        </from>
                    </table>
                </td>
```

```

        </td>
    </tr>
</table>';
}

//*****Formular*****

function PRUEFFormat($format,$value)
{
    $result = 'false';

    if (isset($format) && strlen($format) > 0 && isset($value))
    {
        $pos1 = strpos($format,"(");
        $pos2 = strpos($format,")");
        $formatart = substr($format,0,$pos1);
        $formatlen = (int) substr($format,$pos1+1,$pos2-$pos1-1);
        $valuelen = (int) strlen($value);

        if ($formatart == 'X' || $formatart == 'F+' || $formatart == 'F0+' || $formatart == 'F0')
        {
            if ($valuelen <= $formatlen)
            {
                $result = 'true';
            }
            elseif ($formatart == 'F') {
                if ($valuelen <= $formatlen+1)
                {
                    $result = 'true';
                }
            }
            elseif ($formatart == 'F-') {
                if ($valuelen <= $formatlen+2)
                {
                    $result = 'true';
                }
            }
        }
    }
    return $result;
}

//*****Formular*****

public function GETFormularHeader($label)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
        echo '<tr id="frmLevel1"><td colspan="3">'. $label. '&nbsp;  ';</td></tr>';
    }

//*****Formular*****

public function GETFormular2Filter($filter1,$lfilter1,&$arrfilter1,$filter2,$lfilter2,&$arrfilter2)
{
    echo '<table><table width="100%" border="0" cellspacing="0" cellpadding="0">';
        echo '<tr id="frmLevelFilter" height="17">';
            echo '<td>&nbsp;  ';</td>';
            echo '<td><b>Schnellsuche:</b>&nbsp;  ';</td>';
            echo '<form>';
        echo '<td>'. $filter1. ':&nbsp;  ';</td>';
            echo '<td colspan="1"><select name="'. $lfilter1. '" size="1"><option selected value="0">Bitte auswählen</option>';
        if (is_array($arrfilter1))
        {
            foreach ($arrfilter1 as $fldFilter1)
            {
                echo '<option value="'. $fldFilter1["TID"]. '">'. $fldFilter1["NAME"]. '</option>';
            }
        }
        echo '<td><input type="submit" value="Suchen"></td>';
        echo '</form></td>';
        echo '<td>&nbsp;  &nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&';</td>';
        echo '<td>'. $filter2. ':&nbsp;  ';</td>';
            echo '<td colspan="1"><select name="'. $lfilter2. '" size="1"><option selected value="0">Bitte auswählen</option>';
        if (is_array($arrfilter2))
        {
            foreach ($arrfilter2 as $fldFilter2)

```



```

    {
        echo '<option value="'.fldFilter2['TID'].'">'.fldFilter2['NAME'].</option>';
    }
}

echo '<td><input type="submit" value="Suchen"></td>';
echo '</form>';
echo '</tr><tr><td>&nbsp;</td></tr></table>';
}

//*****Formular*****

public function GETFormular3Filter($filter1,$lkfilter1,&$arrfilter1,$filter2,$lkfilter2,&$arrfilter2,$filter3,$lkfilter3,&$arrfilter3)
{
    echo '<table><table width="100%" border="0" cellspacing="0" cellpadding="0">';
        echo '<tr id="frmLevelFilter" height="17">';
            echo '<td>&nbsp;</td>';
            echo '<td><b>Schnellsuche:</b>&nbsp;</td>';
            echo '<form>';
echo '<td>'. $filter1 .'&nbsp;</td>';
            echo '<td colspan="1"><select name="'. $lkfilter1 .'" size="1"><option selected value="0">Bitte auswählen</option>';
if (is_array($arrfilter1))
{
    foreach ($arrfilter1 as $fldFilter1)
    {
        echo '<option value="'. $fldFilter1['TID'].'">'. $fldFilter1['NAME'].</option>';
    }
}

        echo '<td><input type="submit" value="Suchen"></td>';
        echo '<td>&nbsp;</td>';
        echo '<td>&nbsp;</td>';
        echo '<td>&nbsp;</td>';
        echo '<td>&nbsp;</td>';
        echo '</form></tr>';
        echo '<tr id="frmLevelFilter" height="17"><td>&nbsp;</td><form>';
        echo '<td>&nbsp;</td>';
echo '<td>'. $filter2 .'&nbsp;</td>';
            echo '<td colspan="1"><select name="'. $lkfilter2 .'" size="1"><option selected value="0">Bitte auswählen</option>';
if (is_array($arrfilter2))
{
    foreach ($arrfilter2 as $fldFilter2)
    {
        echo '<option value="'. $fldFilter2['TID'].'">'. $fldFilter2['NAME'].</option>';
    }
}

        echo '<td><input type="submit" value="Suchen"></td>';
        echo '</form><form>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;</td>';
        echo '<td>'. $filter3 .'&nbsp;</td>';
        echo '<td colspan="1"><select name="'. $lkfilter3 .'" size="1"><option selected value="0">Bitte auswählen</option>';
if (is_array($arrfilter3))
{
    foreach ($arrfilter3 as $fldFilter3)
    {
        echo '<option value="'. $fldFilter3['TID'].'">'. $fldFilter3['NAME'].</option>';
    }
}
echo '<td><input type="submit" value="Suchen"></td>';
echo '</form>';
echo '</tr><tr><td>&nbsp;</td></tr></table>';
}

//*****Formular*****

public function GETFormular1Filter($filter,$lkfilter,&$arrfilter,&$arrhidden)
{
    echo '<table><table width="100%" border="0" cellspacing="0" cellpadding="0">';
        echo '<tr id="frmLevelFilter" height="17">';
            echo '<td>&nbsp;</td>';
            echo '<td><b>Schnellsuche:</b>&nbsp;</td>';
            echo '<form>';
            if(is_array($arrhidden))
            {
                foreach ($arrhidden as $hidden)
                {
                    echo '<input type="hidden" name="'. $hidden['NAME'].'" value="'. $hidden['TID'].'">';
                }
            }
        echo '<td><input type="submit" value="Suchen"></td>';
        echo '</form>';
}

```

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

Stephan Hordynski (Matrikelnr.: a9227169)

```

        {
            echo '<input type="hidden" name="'. $hidden['NAME']. '" value="'. $hidden['TID']. '">';
        }
    }
    echo ' </form>
        </tr>
        <form action="'. $file. '">
            <input type="hidden" name="frmstatus" value="'. $frmstatus. '">';
    if(is_array($arrhidden))
    {
        foreach ($arrhidden as $hidden)
        {
            echo '<input type="hidden" name="'. $hidden['NAME']. '" value="'. $hidden['TID']. '">';
        }
    }
}

//*****Formular*****

public function GETFormularHeaderUFormUPD2($label,$file,$frmstatus,&$arrhidden,$arrhidden2)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
    echo '<tr id="frmLevel1">
        <form action="'. $file. '">
            <td colspan="2">'. $label. '&nbsp;</td>
            <td>&nbsp;<input type="submit" value="zur&uuml;ck">&nbsp;</td>';
    if(is_array($arrhidden2))
    {
        foreach ($arrhidden2 as $hidden)
        {
            echo '<input type="hidden" name="'. $hidden['NAME']. '" value="'. $hidden['TID']. '">';
        }
    }
    echo ' </form>
        </tr>
        <form action="'. $file. '">
            <input type="hidden" name="frmstatus" value="'. $frmstatus. '">';
    if(is_array($arrhidden))
    {
        foreach ($arrhidden as $hidden)
        {
            echo '<input type="hidden" name="'. $hidden['NAME']. '" value="'. $hidden['TID']. '">';
        }
    }
}

//*****Formular*****

public function GETFormularHeaderUserverwaltungUPD($label,$file,$sidfeld,$sid,$frmstatus)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
    echo '<tr id="frmLevel1">
        <form action="'. $file. '">
            <td colspan="2">'. $label. '&nbsp;</td>
            <td>&nbsp;<input type="submit" value="zur&uuml;ck">&nbsp;</td>
        </form>
    </tr>
        <form action="'. $file. '">
            <input type="hidden" name="frmstatus" value="'. $frmstatus. '">
            <input type="hidden" name="'. $sidfeld. '" value="'. $sid. '">';
    }

//*****Formular*****

public function GETFormularFooterUserverwaltungUPD()
{
    echo '<tr id="frmLevel2">
        <td colspan="2">&nbsp;</td>
        <td align="center">&nbsp;<input type="submit" value="speichern">&nbsp;</td>
    </tr>';
    echo '</form>';
    echo '</table></tr></table>';
}

```

```
//*****Formular*****  
public function GETFormularFooterUserverwaltungORI($arrhidden)  
{  
    echo '<tr id="frmLevel2">  
        <td colspan="3" align="center">&nbsp;<input type="submit" value="speichern">&nbsp;</td>  
    </tr>;'  
    echo '</form>;'  
    echo '</form>;'  
    if(is_array($arrhidden))  
    {  
        foreach ($arrhidden as $shidden)  
        {  
            echo '<input type="hidden" name="'. $shidden['NAME'].'." value="'. $shidden['TID'].'.'.">;'  
        }  
    }  
    echo '<tr id="frmLevel3"><td colspan="3">&nbsp;&nbsp;</td></tr>;'  
    echo '<tr id="frmLevel2">  
        <td colspan="3" align="center">&nbsp;<input type="submit" value="!ouml;schen">&nbsp;</td>  
    </tr>;'  
    echo '</form>;'  
    echo '</table></tr></table>;'  
}  
  
//*****Formular*****  
public function GETFormularFooterUploadORI(&$arrhidden,&$arrprojekt,$projekt,&$armmethode,$methode)  
{  
    echo '<tr id="frmLevel1"><td colspan="3">&nbsp;</td></tr>;'  
    echo '<tr id="frmLevel4"><td colspan="3">&nbsp;</td></tr>;'  
    echo '<tr id="frmLevel4"><td colspan="3">&nbsp;</td></tr>;'  
    echo '</form>;'  
  
    if(is_array($arrhidden))  
    {  
        foreach ($arrhidden as $shidden)  
        {  
            echo '<input type="hidden" name="'. $shidden['NAME'].'." value="'. $shidden['TID'].'.'.>;'  
        }  
    }  
    if(is_array($arrprojekt))  
    {  
        echo '<tr id="frmLevel4">  
            <td colspan="3" align="center">&nbsp;<b>Projekt: </b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<br>  
            <select name="projekt" size="1">  
                <option>Bitte whlen Sie aus</option>;'  
            $count = count($arrprojekt);  
            for ($i=0; $i<$count; $i++)  
            {  
                if ($arrprojekt[$i] == $projekt)  
                {  
                    echo '<option selected value="'. $arrprojekt[$i].'.'">. $arrprojekt[$i].</option>;'  
                } else {  
                    echo '<option value="'. $arrprojekt[$i].'.'">. $arrprojekt[$i].</option>;'  
                }  
            }  
        echo '</select></td></tr>;'  
        echo '<tr id="frmLevel4"><td colspan="3">&nbsp;</td></tr>;'  
    }  
    if(is_array($arrprojekt))  
    {  
        echo '<tr id="frmLevel4">  
            <td colspan="3" align="center">&nbsp;<b>Methode: </b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<br>  
            <select name="methode" size="1">  
                <option>Bitte whlen Sie aus</option>;'  
            $count = count($armmethode);  
            for ($i=0; $i<$count; $i++)  
            {  
                if ($armmethode[$i] == $methode)  
                {  
                    echo '<option selected value="'. $armmethode[$i].'.'">. $armmethode[$i].</option>;'  
                } else {  
                    echo '<option value="'. $armmethode[$i].'.'">. $armmethode[$i].</option>;'  
                }  
            }  
        }
```

```

    }
    echo '</select></td><tr>';
    echo '<tr id="frmLevel4"><td colspan="3">&nbsp;</td></tr>';
}
echo '<tr id="frmLevelError4"><td colspan="3" align="center">
    &nbsp;<td colspan="3">&nbsp;</td></tr>
    <tr id="frmLevel4"><td colspan="3">&nbsp;</td></tr>
    <tr id="frmLevel2">
        <td colspan="3" align="center">&nbsp;<input type="submit" value="speichern">&nbsp;</td>
    </tr>';
echo '</form>';
echo '<tr id="frmLevel1"><td colspan="3">&nbsp;</td></tr>';
echo '</table></tr></table>';
}

//*****Formular*****

public function GETFormularHeaderStammdaten($label)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
    echo '<tr id="frmLevel1"><td colspan="3">'. $label. '&nbsp;</td></tr>';
    echo '<tr id="frmLevel2"><td colspan="3">Stammdaten&nbsp;</td></tr>';
}

//*****Formular*****

public function GETFormularHeaderAufnahme($label)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
    echo '<tr id="frmLevel1"><td colspan="3">'. $label. '&nbsp;</td></tr>';
    echo '<tr id="frmLevel2"><td colspan="3">Aufnahmefläche&nbsp;</td></tr>';
}

//*****Formular*****

public function GETFormRowStyle1($label)
{
    echo '<tr id="frmLevel1"><td colspan="3">'. $label. '&nbsp;</td></tr>';
}

//*****Formular*****

public function GETFormRowStyleUPDATE1($label,$file,$id,$field,$arrhidden)
{
    echo '<tr id="frmLevel1">
        <form action="'. $file. '">
            <input type="hidden" name="frmstatus" value="EDIT">
            <input type="hidden" name="'. $field. '" value="'. $id. '">
            <td colspan="2">'. $label. '&nbsp;</td>
            <td>&nbsp;<input type="submit" value="&auml;ndern">&nbsp;</td>';
    if(is_array($arrhidden))
    {
        foreach ($arrhidden as $hidden)
        {
            echo '<input type="hidden" name="'. $hidden['NAME']. '" value="'. $hidden['TID']. '">';
        }
    }
    echo ' </td>
        </form>
    </tr>';
}

//*****Formular*****

public function GETFormRowStyleGESPERRT1($label)
{
    echo '<tr id="frmLevel1">
        <td colspan="2">'. $label. '&nbsp;</td>
        <td>&nbsp;gesperrt&nbsp;</td>
    </tr>';
}

//*****Formular*****

```

```

public function GETFormRowStyle2($label)
{
    echo '<tr id="frmLevel2"><td colspan="3">'. $label.'&nbsp;&nbsp;&nbsp;</td></tr>';
}

//*****Formular*****

public function GETFormRowStyleUPDATE2($label,$file,$id,$field)
{
    echo '<tr id="frmLevel2">
        <form action="'. $file.'">
            <input type="hidden" name="frmstatus" value="EDIT">
            <input type="hidden" name="'. $field.'" value="'. $id.'">
            <td colspan="2">'. $label.'&nbsp;&nbsp;&nbsp;</td>
            <td>&nbsp;&nbsp;&nbsp;<input type="submit" value="&auml;ndern">&nbsp;&nbsp;&nbsp;</td>
        </form>
    </tr>';
}

//*****Formular*****

public function GETFormRowStyleUPDATE2a($label,$file,$id,$field)
{
    echo '<tr id="frmLevel2">
        <form action="'. $file.'">
            <input type="hidden" name="frmstatus" value="EDIT">
            <input type="hidden" name="'. $field.'" value="'. $id.'">';
    if(is_array($sarrhidden))
    {
        foreach ($sarrhidden as $hidden)
        {
            echo '<input type="hidden" name="'. $hidden['NAME'].'" value="'. $hidden['TID'].'">';
        }
    }
    echo '    <td colspan="2">'. $label.'&nbsp;&nbsp;&nbsp;</td>
        <td>&nbsp;&nbsp;&nbsp;<input type="submit" value="&auml;ndern">&nbsp;&nbsp;&nbsp;</td>
        </form>
    </tr>';
}

//*****Formular*****

public function GETFormRowStyleGESPERRT2($label)
{
    echo '<tr id="frmLevel2">
        <td colspan="2">'. $label.'&nbsp;&nbsp;&nbsp;</td>
        <td>&nbsp;&nbsp;&nbsp;gesperrt&nbsp;&nbsp;&nbsp;</td>
    </tr>';
}

//*****Formular*****

public function GETFormRowStyle3($label)
{
    echo '<tr id="frmLevel3"><td colspan="3">'. $label.'&nbsp;&nbsp;&nbsp;</td></tr>';
}

//*****Formular*****

public function GETFormRowStyleUPDATE3($label,$id,$file)
{
    echo '<tr id="frmLevel3">
        <form action="'. $file.'">
            <input type="hidden" name="frmstatus" value="EDIT">
            <input type="hidden" name="usrtid" value="'. $id.'">
            <td colspan="2">'. $label.'&nbsp;&nbsp;&nbsp;</td>
            <td>&nbsp;&nbsp;&nbsp;<input type="submit" value="&auml;ndern">&nbsp;&nbsp;&nbsp;</td>
        </form>
    </tr>';
}

//*****Formular*****

public function GETFormRowStyleUPDATE3a($label,$file,$id,$field,&$sarrhidden)

```

```

    echo '<tr id="frmLevel3">
        <form action="'&$file.'">
            <input type="hidden" name="frmstatus" value="EDIT">
            <input type="hidden" name="'&$.field.'" value="'&$.id.'">;
if(is_array($arrhidden))
{
    foreach ($arrhidden as $shidden)
    {
        echo '<input type="hidden" name="'&$.shidden['NAME'].'" value="'&$.shidden["TID"].'">;
    }
}
print '   <td colspan="2">'&$.label.'&nbsp;&nbsp;&nbsp;</td>
      <td>&nbsp;&nbsp;&nbsp;<input type="submit" value="&auml&nderrn">&nbsp;&nbsp;&nbsp;</td>
      </form>
      </tr>';
}

//*****Formular*****

public function GETFormRowStyleGESPERRT3($label,$sid)
{
    echo '<tr id="frmLevel3">
        <td colspan="2">'&$.label.'&nbsp;&nbsp;&nbsp;</td>
        <td>&nbsp;&nbsp;&nbsp;gesperrrt&nbsp;&nbsp;&nbsp;</td>
    </tr>';
}

//*****Formular*****

public function GETFormRowStyleTextarea($text)
{
    echo '<tr id="frmLevel4">
        <td colspan="3">
            <textarea name="PARKKOMMENTAR" cols="80" rows="5">'&$.text.'</textarea>
        </tr>';
}

//*****Formular*****

public function GETFormRowStyle4($label,$value,$parid)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'&$.label.'&nbsp;&nbsp;&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'&$.value.'" name="PARID'&$.parid.'" ';
    echo '<td colspan="1"><input type="text" value="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" >&nbsp;&nbsp;&nbsp;</td>';
    echo '<td valign="top"><input type="button" onClick="javascript:help(\''&$.parid.'\')"' value="?">&nbsp;&nbsp;&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

public function GETFormRowStyle4Err($label,$value,$parid)
{
    echo '<tr id="frmLevelError4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'&$.label.'&nbsp;&nbsp;&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'&$.value.'" name="PARID'&$.parid.'" ';
    echo '<td colspan="1"><input type="text" value="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" >&nbsp;&nbsp;&nbsp;</td>';
    echo '<td valign="top"><input type="button" onClick="javascript:help(\''&$.parid.'\')"' value="?">&nbsp;&nbsp;&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

public function GETFormRowStylePARFORMAT4($label,$value,$parid)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'&$.label.'&nbsp;&nbsp;&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'&$.value.'" name="PARID'&$.parid.'" ';
    echo '<td colspan="1"><input type="text" value="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" readonly>&nbsp;&nbsp;&nbsp;</td>';
    echo '<td colspan="1"><input type="button" onClick="javascript:help(\''&$.parid.'\')"' value="?">&nbsp;&nbsp;&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

```

```
public function GETFormRowStyleADMPARFORMAT4($label,$value,$parid)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" >&nbsp;</td>';
    echo '<td valign="top">&nbsp;</td>';
    echo '</tr>';
}

/*****Formular*****/

public function GETFormRowStyleADMPARFORMATRADIO4($label,$value,$parid,$radio)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="">&nbsp;</td>';
    echo '<td align="center"><input type="radio" name="'. $radio. '" value="INS"></td>';
    echo '</tr>';
}

/*****Formular*****/

public function GETFormRowStyleUPDATE4($label,$value,$parid)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" readonly>&nbsp;</td>';
    echo '<td valign="top"><input type="button" onClick="javascript:help(\''. $parid. '\')" value="&auml;ndern">&nbsp;</td>';
    echo '</tr>';
}

/*****Formular*****/

public function GETFormRowStyleUPDATE4temp($label,$value,$feldname,$parid,$file)
{
    echo '<tr id="frmLevel4"><form action="https://'. $_SERVER['SERVER_NAME']. $file. '" method="get">';
    echo '<input type="hidden" name="'. $feldname. '" value="'. $parid. '">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" ';
    echo 'style="" readonly>&nbsp;</td>';
    echo '<td valign="top"><input type="submit" value="&auml;ndern">&nbsp;</td>';
    echo '</form></tr>';
}

/*****Formular*****/

public function GETFormRowStyleUPDATE4Err($label,$value,$feldname,$parid,$file)
{
    echo '<tr id="frmLevelError4"><form action="https://'. $_SERVER['SERVER_NAME']. $file. '" method="get">';
    echo '<input type="hidden" name="'. $feldname. '" value="'. $parid. '">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" ';
    echo 'style="" readonly>&nbsp;</td>';
    echo '<td valign="top"><input type="submit" value="&auml;ndern">&nbsp;</td>';
    echo '</form></tr>';
}

/*****Formular*****/

public function GETFormRowStyleSELECTAdmin4($label,$value,$parid)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" readonly>&nbsp;</td>';
    echo '<td valign="top">&nbsp;</td>';
    echo '</tr>';
}
```



```
//*****Formular*****
```

```
public function GETFormRowStyleUPDATEAdminDEL4a($label,$value,$parid,$field,$id,$format,$size,$radio,$selradio)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="'. $size. '" title="'. $format. '" STYLE="cursor:pointer; cursor:hand;"';
    echo 'size="'. $size. '" style="" readonly>&nbsp;</td>';
    if ($selradio == 'true')
    {
        echo '<td align="center"><input type="radio" name="'. $radio. '" value="'. $id. '">';
    } else {
        echo '<td align="center"><input type="radio" name="'. $radio. '" value="'. $id. '" checked="checked">';
    }
    echo '</td></tr>';
}
```

```
//*****Formular*****
```

```
public function GETFormRowStyleUPDATEAdminDEL4($label,$value,$parid,$field,$id,$format,$size)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '"';
    echo 'maxlength="'. $size. '" title="'. $format. '" STYLE="cursor:pointer; cursor:hand;" size="'. $size. '" style="" readonly>&nbsp;</td>';
    echo '<td align="center"><input type="radio" name="'. $field. '" value="'. $id. '"></td>';
    echo '</tr>';
}
```

```
//*****Formular*****
```

```
public function GETFormRowStyleUPDAdmin4($label,$value,$fieldname,$format,$size)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="'. $label. '" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="'. $fieldname. '"';
    echo 'maxlength="'. $size. '" title="'. $format. '" STYLE="cursor:pointer; cursor:hand;" size="'. $size. '" style="">&nbsp;</td>';
    echo '<td valign="top">&nbsp;<input type="hidden" name="frm'. $fieldname. '" value="'. $format. '"></td>';
    echo '</tr>';
}
```

```
//*****Formular*****
```

```
public function GETFormRowStyleUPDAdminRADIO4($label,$value,$fieldname,$format,$size,$radio,$selradio)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="'. $label. '" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="'. $fieldname. '"';
    echo 'maxlength="'. $size. '" title="'. $format. '" STYLE="cursor:pointer; cursor:hand;" size="'. $size. '"';
    echo 'style="">&nbsp;</td>';
    if ($selradio == 'true')
    {
        echo '<td align="center"><input type="radio" name="'. $radio. '" value="'. $fieldname. '" checked="checked">';
    } else {
        echo '<td align="center"><input type="radio" name="'. $radio. '" value="'. $fieldname. '">';
    }
    echo '<input type="hidden" name="frm'. $fieldname. '" value="'. $format. '"></td>';
    echo '</tr>';
}
```

```
//*****Formular*****
```

```
public function GETFormRowStyleUPDAdminTA4($label,$value,$fieldname,$format,$size)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="'. $label. '" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;</td>';
    echo '<td><textarea name="'. $fieldname. '" title="'. $format. '"';
    echo 'STYLE="cursor:pointer; cursor:hand;" maxlength="'. $size. '" cols="70" rows="5">'. $value. '</textarea></td>';
    echo '<td valign="top">&nbsp;<input type="hidden" name="frm'. $fieldname. '"';
    echo 'value="'. $format. '"></td>';
    echo '</tr>';
}
```

```
/*******Formular*****

public function GETFormRowStyleERRORAdmin4($error)
{
    echo '<tr id="frmLevelError">';
    echo '<td colspan="3">'. $error.'</td>';
    echo '</tr>';
}

/*******Formular*****

public function GETFormRowStyleINSARRAdmin4($label,$value,$fieldname,&$arrSelect)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="'. $label.'" STYLE="cursor:pointer; cursor:hand;">'. $label.'&nbsp;</td>';
    echo '<td colspan="1"><select name="'. $fieldname.'" size="1"><option value="0">Bitte auswählen</option>';
    if (is_array($arrSelect))
    {
        $count = count($arrSelect);
        for ($i=0; $i<$count; $i++)
        {
            if ($arrSelect[$i] == $value)
            {
                echo '<option selected value="'. $arrSelect[$i].">'. $arrSelect[$i].</option>';
            } else {
                echo '<option value="'. $arrSelect[$i].">'. $arrSelect[$i].</option>';
            }
        }
    }
    echo '</select></td>';
    echo '<td valign="top">&nbsp;</td>';
    echo '</tr>';
}

/*******Formular*****

public function GETFormRowStyleINSARRAdminRADIO4($label,$value,$fieldname,&$arrSelect,$radio,$selradio)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="'. $label.'" STYLE="cursor:pointer; cursor:hand;">'. $label.'&nbsp;</td>';
    echo '<td colspan="1"><select name="'. $fieldname.'" size="1"><option value="0">Bitte auswählen</option>';
    if (is_array($arrSelect))
    {
        $count = count($arrSelect);
        for ($i=0; $i<$count; $i++)
        {
            if ($arrSelect[$i] == $value)
            {
                echo '<option selected value="'. $arrSelect[$i].">'. $arrSelect[$i].</option>';
            } else {
                echo '<option value="'. $arrSelect[$i].">'. $arrSelect[$i].</option>';
            }
        }
    }
    echo '</select></td>';
    if ($selradio == 'true')
    {
        echo '<td align="center"><input type="radio" name="'. $radio.'" value="'. $fieldname.'" checked="checked">';
    } else {
        echo '<td align="center"><input type="radio" name="'. $radio.'" value="'. $fieldname.'">';
    }
    echo '</tr>';
}

/*******Formular*****

public function GETFormRowStyleUPDAdminPWD4($label,$value,$fieldname,$format,$size)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="'. $label.'" STYLE="cursor:pointer; cursor:hand;">'. $label.'&nbsp;</td>';
    echo '<td colspan="1"><input type="password" value="'. $value.'" name="'. $fieldname.'"';
    echo 'maxlength="'. $size.'" title="'. $format.'" STYLE="cursor:pointer; cursor:hand;" size="'. $size.'" style="" >&nbsp;</td>';
    echo '<td valign="top"><input type="hidden" name="frm'. $fieldname.'" value="'. $format.'">&nbsp;</td>';
    echo '</tr>';
}

}
```

```
//*****Formular*****

public function GETFormRowStyle6($label,$value,$parid,$file)
{
    echo '<tr id="frmLevel4">';
    echo '<td align="center" colspan="3"><span title="Parameter">';
    echo '<a href="'.$file.'">'. $value.'</a>&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

public function GETFormRowArt($Art,$value)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $Art.'&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value="'.$value.'"';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" >&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

public function GETFormularFooter()
{
    echo '</table></tr></table>';
}

//*****Formular*****

public function GETFormDLPrintLink($excelfile,$pdffile,$asciifile)
{
    echo '<table><tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>';
    echo '<tr><td><b>DOWNLOAD: </b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';

    if ($excelfile <> "0")
    {
        echo '<td><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$excelfile.'">Excel</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
    if ($pdffile <> "0")
    {
        echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$pdffile.'">PDF</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
    if ($asciifile <> "0")
    {
        echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$asciifile.'">Text</a></td>';
    }

    echo '</tr></table>';
}

//*****Formular*****

public function GETFormPrintLink($file,$label)
{
    echo '<p><b>VERWEIS (Link):&nbsp;&nbsp;&nbsp;&nbsp;</b><a href="https://'.$_SERVER['SERVER_NAME'].'/iFlora/admin/'.$file.'">'. $label.'</a></p>';
}

//*****Formular*****

public function GETFormDLPrintLinkAF($excelfile,$pdffile,$asciifile,$orifile)
{
    echo '<table><tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>';
    echo '<tr><td><b>DOWNLOAD: </b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';

    if ($excelfile <> "0")
    {
        echo '<td><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$excelfile.'">Excel</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
    if ($pdffile <> "0")
```

```

{
    echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$pdffile.'">PDF</a></td>';
    echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
}
if ($asciifile <> "0")
{
    echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$asciifile.'">Text</a></td>';
    echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
}
if ($sorifile <> "0")
{
    echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$sorifile.'">ORI-Datei</a></td>';
}
echo '</tr></table>';
}

//*****ENDE*****

}

?>

```

9.3.1.1.3 LayoutManager.php

```

<?php

// This script is called by all Controller-Scripts to print Layout
// DHTML DB elements.
//
// history
// 2005-10-01 1.0 SH begin GETSelect
//
// description
// GETFloraheader => GET Layout Header
// GETFlorafooter => Get Layout Footer

class LAYOUTManager
{
    //*****

function GETFloraheader($ueberschrift,$news,$lblNavigation,$lblSelect,$lblNews,&$arrLINK,&$arrLAYOUT)
{
    echo '<html>
        <head>
            <title>iFLORA Public</title>
            <link rel="STYLESHEET" type="text/css" href="."/style/newiFlora.css">
        </head>
        <body bgcolor="#FFFFFF" link="#FFCC00" vlink="#FFCC00" alink="#FFCC00" text="#000000">
            <table border="0" cellpadding="0" cellspacing="0" width="100%" bgcolor="#004080" text="#FFFFFF">
                <colgroup>
                    <col width="200">
                    <col>
                </colgroup>
                <tr>
                    <td colspan="2">
                        <table border="0" cellpadding="10" cellspacing="0">
                            <tr>
                                <td>
                                    <h1 style="color:#FFFFFF">iFLORA '.$ueberschrift.'</h1>
                                </td>
                                <tr>
                                    <td>
                                        <table border="0" cellpadding="10" cellspacing="0">
                                            <tr>
                                                <td>
                                                    <h3 style="color:#FFFFFF">'.$lblNavigation.'</h3>
                                                    <p>
                                                        <a href=".'.$arrLINK["ref"].'"><b>'.$arrLINK["name"].'</b></a><br>
                                                    </p>

```

[illegible]

```
//*****

function GETFooter()
{
    echo ' </body>
        </html>';
}

//*****

function GETFlorafooter()
{
    echo '          </td>
            </tr>
        </table>
    </td>
    </tr>
</table>
</body>
</html>';
}

//*****

public function GETLabel($label,$format)
{
    print "<$format>$label</$format>";
}

//*****

public function GETRow()
{
    print "<p />";
}

//*****

public function GETReferenceSelect(&$arrREFERENCE)
{
    foreach ($arrREFERENCE as $Daten)
    {
        print '<a href="'. $Daten["ref"]."'><b>'. $Daten["name"].'</b></a><br>';
    }
}

?>
```

9.3.1.2 Controller für einzelnen Seiten

9.3.1.2.1 Startseite – Rahmen mit Menü

```
<?php

//*****

function GET_HTMLHEAD($version,$db)
{
    print '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
    <html lang="de">
    <head>
    <title>iFlora@'. $db.' (R'. $version.')</title>
        <link rel="STYLESHEET" type="text/css" href="/style/newiFlora.css">
        <link rel="STYLESHEET" type="text/css" href="/tree/dtree.css">
    <script language="JavaScript">
    <!--

        var loaded = 0;
        var counter = 0;

        function init() {
```

```

        this.focus();
        var loaded = 1;
    }

    function destroy() {
        loaded = 0;
    }

    function popup(url,features) {
        counter ++;
        fenster = window.open(url,'\window'+counter,features);
        fenster.focus();
    }

    //-->
</script>
</head>
<body style="margin: 2px 2px 2px 2px;" onLoad="init()" onunload="destroy()">
<table width="100%" height="100%" border="0" cellspacing="0" cellpadding="0">
<tr bgcolor="#008000" text="#FFFFFF" height="57">
<td width="100%" align="left" class="header1">&nbsp;&nbsp;&nbsp;iFLORA Administration</td>
<td width="150" align="right">&nbsp;&nbsp;&nbsp;</td>
<td width="120" align="center" class="smalltext" style="background-color:rgb(220, 20, 60);">
<strong>iFloraPublic@'. $db.'<br />R'. $version.'</strong>
<td width="150" align="right">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan=3 height="2"></td>
</tr>
<tr>
<td colspan=3 height="15" background="/layout/hgwasser1.jpg"></td>
</tr>
<tr>
<td valign="top" colspan=3>
<table width="100%" height="100%" border="0" cellspacing="0" style="padding: 2px; padding-left: 0px;">
<tr>;
}

//*****

function GET_HTMLFOOT($usr)
{
    If ($usr != "")
    {
        $usr = ' - '.strtoupper($usr);
    }
    print '
        </tr>
        </table>
        </td>
        </tr>
        <tr height="20" >
        <td id="footer">&nbsp;&nbsp;&nbsp;<strong>iFlora Administration'. $usr.'</strong>&nbsp;&nbsp;&nbsp;</td>
        <td id="footer" align="right" colspan=3>&copy;&nbsp;&nbsp;&nbsp;<i>sh</i>Informatics&nbsp;&nbsp;&nbsp;</td>
        </tr>
        </table>
        </body>
        </html>;
}

//*****

function GET_HTMLTREE($login,$recht,$usr,$usrid,$arobjekt)
{
    print '
        <td width=180 bgcolor="#F0F0F0" valign="top">;
    if ($login == 'true')
    {
        print '<script src="/tree/dtree.js" type="text/javascript"></script>
        <script language="JavaScript">
        <!--

        mytree = new dTree(\'mytree\');
        mytree.config.useCookies=false;
        mytree.config.target=\'work\';

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
mytree.config.folderLinks=false;
mytree.config.useIcons=true;

mytree.icon = {
    root                : \'./tree/images/base.gif\',
    folder              : \'./tree/images/folder.gif\',
    folderOpen          : \'./tree/images/folderopen.gif\',
    node                : \'./tree/images/page.gif\',
    empty               : \'./tree/images/empty.gif\',
    line                : \'./tree/images/line.gif\',
    join                : \'./tree/images/join.gif\',
    joinBottom          : \'./tree/images/joinbottom.gif\',
    plus                : \'./tree/images/plus.gif\',
    plusBottom          : \'./tree/images/plusbottom.gif\',
    minus               : \'./tree/images/minus.gif\',
    minusBottom: \'./tree/images/minusbottom.gif\',
                    nlPlus                : \'./tree/images/nolines_plus.gif\',
                    nlMinus               : \'./tree/images/nolines_minus.gif\'
};

mytree.add(0,-1,\'iFlora\',\'javascript: void(0);\');
$counter = 1;
if ($recht == 'ADMIN' || $recht == 'PROJ' || $recht == 'USER' || $recht == 'TEMP')
{
    $knoten = $counter;
    print 'mytree.add(.$counter.',0,\'Systemverwaltung\',\'./Systemverwaltung\');';
    $counter++;
    if ($recht == 'ADMIN' || $recht == 'PROJ')
    {
        print ' mytree.add(.$counter.',.$knoten.',\'Benutzerverwaltung\',\'./admin/ADMINFormBenutzer.php\',\'Benutzerverwaltung\');';
        $counter++;
        if ($recht == 'ADMIN')
        {
            print ' mytree.add(.$counter.',.$knoten.',\'Projektverwaltung\',\'./admin/ADMINFormProjekt.php\',\'Projektverwaltung\');';
            $counter++;
            print ' mytree.add(.$counter.',.$knoten.',\'Objektartverwaltung\',\'./admin/ADMINFormObjektart.php\',\'Objektartverwaltung\');';
            $counter++;
            print ' mytree.add(.$counter.',.$knoten.',\'IS-IN-Artverwaltung\',\'./admin/ADMINFormISINArt.php\',\'IS-IN-Artverwaltung\');';
            $counter++;
            print ' mytree.add(.$counter.',.$knoten.',\'IS-IN-Objektartverw.',\'./admin/ADMINFormISINOBA.php\',\'IS-IN-Objektartverwaltung\');';
            $counter++;
        }
    }
    else {
        print ' mytree.add(.$counter.',.$knoten.',\'Benutzerpasswort\',\'./work.php?usr=.$usr.&usrid=.$usrid.\',\'neues Benutzerpasswort\');';
        $counter++;
    }
}
if ($recht == 'PROJ' || $recht == 'ADMIN')
{
    $knoten = $counter;
    print 'mytree.add(.$counter.',0,\'Administration\',\'./Administration\');';
    $counter++;
    print ' mytree.add(.$counter.',.$knoten.',\'Parameterverwaltung\',\'./admin/ADMINFormParameter.php\',\'Parameterverwaltung\');';
    $counter++;
    print ' mytree.add(.$counter.',.$knoten.',\'Formularverwaltung\',\'./admin/ADMINFormFormular.php\',\'Formularverwaltung\');';
    $counter++;
}
if ($recht == 'ADMIN' || $recht == 'PROJ' || $recht == 'USER' || $recht == 'TEMP')
{
    $knoten = $counter;
    print 'mytree.add(.$counter.',0,\'Datenbearbeitung\',\'./Datenbearbeitung\');';
    $counter++;
    if (is_array($arobjekt))
    {
        foreach ($arobjekt as $Objekt)
        {
            print '
mytree.add(.$counter.',.$knoten.',\'$Objekt[\'NAME\']\',\'./admin/ADMINSelObjekt.php?obtid=.$Objekt[\'TID\']&obaname=.$Objekt[\'NAME\']\',\'Aufnahmen\');';
            $counter++;
        }
    }
    $arobjekt = NULL;
    print 'mytree.add(.$counter.',.$knoten.',\'Objektbeziehungen\',\'./admin/ADMINFormObjektBez.php\',\'Aufnahmen\');';
    $counter++;
}
```



```

    }
    print ' mytree.add('.$counter.',0,\Schnittstellen\',\'\',\'Schnittstellen\');';
    $knoten = $counter;
    $counter++;
    print ' mytree.add('.$counter.', '.$knoten.',\'Download\',\'./work.php\',\'Download\');';
    $counter++;
    print ' mytree.add('.$counter.', '.$knoten.',\'ORI-Upload\',\'./admin/ADMINUploadORI.php\',\'Download\');';
}

print 'document.write(mytree);
//-->
</script>';
} else {
    print "&nbsp;&nbsp;<br />Nicht angemeldet!";
}
print ' </td>';
}

//*****

function GET_HTMLLogin($fHPueberschrift,$fmeldung,$ffile,$fname,$fpwd,$fbtnLabel)
{
    print '<td valign="top">
        <h1>'.$fHPueberschrift.</h1>
        <table>
            <tr>
                <td>
                    <p />'.$fmeldung.<p />
                    <form method="POST" action="'.$ffile.'">
                        <p />'.$fname.<br>
                        <input name="inputusr" type="text" size="30" maxlength="30">
                        <p />'.$fpwd.<br>
                        <input name="inputpwd" type="password" size="30" maxlength="30">
                        <p />
                        <input type="submit" value="'.$fbtnLabel.'">
                    </form>
                </td>
            </tr>
        </table>
    </td>';
}

//*****

function GET_HTMLData()
{
    print '<td><iframe src="./work.php" name="work" id="work" width="100%" height="100%"
        marginwidth="0" marginheight="0" frameborder="0" border="1"></iframe></td>';
}

//***** MAIN *****

// INCLUDE
include_once('./lib/DBManager.php');
include_once('./lib/LAYOUTManager.php');
include_once('./../config_iflora/sqlpublic.php.inc');
include_once('./../config_iflora/AdminTools.php');

// INIT Konstanten
define("version","0.9");
define("database",$sql_db);

// INIT Variablen
$HPueberschrift = "Anmeldung Administration iFlora";
$meldung = "Bitte melden Sie sich mit Usernamen und Passwort an:";
$meldungProblem = "Die Anmeldung war nicht erfolgreich!<br />
    .Bitte melden Sie sich mit einem gültigen Usernamen und Passwort an:";

$btnLabel = "Anfrage senden";
$file = "index.php";
$name = "Name:";
$pwd = "Passwort:";
$SQLite = './../config_iflora/data/userdb.dat';

$arrPRUEF = NULL;
$arrSQL = NULL;

```

```

$status = "ERROR";

if($_REQUEST['inputusr'] != ""){$in_usr=$_REQUEST['inputusr'];}
if($_REQUEST['inputpwd'] != ""){$in_pwd=$_REQUEST['inputpwd'];}

// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

// INIT AdminTool Objekt
$oPruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

// WICHTIG!!!!
// Bevor etwas ausgegeben (HTML) wird muss das COOKIE gesetzt werden (v.a. LINUX ist da sehr empfindlich)!!!!

// PRÜFPROZEDURE
if ($in_usr != "")
{
    // SET COOKIE
    //if($db=sqlite_open($SQLite, '0666'))
    //{
        $md5pwd = $oPruefDBM->GET_Key($in_pwd);

        // Prüfen, ob Username in der DB vorhanden ist => zurückgegeben wird USERID, STATUS
        $oPruefDBM->GETUserStatus($in_usr,$md5pwd,$arrPRUEF);

        if (is_array($arrPRUEF))
        {
            foreach ($arrPRUEF as $PRUEF)
            {
                $userid = $PRUEF['USERID'];
                $dbstatus = $PRUEF['STATUS'];
            }

            $maxRecht = $oPruefDBM->GETUserRecht($userid);

            If($maxRecht == 'ADMIN' || $maxRecht == 'PROJ')
            {
                $dbuser = 'ADMIN';
            } else {
                $dbuser = 'WORK';
            }

            if ($dbstatus == "aktiv")
            {
                $status = $dbstatus;
                // Cookie setzen und Eintrag in SQLite SESSION
                $sessionidneu = $oPruefDBM->SET_Sessiondata($SQLite,$in_usr,$userid,$dbuser,$maxRecht);
                setcookie("user", $sessionidneu);
            }
        }
    }
} else {
    // $status = "FATAL ERROR[SQLite001] => Problem with SessionDB! Please call your Systemadministrator!";
    //}
    if ($status == "aktiv")
    {
        // Startseite für Anwendung - Anmeldung OK!
        $SQLquery = "SELECT DISTINCT TID, OBANAME AS NAME "
            ."FROM afn_objektart "
            ."ORDER BY OBANAME";

        // GET DB-Daten
        $oDBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);
        GET_HTMLHEAD(version,database);
        GET_HTMLTREE('true',$maxRecht,$in_usr,$userid,$arrSQL);
        GET_HTMLData();
    } else {
        // Startseite für Anwendung - Problem mit Anmeldung!
        GET_HTMLHEAD(version,database);
        GET_HTMLTREE('false','0','','');
        GET_HTMLLogin($HPueberschrift,$meldungProblem,$file,$name,$pwd,$btnLabel);
    }
} else {

```

```
// prüfen, ob $in_usr tatsächlich vorhanden und gültig ist!!!
// RECHTE ermitteln, damit man MENÜ zusammenstellen kann!

// Startseite Kopf
GET_HTMLHEAD(version,database);
// TREE und LOGIN
GET_HTMLTREE('false','0',' ',' ');
GET_HTMLLogin($HPueberschrift,$meldung,$file,$name,$pwd,$btnLabel);
}

GET_HTMLFOOT($in_usr);

?>
```

9.3.1.2.2 Startseite – Work (im Rahmen)

```
<?php

// INIT
include_once(' ../config_iflora/sqladmin.php.inc');
include_once(' ../config_iflora/AdminTools.php');

//INIT
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
$SQLite = " ../config_iflora/data/userdb.dat";

print '<html>
<head>
</head>
<body>';

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $arrCookie = explode(';',$Cookie);
        $aktiv = $pruefDBM->GETUserAktiv($arrCookie[0]);
        if ($aktiv <> "aktiv")
        {
            print 'Ihre Benutzerberechtigung ist abgelaufen! Bitte setzen Sie sich mit Ihren Administrator in Verbindung!';
        }
    } else {
        print "Es ist keine Cookie gesetzt! Vorgang wurde abgebrochen!<br />Bitte aktivieren sie bei Ihrem Browser die Cookies";
    }
} else {
    print "Es ist keine Cookie gesetzt! Vorgang wurde abgebrochen!<br />Bitte aktivieren sie bei Ihrem Browser die Cookies";
}

print ' </body>
</html>';

?>
```

9.3.1.2.3 Controller – „Benutzerverwaltung“ (ADMINFormBenutzer.php)

```
<?php

// INCLUDE
include_once(' ../lib/DBManager.php');
include_once(' ../lib/VIEWManager.php');
include_once(' ../lib/LAYOUTManager.php');
include_once(' ../config_iflora/sqladmin.php.inc');
include_once(' ../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht","PROJ");
$ERROR = "OK";
$DBERROR = "OK";
$arrFehlermeldung = NULL;
$setKey = 'false';
$DBDateFkt = 'CURDATE()';

// INIT Variables
// Formularvariablen
```

```

if($_REQUEST['frmstatus'] != ""){$in_frmstatus=$_REQUEST['frmstatus'];} else {$in_frmstatus = '0';};
// Pflicht
if($_REQUEST['usrtid'] != ""){$in_usrtid=$_REQUEST['usrtid'];} else {$in_usrtid = 0;};
if($_REQUEST['status'] != ""){$in_status=$_REQUEST['status'];} else {$in_status = '0';};
if($_REQUEST['name'] != ""){$in_name=$_REQUEST['name'];} else {$in_name = '0';};
if($_REQUEST['username'] != ""){$in_username=$_REQUEST['username'];} else {$in_username = '0';};
if($_REQUEST['pwd1'] != ""){$in_pwd1=$_REQUEST['pwd1'];} else {$in_pwd1 = '0';};
if($_REQUEST['pwd2'] != ""){$in_pwd2=$_REQUEST['pwd2'];} else {$in_pwd2 = '0';};
// optional
if($_REQUEST['vorname'] != ""){$in_vorname=$_REQUEST['vorname'];} else {$in_vorname = '-';};
if($_REQUEST['email'] != ""){$in_email=$_REQUEST['email'];} else {$in_email = '-';};
// Rechte&Projekt
if($_REQUEST['urnewprj'] != ""){$in_urnewprj=$_REQUEST['urnewprj'];} else {$in_urnewprj = '0';};
if($_REQUEST['urnewrht'] != ""){$in_urnewrht=$_REQUEST['urnewrht'];} else {$in_urnewrht = '0';};
if($_REQUEST['delurid'] != ""){$in_delurid=$_REQUEST['delurid'];} else {$in_delurid = '0';};
// Formate
if($_REQUEST['frmname'] != ""){$in_frmname=$_REQUEST['frmname'];} else {$in_frmname = '0';};
if($_REQUEST['frmusername'] != ""){$in_frmusername=$_REQUEST['frmusername'];} else {$in_frmusername = '0';};
if($_REQUEST['frmpwd1'] != ""){$in_frmpwd=$_REQUEST['frmpwd1'];} else {$in_frmpwd = '0';};
if($_REQUEST['frmvorname'] != ""){$in_frmvorname=$_REQUEST['frmvorname'];} else {$in_frmvorname = '0';};
if($_REQUEST['frmemail'] != ""){$in_frmemail=$_REQUEST['frmemail'];} else {$in_frmemail = '0';};
// Filter
if($_REQUEST['fltname'] != ""){$in_fltname=$_REQUEST['fltname'];} else {$in_fltname = '0';};
if($_REQUEST['fltusername'] != ""){$in_fltusername=$_REQUEST['fltusername'];} else {$in_fltusername = '0';};

$SarrFormular = NULL;
$SarrRight = NULL;
$SarrStatus = array("aktiv",
                    "gesperrt");
$SarrProjekt = NULL;
$SarrRecht = NULL;
$Scounter = 0;
$Sfile = ".ADMINFormBenutzer.php";

// INIT AdminTool Objekt
$SQLite = "../././config_iflora/data/userdb.dat";
$SpruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$SoVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$SoLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $SCookie = $SpruefDBM->GET_Cookie($SQLite);
    if ($SCookie <> "ERROR")
    {
        $SarrCookie = explode(';', $SCookie);
        $Saktiv = $SpruefDBM->GETUserAktiv($SarrCookie[0]);
        if ($Saktiv == "aktiv")
        {
            $SPRUEFBerechtigung = $SpruefDBM->PRUEFBerechtigung(formularrecht,$SarrCookie[2]);
            if ($SPRUEFBerechtigung > 0)
            {
                if ($SarrCookie[1] == 'ADMIN')
                {
                    $Suser = $Sadmin_user;
                    $Spasswd = $Sadmin_passwd;
                } elseif ($SarrCookie[1] == 'WORK') {
                    $Suser = $Swork_user;
                    $Spasswd = $Swork_passwd;
                } else {
                    $Suser = $Sdata_user;
                    $Spasswd = $Sdata_passwd;
                }
            }
        }

        // INIT DBManager Objekt
        $SoDBM = new DBManager($database,$sql_server,$Suser,$Spasswd,$sql_db,$schema);

        // INIT - Projektnamen holen
        $SqueryProjekte = "SELECT PRJNAME "
                        ."FROM afn_projekt ";
        if ($SarrCookie[2] != 'ADMIN')
        {
            $SqueryProjekte = $SqueryProjekte."WHERE PRJNAME <> 'ADMIN' ";
        }
    }
}

```

```

} elseif ($arrCookie[2] != 'PROJ' && $arrCookie[2] != 'ADMIN') {
    $queryProjekte = $queryProjekte."WHERE TID IN (SELECT PROJEKTID "
        ."FROM adm_user_rechte "
        ."WHERE USERID = $arrCookie[0]";
}
$arrProjekt = $oDBM->GETSelectSQLSelect($queryProjekte);

// INIT - Rechtenamen holen
$queryRechte = "SELECT NAME "
    ."FROM adm_recht ";
if ($arrCookie[2] != 'ADMIN')
{
    $queryRechte = $queryRechte."WHERE KUERZEL <> 'ADMIN' ";
}
$arrRecht = $oDBM->GETSelectSQLSelect($queryRechte);

// ANWENDUNG
// 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
// 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
// UPDATE
if ($in_frmstatus == 'UPDATE')
{
    if ($arrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $PREUFqueryDS_USER = "SELECT DS_USER "
            ."FROM adm_username "
            ."WHERE TID = '$in_usrtid' ";
        $PRUEFds_user = $oDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
        $PRUEFProjekt = $pruefDBM->PRUEFProjekt($arrCookie[0], $in_usrtid);
        // Ein Projektleiter darf alle Benutzer bearbeiten, die seinem Projekt zugeordnet oder von ihm angelegt sind!
        if ($PRUEFProjekt > 0 || $arrCookie[0] == $PRUEFds_user)
        {
            $DBchange = 'true';
        } else {
            $DBchange = 'false';
            $arrFehlermeldung[] = "Ein Projektleiter darf nur Benutzer seines Projektes oder Benutzer, die er selbst angelegt hat,
bearbeiten! Das trifft auf den gewählten Benutzer nicht zu!";
            $in_frmstatus = "EDIT";
        }
    }
    if ($DBchange == 'true')
    {
        if ($in_usrtid > 0 && $in_username != '0' && $in_name != '0' && $in_status != '0')
        {
            # Prüfen, ob Datensatz vorhanden und eindeutig ist!
            $PRUEFqueryUPDATE = "SELECT count(*) "
                ."FROM adm_username "
                ."WHERE TID = '$in_usrtid' ";
            $PRUEFds = $oDBM->GETSelectSQLDatum($PRUEFqueryUPDATE);
            if ($PRUEFds == 1)
            {
                $ds = 'true';
            } else {
                $ds = 'false';
                $arrFehlermeldung[] = "Der ausgewählte Benutzer ist nicht (mehr) in der Datenbank vorhanden!";
            }
        }

        # Prüfen, ob "username" (UNIQUE INDEX) eindeutig ist!
        $PRUEFqueryUSERNAME = "SELECT count(*) "
            ."FROM adm_username "
            ."WHERE USERNAME = '$in_username' "
            ."AND TID <> '$in_usrtid' ";
        $PRUEFun = $oDBM->GETSelectSQLDatum($PRUEFqueryUSERNAME);
        if ($PRUEFun == 0)
        {
            $un = 'true';
        } else {
            $un = 'false';
            $arrFehlermeldung[] = "Der der eingegebene <b>USERNAME</b> muss eindeutig sein - der eingegebene Username wird
bereits für einen anderen Benutzer verwendet!";
        }
    }
}

```

```
// Passwort prüfen
if ($in_pwd1 == $in_pwd2)
{
    if ($in_pwd1 != '0')
    {
        $Key = $pruefDBM->GET_Key($in_pwd1);
        $setKey = 'true';
        $PRUEFpwd = $oVWM->PRUEFFormat($in_frmpwd,$in_pwd1);
    } else {
        $PRUEFpwd = 'true';
    }
} else {
    $DBERROR = 'ERROR';
    $arrFehlermeldung[] = "Beim &Auml;ndern des Passwortes m&uuml;ssen beide Felder mit dem gleichen (neuen) Passwort
befüllt werden! ";
}

// Formatprüfung der eingegebenen Werte
$PRUEFname = $oVWM->PRUEFFormat($in_frmname,$in_name);
$PRUEFusername = $oVWM->PRUEFFormat($in_frmusername,$in_username);
$PRUEFvorname = $oVWM->PRUEFFormat($in_frmvorname,$in_vorname);
$PRUEFemail = $oVWM->PRUEFFormat($in_frmemail,$in_email);

if ($ds == 'true' &&
    $un == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFname == 'true' &&
    $PRUEFusername == 'true' &&
    $PRUEFpwd == 'true' &&
    $PRUEFvorname == 'true' &&
    $PRUEFemail == 'true')
{
    $queryUPDATE = "UPDATE adm_username SET DS_DATE = $DBDateFkt, "
                    . "DS_USER = '$arrCookie[0]', "
                    . "NAME = '$in_name', "
                    . "VORNAME = '$in_vorname', "
                    . "USERNAME = '$in_username', "
                    . "E_MAIL = '$in_email', "
                    . "STATUS = '$in_status' ";

    if ($setKey == 'true')
    {
        $queryUPDATE = $queryUPDATE.", PWD = '$Key' ";
    }
    $queryUPDATE = $queryUPDATE."WHERE TID = '$in_usrtid' ";
    $oDBM->GETUpdateSQLDatum($queryUPDATE);

    # Löschen eines Userrechtes!
    if ($in_delurid != '0')
    {
        # Prüfen, ob zu löschende "UserRecht" vorhanden ist!
        $PRUEFqueryUSERRECHT = "SELECT count(*) "
                                . "FROM adm_user_rechte "
                                . "WHERE TID <> '$in_delurid' ";
        $PRUEFUserrecht = $oDBM->GETSelectSQLDatum($PRUEFqueryUSERRECHT);

        if ($PRUEFUserrecht > 0)
        {
            $DELqueryUSERRECHT = "DELETE FROM adm_user_rechte WHERE TID = '$in_delurid' ";
            $oDBM->GETUpdateSQLDatum($DELqueryUSERRECHT);
        }
    }

    if ($in_urnewrht != "0" && $in_urnewprj != "0")
    {
        $queryINS_UR = "INSERT adm_user_rechte SET DS_DATE = $DBDateFkt, "
                      . "DS_USER = '$arrCookie[0]', "
                      . "USERID = '$in_usrtid', "
                      . "RECHTEID = (SELECT TID "
                      . "FROM adm_recht "
                      . "WHERE NAME = '$in_urnewrht'), "
                      . "AND PROJEKTID = (SELECT TID "
                      . "FROM afn_projekt "
                      . "WHERE PRJNAME = '$in_urnewprj') ";
        $oDBM->GETUpdateSQLDatum($queryINS_UR);
    }
}
```

```

    } else {
        if ($PRUEFname == 'false'){
            $arrFehlermeldung[] = "<b>Name:</b> Der Name darf nicht länger als 40 Buchstaben sein!";
        }
        if ($PRUEFusername == 'false')
        {
            $arrFehlermeldung[] = "<b>Username:</b> Der Username darf nicht länger als 40 Buchstaben sein!";
        }
        if ($PRUEFpwd == 'false')
        {
            $arrFehlermeldung[] = "<b>Passwort:</b> Das Passwort darf nicht länger als 30 Buchstaben sein!";
        }
        if ($PRUEFvorname == 'false')
        {
            $arrFehlermeldung[] = "<b>Vorname:</b> Der Vorname darf nicht länger als 40 Buchstaben sein!";
        }
        if ($PRUEFemail == 'false')
        {
            $arrFehlermeldung[] = "<b>Email:</b>Die email-Adresse darf nicht länger als 40 Buchstaben sein!";
        }
        $in_frmstatus = "EDIT";
    }
} else {
    if ($in_username == '0')
    {
        $arrFehlermeldung[] = "<b>Username:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_username = "";
    }
    if ($in_name == '0')
    {
        $arrFehlermeldung[] = "<b>Name:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_name = "";
    }
    if ($in_status == '0')
    {
        $arrFehlermeldung[] = "<b>Status:</b> Dieses Feld ist ein Pflichtfeld und es muss eine Wert aus der Auswahlliste ausgewählt werden!";
        $in_status = "";
    }
    $in_frmstatus = "EDIT";
}
}

// INSERT
if ($in_frmstatus == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN' || $arrCookie[2] == 'PROJ')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "Nur ein Administrator oder ein Projektleiter darf nur einen neuen Benutzer anlegen!";
        $in_frmstatus = "NEU";
    }
}

if ($DBchange == 'true')
{
    if ($in_username != '0' && $in_name != '0' && $in_status != '0' && $in_pwd1 != '0')
    {
        # Prüfen, ob "username" (UNIQUE INDEX) eindeutig ist!
        $PRUEFqueryUSERNAME = "SELECT count(*) "
            . "FROM adm_username "
            . "WHERE USERNAME = '$in_username' ";
        $PRUEFun = $oDBM->GETSelectSQLDatum($PRUEFqueryUSERNAME);

        if ($PRUEFun == 0)
        {
            $inINS = 'true';
        } else {
            $inINS = 'false';
            $arrFehlermeldung[] = "<b>Username:</b> Der <b>USERNAME</b> muss eindeutig sein. Der von Ihnen eingegebene <b>USERNAME</b> wird bereits bei einem anderen Benutzer verwendet!";
        }
    }
}

```

```
// Passwort prüfen
if ($in_pwd1 == $in_pwd2)
{
    if ($in_pwd1 != '0')
    {
        $Key = $pruefDBM->GET_Key($in_pwd1);
    } else {
        $DBERROR = 'ERROR';
        $arrFehlermeldung[] = "<b>Passwort:</b> Beim Anlegen eines neuen Benutzers muss ein Passwort eingegeben werden! ";
    }
} else {
    $DBERROR = 'ERROR';
    $arrFehlermeldung[] = "<b>Passwort:</b> Beim Anlegen eines neuen Benutzers müssen zur Kontrolle in beide
Passwortfelder das gleiche Passwort eingegeben werden! ";
}

// Formatprüfung der eingegebenen Werte
$PRUEFname = $oVWM->PRUEFFormat($in_frmname,$in_name);
$PRUEFusername = $oVWM->PRUEFFormat($in_frmusername,$in_username);
$PRUEFpwd = $oVWM->PRUEFFormat($in_frmpwd,$in_pwd1);
$PRUEFvorname = $oVWM->PRUEFFormat($in_frmvorname,$in_vorname);
$PRUEFemail = $oVWM->PRUEFFormat($in_frmemail,$in_email);

if ($unINS == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFname == 'true' &&
    $PRUEFusername == 'true' &&
    $PRUEFpwd == 'true' &&
    $PRUEFvorname == 'true' &&
    $PRUEFemail == 'true')
{
    $queryINSERT = "INSERT INTO adm_username SET DS_DATE = $DBDateFkt, "
        . "DS_USER = '$arrCookie[0]', "
        . "NAME = '$in_name', "
        . "VORNAME = '$in_vorname', "
        . "USERNAME = '$in_username', "
        . "E_MAIL = '$in_email', "
        . "STATUS = '$in_status', "
        . "PWD = '$Key' ";

    $oDBM->GETUpdateSQLDatum($queryINSERT);

    if ($in_urnewrht != "0" && $in_urnewprj != "0")
    {
        $GETqueryUSERNAME = "SELECT TID "
            . "FROM adm_username "
            . "WHERE USERNAME = '$in_username' ";
        $sql_usrtid = $oDBM->GETSelectSQLDatum($GETqueryUSERNAME);

        $queryINS_UR = "INSERT adm_user_rechte SET DS_DATE = $DBDateFkt, "
            . "DS_USER = '$arrCookie[0]', "
            . "USERID = '$sql_usrtid', "
            . "RECHTEID = (SELECT TID "
                . "FROM adm_recht "
                . "WHERE NAME = '$in_urnewrht'), "
            . "PROJEKTID = (SELECT TID "
                . "FROM afn_projekt "
                . "WHERE PRJNAME = '$in_urnewprj') ";

        $oDBM->GETUpdateSQLDatum($queryINS_UR);
    }
} else {
    if ($PRUEFname == 'false')
    {
        $arrFehlermeldung[] = "<b>Name:</b> Der Name darf nicht länger als 40 Buchstaben sein!";
    }
    if ($PRUEFusername == 'false')
    {
        $arrFehlermeldung[] = "<b>Username:</b> Der Username darf nicht länger als 40 Buchstaben sein!";
    }
    if ($PRUEFpwd == 'false')
    {
        $arrFehlermeldung[] = "<b>Passwort:</b> Das Passwort darf nicht länger als 30 Buchstaben sein!";
    }
    if ($PRUEFvorname == 'false')

```



```

        {
            $arrFehlermeldung[] = "<b>Vorname:</b> Der Vorname darf nicht länger als 40 Buchstaben sein!";
        }
        if ($PRUEFemail == 'false')
        {
            $arrFehlermeldung[] = "<b>Email:</b>Die email-Adresse darf nicht länger als 40 Buchstaben sein!";
        }
        $in_frmstatus = "NEU";
    }
    } else {
        if ($in_username == '0')
        {
            $arrFehlermeldung[] = "<b>Username:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
            $in_username = "";
        }
        if ($in_name == '0')
        {
            $arrFehlermeldung[] = "<b>Name:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
            $in_name = "";
        }
        if ($in_status == '0')
        {
            $arrFehlermeldung[] = "<b>Status:</b> Dieses Feld ist ein Pflichtfeld und es muss eine Wert aus der Auswahlliste ausgewählt
werden!";
            $in_status = "";
        }
        if ($in_pwd1 == '0')
        {
            $arrFehlermeldung[] = "<b>Passwort:</b> Es muss eine Passwort vergeben werden!";
            $in_pwd1 = "";
            $in_pwd2 = "";
        }
        $in_frmstatus = "NEU";
    }
}
}

// EDITMODE
if ($in_frmstatus == 'EDIT')
{
    $SELECTquery = "SELECT TID AS USRID, NAME, VORNAME, USERNAME, STATUS, E_MAIL, PWD, DS_USER "
        ."FROM adm_username "
        ."WHERE TID = $in_usrtid "
        ."ORDER BY NAME ";

    $oDBM-
>GETSelectSQL($SELECTquery,'USRID,NAME,VORNAME,USERNAME,STATUS,E_MAIL,PWD,DS_USER',$arrFormular);

    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltungUPD('Benutzerdaten ändern',$file,'usrtid',$in_usrtid,'UPDATE');

    if (is_array($arrFormular))
    {
        foreach ($arrFormular as $Daten)
        {
            if ($arrCookie[2] == 'ADMIN')
            {
                $change = 'true';
            } else {
                $PRUEFProjekt = $pruefDBM->PRUEFProjekt($arrCookie[0],$in_usrtid);
                // Ein Projektleiter darf alle Benutzer bearbeiten, die seinem Projekt zugeordnet oder von ihm angelegt sind!
                if ($PRUEFProjekt > 0 || $arrCookie[0] == $Daten['DS_USER'])
                {
                    $change = 'true';
                } else {
                    $change = 'false';
                }
            }
        }

        if ($change == 'true')
        {
            if (is_array($arrFehlermeldung))
            {

```

```

$VWM->GETFormRowStyle2('FEHLERMELDUNGEN');
$count = count($arrFehlermeldung);
for ($i=0; $i<$count; $i++)
{
    $VWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
}
$VWM->GETFormRowStyle2('&nbsp;');
$VWM->GETFormRowStyleUPDAdmin4('Name',$sin_name,'name','X(40)',40);
$VWM->GETFormRowStyleUPDAdmin4('Vorname',$sin_vorname,'vorname','X(40)',40);
$VWM->GETFormRowStyleUPDAdmin4('Username',$sin_username,'username','X(40)',40);
$VWM->GETFormRowStyleINSARRAdmin4('Status',$sin_status,'status',$sarrStatus);
$VWM->GETFormRowStyleUPDAdmin4('email Adresse',$sin_email,'email','X(40)',40);
$VWM->GETFormRowStyleUPDAdminPWD4('neue Passwort','pwd1','X(30)',30);
$VWM->GETFormRowStyleUPDAdminPWD4('neue Passwort<br />wiederholen','pwd2','X(30)',30);
// Rechtedaten des Users aus der Datenbank holen!
$queryRight = "SELECT r.name AS RECHTNAME, ur.tid AS TID, p.prjname AS PROJNAME "
               ."FROM adm_user_rechte ur, "
               ."adm_recht r, "
               ."afn_projekt p "
               ."WHERE ur.rechteID = r.tid "
               ."AND ur.userID = ".$Daten['USRID']. " "
               ."AND ur.projektID = p.tid";
$DBM->GETSelectSQL($queryRight,'RECHTNAME,TID,PROJNAME',$sarrRight);
if (is_array($sarrRight))
{
    $VWM->GETFormRowStyle2('Rechte/Projekt löschen');
    foreach ($sarrRight as $Right)
    {
        $VWM->GETFormRowStyleUPDATEAdminDEL4('Recht => Projekt&nbsp;',$Right['RECHTNAME']. " =>
        ".$Right['PROJNAME'],$Daten['USRID'],'delurid',$Right['TID'],'X(40)',40);
    }
    $sarrRight = NULL;
}
$VWM->GETFormRowStyle2('Rechte/Projekt neu');
$VWM->GETFormRowStyleINSARRAdmin4('Recht','urnewrht',$sarrRecht);
} else {
    $VWM->GETFormRowStyle2('&nbsp;');
    $VWM->GETFormRowStyleUPDAdmin4('Name',$Daten['NAME'],'name','X(40)',40);
    $VWM->GETFormRowStyleUPDAdmin4('Vorname',$Daten['VORNAME'],'vorname','X(40)',40);
    $VWM->GETFormRowStyleUPDAdmin4('Username',$Daten['USERNAME'],'username','X(40)',40);
    $VWM->GETFormRowStyleINSARRAdmin4('Status',$Daten['STATUS'],'status',$sarrStatus);
    $VWM->GETFormRowStyleUPDAdmin4('email Adresse',$Daten['E_MAIL'],'email','X(40)',40);
    $VWM->GETFormRowStyleUPDAdminPWD4('neue Passwort','pwd1','X(30)',30);
    $VWM->GETFormRowStyleUPDAdminPWD4('neue Passwort<br />wiederholen','pwd2','X(30)',30);
    // Rechtedaten des Users aus der Datenbank holen!
    $queryRight = "SELECT r.name AS RECHTNAME, ur.tid AS TID, p.prjname AS PROJNAME "
                  ."FROM adm_user_rechte ur, "
                  ."adm_recht r, "
                  ."afn_projekt p "
                  ."WHERE ur.rechteID = r.tid "
                  ."AND ur.userID = ".$Daten['USRID']. " "
                  ."AND ur.projektID = p.tid";
    $DBM->GETSelectSQL($queryRight,'RECHTNAME,TID,PROJNAME',$sarrRight);
    if (is_array($sarrRight))
    {
        $VWM->GETFormRowStyle2('Rechte/Projekt löschen');
        foreach ($sarrRight as $Right)
        {
            $VWM->GETFormRowStyleUPDATEAdminDEL4('Recht => Projekt&nbsp;',$Right['RECHTNAME']. " =>
            ".$Right['PROJNAME'],$Daten['USRID'],'delurid',$Right['TID'],'X(40)',40);
        }
        $sarrRight = NULL;
    }
    $VWM->GETFormRowStyle2('Rechte/Projekt neu');
    $VWM->GETFormRowStyleINSARRAdmin4('Projekt','urnewprj',$sarrProjekt);
    $VWM->GETFormRowStyleINSARRAdmin4('Recht','urnewrht',$sarrRecht);
}
}
}
$VWM->GETFormularFooterUserverwaltungUPD();
// HTML-Seite Ende
$oLTM->GETFooter();
if ($change == 'false')
{

```

```

$ERROR = "Diese Seite darf nur von Administratoren und Projektleitern des Users bearbeitet werden!";
}
// INSERTMODE
} elseif ($in_frmstatus == 'NEU'){
// HTML-Seite Anfang
SoLTM->GETFORMHeader();

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltungUPD('neuen Benutzer anlegen',$file,'usrtid',$in_usrtid,'INSERT');
if (is_array($sarrFehlermeldung))
{
    $soVWM->GETFormRowStyle2('FEHLERMELDUNGEN');
    $count = count($sarrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $soVWM->GETFormRowStyleERRORAdmin4($sarrFehlermeldung[$i]);
    }
    $soVWM->GETFormRowStyle2('&nbsp;');
    $soVWM->GETFormRowStyleUPDAdmin4('Name',$in_name,'name','X(40)',40);
    $soVWM->GETFormRowStyleUPDAdmin4('Vorname',$in_vorname,'vorname','X(40)',40);
    $soVWM->GETFormRowStyleUPDAdmin4('Username',$in_username,'username','X(40)',40);
    $soVWM->GETFormRowStyleINSARRAdmin4('Status',$in_status,'status',$sarrStatus);
    $soVWM->GETFormRowStyleUPDAdmin4('email Adresse',$in_email,'email','X(40)',40);
    $soVWM->GETFormRowStyleUPDAdminPWD4('Passwort',$in_pwd1,'pwd1','X(30)',30);
    $soVWM->GETFormRowStyleUPDAdminPWD4('Passwort<br />wiederholen',$in_pwd1,'pwd2','X(30)',30);
    $soVWM->GETFormRowStyle2('Rechte/Projekt');
    $soVWM->GETFormRowStyleINSARRAdmin4('Projekt',$in_urnewprj,'urnewprj',$sarrProjekt);
    $soVWM->GETFormRowStyleINSARRAdmin4('Recht',$in_urnewrht,'urnewrht',$sarrRecht);
    $soVWM->GETFormularFooterUserverwaltungUPD();
} else {
    $soVWM->GETFormRowStyle2('&nbsp;');
    $soVWM->GETFormRowStyleUPDAdmin4('Name','','name','X(40)',40);
    $soVWM->GETFormRowStyleUPDAdmin4('Vorname','','vorname','X(40)',40);
    $soVWM->GETFormRowStyleUPDAdmin4('Username','','username','X(40)',40);
    $soVWM->GETFormRowStyleINSARRAdmin4('Status','','status',$sarrStatus);
    $soVWM->GETFormRowStyleUPDAdmin4('email Adresse','','email','X(40)',40);
    $soVWM->GETFormRowStyleUPDAdminPWD4('Passwort','','pwd1','X(30)',30);
    $soVWM->GETFormRowStyleUPDAdminPWD4('Passwort<br />wiederholen','','pwd2','X(30)',30);
    $soVWM->GETFormRowStyle2('Rechte/Projekt');
    $soVWM->GETFormRowStyleINSARRAdmin4('Projekt','','urnewprj',$sarrProjekt);
    $soVWM->GETFormRowStyleINSARRAdmin4('Recht','','urnewrht',$sarrRecht);
    $soVWM->GETFormularFooterUserverwaltungUPD();
}

// HTML-Seite Ende
SoLTM->GETFooter();
// READMODE (auch nach UPDATE!)
} else {
// GET Filter-Daten
$GETqueryFILTER = "SELECT TID AS USRID, concat(NAME,' ',VORNAME) AS NAME, USERNAME "
                ."FROM  adm_username "
                ."ORDER BY NAME ";
$soDBM->GETSelectSQL($GETqueryFILTER,'USRID,NAME,USERNAME',$sarrFilter);

// HTML-Seite Anfang
SoLTM->GETFORMHeader();

// Filter Anfang
if (is_array($sarrFilter))
{
    foreach ($sarrFilter as $Filter)
    {
        $HashName["TID"] = $Filter['USRID'];
        $HashName["NAME"] = $Filter['NAME'];
        $HashUsername["TID"] = $Filter['USRID'];
        $HashUsername["NAME"] = $Filter['USERNAME'];
        $Name[] = $HashName;
        $Username[] = $HashUsername;
    }
}
$soVWM->GETFormular2Filter('Name','fltname',$Name,'Username','fltusername',$Username);

// GET DB-Daten
$query = "SELECT TID AS USRID, concat(NAME,' ',VORNAME) AS NAME, USERNAME, STATUS, E_MAIL, DS_USER "
        ."FROM  adm_username ";

```

```

// Filterdaten einbinden
if ($in_fltname != '0')
{
    $query = $query."WHERE TID = $in_fltname ";
} elseif ($in_ftusername != '0') {
    $query = $query."WHERE TID = $in_ftusername ";
}
$query = $query."ORDER BY NAME ";
$soDBM->GETSelectSQL($query,'USRID,NAME,USERNAME,STATUS,E_MAIL,DS_USER',$sarrFormular);

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltung('Benutzerverwaltung',$file);
if (is_array($sarrFormular))
{
    foreach ($sarrFormular as $Daten)
    {
        {
            if ($scounter > 0)
            {
                $soVWM->GETFormRowStyle1('&nbsp;');
            }
            $soVWM->GETFormRowStyle2('&nbsp;');
            if ($sarrCookie[2] == 'ADMIN')
            {
                $soVWM->GETFormRowStyleUPDATE3($Daten['NAME'],$Daten['USRID'],$file);
            } else {
                $PRUEFProjekt = $pruefDBM->PRUEFProjekt($sarrCookie[0],$Daten['USRID']);
                // Ein Projektleiter darf alle Benutzer bearbeiten, die seinem Projekt zugeordnet oder von ihm angelegt sind!
                if ($PRUEFProjekt > 0 || $sarrCookie[0] == $Daten['DS_USER'])
                {
                    {
                        $soVWM->GETFormRowStyleUPDATE3($Daten['NAME'],$Daten['USRID'],$file);
                    } else {
                        $soVWM->GETFormRowStyleGESPERRT3($Daten['NAME'],$Daten['USRID']);
                    }
                }
            }
            $soVWM->GETFormRowStyleSELECTAdmin4('Username',$Daten['USERNAME'],$Daten['USRID']);
            $soVWM->GETFormRowStyleSELECTAdmin4('Status',$Daten['STATUS'],$Daten['USRID']);
            $soVWM->GETFormRowStyleSELECTAdmin4('email Adresse',$Daten['E_MAIL'],$Daten['USRID']);

            // Rechtedaten des Users aus der Datenbank holen!
            $queryRight = "SELECT  r.name AS RECHTNAME, ur.projektid AS PRJID, p.prjname AS PROJNAME "
                ."FROM    adm_user_rechte ur, "
                ."        adm_recht r, "
                ."        afn_projekt p "
                ."WHERE   ur.rechteID = r.tid "
                ."AND     ur.userID = ".$Daten['USRID']. " "
                ."AND     ur.projektID = p.tid";
            $soDBM->GETSelectSQL($queryRight,'RECHTNAME,PRJID,PROJNAME',$sarrRight);

            if (is_array($sarrRight))
            {
                {
                    foreach ($sarrRight as $Right)
                    {
                        {
                            $soVWM->GETFormRowStyleSELECTAdmin4('Recht => Projekt&nbsp;',$Right['RECHTNAME']. " =>
". $Right['PROJNAME'],$Daten['USRID']);
                        }
                        $sarrRight = NULL;
                    }
                    $scounter++;
                }
            }
        }
    }
    $soVWM->GETFormularFooter();

    // Filter (Tabelle) Ende
    $soVWM->GETFormularFilterEnde();

    // HTML-Seite Ende
    $soLTM->GETFooter();
}
// ENDE ANWENDUNG
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihren Administrator in Verbindung!";
}
} else {

```

```

$ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihren Systemadministrator in
Verbindung!";
}
} else {
$ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle
Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://".$_SERVER['SERVER_NAME']."'iFlora>Startseite</a> oder aktivieren
sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
$oLTM->GETHeader();
$oVWM->GETLabel("iFlora Parameterverwaltung", "H1");
$oVWM->GETTextfield($ERROR);
$oLTM->GETFooter();
}

?>

```

9.3.1.2.4 Controller – „Projektverwaltung“ (ADMINFormProjekt.php)

```

<?php

// INCLUDE
include_once('..lib/DBManager.php');
include_once('..lib/VIEWManager.php');
include_once('..lib/LAYOUTManager.php');
include_once('..../config_iflora/sqladmin.php.inc');
include_once('..../config_iflora/sqladmin.php.inc');
include_once('..../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht", "ADMIN");
$ERROR = "OK";
$DBERROR = "OK";
$arrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$stable = 'afn_projekt';
$arrHidden = NULL;

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$sin_frmstatus=$_REQUEST['frmstatus'];} else {$sin_frmstatus = '0';};
// Parametervariablen
if($_REQUEST['prjtid'] != ""){$sin_prjtid=$_REQUEST['prjtid'];} else {$sin_prjtid = 0;};
// Pflicht
if($_REQUEST['prjname'] != ""){$sin_prjname=$_REQUEST['prjname'];} else {$sin_prjname = 0;};
// optional
if($_REQUEST['prjkommentar'] != ""){$sin_prjkommentar=$_REQUEST['prjkommentar'];} else {$sin_prjkommentar = ""};
// Formate
if($_REQUEST['frmprjname'] != ""){$sin_frmprjname=$_REQUEST['frmprjname'];} else {$sin_frmprjname = 0;};
if($_REQUEST['frmprjkommentar'] != ""){$sin_frmprjkommentar=$_REQUEST['frmprjkommentar'];} else {$sin_frmprjkommentar = 0;};
// Filter
if($_REQUEST['fltprj'] != ""){$sin_fltprj=$_REQUEST['fltprj'];} else {$sin_fltprj = '0';};

$arrFormular = NULL;
$arrFilter = NULL;
$arrFilter2 = NULL;
$counter = 0;
$file = "../ADMINFormProjekt.php";

$arrProjekt = NULL;
$arrWLKlasse = NULL;
$arrISINArt = NULL;
$arrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = "../..../config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$oVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

```

```

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $arrCookie = explode(';', $Cookie);
        $aktiv = $pruefDBM->GETUserAktiv($arrCookie[0]);
        if ($aktiv == "aktiv")
        {
            $PRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht, $arrCookie[2]);
            if ($PRUEFBerechtigung > 0)
            {
                if ($arrCookie[1] == 'ADMIN')
                {
                    $user = $admin_user;
                    $passwd = $admin_passwd;
                } elseif ($arrCookie[1] == 'WORK') {
                    $user = $work_user;
                    $passwd = $work_passwd;
                } else {
                    $user = $data_user;
                    $passwd = $data_passwd;
                }

                // INIT DBManager Objekt
                $oDBM = new DBManager($database, $sql_server, $user, $passwd, $sql_db, $schema);

                // ANWENDUNG
                // 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
                // 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
                / UPDATE
                if ($in_frmstatus == 'UPDATE')
                {
                    if ($arrCookie[2] == 'ADMIN')
                    {
                        $DBchange = 'true';
                    } else {
                        $PREUFqueryDS_USER = "SELECT DS_USER "
                                                ."FROM   adm_username "
                                                ."WHERE  TID = '$in_usrtid' ";
                        $PRUEFds_user = $oDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
                        $PRUEFProjekt = $pruefDBM->PRUEFProjTable($arrCookie[0], $in_prjtid, $table);
                        if ($PRUEFProjekt > 0 || $arrCookie[0] == $PRUEFds_user)
                        {
                            $DBchange = 'true';
                        } else {
                            $DBchange = 'false';
                            $arrFehlermeldung[] = "Nur ein Projektleiter darf die Parameter seines Projektes oder Parameter, die er selbst angelegt hat,
bearbeiten! Das trifft auf den gewählten Parameter nicht zu!";
                            $in_frmstatus = "EDIT";
                        }
                    }
                }
                if ($DBchange == 'true')
                {
                    if ($in_prjtid > '0') // Der Projektname darf vorerst nicht verändert werden!
                    {
                        # Prüfen, ob Datensatz vorhanden und eindeutig ist!
                        $PREUFqueryUPDATE = "SELECT count(*) "
                                                ."FROM   afn_projekt "
                                                ."WHERE  TID = '$in_prjtid' ";
                        $PRUEFds = $oDBM->GETSelectSQLDatum($PREUFqueryUPDATE);
                        if ($PRUEFds == 1)
                        {
                            $ds = 'true';
                        } else {
                            $ds = 'false';
                            $arrFehlermeldung[] = "Das ausgewählte Projekt ist nicht (mehr) in der Datenbank vorhanden!";
                        }
                    }
                    # Prüfen, ob "parname" (UNIQUE INDEX) eindeutig ist!
                    $PREUFqueryPRJNAME = "SELECT count(*) "
                                                ."FROM   afn_projekt "
                                                ."WHERE  PRJNAME = '$in_prjname' "
                                                ."AND    TID <> '$in_prjtid' ";
                    $PRUEFp = $oDBM->GETSelectSQLDatum($PREUFqueryPRJNAME);
                    if ($PRUEFp == 0)

```

```

{
    $pn = 'true';
} else {
    $pn = 'false';
    $arrFehlermeldung[] = "Jedes Projekt muss einen eindeutigen <b>PROJEKTNAMEN</b> haben - der eingegebene
Projektname wird bereits für ein anderes Projekt verwendet!";
}
// Formatprüfung der eingegebenen Werte
$PRUEFprjkommentar = $oVWM->PRUEFFormat($in_frmprjkommentar,$in_prjkommentar);
if ($ds == 'true' &&
    $pn == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFprjkommentar == 'true')
{
    $queryUPDATE = "UPDATE afn_projekt SET DS_DATE = $DBDateFkt, "
                    ."DS_USER = '$arrCookie[0]', "
                    ."DS_STATUS = 'NEU', "
                    ."PRJKOMMENTAR = '$in_prjkommentar' "
                    ."WHERE TID = '$in_prjtid' ";
    $oDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
    $in_frmstatus = "EDIT";
}
} else {
    if ($in_prjname == '0')
    {
        $arrFehlermeldung[] = "<b>Projektname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parname = "";
    }
    $in_frmstatus = "EDIT";
}
}
}
// INSERT
if ($in_frmstatus == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Nur eine Administrator darf ein neues Projektes anlegen!";
        $in_frmstatus = "NEU";
    }
    if ($DBchange == 'true')
    {
        if ($in_prjname != '0')
        {
            # Prüfen, ob "parname" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryPARNAME = "SELECT count(*) "
                                ."FROM afn_projekt "
                                ."WHERE PRJNAME = '$in_prjname' "
                                ."AND TID <> '$in_prjtid' ";
            $PRUEFparn = $oDBM->GETSelectSQLDatum($PRUEFqueryPARNAME);
            if ($PRUEFparn == 0)
            {
                $pn = 'true';
            } else {
                $pn = 'false';
                $arrFehlermeldung[] = "Jedes Projekt muss einen eindeutigen <b>PROJEKTNAME</b> haben - der eingegebene
Projektname wird bereits für einen anderes Projekt verwendet!";
            }
            // Formatprüfung der eingegebenen Werte
            $PRUEFprjname = $oVWM->PRUEFFormat($in_frmprjname,$in_prjname);
            $PRUEFprjkommentar = $oVWM->PRUEFFormat($in_frmprjkommentar,$in_prjkommentar);

            if ($pn == 'true' &&
                $DBERROR == 'OK' &&
                $PRUEFprjname == 'true' &&
                $PRUEFprjkommentar == 'true')
            {
                $queryINSERT = "INSERT INTO afn_projekt SET DS_DATE = $DBDateFkt, "
                                ."DS_USER = '$arrCookie[0]', "
                                ."DS_STATUS = 'NEU', "
                                ."PRJNAME = '$in_prjname', "

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
. "PRJKOMMENTAR = '$in_prjkommentar' ";
$soDBM->GETUpdateSQLDatum($queryINSERT);
} else {
  if ($PRUEFprjname == 'false')
  {
    $arrFehlermeldung[] = "<b>Projektname:</b> Der Projektname darf nicht länger als 50 Buchstaben sein!";
  }
  if ($PRUEFprjkommentar == 'false')
  {
    $arrFehlermeldung[] = "<b>Projektkommentar:</b> Der Projektkommentar darf nicht länger als 255 Buchstaben
sein!";
  }
  $in_frmstatus = "NEU";
}
} else {
  if ($in_prjname == '0')
  {
    $arrFehlermeldung[] = "<b>Projektname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
    $in_parname = "";
  }
  $in_frmstatus = "NEU";
}
}
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
  $SELECTquery = "SELECT TID, PRJNAME, PRJKOMMENTAR, DS_USER "
    . "FROM afn_projekt "
    . "WHERE TID = $in_prjtid ";
  $soDBM->GETSelectSQL($SELECTquery, 'PRJID, PRJNAME, PRJKOMMENTAR, DS_USER', $arrFormular);

  // HTML-Seite Anfang
  $soLTM->GETFORMHeader();

  // Formular zusammenstellen
  $soVWM->GETFormularHeaderUserverwaltungUPD('Projekt ändern', $file, 'prjtid', $in_prjtid, 'UPDATE');

  if (is_array($arrFormular))
  {
    foreach ($arrFormular as $Daten)
    {
      if ($arrCookie[2] == 'ADMIN')
      {
        $change = 'true';
      }

      if ($change == 'true')
      {
        if (is_array($arrFehlermeldung))
        {
          $soVWM->GETFormRowStyle2('&nbsp;');
          $count = count($arrFehlermeldung);
          for ($i=0; $i<$count; $i++)
          {
            $soVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
          }
          $soVWM->GETFormRowStyle2('&nbsp;');
          $soVWM->GETFormRowStyle3($Daten['PRJNAME']);
          $soVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $in_prjkommentar, 'prjkommentar', 'X(255)', 255);
        } else {
          $soVWM->GETFormRowStyle2('&nbsp;');
          $soVWM->GETFormRowStyle3($Daten['PRJNAME']);
          $soVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $Daten['PRJKOMMENTAR'], 'prjkommentar', 'X(255)', 255);
        }
      }
    }
  }
  $soVWM->GETFormularFooterUserverwaltungUPD();
  // INSERTMODE
} elseif ($in_frmstatus == 'NEU') {
  // HTML-Seite Anfang
  $soLTM->GETFORMHeader();
```



```
// Formular zusammenstellen
$oVWM->GETFormularHeaderUserverwaltungUPD('neues Projekt anlegen',$file,'prjtid',$in_prjtid,'INSERT');

if (is_array($arrFehlermeldung))
{
    $oVWM->GETFormRowStyle2('&nbsp;');
    $count = count($arrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
    }
    $oVWM->GETFormRowStyle2('&nbsp;');
    $oVWM->GETFormRowStyle3('Daten');
    $oVWM->GETFormRowStyleUPDAdmin4('Projektname',$in_prjname,'prjname','X(50)',50);
    $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$in_prjkommentar,'prjkommentar','X(255)',255);
} else {
    $oVWM->GETFormRowStyle2('&nbsp;');
    $oVWM->GETFormRowStyle3('Daten');
    $oVWM->GETFormRowStyleUPDAdmin4('Projektname','','prjname','X(50)',50);
    $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe','','prjkommentar','X(255)',255);
}
$oVWM->GETFormularFooterUserverwaltungUPD();

// READMODE (auch nach UPDATE!)
} else {
    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // GET Filter-Daten
    $GETqueryFILTER = "SELECT TID AS TID, PRJNAME AS NAME "
        ."FROM   afn_projekt "
        ."ORDER BY PRJNAME ";
    $oDBM->GETSelectSQL($GETqueryFILTER,'TID,NAME',$arrFilter);
    $oVWM->GETFormular1Filter('Projekt','fltprj',$arrFilter,$arrHidden);

    // GET DB-Daten
    $query = "SELECT p.TID, p.PRJNAME, p.PRJKOMMENTAR "
        ."FROM   afn_projekt p ";
    // Filterdaten einbinden
    if ($in_fltprj != '0')
    {
        $query = $query."WHERE p.TID = '$in_fltprj' ";
    }
    $query = $query."ORDER BY PRJNAME ";

    $oDBM->GETSelectSQL($query,'PRJID,PRJNAME,PRJKOMMENTAR',$arrFormular);

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltung('Projektverwaltung',$file);
    if (is_array($arrFormular))
    {
        foreach ($arrFormular as $Daten)
        {
            {
                if ($counter > 0)
                {
                    $oVWM->GETFormRowStyle1('&nbsp;');
                }
                $oVWM->GETFormRowStyleUPDATE2($Daten['PRJNAME'],$file,$Daten['PRJID'],'prjtid');
                $oVWM->GETFormRowStyle3('Kommentar/Hilfe');
                $oVWM->GETFormRowStyleTextarea($Daten['PRJKOMMENTAR']);
                $counter++;
            }
        }
        $oVWM->GETFormularFooter();
        // HTML-Seite Ende
        $oLTM->GETFooter();
    }
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihrem Administrator in Verbindung!";
}
} else {
```

```

$ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihren Systemadministrator in
Verbindung!";
}
} else {
$ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle
Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://'.$_SERVER['SERVER_NAME'].'/iFlora'>Startseite</a> oder aktivieren
sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
$oLTM->GETHeader();
$oVWM->GETLabel("iFlora Parameterverwaltung", "H1");
$oVWM->GETTextfield($ERROR);
$oLTM->GETFooter();
}

?>

```

9.3.1.2.5 Controller – „Objektartverwaltung“ (ADMINFormObjektart.php)

```

<?php

// INCLUDE
include_once('..lib/DBManager.php');
include_once('..lib/VIEWManager.php');
include_once('..lib/LAYOUTManager.php');
include_once('..../config_iflora/sqladmin.php.inc');
include_once('..../config_iflora/sqladmin.php.inc');
include_once('..../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht", "ADMIN");
$ERROR = "OK";
$DBERROR = "OK";
$sarrFehlermeldung = NULL;
$SDBDateFkt = 'CURDATE()';
$stable = 'afn_objektart';
$sarrHidden = NULL;

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$sin_frmstatus=$_REQUEST['frmstatus'];} else {$sin_frmstatus = '0';};
// Parametervariablen
if($_REQUEST['obcatid'] != ""){$sin_obcatid=$_REQUEST['obcatid'];} else {$sin_obcatid = 0;};
// Pflicht
if($_REQUEST['obaname'] != ""){$sin_obaname=$_REQUEST['obaname'];} else {$sin_obaname = 0;};
if($_REQUEST['obakuerzel'] != ""){$sin_obakuerzel=$_REQUEST['obakuerzel'];} else {$sin_obakuerzel = 0;};
// optional
if($_REQUEST['obakommentar'] != ""){$sin_obakommentar=$_REQUEST['obakommentar'];} else {$sin_obakommentar = "";};
// Formate
if($_REQUEST['frmobaname'] != ""){$sin_frmobaname=$_REQUEST['frmobaname'];} else {$sin_frmobaname = 0;};
if($_REQUEST['frmobakuerzel'] != ""){$sin_frmobakuerzel=$_REQUEST['frmobakuerzel'];} else {$sin_frmobakuerzel = 0;};
if($_REQUEST['frmobakommentar'] != ""){$sin_frmobakommentar=$_REQUEST['frmobakommentar'];} else {$sin_frmobakommentar =
0;};
// Filter
if($_REQUEST['fltoba'] != ""){$sin_fltoba=$_REQUEST['fltoba'];} else {$sin_fltoba = '0';};
if($_REQUEST['fltobakuerzel'] != ""){$sin_fltobakuerzel=$_REQUEST['fltobakuerzel'];} else {$sin_fltobakuerzel = 0;};

$sarrFormular = NULL;
$sarrFilter = NULL;
$sarrFilter2 = NULL;
$scounter = 0;
$file = "ADMINFormObjektart.php";

$sarrProjekt = NULL;
$sarrWLKlasse = NULL;
$sarrISINArt = NULL;
$sarrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = " ../config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$oVWM = new VIEWManager;

```

```
// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_CookieData($SQLite);
    if ($Cookie <> "ERROR")
    {
        $arrCookie = explode(';', $Cookie);
        $aktiv = $pruefDBM->GETUserAktiv($arrCookie[0]);
        if ($aktiv == "aktiv")
        {
            $PRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht, $arrCookie[2]);
            if ($PRUEFBerechtigung > 0)
            {
                if ($arrCookie[1] == 'ADMIN')
                {
                    $user = $admin_user;
                    $passwd = $admin_passwd;
                } elseif ($arrCookie[1] == 'WORK') {
                    $user = $work_user;
                    $passwd = $work_passwd;
                } else {
                    $user = $data_user;
                    $passwd = $data_passwd;
                }
            }

            // INIT DBManager Objekt
            $oDBM = new DBManager($database, $sql_server, $user, $passwd, $sql_db, $schema);

            // ANWENDUNG
            // 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
            // 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
            // UPDATE
            if ($in_frmstatus == 'UPDATE')
            {
                if ($arrCookie[2] == 'ADMIN')
                {
                    $DBchange = 'true';
                } else {
                    $PREUFqueryDS_USER = "SELECT DS_USER "
                        ."FROM adm_username "
                        ."WHERE TID = '$in_usrtid' ";
                    $PRUEFds_user = $oDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
                    $PRUEFProjekt = $pruefDBM->PRUEFProjTable($arrCookie[0], $in_prjt, $table);
                    if ($PRUEFProjekt > 0 || $arrCookie[0] == $PRUEFds_user)
                    {
                        $DBchange = 'true';
                    } else {
                        $DBchange = 'false';
                        $arrFehlermeldung[] = "Nur ein Projektleiter darf die Objektarten bearbeiten! Das trifft auf den gewählten Parameter nicht zu!";
                    }
                    $in_frmstatus = "EDIT";
                }
            }
            if ($DBchange == 'true')
            {
                if ($in_obatid > '0' && $in_obaname > '0')
                {
                    # Prüfen, ob Datensatz vorhanden und eindeutig ist!
                    $PREUFqueryUPDATE = "SELECT count(*) "
                        ."FROM afn_objektart "
                        ."WHERE TID = '$in_obatid' ";
                    $PRUEFds = $oDBM->GETSelectSQLDatum($PREUFqueryUPDATE);
                    if ($PRUEFds == 1)
                    {
                        $ds = 'true';
                    } else {
                        $ds = 'false';
                        $arrFehlermeldung[] = "Die ausgewählte Objektart ist nicht (mehr) in der Datenbank vorhanden!";
                    }
                    # Prüfen, ob "parname" (UNIQUE INDEX) eindeutig ist!
                    $PREUFqueryOBANAME = "SELECT count(*) "
                        ."FROM afn_objektart "
                        ."WHERE OBANAME = '$in_obaname' "
```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
        ."AND TID <> '$in_obatid' ";
$PRUEFp = $oDBM->GETSelectSQLDatum($PRUEFqueryOBANAME);
if ($PRUEFp == 0)
{
    $pn = 'true';
} else {
    $pn = 'false';
    $arrFehlermeldung[] = "Jede Objektart muss einen eindeutigen <b>OBJEKTARTNAMEN</b> haben - der eingegebene
Objektartname wird bereits für eine andere Objektart verwendet!";
}
// Formatprüfung der eingegebenen Werte
$PRUEFobaname = $oVWM->PRUEFFormat($in_frmobaname,$in_obaname);
$PRUEFobakommentar = $oVWM->PRUEFFormat($in_frmobakommentar,$in_obakommentar);
if ($ds == 'true' &&
    $pn == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFobaname == 'true' &&
    $PRUEFobakommentar == 'true')
{
    $queryUPDATE = "UPDATE afn_objektart SET DS_DATE = $DBDateFkt, "
                    ."DS_USER = '$arrCookie[0]', "
                    ."DS_STATUS = 'NEU', "
                    ."OBANAME = '$in_obaname', "
                    ."OBAKOMMENTAR = '$in_obakommentar' "
                    ."WHERE TID = '$in_obatid' ";
    $oDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
    if ($PRUEFobaname == 'false')
    {
        $arrFehlermeldung[] = "<b>Objektartname:</b> Der Objektartname darf nicht länger als 50 Buchstaben sein!";
    }
    if ($PRUEFobakommentar == 'false')
    {
        $arrFehlermeldung[] = "<b>Objektartkommentar:</b> Der Objektartkommentar darf nicht länger als 255
Buchstaben sein!";
    }
    $in_frmstatus = "EDIT";
}
} else {
    if ($in_obaname == '0')
    {
        $arrFehlermeldung[] = "<b>Objektartname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_obaname = "";
    }
    $in_frmstatus = "EDIT";
}
}
}
// INSERT
if ($in_frmstatus == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Nur eine Administrator darf eine neue Objektart anlegen!";
        $in_frmstatus = "NEU";
    }
    if ($DBchange == 'true')
    {
        if ($in_obaname != '0' && $in_obakuerzel != '0')
        {
            # Prüfen, ob "obaname" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryOBANAME = "SELECT count(*) "
                                ."FROM afn_objektart "
                                ."WHERE OBANAME = '$in_obaname' "
                                ."AND TID <> '$in_obatid' ";
            $PRUEFon = $oDBM->GETSelectSQLDatum($PRUEFqueryOBANAME);
            if ($PRUEFon == 0)
            {
                $on = 'true';
            } else {
                $on = 'false';
            }
        }
    }
}
```

```

    $arrFehlermeldung[] = "Jede Objektart muss einen eindeutigen <b>OBJEKTARTNAME</b> haben - der eingegebene
Objektartname wird bereits für eine andere Objektart verwendet!";
}
# Prüfen, ob "obakuerzel" (UNIQUE INDEX) eindeutig ist!
$PRUEFqueryOBAKUERZEL = "SELECT count(*) "
                        ."FROM afn_objektart "
                        ."WHERE OBANAME = '$sin_obakuerzel' "
                        ."AND TID <> '$sin_obatid' ";
$PRUEFok = $oDBM->GETSelectSQLDatum($PRUEFqueryOBAKUERZEL);
if ($PRUEFok == 0)
{
    $ok = 'true';
} else {
    $ok = 'false';
    $arrFehlermeldung[] = "Jede Objektart muss ein eindeutiges <b>OBJEKTARTKUERZEL</b> haben - das eingegebene
Kürzel wird bereits für eine andere Objektart verwendet!";
}
// Formatprüfung der eingegebenen Werte
$PRUEFobaname = $oVWM->PRUEFFormat($sin_frmobaname,$sin_obaname);
$PRUEFobakuerzel = $oVWM->PRUEFFormat($sin_frmobakuerzel,$sin_obakuerzel);
$PRUEFobakommentar = $oVWM->PRUEFFormat($sin_frmobakommentar,$sin_obakommentar);

if ($son == 'true' &&
    $ok == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFobaname == 'true' &&
    $PRUEFobakommentar == 'true')
{
    $queryINSERT = "INSERT INTO afn_objektart SET DS_DATE = $DBDateFkt, "
                  ."DS_USER = '$arrCookie[0]', "
                  ."DS_STATUS = 'NEU', "
                  ."OBANAME = '$sin_obaname', "
                  ."OBAKUERZEL = '$sin_obakuerzel', "
                  ."OBAKOMMENTAR = '$sin_obakommentar' ";

    $oDBM->GETUpdateSQLDatum($queryINSERT);
} else {
    if ($PRUEFobaname == 'false')
    {
        $arrFehlermeldung[] = "<b>Objektartname:</b> Der Objektartname darf nicht länger als 50 Buchstaben sein!";
    }
    if ($PRUEFobakuerzel == 'false')
    {
        $arrFehlermeldung[] = "<b>Objektartkürzel:</b> Das Objektartkürzel darf nicht länger als 10 Buchstaben sein!";
    }
    if ($PRUEFobakommentar == 'false')
    {
        $arrFehlermeldung[] = "<b>Objektartkommentar:</b> Der Objektartkommentar darf nicht länger als 255
Buchstaben sein!";
    }
    $sin_frmstatus = "NEU";
}
} else {
    if ($sin_obaname == '0')
    {
        $arrFehlermeldung[] = "<b>Objektartname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $sin_obaname = "";
    }
    if ($sin_obakuerzel == '0')
    {
        $arrFehlermeldung[] = "<b>Objektartkürzel:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $sin_obakuerzel = "";
    }
}

$sin_frmstatus = "NEU";
}
}
// EDITMODE
if ($sin_frmstatus == 'EDIT')
{
    $SELECTquery = "SELECT TID, OBANAME, OBAKUERZEL, OBAKOMMENTAR, DS_USER "
                  ."FROM afn_objektart "
                  ."WHERE TID = $sin_obatid ";

    $oDBM-
>GETSelectSQL($SELECTquery,'OBAID,OBANAME,OBAKUERZEL,OBAKOMMENTAR,DS_USER',$arrFormular);

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
// HTML-Seite Anfang
$oSMT->GETFORMHeader();

// Formular zusammenstellen
$oSMT->GETFormularHeaderUserverwaltungUPD('Objektart ändern',$file,'obatid',$sin_obatid,'UPDATE');

if (is_array($arrFormular))
{
    foreach ($arrFormular as $Daten)
    {
        if ($arrCookie[2] == 'ADMIN')
        {
            $change = 'true';
        }

        if ($change == 'true')
        {
            if (is_array($arrFehlermeldung))
            {
                $oSMT->GETFormRowStyle2('&nbsp;');
                $count = count($arrFehlermeldung);
                for ($i=0; $i<$count; $i++)
                {
                    $oSMT->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
                }
                $oSMT->GETFormRowStyle2('&nbsp;');
                $oSMT->GETFormRowStyle3($Daten['OBJAKUERZEL']);
                $oSMT->GETFormRowStyleUPDAdmin4('Objektartname',$sin_obaname,'obaname','X(50)',50);
                $oSMT->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$sin_obakommentar,'obakommentar','X(255)',255);
            } else {
                $oSMT->GETFormRowStyle2('&nbsp;');
                $oSMT->GETFormRowStyle3($Daten['OBJAKUERZEL']);
                $oSMT->GETFormRowStyleUPDAdmin4('Objektartname',$Daten['OBANAME'],'obaname','X(50)',50);
                $oSMT->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$Daten['OBAKOMMENTAR'],'obakommentar','X(255)',255);
            }
        }
    }
}
$oSMT->GETFormularFooterUserverwaltungUPD();
// INSERTMODE
} elseif ($sin_frmstatus == 'NEU'){
// HTML-Seite Anfang
$oSMT->GETFORMHeader();

// Formular zusammenstellen
$oSMT->GETFormularHeaderUserverwaltungUPD('neue Objektart anlegen',$file,'obatid',$sin_obatid,'INSERT');

if (is_array($arrFehlermeldung))
{
    $oSMT->GETFormRowStyle2('&nbsp;');
    $count = count($arrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $oSMT->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
    }
    $oSMT->GETFormRowStyle2('&nbsp;');
    $oSMT->GETFormRowStyle3($Daten);
    $oSMT->GETFormRowStyleUPDAdmin4('Objektartname',$sin_obaname,'obaname','X(50)',50);
    $oSMT->GETFormRowStyleUPDAdmin4('Objektartkürzel',$sin_obakuerzel,'obakuerzel','X(10)',10);
    $oSMT->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$sin_obakommentar,'obakommentar','X(255)',255);
} else {
    $oSMT->GETFormRowStyle2('&nbsp;');
    $oSMT->GETFormRowStyle3($Daten);
    $oSMT->GETFormRowStyleUPDAdmin4('Objektartname','','obaname','X(50)',50);
    $oSMT->GETFormRowStyleUPDAdmin4('Objektartkürzel','','obakuerzel','X(10)',10);
    $oSMT->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe','','obakommentar','X(255)',255);
}
$oSMT->GETFormularFooterUserverwaltungUPD();

// READMODE (auch nach UPDATE!)
} else {
// HTML-Seite Anfang
$oSMT->GETFORMHeader();
```

```

// GET Filter-Daten
$GETqueryFILTER = "SELECT TID, OBANAME "
                  ."FROM   afn_objektart "
                  ."ORDER BY OBANAME ";
$soDBM->GETSelectSQL($GETqueryFILTER,'TID,NAME',$sarrFilter);
// GET Filter-Daten
$GETqueryFILTER2 = "SELECT DISTINCT p.TID, p.OBAKUERZEL "
                  ."FROM   afn_objektart p "
                  ."ORDER BY p.OBAKUERZEL ";
$soDBM->GETSelectSQL($GETqueryFILTER2,'TID,NAME',$sarrFilter2);
$soVWM->GETFormular2Filter('Objektart','fltoba',$sarrFilter,'Objektartkürzel','fltobakuerzel',$sarrFilter2);

// GET DB-Daten
$query = "SELECT p.TID, concat(p.OBANAME, ' ('.p.OBAKUERZEL,')', p.OBAKOMMENTAR "
        ."FROM   afn_objektart p ";
// Filterdaten einbinden
if ($sin_fltoba != '0')
{
    $query = $query."WHERE p.TID = '$sin_fltoba' ";
} elseif ($sin_fltobakuerzel != '0') {
    $query = $query."WHERE p.TID = '$sin_fltobakuerzel' ";
}
$query = $query."ORDER BY p.OBANAME ";

$soDBM->GETSelectSQL($query,'OBAID,OBANAME,OBAKOMMENTAR',$sarrFormular);

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltung('Objektartverwaltung',$file);
if (is_array($sarrFormular))
{
    foreach ($sarrFormular as $Daten)
    {
        {
            if ($counter > 0)
            {
                $soVWM->GETFormRowStyle1('&nbsp;');
            }
            $soVWM->GETFormRowStyleUPDATE2($Daten['OBANAME'],$file,$Daten['OBAID'],'obatud');
            $soVWM->GETFormRowStyle3('Kommentar/Hilfe');
            $soVWM->GETFormRowStyleTextarea($Daten['OBAKOMMENTAR']);
            $counter++;
        }
    }
    $soVWM->GETFormularFooter();
    // HTML-Seite Ende
    $soLTM->GETFooter();
}
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihren Administrator in Verbindung!";
}
} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihren Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://'. $_SERVER['SERVER_NAME']. '/iFlora'>Startseite</a> oder aktivieren sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
    $soLTM->GETHeader();
    $soVWM->GETLabel("iFlora Parameterverwaltung", "H1");
    $soVWM->GETTextfield($ERROR);
    $soLTM->GETFooter();
}
?>

```

9.3.1.2.6 Controller – „IS-IN-Artverwaltung“ (ADMINFormISINArt.php)

```
<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../././config_iflora/sqladmin.php.inc');
include_once('../././config_iflora/sqladmin.php.inc');
include_once('../././config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht","ADMIN");
$ERROR = "OK";
$DBERROR = "OK";
$SarrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$stable = 'afn_isinart';
$SarrHidden = NULL;

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$_frmstatus=$_REQUEST['frmstatus'];} else {$_frmstatus = '0';};
// Parametervariablen
if($_REQUEST['isatid'] != ""){$_isatid=$_REQUEST['isatid'];} else {$_isatid = 0;};
// Pflicht
if($_REQUEST['isaname'] != ""){$_isaname=$_REQUEST['isaname'];} else {$_isaname = 0;};
if($_REQUEST['isakuerzel'] != ""){$_isakuerzel=$_REQUEST['isakuerzel'];} else {$_isakuerzel = 0;};
// optional
if($_REQUEST['isakommentar'] != ""){$_isakommentar=$_REQUEST['isakommentar'];} else {$_isakommentar = ""};
// Formate
if($_REQUEST['frmisaname'] != ""){$_frmisaname=$_REQUEST['frmisaname'];} else {$_frmisaname = 0;};
if($_REQUEST['frmisakuerzel'] != ""){$_frmisakuerzel=$_REQUEST['frmisakuerzel'];} else {$_frmisakuerzel = 0;};
if($_REQUEST['frmisakommentar'] != ""){$_frmisakommentar=$_REQUEST['frmisakommentar'];} else {$_frmisakommentar = 0;};
// Filter
if($_REQUEST['fltisa'] != ""){$_fltisa=$_REQUEST['fltisa'];} else {$_fltisa = '0';};
if($_REQUEST['fltisakuerzel'] != ""){$_fltisakuerzel=$_REQUEST['fltisakuerzel'];} else {$_fltisakuerzel = 0;};

$SarrFormular = NULL;
$SarrFilter = NULL;
$SarrFilter2 = NULL;
$counter = 0;
$file = "../ADMINFormISINArt.php";

$SarrProjekt = NULL;
$SarrWLKlasse = NULL;
$SarrISINArt = NULL;
$SarrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = "../././config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$SoVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$SoLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $SarrCookie = explode(';', $Cookie);
        $Saktiv = $pruefDBM->GETUserAktiv($SarrCookie[0]);
        if ($Saktiv == "aktiv")
        {
            $SPRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht,$SarrCookie[2]);
            if ($SPRUEFBerechtigung > 0)
            {
                if ($SarrCookie[1] == 'ADMIN')
                {
                    $Suser = $admin_user;
                    $Spasswd = $admin_passwd;
                } elseif ($SarrCookie[1] == 'WORK') {
```



```

    $user = $work_user;
    $passwd = $work_passwd;
} else {
    $user = $data_user;
    $passwd = $data_passwd;
}

// INIT DBManager Objekt
$soDBM = new DBManager($database,$sql_server,$user,$passwd,$sql_db,$schema);

// ANWENDUNG
// 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
// 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
// UPDATE
if ($in_frmstatus == 'UPDATE')
{
    if ($sarrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $PREUFqueryDS_USER = "SELECT DS_USER "
                                ."FROM   adm_username "
                                ."WHERE  TID = '$in_usrtid' ";
        $PREUFds_user = $soDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
        $PREUFProjekt = $pruefDBM->PRUEFProjTable($sarrCookie[0],$in_prjtid,$stable);
        if ($PREUFProjekt > 0 || $sarrCookie[0] == $PREUFds_user)
        {
            $DBchange = 'true';
        } else {
            $DBchange = 'false';
            $sarrFehlermeldung[] = "Nur ein Projektleiter darf die IS-IN-Arten bearbeiten! Das trifft auf die gewählte IS-IN-Art nicht zu!";
            $in_frmstatus = "EDIT";
        }
    }
}
if ($DBchange == 'true')
{
    if ($in_isatid > '0' && $in_isaname > '0')
    {
        # Prüfen, ob Datensatz vorhanden und eindeutig ist!
        $PREUFqueryUPDATE = "SELECT count(*) "
                                ."FROM   afn_isinart "
                                ."WHERE  TID = '$in_isatid' ";
        $PREUFds = $soDBM->GETSelectSQLDatum($PREUFqueryUPDATE);
        if ($PREUFds == 1)
        {
            $ds = 'true';
        } else {
            $ds = 'false';
            $sarrFehlermeldung[] = "Die ausgewählte IS-IN-Art ist nicht (mehr) in der Datenbank vorhanden!";
        }
        # Prüfen, ob "parname" (UNIQUE INDEX) eindeutig ist!
        $PREUFqueryISANAME = "SELECT count(*) "
                                ."FROM   afn_isinart "
                                ."WHERE  OBANAME = '$in_isaname' "
                                ."AND    TID <> '$in_isatid' ";
        $PREUFp = $soDBM->GETSelectSQLDatum($PREUFqueryISANAME);
        if ($PREUFp == 0)
        {
            $pn = 'true';
        } else {
            $pn = 'false';
            $sarrFehlermeldung[] = "Jede IS-IN-Art muss einen eindeutigen <b>IS-IN-ARTNAMEN</b> haben - der eingegebene IS-IN-
Artname wird bereits für eine andere IS-IN-Art verwendet!";
        }
        // Formatprüfung der eingegebenen Werte
        $PREUFisaname = $soVWM->PRUEFFormat($in_frmisaname,$in_isaname);
        $PREUFisakommentar = $soVWM->PRUEFFormat($in_frmisakommentar,$in_isakommentar);
        if ($ds == 'true' &&
            $pn == 'true' &&
            $DBERROR == 'OK' &&
            $PREUFisaname == 'true' &&
            $PREUFisakommentar == 'true')
        {
            $queryUPDATE = "UPDATE afn_isinart SET DS_DATE = $DBDateFkt, "
                            ."DS_USER = '$sarrCookie[0]', "

```

```

        ."DS_STATUS = 'NEU', "
        ."ISANAME = '$in_isaname', "
        ."ISAKOMMENTAR = '$in_isakommentar' "
        ."WHERE TID = '$in_isatid' ";
        $oDBM->GETUpdateSQLDatum($queryUPDATE);
    } else {
        if ($PRUEFisaname == 'false')
        {
            $arrFehlermeldung[] = "<b>IS-IN-Artname:</b> Der IS-IN-Artname darf nicht länger als 50 Buchstaben sein!";
        }
        if ($PRUEFisakommentar == 'false')
        {
            $arrFehlermeldung[] = "<b>IS-IN-Artkommentar:</b> Der IS-IN-Artkommentar darf nicht länger als 255 Buchstaben
sein!";
        }

        $in_frmstatus = "EDIT";
    }
} else {
    if ($in_isaname == '0')
    {
        $arrFehlermeldung[] = "<b>IS-IN-Artname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_isaname = "";
    }
    $in_frmstatus = "EDIT";
}
}
}
// INSERT
if ($in_frmstatus == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Nur eine Administrator darf eine neue IS-IN-Art angelegen!";
        $in_frmstatus = "NEU";
    }
    if ($DBchange == 'true')
    {
        if ($in_obaname != '0' && $in_obakuerzel != '0')
        {
            # Prüfen, ob "isaname" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryISANAME = "SELECT count(*) "
                ."FROM afn_isinart
                ."WHERE ISANAME = '$in_isaname' "
                ."AND TID <> '$in_isatid' ";
            $PRUEFon = $oDBM->GETSelectSQLDatum($PRUEFqueryISANAME);
            if ($PRUEFon == 0)
            {
                $on = 'true';
            } else {
                $on = 'false';
                $arrFehlermeldung[] = "Jede IS-IN-Art muss einen eindeutigen <b>IS-IN-ARTNAMEN</b> haben - der eingegebene IS-IN-
Artname wird bereits für eine andere IS-IN-Art verwendet!";
            }
            # Prüfen, ob "obakuerzel" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryISAKUERZEL = "SELECT count(*) "
                ."FROM afn_isinart "
                ."WHERE OSANAME = '$in_isakuerzel' "
                ."AND TID <> '$in_isatid' ";
            $PRUEFok = $oDBM->GETSelectSQLDatum($PRUEFqueryISAKUERZEL);
            if ($PRUEFok == 0)
            {
                $ok = 'true';
            } else {
                $ok = 'false';
                $arrFehlermeldung[] = "Jede IS-IN-Art muss ein eindeutiges <b>IS-IN-ARTKUERZEL</b> haben - das eingegebene Kürzel
wird bereits für eine andere IS-IN-Art verwendet!";
            }
            // Formatprüfung der eingegebenen Werte
            $PRUEFisaname = $oVWM->PRUEFFormat($in_frmisaname,$in_isaname);
            $PRUEFisakuerzel = $oVWM->PRUEFFormat($in_frmisakuerzel,$in_isakuerzel);
            $PRUEFisakommentar = $oVWM->PRUEFFormat($in_frmisakommentar,$in_isakommentar);

```

```

if ($on == 'true' &&
    $ok == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFisaname == 'true' &&
    $PRUEFisakuerzel == 'true' &&
    $PRUEFisakommentar == 'true')
{
    $queryINSERT = "INSERT INTO afn_isinart SET DS_DATE = $DBDateFkt, "
                    . "DS_USER = '$sarrCookie[0]', "
                    . "DS_STATUS = 'NEU', "
                    . "ISANAME = '$in_isaname', "
                    . "ISAKUERZEL = '$in_isakuerzel', "
                    . "ISAKOMMENTAR = '$in_isakommentar' ";

    $oDBM->GETUpdateSQLDatum($queryINSERT);
} else {
    if ($PRUEFisaname == 'false')
    {
        $sarrFehlermeldung[] = "<b>IS-IN-Artname:</b> Der IS-IN-Artname darf nicht länger als 50 Buchstaben sein!";
    }
    if ($PRUEFisakuerzel == 'false')
    {
        $sarrFehlermeldung[] = "<b>IS-IN-Artkürzel:</b> Das IS-IN-Artkürzel darf nicht länger als 10 Buchstaben sein!";
    }
    if ($PRUEFisakommentar == 'false')
    {
        $sarrFehlermeldung[] = "<b>IS-IN-Artkommentar:</b> Der IS-IN-Artkommentar darf nicht länger als 255 Buchstaben
sein!";
    }
    $in_frmstatus = "NEU";
}
} else {
    if ($in_isaname == '0')
    {
        $sarrFehlermeldung[] = "<b>IS-IN-Artname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_isaname = "";
    }
    if ($in_isakuerzel == '0')
    {
        $sarrFehlermeldung[] = "<b>IS-IN-Artkürzel:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_isakuerzel = "";
    }
}

$in_frmstatus = "NEU";
}
}
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
    $SELECTquery = "SELECT TID, ISANAME, ISAKUERZEL, ISAKOMMENTAR, DS_USER "
                  . "FROM afn_isinart "
                  . "WHERE TID = $in_isatid ";
    $oDBM->GETSelectSQL($SELECTquery,'ISAID,ISANAME,ISAKUERZEL,ISAKOMMENTAR,DS_USER',$sarrFormular);

    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltungUPD('IS-IN-Art ändern',$file,'isatid',$in_isatid,'UPDATE');

    if (is_array($sarrFormular))
    {
        foreach ($sarrFormular as $Daten)
        {
            if ($sarrCookie[2] == 'ADMIN')
            {
                $change = 'true';
            }
        }

        if ($change == 'true')
        {
            if (is_array($sarrFehlermeldung))
            {

```

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

Stephan Hordynski (Matrikelnr.: a9227169)

```

    } elseif ($in_fltisakuerzel != '0') {
        $query = $query."WHERE p.TID = '$in_fltisakuerzel' ";
    }
    $query = $query."ORDER BY p.ISANAME ";

    $oDBM->GETSelectSQL($query,'ISaid,ISANAME,ISAKOMMENTAR',$arrFormular);

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltung('IS-IN-Artverwaltung',$file);
    if (is_array($arrFormular))
    {
        foreach ($arrFormular as $Daten)
        {
            {
                if ($counter > 0)
                {
                    $oVWM->GETFormRowStyle1('&nbsp;');
                }
                $oVWM->GETFormRowStyleUPDATE2($Daten['ISANAME'],$file,$Daten['ISaid'],'isaid');
                $oVWM->GETFormRowStyle3('Kommentar/Hilfe');
                $oVWM->GETFormRowStyleTextarea($Daten['ISAKOMMENTAR']);
                $counter++;
            }
        }
        $oVWM->GETFormularFooter();
        // HTML-Seite Ende
        $oLTM->GETFooter();
    }
    } else {
        $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
    }
    } else {
        $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihrem Administrator in Verbindung!";
    }
    } else {
        $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihrem Systemadministrator in Verbindung!";
    }
    } else {
        $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle
        Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://".$_SERVER['SERVER_NAME']."'>iFlora</a>Startseite</a> oder aktivieren
        sie die Cookie-Funktion bei Ihrem Internetbrowser!";
    }
}

If($ERROR <> OK) // Fehlermeldung
{
    $oLTM->GETHeader();
    $oVWM->GETLabel("iFlora Parameterverwaltung","H1");
    $oVWM->GETTextfield($ERROR);
    $oLTM->GETFooter();
}

?>

```

9.3.1.2.7 **Controller – „IS-IN-Objektartverwaltung“ (ADMINFormISINOBA.php)**

```

<?php

// BspAbfrage
//select  isinoba.obaname,isa.isaname,oba.obaname
//from    afn_isinArt isa,
//         afn_objektart oba,
//         afn_isinobjart isinobabez,
//         afn_objektart isinoba
//where    isa.tid = isinobabez.isaid => IS-IN THEMA ("PFAD")
//and      oba.tid = isinobabez.obaid => Objektart Unterebene
//and      isinoba.tid = isinobabez.isinobaid => Objektart Übergeordnete Ebene

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../../config_iflora/sqladmin.php.inc');
include_once('../../config_iflora/sqladmin.php.inc');
include_once('../../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung

```

```

define("formularrecht","ADMIN");
$ERROR = "OK";
$DBERROR = "OK";
$sarrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$stable = 'afn_objektart';
$sarrHidden = NULL;

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$in_frmstatus=$_REQUEST['frmstatus'];} else {$in_frmstatus = '0';};
// TID
if($_REQUEST['obtid'] != ""){$in_obtid=$_REQUEST['obtid'];} else {$in_obtid = 0;};
if($_REQUEST['isinobtid'] != ""){$in_isinobtid=$_REQUEST['isinobtid'];} else {$in_isinobtid = 0;};
// Pflicht
if($_REQUEST['obaname'] != ""){$in_obaname=$_REQUEST['obaname'];} else {$in_obaname = 0;};
if($_REQUEST['isaname'] != ""){$in_isaname=$_REQUEST['isaname'];} else {$in_isaname = 0;};
if($_REQUEST['isinobaname'] != ""){$in_isinobaname=$_REQUEST['isinobaname'];} else {$in_isinobaname = 0;};
// optional
if($_REQUEST['isinobakommentar'] != ""){$in_isinobakommentar=$_REQUEST['isinobakommentar'];} else {$in_isinobakommentar =
";};
// Formate
if($_REQUEST['frmisinobakommentar'] != ""){$in_frmisinobakommentar=$_REQUEST['frmisinobakommentar'];} else
{$in_frmisinobakommentar = 0;};
// Filter
if($_REQUEST['fltobaid'] != ""){$in_fltobaid=$_REQUEST['fltobaid'];} else {$in_fltobaid = '0';};
if($_REQUEST['fltisinobaid'] != ""){$in_fltisinobaid=$_REQUEST['fltisinobaid'];} else {$in_fltisinobaid = '0';};
if($_REQUEST['fltisinartid'] != ""){$in_fltisinartid=$_REQUEST['fltisinartid'];} else {$in_fltisinartid = 0;};

$sarrFormular = NULL;
$sarrFilter1 = NULL;
$sarrFilter2 = NULL;
$sarrFilter3 = NULL;
$scounter = 0;
$file = "../ADMINFormISINOBA.php";

$sarrProjekt = NULL;
$sarrWLKlasse = NULL;
$sarrISINArt = NULL;
$sarrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = ".././../config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$soVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$soLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $sarrCookie = explode(';', $Cookie);
        $aktiv = $pruefDBM->GETUserAktiv($sarrCookie[0]);
        if ($aktiv == "aktiv")
        {
            $PRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht,$sarrCookie[2]);
            if ($PRUEFBerechtigung > 0)
            {
                if ($sarrCookie[1] == 'ADMIN')
                {
                    $user = $admin_user;
                    $passwd = $admin_passwd;
                } elseif ($sarrCookie[1] == 'WORK') {
                    $user = $work_user;
                    $passwd = $work_passwd;
                } else {
                    $user = $data_user;
                    $passwd = $data_passwd;
                }
            }
        }

        // INIT DBManager Objekt
        $soDBM = new DBManager($database,$sql_server,$user,$passwd,$sql_db,$schema);
    }
}

```

```
// INIT - Objektartnamen holen
$queryObjektart = "SELECT OBANAME "
                ."FROM afn_objektart ";
$sarrObjektart = $oDBM->GETSelectSQLSelect($queryObjektart);

// ANWENDUNG
// 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
// 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
// UPDATE
if ($in_frmstatus == 'UPDATE')
{
    if ($sarrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $PREUFqueryDS_USER = "SELECT DS_USER "
                            ."FROM adm_username "
                            ."WHERE TID = '$in_usrtid' ";
        $PRUEFds_user = $oDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
        $PRUEFProjekt = $pruefDBM->PRUEFProjTable($sarrCookie[0], $in_prjtid, $stable);
        if ($PRUEFProjekt > 0 || $sarrCookie[0] == $PRUEFds_user)
        {
            $DBchange = 'true';
        } else {
            $DBchange = 'false';
            $sarrFehlermeldung[] = "Nur ein Projektleiter darf die IS-IN Objektartbeziehungen bearbeiten!";
            $in_frmstatus = "EDIT";
        }
    }
}
if ($DBchange == 'true')
{
    if ($in_isinobatid > '0' && $in_obaname > '0' && $in_isinobaname > '0')
    {
        # Prüfen, ob Datensatz vorhanden und eindeutig ist!
        $PRUEFqueryUPDATE = "SELECT count(*) "
                            ."FROM afn_isinobjart "
                            ."WHERE TID = '$in_isinobatid' ";
        $PRUEFds = $oDBM->GETSelectSQLDatum($PRUEFqueryUPDATE);
        if ($PRUEFds == 1)
        {
            $ds = 'true';
        } else {
            $ds = 'false';
            $sarrFehlermeldung[] = "Die ausgewählte Objektart ist nicht (mehr) in der Datenbank vorhanden!";
        }
        # Prüfen, ob "IS-IN Objektartbeziehung" (UNIQUE INDEX) eindeutig ist!
        # d.h.: OBAID&ISINOBAID&ISAID darf noch nicht vorkommen (ausser beim geänderten Datensatz)
        $PRUEFqueryISINOBABEZ = "SELECT count(*) "
                                ."FROM afn_isinobjart "
                                ."WHERE OBAID = (SELECT TID "
                                ."      FROM afn_objektart "
                                ."      WHERE OBANAME = '$in_obaname') "
                                ."AND ISINOBAID = (SELECT TID "
                                ."      FROM afn_objektart "
                                ."      WHERE OBANAME = '$in_isinobaname') "
                                ."AND ISAID = (SELECT ISAID "
                                ."      FROM afn_isinobjart "
                                ."      WHERE TID = '$in_isinobaname') "
                                ."AND TID <> '$in_isinobatid' ";
        $PRUEFp = $oDBM->GETSelectSQLDatum($PRUEFqueryISINOBABEZ);
        if ($PRUEFp == 0)
        {
            $pn = 'true';
        } else {
            $pn = 'false';
            $sarrFehlermeldung[] = "Jede IS-IN Objektartbeziehung muss einen eindeutigen sein - die eingegebene IS-IN
Objektartbeziehung gib es bereits!";
        }
    }
    // Formatprüfung der eingegebenen Werte
    $PRUEFisinobakommentar = $oVWM->PRUEFFormat($in_frmisinobakommentar, $in_isinobakommentar);
    if ($ds == 'true' &&
        $pn == 'true' &&
        $DBERROR == 'OK' &&
        $PRUEFisinobakommentar == 'true')
    {

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
{
  $queryUPDATE = "UPDATE afn_isinobjart SET DS_DATE = $DBDateFkt, "
                  ."DS_USER = '$arrCookie[0]', "
                  ."DS_STATUS = 'NEU', "
                  ."OBAID = (SELECT TID "
                  ."      FROM afn_objektart "
                  ."      WHERE OBANAME = '$in_obaname'), "
                  ."ISINOBAID = (SELECT TID "
                  ."      FROM afn_objektart "
                  ."      WHERE OBANAME = '$in_isinobaname'), "
                  ."ISINOBAKOMMENTAR = '$in_isinobakommentar' "
                  ."WHERE TID = '$in_isinobatid' ";
  $oDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
  if ($PRUEFobakommentar == 'false')
  {
    $arrFehlermeldung[] = "<b>IS-IN Objektartkommentar:</b> Der IS-IN Objektartkommentar darf nicht länger als 255
Buchstaben sein!";
  }

  $in_frmstatus = "EDIT";
}
} else {
  if ($in_isinobaname == '0')
  {
    $arrFehlermeldung[] = "<b>IS-IN Objektart(Ebene 1):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
  }
  if ($in_isinobatid == '0')
  {
    $arrFehlermeldung[] = "Es wurde keine IS-IN Objektartbeziehung ausgewählt!";
  }
  if ($in_obaname == '0')
  {
    $arrFehlermeldung[] = "<b>Objektart(Ebene 2):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
  }
  $in_frmstatus = "EDIT";
}
}
}
// INSERT
if ($in_frmstatus == 'INSERT')
{
  if ($arrCookie[2] == 'ADMIN')
  {
    $DBchange = 'true';
  } else {
    $DBchange = 'false';
    $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Nur eine Administrator darf eine neue Objektart anlegen!";
    $in_frmstatus = "NEU";
  }
}
if ($DBchange == 'true')
{
  if ($in_obaname != '0' && $in_obakuerzel != '0')
  {
    # Prüfen, ob "IS-IN Objektartbeziehung" (UNIQUE INDEX) eindeutig ist!
    # d.h.: OBAID&ISINOBAID&ISAID darf noch nicht vorkommen (ausser beim geänderten Datensatz)
    $PRUEFqueryISINOBAID = "SELECT count(*) "
                          ."FROM afn_isinobjart "
                          ."WHERE OBAID = (SELECT TID "
                          ."      FROM afn_objektart "
                          ."      WHERE OBANAME = '$in_obaname') "
                          ."AND ISINOBAID = (SELECT TID "
                          ."      FROM afn_objektart "
                          ."      WHERE OBANAME = '$in_isinobaname') "
                          ."AND ISAID = (SELECT ISAID "
                          ."      FROM afn_isinobjart "
                          ."      WHERE TID = '$in_isinobaname') "
                          ."AND TID <> '$in_isinobatid' ";
    # Prüfen, ob "obaname" (UNIQUE INDEX) eindeutig ist!
    $PRUEFon = $oDBM->GETSelectSQLDatum($PRUEFqueryISINOBAID);
    if ($PRUEFon == 0)
    {
      $on = 'true';
    } else {
      $on = 'false';
    }
  }
}
```



```

    $arrFehlermeldung[] = "Jede IS-IN Objektartbeziehung muss einen eindeutigen sein - die eingegebene IS-IN
Objektartbeziehung gib es bereits!";
}
// Formatprüfung der eingegebenen Werte
$PRUEFisinobakommentar = $oVWM->PRUEFFormat($in_frmisinobakommentar,$in_isinobakommentar);

if ($on == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFisinobakommentar == 'true')
{
    $queryINSERT = "INSERT INTO afn_isinobjart SET DS_DATE = $DBDateFkt, "
                    ."DS_USER = '$arrCookie[0]', "
                    ."DS_STATUS = 'NEU', "
                    ."ISAID = (SELECT TID "
                    ."    FROM afn_isinart "
                    ."    WHERE ISANAME = '$in_isaname'), "
                    ."OBAID = (SELECT TID "
                    ."    FROM afn_objektart "
                    ."    WHERE OBANAME = '$in_obaname'), "
                    ."ISINOBAID = (SELECT TID "
                    ."    FROM afn_objektart "
                    ."    WHERE OBANAME = '$in_isinobaname'), "
                    ."ISINOBAKOMMENTAR = '$in_isinobakommentar' ";

    $oDBM->GETUpdateSQLDatum($queryINSERT);
} else {
    if ($PRUEFisinobakommentar == 'false')
    {
        $arrFehlermeldung[] = "<b>IS-IN Objektartkommentar:</b> Der IS-IN Objektartkommentar darf nicht länger als 255
Buchstaben sein!";
    }
    $in_frmstatus = "NEU";
}
} else {
    if ($in_isinobaname == '0')
    {
        $arrFehlermeldung[] = "<b>IS-IN Objektart(Ebene 1):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
    }
    if ($in_isinobaid == '0')
    {
        $arrFehlermeldung[] = "Es wurde keine IS-IN Objektartbeziehung ausgewählt!";
    }
    if ($in_obaname == '0')
    {
        $arrFehlermeldung[] = "<b>Objektart(Ebene 2):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
    }
    $in_frmstatus = "NEU";
}
}
}
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
    $SELECTquery = "SELECT isoa.TID, isa.ISANAME, oba1.OBANAME, "
                    ."    oba2.OBANAME AS ISINOBANAME, isoa.ISINOBAKOMMENTAR, isoa.DS_USER "
                    ."FROM afn_isinart isa, "
                    ."    afn_isinobjart isoa, "
                    ."    afn_objektart oba1, "
                    ."    afn_objektart oba2 "
                    ."WHERE isoa.TID = $in_isinobaid "
                    ."AND isa.TID = isoa.ISAID "
                    ."AND oba1.TID = isoa.OBAID "
                    ."AND oba2.TID = isoa.ISINOBAID ";

    $oDBM->GETSelectSQL($SELECTquery,'ISINOBAID,ISANAME,OBANAME,ISINOBANAME,ISINOBAKOMMENTAR,DS_USER',$arrFormular);

    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltungUPD('IS-IN-Objektart ändern',$file,'isinobaid',$in_isinobaid,'UPDATE');

    if (is_array($arrFormular))
    {
        foreach ($arrFormular as $Daten)

```

Diplomarbeit

```

    {
        if ($arrCookie[2] == 'ADMIN')
        {
            $change = 'true';
        }

        if ($change == 'true')
        {
            if (is_array($arrFehlermeldung))
            {
                $oVWM->GETFormRowStyle2('&nbsp;');
                $count = count($arrFehlermeldung);
                for ($i=0; $i<$count; $i++)
                {
                    $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
                }
                $oVWM->GETFormRowStyle2('&nbsp;');
                $oVWM->GETFormRowStyle3($Daten['ISANAME']);
                $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)', $sin_isinobaname, 'isinobaname', $arrObjektart);
                $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)', $sin_obaname, 'obaname', $arrObjektart);
                $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $sin_isinobakommentar, 'isinobakommentar', 'X(255)', 255);
            } else {
                $oVWM->GETFormRowStyle2('&nbsp;');
                $oVWM->GETFormRowStyle3($Daten['ISANAME']);
                $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)', $sin_isinobaname, 'isinobaname', $arrObjektart);
                $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)', $sin_obaname, 'obaname', $arrObjektart);
                $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $Daten['ISINOBAKOMMENTAR'], 'isinobakommentar', 'X(255)', 255);
            }
        }
    }
    $oVWM->GETFormularFooterUserverwaltungUPD();
    // INSERTMODE
} elseif ($sin_frmstatus == 'NEU'){
    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // INIT - Objektartnamen holen
    $queryisinart = "SELECT ISANAME "
        . "FROM afn_isinart ";
    $arrISINArt = $oDBM->GETSelectSQLSelect($queryisinart);

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltungUPD('neue IS-IN-Objektart anlegen', $file, 'isinobaid', $sin_isinobaid, 'INSERT');

    if (is_array($arrFehlermeldung))
    {
        $oVWM->GETFormRowStyle2('&nbsp;');
        $count = count($arrFehlermeldung);
        for ($i=0; $i<$count; $i++)
        {
            $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
        }
        $oVWM->GETFormRowStyle2('&nbsp;');
        $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Art', $sin_isaname, 'isaname', $arrISINArt);
        $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)', $sin_isinobaname, 'isinobaname', $arrObjektart);
        $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)', $sin_obaname, 'obaname', $arrObjektart);
        $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $sin_isinobakommentar, 'isinobakommentar', 'X(255)', 255);
    } else {
        $oVWM->GETFormRowStyle2('&nbsp;');
        $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Art', $sin_isaname, 'isaname', $arrISINArt);
        $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)', $sin_isinobaname, 'isinobaname', $arrObjektart);
        $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)', $sin_obaname, 'obaname', $arrObjektart);
        $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $sin_isinobakommentar, 'isinobakommentar', 'X(255)', 255);
    }
    $oVWM->GETFormularFooterUserverwaltungUPD();

    // READMODE (auch nach UPDATE!)
} else {
    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();
    // GET Filter-Daten 1
    $GETqueryFILTER1 = "SELECT DISTINCT MIN(t.TID), isa.ISANAME "

```

```

        ."FROM afn_isinobjart t, "
        ." afn_isinart isa "
        ."WHERE isa.TID = t.ISAID "
        ."GROUP BY t.ISAID "
        ."ORDER BY isa.ISANAME ";
$SoDBM->GETSelectSQL($GETQueryFILTER1,'TID,NAME',$arrFilter1);
// GET Filter-Daten 2
$GETQueryFILTER2 = "SELECT DISTINCT oba.TID, oba.OBANAME "
        ."FROM afn_isinobjart t, "
        ." afn_objektart oba "
        ."WHERE oba.TID = t.ISAID "
        ."ORDER BY oba.OBANAME";
$SoDBM->GETSelectSQL($GETQueryFILTER2,'TID,NAME',$arrFilter2);
// GET Filter-Daten 3
$GETQueryFILTER3 = "SELECT DISTINCT oba.TID, oba.OBANAME "
        ."FROM afn_isinobjart t, "
        ." afn_objektart oba "
        ."WHERE oba.TID = t.ISINOBAID "
        ."ORDER BY oba.OBANAME ";
$SoDBM->GETSelectSQL($GETQueryFILTER3,'TID,NAME',$arrFilter3);
$SoVWM->GETFormular3Filter('Thema (Pfad)', 'fltisinartid', $arrFilter1, 'IS-IN Objektart E1', 'fltisinobaid', $arrFilter3, 'Objektart
(E2)', 'fltobaid', $arrFilter2);

// GET DB-Daten
$query = "SELECT isinobabez.TID, isinoba.OBANAME, isa.ISANAME, oba.OBANAME, isinobabez.ISINOBAKOMMENTAR "
        ."FROM afn_isinArt isa, "
        ." afn_objektart oba, "
        ." afn_isinobjart isinobabez, "
        ." afn_objektart isinoba "
        ."WHERE isa.tid = isinobabez.isaid "
        ."AND oba.tid = isinobabez.obaid "
        ."AND isinoba.tid = isinobabez.isinobaid ";
// Filterdaten einbinden
if ($in_ftobaid != '0')
{
    $query = $query."AND isinobabez.OBAID = '$in_ftobaid' ";
} elseif ($in_fltisinobaid != '0') {
    $query = $query."AND isinobabez.ISINOBAID = '$in_fltisinobaid' ";
} elseif ($in_fltisinartid != '0') {
    # TODO => wenn es zu einem ISINART mehrere DS gibt => dann über KUERZEL auswählen!!!
    $query = $query."AND isinobabez.TID = '$in_fltisinartid' ";
}
$query = $query."ORDER BY isa.ISANAME, isinoba.TID ";
$SoDBM-
>GETSelectSQL($query, 'ISINOBAID, ISINOBANAME, ISANAME, ISOBANAME, ISINOBAKOMMENTAR', $arrFormular);

// Formular zusammenstellen
$SoVWM->GETFormularHeaderUserverwaltung('IS-IN-Objektartverwaltung', $file);
if (is_array($arrFormular))
{
    foreach ($arrFormular as $Daten)
    {
        {
            if ($counter > 0)
            {
                $SoVWM->GETFormRowStyle1('&nbsp;');
            }
            $SoVWM->GETFormRowStyleUPDATE2($Daten['ISANAME'], $file, $Daten['ISINOBAID'], 'isinobaid');
            $SoVWM->GETFormRowStyle3('Element');
            $SoVWM->GETFormRowStyleSELECTAdmin4(",$Daten['ISINOBANAME']"." =>
            ".$Daten['ISOBANAME'],$Daten['ISINOBAID']);
            $SoVWM->GETFormRowStyle3('Kommentar/Hilfe');
            $SoVWM->GETFormRowStyleTextarea($Daten['ISINOBAKOMMENTAR']);
            $counter++;
        }
    }
    $SoVWM->GETFormularFooter();
    // HTML-Seite Ende
    $SoLTM->GETFooter();
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihrem Administrator in Verbindung!";
}
}

```

```

} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihren Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle
    Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://'. $_SERVER['SERVER_NAME']. "/iFlora">Startseite</a> oder aktivieren
    sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
    $oLTM->GETHeader();
    $oVWM->GETLabel("iFlora Parameterverwaltung", "H1");
    $oVWM->GETTextfield($ERROR);
    $oLTM->GETFooter();
}

?>

```

9.3.1.2.8 Controller – „Benutzerpasswort“ (ADMINFormPW.php)

Im Prototyp nicht enthalten!

9.3.1.2.9 Controller – „Parameterverwaltung“ (ADMINFormParameter.php)

```

<?php

// INCLUDE
include_once(' ../lib/DBManager.php');
include_once(' ../lib/VIEWManager.php');
include_once(' ../lib/LAYOUTManager.php');
include_once(' ../config_iflora/sqladmin.php.inc');
include_once(' ../config_iflora/sqladmin.php.inc');
include_once(' ../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht", "PROJ");
$ERROR = "OK";
$DBERROR = "OK";
$SarrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$Stable = 'afn_parameter';

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$sin_frmstatus=$_REQUEST['frmstatus'];} else {$sin_frmstatus = '0';};
// Parametervariablen
if($_REQUEST['partid'] != ""){$sin_partid=$_REQUEST['partid'];} else {$sin_partid = 0;};
// Pflicht
if($_REQUEST['parname'] != ""){$sin_parname=$_REQUEST['parname'];} else {$sin_parname = 0;};
if($_REQUEST['parformat'] != ""){$sin_parformat=$_REQUEST['parformat'];} else {$sin_parformat = 0;};
if($_REQUEST['parnr'] != ""){$sin_parnr=$_REQUEST['parnr'];} else {$sin_parnr = 0;};
if($_REQUEST['prjname'] != ""){$sin_prjname=$_REQUEST['prjname'];} else {$sin_prjname = 0;};
// optional
if($_REQUEST['parkommentar'] != ""){$sin_parkommentar=$_REQUEST['parkommentar'];} else {$sin_parkommentar = ""};
if($_REQUEST['wlkname'] != ""){$sin_wlkname=$_REQUEST['wlkname'];} else {$sin_wlkname = 0;};
if($_REQUEST['isa'] != ""){$sin_isa=$_REQUEST['isa'];} else {$sin_isa = 0;};
if($_REQUEST['oba'] != ""){$sin_oba=$_REQUEST['oba'];} else {$sin_oba = 0;};
// Formate
if($_REQUEST['frmparname'] != ""){$sin_frmparname=$_REQUEST['frmparname'];} else {$sin_frmparname = 0;};
if($_REQUEST['frmparnr'] != ""){$sin_frmparnr=$_REQUEST['frmparnr'];} else {$sin_frmparnr = 0;};
if($_REQUEST['frmparformat'] != ""){$sin_frmparformat=$_REQUEST['frmparformat'];} else {$sin_frmparformat = 0;};
if($_REQUEST['frmparkommentar'] != ""){$sin_frmparkommentar=$_REQUEST['frmparkommentar'];} else {$sin_frmparkommentar = 0;};
// Filter
if($_REQUEST['fltparkl'] != ""){$sin_fltparkl=$_REQUEST['fltparkl'];} else {$sin_fltparkl = '0';};
if($_REQUEST['fltparameter'] != ""){$sin_fltparameter=$_REQUEST['fltparameter'];} else {$sin_fltparameter = '0';};

$SarrFormular = NULL;
$SarrFilter = NULL;
$SarrFilter2 = NULL;
$Scounter = 0;
$Sfile = "../ADMINFormParameter.php";

$SarrProjekt = NULL;

```

```

$arrWLKlasse = NULL;
$arrISINArt = NULL;
$arrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = ".././../config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$soVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$soLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_CookieData($SQLite);
    if ($Cookie <> "ERROR")
    {
        $arrCookie = explode(';', $Cookie);
        $aktiv = $pruefDBM->GETUserAktiv($arrCookie[0]);
        if ($aktiv == "aktiv")
        {
            $PRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht,$arrCookie[2]);
            if ($PRUEFBerechtigung > 0)
            {
                if ($arrCookie[1] == 'ADMIN')
                {
                    $user = $admin_user;
                    $passwd = $admin_passwd;
                } elseif ($arrCookie[1] == 'WORK') {
                    $user = $work_user;
                    $passwd = $work_passwd;
                } else {
                    $user = $data_user;
                    $passwd = $data_passwd;
                }
            }

            // INIT DBManager Objekt
            $soDBM = new DBManager($database,$sql_server,$user,$passwd,$sql_db,$schema);

            // ANWENDUNG
            // 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
            // 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
            // UPDATE
            if ($in_frmstatus == 'UPDATE')
            {
                if ($arrCookie[2] == 'ADMIN')
                {
                    $DBchange = 'true';
                } else {
                    $PREUFqueryDS_USER = "SELECT DS_USER "
                        ."FROM adm_username "
                        ."WHERE TID = '$in_usrtid' ";
                    $PRUEFds_user = $soDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
                    // $PRUEFProjekt = $pruefDBM->PRUEFProjekt($arrCookie[0],$in_usrtid);
                    $PRUEFProjekt = $pruefDBM->PRUEFProjTable($arrCookie[0],$in_partid,$stable);
                    // Ein Projektleiter darf alle Benutzer bearbeiten, die seinem Projekt zugeordnet oder von ihm angelegt sind!
                    if ($PRUEFProjekt > 0 || $arrCookie[0] == $PRUEFds_user)
                    {
                        $DBchange = 'true';
                    } else {
                        $DBchange = 'false';
                        $arrFehlermeldung[] = "Nur ein Projektleiter darf die Parameter seines Projektes oder Parameter, die er selbst angelegt hat, bearbeiten! Das trifft auf den gewählten Parameter nicht zu!";
                        $in_frmstatus = "EDIT";
                    }
                }
            }
            if ($DBchange == 'true')
            {
                if ($in_partid > '0' && $in_parnr != '0' && $in_parname != '0' && $in_parformat != '0' && $in_prjname != '0')
                {
                    # Prüfen, ob Datensatz vorhanden und eindeutig ist!
                    $PREUFqueryUPDATE = "SELECT count(*) "
                        ."FROM afn_parameter "
                        ."WHERE TID = '$in_partid' ";

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
$PRUEFDs = $oDBM->GETSelectSQLDatum($PRUEFqueryUPDATE);
if ($PRUEFDs == 1)
{
    $ds = 'true';
} else {
    $ds = 'false';
    $arrFehlermeldung[] = "Der ausgewählte Parameter ist nicht (mehr) in der Datenbank vorhanden!";
}
# Prüfen, ob "parname" (UNIQUE INDEX) eindeutig ist!
$PRUEFqueryPARNAME = "SELECT count(*) "
                    . "FROM afn_parameter "
                    . "WHERE PARNAME = '$in_parname' "
                    . "AND TID <> '$in_partid' ";
$PRUEFparn = $oDBM->GETSelectSQLDatum($PRUEFqueryPARNAME);
if ($PRUEFparn == 0)
{
    $pn = 'true';
} else {
    $pn = 'false';
    $arrFehlermeldung[] = "Jeder Parameter muss einen eindeutigen <b>PARAMETERNAME</b> haben - der eingegebene
Parametername wird bereits für einen anderen Parameter verwendet!";
}
# Prüfen, ob "parnr" (UNIQUE INDEX) eindeutig ist!
$PRUEFqueryPARNR = "SELECT count(*) "
                    . "FROM afn_parameter "
                    . "WHERE PARNR = '$in_parnr' "
                    . "AND TID <> '$in_partid' ";
$PRUEFparnr = $oDBM->GETSelectSQLDatum($PRUEFqueryPARNR);
if ($PRUEFparnr == 0)
{
    $pnr = 'true';
} else {
    $pnr = 'false';
    $arrFehlermeldung[] = "Wenn eine <b>PARAMETERNUMMER</b> für einen Parameter verwendet wird, dann muss er
eindeutigen sein - die eingegebene Parameternummer wird bereits für in einen anderen Parameter verwendet!";
}
// Formatprüfung der eingegebenen Werte
$PRUEFparname = $oVWM->PRUEFFormat($in_frmparname,$in_parname);
$PRUEFparnr = $oVWM->PRUEFFormat($in_frmparnr,$in_parnr);
$PRUEFparformat = $oVWM->PRUEFFormat($in_frmparformat,$in_parformat);
$PRUEFparkommentar = $oVWM->PRUEFFormat($in_frmparkommentar,$in_parkommentar);

if ($ds == 'true' &&
    $pn == 'true' &&
    $pnr == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFparname == 'true' &&
    $PRUEFparnr == 'true' &&
    $PRUEFparformat == 'true' &&
    $PRUEFparkommentar == 'true')
{
    $queryUPDATE = "UPDATE afn_parameter SET DS_DATE = $DBDateFkt, "
                    . "DS_USER = '$arrCookie[0]', "
                    . "PARNAME = '$in_parname', "
                    . "PARNR = '$in_parnr', "
                    . "PARFORMAT = '$in_parformat', "
                    . "PARKOMMENTAR = '$in_parkommentar' "
                    . "WHERE TID = '$in_partid' ";
    $oDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
    if ($PRUEFparname == 'false')
    {
        $arrFehlermeldung[] = "<b>Parametername:</b> Der Parametername darf nicht länger als 50 Buchstaben sein!";
    }
    if ($PRUEFparnr == 'false')
    {
        $arrFehlermeldung[] = "<b>Parameternummer:</b> Die Parameternummer darf nicht länger als 4 Buchstaben
sein!";
    }
    if ($PRUEFparformat == 'false')
    {
        $arrFehlermeldung[] = "<b>Parameterformat:</b> Das Parameterformat darf nicht länger als 10 Buchstaben
sein!";
    }
    if ($PRUEFparkommentar == 'false')
```

```

    {
        $arrFehlermeldung[] = "<b>Parameterkommentar:</b> Der Parameterkommentar darf nicht länger als 255
Buchstaben sein!";
    }
    $in_frmstatus = "EDIT";
}
} else {
    if ($in_parname == '0')
    {
        $arrFehlermeldung[] = "<b>Parametername:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parname = "";
    }
    if ($in_parnr == '0')
    {
        $arrFehlermeldung[] = "<b>Parameternummer:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parnr = "";
    }
    if ($in_parformat == '0')
    {
        $arrFehlermeldung[] = "<b>Parameterformat:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parname = "";
    }
    if ($in_prjname == '0')
    {
        $arrFehlermeldung[] = "<b>Projekt:</b> Es muss ein gültiges Projekt ausgewählt werden!";
        $in_parname = "";
    }
    $in_frmstatus = "EDIT";
}
}
}
// INSERT
if ($in_frmstatus == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN' || $arrCookie[2] == 'PROJ')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Ein Projektleiter darf nur für sein Projekt einen neuen Parameter
angelegen!";
        $in_frmstatus = "NEU";
    }
    if ($DBchange == 'true')
    {
        if ($in_parnr != '0' && $in_parname != '0' && $in_parformat != '0' && $in_prjname != '0')
        {
            # Prüfen, ob "parname" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryPARNAME = "SELECT count(*) "
                                . "FROM   afn_parameter "
                                . "WHERE  PARNAME = '$in_parname' "
                                . "AND    TID <> '$in_partid' ";
            $PRUEFpnr = $oDBM->GETSelectSQLDatum($PRUEFqueryPARNAME);
            if ($PRUEFpnr == 0)
            {
                $pn = 'true';
            } else {
                $pn = 'false';
                $arrFehlermeldung[] = "Jeder Parameter muss einen eindeutigen <b>PARAMETERNAME</b> haben - der eingegebene
Parametername wird bereits für einen anderen Parameter verwendet!";
            }
            # Prüfen, ob "parnr" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryPARNR = "SELECT count(*) "
                              . "FROM   afn_parameter "
                              . "WHERE  PARNR = '$in_parnr' "
                              . "AND    TID <> '$in_partid' ";
            $PRUEFpnr = $oDBM->GETSelectSQLDatum($PRUEFqueryPARNR);
            if ($PRUEFpnr == 0)
            {
                $pnr = 'true';
            } else {
                $pnr = 'false';
                $arrFehlermeldung[] = "Wenn eine <b>PARAMETERNUMMER</b> für einen Parameter verwendet wird, dann muss er
eindeutigen sein - die eingegebene Parameternummer wird bereits für in einen anderen Parameter verwendet!";
            }
        }
    }
}

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
// Formatprüfung der eingegebenen Werte
$PRUEFparname = $oVWM->PRUEFFormat($in_frmparname,$in_parname);
$PRUEFparnr = $oVWM->PRUEFFormat($in_frmparnr,$in_parnr);
$PRUEFparformat = $oVWM->PRUEFFormat($in_frmparformat,$in_parformat);
$PRUEFparkommentar = $oVWM->PRUEFFormat($in_frmparkommentar,$in_parkommentar);

if ($pn == 'true' &&
    $pnr == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFparname == 'true' &&
    $PRUEFparnr == 'true' &&
    $PRUEFparformat == 'true' &&
    $PRUEFparkommentar == 'true')
{
    $queryUPDATE = "INSERT INTO afn_parameter SET DS_DATE = $DBDateFkt, "
                  ."DS_USER = '$arrCookie[0]', "
                  ."PARNAME = '$in_parname', "
                  ."PARNR = '$in_parnr', "
                  ."PARFORMAT = '$in_parformat', "
                  ."PARKOMMENTAR = '$in_parkommentar' ";

    $oDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
    if ($PRUEFparname == 'false')
    {
        $arrFehlermeldung[] = "<b>Parametername:</b> Der Parametername darf nicht länger als 50 Buchstaben sein!";
    }
    if ($PRUEFparnr == 'false')
    {
        $arrFehlermeldung[] = "<b>Parameternummer:</b> Die Parameternummer darf nicht länger als 4 Buchstaben
sein!";
    }
    if ($PRUEFparformat == 'false')
    {
        $arrFehlermeldung[] = "<b>Parameterformat:</b> Das Parameterformat darf nicht länger als 10 Buchstaben
sein!";
    }
    if ($PRUEFparkommentar == 'false')
    {
        $arrFehlermeldung[] = "<b>Parameterkommentar:</b> Der Parameterkommentar darf nicht länger als 255
Buchstaben sein!";
    }
    $in_frmstatus = "NEU";
}
} else {
    if ($in_parname == '0')
    {
        $arrFehlermeldung[] = "<b>Parametername:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parname = "";
    }
    if ($in_parnr == '0')
    {
        $arrFehlermeldung[] = "<b>Parameternummer:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parnr = "";
    }
    if ($in_parformat == '0')
    {
        $arrFehlermeldung[] = "<b>Parameterformat:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_parformat = "";
    }
    if ($in_prjname == '0')
    {
        $arrFehlermeldung[] = "<b>Projekt:</b> Es muss ein gültiges Projekt ausgewählt werden!";
        $in_parname = "";
    }
}

$in_frmstatus = "NEU";
}
}
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
    $SELECTquery = "SELECT TID AS PARID, PARNAME, PARNR, PARFORMAT, PARKOMMENTAR, DS_USER "
                  ."FROM afn_parameter "
                  ."WHERE TID = $in_partid ";
}
```



```

SoDBM-
>GETSelectSQL($SELECTquery,'PARID,PARNAME,PARNR,PARFORMAT,PARKOMMENTAR,DS_USER',$sarrFormular);

// INIT - Projektnamen holen
$queryProjekte = "SELECT PRJNAME "
                ."FROM afn_projekt ";
if ($sarrCookie[2] != 'ADMIN')
{
    $queryProjekte = $queryProjekte."WHERE PRJNAME <> 'ADMIN' ";
} elseif ($sarrCookie[2] != 'PROJ' && $sarrCookie[2] != 'ADMIN') {
    $queryProjekte = $queryProjekte."WHERE TID IN (SELECT PROJEKTID "
                ."FROM adm_user_rechte "
                ."WHERE USERID = $sarrCookie[0])";
}
$sarrProjekt = $oDBM->GETSelectSQLSelect($queryProjekte);

// INIT - WertelistenKlassenNamen holen
$queryWLKasse = "SELECT WTKNAME "
                ."FROM afn_werteklassen ";
$sarrWLKasse = $oDBM->GETSelectSQLSelect($queryWLKasse);

// INIT - $sarrISINArt holen
$queryISINArt = "SELECT ISANAME "
                ."FROM afn_isinart ";
$sarrISINArt = $oDBM->GETSelectSQLSelect($queryISINArt);

// INIT - $sarrObjArt holen
$queryObjArt = "SELECT OBANAME "
                ."FROM afn_objektart ";
$sarrObjArt = $oDBM->GETSelectSQLSelect($queryObjArt);

// HTML-Seite Anfang
$oLTM->GETFORMHeader();

// Formular zusammenstellen
$oVWM->GETFormularHeaderUserverwaltungUPD('Parameter ändern',$file,'partid',$sin_partid,'UPDATE');

if (is_array($sarrFormular))
{
    foreach ($sarrFormular as $Daten)
    {
        if ($sarrCookie[2] == 'ADMIN')
        {
            $schange = 'true';
        } else {
            // $PRUEFProjekt = $pruefDBM->PRUEFProjekt($sarrCookie[0],$sin_usrtid);
            $PRUEFProjekt = $pruefDBM->PRUEFProjTable($sarrCookie[0],$Daten['PARID'],$stable);
            // Ein Projektleiter darf alle Parameter bearbeiten, die spez. für sein Projekt eingegeben wurden oder von ihm angelegt
            if ($PRUEFProjekt > 0 || $sarrCookie[0] == $Daten['DS_USER'])
            {
                $schange = 'true';
            } else {
                $schange = 'false';
            }
        }

        if ($schange == 'true')
        {
            if (is_array($sarrFehlermeldung))
            {
                $oVWM->GETFormRowStyle2('&nbsp;');
                $scount = count($sarrFehlermeldung);
                for ($i=0; $i<$scount; $i++)
                {
                    $oVWM->GETFormRowStyleERRORAdmin4($sarrFehlermeldung[$i]);
                }
                $oVWM->GETFormRowStyle2('&nbsp;');
                $oVWM->GETFormRowStyle3('Pflichtfelder');
                $oVWM->GETFormRowStyleUPDAdmin4('Parametername',$sin_parnr,'parname','X(50)',50);
                $oVWM->GETFormRowStyleUPDAdmin4('Parameternummer',$sin_parnr,'parnr','X(4)',4);
                $oVWM->GETFormRowStyleUPDAdmin4('Parameterformat',$sin_parformat,'parformat','X(10)',10);
                $oVWM->GETFormRowStyleINSARRAdmin4('Projekt',$sin_prjname,'prjname',$sarrProjekt);
                $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$sin_parkommentar,'parkommentar','X(100)',100);
                $oVWM->GETFormRowStyle2('Wahlfelder');
            }
        }
    }
}

```

worden sind!

```

$soVWM->GETFormRowStyleINSARRAdmin4('Wertelisteklasse',$sin_wlklname,'wlklname',$sarrWLKlasse);
$soVWM->GETFormRowStyleINSARRAdmin4('Beziehung',$sin_isa,'isa',$sarrISINArt);
$soVWM->GETFormRowStyleINSARRAdmin4('Objekttyp',$sin_oba,'oba',$sarrObjArt);
} else {
$soVWM->GETFormRowStyle2('&nbsp;');
$soVWM->GETFormRowStyle3('Pflichtfelder');
$soVWM->GETFormRowStyleUPDAdmin4('Parametername',$Daten['PARNAME'],'parname','X(50)',50);
$soVWM->GETFormRowStyleUPDAdmin4('Parameternummer',$Daten['PARNR'],'parnr','X(4)',4);
$soVWM->GETFormRowStyleUPDAdmin4('Parameterformat',$Daten['PARFORMAT'],'parformat','X(10)',10);
$soVWM->GETFormRowStyleINSARRAdmin4('Projekt',$sin_prjname,'prjname',$sarrProjekt);
$soVWM-
>GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$Daten['PARKKOMMENTAR'],'parkommentar','X(255)',255);
$soVWM->GETFormRowStyle3('Wahlfelder');
$soVWM->GETFormRowStyleINSARRAdmin4('Wertelisteklasse',$sin_wlklname,'wlklname',$sarrWLKlasse);
$soVWM->GETFormRowStyleINSARRAdmin4('Beziehung',$sin_isa,'isa',$sarrISINArt);
$soVWM->GETFormRowStyleINSARRAdmin4('Objekttyp',$sin_oba,'oba',$sarrObjArt);
}
}
}
$soVWM->GETFormularFooterUserverwaltungUPD();
// INSERTMODE
} elseif ($sin_frmstatus == 'NEU'){
// INIT - Projektnamen holen
$queryProjekte = "SELECT PRJNAME "
                ."FROM   afn_projekt ";
if ($sarrCookie[2] != 'ADMIN')
{
    $queryProjekte = $queryProjekte."WHERE PRJNAME <> 'ADMIN' ";
} elseif ($sarrCookie[2] != 'PROJ' && $sarrCookie[2] != 'ADMIN') {
    $queryProjekte = $queryProjekte."WHERE TID IN (SELECT PROJEKTID "
                ."FROM   adm_user_rechte "
                ."WHERE  USERID = $sarrCookie[0]";
}
$sarrProjekt = $soDBM->GETSelectSQLSelect($queryProjekte);

// INIT - WertelistenKlassenNamen holen
$queryWLKlasse = "SELECT WTKNAME "
                ."FROM   afn_werteklassen ";
$sarrWLKlasse = $soDBM->GETSelectSQLSelect($queryWLKlasse);

// INIT - $sarrISINArt holen
$queryISINArt = "SELECT ISANAME "
                ."FROM   afn_isinart ";
$sarrISINArt = $soDBM->GETSelectSQLSelect($queryISINArt);

// INIT - $sarrObjArt holen
$queryObjArt = "SELECT OBANAME "
                ."FROM   afn_objektart ";
$sarrObjArt = $soDBM->GETSelectSQLSelect($queryObjArt);

// HTML-Seite Anfang
$soLTM->GETFORMHeader();

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltungUPD('neuen Parameter anlegen',$file,'partid',$sin_partid,'INSERT');

if (is_array($sarrFehlermeldung))
{
    $soVWM->GETFormRowStyle2('&nbsp;');
    $count = count($sarrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $soVWM->GETFormRowStyleERRORAdmin4($sarrFehlermeldung[$i]);
    }
    $soVWM->GETFormRowStyle2('&nbsp;');
    $soVWM->GETFormRowStyle3('Pflichtfelder');
    $soVWM->GETFormRowStyleUPDAdmin4('Parametername',$sin_parname,'parname','X(50)',50);
    $soVWM->GETFormRowStyleUPDAdmin4('Parameternummer',$sin_parnr,'parnr','X(4)',4);
    $soVWM->GETFormRowStyleUPDAdmin4('Parameterformat',$sin_parformat,'parformat','X(10)',10);
    $soVWM->GETFormRowStyleINSARRAdmin4('Projekt',$sin_prjname,'prjname',$sarrProjekt);
    $soVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$sin_parkommentar,'parkommentar','X(255)',255);
    $soVWM->GETFormRowStyle3('Wahlfelder');
    $soVWM->GETFormRowStyleINSARRAdmin4('Wertelisteklasse',$sin_wlklname,'wlklname',$sarrWLKlasse);
    $soVWM->GETFormRowStyleINSARRAdmin4('Beziehung',$sin_isa,'isa',$sarrISINArt);
}

```

```

$VWM->GETFormRowStyleINSARRAdmin4('Objekttyp','sin_oba','oba',$sarrObjArt);
} else {
$VWM->GETFormRowStyle2('&nbsp;');
$VWM->GETFormRowStyle3('Pflichtfelder');
$VWM->GETFormRowStyleUPDAdmin4('Parametername','','parname','X(50)',50);
$VWM->GETFormRowStyleUPDAdmin4('Parameternummer','','parnr','X(4)',4);
$VWM->GETFormRowStyleUPDAdmin4('Parameterformat','','parformat','X(10)',10);
$VWM->GETFormRowStyleINSARRAdmin4('Projekt','','prjname',$sarrProjekt);
$VWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe','','parkommentar','X(255)',255);
$VWM->GETFormRowStyle3('Wahlfelder');
$VWM->GETFormRowStyleINSARRAdmin4('Wertelisteklasse','','wlkname',$sarrWLKlasse);
$VWM->GETFormRowStyleINSARRAdmin4('Beziehung','','isa',$sarrISINArt);
$VWM->GETFormRowStyleINSARRAdmin4('Objekttyp','','oba',$sarrObjArt);
}
$VWM->GETFormularFooterUserverwaltungUPD();

// READMODE (auch nach UPDATE!)
} else {
// HTML-Seite Anfang
$soLTM->GETFORMHeader();

// GET Filter-Daten
$GETqueryFILTER = "SELECT TID AS TID, concat(PARNR, ' - ', PARNAME) AS NAME "
                  ."FROM   afn_parameter "
                  ."WHERE  PARNR <> 'LEER' "
                  ."ORDER BY PARNR, PARNAME ";
$soDBM->GETSelectSQL($GETqueryFILTER,'TID,NAME',$sarrFilter);

// GET Filter-Daten
$GETqueryFILTER2 = "SELECT DISTINCT substring(PARNR,1,1) AS TID, concat(substring(PARNR,1,1),'-Klasse') AS
PARKLASSE "
                  ."FROM   afn_parameter "
                  ."ORDER BY PARKLASSE ";
$soDBM->GETSelectSQL($GETqueryFILTER2,'TID,NAME',$sarrFilter2);

$soVWM->GETFormular2Filter('Parameterklasse','fltparkl',$sarrFilter2,'Parameter','fltparameter',$sarrFilter);

// GET DB-Daten
$query = "SELECT pa.TID, concat(pa.PARNR, ' ', pa.PARNAME) AS PARNAME, pa.PARFORMAT, pa.PARKOMMENTAR,
pr.PRJNAME, "
        ."ifnull(w.WTKNAME,'Keine Auswahl'), ifnull(i.ISANAME,'Keine Auswahl'), "
        ."ifnull(o.OBANAME,'Keine Auswahl') "
        ."FROM   afn_parameter pa, "
        ."       afn_projekt pr "
        ."       LEFT JOIN afn_werteklassen w ON w.TID = pa.WLKID "
        ."       LEFT JOIN afn_isinart i ON i.TID = pa.ISAID "
        ."       LEFT JOIN afn_objektart o ON o.TID = pa.LINKOBAID "
        ."WHERE  pa.PARNR <> 'LEER' "
        ."AND    pr.TID = pa.PRJID ";
// Filterdaten einbinden
if ($sin_fltparkl != '0')
{
$query = $query."AND pa.PARNR LIKE '$sin_fltparkl%' ";
} elseif ($sin_fltparameter != '0') {
$query = $query."AND pa.TID = '$sin_fltparameter' ";
}
$query = $query."ORDER BY PARNR ";
$soDBM-
>GETSelectSQL($query,'PARID,PARNAME,PARFORMAT,PARKOMMENTAR,PRJNAME,WTKNAME,ISANAME,OBANAME',$sarrF
ormular);

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltung('Parameterverwaltung',$file);
if (is_array($sarrFormular))
{
foreach ($sarrFormular as $Daten)
{
if ($counter > 0)
{
$soVWM->GETFormRowStyle1('&nbsp;');
}
$PRUEFProjekt = $pruefDBM->PRUEFProjTable($sarrCookie[0],$Daten['PARID'],$table);
if ($PRUEFProjekt > 0 || $sarrCookie[2] == 'ADMIN')
{
$soVWM->GETFormRowStyleUPDATE2($Daten['PARNAME'],$file,$Daten['PARID'],'partid');
}
}
}

```

```

    } else {
        $oVWM->GETFormRowStyleGESPERRT2($Daten['PARNAME']);
    }
    $oVWM->GETFormRowStyle3('Format und Verknüpfungen');
    $oVWM->GETFormRowStylePARFORMAT4('Parameterformat',$Daten['PARFORMAT'],$Daten['PARID']);
    $oVWM->GETFormRowStylePARFORMAT4('Projekt',$Daten['PRJNAME'],$Daten['PARID']);
    $oVWM->GETFormRowStylePARFORMAT4('Wertelistenkasse',$Daten['WTKNAME'],$Daten['PARID']);
    $oVWM->GETFormRowStylePARFORMAT4('Objektart',$Daten['OBANAME'],$Daten['PARID']);
    $oVWM->GETFormRowStylePARFORMAT4('Beziehung',$Daten['ISANAME'],$Daten['PARID']);
    $oVWM->GETFormRowStyle3('Kommentar/Hilfe');
    $oVWM->GETFormRowStyleTextarea($Daten['PARKOMMENTAR']);
    $counter++;
}
}
$oVWM->GETFormularFooter();
// HTML-Seite Ende
$oLTM->GETFooter();
}
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihrem Administrator in Verbindung!";
}
} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihrem Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://".$_SERVER['SERVER_NAME']."/iFlora>Startseite</a> oder aktivieren sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
    $oLTM->GETHeader();
    $oVWM->GETLabel("iFlora Parameterverwaltung", "H1");
    $oVWM->GETTextfield($ERROR);
    $oLTM->GETFooter();
}
}
?>

```

9.3.1.2.10 Controller – „Formular“ (ADMINFormFormular.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../..../config_iflora/sqladmin.php.inc');
include_once('../..../config_iflora/sqladmin.php.inc');
include_once('../..../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht","PROJ");
$ERROR = "OK";
$DBERROR = "OK";
$arrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$stable = 'afn_h_formular';

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$sin_frmstatus=$_REQUEST['frmstatus'];} else {$sin_frmstatus = '0';};
// Parametervariablen
if($_REQUEST['frmtid'] != ""){$sin_frmtid=$_REQUEST['frmtid'];} else {$sin_frmtid = 0;};
// Pflicht
if($_REQUEST['frmname'] != ""){$sin_frmname=$_REQUEST['frmname'];} else {$sin_frmname = 0;};
if($_REQUEST['prjname'] != ""){$sin_prjname=$_REQUEST['prjname'];} else {$sin_prjname = 0;};
if($_REQUEST['obaname'] != ""){$sin_obaname=$_REQUEST['obaname'];} else {$sin_obaname = 0;};
// Formate
if($_REQUEST['frmfrmname'] != ""){$sin_frmfrmname=$_REQUEST['frmfrmname'];} else {$sin_frmfrmname = 0;};

```

```

if($ _REQUEST['frmprjname'] != ""){$sin_frmprjname=$_REQUEST['frmprjname'];} else {$sin_frmprjname = 0;};
if($ _REQUEST['frmobaname'] != ""){$sin_frmobaname=$_REQUEST['frmobaname'];} else {$sin_frmobaname = 0;};
// Filter
if($ _REQUEST['fltprojekt'] != ""){$sin_fltprojekt=$_REQUEST['fltprojekt'];} else {$sin_fltprojekt = '0';};
if($ _REQUEST['fltformular'] != ""){$sin_fltformular=$_REQUEST['fltformular'];} else {$sin_fltformular = '0';};

$SarrFormular = NULL;
$SarrFilter = NULL;
$SarrFilter2 = NULL;
$Scounter = 0;
$Sfile = ".ADMINFormFormular.php";
$SfrFile = ".ADMINFormFormRow.php";

$SarrProjekt = NULL;
$SarrWlKlasse = NULL;
$SarrISINArt = NULL;
$SarrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = "../././config_iflora/data/userdb.dat";
$SpruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$SoVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$SoLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $SpruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $SarrCookie = explode(';', $Cookie);
        $Saktiv = $SpruefDBM->GETUserAktiv($SarrCookie[0]);
        if ($Saktiv == "aktiv")
        {
            $SPRUEFBerechtigung = $SpruefDBM->PRUEFBerechtigung(formularrecht, $SarrCookie[2]);
            if ($SPRUEFBerechtigung > 0)
            {
                if ($SarrCookie[1] == 'ADMIN')
                {
                    $Suser = $Sadmin_user;
                    $Spasswd = $Sadmin_passwd;
                } elseif ($SarrCookie[1] == 'WORK') {
                    $Suser = $Swork_user;
                    $Spasswd = $Swork_passwd;
                } else {
                    $Suser = $Sdata_user;
                    $Spasswd = $Sdata_passwd;
                }
            }
        }

        // INIT DBManager Objekt
        $SoDBM = new DBManager($database,$sql_server,$Suser,$Spasswd,$sql_db,$schema);

        // ANWENDUNG
        // 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
        // 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
        // UPDATE
        if ($sin_frmstatus == 'UPDATE')
        {
            if ($SarrCookie[2] == 'ADMIN')
            {
                {
                    $DBchange = 'true';
                } else {
                    $PREUFqueryDS_USER = "SELECT DS_USER "
                                        ."FROM adm_username "
                                        ."WHERE TID = 'sin_usrtid' ";
                    $SPRUEFds_user = $SoDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
                    // $SPRUEFProjekt = $SpruefDBM->PRUEFProjekt($SarrCookie[0], $sin_usrtid);
                    $SPRUEFProjekt = $SpruefDBM->PRUEFProjTable($SarrCookie[0], $sin_frmtid, $Stable);
                    // Ein Projektleiter darf alle Benutzer bearbeiten, die seinem Projekt zugeordnet oder von ihm angelegt sind!
                    if ($SPRUEFProjekt > 0 || $SarrCookie[0] == $SPRUEFds_user)
                    {
                        $DBchange = 'true';
                    } else {

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
$DBchange = 'false';
$arrFehlermeldung[] = "Ein Projektleiter darf nur die Formulare seines Projektes bearbeiten! Das trifft auf das gewählte
Formular nicht zu!";
    $in_frmstatus = "EDIT";
    }
    }
if ($DBchange == 'true')
{
    if ($in_frmtid > '0' && $in_frmname != '0' && $in_prjname != '0' && $in_obaname != '0')
    {
        # Prüfen, ob Datensatz vorhanden und eindeutig ist!
        $PRUEFqueryUPDATE = "SELECT count(*) "
            ."FROM afn_h_formular "
            ."WHERE TID = '$in_frmtid' ";

        $PRUEFds = $oDBM->GETSelectSQLDatum($PRUEFqueryUPDATE);
        if ($PRUEFds == 1)
        {
            $ds = 'true';
        } else {
            $ds = 'false';
        }
        $arrFehlermeldung[] = "Das ausgewählte Formular ist nicht (mehr) in der Datenbank vorhanden!";
    }
    # Prüfen, ob "frmname" (UNIQUE INDEX) eindeutig ist!
    $PRUEFqueryFRMNAME = "SELECT count(*) "
        ."FROM afn_formular "
        ."WHERE FRMNAME = '$in_frmname' "
        ."AND TID <> '$in_frmtid' ";

    $PRUEFpnm = $oDBM->GETSelectSQLDatum($PRUEFqueryFRMNAME);
    if ($PRUEFpnm == 0)
    {
        $fn = 'true';
    } else {
        $fn = 'false';
    }
    $arrFehlermeldung[] = "Jedes Formular muss einen eindeutigen <b>FORMULARNAMEN</b> haben - der eingegebene
Formularname wird bereits für einen anderen Parameter verwendet!";
    }
    // Formatprüfung der eingegebenen Werte
    $PRUEFfrmname = $oVWM->PRUEFFormat($in_frmfrmname,$in_frmname);

    if ($ds == 'true' &&
        $fn == 'true' &&
        $DBERROR == 'OK' &&
        $PRUEFfrmname == 'true')
    {
        $queryUPDATE = "UPDATE afn_h_formular SET DS_DATE = $DBDateFkt, "
            ."DS_USER = '$arrCookie[0]', "
            ."FRMNAME = '$in_frmname', "
            ."PRJID = (SELECT TID "
                ."FROM afn_projekt "
                ."WHERE PRJNAME = '$in_prjname'), "
            ."OBAID = (SELECT TID "
                ."FROM afn_objektart "
                ."WHERE OBANAME = '$in_obaname') "
            ."WHERE TID = '$in_frmtid' ";
        $oDBM->GETUpdateSQLDatum($queryUPDATE);
    } else {
        if ($PRUEFfrmname == 'false')
        {
            $arrFehlermeldung[] = "<b>Formularname:</b> Der Formularname darf nicht länger als 50 Buchstaben sein!";
        }
        $in_frmstatus = "EDIT";
    }
    } else {
        if ($in_frmname == '0')
        {
            $arrFehlermeldung[] = "<b>Formularname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
            $in_frmname = "";
        }
        if ($in_prjname == '0')
        {
            $arrFehlermeldung[] = "<b>Projekt:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
            $in_prjname = "";
        }
        if ($in_obaname == '0')
```

```

    {
        $arrFehlermeldung[] = "<b>Objektart:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
        $in_obaname = "";
    }
    $in_frmstatus = "EDIT";
}
}
// INSERT
if ($in_frmstatus == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN' || $arrCookie[2] == 'PROJ')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Ein Projektleiter darf nur für sein Projektes ein neues Formular anlegen!";
        $in_frmstatus = "NEU";
    }
    if ($DBchange == 'true')
    {
        if ($in_frmname != '0' && $in_prjname != '0' && $in_obaname != '0')
        {
            # Prüfen, ob "frmname" (UNIQUE INDEX) eindeutig ist!
            $PRUEFqueryFRMNAME = "SELECT count(*) "
                                ."FROM   afn_formular "
                                ."WHERE  FRMNAME = '$in_frmname' ";
            $PRUEFpam = $oDBM->GETSelectSQLDatum($PRUEFqueryFRMNAME);
            if ($PRUEFpn == 0)
            {
                $fn = 'true';
            } else {
                $fn = 'false';
                $arrFehlermeldung[] = "Jedes Formular muss einen eindeutigen <b>FORMULARNAMEN</b> haben - der eingegebene
Formularname wird bereits für einen anderen Parameter verwendet!";
            }
            // Formatprüfung der eingegebenen Werte
            $PRUEFfrmname = $oVWM->PRUEFFormat($in_frmfrmname,$in_frmname);

            if ($fn == 'true' &&
                $DBERROR == 'OK' &&
                $PRUEFfrmname == 'true')
            {
                $queryINSERT = "INSERT INTO afn_h_formular SET DS_DATE = $DBDateFkt, "
                                ."DS_USER = '$arrCookie[0]', "
                                ."FRMNAME = '$in_frmname', "
                                ."PRJID = (SELECT TID "
                                ."FROM afn_projekt "
                                ."WHERE PRJNAME = '$in_prjname'), "
                                ."OBAID = (SELECT TID "
                                ."FROM afn_objektart "
                                ."WHERE OBANAME = '$in_obaname') ";

                $oDBM->GETUpdateSQLDatum($queryINSERT);
            } else {
                if ($PRUEFfrmname == 'false')
                {
                    $arrFehlermeldung[] = "<b>Formularname:</b> Der Formularname darf nicht länger als 50 Buchstaben sein!";
                }
                $in_frmstatus = "NEU";
            }
        } else {
            if ($in_frmname == '0')
            {
                $arrFehlermeldung[] = "<b>Formularname:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
                $in_frmname = "";
            }
            if ($in_prjname == '0')
            {
                $arrFehlermeldung[] = "<b>Projekt:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
                $in_prjname = "";
            }
            if ($in_obaname == '0')
            {
                $arrFehlermeldung[] = "<b>Objektart:</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
                $in_obaname = "";
            }
        }
    }
}

```

```

    }
    $in_frmstatus = "NEU";
  }
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
  $query = "SELECT f.TID, f.FRMNAME, pr.PRJNAME, oa.OBANAME, f.DS_USER "
    ."FROM   afn_h_formular f, "
    ."       afn_projekt pr, "
    ."       afn_objektart oa "
    ."WHERE  pr.TID = f.PRJID "
    ."AND    oa.TID = f.OBAID "
    ."AND    f.TID = $in_frmid";
  $oDBM->GETSelectSQL($query, 'FRMID, FRMNAME, PRJNAME, OBANAME, DS_USER', $arrFormular);

  // INIT - Projektnamen holen
  $queryProjekte = "SELECT PRJNAME "
    ."FROM   afn_projekt ";
  if ($arrCookie[2] != 'ADMIN')
  {
    $queryProjekte = $queryProjekte."WHERE PRJNAME <> 'ADMIN' ";
  }
  elseif ($arrCookie[2] != 'PROJ' && $arrCookie[2] != 'ADMIN') {
    $queryProjekte = $queryProjekte."WHERE TID IN (SELECT PROJEKTID "
      ."FROM   adm_user_rechte "
      ."WHERE  USERID = $arrCookie[0])";
  }
  $arrProjekt = $oDBM->GETSelectSQLSelect($queryProjekte);

  // INIT - ObjektartNamen holen
  $queryOBA = "SELECT OBANAME "
    ."FROM   afn_objektart ";
  $arrObjektart = $oDBM->GETSelectSQLSelect($queryOBA);

  // HTML-Seite Anfang
  $oLTM->GETFORMHeader();

  // Formular zusammenstellen
  $oVWM->GETFormularHeaderUserverwaltungUPD('Formular ändern', $file, 'frmtd', $in_frmtd, 'UPDATE');

  if (is_array($arrFormular))
  {
    foreach ($arrFormular as $Daten)
    {
      if ($arrCookie[2] == 'ADMIN')
      {
        $change = 'true';
      }
      else {
        // $PRUEFProjekt = $pruefDBM->PRUEFProjekt($arrCookie[0], $in_usrtid);
        $PRUEFProjekt = $pruefDBM->PRUEFProjTable($arrCookie[0], $Daten['FRMID'], $table);
        // Ein Projektleiter darf alle Parameter bearbeiten, die spez. für sein Projekt eingegeben wurden oder von ihm angelegt
        if ($PRUEFProjekt > 0 || $arrCookie[0] == $Daten['DS_USER'])
        {
          $change = 'true';
        }
        else {
          $change = 'false';
        }
      }
    }

    if ($change == 'true')
    {
      if (is_array($arrFehlermeldung))
      {
        $oVWM->GETFormRowStyle2('&nbsp;');
        $count = count($arrFehlermeldung);
        for ($i=0; $i<$count; $i++)
        {
          $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
        }
        $oVWM->GETFormRowStyle2('&nbsp;');
        $oVWM->GETFormRowStyleUPDAdmin4('Formularname', $in_frmname, 'frmname', 'X(50)', 50);
        $oVWM->GETFormRowStyleINSARRAdmin4('Projekt', $in_prjname, 'prjname', $arrProjekt);
        $oVWM->GETFormRowStyleINSARRAdmin4('Objektart', $in_obaname, 'obaname', $arrObjektart);
      }
    }
  }
}

```



```

    } else {
        $oVWM->GETFormRowStyle2('&nbsp;');
        $oVWM->GETFormRowStyleUPDAdmin4('Formularname',$Daten['FRMNAME'],'frmname','X(50)',50);
        $oVWM->GETFormRowStyleINSARRAdmin4('Projekt',$Daten['PRJNAME'],'prjname',$arrProjekt);
        $oVWM->GETFormRowStyleINSARRAdmin4('Objektart',$Daten['OBANAME'],'obaname',$arrObjektart);
    }
}
}
}
$soVWM->GETFormularFooterUserverwaltungUPD();
// INSERTMODE
} elseif ($in_frmstatus == 'NEU'){
    // INIT - Projektnamen holen
    $queryProjekte = "SELECT PRJNAME "
                    ."FROM   afn_projekt ";
    if ($arrCookie[2] != 'ADMIN')
    {
        $queryProjekte = $queryProjekte."WHERE PRJNAME <> 'ADMIN' ";
    } elseif ($arrCookie[2] != 'PROJ' && $arrCookie[2] != 'ADMIN') {
        $queryProjekte = $queryProjekte."WHERE TID IN (SELECT PROJEKTID "
                                ."FROM   adm_user_rechte "
                                ."WHERE  USERID = $arrCookie[0]);"
    }
}
$arrProjekt = $soDBM->GETSelectSQLSelect($queryProjekte);

// INIT - Objektartenamen holen
$queryOBA = "SELECT OBANAME "
            ."FROM   afn_objektart ";
$arrObjektart = $soDBM->GETSelectSQLSelect($queryOBA);

// HTML-Seite Anfang
$soLTM->GETFORMHeader();

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltungUPD('neues Formular anlegen',$file,'frmtid',$in_frmrtid,'INSERT');

if (is_array($arrFehlermeldung))
{
    $soVWM->GETFormRowStyle2('&nbsp;');
    $count = count($arrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $soVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
    }
    $soVWM->GETFormRowStyle2('&nbsp;');
    $soVWM->GETFormRowStyleUPDAdmin4('Formularname',$in_frmname,'frmname','X(50)',50);
    $soVWM->GETFormRowStyleINSARRAdmin4('Projekt',$in_prjname,'prjname',$arrProjekt);
    $soVWM->GETFormRowStyleINSARRAdmin4('Objektart',$in_obaname,'obaname',$arrObjektart);
} else {
    $soVWM->GETFormRowStyle2('&nbsp;');
    $soVWM->GETFormRowStyleUPDAdmin4('Formularname',"','frmname','X(50)',50);
    $soVWM->GETFormRowStyleINSARRAdmin4('Projekt',"','prjname',$arrProjekt);
    $soVWM->GETFormRowStyleINSARRAdmin4('Objektart',"','obaname',$arrObjektart);
}
$soVWM->GETFormularFooterUserverwaltungUPD();

// READMODE (auch nach UPDATE!)
} else {
    // HTML-Seite Anfang
    $soLTM->GETFORMHeader();

    // GET Filter-Daten
    $GETqueryFILTER = "SELECT TID AS TID, FRMNAME "
                    ."FROM   afn_h_formular "
                    ."ORDER BY FRMNAME ";
    $soDBM->GETSelectSQL($GETqueryFILTER,'TID,NAME',$arrFilter);
    // GET Filter-Daten
    $GETqueryFILTER2 = "SELECT DISTINCT p.TID, p.PRJNAME "
                    ."FROM   afn_projekt p, "
                    ."      afn_h_formular h "
                    ."WHERE  p.tid = h.PRJID "
                    ."ORDER BY p.PRJNAME ";
    $soDBM->GETSelectSQL($GETqueryFILTER2,'TID,NAME',$arrFilter2);

    $soVWM->GETFormular2Filter('Projekt','fltprojekt',$arrFilter2,'Formular','fltformular',$arrFilter);

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
// GET DB-Daten
$query = "SELECT f.TID, f.FRMNAME, pr.PRJNAME, oa.OBANAME "
        ."FROM   afn_h_formular f, "
        ."       afn_projekt pr, "
        ."       afn_objektart oa "
        ."WHERE  pr.TID = f.PRJID "
        ."AND    oa.TID = f.OBAID ";

// Filterdaten einbinden
if ($in_fltprojekt != '0')
{
    $query = $query."AND f.PRJID = '$in_fltprojekt' ";
} elseif ($in_fltformular != '0') {
    $query = $query."AND f.TID = '$in_fltformular' ";
}
$query = $query."ORDER BY FRMNAME ";
$soDBM->GETSelectSQL($query,'FRMID,FRMNAME,PRJNAME,OBANAME',$arrFormular);

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltung('Formularverwaltung',$file);
if (is_array($arrFormular))
{
    foreach ($arrFormular as $Daten)
    {
        if ($counter > 0)
        {
            $soVWM->GETFormRowStyle1('&nbsp;');
        }
        $PRUEFProjekt = $pruefDBM->PRUEFProjTable($arrCookie[0],$Daten['FRMID'],$table);
        if ($PRUEFProjekt > 0 || $arrCookie[2] == 'ADMIN')
        {
            $soVWM->GETFormRowStyleUPDATE2($Daten['FRMNAME'],$file,$Daten['FRMID'],'frmtid');
        } else {
            $soVWM->GETFormRowStyleGESPERRT2($Daten['FRMNAME']);
        }
        $soVWM->GETFormRowStylePARFORMAT4('Projekt',$Daten['PRJNAME'],$Daten['FRMID']);
        $soVWM->GETFormRowStylePARFORMAT4('Objektart',$Daten['OBANAME'],$Daten['FRMID']);
        $soFile = $frFile.'?frmtid='.$Daten['FRMID'].'&frmname='.$Daten['FRMNAME'];
        $soVWM->GETFormRowStyle6('Formularzeilen','Formularzeilen',$Daten['FRMID'],$soFile);
        $soFile = "";
        $counter++;
    }
    $soVWM->GETFormularFooter();
    // HTML-Seite Ende
    $soLTM->GETFooter();
}
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihren Administrator in Verbindung!";
}
} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihren Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://'.$_SERVER['SERVER_NAME'].'/iFlora'>Startseite</a> oder aktivieren sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
    $soLTM->GETHeader();
    $soVWM->GETLabel("iFlora Formularverwaltung","H1");
    $soVWM->GETTextfield($ERROR);
    $soLTM->GETFooter();
}

?>
```

9.3.1.2.11 Controller – „Formularzeile“ (ADMINFormFormRow.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../././config_iflora/sqladmin.php.inc');
include_once('../././config_iflora/sqladmin.php.inc');
include_once('../././config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht", "PROJ");
$ERROR = "OK";
$DBERROR = "OK";
$SarrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$stable = 'afn_h_formular';

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$sin_frmstatus=$_REQUEST['frmstatus'];} else {$sin_frmstatus = '0'};
// FormRowVariablen
if($_REQUEST['fmrtid'] != ""){$sin_fmrtid=$_REQUEST['fmrtid'];} else {$sin_fmrtid = 0};
if($_REQUEST['frmtid'] != ""){$sin_frmtid=$_REQUEST['frmtid'];} else {$sin_frmtid = 0};
if($_REQUEST['frmname'] != ""){$sin_frmname=$_REQUEST['frmname'];} else {$sin_frmname = 0};
// Pflicht
if($_REQUEST['fmrposition'] != ""){$sin_fmrposition=$_REQUEST['fmrposition'];} else {$sin_fmrposition = 0};
if($_REQUEST['fmrstyle'] != ""){$sin_fmrstyle=$_REQUEST['fmrstyle'];} else {$sin_fmrstyle = 0};
// optional
if($_REQUEST['fmrlabel'] != ""){$sin_fmrlabel=$_REQUEST['fmrlabel'];} else {$sin_fmrlabel = ""};
if($_REQUEST['parlabel'] != ""){$sin_parlabel=$_REQUEST['parlabel'];} else {$sin_parlabel = 'LEER'};
// Formate
if($_REQUEST['frmparlabel'] != ""){$sin_frmparlabel=$_REQUEST['frmparlabel'];} else {$sin_frmparlabel = 0};
if($_REQUEST['frmfmrposition'] != ""){$sin_frmfmrposition=$_REQUEST['frmfmrposition'];} else {$sin_frmfmrposition = 0};
if($_REQUEST['frmfmrstyle'] != ""){$sin_frmfmrstyle=$_REQUEST['frmfmrstyle'];} else {$sin_frmfmrstyle = 0};
if($_REQUEST['frmfmrlabel'] != ""){$sin_frmfmrlabel=$_REQUEST['frmfmrlabel'];} else {$sin_frmfmrlabel = 0};
// Filter
if($_REQUEST['fltformrow'] != ""){$sin_fltformrow=$_REQUEST['fltformrow'];} else {$sin_fltformrow = '0'};
// Radiobutton
if($_REQUEST['frlabel'] != ""){$sin_frlabel=$_REQUEST['frlabel'];} else {$sin_frlabel = '0'};

if($sin_parlabel == "0"){ $sin_parlabel='LEER';};

$SarrFormular = NULL;
$SarrFtRohData = NULL;
$SarrFilter = NULL;
$SarrFilterhidden = NULL;
$SarrHidden = NULL;
$Scounter = 0;
$Sfile = "../ADMINFormFormRow.php";
$Sradio = "frlabel";

$SarrProjekt = NULL;
$SarrWLKlasse = NULL;
$SarrISINArt = NULL;
$SarrObjArt = NULL;

// INIT AdminTool Objekt
$SQLite = "../././config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$SovWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$Soltm = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $SarrCookie = explode(';', $Cookie);
        $Saktiv = $pruefDBM->GETUserAktiv($SarrCookie[0]);
        if ($Saktiv == "aktiv")

```

```

{
$PRUEFBerechtigung = $spruefDBM->PRUEFBerechtigung(formularrecht,$sarrCookie[2]);
if ($PRUEFBerechtigung > 0)
{
    if ($sarrCookie[1] == 'ADMIN')
    {
        $user = $admin_user;
        $passwd = $admin_passwd;
    } elseif ($sarrCookie[1] == 'WORK') {
        $user = $work_user;
        $passwd = $work_passwd;
    } else {
        $user = $data_user;
        $passwd = $data_passwd;
    }
}

// INIT DBManager Objekt
SoDBM = new DBManager($database,$sql_server,$user,$passwd,$sql_db,$schema);

// ANWENDUNG
// 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
// 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
// UPDATE
if ($in_frmstatus == 'UPDATE')
{
    if ($sarrCookie[2] == 'ADMIN')
    {
        $DBchange = 'true';
    } else {
        $PREUFqueryDS_USER = "SELECT DS_USER "
                                ."FROM adm_h_formular "
                                ."WHERE TID = '$in_fmrtid' ";
        $PRUEFDs_user = $SoDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
        // $PRUEFProjekt = $spruefDBM->PRUEFProjekt($sarrCookie[0],$in_usrtid);
        $PRUEFProjekt = $spruefDBM->PRUEFProjTable($sarrCookie[0],$in_partid,$stable);
        // Ein Projektleiter darf alle Benutzer bearbeiten, die seinem Projekt zugeordnet oder von ihm angelegt sind!
        if ($PRUEFProjekt > 0 || $sarrCookie[0] == $PRUEFDs_user)
        {
            $DBchange = 'true';
        } else {
            $DBchange = 'false';
            $sarrFehlermeldung[] = "Nur ein Projektleiter darf die Formulare seines Projektes bearbeiten! Das trifft auf dieses Formular nicht
zu!";
        }
        $in_frmstatus = "EDIT";
    }
}
if ($DBchange == 'true')
{
    if ($in_fmrtid > '0' && $in_fmrtid != '0' && $in_fmrtid != '0')
    {
        # Prüfen, ob Datensatz vorhanden und eindeutig ist!
        $PREUFqueryUPDATE = "SELECT count(*) "
                                ."FROM afn_h_formrow "
                                ."WHERE TID = '$in_fmrtid' ";
        $PRUEFDs = $SoDBM->GETSelectSQLDatum($PREUFqueryUPDATE);
        if ($PRUEFDs == 1)
        {
            $ds = 'true';
        } else {
            $ds = 'false';
            $sarrFehlermeldung[] = "Die ausgewählte Formularzeile ist nicht (mehr) in der Datenbank vorhanden!";
        }
    }

    // Formatprüfung der eingegebenen Werte
    $PRUEFfmrtid = $SoVWM->PRUEFFormat($in_fmrtid,$in_fmrtid);
    $PRUEFfmrtid = $SoVWM->PRUEFFormat($in_fmrtid,$in_fmrtid);
    $PRUEFfmrtid = $SoVWM->PRUEFFormat($in_fmrtid,$in_fmrtid);

    if ($ds == 'true' &&
        $DBERROR == 'OK' &&
        $PRUEFfmrtid == 'true' &&
        $PRUEFfmrtid == 'true' &&
        $PRUEFfmrtid == 'true')
    {

```

```

$queryUPDATE = "UPDATE afn_h_formrow SET DS_DATE = $DBDateFkt, "
                ."DS_USER = '$arrCookie[0]', "
                ."PARID = (SELECT TID "
                ."      FROM afn_parameter "
                ."      WHERE PARNR = substring('$in_parlabel',1,4)), "
                ."FMRLABEL = '$in_fmrlabel', "
                ."FMRPOSITION = '$in_fmrposition', "
                ."FMRSTYLE = '$in_fmrstyle' "
                ."WHERE TID = '$in_fmrtid' ";

$soDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
    if ($PRUEFfmrlabel == 'false')
    {
        $arrFehlermeldung[] = "<b>&Uuml;berschrift:</b> Die &Uuml;berschrift darf nicht l&auml;nger als 100 Buchstaben sein!";
    }
    if ($PRUEFfmrposition == 'false')
    {
        $arrFehlermeldung[] = "<b>Position:</b> Die Position der Formularzeile muss 4 stellig sein!";
    }
    if ($PRUEFfmrstyle == 'false')
    {
        $arrFehlermeldung[] = "<b>Layoutebene:</b> Die Zahl der Layoutebene darf nicht h&ouml;her als 4 stellig sein!";
    }
    $in_fmrtid = "EDIT";
}
} else {
    if ($in_fmrposition == '0')
    {
        $arrFehlermeldung[] = "<b>Position:</b> Dieses Feld ist ein Pflichtfeld und muss bef&uuml;llt werden!";
        $in_fmrposition = "";
    }
    if ($in_fmrstyle == '0')
    {
        $arrFehlermeldung[] = "<b>Layoutebene:</b> Dieses Feld ist ein Pflichtfeld und muss bef&uuml;llt werden!";
        $in_parnr = "";
    }
    $in_fmrstyle = "EDIT";
}
}
}
// INSERT
if ($in_fmrtid == 'INSERT')
{
    if ($arrCookie[2] == 'ADMIN' || $arrCookie[2] == 'PROJ')
    {
        $DBchange = 'true';
    } else {
        $DBchange = 'false';
        $arrFehlermeldung[] = "<b>KEIN RECHT:</b> Ein Projektleiter darf nur für sein Projekt einen neuen Parameter anlegen!";
        $in_fmrtid = "NEU";
    }
    if ($DBchange == 'true')
    {
        if ($in_fmrtid != '0' && $in_fmrposition != '0' && $in_fmrstyle != '0')
        {
            // Formatprüfung der eingegebenen Werte
            $PRUEFfmrlabel = $soVWM->PRUEFFormat($in_fmfmrlabel,$in_fmrlabel);
            $PRUEFfmrposition = $soVWM->PRUEFFormat($in_fmfmrposition,$in_fmrposition);
            $PRUEFfmrstyle = $soVWM->PRUEFFormat($in_fmfmrstyle,$in_fmrstyle);

            if ($DBERROR == 'OK' &&
                $PRUEFfmrlabel == 'true' &&
                $PRUEFfmrposition == 'true' &&
                $PRUEFfmrstyle == 'true')
            {
                $queryINSERT = "INSERT INTO afn_h_formrow SET DS_DATE = $DBDateFkt, "
                                ."DS_USER = '$arrCookie[0]', "
                                ."FRMID = '$in_fmrtid', "
                                ."PARID = (SELECT TID "
                                ."      FROM afn_parameter "
                                ."      WHERE PARNR = substring('$in_parlabel',1,4)), "
                                ."FMRLABEL = '$in_fmrlabel', "
                                ."FMRPOSITION = '$in_fmrposition', "
                                ."FMRSTYLE = '$in_fmrstyle' ";
            }
        }
    }
}

```

```

SoDBM->GETUpdateSQLDatum($queryINSERT);
} else {
if ($PRUEFfmrlabel == 'false')
{
    $arrFehlermeldung[] = "<b>&Uuml;berschrift:</b> Die &Uuml;berschrift darf nicht l&auml;nger als 100 Buchstaben sein!";
}
if ($PRUEFfmrposition == 'false')
{
    $arrFehlermeldung[] = "<b>Position:</b> Die Position der Formularzeile muss 4 stellig sein!";
}
if ($PRUEFfmrstyle == 'false')
{
    $arrFehlermeldung[] = "<b>Layoutebene:</b> Die Zahl der Layoutebene darf nicht h&ouml;her als 4 stellig sein!";
}
$in_frmstatus = "NEU";
}
} else {
if ($in_frmrposition == '0')
{
    $arrFehlermeldung[] = "<b>Position:</b> Dieses Feld ist ein Pflichtfeld und muss bef&uuml;llt werden!";
    $in_frmrposition = "";
}
if ($in_frmstyle == '0')
{
    $arrFehlermeldung[] = "<b>Layoutebene:</b> Dieses Feld ist ein Pflichtfeld und muss bef&uuml;llt werden!";
    $in_parnr = "";
}
if ($in_frmtid == '0')
{
    $arrFehlermeldung[] = "<b>Formularfehler:</b> Es gibt ein Problem mit dem auge&auml;hlten Formular! Falls diese
Fehlermeldung mehrmals hintereinander auftritt, informieren sie Bitte ihren Administrator!";
    $in_frmrposition = "";
}
$in_frmstatus = "NEU";
}
}
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
// INIT - Parameter holen
$queryParameter = "SELECT concat(PARNR,'-',PARNAME) "
                ."FROM   afn_parameter "
                ."WHERE  PARNR <> 'LEER' "
                ."ORDER BY PARNR ";
$arrParameter = SoDBM->GETSelectSQLSelect($queryParameter);

// GET DB-Daten
$query = "SELECT r.TID, r.FMRLABEL, concat(p.PARNR,'-',p.PARNAME), "
        ."r.FMRPOSITION, r.FMRSTYLE, substring(p.PARNR,1,1), p.PARNR, r.DS_USER, p.TID "
        ."FROM   afn_h_formrow r "
        ."      LEFT JOIN afn_parameter p ON p.TID = r.PARID "
        ."WHERE  r.TID = $in_frmtid ";

SoDBM-
>GETSelectSQL($query,'FMRID,FMRLABEL,PARAMETER,FMRPOSITION,FMRSTYLE,PARTYP,PARNR,DS_USER,PARID',$arrFo
rmular);

// HTML-Seite Anfang
SoLTM->GETFORMHeader();

$hiddenhash["NAME"] = "frmtd";
$hiddenhash["TID"] = $in_frmtd;
$arrHidden[] = $hiddenhash;
$hiddenhash["NAME"] = "frmname";
$hiddenhash["TID"] = $in_frmname;
$arrHidden[] = $hiddenhash;
$hiddenhash["NAME"] = "fmrtid";
$hiddenhash["TID"] = $in_fmrtid;
$arrHidden[] = $hiddenhash;

// Formular zusammenstellen
SoVWM->GETFormularHeaderUFormUPD('Formularzeile bearbeiten',$file,'UPDATE',$arrHidden);

if (is_array($arrFormular))
{

```

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

247/322

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
$arrParameter = $oDBM->GETSelectSQLSelect($queryParameter);

// HTML-Seite Anfang
$oLTM->GETFORMHeader();

$shiddenhash["NAME"] = "frmtid";
$shiddenhash["TID"] = $in_frmtid;
$arrHidden[] = $shiddenhash;
$shiddenhash["NAME"] = "frmname";
$shiddenhash["TID"] = $in_frmname;
$arrHidden[] = $shiddenhash;

// Formular zusammenstellen
$oVWM->GETFormularHeaderUFormUPD('neuen Formularzeile anlegen',$file,'INSERT',$arrHidden);

if (is_array($arrFehlermeldung))
{
    $oVWM->GETFormRowStyle2('&nbsp;');
    $count = count($arrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
    }
    $oVWM->GETFormRowStyle2($in_frmname);
    $oVWM->GETFormRowStyle3('Pflichtfelder');
    $oVWM->GETFormRowStyleUPDAdmin4('Position',$in_fmposition,'fmposition','F+(4)',4);
    $oVWM->GETFormRowStyleUPDAdmin4('Layoutebene',$in_fmstyle,'fmstyle','F+(1)',1);
    $oVWM->GETFormRowStyle3('Auswahl - Pflichtfeld');
    $oVWM->GETFormRowStyleINSARRAdminRADIO4('Parameter',$in_parid,'parlabel',$arrParameter,$radio,"");
    $oVWM->GETFormRowStyleUPDAdminRADIO4('&Uuml;berschrift',$in_fmrlabel,'fmrlabel','X(130)',70,$radio,"");
} else {
    $oVWM->GETFormRowStyle2($in_frmname);
    $oVWM->GETFormRowStyle3('Pflichtfelder');
    $oVWM->GETFormRowStyleUPDAdmin4('Position','','fmposition','F+(4)',4);
    $oVWM->GETFormRowStyleUPDAdmin4('Layoutebene','','fmstyle','F+(1)',1);
    $oVWM->GETFormRowStyle3('Auswahl - Pflichtfeld');
    $oVWM->GETFormRowStyleINSARRAdminRADIO4('Parameter','','parlabel',$arrParameter,$radio,"");
    $oVWM->GETFormRowStyleUPDAdminRADIO4('&Uuml;berschrift','','fmrlabel','X(130)',70,$radio,"");
}
$oVWM->GETFormularFooterUserverwaltungUPD();

// READMODE (auch nach UPDATE/INSERT!)
} else {
    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // GET Filter-Daten
    $GETqueryFILTER = "SELECT distinct r.FMRPOSITION, concat(p.PARNR,'-',p.PARNAME) AS PARNAME, "
        . "r.FMRLABEL, substring(p.PARNR,1,1), r.FMRSTYLE "
        . "FROM afn_h_formrow r "
        . "LEFT JOIN afn_parameter p ON p.TID = r.PARID "
        . "WHERE r.FRMID = $in_frmtid "
        . "ORDER BY PARNAME ";
    $oDBM->GETSelectSQL($GETqueryFILTER,'TID,PARNAME,FMRLABEL,PARTYP,FMRSTYLE',$arrFltRohData);
    if (is_array($arrFltRohData))
    {
        foreach ($arrFltRohData as $FltRohData)
        {
            $shash["TID"] = $FltRohData["TID"];
            if($FltRohData["PARTYP"] == "L")
            {
                $shash["NAME"] = $FltRohData["FMRLABEL"];
            } else {
                $shash["NAME"] = $FltRohData["PARNAME"];
            }
            $arrFilter[] = $shash;
        }
    }
    $shiddenhash["NAME"] = "frmtid";
    $shiddenhash["TID"] = $in_frmtid;
    $arrFilterhidden[] = $shiddenhash;
    $shiddenhash["NAME"] = "frmname";
    $shiddenhash["TID"] = $in_frmname;
    $oVWM->GETFormularFilter('Formularzeile','fltformrow',$arrFilter,$arrFilterhidden);
}
```



```
// GET DB-Daten
$query = "SELECT r.TID, r.FMRLABEL, concat(p.PARNR,'-',p.PARNAME), "
        ."r.FMRPOSITION, r.FMRSTYLE, substring(p.PARNR,1,1), p.PARNR "
        ."FROM   afn_h_formrow r "
        ."      LEFT JOIN afn_parameter p ON p.TID = r.PARID "
        ."WHERE  r.FRMID = $in_frmid ";

// Filterdaten einbinden
if ($in_fltformrow != '0')
{
    $query = $query."AND r.FMRPOSITION = '$in_fltformrow' ";
}
$query = $query."ORDER BY FMRPOSITION ";
$soDBM-
>GETSelectSQL($query,'FMRID,FMRLABEL,PARNAME,FMRPOSITION,FMRSTYLE,PARNR,TYP,PARNR',$sarrFormular);

$counter = 1;

// OFormular zusammenstellen
$soVWM->GETFormularHeaderOForm('Formularzeilenverwaltung',$file,$sarrFilterhidden);
if (is_array($sarrFormular))
{
    $PRUEFProjekt = $pruefDBM->PRUEFProjTable($sarrCookie[0],$in_frmid,$stable);
    $soVWM->GETFormRowStyle2($in_frmname);
    foreach ($sarrFormular as $Daten)
    {
        if ($PRUEFProjekt > 0 || $sarrCookie[2] == 'ADMIN')
        {
            $soVWM->GETFormRowStyleUPDATE3a('Zeile: '.$counter,$file,$Daten['FMRID'],'fmrid',$sarrFilterhidden);
        } else {
            $soVWM->GETFormRowStyleGESPERRT3('Zeile: '.$counter);
        }
        $soVWM->GETFormRowStylePARFORMAT4('Position',$Daten['FMRPOSITION'],$Daten['FMRID']);
        if ($Daten['PARNR'] == "L" && $Daten['PARNR'] != "LEER")
        {
            $soVWM->GETFormRowStylePARFORMAT4('Parameter',$Daten['PARNAME'],$Daten['FMRID']);
            $soVWM->GETFormRowStylePARFORMAT4('Überschrift',$Daten['FMRLABEL'],$Daten['FMRID']);
        } else {
            if ($Daten['FMRLABEL'] == "")
            {
                $soVWM->GETFormRowStylePARFORMAT4('Parameter',$Daten['PARNAME'],$Daten['FMRID']);
            } else {
                $soVWM->GETFormRowStylePARFORMAT4('Überschrift',$Daten['FMRLABEL'],$Daten['FMRID']);
            }
        }
        $soVWM->GETFormRowStylePARFORMAT4('Layoutebene',$Daten['FMRSTYLE'],$Daten['FMRID']);
        $counter++;
    }
}
$soVWM->GETFormularFooter();
// HTML-Seite Ende
$soLTM->GETFooter();
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihrem Administrator in Verbindung!";
}
} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihrem Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle
    Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://".$_SERVER['SERVER_NAME']."'>iFlora</a>Startseite</a> oder aktivieren
    sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
    $soLTM->GETHeader();
    $soVWM->GETLabel("iFlora Formularverwaltung","H1");
    $soVWM->GETTextfield($ERROR);
    $soLTM->GETFooter();
}
```

>

9.3.1.2.12 Controller – „<Objekt>“ (ADMINSelObjekt.php)

<?php

```

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['prjtid'] != ""){$sin_prjtid=$_REQUEST['prjtid'];} else {$sin_prjtid = 0;};
if($_REQUEST['prjname'] != ""){$sin_prjname=$_REQUEST['prjname'];} else {$sin_prjname = 0;};
if($_REQUEST['obtid'] != ""){$sin_obtid=$_REQUEST['obtid'];} else {$sin_obtid = 0;};
if($_REQUEST['obaname'] != ""){$sin_obaname=$_REQUEST['obaname'];} else {$sin_obaname = 0;};

$HPueberschrift = "Auswahl $sin_obaname";
$lbSelect = "Auswahl";

$arrSQLObjekt = NULL;
$arrSQL = NULL;

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$oVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

// HTML Header
$oLTM->GETHeader();

// Projektauswahl
if ($sin_prjtid > 0)
{
    // INIT SelectObjekt
    $SQLQueryObjekt = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM   afn_objekt o "
        ."WHERE  o.OBAID = '$sin_obtid' "
        ."AND    o.PRJID = '$sin_prjtid' "
        ."ORDER BY o.OBJNAME";

    // GET DB-Daten
    $oDBM->GETSelectSQL($SQLQueryObjekt,"TID,NAME",$arrSQLObjekt);

    // Layoutbausteine zusammenstellen
    $oVWM->GETLabel($HPueberschrift,"H1");

    if (is_array($arrSQLObjekt))
    {
        foreach ($arrSQLObjekt as $Daten)
        {
            // bei ca.3000 Datensätzen gibt es ein Problem mit Objekten (=> Speicher?)
            print '<a
href="ADMINFormStammdata.php?objtid='.$Daten['TID'].'&objname='.$Daten['NAME'].'&obtid='.$sin_obtid.'&obaname='.$sin_obaname.'
">'.$Daten['NAME'].'</a><br />';
        }
    }
    else {
        $parameter = "prjtid";
        $SQLQuery = "SELECT  DISTINCT p.TID, p.PRJNAME "
            ."FROM   afn_objekt o, "
            ."       afn_projekt p "
            ."WHERE  p.TID = o.PRJID "
            ."AND    o.OBAID = $sin_obtid ";
        $labelSELECT = "PROJEKT:";

        // GET DB-Daten
        $oDBM->GETSelectSQL($SQLQuery,"TID,NAME",$arrSQL);

        // HiddenArray füllen
        $arrSELHidden["obtid"] = "$sin_obtid";

```

```

    $arrSELHidden["obaname"] = "$in_obaname";

    // Layoutbausteine zusammenstellen
    $oVWM->GETLabel($HPueberschrift,"H1");
    $oVWM->GETIDTable(&$arrHISTDaten);
    $oVWM->GETLabel($labelSELECT,"H3");
    $oVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
}
$oLTM->GETFooter();

?>

```

9.3.1.2.13 Controller – „Objektbeziehungen“ (ADMINFormObjektBez.php)

```

<?php

// BspAbfrage
//select  isinoba.obaname,isa.isaname,oba.obaname
//from    afn_isinArt isa,
//        afn_objektart oba,
//        afn_isinobjart isinobabez,
//        afn_objektart isinoba
//where   isa.tid = isinobabez.isaid => IS-IN THEMA ("PFAD")
//and     oba.tid = isinobabez.obaid => Objektart Unterebene
//and     isinoba.tid = isinobabez.isinobaid => Objektart Übergeordnete Ebene

// INCLUDE
include_once('..lib/DBManager.php');
include_once('..lib/VIEWManager.php');
include_once('..lib/LAYOUTManager.php');
include_once('..../config_iflora/sqladmin.php.inc');
include_once('..../config_iflora/sqladmin.php.inc');
include_once('..../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht","TEMP");
$ERROR = "OK";
$DBERROR = "OK";
$arrFehlermeldung = NULL;
$DBDateFkt = 'CURDATE()';
$table = 'afn_objektart';
$arrHidden = NULL;

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$in_frmstatus=$_REQUEST['frmstatus'];} else {$in_frmstatus = '0';}
// TID
if($_REQUEST['obaid'] != ""){$in_obaid=$_REQUEST['obaid'];} else {$in_obaid = 0;};
if($_REQUEST['isinobaid'] != ""){$in_isinobaid=$_REQUEST['isinobaid'];} else {$in_isinobaid = 0;};
// Pflicht
if($_REQUEST['obaname'] != ""){$in_obaname=$_REQUEST['obaname'];} else {$in_obaname = 0;};
if($_REQUEST['isaname'] != ""){$in_isaname=$_REQUEST['isaname'];} else {$in_isaname = 0;};
if($_REQUEST['isinobaname'] != ""){$in_isinobaname=$_REQUEST['isinobaname'];} else {$in_isinobaname = 0;};
// optional
if($_REQUEST['isinobakommentar'] != ""){$in_isinobakommentar=$_REQUEST['isinobakommentar'];} else {$in_isinobakommentar =
"";};
// Formate
if($_REQUEST['frminisobakommentar'] != ""){$in_frminisobakommentar=$_REQUEST['frminisobakommentar'];} else
{$in_frminisobakommentar = 0;};
// Filter
if($_REQUEST['fltobaid'] != ""){$in_ftobaid=$_REQUEST['fltobaid'];} else {$in_ftobaid = '0';};
if($_REQUEST['fltisinobaid'] != ""){$in_ftisinobaid=$_REQUEST['fltisinobaid'];} else {$in_ftisinobaid = '0';};
if($_REQUEST['fltisaid'] != ""){$in_ftisaid=$_REQUEST['fltisaid'];} else {$in_ftisaid = 0;};

$arrFormular = NULL;
$arrFilter1 = NULL;
$arrFilter2 = NULL;
$arrFilter3 = NULL;
$counter = 0;
$file = "../ADMINFormISINOBA.php";

$arrProjekt = NULL;
$arrWLKlasse = NULL;
$arrISINArt = NULL;
$arrObjArt = NULL;

```

```
// INIT AdminTool Objekt
$SQLite = "../././config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$soVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$soLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);
    if ($Cookie <> "ERROR")
    {
        $sarrCookie = explode(';', $Cookie);
        $aktiv = $pruefDBM->GETUserAktiv($sarrCookie[0]);
        if ($aktiv == "aktiv")
        {
            $PRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht, $sarrCookie[2]);
            if ($PRUEFBerechtigung > 0)
            {
                if ($sarrCookie[1] == 'ADMIN')
                {
                    $user = $admin_user;
                    $passwd = $admin_passwd;
                } elseif ($sarrCookie[1] == 'WORK') {
                    $user = $work_user;
                    $passwd = $work_passwd;
                } else {
                    $user = $data_user;
                    $passwd = $data_passwd;
                }
            }

            // INIT DBManager Objekt
            $soDBM = new DBManager($database,$sql_server,$user,$passwd,$sql_db,$schema);

            // INIT - Objektartnamen holen
            $queryObjektart = "SELECT OBANAME "
                ."FROM afn_objektart ";
            $sarrObjektart = $soDBM->GETSelectSQLSelect($queryObjektart);

            // ANWENDUNG
            // 1.) allfällige Daten werden eingespielt (INSERT/UPDATE)
            // 2.) Je nach STATUS wird jeweilige Formular verwendet (READMODE, EDITMODE, INSERTMODE)
            // UPDATE
            if ($in_frmstatus == 'UPDATE')
            {
                if ($sarrCookie[2] == 'ADMIN')
                {
                    $DBchange = 'true';
                } else {
                    $PREUFqueryDS_USER = "SELECT DS_USER "
                        ."FROM adm_username "
                        ."WHERE TID = '$in_usrtid' ";
                    $PRUEFds_user = $soDBM->GETSelectSQLDatum($PREUFqueryDS_USER);
                    $PRUEFProjekt = $pruefDBM->PRUEFProjTable($sarrCookie[0], $in_prjtid, $stable);
                    if ($PRUEFProjekt > 0 || $sarrCookie[0] == $PRUEFds_user)
                    {
                        $DBchange = 'true';
                    } else {
                        $DBchange = 'false';
                        $sarrFehlermeldung[] = "Nur ein Projektleiter darf die IS-IN Objektartbeziehungen bearbeiten!";
                        $in_frmstatus = "EDIT";
                    }
                }
            }
            if ($DBchange == 'true')
            {
                if ($in_isinobatid > '0' && $in_obaname > '0' && $in_isinobaname > '0')
                {
                    # Prüfen, ob Datensatz vorhanden und eindeutig ist!
                    $PRUEFqueryUPDATE = "SELECT count(*) "
                        ."FROM afn_isinobjart "
                        ."WHERE TID = '$in_isinobatid' ";
                    $PRUEFds = $soDBM->GETSelectSQLDatum($PRUEFqueryUPDATE);
                    if ($PRUEFds == 1)

```

```

{
  $ds = 'true';
} else {
  $ds = 'false';
  $arrFehlermeldung[] = "Die ausgewählte Objektart ist nicht (mehr) in der Datenbank vorhanden!";
}
# Prüfen, ob "IS-IN Objektartbeziehung" (UNIQUE INDEX) eindeutig ist!
# d.h.: OBAID&ISINOBAID&ISAID darf noch nicht vorkommen (ausser beim geänderten Datensatz)
$PRUEFqueryISINOBAID = "SELECT count(*) "
                        ."FROM afn_isinobjart "
                        ."WHERE OBAID = (SELECT TID "
                        ."      FROM afn_objektart "
                        ."      WHERE OBANAME = '$in_obaname') "
                        ."AND ISINOBAID = (SELECT TID "
                        ."      FROM afn_objektart "
                        ."      WHERE OBANAME = '$in_isinobaname') "
                        ."AND ISAID = (SELECT ISAID "
                        ."      FROM afn_isinobjart "
                        ."      WHERE TID = '$in_isinobaname') "
                        ."AND TID <> '$in_isinobatid' ";
$PRUEFp = $oDBM->GETSelectSQLDatum($PRUEFqueryISINOBAID);
if ($PRUEFp == 0)
{
  $pn = 'true';
} else {
  $pn = 'false';
  $arrFehlermeldung[] = "Jede IS-IN Objektartbeziehung muss einen eindeutigen sein - die eingegebene IS-IN
Objektartbeziehung gib es bereits!";
}
// Formatprüfung der eingegebenen Werte
$PRUEFisinobakommentar = $oVWM->PRUEFFormat($in_frmisinobakommentar,$in_isinobakommentar);
if ($ds == 'true' &&
    $pn == 'true' &&
    $DBERROR == 'OK' &&
    $PRUEFisinobakommentar == 'true')
{
  $queryUPDATE = "UPDATE afn_isinobjart SET DS_DATE = $DBDateFkt, "
                ."DS_USER = '$arrCookie[0]', "
                ."DS_STATUS = 'NEU', "
                ."OBAID = (SELECT TID "
                ."      FROM afn_objektart "
                ."      WHERE OBANAME = '$in_obaname'), "
                ."ISINOBAID = (SELECT TID "
                ."      FROM afn_objektart "
                ."      WHERE OBANAME = '$in_isinobaname'), "
                ."ISINOBAKOMMENTAR = '$in_isinobakommentar' "
                ."WHERE TID = '$in_isinobatid' ";
  // $oDBM->GETUpdateSQLDatum($queryUPDATE);
} else {
  if ($PRUEFobakommentar == 'false')
  {
    $arrFehlermeldung[] = "<b>IS-IN Objektartkommentar:</b> Der IS-IN Objektartkommentar darf nicht länger als 255
Buchstaben sein!";
  }

  $in_frmstatus = "EDIT";
}
} else {
  if ($in_isinobaname == '0')
  {
    $arrFehlermeldung[] = "<b>IS-IN Objektart(Ebene 1):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
  }
  if ($in_isinobatid == '0')
  {
    $arrFehlermeldung[] = "Es wurde keine IS-IN Objektartbeziehung ausgewählt!";
  }
  if ($in_obaname == '0')
  {
    $arrFehlermeldung[] = "<b>Objektart(Ebene 2):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
  }
  $in_frmstatus = "EDIT";
}
}
}
// INSERT

```

```

if ($sin_frmstatus == 'INSERT')
{
  if ($sarrCookie[2] == 'ADMIN')
  {
    $DBchange = 'true';
  } else {
    $DBchange = 'false';
    $sarrFehlermeldung[] = "<b>KEIN RECHT:</b> Nur eine Administrator darf eine neue Objektart anlegen!";
    $sin_frmstatus = "NEU";
  }
  if ($DBchange == 'true')
  {
    if ($sin_obaname != '0' && $sin_obakuerzel != '0')
    {
      # Prüfen, ob "IS-IN Objektartbeziehung" (UNIQUE INDEX) eindeutig ist!
      # d.h.: OBAID&ISINOBAID&ISAID darf noch nicht vorkommen (ausser beim geänderten Datensatz)
      $PRUEFqueryISINOBAID = "SELECT count(*) "
        ."FROM afn_isinobjart "
        ."WHERE OBAID = (SELECT TID "
        ."      FROM afn_objektart "
        ."      WHERE OBANAME = '$sin_obaname') "
        ."AND ISINOBAID = (SELECT TID "
        ."      FROM afn_objektart "
        ."      WHERE OBANAME = '$sin_isinobaname') "
        ."AND ISAID = (SELECT ISAID "
        ."      FROM afn_isinobjart "
        ."      WHERE TID = '$sin_isinobaname') "
        ."AND TID <> '$sin_isinobaid' ";

      # Prüfen, ob "obaname" (UNIQUE INDEX) eindeutig ist!
      $PRUEFon = $oDBM->GETSelectSQLDatum($PRUEFqueryISINOBAID);
      if ($PRUEFon == 0)
      {
        $son = 'true';
      } else {
        $son = 'false';
        $sarrFehlermeldung[] = "Jede IS-IN Objektartbeziehung muss einen eindeutigen sein - die eingegebene IS-IN
Objektartbeziehung gib es bereits!";
      }
      // Formatprüfung der eingegebenen Werte
      $PRUEFisinobakommentar = $oVWM->PRUEFFormat($sin_frmisinobakommentar,$sin_isinobakommentar);

      if ($son == 'true' &&
        $DBERROR == 'OK' &&
        $PRUEFisinobakommentar == 'true')
      {
        $queryINSERT = "INSERT INTO afn_isinobjart SET DS_DATE = $DBDateFkt, "
          ."DS_USER = '$sarrCookie[0]', "
          ."DS_STATUS = 'NEU', "
          ."ISAID = (SELECT TID "
          ."      FROM afn_isinart "
          ."      WHERE ISANAME = '$sin_isaname'), "
          ."OBAID = (SELECT TID "
          ."      FROM afn_objektart "
          ."      WHERE OBANAME = '$sin_obaname'), "
          ."ISINOBAID = (SELECT TID "
          ."      FROM afn_objektart "
          ."      WHERE OBANAME = '$sin_isinobaname'), "
          ."ISINOBAKOMMENTAR = '$sin_isinobakommentar' ";

        // $oDBM->GETUpdateSQLDatum($queryINSERT);
      } else {
        if ($PRUEFisinobakommentar == 'false')
        {
          $sarrFehlermeldung[] = "<b>IS-IN Objektartkommentar:</b> Der IS-IN Objektartkommentar darf nicht länger als 255
Buchstaben sein!";
        }
        $sin_frmstatus = "NEU";
      }
    } else {
      if ($sin_isinobaname == '0')
      {
        $sarrFehlermeldung[] = "<b>IS-IN Objektart(Ebene 1):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
      }
      if ($sin_isinobaid == '0')
      {
        $sarrFehlermeldung[] = "Es wurde keine IS-IN Objektartbeziehung ausgewählt!";
      }
    }
  }
}

```

```

    }
    if ($in_obaname == '0')
    {
        $arrFehlermeldung[] = "<b>Objektart(Ebene 2):</b> Dieses Feld ist ein Pflichtfeld und muss befüllt werden!";
    }
    $in_frmstatus = "NEU";
}
}
// EDITMODE
if ($in_frmstatus == 'EDIT')
{
    $SELECTquery = "SELECT    isoa.TID, isa.ISANAME, oba1.OBANAME, "
                    ."        oba2.OBANAME AS ISINOBANAME, isoa.ISINOBAKOMMENTAR, isoa.DS_USER "
                    ."FROM      afn_isinart isa, "
                    ."        afn_isinobjart isoa, "
                    ."        afn_objektart oba1, "
                    ."        afn_objektart oba2 "
                    ."WHERE     isoa.TID = $in_isinobatid "
                    ."AND       isa.TID = isoa.ISAID "
                    ."AND       oba1.TID = isoa.OBAID "
                    ."AND       oba2.TID = isoa.ISINOBAID ";

    $oDBM-
>GETSelectSQL($SELECTquery,'ISINOBAID,ISANAME,OBANAME,ISINOBANAME,ISINOBAKOMMENTAR,DS_USER',$arrFormu
lar);

    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // Formular zusammenstellen
    $oVWM->GETFormularHeaderUserverwaltungUPD('IS-IN-Objektart ändern',$file,'isinobatid',$in_isinobatid,'UPDATE');

    if (is_array($arrFormular))
    {
        foreach ($arrFormular as $Daten)
        {
            if ($arrCookie[2] == 'ADMIN')
            {
                $change = 'true';
            }

            if ($change == 'true')
            {
                if (is_array($arrFehlermeldung))
                {
                    $oVWM->GETFormRowStyle2('&nbsp;');
                    $count = count($arrFehlermeldung);
                    for ($i=0; $i<$count; $i++)
                    {
                        $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
                    }
                    $oVWM->GETFormRowStyle2('&nbsp;');
                    $oVWM->GETFormRowStyle3($Daten['ISANAME']);
                    $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)', $in_isinobaname, 'isinobaname', $arrObjektart);
                    $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)', $in_obaname, 'obaname', $arrObjektart);
                    $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $in_isinobakommentar, 'isinobakommentar', 'X(255)', 255);
                } else {
                    $oVWM->GETFormRowStyle2('&nbsp;');
                    $oVWM->GETFormRowStyle3($Daten['ISANAME']);
                    $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene
1)', $Daten['ISINOBANAME'], 'isinobaname', $arrObjektart);
                    $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)', $Daten['OBANAME'], 'obaname', $arrObjektart);
                    $oVWM-
>GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe', $Daten['ISINOBAKOMMENTAR'], 'isinobakommentar', 'X(255)', 255);
                }
            }
        }
    }
    $oVWM->GETFormularFooterUserverwaltungUPD();
    // INSERTMODE
} elseif ($in_frmstatus == 'NEU'){
    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();

    // INIT - Objektartnamen holen

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
$queryisinart = "SELECT ISANAME "
                ."FROM afn_isinart ";
$arrISINArt = $oDBM->GETSelectSQLSelect($queryisinart);

// Formular zusammenstellen
$oVWM->GETFormularHeaderUserverwaltungUPD('neue IS-IN-Objektart anlegen',$file,'isinobaid',$sin_isinobaid,'INSERT');

if (is_array($arrFehlermeldung))
{
    $oVWM->GETFormRowStyle2('&nbsp;');
    $count = count($arrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
    }
    $oVWM->GETFormRowStyle2('&nbsp;');
    $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Art',$sin_isaname,'isaname',$arrISINArt);
    $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)',$sin_isinobaname,'isinobaname',$arrObjektart);
    $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)',$sin_obaname,'obaname',$arrObjektart);
    $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe',$sin_isinobakommentar,'isinobakommentar','X(255)',255);
} else {
    $oVWM->GETFormRowStyle2('&nbsp;');
    $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Art','',$sin_isaname,$arrISINArt);
    $oVWM->GETFormRowStyleINSARRAdmin4('IS-IN Objektart (Ebene 1)','','$sin_isinobaname',$arrObjektart);
    $oVWM->GETFormRowStyleINSARRAdmin4('Objektart (Ebene 2)','','$sin_obaname',$arrObjektart);
    $oVWM->GETFormRowStyleUPDAdminTA4('Kommentar/Hilfe','','$sin_isinobakommentar','X(255)',255);
}
$oVWM->GETFormularFooterUserverwaltungUPD();

// READMODE (auch nach UPDATE!)
} else {
    // HTML-Seite Anfang
    $oLTM->GETFORMHeader();
    // GET Filter-Daten 1
    $GETqueryFILTER1 = "SELECT DISTINCT MIN(t.TID), isa.ISANAME "
                      ."FROM afn_isinobjart t, "
                      ."      afn_isinart isa "
                      ."WHERE isa.TID = t.ISAID "
                      ."GROUP BY t.ISAID "
                      ."ORDER BY isa.ISANAME ";
    $oDBM->GETSelectSQL($GETqueryFILTER1,'TID,NAME',$arrFilter1);
    // GET Filter-Daten 2
    $GETqueryFILTER2 = "SELECT DISTINCT oba.TID, oba.OBANAME "
                      ."FROM afn_isinobjart t, "
                      ."      afn_objektart oba "
                      ."WHERE oba.TID = t.ISAID "
                      ."ORDER BY oba.OBANAME";
    $oDBM->GETSelectSQL($GETqueryFILTER2,'TID,NAME',$arrFilter2);
    // GET Filter-Daten 3
    $GETqueryFILTER3 = "SELECT DISTINCT oba.TID, oba.OBANAME "
                      ."FROM afn_isinobjart t, "
                      ."      afn_objektart oba "
                      ."WHERE oba.TID = t.ISINOBAID "
                      ."ORDER BY oba.OBANAME ";
    $oDBM->GETSelectSQL($GETqueryFILTER3,'TID,NAME',$arrFilter3);
    $oVWM->GETFormular3Filter('Thema (Pfad)', $ftisaid, $arrFilter1, 'IS-IN Objektart E1', $ftisinobaid, $arrFilter3, 'Objektart (E2)', $ftobaid, $arrFilter2);

    // GET DB-Daten
    $query = "SELECT isin.tid, isa.isaname, obj1.objname, obj2.objname "
            ."FROM afn_isin isin, "
            ."      afn_isinart isa, "
            ."      afn_objekt obj1, "
            ."      afn_objekt obj2 "
            ."WHERE isa.TID = isin.ISAID "
            ."AND obj2.TID = isin.OBJID "
            ."AND obj1.TID = isin.ISINOBJID ";
    // Filterdaten einbinden
    if ($sin_ftobaid != '0')
    {
        // $query = $query."AND isinobabez.OBAID = '$sin_ftobaid' ";
    } elseif ($sin_ftisinobaid != '0') {
        $query = $query."AND isa.ISANAME = (SELECT ISANAME "
                      ."FROM afn_isinart "
                      ."WHERE TID = '$sin_ftisaid') ";
    }
```



```

} elseif ($sin_ftisaid != '0') {
    # TODO => wenn es zu einem ISINART mehrere DS gibt => dann über KUERZEL auswählen!!!
    $query = $query."AND isa.ISANAME = (SELECT ISANAME "
        ."FROM afn_isinart "
        ."WHERE TID = '$sin_ftisaid') ";
}
$query = $query."ORDER BY isa.isaname, obj1.objname ";
$soDBM->GETSelectSQL($query,'ISINID,ISANAME,ISINOBJNAME,OBJNAME',$sarrFormular);

// Formular zusammenstellen
$soVWM->GETFormularHeaderUserverwaltung('Objektbeziehungsverwaltung - IN ARBEIT!!!',$file);
$isaname = '0';
if (is_array($sarrFormular))
{
    foreach ($sarrFormular as $Daten)
    {
        if ($Daten['ISANAME'] != $isaname)
        {
            if ($counter > 0)
            {
                $soVWM->GETFormRowStyle1('&nbsp;');
            }
            $soVWM->GETFormRowStyleUPDATE2($Daten['ISANAME'],$file,$Daten['ISINID'],$isinid);
            $soVWM->GETFormRowStyle3("");
            $isaname = $Daten['ISANAME'];
        }
        // Aus Performancegründen wird das direkt ausgegeben!!!!
        print '<tr id="frmLevel4">
            <td colspan="2">'. $Daten['ISINOBJNAME']. "=>". $Daten['OBJNAME']. '</td>
            <td valign="top"><input type="button" value="Löschen">&nbsp;</td>
        </tr>';
        $counter++;
    }
    $soVWM->GETFormularFooter();
    // HTML-Seite Ende
    $soLTM->GETFooter();
}
} else {
    $ERROR = "Diese Seite darf nur von Administratoren eingesehen und bearbeitet werden!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!<br />Bitte setzen Sie sich mit Ihren Administrator in Verbindung!";
}
} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!<br />Bitte setzen Sie sich mit Ihren Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle Startseite!<br />Bitte gehen sie auf die offizielle <a href='https://".$_SERVER['SERVER_NAME']."'>iFlora'</a> Startseite oder aktivieren sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}

If($ERROR <> OK) // Fehlermeldung
{
    $soLTM->GETHeader();
    $soVWM->GETLabel("iFlora Parameterverwaltung", "H1");
    $soVWM->GETTextfield($ERROR);
    $soLTM->GETFooter();
}

?>

```

9.3.1.2.14 **Controller – „Aufnahme Stammdaten“ (ADMINFormStammdata.php)**

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../lib/config_iflora/sqlpublic.php.inc');

// INIT Variables

```

```
if($_REQUEST['obtid'] != ""){$sin_obtid=$_REQUEST['obtid'];} else {$sin_obtid = 0;};
if($_REQUEST['obaname'] != ""){$sin_obaname=$_REQUEST['obaname'];} else {$sin_obaname = 0;};
if($_REQUEST['objtid'] != ""){$sin_objtid=$_REQUEST['objtid'];} else {$sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$sin_objname=$_REQUEST['objname'];} else {$sin_objname = 0;};
if($_REQUEST['frmid'] != ""){$sin_frmid=$_REQUEST['frmid'];} else {$sin_frmid = 0;};

$dlfile = "OOPExcelStammdata.php";
$pdffile = "OOPpdfStammdata.php";
$textfile = "OOPASCIStammdata.php";

$Afile = "ADMINFormStammdata.php";
$vegfile = "ADMINFormAufnahmedata.php";
$veglabel_xls = "Vegetationdaten";

$SarrFormular = NULL;
$SarrStammdaten = NULL;
$SarrObjektdaten = NULL;
$SarrPOLdaten = NULL;
$SarrisinObjekt = NULL;
$SarrISINobjart = NULL;
$Stamm_id = 0;
$Stamm_value = 'k.a.';

// INIT DBManager Objekt
$SoDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$SoVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$SoLTM = new LAYOUTManager;

// GET DB-Daten
$SoDBM->GETObjektStammdata($sin_obtid,$SarrFormular);

// HTML-Seite Anfang
$SoLTM->GETFORMHeader();

// Formular zusammenstellen
$SoVWM->GETFormularHeader($sin_objname);
if (is_array($SarrFormular))
{
    foreach ($SarrFormular as $Daten)
    {
        if ($Daten['STYLE'] == 2)
        {
            $SoVWM->GETFormRowStyle2($Daten['FORMLABEL']);
        } elseif ($Daten['STYLE'] == 3) {
            $SoVWM->GETFormRowStyle3($Daten['FORMLABEL']);
        } elseif ($Daten['STYLE'] == 4) {
            $SoDBM->GETSelectParamdata($sin_objtid,$Daten['PARID'],$SarrStammdaten);
            if (is_array($SarrStammdaten))
            {
                foreach ($SarrStammdaten as $Stamm)
                {
                    $Stamm_id = $Stamm['TID'];
                    $Stamm_value = $Stamm['STDWERT'];
                }
            }
            $SoVWM->GETFormRowStyle4($Daten['LABEL'],$Stamm_value,$Daten['PARID']);
            $Stamm_id = 0;
            $Stamm_value = 'k.a.';
            $SarrStammdaten = NULL;
        } elseif ($Daten['STYLE'] == 5) {
            $SPATHquery = "SELECT isa.ISAKUERZEL "
                ."FROM   afn_parameter p, afn_isinart isa "
                ."WHERE  p.TID = ".$Daten['PARID']. " "
                ."AND   isa.TID = p.ISAID ";
            $spath = $SoDBM->GETSelectSQLDatum($SPATHquery);

            if ($spath == 'POL')
            {
                $SoDBM->GETIsinartPOL($sin_objtid,$Daten['PARID'],$SarrPOLdaten);
                $gemeinde = 'k.a.';
                $bezirk = 'k.a.';
                $bundesland = 'k.a.';
            }
        }
    }
}
```

```

if (is_array($arrPOLdaten))
{
    foreach ($arrPOLdaten as $Objekt)
    {
        $gemeinde = $Objekt['Gemeinde'];
        $bezirk = $Objekt['Bezirk'];
        $bundesland = $Objekt['Bundesland'];

        $oVWM->GETFormRowStyle4($Daten['LABEL'].' - Gemeinde',$gemeinde,$Daten['PARID']);
        $oVWM->GETFormRowStyle4($Daten['LABEL'].' - Bezirk',$bezirk,$Daten['PARID']);
        $oVWM->GETFormRowStyle4($Daten['LABEL'].' - Bundesland',$bundesland,$Daten['PARID']);
    }
    $arrPOLdaten = NULL;
}
} else {
    $oDBM->GETSelectObjektdaten($sin_objtid,$Daten['PARID'],$arrObjektdaten);
    if (is_array($arrObjektdaten))
    {
        foreach ($arrObjektdaten as $Objekt)
        {
            $stamm_id = $Objekt['TID'];
            $stamm_value = $Objekt['NAME'];

            $oVWM->GETFormRowStyle4($Daten['LABEL'],$stamm_value,$Daten['PARID']);
            $stamm_id = 0;
            $stamm_value = 'k.a.';
        }
        $arrObjektdaten = NULL;
    }
}
} elseif ($Daten['STYLE'] == 6) {
    $oDBM->GETisinOBJART($sin_objtid,$Daten['PARID'],$arrISINobjart);
    if (is_array($arrISINobjart))
    {
        if (count($arrISINobjart) == 1)
        {
            foreach ($arrISINobjart as $isinObjekt)
            {
                if($isinObjekt['ISOBAID'] == $isinObjekt['OBAID'])
                {
                    $oDBM->GETSelectObjektdaten($sin_objtid,$Daten['PARID'],$arrisinObjekt);
                } elseif($isinObjekt['ISINOBAID'] == $isinObjekt['OBAID']) {
                    $oDBM->GETSelectISINObjektdaten($sin_objtid,$Daten['PARID'],$arrisinObjekt);
                }
            }
        }
    }
}
if ($Daten['FRMID'] == 1)
{
    $Linkfrmid = 2;
    $obatid = 4;
    $labelAAF = 'Aufnahmeflächen';
} else {
    $Linkfrmid = 1;
    $obatid = 5;
    $labelAAF = 'Aufnahme';
}
if (is_array($arrisinObjekt))
{
    $oVWM->GETFormRowStyle3($labelAAF);
    foreach ($arrisinObjekt as $isObjekt)
    {
        $stamm_id = $isObjekt['TID'];
        $stamm_value = $isObjekt['NAME'];
        // Hier gehört der LINK bearbeitet!!!
        // $Afile = $Afile.'?objtid='.$stamm_id.'&objname='.$stamm_value.'&frmid='.$Linkfrmid;
        $Afile = $Afile.'?objtid='.$stamm_id.'&objname='.$stamm_value.'&obatid='.$obatid.'&obaname='.$isinObjekt['OBANAME'];
        $oVWM->GETFormRowStyle6($Daten['LABEL'],$stamm_value,$Daten['PARID'],$Afile);
    }
    $arrisinObjekt = NULL;
}
}
} else {
    // Hier müssen die Objekte behandelt werden, die keine eigenes Formular haben => nur Name und Nummer!

```

```

}
SoVWM->GETFormularFooter();

if (is_array($arrFormular))
{
    // Download Excel, pdf, ASCII
    $dlfile = $dlfile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname.'&frmid='.$_sin_frmid;
    $pdffile = $pdffile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname.'&frmid='.$_sin_frmid;
    $textfile = $textfile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname.'&frmid='.$_sin_frmid;
    SoVWM->GETFormDLPrintLink($dlfile,$pdffile,$textfile);

    If($Daten['FRMID'] == 2) // muss in Zukunft nach der OBA(Objektart) ausgewählt werden => umstellen!!!
    {
        // Link auf Aufnahme-Vegetationsdaten
        $vegfile = $vegfile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;
        SoVWM->GETFormPrintLink($vegfile,$veglable_xls);
    }
} else {
    $SoVWM->GETRow();
    $SoVWM->GETDataRow("Zu diesem Objekt sind keine Daten vorhanden!");
}

// HTML-Seite Ende
SoLTM->GETFooter();

?>

```

9.3.1.2.15 Controller – „Veg. der Aufnahmefl.“ (ADMINFormAufnahmedata.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['objtid'] != ""){$_sin_objtid=$_REQUEST['objtid'];} else {$_sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$_sin_objname=$_REQUEST['objname'];} else {$_sin_objname = 0;};

$dlfile = "OOPExcelAufnahmedata.php";
$pdffile = "OOPpdfAufnahmedata.php";
$textfile = "OOPASCIIAufnahmedata.php";
$orifile = "OOPASCIIAufnahmedataORI.php";
$arrAufnahmedaten = NULL;

// INIT DBManager Objekt
$SoDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$SoVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$SoLTM = new LAYOUTManager;

// GET DB-Daten
$SoDBM->GETSelectAufnahmedata($_sin_objtid,$arrAufnahmedaten);

// HTML-Seite Anfang
$SoLTM->GETFORMHeader();

// Formular zusammenstellen
$SoVWM->GETFormularHeaderAufnahme($_sin_objname);
$SoVWM->GETFormRowStyle3("Vegetationsdaten");
if (is_array($arrAufnahmedaten))
{
    foreach ($arrAufnahmedaten as $Daten)
    {
        $SoVWM->GETFormRowArt($Daten['ART'],$Daten['VEGWERT']);
    }
}
$SoVWM->GETFormularFooter();

// Download Excel, pdf, ASCII, ORIFile
$dlfile = $dlfile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;
$pdffile = $pdffile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;

```

```

$textfile = $textfile.'?objtid='.$sin_objtid.'&objname='.$sin_objname;
$orifile = $orifile.'?objtid='.$sin_objtid.'&objname='.$sin_objname;
$soVWM->GETFormDLPrintLinkAF($dlfile,$pdffile,$textfile,$orifile);

```

```

// HTML-Seite Ende
$soLTM->GETFooter();

```

```
?>
```

9.3.1.2.16 Controller – „UPLOAD ORI-Datei“ (ADMINUploadORI.php)

```
<?php
```

```

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../../config_iflora/sqladmin.php.inc');
include_once('../../config_iflora/sqladmin.php.inc');
include_once('../../config_iflora/AdminTools.php');

// RECHT (Konstante), Fehlerbehandlung
define("formularrecht", "TEMP");
$error = "OK";
$dberror = "OK";
$sarrFehlermeldung = NULL;
$dbstatus = "";
$dbdatefkt = 'CURDATE()';
$stable = 'afn_parameter';
$sende = 'false';
$sarrHiddenINS = NULL;
$sarrHiddenDEL = NULL;

// INIT Variables
// Formularvariablen
if($_REQUEST['frmstatus'] != ""){$sin_frmstatus=$_REQUEST['frmstatus'];} else {$sin_frmstatus = '0';};
if($_REQUEST['show'] != ""){$sin_show=$_REQUEST['show'];} else {$sin_show = '0';};
// UPDATE Variablen
if($_REQUEST['temptid'] != ""){$sin_temptid=$_REQUEST['temptid'];} else {$sin_temptid = 0;};
if($_REQUEST['artcode'] != ""){$sin_artcode=$_REQUEST['artcode'];} else {$sin_artcode = 0;};
if($_REQUEST['artwert'] != ""){$sin_artwert=$_REQUEST['artwert'];} else {$sin_artwert = 0;};
// INSERT Variablen
if($_REQUEST['projekt'] != ""){$sin_projekt=$_REQUEST['projekt'];} else {$sin_projekt = 0;};
if($_REQUEST['methode'] != ""){$sin_methode=$_REQUEST['methode'];} else {$sin_methode = 0;};
// DELETE Variablen
if($_REQUEST['deltid'] != ""){$sin_deltid=$_REQUEST['deltid'];} else {$sin_deltid = 0;};

$sarrFormular = NULL;
$sarrFilter = NULL;
$sarrFilter2 = NULL;
$counter = 0;
$file = "../ADMINUploadORI.php";
$button = "Datei &uuml;bertragen";
$label = "UPLOAD ORI-Datei";

$queryVegDat = NULL;
$sarrVegDat = NULL;
$sarrVegWert = NULL;
$sarrProjekt = NULL;
$sarrMethode = NULL;
$aufnahmenr = "";

// INIT AdminTool Objekt
$SQLite = "../../config_iflora/data/userdb.dat";
$pruefDBM = new AdminTools($database,$sql_server,$sql_db,$schema);
// INIT VIEWManager Objekt
$soVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$soLTM = new LAYOUTManager;

if (isset($_COOKIE['user']))
{
    $Cookie = $pruefDBM->GET_Cookiedata($SQLite);

    if ($Cookie <> "ERROR")

```

```

{
    $arrCookie = explode(';', $Cookie);
    $aktiv = $pruefDBM->GETUserAktiv($arrCookie[0]);
    if ($aktiv == "aktiv")
    {
        $PRUEFBerechtigung = $pruefDBM->PRUEFBerechtigung(formularrecht, $arrCookie[2]);
        if ($PRUEFBerechtigung > 0)
        {
            if ($arrCookie[1] == 'ADMIN')
            {
                $user = $admin_user;
                $passwd = $admin_passwd;
            } elseif ($arrCookie[1] == 'WORK') {
                $user = $work_user;
                $passwd = $work_passwd;
            } else {
                $user = $data_user;
                $passwd = $data_passwd;
            }
        }

        // INIT DBManager Objekt
        $oDBM = new DBManager($database, $sql_server, $user, $passwd, $sql_db, $schema);

        // INIT - Projekt holen
        $queryProjekt = "SELECT PRJNAME "
            . "FROM afn_projekt "
            . "ORDER BY PRJNAME ";
        $arrProjekt = $oDBM->GETSelectSQLSelect($queryProjekt);

        // INIT - Methode holen
        $queryMethode = "SELECT MTHNAME "
            . "FROM afn_aufmethode "
            . "ORDER BY MTHNAME ";
        $arrMethode = $oDBM->GETSelectSQLSelect($queryMethode);

        // ANWENDUNG
        // 1.) Daten werden vom File geladen => werden in einem Table AFN_TEMP_VEGDAT gespeichert!
        // 2.) Die Daten werden nach dem Laden angezeigt und können bearbeitet werden (deshalb TEMPTABLE!).
        // 3.) Sind alle Fehler ausgebeßert, werden die Daten gespeichert.

        // UPDATE
        if ($in_frmstatus == 'UPDATE')
        {
            if ($in_temptid > 0 && $in_artcode != '0' && $in_artwert != '0')
            {
                // Formatprüfung der eingegebenen Werte
                $PRUEFartcode = $oVWM->PRUEFFormat($in_fmrcode, $in_artcode);
                if ($PRUEFartcode)
                {
                    $queryUPDATE = "UPDATE afn_temp_vegdat SET DS_DATE = $DBDateFkt, "
                        . "DS_USER = '$arrCookie[0]', "
                        . "TVGARTCODE = '$in_artcode', "
                        . "TVGVEGWERT = '$in_artwert' "
                        . "WHERE TID = '$in_temptid' ";
                    $oDBM->GETUpdateSQLDatum($queryUPDATE);
                } else {
                    if ($PRUEFartcode == 'false')
                    {
                        $arrFehlermeldung[] = "Artcode: Der Artcode darf nicht länger als 10 Buchstaben sein!";
                    }
                }
            } else {
                if ($in_temptid == 0)
                {
                    $arrFehlermeldung[] = "&Uuml;berschrift: Die &Uuml;berschrift darf nicht länger als 100 Buchstaben sein!";
                }
                if ($in_artcode == '0')
                {
                    $arrFehlermeldung[] = "Artcode: Dieses Feld ist ein Pflichtfeld und muss ausgefüllt werden!";
                    $in_artcode = "";
                }
                if ($in_artwert == '0')
                {
                    $arrFehlermeldung[] = "Artwert: Dieses Feld ist ein Pflichtfeld und muss ausgefüllt werden!";
                    $in_artwert = "";
                }
            }
        }
    }
}

```

```

    }
    $in_frmstatus = "";
  }
  // DELETE
} elseif ($in_frmstatus == 'DELETE') {
  $queryPRUEFdeltemp = "SELECT count(*) "
    . "FROM afn_temp_vegdat "
    . "WHERE TID = '" . $in_deltid . "' ";
  $PRUEFdeltemp = $oDBM->GETSelectSQLDatum($queryPRUEFdeltemp);
  if ($PRUEFdeltemp == 1)
  {
    $DELqueryTEMP = "DELETE FROM afn_temp_vegdat "
      . "WHERE DS_USER = '" . $sarrCookie[0] . "' "
      . "AND TID = '" . $in_deltid . "' ";
    $oDBM->GETUpdateSQLDatum($DELqueryTEMP);
  }
  // INSERT
} elseif ($in_frmstatus == 'INSERT') {
  // Daten in Tabelle afn_vegdaten übertragen:
  // 1.) Daten überprüfen
  // 2.) Daten OK => in die Tabelle laden!
  // 3.) Daten aus der TEMP Tabelle löschen!
  // Prüf Projekt
  $queryPRUEFprj = "SELECT count(*) "
    . "FROM afn_projekt "
    . "WHERE PRJNAME = '$in_projekt' ";
  $PRUEFprj = $oDBM->GETSelectSQLDatum($queryPRUEFprj);
  // Prüf Aufnahme Methode
  $queryPRUEFmth = "SELECT count(*) "
    . "FROM afn_aufmethode "
    . "WHERE MTHNAME = '$in_methode' ";
  $PRUEFmth = $oDBM->GETSelectSQLDatum($queryPRUEFmth);
  if ($PRUEFprj > 0 && $PRUEFmth > 0)
  {
    $queryVegDat = "SELECT temp.TID, "
      . "    temp.TVGAFNNR, "
      . "    temp.TVGARTCODE, "
      . "    temp.TVGVEGWERT "
      . "FROM afn_temp_vegdat temp "
      . "WHERE temp.DS_USER = '$sarrCookie[0]' "
      . "ORDER BY temp.TID ";
    $oDBM->GETSelectSQL($queryVegDat, "TID, AFNNR, ARTCODE, ARTWERT", $sarrVegDatINS);
    $sarrFehlermeldung = NULL;
    $spruefAufnahme = '0';
    // Prüfdurchgang
    if (is_array($sarrVegDatINS))
    {
      foreach ($sarrVegDatINS as $DatPRUEF)
      {
        // Prüf ARTCODE
        $queryARTCODE = "SELECT count(*) "
          . "FROM afn_arten "
          . "WHERE SPCCODE = '" . $DatPRUEF["ARTCODE"] . "' ";
        $spruefARTCODE = $oDBM->GETSelectSQLDatum($queryARTCODE);
        if ($spruefARTCODE < 1)
        {
          $sarrFehlermeldung[] = $DatPRUEF["AFNNR"] . ' - ' . $DatPRUEF["ARTCODE"] . ': Es gibt im System keine Art zu diesem
Artcode!';
        }
        // Prüf ARTWERT
        $queryARTWERT = "SELECT count(*) "
          . "FROM afn_werteliste "
          . "WHERE WRTWERT = '" . $DatPRUEF["ARTWERT"] . "' "
          . "AND WTKID = (SELECT TID "
            . "    FROM afn_werteklassen "
            . "    WHERE WTKNAME = 'OEKOWERT') ";
        $spruefARTWERT = $oDBM->GETSelectSQLDatum($queryARTWERT);
        if ($spruefARTWERT < 1)
        {
          $sarrFehlermeldung[] = $DatPRUEF["AFNNR"] . ' - ' . $DatPRUEF["ARTCODE"] . ': Der Artwert "'
          . $DatPRUEF["ARTWERT"] . '" ist ungültig!';
        }
        // Prüf AUFNAHME
        if ($spruefAufnahmenr != $DatPRUEF["AFNNR"])
        {

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
$pruefAufnahmenr = $DatPRUEF["AFNNR"];
$queryARTAFN = "SELECT count(*) "
               ."FROM afn_objekt "
               ."WHERE OBJNUMMER = '$DatPRUEF[\"AFNNR\"]' "
               ."AND OBAID = (SELECT TID "
               ."      FROM afn_objektart "
               ."      WHERE OBAKUERZEL = 'AFN') ";

$pruefAFN = $oDBM->GETSelectSQLDatum($queryARTAFN);
if ($pruefAFN > 0)
{
    $arrFehlermeldung[] = $DatPRUEF["AFNNR"].': Jede Aufnahme (bzw. Aufnahmenummer) muss eindeutig sein - eine Aufnahme mit der übergebenen Aufnahmenummer gibt es bereits in der Datenbank!';
}
}
$pruefAufnahmenr = '0';
if (is_array($arrFehlermeldung))
{
    $in_show = 'true';
    // Daten in Tabelle übertragen (3 Punkte)
    // 1.) Aufnahme anlegen (nur 1x!)
    foreach ($arrVegDatINS as $DatINS)
    {
        if ($pruefAufnahmenr != $DatINS["AFNNR"])
        {
            $pruefAufnahmenr = $DatINS["AFNNR"];
            $INSqueryAFN = "INSERT INTO afn_objekt SET DS_DATE = $DBDateFkt, "
                          ."DS_USER = '$sarrCookie[0]', "
                          ."DS_STATUS = 'NEU', "
                          ."PRJID = (SELECT TID "
                          ."      FROM afn_projekt "
                          ."      WHERE PRJNAME = '$in_projekt.', "
                          ."OBAID = (SELECT TID "
                          ."      FROM afn_objektart "
                          ."      WHERE OBAKUERZEL = 'AFN'), "
                          ."OBJNAME = '$DatINS[\"AFNNR\"]', "
                          ."OBJNUMMER = '$DatINS[\"AFNNR\"]', "
                          ."OBJGISMODE = 'POINT' ";

            $oDBM->GETUpdateSQLDatum($INSqueryAFN);
        }
        // 2.) Vegetationsdaten anlegen
        $INSqueryART = "INSERT INTO afn_vegdaten SET DS_DATE = $DBDateFkt, "
                      ."DS_USER = '$sarrCookie[0]', "
                      ."DS_STATUS = 'NEU', "
                      ."OBJID = (SELECT TID "
                      ."      FROM afn_objekt "
                      ."      WHERE OBJNUMMER = '$DatINS[\"AFNNR\"]', "
                      ."MTHID = (SELECT TID "
                      ."      FROM afn_aufmethode "
                      ."      WHERE MTHNAME = '$in_methode.', "
                      ."SPCID = (SELECT TID "
                      ."      FROM afn_arten "
                      ."      WHERE SPCCODE = '$DatINS[\"ARTCODE\"]', "
                      ."VEGWERT = '$DatINS[\"ARTWERT\"]'";

        $oDBM->GETUpdateSQLDatum($INSqueryART);
    }
    // 3.) Löschen die Daten (des Benutzers) aus der TEMP Tabelle!
    $DELqueryTEMP = "DELETE FROM afn_temp_vegdat WHERE DS_USER = '$sarrCookie[0]'";
    $oDBM->GETUpdateSQLDatum($DELqueryTEMP);
}
} else {
    if ($PRUEFprj < 1)
    {
        $arrFehlermeldung[] = "Für das Abspeichern einer ORI-Datei muss ein Projekt angegeben werden!";
    }
    if ($PRUEFmth < 1)
    {
        $arrFehlermeldung[] = "Für das Abspeichern einer ORI-Datei muss eine Aufnahmemethode angegeben werden!";
    }
    $in_show = 'true';
}
}
// Ansicht TEMP Daten inkl. Datenimport TEMP Tabelle
```



```
// HTML-Seite Anfang
if (isset($send_ori) && $in_tempid == 0)
{
    $filezahl = count($HTTP_POST_FILES);

    $counter = 0;

    $oLTM->GETFORMHeader();
    $oVWM->GETFormularHeader('UPLOAD ORI-Datei');
    $oVWM->GETFormRowStyle2('Daten der geladenden Datei');

    foreach($HTTP_POST_FILES as $strFieldName => $arrPostFiles)
    {
        if ($arrPostFiles['size'] > 0)
        {
            $oVWM->GETFormRowStyle3('Dateiname: '.$arrPostFiles['name']);
            $oVWM->GETFormRowStyle3('Dateigröße(Bit): '.$arrPostFiles['size']);
            $oVWM->GETFormRowStyle3('Dateitype: '.$arrPostFiles['type']);

            $file = $arrPostFiles['tmp_name'];
            $filearray = file($file);
            foreach($filearray as $num => $line)
            {
                $string = trim($line);
                $dslaenge = strlen($string);
                $schicht = substr($string,0,1);
                $artkuerzel = substr($string,1,$dslaenge-2);
                $artbew = substr($string,$dslaenge-1,$dslaenge);
                // Daten in Datenbank laden (AFN_TEMP_VEGDAT)
                if ($num > 0)
                {
                    if ($dslaenge == 1)
                    {
                        if ($schicht == "'")
                        {
                            $counter = 0;
                        }
                    } else {
                        if ($counter == 0)
                        {
                            $DBaufnahme = substr($line,0,$dslaenge);
                            $counter++;
                        } else {
                            if ($DBaufnahme != '0000')
                            {
                                $INSquery = "INSERT INTO afn_temp_vegdat SET DS_DATE = $DBDateFkt, "
                                    ."DS_USER = '$arrCookie[0]', "
                                    ."TVGAFNNR = '$DBaufnahme', "
                                    ."TVGSCHICHT = '$schicht', "
                                    ."TVGARTCODE = '$artkuerzel', "
                                    ."TVGVEGWERT = trim('$artbew') ";

                                $DBStatus = $oDBM->GETUpdateSQLDatumErr($INSquery);
                                if ($DBStatus == 'ERROR')
                                {
                                    $arrFehlermeldung[] = "Datenbankfehler(doppelt?): ".$DBaufnahme." - ".$artkuerzel;
                                }
                                $DBStatus = "";
                            }
                            $counter++;
                        }
                    }
                }
            }
        }
    }

    // DATEN der geladenen AUFNAHMEN!!!
    // INIT - geladen Daten holen
    $queryVegDat = "SELECT temp.TID, "
        ."    temp.TVGAFNNR, "
        ."    concat(temp.TVGARTCODE,['ifnull(a.SPCNAME,?)']) AS ARTCODE, "
        ."    temp.TVGVEGWERT "
        ."FROM    afn_temp_vegdat temp "
        ."    LEFT JOIN afn_arten a ON trim(a.SPCCODE) = trim(temp.TVGARTCODE) "
        ."WHERE temp.DS_USER = '$arrCookie[0]' "
        ."ORDER BY temp.TID ";

    $oDBM->GETSelectSQL($queryVegDat,'TID,AFNNR,ARTCODE,ARTWERT',$arrVegDat);
}
```

```

if (is_array($arrFehlermeldung))
{
    $oVWM->GETFormRowStyle3('<b>FEHLERMELDUNGEN</b>');
    $count = count($arrFehlermeldung);
    for ($i=0; $i<$count; $i++)
    {
        $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
    }
}

// Formular zusammenstellen
if (is_array($arrVegDat))
{
    foreach ($arrVegDat as $Daten)
    {
        if ($aufnahmenr != $Daten["AFNNR"])
        {
            $aufnahmenr = $Daten["AFNNR"];
            $oVWM->GETFormRowStyle1($aufnahmenr);
            //$rowcount++;
        }
        if (preg_match("(\\[?\\])", $Daten["ARTCODE"], $regs))
        {
            $oVWM-
>GETFormRowStyleUPDATE4Err($Daten["ARTCODE"],$Daten["ARTWERT"],'temptid',$Daten["TID"],$PHP_SELF);
            //$rowcount++;
        } else {
            $oVWM-
>GETFormRowStyleUPDATE4temp($Daten["ARTCODE"],$Daten["ARTWERT"],'temptid',$Daten["TID"],$PHP_SELF);
            //$oVWM->GETFormRowStyle4($Daten["ARTCODE"],$Daten["ARTWERT"],$Daten["TID"]);
            //$rowcount++;
        }
    }
}
}
}
// $oVWM->GETFormularFooter();
$hiddenhashINS["NAME"] = "frmstatus";
$hiddenhashINS["TID"] = 'INSERT';
$arrHiddenINS[] = $hiddenhashINS;
$oVWM->GETFormularFooterUploadORI($arrHiddenINS,$arrProjekt,"$arrMethode,");
$oLTM->GETFooter();
} else {
    // EDIT Datensatz => Formular - ändern eines Datensatzes
    if ($in_temptid > 0 && $in_frmstatus != 'UPDATE')
    {
        // HiddenArray füllen
        $hiddenhash["NAME"] = "show";
        $hiddenhash["TID"] = 'true';
        $arrHidden[] = $hiddenhash;
        $arrHidden2[] = $hiddenhash;
        $hiddenhash["NAME"] = "temptid";
        $hiddenhash["TID"] = $in_temptid;
        $arrHidden[] = $hiddenhash;

        $queryVegDat = "SELECT temp.TID, "
            ."      temp.TVGAFNNR, "
            ."      temp.TVGARTCODE AS ARTCODE, "
            ."      ifnull(a.SPCNAME,['?']) AS ART, "
            ."      temp.TVGVEGWERT "
            ."FROM    afn_temp_vegdat temp "
            ."      LEFT JOIN afn_arten a ON trim(a.SPCCODE) = trim(temp.TVGARTCODE) "
            ."WHERE   temp.DS_USER = '$arrCookie[0]' "
            ."AND     temp.TID = '$in_temptid' "
            ."ORDER BY temp.TID ";

        $oDBM->GETSelectSQL($queryVegDat,'TID,AFNNR,ARTCODE,ART,ARTWERT',$arrVegDat);

        if (is_array($arrVegDat))
        {
            $oLTM->GETFORMHeader();
            $oVWM->GETFormularHeaderUFormUPD2('UPLOAD ORI-Daten',$file,'UPDATE',$arrHidden,$arrHidden2);

            if (is_array($arrFehlermeldung))
            {
                $oVWM->GETFormRowStyle3('<b>FEHLERMELDUNGEN</b>');
            }
        }
    }
}

```

```

$count = count($arrFehlermeldung);
for ($i=0; $i<$count; $i++)
{
    $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
}
$oVWM->GETFormRowStyle3("");
$oVWM->GETFormRowStyle2('Datensatz Bearbeiten');
$oVWM->GETFormRowStyle3("");
$oVWM->GETFormRowStyleUPDAdmin4('Artcode',$in_artcode,'artcode','X(10)',10);
$oVWM->GETFormRowStyleUPDAdmin4('Art',$in_art,'art','X(55)',55);
$oVWM->GETFormRowStyleINSARRAdmin4('Artmächtigkeit',$in_artwert,'artwert',$arrVegWert);
} else {
    // INIT - Veg Werte holen
    $queryVegWert = "SELECT wl.WRTWERT "
        ."FROM   afn_werteliste wl, "
        ."       afn_werteklassen kl "
        ."WHERE  kl.TID = wl.WTKID "
        ."AND    kl.WTKNAME = 'OEKOWERT'";
    $arrVegWert = $oDBM->GETSelectSQLSelect($queryVegWert);

    $oVWM->GETFormRowStyle2('Datensatz Bearbeiten');
    $oVWM->GETFormRowStyle3("");
    foreach ($arrVegDat as $Daten)
    {
        $oVWM->GETFormRowStyleUPDAdmin4('Artcode',$Daten["ARTCODE"],'artcode','X(10)',10);
        $oVWM->GETFormRowStyleUPDAdmin4('Art',$Daten["ART"],'art','X(55)',55);
        $oVWM->GETFormRowStyleINSARRAdmin4('Artmächtigkeit',$Daten["ARTWERT"],'artwert',$arrVegWert);
    }
}
$oVWM->GETFormRowStyle3("");
$shidhashDEL["NAME"] = "frmstatus";
$shidhashDEL["TID"] = 'DELETE';
$arrHiddenDEL[] = $shidhashDEL;
$shidhashDEL["NAME"] = "show";
$shidhashDEL["TID"] = 'true';
$arrHiddenDEL[] = $shidhashDEL;
$shidhashDEL["NAME"] = "deltid";
$shidhashDEL["TID"] = $in_temptid;
$arrHiddenDEL[] = $shidhashDEL;
$oVWM->GETFormularFooterUserverwaltungORI($arrHiddenDEL);
}
// Ansicht TEMP Daten
} elseif ($in_show == 'true') {
    $queryVegDat = "SELECT temp.TID, "
        ."    temp.TVGAFNNR, "
        ."    concat(temp.TVGARTCODE,['ifnull(a.SPCNAME,?)']) AS ARTCODE, "
        ."    temp.TVGVEGWERT "
        ."FROM   afn_temp_vegdat temp "
        ."    LEFT JOIN afn_arten a ON trim(a.SPCCODE) = trim(temp.TVGARTCODE) "
        ."WHERE  temp.DS_USER = '$arrCookie[0]' "
        ."ORDER BY temp.TID ";
    $oDBM->GETSelectSQL($queryVegDat,'TID,AFNNR,ARTCODE,ARTWERT',$arrVegDat);

    $oLTM->GETFORMHeader();
    $oVWM->GETFormularHeader('UPLOAD ORI-Datei');
    $oVWM->GETFormRowStyle2('Daten der geladenden Datei(en)');

    if (is_array($arrFehlermeldung))
    {
        $oVWM->GETFormRowStyle3('<b>FEHLERMELDUNGEN</b>');
        $count = count($arrFehlermeldung);
        for ($i=0; $i<$count; $i++)
        {
            $oVWM->GETFormRowStyleERRORAdmin4($arrFehlermeldung[$i]);
        }
    }

    // Formular zusammenstellen
    if (is_array($arrVegDat))
    {
        foreach ($arrVegDat as $Daten)
        {
            if ($aufnahmenr != $Daten["AFNNR"])
            {
                $aufnahmenr = $Daten["AFNNR"];
            }
        }
    }
}

```

```

        $oVWM->GETFormRowStyle1($aufnahmenr);
        //$rowcount++;
    }
    if (preg_match("(\\[?\\])", $Daten["ARTCODE"], $regs))
    {
        $oVWM-
>GETFormRowStyleUPDATE4Err($Daten["ARTCODE"],$Daten["ARTWERT"],'temptid',$Daten["TID"],$PHP_SELF);
        //$rowcount++;
    } else {
        $oVWM-
>GETFormRowStyleUPDATE4temp($Daten["ARTCODE"],$Daten["ARTWERT"],'temptid',$Daten["TID"],$PHP_SELF);
        //$oVWM->GETFormRowStyle4($Daten["ARTCODE"],$Daten["ARTWERT"],$Daten["TID"]);
        //$rowcount++;
    }
}
}
// $oVWM->GETFormularFooter();
$hiddenhashINS["NAME"] = "frmstatus";
$hiddenhashINS["TID"] = "INSERT";
$arrHiddenINS[] = $hiddenhashINS;
$oVWM->GETFormularFooterUploadORI($arrHiddenINS,$arrProjekt,"$arrMethode,");
$oLTM->GETFooter();
} else {
    $oLTM->GETFORMHeader();
    $oVWM->GETUpload($label,$button);
    $oLTM->GETFooter();
}
}
} else {
    $ERROR = "Ihre Rechte sind nicht ausreichend, eine ORI Datei zu laden!Bitte setzen Sie sich mit Ihrem Administrator in
Verbindung!";
}
} else {
    $ERROR = "Ihr Benutzername ist inaktiv!Bitte setzen Sie sich mit Ihrem Administrator in Verbindung!";
}
} else {
    $ERROR = "Es gibt ein technisches Problem (ERROR-L100)!Bitte setzen Sie sich mit Ihrem Systemadministrator in Verbindung!";
}
} else {
    $ERROR = "Es wurde kein COOKIE gesetzt! Für den Betrieb von iFlora ist ein Cookie notwendig und der Einstieg über die offizielle
Startseite!Bitte gehen sie auf die offizielle Startseite oder aktivieren sie die Cookie-Funktion bei Ihrem Internetbrowser!";
}
}

If($ERROR <> OK) // Fehlermeldung
{
    $oLTM->GETHeader();
    $oVWM->GETLabel("iFlora Parameterverwaltung", "H1");
    $oVWM->GETTextfield($ERROR);
    $oLTM->GETFooter();
}
}

?>

```

9.3.2 iFlora Öffentlicher Bereich

9.3.2.1 Programmbibliotheken (Libraries)

9.3.2.1.1 DBManager.php

```

<?php

// This script is called by iFlora CONTROLLER to get Data
// for DHTML DB elements.
//
// history
// 2005-09-24 1.0 SH  begin DBManager and Method GETSelectData
// 2005-09-25 1.1 SH  begin Method GETInputData
//
// description
// GETSelectData => GET Data for a SELECT form
// GETInputData => GET Data for INPUT form

```

```

class DBManager
{
    private $_dbms;
    private $_dbhost;
    private $_sqluser;
    private $_sqlpwd;
    private $_sqldb;
    private $_sqlschema;

    /**
     *
     */

    function __construct($_dbms,$_host,$_user,$_pwd,$_db,$_schema)
    {
        $this->_dbms = $_dbms;
        $this->_dbhost = $_host;
        $this->_sqluser = $_user;
        $this->_sqlpwd = $_pwd;
        $this->_sqldb = $_db;
        $this->_sqlschema = $_schema;
    }

    /**
     *
     */

    function GETSelectDatum($table,$field,$orderby,$where)
    {
        // include
        include_once('adodb/adodb.inc.php');
        include_once('adodb/tohtml.inc.php');

        // ADODB
        $db = ADONewConnection($this->_dbms);
        $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

        $query = "SELECT DISTINCT $field "
                ."FROM $table "
                ."$where "
                ."$orderby ";

        $result = $db->Execute($query);

        if ($result==" " || $result==" " || $result === false)
        {
            print_r($TEXT['ADODB-nottable']);
        } else {
            $datum = $result->fields[0];
        }
        return $datum;
    }

    /**
     *
     */

    function GETObjektStammdaten($obaid,&$arrDaten)
    {
        // include
        include_once('adodb/adodb.inc.php');
        include_once('adodb/tohtml.inc.php');

        // fields
        $fields = "LABEL,POSITION,FORMAT,PARID,FORMLABEL,STYLE,FRMID";
        $arrfield = explode(',',$fields);

        // Querys
        $SQLquery = "SELECT concat(p.PARNR, ' - ', p.PARNAME) AS LABEL, "
                ."r.FMRPOSITION AS POSITION, "
                ."p.PARFORMAT AS FORMAT, "
                ."p.TID AS PARID, "
                ."r.FMRLABEL AS FORMLABEL, "
                ."r.FMRSTYLE AS STYLE, "
                ."f.TID AS FRMID "
                ."FROM   afn_h_formular f, afn_h_formrow r, afn_parameter p "
                ."WHERE  f.OBAID = $obaid "
                ."AND    r.FRMID = f.TID "
                ."AND    p.TID = r.PARID "
                ."ORDER BY r.FMRPOSITION ";
    }

```

```
// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

// Datenbearbeitung
$result = $db->Execute($SQLquery);

if ($result==" " || $result==" " || $result === false)
{
    $status = "ERROR or EMPTY";
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hash[$arrfield[$i]] = $result->fields[$i];
        }
        $arrDaten[] = $hash;
        $result->MoveNext();
    }
}

//*****

function GETSelectSQLDatum($query)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $datum = $result->fields[0];
    }
    return $datum;
}

//*****

function GETSelectSQL($query,$fields,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // $arrfield = preg_split("/,/",$fields);
    $arrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ^)!
    if ($this->_dbms=="mysql")
    {
        $table = "".strtolower($table)."";
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    }
}
```

```

} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hash[$arrfield[$i]] = $result->fields[$i];
        }
        $arrDaten[] = $hash;
        $result->MoveNext();
    }
}

//*****

function GETSelectStammdata($frmid,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // fields
    $fields = "LABEL,POSITION,FORMAT,PARID,FORMLABEL,STYLE";
    $arrfield = explode(',',$fields);

    // Querys
    $SQLquery = "SELECT concat(p.PARNR, ' - ', p.PARNAME) AS LABEL, "
                ."r.FMRPOSITION AS POSITION, "
                ."p.PARFORMAT AS FORMAT, "
                ."p.TID AS PARID, "
                ."r.FMRLABEL AS FORMLABEL, "
                ."r.FMRSTYLE AS STYLE "
                ."FROM   afn_h_formrow r, afn_parameter p "
                ."WHERE r.FRMID = $frmid "
                ."AND    p.TID = r.PARID "
                ."ORDER BY r.FMRPOSITION ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($SQLquery);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($arrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash[$arrfield[$i]] = $result->fields[$i];
            }
            $arrDaten[] = $hash;
            $result->MoveNext();
        }
    }
}

//*****

function GETSelectAufnahmedata($objid,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // fields
    $fields = "ART,VEGWERT";
    $arrfield = explode(',',$fields);

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
// Querys
$SQLquery = "SELECT a.SPCNAME AS ART, v.VEGWERT AS VEGWERT "
            ."FROM   afn_vegdaten v, "
            ."       afn_arten a "
            ."WHERE  a.TID = v.SPCID "
            ."AND    v.OBJID = $objid ";

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

// Datenbearbeitung
$result = $db->Execute($SQLquery);

if ($result==" " || $result==" " || $result === false)
{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hash[$arrfield[$i]] = $result->fields[$i];
        }
        $arrDaten[] = $hash;
        $result->MoveNext();
    }
}

//*****

function GETSelectORIdata($objid,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // fields
    $fields = "ART,VEGWERT";
    $arrfield = explode(',',$fields);

    // Querys
    $SQLquery = "SELECT  ifnull(a.SPCCODE,concat('?',a.SPCNAME,',')) AS ART, v.VEGWERT AS VEGWERT "
                ."FROM    afn_vegdaten v, "
                ."        afn_arten a "
                ."WHERE   a.TID = v.SPCID "
                ."AND     v.OBJID = $objid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($SQLquery);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($arrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash[$arrfield[$i]] = $result->fields[$i];
            }
            $arrDaten[] = $hash;
            $result->MoveNext();
        }
    }
}
```



```

}

//*****

function GETSelectParamdata($objid,$parid,&$arrParam)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $PARquery = "SELECT DISTINCT TID, STDWERT "
                ."FROM    afn_stammdaten "
                ."WHERE   objID = $objid "
                ."AND     parID = $parid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($PARquery);

    // fields
    $fieldes = "TID,STDWERT";
    $arrfields = explode(',',$fieldes);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<2;$i++)
            {
                $hash[$arrfields[$i]] = $result->fields[$i];
            }
            $arrParam[] = $hash;
            $result->MoveNext();
        }
    }
}

//*****

function GETSelectObjektdata($objid,$parid,&$arrParam)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $ISINquery = "SELECT DISTINCT o.tid AS TID, o.objname AS NAME "
                ."FROM    afn_isin i, afn_parameter p, afn_objekt o "
                ."WHERE   p.tid = $parid "
                ."AND     i.isaid = p.isaid "
                ."AND     i.objid = $objid "
                ."AND     o.tid = i.isinobjid ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($ISINquery);

    // fields
    $fieldes = "TID,NAME";
    $arrfields = explode(',',$fieldes);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        // Auswahlliste zusammenstellen
        while (!$result->EOF)

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
{
    for ($i=0;$i<2;$i++)
    {
        $hash[$sarrfields[$i]] = $result->fields[$i];
    }
    $sarrParam[] = $hash;
    $result->MoveNext();
}
}
}

//*****

function GETSelectISINObjektdata($objjid,$parid,&$sarrParam)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $ISINquery = "SELECT DISTINCT o.tid AS TID, o.objname AS NAME "
        ."FROM   afn_isin i, afn_parameter p, afn_objekt o "
        ."WHERE  p.tid = $parid "
        ."AND    i.isaid = p.isaid "
        ."AND    i.isinobjid = $objjid "
        ."AND    o.tid = i.objjid "
        ."ORDER BY o.objname ";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // Datenbearbeitung
    $result = $db->Execute($ISINquery);

    // fields
    $fieldes = "TID,NAME";
    $sarrfields = explode(',',$fieldes);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<2;$i++)
            {
                $hash[$sarrfields[$i]] = $result->fields[$i];
            }
            $sarrParam[] = $hash;
            $result->MoveNext();
        }
    }
}

//*****

function GETIsinartPOL($objjid,$parid,&$sarrPOL)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // Abfrage
    $query = "SELECT DISTINCT ogem.objname AS Gemeinde, ogem.tid AS GemeindeID, "
        ."obez.objname AS Bezirk, obez.tid AS BezirkID, "
        ."obl.objname AS Bundesland, obl.tid AS BundeslandID "
        ."FROM   afn_isin igem, "
        ."afn_isin ibez, "
        ."afn_isin ibl, "
        ."afn_parameter p, "
        ."afn_objekt ogem, "
        ."afn_objekt obez, "
        ."afn_objekt obl "
        ."WHERE  p.tid = $parid "
```

```

        ."AND    igem.isaid = p.isaid "
        ."AND    igem.objid = $objid "
        ."AND    ogem.tid = igem.isinobjid "
        ."AND    ibez.isaid = p.isaid "
        ."AND    ibez.objid = igem.isinobjid "
        ."AND    obez.tid = ibez.isinobjid "
        ."AND    ibl.isaid = p.isaid "
        ."AND    ibl.objid = ibez.isinobjid "
        ."AND    obl.tid = ibl.isinobjid ";

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

// Datenbearbeitung
$result = $db->Execute($query);

// fields
$fieldes = "Gemeinde,GemeindeID,Bezirk,BezirkID,Bundesland,BundeslandID";
$arrfield = explode(',',$fieldes);

if ($result==" " || $result==" " || $result === false)
{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0;$i<$count;$i++)
        {
            $hashPOL[$arrfield[$i]] = $result->fields[$i];
        }
        $arrPOL[] = $hashPOL;
        $result->MoveNext();
    }
}

}

/*****

function GETisinOBJART($objid,$parid,&$arrISINOBA)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // Abfrage
    $query = "SELECT DISTINCT  ioa.isaid AS ISAID, "
                ."ioa.obaid AS ISOBAID, "
                ."ioa.isinobaid AS ISINOBAID, "
                ."o.obaid AS OBAID "
        ."FROM    afn_isinobjart ioa, "
        ."afn_parameter p, "
        ."afn_objekt o "
        ."WHERE    p.tid = $parid "
        ."AND      ioa.isaid = p.isaid "
        ."AND      o.tid = $objid ";

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

// Datenbearbeitung
$result = $db->Execute($query);

// fields
$fieldes = "ISAID,ISOBAID,ISINOBAID,OBAID";
$arrfield = explode(',',$fieldes);

if ($result==" " || $result==" " || $result === false)
{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($arrfield);

```

```

// Auswahlliste zusammenstellen
while (!$result->EOF)
{
    for ($i=0;$i<$count;$i++)
    {
        $hashOBA[$sarrfield[$i]] = $result->fields[$i];
    }
    $arrISINOBA[] = $hashOBA;
    $result->MoveNext();
}
}

//*****

function GETSelectData($table,$fields,$orderby,$where,&$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // $sarrfield = preg_split("/,/",$fields);
    $sarrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = ``.strtolower($table).``;
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $query = "SELECT DISTINCT $fields "
            . "FROM $table "
            . "$where "
            . "$orderby ";

    $result = $db->Execute($query);

    if ($result==" " || $result==" " || $result === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($sarrfield);
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash[$sarrfield[$i]] = $result->fields[$i];
            }
            $arrDaten[] = $hash;
            $result->MoveNext();
        }
    }
}

//*****

function GETSelDistArray($tables,$fields,$orderby,$where,&$arrDaten,&$arrLabel)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $sarrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    //if ($this->_dbms=="mysql")
    //{
    //    $table = ``.strtolower($table).``;
    //}

```

```

// ADODB
$db = ADONewConnection($this->_dbms);
$db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

$query = "SELECT DISTINCT $fields "
        ."FROM $tables "
        ."$where "
        ."$orderby ";

$result = $db->Execute($query);

if ($result==" " || $result==" " || $result === false)
{
    print_r($TEXT['ADODB-nottable']);
} else {
    $count = count($arrfield);
    // Auswahlliste zusammenstellen
    while (!$result->EOF)
    {
        for ($i=0; $i<$count; $i++)
        {
            $label = $arrLabel[$arrfield[$i]];
            $arrDaten[$label] = $result->fields[$i];
        }
        $result->MoveNext();
    }
}

//*****

function GETInputData($table, $select, $fields, $orderby, $where, $params, $labels, $types, &$arrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $arrfield = explode(',', $fields);
    $paramfield = explode(',', $params);
    $labelfield = explode(',', $labels);
    $typfield = explode(',', $types);

    // Convert to MySQL-Standard (LOWER TABLENAME within ~)!
    if ($this->_dbms=="mysql")
    {
        $table = "~".strtolower($table). "~";
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $queryCOUNT = "SELECT count(TID) "
                   ."FROM $table "
                   ."$where ";

    $query = "SELECT DISTINCT $select "
            ."FROM $table "
            ."$where "
            ."$orderby ";

    $resultCOUNT = $db->Execute($queryCOUNT);

    if ($resultCOUNT==" " || $resultCOUNT==" " || $resultCOUNT === false)
    {
        print_r($TEXT['ADODB-nottable']);
    } else {
        $pruef = $resultCOUNT->fields[0];
        if ($pruef>0)
        {
            $result = $db->Execute($query);

            if ($result==" " || $result==" " || $result === false)
            {

```

```

        print_r($TEXT['ADODB-nottable']);
    } else {
        $count = count($sarrfield);
        $j = 0;
        // Auswahlliste zusammenstellen
        while (!$result->EOF)
        {
            for ($i=0;$i<$count;$i++)
            {
                $hash["value"] = $result->fields[$i];
                $hash["label"] = $labelfield[$i];
                $hash["param"] = $paramfield[$i];
                $hash["typ"] = $typfield[$i];
                $harr[] = $hash;
                $hash = NULL;
            }

            $sarrDaten[] = $harr;
            $harr = NULL;

            $result->MoveNext();
        }
    } else {
        $count = count($sarrfield);
        for ($i=0;$i<$count;$i++)
        {
            $hash["value"] = "";
            $hash["label"] = $labelfield[$i];
            $hash["param"] = $paramfield[$i];
            $hash["typ"] = $typfield[$i];
            $harr[] = $hash;
            $hash = NULL;
        }

        $sarrDaten[] = $harr;
        $harr = NULL;
    }
}

}

}

//*****

function GETOutputData($table,$fields,$orderby,$where,&$sarrDaten)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    $sarrfield = explode(',',$fields);

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = ``.strtolower($table).``;
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    $queryCOUNT = "SELECT count(TID) "
        . "FROM $table "
        . "$where ";

    $query = "SELECT DISTINCT $fields "
        . "FROM $table "
        . "$where "
        . "$orderby ";

    $resultCOUNT = $db->Execute($queryCOUNT);

    if ($resultCOUNT==" " || $resultCOUNT==" " || $resultCOUNT=== false)
    {

```

```

print_r($TEXT['ADODB-nottable']);
} else {
    $pruef = $resultCOUNT->fields[0];
    if ($pruef>0)
    {
        $result = $db->Execute($query);

        if ($result==" " || $result==" " || $result === false)
        {
            print_r($TEXT['ADODB-nottable']);
        } else {
            $count = count($arrfield);
            // Auswahlliste zusammenstellen
            while (!$result->EOF)
            {
                for ($i=0;$i<$count;$i++)
                {
                    $harr[] = $result->fields[$i];
                }
                $arrDaten[] = $harr;
                $harr = NULL;
                $result->MoveNext();
            }
        }
    }
}

}

}

}

//*****

function INSERTStandard($table,$sequence,&$arrINSERT)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    if ($this->_dbms=="oci8")
    {
        $sequence = "$sequence.NEXTVAL";
        $saktdate = "SYSDATE";
    } elseif ($this->_dbms=="postgres") {
        $sequence = "NEXTVAL('$sequence')";
        $saktdate = "CURRENT_TIMESTAMP";
    } elseif ($this->_dbms=="mysql") {
        $sequence = "";
        $saktdate = "CURRENT_TIMESTAMP";
        $table = "".strtolower($table)."";
    }

    $user = 0;

    $query = "INSERT INTO ".$this->_sqlschema.$table." VALUES ($sequence, "
        ."$saktdate, "
        ."$user, ";

    // INSERTquery zusammenstellen
    foreach ($arrINSERT as $Daten)
    {
        if ($Daten["typ"]=="T" || $Daten["typ"]=="D")
        {
            $query = $query."".$Daten["value"].",", ";
        } else {
            $query = $query.$Daten["value"].",", ";
        }
    }

    $len = strlen($query);
    $query = substr($query,0,$len-2);
    $query = $query."";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

```

```

$db->Execute($query);

}

//*****

function UPDATEStandard($table,$tid,&$arrUPDATE)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // UPDATEquery zusammenstellen
    $query = "UPDATE ".$this->_sqlschema.$table." SET ";

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = "`".strtolower($table)."`";
    }

    foreach ($arrUPDATE as $Daten)
    {
        if ($Daten["typ"]=="T" || $Daten["typ"]=="D")
        {
            $field = $Daten["field"]." = ".$Daten["value"].", ";
        } else {
            $field = $Daten["field"]." = ".$Daten["value"].", ";
        }
        $query = $query.$field;
    }

    $len = strlen($query);
    $query = substr($query,0,$len-2);
    $query = $query." WHERE TID = $tid";

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $db->Execute($query);
}

//*****

function DELETEStandard($table,$tid)
{
    // include
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // UPDATEquery zusammenstellen
    $query = "DELETE FROM ".$this->_sqlschema.$table." WHERE TID = ".$tid;

    // Convert to MySQL-Standard (LOWER TABLENAME within ``)!
    if ($this->_dbms=="mysql")
    {
        $table = "`".strtolower($table)."`";
    }

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqlldb);

    $db->Execute($query);
}

//*****

public function GETIfnull($text,$ifnull)
{
    if ($this->_dbms=="oci8")
    {
        $text = "NVL($text,'$ifnull')";
    } elseif ($this->_dbms=="postgres") {

```



```

    $text = "ifnull($text,$ifnull)";
} elseif ($this->_dbms=="mysql") {
    $text = "ifnull($text,$ifnull)";
}
return $text;
}

//*****

function GET_Userstatus($sql_user)
{
    // INCLUDE
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // SQL-Abfrage zusammenstellen
    if ($this->_dbms=="oci8")
    {
        $query = "SELECT STATUS, "
            . "      (SYSDATE - DS_DATE), "
            . "      NVL(DAUER,0) "
            . "FROM USERNAME "
            . "WHERE NAME = '$sql_user'";
        $queryPRUEF = "SELECT count(TID) "
            . "FROM USERNAME ";
    } elseif ($this->_dbms=="postgres") {

        $query = "SELECT STATUS, "
            . "      (CURRENT_DATE - DS_DATE), "
            . "      IFNULL(DAUER,0) "
            . "FROM USERNAME "
            . "WHERE NAME = '$sql_user'";
        $queryPRUEF = "SELECT count(TID) "
            . "FROM USERNAME ";
    } elseif ($this->_dbms=="mysql") {
        $query = "SELECT STATUS, "
            . "      (UNIX_TIMESTAMP() - UNIX_TIMESTAMP(DS_DATE))/86400, "
            . "      ifnull(DAUER,0) "
            . "FROM `username` "
            . "WHERE NAME = '$sql_user'";
        $queryPRUEF = "SELECT count(TID) "
            . "FROM `username` ";
    }
}

try
{
    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // SQL Abfrage ausführen
    $result = $db->Execute($query);

    // Auswertung
    if ($result==" " || $result==" " || $result === false)
    {
        $user_aktiv = "N-DB";
    } else {
        $fkt_typ = $result->fields[0];
        $fkt_diff = $result->fields[1];
        $fkt_dauer = $result->fields[2];

        if ($fkt_typ!="")
        {
            if ($fkt_typ == "ADMIN" || $fkt_typ == "USER")
            {
                /*****
                * USER Typen:                               *
                * =====                               *
                * ADMIN => Administrator                 *
                * USER  => Permanent (normaler User)      *
                *****/
            }
        }
    }
}

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
* TEST => Temporär (Testuser) *
*****/

    $user_aktiv = 'Y';
} else {
    if ($fkt_diff > $fkt_dauer)
    {
        $user_aktiv = 'N-TIMEOUT';
    } else {
        $user_aktiv = 'Y';
    }
}
} else {
    $user_aktiv = 'N-NOTVALID';
}
}
} catch (Exception $exception) {
    $user_aktiv = "N_DB";
}

return $user_aktiv.";".$fkt_typ;
}

//*****

function GETCreateUser($sql_user,$pwd)
{
    // INCLUDE
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // DBMS abhängiges USER anlegen!
    if ($this->_dbms=="oci8")
    {
        $queryCREATE = "CREATE USER ".$sql_user." IDENTIFIED BY abc";
        $queryGRANT = "GRANT connect, resource TO ".$sql_user." ";
        $db->Execute($queryCREATE);
        $db->Execute($queryGRANT);
    } elseif ($this->_dbms=="postgres") {
        $queryCREATE = "CREATE USER $sql_user IDENTIFIED BY $pwd";
        $db->Execute($queryCREATE);
    } elseif ($this->_dbms=="mysql") {
        $queryCREATE = "GRANT SELECT ON ".$this->_sqldb.".projekt TO '".$sql_user."' IDENTIFIED BY '".$pwd.'";";
        $db->Execute($queryCREATE);
    }
}

//*****

function GETCreateStandardTable($tablename)
{
    // INCLUDE
    include_once('adodb/adodb.inc.php');
    include_once('adodb/tohtml.inc.php');

    // ADODB
    $db = ADONewConnection($this->_dbms);
    $db->Connect($this->_dbhost, $this->_sqluser, $this->_sqlpwd, $this->_sqldb);

    // DBMS abhängiges USER anlegen!
    if ($this->_dbms=="oci8")
    {
        $queryCREATE = "CREATE TABLE ".$this->_sqlschema.$tablename." ( "
            ."TID          NUMBER(13,3) CONSTRAINT PK_".$tablename." PRIMARY KEY, "
            ."DS_DATE       DATE NOT NULL, "
            ."DS_USER        NUMBER(13,3) NOT NULL)";
        $querySEQUENCE = "CREATE SEQUENCE ".$this->_sqlschema."SEQ_".$tablename;
```

```

$querySYNONYM = "CREATE PUBLIC SYNONYM ".$tablename." FOR ".$this->$_sqlschema.$tablename;
$db->Execute($queryCREATE);
$db->Execute($querySEQUENCE);
$db->Execute($querySYNONYM);
} elseif ($this->_dbms=="postgres") {
$queryCREATE = "CREATE TABLE ".$this->$_sqlschema.$tablename." ( "
    ."TID      NUMBER(13,3) CONSTRAINT PK_". $tablename." PRIMARY KEY, "
    ."DS_DATE  DATE NOT NULL, "
    ."DS_USER  NUMBER(13,3) NOT NULL)";
$querySEQUENCE = "CREATE SEQUENCE ".$this->$_sqlschema."SEQ_". $tablename;
$db->Execute($queryCREATE);
$db->Execute($querySEQUENCE);
} elseif ($this->_dbms=="mysql") {
$queryCREATE = "CREATE TABLE ".$this->$_sqlschema.$tablename." ( "
    ."TID      INT(13) NOT NULL AUTO_INCREMENT, "
    ."DS_DATE  DATETIME NOT NULL, "
    ."DS_USER  INT(13) NOT NULL, "
    ."PRIMARY KEY(TID))";
$db->Execute($queryCREATE);
}

}

}

?>

```

9.3.2.1.2 ViewManager.php

```

<?php

// This script is called by iFlora CONTROLLER to print
// DHTML DB elements.
//
// history
// 2005-09-24 1.0 SH  begin GETSelect
// 2005-09-25 1.1 SH  split GETSelect in GETSelect and
//                  GETSelectForm
//                  begin GETInputForm
// 2005-09-27 1.2 SH  private Arrays
// 2005-09-28 1.3 SH  private Arrays to slow!!!
//                  => references
//
// description
// GETSelect => produce a SELECT
// GETSelectForm => produce a SELECT with a form
// GETOutput => produce a list of data

class VIEWManager
{
    //*****

    // Übergabe von Arrays zu langsam!!! => deshalb REFERENZ!!!

    public function GETSelect($tid,$parameter,&$arrSQL)
    {
        // Auswahlliste BEGIN
        echo ' <p>
            <select name="'.$parameter.'" size="1">
            <option value="0">Bitte auswählen</option>';

        // Auswahlliste zusammenstellen
        foreach ($arrSQL as $Daten)
        {
            foreach ($Daten as $key => $value)
            {
                if ($key=="TID")
                {
                    $hTID = $value;
                } elseif ($key=="NAME") {
                    $hNAME = $value;
                }
            }
        }
    }
}

```

```

    if ($hTID==$tid)
    {
        print '<option selected value="'. $hTID. '">'. $hNAME. '</option>';
    } else {
        print '<option value="'. $hTID. '">'. $hNAME. '</option>';
    }
}
// Auswahlliste END
echo ' </select>';

// ENDE Auswahlliste
}

//*****

public function GETSelectForm($tid,$param,$file,$lbSelect,&$arrSQL)
{
    // Formular BEGIN
    echo '<form method="GET" action="'. $file. '">
        <p>';

    self::GETSelect($tid,$param,$arrSQL);

    // Formular END
    echo ' </p>
        <p>
            <INPUT TYPE="SUBMIT" VALUE="'. $lbSelect. '">
        </p>
        </form>';
}

//*****

public function GETSelFormHidden($tid,$param,$file,$lbSelect,&$arrSQL,&$arrHIDDEN)
{
    // Formular BEGIN
    echo '<form method="GET" action="'. $file. '">
        <p>';

    self::GETSelect($tid,$param,$arrSQL);

    // Formular END
    echo ' </p>
        <p>';

    if (is_array($arrHIDDEN))
    {
        foreach ($arrHIDDEN as $key => $value)
        {
            echo '<input type="hidden" name="'. $key. '" value="'. $value. '">';
        }
    }

    echo ' <INPUT TYPE="SUBMIT" VALUE="'. $lbSelect. '">
        </p>
        </form>';
}

//*****

public function GETInput(&$arrInputSQL)
{
    // Eingabe BEGIN
    echo '<table>';
    $rownr = 1;

    foreach ($arrInputSQL as $Daten)
    {
        foreach ($Daten as $row)
        {
            $rowmod = $rownr%3;
            if ($rowmod == 0)
            {
                echo '<tr>';
            }
        }
    }
}

```

[illegible]

[illegible]

```

$farbe = '#EEEEEE';
} else {
    $farbe = '#FFFFFF';
}
echo '<tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
$tid = $Daten[0];
$maxcount = count($Daten);
for ($i=1;$i<$maxcount;$i++)
{
    echo '<td bgcolor="'. $farbe.'">'. $Daten[$i]. '</td><td bgcolor="'. $farbe.'">&nbsp;&nbsp;&nbsp;</td>';
    //print "Si / $farbe / $Daten[$i] <br />";
}
if ($updel == "U" || $updel == "UD")
{
    echo ' <td bgcolor="'. $farbe.'">
        <form name="update" method="GET" action="tabfieldadmin.php">
            <input type="hidden" name="tid" value="'. $tid.'">
            <input type="hidden" name="upd" value="SELEFORUPD">
            <input type="SUBMIT" value="'. $lblupdate.'">
        </form>
    </td>
    <td bgcolor="'. $farbe.'">
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
    </td>';
}
if ($updel == "UD" || $updel == "D")
{
    echo ' <td bgcolor="'. $farbe.'">
        <form name="delete" method="GET" action="tabfieldadmin.php">
            <input type="hidden" name="tid" value="'. $tid.'">
            <input type="hidden" name="upd" value="DELETE">
            <input type="SUBMIT" value="'. $lbldelete.'">
        </form>
    </td>
    <td bgcolor="'. $farbe.'">
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
    </td>';
}

echo '</tr>';
$rownr++;
}
echo '</table>';
}

//*****

public function GETLabel($label,$format)
{
    print "<$format>$label</$format>";
}

//*****

public function GETRow()
{
    print "<p>&nbsp;&nbsp;&nbsp;&nbsp;&~</p>";
}

//*****

public function GETDataRow($data)
{
    print "&nbsp;&nbsp;&~$data&nbsp;&nbsp;&~<br />";
}

//*****

public function GETDataLink($file,$data)
{
    echo '<a href="http://'. $_SERVER['SERVER_NAME'].' /iFloraPublic/public/'. $file.'">'. $data.'</a><br />';
}

//*****

```

[illegible]


```

//*****

public function GETLink_old($file,$lbSelect,&$arrHIDDEN)
{
    // Formular BEGIN
    echo '<form method="GET" action="'. $file. '">
        <p>';

    if (is_array($arrHIDDEN))
    {
        foreach ($arrHIDDEN as $key => $value)
        {
            echo '<input type="hidden" name="'. $key. '" value="'. $value. '">';
        }
    }

    echo '    <INPUT TYPE="SUBMIT" VALUE="'. $lbSelect. '">
        </p>
    </form>';

}

//*****

public function GETLink($file,$lbSelect)
{
    // Formular BEGIN
    echo '<p>
        <a href="http://'. $_SERVER['SERVER_NAME']. '/iFloraPublic/public/'. $file. '">'. $lbSelect. '</a>
    </p>';

}

//*****Formular*****

public function GETFormularHeader($label)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
    echo '<tr id="frmLevel1"><td colspan="3">'. $label. '&nbsp;&nbsp;&nbsp;';</td></tr>';
    echo '<tr id="frmLevel2"><td colspan="3">Stammdaten&nbsp;&nbsp;&nbsp;';</td></tr>';
}

//*****Formular*****

public function GETFormularHeaderAufnahme($label)
{
    echo '<table border="1" cellspacing="0" cellpadding="0" align="center" ><tr><td colspan="2">';
    echo '<table border="0" cellspacing="0" cellpadding="0" align="center">';
    echo '<tr id="frmLevel1"><td colspan="3">'. $label. '&nbsp;&nbsp;&nbsp;';</td></tr>';
    echo '<tr id="frmLevel2"><td colspan="3">Aufnahmefläche&nbsp;&nbsp;&nbsp;';</td></tr>';
}

//*****Formular*****

public function GETFormRowStyle2($label)
{
    echo '<tr id="frmLevel2"><td colspan="3">'. $label. '&nbsp;&nbsp;&nbsp;';</td></tr>';
}

//*****Formular*****

public function GETFormRowStyle3($label)
{
    echo '<tr id="frmLevel3"><td colspan="3">'. $label. '&nbsp;&nbsp;&nbsp;';</td></tr>';
}

//*****Formular*****

public function GETFormRowStyle4($label,$value,$parid)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $label. '&nbsp;&nbsp;&nbsp;';</td>';
    echo '<td colspan="1"><input type="text" value="'. $value. '" name="PARID'. $parid. '">';
}

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" readonly>&nbsp;</td>';
echo '<td valign="top"><input type="button" onClick="javascript:help(\'$.$.parid.\')" value="?">&nbsp;</td>';
echo '</tr>';
}

//*****Formular*****

public function GETFormRowStyle6($label,$value,$parid,$file)
{
    echo '<tr id="frmLevel4">';
    echo '<td align="center" colspan="3"><span title="Parameter">';
    echo '<a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$file.'">'. $value.'</a>&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

public function GETFormRowArt($Art,$value)
{
    echo '<tr id="frmLevel4">';
    echo '<td><span title="Parameter" STYLE="cursor:pointer; cursor:hand;">'. $Art.'&nbsp;</td>';
    echo '<td colspan="1"><input type="text" value=""'. $value.'';
    echo 'maxlength="32" title="X(32)" STYLE="cursor:pointer; cursor:hand;" size="32" style="" readonly>&nbsp;</td>';
    echo '</tr>';
}

//*****Formular*****

public function GETFormularFooter()
{
    echo '</table></tr></table>';
}

//*****Formular*****

public function GETFormDLPrintLink($excelfile,$pdffile,$asciifile)
{
    echo '<table><tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>';
    echo '<tr><td><b>DOWNLOAD: </b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';

    if ($excelfile <> "0")
    {
        echo '<td><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$excelfile.'">Excel</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
    if ($pdffile <> "0")
    {
        echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$pdffile.'">PDF</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
    if ($asciifile <> "0")
    {
        echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$asciifile.'">Text</a></td>';
    }

    echo '</tr></table>';
}

//*****Formular*****

public function GETFormDLPrintLinkAF($excelfile,$pdffile,$asciifile,$orifile)
{
    echo '<table><tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>';
    echo '<tr><td><b>DOWNLOAD: </b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';

    if ($excelfile <> "0")
    {
        echo '<td><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$excelfile.'">Excel</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
    if ($pdffile <> "0")
    {
        echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/iFloraPublic/public/'.$pdffile.'">PDF</a></td>';
        echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
    }
}
```

```

if ($Sasciifile <> "0")
{
    echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$Sasciifile.'">Text</a></td>';
    echo '<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>';
}

if ($Sorifile <> "0")
{
    echo '<td><b></b><a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$Sorifile.'">ORI-Datei</a></td>';
}

echo '</tr></table>';
}

//*****Formular*****

public function GETFormPrintLink($file,$label)
{
    echo '<p><b>VERWEIS (Link):&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</b>';
    <a href="http://'.$_SERVER['SERVER_NAME'].'/.iFloraPublic/public/'.$file.'">'. $label.'</a></p>';
}

//*****ENDE*****

}

?>

```

9.3.2.1.3 LayoutManager.php

```

<?php

// This script is called by all Controller-Scripts to print Layout
// DHTML DB elements.
//
// history
// 2005-10-01 1.0 SH begin GETSelect
//
// description
// GETFloraheader => GET Layout Header
// GETFlorafooter => Get Layout Footer

class LAYOUTManager
{

//*****

function GETFloraheader($ueberschrift,$news,$lblNavigation,$lblSelect,$lblNews,&$arrLINK,&$arrLAYOUT)
{
    echo '<html>
    <head>
        <title>iFLORA Public</title>
        <link rel="STYLESHEET" type="text/css" href="/style/newiFlora.css">
    </head>
    <body bgcolor="#FFFFFF" link="#FFCC00" vlink="#FFCC00" alink="#FFCC00" text="#000000">
    <table border="0" cellpadding="0" cellspacing="0" width="100%" bgcolor="#004080" text="#FFFFFF">
        <colgroup>
            <col width="200">
            <col>
        </colgroup>
        <tr>
            <td colspan="2">
                <table border="0" cellpadding="10" cellspacing="0">
                    <tr>
                        <td>
                            <h1 style="color:#FFFFFF">iFLORA '. $ueberschrift.'</h1>
                        </td>
                    </tr>
                </table>
            </td>
            <tr>
                <td valign="top" width="200">

```

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

Stephan Hordynski (Matrikelnr.: a9227169)

```
pop(file,\help\','\width=400,height=450,scrollbars=yes,dependent=yes,location=no,resizable=yes\')
    }
    -->
    </script>
</head>
<body>;
}

//*****

function GETFooter()
{
    echo ' </body>
    </html>;'
}

//*****

function GETFlorafooter()
{
    echo '          </td>
          </tr>
          </table>
          </td>
          </tr>
          </table>
          </body>
          </html>;'
}

//*****

public function GETLabel($label,$format)
{
    print "<$format>$label</format>";
}

//*****

public function GETRow()
{
    print "<p />";
}

//*****

public function GETReferenceSelect(&$arrREFERENCE)
{
    foreach ($arrREFERENCE as $Daten)
    {
        print '<a href="' . $Daten["ref"] . "><b>' . $Daten["name"] . '</b></a><br>';
    }
}

}
?>
```

9.3.2.2 Controller für einzelnen Seiten

9.3.2.2.1 Startseite – Rahmen mit Menü (index.php)

```
<?php

//*****

function GET_HTMLHEAD($version,$db)
{
    print '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
    <html lang="de">
    <head>
    <title>iFlora Public@' . $db . ' (R' . $version . ')</title>
    <link rel="STYLESHEET" type="text/css" href="/style/newiFlora.css">
    <link rel="STYLESHEET" type="text/css" href="/tree/dtree.css">
```

```

<script language="JavaScript">
<!--

var loaded = 0;
var counter = 0;

function init() {
    this.focus();
    var loaded = 1;
}

function destroy() {
    loaded = 0;
}

function popup(url,features) {
    counter ++;
    fenster = window.open(url,'window'+counter,features);
    fenster.focus();
}

//-->
</script>
</head>
<body style="margin: 2px 2px 2px 2px;" onLoad="init()" onunload="destroy()">
<table width="100%" height="100%" border="0" cellspacing="0" cellpadding="0">
<tr bgcolor="#008000" text="#FFFFFF" height="57">
    <td width="100%" align="left" class="header1">&nbsp;iFlora Öffentlicher Bereich</td>
    <td width="150" align="right">&nbsp;</td>
    <td width="120" align="center" class="smalltext" style="background-color:rgb(220, 20, 60);">
        <strong>iFloraPublic@'. $db.'<br />R'. $version.'</strong>
    </td>
</tr>
<tr>
    <td colspan=3 height="2"></td>
</tr>
<tr>
    <td colspan=3 height="15" background="/layout/hgwasser1.jpg"></td>
</tr>
<tr>
    <td valign="top" colspan=3>
        <table width="100%" height="100%" border="0" cellspacing="0" style="padding: 2px; padding-left: 0px;">
            <tr>';

}

//*****

function GET_HTMLFOOT()
{
    print '    </tr>
        </table>
        </td>
    </tr>
    <tr height="20" >
        <td id="footer">&nbsp;<strong>iFlora Administration'. $usr.'</strong>&nbsp;</td>
        <td id="footer" align="right" colspan=3>&copy;&nbsp;<i>sh</i>Informatics&nbsp;</td>
    </tr>
    </table>
</body>
</html>';
}

//*****

function GET_HTMLTREE($arobjekt)
{
    print '<td width=180 bgcolor="#F0F0F0" valign="top">&nbsp;<br />';
    print ' <a href="https://'. $ _SERVER['SERVER_NAME']. '/iflora/" title="iFlora Administration">Administration</a><br />&nbsp;<br />';
    print ' <script src="/tree/dtree.js" type="text/javascript"></script>
        <script language="JavaScript">
        <!--

        mytree = new dTree(\'mytree\');
        mytree.config.useCookies=false;
        mytree.config.target=\'work\';
    
```

```

mytree.config.folderLinks=false;
mytree.config.useIcons=true;

mytree.icon = {
    root           : \'./tree/images/base.gif\',
    folder         : \'./tree/images/folder.gif\',
    folderOpen     : \'./tree/images/folderopen.gif\',
    node           : \'./tree/images/page.gif\',
    empty          : \'./tree/images/empty.gif\',
    line           : \'./tree/images/line.gif\',
    join           : \'./tree/images/join.gif\',
    joinBottom     : \'./tree/images/joinbottom.gif\',
    plus           : \'./tree/images/plus.gif\',
    plusBottom     : \'./tree/images/plusbottom.gif\',
    minus          : \'./tree/images/minus.gif\',
    minusBottom    : \'./tree/images/minusbottom.gif\',
    nlPlus         : \'./tree/images/nolines_plus.gif\',
    nlMinus        : \'./tree/images/nolines_minus.gif\'
};

mytree.add(0,-1,\'iFlora\',\'javascript: void(0);\');
$counter = 1;
$knoten = $counter;
print \'mytree.add(\'.$counter.\',0,\'Auswahl\',\'javascript: void(0);\,\'Auswahl\');\';
$counter++;
print \'mytree.add(\'.$counter.\',\'.$knoten.\',\'Auswahl Geographie\',\'./public/OOPSelGeographie.php\',\'Auswahl Geographie\');\';
$counter++;
print \'mytree.add(\'.$counter.\',\'.$knoten.\',\'Auswahl Region\',\'./public/OOPSelRegion.php\',\'Auswahl Region\');\';
$counter++;
print \'mytree.add(\'.$counter.\',\'.$knoten.\',\'Auswahl Naturraum\',\'./public/OOPSelNaturraum.php\',\'Auswahl Naturraum\');\';
$counter++;
print \'mytree.add(\'.$counter.\',\'.$knoten.\',\'Auswahl Großraum\',\'./public/OOPSelGrossraum.php\',\'Auswahl Großraum\');\';
$counter++;
$knoten = $counter;
print \'mytree.add(\'.$counter.\',0,\'Objekte\',\'javascript: void(0);\,\'Objekte\');\';
$counter++;
if (is_array($arobjekt))
{
    foreach ($arobjekt as $Objekt)
    {
        print '
mytree.add(\'.$counter.\',\'.$knoten.\',\'.$Objekt[\'NAME\']\',\'./public/OOPSelObjekt.php?obtid=\'.$Objekt[\'TID\']\'&obaname=\'.$Objekt[\'NAM
E\']\',\'.$Objekt[\'NAME\']\');\';
        $counter++;
    }
    $arobjekt = NULL;
}
$counter++;
$knoten = $counter;
print \'mytree.add(\'.$counter.\',0,\'Auswertungen\',\'javascript: void(0);\,\'Objekte\');\';
$counter++;
print \'mytree.add(\'.$counter.\',\'.$knoten.\',\'Aufnahmen\',\'./work.html\',\'Regionen\');\';
$counter++;
$knoten = $counter;
print \'mytree.add(\'.$counter.\',0,\'Downloads\',\'javascript: void(0);\,\'Objekte\');\';
$counter++;
print \'mytree.add(\'.$counter.\',\'.$knoten.\',\'Aufnahmedaten\',\'./work.html\',\'Regionen\');\';

print \'document.write(mytree);
//-->
</script>;
print \' </td>;
}

//*****

function GET_HTMLData()
{
    print \'<td><iframe src=\'./work.html\' name=\'work\' id=\'work\' width=\'100%\' height=\'100%\'
        marginwidth=\'0\' marginheight=\'0\' frameborder=\'0\' border=\'1\'></iframe></td>;\';
}

//***** MAIN *****

// INCLUDE

```

```
include_once('./lib/DBManager.php');
include_once('./../config_iflora/sqlpublic.php.inc');

// INIT Konstanten
define("version","0.6");
define("database",$sql_db);

// INIT DBManager Objekt
$doBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

// GET DB-Daten - Objekte
$SQLquery = "SELECT DISTINCT TID, OBANAME AS NAME "
            ."FROM afn_objektart "
            ."ORDER BY OBANAME";
$doBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);

GET_HTMLHEAD(version,database);
GET_HTMLTREE($arrSQL);
GET_HTMLData();
GET_HTMLFOOT($in_usr);

?>
```

9.3.2.2.2 Startseite – Work (im Rahmen) (work.html)

```
<html>
<head>
</head>
<body>
</body>
</html>
```

9.3.2.2.3 Controller – „Auswahl Geographie“ (OOPSelGeographie.php)

```
<?php

// INCLUDE
include_once('./lib/DBManager.php');
include_once('./lib/VIEWManager.php');
include_once('./lib/LAYOUTManager.php');
include_once('./../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['prjtid'] != ""){$in_prjtid=$_REQUEST['prjtid'];} else {$in_prjtid = 0;};
if($_REQUEST['prjname'] != ""){$in_prjname=$_REQUEST['prjname'];} else {$in_prjname = 0;};
if($_REQUEST['bdltid'] != ""){$in_bdltid=$_REQUEST['bdltid'];} else {$in_bdltid = 0;};
if($_REQUEST['bdlname'] != ""){$in_bdlname=$_REQUEST['bdlname'];} else {$in_bdlname = 0;};
if($_REQUEST['beztid'] != ""){$in_beztid=$_REQUEST['beztid'];} else {$in_beztid = 0;};
if($_REQUEST['bezname'] != ""){$in_bezname=$_REQUEST['bezname'];} else {$in_bezname = 0;};
if($_REQUEST['gemtid'] != ""){$in_gemtid=$_REQUEST['gemtid'];} else {$in_gemtid = 0;};
if($_REQUEST['gemname'] != ""){$in_gemname=$_REQUEST['gemname'];} else {$in_gemname = 0;};
if($_REQUEST['auftid'] != ""){$in_auftid=$_REQUEST['auftid'];} else {$in_auftid = 0;};
if($_REQUEST['aufname'] != ""){$in_aufname=$_REQUEST['aufname'];} else {$in_aufname = 0;};
if($_REQUEST['aufAtid'] != ""){$in_aufAtid=$_REQUEST['aufAtid'];} else {$in_aufAtid = 0;};
if($_REQUEST['aufAname'] != ""){$in_aufAname=$_REQUEST['aufAname'];} else {$in_aufAname = 0;};
if($_REQUEST['obtid'] != ""){$in_obtid=$_REQUEST['obtid'];} else {$in_obtid = 0;};

$file = "OOPSelGeographie.php";
$HPueberschrift = "GEOGRAPHIE Aufnahmeauswahl (OOP)";
$lbSelect = "Auswahl";
$gemStatus = 0;
$lnkfile = "OOPFormStammdata.php";
$lnklbSelAufnahme = "Aufnahme Stammdaten";
$lnklbSelect = "Aufnahmefläche Stammdaten";

$arrSQL = NULL;
$arrInputSQL = NULL;
$arrUPDATE = NULL;
$arrINSERT = NULL;
$arrSELHidden = NULL;
$arrHISTDaten = NULL;
$arrAUFNDATEN = NULL;

// INIT DBManager Objekt
```



```

SoDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
SoVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
SoLTM = new LAYOUTManager;

// INIT SelectParameter
if ($sin_prjtid > 0)
{
    if ($sin_bdltid > 0)
    {
        if ($sin_beztid > 0)
        {
            if ($sin_gemtid > 0)
            {
                if ($sin_auftid > 0)
                {
                    if ($sin_aufAtid > 0)
                    {
                        // SET allgemeine SQLQuery
                        $SQLQuery = "SELECT DISTINCT TID, OBJNAME AS NAME "
                        . "FROM afn_objekt "
                        . "WHERE TID = $sin_auftid ";

                        // GET DB-Daten
                        $SQLQueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                        . "FROM afn_objekt "
                        . "WHERE TID = $sin_aufAtid ";

                        $aufAname = $oDBM->GETSelectSQLDatum($SQLQueryDatum);
                        $arrSELHidden["aufAtid"] = "$sin_aufAtid";
                        $arrSELHidden["aufAname"] = "$aufAname";
                        $arrHISTDaten["PROJEKT:"] = "$prjname";
                        $arrHISTDaten["BUNDESLAND:"] = "$sin_bdlname";
                        $arrHISTDaten["BEZIRK:"] = "$sin_beztid";
                        $arrHISTDaten["GEMEINDE:"] = "$sin_gemname";
                        $arrHISTDaten["AUFNAHME:"] = "$sin_aufname";
                        $arrHISTDaten["AUFNAHMEFLÄCHE:"] = "$aufAname";
                        $linkobjname = $aufAname;
                        $linkobatid = 4;
                        $linkobjid = $sin_aufAtid;
                        $gemStatus = 1;
                    } else {
                        $parameter = "aufAtid";
                        // SET allgemeine SQLQuery
                        $SQLQuery = "SELECT DISTINCT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
                        . "FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
                        . "WHERE o.obaid = a.TID "
                        . "AND a.OBAKUERZEL = 'ANF' "
                        . "AND ia.ISAKUERZEL = 'AAF' "
                        . "AND ia.TID = i.ISAID "
                        . "AND i.ISINOBJID = $sin_auftid "
                        . "AND o.TID = i.OBJID "
                        . "ORDER BY o.OBJNAME";

                        // GET DB-Daten
                        $SQLQueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                        . "FROM afn_objekt "
                        . "WHERE TID = $sin_auftid ";

                        $aufname = $oDBM->GETSelectSQLDatum($SQLQueryDatum);
                        $arrSELHidden["prjtid"] = "$sin_prjtid";
                        $arrSELHidden["prjname"] = "$prjname";
                        $arrSELHidden["bdltid"] = "$sin_bdltid";
                        $arrSELHidden["bdlname"] = "$sin_bdlname";
                        $arrSELHidden["beztid"] = "$sin_beztid";
                        $arrSELHidden["bezname"] = "$sin_beztid";
                        $arrSELHidden["gemtid"] = "$sin_gemtid";
                        $arrSELHidden["gemname"] = "$sin_gemname";
                        $arrSELHidden["auftid"] = "$sin_auftid";
                        $arrSELHidden["aufname"] = "$aufname";
                        $arrHISTDaten["PROJEKT:"] = "$prjname";
                        $arrHISTDaten["BUNDESLAND:"] = "$sin_bdlname";
                        $arrHISTDaten["BEZIRK:"] = "$sin_beztid";
                        $arrHISTDaten["GEMEINDE:"] = "$sin_gemname";
                        $arrHISTDaten["AUFNAHME:"] = "$aufname";
                        $labelSELECT = "AUFNAHMEFLÄCHE: ";
                        $linkobjname = $aufname;
                        $linkobatid = 5;
                    }
                }
            }
        }
    }
}

```

Diplomarbeit

Computerunterstützte Integration vegetationsökologischer Daten über Biologie, Raum und Zeit

```
$linkobjid = $in_auftid;
$gemStatus = 2;
}
} else {
$parameter = "auftid";
$SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
            ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
            ."WHERE o.obaid = a.TID "
            ."AND a.OBNAME = 'AUFNAHME' "
            ."AND ia.ISAKUERZEL = 'POL' "
            ."AND ia.TID = i.ISAID "
            ."AND i.ISINOBJID = $in_gemtid "
            ."AND o.TID = i.OBJID "
            ."ORDER BY o.OBJNAME";
$SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                ."FROM afn_objekt "
                ."WHERE TID = $in_gemtid ";
$gemname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
$arrSELHidden["prjtid"] = "$in_prjtid";
$arrSELHidden["prjname"] = "$prjname";
$arrSELHidden["bdltid"] = "$in_bdltid";
$arrSELHidden["bdlname"] = "$in_bdlname";
$arrSELHidden["beztid"] = "$in_beztid";
$arrSELHidden["bezname"] = "$bezname";
$arrSELHidden["gemtid"] = "$in_gemtid";
$arrSELHidden["gemname"] = "$gemname";
$arrHISTDaten["PROJEKT:"] = "$prjname";
$arrHISTDaten["BUNDESLAND:"] = "$in_bdlname";
$arrHISTDaten["BEZIRK:"] = "$in_bezname";
$arrHISTDaten["GEMEINDE:"] = "$gemname";
$labelSELECT = "AUFNAHME:";
}
} else {
$parameter = "gemtid";
$SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
            ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
            ."WHERE o.obaid = a.TID "
            ."AND a.OBNAME = 'GEMEINDE' "
            ."AND ia.ISAKUERZEL = 'POL' "
            ."AND ia.TID = i.ISAID "
            ."AND i.ISINOBJID = $in_beztid "
            ."AND o.TID = i.OBJID "
            ."ORDER BY o.OBJNAME";
$SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                ."FROM afn_objekt "
                ."WHERE TID = $in_beztid ";
$bezname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
$arrSELHidden["prjtid"] = "$in_prjtid";
$arrSELHidden["prjname"] = "$prjname";
$arrSELHidden["bdltid"] = "$in_bdltid";
$arrSELHidden["bdlname"] = "$in_bdlname";
$arrSELHidden["beztid"] = "$in_beztid";
$arrSELHidden["bezname"] = "$bezname";
$arrHISTDaten["PROJEKT:"] = "$prjname";
$arrHISTDaten["BUNDESLAND:"] = "$in_bdlname";
$arrHISTDaten["BEZIRK:"] = "$bezname";
$labelSELECT = "GEMEINDE:";
}
} else {
$parameter = "beztid";
$SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
            ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
            ."WHERE o.obaid = a.TID "
            ."AND a.OBNAME = 'BEZIRK' "
            ."AND ia.ISAKUERZEL = 'POL' "
            ."AND ia.TID = i.ISAID "
            ."AND i.ISINOBJID = $in_bdltid "
            ."AND o.TID = i.OBJID "
            ."ORDER BY o.OBJNAME";
$SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                ."FROM afn_objekt "
                ."WHERE TID = $in_bdltid ";
$bdlname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
$arrSELHidden["prjtid"] = "$in_prjtid";
$arrSELHidden["prjname"] = "$prjname";
```

```

    $arrSELHidden["bdltid"] = "$in_bdltid";
    $arrSELHidden["bdlname"] = "$bdlname";
    $arrHISTDaten["PROJEKT:"] = "$prjname";
    $arrHISTDaten["BUNDESLAND:"] = "$bdlname";
    $labelSELECT = "BEZIRK:";
}
} else {
    $parameter = "bdltid";
    $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM afn_objekt o, afn_objektart a "
        ."WHERE o.obaid = a.TID "
        ."AND a.OBANAME = 'BUNDESLAND' "
        ."ORDER BY o.OBJNAME";
    $SQLqueryDatum = "SELECT DISTINCT PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE TID = $in_prjtid ";
    $prjname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrHISTDaten["PROJEKT:"] = "$prjname";
    $labelSELECT = "BUNDESLAND:";
}
} else {
    $parameter = "prjtid";
    $SQLquery = "SELECT DISTINCT TID, PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE PRJNAME <> 'ADMIN' "
        ."ORDER BY PRJNAME";
    $labelSELECT = "PROJEKT:";
}

// GET DB-Daten
$oDBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);

// Layoutbausteine zusammenstellen
$oLTM->GETHeader();
$oVWM->GETLabel($HPueberschrift,"H1");
$oVWM->GETIDTable(&$arrHISTDaten);
$oVWM->GETLabel($labelSELECT,"H3");
If ($gemStatus == 1)
{
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obaid='.$linkobaid.'&objname='.$linkobjname;
    $oVWM->GETLink($lnkfile,$lnklbSelect);
} elseif ($gemStatus == 2){
    $oVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obaid='.$linkobaid.'&objname='.$linkobjname;
    $oVWM->GETLink($lnkfile,$lnklbSelAufnahme);
} else {
    $oVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
}
}
$oLTM->GETFooter();
?>

```

9.3.2.2.4 Controller – „Auswahl Region“ (OOPSelRegion.php)

```
<?php
```

```

// INCLUDE
include_once('..lib/DBManager.php');
include_once('..lib/VIEWManager.php');
include_once('..lib/LAYOUTManager.php');
include_once('..lib/config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['prjtid'] != ""){$in_prjtid=$_REQUEST['prjtid'];} else {$in_prjtid = 0;};
if($_REQUEST['prjname'] != ""){$in_prjname=$_REQUEST['prjname'];} else {$in_prjname = 0;};
if($_REQUEST['regtid'] != ""){$in_regtid=$_REQUEST['regtid'];} else {$in_regtid = 0;};
if($_REQUEST['regname'] != ""){$in_regname=$_REQUEST['regname'];} else {$in_regname = 0;};

if($_REQUEST['beztid'] != ""){$in_beztid=$_REQUEST['beztid'];} else {$in_beztid = 0;};
if($_REQUEST['bezname'] != ""){$in_bezname=$_REQUEST['bezname'];} else {$in_bezname = 0;};
if($_REQUEST['gemtid'] != ""){$in_gemtid=$_REQUEST['gemtid'];} else {$in_gemtid = 0;};

```

```

if($REQUEST['gemname'] != ""){$in_gemname=$REQUEST['gemname'];} else {$in_gemname = 0;};

if($REQUEST['auftid'] != ""){$in_auftid=$REQUEST['auftid'];} else {$in_auftid = 0;};
if($REQUEST['aufname'] != ""){$in_aufname=$REQUEST['aufname'];} else {$in_aufname = 0;};
if($REQUEST['aufAtid'] != ""){$in_aufAtid=$REQUEST['aufAtid'];} else {$in_aufAtid = 0;};
if($REQUEST['aufAname'] != ""){$in_aufAname=$REQUEST['aufAname'];} else {$in_aufAname = 0;};

$file = "OOPSelRegion.php";
$HPueberschrift = "REGION Aufnahmeauswahl (OOP)";
$lbSelect = "Auswahl";
$gemStatus = 0;
$lnkfile = "OOPFormStammdata.php";
$lnklbSelect = "Aufnahmefläche Stammdaten";
$lnklbSelAufnahme = "Aufnahme Stammdaten";

$arrAUFNDATEN = NULL;
$arrSQL = NULL;
$arrInputSQL = NULL;
$arrUPDATE = NULL;
$arrINSERT = NULL;
$arrSELHidden = NULL;
$arrHISTDaten = NULL;

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$oVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

// INIT SelectParameter
if ($in_prjtid > 0)
{
    if ($in_regtid > 0)
    {
        if ($in_auftid > 0)
        {
            if ($in_aufAtid > 0)
            {
                $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                    ."FROM afn_objekt "
                    ."WHERE TID = $in_aufAtid ";
                $aufAname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
                $arrSELHidden["aufAtid"] = "$in_aufAtid";
                $arrSELHidden["aufAname"] = "$aufAname";
                $arrHISTDaten["PROJEKT:"] = "$prjname";
                $arrHISTDaten["REGION:"] = "$in_regname";
                $arrHISTDaten["AUFNAHME:"] = "$in_aufname";
                $arrHISTDaten["AUFNAHMEFLÄCHE:"] = "$aufAname";
                $linkobjname = $aufAname;
                $linkobatid = 4;
                $linkobjid = $in_aufAtid;
                $linkfrmid = 2;
                $gemStatus = 1;
            } else {
                $parameter = "aufAtid";
                $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME, o.OBAID "
                    ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
                    ."WHERE o.obaid = a.TID "
                    ."AND a.OBAKUERZEL = 'ANF' "
                    ."AND ia.ISANAME = 'Aufnahme Beziehung' "
                    ."AND ia.TID = i.ISAID "
                    ."AND i.ISINOBJID = $in_auftid "
                    ."AND o.TID = i.OBJID "
                    ."ORDER BY o.OBJNAME";
                $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                    ."FROM afn_objekt "
                    ."WHERE TID = $in_auftid ";
                $aufname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
                $arrSELHidden["prjtid"] = "$in_prjtid";
                $arrSELHidden["prjname"] = "$prjname";
                $arrSELHidden["regtid"] = "$in_regtid";
                $arrSELHidden["regname"] = "$in_regname";
                $arrSELHidden["auftid"] = "$in_auftid";
            }
        }
    }
}

```

```

    $arrSELHidden["aufname"] = "$aufname";
    $arrHISTDaten["PROJEKT:"] = "$prjname";
    $arrHISTDaten["REGION:"] = "$in_regname";
    $arrHISTDaten["AUFNAHME:"] = "$aufname";
    $labelSELECT = "AUFNAHMEFLÄCHE:";
    $linkobjname = $aufname;
    $linkobjid = $in_auftid;
    $linkobatid = 5;
    $linkfrmid = 1;
    $gemStatus = 2;
}
} else {
    $parameter = "auftid";
    $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME, o.OBAID "
        ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
        ."WHERE o.obaid = a.TID "
        ."AND a.OBANAME = 'AUFNAHME' "
        ."AND ia.ISANAME = 'Region' "
        ."AND ia.TID = i.ISAID "
        ."AND i.ISINOBJID = $in_regtid "
        ."AND o.TID = i.OBJID "
        ."ORDER BY o.OBJNAME";
    $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
        ."FROM afn_objekt "
        ."WHERE TID = $in_regtid ";
    $regname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrSELHidden["regtid"] = "$in_regtid";
    $arrSELHidden["regname"] = "$in_regname";
    $arrSELHidden["regname"] = "$regname";
    $arrHISTDaten["PROJEKT:"] = "$in_prjname";
    $arrHISTDaten["REGION:"] = "$regname";
    $labelSELECT = "AUFNAHME:";
}
} else {
    $parameter = "regtid";
    $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM afn_objekt o, afn_objektart a "
        ."WHERE o.obaid = a.TID "
        ."AND a.OBANAME = 'REGION' "
        ."ORDER BY o.OBJNAME";
    $SQLqueryDatum = "SELECT DISTINCT PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE TID = $in_prjtid ";
    $prjname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrHISTDaten["PROJEKT:"] = "$prjname";
    $labelSELECT = "REGION:";
}
} else {
    $parameter = "prjtid";
    $SQLquery = "SELECT DISTINCT TID, PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE PRJNAME <> 'ADMIN' "
        ."ORDER BY PRJNAME";
    $labelSELECT = "PROJEKT:";
}

// GET DB-Daten
$oDBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);

// Layoutbausteine zusammenstellen
$oLTM->GETHeader();
$oVWM->GETLabel($HPueberschrift,"H1");
$oVWM->GETIDTable(&$arrHISTDaten);
$oVWM->GETLabel($labelSELECT,"H3");
If ($gemStatus == 1)
{
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obatid='.$linkobatid.'&objname='.$linkobjname.'&frmid='.$linkfrmid;
    $oVWM->GETLink($lnkfile,$lnklbSelect);
} elseif ($gemStatus == 2){
    $oVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
}

```

```

$lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obtid='.$linkobtid.'&objname='.$linkobjname.'&frmid='.$linkfrmid;
SoVWM->GETLink($lnkfile,$lnklibSelAufnahme);
} else {
    SoVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$sarrSQL,$sarrSELHidden);
}
SoLTM->GETFooter();

```

?>

9.3.2.2.5 Controller – „Auswahl Naturraum“ (OOPSelNaturraum.php)

<?php

```

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['prjtid'] != ""){$in_prjtid=$_REQUEST['prjtid'];} else {$in_prjtid = 0;};
if($_REQUEST['prjname'] != ""){$in_prjname=$_REQUEST['prjname'];} else {$in_prjname = 0;};
if($_REQUEST['regtid'] != ""){$in_regtid=$_REQUEST['regtid'];} else {$in_regtid = 0;};
if($_REQUEST['regname'] != ""){$in_regname=$_REQUEST['regname'];} else {$in_regname = 0;};

if($_REQUEST['beztid'] != ""){$in_beztid=$_REQUEST['beztid'];} else {$in_beztid = 0;};
if($_REQUEST['bezname'] != ""){$in_bezname=$_REQUEST['bezname'];} else {$in_bezname = 0;};
if($_REQUEST['gemtid'] != ""){$in_gemtid=$_REQUEST['gemtid'];} else {$in_gemtid = 0;};
if($_REQUEST['gemname'] != ""){$in_gemname=$_REQUEST['gemname'];} else {$in_gemname = 0;};

if($_REQUEST['auftid'] != ""){$in_auftid=$_REQUEST['auftid'];} else {$in_auftid = 0;};
if($_REQUEST['aufname'] != ""){$in_aufname=$_REQUEST['aufname'];} else {$in_aufname = 0;};
if($_REQUEST['aufAtid'] != ""){$in_aufAtid=$_REQUEST['aufAtid'];} else {$in_aufAtid = 0;};
if($_REQUEST['aufAname'] != ""){$in_aufAname=$_REQUEST['aufAname'];} else {$in_aufAname = 0;};

$file = "OOPSelNaturraum.php";
$HPueberschrift = "NATURRAUM Aufnahmeauswahl (OOP)";
$lbSelect = "Auswahl";
$gemStatus = 0;
$lnkfile = "OOPFormStammdaten.php";
$lnklibSelAufnahme = "Aufnahme Stammdaten";
$lnklibSelect = "Aufnahmefläche Stammdaten";

$sarrSQL = NULL;
$sarrInputSQL = NULL;
$sarrUPDATE = NULL;
$sarrINSERT = NULL;
$sarrSELHidden = NULL;
$sarrHISTDaten = NULL;

// INIT DBManager Objekt
$soDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$soVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$soLTM = new LAYOUTManager;

// INIT SelectParameter
if ($in_prjtid > 0)
{
    if ($in_regtid > 0)
    {
        if ($in_auftid > 0)
        {
            if ($in_aufAtid > 0)
            {
                $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                    ."FROM afn_objekt "
                    ."WHERE TID = $in_aufAtid ";
                $aufAname = $soDBM->GETSelectSQLDatum($SQLqueryDatum);
                $sarrSELHidden["aufAtid"] = "$in_aufAtid";
                $sarrSELHidden["aufAname"] = "$aufAname";
                $sarrHISTDaten["PROJEKT:"] = "$prjname";
            }
        }
    }
}

```

```

$arrHISTDaten["REGION:" ] = "$in_regname";
$arrHISTDaten["AUFNAHME:" ] = "$in_aufname";
$arrHISTDaten["AUFNAHMEFLÄCHE:" ] = "$aufAname";
$linkobjname = $aufAname;
$linkobjid = $in_aufAtid;
$linkobatid = 4;
$linkfrmid = 2;
$gemStatus = 1;
} else {
    $parameter = "aufAtid";
    $SQLQuery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM   afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
        ."WHERE  o.obaid = a.TID "
        ."AND    a.OBAKUERZEL = 'ANF' "
        ."AND    ia.ISANAME = 'Aufnahme Beziehung' "
        ."AND    ia.TID = i.ISAID "
        ."AND    i.ISINOBJID = $in_auftid "
        ."AND    o.TID = i.OBJID "
        ."ORDER BY o.OBJNAME";
    $SQLQueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
        ."FROM afn_objekt "
        ."WHERE TID = $in_auftid ";
    $aufname = $oDBM->GETSelectSQLDatum($SQLQueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrSELHidden["regtid"] = "$in_regtid";
    $arrSELHidden["regname"] = "$in_regname";
    $arrSELHidden["auftid"] = "$in_auftid";
    $arrSELHidden["aufname"] = "$aufname";
    $arrHISTDaten["PROJEKT:" ] = "$prjname";
    $arrHISTDaten["NATURRAUM:" ] = "$in_regname";
    $arrHISTDaten["AUFNAHME:" ] = "$aufname";
    $labelSELECT = "AUFNAHMEFLÄCHE:";
    $linkobjname = $aufname;
    $linkobjid = $in_auftid;
    $linkobatid = 5;
    $linkfrmid = 1;
    $gemStatus = 2;
}
} else {
    $parameter = "auftid";
    $SQLQuery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM   afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
        ."WHERE  o.obaid = a.TID "
        ."AND    a.OBANAME = 'AUFNAHME' "
        ."AND    ia.ISANAME = 'Naturraum' "
        ."AND    ia.TID = i.ISAID "
        ."AND    i.ISINOBJID = $in_regtid "
        ."AND    o.TID = i.OBJID "
        ."ORDER BY o.OBJNAME";
    $SQLQueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
        ."FROM afn_objekt "
        ."WHERE TID = $in_regtid ";
    $regname = $oDBM->GETSelectSQLDatum($SQLQueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrSELHidden["regtid"] = "$in_regtid";
    $arrSELHidden["regname"] = "$in_regname";
    $arrSELHidden["regname"] = "$regname";
    $arrHISTDaten["PROJEKT:" ] = "$in_prjname";
    $arrHISTDaten["NATURRAUM:" ] = "$regname";
    $labelSELECT = "AUFNAHME:";
}
} else {
    $parameter = "regtid";
    $SQLQuery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM   afn_objekt o, afn_objektart a "
        ."WHERE  o.obaid = a.TID "
        ."AND    a.OBANAME = 'NATURRAUM' "
        ."ORDER BY o.OBJNAME";
    $SQLQueryDatum = "SELECT DISTINCT PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE TID = $in_prjtid ";
    $prjname = $oDBM->GETSelectSQLDatum($SQLQueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";

```

```

    $arrSELHidden["prjname"] = "$prjname";
    $arrHISTDaten["PROJEKT"] = "$prjname";
    $labelSELECT = "NATURRAUM:";
}
} else {
    $parameter = "prjtid";
    $SQLquery = "SELECT DISTINCT TID, PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE PRJNAME <> 'ADMIN' "
        ."ORDER BY PRJNAME";
    $labelSELECT = "PROJEKT:";
}

// GET DB-Daten
$soDBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);

// Layoutbausteine zusammenstellen
$soLTM->GETHeader();
$soVWM->GETLabel($HPueberschrift,"H1");
$soVWM->GETIDTable(&$arrHISTDaten);
$soVWM->GETLabel($labelSELECT,"H3");
If ($gemStatus == 1)
{
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obtid='.$linkobtid.'&objname='.$linkobjname.'&frmid='.$linkfrmid;
    $soVWM->GETLink($lnkfile,$lnklbSelect);
} elseif ($gemStatus == 2){
    $soVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obtid='.$linkobtid.'&objname='.$linkobjname.'&frmid='.$linkfrmid;
    $soVWM->GETLink($lnkfile,$lnklbSelAufnahme);
} else {
    $soVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
}
$soLTM->GETFooter();

?>

```

9.3.2.2.6 Controller – „Auswahl Großraum“ (OOPSelGrossraum.php)

```

<?php

// INCLUDE
include_once('..lib/DBManager.php');
include_once('..lib/VIEWManager.php');
include_once('..lib/LAYOUTManager.php');
include_once('..../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['prjtid'] != ""){$in_prjtid=$_REQUEST['prjtid'];} else {$in_prjtid = 0;};
if($_REQUEST['prjname'] != ""){$in_prjname=$_REQUEST['prjname'];} else {$in_prjname = 0;};
if($_REQUEST['regtid'] != ""){$in_regtid=$_REQUEST['regtid'];} else {$in_regtid = 0;};
if($_REQUEST['regname'] != ""){$in_regname=$_REQUEST['regname'];} else {$in_regname = 0;};

if($_REQUEST['beztid'] != ""){$in_beztid=$_REQUEST['beztid'];} else {$in_beztid = 0;};
if($_REQUEST['bezname'] != ""){$in_bezname=$_REQUEST['bezname'];} else {$in_bezname = 0;};
if($_REQUEST['gemtid'] != ""){$in_gemtid=$_REQUEST['gemtid'];} else {$in_gemtid = 0;};
if($_REQUEST['gemname'] != ""){$in_gemname=$_REQUEST['gemname'];} else {$in_gemname = 0;};

if($_REQUEST['auftid'] != ""){$in_auftid=$_REQUEST['auftid'];} else {$in_auftid = 0;};
if($_REQUEST['aufname'] != ""){$in_aufname=$_REQUEST['aufname'];} else {$in_aufname = 0;};
if($_REQUEST['aufAtid'] != ""){$in_aufAtid=$_REQUEST['aufAtid'];} else {$in_aufAtid = 0;};
if($_REQUEST['aufAname'] != ""){$in_aufAname=$_REQUEST['aufAname'];} else {$in_aufAname = 0;};

$file = "OOPSelGrossraum.php";
$HPueberschrift = "GROSSRAUM Aufnahmeauswahl (OOP)";
$lbSelect = "Auswahl";
$gemStatus = 0;
$lnkfile = "OOPFormStammdaten.php";
$lnklbSelAufnahme = "Aufnahme Stammdaten";
$lnklbSelect = "Aufnahmefläche Stammdaten";

$arrSQL = NULL;
$arrInputSQL = NULL;

```



```

$arrUPDATE = NULL;
$arrINSERT = NULL;
$arrSELHidden = NULL;
$arrHISTDaten = NULL;

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$oVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

// INIT SelectParameter
if ($in_prjtid > 0)
{
    if ($in_regtid > 0)
    {
        if ($in_auftid > 0)
        {
            if ($in_aufAtid > 0)
            {
                $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                    ."FROM afn_objekt "
                    ."WHERE TID = $in_aufAtid ";
                $aufAname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
                $arrSELHidden["aufAtid"] = "$in_aufAtid";
                $arrSELHidden["aufAname"] = "$aufAname";
                $arrHISTDaten["PROJEKT:"] = "$prjname";
                $arrHISTDaten["REGION:"] = "$in_regname";
                $arrHISTDaten["AUFNAHME:"] = "$in_aufname";
                $arrHISTDaten["AUFNAHMEFLÄCHE:"] = "$aufAname";
                $linkobjname = $aufAname;
                $linkobjid = $in_aufAtid;
                $linkobatid = 4;
                $linkfrmid = 2;
                $gemStatus = 1;
            } else {
                $parameter = "aufAtid";
                $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
                    ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
                    ."WHERE o.obaid = a.TID "
                    ."AND a.OBAKUERZEL = 'ANF' "
                    ."AND ia.ISANAME = 'Aufnahme Beziehung' "
                    ."AND ia.TID = i.ISAID "
                    ."AND i.ISINOBJID = $in_auftid "
                    ."AND o.TID = i.OBJID "
                    ."ORDER BY o.OBJNAME";
                $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
                    ."FROM afn_objekt "
                    ."WHERE TID = $in_auftid ";
                $aufname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
                $arrSELHidden["prjtid"] = "$in_prjtid";
                $arrSELHidden["prjname"] = "$prjname";
                $arrSELHidden["regtid"] = "$in_regtid";
                $arrSELHidden["regname"] = "$in_regname";
                $arrSELHidden["auftid"] = "$in_auftid";
                $arrSELHidden["aufname"] = "$aufname";
                $arrHISTDaten["PROJEKT:"] = "$prjname";
                $arrHISTDaten["NATURRAUM:"] = "$in_regname";
                $arrHISTDaten["AUFNAHME:"] = "$aufname";
                $labelSELECT = "AUFNAHMEFLÄCHE:";
                $linkobjname = $aufname;
                $linkobjid = $in_auftid;
                $linkobatid = 5;
                $linkfrmid = 1;
                $gemStatus = 2;
            }
        } else {
            $parameter = "auftid";
            $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
                ."FROM afn_objekt o, afn_objektart a, afn_isin i, afn_isinart ia "
                ."WHERE o.obaid = a.TID "
                ."AND a.OBANAME = 'AUFNAHME' "
                ."AND ia.ISANAME = 'Großraum' "
                ."AND ia.TID = i.ISAID "

```

```

        ."AND i.ISINOBJID = $in_regtid "
        ."AND o.TID = i.OBJID "
        ."ORDER BY o.OBJNAME";
    $SQLqueryDatum = "SELECT DISTINCT OBJNAME AS NAME "
        ."FROM afn_objekt "
        ."WHERE TID = $in_regtid ";
    $regname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrSELHidden["regtid"] = "$in_regtid";
    $arrSELHidden["regname"] = "$in_regname";
    $arrSELHidden["regname"] = "$regname";
    $arrHISTDaten["PROJEKT:"] = "$in_prjname";
    $arrHISTDaten["NATURRAUM:"] = "$regname";
    $labelSELECT = "AUFNAHME:";
    }
} else {
    $parameter = "regtid";
    $SQLquery = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM afn_objekt o, afn_objektart a "
        ."WHERE o.obaid = a.TID "
        ."AND a.OBAKUERZEL = 'GRO' "
        ."ORDER BY o.OBJNAME";
    $SQLqueryDatum = "SELECT DISTINCT PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE TID = $in_prjtid ";
    $prjname = $oDBM->GETSelectSQLDatum($SQLqueryDatum);
    $arrSELHidden["prjtid"] = "$in_prjtid";
    $arrSELHidden["prjname"] = "$prjname";
    $arrHISTDaten["PROJEKT:"] = "$prjname";
    $labelSELECT = "GROSSRAUM:";
    }
} else {
    $parameter = "prjtid";
    $SQLquery = "SELECT DISTINCT TID, PRJNAME AS NAME "
        ."FROM afn_projekt "
        ."WHERE PRJNAME <> 'ADMIN' "
        ."ORDER BY PRJNAME";
    $labelSELECT = "PROJEKT:";
}

// GET DB-Daten
$oDBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);

// Layoutbausteine zusammenstellen
$oLTM->GETHeader();
$oVWM->GETLabel($HPueberschrift,"H1");
$oVWM->GETIDTable(&$arrHISTDaten);
$oVWM->GETLabel($labelSELECT,"H3");
If ($gemStatus == 1)
{
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obaid='.$linkobaid.'&objname='.$linkobjname.'&frmid='.$linkfrmid;
    $oVWM->GETLink($lnkfile,$lnklbSelect);
} elseif ($gemStatus == 2){
    $oVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
    $lnkfile = $lnkfile.'?objtid='.$linkobjid.'&obaid='.$linkobaid.'&objname='.$linkobjname.'&frmid='.$linkfrmid;
    $oVWM->GETLink($lnkfile,$lnklbSelAufnahme);
} else {
    $oVWM->GETSelFormHidden($in_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
}
}
$oLTM->GETFooter();
?>

```

9.3.2.2.7 Controller – „<Objekt>“ (OOPSelObjekt.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('../././config/flora/sqlpublic.php.inc');

```

```

// INIT Variables
if($_REQUEST['prjtid'] != ""){$sin_prjtid=$_REQUEST['prjtid'];} else {$sin_prjtid = 0;};
if($_REQUEST['prjname'] != ""){$sin_prjname=$_REQUEST['prjname'];} else {$sin_prjname = 0;};
if($_REQUEST['obtid'] != ""){$sin_obtid=$_REQUEST['obtid'];} else {$sin_obtid = 0;};
if($_REQUEST['obaname'] != ""){$sin_obaname=$_REQUEST['obaname'];} else {$sin_obaname = 0;};

$HPueberschrift = "Auswahl $sin_obaname";
$lbSelect = "Auswahl";

$$arrSQLObjekt = NULL;
$arrSQL = NULL;

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$oVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$oLTM = new LAYOUTManager;

// HTML Header
$oLTM->GETHeader();

// Projektauswahl
if ($sin_prjtid > 0)
{
    // INIT SelectObjekt
    $SQLqueryObjekt = "SELECT DISTINCT o.TID AS TID, o.OBJNAME AS NAME "
        ."FROM   afn_objekt o "
        ."WHERE  o.OBAID = '$sin_obtid' "
        ."AND    o.PRJID = '$sin_prjtid' "
        ."ORDER BY o.OBJNAME";

    // GET DB-Daten
    $oDBM->GETSelectSQL($SQLqueryObjekt,"TID,NAME",$arrSQLObjekt);

    // Layoutbausteine zusammenstellen
    $oVWM->GETLabel($HPueberschrift,"H1");

    if (is_array($arrSQLObjekt))
    {
        foreach ($arrSQLObjekt as $Daten)
        {
            // bei ca.3000 Datensätzen gibt es ein Problem mit Objekten (=> Speicher?)
            print '<a
href="OOPFormStammdata.php?objtid='.$Daten['TID'].'&objname='.$Daten['NAME'].'&obtid='.$sin_obtid.'&obaname='.$sin_obaname.'">
'.$Daten['NAME'].'</a><br />';
        }
    }
    else {
        $parameter = "prjtid";
        $SQLquery = "SELECT  DISTINCT p.TID, p.PRJNAME "
            ."FROM    afn_objekt o, "
            ."        afn_projekt p "
            ."WHERE   p.TID = o.PRJID "
            ."AND     o.OBAID = $sin_obtid ";
        $labelSELECT = "PROJEKT:";

        // GET DB-Daten
        $oDBM->GETSelectSQL($SQLquery,"TID,NAME",$arrSQL);

        // HiddenArray füllen
        $arrSELHidden["obtid"] = "$sin_obtid";
        $arrSELHidden["obaname"] = "$sin_obaname";

        // Layoutbausteine zusammenstellen
        $oVWM->GETLabel($HPueberschrift,"H1");
        $oVWM->GET1DTable(&$arrHISTDaten);
        $oVWM->GETLabel($labelSELECT,"H3");
        $oVWM->GETSelFormHidden($sin_tid,$parameter,$file,$lbSelect,$arrSQL,$arrSELHidden);
    }
    $oLTM->GETFooter();
}
?>

```

9.3.2.2.8 Controller – „Aufnahme Stammdaten“ (OOPFormStammdata.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/VIEWManager.php');
include_once('../lib/LAYOUTManager.php');
include_once('.././../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['obtid'] != ""){$_sin_obtid=$_REQUEST['obtid'];} else {$_sin_obtid = 0;};
if($_REQUEST['objtid'] != ""){$_sin_objtid=$_REQUEST['objtid'];} else {$_sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$_sin_objname=$_REQUEST['objname'];} else {$_sin_objname = 0;};
if($_REQUEST['frmid'] != ""){$_sin_frmid=$_REQUEST['frmid'];} else {$_sin_frmid = 0;};

$dlfile = "OOPExcelStammdata.php";
$pdffile = "OOPpdfStammdata.php";
$textfile = "OOPASCIStammdata.php";

$Afile = "OOPFormStammdata.php";
$vegefile = "OOPFormAufnahmedata.php";
$veglabel_xls = "Vegetationdaten";

$SarrFormular = NULL;
$SarrStammdaten = NULL;
$SarrObjektdaten = NULL;
$SarrPOLdaten = NULL;
$SarrisinObjekt = NULL;
$SarrISINobjart = NULL;
$stamm_id = 0;
$stamm_value = 'k.a.';

// INIT DBManager Objekt
$soDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT VIEWManager Objekt
$soVWM = new VIEWManager;
// INIT LAYOUTManager Objekt
$soLTM = new LAYOUTManager;

$SQLqueryfrmid = "SELECT DISTINCT TID FROM afn_h_formular WHERE OBAID = ".$_sin_obtid;
$_sin_frmid = $soDBM->GETSelectSQLDatum($SQLqueryfrmid);

// GET DB-Daten
$soDBM->GETObjektStammdata($_sin_obtid,$SarrFormular);

// HTML-Seite Anfang
$soLTM->GETFORMHeader();

// Formular zusammenstellen
$soVWM->GETFormularHeader($_sin_objname);
if (is_array($SarrFormular))
{
    foreach ($SarrFormular as $Daten)
    {
        if ($Daten['STYLE'] == 2)
        {
            $soVWM->GETFormRowStyle2($Daten['FORMLABEL']);
        } elseif ($Daten['STYLE'] == 3) {
            $soVWM->GETFormRowStyle3($Daten['FORMLABEL']);
        } elseif ($Daten['STYLE'] == 4) {
            $soDBM->GETSelectParamdata($_sin_objtid,$Daten['PARID'],$SarrStammdaten);
            if (is_array($SarrStammdaten))
            {
                foreach ($SarrStammdaten as $Stamm)
                {
                    $stamm_id = $Stamm['TID'];
                    $stamm_value = $Stamm['STDWERT'];
                }
            }
            $soVWM->GETFormRowStyle4($Daten['LABEL'],$stamm_value,$Daten['PARID']);
            $stamm_id = 0;
            $stamm_value = 'k.a.';
            $SarrStammdaten = NULL;
        }
    }
}

```

```

} elseif ($Daten['STYLE'] == 5) {
    $PATHquery = "SELECT isa.ISAKUERZEL "
                ."FROM   afn_parameter p, afn_isinart isa "
                ."WHERE  p.TID = ".$Daten['PARID']. " "
                ."AND    isa.TID = p.ISAID ";
    $path = $oDBM->GETSelectSQLDatum($PATHquery);

    if ($path == 'POL')
    {
        $oDBM->GETIsinartPOL($in_objtid,$Daten['PARID'],$arrPOLdaten);
        $gemeinde = 'k.a.';
        $bezirk = 'k.a.';
        $bundesland = 'k.a.';
        if (is_array($arrPOLdaten))
        {
            foreach ($arrPOLdaten as $Objekt)
            {
                $gemeinde = $Objekt['Gemeinde'];
                $bezirk = $Objekt['Bezirk'];
                $bundesland = $Objekt['Bundesland'];

                $oVWM->GETFormRowStyle4($Daten['LABEL'],' - Gemeinde',$gemeinde,$Daten['PARID']);
                $oVWM->GETFormRowStyle4($Daten['LABEL'],' - Bezirk',$bezirk,$Daten['PARID']);
                $oVWM->GETFormRowStyle4($Daten['LABEL'],' - Bundesland',$bundesland,$Daten['PARID']);
            }
            $arrPOLdaten = NULL;
        }
    }
    else {
        $oDBM->GETSelectObjektdaten($in_objtid,$Daten['PARID'],$arrObjektdaten);
        if (is_array($arrObjektdaten))
        {
            foreach ($arrObjektdaten as $Objekt)
            {
                $stamm_id = $Objekt['TID'];
                $stamm_value = $Objekt['NAME'];

                $oVWM->GETFormRowStyle4($Daten['LABEL'],$stamm_value,$Daten['PARID']);
                $stamm_id = 0;
                $stamm_value = 'k.a.';
            }
            $arrObjektdaten = NULL;
        }
    }
} elseif ($Daten['STYLE'] == 6) {
    $oDBM->GETIsinOBJART($in_objtid,$Daten['PARID'],$arrISINobjart);
    if (is_array($arrISINobjart))
    {
        if (count($arrISINobjart) == 1)
        {
            foreach ($arrISINobjart as $isinObjekt)
            {
                if($isinObjekt['ISOBAID'] == $isinObjekt['OBAID'])
                {
                    $oDBM->GETSelectObjektdaten($in_objtid,$Daten['PARID'],$arrisinObjekt);
                } elseif($isinObjekt['ISINOBAID'] == $isinObjekt['OBAID']) {
                    $oDBM->GETSelectISINObjektdaten($in_objtid,$Daten['PARID'],$arrisinObjekt);
                }
            }
        }
    }
}
if ($in_frmid == 1)
{
    $LinkfrmId = 2;
    $obatid = 4;
    $labelAAF = 'Aufnahmeflächen';
} else {
    $LinkfrmId = 1;
    $obatid = 5;
    $labelAAF = 'Aufnahme';
}
if (is_array($arrisinObjekt))
{
    $oVWM->GETFormRowStyle3($labelAAF);
    foreach ($arrisinObjekt as $isObjekt)
    {

```

9.3.2.2.9 Controller – „Vegetationsdat. der Aufnahmevl.“ (OOPFormAufnahmedata.php)

Stephan Hordynski (Matrikelnr.: a9227169)

```

{
    foreach ($SarrAufnahmedaten as $Daten)
    {
        $oVWM->GETFormRowArt($Daten['ART'],$Daten['VEGWERT']);
    }
}
$oVWM->GETFormularFooter();

// Download Excel, pdf, ASCII, ORIFile
$dfile = $dfile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;
$pdffile = $pdffile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;
$textfile = $textfile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;
$orifile = $orifile.'?objtid='.$_sin_objtid.'&objname='.$_sin_objname;
$oVWM->GETFormDLPrintLinkAF($dfile,$pdffile,$textfile,$orifile);

// HTML-Seite Ende
$oLTM->GETFooter();

?>

```

9.3.2.2.10 Controller – „Download Stammdaten ASCII“ (OOPASCIISTammdata.php)

```

<?php

// INCLUDE
include_once('..lib/DBManager.php');
include_once('..../config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['objtid'] != ""){$_sin_objtid=$_REQUEST['objtid'];} else {$_sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$_sin_objname=$_REQUEST['objname'];} else {$_sin_objname = 0;};
if($_REQUEST['frmid'] != ""){$_sin_frmid=$_REQUEST['frmid'];} else {$_sin_frmid = 0;};

$SarrFormular = NULL;
$SarrStammdaten = NULL;
$SarrObjektdaten = NULL;
$ASCIIDatei = NULL;
$stamm_id = 0;
$stamm_value = 'k.a.';
$scouter = 0;

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

// GET DB-Daten
$oDBM->GETSelectStammdata($_sin_frmid,$SarrFormular);

// ASCII Header
header('Content-Type: text/txt');
header('Content-Disposition: attachment;filename=Stammdaten.txt');

// ASCII Formular zusammenstellen
print $_sin_objname."\015\012";

// EXCEL Formularinhalte
if (is_array($SarrFormular))
{
    foreach ($SarrFormular as $Daten)
    {
        if ($Daten['STYLE'] == 2)
        {
            print $Daten['FORMLABEL']."\015\012";
        } elseif ($Daten['STYLE'] == 3) {
            print $Daten['FORMLABEL']."\015\012";
        } elseif ($Daten['STYLE'] == 4) {
            $oDBM->GETSelectParamdata($_sin_objtid,$Daten['PARID'],$SarrStammdaten);
            if (is_array($SarrStammdaten))
            {
                foreach ($SarrStammdaten as $Stamm)
                {
                    $stamm_id = $Stamm['TID'];
                    $stamm_value = $Stamm['STDWERT'];
                }
            }
        }
    }
}

```

```

print $Daten['LABEL'].':'. $stamm_value."015\012";

$stamm_id = 0;
$stamm_value = 'k.a.';
$SarrStammdaten = NULL;
} elseif ($Daten['STYLE'] == 5) {
    $PATHquery = "SELECT isa.ISAKUERZEL "
        ."FROM   afn_parameter p, afn_isinart isa "
        ."WHERE  p.TID = ".$Daten['PARID']. " "
        ."AND   isa.TID = p.ISAID ";
    $path = $oDBM->GETSelectSQLDatum($PATHquery);

    if ($path == 'POL')
    {
        $oDBM->GETIsinartPOL($in_objtid,$Daten['PARID'],$SarrPOLdaten);
        $gemeinde = 'k.a.';
        $bezirk = 'k.a.';
        $bundesland = 'k.a.';
        if (is_array($SarrPOLdaten))
        {
            foreach ($SarrPOLdaten as $Objekt)
            {
                $gemeinde = $Objekt['Gemeinde'];
                $bezirk = $Objekt['Bezirk'];
                $bundesland = $Objekt['Bundesland'];

                print $Daten['LABEL']. ' - Gemeinde:'. $gemeinde."015\012";
                print $Daten['LABEL']. ' - Bezirk:'. $bezirk."015\012";
                print $Daten['LABEL']. ' - Bundesland:'. $bundesland."015\012";
            }
            $SarrPOLdaten = NULL;
        }
    } else {
        $oDBM->GETSelectObjektdaten($in_objtid,$Daten['PARID'],$SarrObjektdaten);
        if (is_array($SarrObjektdaten))
        {
            foreach ($SarrObjektdaten as $Objekt)
            {
                $stamm_id = $Objekt['TID'];
                $stamm_value = $Objekt['NAME'];

                print $Daten['LABEL'].':'. $stamm_value."015\012";

                $stamm_id = 0;
                $stamm_value = 'k.a.';
            }
        }
        $SarrObjektdaten = NULL;
    }
} elseif ($Daten['STYLE'] == 6) {
    $oDBM->GETIsinOBJART($in_objtid,$Daten['PARID'],$SarrISINobjart);
    if (is_array($SarrISINobjart))
    {
        if (count($SarrISINobjart) == 1)
        {
            foreach ($SarrISINobjart as $isinObjekt)
            {
                if($isinObjekt['ISOBAID'] == $isinObjekt['OBAID'])
                {
                    $oDBM->GETSelectObjektdaten($in_objtid,$Daten['PARID'],&$SarrisinObjekt);
                } elseif($isinObjekt['ISINOBaid'] == $isinObjekt['OBAID']) {
                    $oDBM->GETSelectISINObjektdaten($in_objtid,$Daten['PARID'],&$SarrisinObjekt);
                }
            }
        }
    }
}

if ($in_frmid == 1)
{
    $LinkfrmId = 2;
    $labelAAF = 'Aufnahmeflächen';
} else {
    $LinkfrmId = 1;
    $labelAAF = 'Aufnahme';
}

```



```

if (is_array($sarrisinObjekt))
{
    print $labelAAF."\015\012";
    foreach ($sarrisinObjekt as $isObjekt)
    {
        $stamm_id = $isObjekt['TID'];
        $stamm_value = $isObjekt['NAME'];

        print $Daten['LABEL'].':'. $stamm_value. "\015\012";
    }
    $sarrisinObjekt = NULL;
}
}
}
}
?>

```

9.3.2.2.11 Controller – „Download Stammdaten Excel“ (OOPEXcelStammdata.php)

```
<?php
// INCLUDE
include_once "Spreadsheet/Excel/Writer.php";
include_once("../lib/DBManager.php");
include_once("../../config_iflora/sqlpublic.php.inc");

// INIT Variables
if($_REQUEST['objtid'] != ""){$_sin_objtid=$_REQUEST['objtid'];} else {$_sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$_sin_objname=$_REQUEST['objname'];} else {$_sin_objname = 0;};
if($_REQUEST['frmid'] != ""){$_sin_frmid=$_REQUEST['frmid'];} else {$_sin_frmid = 0;};

$arrFormular = NULL;
$arrStammdaten = NULL;
$arrObjektdaten = NULL;
$stamm_id = 0;
$stamm_value = 'k.a.';
$counter = 0;

// INIT DBManager Objekt
$oDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

// GET DB-Daten
$oDBM->GETSelectStammdata($_sin_frmid,$arrFormular);

// EXCEL Formular zusammenstellen
$xls =& new Spreadsheet_Excel_Writer();
$xls->send("Aufnahme.xls");
$sheet =& $xls->addWorksheet('A');

// EXCEL Überschrift
$format =& $xls->addFormat();
$format->setBold();
$format->setColor("blue");
$sheet->write($counter,0,$_sin_objname,$format);
$counter++;

// EXCEL Formularinhalte
if (is_array($arrFormular))
{
    foreach ($arrFormular as $Daten)
    {
        if ($Daten['STYLE'] == 2)
        {
            $format =& $xls->addFormat();
            $format->setBold();
            $format->setColor("black");
            $sheet->write($counter,0,$Daten['FORMLABEL'],$format);
            $counter++;
        }
        elseif ($Daten['STYLE'] == 3) {
            $format =& $xls->addFormat();
            $format->setColor("black");
            $sheet->write($counter,0,$Daten['FORMLABEL'],$format);
            $counter++;
        }
    }
}
```

```

} elseif ($Daten['STYLE'] == 4) {
    $oDBM->GETSelectParamdata($in_objtid,$Daten['PARID'],$arrStammdaten);
    if (is_array($arrStammdaten))
    {
        foreach ($arrStammdaten as $Stamm)
        {
            $stamm_id = $Stamm['TID'];
            $stamm_value = $Stamm['STDWERT'];
        }
    }
    $format =& $xls->addFormat();
    $format->setColor("black");
    $sheet->write($counter,0,$Daten['LABEL'],$format);
    $sheet->write($counter,1,$stamm_value,$format);
    $counter++;

    $stamm_id = 0;
    $stamm_value = 'k.a.';
    $arrStammdaten = NULL;
} elseif ($Daten['STYLE'] == 5) {
    $PATHquery = "SELECT isa.ISAKUERZEL "
                ."FROM   afn_parameter p, afn_isinart isa "
                ."WHERE  p.TID = ".$Daten['PARID']. " "
                ."AND   isa.TID = p.ISAID ";
    $path = $oDBM->GETSelectSQLDatum($PATHquery);

    if ($path == 'POL')
    {
        $oDBM->GETisinartPOL($in_objtid,$Daten['PARID'],$arrPOLdaten);
        $gemeinde = 'k.a.';
        $bezirk = 'k.a.';
        $bundesland = 'k.a.';
        if (is_array($arrPOLdaten))
        {
            foreach ($arrPOLdaten as $Objekt)
            {
                $gemeinde = $Objekt['Gemeinde'];
                $bezirk = $Objekt['Bezirk'];
                $bundesland = $Objekt['Bundesland'];

                $format =& $xls->addFormat();
                $format->setColor("black");
                $sheet->write($counter,0,$Daten['LABEL'].' - Gemeinde',$format);
                $sheet->write($counter,1,$gemeinde,$format);
                $counter++;

                $format =& $xls->addFormat();
                $format->setColor("black");
                $sheet->write($counter,0,$Daten['LABEL'].' - Bezirk',$format);
                $sheet->write($counter,1,$bezirk,$format);
                $counter++;

                $format =& $xls->addFormat();
                $format->setColor("black");
                $sheet->write($counter,0,$Daten['LABEL'].' - Bundesland',$format);
                $sheet->write($counter,1,$bundesland,$format);
                $counter++;
            }
            $arrPOLdaten = NULL;
        }
    }
} else {
    $oDBM->GETSelectObjektdaten($in_objtid,$Daten['PARID'],$arrObjektdaten);
    if (is_array($arrObjektdaten))
    {
        foreach ($arrObjektdaten as $Objekt)
        {
            $stamm_id = $Objekt['TID'];
            $stamm_value = $Objekt['NAME'];

            $format =& $xls->addFormat();
            $format->setColor("black");
            $sheet->write($counter,0,$Daten['LABEL'],$format);
            $sheet->write($counter,1,$stamm_value,$format);
            $counter++;
        }
    }
}

```

```

        $stamm_id = 0;
        $stamm_value = 'k.a.';
    }
}
$arrObjektdaten = NULL;
}
elseif ($Daten['STYLE'] == 6) {
    $oDBM->GETisINOBJART($in_objtid,$Daten['PARID'],$arrISINobjart);
    if (is_array($arrISINobjart))
    {
        if (count($arrISINobjart) == 1)
        {
            foreach ($arrISINobjart as $isinObjekt)
            {
                if($isinObjekt['ISOBaid'] == $isinObjekt['OBaid'])
                {
                    $oDBM->GETSelectObjektdata($in_objtid,$Daten['PARID'],$arrisinObjekt);
                } elseif($isinObjekt['ISINOBAID'] == $isinObjekt['OBaid']) {
                    $oDBM->GETSelectISINObjektdata($in_objtid,$Daten['PARID'],$arrisinObjekt);
                }
            }
        }
    }
}

if ($in_frmid == 1)
{
    $LinkfrmId = 2;
    $labelAAF = 'Aufnahmeflächen';
} else {
    $LinkfrmId = 1;
    $labelAAF = 'Aufnahme';
}
if (is_array($arrisinObjekt))
{
    $format =& $xls->addFormat();
    $format->setColor("black");
    $sheet->write($counter,0,$labelAAF,$format);
    $counter++;

    foreach ($arrisinObjekt as $isObjekt)
    {
        $stamm_id = $isObjekt['TID'];
        $stamm_value = $isObjekt['NAME'];

        $Afile = $Afile.'?objtid='.$stamm_id.'&objname='.$stamm_value.'&frmId='.$LinkfrmId;

        $format =& $xls->addFormat();
        $format->setColor("black");
        $sheet->write($counter,0,$Daten['LABEL'],$format);
        $sheet->write($counter,1,$stamm_value,$format);
        $counter++;
    }
    $arrisinObjekt = NULL;
}
}
}
}
// EXCEL Formular Ende
$xls->close();

?>

```

9.3.2.2.12 Controller – „Download Stammdaten PDF“ (OOPpdfStammdata.php)

```

<?php

// INCLUDE
define('FPDF_FONTPATH','../fpdf/font/');
require('../fpdf/fpdf.php');
include_once('../lib/DBManager.php');
include_once('../lib/config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['objtid'] != ""){$in_objtid=$_REQUEST['objtid'];} else {$in_objtid = 0;};
if($_REQUEST['objname'] != ""){$in_objname=$_REQUEST['objname'];} else {$in_objname = 0;};

```

```
if($_REQUEST['frmid'] != ""){$sin_frmid=$_REQUEST['frmid'];} else {$sin_frmid = 0;};

$arrFormular = NULL;
$arrStammdaten = NULL;
$arrObjektdaten = NULL;
$stamm_id = 0;
$stamm_value = 'k.a.';
$counter = 0;

// INIT DBManager Objekt
$soDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT PDFManager Objekt
$soPDF = new FPDF;

// GET DB-Daten
$soDBM->GETSelectStammdata($sin_frmid,$arrFormular);

// PDF Formular zusammenstellen
$soPDF->SetFont('Arial','',12);
$soPDF->AddPage();

// PDF Überschrift
$soPDF->SetFillColor(70,100,140);
$soPDF->SetTextColor(255,255,255);
$soPDF->SetDrawColor(128,0,0);
$soPDF->SetLineWidth(.3);
$soPDF->SetFont('','B');
$soPDF->Cell(190,7,$sin_objname,1,0,'L',1);
$soPDF->Ln();

// PDF Formularinhalte
if (is_array($arrFormular))
{
    foreach ($arrFormular as $Daten)
    {
        if ($Daten['STYLE'] == 2)
        {
            $soPDF->SetFillColor(202,213,227);
            $soPDF->SetTextColor(0,0,0);
            $soPDF->SetDrawColor(128,0,0);
            $soPDF->SetLineWidth(.3);
            $soPDF->SetFont('','B');
            $soPDF->Cell(190,7,$Daten['FORMLABEL'],1,0,'L',1);
            $soPDF->Ln();
        } elseif ($Daten['STYLE'] == 3) {
            $soPDF->SetFillColor(204,204,204);
            $soPDF->SetTextColor(0,0,0);
            $soPDF->SetDrawColor(128,0,0);
            $soPDF->SetLineWidth(.3);
            $soPDF->SetFont('','B');
            $soPDF->Cell(190,7,$Daten['FORMLABEL'],1,0,'L',1);
            $soPDF->Ln();
        } elseif ($Daten['STYLE'] == 4) {
            $soDBM->GETSelectParamdata($sin_objtid,$Daten['PARID'],$arrStammdaten);
            if (is_array($arrStammdaten))
            {
                foreach ($arrStammdaten as $Stamm)
                {
                    $stamm_id = $Stamm['TID'];
                    $stamm_value = $Stamm['STDWERT'];
                }
            }
            $soPDF->SetFillColor(238,238,238);
            $soPDF->SetTextColor(0,0,0);
            $soPDF->SetDrawColor(128,0,0);
            $soPDF->SetLineWidth(.3);
            $soPDF->Cell(90,7,$Daten['LABEL'],1,0,'L',1);
            $soPDF->SetFillColor(255,255,255);
            $soPDF->SetTextColor(0,0,0);
            $soPDF->SetDrawColor(128,0,0);
            $soPDF->SetLineWidth(.3);
            $soPDF->Cell(100,7,$stamm_value,1,0,'L',1);
            $soPDF->Ln();
        }
    }
}
```

```

$stamm_id = 0;
$stamm_value = 'k.a.';
$arrStammdaten = NULL;
} elseif ($Daten['STYLE'] == 5) {
$PATHquery = "SELECT isa.ISAKUERZEL "
              ".FROM   afn_parameter p, afn_isinart isa "
              ".WHERE  p.TID = ".$Daten['PARID']. " "
              ".AND    isa.TID = p.ISAID ";
$spath = $oDBM->GETSelectSQLDatum($PATHquery);

if ($spath == 'POL')
{
    $oDBM->GETisinartPOL($sin_objtid,$Daten['PARID'],$arrPOLdaten);
    $gemeinde = 'k.a.';
    $bezirk = 'k.a.';
    $bundesland = 'k.a.';
    if (is_array($arrPOLdaten))
    {
        foreach ($arrPOLdaten as $Objekt)
        {
            $gemeinde = $Objekt['Gemeinde'];
            $bezirk = $Objekt['Bezirk'];
            $bundesland = $Objekt['Bundesland'];

            $oPDF->SetFillColor(238,238,238);
            $oPDF->SetTextColor(0,0,0);
            $oPDF->SetDrawColor(128,0,0);
            $oPDF->SetLineWidth(.3);
            $oPDF->Cell(90,7,$Daten['LABEL'],' - Gemeinde',1,0,'L',1);
            $oPDF->SetFillColor(255,255,255);
            $oPDF->SetTextColor(0,0,0);
            $oPDF->SetDrawColor(128,0,0);
            $oPDF->SetLineWidth(.3);
            $oPDF->Cell(100,7,$gemeinde,1,0,'L',1);
            $oPDF->Ln();

            $oPDF->SetFillColor(238,238,238);
            $oPDF->SetTextColor(0,0,0);
            $oPDF->SetDrawColor(128,0,0);
            $oPDF->SetLineWidth(.3);
            $oPDF->Cell(90,7,$Daten['LABEL'],' - Bezirk',1,0,'L',1);
            $oPDF->SetFillColor(255,255,255);
            $oPDF->SetTextColor(0,0,0);
            $oPDF->SetDrawColor(128,0,0);
            $oPDF->SetLineWidth(.3);
            $oPDF->Cell(100,7,$bezirk,1,0,'L',1);
            $oPDF->Ln();

            $oPDF->SetFillColor(238,238,238);
            $oPDF->SetTextColor(0,0,0);
            $oPDF->SetDrawColor(128,0,0);
            $oPDF->SetLineWidth(.3);
            $oPDF->Cell(90,7,$Daten['LABEL'],' - Bundesland',1,0,'L',1);
            $oPDF->SetFillColor(255,255,255);
            $oPDF->SetTextColor(0,0,0);
            $oPDF->SetDrawColor(128,0,0);
            $oPDF->SetLineWidth(.3);
            $oPDF->Cell(100,7,$bundesland,1,0,'L',1);
            $oPDF->Ln();
        }
        $arrPOLdaten = NULL;
    }
} else {
$oDBM->GETSelectObjektdata($sin_objtid,$Daten['PARID'],$arrObjektdaten);
if (is_array($arrObjektdaten))
{
    foreach ($arrObjektdaten as $Objekt)
    {
        $stamm_id = $Objekt['TID'];
        $stamm_value = $Objekt['NAME'];

        $oPDF->SetFillColor(238,238,238);
        $oPDF->SetTextColor(0,0,0);
        $oPDF->SetDrawColor(128,0,0);
        $oPDF->SetLineWidth(.3);
    }
}
}

```

```

$soPDF->Cell(90,7,$Daten['LABEL'],1,0,'L',1);
$soPDF->SetFillColor(255,255,255);
$soPDF->SetTextColor(0,0,0);
$soPDF->SetDrawColor(128,0,0);
$soPDF->SetLineWidth(.3);
$soPDF->Cell(100,7,$stamm_value,1,0,'L',1);
$soPDF->Ln();

$stamm_id = 0;
$stamm_value = 'k.a.';
}
}
$arrObjektdaten = NULL;
}
} elseif ($Daten['STYLE'] == 6) {
$soDBM->GETisinOBJART($sin_objtid,$Daten['PARID'],$arrISINobjart);
if (is_array($arrISINobjart))
{
if (count($arrISINobjart) == 1)
{
foreach ($arrISINobjart as $isinObjekt)
{
if($isinObjekt['ISOBaid'] == $isinObjekt['OBaid'])
{
$soDBM->GETSelectObjektdata($sin_objtid,$Daten['PARID'],&$arrisinObjekt);
} elseif($isinObjekt['ISINOBAID'] == $isinObjekt['OBaid']) {
$soDBM->GETSelectISINObjektdata($sin_objtid,$Daten['PARID'],&$arrisinObjekt);
}
}
}
}
}
if ($sin_frmid == 1)
{
$LinkfrmId = 2;
$labelAAF = 'Aufnahmeflächen';
} else {
$LinkfrmId = 1;
$labelAAF = 'Aufnahme';
}
if (is_array($arrisinObjekt))
{
$soPDF->SetFillColor(204,204,204);
$soPDF->SetTextColor(0,0,0);
$soPDF->SetDrawColor(128,0,0);
$soPDF->SetLineWidth(.3);
$soPDF->SetFont(",B");
$soPDF->Cell(190,7,$labelAAF,1,0,'L',1);
$soPDF->Ln();
foreach ($arrisinObjekt as $isObjekt)
{
$stamm_id = $isObjekt['TID'];
$stamm_value = $isObjekt['NAME'];

$soPDF->SetFillColor(238,238,238);
$soPDF->SetTextColor(0,0,0);
$soPDF->SetDrawColor(128,0,0);
$soPDF->SetLineWidth(.3);
$soPDF->Cell(90,7,$Daten['LABEL'],1,0,'L',1);
$soPDF->SetFillColor(255,255,255);
$soPDF->SetTextColor(0,0,0);
$soPDF->SetDrawColor(128,0,0);
$soPDF->SetLineWidth(.3);
$soPDF->Cell(100,7,$stamm_value,1,0,'L',1);
$soPDF->Ln();
}
$arrisinObjekt = NULL;
}
}
}
}
// PDF Formular Ende
$soPDF->Output();

?>

```

9.3.2.2.13 Controller – „Download Aufn. ASCII“ (OOPASCIIAufnahmedata.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['objtid'] != ""){$sin_objtid=$_REQUEST['objtid'];} else {$sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$sin_objname=$_REQUEST['objname'];} else {$sin_objname = 0;};

$SarrAufnahmedaten = NULL;
$counter = 0;

// INIT DBManager Objekt
$SoDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

// GET DB-Daten
$SoDBM->GETSelectAufnahmedata($sin_objtid,$SarrAufnahmedaten);

// ASCII Header
header('Content-Type: text/txt');
header('Content-Disposition: attachment;filename=Stammdaten.txt');

// ASCII Überschriften
print $sin_objname."\015\012";
print 'Aufnahmefläche'. "\015\012";
print 'Vegetationsdaten'. "\015\012";

if (is_array($SarrAufnahmedaten))
{
    foreach ($SarrAufnahmedaten as $Daten)
    {
        print $Daten['ART'].':'. $Daten['VEGWERT']. "\015\012";
    }
}

?>

```

9.3.2.2.14 Controller – „Download Aufn. ORI-Datei“ (OOPASCIIAufnahmedataORI.php)

```

<?php

// INCLUDE
include_once('../lib/DBManager.php');
include_once('../lib/config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['objtid'] != ""){$sin_objtid=$_REQUEST['objtid'];} else {$sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$sin_objname=$_REQUEST['objname'];} else {$sin_objname = 0;};

$SarrAufnahmedaten = NULL;
$counter = 0;

// INIT DBManager Objekt
$SoDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

$queryOBNR = "SELECT objnummer "
            ."FROM afn_objekt "
            ."WHERE TID = $sin_objtid ";
$objnummer = $SoDBM->GETSelectSQLDatum($queryOBNR);

// GET DB-Daten
$SoDBM->GETSelectORIData($sin_objtid,$SarrAufnahmedaten);

// ASCII Header
header('Content-Type: text/txt');
header('Content-Disposition: attachment;filename=Stammdaten.txt');

// ASCII Überschriften
print $sin_objname."\015\012";
print "8\015\012";
print "1\015\012";
print "1\015\012";

```

```

print "I\015\012";
print $objnummer."I\015\012";

if (is_array($arrAufnahmedaten))
{
    foreach ($arrAufnahmedaten as $Daten)
    {
        print "0".$Daten['ART'].$Daten['VEGWERT']."I\015\012";
    }
}
print "A\015\012";
print "0000\015\012";
print "A\015\012";

?>

```

9.3.2.2.15 Controller – „Download Aufn. Excel“ (OOPEXcelAufnahmedata.php)

```

<?php

// INCLUDE
include_once "Spreadsheet/Excel/Writer.php";
include_once("../lib/DBManager.php");
include_once("../config_iflora/sqlpublic.php.inc");

// INIT Variables
if($_REQUEST['objtid'] != ""){$sin_objtid=$_REQUEST['objtid'];} else {$sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$sin_objname=$_REQUEST['objname'];} else {$sin_objname = 0;};

$arrAufnahmedaten = NULL;
$counter = 0;

// INIT DBManager Objekt
$soDBM = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);

// GET DB-Daten
$soDBM->GETSelectAufnahmedata($sin_objtid,$arrAufnahmedaten);

// EXCEL Formular zusammenstellen
$xls =& new Spreadsheet_Excel_Writer();
$xls->send("Aufnahmedaten.xls");
$sheet =& $xls->addWorksheet('A');

// EXCEL Überschrift
$format =& $xls->addFormat();
$format->setBold();
$format->setColor("blue");
$sheet->write($counter,0,$sin_objname,$format);
$counter++;
$format =& $xls->addFormat();
$format->setBold();
$format->setColor("black");
$sheet->write($counter,0,'Aufnahmefläche',$format);
$counter++;
$format =& $xls->addFormat();
$format->setColor("black");
$sheet->write($counter,0,'Vegetationsdaten',$format);
$counter++;

if (is_array($arrAufnahmedaten))
{
    foreach ($arrAufnahmedaten as $Daten)
    {
        $format =& $xls->addFormat();
        $format->setColor("black");
        $sheet->write($counter,0,$Daten['ART'],$format);
        $sheet->write($counter,1,$Daten['VEGWERT'],$format);
        $counter++;
    }
}
// EXCEL Formular Ende
$xls->close();

?>

```


9.3.2.2.16 Controller – „Download Aufn. PFD“ (OOPpdfAufnahmedata.php)

```

<?php

// INCLUDE
define('FPDF_FONTPATH','../fpdf/font/');
require('../fpdf/fpdf.php');
include_once('../lib/DBManager.php');
include_once('../lib/config_iflora/sqlpublic.php.inc');

// INIT Variables
if($_REQUEST['objtid'] != ""){$sin_objtid=$_REQUEST['objtid'];} else {$sin_objtid = 0;};
if($_REQUEST['objname'] != ""){$sin_objname=$_REQUEST['objname'];} else {$sin_objname = 0;};

$arrAufnahmedaten = NULL;
$counter = 0;

// INIT DBManager Objekt
$dbm = new DBManager($database,$sql_server,$p_user,$p_passwd,$sql_db,$schema);
// INIT PDFManager Objekt
$pdf = new FPDF;

// GET DB-Daten
$dbm->GETSelectAufnahmedata($sin_objtid,$arrAufnahmedaten);

// PDF Formular zusammenstellen
$pdf->SetFont('Arial','',12);
$pdf->AddPage();

// EXCEL Überschriften
$pdf->SetFillColor(70,100,140);
$pdf->SetTextColor(255,255,255);
$pdf->SetDrawColor(128,0,0);
$pdf->SetLineWidth(.3);
$pdf->SetFont('','B');
$pdf->Cell(190,7,$sin_objname,1,0,'L',1);
$pdf->Ln();
$pdf->SetFillColor(202,213,227);
$pdf->SetTextColor(0,0,0);
$pdf->SetDrawColor(128,0,0);
$pdf->SetLineWidth(.3);
$pdf->SetFont('','B');
$pdf->Cell(190,7,'Aufnahmefläche',1,0,'L',1);
$pdf->Ln();
$pdf->SetFillColor(204,204,204);
$pdf->SetTextColor(0,0,0);
$pdf->SetDrawColor(128,0,0);
$pdf->SetLineWidth(.3);
$pdf->SetFont('','B');
$pdf->Cell(190,7,'Vegetationsdaten',1,0,'L',1);
$pdf->Ln();

if (is_array($arrAufnahmedaten))
{
    foreach ($arrAufnahmedaten as $Daten)
    {
        $pdf->SetFillColor(238,238,238);
        $pdf->SetTextColor(0,0,0);
        $pdf->SetDrawColor(128,0,0);
        $pdf->SetLineWidth(.3);
        $pdf->Cell(90,7,$Daten['ART'],1,0,'L',1);
        $pdf->SetFillColor(255,255,255);
        $pdf->SetTextColor(0,0,0);
        $pdf->SetDrawColor(128,0,0);
        $pdf->SetLineWidth(.3);
        $pdf->Cell(100,7,$Daten['VEGWERT'],1,0,'L',1);
        $pdf->Ln();
    }
}

// PDF Formular Ende
$pdf->Output();

?>

```



Lebenslauf

Daten zur Person:

| | |
|---------------------|-------------------|
| Name: | Hordynski Stephan |
| Geburtsdatum: | 3.4.1966 |
| Geburtsort: | Wien |
| Staatsbürgerschaft: | Österreich |

Chronologische Abfolge:

| | |
|--|----------------------------|
| Gartenbaufachschule: | 1981 - 1985 |
| Bundesheer: | 1.10.1985 - 31.8.1986 |
| Arbeit (angestellt): | Herbst 1986 – Februar 1992 |
| Facharbeiterintensivausbildung-Elektromechaniker-SW: | Feb.1992 - Feb.1993 |
| Siemens-EDV-Kurs: | März 1993 – August 1993 |
| Studienberechtigungsprüfung (Biologie und ET): | Okt. 1993 |
| Arbeit (angestellt): | 1.4.1994 - 30.9.1994 |
| Studium Elektrotechnik (ET): | WS 1994 – Abbruch 2001 |
| Studium Biologie - Ökologie: | SS 1996 – heute |
| Arbeit (angestellt/selbständig): | 1.7.2001 - heute |
| Diplomarbeit: | Jänner 2005 – Juli 2008 |