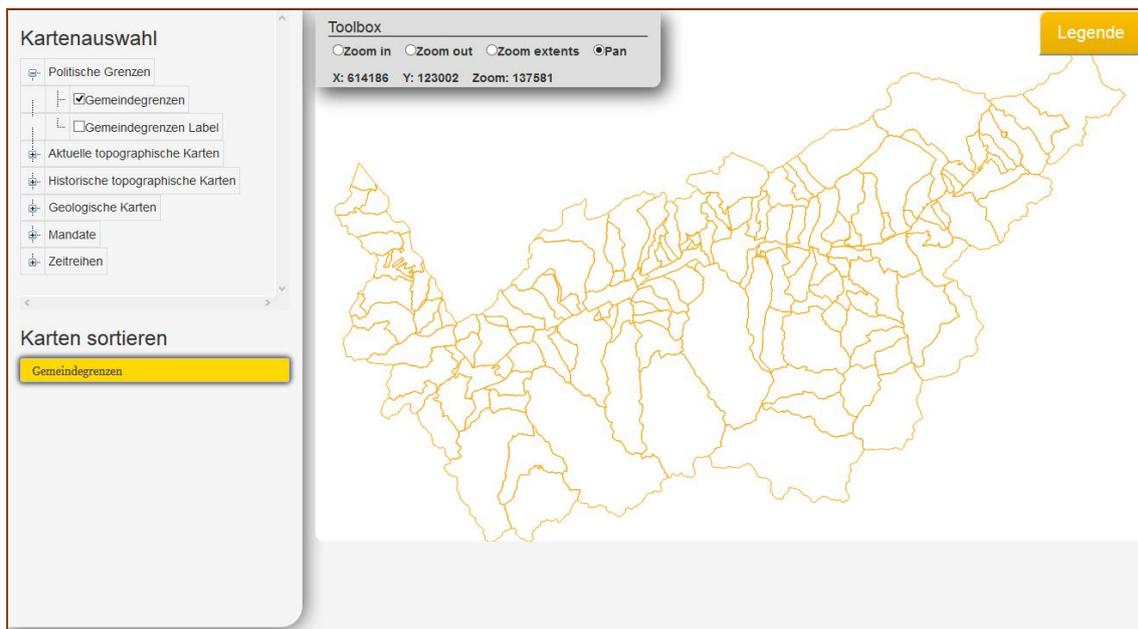


Studiengang Wirtschaftsinformatik

Bachelorarbeit 2017

Beurteilung der Anwendbarkeit vorhandener Web-GIS Module für die Rovina & Partner AG



Student : Michael Varonier

Dozent : Prof. Dr. René Schumann

Auftraggeber : Hermann Rovina (Rovina & Partner AG)

Abgabetermin : 02. August 2017

I. Zusammenfassung

Im Rahmen dieser Bachelorarbeit für die Rovina & Partner AG werden drei Ziele verfolgt. Der erste Schritt der Arbeit erfordert, dass verschiedene Web-GIS Module im .NET-Bereich beurteilt und miteinander verglichen werden. Anschliessend wird das Modul mit der besten Beurteilung als Entwicklungsgrundlage für die Erstellung eines Web-Geoinformationssystems eingesetzt, was als zweites Ziel der Arbeit gilt. Zum Schluss soll das System in die IT-Infrastruktur der Unternehmung integriert werden.

Hinsichtlich der zu erreichenden Ziele wird zuerst eine Analyse über die Unternehmung durchgeführt, in der die IT-Infrastruktur sowie die laufenden Datenverarbeitungsprozesse im Vordergrund stehen. Im Anschluss liegt der Fokus auf der Beurteilung vorhandener Web-GIS Module. Dazu werden verschiedene Vergleichskriterien definiert. Nach der Evaluation wird mithilfe des bestgeeigneten Moduls ein Web-GIS entwickelt, das in die Unternehmung integriert werden soll. Zuletzt kann beurteilt werden, ob sich das ausgewählte Web-GIS Modul für die Entwicklung des Geoinformationssystems geeignet hat.

II. Vorwort

Während meines Studiums beschäftigte mich die Rovina & Partner AG als Entwickler. Aufgrund der verschiedenen Projekte, für die ich in der Unternehmung verantwortlich war, kam mir die Idee, dass ich eines davon mit meiner damals zukünftigen Bachelorarbeit verbinden könnte. Demzufolge hat der Geschäftsführer Herr Hermann Rovina zusammen mit mir im Sommer 2016 ein passendes Thema ausgearbeitet.

Aufgrund den vorhandenen Projekten in der Unternehmung wurde ersichtlich, dass sich die Arbeit um die Entwicklung eines Web-Geoinformationssystems handeln könnte. Nach Einverständnis seitens der Rovina & Partner AG, dass die Bachelorarbeit mit einem internen Projekt verbunden werden kann, wurde ein erster Entwurf für das Einreichen des Themas erstellt.

Indes habe ich mir Gedanken gemacht, welcher Dozent ich für die Betreuung der Arbeit anfragen könnte. Schnell war für mich klar, dass es sich dabei um Prof. Dr. René Schumann handeln würde. Grund für die Wahl ist, dass ich seine Kompetenzen als Dozent wie auch seine Umgangsart mit den Studierenden während meiner Studienzeit sehr geschätzt habe.

Die Motivation für die Durchführung der Arbeit galt der Erstellung eines Systems, mit dem eine Automatisierung auf höchstem Niveau bezüglich den Datenverarbeitungsprozessen in der Unternehmung erreicht werden konnte.

Gerne möchte ich mich deshalb bei der Rovina & Partner AG bedanken, besonders bei Herrn Hermann Rovina, der mich während der gesamten Bachelorarbeit intensiv unterstützt hat. Dieselbe Danksagung gilt Prof. Dr. René Schumann, der ohne überlegen zu müssen die Funktion als Betreuer für meine Bachelorarbeit übernahm und mich ab diesem Zeitpunkt in gleicher Weise unterstützte.

III. Inhaltsverzeichnis

I.	Zusammenfassung	I
II.	Vorwort.....	II
III.	Inhaltsverzeichnis.....	III
IV.	Tabellenverzeichnis.....	V
V.	Abbildungsverzeichnis.....	VI
	Einleitung.....	1
1.	Unternehmensanalyse	3
1.1	Unternehmensbeschreibung.....	3
1.2	Datenverarbeitung in der Unternehmung.....	4
1.2.1	Datenstruktur	4
1.2.2	Fallbeispiel Datenstruktur.....	6
1.2.3	Verarbeitungsprozess	7
1.3	Informationssysteme und Technologien in der Unternehmung	9
1.3.1	Tätigkeitsfeld Informatik	10
2.	Problemstellung	13
3.	Analyse & Evaluation vorhandener Web-GIS Module.....	16
3.1	Marktübersicht.....	16
3.2	Wahl der Vergleichskriterien.....	18
3.3	Wahl der Module	24
3.4	Modulvergleiche durchführen	25
3.5	Bewertung der Module	30
3.6	Wahl des bestgeeignetsten Moduls.....	31

4.	Web-Geoinformationssystem	32
4.1	Drei-Schichten-Architektur.....	32
4.1.1	Datenmodelle und Objekte.....	33
4.1.2	Präsentationslayer	35
4.1.3	Applikationslayer.....	53
4.1.4	Datenbanklayer und GIS-Datenbank.....	60
4.2	Integration in die IT-Infrastruktur der Unternehmung	69
5.	Evaluation	71
5.1	Evaluation zur Anwendbarkeit des gewählten Web-GIS Moduls.....	71
5.2	Evaluation zum erstellten Web-GIS	77
	Schlussfolgerung und Ausblick	80
VI.	Glossar.....	82
VII.	Literaturverzeichnis	84
	Anhang	90
	Anhang I: Methode initializeTree	90
	Anhang II: Methode checkILayer	91
	Anhang III: Methode InitializeMap	92
	Anhang IV: Methode initializeLegende	93
	Anhang V: Methode createKlickBaum	94
	Anhang VI: Methode setKonkreteLayer	96
	Anhang VII: Methode getSpezialLayer	97
	Selbstständigkeitserklärung	99

IV. Tabellenverzeichnis

Tabelle 1: Marktübersicht vorhandener Web-GIS Module	16
Tabelle 2: Beispiel internes Bewertungssystem	18
Tabelle 3: Angeforderter Funktionsumfang	23
Tabelle 4: Modulauswahl	25
Tabelle 5: Internes Bewertungssystem «Technologie»	27
Tabelle 6: Internes Bewertungssystem «Dokumentation»	29
Tabelle 7: Internes Bewertungssystem «Funktionsumfang»	29
Tabelle 8: Internes Bewertungssystem «Modularität des GIS-Moduls»	30
Tabelle 9: Schlussbewertung Evaluation der Module	30

V. Abbildungsverzeichnis

Abbildung 1: Auftraggeber RPGeol AG (Rovina & Partner AG, 2017, S. 3).....	4
Abbildung 2: Datenstruktur (Rovina & Partner AG, 2011, S. 17)	5
Abbildung 3: Beispiel Datenmodellierung der Datenstruktur.....	6
Abbildung 4: Fallbeispiel der Datenstruktur (Mattertal) (Rovina & Partner AG, 2011, S. 17)	7
Abbildung 5: Verarbeitungsprozess (Rovina & Partner AG, 2011, S. 18).....	8
Abbildung 6: Informationssysteme und Technologien der RPGeol AG (Rovina & Partner AG, 2017, S. 14).....	9
Abbildung 7: Beispiel Webportal (Rovina & Partner AG, 2011, S. 19).....	12
Abbildung 8: Drei-Schichten-Architektur Web-GIS	32
Abbildung 9: Datenmodelle und Objekte.....	33
Abbildung 10: Modell LayerInformationen.....	34
Abbildung 11: UML-Diagramm der Datenmodelle	35
Abbildung 12: Web-GIS Benutzeroberfläche.....	36
Abbildung 13: Kartenverwaltung	37
Abbildung 14: Methode initializeTree (Kartenauswahl).....	38
Abbildung 15: Methode isErlaubt (Kartenauswahl).....	39
Abbildung 16: Methode addSortableLayer (Kartensortierung).....	40
Abbildung 17: Methode checkLayer (Kartensortierung).....	41
Abbildung 18: Einbinden JQueryUI (Kartensortierung).....	42
Abbildung 19: Bereich für die Kartenanzeige	42
Abbildung 20: Methode InitializeMap (Kartenanzeige)	43
Abbildung 21: Methode CreateMap (Kartenanzeige).....	43

Abbildung 22: Methode setMapFunctions (Kartenanzeige)	44
Abbildung 23: Beispiel Legende.....	45
Abbildung 24: Methode initializeLegende (Legende).....	46
Abbildung 25: Klickbarer "logischer Layer	47
Abbildung 26: Beispiel Kartenausschnitt Speziallayer	47
Abbildung 27: Methode isClicked (SpezialLayer)	48
Abbildung 28: Methode sLayerClicked (SpezialLayer)	49
Abbildung 29: Dokumentenbereich	50
Abbildung 30: Methode createKlickBaum (Dokumentenbereich).....	51
Abbildung 31: Methode AnzeigeButton_Click (Download Dokument)	52
Abbildung 32: Dokumente herunterladen	53
Abbildung 33: Vorgang Datentransformation	54
Abbildung 34: Methode InitializeMap (Kartenschichten).....	55
Abbildung 35: Methode setKonkreteLayer (Kartenschichten).....	55
Abbildung 36: Zugriff auf Datenquellen (Applikationsschicht).....	56
Abbildung 37: Code Erstellung GDAL-Kartenschicht (GDALManager).....	56
Abbildung 38: Code Erstellung ShapeFile-Kartenschicht (ShapeFileManager)	57
Abbildung 39: Methode getSpezialLayer (SpezLayerManager).....	58
Abbildung 40: "IGeometry"-Collection füllen (getSpezialLayer)	59
Abbildung 41: Erweiterung DataSource (getSpezialLayer).....	59
Abbildung 42: CustomTheme für Symbole (getSpezialLayer)	60
Abbildung 43: Datenbank-Diagramm	61
Abbildung 44: Kartenverwaltung DB-Modell.....	62

Abbildung 45: Kartenanzeige DB-Modell.....	63
Abbildung 46: Vordefinierte Einträge - Tabelle "Layertyp"	64
Abbildung 47: Tabelle "KonkreteLayer" (Kartendateien)	64
Abbildung 48: Tabelle "KonkreteLayer" (Spezial- und Dokumentenlayer)	65
Abbildung 49: Tabelle "KonkreteLayer" (Label).....	68
Abbildung 50: Legendenbereich DB-Modell	69
Abbildung 51: Integration in die IT-Infrastruktur.....	70
Abbildung 52: Web-GIS nach Deployment.....	70
Abbildung 53: Beispiel der Arten von Kartenschichten in SharpMap (SharpMap, 2017b).	76

Einleitung

Die Rovina & Partner AG (RPGeol AG) ist eine Unternehmung im Bereich Ingenieurwesen, insbesondere Geologie und Hydrologie mit Hauptsitz in Varen und einer Filiale in Visp. Aufgrund ihrer Tätigkeitsgebiete werden viele Messdaten erhoben, die im Rahmen von verschiedenen Aufträgen zur Weiterverarbeitung benötigt und anschliessend den Kunden in Form von Auswertungen oder Beurteilungen zur Verfügung gestellt werden.

Bei den erhobenen Daten der RPGeol AG handelt es sich grundsätzlich um Informationen, die ortsgebundene Eigenschaften aufweisen. Deshalb arbeitet die Unternehmung intern mit verschiedenen geografischen Informationssystemen (GIS), um sie zu verwalten.

Allerdings sind diese Systeme nur lokal verfügbar und können nicht als Online-Anwendungen eingesetzt werden, was folglich den Zugriff auf solche Daten über das Internet verhindert. Diese Sachlage gilt als Problemstellung dieser Arbeit, die von der RPGeol AG in Auftrag gegeben wurde. Mit der Integration eines Web-GIS in die Unternehmung soll das Problem gelöst werden.

Im Rahmen dieser Bachelorarbeit werden drei Ziele verfolgt. Der erste Schritt der Arbeit erfordert, dass verschiedene Web-GIS Module im .NET-Bereich beurteilt und miteinander verglichen werden. Anschliessend wird das Modul mit der besten Beurteilung als Entwicklungsgrundlage für die Erstellung eines Web-Geoinformationssystems eingesetzt, was als zweites Ziel der Arbeit gilt. Zum Schluss soll das System in die IT-Infrastruktur der Unternehmung integriert werden.

Nachdem die Integration in die Unternehmung erfolgt ist, soll das Abfragen von Daten, wie beispielsweise Auswertungen oder Beurteilungen, über das Internet möglich sein. Dadurch haben die Kunden direkten Online-Zugriff auf die von der RPGeol AG zur Verfügung gestellten Daten.

Damit die Ziele der Arbeit erreicht werden können, muss zuerst im Kapitel eins die RPGeol AG genauer analysiert werden. Diesbezüglich stehen vor allem die vorhandene IT-Infrastruktur sowie laufende Datenverarbeitungsprozesse im Vordergrund.

Das zweite Kapitel bezieht sich auf die Problemstellung, die anhand eines konkreten Fallbeispiels genauer erläutert wird. Des Weiteren werden in diesem Kapitel die Funktionalitäten und Bedingungen definiert, die das Web-GIS beinhalten soll.

Im Kapitel drei geht es um die Analyse und Evaluation vorhandener Web-GIS Module. Dazu werden verschiedene Vergleichskriterien definiert. Darüber hinaus soll in diesem Kapitel entschieden werden, welches Modul als Entwicklungsgrundlage am besten geeignet ist.

Anschliessend wird das Geoinformationssystem erstellt und in die IT-Infrastruktur integriert. Die Beschreibung dieses Vorgangs steht in Kapitel vier.

Der letzte Teil der Arbeit wird im Kapitel fünf behandelt und bezieht sich auf die Evaluation zur Anwendbarkeit des gewählten Web-GIS Moduls sowie die Evaluation zum erstellten Web-GIS.

Für spezielle Begriffe in dieser Arbeit wurde ein Glossar erstellt, in dem die dazugehörigen Erklärungen stehen.

1. Unternehmensanalyse

Dieses Kapitel beschäftigt sich mit der Analyse der Rovina & Partner AG. Hinsichtlich dessen erfolgt zu Beginn eine grundlegende Beschreibung über die Unternehmung. Anschliessend werden die Datenverarbeitungsprozesse in der Firma untersucht. Zuletzt wird der Fokus auf die Analyse über vorhandene Informationssysteme und Technologien in der RPGeol AG gelegt.

1.1 Unternehmensbeschreibung

Die Rovina & Partner AG ist ein Büro im Bereich Ingenieurwesen, insbesondere Geologie und Hydrologie mit Hauptsitz in Varen (Mittelwallis) und einer Filiale in Visp (Oberwallis). Sie wurde im Jahr 1990 von Herrn Hermann Rovina gegründet, der bis heute Geschäftsführer der Unternehmung ist. Die Firma zählt 17 Mitarbeiter. Darunter befinden sich Geologen, Hydrogeologen, Mess- und Informatikspezialisten sowie Personal in der Administration. Der geplante Jahresumsatz für das Jahr 2017 beläuft sich auf zirka zwei Millionen Franken (Rovina & Partner AG, 2017, S. 2).

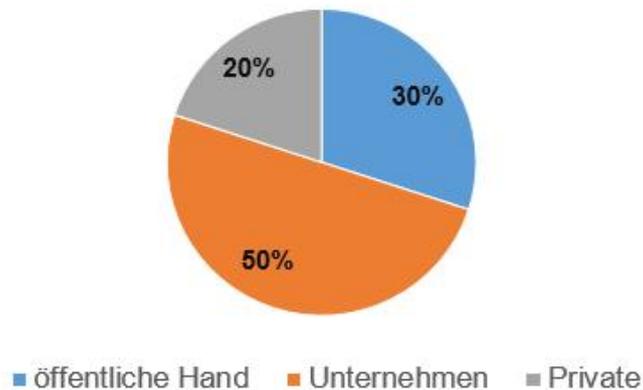
Zu den Tätigkeitsgebieten der Unternehmung gehören eine Vielzahl von verschiedenen Bereichen. Von Geologie und Geotechnik, Hydrogeologie, Naturgefahren und Geomesstechniken bis hin zum Bereich der Informatik erstreckt sich das Tätigkeitsfeld der RPGeol AG.

In der Geologie und Geotechnik werden Arbeiten wie beispielsweise Baugrunduntersuchungen, Baugruben- und Böschungssicherungen sowie Grundlagenstudien durchgeführt. Im Bereich der Hydrogeologie beschäftigt sich die Unternehmung mit Quell- und Grundwassererfassungen, Quellschutzzonen und Grundwassermodellierungen. Für die Prävention von Naturgefahren gilt der Einsatz von Naturgefahrenmanagement und -monitoring, die Erstellung von Studien sowie die Durchführung von Sanierungsprojekten. Des Weiteren wendet die Unternehmung verschiedene Geomesstechniken in den genannten Tätigkeitsgebieten an (Rovina & Partner AG, 2017, S. 5 - 9).

Die aufgeführten Arbeiten zählen zu den Kernelementen im Arbeitsbereich der RPGeol AG und decken nur einen Teil der Arbeitskompetenz in den jeweiligen Bereichen ab. Darüber hinaus wird das Tätigkeitsgebiet der Informatik in diesem Kapitel genauer beschrieben.

Aufträge erhält die RPGeol AG von verschiedenen Instanzen, die sich gemäss Abbildung 1 wie folgt zusammensetzen:

Abbildung 1: Auftraggeber RPGeol AG (Rovina & Partner AG, 2017, S. 3)



Der grösste Anteil an Arbeitsaufträgen stammt von Unternehmen. Dazu gehören vor allem Bahnen, Bauunternehmen, Kraftwerke sowie die Industrie. Sie decken die Hälfte aller Aufträge ab. Weitere 30 Prozent der Arbeiten werden von der öffentlichen Hand vergeben. Damit sind der Bund, die Kantone sowie die Gemeinden der Schweiz gemeint. Die restlichen 20 Prozent stammen von Privatpersonen (Rovina & Partner AG, 2017, S. 3).

1.2 Datenverarbeitung in der Unternehmung

Aufgrund der in der Unternehmungsbeschreibung genannten Tätigkeitsgebiete werden von der RPGeol AG viele Messdaten erhoben (Rovina & Partner AG, 2011, S. 6), die im Rahmen von verschiedenen Aufträgen zur Weiterverarbeitung benötigt werden.

Um diesen Datenverarbeitungsprozess übersichtlich darzulegen, werden drei Punkte betrachtet. An erster Stelle wird die Datenstruktur aufgezeigt. Sie gewährleistet einen Überblick, in welcher Form die erhobenen Daten gespeichert werden. Anschliessend wird anhand eines konkreten Fallbeispiels gezeigt, was als Ursprung für die definierte Struktur gilt. Zuletzt werden die einzelnen Schritte des Prozessablaufs analysiert. Diesbezüglich wird jede Etappe des Prozesses im Detail erklärt.

1.2.1 Datenstruktur

Die Abbildung 2 zeigt auf, wie die gesammelten Informationen aus einer Datenerhebung strukturiert werden. Diese Struktur ist in der Praxis flexibel und wird auf die jeweilige Situation angepasst. Des Weiteren sind in der Abbildung Beispielwerte aufgeführt. Sie stammen aus dem Fallbeispiel, das im zweiten Punkt der Datenverarbeitung genauer beschrieben wird.

Abbildung 2: Datenstruktur (Rovina & Partner AG, 2011, S. 17)



Die erhobenen Daten sind in drei Bereiche aufgeteilt. Dazu gehören der Messstandort, der Messpunkt sowie die Messdaten. Mittels diesen Informationen werden intern von der RPGeol AG diverse Auswertungen und Beurteilungen erstellt (Rovina & Partner AG, 2011, S. 6).

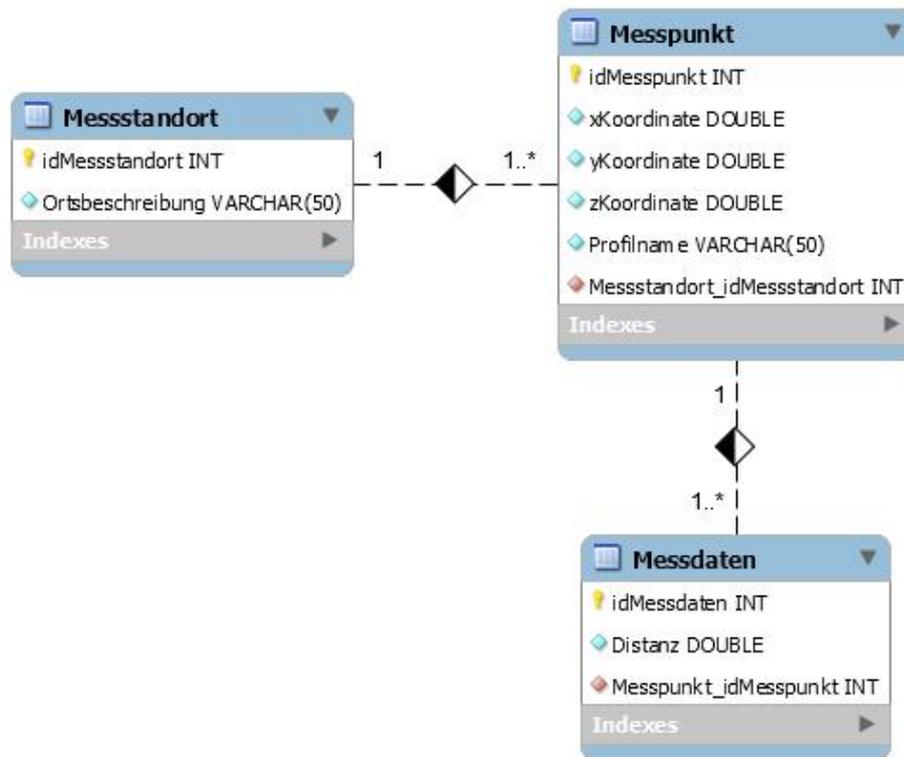
Der erste Bereich besteht aus einer Variable, die den Standort der Messung beschreibt. In diesem Fall wurde eine Messung am Felsen «Goldfad Süd» durchgeführt, der sich im Mattertal (Oberwallis) befindet. Es können mehrere Messungen am selben Ort getätigt werden.

Für den Messpunkt werden mehrere Informationen benötigt. Nebst den Koordinatenpunkten muss die Messung mit einem Profilnamen aufgezeichnet werden, wie beispielsweise «CM07». In der Praxis wird im Anschluss diese Benennung einem entsprechenden Mandat zugeteilt. Die Kombination aus dem Namen des Mandates und dem Messprofil gewährleistet eine Eindeutigkeit der entsprechenden Datenerhebung. Falls mehrere Messungen am selben Punkt stattfinden, werden sie aufsteigend nummeriert.

Zu den Messdaten gehören abschliessend die Resultate, die aus der jeweiligen Untersuchung hervorgingen. Es werden Distanzen, Verschiebungen, Kluftweiten und weitere Berechnungen erfasst (Rovina & Partner AG, 2011, S. 18). Im Arbeitsalltag der Unternehmung werden meistens mehrere Variablen für die Aufzeichnung dieser Informationen benötigt.

Die Abbildung 3 zeigt als Beispiel eine Datenmodellierung für die Struktur der Datenerhebung:

Abbildung 3: Beispiel Datenmodellierung der Datenstruktur

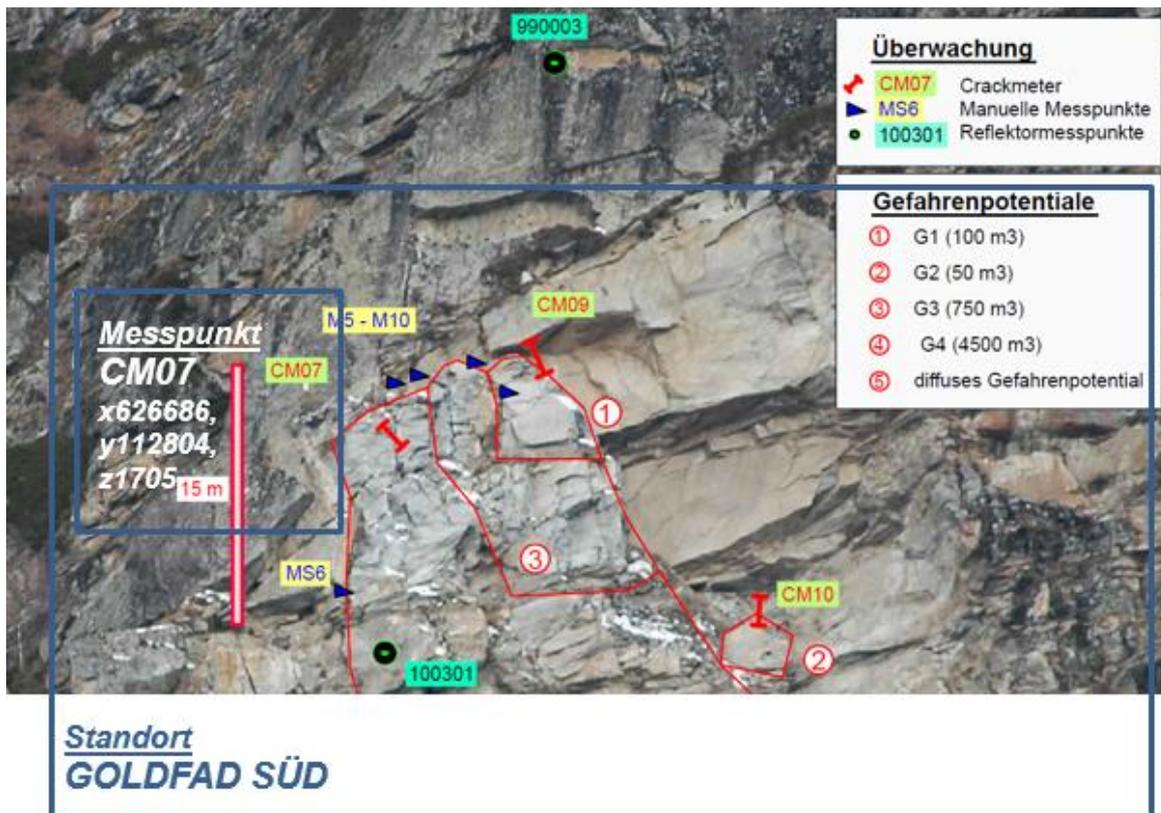


Wichtig zu beachten sind die Kardinalitäten zwischen den einzelnen Objekten. So beinhaltet ein Messstandort einen oder mehrere Messpunkte. Dasselbe gilt für die Messdaten. Bei jedem Messpunkt können mehrere Messungen durchgeführt werden. Umgekehrt gehören die erhobenen Messdaten genau zu einem Messpunkt und jeder Punkt zu einem Standort. Wie bereits erwähnt wird diese Struktur in der Praxis auf die jeweilige Situation angepasst. Beispielsweise werden bei den Messdaten grösstenteils mehrere Variablen benötigt.

1.2.2 Fallbeispiel Datenstruktur

Mit dem konkreten Beispiel soll der Ursprung für die definierte Datenstruktur aufgezeigt werden. Es wird ersichtlich, wie die Datenerhebung in der Praxis aussieht. In diesem Fall wurde eine Situation im Matteredal betrachtet, die in der Abbildung 4 dargestellt ist.

Abbildung 4: Fallbeispiel der Datenstruktur (Mattertal) (Rovina & Partner AG, 2011, S. 17)



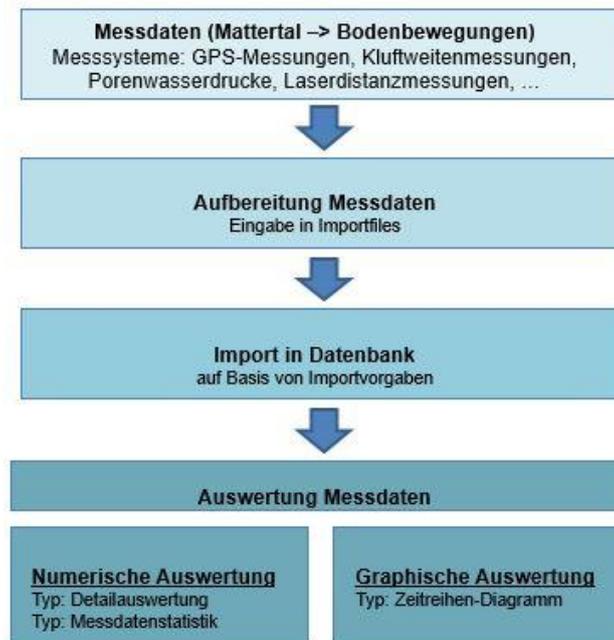
Hier ist die Felswand «Goldfad Süd» abgebildet. Sie gilt als Standort für alle in ihr getätigten Messungen. Im Weiteren ist die Messung mit dem Profilnamen «CM07» markiert, die zusammen mit den Koordinaten den Messpunkt darstellen. Als Messdaten dienen in diesem Beispiel die mit einem Crackmeter erfasste 15 Meter lange Riss in der Felswand. Ein solches Instrument kann Verschiebungen zwischen zwei Elementen messen und erkennen (Swiss Environment, kein Datum).

Nachdem die Daten erfasst sind, werden sie für die weiteren Schritte im Verarbeitungsprozess freigegeben.

1.2.3 Verarbeitungsprozess

In diesem Abschnitt erfolgt die Darlegung des Prozesses, wie die Daten von der Erhebung bis hin zur Weiterverarbeitung behandelt werden. In der Abbildung 5 wird der Vorgang in einzelnen Arbeitsschritten aufgezeigt:

Abbildung 5: Verarbeitungsprozess (Rovina & Partner AG, 2011, S. 18)



Der erste Schritt im Verarbeitungsprozess bezieht sich auf die Entstehung der Daten. Dementsprechend führt die RPGeol AG diverse Untersuchungen und Messungen an verschiedenen Standorten durch (Rovina & Partner AG, 2011, S. 16). Die daraus gewonnenen Resultate und Erkenntnisse sind die erhobenen Daten und werden in den folgenden Schritten im Prozess verarbeitet.

Anknüpfend an die Datenentstehung werden die Resultate in das richtige Format gebracht und in Dateien (Excel oder ASCII) zwischengespeichert (Rovina & Partner AG, 2011, S. 18), die für den Datenimport in die interne Unternehmensinfrastruktur benötigt werden. Die Struktur der Dateien ist im Punkt der Datenstruktur aufgezeigt.

Diese gespeicherten Daten werden im Anschluss in die Datenbanken der RPGeol AG importiert, damit sie der Unternehmung persistent zur Verfügung stehen und wiederverwendet werden können.

Der letzte Schritt des Prozesses gilt der Erstellung von Auswertungen und Beurteilungen, die in zwei unterschiedlichen Formen auftreten. Dazu gehören einerseits numerische und andererseits graphische Auswertungen, die mittels diversen Berechnungs- und Verarbeitungsprogrammen verfasst werden.

Nach abgeschlossener Datenverarbeitung werden die aus dem Prozess hervorgehenden gesammelten Resultate und Erkenntnisse zusätzlich in Dokumente festgehalten und für den praktischen Einsatz freigegeben. Diese sind ebenfalls auf den Datenbanken der RPGeol AG

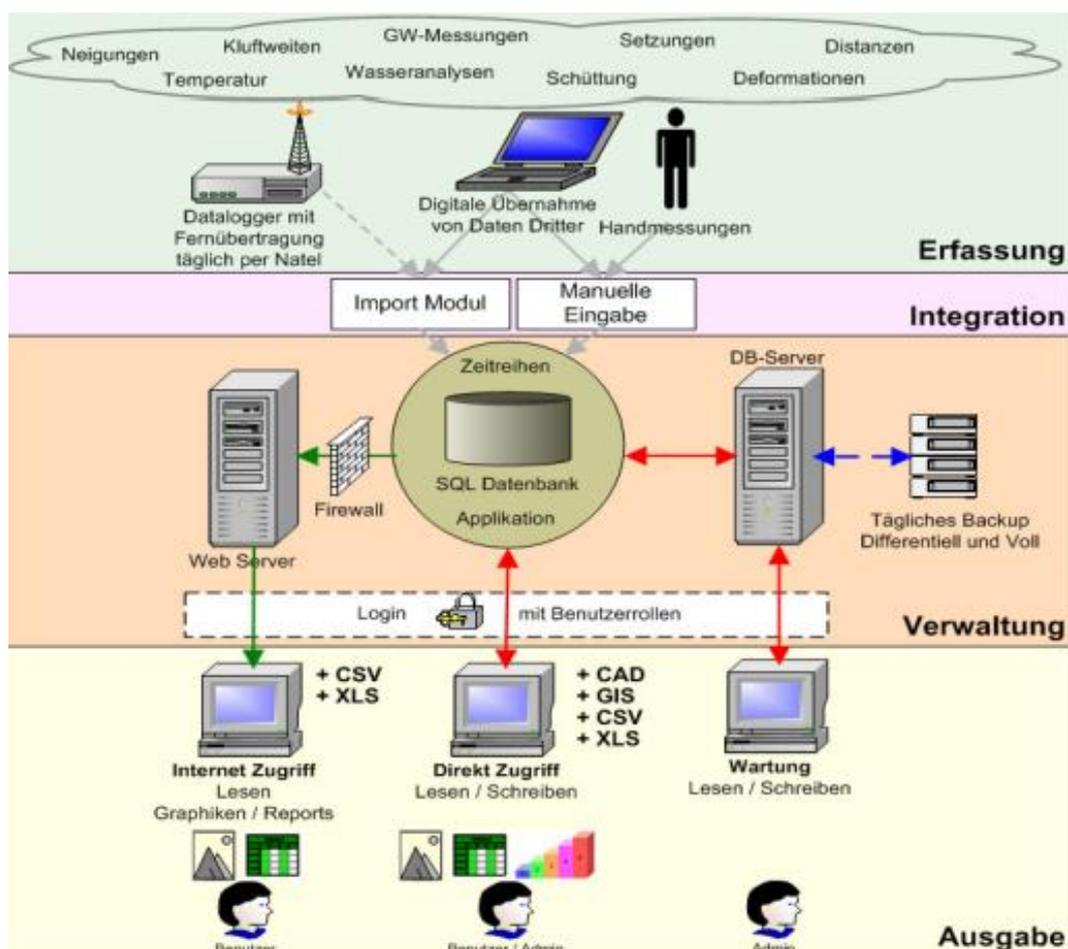
gespeichert (Rovina & Partner AG, 2011, S. 6). Zudem werden die internen geographischen Informationssysteme mit den neuen ortsgebundenen Messdaten erweitert. Weitere Daten, die mit den internen GIS verarbeitet werden, sind meistens im Shapefile, ECW- oder JPG2000-Format (Rovina & Partner AG, 2017, S. 12) (siehe Glossar).

1.3 Informationssysteme und Technologien in der Unternehmung

In diesem Abschnitt werden zuerst die vorhandenen Informationssysteme und Technologien der RPGeol AG aufgezeigt. Dadurch wird ersichtlich, wie der Datenverarbeitungsprozess in der vorhandenen IT-Infrastruktur integriert ist. Des Weiteren sind die Arbeiten aufgelistet, die im Tätigkeitsfeld der Informatik ausgeführt werden.

Die Abbildung 6 stellt den Architekturaufbau der Systeme und Technologien der Unternehmung dar:

Abbildung 6: Informationssysteme und Technologien der RPGeol AG (Rovina & Partner AG, 2017, S. 14)



Gemäss der Abbildung ist die Systemarchitektur der Unternehmung in vier Themenbereiche aufgeteilt. Dazu gehören die Themen Erfassung, Integration, Verwaltung sowie die Ausgabe.

Der erste Bereich bezieht sich auf die Datenerhebung. Sie ist im Kapitel der Datenverarbeitung genauer beschrieben worden. Diverse Messinstrumente erfassen Daten, mit denen die Unternehmung im Anschluss gemäss dem Datenverarbeitungsprozess Beurteilungen und Auswertungen erstellt.

Das Gebiet der Integration dient als Schnittstelle zwischen dem Umfeld ausserhalb der Unternehmung sowie deren internen Systemlandschaft. In diesem Bereich kommen die im Verarbeitungsprozess aufgezeigten Import-Dateien zum Einsatz, auf denen die erhobenen Daten abgespeichert werden. Einzelne Messinstrumente sind in der Lage, über Module die erfassten Resultate automatisch abzuspeichern, bei Handmessungen beispielsweise müssen die Erkenntnisse manuell in die Dateien eingetragen werden. Diese Dateien werden anschliessend per E-Mail an die Rovina & Partner AG gesendet, damit die gespeicherten Daten in die Datenbanken importiert werden können.

Im Bereich der Verwaltung ist die interne IT-Infrastruktur der Unternehmung aufgeführt. Als Grundlage für die Datenverwaltung dienen MS-SQL Datenbanken. Der Datenbankserver läuft auf einem internen Dateisystem, von denen mehrere in der RPGeol AG vorhanden sind. Um seine Sicherheit zu gewährleisten werden Backups von den Datenbanken durchgeführt. Des Weiteren ist die gesamte interne Infrastruktur mit einer Firewall geschützt. Unerlaubte Verbindungen werden dadurch unterbunden. Darüber hinaus existieren von der Unternehmung eigens erstellte Benutzerrollen, die als zusätzliche Sicherheit dienen. Der Zugriff von aussen wird über einen Webserver gewährleistet.

Der letzte Teil der Architektur ist für den Endbenutzer bestimmt. Als Anwender gelten interne sowie externe Personen. Benutzer ausserhalb der Unternehmung können nur über zur Verfügung gestellte Webportale der RPGeol AG auf Daten zugreifen. Die Portale werden im Tätigkeitsfeld Informatik näher beschrieben. Benutzer innerhalb der Unternehmung, wie beispielsweise Mitarbeiter, können interne Programme verwenden. Sie müssen sich jedoch lokal in der Unternehmung befinden.

1.3.1 Tätigkeitsfeld Informatik

Zu den Arbeiten im Bereich der Informatik gehören nebst der Verwaltung der IT-Infrastruktur die Anwendung verschiedener Programme (Rovina & Partner AG, 2017, S. 8).

Für die Administration wird ein internes Enterprise-Resource-Planning (ERP) System verwendet. Mit dem System können Arbeiten wie Projektplanungen, Mandatsverwaltungen, Stundenerfassungen bis hin zu Pendenzenverwaltungen durchgeführt werden. Das ERP wurde von der Unternehmung entwickelt und wird dadurch fortlaufend mit Modulen erweitert. Das System basiert auf Windows Forms und ist in der Programmiersprache C# (.NET) geschrieben.

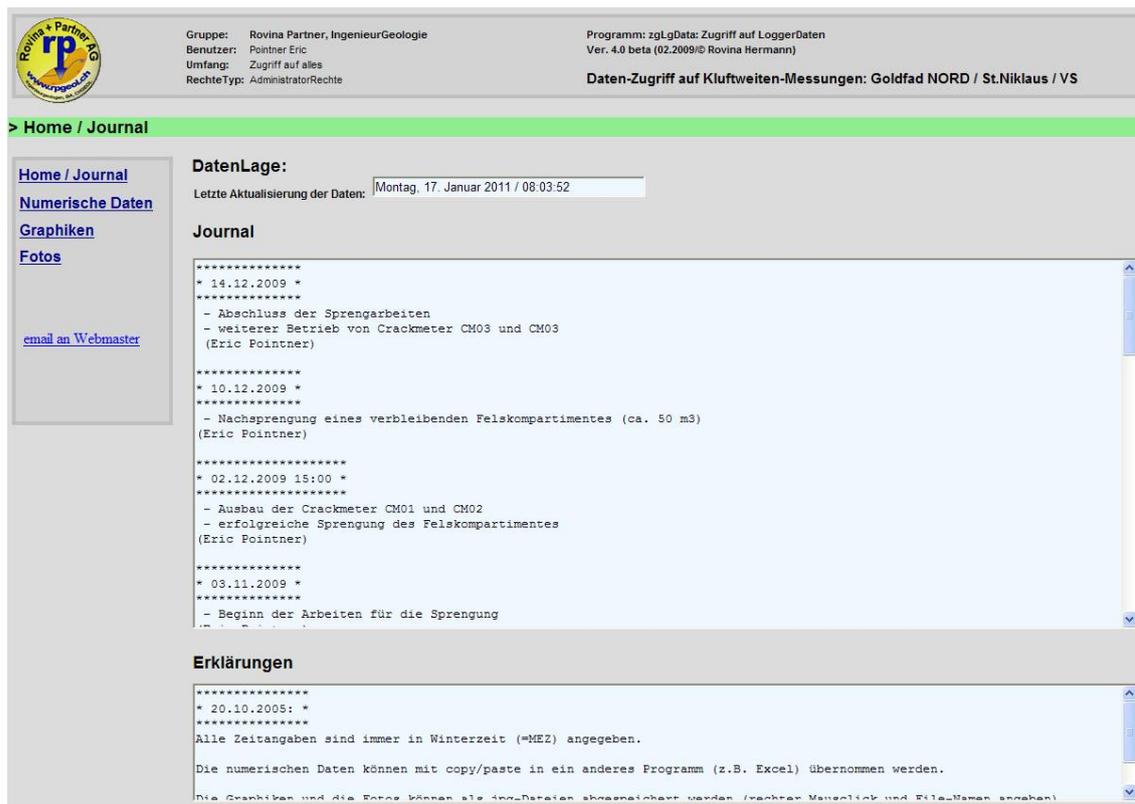
Zur Erstellung von Beurteilungen und Auswertungen dienen die internen «Zeitreihen». Darunter gehören eine Reihe von der RPGeol AG entwickelte Berechnungsprogramme sowie spezielle Datenbanken. Sie werden für den letzten Schritt im Datenverarbeitungsprozess angewendet und ebenfalls fortlaufend weiterentwickelt. Zur Erstellung dieser Programme wurde dieselbe Technologie wie beim ERP eingesetzt.

Die aus den «Zeitreihen» gewonnen Erkenntnisse und Daten werden im Anschluss in verschiedenen Programmen für die Weiterverarbeitung integriert. Darunter befinden sich Simulationsprogramme, geographische Informationssysteme sowie Statistik-Programme.

Der derzeitige Web-Bereich der Unternehmung beschränkt sich auf die Anzeigemöglichkeit von Naturgefahrenmonitoring. Dementsprechend können nur wenige interne Daten über das Internet aufgerufen und abgefragt werden, was der Grund dafür ist, dass die RPGeol AG zukünftig vermehrt den Fokus auf diesen Bereich legt (Rovina & Partner AG, 2017, S. 12).

Die Abbildung 7 ist ein Beispiel dafür, wie ein entsprechendes Webportal der Unternehmung aussieht. Darüber hinaus wird ersichtlich, wie Benutzer ausserhalb der RPGeol AG auf Auswertungen und Beurteilungen zugreifen können:

Abbildung 7: Beispiel Webportal (Rovina & Partner AG, 2011, S. 19)



The screenshot displays a web portal interface for Rovina & Partner AG. The header includes the company logo and user information: Gruppe: Rovina Partner, IngenieurGeologie; Benutzer: Pointner Eric; Umfang: Zugriff auf alles; RechteTyp: AdministratorRechte. The program information is: Programm: zgLgData: Zugriff auf LoggerDaten; Ver. 4.0 beta (02.2009)© Rovina Herrmann. The main content area is titled 'Daten-Zugriff auf Kluffweiten-Messungen: Goldfad NORD / St.Niklaus / VS'. The sidebar contains navigation links: Home / Journal, Numerische Daten, Graphiken, Fotos, and email an Webmaster. The main content area is divided into sections: DatenLage (Montag, 17. Januar 2011 / 08:03:52), Journal (listing activities from 2009), and Erklärungen (providing instructions on data usage).

Header Information:

- Gruppe: Rovina Partner, IngenieurGeologie
- Benutzer: Pointner Eric
- Umfang: Zugriff auf alles
- RechteTyp: AdministratorRechte
- Programm: zgLgData: Zugriff auf LoggerDaten
- Ver. 4.0 beta (02.2009)© Rovina Herrmann
- Daten-Zugriff auf Kluffweiten-Messungen: Goldfad NORD / St.Niklaus / VS

Navigation:

- Home / Journal
- Numerische Daten
- Graphiken
- Fotos
- [email an Webmaster](#)

DatenLage:

Letzte Aktualisierung der Daten: Montag, 17. Januar 2011 / 08:03:52

Journal

```

*****
* 14.12.2009 *
*****
- Abschluss der Sprengarbeiten
- weiterer Betrieb von Crackmeter CM03 und CM03
(Eric Pointner)
*****
* 10.12.2009 *
*****
- Nachsprengung eines verbleibenden Felskompartimentes (ca. 50 m3)
(Eric Pointner)
*****
* 02.12.2009 15:00 *
*****
- Ausbau der Crackmeter CM01 und CM02
- erfolgreiche Sprengung des Felskompartimentes
(Eric Pointner)
*****
* 03.11.2009 *
*****
- Beginn der Arbeiten für die Sprengung
..

```

Erklärungen

```

*****
* 20.10.2005: *
*****
Alle Zeitangaben sind immer in Winterzeit (=MEZ) angegeben.

Die numerischen Daten können mit copy/paste in ein anderes Programm (z.B. Excel) übernommen werden.

Die Graphiken und die Fotos können als jpg-Dateien abgespeichert werden. (rechter Mausklick und File-Namen angeben)

```

Das aufgezeigte Webportal bezieht sich auf den Datenzugriff von durchgeführten Kluffweiten-Messungen an der Felswand «Goldfad Nord» in St. Niklaus (Oberwallis). Der Benutzer kann nebst dem Journal numerische Daten, diverse Graphiken sowie Fotos abfragen.

2. Problemstellung

In diesem Bereich der Arbeit wird die Problematik der Rovina & Partner AG dargelegt. Zu Beginn steht die Erklärung, worin der Ursprung des Problems liegt. Anschliessend wird die Situation mit einem konkreten Fallbeispiel erläutert. Schliesslich wird anhand der Problemstellung festgelegt, was als Lösung für die gestellte Problematik dienen soll.

Wie die Unternehmensanalyse aufzeigt, handelt es sich bei den erhobenen Daten der RPGeol AG grundsätzlich um Messdaten, die ortsgebundene Eigenschaften aufweisen. Solche Daten werden intern mit verschiedenen geografischen Informationssystemen verarbeitet. Allerdings sind diese Systeme nur lokal verfügbar und können nicht als Online-Anwendungen eingesetzt werden, was eine geographische Datenrepräsentation gegen Aussen unmöglich macht.

Für das konkrete Fallbeispiel wird das Mattertal betrachtet, das von zahlreichen Naturgefahren bedroht wird. Dementsprechend messen verschiedene Systeme vor Ort Bodenbewegungen, die an die Unternehmung weitergeleitet werden (Rovina & Partner AG, 2011, S. 6 - 16). Solche Daten sowie deren Auswertungen und Beurteilungen müssen den Gemeinden und den wichtigsten Infrastrukturbetreibern, wie zum Beispiel Schmalspurbahnen oder dem Kanton, zur Verfügung gestellt werden. Ein Beispiel für die Datenerhebung im Mattertal ist im Kapitel der Unternehmensanalyse übersichtlich dargestellt. Aufgrund der Sensibilität der Daten soll der Datenzugriff gegenüber Drittpersonen benutzerspezifisch geregelt sein. Zusätzlich sollen die Auswertungen und Beurteilungen über das Internet erreichbar sein, damit Benutzer ausserhalb der Unternehmung darauf zugreifen können.

Als Lösung der Problematik gilt die Entwicklung eines Web-Geoinformationssystems, das den Zugriff auf geographische Daten der RPGeol AG über das Internet erlaubt. Des Weiteren soll das System in die IT-Infrastruktur der Unternehmung integriert werden, damit interne Daten abgefragt werden können.

Um die Erstellung eines derartigen GIS zu ermöglichen, bedarf es der Anwendung eines Web-GIS Moduls (Programm-bibliothek). Ein solches Modul enthält alle wichtigen Komponenten, die für die Entwicklung eines Geoinformationssystems programmiertechnisch von Nöten sind. Wichtig zu beachten ist der Unterschied zwischen Desktop- und Webanwendungen. Desktopanwendungen werden lokal eingesetzt, was in diesem Fall den eigentlichen Zweck verfehlt. Deshalb soll es mit dem Modul möglich sein, ein geographisches Informationssystem auf einer Website aufzubauen. Die Website kann über das Internet erreicht werden und bietet somit den verlangten Datenzugriff. Zudem darf das Modul nicht

kostenpflichtig sein und muss über eine Open Source Lizenz verfügen, damit die RPGeol AG zusätzliche Kosten sowie rechtliche Restriktionen vermeiden kann.

Angesichts der gewonnen Erkenntnisse aus der durchgeführten Unternehmensanalyse und der aufgezeigten Lösung wird ein Anforderungsprofil für das Geoinformationssystem erstellt. In diesem Profil sind die Funktionalitäten sowie Bedingungen aufgelistet, die das GIS beinhalten soll.

Es gelten fünf Anforderungen für die Zusammenstellung des Profils:

1. Erweiterbarer Inhalt

Die erste Anforderung bezieht sich auf den Aufbau des Geoinformationssystems. Das Ziel ist, das GIS so zu entwickeln, dass es sich dynamisch erstellt. Damit ist gemeint, dass seitens der Unternehmung keine Programmcodeänderungen nötig sind, um das System zu erweitern oder zu verringern. Als Erweiterung gilt beispielsweise das Hinzufügen von neuen Karten.

2. Benutzermanagement

Eine weitere Kondition bezüglich des Informationssystems bewegt sich im Bereich der Sicherheit. Um Schutz vor unerlaubtem Zugriff zu gewährleisten, sollen Benutzergruppen erstellt werden, die verschiedene Rechte erhalten. Dementsprechend kann jeder Anwender nur diejenigen Daten abrufen, die für ihn bestimmt worden sind. So hat zum Beispiel ein öffentlicher Benutzer nicht dieselben Rechte wie ein Mitarbeiter der RPGeol.

3. GIS-Funktionalitäten

Zu den Funktionalitäten, die das Geoinformationssystem beinhalten soll, gehören vier Bereiche. Der erste Bereich liegt im Datenzugriff, der definiert, welche Dateitypen vom System gelesen werden können. Der Unternehmensanalyse entsprechend werden intern meistens mit den Typen Shapefile, ECW sowie JPG2000 gearbeitet. Deshalb soll das GIS die Verarbeitung dieser Dateien ermöglichen.

Damit die Unternehmung die eigens erhobenen Daten in das Informationssystem einbinden kann, soll des Weiteren die Möglichkeit bestehen, geometrische Formen darzustellen. Als Geometrische Form gilt zum Beispiel die Erstellung eines Punktes. Solche müssen mit passenden Textangaben gekennzeichnet werden können. Für die Darstellung dieser Formen müssen zwangsläufig die Koordinaten X und Y angegeben werden.

Der dritte Bereich bezieht sich auf die Funktionen, die für die Steuerung des GIS notwendig sind. Die Mindestanforderung verlangt, dass sich der Benutzer durch die Karte navigieren

sowie den Massstab verändern kann. Des Weiteren soll er selbst bestimmen können, welche Karten auf der Map angezeigt werden.

Der letzte Teil verlangt, dass die angezeigten Kartenschichten (Kartenlayer) vom Anwender sortiert werden können, was den Umgang mit den von der RPGeol AG zur Verfügung gestellten Daten erleichtert und die Regelung der Sichtbarkeit erlaubt.

Diese Funktionalitäten sind wichtig, damit die Unternehmung die im Berufsalltag verwendeten Daten in das System einbinden kann. Darüber hinaus wird dadurch die Steuerung des Systems für den Benutzer ermöglicht.

4. Datenquellen

Aus der durchgeführten Analyse zur IT-Infrastruktur der Rovina & Partner AG wird ersichtlich, dass intern zwei Arten von Datenquellen existieren. Einerseits gilt als Quelle das laufende Dateisystem, andererseits die vorhandenen Datenbanken. Um den Arbeitsfluss nicht einzuschränken, soll das GIS vollumfänglich auf diese Quellen zugreifen können. So müssen zum Beispiel Dateien nicht an einem spezifischen Ort abgelegt werden, was unnötigen Zeitaufwand seitens der Unternehmung erzeugen würde. Der Datenzugriff soll direkt auf die originalen Daten erfolgen.

5. Dokumenten-Zugriff

Angesichts der bei der Unternehmungsanalyse aufgezeigten Datenverarbeitung ist die Unternehmung im Besitz von Dokumenten, auf denen beispielsweise Auswertungen oder Beurteilungen von Messungen gespeichert sind. Solche Daten sollen den wichtigsten Infrastrukturbetreibern zur Verfügung gestellt werden, wie aus der Problemstellung zu entnehmen ist. Dementsprechend gilt das Abfragen dieser Daten über das Geoinformationssystem als Bedingung fünf im Anforderungsprofil.

Die genannten Funktionalitäten und Bedingungen stellen somit das Anforderungsprofil des Geoinformationssystems zusammen. Im weiteren Verlauf der Arbeit werden diverse Entscheidungen mithilfe des erstellten Profils getroffen.

3. Analyse & Evaluation vorhandener Web-GIS Module

Um für die Entwicklung des geographischen Informationssystems ein passendes Modul zu finden, werden verschiedene GIS-Bibliotheken, die auf der .NET Technologie basieren, analysiert und anschliessend evaluiert. Das anhand der Bewertung zweckmässigste Modul wird anschliessend als Entwicklungsgrundlage für die Erstellung des GIS verwendet.

3.1 Marktübersicht

Als wichtigste Rahmenbedingung für den Aufbau der Marktübersicht gilt die Berücksichtigung der .NET Technologie, da die Entwicklung in die IT-Infrastruktur der RPGeol AG integriert werden soll. Zudem muss die Bibliothek die Erstellung eines Web-GIS ermöglichen und darf sich nicht auf den Bereich einer Desktopanwendung beschränken.

Die Tabelle 1 stellt anhand der durchgeführten Internetrecherche die Marktübersicht vorhandener Web-GIS Module zusammen.

Tabelle 1: Marktübersicht vorhandener Web-GIS Module

Modul 1	Modul 2	Modul 3	Modul 4	Modul 5	Modul 6
Easy GIS .NET	DotSpatial	SharpMap	MapAround	Cadcorp	ArcGIS

Easy GIS .NET

Die Easy GIS .NET Programmbibliothek wurde von Winston Fletcher im Jahre 2007 entwickelt. Der ursprüngliche Grund für die Erstellung des GIS-Moduls lag in der Absicht eine Bibliothek zu schaffen, die grosse Shapefiles verarbeiten kann. Nun umfasst das Modul die Möglichkeiten, geographische Informationssysteme im Bereich Desktopanwendung und Webapplikation zu entwickeln (Fletcher, kein Datum a).

DotSpatial

Die GIS-Bibliothek von DotSpatial bewegt sich ebenfalls im Bereich der .NET Technologie. Ihr derzeitiger Manager ist Dr. Dan Ames. Das Modul dient hauptsächlich zur Verarbeitung geographischer Daten sowie zur Abbildung von Karten. Darüber hinaus stellt DotSpatial die Möglichkeit zur Verfügung, ihre Bibliothek als Grundlage für die Entwicklung weiterer GIS-Module zu benutzen (Ames, 2016).

SharpMap

SharpMap ist eine Programmbibliothek, die für die Erstellung von Geoinformationssysteme entwickelt wurde. Es können Desktop- sowie Webapplikationen unter Benutzung derselben Bibliothek erstellt werden. Erste Veröffentlichungen des Moduls erfolgten im Jahre 2006. Weitere Details über die Entwicklung sowie die Geschichte des Moduls sind nicht vorhanden (SharpMap, 2015).

MapAround

Ein weiteres Mittel für die Entwicklung geographischer Systeme stellt die Bibliothek von MapAround dar. Als Rich Client-Applikation erlaubt sie die Erstellung von Karten und geometrischen Formen, die aus verschiedenen Datenquellen stammen. Mögliche Datenquellen sind Datenbanken sowie Dateien. Das Modul kommt beispielsweise beim Russischen Staatenbund für die Anzeige diverser Statistiken zum Einsatz. Ähnlich wie bei SharpMap gibt es wenige Informationen bezüglich der Entstehung, abgesehen vom russischen Ursprung des Projekts (MapAround, kein Datum b).

Cadcorp

Die Computer Aided Development Corporation Limited (Cadcorp) ist eine Software-Entwicklungsfirma, die geographische Informationssysteme sowie diverse Dienstleistungen anbietet. Nebst den eigens erstellten Geoinformationssystemen bietet Sie Module für Entwickler an, um eigene Software-Lösungen herzustellen (Cadcorp, kein Datum b). Ein angebotenes Werkzeug ist eine Programmbibliothek, mit der von Grund auf ein geographisches Informationssystem aufgebaut werden kann (Cadcorp, kein Datum a). Zusätzlich ermöglicht das Tool die Weiterentwicklung und den Ausbau der von Cadcorp bereitgestellten GIS (Cadcorp, 2016a).

ArcGIS

ArcGIS besteht aus mehreren Produkten der international tätigen Unternehmung ESRI Inc (Environmental Systems Research Institute). Sie gilt als globaler Marktführer im Bereich der geographischen Informationssysteme (ESRI Inc., kein Datum a). Zu den angebotenen Produkten im Entwicklungsbereich gehören zum Beispiel GIS-Programmbibliotheken, Software Development Kits und Anwendungsprogrammierschnittstellen (API). Darüber hinaus beschränkt sich ArcGIS, im Gegensatz zu den anderen aufgeführten Modulen, nicht auf den Bereich der .NET Technologie (ESRI Inc., kein Datum b).

3.2 Wahl der Vergleichskriterien

Bei der Aufstellung der Vergleichskriterien stehen die in der Problemstellung definierten Anforderungen an das Geoinformationssystem im Vordergrund. Darüber hinaus sind die bereits in die RPGeol AG integrierten Technologien von grosser Bedeutung. Jedes Kriterium erhält dementsprechend eine Priorität, die später für die Schlussbewertung relevant ist.

Die Abfolge der Definition eines Vergleichskriteriums besteht aus drei Teilen. Zu Beginn steht die Erklärung, was der ausschlaggebende Grund für die Wahl des angegebenen Kriteriums ist und wie stark es für die Schlussbewertung gewichtet wird. Der zweite Teil befasst sich mit der Erstellung eines internen Bewertungssystems. Das System beinhaltet messbare Einheiten, die den Modulen zugewiesen werden, damit sie miteinander verglichen werden können. Letztlich liegt der Fokus auf die Bestimmung, welcher Wert unter welchen Bedingungen verteilt wird.

Tabelle 2: Beispiel internes Bewertungssystem

Module	Bedingungen			Total	SB 3 Pkt.
	Bedingung 1 (max. 5 Punkte)	Bedingung 2 (max. 3 Punkte)	Bedingung 3 (max. 2 Punkte)		
1. Modul	5	3	2	10	3
2. Modul	2	3	2	7	
3. Modul	4	2	2	8	
4. Modul	3	3	2	8	

Die Tabelle 2 stellt ein Beispiel für ein internes Bewertungssystem dar. In diesem Fall bestehen die messbaren Einheiten aus Punkten, die den Modulen zugewiesen werden. Grüne Markierungen zeigen die Module mit der gesamthaft besten Bewertung. Die letzte Spalte «SB» dient für die Schlussbewertung und enthält die Punktzahl, die bei der Gewichtung dem Vergleichskriterium zugeteilt wurde. Nur die Module mit der jeweils besten Bewertung erhalten diese Punktzahl, wie hier beispielsweise das erste Modul.

Nachdem die internen Bewertungen aller Vergleichskriterien abgeschlossen sind, werden für jedes Modul die erreichten Punkte für die Schlussbewertung zusammengezählt. Das Modul mit der am höchsten erzielten Punktzahl dient im weiteren Verlauf als Entwicklungsgrundlage für die GIS-Implementation. Bei zwei gleichhoch bewerteten Modulen entscheidet der Entwickler anhand der restlichen Resultate über die Wahl des Moduls.

Für die Evaluationen werden vier Hauptkriterien erstellt. Dazu gehören die Technologie, die Dokumentation, der Funktionsumfang sowie die Modularität des GIS-Moduls. Mit folgenden Methoden werden die Module bewertet:

1. Technologie

Grund

Technologie steht für den wichtigsten Prüfstein im Bewertungsverfahren, da das geographische Informationssystem in die IT-Infrastruktur der Unternehmung integriert werden soll. Indes soll der geschriebene Programmcode des GIS im Anschluss der Implementation eine hohe Wiederverwendbarkeit sowie Modularität für Erweiterungen und Weiterentwicklungen ermöglichen, was je nach angewandter Technologie grosse Abweichungen haben kann. Aufgrund den bereits vorhandenen Technologien in der Unternehmung, die sich im .NET-Raum befinden, werden dynamische Web-Formen (ASP) für den Aufbau der Website verwendet. Dieses Vergleichskriterium steht an oberster Prioritätsstufe und erhält daher eine Punktzahl von vier für die Schlussbewertung.

Internes Bewertungssystem

Um Vergleiche in diesem Kriterium durchführen zu können, wird ein Punktesystem mit einer Skala von null bis drei verwendet. Folglich können maximal drei Punkte erreicht werden, die in Form von Pluszeichen auf der Tabellenauswertung ersichtlich sind. Der Bereich der Skala stellt sich aus drei technischen Bedingungen zusammen, die das GIS-Modul dem Entwickler zur Verfügung stellen muss.

Bedingungen

Die erste Kondition verlangt, dass die ausgewählte Bibliothek mit der Programmiersprache C# entwickelt wurde und demzufolge die Erstellung eines Geoinformationssystems mit Hilfe derselben Sprache ermöglicht, ohne andere Technologien miteinbeziehen zu müssen. Angewendete Sprachen wie HTML oder CSS werden für diese Bedingung nicht berücksichtigt. Grund für die Stellung dieser Forderung ist die Gewährleistung der Wiederverwendbarkeit und Modularität des geschriebenen Programmcodes für die RPGeol AG. Durch die Erfüllung dieser Kondition wird dem Modul ein Punkt angerechnet.

Des Weiteren soll das Element, auf dem die Karten angezeigt werden, in der von Microsoft zur Verfügung gestellten Toolbox vorhanden sein (Microsoft Corporation, 2007b). Von der GIS-Bibliothek eigens entwickelte Webserversteuerelemente (siehe Glossar), die als Kartengrundlage dienen, werden bei der Evaluation als nicht erfüllt gewertet. Dieses Kriterium

beabsichtigt die Verwendung vordefinierter Elemente, um folgende Weiterentwicklungen sowie technologische Veränderungen zu vereinfachen.

Beim letzten Bewertungspunkt geht es um die Anzeige von Karten ohne zusätzlichen Dateien, die in das Projekt eingebunden werden müssen. Beispielsweise soll für die Anzeige von einem Shapefile keine Referenz auf eine Zusatzdatei erforderlich sein. Solche Dateien können zum Beispiel exportierte Eigenschaften vom Shapefile in einem speziellen Format enthalten. Diese Bedingung wurde aufgestellt, damit der Entwickler unnötigen Aufwand zur Verwaltung solcher Dateien vermeiden kann.

2. Dokumentation

Grund

Ein weiteres Bewertungskriterium von Bedeutsamkeit bezieht sich auf den Bereich Dokumentation. Grund für die Wahl dieses Kriteriums sind die Vorteile, die ein Entwickler durch eine gutgeführte Wissensvermittlung über das angewendete Modul erhält. Nützliche sowie ausführliche Informationen über die Anwendbarkeit der Bibliothek erleichtern nicht nur die Arbeit des Entwicklers, sondern verringern auch die Zeit für das Lösen allfälliger Probleme. Darüber hinaus müssen dadurch die Entwickler der einzelnen Module nur bei spezifischen Fragen kontaktiert werden. Anlässlich der Wichtigkeit erhält dieses Vergleichskriterium eine Punktzahl von drei für die Schlussbewertung.

Internes Bewertungssystem

Das Kriterium wird ebenfalls mit einem System evaluiert, das für erfüllte Anforderungen Bewertungspunkte den jeweiligen Modulen zuteilt. Allerdings erhält jede Bedingung eine unterschiedliche Maximalpunktzahl, die aufgrund ihrer Wichtigkeit zugeteilt wird.

An erster und wichtigster Stelle stehen die Informationsangaben über die angebotenen Funktionen des GIS-Moduls. Hintergrund für eine derartig hohe Gewichtung liegt in der Erklärung, dass dem Anwender die bereitgestellten Möglichkeiten in einem Überblick dargelegt werden sollen und er nicht selbst herausfinden muss, wozu das Modul imstande ist. Bei dieser Kondition können vier Punkte erreicht werden.

Die zweite Bedingung bewegt sich im Rahmen der Tutorials. Durch ausführliche Anweisungen für die Benutzung diverser Funktionen wird dem Entwickler eine Wissensbasis vermittelt, die die Anwendung der Web-GIS Bibliothek im praktischen Bereich vereinfacht und unterstützt. Überdies können häufig gestellte Fragen beantwortet werden, um unnötigen

Kommunikationsaufwand zu vermeiden. Aufgrund der erwähnten Vorteile gut geführter Tutorials erhält diese Kondition eine maximal zu erreichende Punktzahl von drei.

Um sich einen besseren Überblick zu verschaffen, wie ein mögliches Geoinformationssystem mit dem ausgesuchten Modul kreiert werden kann, ist es in der Praxis von Vorteil, wenn dem Anwender bereits erstellte Beispielprojekte angeboten werden. Solche Projekte können als Strukturgrundlage für definitive Entwicklungen dienen. Dieses Kriterium kann höchstens zwei Punkte vergeben.

Zum Schluss werden bereits auf einem Server laufende Web-Beispiele betrachtet, die hinsichtlich der Vorstellung, wie ein fertiges System aussieht und eingesetzt wird, für den Anwender interessant sein können. Diese Bedingung wird im Vergleich zu den anderen aufgeführten Konditionen als am wenigsten wichtig erachtet. Deshalb kann hier maximal nur ein Punkt erreicht werden.

Folglich den zugewiesenen Berechnungsfaktoren kann ein GIS-Modul in der internen Bewertung eine Gesamtpunktzahl von 10 erreichen.

Bedingungen

Die erste Forderung verlangt von der ausgewählten GIS-Bibliothek eine oder mehrere Übersichtstabellen, auf denen die bereitgestellten Funktionen und Möglichkeiten aufgezeichnet sind. Falls die Quelle des Projekts eine derartige Form der Übersicht gewährleistet, erfüllt das Modul diese Forderung. Andernfalls wird kein Punkt vergeben. Als Quelle kann zum Beispiel die Website dienen, auf der das jeweilige Projekt hinterlegt ist.

Bei den angebotenen Tutorials der Module sind zwei wichtige Aspekte zu betrachten. Dazu gehören Instruktionen für die Erstellung einer Desktopanwendung sowie Instruktionen zur Entwicklung einer Webanwendung, da zwischen den beiden Bereichen unterschiedliche Programmcodes oder Elemente zum Aufbau eines Geoinformationssystems benötigt werden. Das Präsentieren eines Tutorials in den genannten Bereichen trägt jeweils zur Addition eines Punktes für das Modul bei. Diese Instruktion darf sich auf die Erstellung einer einfachen Karte auf der Mappe beschränken. Um die Maximalpunktzahl zu erreichen, müssen mindestens vier zusätzliche Tutorials, die weitere Funktionen beschreiben, aufgeführt sein. Die alleinige Zurverfügungstellung von Programmcodes wird auch als Instruktion gewertet.

Für die Bedingung, dass vorhandene Beispielprojekte eine Aufwertung der Module bedeuten, werden zwei Punkte vergeben. Damit eine Bibliothek beide Punkte erhält, muss mindestens ein Projekt im Internet auffindbar sein und für die Erstellung einer Webapplikation dienen. Die Beispiele müssen nicht zwangsläufig von derselben Quelle stammen, auf der das

Modul repräsentiert wird. Jedoch muss es sich bei der Anwendung auf das untersuchte Modul beschränken. Besteht beispielsweise die Projektdemonstration aus der Kombination zweier Module, wird nur ein Punkt verteilt.

Der letzte Punkt wird hinsichtlich der Web-Beispiele von der ausgewählten Bibliothek vergeben, insofern derartige vorhanden sind. Bedingung ist, dass die Webapplikation von einem erreichbaren Webserver gehostet wird und somit über das Internet aufgerufen werden kann.

3. Funktionsumfang

Grund

Das dritte Vergleichskriterium bezieht sich auf den Funktionsumfang. Damit sind die Funktionen gemeint, die das Modul dem Benutzer zur Verfügung stellt. Dieses Kriterium ist von grosser Bedeutung, da mit den angebotenen Funktionen das Geoinformationssystem erstellt wird. Aufgrund der Bedingung der vorhandenen Übersichtstabellen beim Vergleichskriterium «Dokumentation», die für diese Bewertung benötigt werden, wurde dieses Bewertungskriterium an dritter Stelle gewählt. Für die Schlussbewertung erhält es zwei Punkte.

Internes Bewertungssystem

Das System zur Durchführung der Modulvergleiche in diesem Bereich besteht aus zwei Einheiten: Erfüllt und nicht erfüllt. Das bedeutet, dass die Bibliothek den im Rahmen der kartenbezogenen Funktionsanforderungen an das Geoinformationssystem entweder nachkommt oder nicht.

Bedingungen

Die Auswertung für dieses Vergleichskriterium ist einfach. Als Möglichkeiten gelten sieben verschiedene Aspekte, die das GIS-Modul dem Anwender zur Verfügung stellen muss. Diese sieben Bedingungen stellen sich wie folgt zusammen:

Tabelle 3: Angeforderter Funktionsumfang

Bereich	Bedingung
Datenzugriff	Shapefile
	ECW
	JPG2000
Darstellung	Labels
	Geometrische Formen
Kartenlayer Management	Pan
	Zoom

Der erste Bereich in der Tabelle 3 bezieht sich auf die Nutzungsmöglichkeit der Dateien Shapefile, ECW und JPG2000, da für die RPGeol AG der Umgang mit diesen Dateitypen zum Arbeitsalltag gehört. Wie aus der Problemstellung zu entnehmen ist, soll das Geoinformationssystem des Weiteren imstande sein, geometrische Formen wie zum Beispiel Punkte, darzustellen. Solche Formen müssen mit passenden Textangaben gekennzeichnet werden. Als letzte Bedingungen sind die zwei Standardfunktionen aufgeführt, von denen der Benutzer bei der Anwendung eines geographischen Informationssystems Gebrauch machen kann, nämlich das Wechseln sowie das Zoomen der Kartenansicht.

Damit das Modul bei der Bewertung als «erfüllt» gekennzeichnet wird, müssen alle aufgeführten Funktionen dem Benutzer angeboten werden. Bei einer nicht erfüllten Bedingung gilt für die Gesamtauswertung der Status «nicht erfüllt».

4. Modularität des GIS-Moduls

Grund

Beim letzten Bewertungskriterium geht es um die Modularität der GIS-Bibliotheken. Angemessene Aufteilungen des Programmcodes in einzelne Arbeitsbereiche sollen dem Benutzer eine gute Übersicht sowie die Wiederverwendbarkeit der Programmcodes gewährleisten. Je strukturierter die einzelnen Bibliotheken aufgebaut sind, desto einfacher ist es für den Entwickler, sich bei den angebotenen Funktionen zurecht zu finden. Dieses Vergleichskriterium kann bei der Schlussbewertung einen Punkt vergeben.

Internes Bewertungssystem

Für das Messen der Modularität sind gleichermaßen wie beim Funktionsumfang die zwei Einheiten, erfüllt und nicht erfüllt, von Bedeutung. Damit das ausgewählte Modul den Status «erfüllt» erhält, müssen alle nachfolgenden Bedingungen eingehalten werden.

Bedingungen

Zur Zielerreichung müssen die Arbeitsbereiche der jeweiligen Bibliotheken in mindestens drei Teile gekapselt werden:

Der erste Teil beläuft sich auf die Standardbibliothek des angewendeten Moduls. Hier sollen die Standardfunktionen dem Benutzer zur Verfügung gestellt werden. Der Anwender kann beispielsweise direkt erkennen, welche Methoden für die Datenzugriffe eingesetzt werden oder welche Darstellungsmöglichkeiten er hat. Als Darstellungsmöglichkeiten gelten zum Beispiel die verschiedenen Arten der Kartenlayer.

Zweitens ist es für den Entwickler ein Vorteil, wenn die Bibliotheken, die zur Erstellung einer Webapplikation benötigt werden, von den Standardfunktionen getrennt werden. Damit ist die sachliche Trennung zwischen den benötigten Funktionen im Desktop- und Webbereich gewährleistet.

Als dritte und letzte Kondition soll das Modul dem Benutzer die GDAL-Bibliotheken (siehe Glossar) bereitstellen, die für das Erstellen von ECW- und JPG2000-Karten benötigt werden (Gdal.org, kein Datum). Grund für die Aufstellung dieser Bedingung ist, dass der Entwickler nicht selbst die verlangten Bibliotheken finden und in das Projekt einbinden muss.

Punkte für die Schlussbewertung

Angesichts der bei den Gewichtungen vergebenen Punkte in den Vergleichskriterien können bei der Schlussbewertung maximal zehn Punkte erreicht werden.

3.3 Wahl der Module

Die erstellte Marktübersicht gilt nun als Ausgangspunkt für die Wahl der Module, die für die Erstellung eines Geoinformationssystems verwendet werden sollen. Allerdings beziehen sich die Rahmenbedingungen für die Durchführung dieses Auswahlverfahrens nicht nur auf die .NET Technologie und die Möglichkeit, eine Webanwendung zu kreieren, sondern auch auf die Lizenzart sowie den Preis. Damit eine Bibliothek ausgewählt wird, muss es über eine Open Source Lizenz verfügen und darf nicht kostenpflichtig sein.

Anhand der definierten Bedingungen werden folgende vier GIS-Module bewertet:

Tabelle 4: Modulauswahl

Modul 1	Modul 2	Modul 3	Modul 4
Easy GIS .NET	DotSpatial	SharpMap	MapAround

Die Tabelle 4 zeigt, dass die Module von Cadcorp und ArcGIS durch die erweiterten Bedingungen für die Bewertungen nicht miteinbezogen werden, da beide weder kostenfrei sind (Cadcorp, kein Datum c) (ESRI Inc., kein Datum d) noch über eine entsprechende Lizenz (Cadcorp, 2016b) (ESRI Inc., kein Datum c) verfügen.

3.4 Modulvergleiche durchführen

In diesem Teil der Arbeit wird der detaillierte Modulvergleich durchgeführt. Diesbezüglich stehen bei jedem Vergleichskriterium die aus den Untersuchungen gewonnenen Resultate. Des Weiteren dienen die internen Bewertungssysteme als Zusammenfassungen der Erkenntnisse. Die Programmbibliotheken mit den jeweils höchsten Bewertungen werden dadurch ersichtlich.

1. Technologie

Programmiersprache

1. **EasyGIS .NET:** Wie im Name des Moduls bereits erwähnt wird, bewegt sich das Modul im .NET-Bereich und erlaubt die Erstellung eines Geoinformationssystems mit der Programmiersprache C# (Fletcher, kein Datum e). Bei einem zusätzlichen Beispielprojekt kommt die Skriptsprache Javascript zum Einsatz, was bei Entwickler zu Problemen führen kann, die nur im .NET Bereich Kenntnisse aufweisen (Fletcher, kein Datum b).
2. **DotSpatial:** Ein von DotSpatial angebotenes Projekt für die Erstellung eines Web-GIS zeigt, dass das Modul darauf Wert legt, die im Anforderungsprofil definierte Technologie zu berücksichtigen (DotSpatial, 2016a).
3. **SharpMap:** Von SharpMap gibt es ein strukturiert aufgebautes Beispielprojekt, wo die Anzeige einer einfachen Shapefile-Karte auf ASP Formen sowie der Programmiersprache C# entwickelt wurde (Nielsen, 2005). Dementsprechend hält diese Bibliothek die gestellte Bedingung ein.

4. **MapAround:** Wie aus dem zur Verfügung gestellten Programmcode ersichtlich ist, wird für die Erstellung eines GIS die Skriptsprache Javascript benötigt. Grund dafür sind die selbsterstellten Webserversteuerelemente, die auf Ajax (siehe Glossar) basieren. Durch den Einsatz einer anderen Technologie erfüllt das Modul von MapAround die Anforderungen nicht (MapAround, 2013c).

ASP Steuerelemente

1. **EasyGIS .NET:** Für die Erstellung einer Karte muss beim EasyGIS-Modul das selbsterstellte Webserversteuerelement «SFMap» angewendet werden, das als Kartengrundlage dient (Fletcher, kein Datum d). Aus diesem Grund gilt die Bedingung als nicht erfüllt.
2. **DotSpatial:** DotSpatial stellt den eigens entwickelten Assembly «DotSpatial.WebControls» für den Aufbau einer Webapplikation zur Verfügung, der integrierte Steuerelemente beinhaltet (DotSpatial, 2016c). In diesem Fall werden die Anforderungen nicht berücksichtigt (DotSpatial, 2016b).
3. **SharpMap:** Bei Sharpmap können die Karten auf einem «ImageButton» erstellt werden (SharpMap, 2012d). Da dieses Element von Microsoft vordefiniert ist (Microsoft Corporation, 2016), wird die Kondition vom Modul eingehalten.
4. **MapAround:** Das Modul stellt dem Anwender mehrere technische Arten für die Erstellung eines GIS zur Verfügung (MapAround, kein Datum a). Deshalb bietet MapAround diverse Steuerelemente an. Eines davon ist das sogenannte «WebMapControl»-Element, das bei Webformen als Kartengrundlage dient (MapAround, 2013a). Ein ähnliches wurde für Windows Forms Anwendungen entwickelt, wo es ebenfalls als Grundlage für die Erstellung der Karten dient (MapAround, 2013b). Durch den zwangsläufigen Einsatz dieser Elemente kommt das Modul der gestellten Anforderung nicht nach.

Zusatzdateien

1. **EasyGIS .NET:** Die Bibliothek verlangt für die korrekte Anwendung des Serversteuerelements «SFMap» das Referenzieren auf eine von EasyGIS entwickelte Projektdatei (Fletcher, kein Datum f). Diese Datei kann mit der angebotenen Desktopanwendung des Moduls erstellt werden und beinhaltet Eigenschaften eines Shapefiles (Fletcher, kein Datum d).
2. **DotSpatial:** Es werden keine Zusatzdateien benötigt (DotSpatial, 2016a).

3. **SharpMap**: Es werden keine Zusatzdateien benötigt (Nielsen, 2005).
4. **MapAround**: Es werden keine Zusatzdateien benötigt (MapAround, kein Datum d). Allerdings stellt MapAround die weitere Möglichkeit zur Verfügung, einen sogenannten «Workspace» als Kartengrundlage einzusetzen, in dem Informationen über bestimmte Karten vordefiniert sind (MapAround, 2013d).

Die Detailergebnisse lassen sich in der Tabelle 5 wie folgt darstellen:

Tabelle 5: Internes Bewertungssystem «Technologie»

Module	Bedingungen			Total	SB 4 Pkt.
	Programmiersprache	ASP Steuerelemente	Zusatzdateien		
1. EasyGIS .NET	+	-	-	+	
2. DotSpatial	+	-	+	++	
3. SharpMap	+	+	+	+++	4
4. MapAround	-	-	+	+	

2. Dokumentation

Funktionsübersicht

1. **EasyGIS .NET**: Das Modul stellt die benötigten Übersichtstabellen zur Verfügung (Fletcher, kein Datum h).
2. **DotSpatial**: Das Modul stellt die benötigten Übersichtstabellen zur Verfügung (Ames, 2013).
3. **SharpMap**: Das Modul stellt die benötigten Übersichtstabellen zur Verfügung (SharpMap, 2008).
4. **MapAround**: Das Modul stellt die benötigten Übersichtstabellen zur Verfügung (MapAround, kein Datum a).

Tutorials

1. **EasyGIS .NET**: Das Modul stellt den Anforderungen entsprechend genügend Tutorials zur Verfügung (Fletcher, kein Datum c).
2. **DotSpatial**: Das Modul stellt den Anforderungen entsprechend genügend Tutorials zur Verfügung (DotSpatial, 2017c). Für die Darlegung der Instruktionen bezüglich des Web-Bereichs dient ein Beispielprojekt (DotSpatial, 2016a).

3. **SharpMap**: Das Modul stellt den Anforderungen entsprechend genügend Tutorials zur Verfügung (SharpMap, 2012f).
4. **MapAround**: Für die Erstellung einer Karte im Bereich der Desktop- (MapAround, 2013e) sowie Webanwendung (MapAround, 2013c) werden Tutorials oder entsprechende Programmcodes angeboten. Allerdings existieren zu wenig weitere Instruktionen, um alle Bedingungen in diesem Bereich zu erfüllen (MapAround, kein Datum d).

Beispielprojekte

1. **EasyGIS .NET**: Von diesem Modul existiert ein Beispielprojekt zur Erstellung einer Webanwendung (Fletcher, kein Datum c).
2. **DotSpatial**: Von diesem Modul existiert ein Beispielprojekt zur Erstellung einer Webanwendung (DotSpatial, 2016a).
3. **SharpMap**: Von diesem Modul existiert ein Beispielprojekt zur Erstellung einer Webanwendung (SharpMap, 2017a).
4. **MapAround**: Von diesem Modul existiert ein Beispielprojekt zur Erstellung einer Webanwendung, allerdings aus einer Kombination mit dem Modul von Leaflet (siehe Glossar). Aus diesem Grund wird die gestellte Anforderung nicht erfüllt (MapAround, 2013c).

Web-Beispiele

1. **EasyGIS .NET**: Es besteht ein laufendes Web-Beispiel von diesem Modul (Fletcher, kein Datum g).
2. **DotSpatial**: Es sind keine Web-Beispiele im Internet auffindbar.
3. **SharpMap**: Es sind keine Web-Beispiele im Internet auffindbar.
4. **MapAround**: Es besteht ein laufendes Web-Beispiel von diesem Modul (MapAround, kein Datum c).

Die Detailergebnisse lassen sich in der Tabelle 6 wie folgt darstellen:

Tabelle 6: Internes Bewertungssystem «Dokumentation»

Module	Bedingungen				Total	SB 3 Pkt.
	Funktionsübersicht (max. 4 Punkte)	Tutorials (max. 3 Punkte)	Beispielprojekte (max. 2 Punkte)	Web-Beispiele (max. 1 Punkt)		
1. EasyGIS .NET	4	3	2	1	10	3
2. DotSpatial	4	3	2	0	9	
3. SharpMap	4	3	2	0	9	
4. MapAround	4	2	1	1	8	

3. Funktionsumfang

1. **EasyGIS .NET:** Beim ersten Modul wird anhand der gebotenen Übersichtstabellen ersichtlich, dass die Erstellung von ECW- und JPG2000-Karten nicht möglich ist. Deshalb gilt die Bedingung für dieses Modul als nicht erfüllt (Fletcher, kein Datum h).
2. **DotSpatial:** Die Übersichtstabellen des Moduls bestätigen die Erfüllung der gestellten Anforderungen (Ames, 2013).
3. **SharpMap:** Die Übersichtstabellen des Moduls bestätigen die Erfüllung der gestellten Anforderungen (SharpMap, 2008).
4. **MapAround:** Die Übersichtstabellen des Moduls bestätigen die Erfüllung der gestellten Anforderungen (MapAround, kein Datum a).

Die Detailergebnisse lassen sich in der Tabelle 7 wie folgt darstellen:

Tabelle 7: Internes Bewertungssystem «Funktionsumfang»

Module	Bedingungen							SB 3 Pkt.
	Datenzugriff			Darstellung		Kartenlayer Management		
	Shapefile	ECW	JPG2000	Labels	Geometrische Formen	Pan	Zoom	
1. EasyGIS .NET	X			X	X	X	X	
2. DotSpatial	X	X	X	X	X	X	X	2
3. SharpMap	X	X	X	X	X	X	X	2
4. MapAround	X	X	X	X	X	X	X	2

4. Modularität des GIS-Moduls

1. **EasyGIS .NET:** EasyGIS .NET teilt die angebotenen Bibliotheken sowohl in einen Bereich für die Standardfunktionen und einen Bereich für die Entwicklung einer Webanwendung auf. Allerdings kann das Modul keine ECW-Karten erstellen, weshalb die benötigten Bibliotheken hierfür fehlen (Fletcher, kein Datum h). Die Bedingung gilt als nicht erfüllt.
2. **DotSpatial:** DotSpatial hat die Bibliotheken korrekt in die angeforderten Bereiche unterteilt (DotSpatial, 2017b). Allerdings müssen die für die Webanwendung benötigten Programm-Bibliotheken separat heruntergeladen werden (DotSpatial, 2016c), da die Entwickler der Ansicht sind, dass sie nicht repräsentationsfähig sind und Ausarbeitungsarbeiten erfordern (DotSpatial, 2017a).
3. **SharpMap:** Das Modul erfüllt die verlangte Datenkapselung (SharpMap, 2014).
4. **MapAround:** Das Modul erfüllt die verlangte Datenkapselung (MapAround, 2017).

Die Detailergebnisse lassen sich in der Tabelle 8 wie folgt darstellen:

Tabelle 8: Internes Bewertungssystem «Modularität des GIS-Moduls»

Module	Bedingungen			SB 1 Pkt.
	Standardbibliotheken	Web-Bibliotheken	GDAL-Bibliotheken	
1. EasyGIS .NET	X	X		
2. DotSpatial	X	X	X	1
3. SharpMap	X	X	X	1
4. MapAround	X	X	X	1

3.5 Bewertung der Module

Die Schlussbewertung wird mit der Tabelle 9 repräsentiert:

Tabelle 9: Schlussbewertung Evaluation der Module

Module	Vergleichskriterien				Total (Pkt)
	Technologie (4 Punkte)	Dokumentation (3 Punkte)	Funktionsumfang (2 Punkte)	Modularität (1 Punkt)	
1. EasyGIS .NET	[+]	10			3
2. DotSpatial	[++]	8	X	X	3
3. SharpMap	[+++]	9	X	X	7
4. MapAround	[+]	8	X	X	3

Die Resultate aus der Übersicht zeigen, dass SharpMap in Bezug auf die durchgeführten Vergleiche als das zweckmässigste Modul hervorgeht. Zudem erfüllen drei der vier Bibliotheken die Bewertungskriterien «Funktionsumfang» sowie «Modularität», was die OpenSource Projekte in ein positives Licht rückt. Erstaunlicherweise ist die Dokumentation von EasyGIS .NET die ausführlichste, obwohl das Modul am wenigsten Funktionen zur Verfügung stellt.

3.6 Wahl des bestgeeignetsten Moduls

Anhand der Vergleichsresultate aus der Tabelle 9 wird das GIS-Modul von SharpMap im weiteren Verlauf der Arbeit für den Aufbau sowie die Entwicklung eines geographischen Informationssystems mit anschliessender Integration in die IT-Infrastruktur der Rovina & Partner AG eingesetzt.

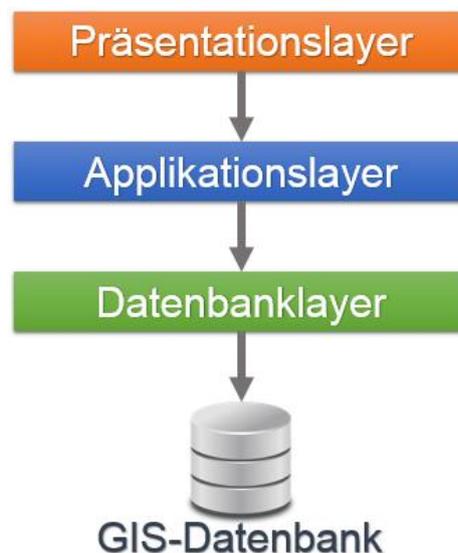
4. Web-Geoinformationssystem

In diesem Kapitel wird aufgezeigt, wie das geographische Informationssystem entwickelt und anschliessend in die IT-Infrastruktur der RPGeol AG integriert wurde. Hier wird ersichtlich, welche Entwicklungsschritte zur Erfüllung des Anforderungsprofils nötig waren, das in der Problemstellung definiert wurde. Für den Aufbau des Systems wurde eine Drei-Schichten-Architektur angewendet.

4.1 Drei-Schichten-Architektur

Die ausgewählte Drei-Schichten-Architektur bezieht sich auf drei verschiedene Schichten (Layer). Dazu gehören der Präsentationslayer, der Applikations- sowie der Datenbanklayer. Jede Schicht wird für einen bestimmten Zweck eingesetzt und ist wichtig, damit das gesamte System funktionieren kann. Nebst den drei verschiedenen Layern wurde ebenfalls eine Programmbibliothek zur Definition der einzelnen Klassen und Objekte erstellt. Diese Bibliothek wird in jeder Schicht benötigt. Die Abbildung 8 zeigt, wie die ausgewählte Architektur aufgebaut ist:

Abbildung 8: Drei-Schichten-Architektur Web-GIS



Um die Zusammenhänge zwischen den einzelnen Schichten übersichtlich darzustellen, wird in diesem Kapitel jede Schicht im Detail beschrieben. Zuerst jedoch wird die Programmbibliothek mit den Datenmodellen betrachtet, da sich die darin enthaltenen Klassen und Objekte auf die Elemente beziehen, die in den verschiedenen Schichten beschrieben werden. Demzufolge liegt die Präsentationsschicht im Vordergrund. Hinsichtlich dessen wird ein Überblick gewährleistet, wie das geographische Informationssystem aussieht und welche

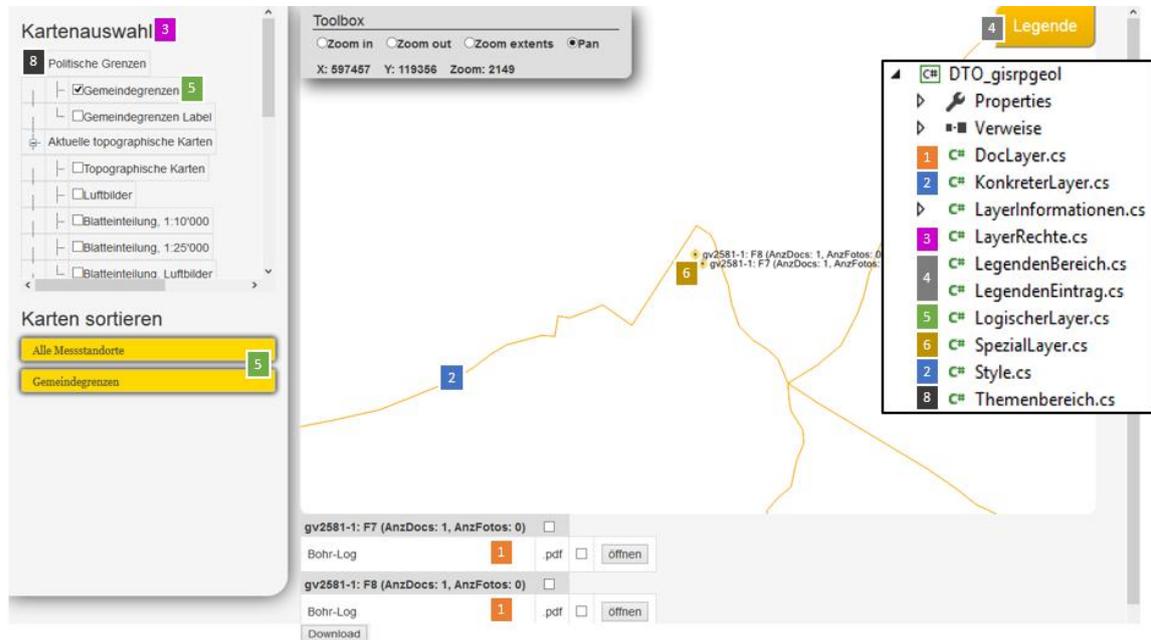
Funktionen es beinhaltet. Anschliessend liegt der Fokus auf der Applikationsschicht, die für die Datentransformation zuständig ist. Zum Schluss wird erklärt, wie die Datenbank mit dem entsprechenden Layer aufgebaut ist.

Wichtig zu beachten in der Architektur sind die einzelnen Verbindungen zwischen den Schichten. So kennt der Präsentationslayer nur die Applikationsschicht, dieser Schicht wiederum ist nur die Datenbankschicht bekannt. Jedoch können alle Layer auf die Programmbibliothek mit den Modellen zugreifen. Der Datenbanklayer dient als direkte Verbindung zur physischen Datenbank.

4.1.1 Datenmodelle und Objekte

In dieser Programmbibliothek sind alle Klassen und Objekte vorhanden, die für die Entwicklung des GIS programmiertechnisch benötigt werden. Die Abbildung 9 zeigt in einer Übersicht die Modelle, die in dieser Bibliothek erstellt wurden. Des Weiteren wird hinsichtlich der Präsentationsschicht ein erster Überblick gewährleistet, wo genau sich diese Klassen und Objekte im GIS befinden. Wichtig zu erwähnen ist, dass diese Klassen keine Methoden mit speziellen Funktionalitäten beinhalten. Sie dienen nur zur Datendefinition und Datenhaltung.

Abbildung 9: Datenmodelle und Objekte



Die nummerierten Klassen und Objekte werden in den nachfolgenden Kapiteln genauer beschrieben, mit Ausnahme der Klasse «LayerInformationen». Sie dient dazu, alle in der GIS-Datenbank gespeicherten Informationen festzuhalten, damit sie den Benutzern während der Programmlaufzeit permanent zur Verfügung stehen. So greifen alle Anwender auf die

Datenmodelle über diese Klasse zu. In der Abbildung 10 ist ersichtlich, wie die Daten in der Klasse implementiert werden.

Abbildung 10: Modell LayerInformationen

```

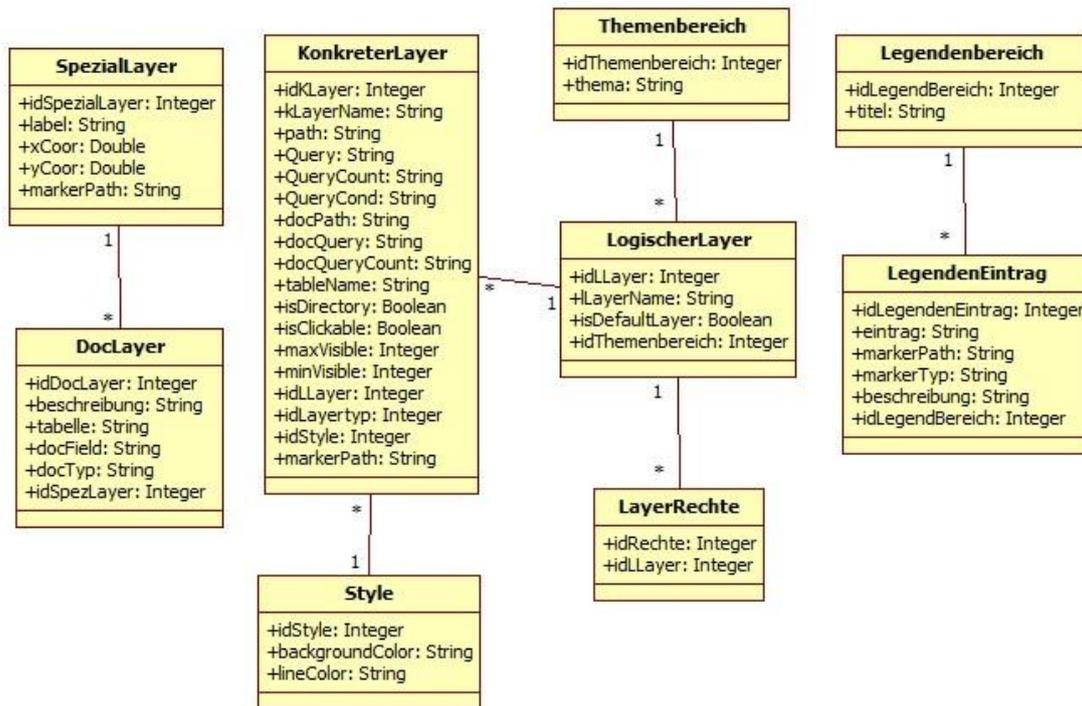
16 Verweise
public class LayerInformationen
{
    //Arrays mit allen Startinformationen über Layer, Rechten, Bereiche, Styles
    2 Verweise
    public static Themenbereich[] tBereichArray { get; set; }
    3 Verweise
    public static LogischerLayer[] lLayerArray { get; set; }
    3 Verweise
    public static KonkreterLayer[] kLayerArray { get; set; }
    2 Verweise
    public static LayerRechte[] layerRechteArray { get; set; }
    2 Verweise
    public static Style[] styleArray { get; set; }
    2 Verweise
    public static LegendenBereich[] lBereichArray { get; set; }
    2 Verweise
    public static LegendenEintrag[] lEintragArray { get; set; }
}

```

Alle Informationen werden zuerst in die passenden Modelle geladen. Dieser Schritt erfolgt auf der Datenbankschicht. Im Anschluss dienen statische Arrays für die Speicherung dieser Modelle. Der Datenzugriff auf die Datenbank für die Initialisierung dieser Listen erfolgt nur ein einziges Mal, nämlich direkt nach dem Start des Webservers.

Mit der Abbildung 11 werden die aufgezeigten Datenmodelle nochmals in Form eines UML-Diagrammes dargestellt.

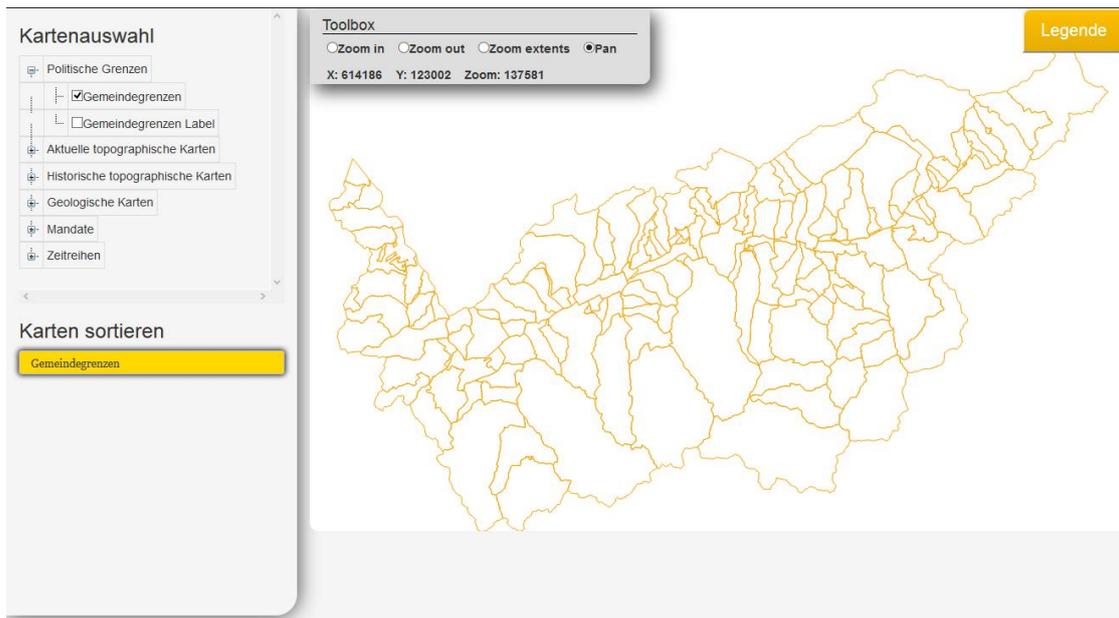
Abbildung 11: UML-Diagramm der Datenmodelle



4.1.2 Präsentationslayer

Bei der ersten Schicht, dem Präsentationslayer, handelt es sich um die visuelle Repräsentation des Geoinformationssystems. Sie stellt somit die graphische Benutzeroberfläche dar. Des Weiteren ist sie dafür verantwortlich, die für den Anwender verfügbaren Funktionen zu veranschaulichen. Da alle Interaktionen zwischen dem Benutzer und dem System über die graphische Oberfläche stattfinden, ist dieser Layer von grosser Wichtigkeit.

Abbildung 12: Web-GIS Benutzeroberfläche



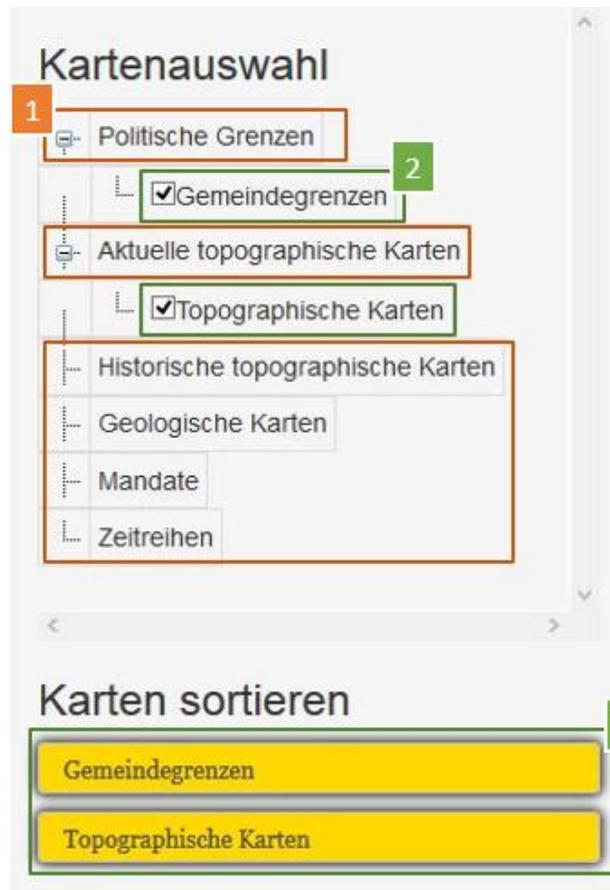
Die Abbildung 12 zeigt, wie das GIS dem Endbenutzer angezeigt wird. Darüber hinaus werden alle Arbeiten und Interaktionen mit dem System über diese graphische Oberfläche durchgeführt. Für die Erstellung dieser Oberfläche wurden dynamische Web-Formen (ASP) eingesetzt.

In den nachfolgenden Schritten des Kapitels werden die im Geoinformationssystem enthaltenen Elemente genauer beschrieben und die Funktionen erklärt, die dem Anwender über diese Oberfläche zur Verfügung stehen. Zuerst bezieht sich die Situation auf die Grundanwendung, hinterher auf spezielle Anwendungsbereiche.

Zusätzlich werden jeweils die dazu benötigten Implementierungen mit den entsprechenden Programmcodes gezeigt. Um das Ganze übersichtlich darzustellen, werden grosse Methoden in einem Pseudocode vereinfacht. Der vollständige Programmcode wird für diese im Anhang beigelegt. Zudem sind alle Methoden in der Programmiersprache C# geschrieben. Falls eine andere Sprache angewendet wurde, wird dies an der entsprechenden Stelle erwähnt.

Elemente und Funktionen für die Grundanwendung

Abbildung 13: Kartenverwaltung



In der Abbildung 13 ist der Bereich der Kartenverwaltung aufgeführt. Dieser Abschnitt legt den Fokus auf zwei verschiedene Elemente sowie zwei verschiedenen Funktionen.

Die Elemente mit der Nummer Eins sind sogenannte «Themenbereiche». Sie repräsentieren jeweils ein Thema, das zu den darunterliegenden Karten passt.

Jedes Thema beinhaltet dementsprechend eine oder mehrere «logische Layer». Damit sind die Karten für die Anzeige gemeint, die von der RPGeol AG dem Anwender zur Verfügung gestellt werden. Sie sind mit der Nummer Zwei in der Abbildung versehen. Es werden dem Benutzer für die Auswahl nur diejenigen Karten bereitgestellt, für die er entsprechende Rechte besitzt. Deshalb sind in dem aufgeführten Beispiel nur Karten für zwei Themenbereiche verfügbar. Die Benutzerrechte werden im Datenbanklayer genauer beschrieben.

Überdies ist wichtig zu erwähnen, dass die «logischen Layer» die Karten nur beschreiben. Die Objekte, die anschliessend für die tatsächliche Erstellung der Karten verantwortlich sind, heissen «konkrete Layer». Sie beinhalten spezielle Eigenschaften wie Pfadangaben auf

Kartendateien, SQL-Abfragen oder Objekte (Style-Objekt), die zum Designen der Karten wichtig sind. Zudem werden sie dem Benutzer nicht angezeigt, da sie im Hintergrund eingesetzt werden. Jeder «logische Layer» kann einen oder mehrere «konkrete Layer» beinhalten.

Hinsichtlich den Funktionen im Bereich der Kartenverwaltung bezieht sich eine davon auf das Auswählen der einzelnen Karten. Dazu muss der Benutzer lediglich die Checkbox aktivieren, die links neben der Beschreibung des «logischen Layer» angezeigt wird. Im Anschluss können die ausgewählten Karten per Drag and Drop sortiert werden, was die zweite Funktion in diesem Bereich darstellt. Anhand der erstellten Reihenfolge werden die Karten geladen, wobei der erste Layer in der Sortierung als oberste Karte auf der Map angezeigt wird.

Zur Erstellung der Kartenauswahl wurde das Steuerelement «TreeView» eingesetzt. Die Abbildung 14 zeigt den Pseudocode, der im Hintergrund des Programmes für die Erstellung des Baumes verantwortlich ist.

Abbildung 14: Methode initializeTree (Kartenauswahl)

```

1  initializeTree
2  {
3      foreach Themenbereich in alleThemenbereiche
4      {
5          Erstelle Treenode für Themenbereich
6          Treenode Themenbereich auf Baum setzen
7          foreach (LogischerLayer in alleLogischerLayer)
8          {
9              if Benutzerrechte vorhanden für LogischerLayer
10             && LogischerLayer gehört zu Themenbereich
11             {
12                 Erstelle Treenode mit Checkbox für LogischerLayer
13
14                 if LogischerLayer ist klickbar
15                 Klickbild auf Treenode LogischerLayer setzen
16
17                 if LogischerLayer ist per Default aktiviert
18                 LogischerLayer auf sortierbare Liste setzen
19
20                 Treenode LogischerLayer auf Baum unter Treenode Themenbereich setzen
21             }
22         } //end-foreach LogischerLayer
23     } //end-foreach Themenbereich
24 }

```

Mit diesem Programmcode wird bei der Initialisierung des Systems der Baum erstellt. Dadurch wird dem Anwender die dazugehörige Kartenauswahl ermöglicht. Der ausführliche Code ist im «Anhang I: Methode initializeTree» aufgeführt.

Wie bereits beschrieben wurde, beinhaltet ein «Themenbereich» einen oder mehrere «logische Layer». Dementsprechend werden zwei ineinander geschachtelte Foreach-

Schleifen eingesetzt, um diese Objekte miteinander zu verbinden und sie in der richtigen Reihenfolge auf den Baum zu setzen. Um die Einträge auf die «TreeView» zu setzen, werden «TreeNode»-Objekte benötigt. Sie erhalten einen Namen und eine ID, die jeweils von einem «Themenbereich» oder einem «logischen Layer» stammen.

Zudem werden gemäss der Methode per Default aktivierte Layer direkt auf die Sortierliste geladen und klickbare mit einem entsprechenden Symbol gekennzeichnet. «Klickbare Layer» werden zu einem späteren Zeitpunkt in diesem Kapitel behandelt, die Sortierliste wird im Anschluss erklärt.

Zur Überprüfung der Benutzerrechte dient eine spezifische Methode. Sie gewährleistet, dass dem Anwender nur die für ihn bestimmten «logischen Layer» angezeigt werden. Sobald diese Bedingung erfüllt ist und der «logische Layer» zum aktuellen Thema gehört, wird der «TreeNode» für ihn erstellt. Bei den «Themenbereichen» ist keine Überprüfung notwendig, da nur die «logischen Layer» Rechte beinhalten.

Die Methode, in der die Rechte überprüft werden, wird mit der Abbildung 15 gezeigt.

Abbildung 15: Methode *isErlaubt* (Kartenauswahl)

```

3 Verweise
private bool isErlaubt(int idLLayer)
{
    foreach (LayerRechte lRechte in LayerInformationen.layerRechteArray)
    {
        if (lRechte.idLLayer == idLLayer && lRechte.idRechte == userRechte) return true;
    }
    return false;
}

```

Als Parameter erhält diese Methode die ID vom zu überprüfenden «logischen Layer». Zur Untersuchung dienen im Anschluss die «LayerRechte»-Objekte. Dieses Objekt enthält immer zwei Schlüssel, nämlich eine «logische Layer»-ID sowie eine ID für den «Benutzertypen». Zum Beispiel kann die Nummer Eins einen öffentlichen Benutzer repräsentieren und die Nummer Zwei eine spezifische Unternehmung.

Die Variable «userRechte» beinhaltet ebenfalls einen «Benutzertypen». Der Wert dieser Variable stammt aus einer «Session» von einem Unternehmensportal der RPGeol AG. Diese «Session» wird nach dem Login in das Portal dem Benutzer zugewiesen und im GIS wiederverwendet.

Im Anschluss werden die Werte vom «LayerRechte»-Objekt mit der übergebenen «logischen Layer»-ID und der «userRechte» Variable verglichen. Falls also der Anwender die

Rechte für den angegebenen Layer hat, wird der Wert «true» zurückgegeben, andernfalls «false».

Für die Liste der Kartensortierung wurde ein «Placeholder»-Steuerelement eingesetzt, auf dem die aktivierten «logischen Layer» in Form einer «HTML-Liste» dynamisch angezeigt werden können. Nach dem Applikationsstart werden gemäss der Abbildung 14 zuerst alle per Default aktivierten Layer direkt auf diese Liste geladen. Der Programmcode dafür sieht wie folgt aus.

Abbildung 16: Methode addSortableLayer (Kartensortierung)

```
private void addSortableLayer(LogischerLayer lLayer)
{
    //Neues listen-html element erstellen
    HtmlGenericControl li = new HtmlGenericControl("li");
    li.Attributes.Add("class", "ui-state-default");
    li.Attributes.Add("id", lLayer.idLLayer.ToString() + "SL");
    li.Attributes.Add("value", lLayer.idLLayer.ToString());
    li.Controls.Add(new LiteralControl(lLayer.lLayerName.ToString()));

    sortableList.Controls.Add(li);
}
```

Entsprechend der Abbildung 16 wird dieser Methode ein per Default aktiver «logischer Layer» übergeben. Anhand den in diesem Objekt vorhandenen Informationen wird ein Listeneintrag für die Sortierliste kreiert. Für diesen Vorgang werden «HtmlGenericControl»-Objekte benutzt. Die Daten für den Eintrag sind die ID sowie der Name des «logischen Layers».

Wie bereits erwähnt wurde, kann der Benutzer während der Programmlaufzeit die Sortierliste dynamisch vergrössern oder verkleinern, indem er bei der Kartenauswahl «logische Layer» aktiviert oder deaktiviert. In der Abbildung 17 wird nun der Pseudocode für das Klickevent gezeigt, nachdem ein «logischer Layer» mit der Checkbox ausgewählt wird. Der vollständige Code ist im «Anhang II: Methode checkLayer» verfügbar. Wichtig zu erwähnen ist, dass dieses Event in der Programmiersprache Javascript geschrieben wurde, damit kein «PostBack» nach jedem Klickevent an den Server geschickt werden muss.

Abbildung 17: Methode checkLayer (Kartensortierung)

```

1  checkLayer
2      {
3          Informationen aus angeklickter Checkbox lesen
4          if Checkbox aktiviert {
5              Listenobjekt für LogischerLayer mit Daten von Checkbox erstellen
6              Listenobjekt auf Placeholder setzen
7          }
8          else {
9              if mehr als ein Logischer Layer aktiviert
10                 deaktiviertes Listenobjekt von Placeholder löschen
11             }
12             else {
13                 Checkbox aktivieren
14             }
15         }
16     }

```

Zuerst werden alle benötigten Informationen vom angeklickten «logischen Layer» aus den automatisch kreierte HTML-Elementen der «TreeView» geladen. Dazu gehören die ID des Layers sowie dessen Name. Anschliessend wird, falls die Checkbox durch den Klick aktiviert wurde, mittels dieser Daten ein neuer Listeneintrag erstellt, der auf die HTML-Liste des «Placeholders» gesetzt wird. Bei einer Deaktivierung der Checkbox wird der angeklickte «logische Layer» aus der HTML-Liste entfernt. Der letzte Eintrag in der Liste kann nicht entfernt werden, da immer mindestens ein Layer ausgewählt sein muss.

Die Methoden, die für die Sortierung dieser erstellen Listenobjekte eingesetzt wurden, stammen vom Framework «jQueryUI». Es stellt dem Anwender vordefinierte Programmibliotheken basierend auf JQuery zur Verfügung, mit denen interaktive Funktionen, wie beispielsweise das Sortieren einer Liste, ausgeführt werden können. Da diese Methoden nicht selbst erstellt wurden, werden sie in dieser Arbeit nicht beschrieben (jQueryUI, kein Datum).

Abbildung 18: Einbinden JQueryUI (Kartensortierung)

```

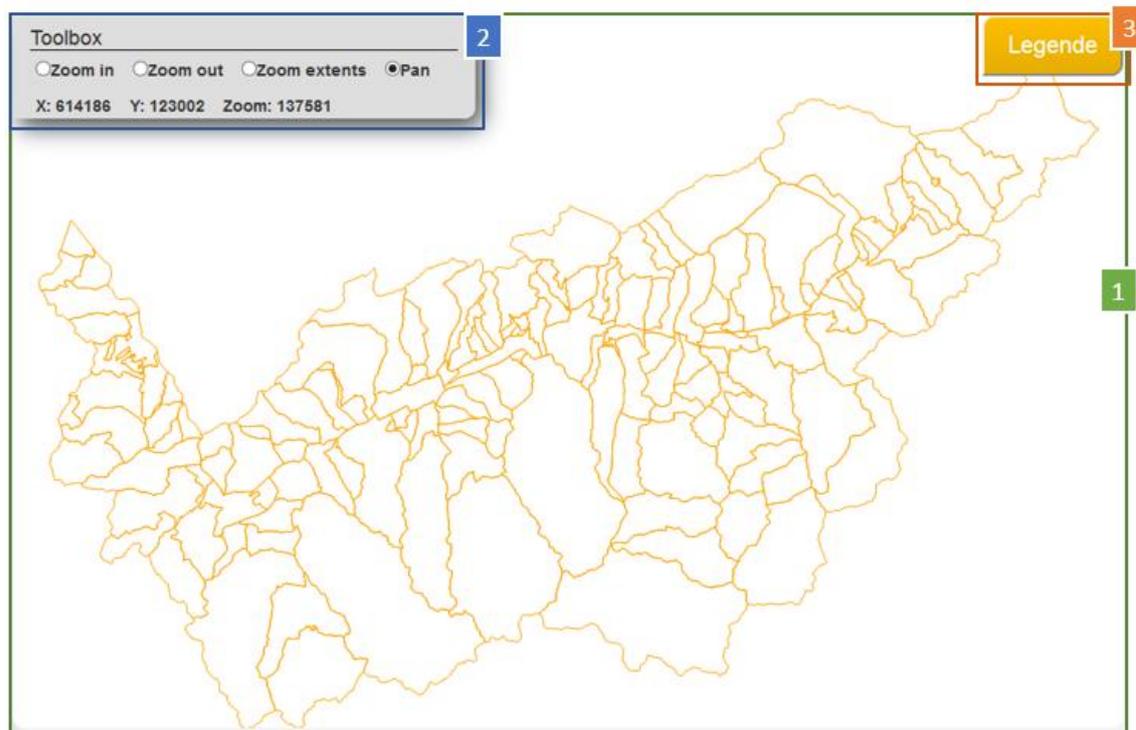
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script>
  //sortierfunktion für Layersortierung
  $(function () {
    $("#sortable").sortable();
    $("#sortable").disableSelection();
  });
</script>

<div style=" height: 350px;">
  <h3>Karten sortieren</h3>
  <ul id="sortable">
    <asp:Placeholder ID="sortableList" runat="server" />
  </ul>
</div>

```

Zur Aktivierung der Funktion für die Sortierung musste lediglich das «UL»-Element der Sortierliste mit der ID «sortable» gekennzeichnet und die Methoden gemäss Abbildung 18 eingebunden werden.

Abbildung 19: Bereich für die Kartenanzeige



Für die Repräsentation des Bereichs der Kartenanzeige dient die Abbildung 19. Hier stehen drei Elemente mit verschiedenen Funktionen im Vordergrund.

Das Element mit der Nummer Eins bezieht sich auf die Map, die als Kartengrundlage dient. Auf ihr werden diejenigen Karten abgebildet, die im Bereich der Kartenverwaltung aktiviert wurden. Zur Initialisierung der Map gehört die Methode in der Abbildung 20. Sie wird nach jedem Klick auf die Map ausgeführt. Der konkrete Programmcode ist im «Anhang III: Methode InitializeMap» verfügbar.

Abbildung 20: Methode InitializeMap (Kartenanzeige)

```

1 InitializeMap
2     {
3         Map-Objekt erstellen
4
5         foreach LogischerLayer in aktivierteLogischeLayer
6         {
7             foreach KonkreterLayer in alleKonkretenLayer
8             {
9                 if KonkreterLayer gehört zu LogischerLayer
10                && Benutzerrechte vorhanden für LogischerLayer
11                {
12                    Kartenschicht mithilfe KonkreterLayer auf Map setzen
13                }
14            }
15        }
16        return Map-Objekt
17    }

```

Wichtig zu beachten ist, dass diese Methode das vom SharpMap-Modul zur Verfügung gestellte Objekt «Map» initialisiert. Dieses Objekt wird, nachdem auf ihm alle bei der Kartenauswahl aktivierten Karten hinzugefügt wurden, in ein Bild umgewandelt und auf das Steuerelement «ImageButton» geladen. Der «ImageButton» ist das Element, das dem Endbenutzer schliesslich angezeigt wird. Für diesen Vorgang dient die Methode «CreateMap», die in der Abbildung 21 gezeigt wird.

Abbildung 21: Methode CreateMap (Kartenanzeige)

```

private void CreateMap()
{
    //Sicherheitscheck ob Layer hinzugefügt wurden -> sonst nicht erstellen weil nullObjekt
    if (this.myMap.Layers.Count != 0)
    {
        System.Drawing.Image img = myMap.GetMap();

        string imgID = SharpMap.Web.Caching.InsertIntoCache(1, img);
        imgMap.ImageUrl = "getmap.aspx?ID=" + HttpUtility.UrlEncode(imgID);
    }
}

```

Das Objekt «myMap» repräsentiert die initialisierte Map. Zuerst wird sie mit der Methode «getMap» in ein Bild konvertiert und im Anschluss mithilfe der Methode «insertIntoCache» von SharpMap in den Zwischenspeicher geladen. Der HttpHandler vom Web-GIS Modul lädt zuletzt das gespeicherte Bild auf den ImageButton «imgMap» (SharpMap, 2012d).

Die Logik für die Erstellung der Map ist ähnlich wie die für die Initialisierung des Baumes in der Kartenauswahl. So gehört ein «konkreter Layer» immer zu einem «logischen Layer». Deshalb werden wiederum zwei Foreach-Schleifen für das Durchlaufen dieser Objekte benötigt. In diesem Fall jedoch werden nur diejenigen «logischen Layer» berücksichtigt, die bei der Kartenverwaltung aktiviert wurden.

Nach der Überprüfung, ob der Benutzer die Rechte für den entsprechenden Layer besitzt, wird eine neue Kartenschicht erstellt und auf die Map geladen. Die dazu benötigten Methoden und Implementierungsschritte sowie die Bedeutung einer Kartenschicht werden im Kapitel der Applikationsschicht genauer beschrieben.

Im zweiten Element der Abbildung 19 ist die Toolbox enthalten, die für das Kartenlayer-Management verantwortlich ist. Die verfügbaren Funktionen können über entsprechende Radio-Buttons aktiviert werden und dienen zur Veränderung der Kartenansicht. Per Mausklick auf die Map wird die ausgewählte Funktion ausgeführt. Darüber hinaus werden die jeweils angeklickten Koordinaten sowie das aktuelle Zoom in der Toolbox angezeigt. Mit der Abbildung 22 wird veranschaulicht, wie das Programm im Hintergrund die Funktionen ausführt.

Abbildung 22: Methode setMapFunctions (Kartenanzeige)

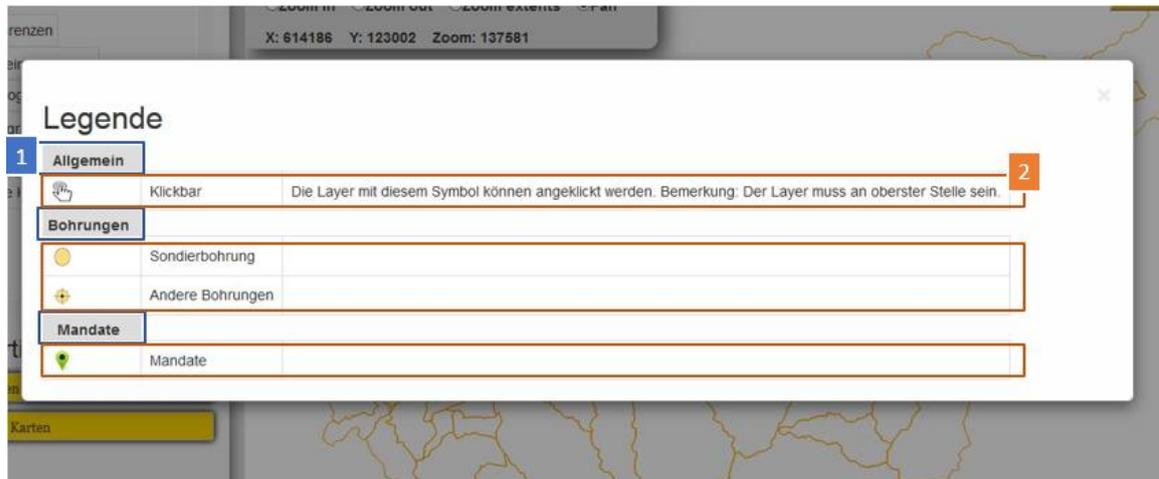
```
private void setMapFunctions()
{
    switch (this.rblMapTools.SelectedValue)
    {
        case "0": //Zoom in
            this.myMap.Zoom = myMap.Zoom * 0.5;
            break;
        case "1": //Zoom out
            this.myMap.Zoom = myMap.Zoom * 2;
            break;
        case "2": //Zoom to extents
            this.myMap.ZoomToExtents();
            break;
        default:
            break;
    }
}
```

Entsprechend der aktiven Funktionsauswahl in der Toolbox wird nach dem Klick auf die Map ein Code-Block in der «Switch-Case-Methode» ausgeführt. Für das Verändern der Kartenansicht stellt das «Map»-Objekt von SharpMap zwei Arten zur Verfügung. Einerseits

kann sie über die «Zoom» Variable verändert werden, indem der angegebene Wert vergrößert oder verkleinert wird. Andernfalls gibt es die Methode «ZoomToExtents», mit der der Massstab direkt auf die grösste angezeigte Karte angepasst wird. Diese Werte gelten im Anschluss als Informationsanzeige auf der Toolbox (SharpMap, 2012d).

Beim dritten Element der Abbildung 19 handelt es sich um die Legende für das Web-GIS. Die Abbildung 23 dient als Beispiel dafür, wie diese aussieht.

Abbildung 23: Beispiel Legende



Die blau markierten Elemente sind die Legendenbereiche. Sie beschreiben den Bereich für die darunterliegenden Einträge, die mit der Nummer Zwei markiert sind. Die Legendeneinträge beschreiben die Bedeutung der einzelnen Symbole, die im GIS auftauchen können. Der dazu benötigte Programmcode ist in Form von Pseudocode in der Abbildung 24 aufgeführt. Im «Anhang IV: Methode initializeLegende» ist der detaillierte Code verfügbar.

Abbildung 24: Methode initializeLegende (Legende)

```

1 initializeLegende
2     {
3         foreach Legendenbereich in alleLegendenbereiche
4         {
5             TableHeader erstellen für LegendTable
6             TableHeader Legendenbereich auf den Table setzen
7
8             foreach LegendenEintrag in alleLegendenEintraege
9             {
10                if LegendenEintrag gehört zu Legendenbereich
11                {
12                    TableRow erstellen
13                    TableRow auf Table setzen
14
15                    TableCell mit Symbol erstellen
16                    TableCell Symbol auf TableRow setzen
17
18                    TableCell für Titel erstellen
19                    TableCell Titel auf TableRow setzen
20
21                    TableCell für Beschreibung erstellen
22                    if LegendenEintrag enthält Beschreibung
23                        TableCell Beschreibung auf TableRow setzen
24                }
25            }
26        }
27    }

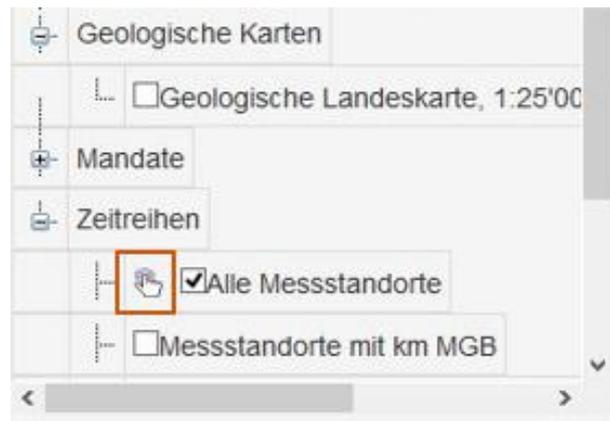
```

Das Vorgehen ist ebenfalls ähnlich wie bei der Initialisierung des Baumes in der Kartenauswahl, insofern zwei Foreach-Schleifen eingesetzt werden. Jedoch wurde als Steuerelement in diesem Fall eine «Table» benutzt. Dementsprechend werden alle Informationen der «Legendenbereiche» und den dazugehörigen «Legendeneinträgen» mittels «TableCell»-Objekte auf dieses Element geladen.

Elemente und Funktionen für den speziellen Anwendungsbereich

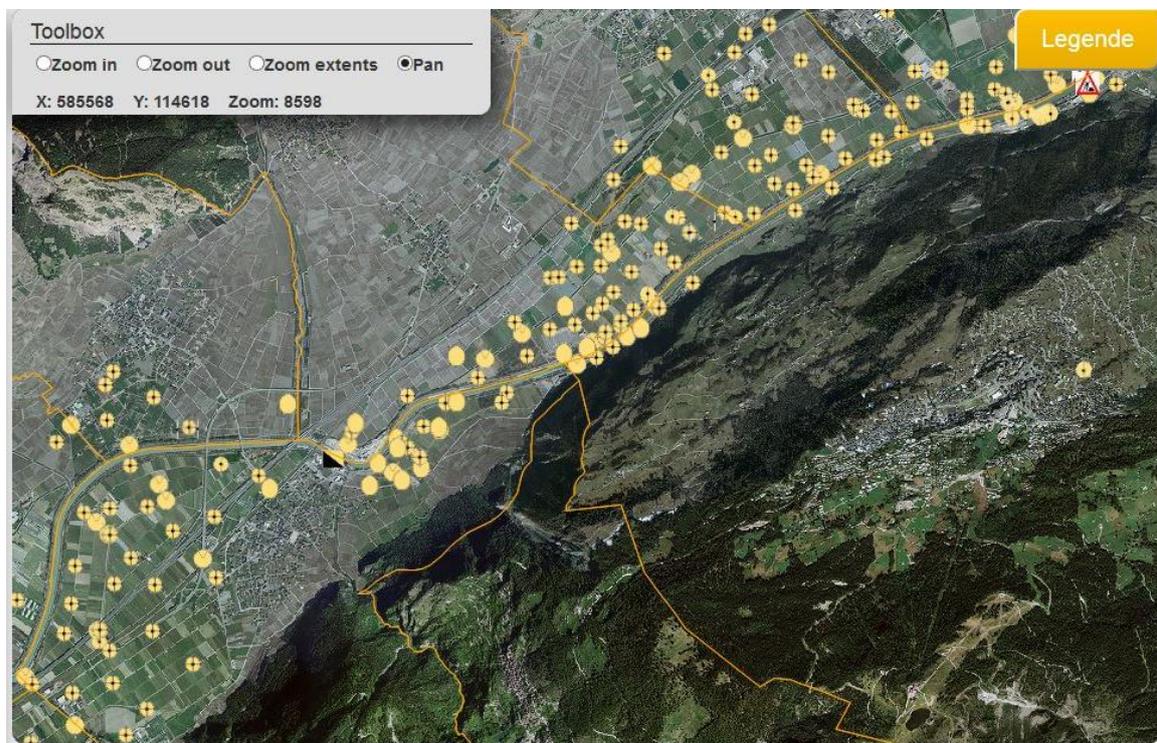
Nebst der aufgeführten Grundanwendung beinhaltet das Web-Geoinformationssystem ausserdem spezifische Elemente und Funktionen, die für den «Dokumenten-Zugriff» benötigt werden.

Abbildung 25: Klickbarer "logischer Layer"



Um diese Anforderung zu erfüllen, wurden sogenannte «Speziallayer» erstellt. Sie werden benötigt, um spezifische Dokumente wie Auswertungen und Beurteilungen von diversen Messungen oder Untersuchungen abfragen zu können. Deshalb sind die dazugehörigen «logischen Layer» im Bereich der Kartenauswahl mit einem Klick-Symbol gekennzeichnet, wie in der Abbildung 25 gezeigt wird. Allerdings sind nicht alle «SpezialLayer» zwangsläufig klickbar. Sie können auch nur zur Darstellung dienen, wie es bei einfachen Karten der Fall ist.

Abbildung 26: Beispiel Kartenausschnitt Speziallayer



Damit ein Dokument abgefragt werden kann, muss der klickbare «logische Layer» aktiviert werden und an oberster Stelle in der Kartensortierung liegen. Die Abbildung 26 dient als

Beispiel für einen Kartenausschnitt, nachdem ein solcher Layer aktiviert wurde. Die Symbole zeigen die «SpezialLayer» und stellen in diesem Fall verschiedene Messstandorte dar. Die «speziellen Layer» enthalten immer die Koordinatenpunkte X, Y und sind mit einem passenden Symbol sowie einer Textangabe verbunden. In der Legende können die Bedeutungen der Symbole nachgelesen werden. Die Implementation für die Erstellung von «SpezialLayer» wird in der Applikationsschicht näher beschrieben.

Anknüpfend an der Aktivierung des klickbaren «logischen Layer» kann der Anwender mit dem Mauszeiger ein Symbol seiner Wahl auf der Karte anklicken. Demzufolge wird im Hintergrund des Programmes anhand des Massstabs und des ausgewählten Koordinatenpunktes eine Fläche erstellt, die für den Benutzer nicht sichtbar ist. Sie dient nur dem Entwickler zur Erfassung der angeklickten Symbole. Dazu wird die Methode in Abbildung 27 benötigt.

Abbildung 27: Methode *isClicked* (*SpezialLayer*)

```
private bool isClicked(int cliCoord, int realCoord, int zoom)
{
    bool isClicked = false;
    int mynumbercheck = zoom;
    if (zoom <= 800)
    {
        if ((realCoord - 3) <= cliCoord && cliCoord <= (realCoord + 3)) isClicked = true;
    }
    else if (800 < zoom && zoom <= 10000)
    {
        if ((realCoord - 30) <= cliCoord && cliCoord <= (realCoord + 30)) isClicked = true;
    }
    else if (10000 < zoom && zoom <= 25000)
    {
        if ((realCoord - 60) <= cliCoord && cliCoord <= (realCoord + 60)) isClicked = true;
    }
    else if (25000 < zoom && zoom <= 50000)
    {
        if ((realCoord - 140) <= cliCoord && cliCoord <= (realCoord + 140)) isClicked = true;
    }
    else if (50000 < zoom)
    {
        if ((realCoord - 160) <= cliCoord && cliCoord <= (realCoord + 160)) isClicked = true;
    }
    return isClicked;
}
```

Jeder einzelne «SpezialLayer» wird nach einem Klick auf die Map mit dieser Methode überprüft. Sie benötigt als Parameter drei Integer-Variablen, nämlich die angeklickte Koordinate des Benutzers, die Koordinate vom «SpezialLayer» sowie das aktuelle Zoom. Da immer nur eine Koordinate angegeben werden kann, muss die Methode zweimal aufgerufen werden. Einmal für die X und einmal für die Y Koordinate. Anhand des angegebenen Zooms

wird dann die entsprechende Überprüfung durchgeführt, ob sich die angeklickte Koordinate im Bereich der Koordinate des übergebenen «SpezialLayers» befindet.

Die Abbildung 28 zeigt die Methode, in der jeder einzelne «SpezialLayer» mit der Methode «isClicked» geprüft wird.

Abbildung 28: Methode sLayerClicked (SpezialLayer)

```
private void sLayerClicked()
{
    if (this.sLayerKlickArray != null)
    {
        foreach (SpezialLayer sLayer in this.sLayerKlickArray)
        {
            if (sLayer.xCoor != null && sLayer.yCoor != null)
            {
                if (isClicked((int)this.myMap.Center.X, (int)sLayer.xCoor.Value, (int)this.myMap.Zoom)
                    && isClicked((int)this.myMap.Center.Y, (int)sLayer.yCoor.Value, (int)this.myMap.Zoom))
                {
                    if (!string.IsNullOrEmpty(kLayerKlick.docQuery))
                    {
                        createKlickBaum(LayerManager.getDocLayer(this.kLayerKlick, sLayer.idSpezialLayer), sLayer);
                    }
                }
            }
        }
    }
    //Setzt variable ob etwas angeklickt wurde -> für postback reload
    if (this.docLayerKlickList.Count != 0)
    {
        ViewState["hasClicked"] = true;
        this.hiddenClickedVariable.Value = "true";
        createDownloadButton();
    }
}
```

Zuerst überprüft das System, ob «SpezialLayer» vorhanden sind. Anschliessend durchläuft die Methode die «SpezialLayer» mit einer Foreach-Schleife. So wird jeder einzelne Layer getestet, ob sich der angeklickte Koordinatenpunkt des Benutzers in seiner Nähe befindet. Die vom Anwender ausgewählten Koordinaten X und Y können aus dem «Map»-Objekt von SharpMap mittels dem Aufruf «Map.Center» gelesen werden.

Mit der Abbildung 29 wird veranschaulicht, wie das geographische Informationssystem reagiert, falls dieser Fall eintritt und sich im entsprechenden «SpezialLayer» Dokumente befinden.

Abbildung 29: Dokumentenbereich



In dieser Situation wurde ein Klick auf eines der zwei aufgeführten Symbole ausgeführt. Da sich diese Messungen in unmittelbarer Nähe befinden, wurden beide mit der im Hintergrund erzeugten Fläche erfasst. Im Folgenden werden die Dokumente der ausgewählten «SpezialLayer» geladen und in Form einer Tabelle unterhalb der Map dargestellt. Der Bereich mit der Nummer Zwei in der Abbildung 29 zeigt die automatisch kreierte Tabelle sowie die darin enthaltenen Elemente.

Darüber hinaus ist im ersten Bereich der aufgezeigten Abbildung ersichtlich, dass die «SpezialLayer» immer mit passenden Textangaben gekennzeichnet sind. Dazu gehören beispielsweise bei einer Messung der Profilname sowie die Anzahl hinterlegter Dokumente. Diese Beschriftungen erscheinen erst ab einem Zoom von 5000 in der Kartenansicht.

Für die Erstellung der Tabelle mit den Dokumenten ist die «createKlickBaum»-Methode verantwortlich. Sie wird in der Methode «sLayerClicked» gemäss der Abbildung 28 aufgerufen und enthält zwei Parameter.

Dazu gehören einerseits die aus der Datenbank gelesenen «Dokumentenlayer», andererseits der passende «SpezialLayer». Das Element mit der Nummer Drei ist ein sogenannter «Dokumentenlayer». Diese Layer repräsentieren jeweils ein Dokument, das zu

einem entsprechenden «SpezialLayer» gehört und eine Beschreibung sowie einen Dateitypen beinhaltet. Darüber hinaus werden sie mittels einer SQL-Abfrage, die im passenden «konkreten Layer» gespeichert ist, aus der Datenbank gelesen. Der Programmcode für diesen Lesezugriff wird nicht gezeigt, da keine spezifische Logik für das Lesen von Informationen aus der Datenbank verwendet wird.

Die Methode «createKlickBaum» sieht gemäss Abbildung 30 wie folgt aus. Der originale Code ist im «Anhang V: Methode createKlickBaum» verfügbar.

Abbildung 30: Methode createKlickBaum (Dokumentenbereich)

```

1 createKlickBaum -> Liste-Dokumentenlayer, SpezialLayer
2     {
3         if Liste-Dokumentenlayer ungleich null
4         {
5             TableHeader erstellen für SpezialLayer
6             TableHeader Speziallayer auf Table setzen;
7
8             TableHeader Checkbox erstellen für SpezialLayer
9             TableHeader Checkbox auf Table setzen
10            foreach Dokumentenlayer in alleDokumentenlayer
11            {
12                TableRow erstellen
13                TableRow auf Table setzen
14
15                for (int i = 0; i < 4; i++)
16                {
17                    TableCell erstellen für Dokumentenlayer
18                    switch (i)
19                    {
20                        case 0: TableCell mit Dokumenten-Beschreibung füllen
21                            break;
22                        case 1: TableCell mit Dokumenten-Typ füllen
23                            break;
24                        case 2: TableCell mit Checkbox für Download füllen
25                            break;
26                        case 3: TableCell mit Button für Download füllen
27                            break;
28                        default:
29                            break;
30                    }
31                    TableCell auf TableRow setzen
32                }
33            }
34        }
35    }

```

Der Ablauf für die Erstellung der Tabelle ist derselbe wie der für die Legende. Es werden alle Informationen der «Dokumentenlayer» gelesen und anschliessend mithilfe den «TableCell»-Elementen auf das «Table»-Steuerelement gesetzt.

Die Dokumente können nun einzeln mithilfe des Buttons «öffnen» entweder heruntergeladen oder direkt angesehen werden, wie im vierten Bereich der Abbildung 29 gezeigt wird. In Abbildung 31 ist der Programmcode für den Download einer Datei dargestellt.

Abbildung 31: Methode `AnzeigeButton_Click` (Download Dokument)

```
protected void AnzeigeButton_Click(object sender, EventArgs e)
{
    Button clickDocButton = (Button)sender;
    string[] idsClickedDoc = clickDocButton.ID.Split(';');
    DocLayer anzeigeDocLayer = auswahlDokument(Convert.ToInt16(idsClickedDoc[1].ToString()),
        Convert.ToInt16(idsClickedDoc[2].ToString()));

    string docName = getDocumentName(anzeigeDocLayer);

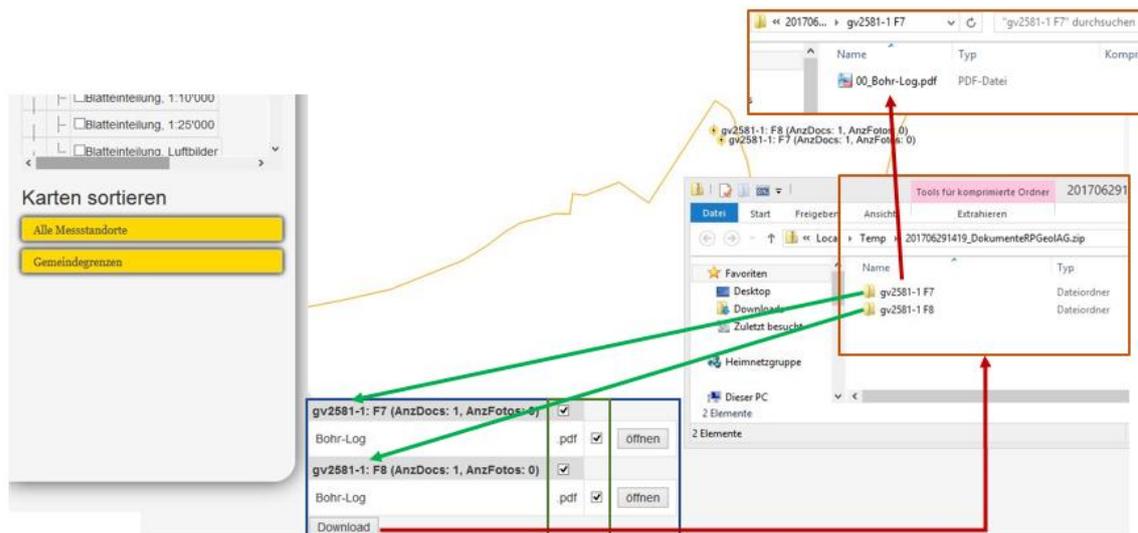
    byte[] document = LayerManager.getDocument(anzeigeDocLayer, this.kLayerKlick);

    //GetBild -> anzeigen oder downloaden
    Response.Clear();
    Response.ClearHeaders();
    Response.AddHeader("Content-Type", "Application/octet-stream");
    Response.AddHeader("Content-Length", document.Length.ToString());
    Response.AddHeader("Content-Disposition", "attachment; filename=" + docName.Trim());
    Response.BinaryWrite(document);
    Response.Flush();
}
```

Zuerst wird anhand des angeklickten Buttons der entsprechende «Dokumentenlayer» samt Informationen geladen und in das Objekt «anzeigeDocLayer» gespeichert. Im Anschluss dient die Methode «getDocument» dazu, das originale Dokument aus der Datenbank zu laden und im byte-Array «document» festzuhalten. Hier wird wieder ein einfacher Lesezugriff mit einer SQL-Abfrage durchgeführt, weshalb diese Methode nicht im Detail erklärt wird. Nachdem das Dokument geladen wurde, kann es mit dem ASP «Response»-Objekt heruntergeladen werden.

Des Weiteren besteht für den Anwender gemäss Abbildung 29 die Möglichkeit, mehrere Dokumente gleichzeitig herunterzuladen. Dazu müssen sie zuerst über die Kontrollkästchen aktiviert werden. Zuletzt muss auf den «Download»-Button im Bereich der Nummer Fünf geklickt werden.

Abbildung 32: Dokumente herunterladen



Nach dem Herunterladen werden die ausgewählten Dokumente dem Anwender in einem komprimierten Ordner zur Verfügung gestellt, wie in Abbildung 32 übersichtlich dargestellt ist. Dieser Ordner beinhaltet die einzelnen Dokumente, festgehalten in derselben Struktur wie die automatisch erstellte Tabelle aufgebaut ist.

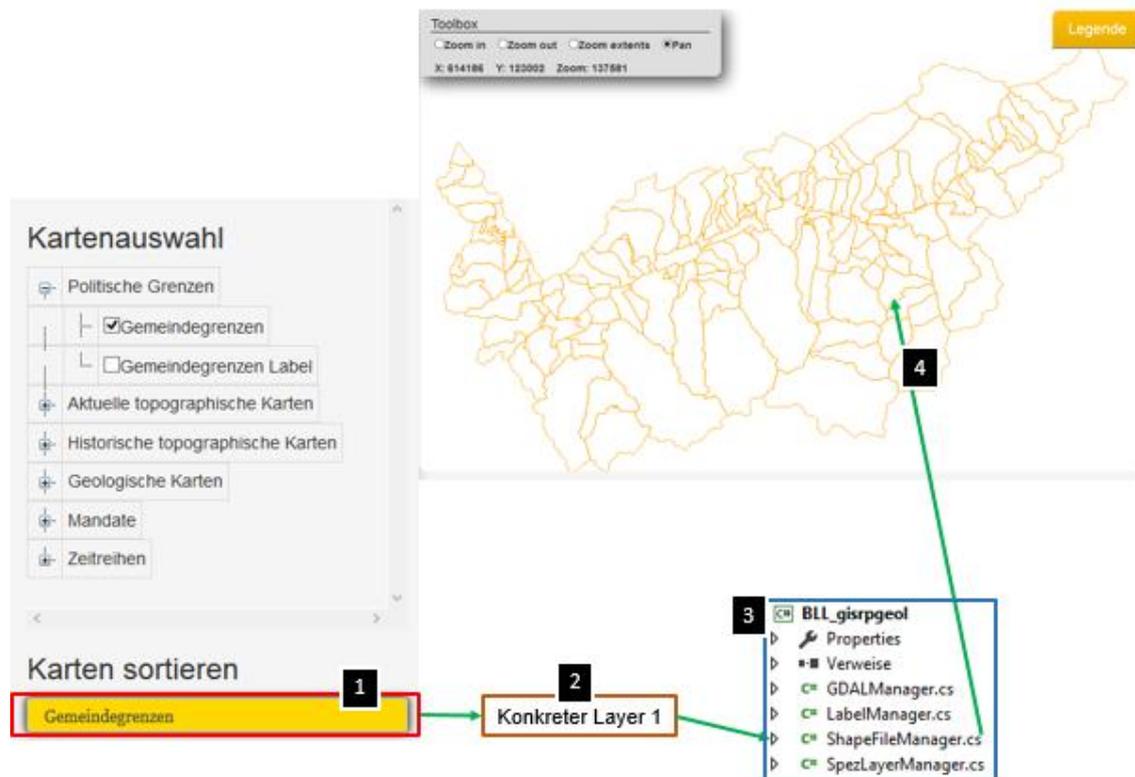
Für das Komprimieren der ausgewählten Dokumente wurde das Projekt «DotNetZip» eingesetzt, mit dem ZIP-Dateien verwaltet werden können (Chiesa, 2016). Da der Programmcode für das Herunterladen von mehreren Dokumenten dem Code für das Herunterladen von einem Dokument entspricht, wird er nicht mehr im Detail aufgezeigt.

Durch die Möglichkeit, Dokumente abfragen und herunterladen zu können, haben die wichtigsten Infrastrukturbetreiber direkten Zugriff auf Daten, die von der RPGeol AG zur Verfügung gestellt werden. So kann die Unternehmung nun Auswertungen und Beurteilungen über das Web-Geoinformationssystem anbieten.

4.1.3 Applikationslayer

In diesem Layer wird der Fokus auf die Datentransformation gelegt. Er ist dafür verantwortlich, sogenannte Kartenschichten zu erstellen. In einem solchen Objekt ist die tatsächliche Karte gespeichert, die dem Anwender auf der Map angezeigt wird. Dazu benötigt die Applikationsschicht Informationen, die mittels den Objekten «konkrete Layer» zur Verfügung gestellt werden. Die erstellten Kartenschichten werden im Anschluss auf das «Map»-Objekt geladen. In der Abbildung 33 wird ersichtlich, wie dieser Vorgang in der Praxis aussieht.

Abbildung 33: Vorgang Datentransformation



Der Vorgang für die Anzeige einer Karte auf der Map erfolgt in vier Schritten:

Zuerst muss der Benutzer die gewünschten Karten bei der Kartenverwaltung aktivieren und sortieren. In diesem Fall wurden die Gemeindegrenzen aktiviert. Sobald nun ein Mausklick auf die Map durchgeführt wird, lädt das Web-GIS alle «konkreten Layer», die zu dem ausgewählten «logischen Layer» gehören. Die geladenen Informationen und Daten werden anschliessend an die Applikationsschicht übertragen, in der die Datentransformation zur Erstellung der Kartenschichten-Objekte erfolgt. Zum Schluss werden diese erstellten Objekte auf die Map geladen. Die Abbildung 34 zeigt nochmals auf, wie die Map gemäss diesem Vorgang initialisiert wird. Der vollständige Programmcode ist im «Anhang III: Methode InitializeMap» aufgeführt.

Abbildung 34: Methode InitializeMap (Kartenschichten)

```

1 InitializeMap
2 {
3     Map-Objekt erstellen
4
5     foreach LogischerLayer in aktivierteLogischeLayer
6     {
7         foreach KonkreterLayer in alleKonkretenLayer
8         {
9             if KonkreterLayer gehört zu LogischerLayer
10            {
11                // Benutzerrechte vorhanden für LogischerLayer
12                {
13                    Kartenschicht mithilfe KonkreterLayer auf Map setzen
14                }
15            }
16        }
17    }
18 }
19 return Map-Objekt
20 }

```

Nun gilt der Fokus der Erstellung von Kartenschichten durch die verschiedenen Verwaltungsklassen des Applikationslayers, da das Geoinformationssystem Schichten aus unterschiedlichen Datenquellen erstellen kann. Zur Auswahl der verlangten Verwaltungsklasse dienen «Layertypen». Sie sind ebenfalls in den «konkreten Layer»-Objekten gespeichert und repräsentieren jeweils eine Datenquelle, wie beispielsweise ein Shapefile. Der Pseudocode, in dem die tatsächliche Selektion erfolgt, wird mit der Abbildung 35 gezeigt. Im «Anhang VI: Methode setKonkreteLayer» ist der konkrete Code beigelegt.

Abbildung 35: Methode setKonkreteLayer (Kartenschichten)

```

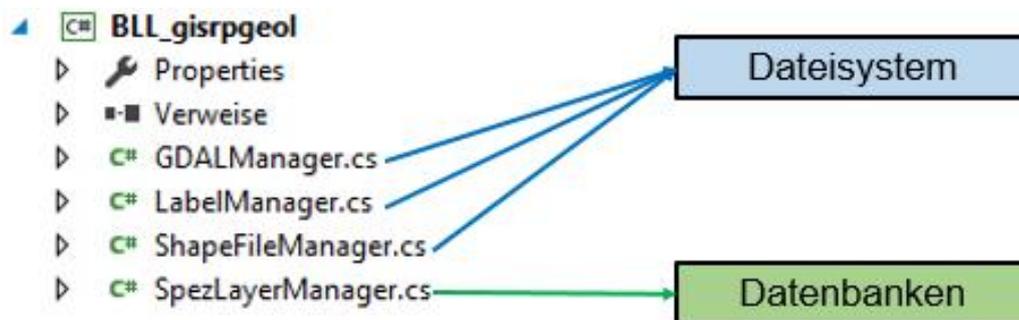
1 setKonkreteLayer
2 {
3     switch (Layertyp von KonkreterLayer)
4     {
5         case 1: neue ECW-Kartenschicht erstellen
6             break;
7         case 2: neue Shapefile-Kartenschicht erstellen
8             break;
9         case 3: neue SpezialLayer-Kartenschicht erstellen
10            break;
11        case 4: neue GeoTIFF-Kartenschicht erstellen
12            break;
13        case 5: neue JPG2000-Kartenschicht erstellen
14            break;
15        case 6: neue Label-Kartenschicht erstellen
16            break;
17        default:
18            break;
19    }
20 }

```

Dieses «Switch-Case» wird an der gelb markierten Stelle der Abbildung 34 aufgerufen und ist verantwortlich dafür, dass die richtige Verwaltungsklasse der Applikationsschicht je nach «Layertyp» aufgerufen wird. Diese Typen sind immer mit einer Nummer dargestellt.

In Abbildung 36 werden die Quellen gezeigt, auf die im Anschluss die Verwaltungsklassen zugreifen:

Abbildung 36: Zugriff auf Datenquellen (Applikationsschicht)



Zum «GDALManager» gehören die Methoden, mit denen Dateien wie ECW, JPG2000 und GeoTiff (siehe Glossar) direkt aus dem Dateisystem gelesen werden können. Der Code in der Abbildung 37 zeigt, was benötigt wird, um eine GDAL-Kartenschicht zu erstellen und sie auf die Map zu laden.

Abbildung 37: Code Erstellung GDAL-Kartenschicht (GDALManager)

```
SharpMap.Layers.GdalRasterLayer newEcw =
    new SharpMap.Layers.GdalRasterLayer(kLayer.kLayerName, @kLayer.path);
if (kLayer.maxVisible != 0)
    newEcw.MaxVisible = kLayer.maxVisible;
newEcw.MinVisible = kLayer.minVisible;
map.Layers.Add(newEcw);
```

Wie ersichtlich ist, wird das von SharpMap zur Verfügung gestellte Objekt «GdalRasterLayer» benötigt. Für dessen Initialisierung sind nur zwei Variablen nötig, nämlich der Layername, den der erstellte Layer erhalten soll sowie der Pfad zu einer GDAL-Datei (Searchcode, kein Datum).

Zusätzlich beinhalten diese Layer-Objekte von SharpMap spezifische Variablen, mit denen Eigenschaften für die Darstellung der Karte gesetzt werden können. Beispielsweise kann mit den Variablen «maxVisible» und «minVisible» definiert werden, ab welchem Zoom der Layer angezeigt wird, mit «BackColor» kann zum Beispiel die Hintergrundfarbe verändert werden. Solche Variablen werden in dieser Arbeit nicht genauer beschrieben. Mit der Funktion

«map.Layers.Add» kann im Anschluss die neue Kartenschicht auf das «Map»-Objekt geladen werden.

Der «ShapeFileManager» ist dementsprechend für Shapefiles verantwortlich. Die Abbildung 38 zeigt den Programmcode, der für das Erstellen eines Shapefiles benötigt wird.

Abbildung 38: Code Erstellung ShapeFile-Kartenschicht (ShapeFileManager)

```
SharpMap.Layers.VectorLayer newLayer = new SharpMap.Layers.VectorLayer(kLayer.kLayerName);
newLayer.DataSource = new SharpMap.Data.Providers.ShapeFile(@kLayer.path);
newLayer.DataSource.Open();
newLayer.MinVisible = kLayer.minVisible;
newLayer.MaxVisible = kLayer.maxVisible;
newLayer.DataSource.Close();
map.Layers.Add(newLayer);
```

In diesem Fall muss zuerst ein «VectorLayer» mit einem gewünschten Namen initialisiert werden, auf dem im Anschluss eine «DataSource» gesetzt wird. Das Objekt «DataSource» ist ebenfalls von SharpMap und verweist auf eine Datenquelle, wie in dieser Situation ein Shapefile. Wichtig ist, dass für die Erstellung des «DataSource»-Objektes der Pfad zur gewünschten Datei angegeben wird (SharpMap, 2012c).

Im «LabelManager» werden die Textangaben kreiert, die zu den Shapefiles gehören. Sie werden auf dieselbe Weise wie die Shapefile-Kartenschicht erstellt. Jedoch wird anstatt eines «VectorLayer»-Objektes ein «LabelLayer»-Objekt benötigt. Zudem muss in der Variable «LabelColumn» des Layers angegeben werden, welche Kolonne in der Shapefile-Tabelle angezeigt werden soll (SharpMap, 2012b).

Die «speziellen Layer» wiederum werden in der Verwaltungsklasse «SpezLayerManager» erstellt. Als Datenquelle dienen hierfür Datenbanken. Mit der Abbildung 39 ist in einem Pseudocode dargestellt, wie die Implementation für die Erstellung der «SpezialLayer» aussieht. Im «Anhang VII: Methode getSpezialLayer» ist der vollständige Code aufgeführt.

Abbildung 39: Methode `getSpezialLayer` (`SpezLayerManager`)

```

1  getSpezialLayer
2      {
3          SpezialLayer aus der Datenbank lesen
4
5          if LogischerLayer an oberster Stelle der Sortierliste
6              && LogischerLayer ist klickbar
7              setze SpezialLayer in den Zwischenspeicher
8
9          VectorLayer erstellen
10
11         KoordinatenPunkte von SpezialLayer auf VectorLayer laden
12
13         Symboleigenschaften zu KoordinatenPunkte hinzufügen
14         Texteeigenschaften zu KoordinatenPunkte hinzufügen
15
16         Symbole auf VectorLayer laden
17
18         if aktueller Map-Zoom < 5000
19             LabelLayer für SpezialLayer laden
20
21         VectorLayer auf Map setzen
22     }

```

Zuerst werden die «speziellen Layer» von der Datenbank mittels einer SQL-Abfrage gelesen, die von dem übergebenen «konkreten Layer» zur Verfügung gestellt wird. Als nächstes wird überprüft, ob es sich um einen klickbaren «logischen Layer» handelt und ob er sich an oberster Stelle in der Sortierliste befindet, da nur die oberste Kartenschicht angeklickt werden kann. Falls diese Bedingungen stimmen, werden die «speziellen Layer»-Objekte in den Zwischenspeicher gelegt, damit sie für die im Dokumentenbereich aufgezeigte Klick-Methode wiederverwendet werden können.

Anschliessend wird ein «VectorLayer»-Objekt erstellt, wie beim «ShapeFileManager» gezeigt wird. Für die Initialisierung der dazugehörigen «DataSource» wird das Objekt «GeometryFeatureProvider» benötigt, das ebenfalls von SharpMap zur Verfügung gestellt wird. Als Konstruktor für die Erstellung dieses Objektes muss eine Collection von «IGeometry»-Objekten angegeben werden. In dieser Liste sind alle Koordinatenpunkte der «SpezialLayer» aufgelistet. Zum Füllen der Collection dient der Programmcode in der Abbildung 40.

Abbildung 40: "IGeometry"-Collection füllen (getSpezialLayer)

```

foreach (SpezialLayer sLayer in sLayerArray)
{
    if (sLayer.xCoor != null && sLayer.yCoor != null)
        geomColl.Add(gf.CreatePoint
            (new GeoAPI.Geometries.Coordinate(sLayer.xCoor.Value, sLayer.yCoor.Value)));
}

```

Alle aus der Datenbank gelesenen «SpezialLayer» werden mit einer Foreach-Schleife durchlaufen. Zuerst wird überprüft, ob der aktuelle Layer die Koordinatenpunkte X, Y enthält. Anschliessend werden sie auf die Collection geladen. In der aufgezeigten Situation ist die Collection in der «geomColl» Variable gespeichert. Mit der Methode «Add» können die Koordinatenpunkte über das von SharpMap zur Verfügung gestellte Objekt «Coordinate» auf die Liste geladen werden (SharpMap, 2012a).

Damit diese Punkte auf der Map als Symbole mit Textangaben angezeigt werden, muss die «DataSource» mit diesen Daten erweitert werden. In der Abbildung 41 wird aufgezeigt, wie der Code dazu aussieht.

Abbildung 41: Erweiterung DataSource (getSpezialLayer)

```

foreach (SpezialLayer sLayer in sLayerArray)
{
    if (sLayer.xCoor != null && sLayer.yCoor != null)
    {
        styleProp[0] = "\t \t " + sLayer.label.TrimEnd();
        if (!string.IsNullOrEmpty(sLayer.markerPath))
            styleProp[1] = sLayer.markerPath;
        gp.Features[sLayerStyleCounter].ItemArray = styleProp;
        sLayerStyleCounter++;
    }
}

```

Ähnlich wie bei der Methode zum Füllen der Collection, werden nochmals alle «SpezialLayer» mit einer Foreach-Schleife durchlaufen. Für jeden Layer dient im Anschluss das «styleProp»-Array dazu, den Pfad zum Symbol sowie die Textangabe zu speichern. Dieses Array wird an der passenden Stelle in der «DataSource» hinzugefügt, die in diesem Fall mit dem Namen «gp» repräsentiert wird.

Zum Anzeigen der Symbole musste ein sogenanntes «CustomTheme»-Objekt erstellt werden, das für die Initialisierung als Parameter eine Methode erhält. Dieses Vorgehen ist in der Abbildung 42 aufgeführt und zeigt, wie auf jedem einzelnen «SpezialLayer» das passende Symbol geladen wird.

Abbildung 42: CustomTheme für Symbole (getSpezialLayer)

```

SharpMap.Rendering.Thematics.CustomTheme sLayerTheme =
    new SharpMap.Rendering.Thematics.CustomTheme(GetSLayerStyle);

private static SharpMap.Styles.VectorStyle GetSLayerStyle(SharpMap.Data.FeatureDataRow row)
{
    //Wenn kein Marker-Pfad vorhanden -> dann default Style anwenden
    if (string.IsNullOrEmpty(row.ItemArray[1].ToString()))
        return null;

    SharpMap.Styles.VectorStyle sLayerstyle = new SharpMap.Styles.VectorStyle();

    System.Drawing.Image marker = ResizeImage((System.Drawing.Image)(new Bitmap(row.ItemArray[1].ToString())));

    sLayerstyle.Symbol = marker;

    return sLayerstyle;
}

```



Die bei der «DataSource» hinzugefügten Daten können durch das «FeatureDataRow»-Objekt von SharpMap wiederverwendet werden. So kann im Anschluss für jeden Punkt ein anderer «VectorStyle» erstellt werden. Ein solches Objekt beinhaltet spezifische Design-Eigenschaften, wie in diesem Fall das Symbol, das mit der Variable «marker» repräsentiert wird (SharpMap, 2012e).

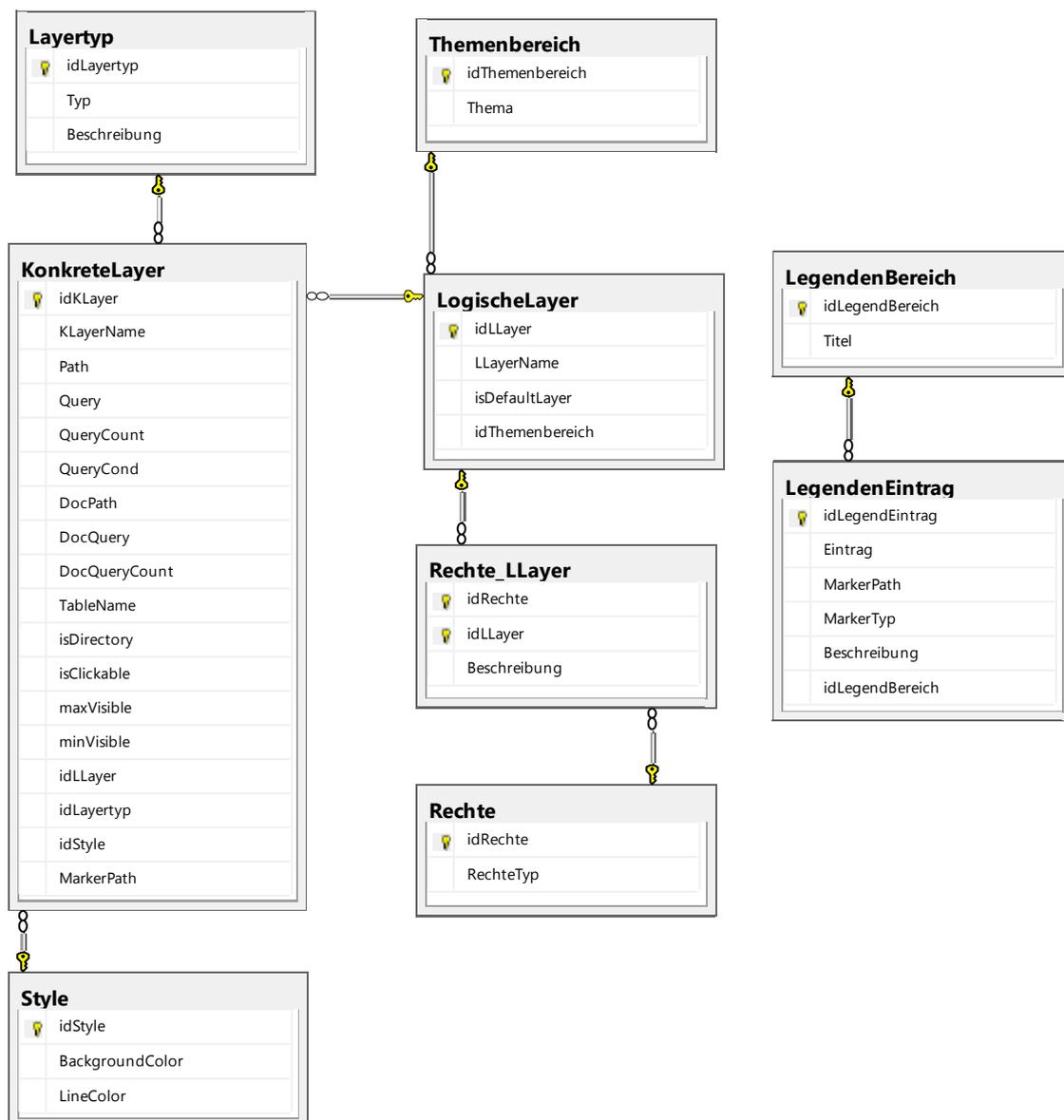
Bevor schliesslich das «VectorLayer»-Objekt mit den «SpezialLayer» auf die Map geladen wird, überprüft das System, ob der aktuelle Zoom der Map unter 5000 liegt. Wenn diese Bedingung zutrifft, werden die Labels angezeigt. Diese sind gemäss der Beschreibung im «LabelManager» zu erstellen, wobei in diesem Fall die «DataSource» angegeben werden muss, die in der Methode «getSpezialLayer» kreiert wurde. Zum Schluss wird die Kartenschicht über den «VectorLayer» auf die Map geladen.

4.1.4 Datenbanklayer und GIS-Datenbank

Im Datenbanklayer geht es darum, die Informationen und Daten aus der GIS-Datenbank zu lesen und in die passenden Objekte abzuspeichern, die im Kapitel der Datenmodelle und Objekte beschrieben wurden. Des Weiteren wird mittels der GIS-Datenbank das Geoinformationssystem gesteuert, insofern sich das System anhand der Datenbankeinträge dynamisch erstellt.

Im Folgenden wird die Datenbank genauer untersucht. Dementsprechend werden die einzelnen definierten Tabellen mit ihren Relationen und Einträgen aufgezeigt. Da der Datenbanklayer nur für die primitiven Datenzugriffe eingesetzt wird, sind keine weiteren Beschreibungen diesbezüglich nötig. Die Abbildung 43 zeigt als Gesamtüberblick das aufgebaute Datenbank-Diagramm.

Abbildung 43: Datenbank-Diagramm

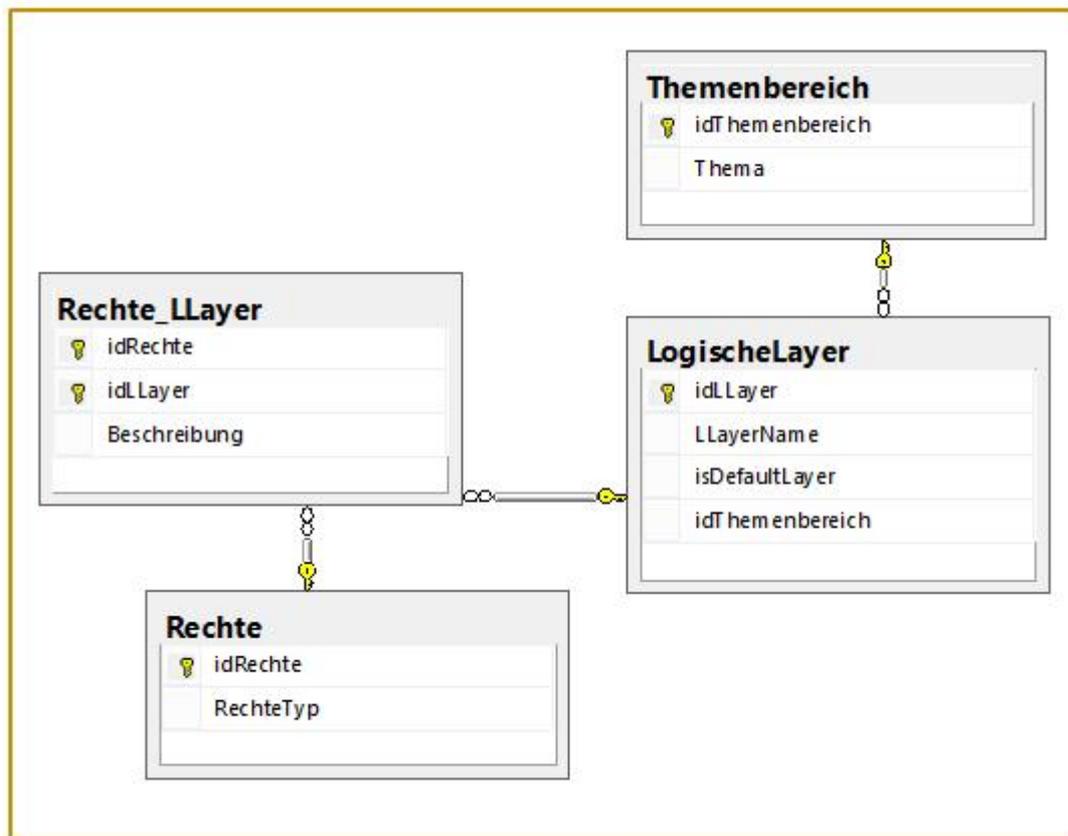


Gemäss dem dargelegten Diagramm wird ersichtlich, dass die Tabellen und deren Relationen zu den in den Schichten angewandten Elementen passen. Zusätzlich beinhalten alle Tabellen einen numerischen Identifikationsschlüssel.

Im Folgenden wird nun das Datenbankmodell in einzelne Bereiche aufgeteilt, die anschliessend näher erläutert werden. Wichtig zu beachten ist, dass nur spezielle Variablen von einzelnen Tabellen genauer beschrieben werden. Einfache Namen, Beschreibungen, Identifikations- oder Fremdschlüssel gehören nicht dazu.

Kartenverwaltung

Abbildung 44: Kartenverwaltung DB-Modell



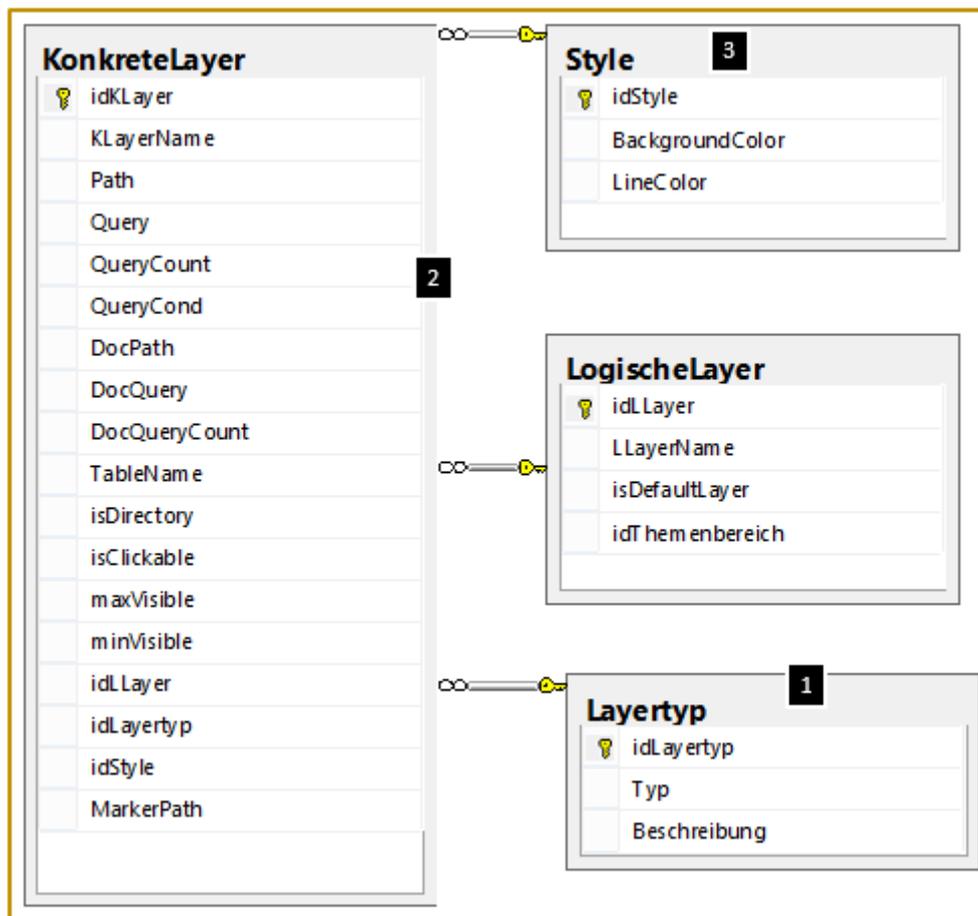
Der erste Bereich bezieht sich entsprechend der Abbildung 44 auf die Kartenverwaltung und stellt sich aus den vier Tabellen «Themenbereich», «LogischeLayer», «Rechte_Layer» sowie «Rechten» zusammen. Diese werden benötigt, um dem Anwender auf der Präsentationsschicht die Auswahl und Sortierung der Karten zu ermöglichen.

Zudem sind in diesem Teil des Modells die Benutzerrechte definiert. Sie können über die Tabelle «Rechte_Layer» vergeben werden, indem jeweils angegeben wird, welcher «LogischeLayer» von welchem «RechteTyp» angewendet werden kann. Als «RechteTyp» kann beispielsweise ein öffentlicher Benutzer gelten oder eine Unternehmung.

Ausserdem muss über die Variable «isDefaultLayer» bestimmt werden, ob ein «Logischer Layer» standardmässig beim Start der Applikation aktiviert sein soll oder nicht. Es wird empfohlen, dass jeder Benutzertyp mindestens einen «logischen Layer» beim Programmstart per Default aktiviert hat. Anderenfalls würde in dieser Situation keine Karte auf der Map angezeigt werden.

Kartenanzeige

Abbildung 45: Kartenanzeige DB-Modell



Wie im Applikationslayer bereits aufgezeigt wurde, werden für die Erstellung der tatsächlichen Karten Informationen von «konkreten Layer» benötigt. Aufgrund der Grösse und Komplexität der Tabelle «KonkreteLayer» werden gemäss Abbildung 45 alle dazugehörigen Variablen und externen Tabellen in einer spezifischen Reihenfolge beschrieben. Die Tabelle «LogischeLayer» stellt hier nur die Verbindung zwischen dem Bereich der Kartenanzeige und der Kartenverwaltung dar.

An erster Stelle wird die Tabelle «Layertyp» betrachtet. Sie ist von grosser Wichtigkeit, da in ihr die verschiedenen Arten von Datenquellen aufgelistet sind, aus denen in der Applikationsschicht Kartenschichten erstellt werden können. Überdies ist sie die einzige Tabelle, die vordefinierte Einträge beinhaltet. Diese sind in der Abbildung 46 aufgeführt und dürfen vom GIS-Verwalter nicht verändert werden, zumal anhand diesen Einträgen im Prozess der Datentransformation die korrekte Verwaltungsklasse in der Applikationsschicht ausgewählt wird. Jeder «KonkreteLayer» beinhaltet immer einen solchen «Layertypen».

Abbildung 46: Vordefinierte Einträge - Tabelle "Layertyp"

	idLayertyp	Typ	Beschreibung
1	1	ecw	ECW File
2	2	shp	ShapeFile
3	3	spezial	SpezialLayer
4	4	tif	GeoTiff
5	5	jpg	JPEG2000
6	6	label	Label

Im zweiten Schritt wird der Fokus auf die Tabelle «KonkreteLayer» gelegt. Aufgrund der verschiedenen Datenquellen kann die Tabelle auf unterschiedliche Arten ausgefüllt werden. Es wird zwischen drei Fällen unterschieden. Dazu gehört zuerst der Normalfall, bei dem der «konkrete Layer» auf eine oder mehrere Kartendateien verweist. Die zweite Situation bezieht sich auf die Erstellung von «Speziallayern» sowie «Dokumentenlayer» und im dritten Fall wird beschrieben, wie «Labels» mittels Shapefiles definiert werden können.

Abbildung 47: Tabelle "KonkreteLayer" (Kartendateien)

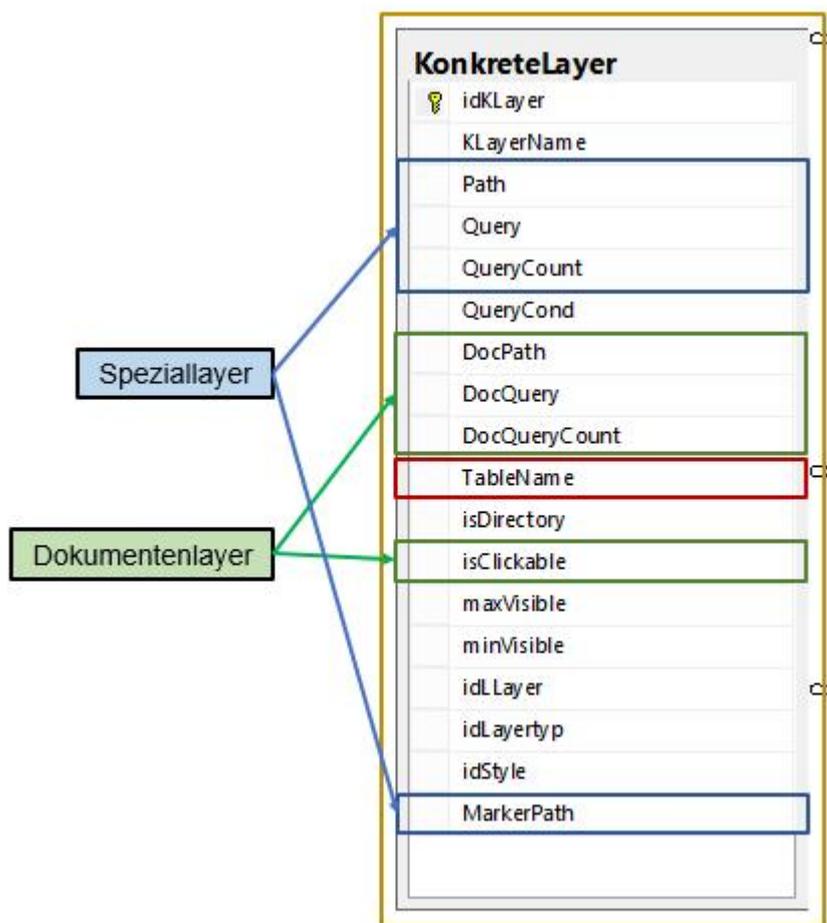
KonkreteLayer	
	idKLayer
	KLayerName
	Path
	Query
	QueryCount
	QueryCond
	DocPath
	DocQuery
	DocQueryCount
	TableName
	isDirectory
	isClickable
	maxVisible
	minVisible
	idLayer
	idLayertyp
	idStyle
	MarkerPath

Um mit dem «konkreten Layer» auf Kartendateien zu verweisen, werden die in der Abbildung 47 blau markierten Variablen benötigt. Die «Path» Variable dient im Normalfall als

Pfadangabe zur entsprechenden Kartendatei, die angezeigt werden soll, wie zum Beispiel ein Shapefile. Es kann aber auch ein Pfad zu einem Dateiordner angegeben werden. In dieser Situation muss jedoch die Variable «isDirectory» aktiviert werden, damit das System weiss, dass es sich um einen Dateiordner handelt. Falls auf einen Ordner referenziert wird, werden alle darin enthaltenen Dateien, die im angegebenen «Layertyp» Format sind, gelesen.

Mit den Variablen «maxVisible» und «minVisible» kann definiert werden, ab welchem Zoom die Karten auf der Map angezeigt werden sollen. Diese Variablen werden immer für diese Funktion eingesetzt, weshalb sie in den nächsten zwei Fällen nicht mehr erwähnt werden.

Abbildung 48: Tabelle "KonkreteLayer" (Spezial- und Dokumentenlayer)



Die markierten Variablen in der Abbildung 48 werden in der zweiten Situation für die Erstellung von «Speziallayer» und «Dokumentenlayer» benötigt. Beide Layer müssen aus einer Datenbank gelesen werden.

Als erstes sind die blauen Markierungen wichtig, insofern sie für die «Speziallayer» verantwortlich sind. Die Variable «Path» dient in diesem Fall dazu, den ConnectionString einzutragen, der die Verbindung zur gewünschten Datenbank aufbaut.

Die Felder «Query» und «QueryCount» werden im Anschluss zum Lesen der Informationen aus der angegebenen Datenbank eingesetzt. Die Struktur der SQL-Abfrage für die «Query» Variable beinhaltet spezifische Regeln. So müssen bei der Antwort die folgenden Rückgabewerte vorhanden sein:

- rp_idSpezialLayer (Integer, muss) = ID eines Koordinatenpunktes
- rp_labelSpeziallayer (nvarchar, muss) = Textangabe für einen Punkt
- rp_xCoor (nvarchar, muss) = X Koordinate für die Darstellung des Punktes auf der Map
- rp_yCoor (nvarchar, muss) = Y Koordinate für die Darstellung des Punktes auf der Map
- rp_markerPath (nvarchar, kann) = Jeder Punkt kann mit einem unterschiedlichen Symbol gekennzeichnet werden. Es wird die Pfadangabe zum Symbol benötigt.

Alle Rückgabewerte müssen mit diesen Namenkonventionen zurückgegeben werden, mit Ausnahme des Wertes «rp_markerPath». Falls jedoch kein Symbol pro Punkt angegeben wird, muss ein Standardsymbol in der Variable «MarkerPath» eingetragen werden. Anderenfalls würden die «Speziallayer» auf der Map nicht angezeigt werden. Wichtig zu beachten ist, dass die eingegebenen Koordinatenpunkte dem «LV03»-System (siehe Glossar) entsprechen, da mit diesem Koordinatensystem gearbeitet wird.

Im Feld «QueryCount» muss nun eine SQL-Count-Abfrage eingetragen werden, die als Rückgabewert dieselbe Anzahl Reihen zurückgibt, wie die Abfrage im Feld «Query» ausgibt.

Der zweite Teil in diesem Fall bezieht sich auf die Erstellung von «Dokumentenlayer». Sie sind mit den «Speziallayer» verbunden, da ein Dokument immer zu einem speziellen Layer gehört.

Die Variablen müssen für die «Dokumentenlayer» auf die gleiche Weise wie für die «Speziallayer» ausgefüllt werden. Jedoch benötigt die Abfrage im Feld «DocQuery» andere Rückgabewerte:

- rp_idDocLayer (Integer), muss = ID von einem Dokument
- rp_beschreibung (nvarchar), muss = Name von einem Dokument
- rp_docTyp (nvarchar), muss = Dokumenttyp (z.B. .pdf). Der Punkt vor der Typbezeichnung muss vorhanden sein.

- rp_dokumentFeld (nvarchar), muss = Name der Kolonne in der SQL-Tabelle, in der das Dokument gespeichert ist. Das Dokument muss im Image-Format gespeichert sein.
- rp_tabelle (nvarchar), muss = Name der SQL-Tabelle, in der das Dokument gespeichert ist.

Wichtig ist, dass alle Felder einen Rückgabewert beinhalten. Darüber hinaus muss die eingetragene SQL-Abfrage eine WHERE-Bedingung aufweisen. Mit dieser Bedingung wird ein Vergleich auf die ID vom geklickten «Speziallayer» durchgeführt, da ein Dokument immer zu einem solchen Layer gehört:

- select (Rückgabewerte) from Dokumente WHERE Dokumente.IDSpeziallayer = @idHandleKlick

Die ID vom «Speziallayer» muss mit «@idHandleKlick» ersetzt werden, da sie vom System anhand des angeklickten Symbols automatisch erkannt wird. Die Klick-Funktion ist im Kapitel der Präsentationsschicht erklärt worden.

Beim Feld «TableName» handelt es sich lediglich um eine Beschreibungsmöglichkeit, die für den Verwalter der Datenbank nützlich sein kann. Sie hat keinen Einfluss auf die Funktionalitäten des Systems. Nachdem die aufgezeigten Variablen ausgefüllt sind, können «Speziallayer» und Dokumente abgefragt werden. Falls keine «Dokumentenlayer» erstellt werden sollen, können die dazugehörigen Felder leer gelassen werden.

Abbildung 49: Tabelle "KonkreteLayer" (Label)

KonkreteLayer	
idKLayer	
KLayerName	
Path	
Query	
QueryCount	
QueryCond	
DocPath	
DocQuery	
DocQueryCount	
TableName	
isDirectory	
isClickable	
maxVisible	
minVisible	
idLayer	
idLayerTyp	
idStyle	
MarkerPath	

Die Definition von Labels mittels Shapefiles ist die dritte Art, wie die Tabelle «KonkreteLayer» ausgefüllt werden kann. Dafür werden gemäss der Abbildung 49 die zwei Felder «Path» und «QueryCond» benötigt.

In diesem Fall dient die Variable «Path» dazu, eine Pfadangabe auf ein bestehendes Shapefile einzutragen. Es darf in dieser Situation auf keine Ordner verwiesen werden. Das Feld «isDirectory» hat beim «Label» keinen Einfluss auf das Verhalten des Systems.

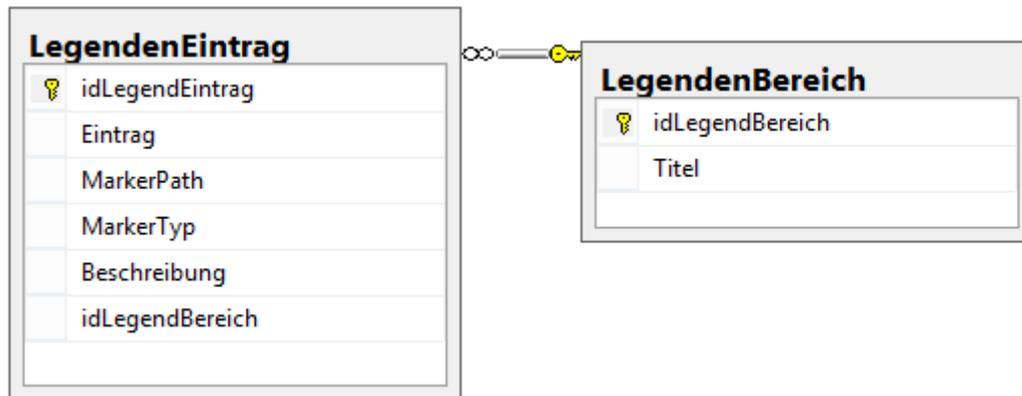
Beim zweiten Feld «QueryCond» muss der Kolonnenname angegeben werden, aus der die Informationen für die Anzeige der Labels stammt. Shapefiles beinhalten die dazu benötigten Tabellen. Sie müssen dementsprechend nicht selbst erstellt werden.

Der dritte Schritt in der Kartenanzeige bezieht sich gemäss Abbildung 45 auf das Design. In dieser Tabelle können Linien- und Hintergrundfarben bestimmt werden. Je nach «LayerTyp» werden die Farben unterschiedlich eingesetzt. So können nur Shapefiles, «Speziallayer» und Labels benutzerdefiniert gestaltet werden. Für die anderen Layer wird das «Style-Objekt» nicht

berücksichtigt. Bei den «Speziallayer» und Labels dienen die «Styles» zur Veränderung der Schriftfarben.

Legendenbereich

Abbildung 50: Legendenbereich DB-Modell



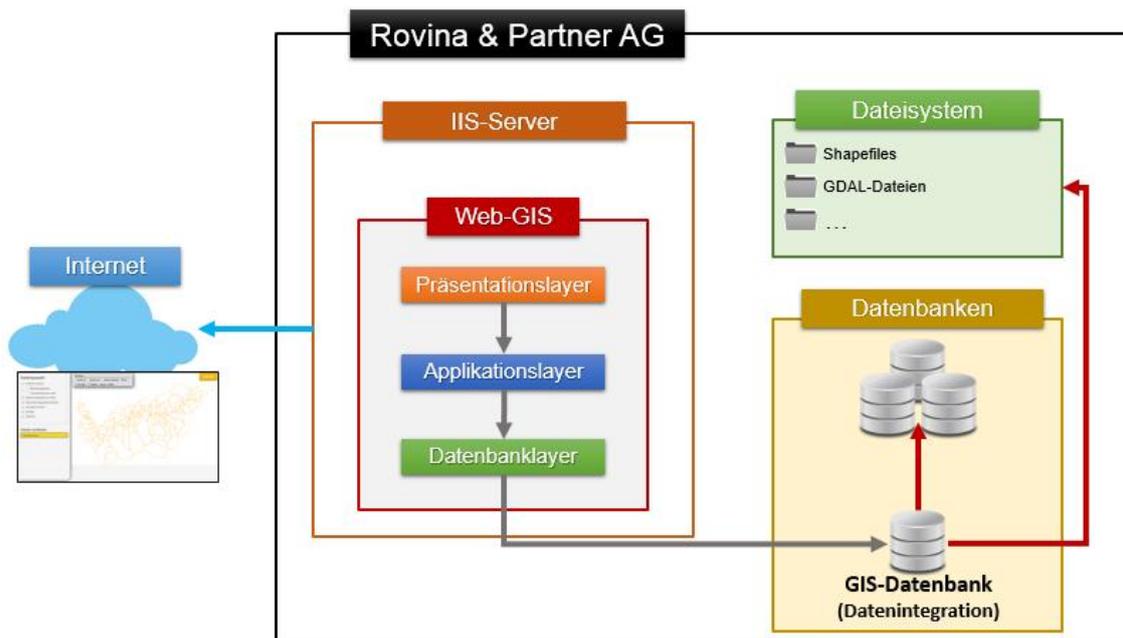
Um die auf der Präsentationsschicht aufgezeigte Legende zu erstellen, werden gemäss der Abbildung 50 die zwei Tabellen «LegendenBereich» sowie «LegendenEintrag» benötigt.

Nachdem ein Legendenbereich definiert wurde, können die dazugehörigen Einträge kreiert werden. Ein Eintrag beinhaltet immer einen Titel, der mit dem Feld «Eintrag» vergeben werden kann. Dazu gehört ein Symbol, das mittels Pfadangabe in der Variable «MarkerPath» angegeben wird. Der Dateityp des Symbols muss im Feld «MarkerTyp» eingetragen sein, zum Beispiel «png». Falls noch eine Beschreibung hinzugefügt werden soll, kann dies über die Variable «Beschreibung» bewerkstelligt werden.

4.2 Integration in die IT-Infrastruktur der Unternehmung

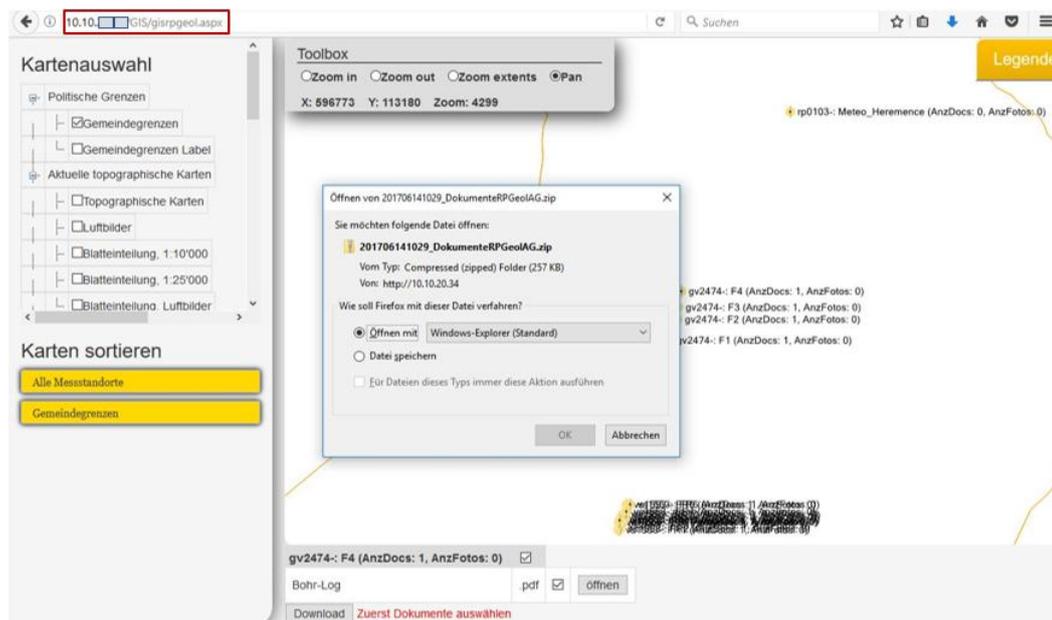
Nach der Entwicklung wird das Web-Geoinformationssystem in die IT-Infrastruktur der Unternehmung integriert, wie in der Abbildung 51 gezeigt wird.

Abbildung 51: Integration in die IT-Infrastruktur



In der dargestellten Architektur ist ersichtlich, wie das System auf dem IIS-Server der RPGeol AG installiert wird. Darüber hinaus werden nochmals die aufgebauten Verbindungen visualisiert. Wichtig zu beachten ist, wie die GIS-Datenbank für die Datenintegration eingesetzt wird und auf die verschiedenen Datenquellen verweist, die bereits in der Unternehmung vorhanden sind. Nachdem die Entwicklung auf dem Webserver läuft, kann die Website über das Internet aufgerufen werden. Mit der Abbildung 52 wird veranschaulicht, wie das Geoinformationssystem im Anschluss in der Praxis aussieht. Hierfür wurde das System ans interne Netzwerk der Unternehmung angeschlossen.

Abbildung 52: Web-GIS nach Deployment



5. Evaluation

Nach der Entwicklung des Web-GIS sowie der Integration in die IT-Infrastruktur der Unternehmung steht die Beurteilung zur Anwendbarkeit des ausgewählten Moduls im Vordergrund sowie die Evaluation zum erstellten Web-Geoinformationssystem.

5.1 Evaluation zur Anwendbarkeit des gewählten Web-GIS Moduls

In diesem Fall wurde das Modul SharpMap eingesetzt. Es soll geklärt werden, ob das Modul die im Kapitel «Analyse und Evaluation vorhandener Web-GIS Module» definierten Bewertungskriterien erfüllt hat. Des Weiteren wird aufgezeigt, welche Vor- und Nachteile sich dadurch ergeben haben. Zum Schluss wird ersichtlich, ob das Modul für die Erstellung des verlangten GIS geeignet war.

1. Technologie

Programmiersprache

Alle Funktionen, Objekte und Methoden die benötigt wurden, um das Geoinformationssystem zu erstellen, konnten mit der Programmiersprache C-Sharp entwickelt werden. Dementsprechend mussten keine weiteren Sprachen für die Anzeige von Karten eingesetzt werden (Nielsen, 2005).

Die dadurch entstandenen Vorteile sind, dass die RPGeol AG den geschriebenen Programmcode wiederverwenden sowie von der bereitgestellten Modularität Gebrauch machen kann. Zum Beispiel können alle definierten Klassen für die Erweiterung eines internen GIS genutzt werden, ohne sie verändern zu müssen.

In diesem Kontext entstanden keine Nachteile, da der Einsatz der Programmiersprache C-Sharp für die Unternehmung zum Alltag gehört.

ASP-Steuerelemente

Wie bei den durchgeführten Modulvergleichen ersichtlich ist, verwendet SharpMap für die Anzeige der Karten das von Microsoft vordefinierte Websteuerelement «ImageButton». Das Element dient beim Modul nicht nur für die Anzeige von Karten, sondern auch für die Funktionen, die Kartenansicht wechseln zu können (SharpMap, 2012d).

Durch den Einsatz eines vordefinierten Steuerelementes müssen keine Anpassungen am Element bei der Verwendung zukünftiger Versionen des Moduls vorgenommen werden. Zusätzlich sind vordefinierte Elemente ausführlich von Microsoft dokumentiert, was die

Anwendbarkeit gegenüber dem Programmierer vereinfacht (Microsoft Corporation, 2016). Ergänzend sind keine weiteren technischen Kenntnisse seitens der Unternehmung nötig.

Da vordefinierte Elemente in sich nicht verändert werden können, liegen die Nachteile in der begrenzten Funktionsmöglichkeit. Das bedeutet, dass spezielle Funktionen wie beispielsweise das Scrollen oder dynamische Navigieren durch die Karte nur mit selbsterstellten Steuerelementen bewerkstelligt werden können. Beim SharpMap-Modul existiert zum Beispiel eine Erweiterung für ein Ajax-Element, das diese Funktionalitäten zur Verfügung stellt (SharpMap, 2006). Im erstellten Web-GIS wurde allerdings das vordefinierte Element «ImageButton» angewendet.

Zusatzdateien

Für die Anzeige von Karten verlangt das ausgewählte Modul keine zusätzlichen Dateien, die in das Projekt eingebunden werden müssen. Alle Karten können direkt vom Dateisystem oder aus diversen Datenbanken gelesen und anschliessend angezeigt werden. (Nielsen, 2005)

Das Vermeiden solcher Dateien hat den Vorteil, dass Arbeitsschritte für die Erstellung und Verwaltung von zusätzlichen Kartendateien vermieden werden können. So müssen beispielsweise keine Eigenschaften von einem Shapefile in eine Zusatzdatei exportiert werden, um die darauf gespeicherten Karten anzeigen zu können. Der Zugriff auf die Kartendatei, in diesem Fall das Shapefile, erfolgt direkt. Deshalb war es möglich, mit der Datenbank die Datenintegration zu regeln und durch Pfadangaben auf darzustellende Kartendateien zu verweisen. Das heisst, damit die RPGeol AG neue Karten in das GIS einbinden kann, werden lediglich die entsprechenden Kartendateien benötigt.

Nachteile bezüglich dieses Kriteriums sind keine vorhanden, da der Einsatz von Zusatzdateien einen Mehraufwand für den Entwickler bedeutet.

2. Dokumentation

Funktionsübersicht

SharpMap stellt dem Anwender eine umfängliche Übersicht zur Verfügung, auf der alle Funktionen aufgelistet sind, die das Modul beinhaltet. Es werden ebenso alle Datenquellen angegeben, die mithilfe der Programmbibliothek angesprochen werden können (SharpMap, 2008).

Die vorhandene Funktionsübersicht ist gegenüber dem Entwickler ein Vorteil, zumal sofort sichtbar ist, ob ein verlangtes Kriterium bezüglich des Geoinformationssystem erfüllt werden

kann oder nicht. Dieser Vorteil wird für die Unternehmung unter anderem auch für zukünftige Entwicklungen am GIS nützlich sein.

Nachteile entstehen durch eine solche Zusammenfassung keine.

Tutorials

Das ausgewählte Modul bietet dem Entwickler detaillierte Instruktionen an, die zur Anwendung diverser Funktionen verhelfen. Im Weiteren sind in vielen verschiedenen Bereichen Tutorials vorhanden, was dem Anwender einen guten Überblick verschafft, wie das Modul funktioniert (SharpMap, 2012f).

Nebst der gut vermittelten Wissensbasis über den Einsatz unterschiedlicher Methoden und Funktionen konnte Zeit für das Lösen allfälliger Probleme eingespart werden, wie etwa bei der Frage, welche Methoden und Attribute für die Erstellung einer klickbaren Karte benötigt werden. Zudem gab es zu allen beim Funktionsumfang definierten Kriterien ein entsprechendes Tutorial, was eine Verminderung des Arbeitsaufwandes darstellte.

Nachteile bezüglich Tutorials gibt es nur, falls unvollständige Anleitungen angeboten werden. SharpMap bietet jedoch in sich vollständig abgeschlossene Dokumentationen an, die in diesem Fall nur Vorteile mit sich bringen.

Beispielprojekte

Für die Erstellung eines einfachen Geoinformationssystems besteht ein Beispielprojekt, das unter anderem in dieser Arbeit als Strukturgrundlage für die Entwicklung des GIS diente (SharpMap, 2017a). Überdies galten andere vorhandene Demoprojekte als weitere Informationsquellen, um einen Überblick vom praktischen Einsatz des Moduls zu erhalten.

Der Vorteil von vorhandenen Beispielprojekten liegt in der Wiederverwendbarkeit des dadurch zur Verfügung gestellten Programmcodes. Ferner fungierten sie nebst den angebotenen Tutorials als zusätzliche Informationsquelle.

Einzigster Nachteil bei Beispielprojekten kann sein, wenn Demoprojekte mit älteren Modulversionen nicht mehr kompatibel mit neueren Versionen sind. Allerdings existiert bei SharpMap keine derartige Situation, sodass im Bereich der Beispielprojekte für das Modul keine Nachteile aufgetreten sind.

Web-Beispiele

Laufende Web-Beispiele sind von SharpMap keine vorhanden. Diesbezüglich wurden die angebotenen Beispielprojekte hinsichtlich der Vorstellung betrachtet, wie ein fertiges System aussieht und eingesetzt wird.

Aufgrund der Tatsache, dass keine Beispiele vom Modul angeboten werden, haben sich dadurch keine Vorteile gegenüber dem Entwickler ergeben.

Der Nachteil in dieser Situation ist die fehlende Repräsentation des Einsatzes vom Modul im Web gegenüber dem Entwickler. Es ist nicht ersichtlich, wie ein GIS, das unter der Anwendung von SharpMap geschrieben wurde, im täglichen Arbeitsbereich aussehen könnte. Das entwickelte Web-Geoinformationssystem in dieser Arbeit wäre für diesen Zweck ideal gewesen.

3. Funktionsumfang

Datenzugriff

Wie die Entwicklung des geographischen Informationssystems gezeigt hat, kann das Modul die Dateien ShapeFile, ECW und JPG2000 verarbeiten und als Karten darstellen. Das bedeutet, dass SharpMap mit den von der RPGeol AG im Arbeitsalltag eingesetzten Dateitypen umgehen kann.

Es ergibt sich für die Unternehmung ein bedeutender Vorteil, dass das ausgewählte Modul diese Kartendateien direkt wiederverwenden kann, da keine Umwandlungen zwischen den einzelnen Typen gemacht werden müssen. Angenommen das Modul könnte nicht mit ECW Files umgehen. Demzufolge müssten diese Dateien zuerst ein entsprechendes Format, wie beispielsweise in eine JPG2000 Datei konvertiert werden, um sie korrekt im GIS zu verwenden.

Da das Modul mit allen beim Funktionsumfang definierten Dateien umgehen kann, entstehen keine Nachteile. Der einzige Nachteil, der beim Datenzugriff vorkommen kann, wäre falls gewisse Dateien nicht verarbeitet werden könnten.

Darstellung

SharpMap ermöglicht es, geometrische Formen sowie passende Überschriften darzustellen. Formen müssen entweder mittels angegebenen Koordinaten oder Shapefiles gezeichnet werden. Somit können beispielsweise auf der Datenbank gespeicherte geographische Daten, die die Koordinatenpunkte X und Y beinhalten, ohne spezielle

Transformationen dargestellt werden (SharpMap, 2012a). Indes können die selbstdefinierten Formen mit Textangaben gekennzeichnet werden, die individuell veränderbar sind. Die Überschriften der Shapefiles sind allerdings vorgegeben und nicht veränderbar.

Es ergeben sich mehrere Vorteile bezüglich der Darstellung, insofern für das Zeichnen diverser geometrischen Formen seitens der Unternehmung keine spezifischen Transformationen erfolgen müssen. Demzufolge können die Daten der einzelnen Datenquellen, wie vom Dateisystem oder den Datenbanken, direkt wiederverwendet und angesprochen werden. Darüber hinaus ist es nützlich, dass die eigens definierten Formen mit Überschriften gekennzeichnet werden können, die die RPGeol AG selbst vergibt. Eine solche von der Unternehmung vergebene Textangabe beinhaltet im entwickelten Geoinformationssystem beispielsweise den Namen einer Bohrung sowie die Anzahl vorhandener Dokumente, die sie hinterlässt. Eine Bohrung wird im GIS als Punkt mit einem entsprechenden Symbol dargestellt.

Aufgrund der erfüllten Bedingung in diesem Bereich sind keine Nachteile erkennbar, da die bereits bestehenden geographischen Daten, die auf den Datenbanken der Unternehmung abgespeichert sind, ohne Veränderungen wiederverwendet werden können.

Kartenlayer-Management

Zum Kartenlayer-Management gehörten die Bedingungen, dass der Benutzer manuell die Kartenansicht verändern kann. Dazu gehört das Zoomen sowie das Wechseln der Perspektive. Dieses Kriterium stellte für das ausgewählte Modul kein Problem dar (SharpMap, 2012d).

Die zwei Standardfunktionen sind für ein GIS elementar wichtig, weil die Nichterfüllung dieser Kondition bedeuten würde, dass die Karten nur aus einer Perspektive betrachtet werden könnten. Deshalb besteht der Vorteil in diesem Bereich aus dem Verändern der Kartenansicht nach Wunsch des Benutzers.

Nachteile erben sich in diesem Fall keine, da SharpMap den verlangten Bedingungen nachkommt.

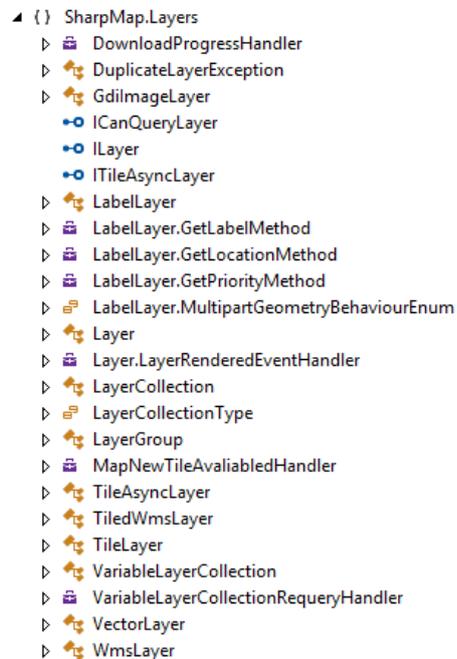
4. Modularität des GIS-Moduls

Standardbibliotheken

Die Standardbibliotheken von SharpMap sind unter dem Namen «SharpMap (Core + UI)» veröffentlicht (SharpMap, 2014). Sie beinhalten alle Standardfunktionen, wie beispielsweise die Anzeige der Darstellungsmöglichkeiten oder diverse Methoden für die Datenzugriffe. Die

Abbildung 53 zeigt als Beispiel die verschiedenen Arten der Kartenschichten, die mit dem Modul erstellt werden können.

Abbildung 53: Beispiel der Arten von Kartenschichten in SharpMap (SharpMap, 2017b)



Durch das Bereitstellen dieser Bibliothek entsteht gegenüber dem Entwickler der Vorteil, dass erkennbar ist, welche Grundfunktionen das Modul anbietet.

Nachteile existieren keine, solange eine gewisse Modularität gewährleistet ist. Wichtig für den Entwickler sind jedoch die vergebenen Namenskonventionen, damit er sich besser zurechtfinden kann.

Web-Bibliotheken

SharpMap bietet eine saubere Trennung zwischen den benötigten Bibliotheken für die Erstellung einer Webapplikation und den Standardbibliotheken. In diesem Modul beinhaltet die Web-Bibliothek den http-Handler, der für die Logik der Webapplikation verantwortlich ist sowie das Webserversteuerelement von Ajax, mit dem dynamische Veränderungen der Kartenansichten möglich sind (SharpMap, 2017d).

Diese Trennung der Bibliotheken für die Erstellung der Web-Applikation und den Standardbibliotheken dient dem Anwender nicht nur als gute Übersicht, sondern auch dazu, dass beispielsweise bei der Entwicklung einer lokalen GIS-Applikation nicht unnötige Bibliotheken eingebunden werden müssen.

Nachteile bezüglich dieser Modularität gibt es keine.

GDAL-Bibliotheken

Für die Anzeige von GDAL-Karten werden spezielle Bibliotheken benötigt (Gdal.org, kein Datum). SharpMap stellt dem Entwickler diese Bibliotheken in einer Erweiterung zur Verfügung, was bedeutet, dass keine zusätzlichen Module vom Anwender heruntergeladen werden müssen (SharpMap, 2017c).

Das Bereitstellen dieser Bibliotheken vermindert den Zeit- und Arbeitsaufwand bezüglich der Suche und Integration der verlangten GDAL-Module.

Im anderen Fall müsste der Entwickler zuerst wissen, welche Bibliotheken er benötigt und wo diese zu finden sind. Darüber hinaus werden die GDAL-Module direkt in das Projekt eingebunden, was ebenfalls eine Herausforderung darstellen würde.

Schlussfolgerung Evaluation Web-GIS Modul

Wie aus den genannten Beurteilungen hervorgeht, erfüllt das Modul SharpMap alle die im Kapitel «Analyse und Evaluation vorhandener Web-GIS Module» definierten Bewertungskriterien, mit Ausnahme des Kriteriums der Web-Beispiele. Trotzdem überwiegen die dargelegten Vorteile klar gegenüber den wenigen Nachteilen, die sich aus der Evaluation ergeben haben.

Überdies wurde in dieser Arbeit erfolgreich unter Anwendung dieses Moduls das verlangte Web-Geoinformationssystem entwickelt und anschliessend in die IT-Infrastruktur der Unternehmung integriert. Folglich lässt sich daraus schliessen, dass SharpMap für die Erstellung eines solchen Web-GIS bestens geeignet ist.

5.2 Evaluation zum erstellten Web-GIS

In diesem Abschnitt der Arbeit wird geklärt, ob das entwickelte Web-Geoinformationssystem die in der Problemstellung gestellten Anforderungen erfüllt. Diesbezüglich wird jeder Punkt im Anforderungsprofil betrachtet und erläutert.

1. Erweiterbarer Inhalt

Da das Web-GIS Modul auf die verschiedenen Datenquellen der Unternehmung direkt zugreifen kann, war es möglich, die Datenbank für die Datenintegration einzusetzen. Die Datenbank wurde so konzipiert, dass sich das entwickelte Geoinformationssystem anhand den Datenbankeinträgen dynamisch erstellt. Dadurch kann das System beliebig erweitert oder verringert werden. Beispielsweise können Karten durch Pfadangaben zum Dateisystem hinzugefügt werden.

2. Benutzermanagement

Wie bei der GIS-Datenbank im Kapitel «Drei-Schichten-Architektur» aufgezeigt, ist die Sicherheit durch Benutzergruppen gewährleistet. Somit kann der GIS-Verwalter durch Datenbankeinträge definieren, welche Benutzer welche «logischen Layer» sehen und verwenden können. Die Benutzergruppen sind aufgrund der flexiblen Datenbank beliebig erstellbar. Diese Bedingung wird dadurch erfüllt.

3. GIS-Funktionalitäten

Mit dem eingesetzten GIS-Modul können die Dateien Shapefile, ECW sowie JPG2000 direkt angesprochen und verarbeitet werden (SharpMap, 2008). So können die am meisten eingesetzten Dateitypen von der Unternehmung im System wiederverwendet werden.

Weiter haben die Evaluation des Moduls sowie die anschliessende Entwicklung des Systems gezeigt, dass geometrische Formen dargestellt werden können. Der Vorgang dazu wurde im Kapitel «Präsentationslayer» genauer erläutert. Nun können anhand von SQL-Abfragen die sogenannten «SpezialLayer» erstellt werden, die beispielsweise Messungen repräsentieren und mit einem entsprechenden Symbol gekennzeichnet sind.

Der dritte Bereich der Anforderung gilt ebenfalls als erfüllt, da der Benutzer sich mit der Toolbox durch die Karte navigieren und durch das Zoom die Kartenansicht verändern kann. Zusätzlich dient die erstellte Kartenauswahl dazu, dass der Anwender die Karten selbst auswählen kann, die er auf der Map anzeigen will.

Im letzten Teil der Anforderung steht die Kartensortierung im Vordergrund. Durch die bereitgestellte Funktionalität, die Reihenfolge der Karten mit Drag-And-Drop verändern zu können, gilt auch diese Bedingung als erfüllt.

4. Datenquellen

Mit dem ausgewählten Web-GIS Modul können die Datenquellen, also das Dateisystem und die Datenbanken, direkt angesprochen werden. Einerseits kann der GIS-Verwalter mit Pfadangaben auf Kartendateien verweisen, andererseits werden Informationen aus den Datenbanken mit SQL-Abfragen geladen. Aufgrund des vollumfänglichen Zugriffs auf die Quellen kann die Unternehmung direkt mit originalen Daten arbeiten.

5. Dokumenten-Zugriff

Durch die Erstellung der «SpezialLayer» sowie die dazugehörigen «DokumentenLayer» besteht für die Unternehmung die Möglichkeit, Daten wie beispielsweise Auswertungen oder

Beurteilungen, den Kunden zur Verfügung zu stellen. Die Dokumente können über die automatisch generierte Tabelle vom Benutzer entweder selektiv oder gesamthaft als ZIP-Datei heruntergeladen werden. Aufgrund dieser Funktion gilt auch die letzte Bedingung im Anforderungsprofil als erfüllt.

Schlussfolgerung Evaluation zum erstellten Web-GIS

Gemäss den aufgezeigten Punkten ist ersichtlich, dass alle im Anforderungsprofil definierten Bedingungen und Funktionalitäten mit dem entwickelten Web-Geoinformationssystem erfüllt wurden.

Schlussfolgerung und Ausblick

In dieser Arbeit wurde das Thema «Beurteilung der Anwendbarkeit vorhandener Web-GIS Module für die Rovina & Partner AG» behandelt. Dementsprechend wurde eine Evaluation über vorhandene GIS-Programmbibliotheken durchgeführt. Das Modul mit der besten Beurteilung diente im Anschluss als Entwicklungsgrundlage für die Erstellung eines Web-Geoinformationssystem. Dieses System ist nach dem Aufbau in die IT-Infrastruktur der Unternehmung integriert worden.

Anhand den aus der Evaluation gewonnenen Ergebnissen wurde für die Entwicklung des geographischen Informationssystem das Modul von SharpMap angewendet, da es hinsichtlich den gestellten Vergleichskriterien als das Zweckmässigste hervorging.

Die konkrete Ausarbeitung des Web-GIS hat anschliessend gezeigt, dass mithilfe des ausgewählten Moduls alle Anforderungen am System, die in der Problemstellung definiert wurden, erfüllt werden konnten. Für die Kartensortierung waren der Einsatz von Javascript und JQuery nötig. Alle anderen Funktionen konnten mit der Programmiersprache C# sowie den ASP-Steuerelementen (.NET) erstellt werden.

Das entwickelte Geoinformationssystem ist im Grunde genommen ein Datenintegrationssystem, das sämtliche in der RPGeol AG anfallenden Daten einbinden und abfragen kann. Die Datenzugriffe sind benutzerspezifisch geregelt. Des Weiteren musste die Unternehmung keine Anpassungen an der vorhandenen Datenstruktur, weder bei den SQL-Datenbanken oder im Dateisystem, vornehmen. Dies hat sich bei der Integration des Systems in die IT-Infrastruktur der Unternehmung als entscheidender Vorteil erwiesen.

Zusätzlich beinhaltet das Web-GIS einige interessante Funktionen. So können die internen Datenbanken mit SQL-Abfragen und die Dateisysteme mit Pfadangaben direkt angesprochen werden, was eine hohe Flexibilität hinsichtlich der Datenintegration bedeutet. Zudem wird dem Anwender durch die Möglichkeit der Kartensortierung eine ideale Steuerung der Map-Sichtbarkeit geboten. Überdies können spezifische Informationen von angeklickten Punkten abgefragt und anschliessend selektiv oder gesamthaft als ZIP-Datei heruntergeladen werden.

Aufgrund des erstellten Web-GIS können nun vorhandene Daten wie Auswertungen oder Beurteilungen den wichtigsten Infrastrukturbetreibern zur Verfügung gestellt werden. Der praktische Einsatz des Geoinformationssystem ist für Anfang September 2017 geplant, da bereits erste Kunden der RPGeol AG ihr Interesse am System geäussert haben.

Zukünftige Arbeiten am GIS

Die Entwicklung des Geoinformationssystems brachte einige Erkenntnisse mit sich. Darunter befinden sich Funktionen, die in Zukunft von Nutzen sein könnten und als nächste Schritte in das System implementiert werden.

Eine Funktion davon bezieht sich auf die Erstellung einer Anwendung, mit der die GIS-Datenbank (Datenintegration) auf eine einfache Art und Weise verwaltet werden kann.

In einem zweiten Schritt gilt der Fokus der Auswahl eines Koordinatenfensters durch den Benutzer, da er bisher nur einen Punkt anklicken kann. Ein solches Fenster ist eine explizit angegebene Fläche. Das Prinzip ist ähnlich wie bei der bereits erstellten Funktion, in der eine Fläche automatisch im Hintergrund erzeugt wird.

Die dritte Funktion gilt dem Dokumentenbereich. So sollte der Anwender zukünftig in der Lage sein, sich eine Dokumentenliste selbst zusammenzustellen, indem er verschiedene Punkte seiner Wahl nacheinander auswählen kann. Alle Dokumente der verschiedenen angeklickten Punkte werden im Anschluss in einer Liste zusammengefasst.

VI. Glossar

Ajax	«Asynchronous JavaScript And XML» wird nicht als eine Programmiersprache anerkannt. Mit Ajax können Webseiten, oder auch nur Teile davon, aktualisiert werden. Für diese Aktualisierung muss die Seite nicht neu geladen werden, da der Datenaustausch zwischen Client und Server asynchron im Hintergrund passiert (W3schools, kein Datum).
ASP.NET-Webserversteuerelement	Webserversteuerelemente sind in der ASP.NET Technologie Elemente, die auf einer Webseite angezeigt werden können. Darüber hinaus bieten sie dem Anwender vordefinierte Funktionen an und können mittels Programmcode serverseitig gesteuert werden (Microsoft Corporation, 2007a).
ECW	Enhanced Compressed Wavelet ist ein Dateityp im Rasterformat, der mittels GDAL-Bibliotheken gelesen werden kann (Gdal.org, 2016).
GDAL	Um Dateien mit Raster- oder Vectorformat lesen und verarbeiten zu können, werden spezielle Transformationen benötigt. GDAL ist eine Bibliothek, die solche Transformationen anbietet (Gdal.org, kein Datum).
GeoTIFF	GeoTIFF ist ein Dateityp im Rasterformat, der mittels GDAL-Bibliotheken gelesen werden kann (Gdal.org, 2016).
JPEG 2000	JPEG2000 ist ein Dateityp im Rasterformat, der mittels GDAL-Bibliotheken gelesen werden kann (Gdal.org, 2016).
Leaflet	Leaflet ist eine GIS-Programmibibliothek für die Erstellung interaktiver Karten. Die Bibliothek ist mit

Javascript geschrieben und obliegt einer OpenSource Lizenz (Leaflet, kein Datum).

LV03

LV03 ist ein Koordinatensystem der Schweiz. Es wurde im Jahre 1900 entwickelt und gilt bis heute als offizielle Grundlage für die Durchführung von verschiedenen Messungen (Bundesamt für Landestopografie swisstopo, kein Datum).

Shapefile

Ein Shapefile ist eine Datei, die geographische Daten beinhaltet. Sie können als Formen, Attribute, Koordinaten sowie Zusatzinformationen, die als Features bezeichnet werden, abgespeichert werden. Dieser Dateityp wurde von der Firma ESRI entwickelt. In der Praxis bilden mehrere Dateien zusammen das Shapefile. Um ein Shapefile zu bilden, müssen zumindest die drei Dateien mit den folgenden Typen involviert sein: .shp, .dbf und .shx (ESRI Inc., kein Datum e).

VII. Literaturverzeichnis

Ames, D. (1. Februar 2013). *DotSpatial - Documentation*. Abgerufen am 22. Mai 2017 von CodePlex:

<https://dotspatial.codeplex.com/wikipage?title=Roadmap&referringTitle=Documentation>

Ames, D. (9. Mai 2016). *DotSpatial*. Abgerufen am 15. Mai 2017 von CodePlex:

<https://dotspatial.codeplex.com/>

Bundesamt für Landestopografie swisstopo. (kein Datum). *LV03*. Abgerufen am 10. Juli 2017 von Swisstopo.admin.ch: <https://www.swisstopo.admin.ch/de/wissens-fakten/geodaesie-vermessung/bezugsrahmen/lokal/lv03.html>

Cadcorp. (2016a). *About Cadcorp*. Abgerufen am 5. Mai 2017 von Cadcorp:

<https://www.cadcorp.com/about-us/>

Cadcorp. (16. März 2016b). *Cadcorp announces new enterprise GIS software licensing model*. Abgerufen am 18. Mai 2017 von Cadcorp: <https://www.cadcorp.com/news-events/cadcorp-announces-new-enterprise-gis-software-licensing-model/>

Cadcorp. (kein Datum a). *.NET web mapping development environment*. Abgerufen am 5. Mai 2017 von Cadcorp: <https://www.cadcorp.com/products/developers-tools/web-map-add-ins/>

Cadcorp. (kein Datum b). *GIS Developer Tools*. Abgerufen am 18. Mai 2017 von Cadcorp: <https://www.cadcorp.com/products/developers-tools/>

Cadcorp. (kein Datum c). *Products*. Abgerufen am 18. Mai 2017 von Cadcorp:

<https://www.cadcorp.com/products/>

Chiesa, D. (22. Juni 2016). *DotNetZip 1.10.1*. Abgerufen am 6. Juli 2017 von Nuget:

<https://www.nuget.org/packages/DotNetZip/1.10.1/>

DotSpatial. (5. Juli 2016a). *DotSpatial - DemoWEB*. Abgerufen am 17. Mai 2017 von GitHub:

<https://github.com/DotSpatial/DotSpatial/tree/master/Source/Examples/DemoWEB>

DotSpatial. (11. Mai 2016b). *DotSpatial - DemoWEB default page*. Abgerufen am 17. Mai 2017 von GitHub:

<https://github.com/DotSpatial/DotSpatial/blob/master/Source/Examples/DemoWEB/default.aspx>

DotSpatial. (21. Dezember 2016c). *DotSpatial - DotSpatial.WebControls*. Abgerufen am 17. Mai 2017 von GitHub:
<https://github.com/DotSpatial/DotSpatial/tree/master/Source/DotSpatial.WebControls>

DotSpatial. (16. März 2017a). *DotSpatial - Issue 951*. Abgerufen am 22. Mai 2017 von GitHub: <https://github.com/DotSpatial/DotSpatial/issues/951>

DotSpatial. (19. Mai 2017b). *DotSpatial - Source*. Abgerufen am 22. Mai 2017 von GitHub:
<https://github.com/DotSpatial/DotSpatial/tree/master/Source>

DotSpatial. (28. Februar 2017c). *Tutorials - Documentation*. Abgerufen am 22. Mai 2017 von CodePlex:
<https://dotspatial.codeplex.com/wikipage?title=DotSpatial%20Tutorials&referringTitle=Documentation>

ESRI Inc. (kein Datum a). *About ESRI*. Abgerufen am 12. Mai 2017 von ESRI:
<http://www.esri.com/about-esri>

ESRI Inc. (kein Datum b). *Documentation - ArcGIS for Developer*. Abgerufen am 12. Mai 2017 von ArcGIS for Developer: <https://developers.arcgis.com/documentation/>

ESRI Inc. (kein Datum c). *Lizenzierung - ESRI Schweiz*. Abgerufen am 18. Mai 2017 von ESRI: <https://www.esri.ch/de/produkte/arcgis-for-desktop/lizenzierung>

ESRI Inc. (kein Datum d). *Pricing*. Abgerufen am 18. Mai 2017 von ArcGIS for Developers:
<https://developers.arcgis.com/pricing/>

ESRI Inc. (kein Datum e). *Shapefiles-Hilfe zu ArcGIS Online*. Abgerufen am 10. Mai 2017 von ArcGIS Doc: <http://doc.arcgis.com/de/arcgis-online/reference/shapefiles.htm>

Fletcher, W. (kein Datum a). *About: Easy GIS .NET*. Abgerufen am 5. Mai 2017 von Easy GIS .NET: <http://www.easygisdotnet.com/About.aspx>

Fletcher, W. (kein Datum b). *Developer Resources. Adding Map Markers to the TiledSFMap Web Control*. Abgerufen am 17. Mai 2017 von Easy GIS .NET:
<http://www.easygisdotnet.com/api/WebExample3.aspx>

Fletcher, W. (kein Datum c). *Developer Resources. Example Code and API Reference*. Abgerufen am 22. Mai 2017 von Easy GIS .NET: <http://www.easygisdotnet.com/api/>

Fletcher, W. (kein Datum d). *Developer Resources. Loading an Easy GIS .NET project in a Web Page*. Abgerufen am 17. Mai 2017 von Easy GIS .NET:
<http://www.easygisdotnet.com/api/WebExample1.aspx>

Fletcher, W. (kein Datum e). *Developer Resources. Using Custom Render Settings in the SFMap Web Control*. Abgerufen am 17. Mai 2017 von Easy GIS .NET:
<http://www.easygisdotnet.com/api/WebExample2.aspx>

Fletcher, W. (kein Datum f). *Easy GIS .NET 3.0 API Reference - SFMap Class*. Abgerufen am 17. Mai 2017 von Easy GIS .NET:
<http://www.easygisdotnet.com/api/ReferenceAPI.aspx>

Fletcher, W. (kein Datum g). *Live Demo 1*. Abgerufen am 22. Mai 2017 von Easy GIS .NET:
<http://www.easygisdotnet.com/demos/Demo1.aspx>

Fletcher, W. (kein Datum h). *Products*. Abgerufen am 22. Mai 2017 von Easy GIS .NET:
<http://www.easygisdotnet.com/Products.aspx>

Gdal.org. (28. Dezember 2016). *GDAL Raster Formats*. Abgerufen am 8. Juni 2017 von gdal:
http://www.gdal.org/formats_list.html

Gdal.org. (kein Datum). *GDAL - Geospatial Data Abstraction Library*. Abgerufen am 15. Juni 2017 von gdal: <http://www.gdal.org/>

JQueryUI. (kein Datum). *Sortable*. Abgerufen am 5. Juni 2017 von JQueryUI:
<https://jqueryui.com/sortable/>

Leaflet. (kein Datum). *Leaflet — an open-source JavaScript library for interactive maps*. Abgerufen am 25. Mai 2017 von Leafletjs: <http://leafletjs.com/>

MapAround. (4. Februar 2013a). *MapAround core - WebMapControl*. Abgerufen am 17. Mai 2017 von GitHub:
<https://github.com/gkrsu/maparound.core/tree/master/src/WebMapControl>

MapAround. (4. Februar 2013b). *MapAround core - WinFormsMapControl*. Abgerufen am 17. Mai 2017 von GitHub:
<https://github.com/gkrsu/maparound.core/tree/master/src/WinFormsMapControl>

MapAround. (18. Januar 2013c). *MapAround example - WebExample*. Abgerufen am 17. Mai 2017 von GitHub:
<https://github.com/gkrsu/maparound.example/tree/master/src/WebExample>

- MapAround. (4. Februar 2013d). *MapAround example - WebExample - workspace*. Abgerufen am 17. Mai 2017 von GitHub: <https://github.com/gkrsu/maparound.example/blob/master/src/WebExample/workspac e.xml>
- MapAround. (15. Juli 2013e). *MapAround example - WinFormsExample*. Abgerufen am 22. Mai 2017 von GitHub: <https://github.com/gkrsu/maparound.example/tree/master/src/WinFormsExample>
- MapAround. (15. Juli 2017). *MapAround example - Source*. Abgerufen am 17. Juli 2017 von GitHub: <https://github.com/gkrsu/maparound.example/tree/master/bin>
- MapAround. (kein Datum a). *MapAround Features*. Abgerufen am 17. Mai 2017 von MapAround: <http://maparound.ru/Features.aspx>
- MapAround. (kein Datum b). *MapAround Home*. Abgerufen am 17. Mai 2017 von MapAround: <http://maparound.ru/>
- MapAround. (kein Datum c). *MapAround Live Demo*. Abgerufen am 22. Mai 2017 von MapAround: <http://maparound.ru/Demo.aspx>
- MapAround. (kein Datum d). *MapAround Samples*. Abgerufen am 22. Mai 2017 von MapAround: <http://maparound.ru/Samples.aspx>
- Microsoft Corporation. (November 2007a). *ASP.NET-Serversteuerelemente*. Abgerufen am 7. Mai 2017 von Msdn.microsoft: <https://msdn.microsoft.com/de-de/library/bb386416.aspx>
- Microsoft Corporation. (November 2007b). *Toolbox-Standardsteuerelemente*. Abgerufen am 12. Juni 2017 von Msdn.microsoft: <https://msdn.microsoft.com/de-de/library/x8k61whf.aspx>
- Microsoft Corporation. (Oktober 2016). *ImageButton-Klasse (System.Web.UI.WebControls)*. Abgerufen am 20. Juni 2017 von Msdn.microsoft: [https://msdn.microsoft.com/de-de/library/system.web.ui.webcontrols.imagebutton\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/system.web.ui.webcontrols.imagebutton(v=vs.110).aspx)
- Nielsen, M. (25. November 2005). *Creating an Interactive Map in ASP.NET 2.0 Using SharpMap - CodeProject*. Abgerufen am 17. Mai 2017 von Codeproject: <https://www.codeproject.com/Articles/12457/Creating-an-Interactive-Map-in-ASP-NET-Using-S>

Rovina & Partner AG. (2. November 2011). Vortrag_ÜberwachungMattertal_20110119.
Varen, Wallis, Schweiz.

Rovina & Partner AG. (20. April 2017). Firmenpräsentation_rpgeol_20170413. Varen, Wallis,
Schweiz.

Searchcode. (kein Datum). *GdalRasterLayer.cs in SharpMap - source code search engine.*
Abgerufen am 13. Juli 2017 von Searchcode:
<https://searchcode.com/codesearch/view/2515369/>

SharpMap. (12. August 2006). *SharpMap - Geospatial Application Framework for the CLR -
Ajax map control.* Abgerufen am 20. Juni 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=AJAX%20map%20control&referringTitle=Extensions>

SharpMap. (17. Januar 2008). *SharpMap - Geospatial Application Framework for the CLR -
Features.* Abgerufen am 22. Mai 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=Features&referringTitle=Home>

SharpMap. (31. August 2012a). *SharpMap - Geospatial Application Framework for the CLR -
Add a primitive Geometry.* Abgerufen am 22. Juni 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=Add%20a%20primitive%20geometry>

SharpMap. (31. August 2012b). *SharpMap - Geospatial Application Framework for the CLR -
Add Label layer.* Abgerufen am 13. Juli 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=Add%20a%20Label%20layer&referringTitle=Add%20layers>

SharpMap. (31. August 2012c). *SharpMap - Geospatial Application Framework for the CLR -
Add Shapefile layer.* Abgerufen am 13. Juli 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=Add%20a%20ShapeFile%20layer&referringTitle=Add%20layers>

SharpMap. (31. August 2012d). *SharpMap - Geospatial Application Framework for the CLR -
Create a clickable map.* Abgerufen am 17. Mai 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=Create%20a%20clickable%20map>

SharpMap. (31. August 2012e). *SharpMap - Geospatial Application Framework for the CLR -
Custom Theme rendering.* Abgerufen am 13. Juli 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=CustomTheme>

SharpMap. (7. Dezember 2012f). *SharpMap - Geospatial Application Framework for the CLR - How to - Documentation*. Abgerufen am 22. Mai 2017 von CodePlex:
<https://sharpmap.codeplex.com/wikipage?title=How%20to...&referringTitle=Documentation>

SharpMap. (11. Februar 2014). *SharpMap - Geospatial Application Framework for the CLR - Release Version 1.1*. Abgerufen am 22. Mai 2017 von CodePlex:
<https://sharpmap.codeplex.com/releases/view/116326>

SharpMap. (26. Oktober 2015). *SharpMap - Geospatial Application Framework for the CLR*. Abgerufen am 17. Mai 2017 von CodePlex: <https://sharpmap.codeplex.com>

SharpMap. (31. Mai 2017a). *SharpMap - Examples DemoWebSite*. Abgerufen am 23. Juni 2017 von GitHub:
<https://github.com/SharpMap/SharpMap/tree/Branches/1.0/Examples/DemoWebSite>

SharpMap. (1. Juni 2017b). *SharpMap - Layers*. Abgerufen am 23. Juni 2017 von GitHub:
<https://github.com/SharpMap/SharpMap/tree/Branches/1.0/SharpMap/Layers>

SharpMap. (31. Mai 2017c). *SharpMap - SharpMap.Extensions*. Abgerufen am 8. Juni 2017 von GitHub:
<https://github.com/SharpMap/SharpMap/tree/Branches/1.0/SharpMap.Extensions>

SharpMap. (24. April 2017d). *SharpMap - SharpMap.Web - Web*. Abgerufen am 22. Juni 2017 von GitHub:
<https://github.com/SharpMap/SharpMap/tree/Branches/1.0/SharpMap.Web/Web>

Swiss Environment. (kein Datum). *Geot. Instrumentierung - Crackmeter*. Abgerufen am 17. Juli 2017 von Swiss Environment.

W3schools. (kein Datum). *AJAX Introduction*. Abgerufen am 15. Mai 2017 von W3schools:
https://www.w3schools.com/xml/ajax_intro.asp

Anhang

Anhang I: Methode initializeTree

```

//Baumstruktur
1 Verweis
private void initializeTree()
{
    //Counter für die Baumstruktur zu erstellen -> weiss genau in welcher Root-Node er sich befindet
    Int16 treeRootCounter = -1;
    //Counter für checkbox nummern bei onclick Methode -> für sortable liste
    Int16 treeLayerCounter = -1;
    //setzt eine Kommagetrennte Liste mit Nummern, für onclick events auf frontpage
    //(wird in javascript für sortable bei pageload verwendet)
    string treeLayerCBList = "";

    //Alle Themenbereiche LADEN
    foreach (Themenbereich tb in LayerInformationen.tbBereichArray)
    {
        treeRootCounter++;
        treeLayerCounter++;
        TreeNode tntb = new TreeNode(tb.thema);
        tntb.SelectAction = TreeNodeSelectAction.None;
        tntb.ShowCheckBox = false;
        this.layerTreeView.Nodes.Add(tntb);

        //Alle logischen Layer zum passenden Themenbereich laden
        foreach (LogischerLayer lLayer in LayerInformationen.lLayerArray)
        {
            if (isErlaubt(lLayer.idLLayer) && lLayer.idThemenbereich == tb.idThemenbereich)
            {
                //Nur Logische Layer Checkboxes müssen onclicks zugewiesen werden
                treeLayerCounter++;
                treeLayerCBList += treeLayerCounter.ToString() + ",";

                TreeNode tnllayer = new TreeNode(lLayer.lLayerName);
                tnllayer.SelectAction = TreeNodeSelectAction.None;
                tnllayer.ShowCheckBox = true;
                tnllayer.Value = lLayer.idLLayer.ToString();

                //Nur layer die klickbar sind erhalten ein Icon
                if (isKlickbar(lLayer.idLLayer))
                    tnllayer.ImageUrl = "~/Images/klickIcon1010.png";

                //id vom Layer auf tooltip wird benötigt in Javascript Methode -> für sortable list
                tnllayer.ToolTip = lLayer.idLLayer.ToString();
                tnllayer.Checked = lLayer.isDefaultLayer;
                this.layerTreeView.Nodes[treeRootCounter].ChildNodes.Add(tnllayer);

                //sortierliste initialisieren
                if (lLayer.isDefaultLayer)
                {
                    addSortableLayer(lLayer);
                }
            }
        }
    }
}
//end-foreach logischer Layer
}
//end-foreach Themenbereich

//Liste mit TreeCheckbox nummern setzen -> für onclicks auf clientseite
this.hiddenClickList.Value = treeLayerCBList;
}

```

Anhang II: Methode checkLayer

```
//onclick event für treeView checkboxes
function checkLayer(clickedCB) {
    var clickedCB = clickedCB;
    //id nummer auslesen aus stringId
    var idCheckbox = clickedCB.id.toString().replace(/[\d.]/g, '');
    var checked = clickedCB.checked;

    //Sortable liste holen
    var ul = document.getElementById("sortable");
    //richtige LogischeLayer Id holen
    var liValue = clickedCB.title.toString();

    //wenn checked wird sortable liste ergänzt
    if (checked) {
        //Listenelement erstellen und der liste hinzufügen
        var liAdd = document.createElement("li");
        //value -> Logische Layer ID, Text -> text der auf liste angezeigt wird
        var liText = document.getElementById("layerTreeViewt" + idCheckbox.toString()).textContent.toString();

        liAdd.appendChild(document.createTextNode(liText.toString()));
        liAdd.setAttribute("id", liValue.toString() + "SL");
        liAdd.setAttribute("value", liValue);
        liAdd.setAttribute("class", "ui-state-default");
        ul.appendChild(liAdd);
    }
    else {
        if (ul.getElementsByTagName("li").length != 1) {
            var liRemove = document.getElementById(liValue.toString() + "SL");
            ul.removeChild(liRemove);
        }
        else {
            clickedCB.checked = true;
        }
    }
}
}
```

Anhang III: Methode InitializeMap

```

2 Verweise
private SharpMap.Map InitializeMap(System.Drawing.Size outputsize)
{
    //Security -> wenn jemand die Layer austricksen will.
    doSecurityCheck();

    //Mappe mit neuer Größe erstellen
    SharpMap.Map map = new SharpMap.Map(outputsize);

    //Wissen, welcher der letzte Layer ist -> für Klickmethode, nur letzter Layer ist klickbar
    bool lastLayer = false;
    int lastLayerCount = 0;

    //Alle ausgewählten Layer durchlaufen und auf die Map laden.
    //alle logischen Layer durchlaufen
    foreach (int idLayer in this.layerReihenfolge)
    {
        lastLayerCount++;
        KonkreterLayer newKLayer = null;
        //Konkreter Layer setzen -> es wird immer einer gefunden da mindestens 1 ausgewählt sein muss.
        foreach (KonkreterLayer kLayer in LayerInformationen.kLayerArray)
        {
            //Check if KonkreterLayer zu Logischem Layer gehört und Zusatzcheck ob der LogischeLayer erlaubt ist.
            if (kLayer.idLayer == idLayer && isErlaubt(kLayer.idLayer))
            {
                newKLayer = new KonkreterLayer(kLayer);
                if (lastLayerCount == this.layerReihenfolge.Length)
                    lastLayer = true;

                //setzt jeweils den Konkreten layer
                setKonkreteLayer(ref map, newKLayer, lastLayer);
            }
        }
    }
    return map;
}

```

Anhang IV: Methode initializeLegende

```

1 Verweis
private void initializeLegende()
{
    foreach (LegendenBereich lBereich in LayerInformationen.LBereichArray)
    {
        //TableHeader laden
        TableHeaderRow bLayerHeaderRow = new TableHeaderRow();
        TableHeaderCell sLayerHeaderCell = new TableHeaderCell();
        sLayerHeaderCell.Text = lBereich.titel;
        bLayerHeaderRow.Cells.Add(sLayerHeaderCell);
        LegendTable.Rows.Add(bLayerHeaderRow);

        foreach (LegendenEintrag lEintrag in LayerInformationen.LEintragArray)
        {
            if (lBereich.idLegendBereich == lEintrag.idLegendBereich)
            {
                //Neue Row erstellen für Tabelle
                TableRow tRow = new TableRow();
                LegendTable.Rows.Add(tRow);

                TableCell tCellIcon = new TableCell();
                HtmlImage legendImage = new HtmlImage();
                legendImage.ID = "lImg;" + lEintrag.idLegendEintrag;
                legendImage.Src = "data:image/" + lEintrag.markerTyp + ";base64," +
                    LayerManager.getBase64StringFromImage(new Bitmap(lEintrag.markerPath));
                legendImage.Height = 20;
                legendImage.Width = 20;
                tCellIcon.Controls.Add(legendImage);

                TableCell tCellTitel = new TableCell();
                tCellTitel.Text = lEintrag.eintrag;

                TableCell tCellBeschreibung = new TableCell();
                if (!string.IsNullOrEmpty(lEintrag.beschreibung))
                    tCellBeschreibung.Text = lEintrag.beschreibung;

                //Zellen der Reihe hinzufügen
                tRow.Cells.Add(tCellIcon);
                tRow.Cells.Add(tCellTitel);
                tRow.Cells.Add(tCellBeschreibung);
            }
        }
    }
}

```

Anhang V: Methode createKlickBaum

```

//Baum für die Dokumentauswahl erstellen
1 Verweis
private void createKlickBaum(DocLayer[] docLayerKlick, SpezialLayer sLayer)
{
    if (docLayerKlick.Length != 0)
    {
        this.docLayerKlickList.Add(docLayerKlick);
        //TableHeader laden
        TableHeaderRow sLayerHeaderRow = new TableHeaderRow();
        //Setzt tooltip als hint -> Reihe für Download nicht berücksichtigen
        sLayerHeaderRow.ToolTip = "no";
        TableHeaderCell sLayerHeaderCell = new TableHeaderCell();
        sLayerHeaderCell.Text = sLayer.label;
        sLayerHeaderRow.Cells.Add(sLayerHeaderCell);

        TableHeaderCell sLayerHeaderCellCB = new TableHeaderCell();
        CheckBox headerCheckbox = new CheckBox();
        headerCheckbox.Checked = false;
        //00 für HeaderCheckbox, in welchem Klickarray
        headerCheckbox.ID = "00;" + (this.docLayerKlickList.Count - 1);
        headerCheckbox.Attributes.Add("onclick", "checkdLayer(this)");
        sLayerHeaderCellCB.Controls.Add(headerCheckbox);
        sLayerHeaderRow.Cells.Add(sLayerHeaderCellCB);

        KlickTable.Rows.Add(sLayerHeaderRow);

        //Jedes Control muss eindeutige ID erhalten -> hier Dokument mit ArrayPosition
        int idControl = 0;

        foreach (DocLayer docLayer in docLayerKlick)
        {
            //Neue Row erstellen für Tabelle
            TableRow tRow = new TableRow();
            KlickTable.Rows.Add(tRow);
            for (int i = 0; i < 4; i++)
            {

```

```

TableCell tCell = new TableCell();
//ID Zelle -> Reihe, 0 für Zelle, in welchem Klickarray, idControl
tCell.ID = i + ";" + "0" + ";" + (this.docLayerKlickList.Count - 1) + ";" + idControl;

switch (i)
{
    case 0:
        if (!string.IsNullOrEmpty(docLayer.beschreibung))
            tCell.Text = docLayer.beschreibung;
        else tCell.Text = "unbenannt";
        tRow.Cells.Add(tCell);
        break;
    case 1:
        if (!string.IsNullOrEmpty(docLayer.docTyp))
            tCell.Text = docLayer.docTyp;
        break;
    //hinzufügen Checkbox für herunterladen der Files
    case 2:
        CheckBox downloadBox = new CheckBox();
        downloadBox.Checked = false;
        //ID DownloadBox -> cb2 für Checkbox, in welchem Klickarray, idControl
        downloadBox.ID = "cb2;" + (this.docLayerKlickList.Count - 1) + ";" + idControl;
        tCell.Controls.Add(downloadBox);
        break;
    //hinzufügen Button für Anzeige von Files
    case 3:
        Button anzeigeButton = new Button();
        anzeigeButton.Text = "öffnen";
        anzeigeButton.Click += AnzeigeButton_Click;
        //ID Button -> b3 für AnzeigeButton, in welchem Klickarray, idControl
        anzeigeButton.ID = "b3;" + (this.docLayerKlickList.Count - 1) + ";" + idControl;
        tCell.Controls.Add(anzeigeButton);
        break;
    default:
        break;
}
tRow.Cells.Add(tCell);
}
idControl++;
}
}
}

```

Anhang VI: Methode setKonkreteLayer

```

1 Verweis
private void setKonkreteLayer(ref SharpMap.Map map, KonkreterLayer newKLayer, bool lastLayer)
{
    switch (newKLayer.idLayertyp)
    {
        //ECW Layer
        case 1:
            //Neue ecw Layer erstellen
            GDALManager.getECWLayer(ref map, newKLayer, "ecw");
            break;
        //ShapeFile Layer
        case 2:
            DTO_gisrpgool.Style newStyle = getStyle(newKLayer.idStyle);
            //Neue SHP Layer erstellen
            ShapeFileManager.getVectorLayer(ref map, newKLayer, newStyle);
            break;
        //Speziallayer
        case 3:
            DTO_gisrpgool.Style newStyleSLayer = getStyle(newKLayer.idStyle);
            //Neuen Speziallayer erstellen
            SpezLayerManager.getSpezialLayer(ref map, newKLayer, ref sLayerKlickArray,
                ref kLayerKlick, newStyleSLayer, lastLayer);
            break;
        case 4:
            //Neuer tif layer erstellen
            GDALManager.getECWLayer(ref map, newKLayer, "tif");
            break;
        case 5:
            //Neuer jpg2000 Layer erstellen
            GDALManager.getECWLayer(ref map, newKLayer, "jpg");
            break;
        case 6:
            DTO_gisrpgool.Style newStyleLblLayer = getStyle(newKLayer.idStyle);
            LabelManager.getLabelLayer(ref map, newKLayer, newStyleLblLayer);
            break;
        default:
            break;
    }
    GC.Collect();
}

```

Anhang VII: Methode getSpezialLayer

```

1 Verweis
public static void getSpezialLayer(ref SharpMap.Map map, KonkreterLayer kLayer,
    ref SpezialLayer[] sLayerKlickArray, ref KonkreterLayer kLayerKlick, Style kLayerStyle, bool lastLayer)
{
    try
    {
        SpezialLayer[] sLayerArray = SpezialLayerDB.getSpezialLayer(kLayer.path, kLayer.query, kLayer.queryCount);

        //Setzt für den User das Klickarray -> nur oberster Layer is Klickbar
        if (lastLayer && kLayer.isClickable)
        {
            sLayerKlickArray = sLayerArray;
            kLayerKlick = kLayer;
        }

        //VectorLayer erstellen
        SharpMap.Layers.VectorLayer sVectorLayer = new SharpMap.Layers.VectorLayer(kLayer.kLayerName);
        //Kollektion für Punktedarstellung erstellen
        Collection<GeoAPI.Geometries.IGeometry> geomColl = new Collection<GeoAPI.Geometries.IGeometry>();
        //Geometry Bibliothek mit ntsService holen um sie zu erstellen -> um Punkt erstellen zu können
        GeoAPI.GeometryServiceProvider.Instance = new NetTopologySuite.NtsGeometryServices();
        GeoAPI.Geometries.IGeometryFactory gf = GeoAPI.GeometryServiceProvider.Instance.CreateGeometryFactory();

        //Punkte erstellen
        foreach (SpezialLayer sLayer in sLayerArray)
        {
            if (sLayer.xCoord != null && sLayer.yCoord != null)
                geomColl.Add(gf.CreatePoint(new GeoAPI.Geometries.Coordinate(sLayer.xCoord.Value, sLayer.yCoord.Value)));
        }

        //Punkte auf GeometryFeatureProvider laden
        #region Datasource Speziallayer
        SharpMap.Data.Providers.GeometryFeatureProvider gp = new SharpMap.Data.Providers.GeometryFeatureProvider(geomColl);

        //Zwei Kolonnen um den custom style zu erstellen hinzufügen zu FeatureDataTable
        gp.Features.Columns.Add("labelColumn");
        gp.Features.Columns.Add("pathColumn");

        //Counter um die DataTable zu füllen mit werten -> ID vom Layer und Pfad
        int sLayerStyleCounter = 0;
        object[] styleProp = new object[2];
        //Für die Anzeige der Speziallayer - > ID und Marker
        foreach (SpezialLayer sLayer in sLayerArray)
        {
            if (sLayer.xCoord != null && sLayer.yCoord != null)
            {
                styleProp[0] = "\t \t " + sLayer.label.TrimEnd();
                if (!string.IsNullOrEmpty(sLayer.markerPath))
                    styleProp[1] = sLayer.markerPath;
                gp.Features[sLayerStyleCounter].ItemArray = styleProp;
                sLayerStyleCounter++;
            }
        }
        //Datasource setzen auf Layer
        sVectorLayer.DataSource = gp;
        #endregion Datasource Speziallayer

        //Style für Speziallayer setzen
        #region Style Speziallayer
        SharpMap.Rendering.Thematics.CustomTheme sLayerTheme = new SharpMap.Rendering.Thematics.CustomTheme(GetSLayerStyle);

        //Default marker setzen falls vorhanden
        if (!string.IsNullOrEmpty(kLayer.markerPath))
            sLayerTheme.DefaultStyle = GetDefaultSLayerStyle(kLayer.markerPath);

        //Max visible setzen
        if (kLayer.maxVisible != 0)
            sVectorLayer.MaxVisible = kLayer.maxVisible;
        sVectorLayer.MinVisible = kLayer.minVisible;
    }
}

```

```
//Theme auf layer setzen
svectorLayer.Theme = sLayerTheme;
#endregion Style Speziallayer

#region Label Speziallayer

//Labels nur anzeigen bei Zoom kleiner als 10000
map.Layers.Add(GetSpezialLabelLayer(kLayer.kLayerName + "Label", gp, kLayerStyle));
#endregion Label Speziallayer

map.Layers.Add(svectorLayer);
}
catch (Exception e)
{
}
}
```

Selbstständigkeitserklärung

Ich bestätige hiermit, dass ich die vorliegende Bachelorarbeit alleine und nur mit den angegebenen Hilfsmitteln realisiert habe und ausschliesslich die erwähnten Quellen benutzt habe. Ohne Einverständnis des Studiengangsleiters und des für die Bachelorarbeit verantwortlichen Dozierenden sowie des Forschungspartners, mit dem ich zusammengearbeitet habe, werde ich diesen Bericht an niemanden verteilen, ausser an die Personen, die mir die wichtigsten Informationen für die Verfassung dieses Berichts geliefert haben und die ich nachstehend aufzähle: Prof. Dr. René Schumann, Hermann Rovina

Siders, August 2017

Michael Varonier