Magisterarbeit

Titel der Magisterarbeit

ERP – Mobile Computing

Anbindung eines mobilen Clients an ein bestehendes ERP – System über Webservices

Verfasser
Stefan Schabel

angestrebter akademischer Grad
Mag. rer. soc.oec.

Wien, Jänner 2009

Studienkennzahl lt. Studienblatt:	066 926
Studienrichtung lt. Studienblatt:	Wirtschaftsinformatik
Betreuer:	Ao.Univ.Prof. Dipl.Ing. Dr.techn.Kurt Matyas

Inhalt

1. Begriffsdefinition

POLLEX-LC TaskCenter	ERP Stammsoftware
POLLEX-LC SmartCenter	Mobiler Client für das TaskCenter
ERP	Enterprise Ressource Planning
IP-Zahlen	Ingress Protection
KMU	Kleine und mittlere Unternehmen
MDE	Mobile Datenerfassung
GUI	Graphical User Interface
SOA	Service Orientated Architecture
WSDL	Webservice Description Language
UDDI	Universal Description, Discovery and Integration
XML	Extansible Markup Language
SOAP	Simple Object Access Protocol
RPC	Remote Procedure Call
RFID	Radio Frequency Identification
PDA	Personal Digital Assistent
CLR	Common Language Runtime
.NET CF	.NET Compact Framework
ASP.NET	Active Server Pages von .NET
.NET	Framework von Microsoft
ASMX	Webservice in ASP.NET

2. Einleitung

Enterprise Resource Planning hat sich aus den Schlagworten MRP (Material Requirements Planning) und MRP II (Manufacturing Resource Planning) heraus entwickelt. Wurden MRP bzw. MRP II speziell für Unternehmen, welche im produzierenden Sektor anzusiedeln sind, entwickelt, so ist ERP im Einsatz wesentlich branchenneutraler und in seinen Funktionen umfangreicher. Diese umfassen Stammdatenverwaltung von Mitarbeitern, Kunden, Lieferanten, etc., ebenso wie Funktionen für das Rechnungswesen, CRM usw..[1]

Der Einsatz von ERP-Software ist selbst in Klein- und Mittelbetrieben in den letzten Jahrzehnten nicht mehr wegzudenken gewesen. Durch eine komplexe Softwarelösung werden sämtliche Betriebsabläufe und Ressourcenplanungen in einer einzigen Applikation erfasst und für alle Betriebsbereiche zur Verfügung gestellt.

Was jedoch umfasst das Schlagwort „ERP“?
ERP-Software hat zum Ziel, die Prozesse der internen „Value-chain“, der Wertschöpfungskette, zu optimieren. Viele Unternehmen mussten, oder vielmehr konnten, in den 90er Jahren ihre Prozesse aufgrund von ERP-Software umstellen und so Arbeitsprozesse, welche nicht zur Wertschöpfungskette beitrugen, identifizieren und eliminieren.[2] Sind die Arbeitsprozesse eines Unternehmens einmal angepasst, so bieten ERP-Systeme eine unternehmensweite Datenbasis für alle Module, was Medienbrüche vermeidet und die Informationswege verkürzt.

ERP ist demnach keine genaue Definition von Prozessen und Funktionen, sondern vielmehr ein Überbegriff für ein Konzept, das Funktionalitäten, die zur Verbesserung und Unterstützung der wertschöpfenden Ablaufprozesse beitragen, bereitstellt.
Derartige Softwaresysteme können als Standardsoftware von verschiedenen Anbietern erworben, oder in Eigeninitiative als Individualsoftware selbst

[1] Vgl. Becker/Kugeler/Rosemann, 2000, S. 286
[2] Vgl. Norris/Hurley/Hartley/Dunleavy/Balls, 2002, S. 12

entwickelt oder in Auftrag gegeben werden. Der Trend in den letzten Jahrzehnten geht eindeutig in Richtung Standardsoftware. Einer der Hauptgründe dafür sind die enormen Kosten, die durch die Eigenentwicklung und auch durch die Wartung anfallen. Die meisten Individuallösungen werden nicht durch ein professionelles Softwareengineering unterstützt, sondern sind schlecht spezifiziert und werden durch mangelndes Betriebsverständnis der Programmierer nur bedingt den Anforderungen gerecht. Terminverzögerungen und eingeschränkte Funktionalität sind die Folgen. Hohe Kosten für die Weiterentwicklung bzw. Behebung von Fehlern sind an der Tagesordnung.[3] Diese Gründe lassen für den Einsatz einer standardisierten Software, welche durch einen professionellen Softwareentwicklungsprozess entstanden ist und durch eine Vielzahl von Kunden aus den unterschiedlichsten Branchen gestestet wurde, sprechen.
Ein weiteres Problem von *„Eigenbau“* - Lösungen ist, dass derartige Systeme mit der Zeit gewachsen sind. Dies bedeutet, in den meisten Fällen existieren Insellösungen für verschiedene Aufgabenbereiche ohne konsistente Datenbasis. Eine Datenübernahme von einer Software zur nächsten ist so meist gar nicht oder nur schwer möglich. Historische Daten können auf diese Weise verloren gehen. Ebenso hat diese Bildung von Softwareinseln zur Folge, dass Daten mehrfach eingegeben und somit auch mehrfach gepflegt werden müssen. Die Konsequenzen dieser redundanten Dateneingaben sind leicht zu erkennen. Neben dem zusätzlichen Aufwand sind inkonsistente Datensätze wohl nur schwer zu vermeiden.

2.1. Mobile Computing

Die Nutzung von mobilen Endgeräten zur Erlangung und Bearbeitung von Information ist in den letzten Jahren, sowohl für private, als auch für kommerzielle Zwecke, rasant angestiegen. Eine mögliche Definition von mobile Computing ist die Verwendung und Bereitstellung von Diensten für mobile Endgeräte.[4]
Die ortsunabhängigen Endgeräte verwenden kabellose Übertragungstechnologien für den Austausch von Daten.

[3] Vgl. Becker/Kugeler/Rosemann, 2000, S. 294
[4] Vgl. e-teaching.org

Diese gewonnene Mobilität der Benutzer ermöglicht die Nutzung von Services, die Informationen zum jeweils aktuellen Aufenthaltsort liefern. Diese neuen Technologien bringen natürlich weitere Probleme mit sich:[5]

- Datenschutzverletzungen:
  Durch die Vernetzung kann der Benutzer lokalisiert werden und somit können Bewegungsnachweise erstellt werden.
- Unsichere Datenübertragungen können von Dritten mitgelesen oder beeinflusst werden.
- Des Weiteren können Daten durch Verlust des mobilen Endgerätes verloren gehen.

2.2. Problemstellung und Zieldefinition

Ziel dieser Arbeit ist es, einen mobilen Client für die ERP-Lösung POLLEX-LC zu entwickeln.
Als Entwickler dieses mobilen Clients hat der Verfasser dieser Diplomarbeit die Möglichkeit, allen wesentlichen Bereichen einer professionellen Softwareentwicklung aktiv beizuwohnen und die dabei gewonnenen Erfahrungen hier einfließen zu lassen.

Ausgangspunkt war die Anfrage eines Kunden nach einer mobilen Auftragserfassungssoftware, die speziell an die Anforderungen dieses Kunden angepasst sein sollte.
Diese Anforderungen umfassten die Abfrage von aktuellen Stammdaten von Kunden und Artikeln. Aufgrund dieser Daten sollte eine Auftragserfassung mit Rabatt- und Skontovergabe direkt beim Kunden möglich werden. Gleichzeit sollte der Auftrag in Echtzeit in das Stammsystem eingepflegt werden.

Firmenintern wurde schließlich geprüft, ob eine derartige Individuallösung allgemein in der Softwarelösung POLLEX-LC abgebildet und somit als zusätzliches Modul, auch für andere Kunden, angeboten werden könnte.
Während dieser Überprüfung wurden Recherchen über bereits existierende mobile Softwarelösungen und deren Funktionsumfang betrieben, um die Funktionalität des neuen Moduls abzugrenzen.

[5] Vgl. Federath, 2001

Im Zuge dieser Überprüfung wurde die Aufgabenstellung noch etwas erweitert und schließlich hat man sich auf die Entwicklung einer Software für folgende Geschäftsfälle festgelegt:

- Auftragserfassung
- Inventur
- Wareneingangs- bzw. Warenausgangskommissionierung

Da diese Anwendung als offenes Projekt konzipiert ist, schließt es Weiterentwicklungen von zusätzlichen Anwendungsfällen nicht aus.

2.3. Methodik

Beginnend mit der Problemstellung durch eine Kundenanfrage wurde der Entwicklungsprozess dieses Softwaremoduls angeworfen.
Der erste Schritt ist die Erläuterung der grundlegenden Technologien, welche für die Durchführung der Entwicklungsarbeiten von Bedeutung sind.
Die Ist-Analyse der gegenwärtigen Situation soll ebenso Aufschluss über die am Markt verfügbare Hardware beziehungsweise Software, als auch einen Überblick über das Stammsystem, welches die Grundlage dieses Softwaremoduls stellt, geben.
Aufgrund dieser IST-Analyse werden Faktoren, die für die Durchführung des Entwicklungsprozesses wichtig sind, erarbeitet.
Der nächste Schritt ist die Erfassung und Beschreibung der notwendigen Usecases, die für die Durchführung der geforderten Geschäftsprozesse notwendig sind.
Da das Projekt, welches Thema dieser Arbeit ist, ein Teilprojekt innerhalb eines Großprojektes darstellt, ist die Systemarchitektur bereits vorgegeben und muss nicht mehr eigens entworfen werden.
Nachdem die Anforderungen definiert worden sind, folgt dieses Projekt dem Prototyping – Prinzip. Mehr zum Thema Prototyping im Kapitel 3.4

2.4. Beteiligte Institutionen

Entwickler und Publisher der ERP-Software POLLEX-LC, welche aktuell in über 10 Ländern in mehreren Sprachen zum Einsatz kommt.
Bei der Durchführung dieses Projektes wurde der Autor von Hr. Dr. Helmut Lexen, Geschäftsführer, sowie von Hr. Mag. Andreas Lexen, Projektleiter, betreut.
Die Erstellung dieser Arbeit betreute an der der Technischen Universität Wien, Hr. DI Dr. Kurt Matyas, Institut für Managementwissenschaften.

3. Grundlagen

In diesem Kapitel wird das, für die Erbringung des Projektziels notwendige, technische Grundlagenwissen vermittelt.

3.1. Webservices

3.1.1. Definition

In der modernen Welt der Informationstechnologie hat der Trend zur serviceorientierten Architektur (SOA) von Software bereits lange Einzug gehalten.
Webservices sind der Grundbaustein jeder serviceorientierten Architektur und werden über ein Netzwerk, in den meisten Fällen über das Internet, zur Verfügung gestellt. Sie sind Anwendungen, die auf den Schnittpunkten eines verteilten Systems liegen und verwenden XML als Übertragungsmedium für den Informationsaustausch.[6] Durch die Nutzung von definierten Standards wird eine Plattformunabhängigkeit für Webservices erreicht. Webservices sind jedoch nicht nur Interfaces, die von Personen benutzt werden können, sondern werden sie in vielen Fällen auch für die Maschine-Maschine-Kommunikation eingesetzt.

*„A Web service is a self-*describing, *self-containing software module available via a network, such as the Internet, which completes tasks, solves problems, or conducts transactions on behalf of a user or application.“*[7]

Papazouglu[8] gruppiert Web-Services in zwei große Lager. Webservices, welche einem einfachen Request/Response-Schema folgen und solche die komplexe Koordinationsaufgaben zwischen „*Inbound*“- und „*Outbound*“-Operationen vollziehen.[9]

[6] Vgl. Hammerschall, 2005, S. 110
[7] Papazoglou, 2008, S. 5
[8] Michael Papazoglou ist Direktor des INFOLAB an der University of Tilburg, Niederlande.
[9] Vgl. Papazoglou, 2008, S. 12

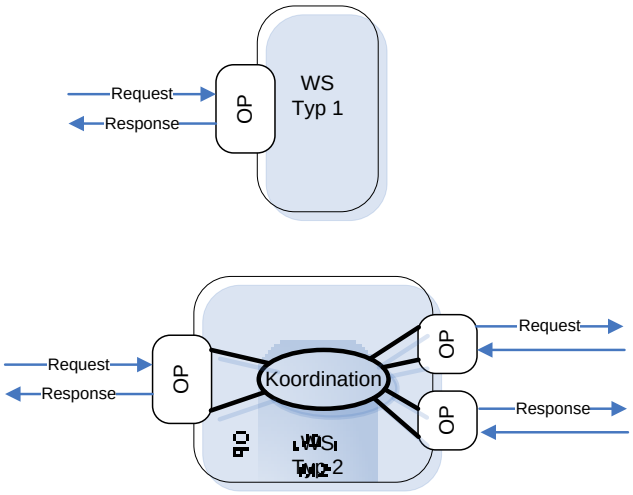


Abbildung 1: Webservice-typen nach Papazoglou[10]

Die lose Koppelung der serviceorientierten Architektur von Webservices basiert auf der Beziehung zwischen den Rollen *Provider*, *Requestor* und *Registry*. Die Operationen zwischen den Rollen innerhalb einer SOA werden in Abbildung 2: Operationen und Rollen in SOA dargestellt. [11]

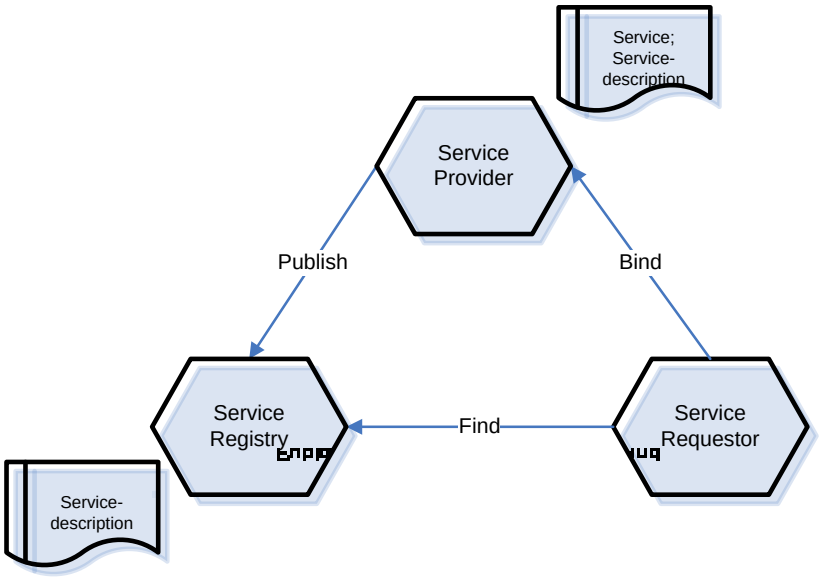


Abbildung 2: Operationen und Rollen in SOA[12]

[10] Abbildung entnommen und verändert aus: Papazoglou (2000:13)
[11] Vgl. Papazoglou, 2008, S. 24
[12] Abbildung entnommen und verändert aus: Papazoglou (2000:24)

Der Ablauf einer Webservicekommunikation erfolgt unter Anwendung von definierten WS-Standards (WSDL, UDDI, SOAP), in drei Schritten:[13]

- Nach der Entwicklung eines Services kann der *Service-Provider* den Service publizieren. Hierzu ist die Beschreibung der Applikation mit Hilfe der WSDL und die Registrierung bei einer *Service Registry* wie der UDDI Vorraussetzung.
- Nun ist es für Service-Clients (oder auch *Service-Requestor*) möglich, den Service über die *Service Registry* zu finden.
- Wurde ein Service gefunden, der den Anforderungen des *Requestors* genügt, wird der Service an den Client gebunden. Während des Bindens stellt der *Requestor* eine Verbindung mit dem Service zur Laufzeit her (*invoke*), indem er die technischen Informationen aus der Web-Servicebeschreibung nutzt.

3.1.2. Voraussetzungen einer WS-Kommunkation

Im Folgenden werden die eingeführten Standards, die für den Ablauf einer Webservicekommunikation notwendig sind, näher beschrieben.

WSDL

Die WSDL ist ein Dokument im XML-Schema, welches sämtliche Informationen bereithält, die ein Benutzer benötigt, um mit dem Service zu kommunizieren. Prinzipiell kann die WSDL beliebig editiert werden, jedoch wird sie meist automatisch generiert. Die XML-Struktur der WSDL ist in Abbildung 3: Aufbau eines WSDL Dokuments dargestellt.[14]

[13] Vgl. Papazoglou, 2008, S. 24
[14] Vgl. Schatten, 2006, S. 14

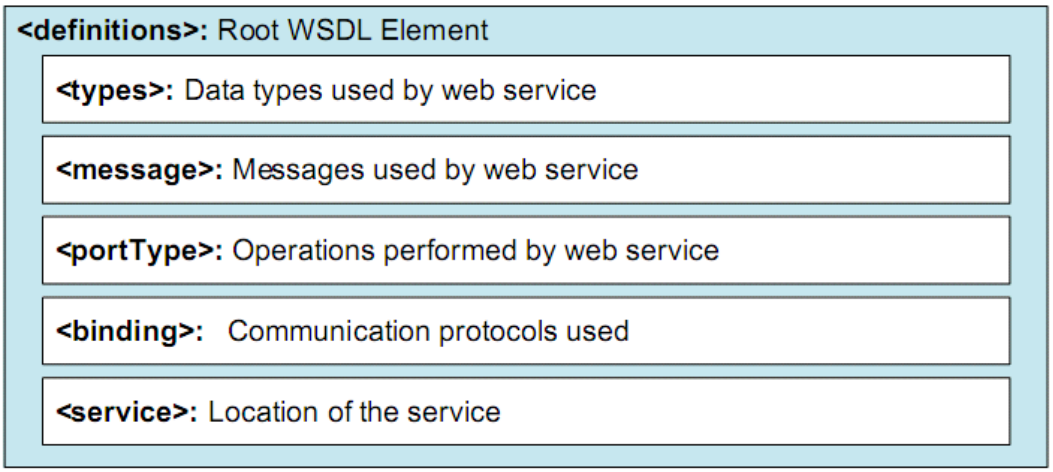


Abbildung 3: Aufbau eines WSDL Dokuments[15]

In dem Wurzelelement <**definitions**> werden die verwendeten Namen und Namespaces deklariert.[16]
Innerhalb der Wurzel werden weitere Elemente zur Beschreibung des Services verwendet.

<types>

Hier sind sämtliche Datentypen definiert, welche nicht im XML-Standard beschrieben sind.

<messages>

Definiert die Struktur der zu übertragenden Nachricht. Es kann natürlich sein, dass die Antwortnachricht von der Anfragenachricht differiert. In diesem Fall werden einfach 2 Nachrichten in der WSDL beschrieben. Auch kann eine Nachricht aus mehreren Teilnachrichten bestehen (<**parts**>).

<portType>

Die Methoden, die ein Webservice anbietet, werden hier beschrieben. Als Parameter werden die im Block <messages> definierten Nachrichten übergeben. Hierzu unterstützt WSDL vier Möglichkeiten:[17][18][19]

- Request/Response: Senden einer Nachricht des Clients und Antwort des Servers
- One-Way: Senden einer Nachricht des Clients ohne Antwort

[15] Abbildung entnommen aus: Schatten (2006:19)
[16] Vgl. Hammerschall, 2005, S. 116
[17] Vgl. Kappl/Kramler, S. 21
[18] Vgl. Schatten, 2006, S. 20
[19] Vgl. W3Schools – WSDL-Tutorial

- Solicit Response: Senden einer Nachricht des Servers und Antwort des Clients
- Notification: Senden einer Nachricht des Servers ohne Antwort

<bindings>
Nachdem die Operationen definiert worden sind, können hier deren Übertragungsprotokoll und das Format der Nachricht festgelegt werden. Üblicherweise wird hier SOAP gebindet, jedoch sind auch andere Protokolle, wie HTTP, FTP oder ähnliche möglich.[20].

<service>
Hier erfolgt die Angabe sämtlicher Informationen, um den Service aufrufen zu können, wie zum Beispiel die Netzwerkadresse.[21]

UDDI

Für die Verbreitung von Webservices hat sich UDDI (Universal Description, Discovery and Integration) als Verzeichnisdienst durchgesetzt. Im Wesentlichen muss dieser Dienst drei Anforderungen genügen:[22]

- Auffinden des Webservices
- Das Beschreiben des Webservices
- Die Art der Kommunikation des Webservices

Diese Operationen sollen im gesamten Netzwerk (Internet) verfügbar sein. Aus diesem Grund wurde ein übergreifender Verzeichnisdienst wie UDDI eingeführt.
Die Struktur eines solchen Verzeichnisdienstes ist im UDDI-Schema enthalten und erlaubt die Modellierung der Informationen zu Webservices. Bei dieser Modellierung wird zwischen folgenden Entitäten unterschieden:[23][24]

- *Business Entity* oder auch *White Pages* beinhalten Informationen über das Unternehmen oder die Abteilung eines Unternehmens, welche den Webservice anbietet.
- *Business Service* oder auch *Yellow Pages* kategorisieren sämtliche Webservices nach ihren Business Entities.

[20] Vgl. Hammerschall, 2005, S. 116
[21] Vgl. Hammerschall, 2005, S. 117
[22] Vgl. Papazoglou, 2008, S. 175
[23] Vgl. Schatten, 2006, S. 25
[24] Vgl. Hammerschall, 2005, S. 118

- *Binding Template* oder *Green Pages* schließlich beschreiben den technischen Zugriff auf einen speziellen Service.
- Mit Hilfe des *taxonomy Model* (*tModel*) können angebotene Webservices innerhalb des Verzeichnisdienstes aufgrund ihrer angebotenen Schnittstellen (bzw. Methoden) kategorisiert werden.

Für einen vollständigen UDDI-Eintrag eines Webservices sind alle vier oben genannten Einträge durchzuführen.

SOAP

Wurde ein Webservice schließlich erfolgreich in einem UDDI-Verzeichnis registriert und hat sich ein Client die Verbindungsinformation von dort geholt, so kann die Kommunikation mit dem Service beginnen.
Der Informationsaustausch von Webservices wird über das SOAP-Protokoll durchgeführt.
Wobei SOAP im eigentlichen Sinne kein Protokoll darstellt, sondern lediglich eine standardisierte Strukturierung von Nachrichten vorgibt und gängige Übertragungsprotokolle (HTTP, FTP, SMTP,…) zum Transfer nutzt.[25]
W3C[26] definiert SOAP als XML-basierendes Protokoll für den Austausch von Informationen innerhalb einer dezentralisierten Umgebung, welches drei Teile beinhaltet. Siehe Abbildung 4: Bestandteile von SOAP

[25] Vgl. Hammerschall, 2005, S. 118
[26] World Wide Web Consortium: http://www.w3.org/

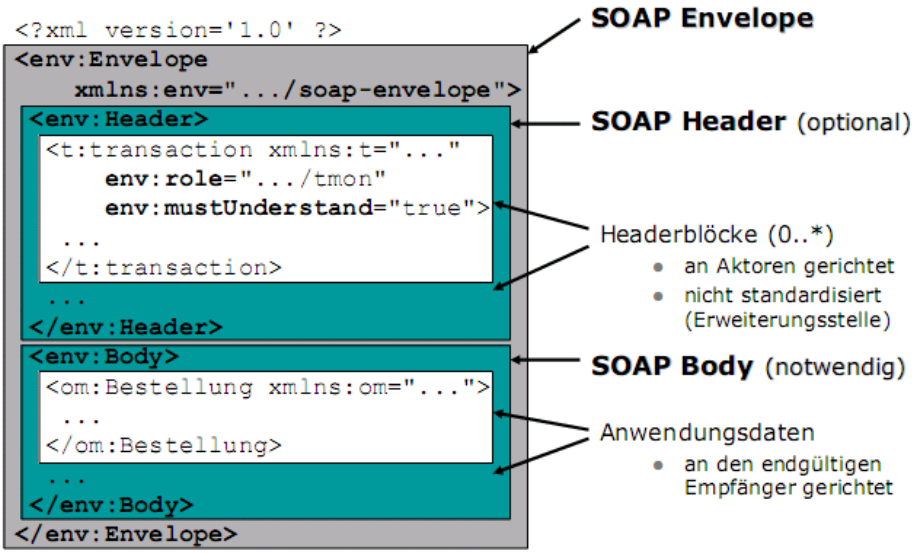


Abbildung 4: Bestandteile von SOAP[27]

SOAP-Envelope:
Die gesamte Nachricht wird von einer Hülle, der SOAP-Envelope, umgeben. Sie ist zwingend für jede Nachricht mit dem XML-tag <*envelope*> zu deklarieren.
Ebenso werden in der Envelope die Namensräume und das Encoding festgelegt.[28] Namesräume werden für SOAP über einen URI definiert. (http://schemas.xmlsoap.org/sopa/envelope)
Die Namensräume und das Encoding müssen für den *Requestor*, wie auch für den *Provider*, dieselben sein, um eine erfolgreiche Nachrichtenübertragung zu erlangen.

SOAP-Header:
Dieser Teil einer SOAP-Nachricht ist nicht zwingend erforderlich und kann für spezielle Verarbeitungsmodalitäten benutzt werden. Innerhalb des <*header*>-tags können zusätzliche Informationen für Security, Authentifikation, Bezahlungsmodalitäten, Adressierungen und dergleichen gespeichert werden. Einfache Webservicenachrichten besitzen häufig keine <*header*>-deklaration.[29]

SOAP-Body:

[27] Abbildung entnommen aus: Kappl (2003:11)
[28] Vgl. Pötscher, 2007, S. 13
[29] Vgl. Pötscher, 2007, S. 13

Der <*body*>-tag beinhaltet schlussendlich die eigentliche Nachricht für die Übertragung und ist für die Gesamtheit einer SOAP-Nachricht zwingend notwendig.[30]
Prinzipiell wird zwischen zwei Kommunikationsmodellen unterschieden. Zum Einen, RPC-style und zum Anderen document-style Webservices.
Bei RPC-style Webservices funktioniert der Service wie bei einem RPC-Call.
Innerhalb der SOAP-Nachricht werden Methodennahme und Eingabeparameter an eine Verarbeitungslogik auf einem Server übergeben. Das Ergebnis der Verarbeitungslogik wird als Antwortdokument in XML zurückgeliefert.[31]
Bei document-style Webservices besteht die Möglichkeit, durch Festlegen der Namensräume und des Encodings ganze XML-Dokumente als Nachricht zu übermitteln.[32]
Der SOAP-Standard gibt keinerlei Bedingungen für das Aussehen und die Struktur des SOAP-Bodys vor.

Attachements

Die Übertragung von binären Daten ist im SOAP-Body nicht vorgesehen. So müssen diese binären Daten zunächst in ein XML-Schema codiert werden. Bei großen Datenmengen ist dies jedoch zu aufwendig. Es wurde daher nach Lösungen für Anhänge (Attachements) an SOAP-Nachrichten gesucht. Derzeit sind zwei gängige Lösungen im Einsatz:
MIME-Attachements (Multipurpose Internet Mail Exentension), welche schon bei E-Mails zum Einsatz kommen und die von Microsoft .NET unterstützten DIME-Attachements (Direct Internet Message Encapsulation).[33]

3.2. Framework

Für die Entwicklung des POLLEX-LC TaskCenters wurde das Microsoft .NET Framework mit der Programmiersprache C# verwendet. In Abbildung 5: .NET Framework Konzept ist das Framework schematisch aufbereitet.

[30] Vgl. Pötscher, 2007, S. 13
[31] Vgl. Papazoglou, 2008, S. 135
[32] Vgl. Papazoglou, 2008, S. 137
[33] Vgl. Pötscher, 2007, S. 15 ff.

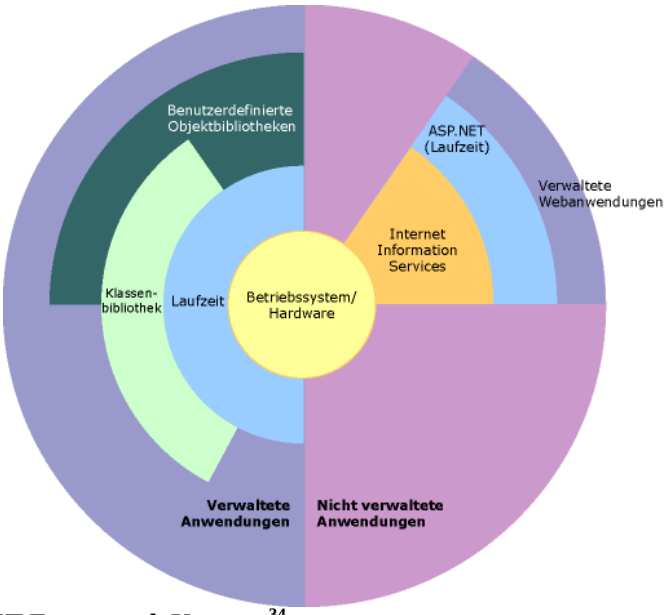


Abbildung 5: .NET Framework Konzept[34]

Prinzipiell umfasst das Framework zwei generelle Bereiche, die Common Language Runtime und die Base Class Library.

Common Language Runtime (CLR): In Microsoft .NET hat der Entwickler die Möglichkeit, zwischen mehreren Programmiersprachen, wie zum Beispiel C# oder Visual Basic, zu wählen. Der Grund dafür ist die CLR, die Laufzeitumgebung, die von allen Sprachen gleichermaßen benutzt wird. Dies wird über den MSIL-Code (Microsoft Intermediate Language), einer Zwischensprache, bewerkstelligt. Die Programmiersprache benötigt lediglich einen Compiler, der den Benutzercode in diese Zwischensprache übersetzt, welche von der CLR interpretiert werden kann. Dies bedeutet, dass unterschiedliche Klassen verschiedene Programmiersprachen besitzen können und dennoch zueinander vererbbar sind.[35] Zusätzlich sind in der CLR mehrere Dienste implementiert, welche für die Thread- und Speicherverwaltung, zur Sicherung des Codezugriffs usw., zuständig sind.[36]

[34] Abbildung übernommen aus: MSDN – Konzeptionelle Übersicht über .NET Framework
[35] Vgl. Baghi S.3
[36] Vgl. MSDN – Webservices mit .NET Framework 2.0 und Visual Studio 2005

Base Class Library: Die über 100 Namensräume umfassende Klassenbibliothek des .NET Frameworks beinhaltet sämtliche Typen, die vom Entwickler verwendet werden können.[37]
Als Programmierschnittstelle werden innerhalb der Klassenbibliothek API's zu unterschiedlichen Themen, wie Entwicklung der GUI oder Dateizugriff, angeboten.[38]

3.2.1. .NET Compact Framework

Für die Implementierung des SmartCenters wird das Microsoft .NET Compact Framework zum Einsatz kommen. Aufgrund des großen Speicherplatzbedarfes des .NET Frameworks, welches wahrscheinlich die Kapazitäten vieler mobiler Geräte übersteigen würde, hat Microsoft eine „kleinere" Variante des Frameworks, speziell für die Entwicklung von Software für PDA's, Smartphones und dergleichen, auf den Markt gebracht.

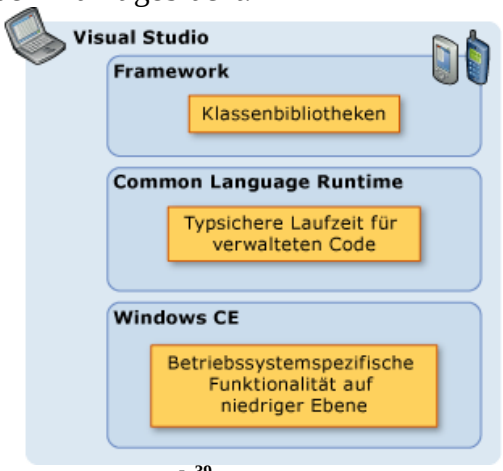


Abbildung 6: .NET Compact Framework[39]

In Abbildung 6: .NET Compact Framework ist die Architektur des .NET Compact Frameworks grafisch aufbereitet.
Die Kernfunktionen beider Frameworks sind identisch:[40]

[37] Vgl. MSDN – Webservices mit .NET Framework 2.0 und Visual Studio 2005
[38] Vgl. Pötscher, 2007, S.52
[39] Abbildung entnommen aus: MSDN - .NET Compact Framework - Architektur
[40] Vgl. Yao/Durant, 2004, S. 76

- Ein gemeinsamer Pool von Typen, der die Interoperabilität zwischen verschiedenen Programmiersprachen unterstützt
- Ausführbare Dateien, die mit Unterstützung der Intermediate Language zur Laufzeit in Maschinenbefehle übersetzt werden
- gleiches Speichermanagement

Microsoft hat für das CF eine Reduktion der möglichen Programmiersprachen durchgeführt. So können derzeit vom .NET CF nur C# und Visual Basic verarbeitet werden und nicht, wie beim kompletten .NET Framework, über 20 verschiedene.[41]

Windows CE: Das Windows CE Betriebssystem ist die Grundlage für das .NET CF. Dafür wurden einige Typen der Klassenbibliotheken überarbeitet und für den Einsatz auf Windows CE – Geräten angepasst.
CLR: Die CLR wurde ebenfalls für den Gebrauch auf mobilen Geräten überarbeitet. So wurde sie hinsichtlich geringeren Arbeitsspeicherbedarfes und effektiveren Stromverbrauchs optimiert.
Zusätzlich ist ein Zwischenlayer zwischen der CLR und Windows CE implementiert worden, in dem Dienste, die vom Framework gefordert werden, zu Windows CE-Diensten zugewiesen werden können.[42]

3.2.2. Webservices in .NET

Ein wesentlicher Bestandteil des .NET Frameworks ist ASP.NET (Active Server Pages). Durch sie wird dem Programmierer ermöglicht serverseitige Webanwendungen zu entwickeln. Zur Erstellung einer ASP.NET – Anwendung wird weiters ein Hosting-Server benötigt, üblicherweise wird dafür der Internet Information Server (IIS) von Microsoft verwendet. Für einen Webservice wird eine Datei mit der Endung *.asmx* erstellt. Durch diese Endung ist erkennbar, dass es sich bei diesem Service um einen ASP.NET Webservice handelt. Innerhalb dieser Datei wird dann lediglich auf den eigentlich C# - Code verwiesen.[43]
Die ASMX-Engine, welche direkt in die Laufzeitumgebung von ASP.NET eingebunden ist, übernimmt die Aufgaben der XML-Serialisierung und der

[41] Vgl. Yao/Durant, 2004, S. 64
[42] Vgl. MSDN - .NET Compact Framework - Architektur
[43] Vgl. Baghi, 2006, S. 4

Request/Response – Methoden zur Übertragung von SOAP-Nachrichten über HTTP.[44]
Für die Erstellung einfacher Webservices sind die Standardklassen des .NET Frameworks in jedem Fall ausreichend. Zur Implementierung komplexerer Services wurden von Microsoft die Webservice Enhancements (WSE) entwickelt. Dieses Addon existiert bis zu diesem Zeitpunkt in seiner dritten Auflage (WSE 3.0) und ermöglicht dem Entwickler echtes SOAP-Messaging. Dadurch ist es möglich, sich vom HTTP-Protokoll zu trennen und Alternativen zu verwenden. Weiters werden durch die Enhancements zusätzliche Sicherheitsroutinen zur WS-Security, WS-Trust, WS-SecureConversation oder WS-SecurityPolicy angeboten.[45]

3.2.3. Sicherheit

Das SmartCenter verwendet aus Gründen der Performance keine Sicherheitsroutinen der Webservice Enhancements. Da zum Beispiel ein SSL-Handshake bei jedem Serviceaufruf die ganze Applikation schwerfällig werden ließe, wird für die Sicherheit ein anderer Weg eingeschlagen. Bei erfolgreichem *Login* bekommt der Client einen verschlüsselten *ConnectionString* zurück. Mit Hilfe dieses verschlüsselten Strings ist der Client in der Lage, Webservices ohne zusätzliche Sicherheitsroutinen aufzurufen. Durch diese Art des Webserviceaufrufes ist es auch nicht notwendig, mögliche Firewalls für jeden Webservice zu konfigurieren, da die Kommunikation ausschließlich über den HTTP-Port stattfindet.

3.2.4. Performance

Zu dem Zeitpunkt als der Autor die ersten Tests bei der Entwicklung mobiler Clients in .NET durchführte, war Breitband-Anbindung bei mobilen Geräten kaum vorhanden beziehungsweise nicht rentabel. Die ersten Tests liefen dabei über GPRS, welches sich als Flaschenhals bei der Übertragung von größeren Datenmengen herausstellte. Auch minderten häufige Webservice-Calls die Performance der Applikationen beträchtlich.

[44] Vgl. Baghi, S. 4
[45] Vgl. Baghi, S.4

Heute existieren bereits erschwingliche Highspeed-Anbindungen, welche den Datenfluss, der durch das SmartCenter produziert wird, ohne größere Probleme bewältigen können.

3.3. Datenübertragung

Eine notwendige Grundlage für eine mobile Anwendung, die einen Informationsaustausch zu ihrem Stammsystem benötigt, ist die Technologie der Datenübertragung. Die Vorraussetzung der Mobilität der Anwendungssoftware lässt auf den Einsatz von „kabelloser" Übertragung schließen.
Da *W-LAN* aufgrund der maximalmöglichen Reichweite eingeschränkt ist, kommt diese Übertragungstechnik nur für den Einsatz im Lager in Frage. Dies bedeutet, dass für die mobile Auftragserfassung nur die derzeit gängigen mobilen Internettechniken der Handynetzbetreiber zur Verfügung stehen.
Folgende Übertragungstechniken werden über das Handynetz weitgehend flächendeckend angeboten:[46]
UMTS (Universal Mobile Telecommunications System) wurde erstmals 2003 eingesetzt und ist seither stets weiterentwickelt worden.
Breitband-UMTS (HSDPA – High Speed Downlink Packet Access) entwickelte sich aus dem UMTS-Netz heraus mit erhöhter Datenübertragungsrate. HSDPA gibt es bereits in drei Ausbaustufen, wobei die Letzte 2008 in Betrieb ging.
GPRS (General Packet Radio System) ist eine etwas behäbige Technik zur Datenübertragung innerhalb von GSM-Netzen.
EDGE ist eine Weiterentwicklung von GPRS mit erhöhten Durchsatzraten.

Ein Vergleich zwischen den unterschiedlichen Übertragungsraten ist in Tabelle 1: Übertragungsraten dargestellt.

Tabelle 1: Übertragungsraten[47]

Übertragungstechnik	Uplink (Senden)	Downlink (Empfangen)
GPRS	1-2 kb/s	3-5 kb/s
EDGE (EGPRS)	10-12 kb/s	15-22 kb/s
UMTS	7,5 kb/s	40 kb/s
HSDPA (3. Ausbaustufe)	3,6 mb/s	7,2 mb/s

[46] Vgl. Welt online
[47] Vgl. Das Elektro-Kompendium

3.4. Prototyping

3.4.1. Historie

In den frühen 80iger Jahren wurde im Bereich der Softwareentwicklung vehement nach neuen Methodiken gesucht, da immer häufiger Differenzen zwischen den Anforderungen und dem Endprodukt auftraten. Das Schlagwort Prototyping wurde zu dieser Zeit erstmals als Alternative zu den herkömmlichen klassischen Entwicklungsstrategien diskutiert. Diese klassischen Designkonzepte, wie zum Beispiel das Wasserfallmodell, zeichneten sich zumeist durch klar definierte Arbeitsschritte aus, welche jedoch zu starr konzipiert waren und den tatsächlichen Endverbraucher der Software nur zu Beginn der Entwicklungsphase mit einbezogen hat.
Diese genau abgegrenzten Arbeitsschritte hatten den Vorteil, gut strukturiert zu sein, und damit waren sie genauer zu kalkulieren und zu controllen.
Fehler in der Analysephase wurden jedoch erst spät entdeckt und konnten mitunter auch die gesamte Software unbrauchbar machen.
Das Prototyping sollte schließlich Abhilfe schaffen und die Kommunikation zwischen den Stakeholdern verbessern und intensivieren.

Hierbei darf das Prototyping nicht zwangsläufig als eigenständiges Konzept gesehen werden, vielmehr kann es ebenso als begleitendes Instrument zu anderen Entwicklungsstrategien verwendet werden.

3.4.2. Ansatzmethoden und Typen

Es existieren drei grundlegende Ansätze des Prototypings:
Throw-away – Ansatz:
Hier wird ein lauffähiger Prototyp entwickelt, welcher zwar noch nicht den vollen Funktionsumfang besitzt, jedoch durchaus die grundlegenden Einsatzbereiche bereits abdeckt. Dieser entwickelte Prototyp wird dem Auftraggeber zu Testzwecken übergeben. Wie der Name dieses Ansatzes bereits schließen lässt, wird dieser Prototyp völlig verworfen und dient lediglich als Lernphase für das spätere Endprodukt.[48]
Inkrementeller Ansatz:
Beim inkrementellen Ansatz wird zunächst an einem stabilen Programmkern gearbeitet, welcher als Grundlage für die weitere Entwicklung verwendet wird.

[48] Vgl. Sattler/Schmied

Nach und nach werden schließlich durch Simulationen alle fehlenden Funktionen implementiert. Das Problem hierbei ist, dass sich das Gesamtsystem durch neu hinzugefügte Programmteile maßgeblich ändern kann, sodass konzeptuelle Änderungen am Programmkern durchgeführt werden müssen.[49]

Evolutionärer Ansatz:

Sämtliche Architekturansätze sind zu keiner Zeit der Entwicklung festgelegt. Somit können sie jederzeit an neue Anforderungen angepasst werden. Der Prototyp, der durch diesen Ansatz entwickelt wird, entspricht dann dem endgültigen Produkt.[50]

Für diese unterschiedlichen Ansätze der Prototypingmethode kommen zwei verschiedene Prototyparten zum Einsatz:[51]

- **Demonstrationsprototyp**: Beim Demonstrationsprototyp oder auch horizontaler Prototyp handelt es sich um einen Prototyp, bei dem das Hauptaugenmerk auf das User-Interface gelegt wird. Er wird zur Veranschaulichung der Prozessabläufe und der Handhabung eingesetzt. Diese Art von Prototyp kommt zumeist in Verbindung mit der Throw-away-Methode zum Einsatz.
- **Funktionaler Prototyp:** Im Gegensatz zum Demonstrationsprototyp wird beim funktionalen Prototyp, oder auch vertikaler Prototyp, mehr auf die Funktionalität eines bestimmten Programmmoduls Bezug genommen.

In Abbildung 7: horizontales und vertikales Prototyping ist der Unterschied dieser beiden Typen veranschaulicht.

[49] Vgl. Sattler/Schmied
[50] Vgl. Sattler/Schmied
[51] Vgl. Sattler/Schmied

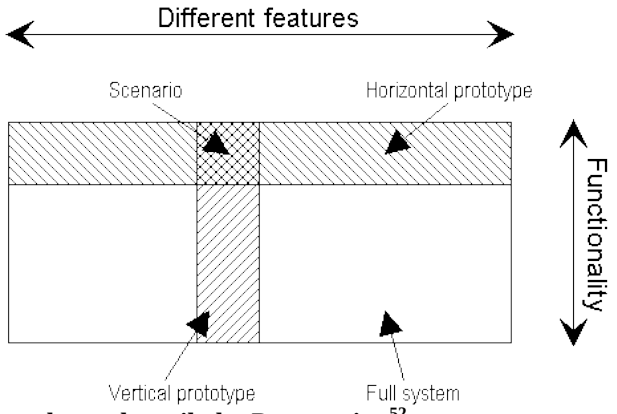


Abbildung 7: horizontales und vertikales Prototyping[52]

3.4.3. SmartCenter - Prototyping

Da es sich beim SmartCenter um ein Programmmodul des TaskCenters handelt, wird für den Entwicklungsprozess zunächst für jeden der drei Geschäftsfälle ein funktionaler Prototyp entwickelt. Durch die Verwendung des inkrementellen Ansatzes werden sämtliche Funktionen jedes Geschäftsfalles simuliert und gegebenenfalls geändert beziehungsweise erweitert.
Die Prototypen spiegeln die modulare Aufbauweise des SmartCenters wieder, was die Erweiterung mit zusätzlichen Geschäftsfällen unterstützt. Jeder der einzelnen Prototypen wird für sich simuliert und getestet.

[52] Abbildung entnommen aus: Sattler/Schmied

4. IST-Analyse

4.1. POLLEX-LC TaskCenter

4.1.1. Aufbau

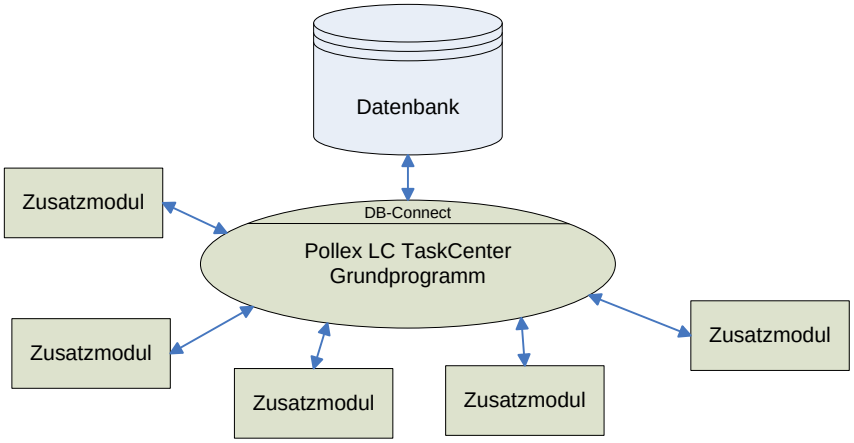


Abbildung 8: Struktur TaskCenter

Die Programmstruktur des TaskCenters sieht ein Grundprogramm vor, welches sämtliche grundlegende Funktionen eines ERP-Systems beinhaltet. Durch zusätzliche Programmmodule kann dieses Grundprogramm auf die Anforderungen des einzelnen Kunden zugeschnitten und erweitert werden.
Das Grundmodul umfasst die gängigen kaufmännischen Prozesse von:[53]

- Einkauf,
- Lagerverwaltung,
- Lieferscheinerstellung,
- Fakturierung,
- Auftragserfassung,
- Angebotserstellung

[53] Vgl. Lexen, 2007, S. 34

Nachfolgend eine kurze Beschreibung der Funktionalitäten des Grundmoduls nach Zusammengehörigkeit gegliedert:

Tabelle 2: Funktionen Grundmodul[54]

Programm-gruppen	**Stichworte**	**Kurzbeschreibung**
Grundlagen		Verwaltung für Systemvorgabewerte und Programmsteuerungsparameter
Stammdaten		Verwaltung der Stammdaten: Artikel, Kunden, Personal, Vermittler, Lieferanten, Banken, …
Verkauf	Auftragsbearbeitung und Fakturierung	Angebotserstellung, Auftragserfassung inkl. Bestätigung, Rechnungen und Gutschriften, Fakturierung, Lieferscheine
	Vor- und Nachkalkulation	Kalkulation mit Vergleichsfunktion
Einkauf	Bestellwesen	Containeroptimierung, Rahmenverträge, Bestellungen (inkl. Vorschläge)
	Wareneingang	Eingangsbuchung, Einstandspreisermittlung, Ermittlung mittlerer Preise, Lagerbuchung, technische und kaufmännische Freigabe, …
Lager	Artikelkartei	Lagerjournal, Bewegungsnachweis
	Inventur	Permanente Inventur, Stichtagsinventur, …
Auswertungen	Statistiken (Umsatz/Absatz/Deckungs-beitrag) inkl. Vorjahresvergleich	Auswertungen für Kunden, Artikel, Vertreter, …
Sonstige	Mailing	Verwaltung des Mailverkehrs

[54] Tabelle entnommen aus Lexen, S. 34 ff.

	Jahres Soll/Ist Plan	Deckungsbeitragsplanung und Vergleichsfunktionen für Soll- und Istwerte auf Quartalsebene
	Überleitung FIBU/KORE	Folgende Bereiche können übergeleitet werden: • Stammdaten • Aus- und Eingangsrechnungen • Wareneinsatzbuchungen • Kostenrechnungsdaten • POS-Kassa
	Datenübernahme	Artikelstamm, Artikelpreise, Kundenstamm, Lieferantenstamm, Ansprechpartner, Inventurdaten, Ausgangsrechnungen
	Schnellbrieffunktion	
	Verfügbarkeitsauskunft	Lagerbestand, offene Kundenaufträge, offene Bestellungen, Warenkorbfunktionen
	Doppelpreisauszeichnung	
	Drucksteuerung	Druckprofilverwaltung
	Chargennummern und Seriennummern	Verwaltung von Nummernkreisen, Inventur und Lagerführung, Tausch von Seriennummern
	Rapporterfassung	
	Vertreter-Provisionsabrechnung	

Die zusätzlich zu dem Grundprogramm zu erwerbenden Module umfassen folgende Aufgabengebiete:[55]

- Ausschreibungen
- CRM
- Datanorm-Import

[55] Vgl. Lexen, 2007, S. 38 ff.

- Edifact
- FIBU-Schnittstellen
- Fremdsprache
- Fremdwährung
- Leergutverwaltung
- Intrastat
- Kundensonderpreise
- Lieferantenanfragen
- Multilager
- OP-Verwaltung
- Organisatorische Einheiten
- POS-Kassa
- Produktion
- Projektverwaltung
- Reparaturverwaltung
- Reservierungssystem
- Setartikel
- Standardfertigung
- Web-Shop

Weitere Module befinden sich derzeit in Arbeit, wie auch jenes, welches diese Arbeit beschreibt.
Da das Internet sich in den letzten Jahren zu immer höheren Bandbreiten weiterentwickelt hat, besteht die Möglichkeit, auf eigene Server in den Filialen zu verzichten und stattdessen auf die Webservices des Stammsystems der Zentrale zuzugreifen. Dies führt wiederum zu Einsparungen in den Anschaffungskosten für den Kunden.[56]

[56] Vgl. Lexen, S. 73

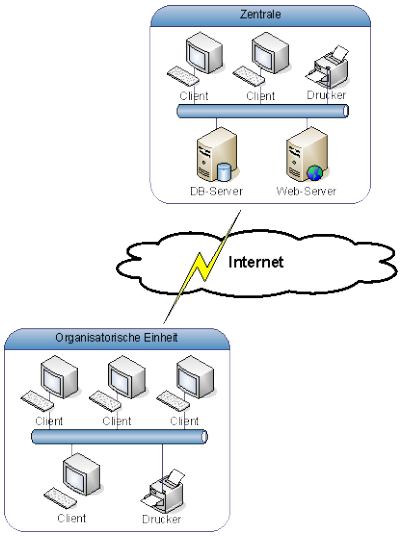


Abbildung 9: Topologie[57]

4.1.2. Artikelkartei

Da die in dieser Arbeit behandelten Prozessabläufe maßgebliche Änderungen in der Artikelkartei hervorrufen, folgt ein kurzer Überblick über die Funktion eben jener Kartei.
Die Artikelkartei ist jene Tabelle in der Datenbank, welche sämtliche Bewegungen eines Artikels über seinen historischen Lebenszyklus hinweg innerhalb des Unternehmens speichert. Hier werden für jeden, im Artikelstamm angelegten Artikel, der je produziert, eingekauft, verkauft oder vernichtet wurde, die entsprechenden Buchungen durchgeführt. Im Idealfall würde eine physische Inventur dieselben Mengen liefern, die auch in der Artikelkartei gespeichert sind.
Mit Hilfe der Artikelkartei können innerhalb eines Unternehmens folgende Aufgaben erfüllt werden:[58]

- **Bewegungsnachweis**
  Die Tabelle der Artikelkartei speichert für jede Artikelbewegung, die durch ein Dokument, wie zum Beispiel einen Wareneingangsschein, ausgelöst wird, die Dokumentennummer, eine Typenkennzeichnung des Dokuments, die Bezeichnung und das Bewegungsdatum mit. Somit können sämtliche Artikelbewegungen und ihre Auslöser nachvollzogen werden. Es werden außerdem Seriennummern in der Artikelkartei

[57] Abbildung entnommen und verändert aus: Lexen (2007:S.74)
[58] Vgl. POLLEX-LC Software GmbH

mitgeführt, um die Bewegungen dieser überprüfen zu können. Bei Kunden, die das Zusatzmodul *Multilager* erworben haben, werden auch die Lagernummer und die Lagerortnummer mitgespeichert.

- **Bestandsführung**
  Im täglichen Prozessablauf kann es zu Änderungen in bereits bestehenden, lagerführenden Dokumenten, wie zum Beispiel einem Lieferschein, kommen. Bei Mengenänderungen in den Positionen dieser Dokumente, werden auch die entsprechenden Buchungszeilen in der Artikelkartei aktualisiert.

- **Lagerinformationen**
  Mit Hilfe einer korrekt geführten Artikelkartei ist das Drucken von aktuellen Bestands- und Bewegungslisten von Artikeln, gegliedert nach Lagerorten, Zeiträumen und Lieferanten, möglich.

Die Artikelkartei garantiert somit eine ISO9000 gerechte Nachverfolgung des Lieferwegs vom Lieferanten zum Kunden.

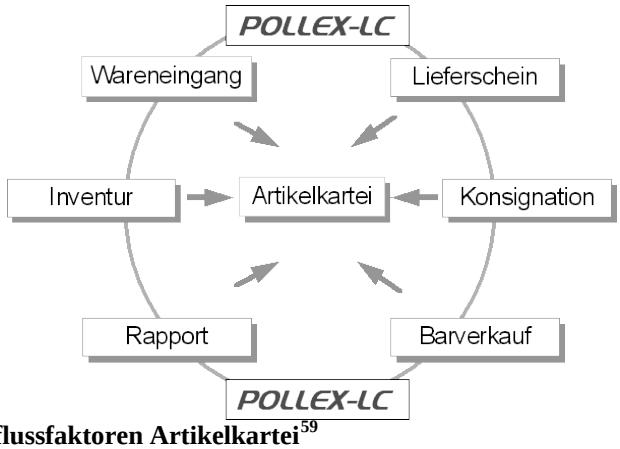


Abbildung 10: Einflussfaktoren Artikelkartei[59]

4.1.3. Barcodes

Das TaskCenter bietet die Funktion, an sämtliche Dokumente eine beliebige Nummer als Barcode anzudrucken. In den für diese Arbeit relevanten Prozessen

[59] Abbildung entnommen aus: POLLEX-LC Software GmbH

sind dies die Wareneingangsscheinnummer und die Auftragsnummer. Zusätzlich zu diesen Nummern sind für sämtliche Lagerorte und Artikel Etiketten mit den Lagernummern und Artikelnummern zu drucken.
Das TaskCenter verwendet hierfür zwei verschiedene Barcode-Standards:

EAN-13:

Abbildung 11: EAN-13 Barcode[60]

Für den Ausdruck von Artikelnummern wird der EAN-13 Barcode verwendet. Dieser Barcode besitzt 13 rein numerische Stellen, wovon eine als Prüfziffer ausgewiesen ist. Das Eingabealphabet sind die neun Ziffern, die folgendem Aufbau unterliegen:[61]

- Stelle 1 und 2 geben das Herkunftsland des Artikels an (Länderpräfix).
- Die Stellen 3-7 kennzeichnen den Hersteller des Artikels.[62]
- Die Stellen 8-12 werden vom Hersteller vergeben und bilden die eigentliche Artikelnummer.
- Die letzte Stelle ist die Prüfziffer, die aus den vorangegangenen Stellen berechnet wird. Bei dieser Berechnung wird die Quersumme der Stellen gebildet und der erhaltene Wert Modulo10 genommen.

Code39:

Abbildung 12: Code39 Barcode[63]

Für alle anderen Barcodes, die zum Einsatz kommen, wird der Code39 Standard verwendet. Das alphanumerische Eingabealphabet dieses Codes besteht aus 26 Buchstaben, 10 Ziffern und einigen Sonderzeichen, wobei die Anzahl der Stellen des Codes beliebig wählbar ist.

---

[60] Barcode generiert mit http://www.barcodesoft.com/de-de/online-barcode-generator.aspx
[61] Vgl. ActiveBarcode - Barcodetypen
[62] Die Codes für Hersteller werden in Österreich von GS1 Austria verwaltet. (http://www.gs1austria.at/)
[63] Barcode generiert mit http://www.barcodesoft.com/de-de/online-barcode-generator.aspx

Optional kann der Barcode um ein Prüfzeichen erweitert werden, welches sich, wie beim EAN-13, aus den anderen Stellen berechnet.[64]
Für die Berechnung des Prüfzeichens müssen zunächst die alphanumerischen Zeichen des Eingabealphabets in numerische Werte umgewandelt und davon die Summe gebildet werden. Diese Umwandlung erfolgt auf der Grundlage einer Referenztabelle (siehe Tabelle 3: Referenztabelle für Code39). Die Summe Modulo43 ergibt einen Wert, der wieder aufgrund der Referenztabelle einem Zeichen zugeordnet werden kann. Dieses Zeichen ist somit das Prüfzeichen des Codes.

Tabelle 3: Referenztabelle für Code39[65]

00	0	11	B	22	M	33	X
01	1	12	C	23	N	34	Y
02	2	13	D	24	O	35	Z
03	3	14	E	25	P	36	-
04	4	15	F	26	Q	37	.
05	5	16	G	27	R	38	Space
06	6	17	H	28	S	39	$
07	7	18	I	29	T	40	/
08	8	19	J	30	U	41	+
09	9	20	K	31	V	42	%
10	A	21	L	32	W		

4.1.4. Prozessabläufe

Ein Geschäftsprozess in sich ist eine

- inhaltlich abgeschlossene,
- sachlogische Abfolge von Funktionen,

welche für die Bearbeitung eines betriebswirtschaftlichen Objektes vonnöten sind. Das BWO (betriebswirtschaftliches Objekt) ist das prägende Element jedes Prozesses und wird gegebenenfalls von weiteren Objekten unterstützt.
Für die Darstellung eines Geschäftsprozesses wird das EPK-Modell[66] verwendet. Die ereignisgesteuerten Prozessketten werden durch ein Ereignis ausgelöst (Start-Event). Durch eine Abfolge von Funktionen und weiteren Ereignissen wird das betriebswirtschaftliche Objekt bis zu einem abschließenden Ereignis bearbeitet.

[64] Vgl. ActiveBarcode - Barcodetyen
[65] Tabelle entnommen aus: ActiveBarcode - Prüfziffernberechnung
[66] Dieses Modell wurde an der Universität des Saarlandes in Saarbrücken im Zuge eines Forschungsprojektes im Jahre 1992 entwickelt.

Betriebswirtschaftliche Objekte können sowohl Informationsobjekte (z.B. Auftrag) oder materialisierte Objekte (z.B. Artikel) darstellen.
Die Geschäftsprozesse Inventur, Auftragserfassung Wareneingangs –und Warenausgangskommissionierung und Auftragserfassung, werden nun in ihrem derzeitigen Prozessablauf etwas genauer betrachtet, um die Schwachstellen innerhalb des Ablaufprozesses identifizieren zu können. Später können daraus Anforderungen an das SmartCenter besser erkannt und definiert werden.

Mobile Auftragserfassung

Die mobile Auftragserfassung ist speziell für Vertreter von hohem Nutzwert, da sie nicht mehr schwere Kataloge mit der Produktauswahl zu den Kunden mitbringen müssen, sondern es bietet ihnen Zugriff auf den gesamten Artikelstamm des Unternehmens. Neben der Gewichtserleichterung ist auch die Tatsache, dass so immer sämtliche Kundendaten und aktuelle Preise verfügbar sind, auf der Seite der Vorteile einer mobilen Auftragserfassung zu finden.
Derzeit wird für Vertreter eine *Offline*-Lösung der mobilen Auftragserfassung angeboten.

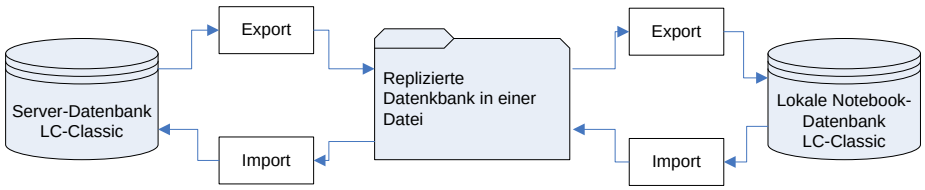


Abbildung 13: mobile Auftragserfassung "offline"

Der Prozessablauf wird in Abbildung 14: Prozessablauf der mobilen Auftragserfassung dargestellt und beschreibt den Auftrag als betriebswirtschaftliches Objekt.
Ein Notebook wird vor dem tatsächlichen Einsatz mit einer „*mobilen*“ Version des TaskCenters ausgerüstet. Diese, im Funktionsumfang eingeschränkte Version des TaskCenters, benützt eine lokale Datenbank, die einen Dump der Serverdatenbank darstellt. Diese Datenbankübertragung ist in Abbildung 13: mobile Auftragserfassung "offline" dargestellt. Dabei wird der derzeitige Inhalt der Serverdatenbank über ein Exportprogramm in eine Datei verfrachtet. Aus dieser Datei kann ein Importprogramm des mobilen TaskCenter-Clients die Daten für die lokale Datenbank importieren.
Nachdem die Datenbank übertragen worden ist, ist das Notebook gerüstet für den Außendiensteinsatz. Mit dem mobilen TaskCenter-Client hat ein Vertreter nun vollen Zugriff auf sämtliche Auftragserstellungsfunktionen, Stammdaten von

Kunden und Artikeln und Ähnliches. Nach der Rückkehr vom Außendienst werden sämtliche Änderungen innerhalb der Datenbank, das heißt sämtliche erfassten Aufträge, von einem Exportprogramm wieder in die Datenbankdatei transferiert, von wo sie weiter in die Serverdatenbank importiert werden. Erst jetzt sind die erfassten Aufträge tatsächlich ins System eingepflegt.
Erst nach Einpflege der lokalen Notebookdaten ist der Auftrag tatsächlich im Stammsystem erfasst und der Prozess erfolgreich beendet.
Die Darstellung des Ablaufprozesses zur mobilen Auftragserfassung erfolgt in Abbildung 14: Prozessablauf der mobilen Auftragserfassung.

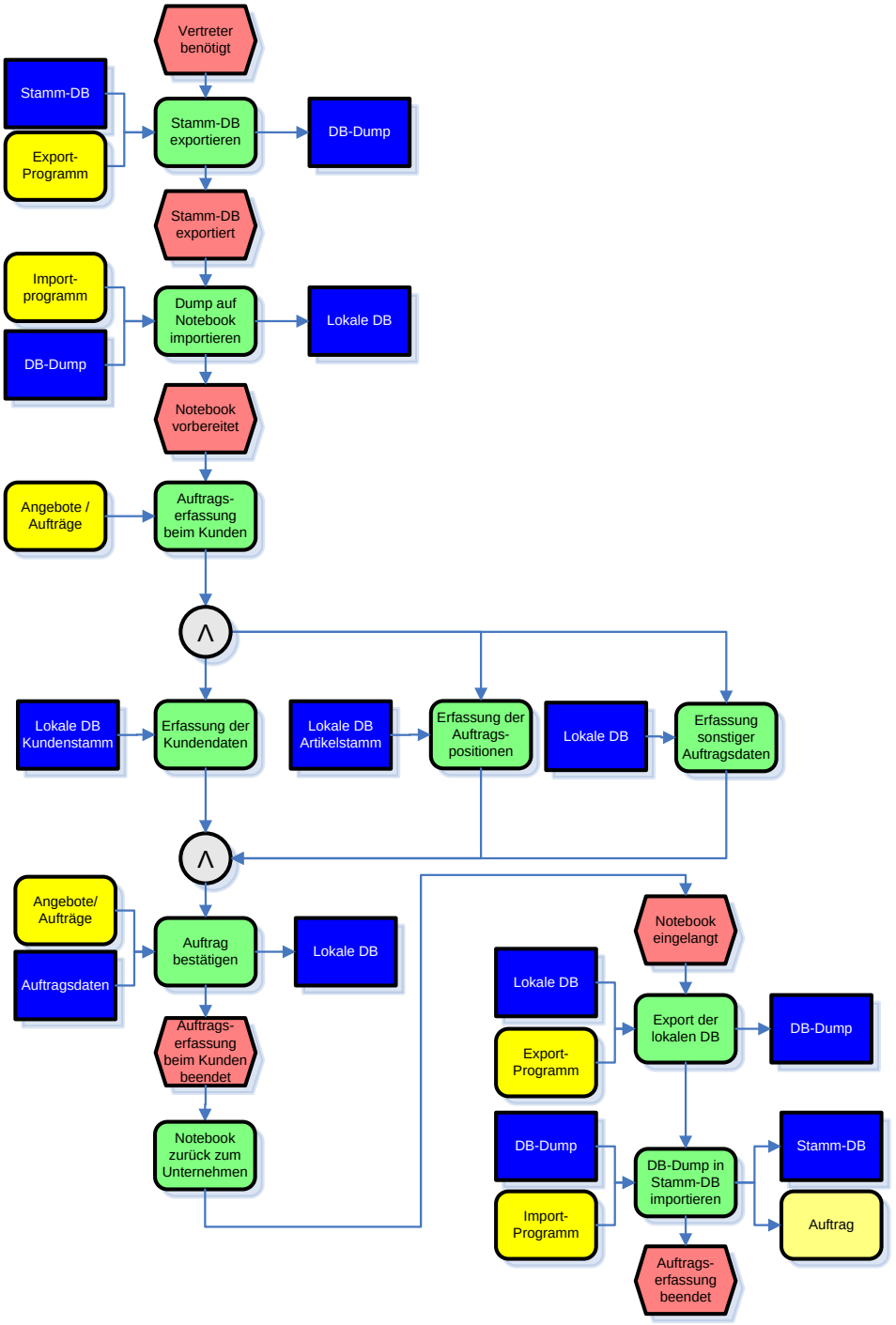


Abbildung 14: Prozessablauf der mobilen Auftragserfassung

Inventur

Eine Inventur ist nicht nur eine gesetzlich vorgeschriebene Aktion zur Ermittlung der Vermögenswerte, sondern sie verhilft dem Unternehmen auch, seine Artikelkartei auf Fehlmengen zu überprüfen und gegebenenfalls zu korrigieren.
In den meisten Fällen werden vor einer Inventur eine oder mehrere Zähllisten erstellt. Diese Listen werden an die Mitarbeiter ausgegeben, die schlussendlich die physische Inventur durchführen. Zähllisten werden aufgrund von Lagern beziehungsweise Lagerorten erstellt. So werden alle Artikel des ausgewählten Lagers auf die Liste gedruckt. Natürlich können mehrere Lager oder Lagerorte miteinander kombiniert werden.
Auf diesen Zähllisten ist ein Feld für jede Position reserviert, auf dem der Mitarbeiter die gezählte Menge eintragen kann.
Nach Abschluss der physischen Inventur werden die erhaltenen Ergebnisse in der Inventurschnellerfassung eingetragen.
In der Inventurschnellerfassung sind sämtliche Artikel, die erfasst wurden, mit ihrem Sollbestand aus der Artikelkartei und ihrem Istbestand aus der physischen Inventur verzeichnet.
In diesem Fenster kann die erfolgte Inventur nun manuell überprüft und gegebenenfalls editiert werden. Nach Beenden der Überprüfung werden die Positionen der Inventurschnellerfassung in die tatsächliche Inventur übernommen und somit die neuen Mengenbestände in die Artikelkartei zurückgespeichert.

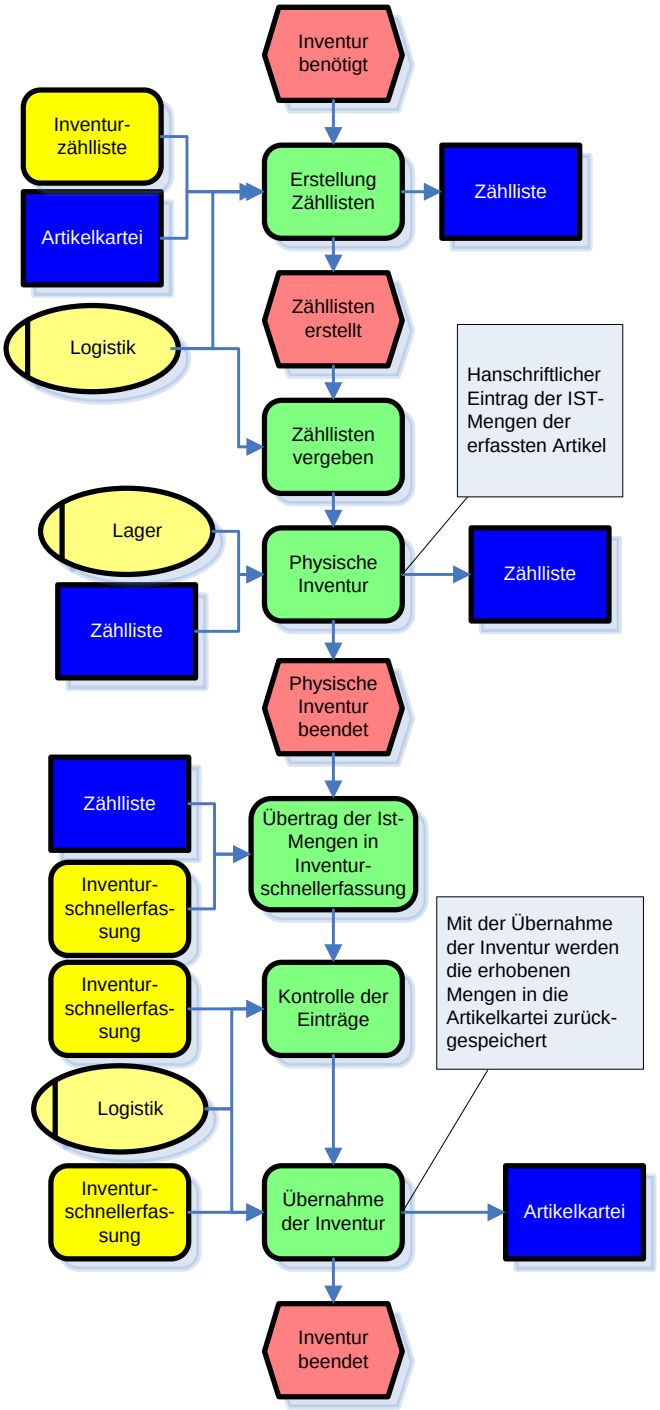


Abbildung 15: Prozessablauf Inventur

Wareneingangskommissionierung

Dem Prozess des Wareneingangs geht meist ein Bedarf voraus. Dieser Bedarf veranlasst nun die Auswahl eines Lieferanten, der den Bedarf decken kann. Nächster Schritt ist die Bestellung beim gewählten Lieferanten und die Anlieferung der bestellten Artikel. Die Anlieferung ist nun der Anstoß zum Prozess des Wareneinganges. Abbildung 16: Prozess Wareneingang

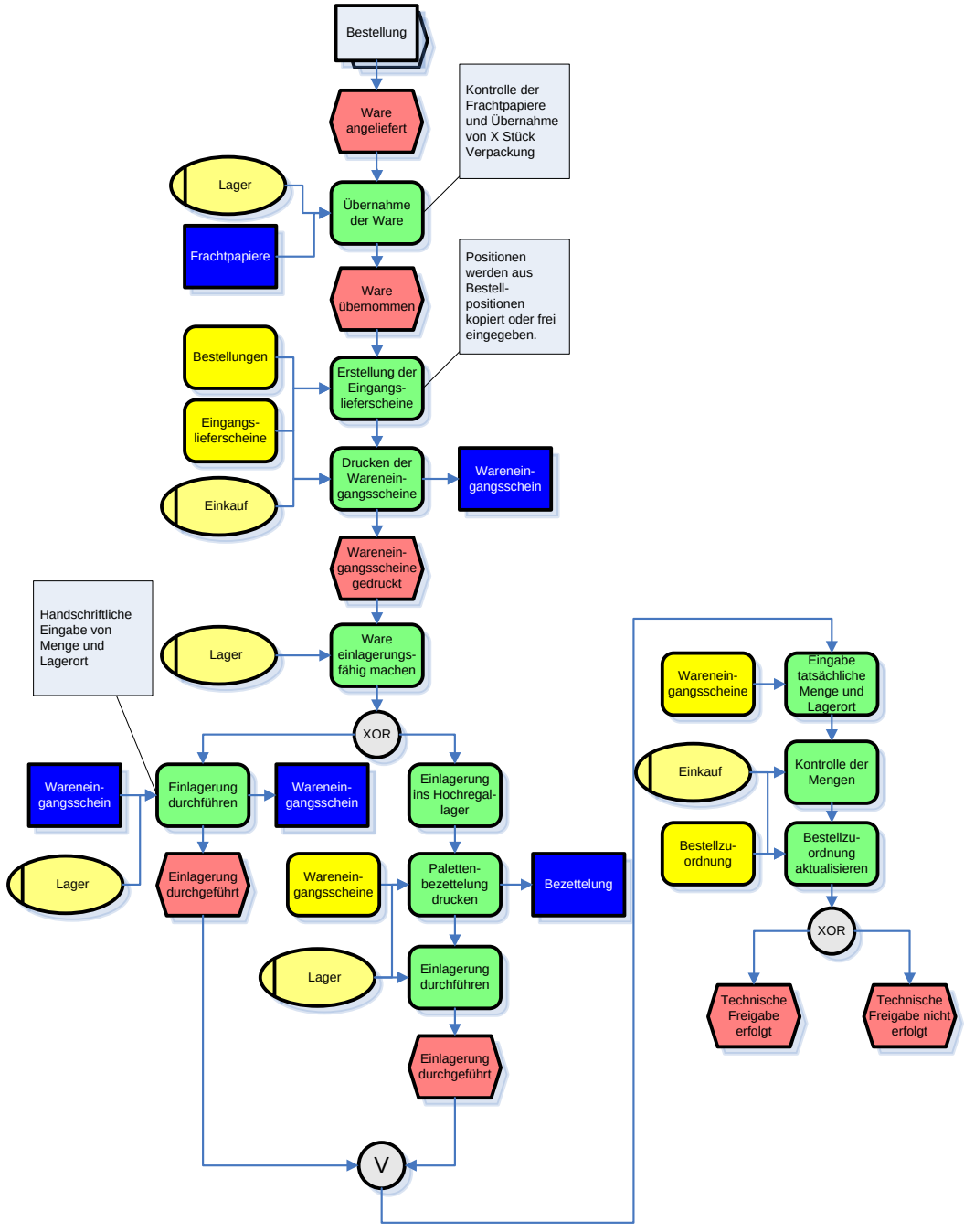


Abbildung 16: Prozess Wareneingang

Beim Wareneingang wird zunächst eine bestimmte Anzahl von „*Verpackung*“ übernommen. Bei dieser Übernahme wird der Inhalt der Verpackung noch nicht überprüft, es werden lediglich die Frachtpapiere mit der tatsächlich entladenen Menge an Gebinden kontrolliert.

Der nächste Schritt ist dann die Erstellung eines neuen Eingangslieferscheines. Die Kopfdaten werden mit den Daten des Lieferanten, dem Datum des Lieferantenlieferscheines und der Lieferantenlieferscheinnummer gespeichert. Die Positionen können entweder manuell eingefügt, oder aus Bestellungen kopiert werden. In Bestellungen können alle offenen Bestellungen dieses Lieferanten angezeigt werden. Diese Positionen werden mit der Menge „0“ in die Eingangslieferscheinpositionen übernommen.
Aus diesem im System erstellten Eingangslieferschein wird nun je Wareneingangsposition (je gelieferten Artikels) ein Wareneingangsschein gedruckt. Dieser Wareneingangsschein enthält eine eindeutige Nummer, die Artikelnummer bzw. –bezeichnung und, wenn vorhanden, einen Lagerplatz. Die eindeutige Nummer des Wareneingangsscheines ist der Positionsindex aus den Eingangslieferscheinpositionen.
Bei Einlagerung in Hochregallager werden für die Wareneingangspositionen Bezettelungen gedruckt und die physischen Waren damit versehen.
Nach einer manuellen Einlagerung müssen endgültigen Lagerorte der Positionen nachträglich in den Wareneingangsscheinen aktualisiert werden.
Stimmen die Ware und die Menge der Wareneingangsscheine mit den Mengen des Lieferantenlieferscheines überein, und ist für jeden Artikel der Lagerort eingetragen, so erfolgt für diesen Artikel die technische Freigabe. Ein Artikel gilt als technisch freigegeben, wenn er in der korrekten Menge geliefert und korrekt eingelagert wurde. Durch diese Freigabe wird die Artikelkartei für diesen Artikel aktualisiert und die freigegebenen Positionen des Eingangslieferscheines werden gesperrt.
Mit der technischen Freigabe des Artikels endet der Prozess des Wareneingangs.

Warenausgangskommissionierung

Prinzipiell geht einem Warenausgang ein Kundenauftrag voraus. Siehe Abbildung 17: Prozess Warenausgang

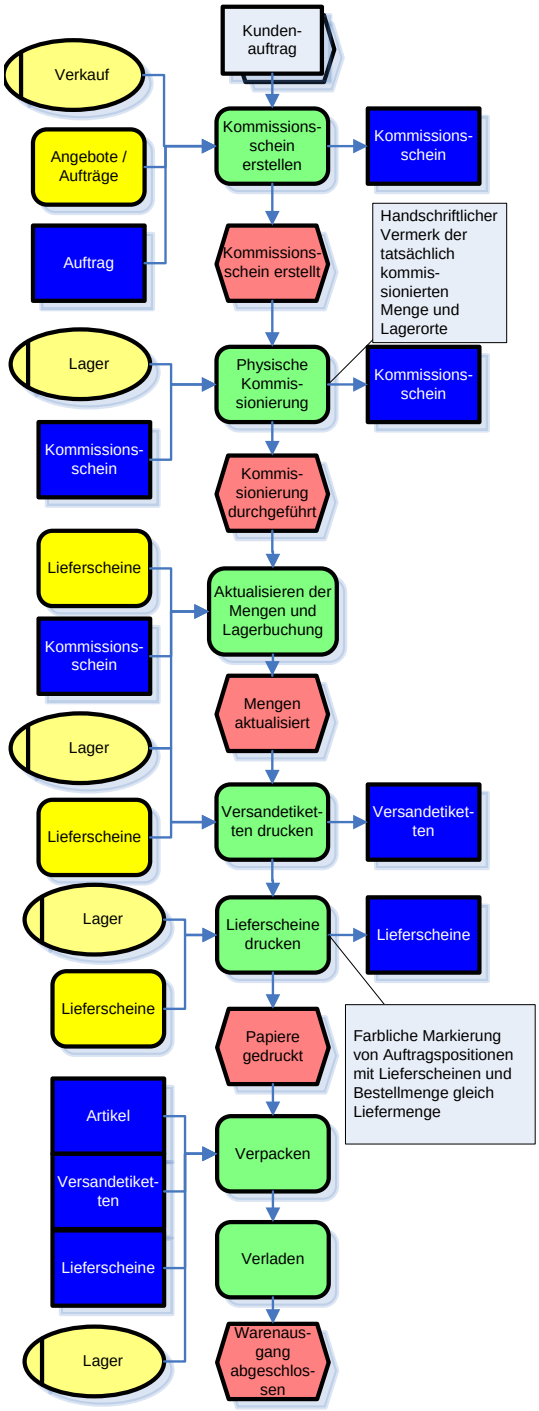


Abbildung 17: Prozess Warenausgang

Durch den Druck eines Kommissionsscheines aus dem Kundenauftrag wird auch in der Datenbank ein Kommissionsschein mit den Auftragspositionen angelegt.
Anhand des gedruckten Kommissionsscheines erfolgt dann die physische Kommissionierung im Lager.
Auf den Kommissionsscheinen werden zunächst handschriftlich die tatsächlich bereitgestellten Artikel mit ihrer Menge und gegebenenfalls ihren Lagerorten eingetragen und danach im System in den Kommissionsscheinen aktualisiert. Sämtliche Positionen werden nun lagergebucht, dies bewirkt eine Aktualisierung des Lagerbestandes. Wurden alle Positionen des Kommissionsscheines korrekt gebucht, so wird aus dem Kommissionsschein automatisch ein Lieferschein generiert. Über einen Datenbanktrigger werden die offenen Auftragsmengen in der Datenbank aktualisiert.
Nach dem Druck der Versandetiketten und der Lieferscheine wird die Sendung verpackt und zur Verladung freigegeben. In den Kundenaufträgen werden Positionen farblich markiert, bei denen die versandte Menge gleich der Auftragsmenge ist und dazu ein Lieferschein erstellt wurde. Dies bedeutet, dass diese Position des Auftrages komplett versandt wurde und keine offenen Mengen mehr zu berücksichtigen sind. Wurden alle Positionen korrekt versandt, so wird der Auftrag automatisch auf *erledigt* gesetzt.

4.2. Marktanalyse

Vor dem eigentlichen Entwurf und der Ausarbeitung der Anforderungen an die neue Applikation, ist es sinnvoll, eine „*Bestandsaufnahme*“ der am Markt befindlichen Soft- beziehungsweise Hardware im Bereich der mobilen Datenerfassung durchzuführen.
In vielen Geschäftsbereichen hat die mobile Datenerfassung bereits Einzug gehalten, da sie die Ablaufprozesse für den Anwender effizienter und einfacher gestaltet. Beispiele wie in der Gastronomie, im Handel oder in der Lagerverwaltung sind schnell und zahlreich zu finden.
Die Anforderungen an die Endgeräte, wie auch an die Applikationen, sind sehr hoch, da sie oft in der alltäglichen Arbeitsroutine eingesetzt werden und Gerätefehlfunktionen, Softwarefehler oder schlechte Performance bei der Verarbeitung von hohen Datenmengen besonders frustrierend sind.

4.2.1. Hardware

Es lohnt sich einen kurzen Blick auf die derzeit am Markt angebotene Hardware zu werfen, um sich über die Möglichkeiten zu informieren, für die diese Geräte einsetzbar sind.
Um die Vielfalt der Geräte ein wenig einschränken zu können, werden nur Geräte betrachtet, welche Windows CE, PocketPC oder Windows Mobile als Betriebssystem verwenden, da diese mit dem Microsoft® .NET Compact Framework 2.0 ausgestattet werden können. Diese Einschränkung ist für die vorliegende Arbeit zulässig, da die geplante Softwareapplikation nur unter diesen Voraussetzungen entwickelt wird.
Auch hier gibt es unzählige Anbieter mit einer Vielzahl verschiedener Geräte. Aus diesem Grund werde ich nur die wichtigsten Eckdaten beschreiben und nicht auf einzelne Typen eingehen.

Ausführungsvarianten

Die einfachste Variante sind Barcodeleser ohne jegliche Verarbeitungslogik. Diese sind häufig an Supermarktkassen und dergleichen zu finden. Hier werden lediglich Barcodes gescannt und die darin gespeicherte Information an ein Verarbeitungssystem weitergeleitet.
Eine zweite Variante sind mobile Datenerfassungsgeräte, auf welchen Anwendungen zur Verarbeitung von erfassten Daten entwickelt und installiert werden können.
Im Folgenden wird auf einige Punkte eingegangen, die bei der Auswahl der richtigen Hardware, ausschlaggebend sein können:[67]

Ergonomie: Es gibt verschiedene Bauformen für beide Varianten, wie zum Beispiel Datenerfassungsgeräte mit und ohne Pistolengriff. Bei vielen manuellen Eingaben sind jene ohne Pistolengriff den Geräte mit Pistolengriff, welche beim Scannen über größere Distanzen ihre Vorteile zeigen, zu bevorzugen.
Barcodeleser in Stiftform eignen sich lediglich für die Variante ohne Verarbeitungslogik.

Display:
Die meisten Geräte besitzen Displays mit dem Standard QVGA (240x320). Die in dieser Arbeit entwickelte Software ist auf diese Auflösung optimiert, die Dimension der Schirmdiagonale hat dabei keinen Einfluss auf die Auflösung.

[67] Vgl. Logistic to go – Software für mobile Datenerfassung

Für das SmartCenter wird ein Gerät mit Touchscreen vorausgesetzt.

Datenkommunikation:
Für Geräte der mobilen Datenerfassung existieren verschiedene Techniken der Datenkommunikation. Die für das SmartCenter relevanten Techniken sind in Kapitel 3.3 genauer beschrieben.
Zusätzlich können die Geräte über Bluetooth oder IrDA, für den Datenaustausch zwischen zwei Geräten über kurze Distanzen, verfügen (z.B. Mobiltelefon, Drucker, …). Eine Schnittstelle, wie USB oder RS232, sollte für die Synchronisation mit einem PC oder für die Einspielung der Software vorhanden sein.[68]

Scanner:
Der Einsatz der Scanfunktion setzt Geräte mit einem Barcodescanner voraus. Hierbei gibt es Ausführungen für 2D und 3D Barcodes. Für das SmartCenter sind 2D-Scanner ausreichend. Hierbei ist zu beachten, dass die Geräte hinsichtlich ihrer Scannerreichweite auf das Einsatzgebiet optimal ausgewählt werden.

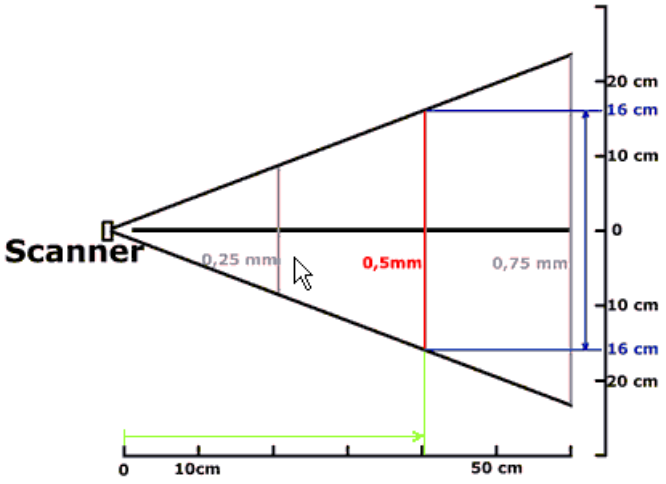


Abbildung 18: Bsp. Scannerreichweite[69]
Die Ermittlung eines optimalen Scannermoduls für das Einlesen eines Barcodes in 40 cm Entfernung ist in Abbildung 18: Bsp. Scannerreichweite dargestellt. Die Mindestmodulbreite des Barcodes (Strichbreite) wäre in diesem Fall 0,5mm und die Breite des gesamten Barcodes dürfte 32 cm nicht überschreiten.[70]

[68] Vgl. Logistic to go – Software für mobile Datenerfassung
[69] Abbildung entnommen aus: Logistic to go – Aufbau einer Lesefeldkurve
[70] Vgl. Logistic to go – Aufbau einer Lesefeldkurve

Da das TaskCenter für die Etikettierung Barcodes in Code39 und EAN-13 – Standard verwendet, müssen die verwendeten Geräte diese Codierungen unterstützen. Die Scanner müssen den gescannten Standard automatisch erkennen und auslesen können.

Schutzklassen

Da an die Geräte hinsichtlich ihrer Robustheit, je nach Einsatzgebiet, unterschiedliche Anforderungen gestellt werden, ist ein kurzer Blick auf die Klassifizierung nach Schutzklassen notwendig.
Um Angaben über die Robustheit eines elektronischen Gerätes machen zu können, wurden für den europäischen Raum die sogenannten IP-Zahlen eingeführt.
IP steht in diesem Fall nicht für Internet Protocol, sondern für Ingress Protection oder wie in DIN nachzulesen ist: International Protection.
IP – Zahlen werden nach folgendem Schema klassifiziert: IP XXXX
Die einzelnen Stellen der IP-Zahlen sind Kennzeichen für folgende Schutzarten:[71]

- 1.Stelle: Berührungsschutz / Fremdkörperschutz
- 2.Stelle: Wasserschutz
- 3.Stelle: Zusätzlicher Berührungsschutz
- 4.Stelle: Ergänzende Buchstaben

Die dritte und vierte Stelle müssen nicht zwangsläufig angegeben werden. In der folgenden Tabelle werden die Schutzarten zusammengefasst:

Tabelle 4: IP-Zahlen[72]

IP-Zahl	**Beschreibung**
Berührungsschutz	
IP 0X	Weder Berührungs- noch Fremdkörperschutz
IP 1X	Handrückenschutz; Fremdkörperschutz mit Durchmesser > 50 mm
IP 2X	Fremdkörperschutz mit Durchmesser > 12 mm; Fingerschutz
IP 3X	Fremdkörperschutz mit Durchmesser > 2.5 mm

[71] Vgl. Elektronic-Magazin, 2007
[72] Tabelle zusammengeführt aus Elektronic-Magazin (2007) und omega.com (S. Z-194)

IP 4X	Fremdkörperschutz mit Durchmesser > 1 mm
IP 5X	Schutz gegen schädliche Staubablagerungen
IP 6X	Schutz gegen Feinststaub; Vollständiger Schutz gegen Eindringen von Staub.
Wasserschutz	
IP X0	Kein Schutz gegen Wasser
IP X1	Schutz gegen vertikal tropfendes Wasser
IP X2	Schutz gegen tropfendes Wasser bis zum Winkel von 15° abweichend der Senkrechten.
IP X3	Schutz gegen sprühendes Wasser bis zum Winkel von 60° abweichend der Senkrechten.
IP X4	Schutz gegen sprühendes Wasser von allen Seiten.
IP X5	Schutz gegen Wasserstrahlen von allen Seiten.
IP X6	Schutz gegen temporäre Überflutung.
IP X7	Schutz gegen Wasser beim Eintauchen.
IP X8	Schutz gegen Wasser bei langfristigem Eintauchen.
IP X9	Schutz gegen Wasser von allen Seiten auch bei erhöhtem Wasserdruck.
3.Stelle	
A	Handrückenschutz oder Fremdkörper mit Durchmesser > 50 mm
B	Fingerschutz
C	Werkzeugschutz
D	Drahtschutz
4.Stelle	
H	Hochspannungs-Betriebsmittel
M	Schutz überprüft bei beweglichen Teilen in Betrieb
S	Schutz überprüft bei beweglichen

	Teilen im Stillstand
W	Schutz überprüft bei festgelegten Wetterbedingungen

In den USA wird nicht nach IP-Codes klassifiziert sondern nach dem sogenannten NEMA-Standard (National Electrical Manufacteurs Association).[73]

4.2.2. Software

Um einen Überblick über das Angebot anderer Firmen zu bekommen, wird eine Recherche über mobile Applikationen für den ERP-Bereich durchgeführt. Im Folgenden ein kurzer Auszug von Firmen, die ihrerseits Softwarelösungen anbieten.

Shipping.net

Shipping.net ist eine Logistiksoftware der Firma CAPO IT Solutions GmbH[74]. Sie unterstützt vor allem Firmen beim Warenausgang. Die Software unterstützt dabei den Weg vom Import der Aufträge bis hin zum Versenden der Ware. Die Auftragspositionen werden über eine XML-Schnittstelle in die Software eingelesen.
Die Software bietet dem Benutzer die Möglichkeit, die zu versendenden Einzelstücke zu Packstücken zusammenzufassen und entsprechend zu etikettieren. Bei kleineren Packstücken, welche an denselben Auftraggeber gehen, kann die Software diese zu Paletten zusammenfassen und entsprechend mit Labels versehen.
Nach erfolgreicher Etikettierung der Packstücke generiert die Software mit Hilfe der Auftragsdaten die schlussendlichen Versendungen.[75]

CodeSnap

Die Firma Flexicom bietet Software für den Wareneingang, die mobile Inventur und Inventarisierung, sowie für die Einsatzplanung von Außendienstmitarbeiter

[73] Da die Klassifizierung nach NEMA nach einer anderen Bewertungsbasis erfolgt, ist keine direkte Umrechnung möglich. Es werden von einigen Firmen Tabellen für eine angenäherte Umrechnungen angeboten. Ein Beispiel dafür siehe http://www.gett.de/content/technical/nemaik.html.
[74] http://www.capto.at
[75] Vgl. Capto IT

an. Für sämtliche Applikationen werden Zusatzmodule für RFID, Magnetkartenleser, Druckmodule, etc. angeboten.[76]

Aisci

AISCI ist eine Firma, die sich auf die Entwicklung automatischer Identifikations-Systeme spezialisiert hat. Sie bietet eine Lösung zur Warenkommissionierung an, bei der die Pickaufträge aus dem Lager direkt an das mobile Endgerät gesendet werden.[77]

[76] Vgl. Flexicom
[77] Vgl. Aisci

5. Motivation

5.1. Entwicklung von Standardsoftware

Die Frage, die sich nun stellt ist, weshalb soll der Mehraufwand betrieben werden, aus einer individuellen Kundenanfrage eine standardisierte Software zu entwickeln.

Der der Grundgedanke von POLLEX-LC TaskCenter ist es, einen branchenunabhängigen Standard zu schaffen, welcher dem Kunden sämtliche Freiheiten lässt, seine wertschöpfenden Geschäftsprozesse in der Software, ohne zusätzlichen Programmieraufwand, abzubilden. Dieses Konzept hat den Vorteil, dass sämtliche Benutzer, egal welcher Branche, denselben Entwicklungsstand ausgeliefert bekommen und keinerlei Probleme, wie bei Individualprogrammierung, auftreten können.

Aufgrund dieses Gedankens ist zu entscheiden, ob aus einer individuellen Anfrage eventuell ein Standard geschaffen werden soll, welcher als zusätzliche Funktion auch anderen Kunden zur Verfügung gestellt werden kann, oder ob diese Anfrage als einmalige Individualprogrammierung behandelt werden soll.

Für die Einbindung des SmartCenters in die Standardsoftware sprechen folgende Punkte:

- Sicherung der gegenwärtigen Marktposition: Die Abwanderung durch Kunden soll vermieden werden → Festigen der Kundenbindung.
- Neuakquirierung von Kunden: Durch den Einsatz neuer Technologien soll ein Produktvorteil gegenüber Konkurrenzprodukten erworben werden.
- Weiterentwicklung der Software: Der „*natürlichen Alterung*“ soll entgegengewirkt werden.

Diesen Punkten stehen der zusätzliche Aufwand und die Risiken einer Softwareentwicklung gegenüber.

5.2. Vorteile des SmartCenters

Um zu verdeutlichen, welche Vorteile eine Umstellung auf das SmartCenter für den Kunden mit sich bringt, nachfolgend eine Gegenüberstellung von Anwendungsfällen mit dem derzeitigen System und der neu zu entwickelnden Applikation.

Tabelle 5: Vorteile SmartCenter

Herkömmliches System	**SmartCenter**	**Vorteile**
Inventur		
Sichtkontrolle von Artikelnummern	Scan von Artikelnummern inklusive sofortiger Überprüfung	Keine falschen Einträge von Artikel durch menschliche Fehler
Handschriftliche Einträge von Inventurergebnisse	Eindeutige numerische Einträge für Mengen	Keine Nachprüfungen oder Fehler durch Unleserlichkeit
Handschriftliche Einträge müssen nachträglich in das System eingepflegt werden	Per Knopfdruck Einpflege der kompletten, gescannten Artikellisten	Beschleunigung der Inventur
Auftragserfassung		
Notebook als „mobiles“ Arbeitsgerät	Nutzung handlicher Endgeräte (z.B. PDA)	Verwendung praktikabler Endgeräte
Verwendung eines Datenbank-Dumps	Echtzeit-Daten	Verwendung von veralteten Daten wird verhindert
Akquirierte Aufträge werden erst nach Rückkehr in das System eingepflegt	Neuakquirierungen sind sofort im System integriert	Artikelreservierungen sofort verfügbar, dadurch keine Dateninkonsistenzen
Wareneingang		
Sichtkontrolle von Artikelnummern, Lager/Lagerorten	Scan von Artikelnummern, Lager/Lagerorten inklusive sofortiger	Keine falschen Einträge durch menschliche Fehler

	Überprüfung	
Handschriftliche Mengeneingabe	Numerische Mengenangaben	Keine Nachprüfungen oder Fehler durch Unleserlichkeit
Nachträgliche Einpflege in das System	Per Knopfdruck Einpflege in das System	Beschleunigung des Vorgangs *Wareneingang*
Warenausgang		
Sichtkontrolle von Artikelnummern, Lager/Lagerorten	Scan von Artikelnummern und Lager/Lagerorten inklusive sofortiger Überprüfung	Keine falschen Einträge durch menschliche Fehler
Handschriftliche Mengeneingabe	Numerische Mengenangaben	Keine Nachprüfungen oder Fehler durch Unleserlichkeit
Nachträgliche Einpflege in das System	Per Knopfdruck Einpflege in das System	Beschleunigung des Vorgangs *Warenausgang*

6. Entwurf und Implementierung

6.1. Systemdesign

Das SmartCenter wird als Zusatzmodul in das TaskCenter-Grundprogramm eingebunden und kann vom Kunden durch Erlangung der entsprechenden Lizenz verwendet werden. Siehe Abbildung 19:
Das neu entwickelte Zusatzmodul wird die Datenbankanbindung des Grundprogramms in Anspruch nehmen und keinerlei eigene Funktionen für Datenbanktransaktionen bereitstellen. Dies ermöglicht, unter Verwendung von klar definierten Schnittstellen, den Zugriff auf bereits etablierte und stabile Programmabläufe.

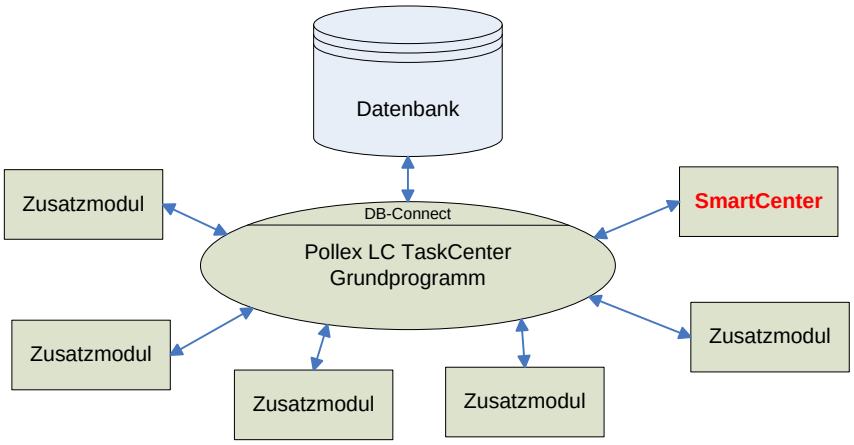


Abbildung 19: Modul SmartCenter

Das SmartCenter verwendet die serviceorientierte 3-Tier-Architektur des POLLEX-LC TaskCenters.

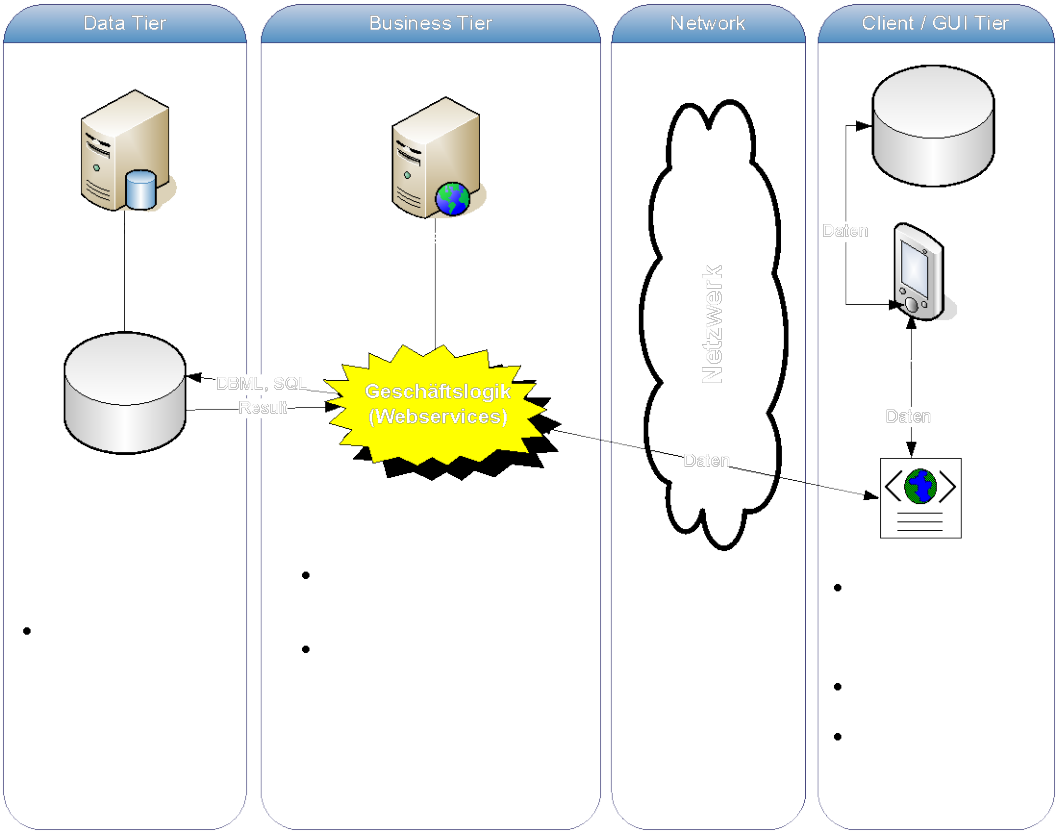


Abbildung 20: Systemarchitektur POLLEX-LC SmartCenter[78]

Temporäre Laufzeitdaten werden in Datasets im Arbeitsspeicher des Gerätes gehalten und nur durch vordefinierte Ereignisse, wie zum Beispiel das Abschließen eines Vorganges, per XML in den Webservices zur Weiterverarbeitung übertragen und dann gegebenenfalls in der Datenbank gespeichert. Diese Datensätze sind beim Beenden der Software oder beim Deaktivieren des Gerätes unwiderruflich verloren.
Permanente Daten werden im lokalen Speicher des Gerätes in XML-Dateien gespeichert und haben nach Beenden der Applikation oder Abschalten des Gerätes weiter Bestand.

[78] Abbildung entnommen und verändert aus: Lexen (2007, S.73)

6.2. Requirement study

Da der Ausgangspunkt dieses Projektes aus einer Kundenanfrage hervorging, wurden Ideen und wichtige Grundfunktionen für diese Applikation in Gesprächen mit dem Kunden erfasst. Die langjährige Erfahrung der Betreuer dieser Arbeit in dieser Branche und die vorangegangene IST-Analyse ließen schließlich die Eckdaten für diesen Client entstehen. Im Folgenden werden die, auf diese Weise gewonnenen, Erkenntnisse zu den einzelnen Geschäftsprozessen unter die Lupe genommen.

Prinzipiell folgen sämtliche Vorgänge dem gleichen Prinzip. Alle Eingaben, die Daten aus dem Stammsystem betreffen, werden sofort nach der Durchführung der Eingabe verifiziert.
Die Daten werden temporär auf dem Gerät gehalten und erst durch aktives „*Beenden*“ des Vorganges über die Schnittstellen in die richtigen Tabellen des Stammsystems eingepflegt.
Jeder der Vorgänge beginnt mit dem *Login* des Benutzers ins System.
Nach einer erfolgreichen Authentifikation befindet sich das Gerät im Status „*logged*“ und es kann zwischen den einzelnen Vorgängen gewählt werden. Die Liste der Vorgänge, welche diese Arbeit behandelt, kann in Zukunft beliebig erweitert werden.
Ein Konfigurationstool bietet die Möglichkeit, Vorgabewerte für den mobilen Client zu pflegen und zu verwalten.
Zusätzlich zu diesen Funktionalitäten ist das SmartCenter mehrsprachig ausgelegt.

Im Folgenden werden sämtliche Usecases, die für die Applikation von Bedeutung sind, erfasst und beschrieben.

6.2.1. Login

Status: *none*
Durch die Eingabe von Benutzername und Kennwort authentifiziert sich der Benutzer des SmartCenters am System. Die Eingaben werden mit dem Stammsystem verifiziert und bei Erfolg wechselt das Gerät in den Status *logged*.
Bei erfolglosem Versuch kann die Eingabe bis zu dreimal wiederholt werden, bevor die Applikation sich selbst beendet.

6.2.2. Konfiguration

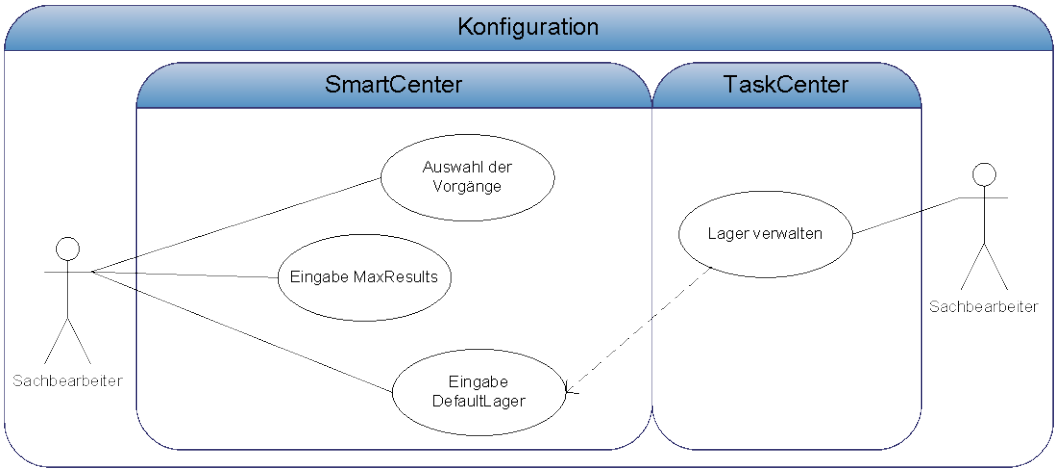


Abbildung 21: Anwendungsfalldiagramm Konfiguration

Das Konfigurationstool bietet die Möglichkeit, diverse Einstellungen der Software am Gerät durchzuführen. Sämtliche Konfigurationen werden in einer XML-Datei am Gerät gespeichert, dadurch sind alle getroffenen Einstellungen geräte- und nicht benutzerspezifisch. Dies macht durchaus Sinn, da Geräte zumeist für spezielle Aufgaben eingesetzt werden und nicht von denselben Benutzern genützt werden. Zum Beispiel wird ein Gerät nur zur Inventur eingesetzt, unabhängig davon wer die Inventur durchführt.

Auswahl der Vorgänge:
Status: *logged*
Hier erfolgt die Auswahl der Vorgänge, die für dieses Gerät im Hauptmenü angezeigt werden sollen.

Eingabe MaxResults:
Status: *logged*
Es kann die Anzahl der maximalen Datensätze, die von den Webservices bei der Kunden- und Artikelsuche zurückgeliefert werden, festgelegt werden. Hiermit soll einem zu hohen Traffic entgegengewirkt werden, dies ist besonders bei langsamen Datenübertragungsraten sinnvoll. Übersteigt die Menge der resultierenden Datensätze den hier eingetragenen Maximalwert, so wird der Benutzer aufgefordert, seine Suchspezifikationen neu zu formulieren.

Eingabe DefaultLager:

Status: *logged*
Für einige Vorgänge können Defaultwerte für Lager/Lagerort hinterlegt werden, um den Vorgang bei Nutzung nur eines einzelnen Lagers zu beschleunigen. Die Eingaben werden sofort überprüft , um festzustellen, ob die vergebene Lagernummer tatsächlich existiert.

6.2.3. Inventur

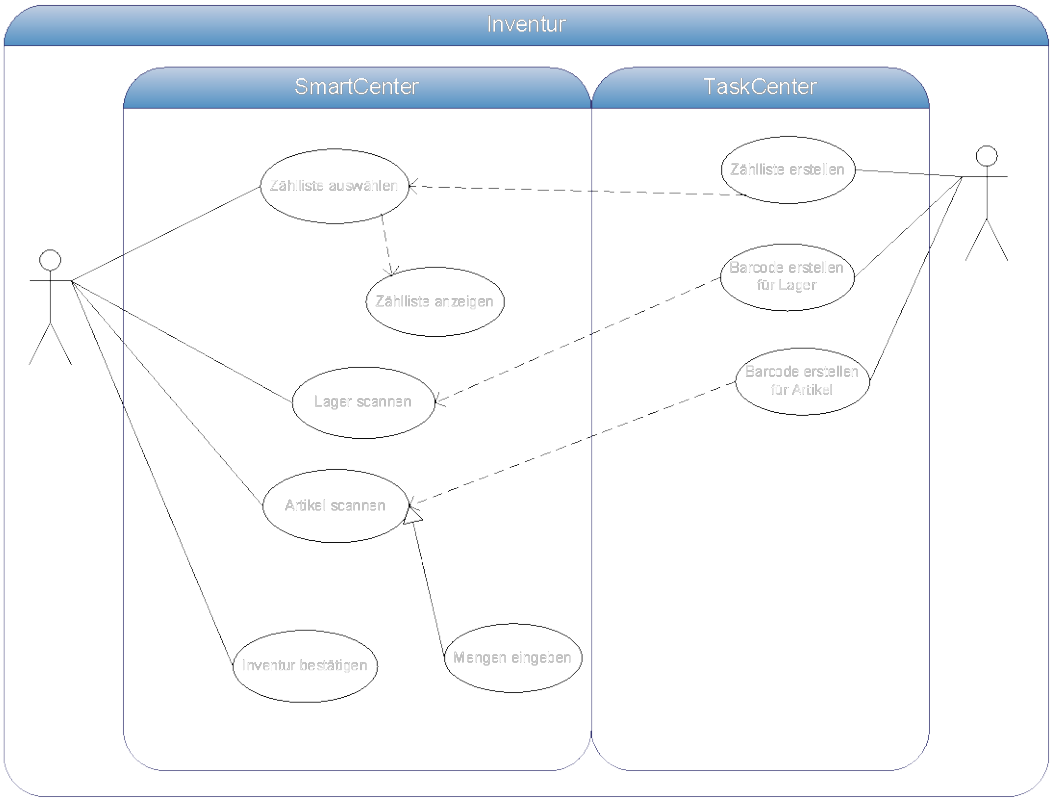


Abbildung 22: Anwendungsfalldiagramm Inventur

Die Anwendungsfälle, die für den Lagerarbeiter zu Verfügung stehen müssen, sind in Abbildung 22: Anwendungsfalldiagramm Inventur dargestellt.

Zählliste auswählen:
Status: *logged*

Zähllisten werden im TaskCenter durch die Auswahl von Lagern bzw. Lagerorten erstellt. Die Artikel, welche laut Artikelkartei dort gelagert sind, werden durch das System automatisch der Liste hinzugefügt. Nachdem die Zähllisten erstellt wurden, können sie im SmartCenter ausgewählt werden. Die Auswahl hat die Anzeige der in der Liste enthaltenen Artikel in einer Tabelle zur Folge. Alternativ kann auch keine Zählliste ausgewählt werden.
Wurde der Schritt der Zähllistenauswahl erfolgreich durchgeführt, wechselt das System in den Status „*bereit*“.

Zählliste anzeigen:
Status: *bereit*
Es werden sämtliche Artikel der Zählliste in einer Tabelle angezeigt, sollte keine Zählliste ausgewählt worden sein, so ist die Tabelle leer.
Außer der Artikelnummer, der Artikelbezeichnung, der Lagernummer, der Lagerortnummer und der Ist-Menge, kann auch die Soll-Menge angezeigt werden. Dies wird im Konfigurationstool festgelegt. Oft ist es nicht empfehlenswert, Sollmengen mit anzugeben, da es sonst passieren kann, dass Mengen nicht mehr gezählt, sondern einfach eingetragen werden.

Lager scannen:
Status: *bereit*
Für die im TaskCenter verwalteten Lager/Lagerorte müssen Barcodes gedruckt und an den Lagerstellen angebracht werden. Somit kann nun der Barcode am Lagerort mit dem Gerät gescannt werden. Hier prüft das System sofort, ob es sich bei dem gescannten Code um ein gültiges Lager handelt, oder ob ein falscher Barcode gescannt wurde. Bei einem Fehlscan ertönt ein akustisches Signal und die Eingabe wird annulliert. Um auch ohne Scannen von Lagercodes die Inventur durchführen zu können, ist es im Konfigurationstool möglich, Default-Werte für Lager zu hinterlegen, welche automatisch ausgewählt werden.
Das ausgewählte Lager wird mit seiner Nummer am Schirm mit angezeigt.
Nach einer gültigen Lagereingabe wechselt das System in den Status *Artikel eingeben.*

Artikel scannen:
Status: *Artikel eingeben*
Ebenso wie für die Lagernummern, sind auch für sämtliche im Artikelstamm befindlichen Artikelnummern, Barcodes zu drucken und an den physischen Artikeln anzubringen. Beim Scannen des Artikelbarcodes wird sofort mit dem Artikelstamm validiert, ob die Artikelnummer tatsächlich vorhanden ist. Bei

einem Fehlscan ertönt ein akustisches Signal, das mit der Annullierung der Eingabe einhergeht.
Artikelscans werden nur zugelassen, wenn ein gültiges Lager ausgewählt wurde, oder ein gültiges Lager aus der Konfiguration ausgelesen werden konnte.
Befindet sich ein Artikel nicht auf der Zählliste, so wird ebenfalls ein akustisches Signal ausgegeben. Weiters erscheint die Meldung „*Artikel nicht auf Liste! Fortfahren?*“. Diese Meldung kann mit „*Ja*“ oder „*Nein*“ bestätigt werden. Im Fall von „Ja“ wird der Artikel in die Tabelle mit aufgenommen, bei „Nein“ wird die Eingabe annulliert.
Bei erfolgreicher Eingabe eines Artikels wird die Zeile in der Tabelle mit Lager, Artikelnummer und Ist-Menge aktualisiert und automatisch am Schirm markiert. Dabei gilt, bei jeder erfolgreichen Artikeleingabe wird die Spalte Ist-Menge um eins inkrementiert.

Mengen eingeben:
Status: *Artikel eingeben*
Die Spalte Ist-Menge kann in der Tabelle manuell editiert werden, wobei hier nur numerische Eingaben zugelassen werden dürfen.

Inventur bestätigen:
Status: *Artikel eingeben*
Die Abfolge der Eingabe der Lagernummer, Artikelnummer und Mengenangabe kann beliebig oft durchgeführt werden. Wurden alle Artikel erfasst, so kann der Benutzer die Inventurerfassung bestätigen, und das SmartCenter aktualisiert nun die Tabelle für die Inventurschnellerfassung.
Ab diesem Zeitpunkt greifen wieder die üblichen Prozesse des TaskCenters und die Arbeit des mobilen Clients ist abgeschlossen. Für einen neuen Eintrag ist neben dem Datentriple Lagernummer, Artikelnummer und Menge, nun auch der Timestamp des Scans unumgänglich. Der Timestamp des Scans ist aus Gründen der Nachvollziehbarkeit wichtig, um den genauen Zeitpunkt der Inventur eines Artikels ermitteln zu können.
Das SmartCenter wechselt danach wieder in den Status *logged* und das Hauptmenü wird angezeigt.

6.2.4. Warenkommissionierung

Eine konsistente und lückenlose Warenkommissionierung ermöglicht korrekte Artikelbestände innerhalb der Artikelkartei und ist somit die wichtigste Grundlage der permanenten Inventur.

Wareneingang:

In Abbildung 22: Anwendungsfalldiagramm Inventur sind sämtliche Anwendungsfälle dargestellt, die für einen Lagerarbeiter zur Wareneingangskommissionierung notwendig sind.

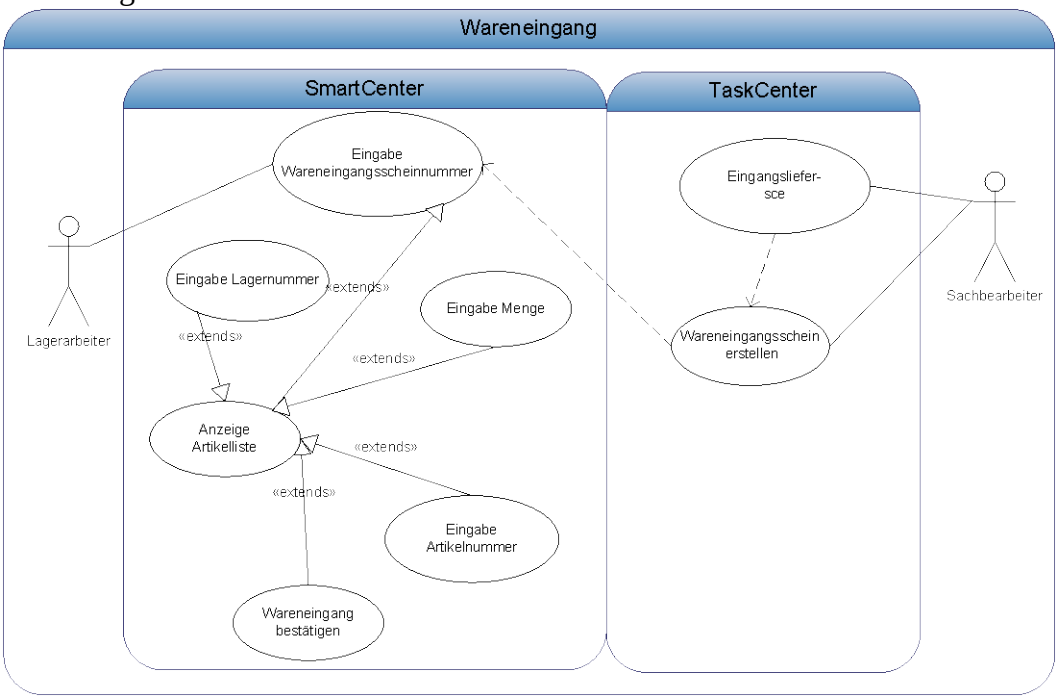


Abbildung 23: Anwendungsfallsdiagramm Wareneingang

Eingabe Wareneingangsscheinnummer

Status: *logged*

Wie in der IST-Analyse bereits beschrieben, wird nach Anlieferung der bestellten Artikel ein Eingangslieferschein im System erstellt. Aus diesem Lieferschein wird je Artikel ein Wareneingangsschein gedruckt, eventuell mit Lagervorschlag versehen. Nun wird im Lager die angelieferte Ware einlagerungsfähig gemacht (Auspacken, auf Paletten verfrachten, etc.).

Den Wareneingangsschein kann der Benutzer des SmartCenters auf dem Gerät durch Eingabe oder Scan des Barcodes der Wareneingangsscheinnummer eindeutig identifizieren.

Die Eingabe wird mit dem TaskCenter validiert und bei Erfolg werden die Artikeldaten und, wenn vorhanden, der Lagervorschlag angezeigt.

Das Gerät wechselt dabei in den Status *bereit*.

Anzeige Artikelliste:
Status: *bereit*
Der Artikel wird am Bildschirm des mobilen Gerätes inklusive optionalen Lagervorschlags angezeigt.
Der Lagerarbeiter kann nun aufgrund der angezeigten Daten, den richtigen Artikel am korrekten Lagerort einlagern.

Eingabe Lagernummer:
Status: *bereit*
Vorraussetzung ist die Vergabe von Barcodes an alle Lagerplätze. Auch hier wird der Scan sofort mit dem Stammsystem validiert. Anhand des Barcodes werden die Lagernummer und die Lagerortnummer ermittelt.
Bei erfolgreicher Eingabe des Lagers wird eine neue Zeile im Tabellenbereich eingefügt und die Spalten *Lagernummer* und *Lagerortnummer* mit den ermittelten Werten befüllt. Es wird keine neue Zeile eingefügt, wenn dasselbe Lager noch einmal gescannt wird, sondern es wird lediglich die Zeile des Lagers markiert.
Das Gerät wechselt in den Status *Artikel einlagern* und ist somit bereit für die Eingabe von Artikeln.
Beim Scannen eines abweichenden Lagerortes wird eine neue Zeile in die Tabelle eingefügt und diese markiert. Dadurch ist es möglich den Wareneingang auf mehrere Lagerorte aufzusplitten.

Eingabe Artikelnummer:
Status: *Artikel einlagern*
Der Lagerarbeiter lagert den Artikel nun an seinem Platz ein, und scannt dabei den Barcode des Artikels.
Dieser Vorgang dient lediglich der Kontrolle, ob der richtige Artikel eingelagert wird. Auch hier wird überprüft, ob die Eingabe eine gültige Artikelnummer ist und wenn ja, ob die Artikelnummer tatsächlich mit dem aktuellen Wareneingangsschein korrespondiert.
Bei erfolgreicher Eingabe der Artikelnummer wird im unteren Tabellenbereich die Spalte *Ist-Menge* aktualisiert.

Eingabe Menge:
Status: *Artikel einlagern*
Die *Ist-Menge* kann manuell eingetragen werden oder wird durch mehrmaliges Scannen der Artikelnummer automatisch inkrementiert.

Ist für jede Zeile im Tabellenbereich der vollständige Datensatz von Lagernummer, Lagerortnummer und Ist-Menge gewährleistet, so kann der Wareneingang bestätigt werden.

Wareneingang bestätigen:
Status: *Artikel einlagern*
Nachdem der Lagerarbeiter den Artikel in der korrekten Menge und am richtigen Platz eingelagert hat, bestätigt er die Einlagerung. Dieses Bestätigen aktualisiert die Eingangslieferscheinpositionen mit den durch die Einlagerung erhaltenen Daten. Im Speziellen werden die erhobenen Mengenangaben mit den richtigen Lagerplatzangaben in die Positionen eingepflegt.
Wie bereits erwähnt, wird bei Übereinstimmung der Mengen vom Eingangslieferschein und vom Lieferantenlieferschein die technische Freigabe des Artikels erwirkt und somit die Einlagerung des Artikels abgeschlossen. Durch dieses Abschließen wird die Eingangslieferscheinposition gesperrt und die Artikelkartei aktualisiert.
Das Gerät wechselt in den Status *logged* und öffnet das Fenster der Wareneingangsscheinauswahl.

Warenausgang:

Wie bereits erwähnt ist ein Auftrag Anstoß für den Prozess der Warenausgangskommissionierung. In den meisten Fällen wird dies ein Auftrag eines Kunden sein und hat somit auch eine Auslieferung zur Folge. Im TaskCenter werden aufgrund des Kundenauftrages die Kommissionsscheine erstellt. Anhand dieser Kommissionsscheine wird nun die physische Kommissionierung, mit Unterstützung der neuen Software, vom Lagerarbeiter durchgeführt.
Die für das SmartCenter relevanten Anwendungsfälle dieses Geschäftsprozesses sind in Abbildung 24: Anwendungsfalldiagramm Warenausgang dargestellt.

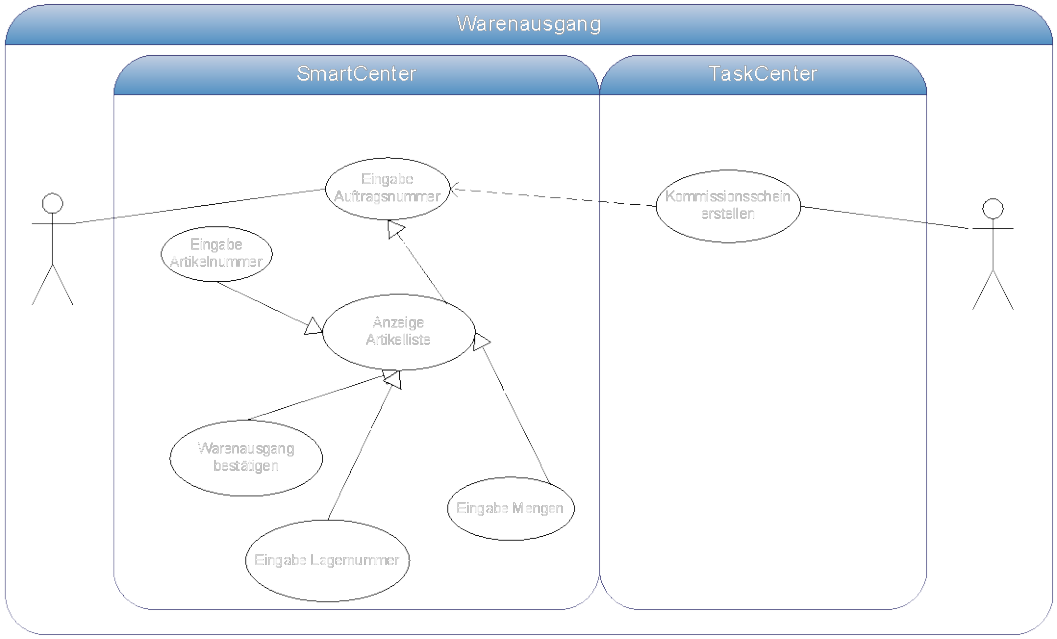


Abbildung 24: Anwendungsfalldiagramm Warenausgang

Eingabe Auftragsnummer:

Status: *online*

Aufgrund der Auftragsnummer können die Auftragspositionen ermittelt und am Gerät angezeigt werden. Die Eingabe kann entweder manuell oder durch das Scannen eines Barcodes erfolgen, welcher auf dem Kommissionsschein angedruckt wird. Das System überprüft bei Eingabe sofort, ob ein derartiger Auftrag (Kommissionsschein) im System existiert. Bei fehlerhafter Eingabe ertönt ein akustisches Warnsignal und der eingegebene Wert wird annulliert. Erst nach erfolgreicher Eingabe einer gültigen Auftragsnummer ist das Gerät bereit für die Warenkommissionierung und wechselt in den Status *bereit.*

Anzeige Artikelliste:

Status: *bereit*

Am Schirm werden sämtliche Positionen des Kommissionsscheines mit ihren Soll-Mengen, Artikelnummern, Bezeichnungen und Lagerorten angezeigt. Um jede Zeile des Kommissionsscheines eindeutig identifizieren zu können, wird der Positionsindex (LSPOSINX) mitselektiert, jedoch nicht am Schirm angezeigt. Ist in der Konfiguration ein Default-Wert für das Lager hinterlegt worden, so wird das Lager am Schirm angezeigt und das Gerät wechselt automatisch in den Status *Eingabe Artikel.*

Eingabe Lagernummer:
Status: *bereit, Eingabe Artikel*
Die Eingabe der Lagernummer ist nicht zwingend erforderlich, sie kann auch als Default-Wert in der Konfiguration hinterlegt werden. Nach Eingabe (oder Scan) der Lagernummer wird sofort überprüft, ob es sich hierbei um eine gültige Eingabe handelt. Ist dies nicht der Fall, wird die Annullierung der Eingabe durch ein akustisches Signal begleitet. Bei erfolgreicher Eingabe wechselt das Gerät in den Status *Eingabe Artikel*.

Eingabe Artikelnummer:
Status: *Eingabe Artikel*
Der Lagerarbeiter kann die am Schirm angezeigten Artikel nur dann bereitstellen und erfassen, wenn ein Lager ausgewählt wurde. Dazu holt er die physischen Artikel aus dem Lager (Scannen des Lagerorts und der Artikelnummer) und deponiert sie am Kommissionierplatz.
Die Artikelnummern können entweder gescannt oder manuell eingegeben werden, wobei Eingaben, welche sich nicht auf der Artikelliste befinden, durch ein akustisches Signal angezeigt und annulliert werden.

Eingabe Mengen:
Status: *bereit*
Die Ist-Mengen können entweder manuell editiert oder durch wiederholtes Scannen der Artikelnummer, um eins inkrementiert werden.

Warenausgang bestätigen:
Status: *bereit*
Nach der Erfassung aller bereitgestellten Artikel und ihrer entsprechenden Mengen kann der Lagerarbeiter die am Schirm angezeigte Liste akzeptieren und somit die Kommissionierung bestätigen. Das Gerät schließt den Vorgang, zeigt das Hauptmenü an und wechselt in den Status *online.*

6.2.5. Auftragserfassung

Für Außendienstmitarbeiter der Lizenznehmer soll der Prozess der Auftragserfassung beim Kunden durch die mobile Auftragserfassung wesentlich vereinfacht werden. In Abbildung 25: Anwendungsfalldiagramm Auftragserfassung sind sämtliche Anwendungsfälle der Auftragserfassung dargestellt.

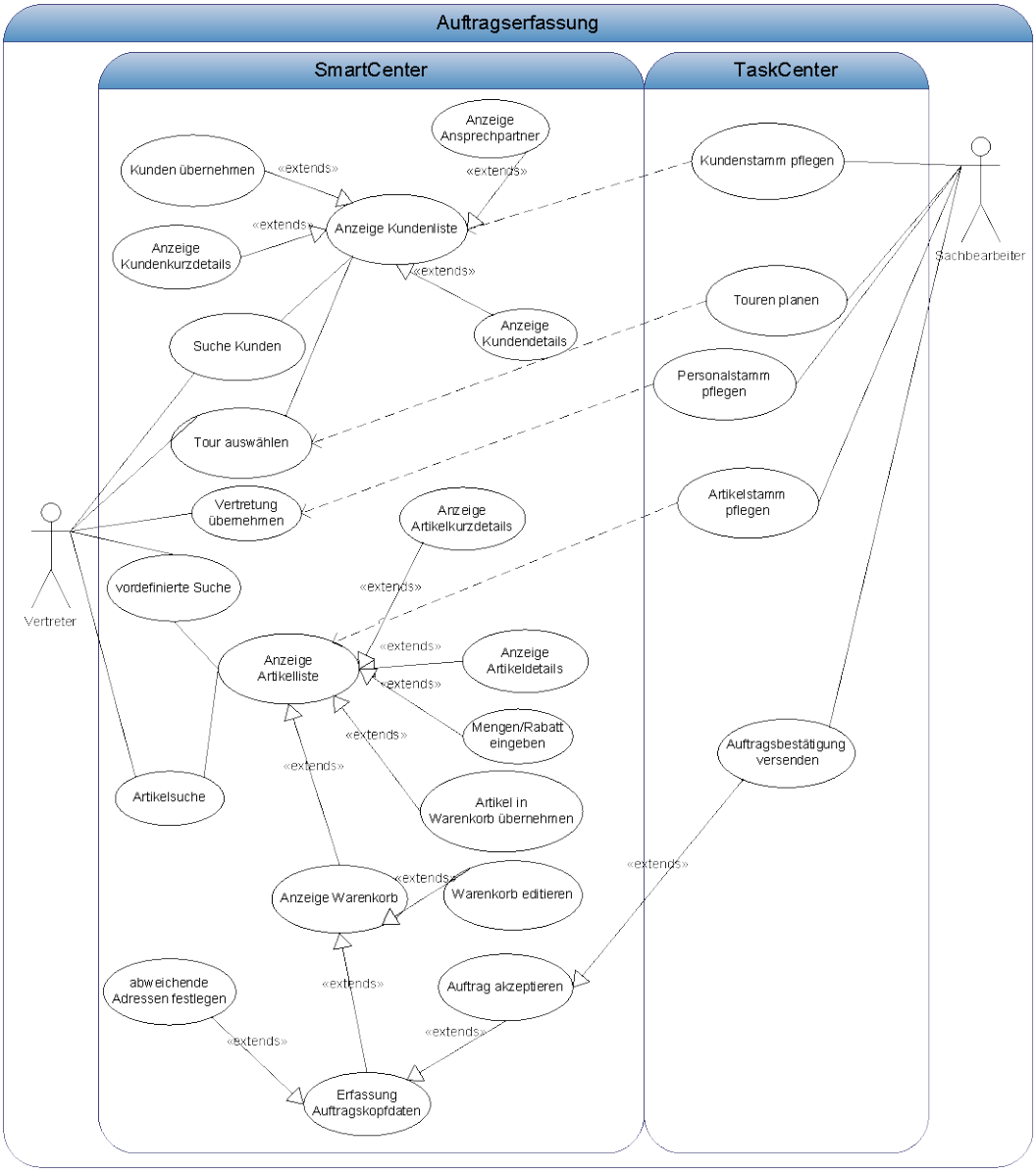


Abbildung 25: Anwendungsfalldiagramm Auftragserfassung

Anzeige Kundenliste:

Status: *logged*

Es werden sämtliche Kunden des eingeloggten Vertreters angezeigt.

Zur Einschränkung der Kundenliste existieren zwei Möglichkeiten:

- Einschränkung durch Eingabe direkter Suchkriterien

- Einschränkung durch Auswahl einer Tour führt zur Anzeige sämtlicher Kunden auf dieser Tour. Vorraussetzung hierfür ist die Erstellung von Touren im TaskCenter.

In beiden Fällen werden die Suchergebnisse in einer Tabelle, der Kundenliste, angezeigt.

Tour auswählen:
Status: *logged*
Das TaskCenter bietet im Kundenstamm die Möglichkeit, Kunden zu Touren zuzuordnen. Diese Touren können im Vorgang Auftragserstellung des SmartCenters ausgewählt werden. Durch die Auswahl einer bestimmten Tour werden auf dem Gerät alle Kunden, die dieser Tour zugeordnet sind, angezeigt.

Suche Kunde:
Status: *bereit, logged*
Für die Suche von Kunden kann zunächst das Kriterium der Suche ausgewählt werden. Mögliche Kriterien für die Kundensuche sind:

- Kundennummer
- Name
- Postleitzahl
- Alle

Nachdem ein Kriterium ausgewählt wurde, kann ein Suchwert eingegeben werden, und das System sucht im Kundenstamm nach Einträgen, die mit dem Suchkriterium und dem Wert übereinstimmen. Die zurückgelieferten Datensätze werden in der Kundenliste angezeigt.

Anzeige Kundenkurzdetails:
Status: *bereit, logged*
Unterhalb der Tabelle der Kundenliste werden für den markierten Kunden kurze Detailinformationen angezeigt. Diese können ausgeblendet werden, um die Kundenliste größer darstellen zu können.

Anzeige Ansprechpartner:
Status: *logged*
Ist in der Kundenliste ein Datensatz markiert, so können in einem eigenen Fenster die Ansprechpartner des markierten Kunden angezeigt werden. Diese Ansprechpartner werden im Kundenstamm des TaskCenters gepflegt.

Kunde übernehmen:
Status: *bereit, logged*
Aus der Kundenliste wird der Kunde markiert und mit *Übernehmen* ausgewählt. Wird ein Kunde ausgewählt, so ist dessen Name, seine Bonität und die Möglichkeit, die Kundendetails zu öffnen, am Schirm immer erkennbar und das Gerät wechselt in den Status *bereit*. Des Weiteren werden kundenspezifische Daten in den globalen Variablen gespeichert.

Anzeige Kundendetails:
Status: *bereit*
Wird ein Kunde übernommen, so können die Kundendetails in einem eigenen Fenster angezeigt werden.
Die Kundendetails beinhalten umfangreiche Detaildaten zu jedem Kunden.

Vertretung übernehmen:
Status: *bereit, logged*
Das SmartCenter bietet des Weiteren eine Vertretungsfunktion, die es einem Mitarbeiter erlaubt, die Kunden bzw. Touren eines Kollegen als dessen Vertretung zu übernehmen. Die Vertretungsberechtigungen werden im Personalstamm des TaskCenters vergeben und können danach am mobilen Client in einem eigenen Fenster angezeigt werden.

Aus diesen möglichen Vertretungen kann schließlich ein Mitarbeiter ausgewählt werden. Die Kundenauswahl muss danach von erneut erfolgen und das Gerät wechselt in den Status *logged*.

Anzeige Artikelliste:
Status: *logged, bereit*
In der Artikelliste werden die Artikel des Artikelstamms angezeigt. Zu Beginn ist die Liste leer und kann durch folgende Möglichkeiten befüllt werden:

- Durch eine vordefinierte Artikelsuche oder
- durch eine individuelle Suche

Hinweis: Artikel, welche sich bereits im Warenkorb befinden, werden grün hinterlegt und die Spalte Menge ist befüllt.

Vordefinierte Suche:
Status: *logged, bereit*

Eine Auswahl von vordefinierten Suchen wird angeboten, um die Artikelsuche für häufig verwendete Suchkriterien zu beschleunigen. Diese Suche ist hardcodiert und kann nur durch den Programmierer erweitert werden. Zur Zeit der Erstellung dieses Dokumentes sind folgende vordefinierte Suchen vorgesehen:

- Suche der zuletzt bestellten Artikeln: Wenn ein Kunde ausgewählt ist, so werden die Artikel der letzten Bestellung angezeigt.
- Suche nach Aktionsartikel: Es werden alle Artikel ausgegeben, welche Sonderpreise hinterlegt haben.

Artikelsuche:
Status: *logged, bereit*
Als zweite Möglichkeit gibt es die Anzeige von Artikeln, die über eine individuelle Suche aufgefunden wurden. Für eine individuelle Suche muss zunächst das Kriterium, nach dem gesucht wird, ausgewählt werden.
Folgende Suchkriterien sind vorgesehen:

- Bezeichnung,
- Artikelnummer,
- Suchbezeichnung,
- Hauptgruppe,
- Sortiment,
- Reihe,
- Gruppe
- Kundenartikelnummer
- Alle

Manche Kriterien haben aufgrund ihrer Funktion vordefinierte Werte, wie zum Beispiel Sortiment[79]. Ist dies der Fall, so werden diese Werte extra zur Auswahl vorgeschlagen.
Ansonsten kann nun eine beliebige Sucheingabe erfolgen und mit der Bestätigung wird im Artikelstamm nach allen Artikeln selektiert, die mit dem eingegebenen Suchwert übereinstimmen.

Für den Einsatz der Scanfunktion des SmartCenters sind bei der individuellen Artikelsuche zwei Vorgehensweisen denkbar:

[79] Anmerkung des Autors: Sortiment ist ein Kriterium der standardisierten Artikelprofilierung und kann im Stammsystem vom Benutzer mit beliebigen Werten verwaltet werden. Sämtliche dort verwaltete Werte werden dann im SmartCenter zur Auswahl angeboten.

- Durch die manuelle Eingabe oder durch das Scannen eines Barcodes kann die Artikelnummer eingegeben werden. Vorstellbar für diesen Vorgang wäre, dass bei häufig gewählten Artikeln, die Nummern bereits bekannt sind, oder durch das Scannen aus einem Katalog mit Barcodes ausgesucht werden.
- Durch die manuelle Eingabe oder Scannen eines Barcodes kann die Kundenartikelnummer eingegeben werden. Zum Beispiel besichtigt ein Vertreter mit dem Kunden das Lager oder ein Regal vorort und scannt jede Artikelnummer, die nachbestellt werden muss. Ausschlaggebend ist hier die kundenspezifische Artikelnummer und nicht die tatsächliche Artikelnummer.

Anzeige Artikeldetails:
Status: *logged, bereit*
Innerhalb der Tabelle der Artikelliste können durch Betätigen des PictureButtons für Artikeldetails, sämtliche Details zu einem bestimmten Artikel in einem eigenen Fenster angezeigt werden.

Anzeige Artikelkurzdetails:
Status: *logged, bereit*
Es kann unterhalb der Artikelliste ein Bereich eingeblendet werden, wo die wichtigsten Artikeldetails für den in der Liste markierten Artikel angezeigt werden. Falls ein Kunde ausgewählt ist, wird auch der kundenspezifische Preis für diesen Artikel ausgewiesen.

Mengen angeben:
Status: *logged, bereit*
Innerhalb der Liste ist eine Spalte für die Eingabe der Menge, mit welcher ein Artikel in den Warenkorb übernommen werden soll, reserviert. Hier sind lediglich numerische Einträge zulässig.

Artikel in Warenkorb übernehmen:
Status: *bereit*
Diese Funktion ist nur aktiv, wenn ein Kunde ausgewählt wurde.
Für die Übernahme in den Warenkorb existieren zwei Möglichkeiten. Zum Einen können einzelne Artikel in den Warenkorb übernommen werden, dann bleibt die Artikelanzeige geöffnet. Zum Anderen kann der derzeitige Anzeigeschirm mit allen Artikeln, die eine Mengenangabe von „*> 0*“ besitzen, in den Warenkorb

übernommen werden. In diesem Fall wird die Artikelanzeige verlassen und der Warenkorb geöffnet.

Anzeige Warenkorb:
Status: *bereit*
Im Warenkorb werden sämtliche übernommene Artikel mit ihren Mengen angezeigt. Hier kann der Warenkorb editiert, oder zurück in die Artikelsuche gewechselt werden, um weitere Artikel hinzuzufügen.
Ebenfalls wird die Gesamtsumme und das Gesamtgewicht der sich im Warenkorb befindlichen Artikel am Schirm ausgegeben.

Warenkorb editieren:
Status: *bereit*
Innerhalb des Warenkorbs werden Funktionen zur Verfügung gestellt, mit deren Hilfe der Warenkorb editiert werden kann. Zu diesen Funktionen zählen:

- Editieren der Mengen
- Löschen von Artikeln
- Wechsel in die Artikelsuche, um Artikel hinzuzufügen

Ist der Warenkorb korrekt für die Anforderungen des Kunden zusammengestellt, so kann dieser übernommen werden und die Auftragskopfdatenerfassung wird geöffnet.

Erfassung der Auftragskopfdaten:
Status: *bereit*
Innerhalb der Kopfdatenerfassung können Lieferdatum, Zahlungs- und Lieferbedingung, Versandart und Schlussrabatt festgelegt werden. Ebenso können zusätzliche Informationen als Freitext zu dem Auftrag eingegeben werden.

Abweichende Adressen festlegen:
Status: *bereit*
Innerhalb der Kopfdatenerfassung kann auch ein Fenster für die Festlegung von Rechnungs- und Lieferadresse geöffnet werden. Hier werden sämtliche im Kundenstamm hinterlegte Adressen zur Auswahl angeboten.

Auftrag bestätigen:
Status: *bereit*

Wurden alle relevanten Daten für den Auftrag erfasst, so kann der Auftrag bestätigt werden. Das SmartCenter übermittelt die Daten an das Stammsystem, wo der Auftrag sofort angelegt und eine Auftragsnummer dafür vergeben wird.
Nach erfolgreicher Anlage des Auftrages wird die *Auftragsnummer* angezeigt und somit der Prozess der Auftragserfassung beendet. Mit dem Beenden des Prozesses wechselt das Gerät in den Status *logged* und das Hauptmenü wird angezeigt.

6.3. Entwicklungsumgebung

Für die Entwicklung der Applikation wird das Microsoft Visual Studio 2005 verwendet. Diese Umgebung ist der Standardeditor von Microsoft für das .NET Framework. Da jedoch, wie bereits eingangs dieser Arbeit erwähnt, das .NET Compact Framework eingesetzt wird, sind Erweiterungen für das Studio notwendig. Diese Erweiterungen werden von Microsoft als *Smart Device Extensions (SDE)* zur Verfügung gestellt. Dabei ist die Bezeichnung *Extensions* auch als solche zu bewerten, da es sich hierbei um keine *standalone* - Variante handelt, sondern das Microsoft Visual Studio oder Microsoft Visual Basic als Grundlage vorausgesetzt wird. SDE beinhaltet einen Editor zum Designen von Compact Framework Formularen, sowie Emulatoren von Mobile Devices zum Testen ohne tatsächliches Endgerät.[80] Siehe Abbildung 26: Pocket PC Emulator

[80] Vgl. MSDN – Smart Device Extensions

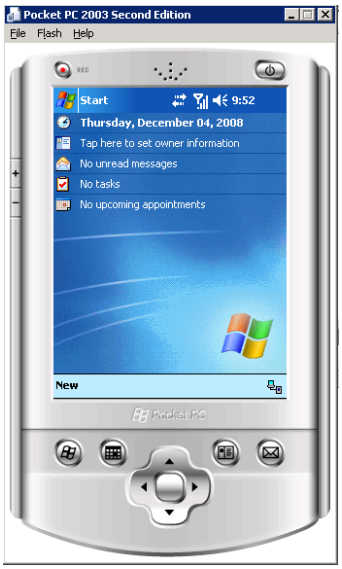


Abbildung 26: Pocket PC Emulator

Code Management

Die Umsetzung des Code-Managements und der Versionskontrolle erfolgt mit Hilfe von Microsoft Visual SourceSafe.

Das von Microsoft zur Verfügung gestellte Versionsverwaltungssystem Visual SourceSafe garantiert für verschiedene Organisationen die parallele Entwicklung an mehreren Versionen auf Dateiebene.[81]

Es unterstützt sowohl die Nachvollziehbarkeit von Dateiänderungen als auch, für die Softwareentwicklung von großer Bedeutung, Funktionen für die Wiederverwendung von Codesnipplets.

Folgende Hauptpunkte werden durch SourceSafe sichergestellt:[82]

- Schutz vor Daten- beziehungsweise Dateiverlust.
- Trackingfunktionen zur Rückverfolgung von früheren Dateiversionen.
- Aufteilung, Freigabe, Merging und Verwaltung unterschiedlicher Versionen.
- Nachverfolgung von einzelnen Modulen oder von gesamten Projekten.

[81] Vgl. MSDN – Visual Source Safe
[82] Vgl. MSDN – Visual Source Safe

6.4. Datenbankzugriff

Innerhalb des TaskCenters erfolgen sämtliche Datenbankzugriffe über Webservices und zwar ausschließlich mit *Oracle* und *SQLServer*, welche in .NET beinhaltet sind.
Die Abfragen auf der Datenbank werden von spezifischen Funktionen der Provider in ihren DAL-Layern (Data Access Layer) durchgeführt. (Abbildung 27: Datenbankzugriff vereinfacht)
Diese spezifischen Layer werden von der abstrakten Klasse DAL abgeleitet. Der *DataAccesLayer* wiederum wird von der Klasse *DBConnect*, welche die Methoden für die Programmierung zur Verfügung stellt, benutzt. Da sich Oracle und SQLServer in der Syntax der Abfragesprache unterscheiden, wird von den Programmierern ausschließlich mit der Oracle-Syntax gearbeitet. Die Übersetzung für SQLServer-Datenbanken erfolgt automatisch im DAL-Layer.[83]

[83] Vgl. Lexen, S. 86

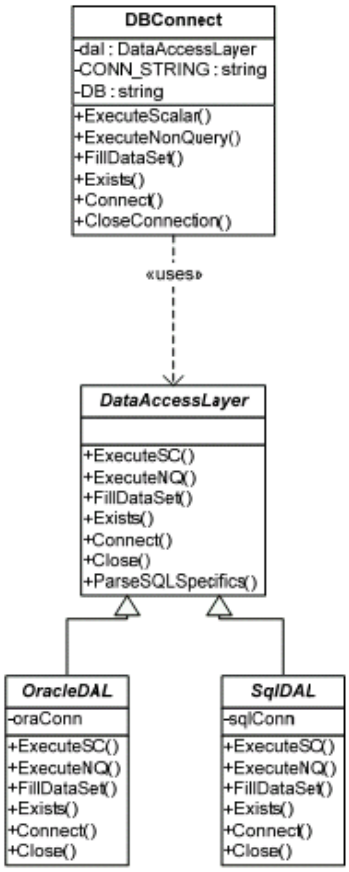


Abbildung 27: Datenbankzugriff vereinfacht[84]

6.5. Webservices

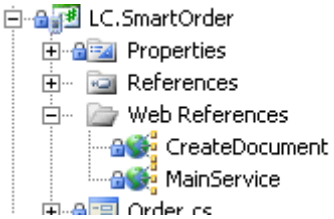


Abbildung 28: Webservicereferenzierung

[84] Abbildung entnommen aus: Lexen (2007:S.87)

Anders als beim TaskCenter, wo die Instanzierung der Webservices in einer globalen Klasse durchgeführt wird, erfolgt dies beim SmartCenter direkt in der Klasse, in welcher der Webservice aufgerufen werden soll. Dies bedeutet, dass der Webservice in den Webreferenzen jedes einzelnen Projektes, das den Service verwendet, referenziert werden muss (siehe Abbildung 28: Webservicereferenzierung):

```
CreateDocument.Warenkorb cd = new
LC.SmartOrder.CreateDocument.Warenkorb();
```

Die Zuweisung der URL des Webservices erfolgt vor dem eigentlichen Serviceaufruf:

```
ms.Url = LC.SmartGlobals.Globals.urlMainService;
```

In der Konfigurationsdatei *config.xml* wird die URL des Webservices *WebServiceUrl* gespeichert, welcher auf die Konfigurationsdatei des TaskCenters *WebServiceUrlSettting.config* zugreift. Bei der Installation des TaskCenter-Servers werden in dieser Datei die URL's aller zur Verfügung stehender Webservices eingetragen.
Nach dem Login des SmartCenters wird über den *WebServiceUrl* die *WebserviceUrlSetting.config* ausgelesen und die ermittelten Werte werden im XML-Format (*doc_node*) zurückgegeben und schließlich in den globalen Variablen des SmartCenters gespeichert:

```
WebServiceUrl.WebServiceUrl ws = new
LC.SmartLogin.WebServiceUrl.WebServiceUrl();
ws.Url = urlWebService;
doc_node = ws.GetWebServiceUrl();

SmartGlobals.Globals.SetWebServiceUrl(doc_node);
```

Ein typischer Webserviceaufruf im SmartCenter hat folgende Struktur:

```
DataSet ds = ms.FillDataSet(LC.SmartGlobals.Globals.CONN_STRING,
LC.SmartGlobals.Globals.DB_TYPE, sqlSelect, out ex);
```

Innerhalb des SmartCenter-Projektes gilt grundsätzlich, dass für einen Aufruf drei Parameter Voraussetzung sind:

- CONN_STRING: Über den verschlüsselten Connection-string werden sämtliche, für einen erfolgreichen Datenbankzugriff notwendige, Daten an den Webservice übergeben.

- DB_TYPE: Definiert den Datenbanktyp (Oracle, SqlServer)
- OUT-Parameter: Falls Fehler während der Ausführung des Webservice auftreten, wird die entsprechende Fehlermeldung des IIS an diesen Parameter zurückgegeben.

6.6. gerätespezifische Konfiguration

Konfigurationseinstellungen, WS-Adressierungen und Sprache werden in XML-Dateien auf dem Gerät gespeichert. Das .NET Compact Framework stellt Klassen und Methoden zum Generieren und Auslesen von XML-Dateien zur Verfügung.
Für das SmartCenter wurden die Daten für die Weiterverarbeitung auf zwei unterschiedlichen Wegen ausgelesen:

- direktes Auslesen eines konkreten Knotens innerhalb der XML-Datei oder
- Einlesen der kompletten XML-Datei in ein strukturgleiches Dataset.

6.6.1. Scanfunktion

Falls die Scanfunktion des Gerätes verwendet werden soll, muss in den Einstellungen ein Abschluss-String für die eingescannte Zeichenkette angegeben werden. Für das SmartCenter ist dies der ASCII-Code[85] 13 (Enter).

6.6.2. Config-file

Innerhalb der *config.xml* werden sämtliche im Konfigurationstool getroffenen Einstellungen gespeichert. Zusätzlich zu den vorgangsspezifischen Einstellungen wird hier auch die URL des Webservice *WebserviceUrl* und der Name der Datenbank angefügt.

[85] American Standard Code for Information Interchange

Ein Ausschnitt aus der *config.xml*:

```
<?xml version="1.0" encoding="utf-8" ?>
<Settings>
  <URL
value="http://programmierung/LCWebServices/LC.WebServiceUrl/WebSer
viceUrl.asmx"></URL>
  <DB value="LC5"></DB>
  <ORDER>

    <VISIBLE value="1"></VISIBLE>
    <MAXRESULTS value="100"></MAXRESULTS>
    <MAXDISCOUNT value="3"></MAXDISCOUNT>
    <DEFAULTSEARCH value="9"></DEFAULTSEARCH>
  </ORDER>
  <INVENTUR>
    …
  </INVENTUR>
  <KOMM>
    …
  </KOMM>
</Settings>
```

Die URL der Webservices, sowie die Angabe der Datenbank müssen manuell in der *config.xml* editiert und auf das Gerät kopiert werden.

6.6.3. Multilanguage

Da das POLLEX-LC TaskCenter eine international eingesetzte Software ist, welche volle Unterstützung in verschiedenen Sprachen bieten muss, ist auch für das SmartCenter eine multilinguale Lösung vorgesehen. Das gesamte Softwarepaket des TaskCenters wird standardmäßig in den Sprachen Deutsch und Englisch ausgeliefert. Im Softwarepaket beinhaltet ist jedoch ein Übersetzungstool, mit dessen Hilfe die gesamte Software in jede beliebige Sprache übersetzt werden kann. (siehe Abbildung 29: Übersetzungstool)

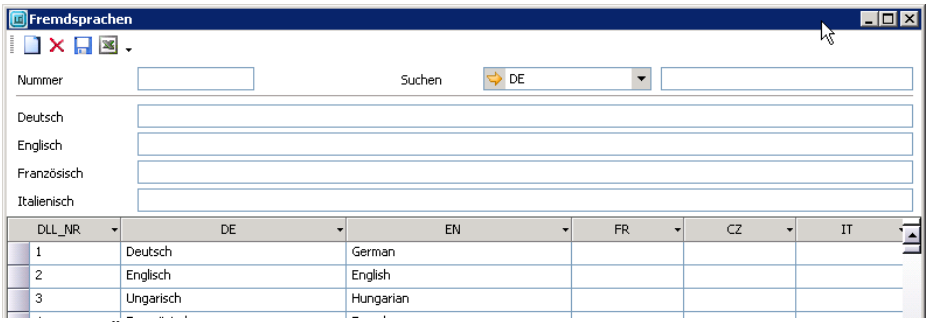


Abbildung 29: Übersetzungstool

Aus diesem Übersetzungstool können nun für die unterschiedlichen Sprachen XML-Dateien exportiert werden. Dies bedeutet, dass für jede Sprache eine eigene XML – Datei angelegt wird, die folgende Struktur aufweist:

```
<?xml version="1.0" standalone="yes"?>
<Root xmlns="http://www.pollex-lc.com/SpracheSchema.xsd">
  <DLL_NR nummer="1">
    <Text>Deutsch</Text>
    <Beschreibung>Sprachbezeichnung</Beschreibung>
  </DLL_NR>
  <DLL_NR nummer="2">
    <Text>Vertreter</Text>
  </DLL_NR>
  <DLL_NR nummer="3">
    <Text>Kundensuche</Text>
  </DLL_NR>
```

Der <DLL_NR> – tag beinhaltet die systemweit eindeutige *dll*-Nummer für die im <TEXT> - tag gespeicherte Zeichenkette. Die so erstellte XML-Datei wird vor Start der Anwendung in das Ausführungsverzeichnis der Software kopiert. Beim Login wird nun nach einer solchen Sprachdatei gesucht und, wenn die Suche erfolgreich war, der Inhalt dieser Sprachdatei in den globalen Variablen in ein Dataset gespeichert.
Es werden nun sämtliche Titel und Feldbezeichner in allen Fenstern des Benutzerinterfaces auf folgende Weise belegt.

```
LC.SmartGlobals.Languages.GetText(dll_nr, "defaultText");
```

Die Methode *GetText(dll_nr, „text“)* aus den *SmartGlobals.Languages* liefert nun die zur übergebenen *dll*-Nummer gespeicherte Zeichenkette. So wird, je nachdem

welche Sprachdatei geladen wurde, die korrekte Zeichenkette für diese Sprache zurückgeliefert.

```
public static string GetText(int dllnr, string text)
{
  if (ds != null)
  {
    if (ds.Tables.Count > 0)
    {
      DataRow[] rows = ds.Tables[0].Select("NUMMER='"
           + dllnr.ToString() + "'");
      if (rows.Length > 0)
        return rows[0].ItemArray[0].ToString();
      else return text;
    }
    else throw new Exception("Sprachdatei wurde nicht
      initialisiert");
  }
  else return "";
}
```

6.7. Indexverwaltung

Standardmäßig werden in fast allen Tabellen der Datenbank Indizes als Primärschlüssel mitgespeichert. Für die Erlangung von uniquen Indizes wurde eine Webservicemethode entwickelt, die bei Aufruf den neuen Index zurückliefert:

```
LC.Globals.WebServices.miscService.NeuerIndex(LC.Globals.Globals.C
ONN_STRING, LC.Globals.Globals.DB_TYPE, "TANINX", out ex);
```

Sämtliche Indizes werden in einer eigenen Datenbanktabelle gespeichert. Der Webservice inkrementiert den angeforderten Index, speichert den neuen Wert in die Datenbank und gibt ihn zurück.

6.8. Schnittstellen

Im Folgenden werden für die einzelnen Geschäftsprozesse die Schnittstellen, über die die erfassten Daten wieder in das Stammsystem eingepflegt werden, erläutert

Wareneingang:

Tabelle 6: Datensatz Wareneingang

Spalte	Typ	Bemerkung
ARTIKELNUMMER	String	Artikelnummer des gescannten Artikels
LAGER	Float	Lagernummer des gescannten Lagers
LAGERORT	Float	Lagerortnummer des gescannten Lagerortes
STUECK	Float	Menge der eingelagerten Artikel in Lagermengeneinheit
WES_NUMMER	String	Systemweit eindeutige Wareneingangsscheinnummer

Ein vollständiger Datensatz für die Wareneingangskommissionierung besteht aus den in Tabelle 6: Datensatz Wareneingang zusammengefassten Daten. Da, wie bereits beschrieben, die Wareneingangsscheinnummer einem eindeutigen Positionsindex innerhalb der Eingangslieferscheinpositionen entspricht, kann die korrekte Zeile innerhalb der Positionen über die *WES_NUMMER* angesprochen werden. Dies bedeutet, dass jede Zeile innerhalb des Datensatzes einer Zeile innerhalb der Tabelle *LS_WEPOSITIONEN* der Datenbank entspricht. Die Zeile innerhalb der Datenbank wird nun durch ein Update-Statement, den Wareneingängen entsprechend, aktualisiert:

```
update set LS_WEPOSITIONEN (stueck,lager,lagerort) values
(stueck,lager,lagerort) where wes_nummer = weposinx
```

Durch das Aktualisieren der Tabelle *LS_WEPOSITIONEN* in der Datenbank sind nun sämtliche, durch die mobile Datenerfassung erhaltenen, Daten wieder korrekt im System eingepflegt und stehen für sämtliche Funktionen der Standardsoftware zur Verfügung.

Ein wichtiger Punkt muss ebenfalls noch berücksichtigt werden. Wird ein Artikel, also ein Wareneingangsschein, an mehreren Lagerorten gelagert, so liefert dieser Wareneingangsschein mehrere Zeilen im Datensatz zurück. Dies bedeutet, dass der Webservice, der diesen Datensatz übernimmt, nun eine zusätzliche Zeile in den *LS_WEPOSITIONEN* mit einem neuen eindeutigen Positionsindex einfügen

muss. Die bereits in den Eingangslieferscheinpositionen vorhandene Artikelposition wird mit der Menge und dem Lagerort des als erstes eingelagerten Artikels, also der ersten Zeile des gescannten Datensatzes, befüllt. Für jede weitere Zeile des Datensatzes wird eine neue Artikelposition in den Eingangslieferscheinen hinzugefügt und mit der Artikelnummer, der Bezeichnung, der Einlagerungsmenge und dem Lagerort befüllt. Damit für sämtliche folgenden Zeilen der Zusammenhang zu diesem Wareneingangsschein erhalten bleibt, existiert in der Tabelle ein Feld *STKL_INDEX*, in dem der Positionsindex der Hauptzeile gespeichert wird. Diese Nebenzeilen werden über folgendes Statement in die Tabelle eingefügt:

```
insert into LS_WEPOSITIONEN stueck =
stueck,lager=lager,lagerort=lagerort,stkl_index = wes_nummer,
weposinx = NeuerIndex(„WEPOSINX“))
```

Warenausgang:

Tabelle 7: Datensatz Warenausgang

Spalte	Typ	Bemerkung
ARTIKELNUMMER	String	Artikelnummer des gescannten Artikels
MENGE	Double	Menge der kommissionierten Artikel
AUFTRAGSNUMMER	Double	Kundenauftragsnummer des kommissionierten Artikels
LAGER	Double	Nummer des Lagers
LAGERORT	Double	Nummer des Lagerorts
POSINX	Double	Positionsindex

Die für einen kompletten Datensatz des Warenausgangs minimal erforderlichen Daten sind in Tabelle 7: Datensatz Warenausgang zusammengefasst.
Die Tabelle der Lieferscheinpositionen (Kommissionsscheinpositionen) muss nun mit den Werten des erfassten Datensatzes aktualisiert werden. Hierzu wird für jede Zeile innerhalb des Datensatzes ein Update-Statement in der Datenbank durchgeführt. Zu beachten ist, dass nur vollständige Datensätze in die Datenbank gespeichert werden.

```
update set LSPOSITIONEN
(auftragsnummer,artikelnummer,stueck,lager,lagerort) values
(auftragsnummer,menge,lager,lagerort) where posinx = lsposinx
```

Mit dem Eintrag in die Datenbanktabelle *LSPOSITIONEN* ist man wieder im standardisierten Prozessablauf des Warenausganges.

Inventur:

Tabelle 8: Datensatz Inventur

Spalte	Typ	Bemerkung
ZEITPUNKT	Timestamp	Speichert den Zeitpunkt des durchgeführten Scans
ARTIKELNUMMER	String	Artikelnummer des gescannten Artikels
LAGER	Double	Lagernummer des gescannten Lagers
LAGERORT	Double	Lagerortnummer des gescannten Lagerortes
MENGE	Double	Menge der eingelagerten Artikel
LISTENNUMMER	Double	Zählistennummer

Das SmartCenter muss für eine korrekte Inventur eines Artikels einen vollständigen Datensatz, wie er in Tabelle 8: Datensatz Inventur dargestellt ist, liefern.
In der Spalte Listennummer wird die Nummer der Zählliste mitgespeichert, dieses Feld kann auch leer sein, wenn ein Artikel ohne Zählliste für die Inventur erfasst wurde.
Die übermittelten Datensätze werden in die Tabelle Inventurschnellerfassung, *INVM*, mit folgendem Statement gespeichert:

```
insert into INVM anr = Artikelnummer, l = lager, lo = lagerort,
stk = menge, nummer = listennummer, bu_datum = zeitpunkt
```

Auftragserfassung
Der Prozess der Auftragserfassung gestaltet sich etwas komplexer und erfordert daher mehrere Schnittstellen. Es wird während des gesamten Ablaufs der Erfassung ein eigener Transaktionsindex (*TANINX*), welcher in den globalen Variablen gespeichert wird, für eine Session angelegt.

```
LC.SmartGlobals.Globals.TanInx =
MiscService.NeuerIndex(LC.SmartGlobals.Globals.CONN_STRING,
LC.SmartGlobals.Globals.DB_TYPE, "TANINX", out ex);
```

Da das TaskCenter für den Webauftritt des Kunden einen eigenen Web-Shop anbietet, existiert bereits ein zuverlässiger Webservice für Warenkorbfunktionen, der *BasketService*. Diese etablierte Standardmethode wird verwendet, um Artikelpositionen in die Tabelle *WARENKORB* einzufügen.

```
public double Insert(string CONN_STRING, string DB_TYPE, string
itemNr, double quant, int taninx, int custNo, double discount, int
empno, out string ex)
```

Die Übergabeparameter setzen sich dabei aus der Artikelnummer, der einzufügenden Menge, dem Transaktionsindex, der Kundennummer, dem Positionsrabatt und der Mitarbeiternummer zusammen. Statt einer einzelnen Artikelnummer kann auch in einer eigenen Überladung des *BasketService.Insert* ein Dataset übergeben werden. Dieser Dataset beinhaltet ein oder mehrer Artikelnummern mit zugehöriger Menge und Positionsrabatt.
Nach Bestätigen der Auftragskopfdaten wird zunächst das erfasste Lieferdatum zu den Warenkorbdaten hinzugefügt. Die eindeutige Identifizierung des Warenkorbs erfolgt über den Transaktionsindex.

```
string sqlSelect = "update warenkorb set lieferdatum = to_date('"
+ dDate + "','dd.MM.yyyy')" + " where taninx=" +
LC.SmartGlobals.Globals.TanInx;
```

Weiters wird ein Auftrag mit Hilfe des *BasketService.CreateDocument* im Stammsystem angelegt:

```
public bool CreateDocument(
                    string CONN_STRING,
                    string DB_TYPE,
                    string USER,
                    double nTaninx,
                    double nKundennummer,
                    double nPersonalnummer,
                    int nTyp,
                    string sKategorie,
                    string sVersandart,
                    double nUseDocumentNo,
                    bool bTFG,
                    bool bWebAuftrag,
                    double nSchlussrabatt,
                    string spBestellnummer,
                    string sVereinbarung,
                    string sLieferbedingung,
                    string sZahlungsbedingung,
                    int nLieferadresseAus,
                    double nLieferadresse,
                    int nRechnungsadresseAus,
                    double nRechnungsadresse,
                    out double nDocument,
                    out string ex)
```

Bei erfolgreicher Durchführung des Webservices wird mit *nDocument* die Auftragsnummer zurückgeliefert, mit der der Auftrag im Stammsystem, mit dem kompletten Warenkorb als Auftragspositionen, angelegt wurde.

6.9. User interface

6.9.1. Usability

„Usability is the measure of the quality of the user experience when interacting with something – whether a web site, a traditional software application, or any other device the user can operate in some way or another“ - Jacob Nielsen, 1998[86]

Da sich die Displaygrößen bei mobilen Geräten hinsichtlich ihrer Dimension in Grenzen halten, sind hier besondere Anforderungen an das User Interface gestellt. Prinzipiell wird in der Usability zwischen drei Prinzipien unterschieden:[87]

[86] Jacob Nielsen wurde 1957 in Kopenhagen, Dänemark geboren und gilt weltweit als Experte auf dem Gebiet der Mensch-Maschine-Kommunikation.
[87] Vgl. INSO – Usability Engineering

- Effektivität: Wurde das Ziel erreicht?
- Effizienz: Mit welchem Aufwand wurde das Ziel erreicht?
- Zufriedenheit: War man über den Verlauf der Zielerreichung zufrieden?

Zunächst versucht man die Benutzer zu definieren, das heißt, welche Benutzergruppen werden direkt oder indirekt mit dieser Applikation arbeiten. Als nächstes gilt es, die Aufgaben zu ermitteln, die an das User Interface gestellt werden und diese zu kapseln. Für die Ermittlung der Anforderungen können die bereits beschriebenen Usecases verwendet werden. Es existieren unterschiedliche Techniken, um für die ermittelten Usecases eine geeignete Oberfläche zu gestalten:

- Card Sorting ist eine Methode, bei der durch Sortieren der Anwendungsfälle, durch einen oder mehrere repräsentative Benutzer, die Funktionalitäten zu Gruppen zusammengefasst werden. Es hilft das „mentale“ Modell eines Benutzers zu verstehen.
- Contextual Inquiry ist das Beobachten der Benutzer in ihrer alltäglichen Arbeitsumgebung. Wichtig ist das Nichtbeachten eigener Vorstellungen und Erwartungen.
- Interview und Fragebögen: Interviews mit den Benutzern unterstützt durch vorgefertigte Fragebögen helfen bei der Erstellung des Konzepts der grafischen Oberfläche.
- Fokusgruppen: Hier wird ein Team gebildet, welches innerhalb einer Diskussionsrunde das Design entwirft.

Im vorliegenden Fall des SmartCenters wird das Design des Userinterfaces innerhalb einer kleinen Fokusgruppe entworfen. Innerhalb dieser Diskussionsrunde werden die Eckdaten für die grafische Oberfläche besprochen und schlussendlich für den Prototyp festgelegt.

6.9.2. Window-management

Das Management der Fensternavigation wird aus einem Hauptfenster *Main.cs* durchgeführt, welches in diesem Fall das Hauptmenü darstellt.
Das Hauptmenü beinhaltet nicht nur die Navigation zu den einzelnen Vorgängen, sondern fungiert auch als Container sämtlicher Child-Fenster. Durch seine Funktion als Container läuft die Kommunikation der Child-Fenster untereinander über das Hauptmenü.

Reaktionen auf Ereignisse innerhalb der Child-Fenster werden in den meisten Fällen durch das Hauptmenü durchgeführt, deshalb werden sämtliche Vorgangsfenster aus dem Hauptmenü geöffnet und alle notwendigen EventHandler dort angelegt:

```
void items_OpenCustomer(object sender, EventArgs e)
{
  if (this.customer == null)
  {
    //Fenster zum ersten Mal öffnen
    LC.SmartGlobals.Methods.ShowWaitCursor(true);
    customer = new LC.SmartCustomer.Customer();
    customer.OpenItems +=new
      EventHandler(customer_OpenItems);
    customer.Show();
    LC.SmartGlobals.Methods.ShowWaitCursor(false);
  }
  customer.BringToFront();
}
```

Vorteil der Abwicklung der Kommunikation über eine Hauptklasse als Eventbroker ist, dass so keine Referenzen der Child-Fenster zueinander notwendig sind. Zirkuläre Referenzen können auf diese Weise vermieden werden und es unterstützt die modulare Aufbauweise der gesamten Software.
Ein Beispiel für das Eventhandling ist in Abbildung 30: Beispiel für Eventhandling dargestellt.[88]

[88] Vgl. Lexen, 2007, S. 90

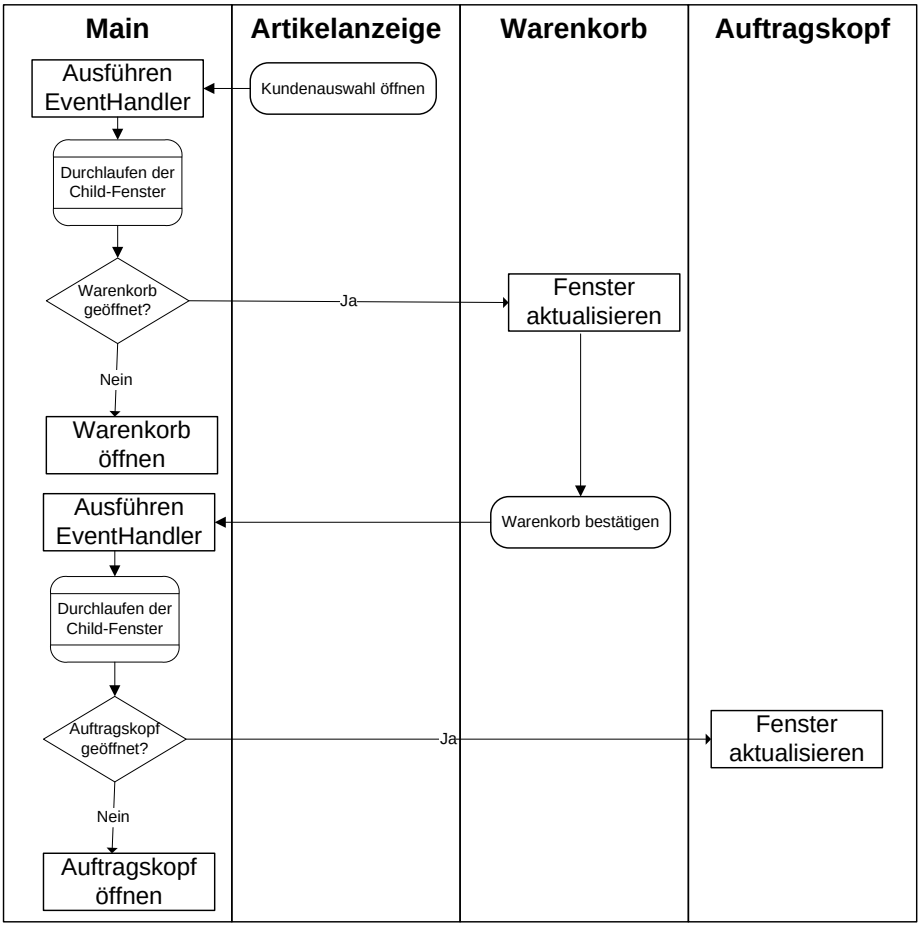


Abbildung 30: Beispiel für Eventhandling[89]

6.9.3. SmartControls

Die Standardtools des .NET Compact Framework bieten oft nicht genügend Funktionalität, aus diesem Grund werden eigene Tools entworfen, welche vom Standard abgeleitet und *customized* werden.
Diese, eigens für das SmartCenter entwickelten, Controls sind innerhalb der Klasse *SmartControls* definiert.
Vorteil bei der Anlage eines eigenen Projektes, welches sämtliche Controls, die im SmartCenter verwendet werden beinhaltet, ist, dass hier Grundeinstellungen,

[89] Abbildung entnommen und verändert aus: Lexen (2007:S.90)

welche für alle Instanzen des Controls innerhalb des SmartCenters gelten sollen, durchgeführt werden können, ohne dies für jedes Control einzeln bewerkstelligen zu müssen.

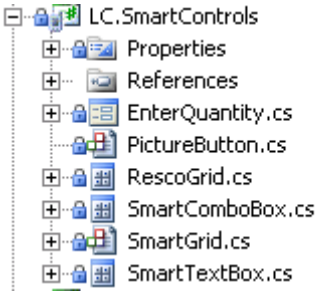


Abbildung 31: SmartControls

SmartComboBox:

Abbildung 32: SmartComboBox

Die Combobox bietet Werte zur freien Auswahl über ein Listfeld an. Dieses Control dient zur Anzeige und Auswahl von Vorgabewerten.

PictureButton:
PictureButtons werden benötigt, um Images als Buttons verwenden zu können. Derartige PictureButtons verbessern die Usability der SmartCenter – Anwendung erheblich.

SmartTextBox:
Die SmartTextBox ist ein Control zur Eingabe von beliebigen Zeichenketten. Da das Compact Framework keine eigenen Controls zur Eingabe von numerischen Werten anbietet, muss die Funktionalität manuell erweitert werden. Die SmartText-Property „*FieldFormat*“ kann nun von *string* auf *numeric* geändert werden und löst somit eine Überprüfung des SmartTextBox-Inhalts auf numerische Werte aus.

SmartGrid:
Da das Standardcontrol eines Grids nur äußerst eingeschränkte Möglichkeiten bietet, die Darstellung des Grids zu beeinflussen, muss nach einer anderen Lösung gesucht werden. In diesem Fall sollen keine eigenen *dll's* zur Erweiterung des vorhandenen Standardtools entwickelt werden. Stattdessen werden Grid-*dll's* der Firma Resco[90] zugekauft, welche bereits umfangreiche Funktionen zur Gestaltung des Tabellenlayouts zur Verfügung stellen. (siehe Abbildung 33: Gridgestaltung) Diese angekauften Programmbibliotheken werden in das Projekt eingebunden.
Das SmartGrid wird vollständig vom zugekauften RescoGrid abgeleitet und beinhaltet somit sämtliche Funktionen dieser Klasse.
Da es sich bei den Grid-*dll's* der Firma Resco ebenfalls um abgeleitete Klassen des .NET Compact Frameworks handelt, ist das Framework die grundlegende Vorraussetzung bei der Verwendung dieser Klassen.

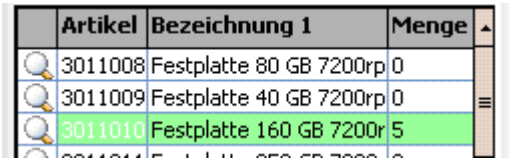


Abbildung 33: Gridgestaltung

6.10. Installation

Die Installation des SmartCenters setzt diverse Vorbereitungsarbeiten voraus:

- **TaskCenter-Server**: Aufgrund der Tatsache, dass diese Softwareapplikation ein Zusatzmodul des TaskCenters ist, wird die Installation der Stammsoftware vorausgesetzt. Hierzu ist es notwendig, die Server-Komponenten des TaskCenters zu installieren, um den Zugriff auf die Datenbank und den Webservices zu gewährleisten.
- **ActiveSync**: Des Weiteren ist die Installation der Synchronisationssoftware ActiveSync[91], ab der Version 3.8, auf einer Workstation notwendig.
- **.NET Compact Framework**: Ist das Endgerät erfolgreich über ActiveSync mit der Workstation verbunden, kann das .NET Compact

[90] Resco.net Developer; http://www.resco.net/developer/default.aspx
[91] Microsoft Active Sync ist derzeit in der Version 4.5 verfügbar und ist die Übertragungssoftware von Microsoft für Geräte mit Windows Mobile.

Framework 2.0 auf das Gerät eingespielt werden. ActiveSync und das .NET Compact Framework werden von Microsoft gratis zum Download angeboten.

Sind diese drei Vorbereitungen getroffen, kann das SmartCenter durch Ausführen der Installationsdatei auf dem mobilen Endgerät installiert werden.
Die Installationsdatei für das SmartCenter wird von POLLEX-LC Software GmbH über einen Download-link für ihre Lizenznehmer zur Verfügung gestellt.

6.11. Test

Der Prototyp wird seitens des Programmierers auf verschiedene Kriterien getestet:

- **Numerische Felder**: Bei Eingabe von Strings in numerische Felder muss die Eingabe annulliert werden.
- **Alphanumerische Eingabe**: Bei Eingabe zu langer Zeichenketten, welche nicht mehr dem Datentyp in der Datenbank genügen, muss die Eingabe annulliert werden.
- **Unvollständige Datensätze**: Überprüfung des Verhaltens bei Eingabe von lückenhaften Daten. Die Applikation darf unvollständige Daten nicht in die Datenbank speichern, darf ihrerseits keinen Applikations- oder Systemabsturz verursachen und muss den Benutzer auf die Falscheingabe aufmerksam machen.
- **Unvollständige Datenbanksätze**: Testen des Verhaltens des SmartCenters bei fehlerhaften Einträgen in der Datenbank. Software- und Systemabstürze sind zu vermeiden. Gefahr besteht hier z.B. bei Division durch Null.
- **Verbindungsabbruch**: Die Applikation darf keinerlei Systemabstürze verursachen bei plötzlichem Verbindungsabbruch der eigenen Internetconnectivity.
- **Servercrash:** Testen des Verhaltens der Software bei nicht Erreichbarkeit des Webservers, beziehungsweise des Datenbankservers.
- **User Interface:** Das gesamte SmartCenter wird in seiner Funktionalität und Performance mit Hilfe von Testdaten getestet. Als Tester kommen dabei alle Personen außer dem Entwickler in Frage. Dies soll zeigen ob das Userinterface intuitiv zu bedienen ist und ob Fehlbedienung weitgehend vorgebeugt wird.

7. Aussicht

Die Entwicklung des SmartCenters hat dem Verfasser dieser Arbeit nicht nur Einblicke in die Abläufe einer professionellen Softwareentwicklung gegeben, sondern auch die Möglichkeit geboten, sich mit einem Thema zu befassen, das sich in den letzten Jahren sehr stark weiterentwickelt hat und auch in der Zukunft weiterentwickeln wird.

Das SmartCenter befindet sich derzeit mit dem Prozess der Auftragserfassung in Echtbetrieb bei einer Kundenfirma von POLLEX-LC Software GmbH. Durch den Schritt der „Mobilmachung“ des TaskCenters erhofft sich die Geschäftsleitung von POLLEX-LC die Stärkung der eigenen Marktposition und Erfahrungen in einem neuen Sektor der ERP-Softwarebranche.
Das SmartCenter wurde von POLLEX-LC Software GmbH bereits in die Produktpalette aufgenommen und steht Kunden ab sofort zur Verfügung.

7.1. Weiterentwicklungen

Da das SmartCenter als offenes Projekt konzipiert wurde, sind Weiterentwicklungen in technischer Hinsicht und Erweiterungen durch neue Vorgänge, möglich. Mögliche Punkte, die in Zukunft für das SmartCenter in Frage kommen könnten:

Anoto:
Eine interessante Möglichkeit der mobilen Datenerfassung wurde erstmals 1996 von der Firma Anoto entwickelt und seitdem ständig verbessert. Ein „digitalisiertes“ Papier ist die Grundlage für diese Technologie. Dabei werden, für das menschliche Auge zu kleine, Punkte in speziellen Mustern auf das Papier gedruckt. Ein Laser, in Form eines einfachen Stiftes, kann diese Punkte auslesen. Somit ist eine exakte Positionsbestimmung des Stiftes auf dem Papier errechenbar, und über die Zeit betrachtet, sind Schriftzüge zu erfassen. Die erfassten Daten können nun per USB in einen PC eingespeist oder über eine Bluetoothverbindung versendet werden. Die Verwendung von Bluetooth

ermöglicht zum Beispiel das Übertragen der Daten mit Hilfe gängiger Datenübertragungsprotokolle von Handynetzen.[92]

RFID:
Das Prinzip von RFID (Radio Frequency Identification) ist einfach: Ein Transponder wird von einem Lesegerät ohne direkten Kontakt über Funk registriert. Dieser Transponder kann zum Beispiel auf einer Ware angebracht werden, wobei die Transpondernummer in der Datenbank mit der Ware in Verbindung gebracht werden muss. Das heisst es muss die Transpondernummer zum Artikel hinzugepflegt werden. Wird eine Ware, oder besser gesagt ihr Transponder, von einem Lesegerät erfasst, so registriert das System dies und kann auf diesem Weg feststellen, wo sich die Ware aktuell befindet. Auf diesem Weg können zum Beipiel Umlagerungen oder Warenausgänge automatisiert abgehandelt werden.[93]

Lagerumbuchungen:
Ein weiterer Vorgang für die Lagerwirtschaft ist bereits in Diskussion – die Lagerumbuchung. Derzeit sind lediglich Wareneingang und Warenausgang mit dem SmartCenter abgedeckt, jedoch ist es nahe liegend, für Umlagerungen ebenfalls einen Prozessablauf zu implementieren.

Verbunden mit den Vorgängen der Lagerumbuchungen und Warenaus- bzw. –eingängen wäre das Empfangen von Pickaufträgen auf den mobilen Endgeräten vorstellbar. Dies bedeutet, ein Lagerarbeiter bekommt den Auftrag, einen Artikel von Lagerort A nach Lagerort B zu transportieren direkt auf sein SmartCenter gesendet. Diese Möglichkeit ist aber erst ab einer gewissen Größe des Lagers und damit des gesamten Unternehmens sinnvoll.

Für den Anwendungsbereich von mobilen Geräten im ERP-Bereich gibt es eine Vielzahl von neuen Technologien und Möglichkeiten, welche immer umfangreicher und komplexer werden. Aus diesem Grund ist das SmartCenter für die Zukunft wohl als Basis für immer weitere Entwicklungen und neue Vorgänge zu sehen.

[92] Vgl. Anoto
[93] Vgl. RFID

8. Abbildungs- und Tabellenverzeichnis

9. Literatur

Für sämtliche Internetadressen wurde der Inhalt zuletzt am 3. Dezember 2008 überprüft.

ActiveBarcode - Barcodetypen
http://www.activebarcode.de/codes/

ActiveBarcode - Prüfziffernberechnung: Module43
http://www.activebarcode.de/codes/checkdigit/modulo43.html

AISCI Ident GmbH
Warenkommissionierung
http://www.aisci.de/pages/beratung/anwendungen/kommissionierung.html

Anoto Group AB
http://www.anoto.com/

Baghi, Ehsan:
Webservices in .NET und Webservice Enhancements,
Technische Universität Darmstadt

Becker, Jörg/Kugeler, Martin/Rosemann, Michael:
Prozessmanagement, Münster, 2000

Capto IT-Solutions GmbH
Shipping.net
http://www.capto.at/?story=39

Das Elektro-Kompendium; Autor unbekannt:
HASDPA – High Speed Downlink Packet Access
http://www.elektronik-kompendium.de/sites/kom/0910251.htm

Elektronic-Magazin.de, Verfasser: Benutzer StevenRun:
IP Schutzarten, 2007
http://www.elektronik-magazin.de/page/ip-schutzklassen-25

e-teaching.org:
Mobile Computing
http://www.e-teaching.org/technik/vernetzung/mobile_computing/

Federrath, Hannes:
Mobile Computing, 2001
http://www.semper.org/sirene/publ/Fede22001MobileComputing.html

Flexicom – Das Systemhaus für Identifikationstechnik:
CodeSnap
http://www.flexicom.de/Software/codesnap/codesnapoverview.htm

Forschungsgruppe für Industrial Software (INSO) TU Wien:
Reader zur LVA Usability Engineering, 2005

Hammerschall, Ulrike:
Verteilte Systeme und Anwendungen, München, 2005

Informationen zur Erstellung und Abgabe von Hochschulschriften
Universität Wien, 2008

Kappl Gerti/Kramler Gerhard:
Web-Services
http://www.inf.fu-berlin.de/inst/ag-se/teaching/V-J2EE-2003/48_Webservices.pdf

Lexen, Andreas:
Diplomarbeit – Migration eines Legacy ERP-Systems in.NET Architektur,Wien, 2007

Logistics to go:
Aufbau einer Lesefeldkurve eines Barcodescanners
http://www.logitogo.com/html/leseabstand_und_kontrast_fur_s.html

Logistic to go:
Software für Mobile Datenerfassung und Geräteauswahl
http://www.logitogo.com/html/mobile_datenerfassung_software.html

MSDN Library; Autor: Mauerer, Jürgen:
Web Services mit .NET Framework 2.0 und Visual Studio 2005
http://msdn.microsoft.com/de-de/library/bb979113.aspx

MSDN Library:
Konzeptionelle Übersicht über .NET Framework, 2008
http://msdn.microsoft.com/de-de/library/zw4w595w.aspx

MSDN Library:
Smart Device Extensions, 2008
http://msdn.microsoft.com/de-de/library/ms837916.aspx

MSDN Library:
Visual SourceSafe, 2008
http://msdn.microsoft.com/de-de/library/3h0544kx(VS.80).aspx

MSDN Library:
.NET Compact Framework – Architektur, 2008
http://msdn.microsoft.com/de-de/library/9s7k7ce5.aspx

Norris, Grant/Hurley, R. James/Hartley, M. Kenneth/Dunleavy, R. John/
Balls, D. John:
E-Business und ERP, Weinheim, 2002

Omega Engineering Inc.
International Standards
http://www.omega.com/temperature/Z/pdf/z194-196.pdf

Papazoglou, Michael:
Web services: Principles and Technology, Harlow, 2008

POLLEX-LC Software GmbH
www.pollex.at

Pötscher, Florian:
Diplomarbeit – .NET Web Services und Grid Computing, Wien, 2007

RFID Journal
RFID http://www.rfid-journal.de/

Sattler, Roland / Schmied, Stefan
Prototyping – Konzepte und Techniken
http://cartoon.iguw.tuwien.ac.at:16080/fit/fit01/prototyping/konzepte.html

Schatten, Alexander, Wien, 2006
Vorlesungsunterlage Web-Services
http://www.schatten.info/lehre/veranstaltungen/
06w_e-commerce-technische-aspekte/web-services.pdf

Welt online; Autor unbekannt:
UMTS, EDGE, HSUPA oder GPRS einfach erklärt
http://www.welt.de/wirtschaft/article1730594/
UMTS_EDGE_HSUPA_oder_GPRS_einfach_erklaert.html

W3Schools - WSDL-Tutorial
http://www.w3schools.com/WSDL/wsdl_documents.asp

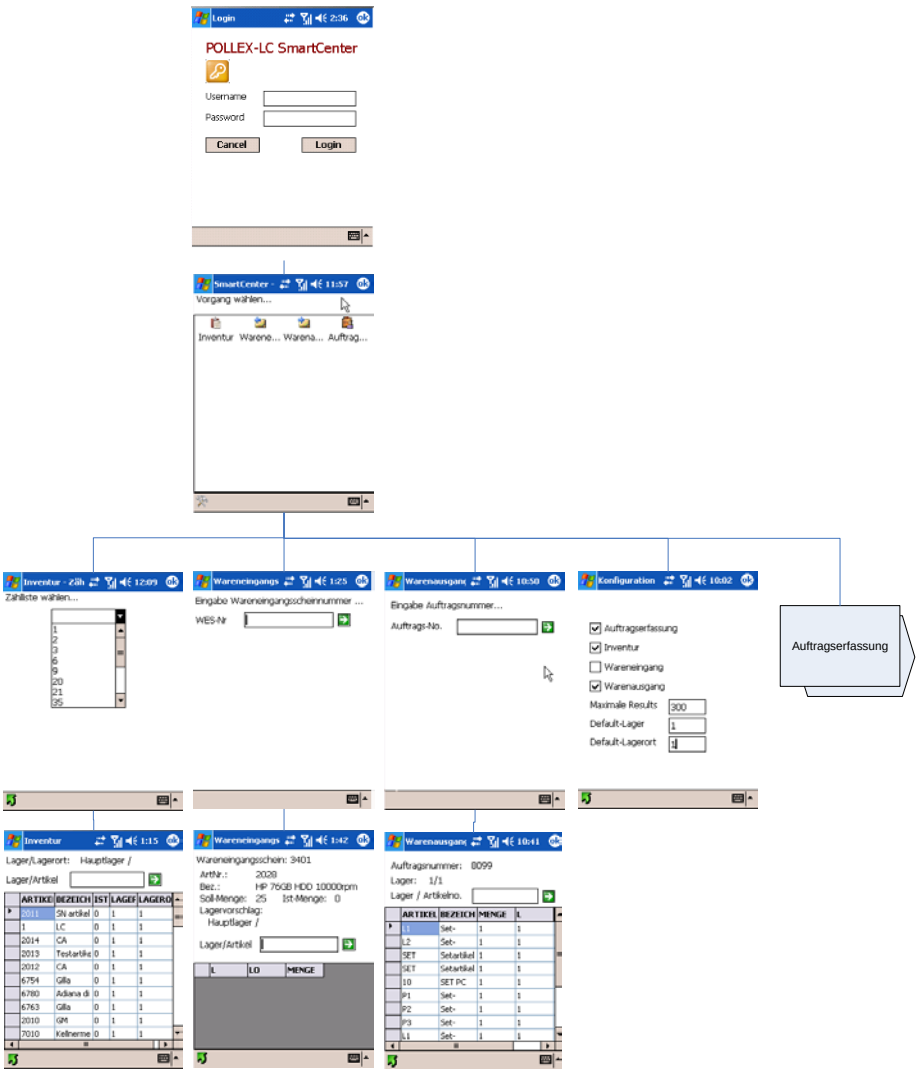
Yao, Paul / Durant, David:
.NET Compact Framework Programming with C#, Boston, 2004

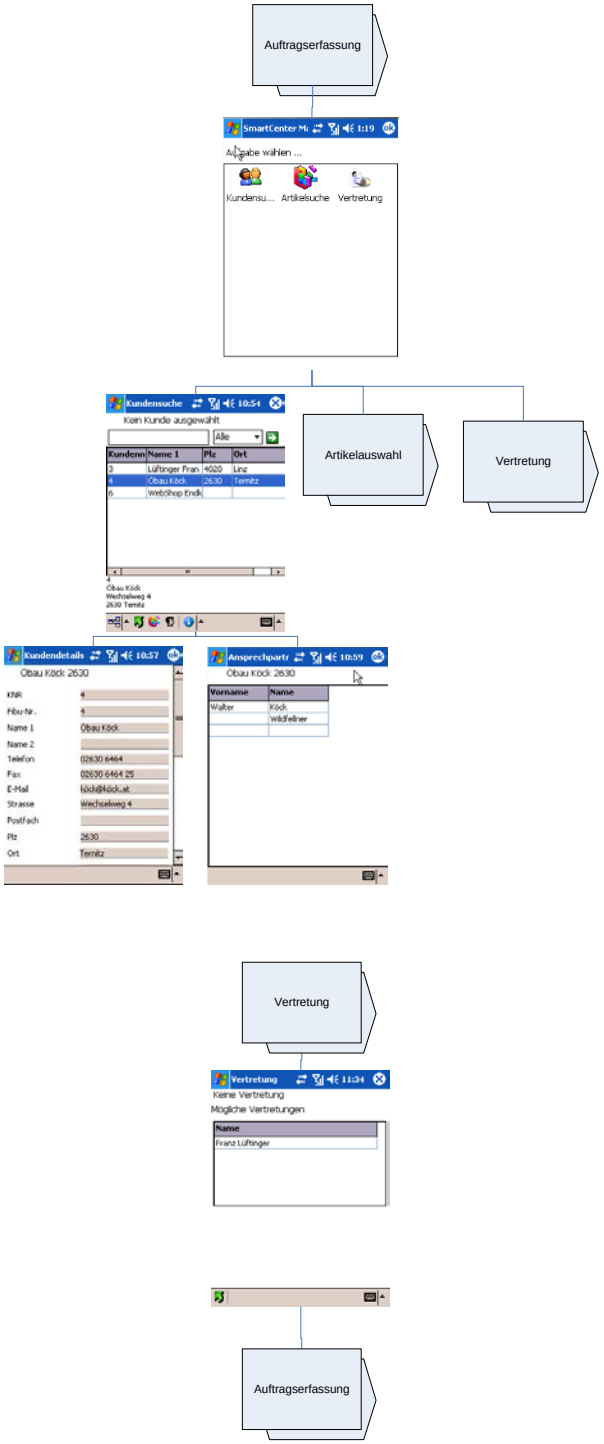
Zuser, Wolfgang/Biffl, Stefan/Grechenig, Thomas:
Software Engineering, 2001

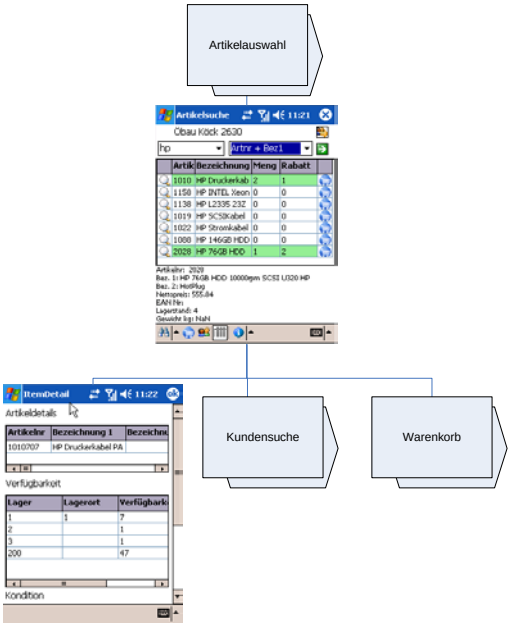
„Ich habe mich bemüht, sämtliche Inhaber der Bildrechte ausfindig zu machen und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Meldung bei mir.“[94]

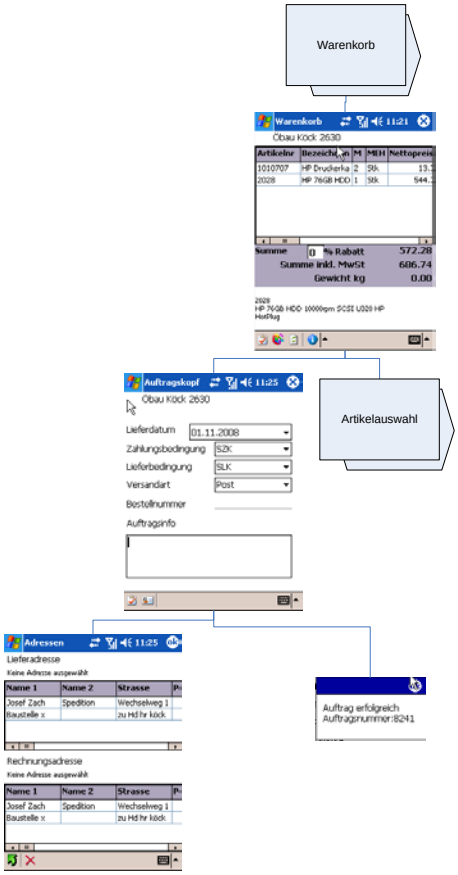
[94] Informationen zur Erstellung und Abgabe von Hochschulschriften, 2008, S. 1

Anhang A: User Interface




Auftragserfassung
Artikelauswahl
Vertretung
Vertretung
Auftragserfassung


Artikelauswahl
Artikelsuche
Obau Köck 2630
hp
Artnr + Bez1
Artik
Bezeichnung
Meng
Rabatt
1010 HP Druckerkab 2 1
1158 HP INTEL Xeon 0 0
1138 HP L2395 23Z 0 0
1019 HP SCSIKabel 0 0
1022 HP Stromkabel 0 0
1008 HP 146GB HDD 0 0
2028 HP 76GB HDD 1 2
ItemDetail
Artikeldetails
Artikelnr
Bezeichnung 1
1010707
HP Druckerkabel PA
Verfügbarkeit
Lager
Lagerort
1
1
7
2
1
3
1
200
47
Kondition
Kundensuche
Warenkorb


Warenkorb
Warenkorb 11:21
Öbau Köck 2630
Artikelnr
Bezeichn
M
ME
Nettopreis
1010707
HP Druckerka
2
Stk
2028
HP 76GB HDD
1
Stk
544.2
Summe
0
% Rabatt
572.28
Summe inkl. MwSt
686.74
Gewicht kg
0.00
Artikelauswahl
Auftragskopf 11:25
Öbau Köck 2630
Lieferdatum
01.11.2008
Zahlungsbedingung
SZK
Lieferbedingung
SLK
Versandart
Post
Bestellnummer
Auftragsinfo
Adressen 11:25
Lieferadresse
Keine Adresse ausgewählt
Name 1
Name 2
Strasse
Josef Zach
Spedition
Wechselweg 1
Baustelle x
zu Hd hr köck
Rechnungsadresse
Keine Adresse ausgewählt
Name 1
Name 2
Strasse
Josef Zach
Spedition
Wechselweg 1
Baustelle x
zu Hd hr köck
Auftrag erfolgreich
Auftragsnummer:8241

Anhang B: Kurzfassung

Im Zuge dieser Diplomarbeit wird ein Softwaremodul für die ERP-Software POLLEX-LC TaskCenter der Firma POLLEX-LC Software GmbH entwickelt. Dieses Modul ist für den Einsatz auf mobilen Endgeräten gedacht und soll die Ablaufprozesse der Vorgänge

- Inventur,
- Wareneingangskommissionierung,
- Warenausgangskommissionierung und
- Auftragserfassung

unterstützen und beschleunigen.

Zur Identifikation der benötigten Funktionalitäten der Software, wird eine Analyse der Stammsoftware, der verfügbaren Geräte am Markt und der derzeitig angebotenen Softwarelösungen durchgeführt. Anhand dieser Erkenntnisse kann schließlich ein Prototyp entworfen werden, welcher den geforderten Funktionen gerecht wird.

abstract

The goal of this thesis is to design a new software module for the ERP-software POLLEX-LC TaskCenter of the publisher POLLEX-LC Software GmbH.
This module should be operating on mobile devices and should provide the working processes of

- inventory,
- goods issuing department,
- incoming goods department and
- mobile order entry

to make them easier and faster.

The analysis of the current hardware and software offerings provides also the identification of the needed functionalities of this software, like a good look at the existing interfaces of POLLEX-LC TaskCenter. Due to this detected functionalities it is possible to design a prototype, which includes all functions for the wanted processes.

Anhang C: Lebenslauf

Angaben zur Person:

Name:	Stefan Schabel
Adresse:	Krieglergasse 15/19 1030 Wien
Email:	stefan@schabel.net
Telefon:	0650/3341853
Geboren am	14.05.1979 in Wien
Staatsbürgerschaft:	Österreich
Familienstand:	ledig

Schulausbildung:

09-1985 bis 06-1989	Volksschule Weikendorf
09-1989 bis 05-1993	Unterstufe Bundesgymnasium und Bundesrealgymnasium Gänserndorf
09-1993 bis 05-1998	HTL/TGM Maschinenbau Ausbildungszweig allgemeiner Maschinenbau; Abschluss mit Matura;

Präsenzdienst:

10-2001 bis 09-2002	Zivildienst als Sanitäter beim niederösterreichischen Roten Kreuz in Gänserndorf

Studium:

10-1998 bis 10-2000	Diplomstudium Informatik an der techn. Universität Wien
10-2000 bis 06-2006	Interuniversitäres Bakkalaureatsstudium Wirtschaftsinformatik an der techn.Universtität und an der Universität Wien mit Kernfachkombination *Decision Support in E-Government* am Institut für Raumentwicklung, Infrastruktur- und

	Umweltplanung im Fachbereich Finanzwissenschaft und Infrastrukturpolitik an der TU Wien. Abschluss am 14.06.2006
Bakkalaureatsarbeiten	-) Erarbeitung eines Konzepts zur Reorganisation von Prozessen -) Straßenlärmschutzmaßnahmen der europäischen Union
Seit 06-2006	Interuniversitäres Magisterstudium Wirtschaftsinformatik an der techn. Universität und der Universität Wien mit Kernfachkombination *Organisationsplanung* am Institut für Managementwissenschaften an der TU Wien.
Diplomarbeit	ERP – Mobile Computing Anbindung eines mobilen Clients an ein bestehendes ERP – System über Webservices

Berufspraktika:

07-1994	Praktikum bei OMV Gänserndorf, Abteilung SOB
08-1995	Praktikum bei OMV Gänserndorf, Abteilung SOB
07-1996	Praktikum bei Schindler, Aufzüge und Rolltreppen; Werkstättentätigkeiten
07-1997 bis 08-1997	Praktikum bei Schindler, Aufzüge und Rolltreppen; Werkstättentätigkeiten
07-1998	Praktikum bei Novoferm, AutoCad – Zeichner
04-1999 bis 08-1999	Netzwerkadministrative Tätigkeiten für diverse Handelsakademien im Raum Wien durchgeführt;
07-2000	Praktikum bei Siemens AG, Softwareentwicklung, Projektorganisation und Durchführung
07-2001	Praktikum bei Siemens AG, Windows NT - Administration
07-2004	Praktikum bei Fa. Wave – Solutions for IT, im Bereich Project Office
07-2005	Praktikum bei Fa. Wave – Solutions for IT, im Bereich Prozessmanagement
Seit 12-2005	Softwareentwickler bei Fa. Pollex-LC

Studienbegleitende Berufserfahrung:

04-1998 bis 08-1999	-) Logistikmitarbeiter bei der Flughafen Wien AG -) Reparatur- und Wartungsarbeiten an Computern -) Zusammenbau einzelner Systeme und Konfiguration

	des Betriebssystems
	-) Zusammenbau von Netzwerken
02-2003 bis 08-2004	-) Fa. VIAS, Vienna International Airport Security
09-2004 bis 10-2005	-) Fa. Contact-Promotion

Persönliche Kentnisse

Sprachkenntnisse:	-) Englisch auf Maturaniveau in Wort und Schrift
	-) Spanisch, Anfängerkurs an der Wirtschaftsuniversität Wien
EDV Kenntnisse:	-) Programmiersprachen: SQL, HTML, XML, Pascal, Java, C#, PHP
	-) Kenntnisse von Datenbanken (Oracle, MySQL, Access, …)
	-) UML Kenntnisse
	-) vertraut mit dem Aufbau eines Computer am aktuellen Stand der Technik, sowie deren Wartung
	-) fundierte Microsoft Office Kenntnisse (Excel, Word, PowerPoint, Frontpage)
	-) Microsoft Navision
	-) vertraut im Umgang mit den aktuellen Betriebssystemen von Microsoft
	-) Netzwerkkenntnisse, sowohl Aufbau als auch Funktionsweise
	-) Kenntnisse mit UNIX/Linux
	-) vertraut mit Ablauf eines Softwareprojekts im Unified Process
	-) fundierte Kenntnisse im Bereich ERP-Software, im Speziellen Pollex-LC TaskCenter.
Zus. Ausbildung	-) Ausbildung zum Sanitäter
	-) Absolvierung eines Lehrgangs für IGM – Schweißrobotersysteme über Programmierung INS am RT 330
	-) Ausbildung zum Stapler- und Kranführer

Ort, Datum:
Wien, 05.01.2009