

Dokumentation

-Projekt-

ContactManager



-Projektmitglieder-

Erwin Braun, Alexander Steffen und Gerrit Böselager

-Datum-

23. März 2012

Inhaltsverzeichnis

1	Anforderungsdefinition.....	2
1.1	Funktionalitäten	2
1.1.1	Muss-Funktionalitäten.....	2
1.1.2	Optionale Funktionalitäten	3
1.1.3	Abgrenzungskriterien.....	3
2	Planung und Fachkonzeption	4
2.1	Benutzerrechte	4
2.2	Use-Case-Diagramm	5
2.3	Klassendiagramm	6
2.4	Entity-Relationship-Model (ERM)	7
2.5	Prototypen	8
2.6	Testplan	13
3	Implementierung.....	14
3.1	Team.....	14
3.2	Programmierstandards	14
3.3	Umgebung	15
3.3.1	Webserver	15
3.3.2	Datenbank Management System (DBMS)	15
3.3.3	Programmiersprachen	15
3.3.4	Entwicklungsumgebung.....	15
3.3.5	Versionsverwaltung	15
3.4	Architektur.....	16
3.4.1	Abweichung zum Fachkonzept.....	16
3.4.2	Drei-Schichten-Modell	16
3.4.3	Technisches Klassendiagramm.....	17
3.4.4	Package-Diagramm	18
3.5	Externe Komponenten, Module und Quellen	19
4	Testphase	21
5	Projektabschluss & Fazit	22
5.1	Erreichung der Anforderungen	22
5.2	Soll / Ist - Vergleich.....	23
5.3	Fazit	24

1 Anforderungsdefinition

Als Ziel gilt die Umsetzung einer Kontaktverwaltung auf Basis einer Web-Applikation, bei der die gesamte Oberfläche in einem herkömmlichen Browser (z.B. Microsoft Internet Explorer, Mozilla Firefox oder Google Chrome) dargestellt werden kann. Die Pflege von Kontakten (Anlage, Bearbeitung, Löschung) wird von Benutzern durchgeführt.

Der ContactManager ist beschränkt auf den Einsatz innerhalb eines Unternehmens. Die von Administratoren gepflegten Benutzer haben die Möglichkeit private Kontakte, sowie auch Firmenkontakte zu pflegen. Zusätzlich können Kontakte als „öffentlich“ gekennzeichnet werden, wodurch der Kontakt auch von allen anderen Kollegen einsehbar, veränderbar und löschar ist.

Die Funktionalitäten werden im Folgenden kurz konkret beschrieben.

1.1 Funktionalitäten

1.1.1 Muss-Funktionalitäten

Die Muss-Funktionalitäten dokumentieren die Funktionen, die explizit vom Auftraggeber verlangt werden und unbedingt umzusetzen sind.

- Benutzerlogin
 - o Benutzer können sich mit Loginnamen und Passwort anmelden und können somit Funktionen, die nur für angemeldete Benutzer zugänglich sind, nutzen.
- Erfassung, Änderung und Löschung der Daten (Name, Adresse, Telefon, Freitext)
 - o Nach dem Login kann der Nutzer Kontakte entsprechend verwalten.
- Suchen nach Kontakten (Volltextsuche)
 - o Kontakte können von Benutzern per Volltextsuche gefunden werden.
- Plausibilitätsprüfung bei der Eingabe
 - o Die angegebenen Kontaktdaten werden auf Plausibilität untersucht. Dazu gehören insbesondere E-Mail-Adresse und Geburtstag.
- Speicherung der Daten in einer Datenbank
 - o Die Daten werden in einer MySQL-Datenbank gespeichert.

1.1.2 Optionale Funktionalitäten

Zusätzlich zu den Muss-Funktionalitäten können die folgend gelisteten Features optional umgesetzt werden.

- Filter für die Suche
 - o Filtermöglichkeit auf einzelne Spalten (z.B. Vorname, Nachname, Ort) eines Kontaktes
- Administratorfunktionen
 - o Benutzerverwaltung (Anlegen, Bearbeiten, Löschen und Sperren von Benutzern)
- „Eigene Daten ändern“ für eingeloggte Benutzer
 - o Benutzer können eigenen Daten, wie z.B. Vorname, Nachname, Passwort oder E-Mail eigenständig ändern
- „Kennwort vergessen“-Funktion
 - o Benutzer können darüber beim Administrator ein neues Kennwort anfordern
- Druck von Einzelkontakten
 - o Kontakte können zur Ausgabe auf Papier ausgedruckt werden.
- Export der Kontaktdaten in eine CSV-Datei
- Möglichkeit zur Erstellung von Kontaktgruppen
 - o Selbst erstellte Kontakte können in Gruppen eingeteilt werden, die durch den Benutzer selbst zu verwalten sind.
- Auswertungsfunktionen
 - o Welche und wie viele Benutzer sind aktuell online?
 - o Welche Kontakte wurden zuletzt geändert?
 - o etc.
- Anzeige der Adresse per GoogleMaps

1.1.3 Abgrenzungskriterien

Folgende Abgrenzungskriterien wurden für das Projekt definiert.

- Schnittstellen zu externen Systemen (beispielsweise zu ERP oder CRM-Systemen) sind nicht vorgesehen.
- Ein Import von Kontaktdaten wird nicht unterstützt.
- Einmal gelöschte Kontakte können nicht wiederhergestellt werden, da diese direkt von der Datenbank entfernt werden.
- Die Datenübertragung zwischen Server und Client geschieht auf unverschlüsseltem Wege.
- Passwörter werden unverschlüsselt in der Datenbank abgespeichert.
- Mehrsprachigkeit ist nicht vorgesehen.

2 Planung und Fachkonzeption

2.1 Benutzerrechte

Im der folgenden Tabelle wird definiert, welche Funktionen von den entsprechenden Akteuren (angemeldeter Benutzer, nicht angemeldeter Benutzer und angemeldeter Administrator) genutzt werden dürfen.

	öffentliche Kontakte einsehen	öffentliche Kontakte bearbeiten und löschen	eigene private Kontakte einsehen	eigene private Kontakte bearbeiten und löschen	Neuen Benutzer anlegen, bearbeiten, löschen, sperrern
Nicht angemeldeter Benutzer	X				
Angemeldeter Benutzer	X	X	X	X	
angemeldeter Administrator	X	X	X	X	X

2.2 Use-Case-Diagramm

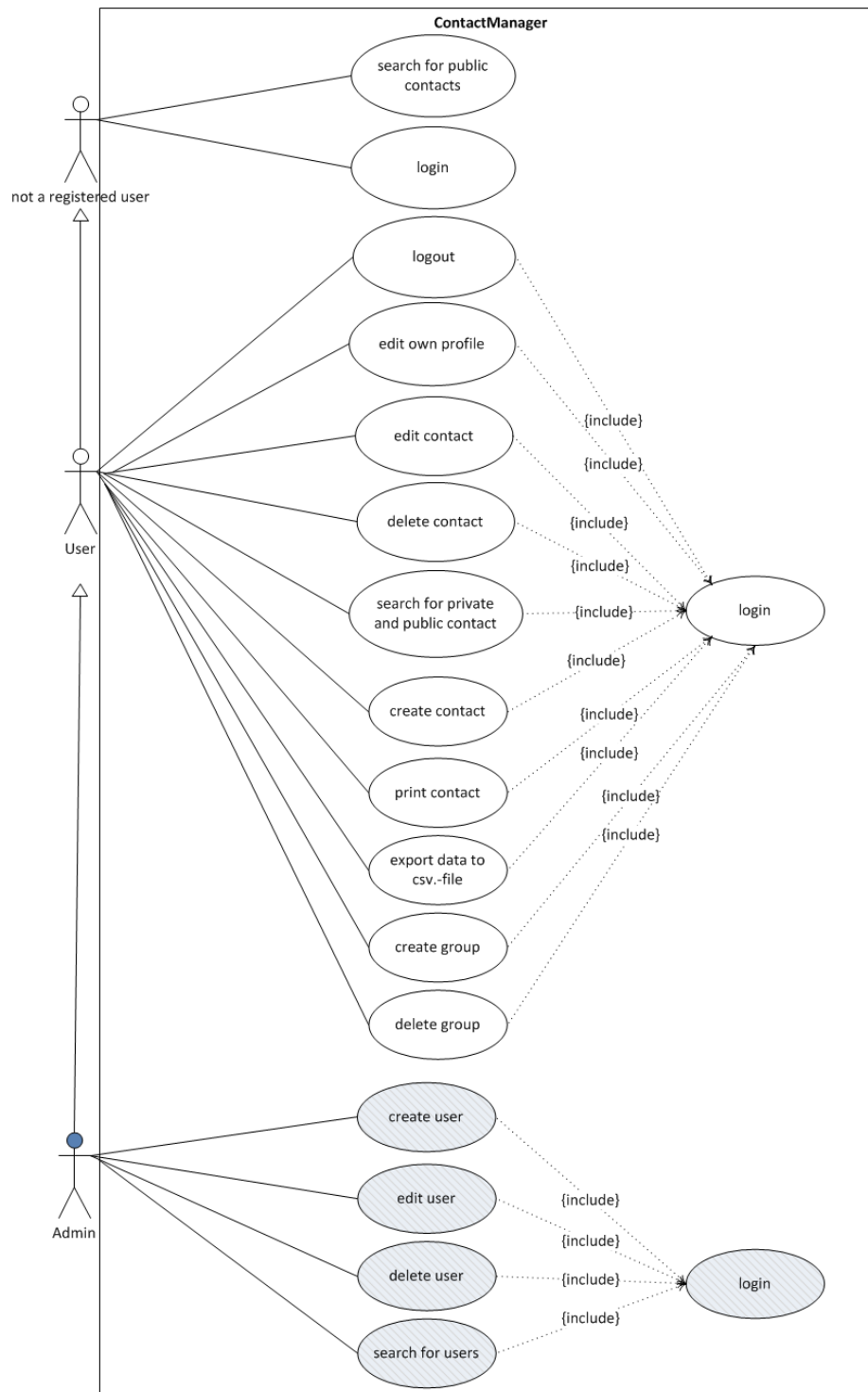


Abbildung 1: Use-Case-Diagramm

Die schraffierten Use-Cases und Akteure sind aus den optionalen Anforderungen entstanden und werden aus dem Grund auch nur dann umgesetzt, wenn entsprechende Kapazitäten im Projekt frei sind.

2.3 Klassendiagramm

Folgend ist das fachliche Klassendiagramm aufgeführt, das die fachlichen Zusammenhänge/Beziehungen zwischen den einzelnen Klassen darstellt. Das technische Klassendiagramm (siehe 3.4.3) zeigt wie dieses Konzept technisch umgesetzt wurde und entsprechende Differenzen.

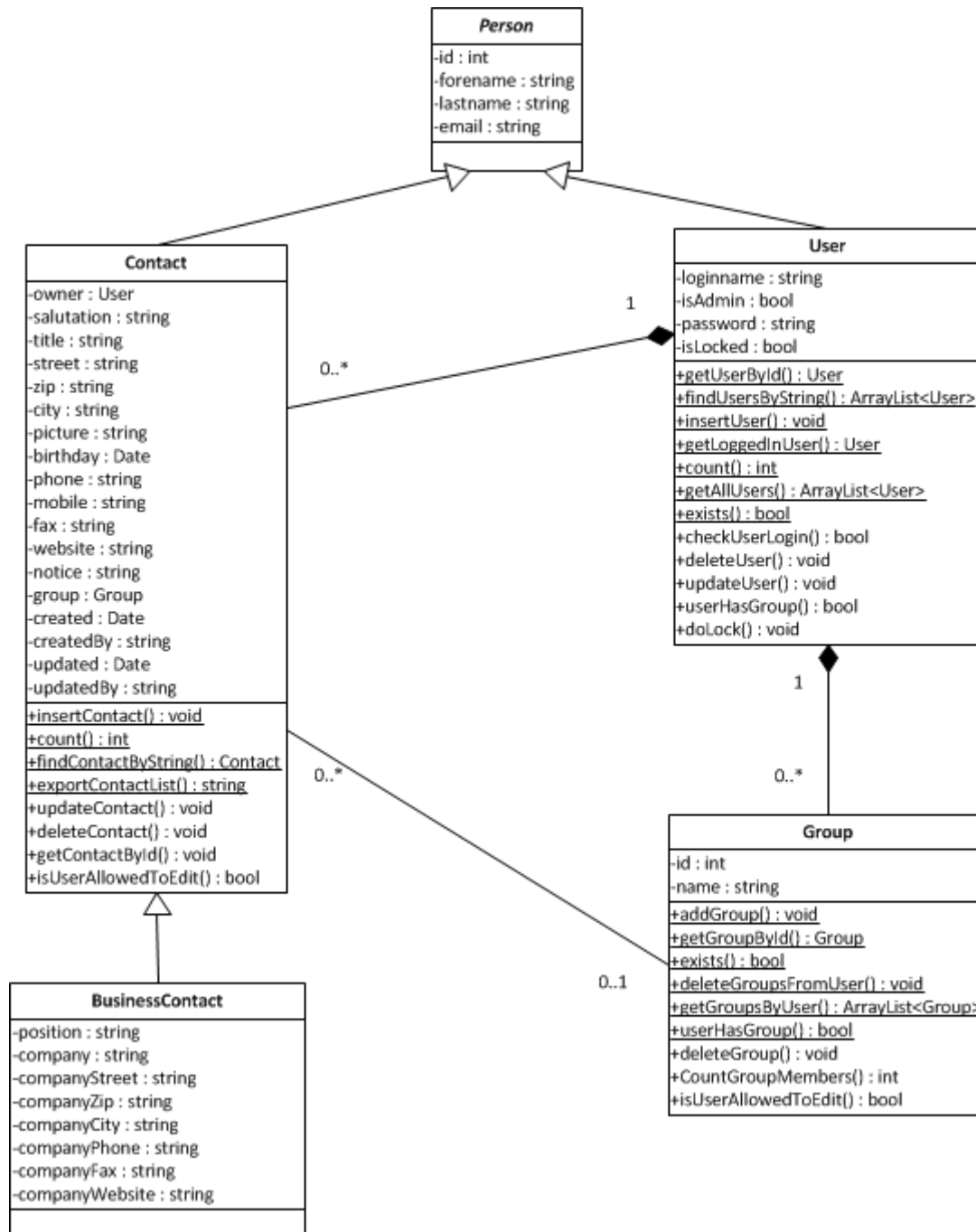


Abbildung 2: fachliches Klassendiagramm

2.4 Entity-Relationship-Model (ERM)

Das folgende Entity-Relationship-Model zeigt die geplante Struktur der Datenbank.

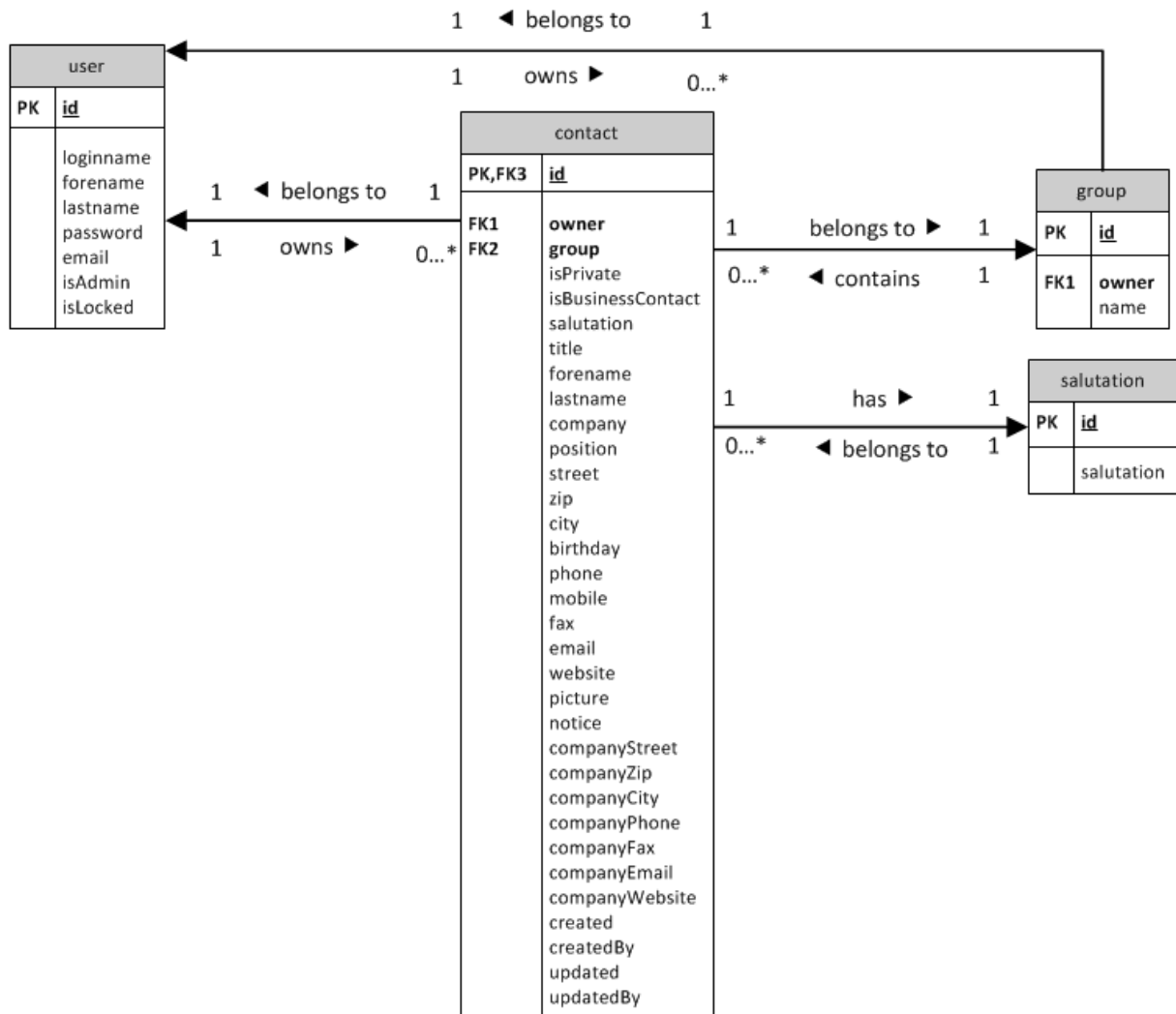


Abbildung 3: Entity-Relationship-Model

2.5 Prototypen

Grundaufbau



Abbildung 4: Prototyp des Grundaufbaus

Kontaktliste für eingeloggte Nutzer

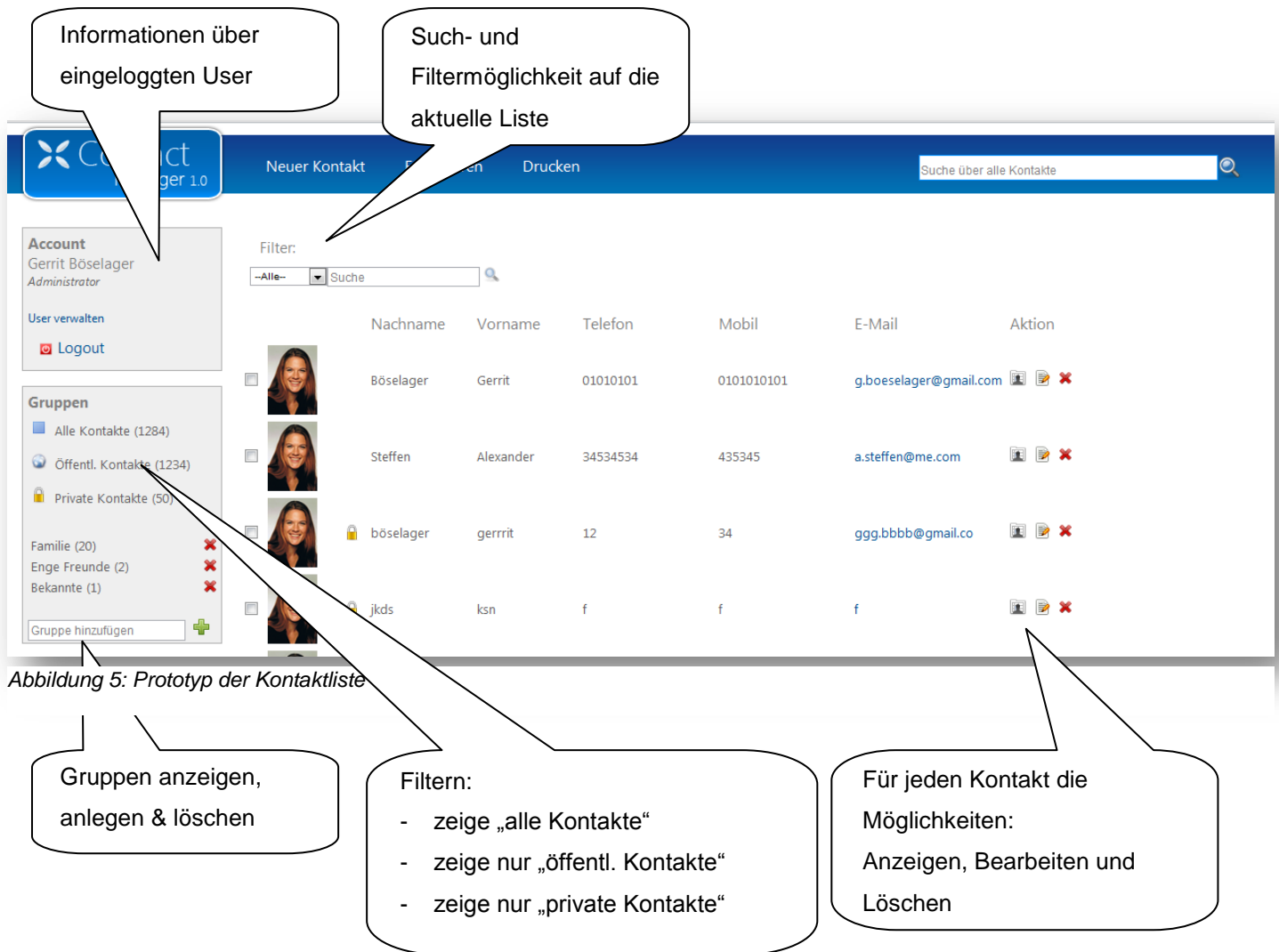



Abbildung 5: Prototyp der Kontaktliste

Kontakt anzeigen


 **Contact**
Manager 1.0


[Neuer Kontakt](#) [Exportieren](#) [Drucken](#)



Account
Gerrit Böselager
Administrator

[User verwalten](#)
[Logout](#)

Gruppen
[Alle Kontakte \(1284\)](#)
[Öffentl. Kontakte \(1234\)](#)
[Private Kontakte \(50\)](#)

Familie (20) ✖
Enge Freunde (2) ✖
Bekannte (1) ✖
 



Allgemein
 nicht öffentlicher Kontakt •  Freunde

Frau Prof. Dr.
Hildegard Schnaggendorf
Geburtstag: 13.01.1980

Adressdaten
Straße: Brunstweg 6
PLZ/Ort: 0815 Klitschendorf

Kontaktinformationen
Telefon: 01234 / 906090
Mobil: 0170 / 6665666
Fax: 01234 / 906091
E-Mail: hildi.schnagge@dorfmail.de
Webseite: www.wer-ist-hildegard.de

Sonstiges
Notiz: Hildegard ist bis 30.3.2012 im Urlaub....

Firmendaten
Firma: Currywurst AG
Position: Wurstprüferin (sausages manager)
Straße: Alte Currystr. 700
PLZ/Ort: 11233 Naturdarmstadt

Abbildung 6: Prototyp der Kontaktansicht

Kontakt bearbeiten/anlegen

[Neuer Kontakt](#)
[Exportieren](#)
[Drucken](#)

Suc

Account

Gerrit Böselager
Administrator

[User verwalten](#)

[Logout](#)

Gruppen

Alle Kontakte (1284)

Öffentl. Kontakte (1234)

Private Kontakte (50)

Familie (20) ✖

Enge Freunde (2) ✖

Bekannte (1) ✖

[Gruppe hinzufügen](#) +

Kontakt bearbeiten

Laden Sie ein Foto des Kontaktes hoch: [Datei auswählen](#) Keine ausgewählt

Zuordnung

Privater Kontakt ☒

Gruppe -Standard- ▾

Allgemein

Anrede: Herr ▾

Titel:

Vorname:

Nachname:

Geburtstag: (Format: TT.MM.JJJJ)

Adressdaten

Straße:

PLZ/Ort:

Kontaktinformationen

Telefon:

Mobil:

Fax:

E-Mail:

Webseite:

Sonstiges

Notiz:

Firmendaten

Firma:

Position:

Straße:

PLZ/Ort:

Speichern

Abbildung 7: Prototyp der Kontakt bearbeiten Ansicht

Kontakt drucken


	Allgemein nicht öffentlicher Kontakt • Freunde Frau Prof. Dr. Hildegard Schnaggendorf Geburtstag: 13.01.1980
	Adressdaten Straße: Brunstweg 6 PLZ/Ort: 0815 Klitschendorf
	Kontaktinformationen Telefon: 01234 / 906090 Mobil: 0170 / 6665666 Fax: 01234 / 906091 E-Mail: hildi.schnagge@dorfmail.de Webseite: www.wer-ist-hildegard.de
	Sonstiges Notiz: Hildegard ist bis 30.3.2012 im Urlaub....
	Firmendaten Firma: Currywurst AG Position: Wurstprüferin (sausages manager) Straße: Alte Currystr. 700 PLZ/Ort: 11233 Naturdarmstadt

Abbildung 8: Prototyp der Druckansicht

2.6 Testplan

In der folgenden Abbildung sind die geplanten Testfälle aufgelistet. Die detaillierte Testphase, inklusive entsprechendem Protokoll, kann aus Kapitel 4 entnommen werden.

Benutzersicht	TestNr.	Testfall
Benutzer	T001	Login
	T002	Logout
	T003	Kontakt anlegen
	T004	Kontakt bearbeiten
	T005	Kontakt Löschen
	T006	Kontakt details öffnen
	T007	Suche nach öffentlichen Kontakten
	T008	Suche nach privaten Kontakten
	T009	Daten in CSV Exportieren
	T010	Kontakte Drucken
	T011	Kontakt Gruppen erstellen
	T012	Kontakt Gruppen Löschen
nicht registrierter Benutzer	T013	Suche nach öffentlichen Kontakten
	T014	Login
Administrator	T016	Benutzer anlegen
	T017	Benutzer bearbeiten
	T018	Benutzer Löschen
	T019	Benutzer sperren
	T020	Benutzer suchen

Abbildung 9: Geplante Testfälle

3 Implementierung

3.1 Team

Die Entwicklung wird in einem Drei-Mann-Team von den Entwicklern

- Braun, Erwin
- Steffen, Alexander
- Böselager, Gerrit

durchgeführt.

3.2 Programmierstandards

Bei der *Benennung von Variablen, Eigenschaften, Methoden, Objekten, Klassen und sonstigen Konstruktionen*, ist darauf zu achten, dass *sinnvolle Bezeichnungen* gewählt werden, um so eine bessere Verständlichkeit des Programmcodes zu gewährleisten.

Die *Trennung von Wörtern* innerhalb eines Bezeichners wird durch *Großschreibung des ersten Buchstabens eines neuen Wortes* gekennzeichnet, wobei der *Anfangsbuchstabe des Bezeichners* immer *klein* zu schreiben ist. Als Sprache soll *Englisch* verwendet werden.

Ausnahmen:

Handelt es sich um einen *Klassenname*, so wird der Anfangsbuchstabe *großgeschrieben*.

Konstantenbezeichnungen werden in **GROßBUCHSTABEN** geschrieben und einzelne Wörter durch einen Unterstrich (**_**) voneinander getrennt.

Beispiele:

```
String myPerfectString;  
public void getPerfectString() { ... };  
public class User { ... }  
public static final int CONSTANT_VALUE = 815;
```

Kommentare sollen möglichst häufig an sinnvollen Stellen eingesetzt werden, wobei diese in Form von „*JavaDoc*“ und in *englischer Sprache* anzufertigen sind.

Konventionen für Dateinamen

JSP-Dateien werden kleingeschrieben. Der Anfangsbuchstabe einer JAVA-Datei wird großgeschrieben und hat den gleichen Namen, wie die darin enthaltene Klasse.

3.3 Umgebung

3.3.1 Webserver

Als Webserver-Software kommt Apache Tomcat 7.0 (Release 7.0.23) zum Einsatz. Tomcat ist kostenlos und ein open source Projekt und ist dazu fähig *Java Servlets* sowie *JavaServerPages* zu verarbeiten. Zusätzlich wird dazu mindestens die JRE 6.0+ (Java Standard Edition Runtime Environment) benötigt.

Quelle: <http://tomcat.apache.org>

Download: <http://tomcat.apache.org/download-70.cgi>

3.3.2 Datenbank Management System (DBMS)

Eingesetzt als DBMS (Datenbank Management System) wird MySQL (Version: 5.0.51b). MySQL ist eine der am häufigsten anzutreffenden Datenbank Web-Bereich und ist kostenfrei zu nutzen.

Quelle & Download: <http://dev.mysql.com/downloads/mysql>

3.3.3 Programmiersprachen

Die hauptsächlich zu verwendende Programmiersprache im Bereich von Java Servlets und JavaServerPages ist *JAVA*. Zur Darstellung der Webseiten wird HTML (Hypertext Markup Language) und zum clientseitigen Scripting JavaScript (Achtung: JavaScript ≠ JAVA).

3.3.4 Entwicklungsumgebung

Als Entwicklungsumgebung kommt „Eclipse Java EE for Web Developers“ in den Versionen „HELIOS“ und „INDIGO“ zum Einsatz.

3.3.5 Versionsverwaltung

Um zu ermöglichen, dass die Teammitglieder zeitgleich am Projekt (bzw. auch an gleichen Dateien) arbeiten können und alle Änderungen an den Projektdateien protokolliert und versioniert werden, wird als Versionsverwaltung Git eingesetzt.

Das Projekt-Repository liegt dabei in der „Cloud“ beim Hoster www.github.com. Der Dienst ist kostenlos und alle Projektmitglieder haben ständig Zugriff auf die gemeinsamen Projektdateien.

Als Client-Tool auf den Windows-Rechnern kommt TortoiseGit (Version 1.7.6.0) zum Einsatz.

Quelle & Download: <http://code.google.com/p/tortoisegit/downloads/list>

3.4 Architektur

3.4.1 Abweichung zum Fachkonzept

Die im Entity-Relationship-Model gebildete Entität „salutation“ (Anrede) wird im Echtbetrieb nicht implementiert. Grund dafür ist, dass der Aufwand größer als der Nutzen ist, wenn die Anreden (Frau, Herr) in eine eigene Tabelle ausgelagert werden. Die Anreden sind daher fest als Text in die Tabelle „contact“ zu schreiben.

3.4.2 Drei-Schichten-Modell

Die Architektur wird nach dem Drei-Schichten-Modell aufgebaut, das der Strukturierung der Software dient. Hierbei wird die Software in die Datenhaltungs-, Fachkonzept- und die Darstellungsschicht aufgeteilt.

In der Präsentationsschicht werden alle Oberflächenkomponenten, die der Interaktion mit dem Benutzer dienen, untergebracht. In der Fachkonzeptschicht befinden sich zentrale Logiken (bzw. Geschäftslogiken) die die Software steuern. Die Datenhaltungsschicht besteht, wie hier, meist aus einer Datenbank und ist für die Datenspeicherung und das Laden von Daten zuständig.¹

Folgend ist die in diesem Projekt gewählte Architektur aufgezeigt.

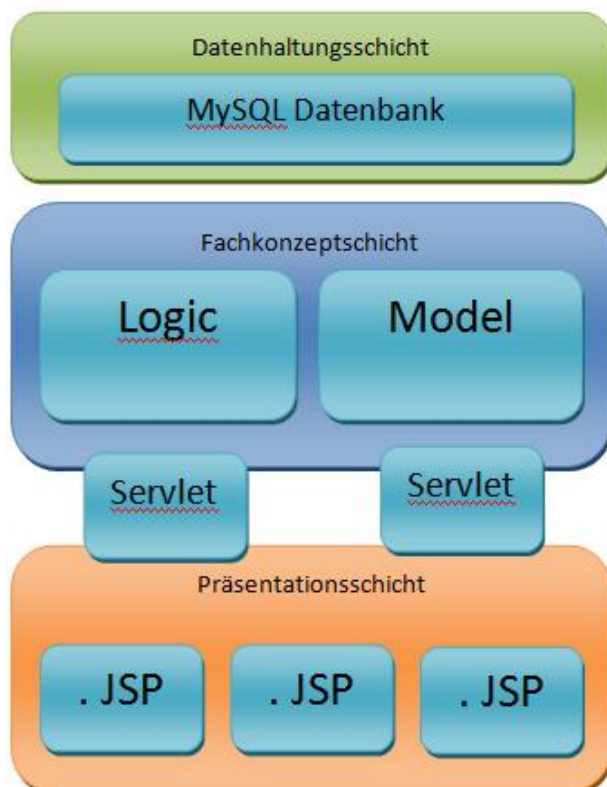


Abbildung 10: Drei-Schichten-Architektur

¹ vgl. <http://www.itwissen.info/definition/lexikon/Drei-Schichten-Architektur-three-tier-architecture.html>

3.4.3 Technisches Klassendiagramm

Das technische Klassendiagramm zeigt die umgesetzten Klassen und deren Beziehungen untereinander. Im Projekt werden statische Klassen als so genannte „Service-Klassen“ eingesetzt. (ContactManager, DataSource, UserManager, GroupManager, Validation, LayoutModules, Convert) Es ist nicht möglich von diesen Klassen Objekte/Instanzen zu bilden, wodurch diese keine direkten Beziehungen zu den fachlichen Klassen haben.

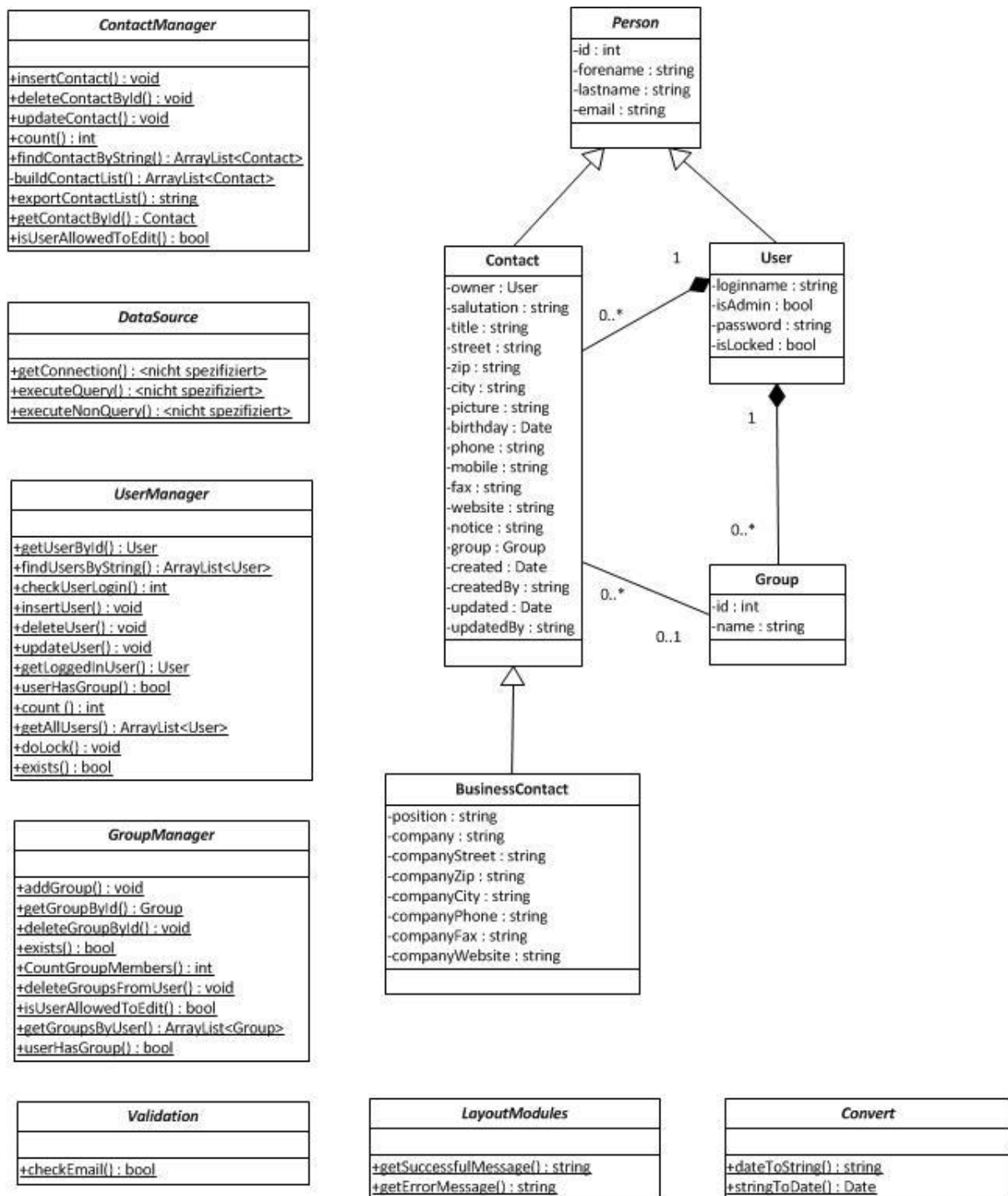


Abbildung 11: Technisches Klassendiagramm

3.4.4 Package-Diagramm

Anhand des folgenden Package-Diagramms wird gezeigt, welche Packages importiert und verwendet werden.

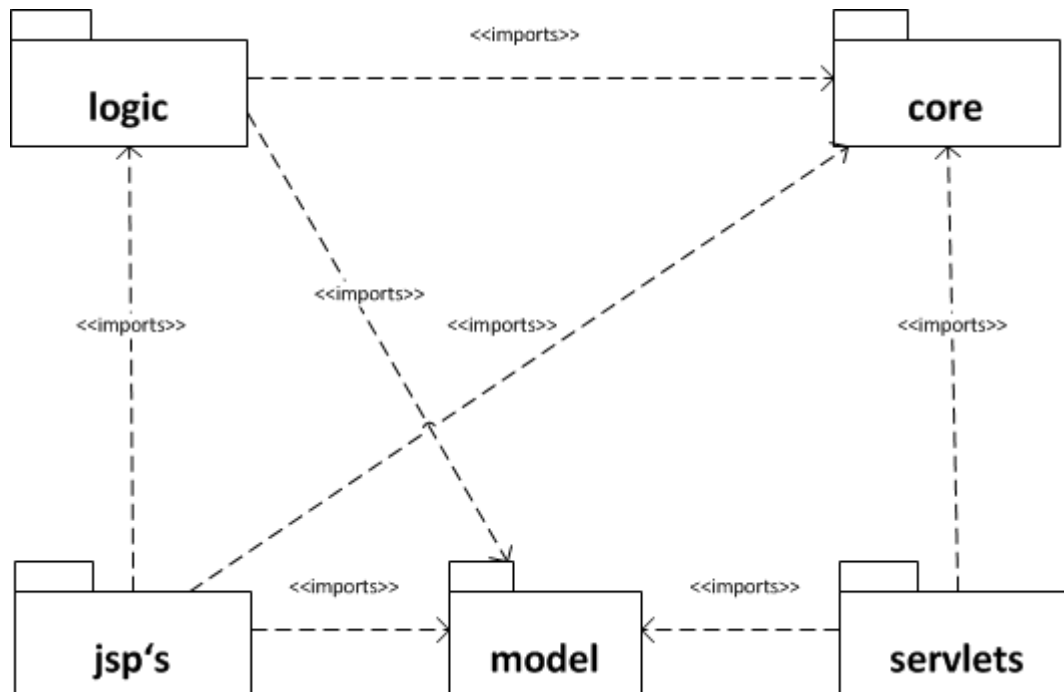


Abbildung 12: Package-Diagramm

3.5 Externe Komponenten, Module und Quellen

Zur Erweiterung des Funktionsumfanges wurden während der Implementierungsphase verschiedene externe Komponenten und Module im Projekt eingesetzt, die folgend beschrieben werden.

Apache Fileupload (Version 1.2.2)

Diese Komponente wurde eingesetzt, um einen Upload von Kontaktfotos zu ermöglichen.

Quelle & Download: <http://commons.apache.org/fileupload>

Dokumentation: <http://commons.apache.org/fileupload/apidocs/index.html>

Apache Commons IO (Version 2.1)

Ist eine weitere Komponente von Apache, die zur Benutzung der Fileupload Library (siehe oben) benötigt wird.

Quelle & Download: <http://commons.apache.org/io>

Dokumentation: <http://commons.apache.org/io/api-release/index.html>

Um die oben beschriebenen Libraries nutzen zu können, müssen die entsprechenden .jar Dateien (commons-fileupload-1.2.2.jar bzw. commons-io-2.1.jar) in das lib-Verzeichnis des Webserver (hier: D:\Programme\Tomcat\lib) kopiert werden und ein Neustart des Tomcat durchgeführt werden.

jQuery JavaScript Library (Version 1.7.1)

Die jQuery ist eine kostenlose JavaScript Library, die das Entwickeln von JavaScript's vereinfacht. Zusätzlich gibt es viele fertige JavaScript-Module, die man schnell auf eigenen Webseiten einbinden kann.

Quelle & Download: <http://jquery.com>

Dokumentation: http://docs.jquery.com/Main_Page

In diesem Projekt wurde zusätzlich die **jQuery UI Library (Version 1.8.17)** eingesetzt, um ein Kalenderelement zum Auswählen eines Geburtsdatums zu vereinfachen.

Quelle & Download: <http://jqueryui.com>

Dokumentation: <http://jqueryui.com/demos>

Regular Expression für Überprüfung einer E-Mail Adresse

Zur Validierung einer E-Mail-Adresse wurde folgender „Regulärer Ausdruck“ verwendet.

Quelle: <http://www.mkyong.com/regular-expressions/how-to-validate-email-address-with-regular-expression/>

Grafiken

Alle im Projekt verwendeten Grafiken wurden aus der „Open Icon Library“ entnommen. Die Verwendung ist kostenlos, wie ein Ausschnitt aus den LICENSES zeigt:

„ All these icons are free and open source. That means they are free to use, modify, and redistribute, without any cost. They can be used for commercial and non-commercial projects or products, royalty free. The only requirement is that they remain free and open source, and you have to give credit where credit is due. So if you use the icons in your project, you have to offer the icons or any derivatives of the icons with the same free/open license.“

Quelle: <http://openiconlibrary.sourceforge.net/>

Session Counter

Zum Zählen der angemeldeten Benutzer wird der SessionCounter von www.startdeveloper.com verwendet.

Quelle & Dokumentation:

<http://www.stardeveloper.com/articles/display.html?article=2001112001&page=1>

4 Testphase

In der Testphase wurden alle im Testplan (siehe Kapitel 2.6) vorgesehenen Szenarien getestet und ggf. aufgetretene Fehler behoben.

Besonders beachtet wird, dass die Entwickler nicht ihre jeweils eigens entwickelten Programmkomponenten testen, sondern jeweils die eines anderen Entwicklers. Dadurch kann das Risiko verringert werden, dass nur das getestet wird, was auf jeden Fall funktioniert.

TestNr.	Testfall	erwartetes Testergebnis	Ergebnis	Fehlerkorrektur
Sicht: Benutzer				
T001	Login	Der Benutzer hat sich mit seinem Benutzernamen und Passwort erfolgreich in das System eingeloggt. Bei falschen Benutzerdaten und falschen Passwort gibt es eine Fehlermeldung.	✓	keine
T002	Logout	Der Benutzer hat sich nach dem anklicken des Feldes "Logout" erfolgreich aus dem System ausgeloggt.	✓	
T003	privaten Kontakt anlegen	Der Benutzer kann einen privaten Kontakt, mit folgenden Feldern anlegen: - Anrede, Titel, Vorname(Pflichtfeld), Nachname(Pflichtfeld), Geburtstag, Straße, PLZ/Ort, Telefon, Mobil, Fax, E-Mail, Webseite, Notiz Zusätzlich kann der Benutzer ein Bild zu dem Kontakt hinterlegen. Desweiteren kann der Benutzer seine Kontakte in einer Gruppe zuordnen. Die E-Mail-Adresse wird auf Gültigkeit überprüft.	✓	Keine
T004	öffentlich Kontakt anlegen	siehe T003	✓	Keine
T004	Kontakt bearbeiten	Der Benutzer kann den Kontakt auf seine einzelnen Felder bearbeiten.	✓	Keine
T005	Kontakt Löschen	Der Benutzer kann den Kontakt vollkommen löschen.	✓	Keine
T006	Kontakt details öffnen	Der Benutzer kann eine übersicht des Kontaktes öffnen.	✓	Keine
T007	Suche nach Kontakten	Der Benutzer kann nach Kontakten suchen.	✗	<u>Fehler:</u> Es werden nur private Kontakte gefunden! Fehler durch Änderung der SQL-Abfrage behoben.
T009	Daten in CSV Exportieren	Alle Kontakte können in CSV Exportiert werden.	✗	<u>Fehler:</u> Umlaute werden nicht korrekt angezeigt! Fehler durch Änderung des Zeichensatzes in ISO 8859-1.
T010	Kontakte Drucken	Der Benutzer kann einzelne Kontakte Drucken.	✓	Keine
T011	Kontakt Gruppen erstellen	Der Benutzer ist in der Lage Gruppen für die Kontakte anzulegen.	✓	Keine
T012	Kontakt Gruppen Löschen	Der Benutzer ist in der Lage die angelegte Gruppe zu löschen.	✓	Keine
Sicht: Nicht registrierter Benutzer				
T013	Suche nach öffentl. Kontakten	Der nicht registrierte Benutzer ist in der Lage nach öffentlichen Kontakten zu suchen.	✓	Keine
Sicht: Administrator				
T016	Benutzer anlegen	Der Administrator ist in der Lage neue Benutzer anzulegen.	✓	Keine
T017	Benutzer bearbeiten	Der Administrator ist in der Lage Benutzer zu bearbeiten.	✗	<u>Fehler:</u> Änderung des Loginnamens wird nicht in die Datenbank übernommen! Fehler durch Anpassung der SQL-Abfrage behoben.
T018	Benutzer Löschen	Der Administrator ist in der Lage Benutzer zu löschen.	✓	Keine
T019	Benutzer sperren	Der Administrator ist in der Lage Benutzer zu sperren.	✓	Keine
T020	Benutzer suchen	Der Administrator ist in der Lage Benutzer zu suchen.	✓	Keine

Abbildung 13: Testplan

5 Projektabschluss & Fazit

5.1 Erreichung der Anforderungen

Die in 1.1.1 definierten Muss-Funktionalitäten der Software konnten vollständig implementiert werden und sind folgend nochmals aufgeführt:

- Benutzerlogin
- Erfassung, Änderung und Löschung der Daten (Name, Adresse, Telefon, Freitext, ...)
- Suchen nach Kontakten (Volltextsuche)
- Kontakte können von Benutzern per Volltextsuche gefunden werden
- Benutzereingaben werden auf Plausibilität untersucht (insbesondere E-Mail-Adresse und Geburtstag)
- Speicherung der Daten in einer Datenbank

Die als optional eingestuften Funktionalitäten

- Filtermöglichkeit auf einzelne Spalten eines Kontaktes
- Benutzerverwaltung durch Administratoren (Anlegen, Bearbeiten, Löschen und Sperren von Benutzern)
- Druck von Einzelkontakten
- Export der Kontaktdaten in eine CSV-Datei
- Möglichkeit zur Erstellung von Kontaktgruppen für persönliche Kontakte
- Auswertungsfunktionen
 - o Welche und wie viele Benutzer sind aktuell online?
 - o Welche Kontakte wurden zuletzt geändert?

wurden ebenfalls erfolgreich in die Software integriert und sind als zusätzliche Funktionen nutzbar.

Um das Zeitbudget des Projektes nicht zu überschreiten, wurden die optionalen Funktionalitäten

- „Kennwort vergessen“-Funktion
- Anzeige der Adresse per GoogleMaps

nicht umgesetzt.

5.2 Soll / Ist - Vergleich

Der geplante Zeitaufwand für die 3 Projektmitglieder von insgesamt 169 Stunden (≈ 57 Stunden/pro Projektmitglied) konnte bis auf eine geringe Abweichung von 14 Stunden eingehalten werden, sodass bei Projektabschluss 183 Stunden Arbeitsaufwand verbucht wurden. Die zusätzlichen Stunden entstanden durch nicht eingeplante Zeiten für Code-Review und Schwierigkeiten im Bereich des Bilduploads.

Aktivität		Bearbeiter BO = Böselager ST = Steffen BR = Braun	Start	Ende	SOLL (Std.)	IST (Std.)
1	Fachkonzepterstellung					
1.1	Zieldefinierung, Funktionsumfang	BO, ST, BR	12. Jan.	12. Jan.	2,0	2,0
1.2	Erstellung Projektplan	BO, ST, BR	12. Jan.	12. Jan.	2,0	2,5
1.3	Erstellung Use-Case-Diagramm	BR	13. Jan.	20. Jan.	3,0	3,5
1.4	Erstellung Klassendiagramm	ST	13. Jan.	20. Jan.	4,0	4,5
1.5	Erstellung Packagediagramm	BR	28. Feb.	5. Mrz.	2,0	1,0
1.6	Erstellung ERM	BO	13. Jan.	20. Jan.	3,0	3,0
1.7	Erstellung Prototyp	BO, ST, BR	20. Jan.	27. Jan.	4,0	9,0
2	Aufsetzen der MySQL-Datenbank	BO	1. Feb.	1. Feb.	4,0	3,0
3	MVC-Modell in Softwarearchitektur implementieren	BO	3. Feb.	5. Feb.	6,0	6,0
4	Fundament der HTML-Oberfläche	BO, ST, BR	2. Feb.	4. Feb.	5,0	8,0
5	Benutzerlogin					
5.1	HTML-Oberfläche	ST	1. Feb.	1. Feb.	4,0	1,0
5.2	Anmelde-logik	BR	2. Feb.	2. Feb.	8,0	4,0
5.3	Eigene Benutzerdaten pflegen	ST, BO	17. Feb.	24. Feb.	2,0	7,0
6	Kontaktverwaltung					
6.1	HTML-Oberfläche					
6.1.1	Formular Anlage/Bearbeitung eines Kontaktes	ST	4. Feb.	29. Feb.	4,5	5,5
6.1.2	Übersicht aller Kontakte in einer Liste	BR	4. Feb.	4. Feb.	3,5	3,0
6.1.3	Einzelkontaktdarstellung	BO	6. Feb.	24. Feb.	3,5	2,0
6.1.4	Suche, Filterung und entsprechende Ergebnisdarstellung	ST	10. Feb.	20. Feb.	3,0	1,0
6.1.5	Druck von Einzelkontakten und Kontaktlisten	BO	15. Feb.	15. Feb.	5,0	3,0
6.2	Logik					
6.2.1	Anlage/Bearbeitung/Löschung eines Kontaktes	BO	4. Feb.	29. Feb.	9,0	15,0
6.2.2	Übersicht aller Kontakte in einer Liste	BR	4. Feb.	29. Feb.	5,0	6,0
6.2.3	Einzelkontaktdarstellung	ST	6. Feb.	17. Feb.	4,5	3,5
6.2.4	Globale Suche	BO	8. Feb.	14. Feb.	2,0	6,0
6.2.5	Filterung	BO	25. Feb.	5. Mrz.	3,0	4,0
6.2.6	Export der Kontaktdaten in CSV-Datei	BO	19. Feb.	19. Feb.	5,0	5,0
6.2.7	Erstellung von Kontaktgruppen implementieren	BR	20. Feb.	22. Feb.	6,0	5,5
6.2.8	Bildupload	BO	8. Feb.	29. Feb.	3,0	10,0
7	Gruppenverwaltung					
7.1	Logiken für Gruppe löschen und anlegen	ST	10. Feb.	21. Feb.	3,0	6,0
7.2	Oberfläche für Gruppenverwaltung mit Logik verknüfen	BO	15. Feb.	23. Feb.	1,0	2,0
7	Administratorbereich					
8.1	Benutzerpflege (anlegen, bearbeiten, löschen, sperren)	ST	20. Feb.	5. Mrz.	16,0	15,0
8.2	Auswertungen (Wer ist online? Wie viele sind online?)	ST	20. Feb.	5. Mrz.	8,0	4,0
9	Testphase					
9.1	Testen	BR	1. Mrz.	10. Mrz.	15,0	12,0
9.2	Korrektur aufgefallener Fehler	BO, ST, BR	1. Mrz.	10. Mrz.	10,0	9,0
10	Code-Review	BO, ST, BR	1. Mrz.	10. Mrz.	0,0	2,0
11	Dokumentation	BO, ST, BR	23. Feb.	15. Mrz.	10,0	9,0
					SOLL	169,0
					IST	183,0
					Abweichung	14,0

Abbildung 14: Projektplan (inkl. SOLL/IST-Vergleich)

5.3 Fazit

Es kann festgehalten werden, dass der Einsatz des Tools „Git“ zur Verwaltung und Versionierung des Quellcodes die Zusammenarbeit der Projektmitglieder während der Implementierungsphase positiv beeinflusst hat. Nur so war gewährleistet, dass keine Änderungen/Anpassungen überschrieben und damit verloren gehen und, dass die Arbeit von mehreren Projektmitgliedern an einer Datei möglich war.

Außerdem lässt sich festhalten, dass die intensive Planung des Projektes erheblich zum Projekterfolg beigetragen hat. Besonders die Einführung von Warnstufen (siehe Projektplan auf der beiliegenden CD) bei Nichteinhaltung von End-Terminen ließ die Projektmitglieder schnell erfassen, welche Arbeitspakete das zeitliche Projektziel akut gefährdeten.

In Anbetracht der Ergebnisse lässt sich abschließend sagen, dass das Projekt „ContactManager“ erfolgreich beendet wurde und die erlangten Kenntnisse in zukünftigen Projekten eingesetzt werden können.