



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Oliver Willhöft

User-centered Design und Realisierung einer Web-
Suchmaschine für Pflegeeinrichtungen

Oliver Willhöft

User-centered Design und Realisierung einer Web-
Suchmaschine für Pflegeeinrichtungen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Olaf Zukunft
Zweitgutachter : Prof. Dr. Stefan Sarstedt

Abgegeben am 16.04.2013

Oliver Willhöft

Thema der Arbeit

User-centered Design und Realisierung einer Web-Suchmaschine für Pflegeeinrichtungen

Stichworte

User-centered Design, Suchmaschine, Pflege, Pflegeheim, Visualisierung, Web, Play!-Framework

Kurzzusammenfassung

Gegenstand dieser Arbeit ist die Entwicklung einer Webanwendung zum Finden einer Pflegeeinrichtung unter Beachtung von Nutzerwünschen. Hierfür werden Nutzerworkshops während des Entwicklungsprozesses durchgeführt und die Ergebnisse umgesetzt. Die Software findet Pflegeeinrichtungen, vergleicht, bewertet und filtert die Ergebnisse nach Benutzerangaben. Für den Nutzer werden die Ergebnisse übersichtlich dargestellt. Entwickelt wird die Software in der Programmiersprache Java unter Verwendung des Play!-Webframeworks.

Oliver Willhöft

Title of the paper

User-centered Design and development of a search engine for care retirement homes.

Keywords

User-centered Design, search engine, nursing, care retirement home, visualization, Web, Play!-Framework

Abstract

Purpose of this thesis is to develop a web application finding retirement homes altered by user requirements. During the development process there will be user workshops performed and their findings will be assimilated into it. The software finds retirement homes, compares, evaluates and screens the results based on user demands. The outcome will be presented to the user in neat way. The application will be developed in the coding language Java with use of the Play! web framework.

Inhaltsverzeichnis

1	Einleitung	7
2	Grundlagen	9
2.1	Verfügbarkeit fachlicher Daten	9
2.2	Usability, User Experience und User Centered Design	13
2.2.1	Klassische Usabilityanalyse mit Eyetracking und Mousetracking im Labor ...	14
2.2.2	Virtuelle Aufmerksamkeitsanalyse.....	14
2.2.3	A/B-Test.....	15
2.3	Anforderungen an die Gestaltung der Oberfläche	15
3	Analyse bestehender Webapplikationen.....	18
3.1	Usability bestehender Pflegeheimfinder	18
3.2	AOK-Pflegeheimnavigator	19
3.3	BKK-Pflegefinder	21
3.4	Der Pflegekompass.....	23
3.5	Der Pflegelotse	25
3.6	Übersicht des Ergebnisses.....	27
4	Soll-Analyse.....	28
4.1	Vision	28
4.2	Überlegungen zur Umsetzung der Vision	29
4.3	Zielgruppenanalyse	30
4.4	Nutzerworkshop.....	31
4.4.1	Vorbereitung des ersten Nutzerworkshops.....	31

4.4.2	Durchführung und Ergebnis des ersten Nutzerworkshops	33
4.4.3	Vorbereitung des zweiten Nutzerworkshops.....	35
4.4.4	Durchführung und Ergebnis des zweiten Nutzerworkshops.....	35
4.5	Anforderungsanalyse	37
4.5.1	Funktionale Anforderungen	37
4.5.2	Nicht-Funktionale Anforderungen	39
4.5.3	Ausgrenzung.....	40
4.5.4	Anwendungsfälle.....	41
4.5.5	Anwendungsfall Pflegeheim finden	41
4.5.6	Anwendungsfall neue Bewertung vornehmen	43
4.5.7	Anwendungsfall Details anzeigen lassen	44
4.6	Auswertung der Daten durch das System.....	45
5	Systemarchitektur	46
5.1	Fachliches Datenmodell	46
5.2	Fachliches Komponentendiagramm	48
5.3	Verwendete Frameworks.....	48
5.4	Technisches Komponentendiagramm	50
5.5	Aufbau der Komponenten	52
5.5.1	Controllers Innensicht	52
5.5.2	findeEinrichtungen Innensicht	54
5.5.3	bewerteEinrichtungen Innensicht.....	55
5.5.4	models Innensicht	57
5.6	Datenbankmodell.....	58
6	Realisierung.....	60
6.1	Weboberfläche mit Play!, Twitter Bootstrap und HTML	60
6.2	Datenbankzugriff.....	62
6.3	Komponente controllers	65
	Implementierung HomeController und SuchergebnisController	65
6.4	Komponente findeEinrichtungen.....	67
6.5	Komponente bewerteEinrichtungen	68
	Implementierung bewerteEinrichtungen	68

6.6	Komponente views.....	70
6.7	Softwaretests	74
6.7.1	Durchgeführte Softwaretests.....	75
6.7.2	Implementierung Softwaretests	76
7	Auswertung.....	77
7.1	Fazit	77
7.2	Mögliche Verbesserungen	78
8	Zusammenfassung und Ausblick	80
8.1	Zusammenfassung	80
8.2	Ausblick	81
A	Anhang.....	82
	Screenshots Mock-Up der Oberfläche für den Workshop.....	82
B	Glossar	86
C	Tabellenverzeichnis.....	87
D	Abbildungsverzeichnis	88
E	Literaturverzeichnis.....	90

1 Einleitung

Aufgrund der demografischen Entwicklung wird der Altersdurchschnitt der deutschen Bevölkerung weiterhin steigen. Damit einhergehend steigen auch die altersbedingten Krankheiten und Behinderungen, die eine professionelle Betreuung notwendig machen. Diese Betreuung könnte durch einen Pflegeheimplatz gegeben sein. Das Angebot ist bereits heute so vielfältig, dass eine analoge Suche langwierig und unübersichtlich ist. Hierbei spielt die ständig steigende Anzahl von Pflegeheimen eine Rolle, die sich in ihren Dienstleistungen und in ihrer Ausstattung stark unterscheiden. Eine Suchmaschine gestaltet diese Arbeit komfortabler.

Die in dieser Arbeit entwickelte Anwendung soll Pflegeeinrichtungen innerhalb Deutschlands finden, sie miteinander vergleichen und eine Rangliste anzeigen. Der Nutzer soll die Reihenfolge der Ergebnisse beeinflussen können, sodass eine Auflistung der passendsten Pflegeeinrichtungen dargestellt wird. Er wird verschiedene Kriterien an ein Pflegeheim auswählen können und so seine persönlichen Wünsche und Prioritäten in die Suche einbringen. Die Software übernimmt dann die Auswertung und präsentiert eine personalisierte Auswahl der Ergebnisse.

„Gut die Hälfte aller Senioren nutzt bereits das Internet“ (Grabitz, 2010). Aus seiner Berufsvorgängerzeit weiß der Autor, dass einige Nutzer Angst vor der Komplexität von Computern haben. Für diese Zielgruppe sollte die Oberfläche so aufbereitet sein, dass sie sich schnell und einfach zurechtfinden. Bisherige auf dem Markt befindliche Pflegeheimfinder bieten bereits eine Suchfunktion an. Die Gestaltung der Oberflächen scheint aber veraltet und nicht benutzerfreundlich. Sie richten sich nicht an die Bedürfnisse der Zielgruppe. Die meisten Suchmaschinen werden zudem von Pflegekassen betrieben und scheinen nur ihre eigenen Daten über Pflegeeinrichtungen abzubilden.

Ein weiteres Ziel dieser Bachelorarbeit ist die Entwicklung einer Applikation, die sich konkret an den Bedürfnissen der Nutzer aus genannter Zielgruppe orientiert. Dazu sollen innerhalb des Entwicklungsprozesses Nutzerworkshops stattfinden. Dort soll identifiziert werden, welche Informationen die Nutzer über eine Pflegeeinrichtung haben möchten. Weiterhin sollen Ideen zur Gestaltung der Oberfläche dazu beitragen die Suchmaschine mit einem hohen Bedienkomfort auszustatten. Ziel ist es den Bedienkomfort der graphischen

Oberfläche so zu erhöhen, so dass sie möglichst intuitiv zu bedienen ist und auch weniger erfahrenen Internetnutzern eine hohe Nutzungsqualität bietet.

Um alle diese Ziele umzusetzen wurde folgendes Vorgehen gewählt. Im nachkommenden Kapitel 2 werden die Grundlagen dargestellt, auf denen diese Arbeit basiert. Dies sind zum einen die öffentlich verfügbaren Pflegedaten, die für die Vergleichbarkeit der Einrichtungen essentiell sind. Zum anderen werden die Thematik der Software-Ergonomie und der Methoden beschrieben, wie benutzerfreundliche Oberflächen unter Einbeziehung des Nutzers in den Entwicklungsprozess erarbeitet werden können. In Kapitel 3 werden bestehende Pflegeheim-Suchmaschinen vorgestellt und ihre Gebrauchstauglichkeit anhand von vier Grundsätzen untersucht. Anschließend werden im Kapitel 4 die Anforderungen der Zielgruppe an eine Pflegeheim-Suchmaschine mithilfe von Workshops identifiziert. Nach einer genauen Analyse der Ergebnisse werden die Anforderungen in Teilaufgaben gruppiert und im Kapitel 5 bei der Entwicklung des Softwaredesigns bearbeitet. Es werden die technischen und fachlichen Komponenten und das Datenbankmodell sowie die Zusammenhänge untereinander dargestellt. Im 6. Kapitel wird die Umsetzung der Software ausführlich beschrieben. Besonderes Augenmerk wird auf die verwendete Technik und die eingesetzten Frameworks gelegt. Und im letzten Kapitel wird ein Fazit gegeben.

2 Grundlagen

Dieses Kapitel liefert einen Überblick über die Qualitäts- und Leistungskriterien von Pflegeheimen sowie einen Ausblick, welche Informationen für einen Kunden zur Verfügung gestellt werden sollten. Weiterhin wird ein kleiner Überblick über die Themen Benutzerfreundlichkeit und Design von grafischen Oberflächen gegeben.

2.1 Verfügbarkeit fachlicher Daten

Um Pflegeeinrichtungen vergleichen zu können werden verschiedene Kennzahlen und Dienstleistungen benötigt, wie etwa Preis, Lage und Leistungsumfang. Die Applikation soll einen Überblick über diese Informationen liefern.

Die Standorte von Pflegeheimen sind entweder im Branchenverzeichnis oder im Internet erhältlich. Um die Leistungen gegenüberstellen zu können benötigt der Nutzer mehr Informationen als nur die Lage der Einrichtung. Diese Informationen werden zwischen Pflegekasse und Pflegeeinrichtung in der sogenannten Leistungs- und Qualitätsvereinbarung (LQV)¹ schriftlich vereinbart. Grundlage für die LQV ist unter anderem der Rahmenvertrag nach §75 (Sozialgesetzbuch) SGB XI, der Versorgungsvertrag nach §132, 132a SGB V und die Vergütungsvereinbarung nach §§84 ff. SGB XI. „Zur Ermittlung der Vergleichsdaten ist vorrangig auf die verfügbaren Daten aus den Versorgungsverträgen, den LQV sowie den Pflegesatz- und Entgeltvereinbarungen [...] zurückzugreifen.“ (Klie, 2005, p. 415)

Eine weitere rechtliche Grundlage für den direkten Vergleich zweier Heime ist im §92a „Rechtsverordnung zum Pflegeheimvergleich“ im Pflege-Qualitätssicherungsgesetz (PQsG)¹ geregelt. Dort steht „(6) Die Spitzen- und Landesverbände der Pflegekassen sind berechtigt, jährliche Verzeichnisse der Pflegeheime mit den im Pflegeheimvergleich ermittelten Leistungs-, Belegungs- und Vergütungsdaten zu veröffentlichen.“ (Klie, 2005, p. 416)

Optimal wäre eine einheitliche Datenerhebung zwischen Pflegeeinrichtungen und Landesverbänden. Zu Vergleichszwecken sollten diese öffentlich zugänglich sein.

Die Literaturrecherche ergab, dass die Gesetze kaum eingehalten oder von den beteiligten Gruppen unterschiedlich ausgelegt werden.

¹ Erläuterung siehe Glossar

„Der Haken hierbei ist, dass Kostenträger [...] mehr Leistungen und [...] Qualität [...] verlangten, aber dafür keineswegs mehr Entgelt als bisher verhandeln und bezahlen wollen. Der ursprüngliche Zweck der LQV, nämlich Wettbewerb und Vergleich der Einrichtungen untereinander, ist so kaum noch gegeben.“ (König, 2007, p. 3)

Die Daten werden nur zum Teil veröffentlicht, denn „Die erhobenen Daten werden nur auf Landesebene der Vereinigungen der Heimträger, den Pflegekassen, dem Medizinischen Dienst der Krankenkassen (MDK)² und dem Sozialhilfeträger zugänglich gemacht. [...] Die Spitzen- und Landesverbände der Pflegekassen sind berechtigt, jedes Jahr eine Liste zu veröffentlichen, die einen Heimvergleich in den Bereichen Entgelt, Leistung und Belegung ermöglicht.“ (König, 2007, p. 95)

Die gesetzliche Regelung bietet die Rahmenbedingungen, um Daten in oben genannten Kriterien zu veröffentlichen. Jedoch entscheidet jeder Landesverband selbst, welche Daten er veröffentlichen möchte und in welcher Struktur dies geschieht. Zudem werden im Bereich Sächliche Ausstattung nur teilweise Angaben und im Bereich Personelle Besetzung keine Angaben veröffentlicht.

Ein elementarer Punkt dieser Daten ist die Pflegenote. „Diese Noten setzen sich aus vielen Einzelbewertungen zusammen, die im Rahmen der gesetzlichen Überprüfung der Pflegeheime und -dienste durch die Medizinischen Dienste der Krankenkassen ermittelt werden.“ (Spitzenverband Pflegenoten, 2012) Dabei reicht das Bewertungssystem von eins bis fünf, wie es aus der Schule bekannt ist. Zudem wird ein landesweiter Durchschnitt angezeigt, damit der Verbraucher sehen kann, wie die Gesamtnote zu bewerten ist.

Der Vergleich von vier Internetveröffentlichungen der Landesverbände ergab stark voneinander abweichende Daten. Das Ergebnis der Recherche wird in *Tabelle 1 Übersicht Pflegeheimdaten der Landesvereine* angezeigt. An dieser Tabelle wird sichtbar, dass die Landesverbände unterschiedliche Daten veröffentlichen.

² Erläuterung siehe Glossar

Tabelle 1 Übersicht Pflegeheimdaten der Landesvereine

	AOK	BKK	Knappschaft	vdek
Standort	- Straße & Nr., PLZ - Ort - Telefon - Email	- Straße & Nr., PLZ - Ort - Telefon - Email	- Straße & Nr., PLZ - Ort - Telefon - Email	- Straße & Nr., PLZ - Ort - Telefon - Email
Versorgungsstruktur	- Vollstationäre Pflege - Kurzzeitpflege - Kein Pflegeschwerpunkt - Pflege demenzerkrankter Menschen - Pflege beatmungspfl. Menschen - Pflege von Menschen im Wachkoma	- Vollstationäre Pflege - Kurzzeitpflege	- Vollstationäre Pflege - Kurzzeitpflege - Kein Pflegeschwerpunkt - Pflege demenzerkrankter Menschen - Pflege beatmungspfl. Menschen - Pflege von Menschen im Wachkoma - Pflege von Menschen mit MS - Pflege alkoholabhängiger Menschen - Pflege von Menschen mit Alzheimer	- Vollstationäre Pflege - Kurzzeitpflege - Kein Pflegeschwerpunkt - Pflege demenzerkrankter Menschen - Pflege beatmungspfl. Menschen - Pflege von Menschen im Wachkoma - Pflege von Menschen mit geistiger Behinderung - Pflege von Menschen mit chronischer psychischer Erkrankung - Pflege von Menschen mit MS - Pflege von Menschen mit gerontopsychiatrischer Erkrankung
Sächliche Ausstattung	Heimgröße; freie Plätze; Ein- oder Mehrbettzimmer; Telefon auf den Zimmern; Sauna; Hallenbad; Internet auf dem Zimmer; Fernsehen auf dem Zimmer; Gartenanlage; Ladengeschäft auf dem Gelände;		Heimgröße; Balkon; Café; Citylage; Einzelzimmer; Garten; Kapelle; Komfort-Einzelzimmer; Menüwahl; Raum der Stille; Sinnesgarten; Snoezelraum ³ ; Stadtrandlage; Terrasse; Wintergarten;	Heimgröße

³ Erläuterung siehe Glossar

	Bibliothek		Bewegungsbad; Sauna; Schwimmbad; Bibliothek; Kegelbahn; Frisör; Kiosk; Veranstaltungsraum	
Heimentgelt	- tägliches Heimentgelt Pflegestufe I, II & III	- tägliches Heimentgelt Pflegestufe I, II & III - monatliches Heimentgelt I, II, & III	- monatliches Heimentgelt I, II, & III	- monatliches Heimentgelt I, II, & III
Leistungen und Pflegequalität	- Pflegenote und MDK-Transparenzbericht - Tagesgestaltung - Beschäftigungstherapie - Ausflüge - Veranstaltung von Festen - Betreuung nach § 87b SGB XI ⁴ - Einsatz von Expertenstandards - Anbindung ans öffentliche Verkehrsnetz in Minuten - Gottesdienste - Tierhaltung - Übernachtung für Angehörige	- Pflegenote und MDK-Transparenzbericht - Betreuung nach § 87b SGB XI ⁴	- Pflegenote und MDK-Transparenzbericht - Betreuung nach § 87b SGB XI ⁴ - Gottesdienste - Fahrdienst - Tierhaltung	- Pflegenote und MDK-Transparenzbericht - Betreuung nach § 87b SGB XI ⁴
Anmerkung	Die zur Verfügung gestellten Daten unterscheiden sich je nach Einrichtung.		Kriterien innerhalb der sächlichen Ausstattung werden nach der Oder-Verknüpfung ausgewählt	

⁴ Erläuterung siehe Glossar

2.2 Usability, User Experience und User Centered Design

Allgemein gültige Erklärungen für Usability, User Experience und User Centered Design zu finden ist mit Schwierigkeiten verbunden, da bisher keine einheitlichen Definitionen festgelegt wurden. Im folgenden Abschnitt werden darum einige Erläuterungen zu jedem der drei Begriffe gegeben. „Usability as a discipline attempts to determine how successfully a user can complete a task and how satisfying a device or interface might be to use.“ (Connor, 2012, p. 271)

„Während sich die Usability eigentlich nur mit den technischen und objektiven Faktoren befasst, schließt die User Experience zusätzlich noch alle psychologischen und subjektiven Aspekte mit ein. [...]Die Usability präsentiert sich also lediglich als Teilbereich der User Experience. Trotzdem: eine einheitliche, klare Definition fehlt nach wie vor, was zeigt, wie jung dieser Fach- und Forschungsbereich eigentlich ist.“ (Broschart, 2011, pp. 329-330)

„User-Centered Design may be considered a practice, field, craft, framework, philosophy, discipline, or method of designing tools for human use by involving humans in the design process.“ (User-Centered Design may be considered a practice, 2009)

Es wird schnell deutlich, dass sich alle Definitionen mit dem Benutzer auseinander setzen, also spielt die Interaktion mit dem Nutzer eine entscheidende Rolle für die Visualisierung und Benutzbarkeit. Um die User Experience möglichst hoch zu halten, muss die Oberfläche verschiedenen Kriterien genügen. Diese sind barrierefreier Zugang, Nutzwert, Usability und Joy of Use. Kommen alle Faktoren zusammen, dann spricht man vom Nutzerspaß.

„Für den Nutzer repräsentieren Accessibility, Utility, Usability und Joy of Use lediglich unterschiedliche Bedürfnisqualitäten (siehe *Abbildung 1 Bedürfnispyramide des Nutzers*), die die Anwendung zu befriedigen hat. Die Befriedigung aller dieser Qualitäten erzeugt ein ganzheitliches Nutzungserlebnis bzw. sog. User Experience.“ (Ebert-Yom, 2010, p. 8)

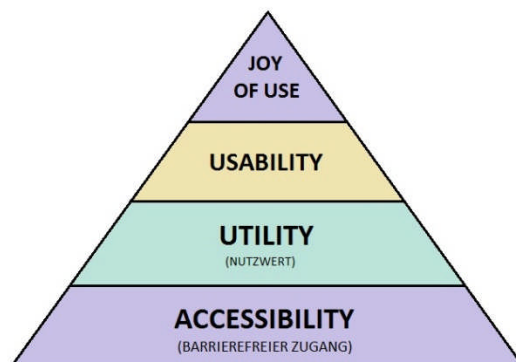


Abbildung 1 Bedürfnispyramide des Nutzers (Knoflicek, 2012)

Wie gut die Bedürfnisqualitäten tatsächlich erfüllt werden, lässt sich nur durch das Einbeziehen der Nutzer erreichen, denn „ohne Tests lassen sich keine Pauschalaussagen darüber treffen, was Besucher wollen [...].“ (Düweke & Rabsch, 2011)

Wie bereits in der Definition von User-Centered Design (UCD) erwähnt wurde, ist es am besten den Nutzer während des gesamten Entwicklungsprozesses teilhaben zu lassen. Die daraus entstehende Frage ist - wie setzt man UCD um? In der Fachliteratur werden unterschiedliche und vielfältige Methoden und Tests vorgestellt, die diese Frage beantworten. Ein guter Einstieg scheint zu sein, folgende Punkte zu beachten. „

- Verifizieren Sie die Zielgruppe
- Welche Eigenschaften kennzeichnen die Zielgruppe [...]?
- Welche Wünsche haben die Benutzer an die Webseite, [...]?
- Wie lautet die Zielsetzung der Webseite [...]?“ (Stiegert, 2011, p. 16)

2.2.1 Klassische Usabilityanalyse mit Eyetracking und Mousetracking im Labor

Bei diesem Verfahren werden Probanden eingeladen, in einem Labor eine Software-Oberfläche zu testen. Zum Einsatz kommen ein Eyetrackingsystem, welches die Augenbewegungen aufzeichnet, und ein Moustrackingsystem, welches die Mausbewegungen und Klicks aufzeichnet. Meist werden die Probanden dabei gefilmt. Die Auswertung der Testreihen erfolgt auf verschiedene Arten. Aus den Augenbewegungen kann eine sogenannte Heatmap erstellt werden. Dabei handelt es sich um eine zusätzliche Schicht, die über die Oberfläche gelegt wird. Die Bereiche der Oberfläche, auf denen die Augen lange verweilen, werden dort farblich hervorgehoben. Die Mausbewegungen können ähnlich visuell dargestellt werden. Auf einer zusätzlichen Ebene werden die einzelnen Bewegungen und Klicks mit Hilfe von Strichen und Kreisen dargestellt. Bei Sichtung des Videomaterials geben vor allem „das Laute Denken“ der Probanden Rückschlüsse auf Probleme mit der Benutzbarkeit.

Dieses klassische Testverfahren liefert die meisten Informationen über die Benutzbarkeit einer Oberfläche. Nachteilig sind dabei die kostenintensive Umsetzung und der mögliche Hawthorne-Effekt. In Untersuchungen wurde festgestellt, dass sich Menschen in einer Laborumgebung nicht so natürlich verhalten wie in einer gewohnten Umgebung. Dies könnte zu einem verfälschten Ergebnis führen.

2.2.2 Virtuelle Aufmerksamkeitsanalyse

Mittlerweile werden virtuelle Eyetrackingsysteme im Internet angeboten. Diese simulieren die ersten drei bis fünf Sekunden der Augenbewegungen eines Nutzers auf einer Webseite. Die Auswertung dauert wenige Sekunden. Die Betreiber versichern eine Genauigkeit von

94% zu realen Tests. Das Ergebnis wird ebenfalls als Heatmap präsentiert. Die geringen Kosten ermöglichen den Einsatz dieser Technik bereits im frühen Entwicklungsstadium anhand eines Mock-Ups. Es kann aber nicht die Genauigkeit und Vielfältigkeit eines Labortests abbilden.

2.2.3 A/B-Test

Beim A/B-Test werden dem Nutzer abwechselnd zwei unterschiedliche Oberflächen zur Verfügung gestellt. Aus dem Verhalten des Nutzers kann anschließend gefolgert werden, welche Version bevorzugt wird. Meist wird nur ein Aspekt der Oberfläche untersucht und nicht die komplette Anwendung, damit man schneller Rückschlüsse ziehen kann, welche Änderung einen positiven Effekt zur Folge hätte. Häufig wird der Test iterativ eingesetzt, um sich langsam dem Optimum zu nähern. Dieser Test eignet sich als Ergänzung zu anderen Verfahren, weil er vergleichsweise kostengünstig ist. Es lassen sich keine elementaren Rückschlüsse aus diesem Test generieren. Vielmehr ist es so, dass man Erkenntnisse aus vorherigen Tests umsetzt und mit dem A/B-Test feststellt, ob eine Verbesserung eingetreten ist.

2.3 Anforderungen an die Gestaltung der Oberfläche

Möchte man eine nutzerfreundliche Webseite gestalten, ist es notwendig sich mit Webdesign auseinander zu setzen. An dieser Stelle werden nur die best practices zum Webseitenlayout aufgezählt, die in dieser Arbeit genutzt werden.

Ebenso soll das Design allgemeinen Erkenntnissen bzw. best practices der Usability genügen, soweit diese vorliegen. „Follow the general principles of navigation design, [...], follow the Web standards, in order to construct self-described Web user interface with high-usability.“ (Yan & Guo, 2010)

Eine Nutzerbefragung zum Thema Nutzergerechte Gestaltung von Online-Shops der eResult GmbH aus dem Oktober 2012 ergab im Bereich Navigation folgendes Ergebnis. Die meisten befragten Nutzer, über 50%, erwarten die Navigationsleiste oder das Menü im linken Bereich der Seite. Und dicht gefolgt mit 21-30% wird diese im oberen Bereich erwartet. Niemand erwartet sie im mittleren, rechten oder unteren Bereich der Website. Siehe *Abbildung 2 Navigation/Menü Position - Nutzererwartung*

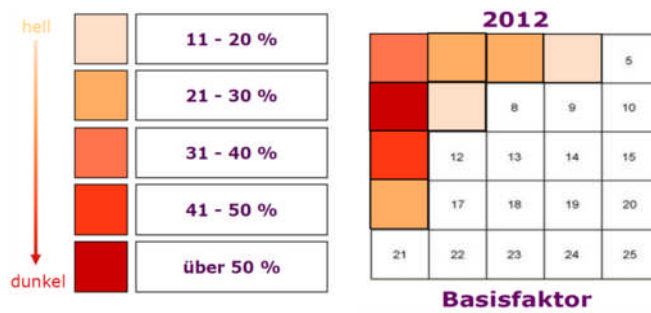


Abbildung 2 Navigation/Menü Position - Nutzererwartung

Nachdem nun feststeht, wo sich das Menü befinden sollte, ist die Wahl des richtigen Ausdrucks die nächste entscheidende Komponente für die Verständlichkeit der Webseite. Denn nur wenn der Nutzer auch versteht und nachvollziehen kann, was die Webseite ausmacht, kann er sie entsprechend nutzen.

„Three guidelines for better Web writing: skip the jargon, [...] avoid acronyms, [...] bar sarcasm, subtle word play [...]“ (Nielsen & Loranger, 2006)

Die Bezeichner aller Felder und der Inhalt der Webseite sollen diesen Grundsätzen entsprechen. Zudem sollen Fachwörter und Fachjargon vermieden werden, um so viele Menschen wie möglich anzusprechen.

Diese Anwendung wird nicht ohne Eingabefelder auskommen, da dies zur Kernfunktionalität gehören wird. Dazu ist es wichtig diese so einfach wie möglich zu gestalten, damit auch weniger versierte Nutzer damit umgehen können. Um dies zu erreichen sollen folgende Regeln beachtet werden:

Formulareigenschaften:

”

- Die visuelle Gestaltung eines Formulars sollte nach konservativen Maßstäben erfolgen.
- Fragen Sie so wenige Daten wie möglich ab.
- Vermeiden Sie viele Pflichtfelder. [...]
- Verzichten Sie auf Reset- und Zurück-Buttons.
- Die Beschriftung sollte so nah wie möglich an ihren Eingabefeldern platziert werden.
- Bieten Sie keine überflüssigen Hilfsfunktionen an. [...]
- Belassen Sie Eingabefelder leer. [...]
- Gestatten Sie die Verarbeitung aller erdenklichen, sinnvollen üblichen Eingabeformate.

“ (Broschart, 2011, p. 369 & 371)

Diese Eigenschaften sind für Formblätter maßgeblich und sollten allen Formularen zu Grunde liegen, um den Nutzer nicht zu verwirren.

Benutzerführung durch Formulare mit vielen Eingabemöglichkeiten:

”
- Gruppierung für bessere Benutzerführung [... Es] sollten umfangreiche Formularblöcke zur Erfassung von Daten in leicht handhabbaren Gruppen aufgeteilt werden. Ziel dieses Ansatzes ist es, eine Gruppierung auf Basis der semantischen Bedeutung der Elemente zueinander herstellen zu können.

“ (Stiegert, 2011)

Falls eine Anwendung viele Fakten abfragen muss, sollten die Eingabefelder übersichtlich gestaltet sein. Ansonsten wird der Nutzer vermutlich abgeschreckt und die Nutzung der Anwendung womöglich verweigern. Um dies zu vermeiden, sollten die Felder wie oben formuliert aufgebaut sein.

Null-Treffer-Anzeige:

”
Auf der Null-Treffer-Seite sollten Sie folgende Hilfestellung anbieten [...]

- Tipps zur Verbesserung des Suchergebnisses. [...]
- Häufigste Suchwörter
- Links zu FAQs oder eine Guided Tour zur richtigen Seite

“

(Ebert-Yom, 2010, p. 85)

Eine umfassende Hilfestellung bietet dem Nutzer die Chance zu einem Ergebnis zu gelangen, obwohl sein vorheriger Versuch fehlschlug. Dabei sollten oben genannte Tipps als Hilfestellung dienen.

3 Analyse bestehender Webapplikationen

Dieses Kapitel stellt vorhandene Pflegeheimfinder und deren Benutzbarkeit vor. Aus Zeitgründen handelt es sich nicht um eine vollständige Aufzählung aller Suchmaschinen, die sich mit dem Finden von Pflegeheimen auseinandersetzen. Auch die Untersuchung der Benutzbarkeit beschränkt sich auf wenige Kriterien und ist nicht repräsentativ. Das Kapitel zeigt eine Vorstellung der zurzeit genutzten Systeme und deren Stärken und Schwächen.

3.1 Usability bestehender Pflegeheimfinder

Im Internet gibt es viele Suchmaschinen, die Pflegeheime suchen. Bei dieser Ist-Analyse wird auf vier Suchmaschinen Bezug genommen. Diese Seiten dürfen offiziell die Pflegenoten, siehe 2.1, veröffentlichen. Als Grundlage dieser Untersuchung dient die ISO 9241-10. Zur Untersuchung werden die Grundsätze Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit und Erwartungskonformität herangezogen. Die Grundsätze Fehlertoleranz, Individualisierbarkeit und Lernförderlichkeit werden aus Zeitgründen nicht untersucht.

„Aufgabenangemessenheit

Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen. [...]

Selbstbeschreibungsfähigkeit

Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt. [...]

Steuerbarkeit

Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist. [...]

Erwartungskonformität

Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z.B. den Kenntnissen aus dem Arbeitsgebiet, der Ausbildung und der Erfahrung des Benutzers sowie den allgemein anerkannten Konventionen.“ (Europäisches Komitee für Normung, 1995)

3.2 AOK-Pflegeheimnavigator

Suchmaske:

Aufgabenangemessenheit:

Im AOK-Pflegeheimnavigator kann man neben dem gewünschten Standort auch verschiedene Kriterien auswählen, die das Pflegeheim erfüllen muss oder kann. Dies wird durch Anklicken des „erweiterte Suche“-Buttons ausgelöst. Dieser Grundsatz ist erfüllt.

Selbstbeschreibungsfähigkeit:

Die Kriterien sind grob unterteilt und können mehrere Unterkriterien enthalten, siehe *Abbildung 3 Suchmaske AOK-Pflegeheimfinder*. Hier wird die Selbstbeschreibungsfähigkeit

verletzt. Der Nutzer kann nicht erkennen, ob nur eine der Unterkategorien für die Suche herangezogen wird oder ob ein Kriterium genügt, um in die Auswahl aufgenommen zu werden. Als Beispiel: In der gewählten Kategorie wird das Unterkriterium aufgezählt, ob Haustiere in der Einrichtung gestattet sind. Der Nutzer weiß aber nicht, ob im Ergebnis auch tatsächlich alle Pflegeheime Haustiere erlauben. Dies zeigt sich erst, wenn das Ergebnis genauer betrachtet wird.

Steuerbarkeit:

Dieser Grundsatz ist erfüllt, siehe Aufgabenangemessenheit.

The screenshot shows the search interface of the AOK-Pflegeheimnavigator. It includes fields for location (Gemeinde/Stadt, PLZ, Umkreis), selection criteria (Auswahl), sorting (Sortierung), and various checkboxes for filters. A legend at the bottom explains the search logic: 'Und' (all criteria) and 'Oder' (at least one criterion). A tooltip is visible over the 'Liegen Informationen vor zu...?' link.

Abbildung 3 Suchmaske AOK-Pflegeheimfinder

Erwartungskonformität:

Der Nutzer kann wählen, ob alle zusätzlichen Kriterien im Suchergebnis enthalten sein müssen oder nur eines. Es geht aber aus der Beschreibung nicht eindeutig hervor, welches Kriterium gewählt wird. Falls ein Nutzer drei Kriterien auswählt, hat er aufgrund der Sortierung keine Möglichkeit herauszufinden, ob drei, zwei oder nur ein Kriterium seiner Wahl erfüllt wurde. Dies entspricht nicht der Erwartungskonformität.

Ergebnis der Suche:

Aufgabenangemessenheit:

Das Ergebnis wird in einer Tabelle dargestellt. Dabei werden Name, Ort, Kontaktmöglichkeiten und Preis angegeben. Um detaillierte Informationen anzeigen zu lassen, ist der Link „mehr Informationen“ auszuwählen. Die Detailansicht einer Einrichtung zeigt alle Leistungen und Kriterien an, die hinterlegt wurden. Die Anzeige findet teilweise mithilfe von Piktogrammen statt. Der Nutzer kann sich hier einen Überblick über die Ausstattung und Leistungen der Einrichtung verschaffen. Es werden nur Felder angezeigt, die Daten enthalten. Dies entspricht der Aufgabenangemessenheit.

Angezeigt wird der Transparenzbericht des MDK, indem die Note der Einrichtung veröffentlicht ist.

Selbstbeschreibungsfähigkeit:

Die Detailansicht zeigt die jeweils angebotenen Leistungen als Piktogramme. Dies ist ein gutes Mittel schnell und bildhaft Informationen zu übermitteln. Bei der Mouseover wird angegeben, worum es sich handelt.

Steuerbarkeit:

Der Grundsatz ist erfüllt. Der Nutzer kann nach seiner eigenen Wahl entscheiden, wann er was auswählt.

Erwartungskonformität:

Die Sortierung der Ergebnisse findet nach der Entfernung zum gewählten Standort statt.

Es erfolgt kein Ranking anderer Kriterien, siehe *Abbildung 4 Suchergebnis AOK-Pflegeheimfinder*. Der Nutzer kann sich, je nachdem welche Auswahl er getätigt hat, sicher sein, dass sein Oberkriterium bei den Ergebnissen enthalten ist.

Im oben genannten Beispiel wurde die erlaubte Tierhaltung im Pflegeheim als mögliche Unterkategorie beschrieben.

The screenshot shows a search results page for care homes. At the top, it indicates '92 Pflegeheime' and provides navigation links. Three listings are visible:

- Pflegeheim der Johann Carl Müller-Stiftung:** Saselkoppel 20, 22393 Hamburg. Contact: 040/226 336 776. Price: Pflegeplätze ab Pflegestufe I bei 67,41 € je Pflegeplatz pro Tag¹⁾. Last updated: 28.02.2012 13:14.
- Fröhlich Senioren- und Pflegepension GmbH:** Rodenbeker Str. 2 - 5, 22395 Hamburg. Contact: 040/60441110. Price: Pflegeplätze ab Pflegestufe I bei 63,96 € je Pflegeplatz pro Tag¹⁾. Last updated: 28.02.2012 13:14.
- Senioren & Pflegeheim Margarethenhof GmbH:** Wohldorfer Damm 156, 22395 Hamburg. Contact: 040 - 604 87 42. Price: Pflegeplätze ab Pflegestufe I bei 66,51 € je Pflegeplatz pro Tag¹⁾. Last updated: 04.09.2008 12:43.

Each listing includes a 'Vollstationäre Pflege' and 'Kurzzeitpflege' button, a link to the MDK transparency report, and a 'mehr Informationen' link. A note at the bottom of each listing states: ¹⁾zzgl. Investitionskosten, hierzu siehe auch "mehr Informationen".

Abbildung 4 Suchergebnis AOK-Pflegeheimfinder

Die drei untersuchten Ergebnisse hätten Informationen dazu liefern müssen. Jedoch enthielt keines der drei Suchergebnisse Angaben zur Tierhaltung. Dies widerspricht der Erwartungskonformität. Als weiteres Ergebnis ist festzustellen, dass zwei der drei Ergebnisse keine Informationen der gewünschten Oberkategorie enthielten, in diesem Fall Informationen zum Qualitätsmanagement der Einrichtung. Dies entspricht nicht der Erwartungskonformität.

3.3 BKK-Pflegefinder

Aufgabenangemessenheit:

Der Nutzer hat schnell einen Überblick und kann sofort suchen. Dies entspricht diesem Grundsatz

Selbstbeschreibungsfähigkeit:

In der Navigationsleiste kann der Nutzer eine Suche nach Preisvergleich auswählen. Die angezeigte Such-Maske ist fast identisch mit der Startseite, siehe *Abbildung 5 Suchmaske BKK-Pflegefinder*. Das Eingabefeld „Name der Einrichtung“ ist durch ein Drop-

Down-Menü „Sortierung“ ersetzt, welches es dem Nutzer ermöglicht, das Ergebnis der Suche nach einem bestimmten Merkmal zu sortieren, siehe *Abbildung 6 Suchergebnis BKK-Pflegefinder*. Die Auswahl „Preis (Pfleigestufe 1)“ gibt keine Auskunft darüber, ob die Sortierung auf- oder absteigend stattfindet. Hier ist die Selbstbeschreibungsfähigkeit verletzt.

Steuerbarkeit:

Im Pflegefinder der BKK kann neben dem Standort des Heims auch die Art der Pflege, z.B. vollstationär oder ambulant, ausgewählt werden. In dieser Maske wird der Grundsatz der Steuerbarkeit erfüllt.

BKK PflegeFinder
DER BKK PFLEGEFINDER PFLEGE-EINRICHTUNGEN SUCHE UND FINDEN

Startseite Preisvergleich Pflegeberatung Kontakt

> Pflegeeinrichtungen suchen und finden.

Finden Sie hier das Versorgungsangebot Ihrer Wahl:

PLZ oder Ort: max. Entfernung:

Name der Einrichtung:

Pflegeart:

> Gezielte Suche zum Preisvergleich

Der BKK PflegeFinder hilft Ihnen dabei, Pflegeeinrichtungen in der von Ihnen gewünschten Region zu finden und die zugehörigen Preise auf einen Blick zu vergleichen. Auf Wunsch können Sie die erhaltene Übersicht auch ausdrucken. Ein Klick auf "Details zur Einrichtung" in den Suchergebnissen zeigt Ihnen weitere Informationen zur jeweiligen Pflegeeinrichtung.

Preisvergleich

PLZ oder Ort: max. Entfernung:

Pflegeart:

Sortierung:

Abbildung 5 Suchmaske BKK-Pflegefinder

Erwartungskonformität:

Es können keine weiteren Kriterien ausgewählt werden, wie z.B. die Pflege von beatmungspflichtigen Menschen oder eine erlaubte Tierhaltung. Dies widerspricht der Erwartungskonformität.

Ergebnis der Suche:

Aufgabenangemessenheit:

Das Ergebnis wird in einer Tabelle dargestellt und enthält den Namen der Einrichtung, Straße, Ort und Entfernung zum eingegeben Standort. Dies ist nicht angemessen, weil der Nutzer erst zur nächsten Seite navigieren muss, um Kontaktinformationen zu erhalten.

Suchergebnisse							
Einrichtung	Angebote*	Kosten nach Pflege-stufe	Gesamtheimzeit (monatlich in €)		Eigenanteil Versicherter (monatlich in €)		Entfernung zur PLZ (Luftlinie)
			zzgl. Investitions-kosten**	inkl. Investitions-kosten ab	zzgl. Investitions-kosten**	inkl. Investitions-kosten ab	
Villa Eilersweg Altenheim Kinne GmbH Eilersweg 10 - 12 22143 Hamburg	▪ Vollstationäre Pflege	1	1.374,68	k.A.	351,68	k.A.	~5 km
	▪ Kurzzeitpflege	2	1.809,08	k.A.	530,08	k.A.	
	▪ Betreuung nach § 87b SGB XI***	3	2.243,17	k.A.	693,17	k.A.	
Details zur Einrichtung							
Altenheim St. Elisabeth Maria-Merkel-Str. 5 21465 Reinbek	▪ Vollstationäre Pflege	1	1.570,89	2.158,91	547,89	1.135,91	~19 km
	▪ Betreuung nach § 87b SGB XI***	2	1.888,78	2.476,80	609,78	1.197,80	
	▪ Härtefall	3	2.206,67	2.794,69	656,67	1.244,69	
			2.574,75	3.162,77	656,75	1.244,77	
Details zur Einrichtung							

Abbildung 6 Suchergebnis BKK-Pflegefinder

Selbstbeschreibungsfähigkeit:

Die Detailansicht zeigt dem Nutzer die Kontaktdaten, den Pflege-transparenzbericht mit den veröffentlichten Noten, detaillierte Preise, Stadtplan und Leistungsangebote der Einrichtung. Dabei ist mit Leistungsangebot die Art der Pflege wie in der Suchmaske gemeint. Dies verletzt die Selbstbeschreibungsfähigkeit, weil die einheitliche Terminologie nicht eingehalten wird.

Steuerbarkeit:

Für beide Suchmöglichkeiten wird das Ergebnis in einer Tabelle dargestellt. Der Nutzer kann das Ergebnis mithilfe eines Auswahlfeldes sortieren lassen. Möglichkeiten zum Sortieren sind der Standort in verschiedenen Variationen oder nach alphabetischer Reihenfolge des Einrichtungsnamens. In der Ergebnistabelle nach der Preissuche kann der Nutzer zusätzlich nach Preis auf- oder absteigend sortieren lassen. Die Tabellen geben Auskunft über Name der Einrichtung, Entfernung zum gesuchten Standort und Preis. Die Grundsatz Steuerbarkeit ist erreicht.

Erwartungskonformität:

Individuell vereinbarte Leistungen oder die Ausstattung der Einrichtung werden nicht beschrieben. Dies entspricht nicht der Erwartungskonformität, weil der Nutzer eventuell eine Einrichtung sucht, die spezielle Anforderungen erfüllen soll, wie das Mitbringen eigener Möbel oder erlaubte Tierhaltung.

3.4 Der Pflegekompass

Aufgabenangemessenheit:

Der Pflegekompass der Knappschaft bietet ebenso wie der Pflegenavigator der AOK eine einfache Suche und eine erweiterte Suche an. Neben der Ortswahl kann der Nutzer hier die Größe der Einrichtung einschränken. Die Art der Pflege, die eine Pflegeeinrichtung anbieten soll, gestattet im Vergleich zum Pflegenavigator und Pflegefinder sechs zusätzliche Pflegearten. Im Gegensatz zu den anderen hier analysierten Pflegeheimfindern, besteht die Möglichkeit mehrere Arten der Pflege auszuwählen. Dabei werden nur Dienstleister angezeigt, die alle Arten der Pflege unterstützen (Und-Verknüpfung). Dies ist angemessen für eine Zielgruppe, die sehr spezielle Bedürfnisse an die Pflege richtet.

Abbildung 7 Suchmaske Pflegekompass

Selbstbeschreibungsfähigkeit und Steuerbarkeit:

Das Auswahlfeld „pflegefachlicher Schwerpunkt“ umfasst zwanzig verschiedene Auswahlmöglichkeiten und bietet damit im Vergleich zu den anderen das größte Sortiment. Die angezeigte Maske entspricht der Steuerbarkeit und Selbstbeschreibungsfähigkeit.

Erwartungskonformität:

Der Pflegekompass bietet dem Nutzer an, nach „besonderen Angeboten“ zu suchen, siehe *Abbildung 7 Suchmaske Pflegekompass*. Dabei können mehrere Kriterien ausgewählt werden. In der Ergebnisliste werden Einrichtungen angezeigt, die mindestens eines dieser Kriterien erfüllen (Oder-Verknüpfung). Der Nutzer kann nicht auswählen, ob alle Kriterien erfüllt sein sollen. Dies verletzt die Erwartungskonformität. Zudem fehlt die Möglichkeit Pflegeeinrichtungen direkt vergleichen zu können.

ASB Sozialeinrichtungen GmbH Pflegezentrum "Lupine"

IK: 510202082

Lupinenweg 12
22549 Hamburg
Telefon: 040 / 83398110
Fax: 040 / 83398154
E-Mail: pflgezentrum@asb-hamburg.de
Internetadresse: <http://www.asb-hamburg.de>
[Adresse in Google-Maps anzeigen](#)

Arten der Pflege

- [Kurzzeitpflege](#)
- [Vollstationäre Pflege](#)

Qualitätsberichte

- [Vollstationäre Pflege](#)

Allgemeine Informationen

Pflegefachliche Schwerpunkte

- Multiple Sklerose (MS)

Besondere Angebote

- Garten
- Stadtrandlage
- Tierhaltung
- Gottesdienste
- Terrasse
- Cafe
- Einzelzimmer
- Menuewahl

Anzahl Pflegeplätze

- 26 Kurzzeitpflegeplätze
- 90 vollstationäre Pflegeplätze

1. ASB Sozialeinrichtungen GmbH Pflegezentrum "Lupine"

IK: 510202082

Lupinenweg 12
22549 Hamburg
Telefon: 040 / 83398110
Fax: 040 / 83398154
E-Mail: pflgezentrum@asb-hamburg.de
Internetadresse: <http://www.asb-hamburg.de>
[Adresse in Google-Maps anzeigen](#)

Arten der Pflege

- [Kurzzeitpflege](#)
- [Vollstationäre Pflege](#)

Qualitätsberichte

- [Vollstationäre Pflege](#)

vollstationäre Pflege

Anzahl Pflegeplätze

- 26 Kurzzeitpflegeplätze
- 90 vollstationäre Pflegeplätze

pflegeteilmabhängige Beträge

Anzahltagemenge (in Euro)	Unterhalt (in Euro)	Verpflegung (in Euro)	Investitionskosten (in Euro)
2,27	12,37	10,75	10,00

Übersicht Pflegesätze

Pflegesätze	Pflegesatz	Heimzeit monatlich (in Euro)	Anteil der Pflegekasse (in Euro)
0	22,47	2.036,31	0,00
1	41,11	2.603,34	1.023,00
2	60,55	3.194,71	1.279,00
3	80,13	3.790,33	1.550,00
Häufel	90,49	4.105,48	1.918,00

Abbildung 8 Suchergebnis Pflegekompass

identisch mit der Übersicht. Anstelle der Preisübersicht erhält der Nutzer eine Auflistung der besonderen Angebote. Wählt der Nutzer den Link „vollstationäre Pflege“ unter der Überschrift „Arten der Pflege“ aus, erhält er einen detaillierten Überblick über die Leistungen, Preise, besonderen Leistungen, Qualitätskriterien, Heimvertrag, etc. Die Note des Transparenzberichts wird weder in der Übersicht noch in der Detailansicht direkt angezeigt. Erst nach Herunterladen des Berichts über einen zur Verfügung gestellten Link

Ergebnis der Suche:

Aufgabenangemessenheit:

Das Ergebnis wird in einer Tabelle dargestellt. Diese Liste ist in alphabetischer Reihenfolge sortiert und bietet eine Übersicht über folgende Daten: Kontaktdaten, Pflegearten, Link zum Transparenzbericht mit Note, Anzahl der Pflegeplätze, Preise. Alle Links werden in roter Schrift hervorgehoben. Der Grundsatz Aufgabenangemessenheit ist nicht erfüllt, weil diese Maske keine Sortierung ermöglicht, siehe *Abbildung 8 Suchergebnis Pflegekompass*.

Selbstbeschreibungsfähigkeit:

An mehreren Stellen im Text befindet sich eine rote Schrift, die auf einen Link hinweist. Es geht aus dem Text oder der Mouseoverhilfe nicht eindeutig hervor, worauf diese Links verweisen. Dabei dient z.B. der Name einer Einrichtung als Link zu einer weiteren Übersicht. Der Mouseover-Text enthält den gleichen Text wie der Link. Diese Tatsache entspricht nicht der Selbstbeschreibungsfähigkeit. Es wird zudem nicht erläutert, wonach die Liste sortiert wird.

Steuerbarkeit:

Der Grundsatz ist erfüllt. Der Nutzer kann nach seiner eigenen Wahl entscheiden, wann er was auswählt.

Erwartungskonformität:

Es besteht keine Möglichkeit nach anderen Kriterien zu sortieren, wie z.B. Entfernung zum Standort. Es kann der Titel (Name der Einrichtung) als Link ausgewählt werden, um eine weitere Sicht auf die Einrichtung zu erhalten. Diese Sicht ist bis auf einen Punkt

kann er im Bericht eingesehen werden. Ein direkter Vergleich der Einrichtungen kann somit nicht erfolgen. Alle in diesem Absatz beschriebenen Punkte widersprechen der Erwartungskonformität.

3.5 Der Pflegelotse

Aufgabenangemessenheit und Steuerbarkeit:

Wie die drei anderen Suchmaschinen auch bietet der Pflegelotse der vdek die Möglichkeit nach Standort und Versorgungsform (z.B. vollstationär) zu suchen. Der Nutzer kann zwischen einer Schnellsuche oder geleiteten Suche nach Versorgungsform wählen. Wird die geleitete Suche ausgewählt, werden weitere Auswahlmöglichkeiten zu Verfügung gestellt. Jedoch kann die Suche nicht verfeinert werden. So werden je nach Wahl die mögliche Pflegeart (vollstationäre Pflege, Tagespflege, etc.) und danach der Pflege-schwerpunkt zur Verfügung gestellt. Dies erfüllt die Grundsätze der Aufgabenangemessenheit und der Steuerbarkeit, siehe *Abbildung 9 Suchmaske vdek*.

Selbstbeschreibungsfähigkeit:

Die Selbstbeschreibungsfähigkeit ist erfüllt, weil alle Felder selbstbeschreibende Namen verwenden.

Erwartungskonformität:

Weitere Kriterien zur Einschränkung der Suche werden nicht angeboten. Falls der Nutzer bestimmte Voraussetzungen an die Pflegeeinrichtung erhebt, kann er dies an dieser Stelle nicht angeben. Dies entspricht nicht der Erwartungskonformität.

Abbildung 9 Suchmaske vdek

Ergebnis der Suche:

Aufgabenangemessenheit:

Die Suchergebnisse werden in Tabellenform dargestellt. Dabei werden folgende Informationen angezeigt: Name der Einrichtung, Entfernung, Plätze (Größe der Einrichtung), Pflegenote direkt als Note, PLZ, Telefon, Verbraucherefreundlichkeit und Preise. Zudem bietet die Tabelle die Möglichkeit nach Entfernung, Pflegenote oder Preis zu sortieren, siehe *Abbildung 10 Suchergebnis vdek*.

Selbstbeschreibungsfähigkeit:

Alle Titel und Bezeichner sind so gewählt, dass sie sich selbst beschreiben. Dieser Grundsatz ist erfüllt.

Steuerbarkeit:

Anhand der Plätze und der Pflegenote des Transparenzberichts kann der Nutzer die Einrichtungen direkt vergleichen. Zusätzlich hat er die Möglichkeit sich Einrichtungen zu merken und diese direkt miteinander zu vergleichen. Dies entspricht der Steuerbarkeit, weil nicht alle Informationen sofort angezeigt werden sondern nur die, die benötigt werden.

Trefferliste (Monatspreise in EURO)										
Ver- gleich	Ent- fernung	Name	Plätze	Pflegenote	Verbraucher- freundlichkeit	Ort	Telefon	Pflegestufe I EURO	Pflegestufe II EURO	Pflegestufe III EURO
1	1,0 km	Seniorenwohn- u. Pflegeheim Johann Carl Müller-Stift	24	1,1		22393 Hamburg	040-6010035	2050,61	2627,68	3208,70
2	1,2 km	Parkresidenz Alsterthal-Groß	65	1,0		22391 Hamburg	040/606080	Einrichtung gemäß § 91 SGB XI	Einrichtung gemäß § 91 SGB XI	Einrichtung gemäß § 91 SGB XI
3	1,7 km	Ev.-luth. Kirchengemeinde Alsterthal am Babsthorst	97	1,3		22391 Hamburg	040-5369740	1985,21	2590,26	3199,27
4	2,0 km	Gast- und Krachshaus Alster- und Pfaffenhamm	167	1,1		22391 Hamburg	040/6068940	2016,24	2630,72	3249,46
5	2,0 km	Stiftung Veteranenheim Deutscher Soldaten	144	1,3		22399 Hamburg	040-60664-0	2037,84	2645,93	3257,98
6	2,1 km	Hospital zum Heiligen Geist	855	1,1		22391 Hamburg	040-60601-120	2098,37	2729,89	3365,67
7	2,3 km	Freizeit, Senioren- und Pflegeheim Gama	34	1,2		22395 Hamburg	040/60441110	1945,66	2500,83	3059,95
8	2,4 km	Seniorenresidenz Wellenbühel	26	1,1		22391 Hamburg	040/640909-0	2025,97	2599,39	3177,06
9	2,6 km	ALTESOCHTEL Seniorenpflege Gut Wellenbühel	169	1,0		22391 Hamburg	040/970700	2096,55	2715,29	3338,29
10	2,6 km	Altenheim am Heiligen Geist Besondere Versorgung der Haus Altenheim Senioren- pflegegesellschaft mbH	...	1,0		22391 Hamburg	040/970700	Einrichtung gemäß § 91 SGB XI	Einrichtung gemäß § 91 SGB XI	Einrichtung gemäß § 91 SGB XI

Abbildung 10 Suchergebnis vdek

Erwartungskonformität:

Die Detailsicht einer Einrichtung bietet folgende weitere Informationen: Straße und Hausnummer, Emailadresse, Internetauftritt sowie eine Aufschlüsselung, wie die Pflegeplätze verteilt sind. Dies entspricht nicht der Erwartungskonformität, denn der Nutzer möchte zusätzliche Informationen über die Einrichtung erhalten und nicht nur Adresse und Kontaktdaten. Eine Angabe über zusätzliche Dienstleistungen und besondere Angebote fehlt, obwohl sie in der Suchmaske ausgewählt werden konnten (z.B. Betreuung atmungspflichtiger Menschen). Diese Tatsache entspricht nicht der Erwartung.

3.6 Übersicht des Ergebnisses

Das Ergebnis der Usability-Untersuchung wird in *Tabelle 2 Ergebnisübersicht Usability-Analyse vorhandener Pflegeheimfinder* in übersichtlicher Form dargestellt.

Tabelle 2 Ergebnisübersicht Usability-Analyse vorhandener Pflegeheimfinder

	Pflegenavigator	Pflegefinder	Pflegekompass	Pflegelotse
Selbstbeschreibungsfähigkeit	++	++	-	++
Steuerbarkeit	+	+	+	++
Auswahlmöglichkeiten	+	-	++	-
Sortierung	+	+	-	++
Preisvergleich möglich	-	-	+	+
Notenvergleich möglich	-	-	-	+
Informationsgehalt auf Übersicht	+	-	++	++
Informationsgrad auf Detailansicht	++	-	-	-
Ranking vorhanden	-	-	-	-

Legende: ++ = sehr gute Umsetzung; + = Standardumsetzung;
- = nicht umgesetzt oder unzureichend umgesetzt

Die Tabelle zeigt deutlich die verschiedenen Vor- und Nachteile der einzelnen Pflegeheimfinder. Fast alle erfüllen die Grundlagen der Selbstbeschreibungsfähigkeit und Steuerbarkeit. Bei allen anderen Aspekten scheinen sich Schwerpunkte zu bilden. Während der Pflegekompass und der Pflegelotse einen Preisvergleich anbieten, sticht der Pflegenavigator mit einem hohen Informationsgrad heraus. Nur der Pflegelotse bietet neben dem Preisvergleich auch einen Notenvergleich an. Keiner der Pflegeheimfinder bietet ein Ranking an oder vergleicht einzelne Leistungen miteinander.

4 Soll-Analyse

Dieses Kapitel beschreibt das Analysieren der Anforderungen für die Anwendung. Dabei werden Nutzerworkshops durchgeführt und die Ergebnisse anschließend und währenddessen in die Anforderungsanalyse integriert.

4.1 Vision

In Deutschland angesiedelte Personen mit einem Alter von 45 Jahren und älter bilden die Zielgruppe dieser Anwendung. Es lässt sich keine Aussage darüber treffen, wie viel sie das Internet nutzen oder wie hoch die Berührungängste sind. Dennoch ist die Zielgruppe nicht mit dem Internet aufgewachsen. Der Autor vermutet: je höher das Alter, desto höher die Berührungängste mit dem WWW. Derzeit werden Pflegeplätze hauptsächlich von den Kindern pflegebedürftiger Senioren gesucht. Die Kinder sind meist über 50 Jahre und älter. Der Pflegebutler soll auch einem Personenkreis zugänglich sein, der andere Bedürfnisse an Webseiten hat, als jüngere Personen, die damit vertrauter sind. Generell möchte der Nutzer die besuchte Webseite auf einen Blick überschauen können und die Bedienung sollte sich ohne Mühe erschließen. Bisherige Pflegeheimfinder erfüllen zwar ihre Funktion, orientieren sich aber nicht an den Bedürfnissen der Zielgruppe. Sie stellen dem Nutzer ihre erhobenen Qualitätsdaten in begrenzter Form zur Verfügung. Gelingt es dem Pflegebutler die Bedürfnisse der oben genannten Zielgruppe zu erfüllen, werden auch Experten, die häufig das Internet nutzen und sich auskennen von der leichten Handhabung profitieren.

Die Anwendung dieser Arbeit, der Pflegebutler, soll eine Suchmaschine für Pflegeheime innerhalb Deutschlands werden. Sie findet für den Anwender das passende Pflegeheim, indem es die Eingabe des Anwenders mit den Daten verschiedener Pflegeeinrichtungen vergleicht und daraus eine Rangliste erstellt.

Die Zielgruppe steht dabei im Vordergrund und es wird angenommen, dass sie bestimmte Bedingungen an Pflegeeinrichtungen stellen. Sie möchten nicht nur die reine Unterbringung und Pflegeleistung erwerben. Vielmehr möchten sie sicherstellen, dass die zu pflegende Person sich gut aufgehoben fühlt. Eine rein pflegerische Versorgung wie im Krankenhaus schreckt die Menschen ab, deshalb sollte die Pflegeeinrichtung ein „Zuhause“ bieten. Die Einrichtung muss einen Raum für Privatsphäre liefern, wie z.B. Einzelzimmer. Weiter ist es wichtig, die neue Umgebung mit eigenen Möbeln gemütlich zu gestalten. Dies sind nur zwei der zahlreichen Aspekte, die den Nutzer bei seiner Suche nach einem Pflegeheim bewegen. Bisherige Pflegeheimfinder gehen auf diese Bedürfnisse nicht ein.

Der Pflegebutler soll anders sein und soll den Bedürfnissen der Zielgruppe gerecht werden, indem er dem Nutzer die Möglichkeit bietet, genau die zu ihm passende Pflegeeinrichtung zu finden. Dazu soll die Anwendung neben den pflegerischen Informationen weitere Daten zur Verfügung stellen, die die Zielgruppe interessieren. Der Nutzer soll direkten Einfluss auf das Ergebnis nehmen können, indem er selbst angibt, welche Kriterien ihm wichtig sind.

Als wichtigster Unterschied zu anderen Pflegeheimfindern soll die Bedienbarkeit herausstechen. Dem Nutzer soll die Angst vor der komplexen Bedienung genommen werden, weil der Pflegebutler ihn durch die einzelnen Schritte der Suche führen wird. Der Aufbau soll so übersichtlich und einfach gestaltet werden, dass keine Berührungängste entstehen können.

Anschließend soll das Ergebnis übersichtlich und in einer nie zuvor dagewesenen und intuitiv zu verstehenden Art präsentiert werden. Damit soll der Nutzer sofort und auf den ersten Blick erkennen können, welches Pflegeheim seinen individuellen Bedürfnissen entspricht.

4.2 Überlegungen zur Umsetzung der Vision

Um eine Suche und späteres Anzeigen der Informationen zu gewährleisten, benötigt die im Rahmen dieser Arbeit erstellte Anwendung Daten, mit denen sie arbeiten kann. Dafür bietet sich eine Datenbank an. Es sollen prototypische Daten für das System zur Verfügung gestellt werden, weil eine richtige Datenerhebung den zeitlichen Rahmen dieser Arbeit sprengen würde. Für eine Weiterentwicklung wäre es vorstellbar, Daten aus dem Internet zu suchen oder eine Schnittstelle für Pflegeheime zur Verfügung zu stellen.

Der Anwender soll bei der Eingabe seiner Kriterien gewichten können, ob Preis, Entfernung vom Standort oder Pflegenote für ihn wichtig sind. Die Anwendung soll die eingegebenen Kriterien des Nutzers mit dem Datenstamm der Pflegeheime vergleichen. Aus den vorhandenen Daten wird das System ein Ranking erstellen, welches es dem Nutzer ermöglicht die besten Treffer zu erkennen.

Die gefundenen Pflegeheime sollen in einer Übersicht tabellarisch und grafisch angezeigt werden. Daten dieser Ansicht wären beispielsweise: Name, Standort, Telefon und Pflegenote. Die grafische Darstellung könnte der Darstellung der Uni-Ranking-Suchmaschine von der Seite „die Zeit“ ähneln. Es wird eine Zielscheibe angezeigt und die gefundenen Einrichtungen werden je nach Position in der Rangliste näher zur Mitte oder näher am Rand positioniert. In der tabellarischen Darstellung steht das ranghöchste Ergebnis an erster Stelle und das zweite an zweiter Stelle, usw.

Aus Übersichtsgründen sollen in den Suchergebnissen nur Teilinformationen angezeigt werden, beispielsweise Name, Entfernung vom gesuchten Standort, Telefon, Pflegenote. In der grafischen Darstellung befänden sich diese Daten innerhalb einer Sprechblase und in der tabellarischen als Spalte in der Tabelle.

Durch Anklicken der Sprechblase oder eines Links sollen dem Nutzer die gesamten Daten einer Einrichtung übersichtlich angezeigt werden, z.B. in Form einer Mindmap.

4.3 Zielgruppenanalyse

Wie bereits beschrieben soll der Nutzer im Mittelpunkt der Entwicklung stehen. Dazu wird als erstes analysiert, welche Zielgruppe diese Anwendung vermutlich nutzen wird. Um im späteren Verlauf die Anforderungen an die Zielgruppe anzupassen, wird eine Befragung aus Vertretern der Zielgruppe stattfinden. Aus der Ist-Analyse und dem Zielgruppenworkshop ergeben sich anschließend die Anforderungen an diese Software.

Die Anwendung soll Personen unterstützen, die ein Pflegeheim suchen. Häufig wird eine Einrichtung von den Kindern der zu pflegenden Person gesucht. Es ist anzunehmen, dass „der Altersdurchschnitt für Frauen im Heim bei 85 Jahren und für Männer bei 75 Jahren liegt und [...] die Verweildauer schätzungsweise bei 30 bis 40 Monaten liegt“ (Winter, 2008). Damit

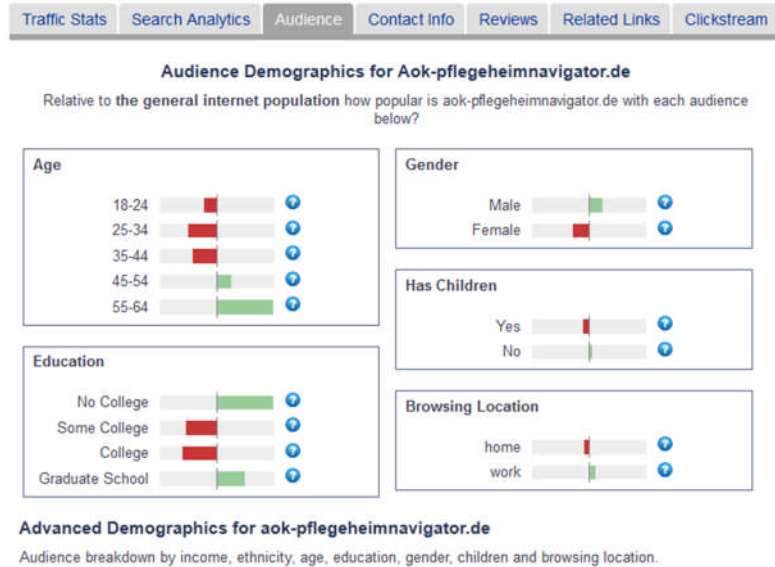


Abbildung 11 Webstatistik AOK-Pflegenavigator

ergibt sich ein Durchschnittsalter bei Heimaufnahme für Frauen zwischen 81 und 82 und für Männer zwischen 71 und 72 Jahren. Weiter wird angenommen, dass diese Personen ihre Kinder zwischen dem 20. und 30. Lebensjahr geboren haben. Somit liegt das Alter der Zielgruppe zwischen 41 und 62 Jahren.

Diese Annahme wird unterstützt durch die Statistik der Alexa Web Information Company (Alexa, 2012), siehe *Abbildung 11 Webstatistik AOK-Pflegenavigator*. Auch wenn diese Information für sich genommen nicht ausreicht, um die Zielgruppe zu analysieren, gibt sie einen deutlichen Hinweis auf das Alter der Besucher von der AOK-Pflegeheimnavigator Webseite. Aus dieser Statistik geht eindeutig hervor, dass Besucher mit einem Durchschnittsalter ab 45 Jahren zahlreicher vertreten sind als üblich. Im Gegenzug dazu sind die Altersgruppen unter 44 deutlich weniger vertreten (siehe *Abbildung 11 Webstatistik AOK-Pflegenavigator*). 100% der Nutzer dieser Seite stammen aus Deutschland.

Zum besseren Verständnis der Zielgruppe wird eine Typologie erstellt. Laut der Web-Typologie des Instituts für Medien- und Konsumentenforschung (IMUK) gehört die Zielgruppe zum „Web-Mainstream und macht einen Anteil von 42% der Gesamtnutzer aus.

Sie sind eher weiblich und älter, besitzen eine niedrige bis mittlere Bildung, nutzen selten das Internet, angepasst“ (Institut für Medien- und Konsumentenforschung, 2012).

Die Informationen beider Quellen ergeben zusammengefasst folgendes Benutzerprofil:

- männliche und weibliche Nutzer
- Alter liegt zwischen 41 und 62 Jahren
- Nutzen selten das Internet
- Wenig Vorkenntnisse im Umgang mit dem Internet
- Niedrige bis mittlere Bildung
- Deutsch als Muttersprache
- Rufen die Webseite von der Arbeit oder von zu Hause aus auf
- Nutzen kaum mobile Geräte zum Internetsurfen
- Haben Kinder

Ein Teil der Zielgruppe wird als ‚Silversurfer‘ bezeichnet. Damit werden Internetnutzer im Alter von 50 Jahren aufwärts bezeichnet. Es können „generationsbedingte Barrieren wie Jargon (z.B. FAQ, Sitemap, About Us, Submit) [...]“ auftreten, sowie weitere Schwierigkeiten wie etwa „unbeabsichtigt auf Seiten zurückkehren, die sie schon gesehen hatten. „Wo bin ich jetzt?“ und „Wo war ich schon?“ sind Fragen, die man ohne unterstützende Anhaltspunkte nur dann beantworten kann, wenn man ein gutes Kurzzeitgedächtnis hat. [...] Häufig klickten sie auf den falschen von mehreren eng nebeneinander liegenden Links. [...] Ältere Menschen hatten motorische Schwierigkeiten, Pull-down-Menüs zu bedienen oder Linkhierarchien zu verfolgen, die nur mit gedrückter Maustaste und gerader, zügiger Handbewegung gesteuert werden können.“ (Schulz, 2004)

Bei der Entwicklung der Anwendung sollten diese Erkenntnisse über die Zielgruppe einfließen.

4.4 Nutzerworkshop

Im Rahmen dieser Arbeit ist nicht genügend Zeit, um zahlreiche, aufwändige oder langfristige Tests durchzuführen. Aus diesem Grund soll es stellvertretend zwei Nutzerworkshops mit einer kleinen Teilnehmerzahl geben, mit deren Durchführung sich dieses Kapitel beschäftigt.

4.4.1 Vorbereitung des ersten Nutzerworkshops

Nachdem die Zielgruppe identifiziert ist, sollen nun auch Vertreter der Zielgruppe am Entwicklungsprozess teilhaben. Der Workshop hat das Ziel, die Bedürfnisse an eine Pflegeeinrichtung zu identifizieren und eine Idee für eine mögliche Oberfläche zu entwickeln. Was erwartet der Nutzer von einer Suchmaschine für Pflegeeinrichtungen?

Welche Informationen muss eine Suchmaschine zur Verfügung stellen? Wie sollte eine gut nutzbare Oberfläche aussehen?

Um den Workshop effektiv zu gestalten sollten maximal fünf Personen teilnehmen. Bei höherer Teilnehmerzahl wird der organisatorische Aufwand so hoch, dass für Diskussionen und Teamarbeit weniger Zeit zur Verfügung steht. Probanden zu finden, die gerade im Internet eine Pflegeeinrichtung suchen, ist schwer. Welche Informationen benötigen Angehörige auf der Suche nach einem Pflegeplatz? Am ehesten kann diese Frage von Personen beantwortet werden, die täglich mit Menschen in Kontakt kommen, welche auf der Suche nach einer Pflegeeinrichtung sind. Aus diesem Grund wendet sich der Autor an das Personal mehrerer Pflegeeinrichtungen, die den Erstkontakt zum Kunden herstellen und bittet um ihre Teilnahme am Workshop. Neben drei Experten aus der Pflege findet der Autor auch eine fachfremde Person aus der Zielgruppe, die sich bereit erklärt am Workshop teilzunehmen.

Die Teilnehmer sind aufgefordert, die ihnen gestellten Fragen kreativ und offen zu beantworten. Es soll nicht vorweg genommen werden, welche Möglichkeiten bei der technischen Umsetzung bestehen und welche nicht. Um dies zu erreichen werden die Frage an die Probanden mit offenen W-Fragen gestellt und nicht mit Ja/Nein-Fragen oder Multiple-Choice-Bausteinen. Die folgenden Fragen scheinen geeignet, das Ziel dieses Workshops zu erreichen:

- Frage 1: Wenn ich ein Pflegeheim suche, welche Eigenschaften sollte es erfüllen, damit ich es mit meinen Eltern anschauen?
- Frage 2: Wie sollte die Suchmöglichkeit aufgebaut sein?
- Frage 3: Wie sollte das Ergebnis auf dem Bildschirm präsentiert werden?
- Frage 4: Wie sollten die Anzahl der Übereinstimmungen angezeigt werden?

Der Verfasser dieser Arbeit übernimmt die Rolle des Moderators. Die Teilnehmer sollen möglichst ihre eigenen Erfahrungen und Ideen vorstellen, ohne von anderen Teilnehmern beeinflusst zu werden. Jeder der Teilnehmer soll beteiligt sein. Die folgende Technik ermöglicht diese Zielerreichung.

Die Teilnehmer schreiben stichwortartig ihre Antworten auf eine Karte. Auf jeder Karte steht eine Antwort. Für die Beantwortung jeder Frage werden fünf Minuten eingeplant. Im Anschluss werden die Karten vom Moderator eingesammelt und an eine Moderationswand gepinnt. In dieser Phase wird vom Moderator eine Diskussion zu jeder Antwort angeregt. Sollten sehr viele Antworten zusammen kommen, werden die Teilnehmer gebeten, die Antworten zu bewerten. Dazu wird abhängig von der Anzahl der Antworten eine gewisse Anzahl an Klebepunkten verteilt. Die Teilnehmer dürfen diese dann an die in ihren Augen wichtigsten Beiträge kleben. Insgesamt wird eine Länge von 1,5 Std. für die Durchführung des Workshops veranschlagt.

4.4.2 Durchführung und Ergebnis des ersten Nutzerworkshops

Mit vier Vertretern der Zielgruppe wurde der Workshop durchgeführt. Er bestand aus zwei männlichen und zwei weiblichen Nutzern. Der Verfasser dieser Arbeit übernahm wie geplant die Moderation. Auch die Durchführung verlief wie geplant. Die ersten Antworten wurden nur kurz und knapp diskutiert, ab der sechsten Antwort entstanden längere Debatten. Die erste Frage ergab viele Antworten, deshalb wurden vom Moderator fünf Klebepunkte an die Teilnehmer verteilt. Die folgenden Ergebnisse dienen als Brainstorming für den weiteren Entwicklungsprozess.

Ergebnis:

Frage 1 Wenn ich ein Pflegeheim suche, welche Eigenschaften sollte es erfüllen, damit ich es mit meinen Eltern anschauen?:

- **Lage (in der Nähe meines Wohnorts)** **4P**
- Sexuelle Ausrichtung (Einrichtung nur für Homosexuelle)
- **Verkehrsanbindung / öffentlich oder Fahrdienst (in die Stadt / Einkaufsmöglichkeiten)** **1P**
- **Qualifikation der Mitarbeiter** **2P**
- **Pflegesätze (Preise)** **4P**
- **Beschäftigungsangebote** **3P**
- Probewohnen möglich
- Ärztliche Betreuung (in der Nähe / Hausbesuche von Fachärzten)
- **Tierhaltung** **1P**
- Internetzugang
- Café für Besucher
- Fitness- oder Sportraum / Wellnessbereich
- Auslegung des Begriffs „Selbstbestimmung“
- Bezugspflegerkraft vorab kennen lernen (eine feste verantwortliche Pflegeperson)
- **Eigene Möbel / Bettwäsche mitbringen können** **1P**
- **Ausstattung der Zimmer (Dusche & Bad im Zimmer)** **1P**
- MDK-Note
- Übernachtungsmöglichkeiten für Angehörige
- Bettnerer möglich (demenziell Erkrankte dürfen zusammen im Bett schlafen)
- Moderne Ausstattung (keine Jagdszenebilder an der Wand, etc.)
- **Größe des Hauses (Anzahl Pflegeplätze)** **2P**
- Angebotene Versorgungsform
- Ein-/ Mehrbettzimmer
- Nachtwache vor Ort
- **Hauseigene Küche** **1P**

Frage 2 Wie sollte die Suchmöglichkeit aufgebaut sein?:

- Keine offenen Fragen (es soll schnell gehen)
- Eigenschaften (der Einrichtung) per Kästchen auswählen
- Feld zum Eingeben, wo die Einrichtung ist, mit Entfernungsangabe
- Fachrichtung soll auswählbar sein
(z.B. Versorgung von demenziell erkrankten Menschen)

Frage 3 Wie sollte das Ergebnis auf dem Bildschirm präsentiert werden?

- Die Einrichtung in der Mitte und um sie herum die einzelnen Kriterien wie bei einer Mindmap
- Wie bei Immobilienscout (Erst eine Liste mit allen verfügbaren Einrichtungen im Umkreis und danach mithilfe einzelner Kriterien das Ergebnis immer weiter eingrenzen)
- Sortiert in der Übersicht (Preis, Übereinstimmung, Lage)
- In der Übersicht anzeigen, was diese Einrichtung von allen anderen unterscheidet
- Bild der Einrichtung
- Kontaktaufnahme in der Übersicht (muss schnell gehen)
- Diashow
- Hilfestellung vom System (Beratungszettel ausdrucken lassen)

Frage 4 Wie sollten die Anzahl der Übereinstimmungen angezeigt werden?

- Ein Regler der von 0% bis 100% anzeigt
- Ein Ampelsystem von grün, gelb und rot
- Ein Balken der die die Übereinstimmungen in Prozent anzeigt und es farblich unterstützt mit rot, gelb und grün (wie bei Passwort)

4.4.3 Vorbereitung des zweiten Nutzerworkshops

Im zweiten Nutzerworkshop soll getestet werden, ob die Probanden mit der Oberfläche ohne Schwierigkeiten interagieren können. Weiter soll überprüft werden, ob eine alternative grafische Darstellung der Suchergebnisse einer tabellarischen Übersicht vorzuziehen ist. Es werden zwei Tests durchgeführt, eine Aufmerksamkeitsanalyse und ein A/B-Test. Den Probanden wird ein Prototyp der Webseite vorgeführt und sie sollen den gesamten Suchprozess einmal durchlaufen. Beim ersten Durchlauf wird das Ergebnis klassisch in einer Tabelle dargestellt. Dabei notiert sich der Moderator die Start- und Endzeit, um feststellen zu können, wie lange eine Suche ungefähr dauert. Anschließend sollen die Probanden erneut den Suchprozess durchlaufen. Beim zweiten Durchlauf wird das Ergebnis grafisch dargestellt. Im Anschluss sollen die Nutzer mitteilen, welche Darstellung sie als angenehmer empfanden. Die Aussagen der Probanden werden vom Moderator zur späteren Auswertung notiert.

Für jeden Test werden 15 Minuten eingeplant, sprich eine Stunde für die gesamte Durchführung.

4.4.4 Durchführung und Ergebnis des zweiten Nutzerworkshops

Da die Probanden aus dem ersten Workshop mit dem Thema bereits vertraut sind und sie eine gewisse Vorstellung von der Oberfläche entwickelt haben, wurden sie auch zum zweiten Workshop eingeladen. Um es ihnen so einfach und angenehm wie möglich zu gestalten findet der Workshop nicht im Labor sondern in der jeweiligen Pflegeeinrichtung statt. Zur Durchführung wird der Laptop des Autors genutzt.

Ergebnis:

1 Person hat abgesagt.

Erste Testperson:

- Hat ca. 2 Minuten bis zum Ergebnis gebraucht
- Aussagen:
 - o schön einfach.
 - o Ampelsystem gut.
 - o „Kritzeleien sehen schön aus“.
 - o Keine Fragen.
 - o Möchte nicht in eine so große Einrichtung. Wäre schön, wenn man die Größe auf den ersten Blick sieht.
 - o In die Auswahlliste „eigener Waschraum für Wäsche“ aufnehmen.

Zweite Testperson:

- Hat ca. 1,5 Minuten bis zum Ergebnis gebraucht
- Aussagen:
 - In der Ergebnisübersicht Bild der Einrichtung oder Größe anzeigen lassen, damit ich gleich weiß wie groß die Einrichtung ist.
 - Auf der Suchseite gleich Prioritäten auswählen können, wie günstiger Preis und Einzelzimmer.
 - Für das Ampelsystem zusätzlich Smileys verwenden.
 - Grafische Ansicht unübersichtlich.

Dritte Testperson:

- Hat ca. 3 Minuten bis zum Ergebnis benötigt
- Aussagen:
 - Auf der ersten Seite Prioritäten setzen können. Auswahl sollte sein: günstiger Preis, in der Nähe, Einzelzimmer.
 - Grafische Ansicht sieht eher nach Stadtplan aus.

Eigene Beobachtung:

- Nur 1 Proband von 3 hat in der Detailansicht nach unten gescrollt.
- Alle Probanden fanden die Tabellenübersicht intuitiver und übersichtlicher.

4.5 Anforderungsanalyse

Aus der Ist-Analyse und dem Zielgruppenworkshop ergeben sich die folgenden Anforderungen an diese Software.

4.5.1 Funktionale Anforderungen

A1: Das System sucht aufgrund der Nutzerkriterien passende Einrichtungen

Das System analysiert die Nutzereingaben und sucht die passenden Pflegeeinrichtungen heraus. Besonders wichtig ist hierbei, dass nur Einrichtungen in die Auswahl aufgenommen werden, die im passenden Umkreis der Suche liegen, die Voraussetzung des Pflegeschwerpunktes erfüllen und die passende Pflegeart anbieten. Dieses Ergebnis wird dem Nutzer angezeigt. Die folgenden Daten müssen vom Nutzer eingegeben werden

- Standort
 - o PLZ oder Stadt, Umkreis der Suche
- Versorgungsstruktur
 - o Vollstationäre Pflege, Kurzzeitpflege
 - o Kein Pflegeschwerpunkt, Pflege demenzerkrankter Menschen, Pflege beatmungspflichtiger Menschen, Pflege von Menschen im Wachkoma

Damit der Nutzer das Ranking beeinflussen kann, soll eine Gewichtung von vier Bereichen möglich sein. So kann der Nutzer entscheiden, ob der Preis, die Entfernung zum Standort, die Pflegenote oder viel Leistung in die Bewertung einfließen sollen.

Falls die Pflichtfelder nicht ausgefüllt werden, wird das System eine Fehlermeldung mit einem Hinweistext ausgeben, welche Felder noch auszufüllen sind.

A2: Der Nutzer kann zusätzliche Profilkriterien auswählen, die das System für eine Bewertung heranzieht.

Der Nutzer kann folgende Daten für die Suche eingeben:

- Ernährungsstruktur
 - o Getränkeangebot, hauseigene Küche, Menüwahl, Diätspeisen, vegetarische Speisen, Schonkost, pürierte Kost
- Sächliche Ausstattung
 - o Heimgröße, freie Plätze, Zimmergröße, Ein- oder Mehrbettzimmer, Telefon auf dem Zimmer, Internet auf dem Zimmer, Sauna, Schwimmbad, Fernsehen auf dem Zimmer, Gartenanlage, Ladengeschäft auf dem Gelände, Bibliothek, Frisör, Dusche auf dem Zimmer

- Heimentgelt
 - o Monatliche und tägliche Kosten der Pflegestufe 0, I, II und III

- Leistungen und Pflegequalität
 - o Pflegenote und MDK-Transparenzbericht
 - o Tagesgestaltung, Beschäftigungstherapie, Ausflüge, Veranstaltung von Festen, Betreuung nach § 87b SGB XI
 - o Einsatz von Expertenstandards
 - o Anbindung ans öffentliche Verkehrsnetz in Minuten, Fahrdienst
 - o Gottesdienste
 - o Tierhaltung
 - o Eigene Möbel
 - o Ausgebildetes Personal

Das System nimmt eine Einschränkung aufgrund der ausgewählten Profilkriterien vor. Sollte dabei das Ergebnis auf 0 Einrichtungen sinken, wird das System die Einrichtungen nach Grad der Übereinstimmung sortieren. Das System soll einen Hilfetext darstellen, der den Nutzer darüber informiert, dass es keine Einrichtung gibt, die alle Kriterien erfüllt und dass stattdessen die Ergebnisse mit den höchsten Übereinstimmungen angezeigt werden. Falls der Nutzer keine Angaben zu diesen Profilkriterien macht, wird das System je nach eingegebener Gewichtung des Nutzers, siehe A5, die Profilkriterien ignorieren oder auswerten.

A3: gefundene Einrichtungen bewerten und in eine Rangliste einordnen

Jede Einrichtung im Suchergebnis wird vom System bewertet. Anhand der Nutzereingaben und den Einrichtungsdaten wird eine prozentuale Treffergenauigkeit errechnet. Die bewerteten Einrichtungen werden anhand der Treffergenauigkeit in eine Rangliste einsortiert.

A4: Das Ergebnis der Suche mithilfe einer Grafik darstellen

Das System stellt das Suchergebnis grafisch dar, beispielsweise in Form einer Mindmap. In dieser Übersicht sollen folgende Daten angegeben werden:

- Name der Einrichtung
- Kontaktdaten
- Preis
- Größe der Einrichtung
- Entfernung zum gesuchten Standort
- Übereinstimmungsgrad mit dem gesuchten Kriterien als Ampelsystem

A5: Nutzer kann Rankingverfahren mit seiner Eingabe beeinflussen

Der Nutzer kann zwischen Preis, Entfernung zum Standort, Übereinstimmung zu Suchkriterien und Pflegenote seine eigene Gewichtung für das System angeben. Das System berücksichtigt die Gewichtung des Benutzers in seiner Berechnung der Rangordnung.

A6: Alle Daten einer Einrichtung in der Detailsicht anzeigen

Das System zeigt dem Nutzer detaillierte Daten einer Einrichtung an, wenn er eine Einrichtung auswählt.

4.5.2 Nicht-Funktionale Anforderungen

NF1: Fehlertoleranz

Der Nutzer erhält Hinweise auf fehlerhafte Eingaben und falls das Suchergebnis keinen Treffer ergibt.

NF2: Benutzbarkeit

Verständlichkeit:

Es wird davon ausgegangen, dass der Standardnutzer dieser Anwendung weiß, wozu Pflegeheime dienen und welche Leistungen sie erbringen. Die Anwendung soll dem Nutzer mithilfe von Hilfetexten die Funktion der Bewertung verständlich machen.

Erlernbarkeit:

Der Nutzer sollte beim Lesen der Bezeichner verstehen, was er in die entsprechenden Felder eingeben soll. Das System soll Hilfestellung anhand von Mouseovertexten geben. Der Nutzer soll anhand der Oberfläche und Hilfetexte die Bedienung der Software erlernen können. Handbücher oder eine zusätzliche Hilfsfunktion sollen nicht nötig sein. Mithilfe eines Mock-Ups soll ermittelt werden, ob die Teilnehmer aus dem vorangegangenen Workshop Fragen zur Bedienung haben. Das System soll keine Fragen offen lassen.

Bedienbarkeit:

Der Nutzer soll geschätzte 30 bis 60 Sekunden benötigen, um ein passendes Pflegeheim zu finden. Dieser Wert soll anhand des Mock-Ups und der Workshop Teilnehmer verifiziert werden.

Konformität:

Die restlichen Eingaben werden per Auswahlfelder zur Verfügung gestellt. Die Anwendung soll der ISO 9241-10 genügen.

NF3: Effizienz

Das System soll nicht länger für die Suche benötigen als die untersuchten Pflegeheimfinder. Daraus ergibt sich, dass bei einer Auswahl von ca. 30.000 Pflegeheimen die Antwortzeit

unter 2 Sekunden liegen soll, sowohl für die Suchanfrage, als auch bei der Filterungs- und Bewertungsanfrage.

NF4: Nutzer sollen das System ohne Installation nutzen können

Für eine bequeme Nutzung des Systems soll der Nutzer die Funktionen sofort verwenden können, ohne es installieren zu müssen.

4.5.3 Ausgrenzung

N1: Benutzerschnittstelle für Pflegeheime

Aus zeitlichen Gründen wird die Anwendung mit fiktiven Daten ausgestattet. Deshalb entfällt eine Schnittstelle für Pflegeheime.

N2: Sicherheit

Es ist nicht geplant eine Schnittstelle nach außen anzubieten, deshalb wird auf ein Benutzer- und Berechtigungssystem verzichtet.

N3: Übertragbarkeit

Es ist nicht geplant, die Anwendung auf ein anderes System zu übertragen.

N4: Einrichtungsdaten werden nicht überprüft

Es wird nicht überprüft, ob die eingepflegten Daten der Wahrheit entsprechen.

4.5.4 Anwendungsfälle

Aus den oben genannten Anforderungen lassen sich mehrere Anwendungsfälle erschließen. Eine Übersicht der abgeleiteten Anwendungsfälle befindet sich im Anwendungsfalldiagramm, siehe *Abbildung 12 Anwendungsfalldiagramm*. In den folgenden Kapiteln werden die Anwendungsfälle einzeln näher erläutert.

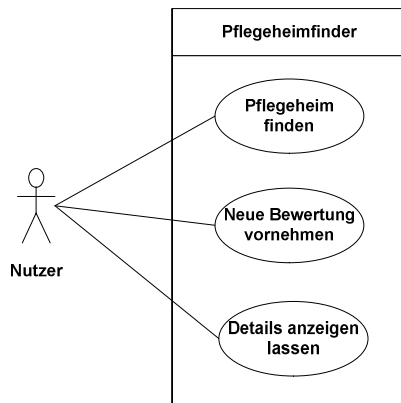


Abbildung 12 Anwendungsfalldiagramm

4.5.5 Anwendungsfall Pflegeheim finden

Titel: Pflegeheim finden
Akteur: Nutzer
Ziel: Der Nutzer erhält eine Übersicht von Pflegeheimen, die in das gesuchte Profil passen.

Auslöser:

Der Nutzer möchte eine Pflegeeinrichtung in Wohnort-Nähe finden, die im Prüfbericht des MDK mit einer Pflegenote von mindestens 1,5 abgeschlossen hat.

Vorbedingungen:

Der Nutzer muss über einen Internetzugang verfügen.

Nachbedingungen:

Dem Nutzer werden ins Profil passende Pflegeheime übersichtlich angezeigt.

Erfolgsszenario:

1. Nutzer ruft die Oberfläche des Pflegeheimfinders auf.
2. Nutzer füllt das Formular für die Kriterien des Pflegeheims aus.
3. Nutzer drückt den Knopf „Suche starten“.

4. Das System erstellt anhand der folgenden Kriterien ein Ranking:
 - Versorgungsform „Vollstationär“;
 - Eingegebene PLZ 22393 und Umkreis von 10 km;
 - Gewichtung liegt auf der Pflegenote.
5. Das System erstellt eine Gewichtung der ins Profil passenden Pflegeheime wie folgt:
 - Das Pflegeheim mit niedrigster Note und mit geringster Entfernung erhält den höchsten Rang.
 - Alle anderen werden in absteigender Reihenfolge einsortiert.
 - Pflegeheime außerhalb des eingegebenen Standorts werden nicht angezeigt;
 - Pflegeheime einer anderen Versorgungsform als gewünscht werden nicht angezeigt.
6. Das System stellt die Ergebnisse grafisch dar.

Erweiterungen:

- 5.a Das System findet kein Pflegeheim, das in das Profil passt.
Das System meldet „0 Pflegeheime gefunden, die in das Profil passen. Bitte ändern Sie Ihre Auswahlkriterien.“

Fehlerfälle:

- 3.a Der Nutzer gibt eine ungültige PLZ oder Ort ein. System meldet einen Eingabefehler und fordert zur Neueingabe auf.
- 3.b Pflichtfeld wurde nicht ausgefüllt. Das System meldet fehlende Eingabe und fordert zur Neueingabe auf.

Anforderungen:

A1 – A6

4.5.6 Anwendungsfall neue Bewertung vornehmen

Titel: neue Bewertung vornehmen
Akteur: Nutzer
Ziel: Das System sortiert die gefundenen Pflegeheime mithilfe einer neuen Bewertung um und zeigt diese an. Die Kriterien zur Bewertung werden vom Nutzer vorher festgelegt.

Auslöser:
Der Nutzer möchte die Suchergebnisse eingrenzen oder neu bewerten. Dazu klickt er die Kriterien an, die ein Pflegeheim erfüllen soll.

Vorbedingungen:
Der Nutzer ist erfolgreich auf die Seite mit der Trefferübersicht navigiert. Das Formular wird korrekt im Browser angezeigt.

Nachbedingungen:
Die Übersicht zeigt das Suchergebnis nach der neuen Sortierung an.

Erfolgsszenario:

1. Nutzer wählt folgende Dienstleistungen und Kriterien aus, die das Pflegeheim erfüllen soll:
 - Diätspeisen
 - Heimentgelt Pflegestufe 1 bis 2.200,-€
 - Telefon auf Zimmer
 - Fernsehen auf Zimmer
 - Frisör
 - Fahrdienst
 - gute PflegenoteAnschließend klickt er auf den Knopf „Ergebnis berechnen“.
2. Das System durchsucht das Suchergebnis und filtert alle Einrichtungen heraus, die die angegebenen Dienstleistungen nicht anbieten.
3. Das System erstellt eine Reihenfolge des neuen Suchergebnisses basierend auf der Pflegenote.
4. Das System zeigt das Ergebnis an.

Erweiterungen:
2.a Sollte das Ergebnis auf 0 Treffer sinken, stellt das System eine Rangliste auf. Die Rangliste wird nach der Anzahl der Übereinstimmungen sortiert.

Anforderungen:
A6

4.5.7 Anwendungsfall Details anzeigen lassen

Titel: Details der Einrichtung vom System anzeigen lassen

Akteur: Nutzer

Ziel: Vom Nutzer ausgewählte Einrichtung wird mit allen vorhandenen Daten angezeigt.

Auslöser:

Der Nutzer klickt auf eine Einrichtung, die er sich genauer ansehen möchte.

Vorbedingungen:

Der Nutzer ist erfolgreich auf die Seite mit der Trefferübersicht navigiert. Die Trefferübersicht wird korrekt im Browser angezeigt.

Nachbedingungen:

Eine neue Seite mit allen Informationen über die ausgewählte Einrichtung wird angezeigt.

Erfolgsszenario:

1. Nutzer klickt auf die Einrichtung, die er genauer betrachten möchte.
2. Das System sucht aus der Datenbank alle Angaben der gewünschten Einrichtung und zeigt diese im Browser an.

Anforderungen:

A2 – A5

4.6 Auswertung der Daten durch das System

Die ausgewerteten Einrichtungen sollen dem Nutzer übersichtlich angezeigt werden. Würde das System ca. „29034“ (Knappschaft, kein Datum) Einrichtungen vergleichen, wäre diese Datenmenge für einen Menschen kaum überschaubar. Deshalb soll das System eine Vorauswahl treffen. Es werden nur Einrichtungen in das Ranking aufgenommen, die folgende Bedingungen erfüllen:

- Einrichtung liegt im gewünschten Umkreis vom gesuchten Standort
- Pflegeschwerpunkt stimmt mit Eingabe überein
- Versorgungsstruktur stimmt mit Eingabe überein

Der Nutzer sucht gezielt Pflegeeinrichtungen, die sich in einem bestimmten Umkreis vom eingegeben Standort befinden, weil sie entweder ihre Angehörigen besuchen möchten oder weil sie selber in die Einrichtung einziehen möchten und genau an diesem Ort leben wollen. Deshalb kommen Pflegeheime, die weiter entfernt liegen, nicht in Betracht.

Ähnlich verhält es sich mit der Versorgungsstruktur und dem Pflegeschwerpunkt. Würden alle Einrichtungen in das Ranking aufgenommen, dann könnten im schlimmsten Fall nur Einrichtungen angezeigt werden, die diese Bedingungen nicht erfüllen. Falls der Benutzer eine vollstationäre Unterbringung benötigt, die beatmungspflichtige Menschen betreut, hat er keinen Nutzen von Einrichtungen, die nur Kurzzeitpflege von demenziell erkrankten Menschen anbieten.

Das System führt eine Auswertung durch, indem es einen Soll-Ist-Abgleich berechnet. Die Sollwerte entstehen durch die Benutzereingabe oder eine natürliche Obergrenze. Für die Kategorien Entfernung und Kosten werden die Maximalwerte aus der Nutzereingabe genutzt. Falls der Maximalwert für Kosten nicht eingegeben wurde, wird der höchste Preis aus dem bisherigen Ergebnis als Sollwert genommen. Für die Kategorie Pflegenote steht die natürliche Obergrenze von eins fest. Für die Kategorie Profilkriterien ergab die Anforderungsanalyse eine Obergrenze von 32.

Der Ist-Wert ergibt sich aus den Daten der Einrichtungen. Das System errechnet für jede Einrichtung in der Ergebnisauswahl die jeweiligen Abweichungen vom Soll-Wert. Die dabei verwendete Formel lautet: $G = \text{Soll-Wert}$, $P = \text{Ist-Wert}$. $|G-P|/G*100$.

Der Vergleichswert für das Ranking bildet sich aus dem Durchschnitt aller berechneten Ergebnisse. Anhand dieses Prozentwerts kann die Rangliste sortiert werden.

5 Systemarchitektur

Dieses Kapitel erläutert die Architektur dieser Anwendung. Es werden die fachlichen, technischen Komponenten und das Datenbankmodell dargestellt.

5.1 Fachliches Datenmodell

In

Abbildung 13 fachliches Datenmodell sind alle fachlichen Daten und deren Zusammenhänge aufgeführt. Die Einrichtung steht dabei im Mittelpunkt. Die Kosten einer Einrichtung sind abhängig von der Pflegestufe und der Dauer des Aufenthalts. Falls eine Kurzzeitpflege gesucht wird, sind die täglichen Kosten von Interesse, bei einem vollstationären Aufenthalt hingegen die monatlichen Kosten.

Um eine Umkreissuche zu realisieren benötigt der Standort einer Einrichtung neben einem Namen auch Längen- und Breitengradangaben, wobei die Postleitzahlen auf eine Stadt referenzieren.

Jede Einrichtung besitzt unterschiedliche Eigenschaften in den Bereichen Versorgungsstruktur, Ernährungsstruktur, Ausstattung, Leistung und Pflegequalität. Jeder Bereich kann dabei den Wert annehmen, der unter Punkt 4.5.1 A1 & A2 beschrieben wurde.

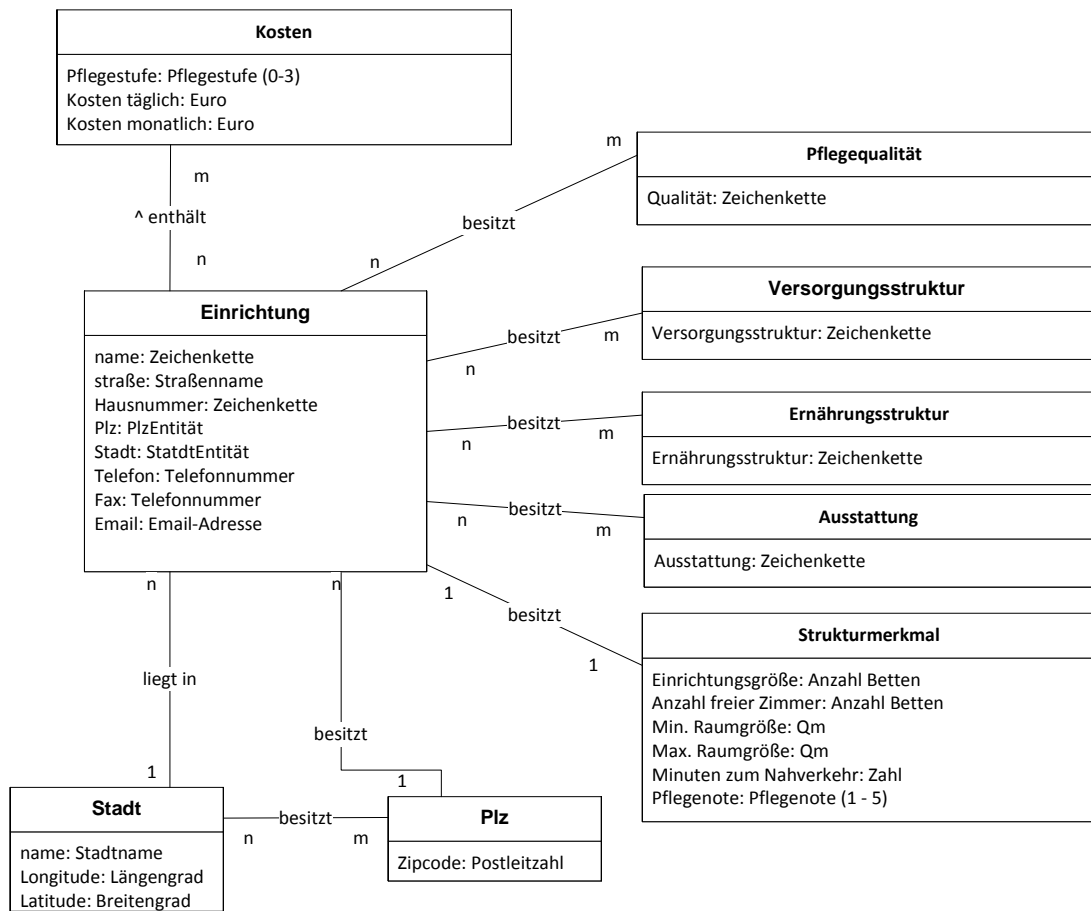


Abbildung 13 fachliches Datenmodell

5.2 Fachliches Komponentendiagramm

In *Abbildung 14 fachliches Komponentendiagramm* sind die fachlichen Komponenten der Web-Anwendung aufgeführt.

<<findeEinrichtungen>>

Diese Komponente ist dafür zuständig die Anforderungen aus 4.5.1 A1 umzusetzen, also das Finden von Einrichtungen innerhalb eines Umkreises und das Einschränken nach der Versorgungsstruktur. Um die Einrichtungen zu bewerten, wird die bewerteEinrichtungen-Komponente genutzt.

<<bewerteEinrichtungen>>

Zur Aufgabe dieser Komponente gehört das Erstellen einer Rangliste mithilfe einer prozentualen Einschätzung, siehe 4.5.1 A3 & A5 und 4.6.

Eine zusätzliche Aufgabe dieser Komponente besteht darin, das Suchergebnis nach Angaben des Nutzers zu filtern, siehe 4.5.1 A2.

<<zeigeEinrichtungAn>>

Falls der Nutzer mehr Informationen über eine Einrichtung anzeigen lassen möchte, siehe 4.5.1 A6, ist diese Komponente für das Suchen innerhalb der Datenbank zuständig.



Abbildung 14 fachliches Komponentendiagramm

5.3 Verwendete Frameworks

Zum Einsatz kommt das Java/Scala Play!-Webframework 2.0.3, um die Implementation dieser Anwendung zu verkürzen. „Play is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption (CPU, memory, threads) for highly-scalable applications thanks to its reactive model, based on Iteratee IO.“ (play!-Framework documentation, unbekannt) Dieses MVC-Framework unterstützt die Entwicklung, indem es viele bewährte Java-Bibliotheken mitliefert. Diese werden über die Play!-Fassade genutzt, wie z.B. Hibernate, Ebeans, Quartz. Play! liefert für die View-Komponente einige Tags mit, die HTML-Konstrukte erzeugen oder eingestreuten Scala-Code ausführen. Der Play!-Dispatcher ist verantwortlich für die Routing-Regeln und leitet Requests automatisch zum passenden Controller.

Für die Implementation wird Java verwendet. Die Entscheidung ist auf Play! gefallen, weil es ein Java-basiertes Web-Framework ist. Während des Studiums hat der Autor erste

Erfahrungen mit dem Framework gesammelt, die er bei dieser Gelegenheit vertiefen möchte.

Die Darstellung einer Web-Oberfläche kann unterschiedlich ausfallen, je nach verwendetem Browser und Endgerät. Um die Darstellung möglichst einheitlich zu gestalten ist die Wahl auf das Twitter-Bootstrap Framework gefallen. So wird die Arbeit für die beste Darstellung dem Twitter-Bootstrap Framework überlassen. „Bootstrap comes equipped with HTML, CSS, and JS for all sorts of things, but they can be summarized with a handful of categories visible at the top of the Bootstrap documentation.“ (@fat, unbekannt)

Der Autor hat sich für Bootstrap entschieden, weil es Funktionen zur Verfügung stellt, die das Entwickeln einer grafischen Oberfläche vereinfacht. Die maßgebliche Funktion ist das integrierte Scaffolding-System, da es unterschiedliche Layout-Möglichkeiten zur Verfügung stellt. Als Beispiel sei das fluid-grid-System aufgeführt, siehe *Abbildung 15 twitter-bootstrap fluid-grid-system*. Bei der Entwicklung kann die Größe, Position und der Inhalt jedes Gitters angegeben werden. Bootstrap berechnet dynamisch prozentual die Position für die entsprechende Auflösung.



Abbildung 15 twitter-bootstrap fluid-grid-system

Weiterhin unterstützt Bootstrap die einheitliche Gestaltung der Oberfläche, weil die verwendeten Elemente, z.B. Buttons oder Eingabeelemente, bereits mit hinterlegten CSS-Klassen gestylt sind. Die JavaScript-Elemente von Twitter-Bootstrap, wie z.B. Alerts oder Collapsible-Groups, werden von fast jedem Browser unterstützt. Die einzige Ausnahme ist der IE7.

5.4 Technisches Komponentendiagramm

Die

Abbildung 16 technisches Komponentendiagramm zeigt das entsprechende Diagramm dieser Anwendung. Der Entwurf entspricht dem MVC-Entwurfsmuster, erweitert um eine Utility-Komponente und der Geschäftslogik. Das Muster soll den leichten Austausch einzelner Komponenten ermöglichen. Ein weiterer Vorteil besteht darin, dass dieses Muster für die Web-Entwicklung weit verbreitet ist und von vielen Frameworks unterstützt wird.

Der Entwurf zeigt deutlich, dass das Play!-Framework eine zentrale Rolle spielt. Es dient als Schnittstelle zwischen allen Komponenten. Für die Model-Komponente übernimmt es den Zugriff auf die Datenbank. Neben dem automatischen Erstellen und Verwalten des Datenbankschemas übernimmt es auch die Lese- und Schreibzugriffe.

Für die View- und Controller-Komponente übernimmt das Play!-Framework das Routing und erstellt die entsprechenden Html-Responses.

Bei der Verwendung von Play! entsteht eine enge Kopplung zwischen Framework und Fachlichkeit. Dies verletzt das Entwurfsprinzip der losen Kopplung. Zum heutigen Zeitpunkt ist noch nicht sicher, ob das junge Framework lange bestehen bleiben wird. Falls neue Technologien entwickelt werden und Play! stagniert, könnte dies auch die Weiterentwicklung dieses Systems beeinflussen. Der Autor ist der Meinung, dass eine schnellere Implementierung diesen Nachteil unter dem gegebenen zeitlichen Rahmen ausgleicht.

<<utility>>

Die Utility-Komponente ist für das Halten aller benötigten Aufzählungstypen verantwortlich. Sie stellt diese für die anderen Komponenten zur Verfügung.

Weiterhin hat sie die Aufgabe, die Mock-Up-Daten zu erstellen und die Anwendung zu konfigurieren.

<<models>>

Diese Komponente enthält die Klassen des objektrelationalen Mappings. Diese können von anderen Komponenten verwendet werden, um Einrichtungen innerhalb eines bestimmten Umkreises oder Städte in einem bestimmten Postleitzahlengebiet zu finden.

Sie stellt die Erzeugen-, Löschen-, Suchen- und Ändern-Operationen für die Einrichtungen zur Verfügung.

<<controllers>>

Diese Komponente verarbeitet http-Requests und ist für das Transformieren der eingehenden Daten in ein für die Auswertung geeignetes Format zuständig. Mithilfe dieser Daten wird eine Auswertung in der geschäftslogik-Komponente angestoßen. Der Rückgabewert wiederum wird in einen http-Response umgewandelt, der dem Nutzer im Browser angezeigt wird.

<<geschäftslogik>>

In dieser Komponente befindet sich die Geschäftslogik, siehe 5.1. Damit das Diagramm übersichtlich bleibt, wurden hier die Komponenten findeEinrichtungen, bewerteEinrichtungen und zeigeEinrichtungenAn zusammengefasst und als Geschäftslogik bezeichnet.

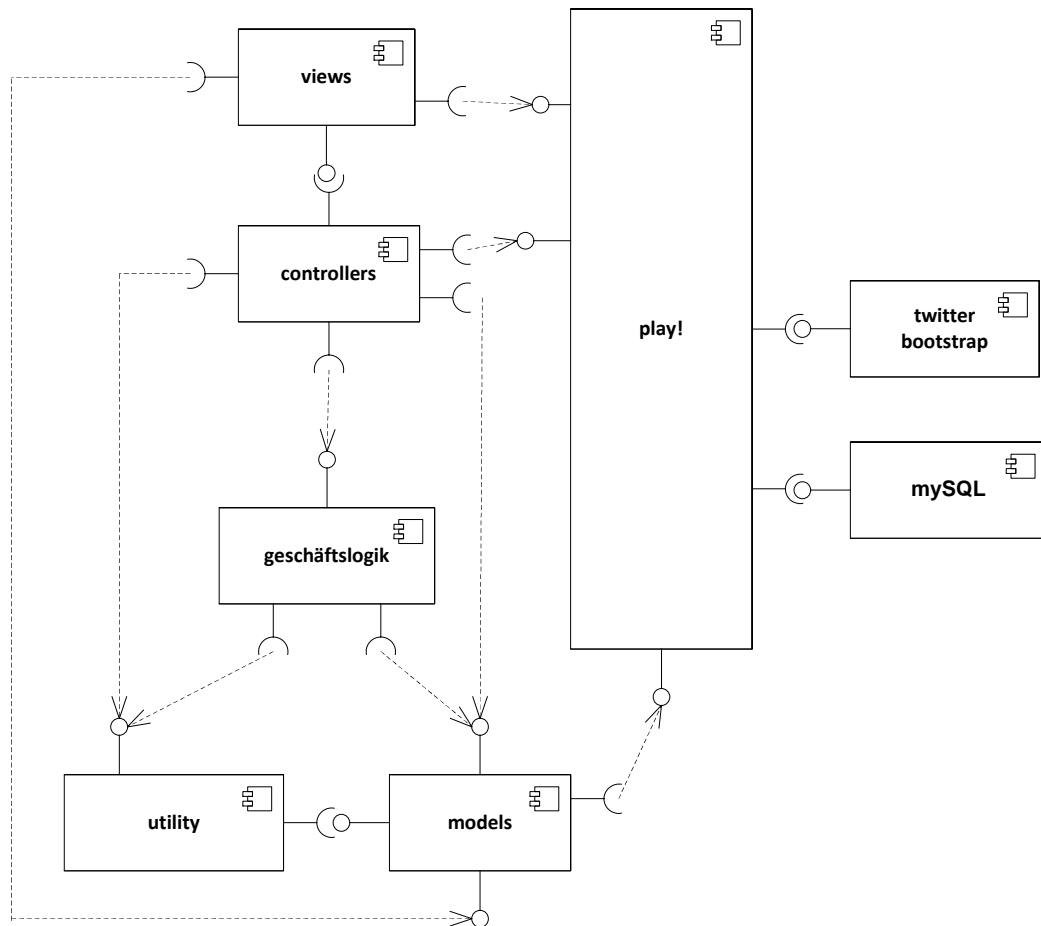


Abbildung 16 technisches Komponentendiagramm

5.5 Aufbau der Komponenten

Dieses Kapitel beschreibt die Funktionsweise der einzelnen Komponenten. Innerhalb jeder Komponente existiert eine Klasse `DataManager`. Diese Klasse dient als Schnittstelle zur `Model`- und `Utility`-Klasse. Diese Technik wird angewandt, damit die einzelnen Komponenten voneinander möglichst unabhängig bleiben und wiederverwendet werden können.

5.5.1 Controllers Innensicht

Der Aufbau des Controllers soll die Geschäftslogik möglichst vom Play!-Framework entkoppeln (siehe Abschnitt 5.4). Dazu dienen die Klassen `HomeController`, `SuchergebnisController` und `DataManager`, siehe *Abbildung 17 controllers Innensicht*. Diese stellen die Schnittstelle zu allen anderen Komponenten und zu Play! dar. Somit bleibt die Geschäftslogik von der eingesetzten Technik möglichst unabhängig.

<<HomeController>>

Diese Klasse nimmt den `index-Request` entgegen und veranlasst das Rendern der Hauptseite. Als Teil der einfachen Benutzbarkeit hat sich der Autor entschieden ein Autocompletefeld für den Ort umzusetzen. Dabei werden bisherige Eingaben des Nutzers verwendet, um passende Vorschläge anzeigen zu lassen. Hinzu werden ankommende `Json`-Objekte in Strings transformiert. Vom `DataManager` werden passende Städte angefordert. Die Antwort wiederum wird in eine `JsonObject`-Liste mit Städtenamen umgewandelt, die an das Formblatt zurückgesendet wird.

Weiterhin nimmt sie den `Request` des `Suche-Einrichtungen-Formulars` entgegen. Sie wandelt die eingegangenen Daten in entsprechende Entitäten um, indem sie diese vom `DataManager` anfordert. Diese werden genutzt, um ein `FormGlobalTransport`-Objekt zu erzeugen. Mithilfe des `Transportobjekts` wird von der Komponente `findeEinrichtungen` `Einrichtungsauswertung`-Objekte angefordert, die zum Rendern an die `SuchergebnisController` weitergereicht werden.

Sie prüft die Fehlerfälle `<FormError>` und `<Stadt konnte nicht gefunden werden>`. Für den `FormError` wird das Rendern eines Alerts und für den anderen das Rendern einer Information veranlasst.

<<SuchergebnisController>>

Diese Klasse veranlasst das Rendern der `Suchergebnis-Seite` und der `Detailansichts-Seite`. Sie nimmt den `Request` des `Filter-Formulars` entgegen. Sie transformiert die Daten so, dass sie ein `FormGlobalTransport`-Objekt vom `DataManager` anfordern kann. Zum Rendern der `Ergebnis-Seite` nutzt sie die Antwort der `bewerteEinrichtung`-Komponente, die sie vorher mit dem `Transport-Objekt` angestoßen hat.

Zum Rendern der `Detailansichts-Seite` wird von der Komponente `zeigeEinrichtungAn` die `Einrichtung-Entity` mit der passenden `Id` angefordert.

Sie prüft die Fehlerfälle <FormError> und <0 Einrichtungen gefunden>. Für den FormError wird das Rendern eines Alerts und für den anderen das Rendern einer Information veranlasst.

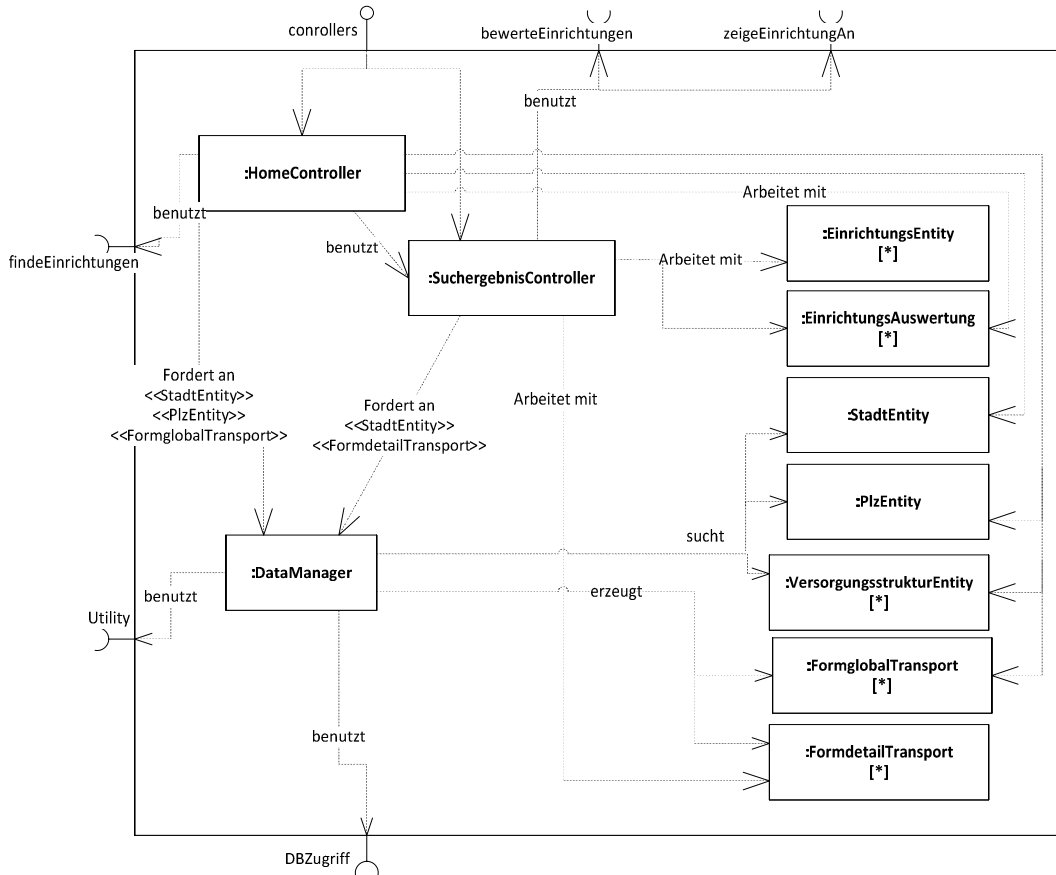


Abbildung 17 controllers Innensicht

5.5.3 bewerteEinrichtungen Innensicht

Zur Aufgabe der Klasse `BewerteEinrichtungenAnwendungsfall` gehört das Erstellen einer Rangliste, siehe *Abbildung 19 bewerteEinrichtungen Innensicht*. Dabei berechnet sie die Abweichungen in den Bereichen Entfernung zum Standort, Anzahl der angebotenen Leistungen, Kosten und gute Pflegenote, siehe 4.6. Bei den Kosten wird dabei unterschieden, ob sich die Suche auf einen Kurzzeitpflegeplatz bezieht (dann wird die Abweichung anhand der täglichen Kosten berechnet) oder ob es sich um eine Vollstationäre Unterbringung handelt. Zur Berechnung der angebotenen Leistung wird die Anzahl der Ernährungsstruktur-, Ausstattungs- und Pflegequalitäts-Entitäten herangezogen.

Als Ergebnis entsteht eine prozentuale Einschätzung, die als Grundlage für die Rangfolge dient, siehe 4.5.1 A3 & A5.

Eine zusätzliche Aufgabe dieser Klasse besteht darin, das Suchergebnis nach Angaben des Nutzers zu filtern. Dabei werden alle Einrichtungen aus dem Suchergebnis entfernt, die nicht zu den eingegangenen Profilkriterien passen, siehe 4.5.1 A2. Sollte dabei das Ergebnis auf 0 sinken, wird eine Einschätzung anhand der Anzahl der angebotenen Leistungen vorgenommen. Dies geschieht, damit der Nutzer nicht mehrfache Eingaben tätigen muss. Sollte das Ergebnis auf 0 Treffer sinken, würde der Nutzer vermutlich eine erneute Suche mit weniger Profilkriterien eingeben und die Suche erneut starten. Mit dieser Vorgehensweise bleibt ihm das erspart. Er sieht wie gut die verfügbaren Einrichtungen auf sein Profil passen.

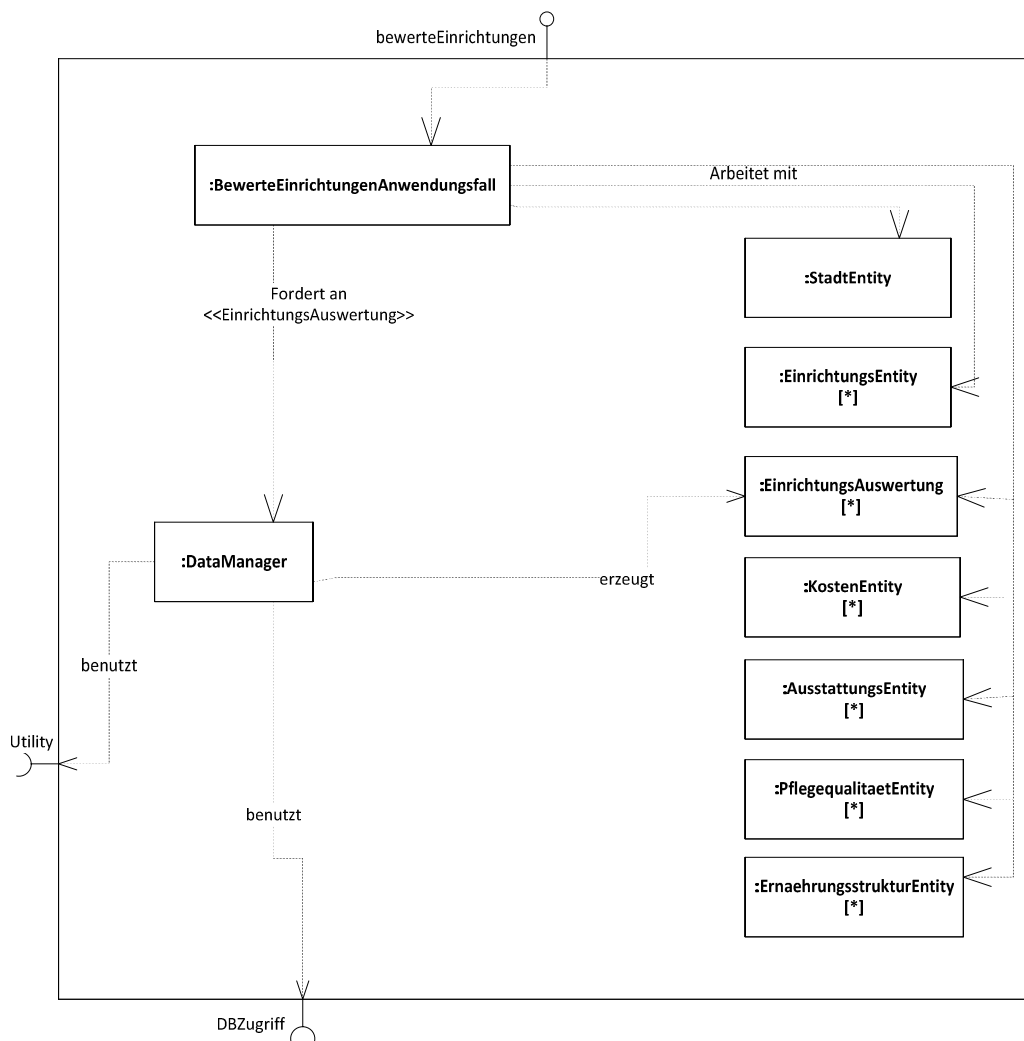


Abbildung 19 bewerteEinrichtungen Innensicht

5.5.4 models Innensicht

Die Klasse DBZugriff dient als Schnittstelle für alle Komponenten, die auf die Datenbank zugreifen. Sie leitet die Erstellen, Löschen, Lesen und Aktualisieren Anfragen an die entsprechenden Entitätsklassen weiter, siehe *Abbildung 20 models Innensicht*.

Die Entitätsklassen übernehmen mithilfe der Play!-Schnittstelle alle Datenbankoperationen.

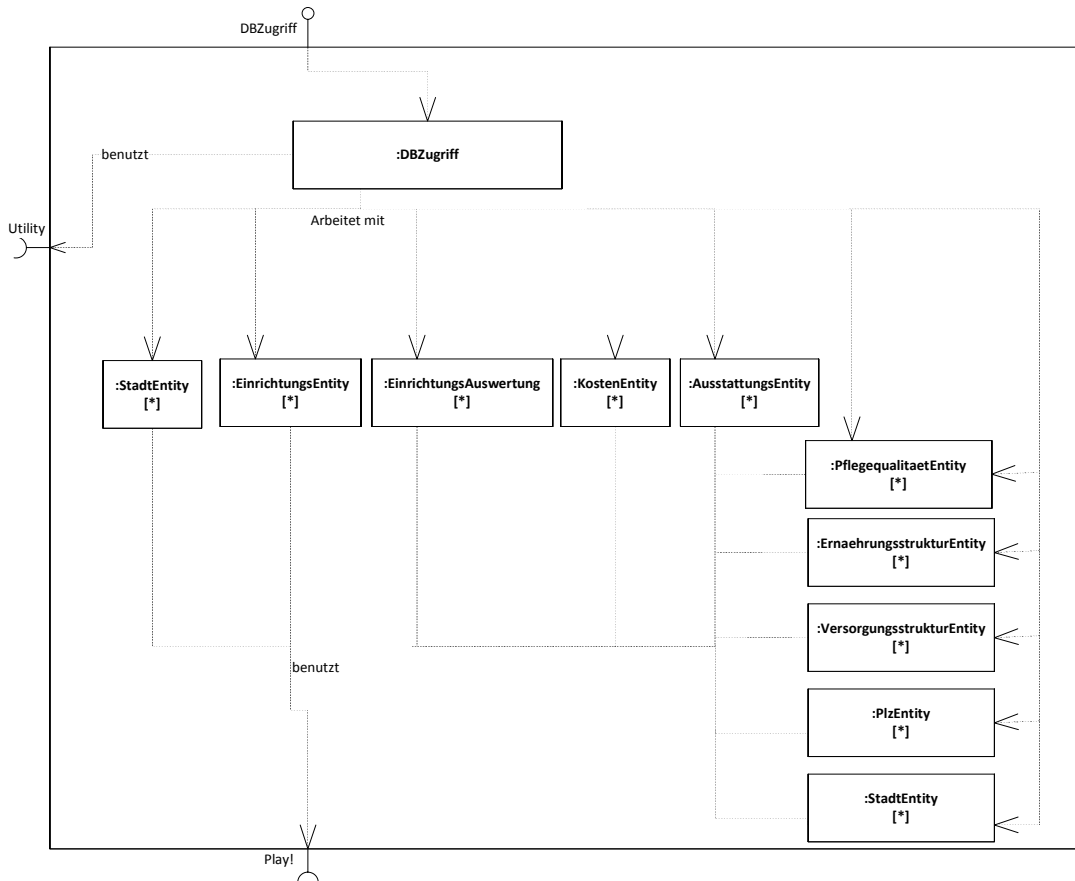


Abbildung 20 models Innensicht

5.6 Datenbankmodell

Die *Abbildung 21 ER-Diagramm* zeigt die Tabellen der Datenbank als ER-Diagramm. Dabei werden die Verwaltung und die Zugriffe durch das OR-Mapping des Play!-Frameworks übernommen. Die dazu notwendigen Klassen werden durch die Models-Komponente bereitgestellt. Die Tabellen Stadt und Plz werden mit OpenGeoDB Daten gefüllt, bereitgestellt durch ein MySQL Dump.

Die Utility-Komponente übernimmt das Speichern der Test-Einrichtungsdaten. Alle anderen Komponenten greifen lesend auf die Datenbank zu, siehe Kapitel 5.5.

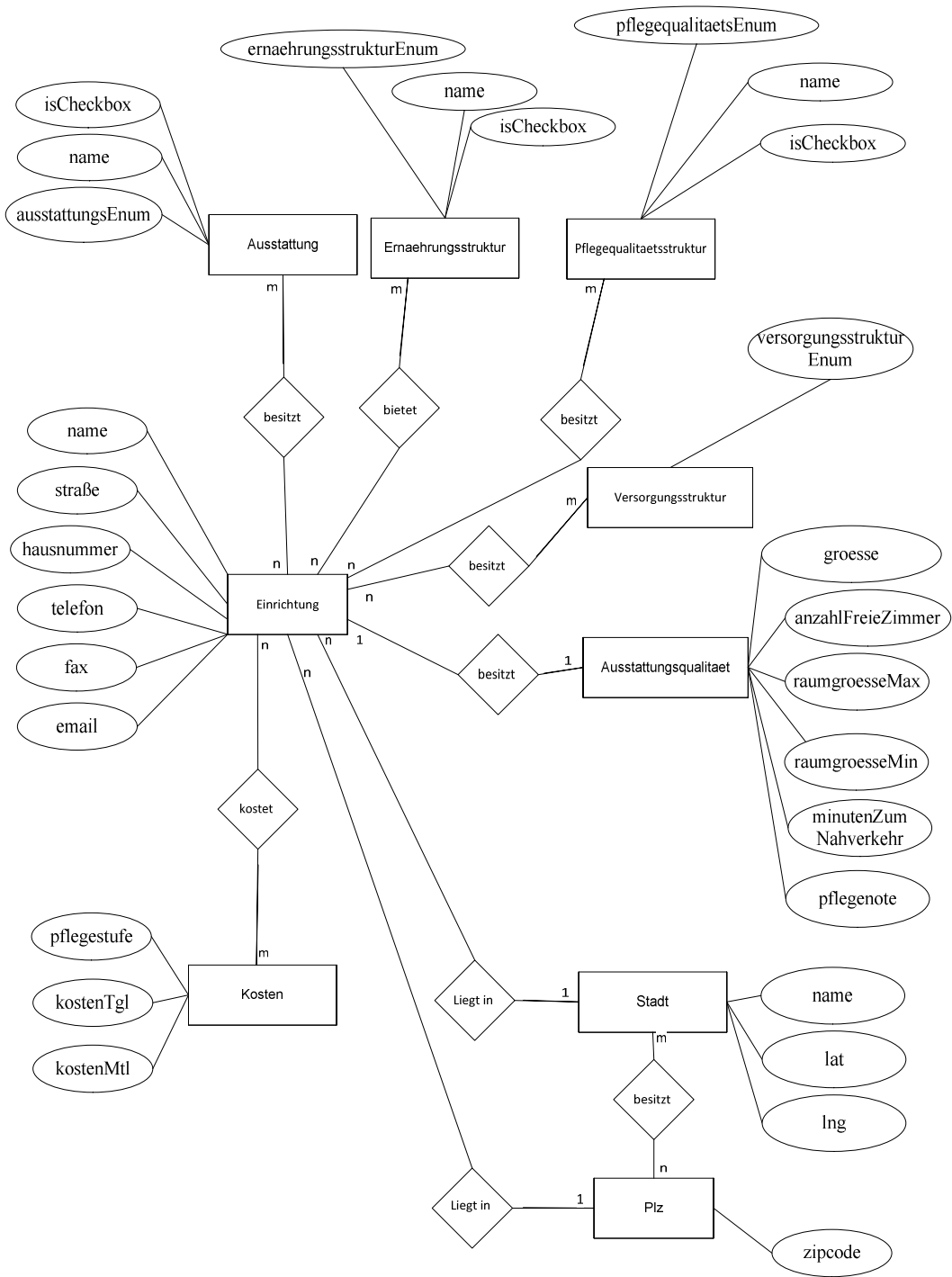


Abbildung 21 ER-Diagramm

6 Realisierung

Im folgenden Kapitel wird im Detail auf die Umsetzung der einzelnen Komponenten eingegangen. Es wird beschrieben, wie die Anforderungen umgesetzt wurden.

6.1 Weboberfläche mit Play!, Twitter Bootstrap und HTML

Wie bereits aus den vorangegangenen Kapiteln zu entnehmen war, besteht diese Anwendung aus der serverseitigen Anwendungslogik und einer clientseitigen Darstellung der Oberfläche im Browser des Benutzers.

Die Darstellung im Browser erfolgt durch HTML, Cascading Style Sheets (CSS) und wenige Java-Script-Elemente. HTML wird erzeugt und an den Browser ausgeliefert. Beim Einsatz von Play! werden HTML-Templates angelegt, die durch die Views-Komponente mit Inhalten gefüllt werden. Beim Kompilieren werden die views-, controllers- und models-Komponenten von Play! geparkt und entsprechende Scala-Klassen erzeugt. Aus diesem Grund müssen sich die HTML-Templates innerhalb der views-Komponente befinden.

Durch dieses Vorgehen ermöglicht es Play! innerhalb der views-Komponente Scala-Programmcode einzustreuen. Statt Inhalte anderer Templates zu erben können andere Templates mit Scala-Methoden „aufgerufen“ werden. So erweckt das Programmieren von HTML-Templates den Eindruck von objektorientierter Programmierung, siehe *Abbildung 23 HTML-Template eingestreuter Scala-Code*.

Die Templates sind so aufgebaut, dass sie kleine wiederverwendbare Einheiten bilden, die durch ihre Semantik getrennt sind. Hauptsächlich ist bei der Programmierung der Oberfläche HTML und CSS zum Einsatz gekommen. Am Beispiel von *Abbildung 22 Aufteilung der Oberfläche in kleine Templates* wird die Aufteilung in kleine Komponenten durch die farbliche Umrandung deutlich. Der blaue Bereich wird durch drei kleinere Bereiche aufgebaut: Der erste Bereich (rot umrahmt) ist das Informationsfeld, welches an unterschiedlichen Stellen der Anwendung wiederverwendet wird. Der zweite Bereich (grün umrahmt) ist Teil des Formulars, das mit Eingabe- und Dropdown-Feldern grundsätzliche Informationen abfragt und der dritte Bereich (violett umrandet) ist der Formularteil, der mit Checkboxes die Priorität festlegt. Durch dieses Vorgehen wird das Implementieren beschleunigt.

Pflegeeinrichtungen suchen

Hinweis: Bitte tragen Sie Ihre Angaben zur Suche einer Pflegeeinrichtung in die Felder ein.

Postleitzahl oder Ort (Pflichtfeld):

Umkreis:

Versorgungsform: Vollstationär Kurzzeitpflege

Pflegeschwerpunkt:

Prioritäten (Optional): günstiger Preis gute Pflegenote
 möglichst nah am gesuchten Standort möglichst viel Leistung

Abbildung 22 Aufteilung der Oberfläche in kleine Templates

In *Abbildung 23 HTML-Template eingestreuter Scala-Code* wird der eingestreute Scala-Code der Home-Seite gezeigt. Das '@-Symbol' dient dabei als Fluchtsymbol. Das 'Form-Keyword' signalisiert, dass der Methodenbody aus einer Form besteht. Der Parameter gibt an, wohin der Request für die Form geroutet werden soll. In Zeile 3 und 4 werden wie oben beschrieben zwei HTML-Templates „aufgerufen“, die erwarteten Parameter werden von Play! in das Template wie gewünscht integriert. Das Prioritäten-Template wiederum wird innerhalb des formglobal-Templates aufgerufen.

```

1. </header>
2.   @form(routes.HomeController.findInstitution()){
3.     @information("Bitte tragen Sie Ihre Angaben zur Suche einer Pflegeeinrichtung in die Felder ein.
4.     ")
5.     @formglobal(hinweis, grund, titel)
6.   }
7. </div>

```

Abbildung 23 HTML-Template eingestreuter Scala-Code

Wie bereits in Kapitel 5.3 beschrieben wird Twitter-Bootstrap hauptsächlich für das einheitliche Design und Layout verwendet. Um Twitter-Bootstrap vollständig einsetzen zu können wird neben der Bootstrap-Bibliothek auch die JQuery-Bibliothek in das Play!-Framework eingebunden. Dies ist notwendig, weil einige Funktionen von Bootstrap die JQuery-Bibliothek nutzen.


Das in Kapitel 5.3 beschriebene Gitter-System für das Layout erspart bei der Implementierung viel Programmierarbeit. Es werden die Größe des Gitters und ein Offset

angegeben und das Framework übernimmt die Berechnung anhand der Fenstergröße , siehe *Abbildung 24 Beispiel HTML für Bootstrap fluid-grid*.

1. `<div class="row-fluid">`
2. `<div class="span4 offset9" >...`
3. `</div>`

Abbildung 24 Beispiel HTML für Bootstrap fluid-grid

Ähnlich verhält es sich bei dem Design. Twitter-Bootstrap ermöglicht ein einheitliches Design und erspart somit dem Entwickler das Erstellen von eigenen Styles. Abweichend davon kann an einigen Stellen auf ein eigenes Styling nicht verzichtet werden. Als Beispiel dient *Abbildung 25 wenig Kontrast Warning-Box*. Für einen Teil der Zielgruppe kann es wegen des mangelnden Kontrasts zu einer erschwerten Lesbarkeit kommen. Dies wurde mit eigenen kontrastreichen Designs behoben, siehe *Abbildung 26 viel Kontrast Warning-Box*.



Warning! Best check yo self, you're not looking too good. ×

Abbildung 25 wenig Kontrast Warning-Box

Information.

Achtung: 0 Pflegeheime gefunden, die in das Profil passen.
Vorschlag: Bitte drücken Sie auf Schliessen und ändern Sie Ihre Auswahlkriterien. Z.B. Umkreis erweitern.

Abbildung 26 viel Kontrast Warning-Box

6.2 Datenbankzugriff

Der Datenbankzugriff in Play! erfolgt über das integrierte OR-Mapping von Hibernate oder Ebeans. So entfällt das Schreiben von SQL-Statements, welches die Entwicklungszeit beschleunigt und vereinfacht. Egal für welches ORM man sich entscheidet, die Tabelle der Datenbank wird als Klasse definiert und ein Objekt der Klasse repräsentiert einen Datensatz. Ein Datensatz wird durch die Instanziierung eines Objekts erzeugt. Der Datensatz kann über geerbte Klassenmethoden gesucht, geändert, gelöscht oder gespeichert werden. Als Erweiterung des ORM merkt sich Play! das aktuelle Datenbank-Schema und führt einen neuen Aufbau⁵ oder eine Aktualisierung der Tabellen aus, falls sich etwas am Schema ändert.

⁵ während der Entwicklungszeit

```
1. @Entity
2. public class EinrichtungsEntity extends Model {
3.
4.     private static final long serialVersionUID = 1L;
5.
6.     @Id
7.     @GeneratedValue(strategy=GenerationType.AUTO)
8.     private Long id;
9.
10.    // Kontaktdaten
11.    @Required
12.    private String name;
13.
14.    ...
15.
16.    @ManyToMany(cascade=CascadeType.ALL)
17.    private List<KostenEntity> kosten = new ArrayList<KostenEntity>();
18.
19.    ...
20.
21.    protected static Finder<Long, EinrichtungsEntity> find = new Finder<Long, EinrichtungsEntity>(Long.
22.        class, EinrichtungsEntity.class);
23.
24.    ...
25.    /**
26.     * loescht EinrichtungsEntity mit der passenden id aus der DB
27.     * @param id
28.     */
29.    protected static void delete(Long id) {
30.        find.ref(id).delete();
31.    }
```

Abbildung 27 Codebeispiel einer Modelklasse

Abbildung 27 Codebeispiel einer Modelklasse steht stellvertretend für den groben Aufbau aller Entitäts-Klassen in der models-Komponente. Die Klasse erbt von der Model-Klasse, die Bestandteil der Play!-API ist. Diese stellt Methoden für die Datenbankzugriffe zur Verfügung, wie z.B. `save()`, `update()`, `find()` und `delete()`, siehe Zeile 30.

Spalten in der Tabelle werden auf die Attribute der Entitätsklasse abgebildet. Bei dem Attribut 'name' handelt es sich um ein Textfeld beliebiger Größe⁶. Der Spaltenname wird nach dem Variablennamen benannt, in diesem Fall 'name'. Obwohl das Framework im Hintergrund die SQL-Statements generiert und ausführt, wird hier vom Entwickler auf dieser Ebene Einfluss auf das Datenbankschema genommen. Mit Hilfe der Java-Annotations können zusätzliche Argumente für die SQL-Statements 'übergeben' werden. So wird z.B. durch '@Required' in Zeile 11 ein 'NOT NULL' definiert. Durch die Annotation '@Id' wird das Attribut in Zeile 8 als Primärschlüssel festgelegt. '@GeneratedValue' kann als 'AUTO_INCREMENT' eingesetzt werden, sofern diese Funktion von der Datenbank

⁶ Die Einschränkung findet auf Datenbankebene statt, abhängig von der eingesetzten Datenbank.

unterstützt wird. Beim Erzeugen eines Objektes, sprich einem neuem Datensatz, wird automatisch ein eindeutiger Primärschlüssel erzeugt.

Auch die Beziehungen zwischen den Tabellen werden mit Hilfe von Annotations angegeben. Die Bezeichnung '@ManyToMany' drückt dabei in diesem Fall eine m:n Beziehung zwischen den Tabellen EinrichtungEntity und KostenEntity aus.

Ein weiteres Tag stellt das '@Entity' dar. Klassen, die sich innerhalb der models-Komponente befinden, können zwar von der views-Komponente zum Rendern genutzt werden, aber die Klasse wird nicht auf die Datenbank abgebildet und somit nicht persistent gehalten.

Die Klasse 'Finder', Zeile 21, stellt einfache Methoden zur Datenbanksuche zur Verfügung. Dabei bildet sie den gefundenen Datensatz auf die entsprechende Klasse ab und erzeugt automatisch ein Objekt dieser Klasse. Eingesetzt wird diese Technik in Zeile 30. Für alle Standardzugriffe auf die Datenbank reichen diese simpel gehaltenen Methoden aus. Für komplexere Zugriffe stellt die API noch weitere Methoden zur Verfügung, siehe *Abbildung 28 Play! Methoden zum Datenbankzugriff*. Anhand der aufgerufenen Methoden in Zeile drei und vier lässt sich erahnen, welches SQL-Statement daraus generiert und ausgeführt wird.

```

1. protected static List<StadtEntity> findStadtByExactName(String name){
2.     List<StadtEntity> cities = Ebean.find(StadtEntity.class)
3.         .where()
4.         .ilike("name", name)
5.         .findList();
6.     return cities;
7. }
```

Abbildung 28 Play! Methoden zum Datenbankzugriff

Für die Umkreissuche ist eine weitere mögliche Nutzung der Schnittstelle eingesetzt worden, siehe *Abbildung 29 Einsatz von reinem SQL*. Der Parameter der Where-Klausel, zwischen Zeile 6 bis 10, besteht aus reinem SQL-Code, der in dem generierten SQL-Statement eingebunden wird. In diesem Fall wird die Distanz jeder eingetragenen Stadt zum angegebenen Standort auf Datenbankebene ausgerechnet. Nur Städte, deren Distanz zum Standort kleiner ist als der Umkreis, werden zur Ergebnisliste 'cities' hinzugefügt.

```

1. StadtEntity start = findStadtByExactName(cityname).get(0);
2. String latStart = start.lat.toString();
3. String lngStart = start.lng.toString();
4.
5. List<StadtEntity> cities = Ebean.find(StadtEntity.class)
6.     .where("(round(1.609563462 "
7.         + " * SQRT( "
8.         + " ((69.1 * (lat - "+latStart+"))*(69.1 * (lat - "+latStart+")) "
9.         + " +((53.0 * (lng - "+lngStart+"))*(53.0 * (lng - "+lngStart+")) "
10.        + " ))<="+perimeter)
11.     .findList();
12. return cities;
```

Abbildung 29 Einsatz von reinem SQL

6.3 Komponente controllers

Diese Komponente ist hauptsächlich für das Transformieren von Daten zuständig, siehe Kapitel 5.5.1. *Abbildung 30 Klassendiagramm controllers* zeigt eine vereinfachte Sicht auf die hier im Kapitel beschriebenen Klassen.

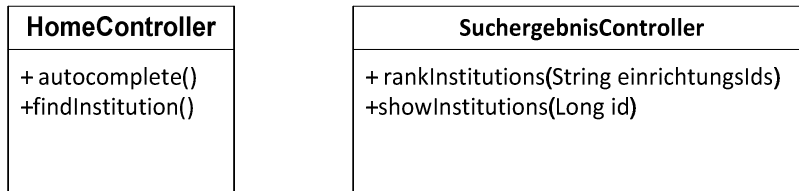


Abbildung 30 Klassendiagramm controllers

Implementierung HomeController und SuchergebnisController

Die Methode `<<autocomplete()>>` ist dafür zuständig dem Nutzer eine Auswahl von möglichen Städten unterhalb des Eingabefeldes „Ort“ anzuzeigen. Dazu liest sie das Json-Objekt aus und greift auf die Datenbank zu, um passende Stadt-Entitäten zu finden. Es liefert ein Json-Objekt mit einer Liste von Städtenamen zurück.

`<<findInstitution()>>` nimmt das Formular-Request der Home-Seite entgegen. Die Methode erstellt ein `FormglobalTransport`-Objekt aus den Daten des Formulars. Dabei handelt es sich um ein Transportobjekt, welches nur dazu dient Informationen in Form von Variablen zu halten, siehe *Abbildung 31 Klasse FormglobalTransport*. Mithilfe der Getter-Methoden können andere Komponenten auf die Informationen zugreifen. Anschließend wird die `findeEinrichtungenAnwendungsfall`-Klasse mit dem Transport-Objekt aufgerufen.

Die eingehenden einfachen Datentypen werden in die benötigten Entitätsobjekte umgewandelt.

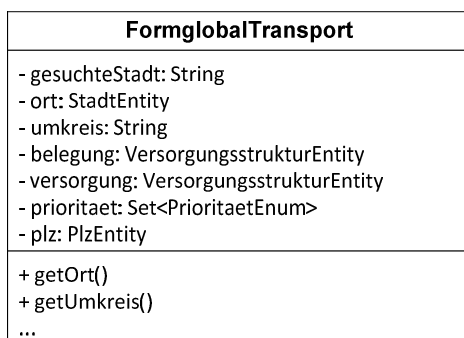


Abbildung 31 Klasse FormglobalTransport

An dieser Stelle bietet Play! keine große Unterstützung. Die Transformation muss komplett selbständig implementiert werden. Die *Abbildung 32 Play! Umwandlung von Form in Model-Objekt* zeigt den üblichen Umgang innerhalb von Play! mit eingehenden Form-Requests. Normalerweise reicht ein Methodenaufruf, um die Form-Daten in ein Model-Objekt zu erstellen und gleichzeitig in der Datenbank zu speichern, siehe *Abbildung 32 Play! Umwandlung von Form in Model-Objekt Zeile 8*.

```
1. public static Result newTask() {
2.     Form<Task> filledForm = taskForm.bindFromRequest();
3.     if(filledForm.hasErrors()) {
4.         return badRequest(
5.             views.html.index.render(Task.all(), filledForm)
6.         );
7.     } else {
8.         Task.create(filledForm.get());
9.         return redirect(routes.Application.tasks());
10.    }
11. }
```

Abbildung 32 Play! Umwandlung von Form in Model-Objekt

Im vorliegenden Szenario sollen die Daten in das passende Format umgewandelt werden. Das Beispiel in *Abbildung 33 Umwandlung String <<Ortfeld>> in Plz- oder Stadt-Entität* zeigt das eingesetzte Vorgehen.

```
1. private static FormglobalTransport createTransportGlobal(DynamicForm filledForm) {
2.     //auslesen Feld ort und interpretieren, ob ort oder plz gemeint ist.
3.     String feldOrt = filledForm.field("ort").value();
4.     PlzEntity plz = DataManager.findPlzByPlzString(feldOrt);
5.     StadtEntity ort = DataManager.findStaedteByExactName(feldOrt);
6.     if(ort==null){
7.         ort = DataManager.findStadtByPlz(feldOrt);
8.     }
9.
10.    //auslesen des Feldes Umkreis
11.    String umkreis = filledForm.field("umkreis").value();
12.
13.    ...
```

Abbildung 33 Umwandlung String <<Ortfeld>> in Plz- oder Stadt-Entität

Nach dem gleichen Prinzip arbeitet die `<<rankInstitutions()>>`-Methode für das Formular der Suchergebnis-Seite. Sie dient zum Erzeugen eines `FormDetailTransport`-Objekts, siehe *Abbildung 34 Klassendiagramm FormDetailTransport*, welches zum Aufrufen der `BewerteEinrichtungenAnwendungsfall`-Klasse verwendet wird.

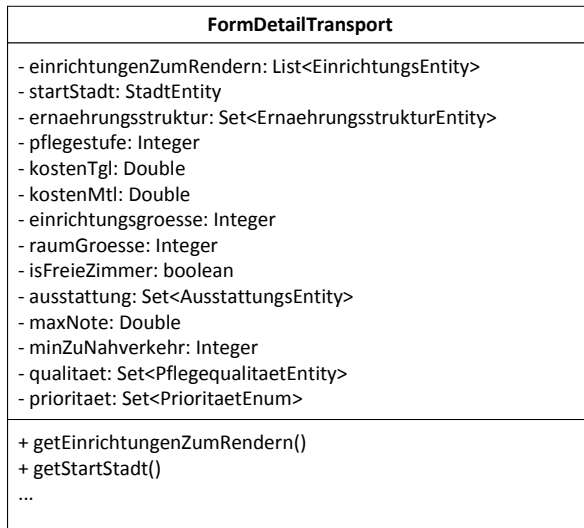


Abbildung 34 Klassendiagramm FormDetailTransport

6.4 Komponente findeEinrichtungen

Wie in Kapitel 5.5.2 bereits beschrieben sucht diese Komponente alle Einrichtungen aus dem Umkreis, die über die geforderte Versorgungsstruktur verfügen. Die Umsetzung hält sich dabei an die Vorgaben von Kapitel 5.5.2 und ist aufgebaut wie in

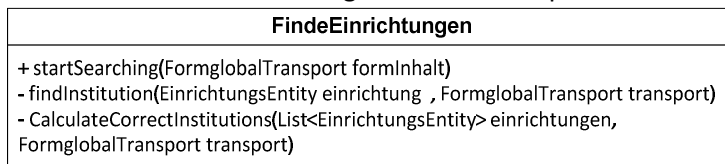


Abbildung 35 Klassendiagramm findeEinrichtungen gezeigt.

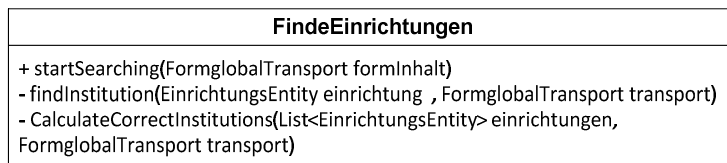


Abbildung 35 Klassendiagramm findeEinrichtungen

6.5 Komponente bewerteEinrichtungen

Kapitel 5.5.3 beschreibt die Arbeitsweise dieser Komponente. Im Unterkapitel Implementierung wird auf Einzelheiten der Umsetzung eingegangen.

Implementierung bewerteEinrichtungen

Die Klasse `BewerteEinrichtungenAnwendungsfall` implementiert die bereits beschriebene Geschäftslogik. Die Bewertung einer Einrichtung geschieht nach der in Kapitel 4.6 vorgestellten Formel. Um die Bewertung vorzunehmen werden vorher die Instanzvariablen `minTgl`, `minMtl`, `maxDistanz`, `startStart` initialisiert. Die Variablen repräsentieren den Sollwert `P`. Bei einer festen Größe, wie z.B. `Pflegenote`, wird diese Zahl als Sollwert verwendet. Bei einer variablen Größe, wie beispielsweise `Kosten`, wird der beste Wert ermittelt und entsprechend in der Formel verwendet. Die Methode `<<doAccelerationFromSearchresult>>` übernimmt die Bewertung der Einrichtungen. Der Parameter `'prioritaeten'` enthält die vom Nutzer angegebenen Bewertungsprioritäten. Es wird geprüft, welche Priorität enthalten ist und für jede enthaltene Priorität wird die entsprechende Auswertung vorgenommen. Das Ergebnis wird dann in der Ergebnisliste gespeichert. Im Anschluss wird der Durchschnitt berechnet. In *Abbildung 37 berechne Distanz Übereinstimmung* ist beispielhaft eine Berechnung aufgeführt.

BewerteEinrichtungenAnwendungsfall
- minTgl: Double - minMtl: Double - maxDistanz: Integer - startStart: StadtEntity - ausgewerteteEinrichtungen: List<Einrichtungsauswertung>
+ rankInstitutions(FormDetailTransport transport) + addInstituionForRender(EinrichtungEntity einrichtung, Integer pflegestufe) + doAccelarationFromSearchresult(Set<PrioritaetEnum> prioritaeten) ...

Abbildung 36 Klassendiagramm `BewerteEinrichtungenAnwendungsfall`

```

1. /**
2.  * berechnet die Uebereinstimmung zur maximalen Distanz. Maximum ist die entfernteste Einrichtung.
3.  * @param distanz, der Einrichtung vom Startpunkt
4.  * @return Prozentuale Uebereinstimmung zum Maximum
5.  */
6. private static Integer computeLocalAcceleration(Integer distanz) {
7.     if(maxDistanz>0){
8.         double tempdistanz = (double)distanz;
9.         return (int)(100-(((maxDistanz - tempdistanz)/maxDistanz)*100));

```

```

10. }else{
11.     return 0;
12. }
13. }

```

Abbildung 37 berechne Distanz Übereinstimmung

Neben der Bewertung übernimmt die Klasse ebenfalls die Transformation von der Einrichtungs-Entität in die EinrichtungsAuswertung. Dabei handelt es sich im Prinzip um ein Transportobjekt, wie es in vorangegangenen Kapiteln bereits beschrieben wurde, siehe *Abbildung 38 Klassendiagramm EinrichtungsAuswertung*. Dieses Objekt ist nötig, um zusätzliche Informationen für die View-Komponente bereitzustellen, wie z.B. Grad der Übereinstimmung zum Sollergebnis. Dabei handelt es sich um kurzfristige Objekte, die nach ihrem Einsatz wieder gelöscht werden. Im Gegensatz zu allen anderen Transportobjekten befindet sich diese Klasse innerhalb der models-Komponente, weil Play! innerhalb der View-Komponente nur auf Objekte der modelKomponente zugreifen kann.

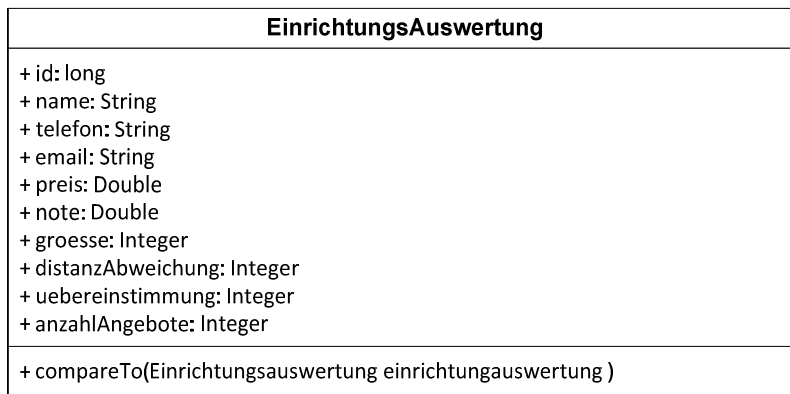


Abbildung 38 Klassendiagramm EinrichtungsAuswertung

Die Methode <<rankInstitutions>> übernimmt das Filtern der Einrichtungen (wie in Kapitel 5.5.3 beschrieben) und das anschließende Auswerten. Das FormDetailTransportobjekt enthält alle Informationen zum Filtern, siehe Kapitel 6.3. Für alle Einrichtungen wird geprüft, ob sie die gewünschten Eigenschaften erfüllen, siehe *Abbildung 39 Code - Filtern der Einrichtungen*.

```

1. /**
2.  * Wertet aus, ob die Einrichtung alle gewuenschten Eigenschaften erfuehlt.
3.  * @param detail, Transportobjekt, welches die Formularinformationen der Suchergebnis-
4.  * Seite beinhalten
5.  */
6. private static void addCorrectInstitution(FormDetailTransport detail) {
7.     for (EinrichtungEntity aktuelleEinrichtung : detail.getEinrichtungenZumRendern()) {
8.         && checkErnaehrungsstruktur(aktuelleEinrichtung, detail)
9.         && checkAusstattung(aktuelleEinrichtung, detail)
10.        && checkKosten(aktuelleEinrichtung, detail)

```

```

11.      && checkEinrichtungsgroesse(aktuelleEnrichtung, detail)
12.      && checkZimmergroesse(aktuelleEnrichtung, detail)
13.      && checkNote(aktuelleEnrichtung, detail)
14.      && checkMinZumNahverkehr(aktuelleEnrichtung, detail){
15.      addInstitutionForRender(aktuelleEnrichtung, detail.getPflegestufe());
16.      }
17.    }
18.  }

```

Abbildung 39 Code - Filtern der Einrichtungen

6.6 Komponente views

Die Benutzeroberfläche ist nach den Kriterien aus Kapitel 2.3 gestaltet. Bei den Eingabefeldern ist nur das Feld für die Ortseingabe als Pflichtfeld umgesetzt. Weiterhin ist auf Fachjargon oder fremdsprachige Ausdrücke verzichtet worden. Als Hilfestellung zur Eingabe für vorhandene Ortschaften ist eine Autovervollständigung implementiert worden, siehe *Abbildung 22 Aufteilung der Oberfläche in kleine Templates* und *Abbildung 40 Autovervollständigung*.

Abbildung 40 Autovervollständigung

Innerhalb jedes Abschnitts einer Seite werden Hinweise zur Benutzung gegeben. Um die Eigenschaften der Benutzbarkeit einzuhalten ist die Oberfläche wie folgt gestaltet worden: Alle Eingabefelder können in beliebiger Reihenfolge ausgefüllt werden. Bei der Eingabe kann sowohl Maus als auch Tastatur genutzt werden, wobei man wie üblich die Felder mit der Tabulator-Taste auswählen kann. Die Reihenfolge von oben nach unten und von links nach rechts wurde beachtet, wie es in der westlichen Kultur „best practice“ ist.

Für die Anzeige der Übereinstimmungsauswertung wurde ein Ampelsystem umgesetzt, wie es im Nutzerworkshop vorgeschlagen wurde. Dieses wird mit einem Piktogramm


umgesetzt, deren Bedeutung in der westlichen Kultur weitläufig bekannt ist. Die Piktogramme zeigen drei Bewertungen per Handzeichen, die je nach Bedeutung eingefärbt sind (Daumen nach oben in grün, Daumen zur Seite in gelb und Daumen nach unten in rot). Ein Ergebnis des Nutzerworkshops war die Anforderung, so wenig freie Texte wie möglich eingeben zu müssen. Dies ist in beiden Formularen berücksichtigt worden. Bei der Ortsangabe konnte nicht auf eine Nutzereingabe verzichtet werden. An allen anderen Stellen sind nur Eingabefelder der Form Dropdown, Checkbox oder Radiobuttons eingesetzt.

Suchergebnis					
Hinweis: Um mehr Informationen über eine Einrichtung zu erhalten, bitte auf den Namen klicken.					
Tabelle mit gefundenen Einrichtungen					
Name	Kontaktdaten	Größe der Einrichtung	Entfernung	Preis	Übereinstimmung
Einrichtung mit Bild Norderstedt 1	040-55 55 55 56 tolle.einrichtung@web.de	130 Betten	0 Km	2294,87 €	
Einrichtung in Norderstedt 2	040-55 55 55 56 tolle.einrichtung@web.de	60 Betten	0 Km	1754,87 €	
Einrichtung Hasloh	040-55 55 55 56 tolle.einrichtung@web.de	130 Betten	7 Km	2254,87 €	
Einrichtung Ellerau	040-55 55 55 59 tolle.einrichtung@web.de	20 Betten	10 Km	2594,87 €	

Abbildung 41 Suchergebnis

Um den Grundsatz der Aufgabenangemessenheit zu gewährleisten ist auf den Seiten Suchergebnis und Detailansicht die Accordion-Technik eingesetzt worden, die es ermöglicht nur die Informationen anzeigen zu lassen, die den Benutzer interessieren, siehe *Abbildung 41 Suchergebnis* und *Abbildung 42 Detailansicht*.

Kontaktdaten von Einrichtung Einrichtung in Norderstedt 2



Name der Einrichtung: Einrichtung in Norderstedt 2
Strasse: einige Attribute 23-25b
Ort: 22844 Norderstedt
Telefon: 040-55 55 55 56
Fax: 040-55 55 55 57
Email: tolle.einrichtung@web.de

Allgemeine Angaben zur Einrichtung
Bitte hier klicken, um mehr Informationen anzuzeigen

Kosten von Einrichtung in Norderstedt 2
Bitte hier klicken, um mehr Informationen anzuzeigen

Angebote Dienstleistung von Einrichtung in Norderstedt 2
Bitte hier klicken, um mehr Informationen anzuzeigen

Transparenzbericht
Bitte hier klicken, um mehr Informationen anzuzeigen

Abbildung 42 Detailansicht

Eine große Herausforderung ist die hohe Anzahl an Filterkriterien, nach denen Einrichtungen gefiltert werden können. Normalerweise sollte ein Formular möglichst kurz und übersichtlich gestaltet sein und den Benutzer nicht vor Eingabemöglichkeiten erschlagen. Diese Möglichkeit ist in diesem Fall nicht gegeben.

Die Lösung besteht darin die Felder in semantische Gruppen aufzuteilen, die durch Überschriften deutlich voneinander getrennt sind. Dadurch entsteht der Anschein, es handele sich um kleine Artikel einer Zeitschrift, die man kurz lesen kann. Dieser Anschein wird durch eine weitere Aufteilung in drei Spalten unterstützt, siehe *Abbildung 43 Filtern des Suchergebnisses*.

Suchergebnis eingrenzen.
 Bitte hier klicken, um die Suche einzugrenzen

Hinweis: Füllen Sie das Formular nach Ihren Wünschen aus und drücken Sie auf den Knopf -Ergebnis berechnen- um weniger Ergebnisse angezeigt zu bekommen. Sollte dies nicht möglich sein, werden Ihnen die besten Ergebnisse am Anfang der Tabelle angezeigt.

Versorgungsstruktur <input type="checkbox"/> Diätspesen <input type="checkbox"/> Getränkeangebot <input type="checkbox"/> Schonkost <input type="checkbox"/> Menüwahl <input type="checkbox"/> pürierte Kost <input type="checkbox"/> vegetarische Speisen <input type="checkbox"/> hausgene Küche Heimentgelt Heimentgelt für Pflegestufe: <input type="text" value="0"/> Eigenleistung Heimentgelt pro Tag: <input type="text" value="egal"/> Eigenleistung Heimentgelt pro Monat: <input type="text" value="egal"/>	Ausstattung und Leistung Heimgröße gemessen in Bettenanzahl: <input type="text" value="egal"/> Zimmergröße in qm: <input type="text" value="egal"/> <input checked="" type="radio"/> Einzelzimmer <input type="radio"/> Mehrbettzimmer <input type="checkbox"/> Einrichtung hat freie Zimmer <input type="checkbox"/> Telefon auf Zimmer <input type="checkbox"/> Fernsehen auf Zimmer <input type="checkbox"/> Internet auf Zimmer <input type="checkbox"/> Dusche auf Zimmer <input type="checkbox"/> Sauna <input type="checkbox"/> Schwimmbad <input type="checkbox"/> Gartenanlage <input type="checkbox"/> Ladengeschäft auf dem Gelände <input type="checkbox"/> Bibliothek <input type="checkbox"/> Frisör	Leistungen im Bereich Pflegequalität maximale Pflegenote: <input type="text" value="egal"/> Anbindung ans öffentliche Verkehrsnetz: <input type="text" value="egal"/> <input type="checkbox"/> MDK-Transparenzbericht <input type="checkbox"/> Tagesgestaltung <input type="checkbox"/> Beschäftigungstherapie <input type="checkbox"/> Ausflüge <input type="checkbox"/> Feste ausrichten <input type="checkbox"/> Betreuung nach § 87b SGB XI <input type="checkbox"/> Einsatz von Expertenstandards <input type="checkbox"/> Fahrdienst <input type="checkbox"/> Gottesdienst <input type="checkbox"/> Tierhaltung ist erlaubt <input type="checkbox"/> Personal ist ausgebildet <input type="checkbox"/> Eigene Möbel dürfen mitgebracht werden
--	---	--

Prioritäten
 günstiger Preis
 möglichst nah am gesuchten Standort
 gute Pflegenote
 möglichst viel Leistung

Abbildung 43 Filtern des Suchergebnisses

6.7 Softwaretests

Softwaretests werden unter anderem eingesetzt, um die Qualität eines Softwaresystems zu messen, um vorhandene Fehler aufzudecken und um zu prüfen, ob gesetzliche Vorgaben und Kundenanforderungen erfüllt wurden. Das Planen und Implementieren von Softwaretests ist ein umfangreiches Thema innerhalb der Softwareentwicklung. Es gibt verschiedene Teststufen, wie z.B. Komponententests, Integrationstests, Systemtests und Abnahmetests und verschiedene Testarten, wie z.B. Black-Box-Tests, White-Box-Tests und Grey-Boy-Tests.

Black-Box-Tests werden ohne Kenntnisse über den Aufbau des Systems entwickelt. Als Basis dienen häufig die Entwicklungsdokumente. Oftmals werden Black-Box-Tests nicht vom Entwickler selbst durchgeführt sondern von fachlich geschulten Testern, damit ein Missverständnis des Entwicklers nicht das Testergebnis verfälscht.

White-Box-Tests werden meist vom Entwickler selbst geschrieben, weil diese Art von Tests Kenntnisse über den inneren Aufbau der Software voraussetzt. Ein falsches Verständnis über die Funktionsweise der Software kann ebenfalls zu „falschen“ Testergebnissen führen. Einige dieser Testarten können mithilfe von Unit-Tests automatisiert durchgeführt werden, um eine gleichbleibende Qualität zu gewährleisten. Während der Entwicklungszeit sollten sich auch die Tests gleichzeitig mit dem Code weiterentwickeln.

Die oben beschriebenen Whitebox-Tests können für Komponententests eingesetzt werden. Sie überprüfen die funktionale Korrektheit der Komponenten unabhängig voneinander. Integrationstests überprüfen hauptsächlich die Schnittstellen zwischen den Komponenten, um die Kompatibilität zwischen ihnen zu gewährleisten. Diese Tests sind besonders wichtig, will man Dienste externer Systeme nutzen, denn die externen Protokolle und Datenformate unterscheiden sich fast immer von den eigenen. Hierfür bieten sich zwei Testverfahren an: Das Top-Down- oder das Bottom-Up-Verfahren. Im Top-Down-Verfahren werden zu Beginn der Entwicklung die angrenzenden Komponenten mithilfe von Mock-Objekten simuliert. Das Erstellen dieser Dummy-Daten ist dabei eine aufwendige Arbeit. Beim Bottom-Up-Verfahren hingegen werden Integrations-Tests nacheinander mit fertigen Komponenten durchgeführt. Falls diese nicht zusammenpassen, kann eine spätere Korrektur aufwendig sein.

Abnahmetests werden meist vom Kunden selbst als Blackbox-Test durchgeführt. Er testet häufig anhand der Oberfläche, ob die Anforderungen an die Software erfüllt wurden. Teilweise können die Oberflächentests auch automatisiert durchgeführt werden.

Lassen sich mithilfe der oben genannten Tests die funktionale Korrektheit und das Erfüllen der funktionalen Anforderungen überprüfen, so gibt es auch Testverfahren, die einige nicht-funktionale Anforderungen und die Benutzbarkeit untersuchen.

Ein Performance-Test dient dazu Fehler im System aufzudecken, die bei den oben genannten Testarten nicht auftauchen, z.B. Deadlocks bei Datenbankzugriffen, Auffinden von Bottlenecks innerhalb der Anwendung, das Auftauchen von Dateninkonsistenzen oder

zu hohe durchschnittliche Antwortzeiten. Die grundsätzliche Idee hierbei ist, das Verhalten der Anwendung im laufenden Betrieb zu untersuchen. Dabei wird auf die laufende Anwendung eine hohe Last erzeugt, z.B. durch virtuelle Nutzer, die zeitgleich auf das System zugreifen, währenddessen wiederholt Test-Fälle bzw. Testprozesse ausgeführt werden. Anhand der gesammelten Daten findet eine Auswertung statt.

Um die Benutzbarkeit einer Anwendung zu untersuchen stehen unterschiedliche Usability-Tests zur Verfügung. Grundsätzlich lassen sich diese in qualitative und quantitative Tests aufteilen. Zu den qualitativen Tests gehören die bereits in den Kapitel 2.2.1 und 2.2.2 beschriebenen klassischen Labor-Experimente und virtuelle Aufmerksamkeitsstudien. Eine Abwandlung dieser Tests bietet ein Remote-Usability-Test, der grundsätzlich ähnlich funktioniert wie der klassische Test. Der Nutzer kann sich mithilfe einer Software mit dem Testrechner, dem Usability-Experten und dem Moderator verbinden. Der Proband führt wie gewohnt den Test durch, mit dem Unterschied, dass er sich zu Hause am eigenen Rechner befindet. Anhand seines Verhaltens und der Gespräche mit dem Moderator kann eine Auswertung stattfinden. Einzig das Eye-Tracking und somit die Erstellung einer Heatmap fallen als Auswertungs-Instrument weg.

Die quantitativen Tests werden meist im laufenden Betrieb durchgeführt, um eine möglichst hohe Anzahl an Testpersonen zu gewinnen. Das Aufzeichnen der Mouse-Bewegung auf Webseiten ist eine beliebte Art von quantitativem Benutzertests. Im Web-Bereich gibt es verschiedene Anbieter für diesen Service. Je nachdem welche Daten für den Test von Interesse sind, können ganze Videos von Einzelsessions aufgezeichnet werden oder Heuristiken von Klickaktionen, Bewegungsmuster oder Interaktionen mit Formularen erstellt werden.

6.7.1 Durchgeführte Softwaretests

Im vorangegangenen Kapitel wird deutlich, dass das Testen einer Anwendung sehr aufwendig und damit zeitintensiv ist. Aus diesem Grund hat sich der Autor auf wenige Testverfahren beschränkt. Alle verfügbaren Tests durchzuführen würde den zeitlichen Rahmen dieser Arbeit bei weitem überschreiten.

Innerhalb der Entwicklung dieser Software wurden Komponenten- und Integrations-Tests mithilfe des White-Box-Ansatzes durchgeführt. Der Autor hat sich aus folgenden Gründen für diese Art der Tests entschieden. Zum einen steigern sie die Qualität der Software, indem die Kernfunktion der Anwendung überprüft wird. Zum anderen benötigen sie keine zusätzliche Software oder Hardware, wie z.B. Performance-Tests oder Usability-Tests. Weiterhin können sie vom Entwickler selbst automatisiert durchgeführt werden.

Während der gesamten Entwicklungszeit wurden die Tests erweitert und an den Stand der Entwicklung angepasst. Am Anfang wurden fehlende Funktionen oder Komponenten durch Mock-Up-Daten ersetzt. Im späteren Verlauf wurden diese durch die echten Funktionen ausgetauscht. Als Testdaten dienten ausgewählte Orte und Dummy-Einrichtungen mit unterschiedlichen Eigenschaften.

6.7.2 Implementierung Softwaretests

Während der gesamten Implementierungsphase wurden mehrere Unit-Tests zum Testen der einzelnen Komponenten implementiert. Play! stellt für Softwaretests eine API zur Verfügung. Mithilfe dieser API kann u.a. eine Serversimulation für die Tests gestartet und gestoppt werden, siehe *Abbildung 44 start und stop fakeApp*. Somit ist es möglich Modeltests und http-Requests durchzuführen.

Mithilfe eines Play! Konsolenbefehls werden alle Tests innerhalb der Testkomponente ausgeführt.

```
1. @BeforeClass
2. public static void startApp() {
3.     app = Helpers.fakeApplication();
4.     Helpers.start(app);
5.
6.     ....
7.
8. @AfterClass
9. public static void stopApp() {
10.     Ebean.execute(Ebean.createCallableSql(dropDdl));
11.     Helpers.stop(app);
12. }
```

Abbildung 44 start und stop fakeApp

Für die Tests wurde JUnit verwendet. Es wurden verschiedene Tests für die Anwendungsfallklassen geschrieben. Alle Tests, die die Geschäftslogik betreffen, verwenden die 'fakeApplikation', damit sie Zugriff auf die Datenbank erhalten. Dieses Vorgehen wurde gewählt, um das Zusammenspiel aller Komponenten zu überprüfen. Die automatisierten Tests der Anwendungsfallklassen kontrollieren, ob das richtige Ergebnis ausliefert wird. Ist dies nicht der Fall, wird von der Testmethode ein 'Error' generiert, das anzeigt, welches Ergebnis falsch ist.

7 Auswertung

7.1 Fazit

Ziel dieser Arbeit war es, eine innovative benutzerfreundliche Suchmaschine für Pflegeheime in Deutschland zu konzipieren und umzusetzen.

Dazu wurden zunächst die Grundlagen zweier Themenbereiche betrachtet. Ein kleiner Einblick in die Welt der Pflege offenbarte, dass die gesetzlichen Vorgaben zur Veröffentlichung von Pflegedaten sehr heterogen ausgelegt werden. Es wurde gezeigt, wie unterschiedlich die Daten der einzelnen Einrichtungen von den Landesverbänden bereitgestellt werden und dadurch eine Vergleichbarkeit erschweren. Weiterhin wurde ein kleiner Einblick in das Thema der Benutzbarkeit von Software gegeben. Die angeführten Daten stellen keineswegs das gesammelte Wissen dieses Themengebietes dar. Vielmehr soll dem Leser ein Grundverständnis für die verschiedenen Aspekte dieses Thema vermittelt werden, damit er den Ausführungen des Autors mühelos folgen kann.

Von den im Internet bestehenden Pflegeheimfindern wurden vier Webseiten exemplarisch ausgewählt und einer Benutzbarkeits-Analyse unterzogen. Ziel war es hier, ein mögliches Verbesserungspotenzial zu ermitteln. Auch wenn eine Seite ausführliche Informationen anbot und zwei andere Seiten Preisvergleiche ermöglichten, konnte keine der vier getesteten Seiten ein rundum gelungene Mischung aus Information und Gegenüberstellung vorweisen. Diese Lücke sollte mit der vorliegenden Suchmaschine gefüllt werden.

Um möglichst nah an den Bedürfnissen der Nutzer operieren zu können, wurde daraufhin eine Zielgruppenanalyse durchgeführt. Es war für die Konzeptionierung der Suchmaschine äußerst hilfreich, die Zielgruppe und ihre speziellen Bedürfnisse identifiziert zu haben. Das Verständnis für die Anforderungen der Nutzer ist dadurch stark gestiegen. Sollen Nutzer am Entwicklungsprozess teilhaben, so ist es nach Meinung des Autors unumgänglich eine Zielgruppenanalyse durchzuführen. Allerdings ist das Ergebnis kritisch zu betrachten, denn es kann immer nur eine Momentaufnahme der aktuellen Beobachtungen wiedergeben. Die Bedürfnisse der Nutzer können sich im Laufe der Zeit wandeln und in der Marktforschung werden stetig neue Erkenntnisse gewonnen. Eine Analyse der Zielgruppe ist solch ein komplexes Thema, dass dies bei einem kommerziellen Projekt durch ein Expertenteam erfolgen sollte.

Der Nutzerworkshop erwies sich als geeignete Methode, die Anforderungen direkt und spontan vom Nutzer zu erhalten oder sie mit ihnen gemeinsam zu erarbeiten. Die dabei verwendeten Attrappen waren ein sehr nützliches Hilfsmittel, weil die Teilnehmer sofort am Bildschirm über eine mögliche Visualisierung diskutieren konnten. Jedem fiel das unfertige Design in Form von kurvigen Strichen sofort auf und der Moderator konnte

erklären, dass es sich bei der Oberfläche um eine Attrappe handele und kein fertiges Produkt sei. Dies führte zu interessanten Denkanstößen und nützlichen Hinweisen seitens der Nutzer. Werden Nutzer am Entwicklungsprozess beteiligt, kann es vorkommen, dass Ideen des Entwicklers und Interessen der Nutzer voneinander abweichen. Da in dieser Arbeit die Usability im Vordergrund steht, ist aus diesem Grund auf große grafische Neuerungen zugunsten der besseren Lesbarkeit verzichtet worden. Bei Befragung der Nutzer im Workshop zeigte sich, dass die tabellarische Anzeige der Ergebnisse deutlich besser abschnitt als ungewohnte Formate. Der Autor hält eine Beteiligung der Nutzer am Entwicklungsprozess für förderlich, selbst wenn die Nutzer eine komplett andere Idee von der Anwendung haben als die Entwickler. Gerade dann ist der Austausch miteinander lehrreich und führt zu zufriedenstellenden Resultaten. Schließlich sollen die Nutzer am Ende mit der Anwendung umgehen.

Nach Festlegung der Anforderungen an die Software wurde die Systemarchitektur erarbeitet und nach üblichen Verfahren textuell und grafisch dargestellt. Dies stellte eine gewisse Herausforderung dar, weil das ausgewählte Framework nur ein bestimmtes Design-Pattern zuließ und somit andere Vorgehensweisen verhinderte. Trotzdem halfen die gewählten Frameworks dabei die Entwicklung schnell voranzutreiben. An einigen Stellen lieferten sie nicht die gewünschte Funktionalität, aber mit den gewählten Workarounds konnte in der Kürze der Zeit ein guter Mittelweg gefunden werden.

Rückblickend betrachtet ist es vor allem dem Input der Nutzer zu verdanken, dass die Entwicklung des Pflegebutlers erfolgreich verlaufen ist, auch wenn eine innovative Komponente fehlt. Es gibt viele verschiedene Möglichkeiten den Nutzer am Entwicklungsprozess zu beteiligen und das in dieser Arbeit beschriebene Vorgehen ist nur eines von vielen. Es ist anzunehmen, dass unter anderen Voraussetzungen andere Verfahren besser geeignet sind. Welches Vorgehen gewählt wird, sollte abhängig von den vorherrschenden Bedingungen neu entschieden werden. In einer realen Umgebung wäre die hier verwendete Methode zeitaufwendig und teuer in der Umsetzung. Trotzdem sollte der Nutzer am Entwicklungsprozess teilhaben, sei es mit Umfragen oder Workshops oder anderen Mitteln, denn er liefert unschätzbare Ideen, die unbedingt in die frühen Phasen der Entwicklung einbezogen werden sollten, um späteres Nachbessern zu vermeiden.

7.2 Mögliche Verbesserungen

Bei der Gestaltung der Oberfläche wurden alle Punkte der Benutzbarkeit aus dem Kapitel 2 berücksichtigt. Allerdings ist aufgrund der Nutzermeinung die grafische Innovation nicht umgesetzt worden, wie bereits im vorherigen Kapitel beschrieben. Man könnte mit den Nutzern weitere neuartige Gestaltungsideen diskutieren. Ein Beispiel wäre eine Magnettafel mit Post-It-Aufklebern, die die Suchergebnisse darstellen. Pro Post-It würde eine Einrichtung abgebildet und die Farbe des Post-Its signalisierte den Rang in der Ergebnisliste.

Dies wäre eine frischere, jugendlichere Form und würde auch das jüngere Publikum ansprechen.

Weiterhin ist die Anwendung nicht auf die Kompatibilität mit mobilen Endgeräten wie z.B. Smartphones oder Tablet-PCs getestet worden. Da es sich bei der Software um eine Webanwendung handelt, wäre die Verwendung auf mobilen Geräten denkbar, obwohl ein Großteil der Zielgruppe vermutlich kein mobiles Endgerät nutzt. Die verwendete Technik sollte zwar die Anzeige auf mobilen Geräten ermöglichen, aber eine Optimierung für die Verwendung mit Touch-Eingabegeräten fehlt.

Um eine gute Usability zu gewährleisten ist es nötig, einen barrierefreien Zugang zu der Anwendung zu ermöglichen. Aus diesem Grund wäre es wünschenswert, einige Benutzbarkeitstests mithilfe einer Sprachausgabe-Software durchzuführen, um festzustellen, ob die Software einen barrierefreien Zugang erlaubt. Denkbar wäre, dass die eingesetzten Icons zur Bewertungsdarstellung zusätzlich in Textform angezeigt werden müssten. Auch wären abschließende Usability-Tests hilfreich, um festzustellen wie der Nutzer mit dieser Anwendung umgeht. Diese könnten im laufenden Betrieb als quantitative Studie durchgeführt werden. So könnte sich ein Profil mit häufig verwendeten Auswahlkriterien ergeben. Dieses Profil könnte dann gleich auf der ersten Seite ausgewählt werden. Ein Ergebnis der Studie könnte auch das Wegfallen einiger nicht genutzter Kriterien bedeuten, was wiederum eine bessere Übersicht ermöglicht.

Innerhalb der Entwicklung sind die Softwaretests nicht in ausreichender Form durchgeführt worden. Daher wäre es erstrebenswert, den Umfang der Softwaretests auszubauen. Dabei sollten die vorhandenen Tests erweitert werden. Ebenfalls könnten automatisierte Oberflächen-Tests entworfen und durchgeführt werden. Es wäre wertvoll Performance-Tests durchzuführen, um vorhandene Schwachpunkte zu identifizieren und zu eliminieren, bevor die Anwendung „marktreif“ genannt werden kann.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

In dieser Arbeit wurde eine Web-Applikation zum Finden und Bewerten von Pflegeeinrichtungen entwickelt. Das Kapitel Grundlagen liefert einen Überblick über die Qualitäts- und Leistungskriterien von Pflegeheimen sowie einen Ausblick, welche Informationen zur Verfügung gestellt werden sollten, damit sie vergleichbar werden. Weiterhin wurde ein kleiner Einblick in die Themen Benutzerfreundlichkeit und Design von grafischen Oberflächen gegeben und mit welchen Mitteln Benutzer in den Entwicklungsprozess einbezogen werden können.

Es wurden bestehende Pflegeheimfinder und deren Gebrauchstauglichkeit anhand von vier Grundsätzen untersucht und die Ergebnisse vorgestellt. Dabei handelt es sich nicht um eine vollständige Aufzählung aller Suchmaschinen, die sich mit Pflege auseinandersetzen. Auch die Untersuchung der Benutzbarkeit beschränkte sich auf wenige Kriterien und ist nicht repräsentativ. Es wurde eine Übersicht der zurzeit genutzten Systeme und deren Stärken und Schwächen vorgestellt.

Durch Zuhilfenahme verschiedener Statistiken konnte eine differenzierte Beschreibung der potentiellen Nutzer zusammengestellt werden. Es handelt sich um Personen im Alter von 41+, die nur wenig Kontakt mit dem Internet haben. Anschließend wurde eine Soll-Analyse unter Einbeziehung der Nutzer durchgeführt, die die Anforderungen für die Anwendung festlegte. Einbezogen wurden die Nutzer mittels zweier Workshops. Die Ergebnisse wurden anschließend in die Anforderungsanalyse integriert und in Teilaufgaben untergliedert.

Im Kapitel Systemarchitektur wurde die Architektur dieser Anwendung erläutert. Es wurden die fachlichen, technischen Komponenten und das Datenbankmodell dargestellt, sowie die Zusammenhänge untereinander beschrieben.

In der Realisierung wurde im Detail auf die Umsetzung der einzelnen Komponenten eingegangen und wie die Anforderungen verwirklicht wurden. Dabei wurde besonderes Augenmerk auf die verwendete Technik und die verwendeten Frameworks gelegt.

8.2 Ausblick

Um korrekte Daten von Einrichtungen zu erhalten, wäre es denkbar eine Schnittstelle für Unternehmen anzubieten, die Ihre Daten der Plattform zur Verfügung stellen. Dies könnte ebenfalls online geschehen. Dabei müsste die Anwendung eine Authentifizierung ermöglichen, damit die Unternehmen nur ihre eigenen Daten manipulieren können.

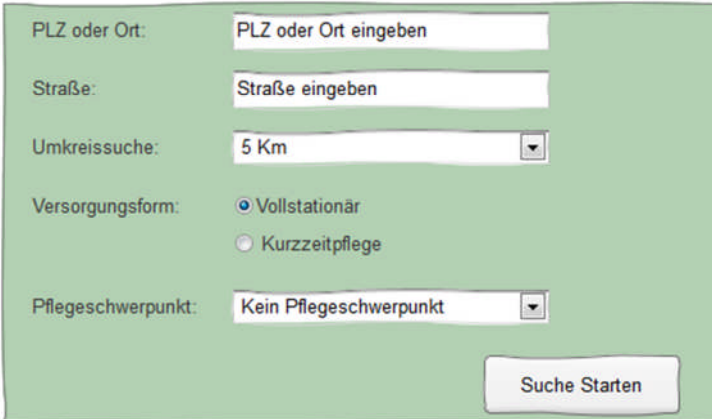
Um sich stärker von anderen Pflegeheimfindern abzuheben, könnte eine Bewertungsstruktur durch die Webnutzer eingeführt werden. Erfahrungsberichte und ein Punktesystem ähnlich wie bei Qype oder verschiedenen Onlineshops würden der abstrakten Bewertung nach Preis und Entfernung zusätzlich eine persönliche Note verleihen. Denkbar wäre zudem, dass nicht nur eine Gesamtbewertung stattfindet, sondern dass sogar einzelne Bereiche wie das Essen oder Freundlichkeit der Mitarbeiter vom Nutzer bewertet werden könnten. Diese Bewertung würde bei Bedarf ebenfalls in die Berechnung der Rangliste einfließen.

Aktuell wird in der Presse die Unterbringung von Senioren in Pflegeheimen diskutiert. Ein Themenschwerpunkt bezieht sich darauf, dass sich Senioren ein kostengünstiges Pflegeheim im Ausland suchen. Es haben sich in verschiedenen Ländern der Welt einige Pflegeheime auf die deutschsprachigen Kunden spezialisiert. Aus diesem Kontext heraus könnte die Anwendung auch solche speziellen Pflegeheime aufnehmen oder sogar eine eigene Rubrik „Pflege im Ausland“ einführen.

Um eine größere Anzahl von Webnutzern auf die Webseite zu locken, könnten außerdem ein Lexikon für Fachbegriffe und ein Ratgeber für aktuelle Fragen zum Thema Pflege angeboten werden. Dieser Ratgeber könnte zusätzlich um ein Forum erweitert werden, damit die Webnutzer sich gegenseitig beraten und austauschen können.

A Anhang

Screenshots Mock-Up der Oberfläche für den Workshop



The screenshot shows a search interface titled "neue Suche" in the top right corner. The interface is set against a light green background and contains the following elements:

- PLZ oder Ort:** A text input field with the placeholder text "PLZ oder Ort eingeben".
- Straße:** A text input field with the placeholder text "Straße eingeben".
- Umkreissuche:** A dropdown menu currently displaying "5 Km".
- Versorgungsform:** Two radio button options: "Vollstationär" (which is selected) and "Kurzzeitpflege".
- Pflegeswerpunkt:** A dropdown menu currently displaying "Kein Pflegeswerpunkt".
- Suche Starten:** A button located at the bottom right of the form.

Abbildung 45 Mock-up Suchmaske

Ihre Auswahl ergab 6 Treffer.

Beinflussen Sie das Suchergebnis durch Detailangaben in Ausstattung und Prioritäten.

Ausstattung wählen

Hauseigene Küche Fernsehen auf Zimmer
 Einbettzimmer ▼
 Tierhaltung erlaubt
 Internet auf Zimmer

Prioritäten wählen

Suchergebnis 1

Name	Adresse	Kontakt	Übereinstimmung	Entfernung km
Einrichtung 1	Hennebergstr 2b, 22393 Hamburg	Tel: 040/52732... email: einrichtung1@entenhausen.de	●	0,5
Einrichtung 2	Hennebergstr 2b, 22393 Hamburg	Tel: 040/52732... email: einrichtung2@entenhausen.de	●	1,2
Einrichtung 3	Hennebergstr 2b, 22393 Hamburg	Tel: 040/52732... email: einrichtung3@entenhausen.de	●	1,9
Einrichtung 4	Hennebergstr 2b, 22393 Hamburg	Tel: 040/52732... email: einrichtung4@entenhausen.de	●	2,6
Einrichtung 5	Hennebergstr 2b, 22393 Hamburg	Tel: 040/52732... email: einrichtung5@entenhausen.de	●	3,2
Einrichtung 6	Hennebergstr 2b, 22393 Hamburg	Tel: 040/52732... email: einrichtung6@entenhausen.de	●	4,9

Abbildung 46 Mock-up Suchergebnis tabellarisch



Name: Einrichtung 1
 Telefon: 040/52732XY
 Email: einrichtung1@entenhausen.com
 Adresse: ABC-Straße 14-28
 22393 Hamburg

Versorgungsstruktur:

- Vollstationäre Pflege
- Kurzzeitpflege

Pflegeschwerpunkt:

- kein Pflegeschwerpunkt

Speisenangebot:

- Diätkost
- hauseigene Küche
- Schonkost

Sächliche Ausstattung

- Größe: 120 Betten
- Zimmergröße: 25qm - 35qm
- Telefon auf Zimmer
- Frisör

Heimentgelt:

Pflegestufe	täglich gesamt €	monatlich gesamt €	Leistungsbetrag der Pflegekasse €	verbleibender Eigenanteil pro Bewohner €
0	73,78	2.213,40	---	2.213,40
1	88,52	2.655,60	1.023,00	1632,60
2	105,94	3.178,20	1.279,00	1.899,20
3	126,57	3.797,10	1.550,00	2.247,10

Pflegenote:

- 1,5 [Bericht anzeigen](#)

Freizeitgestaltung und Beschäftigungsangebot:

- Ausflüge werden alle 2 Monate angeboten
- Enternung nächste Haltestelle: 2 Gehminuten
- Tierhaltung erlaubt

Abbildung 47 Mock-up Detailansicht

Ihre Auswahl ergab 6 Treffer.

Beeinflussen Sie das Suchergebnis durch Detailangaben in Ausstattung und Prioritäten.

Ausstattung wählen
Prioritäten wählen

Suchergebnis 1

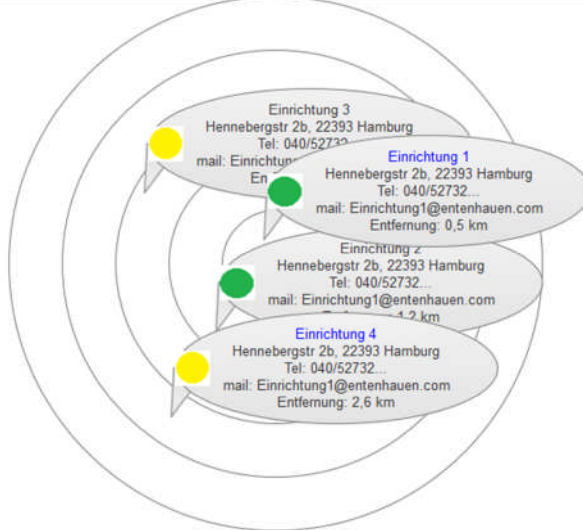


Abbildung 48 Mock-up Suchergebnis grafisch

B Glossar

Leistungs- und Qualitätsvereinbarung(LQV)

In der LQV ist die Struktur der Einrichtung, der Bedarf der versorgten Klientel, Art und Umfang der Leistung und personelle und sächliche Ausstattung.

Pflege-Qualitätssicherungsgesetz (PQsG)

Das PQsG regelt und erweitert die Qualitätsanforderungen des SGB XI, die an Pflegeeinrichtungen gestellt werden.

Medizinischer Dienst der Krankenkassen (MDK)

Der MDK hat vielfältige Aufgaben in vielen medizinischen Bereichen. Im pflegerischen Bereich sind die Aufgaben unter anderem: Berater in medizinischen Versorgungsfragen, Begutachtungen für die Pflegeversicherung, Pflegequalität durch Prüfungen in Pflegeeinrichtungen sichern, Betroffene in eine Pflegestufe einstufen.

Betreuung nach §87b SGB XI

Regelt die Ausbildung von Betreuungskräften sowie die leistungsgerechten Zuschläge an eine Pflegeeinrichtung, die Betreuungskräfte beschäftigt und Personen umsorgt, die einen erheblichen allgemeinen Betreuungsbedarf haben (beispielsweise Menschen mit demenzbedingten Fähigkeitsstörungen).

Snoezelraum

Ein Ruheraum für Bewohner der Pflegeeinrichtung, der alle Sinne ansprechen und beruhigen soll. Die Möblierung ist so gewählt, dass eine möglichst beruhigende Atmosphäre entsteht. Dazu trägt auch eine beruhigende Geräuschkulisse bei.

C Tabellenverzeichnis

Tabelle 1 Übersicht Pflegeheimdaten der Landesvereine	11
Tabelle 2 Ergebnisübersicht Usability-Analyse vorhandener Pflegeheimfinder.....	27

D Abbildungsverzeichnis

Abbildung 1 Bedürfnispyramide des Nutzers (Knoflicek, 2012)	13
Abbildung 2 Navigation/Menü Position - Nutzererwartung.....	16
Abbildung 3 Suchmaske AOK-Pflegeheimfinder	19
Abbildung 4 Suchergebnis AOK-Pflegeheimfinder.....	20
Abbildung 5 Suchmaske BKK-Pflegefinder	21
Abbildung 6 Suchergebnis BKK-Pflegefinder	22
Abbildung 7 Suchmaske Pflegekompass	23
Abbildung 8 Suchergebnis Pflegekompass	24
Abbildung 9 Suchmaske vdek.....	25
Abbildung 10 Suchergebnis vdek	26
Abbildung 11 Webstatistik AOK-Pflegenavigator	30
Abbildung 12 Anwendungsfalldiagramm.....	41
Abbildung 13 fachliches Datenmodell	47
Abbildung 14 fachliches Komponentendiagramm.....	48
Abbildung 15 twitter-bootstrap fluid-grid-system.....	49
Abbildung 16 technisches Komponentendiagramm.....	51
Abbildung 17 controllers Innensicht	53
Abbildung 18 findeEinrichtungen Innensicht.....	54
Abbildung 19 bewerteEinrichtungen Innensicht	56
Abbildung 20 models Innensicht.....	57
Abbildung 21 ER-Diagramm	59
Abbildung 22 Aufteilung der Oberfläche in kleine Templates.....	61
Abbildung 23 HTML-Template eingestreuter Scala-Code.....	61
Abbildung 24 Beispiel HTML für Bootstrap fluid-grid	62
Abbildung 25 wenig Kontrast Warning-Box.....	62
Abbildung 26 viel Kontrast Warning-Box	62
Abbildung 27 Codebeispiel einer Modelklasse	63
Abbildung 28 Play! Methoden zum Datenbankzugriff.....	64
Abbildung 29 Einsatz von reinem SQL	64
Abbildung 30 Klassendiagramm controllers	65
Abbildung 31 Klasse FormglobalTransport	65
Abbildung 32 Play! Umwandlung von Form in Model-Objekt	66
Abbildung 33 Umwandlung String <<Ortfeld>> in Plz- oder Stadt-Entität	66
Abbildung 34 Klassendiagramm FormDetailTransport	67
Abbildung 35 Klassendiagramm findeEinrichtungen	67

Abbildung 36 Klassendiagramm BewerteEinrichtungenAnwendungsfall.....	68
Abbildung 37 berechne Distanz Übereinstimmung	69
Abbildung 38 Klassendiagramm EinrichtungenAuswertung	69
Abbildung 39 Code - Filtern der Einrichtungen.....	70
Abbildung 40 Autovervollständigung.....	70
Abbildung 41 Suchergebnis.....	71
Abbildung 42 Detailansicht	72
Abbildung 43 Filtern des Suchergebnisses.....	73
Abbildung 44 start und stop fakeApp	76
Abbildung 45 Mock-up Suchmaske.....	82
Abbildung 46 Mock-up Suchergebnis tabellarisch.....	83
Abbildung 47 Mock-up Detailansicht.....	84
Abbildung 48 Mock-up Suchergebnis grafisch.....	85

E Literaturverzeichnis

Literaturverzeichnis

Alexa, 2012. *Web Information Company*. [Online]

Available at: <http://www.alexacom/siteinfo/aok-pflegeheimnavigator.de#>

[Zugriff am 26.10 Oktober 2012].

Broschart, S., 2011. *Suchmaschinenoptimierung & Usability Website-Ranking und Nutzerfreundlichkeit verbessern*. 2. Hrsg. Poing: Franzis Verlag GmbH.

Connor, J. O., 2012. *Pro HTML5 Accessibility*. 1. Hrsg. s.l.:Apress.

Düweke, E. & Rabsch, S., 2011. *Erfolgreiche Websites*. 1. Hrsg. Bonn: Galileo Press.

Ebert-Yom, M., 2010. *Usability als Erfolgsfaktor*. 1. Hrsg. Berlin: Cornelsen Verlag Scriptor GmbH.

Europäisches Komitee für Normung, 1995. *Ergonomische Anforderung für Bürotätigkeiten mit Bildschirmgeräten Teil 10: Grundsätze Dialoggestaltung*. Brüssel: s.n.

Grabitz, I., 2010. *Welt.de*. [Online]

Available at: <http://www.welt.de/wirtschaft/webwelt/article10717665/Internet-befluegelt-soziale-Aktivitaeten-der-Senioren.html>

[Zugriff am 25 September 2012].

Institut für Medien- und Konsumentenforschung, 2012. *CN Typologien*. [Online]

Available at: <http://www.imuk.de/cn/cntypologien/cnwebtypologie.html>

[Zugriff am 26 Oktober 2012].

Klie, T., 2005. *Pflegeversicherung*. 7. Hrsg. Hannover: Vinzentz Network.

Knappschaft, kein Datum *der-Pflegekompass*. [Online]

Available at: [http://www.der-](http://www.der-pflegekompass.de/lang_de/nn_158630/DE/3_Service/daten_impress/daten_impress_vorspann.html)

[pflegekompass.de/lang_de/nn_158630/DE/3_Service/daten_impress/daten_impress_vorspann.html](http://www.der-pflegekompass.de/lang_de/nn_158630/DE/3_Service/daten_impress/daten_impress_vorspann.html)

[Zugriff am 19. Oktober 2012].

Knoflicek, D., 2012. *User Experience - Die Bedürfnispyramide*. [Online]

Available at: <http://imb.donau-uni.ac.at/planet-ecommerce/user-experience>

[Zugriff am 07. Februar 2013].

König, J., 2007. *Was die PDL wissen muss*. 3. Hrsg. Hannover: Schlütersche GmbH & Co. KG.

König, J., 2007. *Was die PDL wissen muss*. 3. Hrsg. Hannover: Schlütersche GmbH & Co. KG.

Nielsen, J. & Loranger, H., 2006. *Prioritizing Web Usability*. 1. Hrsg. Berkeley, CA: Pearson Education.

Pflegenoten, S., 2011. *Pflegenoten*. [Online]

Available at: http://www.pflegenoten.de/Veroeffentlichungen_Uebersicht.gkvnet

[Zugriff am 24 September 2012].

- play!-Framework documentation, unbekannt. *Play 2.0 documentation*. [Online]
Available at: <http://www.playframework.com/documentation/2.0.3/Home>
[Zugriff am 22.02.13 Februar 2013].
- Schulz, U., 2004. Web-Usability für Silversurfer. *mer. medien + erziehung*. 48. Jahrgang, Heft 4/04, April, pp. 18-22.
- Spitzenverband Pflegenoten, 2012. *Pflegenoten*. [Online]
Available at: <http://www.pflegenoten.de/Bewertungssystematik.gkvnet>
[Zugriff am 24 September 2012].
- Stiegert, H., 2011. *Modernes Webdesign mit CSS*. 1. Hrsg. Bonn: Galileo Press.
- User-Centered Design may be considered a practice, f. c., 2009. *User-Centered Design may be considered a practice, field, craft,*, Woburn: ACM international conference on Design of communication .
- Winter, M. H.-J., 2008. Pflegeheime auf dem Weg zu Institutionen des Sterbens?. *Gesundheit und Gesellschaft Wissenschaft*, Oktober, pp. 15-22.
- Yan, P. & Guo, J., 2010. *The Research of Web Usability Design*, Chongqing, China: IEEE.
- @fat, @. a., unbekannt. *getting bootstrap*. [Online]
Available at: <http://twitter.github.com/bootstrap/getting-started.html>
[Zugriff am 22. Februar 2013].

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____