



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Fachher Syed

Fernseher Plus - Ein Informations- und
Kommunikationssystem für Senioren

Fachher Syed
Fernseher Plus - Ein Informations- und
Kommunikationssystem für Senioren

Bachelorarbeit eingereicht im Rahmen der Bachelorarbeitprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 28. Februar 2013

Fachher Syed

Thema der Bachelorarbeit

Fernseher Plus - Ein Informations- und Kommunikationssystem für Senioren

Stichworte

Ambient Assisted Living, Soziale Software, Software-Ergonomie, barrierefreie Benutzeroberfläche, barrierefreie Benutzerschnittstelle

Kurzzusammenfassung

In dieser Bachelorarbeit wird eine Prototyp für ein Informations- und Kommunikationssystem entwickelt, welches auf die Nutzung von Senioren ausgerichtet ist. Schwerpunkt bildet hierbei die Erstellung einer barrierefreien Benutzerschnittstelle.

Fachher Syed

Title of the paper

Fernseher Plus - an information and communication system

Keywords

Ambient Assisted Living, Sociale software, Usability, Accessible UI, Accessible user interface

Abstract

In this thesis a prototype for an information and communication system is developed, which is focused on the use of senior citizens. Focus here is on the creation of an accessible user interface.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
1 Einführung	7
1.1 Projekt Fernseher Plus	8
1.2 Ziel der Arbeit	8
1.3 Aufbau der Arbeit	8
2 Analyse	10
2.1 Problemstellung	10
2.2 Eigenschaften von Senioren-gerechter Software	19
2.3 Use-Cases	22
2.3.1 Vorstellung der Charaktere	22
2.3.2 Szenarien	24
2.3.3 Funktionale Anforderungen	25
2.4 Vergleichbare Arbeiten	28
2.4.1 Smart Senior @ Home	28
2.5 Fazit	30
3 Design	31
3.1 System Überblick	31
3.2 Architektur	32
3.2.1 Client Server Architektur	32
3.2.2 3 Schichten Architektur	33
3.3 System Entwurf	35
3.3.1 Komponentendiagramm	35
3.3.2 Laufzeitdiagramm	37
3.3.3 Datenmodell	40
3.3.4 Plugin System	41
3.4 Client Entwurf	42
3.4.1 Model-View-Controller Entwurfsmuster	43
3.4.2 Client Komponente	44
3.4.3 Aufbau der grafischen Benutzeroberfläche	45
3.4.4 Interaktion mit der grafischen Benutzeroberfläche	46

3.5	Weitere Eigenschaften des Systems	49
3.5.1	Sicherheit	49
3.5.2	Kommunikation	49
3.5.3	Darstellung	50
3.6	Fazit	50
4	Realisierung	51
4.1	Verwendete Technologien	51
4.1.1	Verwendete Hardware	51
4.1.2	Verwendete Software	52
4.2	Server Realisierung	53
4.2.1	Java EE 6	53
4.2.2	3 Schichten Architektur	54
4.2.3	Umsetzung der REST Schnittstellen	56
4.2.4	Umsetzung der 3 Schichten Architektur	57
4.2.5	Plugin System	57
4.2.6	Wetter Plugin	58
4.2.7	Facebook Plugin	59
4.2.8	Sicherheit	60
4.3	Client Realisierung	64
4.3.1	Sprachaufzeichnung	64
4.3.2	Grafische Benutzerschnittstelle	66
4.4	Fazit	67
5	Zusammenfassung	68
	Literaturverzeichnis	70

Abbildungsverzeichnis

2.1	Prozentuale Internetnutzung durch Senioren	16
2.2	Wofür nutzen Senioren das Internet	17
2.3	Use Case Diagram	22
2.4	Smart Senior Plus Architektur	28
3.1	Grob Design	32
3.2	3-Schichten Architektur	33
3.3	Client Server Aufteilung	34
3.4	Komponentendiagramm	36
3.5	Nachrichten abrufen und versenden	38
3.6	Fachliches Datenmodell	41
3.7	Model View Controller Architektur	43
3.8	Client Komponente	44
3.9	Pageflow Diagram Nachrichten versandt	46
3.10	Statemachine Nachrichten versandt	47
3.11	Aufbau eines Untermenüpunktes	48
4.1	Infrarot Fernbedienung	52
4.2	Portalseite des Fernseher Plus Servers	53
4.3	Client Wetterabruf	58
4.4	Client des Seniors	64
4.5	Wizard Klassendiagramm	66
4.6	Umsetzung MVC	67

1 Einführung

Unsere Gesellschaft verändert sich. Während wir immer älter werden, stagniert die Geburtenrate in den westlichen Industrienationen. Dies führt dazu dass die Alterspyramide in einigen Jahren auf dem Kopf stehen wird. Die Frage die sich hierbei stellt ist, wie lange dies durch das aktuelle Sozialsystem getragen werden kann. Forscher gehen davon aus, dass bis zum Jahr 2060 die Zahl der über 65-Jährigen ein Drittel der gesamten Bevölkerung in Deutschland ausmachen wird. Dann würde es mehr alte Menschen als junge Menschen geben. Wie kann eine zahlenmäßig unterlegene junge Arbeitergeneration, die ältere Generation tragen? Eine Möglichkeit wären Intelligente Assistenzsysteme, die den Senioren bei alltäglichen Aufgaben unterstützen und eventuell das Sozialsystem entlasten. Die Rede ist hier von Ambient Assisted Living. Intelligente Assistenzsysteme sollen älteren Menschen und benachteiligten Menschen situationsabhängig und unaufdringlich unterstützen.

In den letzten Jahren hat die Bedeutung von Informations- und Kommunikationstechnologien massiv zugenommen. Während Computer früher nur für die Arbeit verwendet wurden, trifft man sie heute allgegenwärtig an. Ältere Menschen haben in dieser modernen Welt oft das Nachsehen, weil sie nicht mit diesen Technologien aufgewachsen sind. Wer einmal einen älteren Mensch an einem modernen Ticketschalter für U-Bahn Tickets gesehen hat, sieht diesen Menschen schnell die Ratlosigkeit im Gesicht an. Oft wird die Ratlosigkeit begleitet durch Angst etwas kaputt zu machen.

„das Bild des älteren Menschen in unserer Gesellschaft ist auch heute noch durch Feststellungen von Isolation und Vereinsamung, von Abhängigkeit und Hilfsbedürftigkeit charakterisiert“ (Lehr,1991)

Ein weiterer Punkt ist die Vereinsamung im Alter unter der viele ältere Menschen leiden. Die häufigsten Ursachen hierfür sind das gleichaltrige Freunde an Altersschwäche sterben, Familienangehörige wegziehen oder die Scheidung vom Ehepartner. Um dieser Einsamkeit entgegenzuwirken, müssen bestehende Sozialkontakte gepflegt und neue Sozialkontakte geknüpft werden. Dies ist jedoch schwierig, wenn die Kontaktdaten verlegt, verloren oder nicht mehr aktuell sind. Soziale Software¹ könnte den Senior dabei helfen. Sozial Software dienen dazu, Gemeinschaften aufzubauen und zu pflegen, und zwar in aller Regel über das Internet (Wikipedia, 2013b).

¹Social Software ist ein Modewort für Software, die der menschlichen Kommunikation und der Zusammenarbeit dient

1.1 Projekt Fernseher Plus

Fernseher Plus ist ein Projekt der Firma Silpion. Ziel des Projektes ist es ein System zu entwickeln, mit dessen Hilfe Senior mit ihren Freunden und Verwandten in Kontakt bleiben können. Hierbei sollen die Senioren von Familienangehörigen und vom Pflegepersonal (im weiteren als Caretaker bezeichnet) unterstützt werden. Dafür stellt der Fernseher Plus Server eine Portal-Seite bereit. Das Projekt Fernseher Plus ging aus dem Projekt Abendsonne hervor. 2006 wurde das Projekt Abendsonne initiiert und 2012 in den Namen Senior Plus umbenannt. Während beim Projekt Abendsonne die Realisierung einer Fernbedienung und einer Set-Top-Box im Vordergrund stand, rückt in dem Projekt Fernseher Plus der Backend Server in den Vordergrund.

1.2 Ziel der Arbeit

Die vorliegende Arbeit beschreibt die Entwicklung einer Informations- und Kommunikationsplattform für Senioren, mit dessen Hilfe Senioren mit Ihren Freunden und Verwandten in Kontakt bleiben können, sowie sich im Internet informieren.

Hierzu soll ein Client entwickelt werden, welcher über eine barrierefreie Benutzeroberfläche verfügen soll. Der Client Prozess läuft hierbei auf einem herkömmlichen Multimedia PC ab. Um die Funktionalität des Clients zu testen, soll zusätzlich ein Server entwickelt werden. Dies ist notwendig, da der Client kaum Anwendungslogik enthalten soll. Die Interaktion mit dem Client Systems soll über eine herkömmliche Fernseher Fernbedienung erfolgen. Zum aufzeichnen von Sprachaufzeichnung, welche für die Beantwortung von Nachrichten genutzt werden, wird ein Mikrofon verwendet. Die Funktionalität des Servers soll durch Plugins erweiterbar sein.

1.3 Aufbau der Arbeit

Die Arbeit ist in 5 Kapitel unterteilt, welche die Struktur und den Entwicklungsablauf dieser Bachelorarbeit wiedergeben.

Wir befinden uns im ersten Kapitel, welches die Einleitung umfasst.

Das 2 Kapitel umfasst die Ergebnisse der Anforderungsanalyse. Hier wird aufgezeigt, welche Funktionen das System bieten soll und worauf dabei die Schwerpunkte gelegt werden.

Das 3 Kapitel legt dem Leser das Design der entwickelten Software nahe. Darin werden die verschiedenen Möglichkeiten für den Anwender aufgeschlüsselt und durch entsprechende Diagramme dargestellt.

Das 4 Kapitel befasst sich mit der Realisierung der Software. Die verschiedenen genutzten Softwarebibliotheken werden dargestellt und deren Nutzungsgründe erläutert.

Das abschließende 5 Kapitel beinhaltet das Fazit sowie einen Ausblick. Hier wird sich kritisch mit der Arbeit auseinander gesetzt und die unterschiedlichen Schritte werden bewertet. Zudem werden verschiedene Erweiterungs- sowie Einsatzmöglichkeiten dargestellt, welche sinnvoll zu realisieren wären, aber durch den begrenzten Zeitrahmen nicht Teil dieser Arbeit geworden sind.

2 Analyse

2.1 Problemstellung

Demographische Entwicklung

Infolge des Rückgangs der Kinder- und Säuglingssterblichkeit und des allgemeinen Rückgangs an Infektionskrankheiten durch moderne hygienische Maßnahmen und Medikamente, nahm die Bevölkerungszahl in Deutschland seit Mitte des vergangenen Jahrhunderts bis heute von ca. 22 Mio. auf ca. 85 Mio. zu. Die durchschnittliche Lebenserwartung stieg von ca. 35 Jahren auf ca. 70 Jahre an. Die Geburtenrate (Fertilität) nahm jedoch ab und beträgt aktuell 1,4 Kindern pro Frau. Für die Reproduktion wären jedoch 2,1 Kinder erforderlich (Umweltbundesamt, 2013).

Dadurch kommt es in Deutschland sowie in anderen Industrienationen zu einer Überalterung der Gesellschaft. So wird erwartet, dass die Zahl der über 65-Jährigen in Deutschland von 2008 bis 2060 von einem Fünftel auf ein Drittel der Gesamtbevölkerung steigt. Es wird also erwartet, dass es bald mehr alte Menschen als Junge gibt (Umweltbundesamt, 2013).

Beschreibung der Zielgruppe

Die Generation 50plus oder auch Best Ager genannt beschreibt die Gruppe der Menschen, welche älter sind als fünfzig. Nicht alle Menschen über fünfzig sind gleich. Das Alter lässt sich in drei Lebensphasen unterteilen:

1. Der Jungsenior ab 50 denkt berufsorientiert, sein Lebensrhythmus wird von der beruflichen Tätigkeit bestimmt.
2. Der Senior ab 60 ist häufig Privatier, denkt familienorientiert und genießt das Leben mit den Enkeln. Allerdings werden in dieser Phase auch noch häufig Ehen geschieden.
3. Ab 70 rückt die Gesundheit ins Zentrum des Lebens.

Der Alltag eines 52-jährigen Berufstätigen unterscheidet sich zum Beispiel stark von dem einer 87-jährigen Rentnerin.

Auch sind die Senioren von Heute nicht mit zukünftigen Senioren vergleichbar. Denn anders als die heutige Generation (Digital Natives) ist die ältere Generation (Digital Immigrants) nicht mit dem Computer aufgewachsen. Es handelt sich hierbei also um ein aussterbendes Forschungsfeld (Müller, 2006).

Pflegeheim oder Zuhause

In Deutschland leben zurzeit etwa 85 Millionen Menschen. Davon sind 24,6 Millionen über 65 Jahre alt. Lediglich 6 Prozent dieser Altersgruppe sind pflegebedürftig. (Hertwig, 2009) Das heißt das 94 % der über 65-Jährigen in keinsten Weise durch ihr Alter beeinträchtigt ist. Sie gehören zu den sogenannten » jungen Alten « und genießen, soweit es ihre finanziellen Mittel zulassen, unbeschwerte Jahre.

2,5 Millionen Menschen die in Deutschland leben sind pflegebedürftig (Stand 2011). Zwei Drittel der Pflegebedürftigen werden zu Hause versorgt. Davon erhielten ca. 1 Million Pflegebedürftige ausschließlich Pflegegeld, das bedeutet, sie wurden in der Regel zu Hause allein durch Angehörige gepflegt. Weitere 576 000 Pflegebedürftige lebten ebenfalls in Privathaushalten. Bei ihnen erfolgte die Pflege jedoch zusammen mit oder vollständig durch ambulante Pflegedienste. 30 % (743 000 Personen) wurden in Pflegeheimen vollstationär betreut.

Familiäres Pflegepotenzial geht jedoch aufgrund der demografischen Entwicklung zurück (Rückgang der Zahl der Kinder pro Familie) weshalb die Pflege zunehmend durch Pflegepersonal erfolgen muss. Das Pflegepersonal ist jedoch Heute schon ausgelastet, weshalb nach alternativen gesucht wird.

Mit zunehmendem Alter sind Menschen in der Regel eher pflegebedürftig. Während bei den 70- bis unter 75-Jährigen „nur“ jeder Zwanzigste (5 %) pflegebedürftig war, wurde für die ab 90-Jährigen die höchste Pflegequote ermittelt: Der Anteil der Pflegebedürftigen an allen Menschen dieser Altersgruppe betrug dabei 58 %.

Die Mehrheit der Pflegebedürftigen waren mit 83 % ältere Menschen im Alter von 65 Jahren und älter. Die Zahl der Pflegebedürftigen steigt weiter. (Pfaff, 2013).

Pflegebedürftige Menschen, die nicht mehr im Stande dazu sind ihr Heim zu verlassen, können viele alltägliche Aufgaben die mit dem verlassen des Heims verbunden sind, nicht selbstständig bewältigen. Hierzu zählen das abschließen von Rechtsgeschäften, sowie Behörden- oder Bankangelegenheiten. Sie sind dadurch auf fremde Hilfe angewiesen.

„Senioren wollen auch wenn sie einmal Pflege und Betreuung benötigen weiterhin selbstständig wohnen“ (Burkhart Brückner, 2006)

Aus Umfragen geht hervor, dass 90% der deutschen Bevölkerung lieber in Frieden zu Hause sterben möchten als in einem Altersheim. Die Realität sieht jedoch anders aus. 70 % der über 65-jährigen sterben im Krankenhaus.

Ein Umzug in ein Pflegeheim sollte deshalb nur dann überlegt werden, wenn alle anderen Möglichkeiten ausgeschöpft sind (Bernhard Reindl, 2009).

Die Singularisierung, das Leben ohne festen Partner, trägt in doppelter Hinsicht dazu bei, häusliche Pflege zu erschweren, wenn nicht sogar unmöglich zu machen. Auch ältere Menschen leben immer häufiger alleine. In einigen Fällen ist der Partner verstorben, in zunehmendem Maße ist inzwischen auch eine frühere Ehescheidung dafür verantwortlich. Werden allein lebende Menschen pflegebedürftig, müssen sie in der Regel sehr früh professionelle Hilfe in Anspruch nehmen. Oft jedoch ist der entstehende Aufwand allein mit dieser Unterstützung nicht mehr zu bewältigen. Hinzu kommt das Gefühl der Einsamkeit und des Alleinseins, so dass der Umzug in ein Heim gerade in einem solchen Fall eine sinnvolle Alternative sein kann (Hertwig, 2009).

Wirtschaftlicher Gesichtspunkt

Das Nettoeinkommen der 45- bis 55-Jährigen liegt bei monatlich 3383 Euro, bei den 55- bis 65-Jährigen bei 3015 Euro. Beide Werte liegen deutlich über dem Bevölkerungsdurchschnitt von 2771 Euro monatlich. Auch beim Vermögen trumpfen die Älteren auf. Etwa 60 Prozent des Vermögens aller Haushalte - das entspricht 2200 Milliarden Euro - liegt heute in Händen der über 50-Jährigen (Müller, 2006).

Aufgrund ihres durchschnittlichen Monatseinkommens sind die Senioren zu einer umworbenen Zielgruppe der Werbebranche geworden (Hertwig, 2009).

Der Eintritt in das Rentenalter ist aber oft mit materiellen Einbußen verbunden. Gesellschaftlich gesehen, wird jedoch die Kaufkraft älterer Menschen in den kommenden Jahrzehnten wachsen. Dementsprechend wird sich das Warenangebot und die Werbestrategien der Unternehmen wandeln (Burkhart Brückner, 2006).

Vereinsamung im Alter

Verbringt man während der Arbeitszeit in der Regel sehr viel Zeit mit Arbeitskollegen, fallen diese sozialen Kontakte im Alter weitgehend weg. Des Weiteren kommt es zu einer Veränderung der sozialen Rolle der Älteren ((Faltermaier, 1992) zitiert nach (Leupolz, 2010)). Teilweise fühlen sich ältere Menschen dann nicht mehr gebraucht und nutzlos. Oft entsteht deshalb ein negatives Selbstbild. Eine neue persönliche Identität, die über die Funktion in

einem Betrieb hinausgeht, muss gefunden werden und stellt eine Herausforderung in dieser Lebensphase dar (Zoch, 2008) zitiert nach (Leupolz, 2010).

Funktionierende Großfamilien, in denen ein kranker alter Mensch versorgt und betreut wird, finden wir heutzutage fast nur noch in ländlichen Gegenden. Die Rahmenbedingungen familiären Lebens haben sich verändert. Um einen gewissen Lebensstandard aufrecht erhalten zu können, sind viele Familien zur Erwerbstätigkeit beider Partner gezwungen. Da bleibt keine Zeit, um die Pflege eines alten Menschen im erforderlichen Umfang leisten zu können (Hertwig, 2009).

Leben die Kinder, inzwischen berufstätig und mit eigener Lebensplanung, als Singles, schaffen diese es in den seltensten Fällen, ihre pflegebedürftigen Eltern zu betreuen. Neben der eigenen Berufstätigkeit bleibt dazu einfach keine Zeit mehr. Viele Kinder leben inzwischen viel zu weit weg von ihren Eltern, um deren Pflege übernehmen zu können (Hertwig, 2009).

Stadt-Umland-Wanderung

Ein Phänomen welches innerhalb der Stadtregionen beobachtet werden kann, ist die Stadt-Umland-Wanderung. Junge Menschen ziehen aufgrund einer Ausbildung in die Stadt und beruflich etablierte Paare und Familien mit Kindern ziehen ins Umland. So kommt es dazu dass in einigen Städten weit aus mehr ältere Menschen leben als jüngere (Vergreisung einer Stadt) und umgekehrt (Verjüngerung einer Stadt). Diese Wanderungsprozesse vollziehen sich in unterschiedlichen Regionen, Kommunen und Stadtteilen in unterschiedlichem Ausmaß und Tempo (umweltbundesamt, 2013). Die berufliche Zukunft kann also dazu führen, dass Kinder wegziehen und ihre Eltern allein zurücklassen.

Körperliche Einschränkung

Die öffentliche Diskussion war lange von der Annahme geprägt, dass im Alter Verluste und Einbußen vorherrschen - diese Position wird als das Defizitmodell des Alters bezeichnet. Demgegenüber spricht man von einem Kompetenzmodell des Alters, wenn die Chancen und Entfaltungsmöglichkeiten betagter Menschen betont werden. Das Kompetenzmodell rückt zurzeit in den Vordergrund der Fachdiskussion (Burkhart Brückner, 2006).

Typische Konflikte alter Menschen betreffen oft ihr Verhältnis zu ihrem Körper und der Gesundheit sowie die Autonomie ihrer Selbstversorgung und Mobilität, aber auch das Arbeits- und Freizeitverhalten und schließlich ihre soziale Beziehungen, insbesondere zu nahen Verwandten wie Ehepartnern und Kindern (Burkhart Brückner, 2006).

Ein Problem ist die Zahl der Demenzerkrankung ab dem 60-zigsten Lebensjahr. Diese beträgt bei den Frauen 47% und bei den Männern 29%.

Abbau der geistige Fähigkeiten

Es wurde lange Zeit behauptet, dass ältere Personen Informationen schwerfälliger verarbeiten können. Neuere Forschung ergeben jedoch ein komplexeres Bild, so müsse man zwischen der fluide Intelligenz und der kristallinen Intelligenz unterscheiden (Weinert 1991 zitiert nach Burkhart Brückner). Die „fluide Intelligenz“ beziehe sich auf das Arbeitsgedächtnis, das abstrakte Denken, das psychische Tempo und die Kombinationsfähigkeit und vermindere sich im Alter. Die „kristalline Intelligenz“ beziehe sich hingegen auf das Allgemeinwissen, den Wortschatz, die Aufmerksamkeit und die Urteilskraft und bliebe unabhängig vom Alter bestehen. Zudem wurde darauf hingewiesen, dass die soziale Kompetenz nicht beeinträchtigt würde und das biologische Alter nur ein Einflussfaktor unter mehreren anderen sei (z.B. Bildung, Beruf, Gesundheit), welche die geistige Leistungsfähigkeit beeinflussen können (Burkhart Brückner, 2006).

Das bedeutet das ältere Menschen aufgrund ihrer physischen Abnahme an Mobilität verlieren können, mental aber keinerlei Einbußen haben.

Physiologischer Abbauprozess

Die physiologischen Abbauprozesse im Alter betreffen vor allem die Verschlechterung der Seh- und Hörfähigkeit, die Motorik und die kognitiven Fähigkeiten (Bmfsfj, 2013). Die rückläufigen physischen und kognitiven Fähigkeiten führen zu Schwierigkeiten im Umgang mit dem Computer. Neben der Verschlechterung der Seh- und Hörfähigkeit nimmt auch der Tastsinn sowie die Koordination der Finger kontinuierlich ab. Dadurch fällt es dem älteren Menschen schwer Aktionen wie das hoch und runter scrollen eines Fensters oder der Doppelklick mit der Maus durchzuführen.

Computer Anxiety

Die Probleme der heutigen älteren Generation (Kriegsgeneration) liegen vor allem darin begründet, dass sie in einer anderen Zeit groß geworden sind. Man bedenke, dass der technische Fortschritt damals erst im Aufbau war. Radio und Fernsehen wurden gerade entwickelt (Gesund älter werden: Stefan Schrank). Viele Rentner haben nie gelernt, mit einem Computer umzugehen und mit dem Alter steigen die Berührungängste weiter (taz, 2013).

Denn erst Mitte der neunziger Jahre (1995) war der Computer für Jedermann erschwinglich und selbstverständlich (Nielsen, 1994).

Studien über Computer Anxiety bei jüngeren und älteren Menschen haben belegt, dass ältere Mensch deutlich mehr Angst vor Computern haben (Phang u. a., 2006) als jüngere Menschen.

Senioren geraten oft in Panik, wenn sie mit einer Fehlermeldung konfrontiert werden, welche sie in den meisten Fällen nicht verstehen. Computer Profis wissen wie sie mit dieser Fehlermeldung umzugehen haben und Kinder ignorieren diese einfach (Phang u. a., 2006).

Deshalb sollten Fehlermeldung leicht und verständlich für den Senior sein. Es sollten Fehlermeldung wie „Bitte kontaktieren sie den Support“ statt „Illegal Operation“ erscheinen, da „Illegal“ von den Senioren als etwas sehr schlimmes angesehen wird. Durch benutzerfreundliche Fehlermeldung kann die Angst vor Computer gemildert werden. Fehlermeldung sollen also konstruktiv und freundlich sein.

Unsicherheiten können auch durch Support in Form von Computerhilfe von Vertrauenspersonen oder Bereitstellung von geeigneter Hardware gemildert werden. Auch Computer Schulung können Computer Ängste nehmen (Phang u. a., 2006).

Senioren haben einen erhöhten Bedarf an Sicherheit und Privatsphäre. Während die heutige Generation leichtsinnig mit ihren privaten Daten umgeht, zögern Senioren diese herauszugeben weil sie um ihre Privatsphäre fürchten. Auch Online Geschäftsabwicklung werden von den Senioren kaum genutzt, da sie der Sache nicht trauen. Deshalb sollte die Wahrung der Privatsphäre sowie die Sicherheit einen hohe Stellenwert bei der Entwicklung von Informationssystemen für Senioren haben (Hans Rudolf Schelling, 2010).

Viele Produkthanbieter berücksichtigen die Senioren nicht als Konsumenten und verwenden Slangs oder englische Begriffe welche die Senioren meistens nicht verstehen. Dies liegt meistens daran, dass viele Produkte von jüngeren Menschen für jüngere Menschen gemacht worden.

Forschung haben ergeben, dass es eine Beziehung zwischen Selbstwirksamkeitserwartung(SWE) und Computer Anxiety gibt (Phang u. a., 2006).

Unter Selbstwirksamkeitserwartung versteht man die eigene Erwartung, aufgrund eigener Kompetenzen gewünschte Handlungen erfolgreich selbst ausführen zu können. Ein Mensch, der daran glaubt, selbst etwas zu bewirken und auch in schwierigen Situationen selbstständig handeln zu können, hat demnach eine hohe Selbstwirksamkeitserwartung (Wikipedia, 2013a).

Das bedeutet das Menschen mit einer höheren Selbstwirksamkeitserwartung motivierter sind etwas neues zu lernen, als solche mit einer niedrigen Selbstwirksamkeitserwartung. Jemand mit einer niedrigen Selbstwirksamkeitserwartung kann durch das meistern einer

Aufgabe (z.B. am Computer) seinen Selbstwirksamkeitserwartung steigern und dadurch seine Computer Ängste mindern (Corporate-eye, 2013).

Aktuelle Internetnutzung durch Senioren

Nur elf Prozent aller Internetnutzer sind Senioren (ARD-ZDF, 2013). Die Hauptgründe hierfür sind unter anderem Sicherheitsbedenken sowie einer Kosten-Nutzen-Kalkulation. Die Kosten sind etwa das Erlernen der Fähigkeiten zur Nutzung eines Computers. Diese werden von den meisten Senioren als immens hoch angesehen im Vergleich zum Nutzen. Auch wenn es für den einen oder anderen eine relative hohe Bereicherung darstellen könnte. Wie man aus der unten stehenden Tabelle in der roten Markierung erkennen kann, nutzen von den Senior ab 60 Jahren nur 39% das Internet 2.1. Wie aus der Abbildung 2.2 hervorgeht nutzen Senioren das Internet hauptsächlich um Sachen zu Suchen und E-Mails zu schreiben.

2 Internetnutzer in Deutschland 1997 bis 2012
gelegentliche Nutzung, in %

	1997	2000	2003	2006	2007	2008	2009	2010	2011	2012
Gesamt	6,5	28,6	53,5	59,5	62,7	65,8	67,1	69,4	73,3	75,9
Männer	10,0	36,6	62,6	67,3	68,9	72,4	74,5	75,5	78,3	81,5
Frauen	3,3	21,3	45,2	52,4	56,9	59,6	60,1	63,5	68,5	70,5
14-19 J	6,3	48,5	92,1	97,3	95,8	97,2	97,5	100,0	100,0	100,0
20-29 J	13,0	54,6	81,9	87,3	94,3	94,8	95,2	98,4	98,2	98,6
30-39 J	12,4	41,1	73,1	80,6	81,9	87,9	89,4	89,9	94,4	97,6
40-49 J	7,7	32,2	67,4	72,0	73,8	77,3	80,2	81,9	90,7	89,4
50-59 J	3,0	22,1	48,8	60,0	64,2	65,7	67,4	68,9	69,1	76,8
ab 60 J	0,2	4,4	13,3	20,3	25,1	26,4	27,1	28,2	34,5	39,2
in Ausbildung	15,1	58,5	91,6	98,6	97,6	96,7	98,0	100,0	100,0	100,0
berufstätig	9,1	38,4	69,6	74,0	78,6	81,8	82,3	82,4	87,0	90,7
Rentner/nicht berufstätig	0,5	6,8	21,3	28,3	32,0	33,6	34,7	36,4	45,0	44,7

Basis: Bis 2009: Deutsche ab 14 Jahren in Deutschland (2009: n=1 806, 2008: n=1 802, 2007: n=1 822, 2006: n=1 820, 2003: n=1 955, 2000: n=3 514, 1997: n=15 431). Ab 2010: Deutschsprachige Bevölkerung ab 14 Jahren (2012: n=1 800, 2011: n=1 800, 2010: n=1 804).

Quelle: ARD-Onlinestudie 1997, ARD/ZDF-Onlinestudien 2000-2012.

Abbildung 2.1: Prozentuale Internetnutzung durch Senioren
Quelle:(ARD-ZDF, 2013)

9 Onlineanwendungen nach Geschlecht und Alter 2012
mindestens einmal wöchentlich genutzt, in %

	Gesamt	Frauen	Männer	14-29 J	30-49 J	50-69 J	ab 70 J
Suchmaschinen nutzen	83	80	86	96	83	74	58
senden/empfangen von E-Mails	79	78	81	81	83	74	69
zielgerichtet bestimmte Angebote/informationen suchen	61	61	62	69	64	53	38
einfach so im Internet surfen	43	41	45	59	40	35	22
Onlinecommunitys nutzen	36	37	36	75	30	12	4
Homebanking	32	31	33	26	39	28	24
Gesprächsforen/Chatten	26	22	29	56	19	8	4
Instant Messaging	18	16	21	36	15	8	4
überspielen/herunterladen von Dateien	17	10	23	30	15	10	3
Kartenfunktionen nutzen	17	15	19	27	15	10	12
Onlinespiele	16	12	19	23	16	7	14
„Apps“ auf Mobilgeräten nutzen, um ins Internet zu gehen	15	12	18	32	11	5	2
Musikdateien aus dem Internet	12	12	13	29	9	1	0
Video/TV zeitversetzt	11	10	12	20	11	5	2
live im Internet Radio hören	10	7	14	12	13	7	5
RSS-feeds/Newsfeeds	10	8	11	15	9	6	1
Onlineauktionen/Versteigerungen	7	6	8	6	9	5	3
andere Waren bestellen/Onlineshopping	7	7	7	8	8	5	1
live im Internet fernsehen	7	6	9	11	7	4	8
Buch- und CD-Bestellungen	6	6	5	7	6	4	1
andere Audiodateien aus dem Internet	4	2	6	8	4	2	—
Kontakt-/Partnerbörsen	4	4	4	7	3	3	5
Audio/Radiosendungen zeitversetzt	3	1	5	2	4	2	3
Audiopodcasts	2	1	2	2	2	1	0
Videopodcasts	2	0	3	2	2	1	—

Basis: Deutschsprachige Onlinenutzer ab 14 Jahren (n=1366).

Quelle: ARD/ZDF-Onlinestudie 2012.

Abbildung 2.2: Wofür nutzen Senioren das Internet

Quelle:(ARD-ZDF, 2013)

Digitalen Agenda für Europa

Die „Digitalen Agenda für Europa“ ist eine Agenda welche im Mai 2010 von der Europäischen Kommission angenommen wurde und dessen Ziel es ist Älteren, sehbehinderten oder in ihren kognitiven Fähigkeiten eingeschränkten Personen durch die Beherrschung von Informations- und Kommunikationstechnologien ein würdiges und unabhängiges Leben und hohe Lebensqualität zu ermöglichen (digitalechancen, 2013).

Digitale Kluft

Die digitale Kluft bezeichnet die Ausgrenzung bestimmter Personengruppen aus dem gesellschaftlichen Leben aufgrund mangelnder Nutzung aktueller Informationstechnologien.

Computer Unterstützung durch andere Personen

Computer Unterstützung bezieht sich auf die Unterstützung durch andere Personen in Form von Ressourcen oder Wissen welches die Senioren bei ihrer Computeraktivität unterstützt.

Hilfe welche durch Freunde und Verwandte geleistet wird, beeinflusst stark die Nutzung von Informationstechnologie durch Senioren (Phang u. a., 2006). Die Internetnutzung des eigenen unmittelbaren sozialen Umfelds (Lebenspartner/in, Kinder, Enkel, Geschwister, Freundeskreis) hängt mit der eigenen Nutzung zusammen (Hans Rudolf Schelling, 2010).

Es hat sich gezeigt, dass Senioren die Hilfe von gleichaltrigen vorziehen, dies liegt darin begründet dass jüngere Menschen ungeduldig sind und oft Jargons benutzen welche die Senioren nicht verstehen (Phang u. a., 2006).

Gründe weshalb Senioren das Internet nicht nutzen

Allgemeines Interesse an Technik und die Einschätzung der Bedienung technischer Geräte als schwierig oder leicht erweisen sich als sehr starke Prädiktoren bezüglich Internetnutzung. In einer Studie gaben Senioren welche das Internet nicht nutzen, vor allem die Kompliziertheit der Benutzung (71 %) und den Aufwand für das Erlernen (60 %) als stärkste Gründe zur Nichtnutzung des Internets an.

Ein weiterer Grund ist das die Nutzung von Informationssystemen zu einer Veränderung des gewohnten Weges zur Erlangung von Dienstleistungen führt. Senioren legen z.B. bei einer Überweisung ihre Sicherheit in die Hände des Bankangestellten welcher sich um die Korrektheit kümmert. Senioren legen viel Wert in Vertrautes und bleiben diesen gern Treu.

Mittels einer repräsentativen telefonischen Befragung an 1105 Personen im Alter von 65 Jahren und älter in der Schweiz wurden die Gründe der (Nicht)Nutzung des Internets durch Senioren erfasst. Bei dieser Befragung gaben 60% der Befragten an, dass Sie Sicherheitsbedenken (bezüglich Datensicherheit und Internetkriminalität) haben. Anschaffungskosten für Hardware stellen nur für 33% einen Hinderungsgrund da. 66% aller Befragten sind nicht bereit für das Internet Geld auszugeben. Gesundheitliche Gründe, spielen eine eher unwichtige Rolle (27% haben damit Probleme).

Für die tatsächliche Nutzung spielen auch die Bildung und das Einkommen eine erhebliche Rolle, weniger aber für das Interesse an Internetanwendungen (Hans Rudolf Schelling, 2010).

Warum das Internet nützlich für Senioren sein kann

Ressourcenersparnis in Form von Zeit und Geld sind starke Motivatoren für die Nutzung von Informationssystemen. So kann durch die Nutzung von Online Banking der Gang zur Bank gespart werden. Dies führt dazu das der Senior auf weniger fremde Hilfe angewiesen ist und dadurch unabhängiger ist.

Nachrichten verschicken vs Telefonieren

In der Studienarbeit "Ämericans and Text Messaging" hat Clive Thompson 2277 Junge Menschen im Alter von 18 und älter bezüglich der Gründe für die Bevorzugung der SMS gegenüber dem telefonieren befragt. Die Mehrheit der Befragten ziehen eine Textnachricht aus folgenden Begründungen einem persönlichen Telefongespräch vor:

- eine Telefongespräch ist zu persönlich
- Textnachrichten sind distanzierter
- kosten nicht viel Überwindung
- geben mehr Zeit zum Antworten und Überlegen(asynchron)

Nach Meinung von Clive Thompson wird das Telefon von E-Mail und SMS verdrängt. (Kuhlmann, 2013)

Dies spricht dafür dass der Senior auch diese Form der Kommunikation nutzen sollte, weil er sonst von der Gesellschaft abgeschnitten wird (siehe: Digitale Kluft).

2.2 Eigenschaften von Senioren-gerechter Software

Ältere Menschen können aufgrund ihrer verminderten physischen und mentalen Fähigkeiten, nicht mit jüngeren Menschen mithalten was das Erlernen von Computerwissen betrifft.

Deshalb sollten Programme, welche für Senioren bestimmt sind eine Reihe von Eigenschaften besitzen.

Folgende Eigenschaften sollte die Benutzerschnittstelle aufweisen, damit das System von Senioren bedient werden kann (Bernhard Reindl, 2009).

- Das Display sollte groß, beleuchtet und kontrastreich sein
- Schriftfarben und Hintergrundfarben sollten anpassbar sein.

- Schriftgröße, Bildsymbole für die Bedienung und Farbhintergrund sollten anpassbar sein.
- Die Tasten sollten möglichst groß und gut fühlbar sein

In der Nachfolgenden Tabelle werden eine Reihe von Richtlinien aufgelistet und ihrer Relevanz, welche bei der Gestaltung von grafischen Benutzerschnittstellen (GUI) für Senioren zu berücksichtigen sind.

Zudem sollten komplizierte Mausektionen wie das Dragging und Doppelclicks vermieden werden. Selbst erfahrene ältere Computerbenutzer haben mit diesen Schwierigkeiten, dies liegt in der Abnahme des Tastsinns begründet. Bei der Erstellung von Multi-Level-Navigation sollte darauf geachtet werden, dass diese eine geringe Tiefe besitzen sollte. Außerdem sollten so wenig Elemente wie nötig auf der Benutzeroberfläche angezeigt werden. Es sollten Symbole mit großen Bildern angezeigt werden mit einer zusätzlichen Beschriftung (Research on hci). Die wichtigsten Faktoren bei der Gestaltung der Benutzeroberfläche sind die Lesbarkeit, Bedienbarkeit und die Verständlichkeit. Seniorengerechte Produkte sollten so gestaltet sein, dass die Basisfunktionalität leicht erlernt werden kann (A.Hart, 2013).

Richtlinien Beschreibung	Relevanz %
Formulierung: Anweisung sollten im Aktive formuliert sein	100.0
Scrollen: Vermeide automatisches scrollen und stelle Scroll Icons bereit	100.0
Maus: Vermeide Doppelklicks	100.0
Beschriftung: Benutze Groß- und Kleinschreibung für Fließtext und Titel sollte in Großbuchstaben stehen	97.2
Ausrichtung: Texte sollten links ausgerichtet sein	97.2
Gestalt: Klare Sätze ohne Zweideutigkeit	97.2
Menüs: Nicht zu viele Pull Down und Cascading Menüs verwenden	97.2
Verständlichkeit: Sätze sollten leicht und verständlich formuliert sein. Für Fachwörter sollte ein Glossar zur Verfügung stehen	91.7
Schriftart: Benutze Schriftarten, dessen Buchstaben nicht zu dicht aneinander stehen	91.4
Farben: Vermeide das die Farbe Gelb, Blau und Grün beieinander stehen	90.6
Hintergrund: Benutze eine dunkle Schriftfarbe auf hellem Hintergrund und umgekehrt	88.9
Konsistente Erscheinung: Das Design und die Navigation sollte auf jeder Seite gleich sein	77.8
Organisieren: Benutze ein Standardformat; längere Dokumente sollten in Absätze aufgeteilt werden	75.0
Navigation: Benutze explizit Schritt für Schritt Navigation; einfach und ziel-führend	72.2
Hilfe & Informationen: Biete Tutorials auf Webseiten an und Kontaktinformationen	78.1
Symbole & Knöpfe: Benutze große Knöpfe; Text sollte wenn möglich mit durch Symbole ergänzend werden	69.4
Text Alternativen: Biete Text Alternativen für alle Medientypen	67.6
Illustration & Fotos: Benutze nur für den Text relevante Fotos	63.9
Schriftbreite: Benutze eine mittlere oder kräftige Schrift	50.0
Schriftgröße: Benutze eine Schriftgröße von 12 bis 14 für den Textkörper	44.4
Inhaltsübersicht: Benutze ein Inhaltsverzeichnis um zu zeigen wie die Seite aufgebaut ist	38.9
Querverweise: Benutze Symbole mit Text für Querverweise (Hyperlinks)	25.0

Tabelle 2.1: Kriterien Senioren Benutzerschnittstelle
Quelle: (A.Hart, 2013)

2.3 Use-Cases

In dieser Arbeit soll ein Informations- und Kommunikationssystem für Senior erstellt werden. Dazu werden zunächst Anwendungsfälle vorgestellt, die das zu entwerfende System genauer beschreiben.

Aus diesen exemplarischen Anwendungsfällen werden anschließend die Anforderungen und Voraussetzungen für das System abgeleitet.

2.3.1 Vorstellung der Charaktere

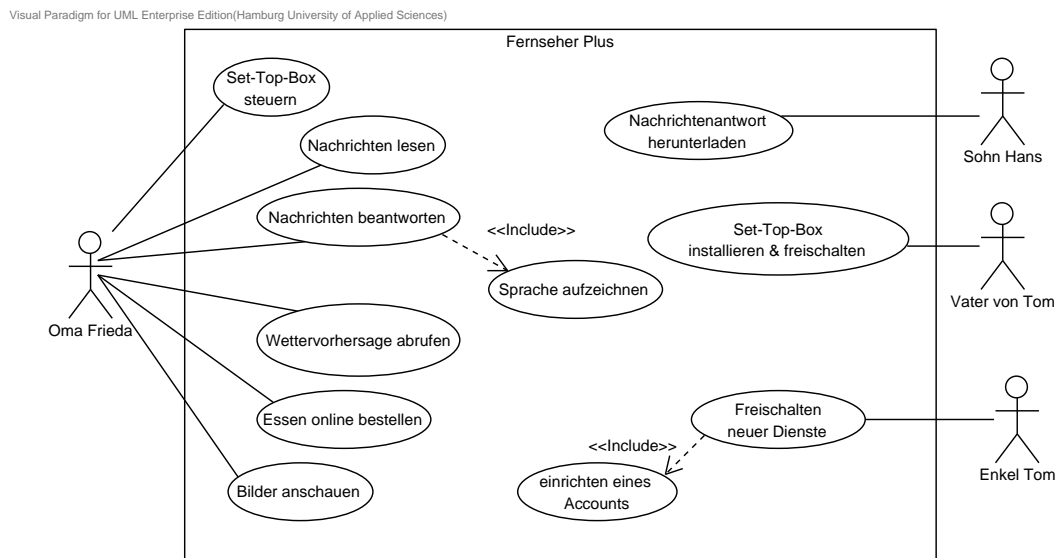


Abbildung 2.3: Use Case Diagram

Oma Frieda

Oma Frieda ist eine alleinstehende Frau im Alter von 75 Jahren. Sie wohnt alleine in einer zwei Zimmer Wohnung in einem ländlichen Gebiet. Ihr Mann ist vor einigen Jahren gestorben weswegen sie alleine lebt. Ihr Wohnblock besitzt einen Internetzugang, welchen Oma Frieda lange Zeit nicht nutzte, da Sie hierfür keine Verwendung sah. Seit neuestem besitzt Oma Frieda eine Set-Top-Box welche mit ihrem Fernseher und dem Internet verbunden ist. Die Installation der Box wurde von ihrem Enkel Tom vorgenommen. Mit der neuen Installation kann Oma Frieda mit ihrem Enkel Tom, Freunden und weiteren Verwandten kommunizieren.

Sie kann sich informieren wie das Wetter bei ihr und ihren Freund wird und kann sich ihr Essen für die kommenden Tage vorbestellen z.B. bei Essen auf Rädern.

Enkel Tom

Tom ist 16 Jahre alt und nutzt leidenschaftlich gern seinen Computer zum spielen und zum surfen im Internet. Einmal in der Woche besucht er seine Oma, welche 70 Kilometer von seinem Wohnort entfernt wohnt. Tom würde gern öfter mit seiner Oma Informationen austauschen, doch besitzt diese nur ein Telefon. Telefonieren findet Tom „Uncool“. Er würde seiner Oma viel lieber auf die Facebook Pinnwand schreiben, aber leider besitzt seine Oma keine Internet, da sie hierfür bisher keine Verwendung vorsah und Angst vor Technik. Als Tom von der „Fernseher Plus Box“ hörte, dachte er sich, das dies doch die ideale Lösung sei, um mit seiner Oma über das Internet zu kommunizieren. Seine Oma braucht keine Erfahrung im Umgang mit dem Computer zu haben und kann trotzdem von den Vorteilen des Internets profitieren. Also bat Tom seinen Vater darum, seiner Oma so eine Box zu kaufen. Das einrichten des Facebook Accounts würde er selbstverständlich übernehmen.

Sohn Hans

Hans ist 45 Jahre alt und ist der Sohn von Oma Frieda. Er lebt seit 6 Jahren in Mexiko. Aufgrund der hohen Telefonkosten kann er nicht so oft mit seiner Mutter telefonieren. Generell telefoniert Hans ungern und Briefe schreiben findet er altmodisch. Dennoch würde er gern mit seiner Mutter in Kontakt bleiben.

Administrator

Der Fernseher Plus Server wird von einem Administrator verwaltet und gepflegt. Der Administrator kann das System um neue Funktionalitäten erweitern. Hierzu werden von ihm Plugins installiert.

Caretaker

Ein Caretaker kann ein Verwandter bzw. ein Pfleger sein, welcher sich um die Verwaltung von Diensten auf der Fernseher Plus Plattform kümmert.

2.3.2 Szenarien

Diese fiktiven Szenarien beschreiben die zu Grunde liegende Vision in beispielhaften Anwendungsfällen.

Oma Frieda kommuniziert

Oma Frieda steht Morgens um sieben Uhr auf und macht sich das Frühstück. Nebenbei lässt sie ihren Fernseher laufen, damit sie sich nicht so allein fühlt. Plötzlich ertönt ein Signal und auf dem Fernseher erscheint die Meldung „Sie haben Post“. Oma Frieda nimmt ihre Fernbedienung zur Hand und wählt anzeigen aus. Dadurch wird ihr der Inhalt der Nachricht angezeigt. Oma Frieda freut sich, denn ihr Sohn Hans aus Mexiko hat sie zu seiner Hochzeit eingeladen. Sie möchte sofort eine Antwort zurücksenden. Oma Frieda klickt auf „Antworten“ und spricht in das eingebaute Mikrofon ihrer Fernbedienung. Das die Nachricht von einem Social Network, E-Mail Provider oder RSS Feed kommt weiß und interessiert die Oma nicht.

Hans erhält eine Antwort

Hans der Sohn von Oma Frieda entdeckt einen neuen Eintrag auf seiner Socialnetwork Pinnwand. Die Nachricht ist von Oma Frieda, die lediglich einen Link enthält. Hans klickt auf den Link und hört sich die Antwort von Oma Frieda an. Hans erfährt so dass Oma Frieda zu seiner Hochzeit kommen wird.

Oma Frieda informiert sich

Oma Frieda weiß absolut nicht was sie zu der Hochzeit ihres Sohnes anziehen soll, da sie nicht weiß wie das Wetter in Mexiko wird. Oma Frieda geht in das Telefonbuch der Set-Top-Box und sucht Hans Kontakt raus. Mit einem weiteren Knopfdruck wird das aktuelle Wetter von Hans abgerufen. Jetzt weiß Oma Frieda dass sie ein Kleid anziehen kann, da es zu dieser Jahreszeit sehr heiß in Mexiko ist.

Oma Frieda interagiert

Früher hat Oma Frieda ihr Essen telefonisch bei „Essen auf Rädern“ bestellt. Damit Oma Frieda ihr Essen bestellen konnte musste sie erstmal wissen was es denn überhaupt gibt.

Dafür gab es jede Woche den ausgedruckten Speiseplan, welchen Oma Frieda ständig verlegte. Dann war es so dass sie nur an den Werktagen während der Bürozeit ihre Bestellung aufgeben konnte. Jetzt kann Oma Frieda dank der Hilfe von „Fernseher Plus“ zu jeder Zeit ihre Bestellung aufgeben. Hierzu wählt Oma Frieda im Menü „Essen“ aus. Im folgenden Menü wählt sie ihr Essen für die folgenden Tage aus. Der Fernseher Plus Server versucht die Vorlieben von Oma Frieda zu ermitteln. Sollte Oma Frieda einmal vergessen ihr Essen auszuwählen, so wird dies durch den Server selbstständig erledigt.

Installation der Box

Toms Vater richtet die Set-Top-Box bei Oma Frieda ein. Hierzu verbindet er die Set-Top-Box mit dem Fernseher und dem Internet. Nach der Installation der Set-Top-Box muss diese noch registriert werden. Hierzu geht er mit seinem Smartphone auf die Fernseher Plus Service Webseite und gibt den mitgelieferten Code ein. Damit ist die Set-Top-Box auch schon eingerichtet. Jede Set-Top-Box besitzt einen eindeutigen Key mit der sie identifiziert werden kann.

Pflegepersonal und Verwandte

Oma Frieda möchte gern Bilder auf ihrer Box empfangen. Zur Zeit ist dieses Feature jedoch nicht aktiviert. Oma Frieda bittet ihren Enkel Tom, ihr dieses Feature freizuschalten. Mit wenigen Klicks auf der Fernsehen Plus Service Webseite kann Tom die neuen Features für seine Oma freischalten. Er erledigt dies unterwegs mit seinem Mobiltelefon.

System um neue Funktionalitäten erweitern

Die Hersteller der Set-Top-Box haben erkannt dass Senioren gern Spiele spielen. Also wurden ein paar Programmierer damit beauftragt ein Plugin zu schreiben, welche das System um ein Spiel erweitert. Der Systemadministrator pflegt dann das neue Plugin in das bestehende System ein.

2.3.3 Funktionale Anforderungen

Nachdem mehrere wesentliche Szenarien einen Ausblick gegeben haben, was das System einmal leisten sollte, werden in diesem Abschnitt zu bestimmten Szenarien die funktionalen Anforderungen abgeleitet und nach Wichtigkeit sortiert. Dabei werden die funktionalen Anforderungen beschrieben, welche den Funktionsumfang des Systems darstellen.

F1 - Multi-Level-Navigation

Die Navigation durch die Menüs der Set-Top-Box erfolgt in einem gerichteten zyklischen Graphen.

- Der Startknoten des Graphen ist das Hauptmenü
- durch Auswahl eines Menüpunktes im Hauptmenü wird vom Startknoten zu einem adjazenten(benachbarten) Knoten traversiert.
- durch drücken des Zurück-Knopfs der Fernbedienung wird vom aktuellen Knoten zum Vorgängerknoten traversiert, sofern es eine Kante zu diesem gibt. Sollte es keine Kante zum Vorgängerknoten geben, so wird zu einem vordefinierten Knoten traversiert. Dies kann z.B. der Fall sein, wenn im Vorgängerknoten eine Datei übertragen wurde. Das traversieren zu diesem Knoten würde zu einer erneuten Versendung der Datei führen.
- durch drücken des Hauptmenü-Knopfs wird vom aktuellen Knoten zum Startknoten traversiert.

F2 - Steuerung der Set-Top-Box mit der Fernbedienung

Die Set-Top-Box wird mithilfe einer gewöhnlichen Fernseher Fernbedienung bedient.

Der Senior soll

- mit den Pfeiltasten der FB die Set-Top-Box steuern können
- mit einem Knopfdruck auf der FB etwas auswählen können
- mit einem Knopfdruck ein Menü-Level zurückgehen können
- mit einem Knopfdruck auf der FB zurück ins Menü kehren, egal in welchem Untermenü er sich gerade befindet

F3 - Abruf des aktuellen Wetters eines Kontaktes

Der Senior soll in der Lage sein, sich über das aktuelle Wetter zu informieren. Sei es über das eigene oder das seiner Kontakte.

Der Senior kann das aktuelle Wetter

- von seinem Ort abrufen
- von den Orten seiner Kontakte

F4 - Nachrichten abrufen

Der Senior kann Nachrichten vom Fernseher Plus Server abrufen.

F5 - Nachrichten per Sprachaufzeichnung zu beantworten

Nachrichten werden mittels einer Sprachaufzeichnung beantwortet.

Der Senior soll durch die Verwendung eines Mikrofons seine Stimme aufzeichnen und das Ergebnis der Aufzeichnung als Antwort versenden können.

F6 - Plugins hinzufügen

Programmierer können das System um weitere Funktionalitäten erweitern. Dies geschieht mithilfe von Plugins.

Der Systemadministrator soll

- Plugins hinzufügen können.
- Plugins entfernen können.

F7 - Dienste hinzufügen und entfernen durch Caretaker

Sollte der Bedarf bestehen Nachrichten von einem Sozialnetzwerk wie z.B. Facebook abzurufen bzw. zu diesem zu versenden, so kann hierfür ein Dienst von einem Caretaker freigeschaltet werden.

Pflegepersonal und Verwandte sollen

- Dienste freischalten können.
- Dienste sperren können.

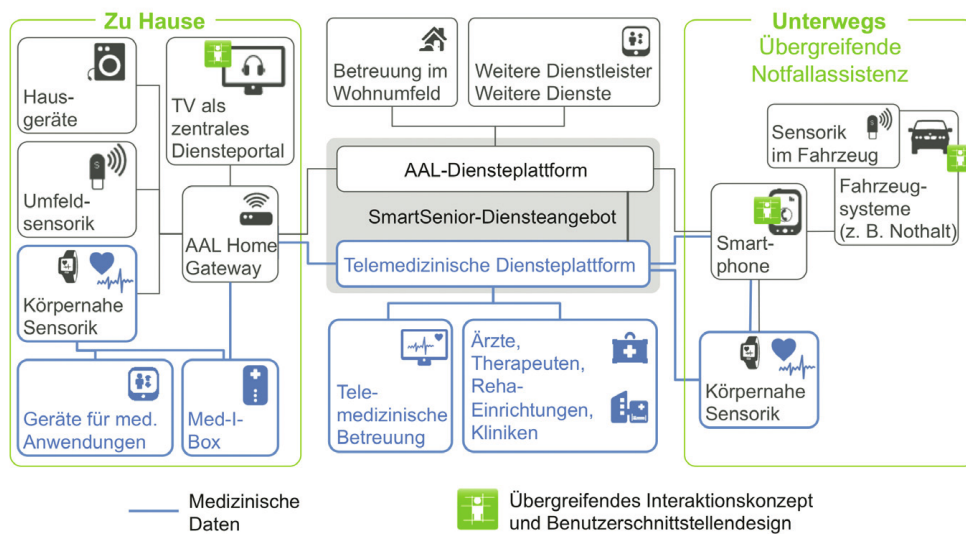


Abbildung 2.4: Smart Senior Plus Architektur
(Quelle: (SmartSenior, 2013))

2.4 Vergleichbare Arbeiten

2.4.1 Smart Senior @ Home

SmartSenior@Home ist ein Projekt welches durch das Bundesministerium für Bildung und Forschung gefördert wird. An diesem Projekt beteiligen sich kleine und mittelständische Unternehmen, Gesundheitsdienstleister sowie Forschungseinrichtung wie das Fraunhofer Institute. Es ist das größte Projekt seiner Art in Deutschland. Das Ziel dieses Zusammenschluss war es mit Technologien, Infrastrukturen und Dienstleistungen die Grundlagen für zuverlässige und einfach zu bedienende Produkte zu entwickeln und das ältere Menschen länger selbstständig im häuslichen Umfeld leben können.

Das Projekt wurde im Jahr 2009 initiiert und erreichte im Jahr 2012 seinen letzten Meilenstein. Die SmartSenior-Dienstplattform besitzt einen modularen Aufbau. So können individualisierte Dienstangebote erstellt werden wodurch Kosten gespart werden. Es wurden bereits während des Projekts Feldstudien durchgeführt um festzustellen ob das Produkt von den Senior angenommen wird. Die gewonnenen Erkenntnisse zeigen, dass für die Senior die Zuverlässigkeit des Systems sowie eine einfache Bedienung die wichtigsten Akzeptanzkriterien sind. Im Unterschied zu vielen anderen Projekten wurden die entwickelten Systeme größtenteils nicht nur im Labor, sondern unter Alltagsbedingungen von Seniorinnen und Se-

nieren zu Hause getestet. In diesem Projekt werden die folgenden 3 Lebensbereiche adressiert:

- Länger selbstständig im häuslichen Umfeld leben
- Sicher unterwegs sein
- Gesund werden und bleiben

Genau wie in dieser Arbeit dient der Fernseher im Projekt SmartSenior@Home als zentraler Kommunikationskanal. Die Form der Bedienung des Fernsehers unterscheidet sich jedoch von dieser Arbeit. Während in dieser Arbeit die Metapher der klassischen Fernseher Fernbedienung als Bedienelement vertreten wird, wird in dem Projekt SmartSenior@Home ein Webpad verwendet. Es wird dort die Idee des „Second Screen“ aufgegriffen, der die Nutzung eines zweiten Bildschirms parallel zum laufenden Fernsehprogramm beschreibt. Die Frage welches Bedienkonzept sich durchsetzen wird, kann nur die Zeit zeigen.

SmartSenior@Home bietet folgende Dienstleistungen an:

- Ein TV-integriertes Serviceportal, das den zentralen Zugang zu den medizinischen und nichtmedizinischen SmartSenior-Diensten ermöglicht.
- Eine sensorbasierte Erkennung von typischen Alltagssituationen in der häuslichen Umgebung, die kritische Zustände bemerkt und ein Assistenz-Center informiert. Nutzer können Einrichtung und Geräte überwachen und steuern.
- Ein System zur Erfassung und Übertragung von Vitaldaten, bei dem klinisch zugelassene Messgeräte Gewicht, Blutdruck und EKG bestimmen und die Daten sicher an das Telemedizinzentrum übertragen.
- Eine TV-integrierte Lösung zur Videokommunikation mit Freunden und Bekannten, mit Dienstleistern und medizinischen Einrichtungen in hoher Bild- und Tonqualität; sie gestattet unter anderem Videokonferenzen mit bis zu sechs Teilnehmern. Die Lösung unterstützt auch das gemeinsame Fernsehen räumlich voneinander getrennter Senioren.
- Die unmittelbare Anbindung an ein Assistenz-Center, das über weitreichende Möglichkeiten zum Remote Management verfügt und dadurch die Installation, Aktualisierung und Deinstallation von Anwendungen und den Umgang mit der Technik für die Senioren erheblich vereinfacht.
- Ein Partnerfinder, der die Senioren bei der sozialen Vernetzung unterstützt, indem er Personen anhand ihrer Hobbys und Interessen findet. Es werden auch Fernsehgewohnheiten berücksichtigt, welche die SmartSenior Set-Top-Box automatisch erfasst

(Quelle: (SmartSenior, 2013))

Eine flächendeckende Bereitstellung des SmartSenior@Home System könnte Aufgrund der hohen Kosten sowie rechtliche und regulatorischen Gründen scheitern.

2.5 Fazit

Das Fernseher Plus System soll Oma Frieda dazu dienen, mit ihren Freunden und Verwandten in Kontakt zu bleiben. Hierzu besitzt Oma Frieda eine Set-Top-Box welche sich mit dem Server von Fernseher Plus verbindet. Für die Bedienung der Set-Top-Box verwendet Oma Frieda eine Fernbedienung, die ihrer alten Fernseher Fernbedienung ähnelt. Mit der Set-Top-Box kann Oma Frieda Nachricht abrufen und diese mit einer Sprachaufzeichnung beantworten. Zudem kann Oma Frieda sich über das aktuelle Wetter von verschiedenen Standorten informieren. Die Fernbedienung soll über sieben Tasten verfügen, die aus einem Steuerkreuz und 3 Spezialtasten besteht.

Im folgenden Kapitel wird untersucht, wie ein derartiges System zu entwerfen ist.

3 Design

Dieses Kapitel dient der Ausarbeitung eines Konzepts für die spätere Realisierung des Fernseher Plus Systems. Hierzu werden für die im vorherigen Kapitel gestellten funktionalen Anforderung Lösung entwickelt. Dafür wird zunächst eine Architektur bestimmt und im Anschluss daran ein Entwurf.

Dynamische Vorgänge innerhalb des Systems werden mittels Sequenzdiagramme und Zustandsautomaten dargestellt.

3.1 System Überblick

In der Abbildung 3.1 wird ein grober Überblick über das Gesamtsystem gegeben.

In dieser Abbildung wird auf der linken Seite der Client-Aufbau beim Senioren gezeigt und auf der rechten Seite der Fernseher Plus Server.

Der Clientaufbau besteht aus einer Set-Top-Box, einem Fernseher und einer Fernbedienung. Der Client ist mit dem Fernseher Plus Server über eine Breitbandverbindung verbunden.

Der Serveraufbau besteht aus einem Server welcher mit einer Datenbank verbunden ist.

Caretaker legen fest welche Services die Set-Top-Box des Seniors in Anspruch nehmen kann. Hierzu gehen Caretaker auf die Portal Seite des Fernseher Plus Servers. Caretaker können Verwandte oder das Pflegepersonal sein.

Freunde und Verwandte können E-Mails oder Nachrichten von Sozialen Netzwerken (z.B. Facebook) an den Fernseher Plus Server senden. Der Fernseher Plus Server hält die Nachrichten für die Set-Top-Box des Seniors zum Abruf bereit.

Die Verbindung zwischen Client und Server ist durch das Transport Layer Security (TLS) Protokoll abgesichert.

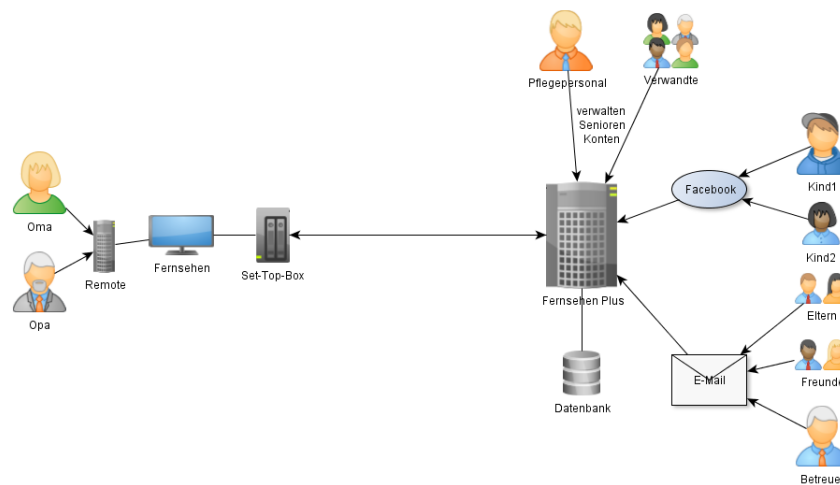


Abbildung 3.1: Grob Design

3.2 Architektur

3.2.1 Client Server Architektur

Für das System Fernseher Plus soll eine Client Server Architektur entwickelt werden.

Der Grundgedanke der Client Server Architektur ist die Trennung zwischen dem Dienstanbieter(Server) und dem Dienstnehmer(Client).

Bei der Client Server Architektur stellen ein oder mehrere Client-Prozesse Anfrage an den Server-Prozess der diese Anfrage abarbeitet und gegebenenfalls eine Antwort sendet.

Es soll zwei verschiedene Arten von Clients geben. Einmal die Set-Top-Box des Seniors und dann die verschiedenen Browser(Chrome,Firefox,Internet Explorer) welche auf die Web-Schnittstelle des Fernseher Plus Server zugreifen um Dienste und Einstellung für die Set-Top-Box festzulegen.

Clients: Set-Top-Box, Webbrowser

Server: Fernseher Plus

Eine Client Server Architektur kann auf zwei Arten realisiert werden. Entweder als Zwei-Schichten-Client-Server Architektur oder eine Drei-Schichten-Client-Server Architektur. Für diese Arbeit soll eine Drei-Schichten-Client-Server Architektur (3SCS) umgesetzt werden. Bei der 3SCS Architektur liegt zwischen dem Client und dem Daten-Server der sogenannte Middle-Tier-Server. Jede Schicht hat einen spezifischen Verantwortungsbereich. Tier 1 bildet der Client mit seiner grafischen Benutzeroberfläche, in die der Anwender seine Eingaben

macht. Tier 2 ist der Application Server, der für die Geschäftsprozesse und die Datenzugriffe verantwortlich ist, und Tier 3, der Daten-Server, der die Geschäftsdaten bereitstellt (ITWissen, 2013a).

3.2.2 3 Schichten Architektur

Das System ist in drei Schichten eingeteilt.

Es gibt zwei Arten von Präsentationen. Einmal die Präsentation für den Caretaker und einmal für den Senior. Beide greifen auf die darunter liegende Anwendungsschicht zu, welche sich auf dem Server befindet. Die Anwendungsschicht wiederum greift auf die darunter liegende Persistenzschicht zu, um Daten auszulesen bzw. abzuspeichern oder Benutzer zu autorisieren. Dabei stellt die untere Schicht der darüber liegenden Schicht eine Schnittstelle, womit die darüber liegende Schicht den Dienst der darunterliegenden Schicht in Anspruch nehmen kann. Jede Schicht kann nur den Dienst der direkt unter ihr liegenden Schicht in Anspruch nehmen (ITWissen, 2013b).

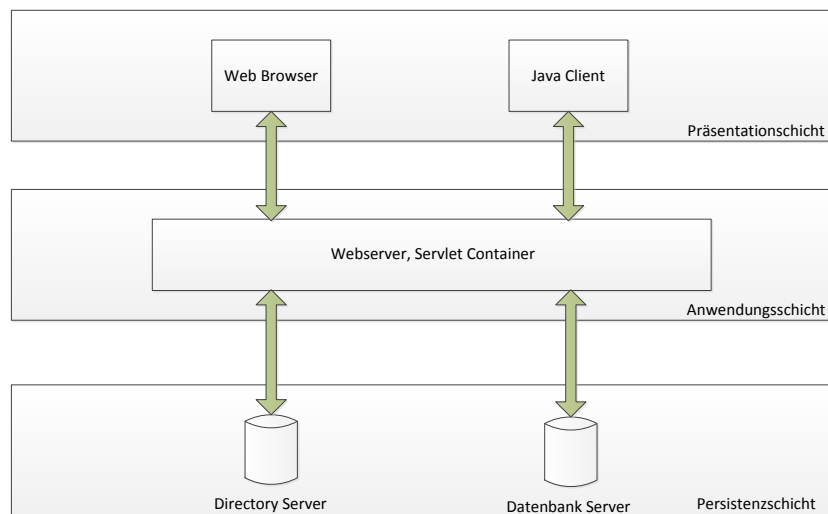


Abbildung 3.2: 3-Schichten Architektur

Client/Server-Funktionalität

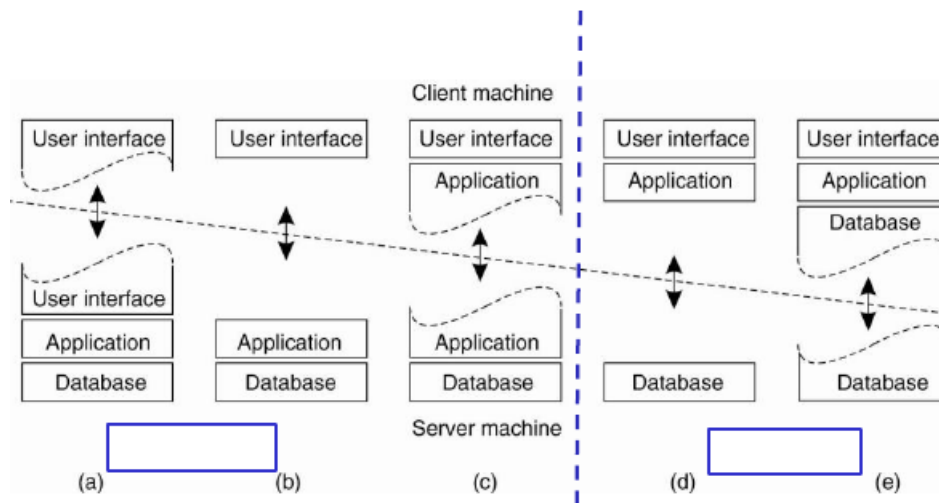


Abbildung 3.3: Client Server Aufteilung
(Andrew S. Tanenbaum, 2007)

Bei einer Client/Server-Architektur wie sie hier verwendet werden soll, muss entschieden werden wie viel Funktionalität der Server und wie viel der Client übernehmen soll. Nach Tanenbaum gibt es 5 verschiedene Möglichkeiten der Aufteilungen. Der Client kann nur für die Präsentation der Anwendung zuständig sein, bis hin zu das er auch Anwendungslogik sowie Persistenz enthält. Sollte der Client nur für die Präsentation zuständig sein, so spricht man von einem Thin-Client. Im Extremfall spricht man von einem Fat-Client (Andrew S. Tanenbaum, 2007).

Da die Aufgabe des Clients (Set-Top-Box) lediglich darin besteht Audiodateien aufzuzeichnen, das Menü darzustellen und mit dem Senior zu interagieren wäre eine Thin-Client Realisierung möglich. Im Allgemeinen bieten Browser keine Möglichkeit um Audiodateien aufzuzeichnen, da es den Browsern nicht gestattet ist auf die Peripherie Geräte des Computers zuzugreifen. Die Hauptgründe hierfür sind Sicherheitsbedenken. „Seit 2009 arbeitet das World Wide Web Consortium daran, Webbrowsern für Voice- beziehungsweise Videokonferenzen den Zugriff auf Mikrofon beziehungsweise Kamera kontrolliert einzuräumen.“ (iX Ausgabe 12). Man spricht hierbei von Web Real-Time Communications kurz WebRTC. Der Opera Browser sowie der Chrome Browser haben die Entwürfe in die aktuellen Desktop-Versionen ihrer Browser eingebunden. Mit diesen Browsern wäre also eine Thin-Client Realisierung möglich. Mehr kann hierzu in den W3C Drafts nachgelesen werden (Rogers, 2013). Sowie auf der offiziellen Seite zum WebRTC <http://www.webrtc.org/>. Da es sich hierbei um eine neue Technologie handelt (Stand 2012) und nur von wenigen Browsern unterstützt wird, wurde diese Möglichkeit der Realisierung verworfen.

Für die Realisierung des Clients wird eine Rich-Client Realisierung in Betracht gezogen, so wie sie in Abbildung 3.3 mit der Markierung c dargestellt ist. Ein Rich-Client besitzt neben der Präsentation auch Teile der Anwendungslogik.

3.3 System Entwurf

3.3.1 Komponentendiagramm

Beschreibung der Komponente

In diesem Abschnitt werden die Komponente des Komponentendiagrammes aus Abbildung 3.4 beschrieben.

FileUploadManager Die FileUploaderManager Komponente soll für das hochladen von Dateien zum Server verantwortlich sein. Hochgeladene Dateien sollen im Filesystem des Servers abgespeichert. Als Dateiname soll eine „Universally Unique Identifier“ kurz UUID verwendet werden. Hierdurch soll verhindert werden, dass eine bestehende Datei aufgrund einer Namenskollision überschrieben wird. Vor dem anlegen der Datei soll geprüft werden ob die UUID bereits vergeben ist. Sollte die UUID nicht vergeben sein, so soll die UUID in der Datenbank abgespeichert. Mit der UUID kann die Datei später wieder gefunden werden. Sollte die UUID bereits vergeben sein, so soll eine neue erzeugt und geprüft werden ob diese bereits vergeben ist. Dies erfolgt so lange, bis eine freie UUID gefunden wird oder ein Timeout erreicht ist. Bei einer UUID handelt es sich um eine 16-Byte-Zahl welche folgenden Aufbau hat: 550e8400-e29b-11d4-a716-446655440000. Bei einem derart großen Namensraum ist es relative Unwahrscheinlich, dass keine UUID gefunden wird.

Mail-User-Agent Der Mail-User-Agent soll für das abrufen und versenden von E-Mails verantwortlich sein. Hierfür stellt er eine Verbindung zu einem E-Mail Provider her.

Bestandteile eines Mail-User-Agent:

- Benutzeragenten(User Agents) mit denen die Nachrichten gelesen und gesendet werden.
- Nachrichtenübertragungsagenten (Message Transfer Agents) die Nachrichten von der Quelle zum Ziel befördern.

Caretaker Manager Mit dem Caretaker Manager sollen neue Caretaker angelegt, bearbeitet und gelöscht werden.

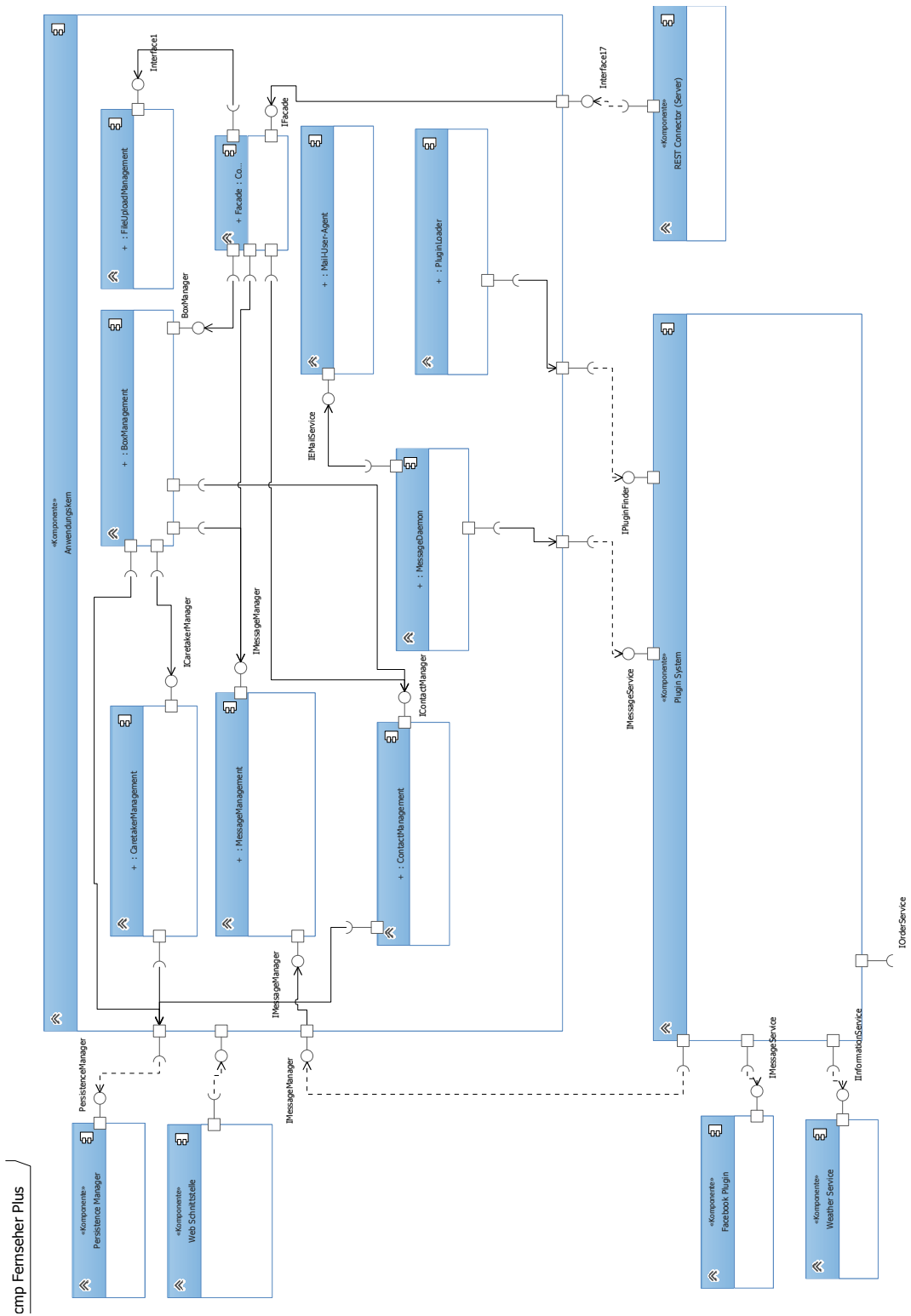


Abbildung 3.4: Komponentendiagramm

Message Manager Der Message Manager soll für die Verwaltung von Nachrichten verantwortlich sein. Nachrichten können von einem E-Mail Konto stammen oder von sozialen Netzwerken wie Facebook oder Twitter. Damit Nachrichten einheitlich behandelt werden, sollen die Nachrichtenköpfe in ein einheitliches Format konvertiert werden. Nach der Konvertierung werden diese in die Datenbank abgelegt und warten dort auf den Abruf. Die wichtigsten Informationen eines Nachrichtenkopfs sind der Absender, Provider(E-Mail,Facebook,Twitter) und der Nachrichteninhalte. Der Absender und der Provider sind wichtig, um bei einer späteren Beantwortung einer Nachricht, das richtige Plugin und den richtigen Adressaten auszuwählen.

Message Daemon Der Message Daemon soll dafür sorgen dass die verschiedenen Message Services in regelmäßigen Abständen Nachrichten abrufen.

Box Manager Der Box Manager soll für das anlegen, bearbeiten und löschen von Boxen zuständig sein. Erst nachdem eine Box angelegt ist, kann sich die Box mit dem Fernseher Plus Server verbinden und sich anmelden.

Plugin Loader Der Plugin Loader soll für das Laden der verschiedenen Plugins verantwortlich sein. Er soll dafür sorgen dass die kompilierten Dateien innerhalb des Plugins dem Classloader bekannt gemacht werden. Erst dadurch kann das System um weitere Funktionalitäten erweitert werden.

Plugin System - Das Plugin System ist für das laden der Bibliotheksdateien der Plugins verantwortlich. Diese Komponente arbeitet eng mit der Plugin Loader Komponente zusammen.

Facebook Plugin - Ein einfaches Plugin um auf Nachrichten von Facebook zuzugreifen.

Weather Plugin - Ein einfaches Plugin um auf den Wetterservice von Worldweatheronline zuzugreifen. Hiermit kann das aktuelle Wetter abgerufen werden.

3.3.2 Laufzeitdiagramm

Dieser Abschnitt beschreibt das Laufzeitdiagramm aus Abbildung 3.5.

In diesem Laufzeitdiagramm sehen wir 2 Akteure. Rechts im Diagramm sehen wir Oma Frieda und links ihren Enkel Tom. In diesem Diagramm werden die Anwendungsfälle „Nachrichten lesen“, „Nachrichten beantworten“ und „Antwort herunterladen“ beschrieben.

Enkel Tom schreibt eine Nachricht an seine Oma

1. Enkel Tom erstellt auf der Facebook Seite von Oma Frieda ein Pinnwand Eintrag. Oma Frieda muss nicht wissen, dass sie ein Facebook Account besitzt. Der Account wurde von einem Caretaker eingerichtet.

2. Der Fernseher Plus Server überprüft in regelmäßigen Abständen ob neue Nachrichten im Posteingang vorliegen. Hierzu werden alle Plugins welche das MessageService Interface implementieren angefragt.
3. Da es eine neue Nachricht von Tom gibt, ruft der Fernseher Plus Server diese ab.
4. Die abgerufene Nachricht wird in die Datenbank geschrieben. Zusätzlich wird die Nachricht als ungelesen markiert, damit sie später ganz oben in der Liste der Nachrichtenliste steht.

Visual Paradigm for UML Enterprise Edition (Hamburg University of Applied Sciences)

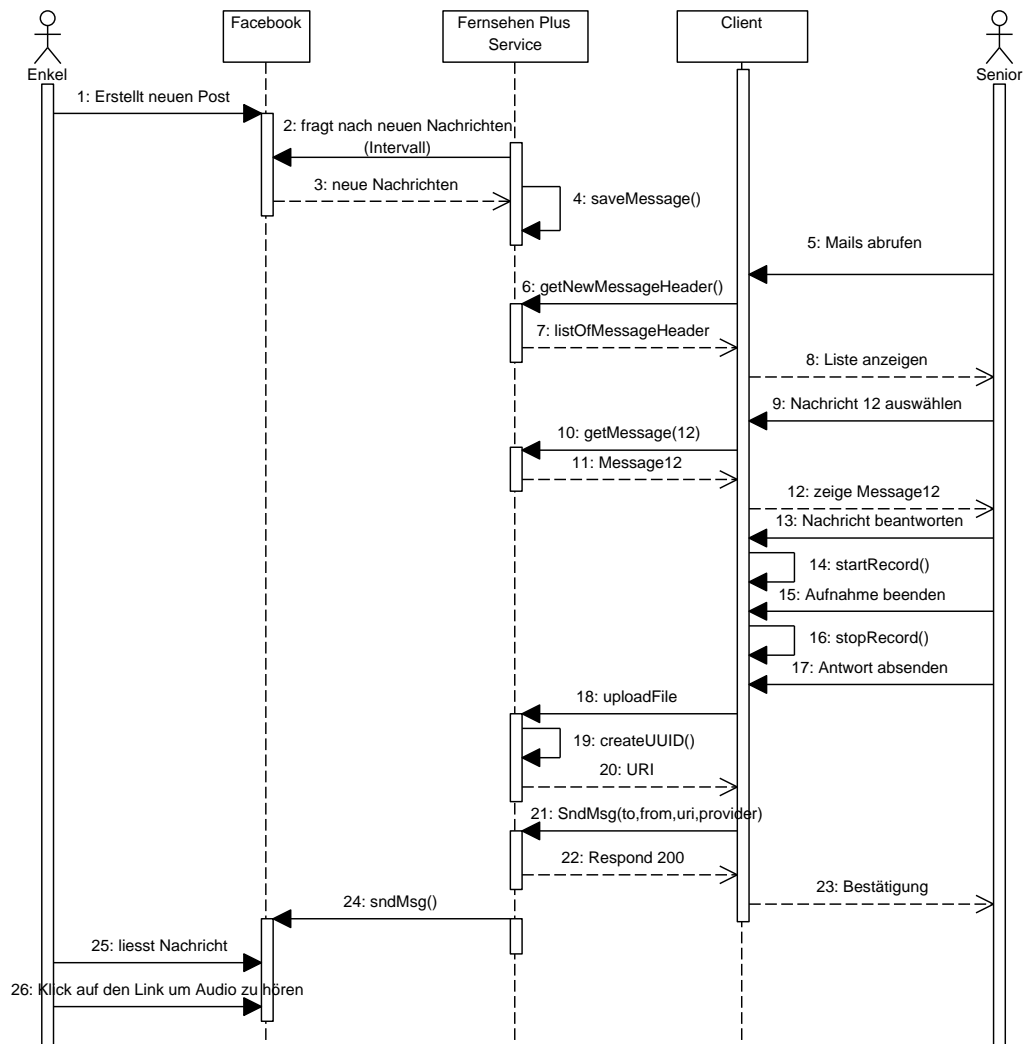


Abbildung 3.5: Nachrichten abrufen und versenden

Oma Frieda schaut ins Postfach

5. Der Senior möchte nachschauen ob es neue Nachrichten in seinem Posteingangsfach gibt. Hierzu wählt er den Menüpunkt „Nachrichten“ im Hauptmenü aus. Die Abfrage ob neue Nachrichten vorliegen, kann auch implizit erfolgen. Hierzu schaut die Set-Top-Box z.B. alle 10 Minuten selbstständig ins Posteingangsfach und teilt dies dem Senior durch ein Popup Menü mit.
6. Der Client wandelt die Aktion des Benutzers in einen REST Request um. Es wird eine Liste mit allen **Message Header** angefordert.
7. Der Server liefert dem Client eine Liste der **Message Header** zurück falls es welche gibt.
8. Der Client bereitet die Liste der **Message Header** für die Präsentation auf und stellt sie dar.
9. Oma Frieda wählt die Nachricht von Tom aus.
10. Der Client fragt den Nachrichten Inhalt der Nachricht mit der ID 12 beim Server an.
11. Der Server liefert dem Client die Nachricht mit der ID 12 zurück.
12. Der Client bereitet den Inhalt der **Message** für die Präsentation auf und stellt sie dar.

Oma Frieda beantwortet die Nachricht von Tom

13. Oma Frieda möchte die Nachricht von Tom beantworten und drückt hierfür auf die **Bestätigungstaste** der Fernbedienung.
14. Der Client beginnt mit der Aufnahme der Sprachaufzeichnung.
15. Oma Frieda möchte die Aufnahme stoppen und drückt erneut auf die Bestätigungstaste.
16. Die Aufnahme wird gestoppt, abgespeichert und noch einmal abgespielt.
17. Oma Frieda ist mit der Aufnahme zufrieden und drückt erneut die Bestätigungstaste. Dadurch wird der Upload der Aufnahme Datei initiiert.
18. Die Datei wird auf den Server hochgeladen.
19. Der Server generiert eine UUID und speichert die Datei mit dem Namen der UUID im Filesystem ab.
20. Der Server sendet dem Client eine Uniform Resource Identifiern kurz URI zu. Mit dessen Hilfe auf die Datei zugegriffen werden kann.
21. Der Client erstellt eine Nachricht mit der URL als Inhalt.

22. Sollte das versenden der Nachricht zum Fernseher Plus Server geklappt haben, so wird der Client darüber informiert.
23. Der Client zeigt Oma Frieda an, dass die Nachricht korrekt versendet wurde, auch wenn der Fernseher Plus Server die Nachricht noch nicht weiter senden konnte(z.B. Aufgrund einer Überlastung der Datenleitung).
24. Der Fernseher Plus Server wählt das passende Plugin zum versenden der Nachricht aus. Die Information hierzu findet der Server in dem **Message Header**. Hier befindet sich die Information, um welchen **Provider** es sich handelt.

Enkel Tom hört sich die Antwort von Oma Frieda an

25. Enkel Tom schaut zu einem späteren Zeitpunkt auf seine Facebook Pinnwand und entdeckt die Nachricht von Oma Frieda.
26. Tom klickt auf den Link und hört sich die Antwort auf seine Nachricht an.

3.3.3 Datenmodell

Die typische Nutzung des Systems wird in folgendem Szenario beschrieben um daraus ein ER-Datenmodell zu erstellen. Es wird die Notationsform nach James Martin verwendet. Das ER-Datenmodell wird später für die Erstellung der Entitäten benötigt.

1. Ein Familienmitglied oder eine Pflegeperson erstellt sich einen **Caretaker** Account auf dem Fernseher Plus Server. Hierzu muss ein eindeutiger Benutzername, Passwort sowie die Anschrift angegeben werden.
2. Der Caretaker fügt eine **Box** dem System hinzu. Dies erfolgt mithilfe der Box UID und dem dazugehörigen Passwort.
 - Wenn die Box noch nicht im System vorhanden ist, so wird eine neue Box angelegt und dem Caretaker-Account hinzugefügt.
 - sonst wird die bestehende Box dem Caretaker-Account hinzugefügt.

Jeder Caretaker kann sich um eine beliebige Anzahl von Boxen kümmern. Boxen können von mehreren Caretaker verwaltet werden. Eine Box kann nicht ohne einen Caretaker existieren. Der Caretaker registriert die Box.

3. Der Caretaker schaltet gewünschte **Services** für den Senior frei. Eine Box kann beliebig viele Services abonnieren. Ein Service kann von mehreren verschiedenen Boxen abonniert werden.

4. Wenn ein oder mehrere Message Service eingerichtet sind, so ruft der Fernseher Plus Server in bestimmten Intervallen die Nachrichten aller eingerichteten Message Service ab und speichert sie in die Datenbank. Der eigentliche Nachrichteninhalte (**Message**) wird separat von den Meta-Informationen **Message Header** abgespeichert, damit Datenbankabfragen schneller verarbeitet werden und kleinere Resultsets zurückgesendet werden. Zu jedem Message Header gehört genau eine Message.
5. Der Benutzer einer Box möchte eine Antwort an den Absender einer empfangenen Nachricht schicken. Hierzu wird eine Audiodatei aufgezeichnet und abgespeichert (**UploadedFile**). Jede Box kann eine Vielzahl von Audiodateien (UploadedFile) besitzen.
6. Damit der Senior mit seinen Freunden und Verwandten in Kontakt bleiben kann, besitzt die Set-Top-Box ein virtuelles Telefonbuch, in dem die Kontaktdaten gespeichert werden. Jede Box kann eine Vielzahl von Kontakten besitzen.

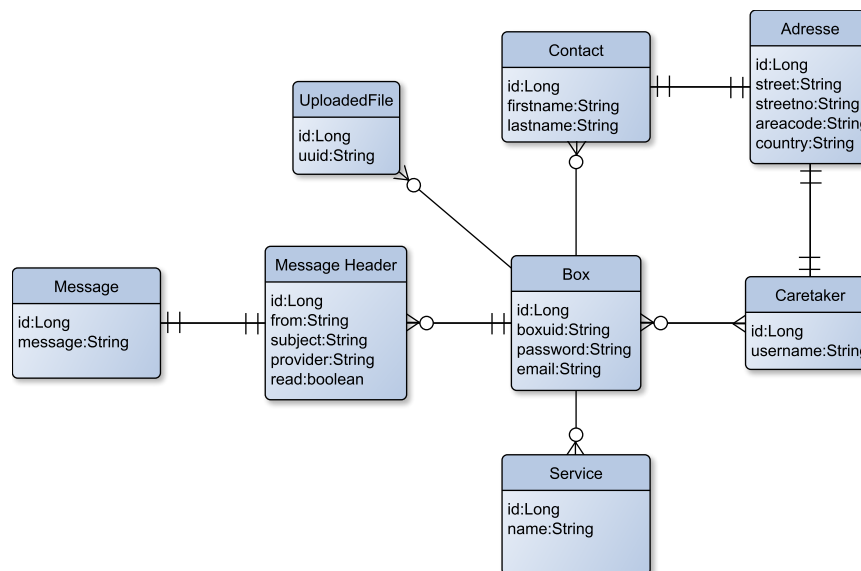


Abbildung 3.6: Fachliches Datenmodell

3.3.4 Plugin System

Das Backend System soll über ein Plugin System verfügen. Dadurch können Programmierer neue Funktionen in Form von Plugins entwickeln, welchen den Server um diese Funktionalitäten erweitern. Das besondere hierbei ist, dass die Programmierer nicht wissen müssen wie der Kern der Anwendung funktioniert. Sie müssen lediglich gegen die Schnittstellen programmieren, welche vom Server bereitgestellt werden.

Für diese Arbeit werden zwei Plugin Schnittstellen benötigt, welche in Listing 3.1 und 3.2 beschrieben sind. Während die Schnittstelle für das Wetter ziemlich speziell ist, ist die Schnittstelle für die MessageService eher generisch.

```
public interface WeatherService {  
    // Zum abrufen des aktuellen Wetters  
    public Weather requestWeather(String city, String country);  
}
```

Listing 3.1: weatherapi

```
public interface MessageService extends Serializable {  
    // Dient der Initialisierung des Plugins. (z.B. Username, Password, Accesstoken etc.)  
    public void init(Properties props);  
    // Eindeutiger Name des Plugins  
    String getName();  
    // Zum erstellen einer Nachricht  
    void sendMessage(Message msg);  
    // Zum abrufen von Nachrichten  
    List<Message> receiveMessage();  
}  
  
public class FacebookService implements MessageService{  
  
}
```

Listing 3.2: Message Service Api

Die Idee den Applikationsserver um ein Plugin System zu erweitern, kam aufgrund der langen Deployment Zeit einer Java EE Anwendung. Diese kann je nach Rechenleistung und neu zu kompilierenden Dateien bis zu 30 Minuten betragen. Da das Backend System oft um neue Funktionalitäten erweitert wird (z.B. ein Service für das abrufen von Twitter Nachrichten), müssen oft viele Dateien neu kompiliert werden und erneut hochgeladen werden. Das Plugin System selbst wird in dieser Arbeit nicht selbst entwickelt. Stattdessen wird die Realisierung aus dem Artikel „Writing Pluggable Java EE Applications“ von Jason Lee verwendet. Hierzu werden lediglich die Schnittstellen der Plugins angepasst.

3.4 Client Entwurf

Die Gestaltung und Bedienung des Clients bildet den Schwerpunkt dieser Arbeit. Nur ein System welches leicht bedienbar ist und einen Mehrwert für den Senior darstellt wird von diesem akzeptiert. Ein System hat nur dann eine Zukunft, wenn es von seinen Endbenutzern als leicht und verständlich angesehen wird (Phang u. a., 2006).

3.4.1 Model-View-Controller Entwurfsmuster

Dieses Entwurfsmuster entstand während der Verbreitung ereignisgesteuerter grafischer Benutzeroberflächen. 1979 formulierte der norwegische Computerwissenschaftler Trygve Reenskaug das Model-View-Controller Architekturmuster kurz MVC zur Strukturierung von Anwendungen mit grafischen Benutzeroberflächen. Ziel des MVC ist die Trennung der Darstellung von der Ablaufsteuerung und der Datenhaltung. Dadurch sind die Komponente lose gekoppelt und wieder verwendbar. Die Anwendung dieses Musters führt durch die Trennung der Verantwortlichkeiten zu einer größeren Flexibilität und besseren Erweiterbarkeit des Systems (Starke, 2011).

Das MVC Entwurfsmuster soll für die spätere Realisierung des Clients(Set-Top-Box) verwendet werden.

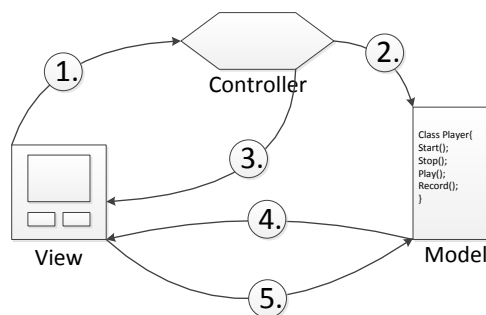


Abbildung 3.7: Model View Controller Architektur

View Views entsprechen den Fachmasken einer Anwendung und repräsentieren einen Zustand. Es nimmt Anwender-Aktionen an und leitet diese als fachliches Ereignis an den Controller weiter. Jeder View kann zu jeder Zeit sein zu gehöriges Model nach seinem Zustand fragen und sich dementsprechend anpassen.

Controller Der Controller sitzt zwischen View und Model und vermittelt zwischen diesen. Er ist für die Ablaufsteuerung der Anwendung zuständig. Zu jedem Zeitpunkt besitzt er einen Zustand und verhält sich wie ein Zustandsautomat. Controller setzen Ereignisse in Domänen-Operationen um und legen den Folge-Zustand des Systems fest. Bei einer Zustandsänderung benachrichtigt er das Model.

Model Das Model beinhaltet die fachlichen Information. Es bereitet diese Information für die View passend zu.

Ergänzend sollte dabei erwähnt werden, dass es zu jedem Model mehrere Controller mit View geben kann. Diese teilen sich dann den Zustand des Models.

Abbildung 3.7 beschreibt beispielhaft wie die MVC Architektur funktioniert.

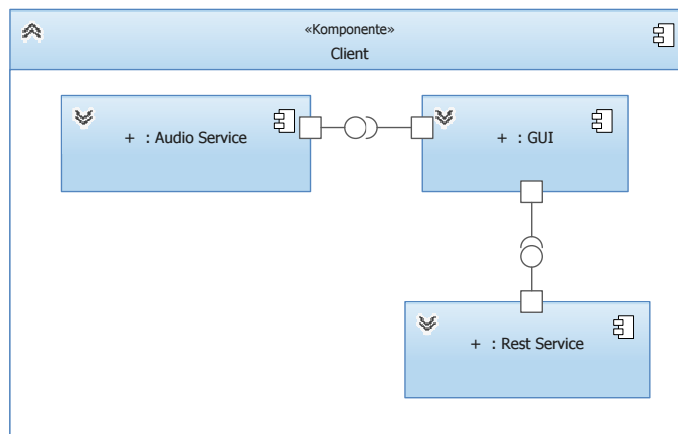


Abbildung 3.8: Client Komponente

1. Der Benutzer drückt einen Button
2. Der Controller interpretiert die Eingabe des Benutzers und stellt fest was es für das Model bedeutet
3. Der Controller kann dem View dazu auffordern die Anzeige zu aktualisieren. Zum Beispiel das der Button gesenkt bleiben soll, nachdem er betätigt wurde.
4. Sollte sich der Zustand des Models ändern, so informiert er alle Observer darüber. Standardmäßig ist dies die View, aber auch der Controller kann sich beim Model registrieren und wird dann über Zustandsänderung informiert.
5. Die View holt sich Information über den aktuellen Zustand des Models.

3.4.2 Client Komponente

Der Client besteht aus den in Abbildung 3.8 genannten Komponenten:

Audio Service Die Audio Service Komponente ist für die Sprachaufzeichnung und das Abspielen einer Sprachaufzeichnung verantwortlich. Nach erstellen einer Sprachaufzeichnung wird diese temporär zwischengespeichert, damit sie vom Rest Service weiter verarbeitet werden kann.

Rest Service Der Rest Service ist für die Kommunikation mit dem Fernseher Plus Server verantwortlich und hat folgende Aufgaben:

- Client Anfragen zum Server senden
- Antworten vom Server entgegen nehmen
- Client Anfragen in das JSON Format umwandeln (Marshalling)
- JSON Nachricht in Java Objekte umwandeln für die weitere Bearbeitung (Unmarshalling)
- Dateien zum Server hochladen

Graphical User Interface (GUI) Die grafische Benutzeroberfläche erlaubt die Interaktion mit dem Computer über grafische Symbole. Die Auswahl der grafischen Symbole erfolgt mittels einer Fernbedienung. Für die Interaktion mit der Set-Top-Box stellt die GUI verschiedene Observer bereit, welche auf die von der Fernbedienung ausgelösten Events reagieren und diese in Aktionen umwandelt. Eine Aktion kann etwa das hoch oder runter scrollen einer Liste von Mails oder Kontakten sein. Die GUI greift auf die Dienste des Rest Service und des Audio Service zu. Der genaue Aufbau der GUI wird weiter unten beschrieben.

3.4.3 Aufbau der grafischen Benutzeroberfläche

Für die grafische Benutzeroberfläche soll ein Wizard Dialog verwendet werden. Ein Wizard Dialog ist eine spezielle Form der Benutzeroberflächengestaltung, bei der ein Benutzer durch eine Sequenz von Dialogen geführt wird. Jeder Dialog stellt dabei einen wohldefinierten Schritt in dem Assistenten dar. Diese Dialogform wird häufig bei der Installation von Anwendungen verwendet, bei dem der Benutzer eine Reihe von Frage beantworten muss.

In Abbildung 3.9 befinden sich 6 Dialoge. Beginnend beim **Main Menu View**(1) Dialog, gelangt der Benutzer durch Auswahl des Menüpunktes „Nachrichten“ zum **Message Overview** Dialog(2) welche dem Benutzer die Liste aller Nachrichten anzeigt. Durch Bestätigung der Auswahl einer Nachricht durch die Taste „OK“ gelangt der Benutzer zum **Message Read View**(3) Dialog in welcher dem Benutzer der Inhalt der ausgewählten Nachricht angezeigt wird. Der Benutzer kann in diesem Dialog durch Betätigung der „OK“ Taste eine Antwort für den Absender der Nachricht erstellen. Dadurch gelangt der Benutzer zum **Record Stop View**(4) Dialog, in welcher der Benutzer gefragt wird ob er eine Antwort aufzeichnen möchte. Durch Betätigung der „OK“ Taste wird die Aufnahme gestartet. Hierzu muss ein neuer Thread gestartet werden, damit der Benutzer weiterhin mit der GUI interagieren kann. Dem Benutzer wird der **Record Start View** Dialog angezeigt. In diesem Dialog kann der Benutzer die Aufnahme durch erneutes drücken der „OK“ Taste beenden. Dadurch gelangt er zum **Record Replay View** Dialog in welchem dem Benutzer die Aufnahme noch einmal abgespielt wird. Sollte dem Benutzer die Aufnahme gefallen, so kann er durch ein letztes mal drücken

der „OK“ Taste die Aufnahme zum Absender senden. Jetzt ist es Aufgabe des Rest Service die Datei hochzuladen. Hierzu muss ein weiterer Thread erzeugt werden.

Die Abbildung 3.9 zeigt wie der Ablauf eines Wizard Dialog aussehen könnte. Die Dialogboxen werden im folgenden beschrieben:

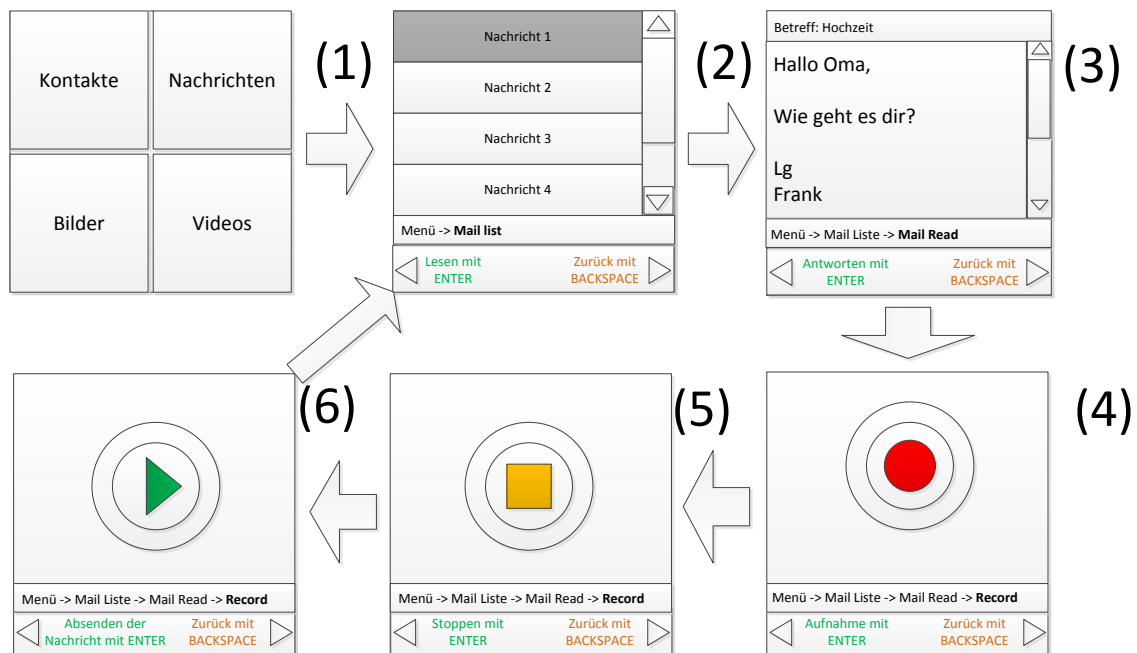


Abbildung 3.9: Pageflow Diagram Nachrichten versandt

Die Abbildung 3.10 zeigt einen Zustandsautomaten, welcher den Ablauf zur Erstellung einer Nachrichtenantwort verdeutlicht.

Die Abbildung 3.11 zeigt den typischen Aufbau eines Untermenüpunktes. In der Navigationsleiste kann der Benutzer die Menütiefe sehen. Zudem sieht er wohin er kommt, wenn er die **Zurücktaste** drückt.

3.4.4 Interaktion mit der grafischen Benutzeroberfläche

Für die Bedienung eines Computers wird in der Regel eine Tastatur und eine Maus verwendet. Neuerdings finden auch Touchscreens eine zunehmende Beliebtheit. Es ist eher selten der Fall, dass ein Computer mit einer Fernseher Fernbedienung bedient wird. Hierbei gilt es zu beachten, dass eine Fernbedienung im Vergleich zu einer Tastatur nur sehr wenig Tasten besitzt. Um dieses Defizit auszugleichen müssen die Tasten der Fernbedienung mehrfach

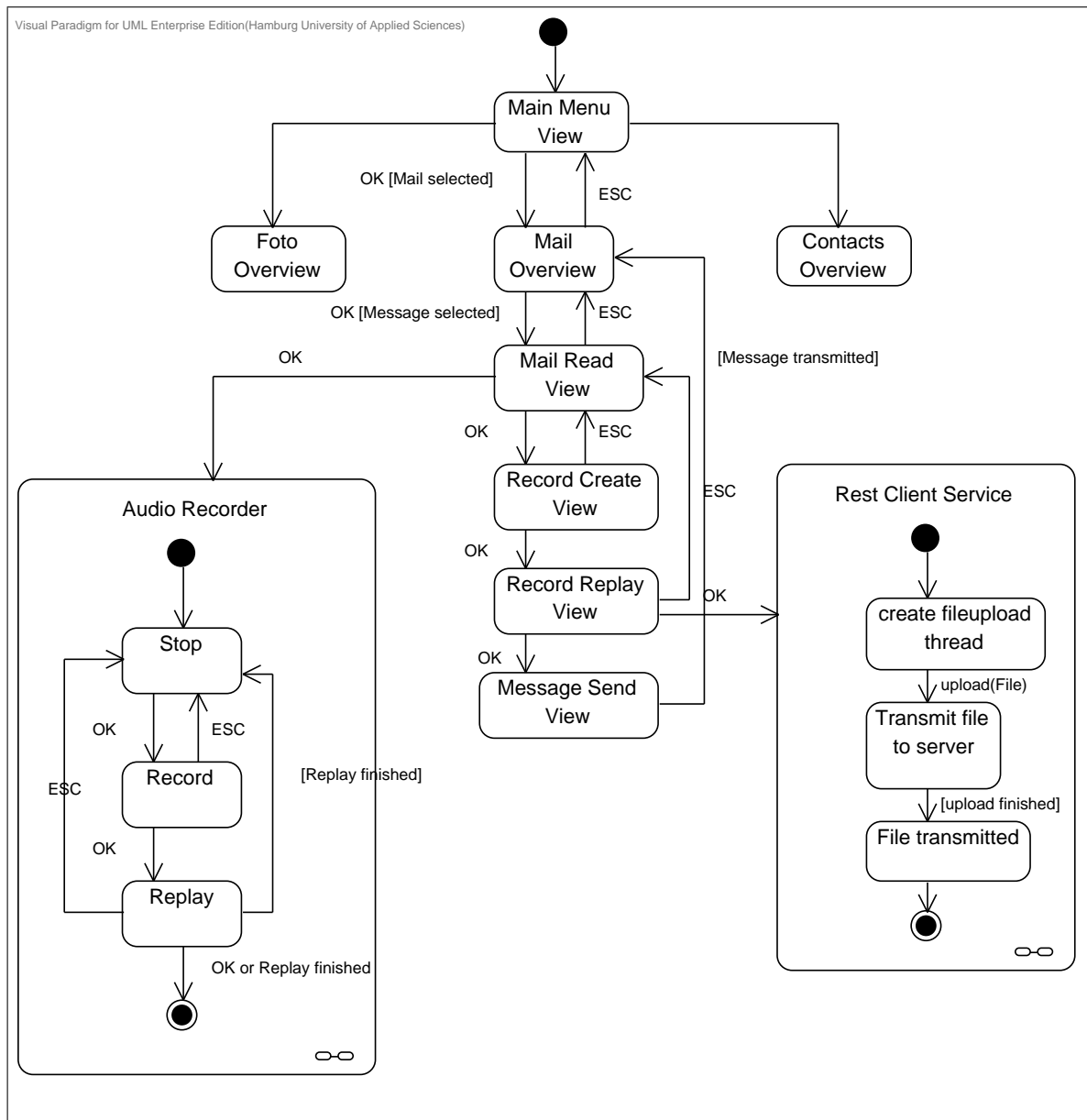


Abbildung 3.10: Statemaschine Nachrichten versandt

belegt werden. Dies jedoch erhöht die Komplexität der Bedienung. Eine weit aus bessere Lösung ist es, dem Benutzer eine Auswahl der möglichen Aktionen auf dem Bildschirm zu präsentieren, aus welchem dieser eine Aktion auswählen kann. Dadurch werden weit aus weniger Tasten benötigt. In dieser Arbeit werden hierzu nur 7 Tasten benötigt. 4 Tasten für das Steuerkreuz und drei weitere Funktionstasten.

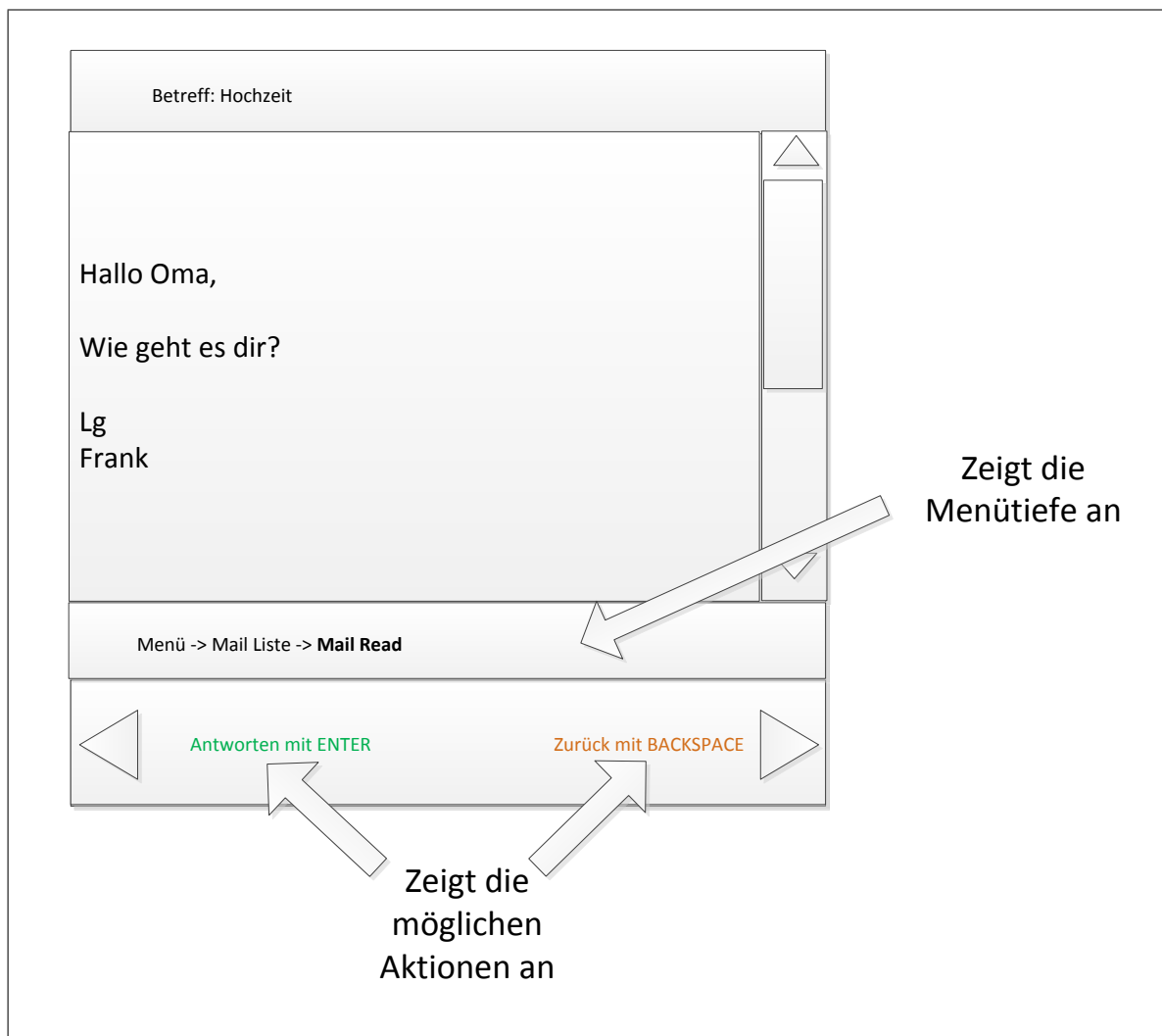


Abbildung 3.11: Aufbau eines Untermenüpunktes

3.5 Weitere Eigenschaften des Systems

3.5.1 Sicherheit

Authentifizierung

Daten die zum Fernseher Plus Server gesendet werden, gehen über das Internet. Das Problem hierbei ist, dass diese Daten über verschiedene Server gehen. Es muss davon ausgegangen werden, dass einer dieser Server die gesendeten Daten abhört. Eine Möglichkeit dies zu erschweren ist es die Daten zu verschlüsseln. Standardmäßig wird hierfür eine SSL Verbindung zwischen Client und Server eingerichtet.

Authentifizierung ist der Prozess bei der, die Identität eines Remote-Users bestätigt wird. In dieser Arbeit soll hierfür die „Client Certificate Authentication“ Methode verwenden, bei der sowohl der Server als auch der Client über ein eigenes Zertifikat verfügen, mit dessen Hilfe sie sich gegenüber authentifizieren können. Das besondere hierbei ist, dass der Senior später weder Username noch Password eingeben muss.

Autorisierung

Mit der Autorisierung soll festgestellt werden ob ein Remote-User berechtigt ist bestimmte Aktionen durchzuführen. Hierfür werden Rollen und Gruppen eingerichtet, in welcher sich der Benutzer befinden kann.

3.5.2 Kommunikation

Protokoll

Damit der Client und der Server sich verständigen können, muss ein Protokoll vereinbart werden. Ein Protokoll legt das Format, den Inhalt, die Bedeutung und die Reihenfolge einer gesendeten Nachricht fest. Die folgende Tabelle beschreibt das Protokoll für die Kommunikation zwischen Set-Top-Box(Client) und dem Fernseher Plus Backend System(Server). Das Protokoll genügt für den Prototypen, jedoch ist es für das endgültige System viel zu kurz gehalten.

Anfrage	Antwort	Bedeutung
getAllMessageHeader	messageHeaderList	Abrufen aller Nachrichtenköpfe
getMessageWithId	message	Abrufen einer bestimmten Nachricht
getAllContacts	contactList	Abrufen aller Kontakte
getContactWithId	contact	Abruf eines bestimmten Kontaktes
getWeatherFromContact	currentLocationWeather	Abruf des Wetters eines Kontaktes

Die gesendeten und empfangenen Nachrichten sollen im JSON Format vorliegen.

3.5.3 Darstellung

Das Hauptmenü soll ein Kachel-Design haben ähnlich dem neuem Windows 8 Betriebssystem (Stand 2012). Jede Kachel führt zu einem Menüpunkt. Ein Menüpunkt kann etwa die Übersicht aller eingegangenen Nachrichten sein oder die Liste aller Kontakte darstellen. Jede Kachel soll eine Beschriftung sowie ein Symbol erhalten, damit es schneller wieder erkannt werden kann. Zunächst sollen nur 4 Kacheln angezeigt werden, damit die Beschriftungen sowie die Symbole ausreichend groß darstellbar sind. Die Bedienlogik folgt einheitlichen seniorengerechten Regeln, indem gewohnte Interaktionsschemata beibehalten werden. Eine geringe Komplexität soll durch die Limitierung der Tiefe des Menubaumes erreicht werden. Dadurch soll erreicht werden, dass Senioren sich nicht in den Menübaum verlaufen.

3.6 Fazit

In diesem Kapitel wurde der Entwurf des Fernseher Plus Servers sowie des Clients dargestellt, um die aus der Analyse gestellten Anforderungen zu erfüllen. Das Gesamtsystem wurde zuerst in Schichten aufgeteilt und dann in Komponenten zerlegt. Dies geschah nach Paradigmen des objektorientierten Entwurfs. Exemplarisch wurde an ein paar Beispielen skizziert, wie das Verhalten des Systems aussehen sollte.

Das folgende Kapitel widmet sich dieser Realisierung.

4 Realisierung

4.1 Verwendete Technologien

4.1.1 Verwendete Hardware

Für einen Prototypen wird ein herkömmlicher Laptop verwendet, welcher über ein eingebautes Mikrofon verfügt. Für die Verbindung der Fernbedienung mit dem Laptop wird ein USB Infrarot Empfänger verwendet. Der Laptop dient in diesem Versuchsaufbau sowohl als Client als auch Server.

Auflistung der verwendeten Hardware Technologien:

1. Laptop Samsung R530 mit eingebauten Mikrofon
2. USB IR-Empfänger
3. Fernbedienung mit Steuerkreuz

Fernbedienung

Für die Bedienung der Set-Top-Box wurde eine Fernbedienung verwendet, welche der Fernbedienung eines Prototypen von Silpion ähnelt. Die Fernbedienung verfügt über ein Steuerkreuz sowie mindestens 3 weitere Knöpfe. Das Besondere an dieser Fernbedienung ist, dass der Treiber das Mapping von Fernbedienungstasten auf PC Keyboard Tasten übernimmt. So führt das Drücken der Taste „BACKSPACE“ dazu, dass in einem Text Editor ein Rückschritt vollzogen wird. Diese Eigenschaft können wir bei der Realisierung der Set-Top-Box nutzen, um die grafische Benutzeroberfläche zu bedienen.



Abbildung 4.1: Infrarot Fernbedienung

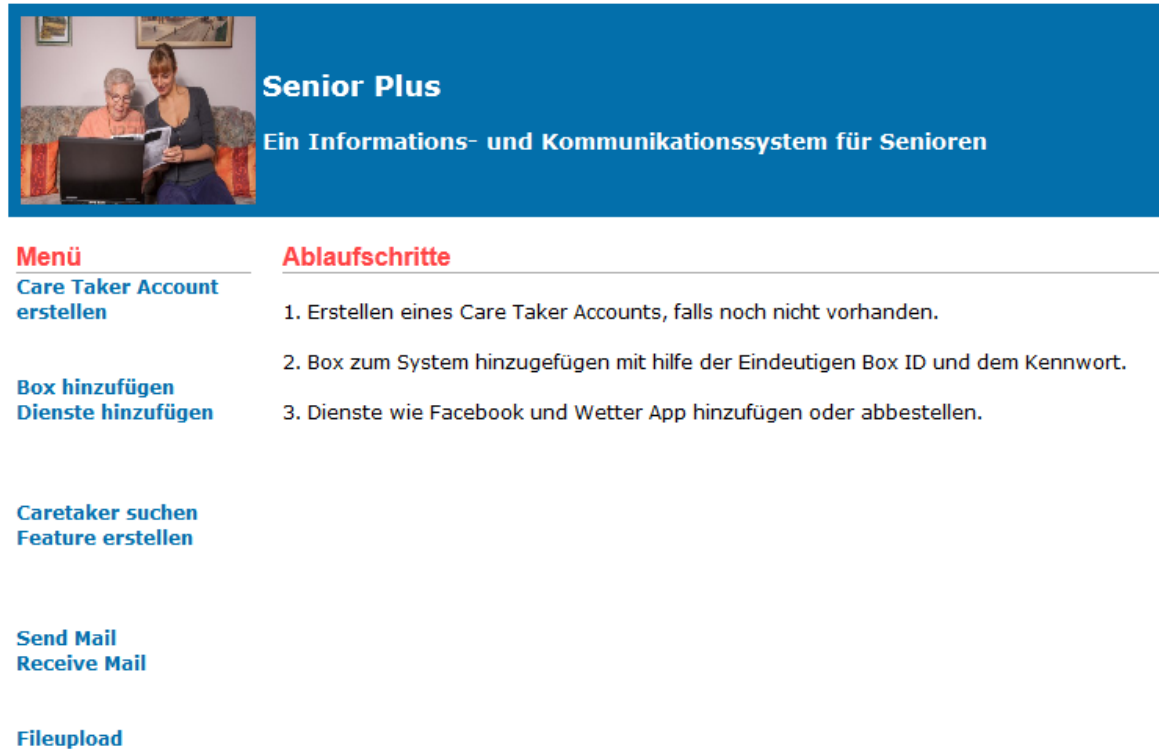
4.1.2 Verwendete Software

Für den Webserver/Application Server wurde Glassfish verwendet mit einer Derby Datenbankbindung. Für das Backend wurde Java als Programmiersprache verwendet, da es sich einerseits um eine Objektorientierte Sprache handelt und andererseits hiermit schon gute Erfahrungen gemacht wurden.

Auflistung der verwendeten Software Technologien:

1. Java 1.6 als Programmiersprache für Client Frontend sowie Server Backend
2. Jersey REST für die REST Anbindung
3. Glassfish für den Webserver/Applikation Server
4. Jackson JSON-processor für das parsen von JSON
5. JavaDB als Datenbank
6. JSF 2.0 für das Server Frontend
7. ApacheDS als Directory Server (LDAP)
8. Swing für das Frontend des Senior Clients
9. RestFB für die Anbindung an die Facebook API
10. EclipseLink für das mappen von Objekten auf Relationalen Tabellen (OR-Mapper)

4.2 Server Realisierung



Senior Plus
Ein Informations- und Kommunikationssystem für Senioren

Menü

- Care Taker Account erstellen
- Box hinzufügen
- Dienste hinzufügen
- Caretaker suchen
- Feature erstellen
- Send Mail
- Receive Mail
- Fileupload

Ablaufschritte

1. Erstellen eines Care Taker Accounts, falls noch nicht vorhanden.
2. Box zum System hinzufügen mit Hilfe der Eindeutigen Box ID und dem Kennwort.
3. Dienste wie Facebook und Wetter App hinzufügen oder abbestellen.

Abbildung 4.2: Portalseite des Fernseher Plus Servers

4.2.1 Java EE 6

Für die Realisierung des Backend Servers wurde ein Java EE¹ Server verwendet werden. Java-EE-Anwendungen lassen sich grob aufteilen in eine Clientschicht, die serverseitige Implementierung und eine Schicht, in der Daten abgelegt oder von anderen Systemen weiterverarbeitet werden.

Als Clients von Java-EE-Anwendungen werden häufig Browser eingesetzt. Die Präsentationslogik wird in diesem Fall serverseitig bereitgestellt, vielfach durch JavaServer Faces in einem Web Container.

¹Java EE ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von in Java programmierten Anwendungen und insbesondere Web-Anwendungen

Desktopclients implementieren diese Präsentationslogik i.d.R. selbst. Sie greifen entweder direkt auf die Geschäftslogik zu oder bedienen sich bspw. Web Services wie REST.

Die Geschäftslogik kann mittels CDI und/oder EJB programmiert werden und diverse weitere serverseitige Techniken wie Java Persistence, Connector, Messaging etc. verwenden.

Die verarbeiteten Daten werden häufig in Datenbanken abgelegt, können aber auch in anderen Backend-Systemen liegen, die auf unterschiedliche Weise angebunden werden.

Quelle: (Weil, 2012)

Context and Dependency Injection

Die Aufgabe eines jeden Softwareentwicklers ist es Komponenten zu erstellen und diese zu verknüpfen. Ziel hierbei ist eine Lose Kopplung zwischen den Komponenten herzustellen. CDI unterstützt den Entwickler hierbei in dem der Entwickler nicht selbst eine Instanz einer Komponente instanziiert, sondern das dies durch einen Applikationsserver (Container) bei Bedarf geschieht. Die Rede ist hierbei von „Inversion of control“, die den aktiven Teil der Komponentenverknüpfung aus der Komponente herauslöst und in die Laufzeitumgebung - den Container - verlagert: Nicht die Komponente besorgt sich die von ihr benötigten Serviceobjekte, sondern der Container liefert sie an. Bei der Dependency Injection obliegt es dem Container, wann die benötigten Objekte erzeugt und zerstört werden, er kann also die Komponenten von der kompletten Lifecycle-Steuerung entlasten (Weil, 2012). CDI wurde bei der Realisierung des Backend Servers verwendet, im speziellen bei der Realisierung der 3 Schichten Architektur.

4.2.2 3 Schichten Architektur

Die Anwendung ist als Webanwendung mit JSF und Facelets für die Präsentation aufgebaut. Die Logik der Anwendung befindet sich in CDI Beans, wobei eine saubere Schichtentrennung eingehalten wurde.

- Die Präsentationslogik befindet sich in Model-Objekten, d.h. CDI Beans mit dem Stereotype `@Model`. Hier ist nur die Aufbereitung der Daten für die Präsentation und die Statusverwaltung der Benutzeroberfläche enthalten.
- Für die Geschäftslogik wurden CDI Beans mit dem Stereotype `@DomainService` verwendet. Hier sind über den reinen Datenzugriff hinausgehende Geschäftsprozesssteile enthalten, z.B. umfangreiche Datenzusammenstellungen oder auch Zugriffe auf technische Dienste der Serverinfrastruktur.

- Der Zugriff auf die Datenbank geschieht mit Repository-Objekten. Dahinter verbergen sich CDI Beans mit dem Stereotype `@DataRepository`. Sie implementieren die CRUD-Funktionalitäten sowie die benötigten Find-Operationen auf Basis von Java Persistence.

Zwischen diesen Anwendungsschichten erfolgen die Zugriffe nur in Richtung Model.

```
@DataRepository
public class CaretakerServiceImpl implements CaretakerService {

    @PersistenceContext
    private EntityManager em;

    @Override
    public void create(Caretaker caretaker){
        em.persist(caretaker);
    }

    @Override
    public void update(Caretaker caretaker){
        em.merge(caretaker);
    }

    @Override
    public void remove(Caretaker caretaker){
        em.remove(em.merge(caretaker));
    }

    @Override
    public Caretaker find(Object id){
        return em.find(Caretaker.class, (Long)id);
    }

    ....
}
```

Listing 4.1: Beispielcode eines Repository-Objektes

```
@Model
public class CaretakerPM implements Serializable {

    private String id;
    private String username;
    private String password;

    // Injektionsstelle
    @Inject
    private CaretakerService ctservice;
```

```

    private Caretaker selectedCaretaker;

    @PostConstruct
    private void init() {
    }

    ...
}

```

Listing 4.2: Beispielcode der Präsentationslogik

```

<h:form id="caretakerForm">
  <h:message for="userName" styleClass="error"/>
  Username:
  <h:inputText value="#{caretakerPM.username}" id="userName" required="true"
               requiredMessage="Bitte_einen_Username_eingeben."/>

  <h:message for="userPassword" styleClass="error"/>
  Password:
  <h:inputText value="#{caretakerPM.password}" id="userPassword"
               required="true"
               requiredMessage="Bitte_ein_Password_eingeben."/>

  <h:commandButton value="Caretaker_erstellen"
                   action="#{caretakerPM.createCaretaker()}" />
  <hr/>
</h:form>

```

Listing 4.3: caretaker.xhtml

4.2.3 Umsetzung der REST Schnittstellen

Jeder Java Applikation Server welcher sich an die Java EE Spezifikation hält, bietet eine REST Schnittstelle an. Für diese Schnittstelle kann der Entwickler eine Implementierung frei wählen. In dieser Arbeit wird Jersey REST verwendet. Im Deployment Descriptor(web.xml) des Applikationsservers muss lediglich angegeben werden, für welche Implementierung man sich entschieden hat. Im Listing 4.4 wird ein Auszug aus der web.xml Datei gezeigt, welcher dem Applikationsserver die REST Implementierung bekannt macht.

```

<servlet>
  <servlet-name>JerseyServlet</servlet-name>
  <servlet-class>
    com.sun.jersey.spi.container.servlet.ServletContainer
  </servlet-class>

```



```

    ....
</servlet>

```

Listing 4.4: Aktivierung von Jersey REST

Rest Services sind ganz normale Java Klassen, welche annotiert sind. Listing 4.5 zeigt einen einfachen Rest Service. Der Aufruf der URL `http://localhost:8080/FernseherPlus/BOX_UID12/message` würde in diesem Beispiel ein JSON Objekt zum Client zurücksenden, welche alle Nachrichten Header enthält.

```

@Path("/{boxuid}/message")
@RequestScoped // Ist notwendig, um CDI zu verwenden
public class MailResource {
    @Inject
    MessageHeaderService messageHeaderService;

    @GET
    @Produces({MediaType.APPLICATION_JSON})
    public List<MessageHeader>getAllMessageHeader(){
        List<MessageHeader>messageHeaderList = messageHeaderService.findAll();
        return messageHeaderList;
    }
}

```

Listing 4.5: Einfacher Rest Service

4.2.4 Umsetzung der 3 Schichten Architektur

4.2.5 Plugin System

Für das Plugin System wird eine Entwicklung von Jason Lee verwendet (Lee, 2013).

Um das Plugin System nutzen zu können muss eine CDI Extension geschrieben werden. Listing 4.6 zeigt einen Auszug aus dem Code.

```

public class PluginLoader implements Extension {
    protected static final String SERVICES_NAME =
        "com.steeplesoft.plummer.finders";
    private static List<PluginFinder> pluginFinders;
    List<Class<? extends PluginFinder>> pluginFinderClasses;

    public void beforeBeanDiscovery(@Observes BeforeBeanDiscovery bbd,
        BeanManager beanManager) {
        for (PluginFinder pluginFinder : getPluginFinders()) {

```

```

    for (Class<?> clazz : pluginFinder.getClasses()) {
        final AnnotatedType<?> annotatedType =
            beanManager.createAnnotatedType(clazz);
        bbd.addAnnotatedType(annotatedType);
    }
}
}
}

```

Listing 4.6: Einbinden des Plugin Loaders

4.2.6 Wetter Plugin

Für das abrufen der Wetterdaten wurde eine öffentliche REST Schnittstelle von worldweatheronline.com verwendet. Damit die API genutzt werden kann, muss zuvor ein Benutzerkonto eingerichtet werden.

Um das aktuelle Wetter der Stadt Hamburg abzurufen, wird folgende URL aufgerufen.

```
http://free.worldweatheronline.com/feed/weather.ashx?q=Hamburg,
Germany&format=json&key=XXXXXXXXXXXXXXXXXXXX
```

Die Query Parameter haben folgende Bedeutung:

q Legt fest für welchen Ort das Wetter abgerufen werden soll

format Legt das Rückgabe Format fest

key Hier wird der Accesstoken angegeben, welchen man durch die Anmeldung auf der Seite erhält

Als Rückgabe Format wurde JSON festgelegt, da hierfür zahlreiche Open Source Parser existieren.

Damit der Server die Daten weiterverarbeiten kann, werden diese in Java Objekte umgewandelt. Hierfür wird der Jackson Java JSON-processor verwendet.

Der genau Ablauf wird in Abbildung 4.3 dargestellt.

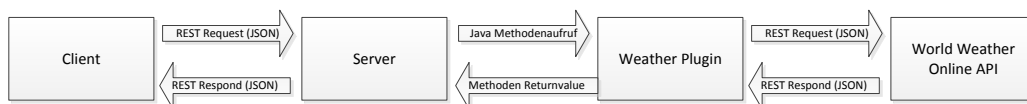


Abbildung 4.3: Client Wetterabruf

Das Wetter Plugin stellt folgende Schnittstelle zur Verfügung.

```
public interface WeatherService {
    public Weather requestWeather(String city, String country);
}
```

Listing 4.7: weatherapi

4.2.7 Facebook Plugin

Für die Realisierung des Facebook Plugins wurde die RestFB Bibliothek verwendet. Listing 4.8 beschreibt wie eine Nachricht auf der Pinnwand eines Facebook Benutzers erstellt wird und wie eigene Pinnwand Einträge abgerufen werden. Damit die Facebook API verwendet werden kann, muss zuvor ein Facebook Account erstellt werden, sowie ein Entwicklerkonto eingerichtet werden.

```
// Initialisierung des RestFB Clients mit Hilfe des Accesstoken.
private FacebookClient facebookClient =
    new DefaultFacebookClient(FACEBOOK_ACCESTOKEN);

// Zum versenden einer Nachricht an die Facebook ID des Empfängers.
facebookClient.publish(FACEBOOK_ID + "/feed",
    FacebookType.class,
    Parameter.with("message", "Dies_ist_die_Nachricht"));

// Zum abrufen der Pinwand Eintraege
Connection<Post> myFeed = facebookClient.fetchConnection("me/feed",
    Post.class,
    Parameter.with("limit", 100));
```

Listing 4.8: Senden und Abrufen von Facebook Feeds

Das Facebook Plugin realisiert das MessageService Interface.

```
public interface MessageService extends Serializable {
    ...
}

public class FacebookService implements MessageService{
    // Dient der Initialisierung des Plugins. (z.B. Username, Password, Accesstoken etc.)
    public void init(Properties props);
    // Eindeutiger Name des Plugins: Facebook
    String getName();
    // Zum erstellen eines Pinnwand Eintrages
```

```
void sendMessage(Message msg);  
// Zum abrufen eigener Pinnwand Eintraege  
List<Message> receiveMessage();  
  
}
```

Listing 4.9: Message Service Api

4.2.8 Sicherheit

Security Rollen

Die Berechtigungssteuerung aus Sicht der Anwendung stützt sich auf sogenannten Security Roles ab. Sie stellen virtuelle Gruppierungen von Usern dar, wobei in der Anwendung nur der Rollenname bekannt ist. Die Zuordnung von Usern zu den Rollen ist Aufgabe des eingesetzten Application Servers.

Für die Zuordnung von Usern zu Rollen wird ein Directory Server (ApacheDS) verwendet. Der Directory Server ist mit dem Applikation Server verbunden.

Die Zuordnung der Rollen zu den Gruppen im Directory Server erfolgt innerhalb des Deployment Descriptors (sun-web.xml bzw glassfish-web.xml).

```
<security-role-mapping>  
  <role-name>admin</role-name>  
  <group-name>appadmin</group-name>  
</security-role-mapping>  
<security-role-mapping>  
  <role-name>user</role-name>  
  <group-name>appuser</group-name>  
</security-role-mapping>
```

Listing 4.10: web.xml

Zugriffsregeln

Die Seiten einer Webanwendung sind zunächst öffentlich erreichbar. Da dies nicht gewünscht ist, sind hierfür security-constraint-Elemente im Deployment Descriptor (web.xml) der Anwendung einzutragen. Durch die Angabe von web-resource-collection kann eine Teilmenge der Webseiten festgelegt werden, welche Passwort geschützt sind. Wie man im Listing 4.11 sehen kann, werden drei Rollen festgelegt Admin, Caretaker und User.

```
<security-constraint>

<!-- Festlegen der URLs, welche nur von Admin betreten werden duerfen. -->
  <web-resource-collection>
    <web-resource-name>Admin Pages</web-resource-name>
    <url-pattern>/faces/admin/*</url-pattern>
  </web-resource-collection>

<!-- Rollen festlegen -->
  <auth-constraint>
    <role-name>admin</role-name>
  </auth-constraint>

  <!-- Eine SSL gesicherte Verbindung einrichten-->
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>

<security-constraint>

<web-resource-collection>
  <web-resource-name>Caretaker Pages</web-resource-name>
  <url-pattern>/faces/caretaker/*</url-pattern>
</web-resource-collection>

<auth-constraint>
  <role-name>caretaker</role-name>
</auth-constraint>

  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>

<security-constraint>
  <web-resource-collection>
    <web-resource-name>All Pages</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>user</role-name>
</auth-constraint>
</security-constraint>

<login-config>
  <auth-method>FORM</auth-method>
```

```
<realm-name>ldap</realm-name>
<form-login-config>
  <form-login-page>/faces/login.xhtml</form-login-page>
  <form-error-page>/faces/loginerror.xhtml</form-error-page>
</form-login-config>
</login-config>
```

Listing 4.11: web.xml

Deklarative Security

Damit bestimmte Funktionen/Methoden nur von berechtigten Usern ausgeführt werden, sind diese durch folgende Annotationen eingeschränkt worden.

@RolesAllowed Zugriffserlaubnis nur für bestimmte Security-Rollen

@PermitAll Zugriffserlaubnis für alle Benutzer

@DenyAll Zugriffsverbot für alle Benutzer

```
@Stateless
@PermitAll
public class SecurityServiceBean
{
  @RolesAllowed("admin")
  public void deleteBox()
  {
  }

  @RolesAllowed("caretaker")
  public void createNewBox()
  {
  }
}
```

Listing 4.12: Deklarative Security

Client Zertifikat Erstellung und Austausch

Die JDK von Java bietet ein Programm welches Keytool heißt, mit dessen Hilfe können selbst signierte Zertifikate erstellt werden. Sowohl für den Server als auch für den Client wurde ein Zertifikat erstellt. Diese wurden untereinander ausgetauscht und für die Erstellung der gesicherten Verbindung genutzt.

Persistenz

Für das erstellen, löschen, aktualisieren und lesen der verschiedenen Entitäten des Datenmodells wird ein Objekt Relationaler Mapper verwendet. Damit die Java Persistenz API (JPA) diese Aufgabe übernehmen kann, sind die Entitäten mit Annotation zu versehen sowie eine persistence.xml anzulegen. In der persistence.xml stehen die Datenbank spezifischen Einstellungen wie Zugangsdaten, Schema, Datenbankbezeichnung, JDBC Treiber und weitere Eigenschaften drin.

Bei der Annotation der Klassen werden unter anderem die Relation der Entitäten untereinander festgelegt sowie die Beziehungsarten (One-To-One, One-To-Many oder Many-To-Many). Das Listing 4.13 zeigt die Entität Caretaker, welche eine N zu M Beziehung zu der Entität Box besitzt. Als Primärschlüssel wird eine Identität vom Datentyp Long angenommen. Jede Entität muss das Interface Serializable implementieren. NamedQueries sind sogenannte Prepared Statements, mit ihrer Hilfe müssen SQL Anweisung nicht wiederholt geparsed werden.

```
@Entity
@NamedQueries({
    @NamedQuery(name = Caretaker.FIND_ALL,
                query="select_c_from_Caretaker_c"),
    @NamedQuery(name = Caretaker.FIND_BY_NAME,
                query="select_c_from_Caretaker_c_where_c.username_=:name")
})

public class Caretaker implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @ManyToMany // (mappedBy = "parent",
                cascade=CascadeType.ALL, orphanRemoval=true)
    private Set<Box> settopboxes=new HashSet<Box>();
```

Listing 4.13: caretaker.java



Abbildung 4.4: Client des Seniors

4.3 Client Realisierung

4.3.1 Sprachaufzeichnung

Für die Aufnahme der Stimme des Seniors wird das integrierte Mikrofon des Laptops verwendet. Mithilfe der Java Audio API wird die Schnittstelle des Mikrofons angesprochen. Die Aufzeichnung der Stimme erfolgt mit einer Abtastrate von 44,1 kHz in Stereo. Folgendes Listings beschreiben kurz wie man mit der Java Audio API eine Sprachaufzeichnung angefertigt 4.14 und wie eine Sprachaufzeichnung 4.15 abgespielt wird.

```
// Mikrofon auswählen und festlegen des Audioformates  
DataLine.Info info = new DataLine.Info(TargetDataLine.class, audioFormat);  
SourceDataLine line = (SourceDataLine) AudioSystem.getLine(info);  
// Sprachaufzeichnung starten
```



```

line.start();
while (thread != null){
    if ((numBytesRead = line.read(data, 0, bufferLengthInBytes)) == -1) {
        break;
    }
    outputStream.write(data, 0, numBytesRead);
}

```

Listing 4.14: Aufnahme einer Sprachaufzeichnung

```

// Lautsprecher auswaehlen und festlegen des Audioformates
DataLine.Info info = new DataLine.Info(TargetDataLine.class, audioFormat);
SourceDataLine line = (SourceDataLine) AudioSystem.getLine(info);
// Wiedergabe der Sprachaufzeichnung
line.open(audioFormat, bufSize);
while (thread != null) {
    try {
        if ((numBytesRead = playbackInputStream.read(data)) == -1) {
            break;
        }
        int numBytesRemaining = numBytesRead;
        while (numBytesRemaining > 0) {
            numBytesRemaining -= line.write(data, 0, numBytesRemaining);
        }
    } catch (Exception e) {
        shutdown("Error_during_playback:_" + e);
        break;
    }
}

```

Listing 4.15: Abspielen einer Sprachaufzeichnung

TargetDataLineClass entspricht hierbei das Mikrofon und SourceDataLineClass die Lautsprecher. Sowohl die Aufnahme als auch die Wiedergabe einer Sprachaufzeichnung erfolgen in Threads. Dies ist notwendig, da sonst keine Interaktion mit der GUI möglich wäre.

Die Audio Service Komponente bietet folgende Schnittstelle an.

```

public interface AudioService {
    public void startPlayback();
    public void stopPlayback();
    public void startRecord();
    public void stopRecord();
    public void loadRecord(File src);
    public void saveRecord(File target, AudioFileFormat.Type type);
}

```

Listing 4.16: AudioService Schnittstelle

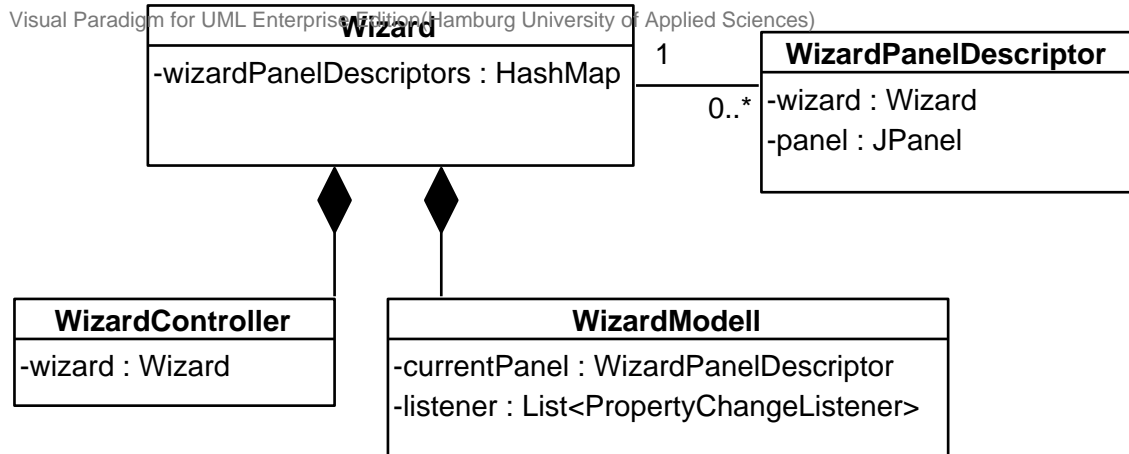


Abbildung 4.5: Wizard Klassendiagramm

4.3.2 Grafische Benutzerschnittstelle

Die grafische Benutzerschnittstelle der Set-Top-Box wurde mit Java Swing realisiert. In diesem Abschnitt wird beschrieben wie das MVC Muster umgesetzt wurde.

Die GUI besteht aus einer Reihe von verschiedenen Views. Für die Views werden Java Swing JPanels verwendet. Zu jedem View gehört ein Controller, welcher die Aktion des Benutzer entgegen nimmt, interpretiert und an das darunter liegende Modell weiterleitet. Eine Ausnahme bildet hierbei die View des Hauptmenüs. Diese besitzt kein Modell, da das Hauptmenü keinen Zustand besitzt. Jede View wird von einem WizardPanelDescriptor ummantelt (gewrapped). Die Menge aller WizardPanelDescriptors bildet einen zyklischen Graphen. Der Graph entsteht durch die Verknüpfung von WizardPanelDescriptors. Die WizardPanelDescriptors legen damit die Navigationsreihenfolge fest. Dadurch enthalten die Views keine Navigation Logik und werden nur für die Anzeige von Bedienelement genutzt (Separation of concerns). Die WizardPanelDescriptors werden durch den Wizard verwaltet. Der WizardController reagiert auf Events welche durch die Fernfernbedienung ausgelöst werden und entscheidet welcher View angezeigt werden soll. Das WizardModell enthält den aktuellen Zustand, also welche View gerade angezeigt wird. Sollte es zu einer Zustandsänderung in den Modells kommen, so informieren diese ihre Observar. Abbildung 4.5 zeigt das Klassendiagramm für den Wizard an. In Abbildung 4.6 wird das MVC Muster beispielhaft demonstriert.

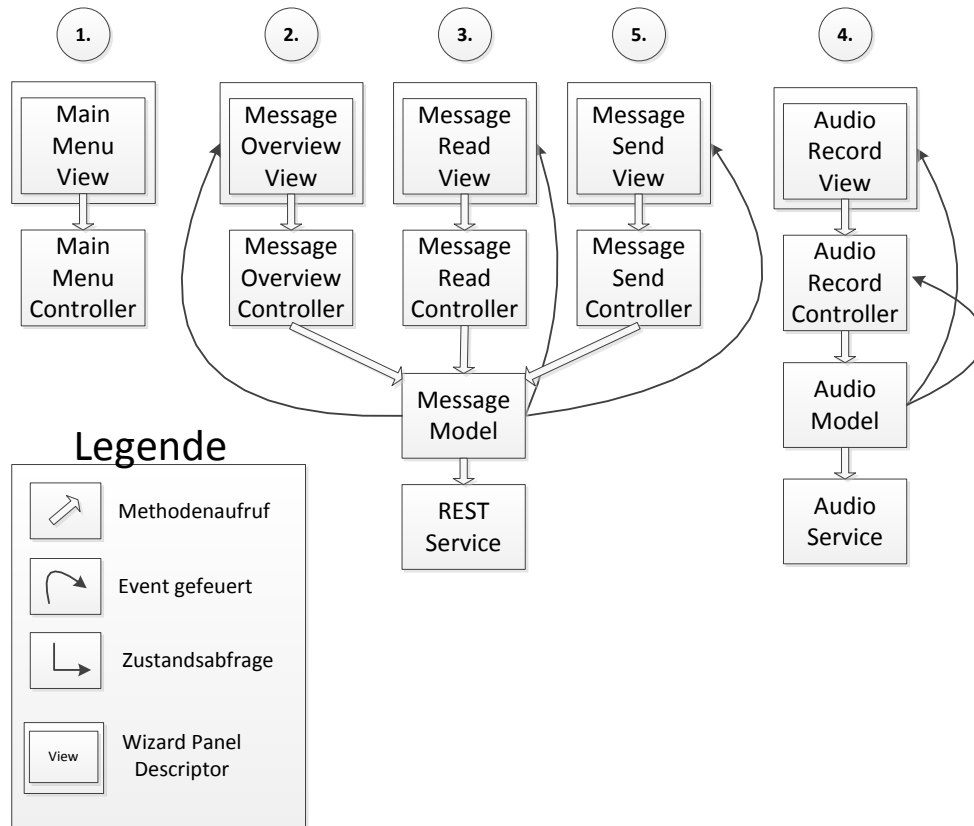


Abbildung 4.6: Umsetzung MVC

4.4 Fazit

Dieses Kapitel hat gezeigt, dass das Design zum größten Teil realisierbar ist. Bei der Realisierung des Entwurfs ergaben sich jedoch eine Reihe von Problemen, welche bei der Erstellung des Designs nicht vorgesehen waren. Das größte Problem ergab sich bei der Verwendung des Plugin Systems. Beim Plugin System war es vorgesehen, dass Plugins sich zur Laufzeit in das System einfügen lassen. Dies war jedoch aufgrund der Technik des Applikationsservers nicht möglich. Plugins können nur vor dem Start des Servers eingefügt werden. Ein weiteres Problem war die Erstellung von grafischen Schnittstellen für die Plugins, mit dessen Hilfe die Plugins konfiguriert werden sollten. Für jedes Plugin sollte eine individuelle grafische Schnittstelle entwickelt werden, was sich jedoch schwer umsetzen ließ. Deshalb wurde eine einheitliche grafische Schnittstelle für alle Plugins entwickelt.

5 Zusammenfassung

Ziel der Arbeit war es einen Prototypen zu entwickeln, mit dessen Hilfe sich ältere Menschen im Internet informieren, sowie mit Verwandten kommunizieren können. Der Prototyp besteht aus einer Set-Top-Box, einer Fernbedienung, einem Mikrofon und einem Server. Die Fernbedienung diente zur Bedienung der Set-Top-Box und das Mikrofon zum Beantworten von Nachrichten. Im Vordergrund dieser Arbeit stand eine benutzerfreundliche Bedienoberfläche.

Im zweiten Kapitel wurden zunächst die grundsätzlichen Probleme der Senioren beschrieben sowie eine Anforderungsanalyse durchgeführt aus der die funktionalen Anforderungen hervorgingen.

Im dritten Kapitel wurde eine Architektur für das System festgelegt und ein Entwurf für die Realisierung erarbeitet.

Im vierten Kapitel wurde schließlich der Entwurf durch die Implementierung des Clients sowie des Servers realisiert.

Abschließend kann gesagt werden, dass das entwickelte System das Potential hat, wie im dargestellten Szenario zu funktionieren. Die gestellten Anforderungen konnten mit einigen Ausnahmen erfüllt werden. Allerdings ist unklar ob dieses System von Senioren genutzt werden kann. Hierzu müsste eine Feldstudie durchgeführt werden.

Ausblick

Das Projekt Smart Senior @ Home hat bereits gezeigt, dass Senioren bereit sind Informationssysteme zu nutzen. Es steht also nicht in Frage ob die Entwicklung eines Assistenzsystems für Senioren sinnvoll ist, sondern wie solch ein System aussehen soll. In dieser Arbeit wurde die Set-Top-Box mithilfe einer Fernbedienung bedient, während beim Projekt Smart Senior @ Home ein Webpad verwendet wird. Ob die Fernbedienung oder das Webpad die geeignetste Form der Interaktion mit dem System ist muss in Studien belegt werden. Eines ist jedoch klar, die Senioren von Heute sind nicht mit zukünftigen Senioren vergleichbar (Siehe Digital Immigrant vs Digital Natives). Das aber Informationssysteme hierbei eine verstärkte Rolle spielen wird ist aufgrund der demographischen Entwicklung unbestreitbar. Meines

Erachtens nach wird sich die Vision von Mark Weiser verwirklichen (Ubiquitous Computing) und Computer verschwinden aus unseren Augen, aber nicht aus unserem Leben. Computer in Kleidung, im Körper und in Möbeln.

Literaturverzeichnis

- [A.Hart 2013] A.HART, Traci: *Evaluation of Websites for Older Adults*. 2013. – URL http://www.researchgate.net/publication/200553037_Evaluation_of_websites_for_older_adults_-_how_senior_friendly_are_they. – [Online; Stand 23. Februar 2013]
- [Andrew S. Tanenbaum 2007] ANDREW S. TANENBAUM, Maarten van S.: *Verteilte Systeme: Prinzipien und Paradigmen*. Pearson Studium IT, 2007
- [ARD-ZDF 2013] ARD-ZDF: *76 Prozent der Deutschen online neue Nutzungssituationen durch mobile Endgeräte*. 2013. – URL http://www.ard-zdf-onlinestudie.de/fileadmin/Online12/0708-2012_Eimeren_Frees.pdf. – [Online; Stand 23. Februar 2013]
- [Bernhard Reindl 2009] BERNHARD REINDL, Dieter K.: *Wohnen im Alter*. Beuth Verlag GmbH DE, 2009
- [Bmfsfj 2013] BMFSFJ: *Berichte zur Lage der älteren Generation Umweltprämie*. 2013. – URL <http://www.bmfsfj.de/RedaktionBMFSFJ/Abteilung3/Pdf-Anlagen/fuenfter-altenbericht,%20property=pdf,bereich=,rwb=true.pdf>. – [Online; Stand 23. Februar 2013]
- [Burkhart Brückner 2006] BURKHART BRÜCKNER, Ulrich K.: *Verstehende Beratung alter Menschen*. S.Roderer-Verlag, 2006
- [Corporate-eye 2013] CORPORATE-EYE: *Computer Anxiety & Older Adults: Using Self-Efficacy to Break Usage Barriers*. 2013. – URL <http://www.corporate-eye.com/blog/2012/08/computer-anxiety-older-adults/>. – [Online; Stand 23. Februar 2013]
- [digitalechancen 2013] DIGITALECHANCEN: *Digitale Agenda: Smartphones und Fernbedienungen helfen Senioren und Menschen mit Behinderungen im Haushalt*. 2013. – URL <http://www.digitale-chancen.de/content/stories/index.cfm/aus.2/key.3002/secid.1/secid2.0>. – [Online; Stand 23. Februar 2013]
- [Faltermaier 1992] FALTERMAIER, Tone: *Entwicklungspsychologie des Erwachsenenalters*. Kohlhammer:Berlin,Stuttgart, 1992

- [Hans Rudolf Schelling 2010] HANS RUDOLF SCHELLING, Alexander S.: Internet-Nutzung im Alter: Gründe der (Nicht-)Nutzung von Informations- und Kommunikationstechnologien (IKT) durch Menschen ab 65 Jahren in der Schweiz. In: *Zürcher Schriften zur Gerontologie* 7 (2010)
- [Hertwig 2009] HERTWIG, Harald: *Ins Heim oder daheim*. Wilhelm Heyne Verlag, 2009
- [ITWissen 2013a] ITWISSEN: *Client-Server-Architektur*. 2013.
– URL <http://www.itwissen.info/definition/lexikon/Client-Server-Architektur-C-S-client-server-architecture.html>.
– [Online; Stand 23. Februar 2013]
- [ITWissen 2013b] ITWISSEN: *Drei-Schichten-Architektur*. 2013.
– URL <http://www.itwissen.info/definition/lexikon/Drei-Schichten-Architektur-three-tier-architecture.html>. – [Online; Stand 23. Februar 2013]
- [Kuhlmann 2013] KUHLMANN, Verena: *SMS vs. Telefonat*. 2013. – URL <http://jetzt.sueddeutsche.de/texte/anzeigen/530850>. – [Online; Stand 23. Februar 2013]
- [Lee 2013] LEE, Jason: *Writing Pluggable Java EE Applications*. 2013. – URL <http://blogs.steeplesoft.com/2012/07/writing-pluggable-java-ee-applications-the-explanation/>. – [Online; Stand 23. Februar 2013]
- [Leupolz 2010] LEUPOLZ, Lisa: *Social Network Sites für Ältere: Motive und Determinanten zur Nutzung von Social Network Sites bei der Generation 50plus*, Augsburg, Univ., Diplomarbeit, 2010. – URL <http://books.google.de/books?isbn=382581839X>
- [Müller 2006] MÜLLER, Wolfgang: *Die neue Traumzielgruppe*. 2006. – URL http://www.focus.de/finanzen/news/50Plus/generation-50plus_aid_22269.html. – [Online; Stand 23. Februar 2013]
- [Nielsen 1994] NIELSEN, Jakob: *Usability Engineering*. Morgan Kaufmann, 1994
- [Pfaff 2013] PFAFF, Heiko: Pflege im Rahmen der Pflegeversicherung Deutschlandergebnisse. In: *Statistisches Bundesamt* 5224001119004 (2013). – URL https://www.destatis.de/DE/Publikationen/Thematisch/Gesundheit/Pflege/PflegeDeutschlandergebnisse5224001119004.pdf?__blob=publicationFile
- [Phang u. a. 2006] PHANG, Chee W. ; SUTANTO, J. ; KANKANHALLI, A. ; LI, Yan ; TAN, B.C.Y. ; TEO, Hock-Hai: Senior Citizens' Acceptance of Information Systems: A Study in the Context of e-Government Services. In: *Engineering Management, IEEE Transactions on* 53 (2006), nov., Nr. 4, S. 555–569. – ISSN 0018-9391

- [Rogers 2013] ROGERS, Chris: *Web Audio API*. 2013. – URL <https://dvcs.w3.org/hg/audio/raw-file/tip/webaudio/specification.html>. – [Online; Stand 23. Februar 2013]
- [SmartSenior 2013] SMARTSENIOR: *SmartSenior: Intelligente Dienste und Dienstleistungen für Senioren*. 2013. – URL http://www.izm.fraunhofer.de/content/dam/izm/de/documents/News-Events/News/2012/SmartSenior%20Projektreport_2012-09-07_final.pdf. – [Online; Stand 23. Februar 2013]
- [Starke 2011] STARKE, Gernot: *Effektive Softwarearchitekturen: Ein praktischer Leitfaden*. Carl Hanser Verlag GmbH & CO. KG, 2011
- [taz 2013] TAZ: *Computer leicht gemacht*. 2013. – URL <http://www.taz.de/!75600/>. – [Online; Stand 23. Februar 2013]
- [umweltbundesamt 2013] UMWELTBUNDESAMT: *Demographischer Wandel*. 2013. – URL <http://www.umweltbundesamt-daten-zur-umwelt.de/umweltdaten/public/theme.do?nodeIdent=2273>. – [Online; Stand 23. Februar 2013]
- [Weil 2012] WEIL, Dirk: *Java EE 6: Enterprise-Anwendungsentwicklung leicht gemacht*. entwickler.press, 2012
- [Wikipedia 2013a] WIKIPEDIA: *Selbstwirksamkeitserwartung*. 2013. – URL <http://de.wikipedia.org/wiki/Selbstwirksamkeitserwartung>. – [Online; Stand 23. Februar 2013]
- [Wikipedia 2013b] WIKIPEDIA: *Soziale Software*. 2013. – URL http://de.wikipedia.org/wiki/Soziale_Software. – [Online; Stand 23. Februar 2013]
- [Zoch 2008] ZOCH, Annette: *Mediennutzung von Senioren: Eine qualitative Untersuchung zu Medienfunktionen, Nutzungsmustern und Nutzungsmotiven*, München, Univ., Diplomarbeit, 2008. – URL <http://books.google.de/books?isbn=382581839X>

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 28. Februar 2013

Ort, Datum

Unterschrift