

Entwicklung eines mobilen Software-Assistenten zur Unterstützung der vernetzten Pflege

Master-Arbeit
von

Thomas Knapp

An der Fakultät für Wirtschaftswissenschaften
Institut für Informationswirtschaft und -management (IISM)
Information & Market Engineering

Gutachter: Prof. Dr. rer. pol. Christof Weinhardt

Betreuer Assistent: Dr. Henner Gimpel

Betreuende Mitarbeiter FZI: Bruno Rosales Saurer

Mathias Schmon

Imanol Bernabeu

31. März 2012

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
Tabellenverzeichnis	v
Abkürzungsverzeichnis	vi
1. Einleitung	1
2. Ziele der Arbeit	3
3. Analyse der Anforderungen und daraus abgeleitete Entwurfsentscheidungen	6
3.1. Analyse der Anforderungen	6
3.2. Entwurfsentscheidungen des Gesamtsystems auf Basis der erhobenen Anforderungen	10
4. Basisanwendung und Plug-Ins zur Realisierung der identifizierten Anforderungen	15
4.1. Aufgaben und Struktur der Basisanwendung	15
4.2. Einbinden von Plug-Ins in die Basisanwendung	21
4.3. Das Plug-In <i>Kontakte</i>	25
4.4. Das Plug-In <i>Touren</i>	28
5. Evaluation	39
5.1. Evaluation der technischen Umsetzung der mobilen Anwendung	39
5.2. Evaluation von Nutzbarkeit, Funktionsumfang und Akzeptanz anhand einer Zielgruppenschulung	44
5.2.1. Vorwissen der Teilnehmer und vorbereitende Maßnahmen	45
5.2.2. Ermittlung des Status Quo - Ablauf und Schlussfolgerungen	48
5.2.3. Anforderungen an CareNetmobile aus Sicht der Zielnutzergruppe	53
5.2.4. Ergebnisse des Abschlussfragebogens	57
6. Fazit	61
7. Erklärung	65
Anhang	66

A.	Kernanforderungen an eine mobile Anwendung zur Dokumentation ärztlich übertragener Tätigkeiten	66
B.	Liste grundsätzliche Entwurfsentscheidungen CareNet <i>mobile</i>	67
C.	Basisanwendung	68
D.	Plug-In-Struktur	69
E.	Plug-In Kontakte	69

Abbildungsverzeichnis

2.1. Zielkomponenten der Arbeit	4
3.1. Verwendete Technologien in der Gesamtarchitektur	11
3.2. Hybride Anwendungsentwicklung mit PhoneGap	13
4.1. Struktur der grafischen Oberfläche von CareNetmobile	16
4.2. Vergleich zweier Beispielimplementierungen von Java und JavaScript	17
4.3. Überprüfung der Benutzerdaten in CareNetmobile	18
4.4. Struktur eines Plug-Ins in CareNetmobile	21
4.5. Übersicht der verwendbaren der Plug-Ins	23
4.6. Grundstruktur einer Plug-In-JavaScript-Datei	24
4.7. Tabellen in der lokalen Datenbank des Plug-Ins <i>Kontakte</i>	26
4.8. Ausschnitt der Darstellung von Kontakten innerhalb einer Sicht	27
4.9. Datenmodell und zugehörige Datenbanken im Plug-In <i>Touren</i>	29
4.10. Übersicht der abzuarbeitenden Stationen einer Tour	31
4.11. Übersicht der zu erbringenden Dienste einer Station	32
4.12. Darstellung einer Standardleistung	33
4.13. Darstellung der Schaltflächen zur Wunddokumentation	34
4.14. Darstellung der Schaltflächen zur Erfassung von Blutdruck und Puls	35
4.15. Darstellung der Klientenenakte im Plug-In <i>Touren</i>	36
5.1. Interne und externe Charakteristika von Software-Produktqualität	40
5.2. Ergebnisse des Fragebogens zur Einstellung gegenüber Computern	46
5.3. Entwicklung eines Diagramms zur Ermittlung des Tagesablaufs mit den Teilnehmern	48
5.4. Ergebnisse der Fragen zur Benutzbarkeit von CareNetmobile	58
5.5. Ergebnisse der Fragen zum benötigten Vorwissen und Lernaufwand zur Nutzung von CareNetmobile	59
5.6. Ergebnisse der Fragen zu potentiellem Nutzen und Akzeptanz von CareNetmobile	60

Tabellenverzeichnis

- 5.1. Liste der identifizierten Probleme bei der Dokumentation von Pflegeleistungen in der Residenz Bad Friedrichshall 52

Abkürzungsverzeichnis

VERAH	Versorgungsassistentin in der Hausarztpraxis
MFP	Medizinische Fachpflegekraft
DRK	Deutsches Rotes Kreuz
GPS	Global Positioning System
WLAN	Wireless Local Area Network
FZI	Forschungszentrum Informatik
REST	Representational State Transfer
URI	Uniform Resource Identifier
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
XML	Extensible Markup Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JSON	Java Script Object Notation
App	Application
ID	Identifier
EDV	Elektronische Datenverarbeitung
CAS	Computer Attitude Scale
AJAX	Asynchronous JavaScript and XML
UTAUT	Unified Theory of Acceptance and Use of Technology
ISO	International Standards Organization
IEC	International Electrotechnical Commission

1. Einleitung

Nach Schätzungen des Statistischen Bundesamtes steigt der Anteil der über 65jährigen in der Bundesrepublik Deutschland von 19% im Jahr 2005 auf 30% im Jahr 2050 [1] bei gleichzeitig steigender Lebenserwartung. Dieser demographische Wandel birgt große Herausforderungen für das Gesundheitswesen, da eine größere Anzahl an Patienten mit altersbedingten Krankheiten versorgt werden muss, die häufig multimorbide und immobil sind.

Insbesondere Hausarztpraxen als zentrale Anlaufstellen für Beschwerden im Alter stehen vor neuen Aufgaben. Wo früher nur Heilkunde angewandt wurde, werden heute beispielsweise auch koordinative Aufgaben zwischen Fachärzten, Pflegediensten und Apotheken übernommen. Mit einer gestiegenen Anzahl an pflegebedürftigen Personen wird sich dieser Trend weiter verstärken. Hinzu kommt, dass häufigere Immobilität der älteren Patienten von den Hausärzten mehr zeitaufwendige Hausbesuche abverlangen wird. Dem gegenüber steht eine sinkende Anzahl an Hausärzten, da auch deren Altersstruktur vom demographischen Wandel geprägt ist. Nach Schätzungen wird die Anzahl der Hausärzte bis 2020 bundesweit um etwa 7000 zurückgehen, was einem Anteil von 13.3% entspricht ([2, S. 65]). Überdurchschnittlich hoch wird der Rückgang an Hausärzten in den ländlichen Regionen der neuen Bundesländer ausfallen ([2, S. 52 ff.]). Der Rückgang der Hausärzte wird somit bundesweit zu signifikanten Versorgungsengpässen führen, die kurzfristig nicht durch junge Ärzte ausgeglichen werden können.

Um auch in Zukunft eine ausreichende Versorgung von Patienten gewährleisten zu können, wurde vom Gemeinsamen Bundesausschuss eine Richtlinie herausgegeben, die Rahmenbedingungen formuliert, in denen ärztliche Tätigkeiten auf Berufsangehörige der Alten- und Krankenpflege übertragen werden können [3]. Diese Maßnahme soll Hausärzte bei Tätigkeiten entlasten, für die nicht unbedingt ärztliche Expertise notwendig ist. Insbesondere die Anzahl der vom Arzt durchzuführenden Hausbesuche könnte sich hierdurch verringern. Ärztliche Tätigkeiten können dabei sowohl an Arzthelfer in einer Praxis, Mitarbeiter eines ambulanten Pflegedienstes, Pfleger in einem Alten- und Pflegeheim sowie Gesundheits-

und Krankenpflegern in einem Krankenhaus delegiert werden. Vor der Herausgabe der Richtlinie im März 2012 wurde bereits in verschiedenen Modellprojekten der Nutzen von Delegation evaluiert. Mit dem Modellprojekt AGnES [4] beispielsweise ließ sich belegen, dass durch Delegation positive Effekte bzgl. Effizienz und Qualität der ärztlichen Leistungen erzielt werden können. Um diese Effizienz- und Qualität zu erreichen, benötigen die Berufsangehörigen der Alten- und Krankenpflege allerdings gewisse Zusatzqualifikationen, die im Rahmen verschiedener Weiterbildungsprogramme erworben werden können. Beispiele hierfür sind VERAH ([5] und [6]) oder MFP [QUELLE]. In einem ca. 200stündigen Fortbildungsprogramm werden unter anderem Kenntnisse aus den Bereichen Case Management, Gesundheits-, Praxis- und Wundmanagement sowie IT vermittelt.

Das Gesamtziel des Gesetzgebers ist, eine integrierte Versorgung von Pflegebedürftigen zu erreichen, bei der die Parteien des Gesundheitssystems (Ärzte, Pflegekräfte, Krankenkassen) intensiv zusammenarbeiten sollen. Eine enge Zusammenarbeit bedeutet einen hohen Koordinationsaufwand und damit verbunden ein hohes Maß an Informationsaustausch. Um das volle Potential der Delegation ärztlicher Leistungen ausschöpfen zu können, müssen entsprechende EDV-Systeme unterstützend eingesetzt werden. Zwar werden auch heute schon vielfältige Verwaltungssysteme zur Dokumentation von entweder ärztlichen Tätigkeiten oder Pflegetätigkeiten eingesetzt, jedoch ist das Angebot an Software zur Unterstützung einer Zusammenarbeit zwischen Ärzten und Pflegedienstleistern gering. Gerade wenn es darum geht, vor Ort beim Klienten Vorgänge zu dokumentieren, wird häufig zunächst auf Papier dokumentiert und die Information erst zu einem späteren Zeitpunkt in ein EDV-System übertragen. Diese Medienbrüche Kosten Zeit und in der Folge Geld und die Fehlerquellen bei der manuellen Übertragung von Informationen sind vielfältig.

Diese Master-Arbeit setzt an der Schnittstelle zwischen Ärzten und Berufsangehörigen der Alten- und Pflegeberufe an.

[Beschreibung der einzelnen Kapitel.]

2. Ziele der Arbeit

Ziel dieser Master-Arbeit ist es, ein elektronisches System zur Unterstützung der Zusammenarbeit von Ärzten und Berufsangehörigen der Alten- und Pflegeberufe, die eine Zusatzqualifikation zur Ausübung ärztlicher Tätigkeiten haben, zu entwerfen, zu implementieren und zu evaluieren. Einerseits soll eine Delegation von Aufgaben möglich sein, andererseits sollen die ausgeführten Tätigkeiten elektronisch dokumentiert werden können. Da die ärztlichen Tätigkeiten, die delegiert werden sollen, vornehmlich vor Ort beim Klienten durchgeführt werden, lassen sich im Kern zwei notwendige Komponenten identifizieren: ein zentrales Verwaltungssystem und eine mobile Anwendung.

Abbildung 2.1 zeigt eine stilistische Darstellung der Komponenten der Arbeit. Die Basis bildet ein zentrales System, in welchem die Daten zu Klienten, auszuführenden und ausgeführten Tätigkeiten verwaltet werden. Daneben gibt es eine Software auf einem mobilen Endgerät, die es ermöglicht, Informationen über auszuführende Tätigkeiten abzurufen und nach der Ausführung die Ergebnisse zu dokumentieren. Verbunden werden beide Systeme über eine Middleware, die Anfragen an das Verwaltungssystem über verschiedene Schnittstellen ermöglichen soll. Da die Kommunikation zwischen der mobilen Anwendung und dem Verwaltungssystem in dieser Arbeit aber lediglich über einen Schnittstellentyp läuft, dient die Middleware vor allem als Platzhalter, um zu einem späteren Zeitpunkt weitere Schnittstellen implementieren zu können. Die Implementierung weiterer Schnittstellen ist nicht Teil dieser Arbeit.

Idee der direkten elektronischen Dokumentation beim Klienten ist die Vermeidung von Medienbrüchen. Anstatt zunächst auf Papier zu dokumentieren und diese Informationen zu einem späteren Zeitpunkt zu übertragen, sollen die Informationen über eine internetgestützte Schnittstelle direkt an das Verwaltungssystem übertragen werden. Ziel ist eine Einsparung der Zeit, die dafür benötigt wird, die schon dokumentierten Informationen noch einmal manuell zu übertragen. Außerdem wird eine Reduzierung möglicher Fehlerquellen angestrebt, um die Qualität der Dokumentation zu erhöhen.

Die Entwicklung der mobilen Anwendung ist Kern der Arbeit. Vorgabe ist, Tablet-PCs zur Dokumentation einzusetzen, da sie den momentan neuesten Stand der Technik darstellen,

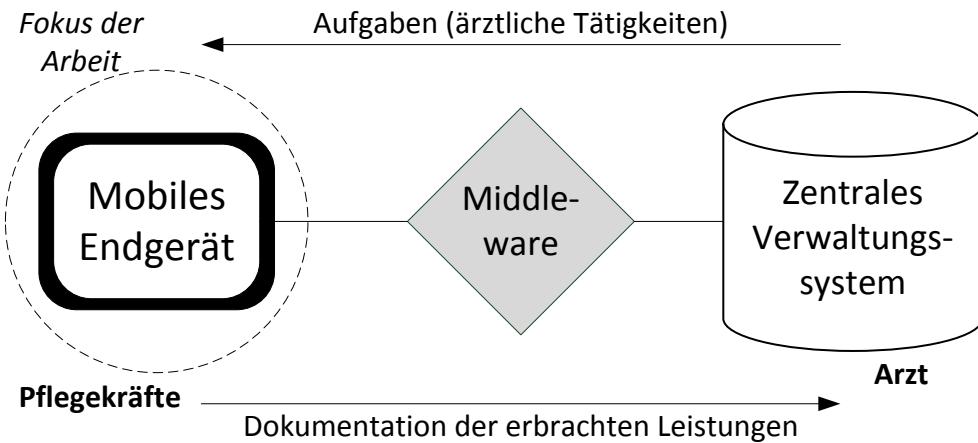


Abbildung 2.1.: Zielkomponenten der Arbeit

übersichtlich und intuitiv bedienbar sind. Welche Plattform (z.B. Android oder iOS) und welche Hardware genau eingesetzt wird, ist offen und wird sich aus den zu ermittelnden Anforderungen ableiten.

Der Entwicklungsprozess gliedert sich in eine Anforderungsanalyse, die Implementierung und eine Evaluation der entstandenen Software und folgt somit vornehmlich einem iterativen Entwicklungsmodell. Es findet vorerst eine auf Literatur basierende Anforderungsanalyse sowie eine Priorisierung nach technischen und zeitlichen Kriterien statt. Die Implementierung konzentriert sich zunächst auf die Umsetzung der wichtigsten Anforderungen, um einen ausreichend reifen Stand der Software zu erreichen.

Die Evaluation gliedert sich **drei (ja?)** Teile. Nach etwa der halben Bearbeitungszeit soll ein erster Prototyp der Software (auf Basis der in der Literatur gefundenen Anforderungen entwickelt) im Rahmen einer Schulung zur Medizinischen Fachpflegekraft (MFP) den Teilnehmern vorgestellt. Hier sollen Akzeptanz, Benutzbarkeit sowie erwarteter Nutzen der Software evaluiert werden. Die hier gesammelte Resonanz wird in die weitere Entwicklung einfließen.

Neben den fachlichen Anforderungen soll die technische Umsetzung der Software evaluiert werden. Hierzu wird die verwendete Architektur nach Kriterien der ISO-25000 (Software Engineering) [7] geprüft.

Im letzten Teil der Evaluation wird überprüft, ob ein typischer Prozess **[oder eine Leistung]** aus der Richtlinie zur Delegation ärztlicher Leistungen mit der entstandenen Software delegierbar und dokumentierbar ist.

Das zentrale Verwaltungssystem wird in dieser Arbeit nicht von Grund auf entwickelt, sondern basiert auf einer im Projekt VitaBit ([8] und [9]) entwickelten Software-Lösung zur Dokumentation von Pflegeleistungen. Bereitgestellt wird es von der nubedian GmbH [10], die außerdem Unterstützung bei der Anpassung des Systems an die Anforderungen der Dokumentation ärztlicher Leistungen anbietet, da das System momentan lediglich zur

Dokumentation von Pflegeleistungen gedacht ist. Im Laufe der Entwicklung der mobilen Anwendung sollen notwendige Anpassungen des zentralen Verwaltungssystem identifiziert und dokumentiert werden.

Neben der nubedian GmbH wird der Entwicklungsprozess dieser Arbeit von Mitarbeitern der DRK Residenz Bad Friedrichshall [11] unterstützt. Hier werden Pflegekräfte mit einem an VERAH angelehnten Fortbildungsprogramm zu Medizinischen Fachpflegekräften (MFP) ausgebildet. Um die Terminologie in dieser Arbeit zu vereinfachen, wird künftig anstatt von "Angehörigen der Alten- und Krankenpflege" oder "Medizinischen Fachpflegekraft" stellvertretend für alle Fortbildungsmöglichkeiten zur Erfüllung der Richtlinie nach § 63 Abs. 3c SGB V lediglich die Bezeichnung "Pflegekraft" verwendet.

3. Analyse der Anforderungen und daraus abgeleitete Entwurfsentscheidungen

In diesem Kapitel werden die aus Literatur und vorangegangenen Projekten hervorgegangenen Anforderungen an eine mobile Anwendung zur Dokumentation von ärztlich übertragenen Tätigkeiten vorgestellt. In Abschnitt 3.1 werden funktionelle Anforderungen beschrieben und technische Anforderungen daraus abgeleitet. Abschnitt 3.2 stellt Entwurfsentscheidungen vor, die zur Realisierung der Anwendung getroffen wurden.

3.1. Analyse der Anforderungen

Zu Beginn der Bearbeitungszeit der Master-Arbeit konnte kein Termin mit den eigentlichen Endnutzern der mobilen Anwendung vereinbart werden, um eine direkte Anforderungserhebung durchzuführen. Aufgrund dessen musste zunächst eine Anforderungsanalyse auf Basis von Literatur, aktuellen und vorangegangenen Arbeiten ergänzt um Gespräche mit mit sachkundigen Mitarbeitern von Kranken- und Pflegeeinrichtungen durchgeführt werden. Da die Grundlage für die Möglichkeit, ärztliche Tätigkeiten an speziell geschulte Pflegekräfte zu übertragen, die Richtlinie des Gemeinsamen Bundesausschusses basierend auf § 63 Abs. 3c SGB V ist, kann die Dokumentation der hier aufgelisteten übertragbaren Leistungen als Basisanforderung gesehen werden. Ein Problem ist aber, dass aus der Auflistung der Tätigkeiten nicht die zu dokumentierenden Daten hervorgehen. Diese konnten in der Literatur jedoch nicht im Detail ermittelt werden, da einige Leistungen medizinisches Fachwissen über deren Erbringung erfordern. Deswegen wurde zunächst angenommen, dass in der Regel eine Bestätigung, ob eine Leistung erbracht wurde, ausreicht, und sonst nur die Dokumentation von Blutdruck und Puls als wertbasierte Leistungen notwendig ist, sowie das Fotografieren von Wunden. Weitere zu dokumentierende Werte konnten erst bei einem Treffen mit den Endnutzern ermittelt werden. Die Ergebnisse werden in Kapitel 5.2 vorgestellt.

Das Ausbildungsprogramm zu einer Medizinische Fachpflegekraft (MFP) nimmt keine Einschränkung des Arbeitsgebietes vor. Eine Pflegekraft kann somit sowohl stationär in einer Pflegeeinrichtung arbeiten, als auch im ambulanten Dienst tätig sein, bei der die jeweiligen Klienten besucht werden. Die mobile Anwendung ist folglich so zu gestalten, dass beide Einsatzbereiche abgedeckt werden können. Es ist zu erwarten, dass die zu erbringenden Leistungen im stationären und im ambulanten Bereich die gleichen sein werden, jedoch können sich Unterschiede bei den technischen Anforderungen oder der Terminologie ergeben. Wird im Folgenden keine explizite Unterscheidung zwischen ambulanten und stationären Anforderungen genannt, so gilt eine Anforderung für beide Bereiche.

Jede ärztlich übertragene Tätigkeit kann direkt einer Person zugeordnet werden. Um eine Interaktion mit der Person zu ermöglichen, müssen bestimmte Daten über diese verfügbar sein. Hierzu gehören Stammdaten wie Name, Adresse, Anschrift und Alter, medizinische Daten über aktuell verabreichte Medikamente, Vorerkrankungen oder Allergien sowie Angaben zu Angehörigen, dem zuständigen Hausarzt und einer eventuellen Pflegebedürftigkeit. Aus der Notwendigkeit der Verfügbarkeit persönlicher Daten ergibt sich die Anforderung, diese Daten ausreichend zu schützen und nur dem jeweiligen Pflegepersonal zur Verfügung zu stellen. Folglich muss eine Authentifizierungsfunktion in Form einer Anmeldung mit Benutzername und Passwort (Log-In) implementiert werden. Ein Log-In ermöglicht es außerdem, der angemeldeten Pflegekraft nur die Informationen anzuzeigen, die sie für die Ausführung ihrer Tätigkeit benötigt (Autorsierungsfunktion). So kann ausgeschlossen werden, dass Informationen über Klienten verfügbar sind, die gerade nicht behandelt werden.

Es wird angenommen, dass Pflegekräfte in Schichten arbeiten, da eine Versorgung von Klienten Tag und Nacht gewährleistet sein muss. Weiter wird angenommen, dass jede Pflegekraft während einer Schicht für eine bestimmte Anzahl von Klienten zuständig ist, für die eine Menge vordefinierter Leistungen erbracht werden muss. Zu Beginn jeder Schicht bekommt eine Pflegekraft einen Plan ausgehändigt, in dem die abzuarbeitenden Stationen und die entsprechenden Leistungen aufgelistet sind. In diesem Plan muss jede Leistung dokumentiert werden (vgl. [12, S. 18 ff.]). Eine Leistung wird entweder durch das Setzen eines Hakens dokumentiert (Bestätigung) oder durch die Eintragung von Messwerten, beispielsweise im Falle der Erfassung von Blutdruck und Puls. Ziel der mobilen Anwendung ist es, dass sowohl die Übermittlung des Tagesplanes pro Pflegekraft als auch die Dokumentation der Leistungen elektronisch erfolgt. In einer Übersicht über alle Klienten/Stationen sollte direkt erkennbar sein, ob bereits alle Leistungen einer Station erbracht wurden, ohne die Einzelleistungen kontrollieren zu müssen.

Konnte eine Leistung nicht oder nicht vollständig erbracht werden, muss es die Möglichkeit geben, dies zu dokumentieren. Neben einer Markierung der Leistung muss ein Feld für eine Begründung (Freitext oder mit vordefinierten Bausteinen) der Nichterbringung der Leistung verfügbar sein. In der Übersicht über die Stationen einer Schicht sollte neben den Zuständen "Erledigt" und "Offen" außerdem der Zustand "Unvollständig abgeschlossen" ersichtlich sein, um zu zeigen, dass zwar alle Leistungen bearbeitet wurden, jedoch nicht alle erbracht werden konnten.

Bei der manuellen Dokumentation kann es immer wieder zu Fehlern kommen. Beispielsweise wenn Werte falsch eingetragen oder Leistungen bestätigt werden, die eigentlich noch nicht vollständig erfüllt sind. Da diese Fehler auch mit einer elektronischen Dokumentation nicht ausgeschlossen werden können, müssen einmal dokumentierte Leistungen stornierbar und korrigierbar sein, ohne dass im zentralen System eingegriffen werden muss. Um einer Verfälschung von Dokumentationsergebnissen durch nachträgliche Veränderungen beispielsweise von Messwerten entgegen zu wirken, sollte protokolliert werden, wer zu welchem Zeitpunkt eine Änderung vornimmt. Diese Protokollfunktion muss zur vollständigen Überwachung auch im zentralen System verfügbar sein und mit der mobilen Anwendung synchron gehalten werden.

Vor allem im ambulanten Bereich kann es immer wieder vorkommen, dass Klienten in Gegenden besucht werden müssen, in denen kein mobiles Internet verfügbar ist. Da der Datenaustausch zwischen mobiler Anwendung und zentralem Verwaltungssystem über eine Internetverbindung geschieht, wäre somit die Arbeit mit dem mobilen System nicht möglich. Darauf zu setzen, dass beim Klienten vor Ort eine lokale Internetverbindung genutzt werden kann, ist auch keine praktikable Lösung, da die Wahrscheinlichkeit für Probleme mit der Einrichtung und Verfügbarkeit hoch ist. Vor Ort beim Klienten anzukommen und festzustellen, dass der Plan der zu erbringenden Leistungen wegen Verbindungsproblemen nicht verfügbar ist, führt zu nicht hinnehmbaren Einschränkungen, da so eventuell wichtige medizinische Tätigkeiten nicht durchgeführt werden. Notwendig ist deshalb, den Plan für eine oder mehrere Schichten sowie alle relevanten Informationen zu Personen (Klienten, Angehörigen, Ärzten) vor dem Beginn einer Schicht lokal auf dem mobilen Endgerät zu speichern und somit auch ohne Internet verfügbar zu haben. Lediglich Aktualisierungen müssten so während einer Schicht nachgeladen werden. Wird das initiale Laden von Informationen über ein lokales Netzwerk durchgeführt, spart dies außerdem Datenvolumen ein, welches bei Verträgen mit Internet über das Mobilfunknetz meist sehr begrenzt ist (**Quelle notwendig?**).

Wichtig bei der Dokumentation von Wunden ist neben einer textuellen Beschreibung eine bildliche Erfassung. Dies könnte mit einer mitgeführten Digitalkamera erfolgen, jedoch besteht dann das Problem, dass die aufgenommenen Bilder nach vollendeter Schicht manuell an das zentrale System übertragen werden müssen und den richtigen Patienten zugeordnet. Dies kostet verhältnismäßig viel Zeit und kann zu Fehlern bei der Zuordnung führen. Wünschenswert ist deswegen eine Funktion in der mobilen Anwendung, die es ermöglicht, direkt über eine interne Kamera Wundfotos aufzunehmen und diese an das zentrale System zu übertragen.

Eine ähnliche Anforderung wurde bei der Erfassung der Vitalwerte wie Puls und Blutdruck identifiziert. Neben der manuellen Messung und Eintragung der ermittelten Werte gäbe es die Möglichkeit, ein automatisches Messgerät via Bluetooth an das mobile Endgerät anzuschließen und damit die Messung und Eintragung vollständig zu automatisieren. Dass dies grundsätzlich möglich ist, wurde im Projekt *Stroke Manager* des Forschungszentrum Informatik (FZI) gezeigt [13]. Die automatische Erfassung würde Fehler bei der manuellen Eintragung von Messwerten würden somit ausgeschlossen. Da die Möglichkeit besteht,

dass das automatische Messgerät ausfällt, muss eine manuelle Eintragung jedoch trotzdem möglich sein.

Das Projekt VitaBit befasste sich, ähnlich wie diese Arbeit, mit der Dokumentation von Pflegeleistungen ambulanter Dienste. Auch hier musste einer Pflegekraft ein Tages-/Tourenplan angezeigt werden. Da die Leistungen ambulanter Dienste ausschließlich vor Ort beim Klienten erbracht werden, die häufig allein leben und nur noch eingeschränkt mobil sind, hat die Pflegekraft in der Regel einen Schlüssel zu den Wohnräumen des Klienten. Die Schlüssel aller Klienten werden meist an einem Band zusammengefasst und mit Nummern zur Zuordnung versehen. Damit die Pflegekraft nicht erst lang nach dem passenden Schlüssel suchen muss, war eine Anforderung, dass die Schlüsselnummer direkt in der Übersicht der Stationen auftaucht. Da die in dieser Arbeit entwickelte Arbeit ebenfalls für den ambulanten Pflegedienst gedacht ist, wird diese Anforderung auch hier übernommen. Im Falle der stationären Betreuung von Klienten soll anstatt einer Schlüsselnummer die Zimmernummer in der Einrichtung angezeigt werden.

Die Zielgruppe, für die die mobile Anwendung entwickelt wird, sind Angehörige der Alten- und Pflegeberufe. Da es sich bei dieser Gruppe in der Regel um wenig IT-affine Personen handelt, muss die Anwendung übersichtlich strukturiert werden und intuitiv bedienbar sein, damit die Hürden der Benutzung durch die Bedienbarkeit möglichst gering sind. Ziel ist es, dass das grundsätzliche Wissen über die Bedienung von Tablet-PCs sowie das Fachwissen über die Dokumentation der Leistungen ausreicht, um die mobile Anwendung zu verwenden. Der Aufwand für Schulungen soll so minimal gehalten werden. Konzeptionell soll eine Orientierung an den graphischen Benutzerschnittstellen von iOS und Android stattfinden, den Betriebssystemen für mobile Geräte von Apple und Google.

Eine Funktionalität, die in erster Linie für ambulante Pflegedienste interessant ist, ist die Möglichkeit die Adresse eines Klienten auf einer Karte anzeigen zu lassen. Alternativ könnte Global Positioning System (GPS) könnte auch die aktuelle Position relativ zur Adresse des Klienten angezeigt werden, um eine Routenplanung durchzuführen und somit Fahrzeiten besser abschätzen zu können. Hierzu ist allerdings eine aktive Internetverbindung notwendig. Eine vollständige Navigation ist nicht geplant.

Um eine Reduzierung der elektronischen Geräte zu erreichen, die eine Pflegekraft mitführen muss, ist eine Funktion zum Führen von Telefongesprächen hilfreich. Einerseits könnten normale Telefongespräche beispielsweise mit der Zentrale, Angehörigen oder Ärzten geführt werden, andererseits wäre es aber auch möglich, einen Notruf-Knopf in die Anwendung zu integrieren, um in Notsituationen direkt Hilfe anfordern zu können. Beschränkt wird die Funktionalität zwar durch die Notwendigkeit der Verfügbarkeit eines Mobilfunknetzes, jedoch kann diese Einschränkung ebenso beim Mitführen eines separaten Mobiltelefons.

Von der bisher beschriebenen Menge an Funktionen ist lediglich die automatische Erfassung von Blutdruck und Puls mittels eines per Bluetooth angeschlossenen Messgerätes durch manuelle Erfassung substituierbar. Alle anderen Anforderungen müssen auf Basis der in der Literatur gewonnenen Erkenntnisse umgesetzt werden, um einen sinnvollen Einsatz der mobilen Anwendung überhaupt erst zu ermöglichen.

Aus den beschriebenen funktionalen Anforderungen an die mobile Anwendung ergeben sich eine Reihe von technischen Anforderungen an das Endgerät. Grundanforderung ist eine Internetverbindung sowohl über ein lokales Wireless Local Area Network (WLAN), als auch über ein Mobilfunknetz, um Tourendaten und Aktualisierungen empfangen und senden zu können. Zur Dokumentation von Wunden per Foto muss eine Kamera mit einer ausreichend hohen Auflösung vorhanden sein, damit das medizinische Personal bei der Beurteilung der Wunden nicht durch schlechte Bildqualität eingeschränkt wird. Zur Unterstützung der intuitiven Bedienung sollte das mobile Gerät einen berührungsempfindlichen Bildschirm haben, was aber bei allen Enderäten der Fall sein sollte, da dies eines der Hauptmerkmale von Tablet-PCs ist. Zur Bestimmung der Position einer Pflegekraft, um die Route zum nächsten Klienten zu berechnen, muss eine GPS-Verbindung möglich sein. Sollen Messgeräte für Blutdruck und Puls angeschlossen werden, muss eine Bluetooth-Verbindung verfügbar sein.

Da Alten- und Krankenpflege dem sozialen Bereich zuzuordnen ist und ein Ziel die Einsparung von Kosten ist, sollten die Endgeräte möglichst geringe Kosten verursachen. Obwohl grundsätzlich Tablet-PCs sowohl mit dem Android-Betriebssystem als auch mit iOS die technischen Anforderungen erfüllen, können Android Geräte im Schnitt zu einem deutlich niedrigeren Preis erworben werden [QUELLE]. Aus diesem Grund fällt die Entscheidung bei der Wahl der Plattform auf Android.

3.2. Entwurfsentscheidungen des Gesamtsystems auf Basis der erhobenen Anforderungen

Zur Dokumentation von Pflegetätigkeiten auf Tablet-PC-ähnlichen Endgeräten arbeitete Christina Hardt [12] in Ihrer Diplomarbeit mit der im Projekt VitaBIT ([8] und [9]) entwickelten Software. Diese bietet bereits Datenstrukturen für Tourenpläne und die Definition von zu erledigenden Tätigkeiten pro Station. Da die Anforderungen dieser Arbeit auch Touren- bzw. Schichtpläne benötigen, um diese auf einem Tablet-PC abrufen und anzeigen zu können, wird VitaBIT die Basis für das in Kapitel 2 beschriebene zentrale System darstellen. Da es im Kontext dieser Arbeit aber nicht mehr um die Dokumentation von Pflegetätigkeiten geht, sondern um die Übermittlung und Dokumentation ärztlicher Tätigkeiten durch Pflegepersonal, müssen Änderungen an VitaBIT vorgenommen werden. Um den Änderungen und dem Kontext der vernetzten Pflege auch namentlich gerecht zu werden, wurde das zentrale System in CareNet (Care = Pflege, Net = Netz) umbenannt. Auf die Änderungen an VitaBIT wird in dieser Arbeit nicht eingegangen, da diese von Mitarbeitern der nubedian GmbH durchgeführt werden.

Abbildung 3.1 zeigt die Komponenten des Gesamtsystems erweitert um die verwendeten Technologien, auf denen diese basieren. Die Kommunikation zwischen den Komponenten erfolgt via Internet mittels REST. REST steht für *Representational State Transfer* und stellt einen Architekturstil zur zustandslosen Interaktion dar. Im Gegensatz zu Sitzungen

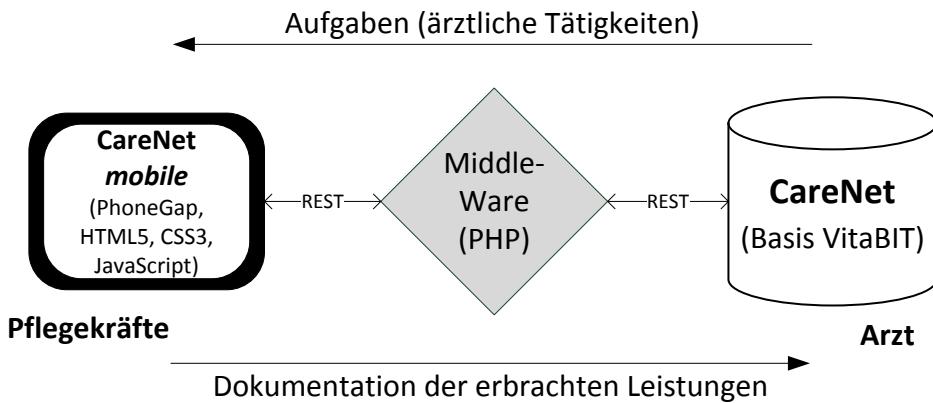


Abbildung 3.1.: Verwendete Technologien in der Gesamtarchitektur

(Sessions), die von Servern während der Interaktion mit einem Benutzer aufwendig verwaltet werden müssen, werden bei dem REST-Prinzip vorberechnete Ressourcen, beispielsweise eine Webseite, über eine URI (Uniform Resource Identifier) referenziert und geladen. Alle Informationen über den Kontext einer Anfrage sind in der Anfrage selbst enthalten und es werden keine weiteren lokal beim Nutzer oder auf dem Server gespeicherten Informationen benötigt. Eine Kommunikation via REST bedeutet eine lose Verkoppelung von vordefinierten Ressourcen, durch die ein Nutzer anhand gegebener URIs navigieren kann. Es werden lediglich die grundlegenden HTTP-Methoden PUT, POST, GET und DELETE verwendet, um Anfragen zu formulieren. Übertragen werden Ressourcen bzw. Datenobjekte in der Regel in Form einer XML- oder JSON-Datei [14, S. 19].

Die in dieser Arbeit zu entwickelnde mobile Anwendung stellt im Wesentlichen eine mobile Erweiterung von CareNet dar, aufgrund dessen der Name CareNet *mobile* gewählt wurde. Die Verwendung von Tablet-PCs als Hardware ist vorgegeben, da sie den neuesten Stand der Technik repräsentieren. Es wurde eine Neuentwicklung einer Lösung zur mobilen Dokumentation angestoßen, da in den vorangegangenen Projekten Probleme mit der verwendeten Hardware identifiziert werden konnten. Im Projekt VitaBIT wurden zur Dokumentation Smartphones der ersten Generation eingesetzt und in [12] wurde ein Panasonic CF-H1 eingesetzt, ein Vorläufer heutiger Tablet-PCs. Dieses Gerät hatte zwar bereits einen berührungssempfindlichen Bildschirm und eine Kamera, jedoch war die Leistungsfähigkeit nicht ausreichend, was zu Performanzproblemen führte. Außerdem konnte die Intuitivität der Bedienung nicht so umgesetzt werden, wie es beispielsweise unter Android oder iOS heute möglich ist. Das in VitaBIT verwendete Smartphone ist zwar sehr handlich und damit leicht mitzuführen, jedoch schränkt der kleine Bildschirm die Möglichkeiten der Darstellung ein, was gegen die Anforderung der Übersichtlichkeit spricht.

Jedes Betriebssystem von Tablet-PCs basiert auf einer eigenen Programmiersprache. Android basiert beispielsweise auf Java, iOS auf Objective-C und Windows Phone auf C#. Anwendungen für die Betriebssysteme können nur in den jeweiligen Sprachen entwickelt werden. Eine so entwickelte Anwendung wird als *nativ* bezeichnet. Dieser Umstand impliziert, dass bei der Entwicklung neuer Anwendungen vorher die Entscheidung getroffen werden muss, auf welcher Plattform diese bereitgestellt werden soll. Sollen mehrere Plattformen bedient werden, müssen mehrere Anwendungen mit der gleichen Funktionalität, aber anderer Quellcodebasis erstellt werden. Dies führt zu hohem zeitlichen Aufwand und ist kostenintensiv.

Eine Alternative ist die Erstellung von *WebApps*. Prinzipiell handelt es sich hierbei um Webseiten, jedoch sind diese für Touch-Oberflächen optimiert und sie sollen optisch an native Anwendungen erinnern. Da WebApps aber in einem Browser auf dem Tablet-PCs aufgerufen werden müssen, können sie nicht das Gefühl einer vollwertigen App (Application = Anwendung) bieten. Ungleich größer sind die Einschränkungen von WebApps bei der bereitstellbaren Funktionalität. Mit reinen Webtechnologien wie HTML5 oder JavaScript ist es nämlich nicht möglich, hardspezifische Funktionen wie die Kamera, den GPS-Sensor oder eine Bluetooth-Verbindung anzusprechen. Dies ist nur über die jeweilige native Programmiersprache möglich. Da in den Anforderungen aber eine Nutzung der Kamera sowie des GPS-Sensors gefordert wird, ist eine Implementierung von CareNet *mobile* als WebApp nicht möglich.

Ein Ansatz zur Lösung dieses Problems bietet das Projekt PhoneGap [15]. Die Idee ist, in dem Gerüst einer nativen Anwendung, einem nativen Container, einen Browser zu platzieren, der Implementierungen in Webtechnologien wie HTML5, CSS3 und JavaScript interpretieren kann. Abbildung 3.2 zeigt den Zusammenhang der verschiedenen Technologien. Um den Zugriff auf endgerätspezifische Funktionen zu ermöglichen, bietet PhoneGap zur Kommunikation mit allen wichtigen Hardware-Funktionen JavaScript-Methoden an, die auf die jeweiligen Befehle in der nativen Programmiersprache abgebildet werden. Anstatt beispielsweise die Kamera unter Android direkt über Java anzusprechen, erfolgt der Aufruf in JavaScript, welcher von einer Java-Klasse verarbeitet und an die Hardware weitergereicht wird. PhoneGap verfolgt somit einen hybriden Ansatz aus nativer Entwicklung und der Nutzung von WebApps.

Die reine Nutzung von Webtechnologien und die Bereitstellung verschiedener nativer Container durch PhoneGap für die jeweils gewünschte Plattform ermöglicht eine Portabilität der Anwendung. Einmal entwickelt kann diese beispielsweise sowohl auf Android, als auch auf PhoneGap oder Windows Phone bereitgestellt werden. Diese Möglichkeit bringt Vorteile in Bezug auf Kosten und Zeit gegenüber der Entwicklung individueller nativer Anwendungen.

Als Anforderung an CareNet *mobile* wurde zwar zunächst nur die Bereitstellung auf Android-Endgeräten gefordert, jedoch ist nicht auszuschließen, dass in Zukunft auch die Nutzbarkeit auf iOS- oder Windows-Geräten gefordert wird. Da es mit der Verwendung von PhoneGap möglich ist, diese Option mit geringem Aufwand offen zu halten, wurde entschieden, CareNet *mobile* in als hybride Anwendung zu entwickeln. [Same Origin Policy reinnehmen?]

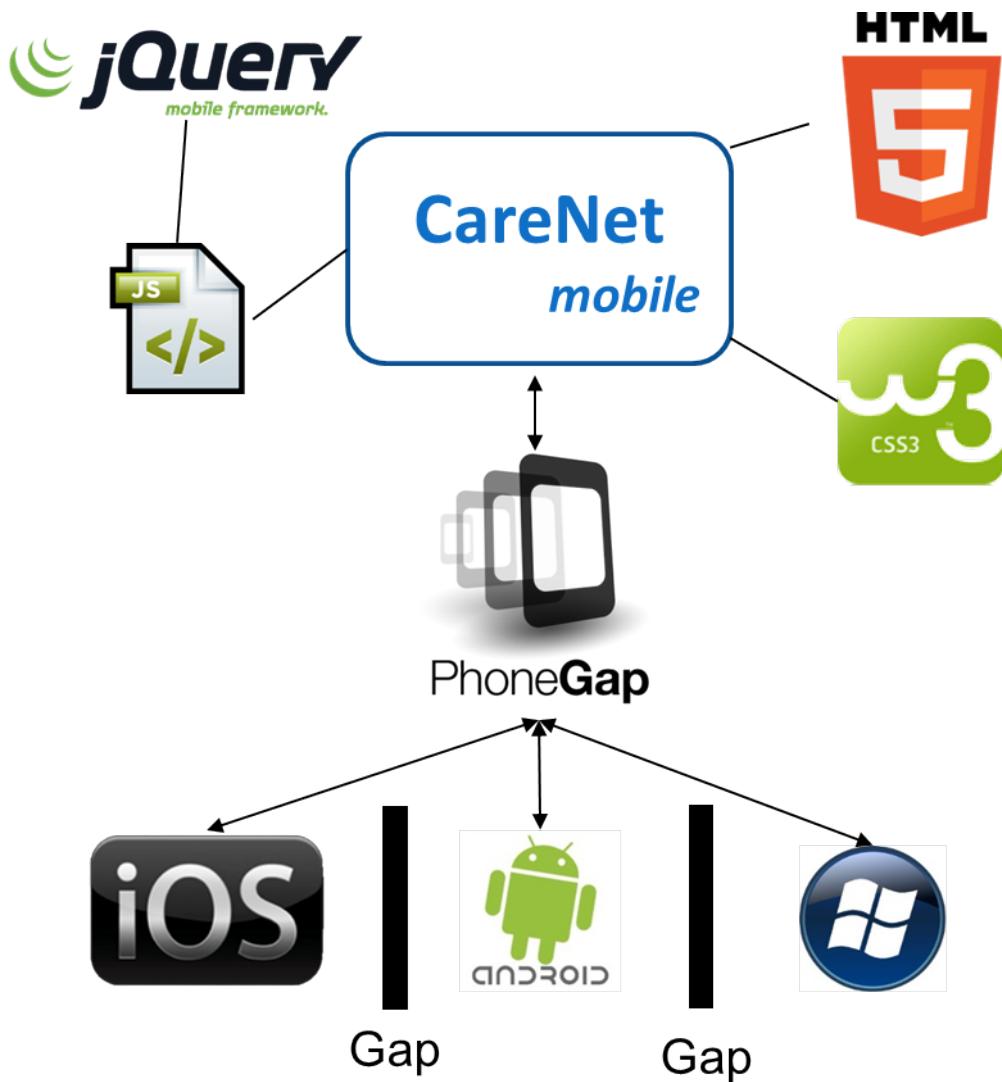


Abbildung 3.2.: Hybride Anwendungsentwicklung mit PhoneGap

Zusammen mit PhoneGap bietet sich die Verwendung bereits existierender JavaScript-Bibliotheken an, die zum Entwurf touch-optimierter Benutzeroberflächen entwickelt wurden. Zwei verbreite Bibliotheken sind jQuery mobile .[16] und Sencha Touch [17]. Im Kontext dieser Arbeit wurde jQuery mobile aufgrund der Einfachheit in der Entwicklung und der Mächtigkeit der gebotenen Funktionalität gewählt.

In [12] hat sich gezeigt, dass sich einmal gestellte Anforderungen rasch ändern können und dass eine Anwendung so entworfen sein sollte, dass Anpassungen mit vertretbarem Aufwand durchgeführt werden können. Zwar ist momentan die Menge an zu dokumentierenden Tätigkeiten durch eine abgeschlossene Liste in Richtlinie § 63 Abs. 3c SGB V beschränkt, jedoch sind Änderungen in Zukunft möglich. Neben zusätzlichen Tätigkeiten könnten auch andere funktionale Erweiterungen hinzukommen. Um diesen Anforderungen gerecht zu werden, wird die Anwendung auf Basis einer Plug-In-Architektur entwickelt, angelehnt an die bewährte Architektur der Entwicklungsumgebung Eclipse [18]. Bei einer Plug-In-Architektur kann gekapselte Funktionalität nach dem Steckerprinzip ("“plug-In”")

= einstecken) hinzugefügt und wieder entfernt werden. Im Idealfall sollten Änderungen an den Plug-Ins keine Auswirkungen auf das Gesamtsystem haben. Plug-Ins untereinander können jedoch Abhängigkeiten aufweisen. Wie die Plug-In-Architektur von CareNet *mobile* genau aussehen wird, wird in Kapitel 4.1 beschrieben.

Wie bereits in Kapitel 2 erwähnt, hat die Middleware in erster Linie eine Platzhalterfunktion, um den Zugang zu CareNet später um weitere Schnittstellen erweitern können. Da im Moment nur eine REST-Schnittstelle angeboten wird, die auch von CareNet *mobile* verwendet wird, wäre die Middleware zur Kommunikation nicht notwendig. Sie nimmt lediglich REST-Aufrufe an und leitet diese unbearbeitet weiter. Es wurde entschieden, die Middleware in PHP [19] zu implementieren, da es sich hierbei um eine mächtige und weit verbreite serverseitige Programmiersprache handelt. PHP wird von einem Großteil der Server-Hosting-Anbieter unterstützt und stellt somit keine Limitierung bei der Auswahl des Ortes dar, an dem die Middleware installiert wird.

4. Basisanwendung und Plug-Ins zur Realisierung der identifizierten Anforderungen

Im vorliegenden Kapitel wird die Implementierung der mobilen Anwendung CareNet *mobile* beschrieben. Da es sich um eine Plug-In Architektur handelt, die sich in eine Basisanwendung und eine Menge von Plug-Ins gliedert, wurde die Aufteilung der Kapitel entsprechend den zu implementierenden Elementen angepasst. Abschnitt 4.1 beschreibt die Struktur der Basisanwendung und Abschnitt 4.2 erklärt Vorgang und Voraussetzungen, um Plug-Ins zur Anwendung hinzuzufügen. Die Abschnitte 4.3 und 4.4 stellen die beiden zentralen Plug-Ins *Kontakte* und *Touren* vor.

4.1. Aufgaben und Struktur der Basisanwendung

Um einem Benutzer von CareNet *mobile* die Bedienung der Anwendung zu erleichtern, sollte ein einheitliches Gesamterscheinungsbild eingehalten werden, da ständige Strukturwechsel in der grafischen Oberfläche zu Unübersichtlichkeit führen können [Quelle]. Hierzu wurde die in Abbildung 4.1 gezeigte feste Struktur der grafischen Oberfläche entworfen. Oben links befindet sich das Logo der Anwendung, darunter ist ein Bereich für die Darstellung von Informationen wie das aktuell verwendete Plug-In oder der gerade behandelte Klient reserviert. Am oberen Rand, rechts des Logos, wurde eine Leiste zum Platzieren von Buttons zur Navigation eingefügt. Hier sollen Schaltflächen zur Navigation zwischen Basisanwendung und Plug-Ins platziert werden (z.B. ein "Home"-Button) und auch zur Navigation zwischen verschiedenen Sichten innerhalb eines Plug-Ins. Der größte Bereich ist zur Darstellung und Bearbeitung des eigentlichen Inhalts gedacht.

Die gesamte Anwendung basiert lediglich auf zwei HTML-Dateien und einer Menge von JavaScript-Dateien. Eine HTML-Datei wird für einen Login-Dialog verwendet, die zweite um den kompletten Inhalt darzustellen. Alle Strukturelemente und Funktionalität wird über JavaScript hinzugefügt. Zur Manipulation der HTML-Elemente wird jQuery und jQuery mobile verwendet.

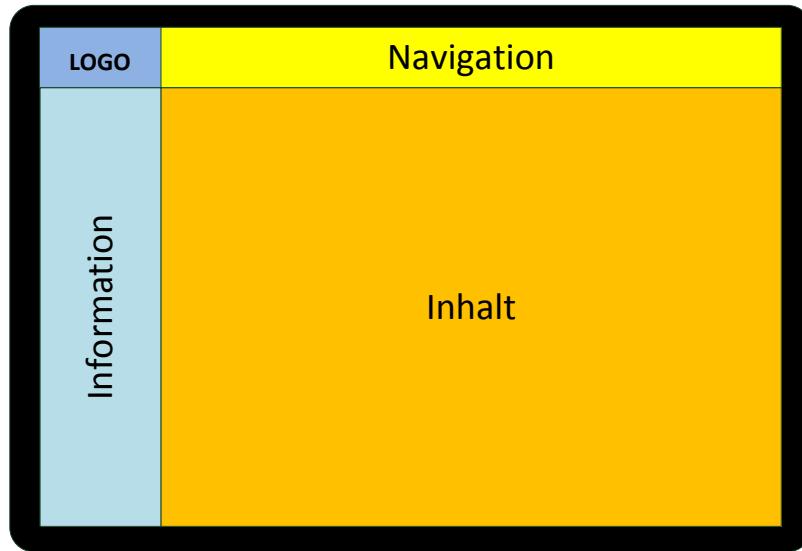


Abbildung 4.1.: Struktur der grafischen Oberfläche von CareNetmobile

Die Strukturierung und Gestaltung der grafischen Oberfläche erfolgt mittels HTML5 und CSS3, die Funktionalität wird mit JavaScript entwickelt. Gegenüber der Entwicklung einer nativen Anwendung für Android ergeben sich bei der Entwicklung mit JavaScript gewisse Schwierigkeiten. Während Java eine objektorientierte Sprache ist, bei der Typsicherheit und Kapselung von Funktionalität in Klassen zu den Eigenschaften gehören, ist JavaScript eine Scriptsprache, die keinerlei Typsicherheit und nur bedingte Kapselungsmöglichkeiten bietet. Ein wichtiger Unterschied ist, dass Java vor der Ausführung zunächst kompiliert, d.h. in eine maschinenverständliche Zwischensprache übersetzt werden muss, während JavaScript direkt vom Browser interpretiert werden kann. Dies impliziert, dass JavaScript-Code, selbst wenn er bereits auf einer Webseite ausgeführt wird, dem Nutzer im Klartext zur Verfügung steht, während der Java-Code nur bedingt und durch aufwendige Verfahren wieder dekompiliert und somit im Klartext verfügbar gemacht werden kann. Dieser Eigenschaft muss sich ein Entwickler besonders im Bezug auf im Quellcode vermerkte sicherheitsrelevante Informationen bewusst sein.

Während der Entwicklung kann vor allem die fehlende Möglichkeit der Kapselung problematisch werden. Abbildung 4.2 zeigt Beispielimplementierungen in Java und JavaScript. Links wurde eine Klasse erstellt, die Methoden und Variablen kapselt. Wird eine Variable außerhalb einer Methode definiert (Abbildung 4.2: *ersteVariableJava*), ist diese in der gesamten Klasse verfügbar. Ist die Variable als *public* gekennzeichnet, kann auf sie über *Klassenname.variablename* zugegriffen werden. Wird sie als *private* deklariert, ist sie nur innerhalb der Klasse verfügbar. Das gleiche gilt für Methoden. Wird eine Variable innerhalb einer Methode definiert (Abbildung 4.2: *zweiteVariable*), kann auch nur innerhalb dieser Methode auf die Variable zugegriffen werden.

In JavaScript sind grundsätzlich alle Methoden global verfügbar. Die Verfügbarkeit beschränkt sich hier nicht auf eine Klasse oder Datei, sondern auf eine Methode kann von

Java	JavaScript
<p>Datei: beispiel.java</p> <pre>class beispiel{ public string ersteVariableJava = ""; //Zugriff: beispiel.ersteVariableJava public void methodeJava(){ //Zugriff: beispiel.methodeJava(); string zweiteVariableJava = ""; //nur in Methode verfügbar } }</pre>	<p>Datei: beispiel.js</p> <pre>//Konstruktor und „Klasse“ function beispiel(){ ... var ersteVariableJS = ""; //global beispiel.methodenJS = function(){ //Zugriff: beispiel.methodenJS zweiteVariableJS = ""; //global var dritteVariableJS = ""; //nur in Methode verfügbar }</pre>

Abbildung 4.2.: Vergleich zweier Beispielimplementierungen von Java und JavaScript

allen im Kontext einer HTML-Datei geladenen Dateien zugegriffen werden. Eine Klassendefinition kann nur in sofern erfolgen, dass einem Methodennamen der Name einer anderen vorher definierten Methode vorangestellt wird (Abbildung 4.2: *beispiel.methodeJS()*). Hiermit wird es möglich, einen Namensraum zu definieren, um Methoden einem bestimmten Kontext (hier: *beispiel*) zuzuordnen. Wird allerdings an anderer Stelle im geladenen JavaScript-Code die selbe Kontext-Methodennamen-Kombination verwendet, kann der Interpreter, der Browser, keine eindeutige Zuordnung des Namens zu ausführbarem Quellcode mehr herstellen und es wird nichts ausgeführt.

Ebenso verhält es sich mit Variablen. Wird eine Variable mit dem Zusatz *var* außerhalb einer Methode definiert, ist sie global verfügbar. Das gleiche gilt, wenn eine Variable ganz ohne Zusatz (Abbildung 4.2: *zweiteVariableJS*) definiert wird. Nur wenn eine Variable innerhalb einer Methode mit dem Zusatz *var* erstellt wird, kann sie nur innerhalb der Methode verwendet werden. Dieser Umstand kann zu massiven Problemen führen. Vergisst ein Entwickler beispielsweise nur ein einziges Mal den Zusatz *var* bei der Definition einer Laufvariable in einer Schleife (häufig mit dem Namen *i*), wird diese nun global geltende Variable alle anderen Schleifen, die ebenfalls die Laufvariable *i* verwenden, beeinflussen. Da dies kein syntaktischer, sondern ein semantischer Fehler ist, kann dieser nicht von automatisierten Fehlerfindungswerkzeugen entdeckt werden und es beginnt die sprichwörtliche Suche nach der Nadel im Heuhaufen. Derartige Umstände können sehr viel Zeit kosten.

Einzelne Plug-Ins sollen die in Abbildung 4.1 gezeigte Struktur nutzen, um Inhalte darzustellen. Damit dies möglich ist, müssen die einzelnen Bereiche, welche als HTML-DIV-Elemente erstellt wurden, eindeutig durch IDs gekennzeichnet werden. Da es in JavaScript nicht, wie in Java, möglich ist, Konstanten zu definieren, wurden in einer separa-

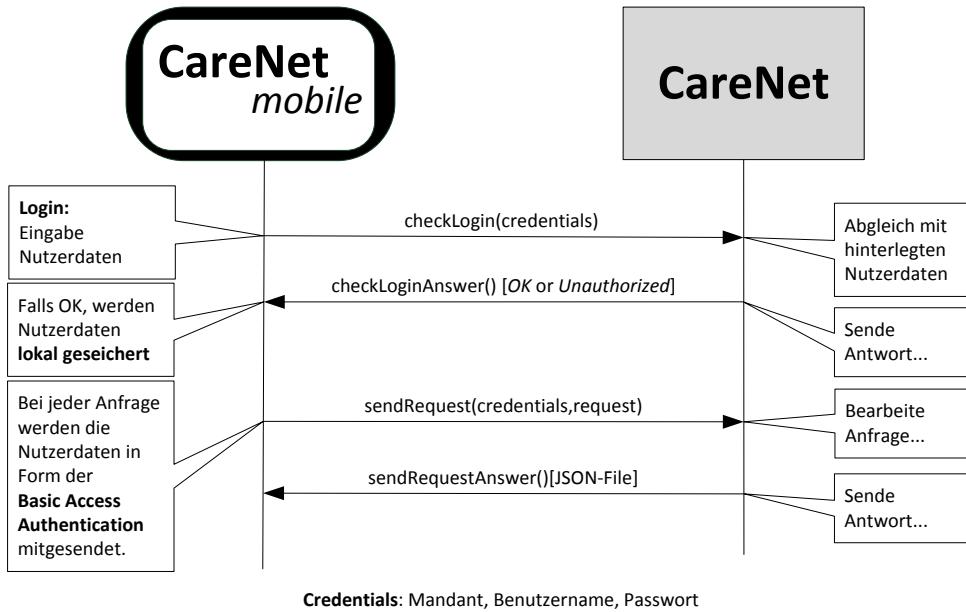


Abbildung 4.3.: Überprüfung der Benutzerdaten in CareNetmobile

ten JavaScript-Datei `CONSTANTS.js` alle IDs in Form von Variablen hinterlegt, welche den DIV-Elementen zugewiesen wurden. Der Nachteil von Variablen ist, dass diese während der Laufzeit aus den Plug-Ins heraus verändert werden können. Aus diesem Grund muss die Konvention eingehalten werden, zentral definierte Variablen nicht zu verändern, da dies Auswirkungen auf andere Plug-Ins haben könnte. Die Eigenschaft einer Plug-In-Architektur, dass Plug-Ins keinen Einfluss auf die Basisanwendung haben, kann somit nicht vollständig in JavaScript umgesetzt werden. Menschliche Fehler durch Nicht-Einhaltung der Konvention bleiben möglich.

Der Vorteil von zentral definierten Variablen ist, dass an einer zentralen, übersichtlichen Stelle nachgeschaut werden kann, wie eine ID lautet, und dass die wichtigen Strukturelemente für Informationen, Navigation und Inhalt zentral referenziert werden können. Die Datei `CONSTANTS.js` wird außerdem dazu genutzt, um alle sonstigen in der Implementierung der Basisanwendung verwendeten Zeichenketten als Variablen zu hinterlegen. Hierzu zählt beispielsweise die Server-Adresse der Middleware oder alle Arten von Meldungen an einen Nutzer. Dieses Verfahren erhöht die Wartbarkeit der Anwendung, da Änderungen an mehrfach verwendeten Zeichenketten nicht *mehr*fach umgesetzt werden müssen, sondern lediglich *einmal* in der zentralen Datei. Die Wahrscheinlichkeit für Fehler durch nicht vollständig umgesetzte Anpassungen kann so verringert werden. Dieses Vorgehen ist angelehnt an die zentrale Verwaltung von *String Ressources* bei der Entwicklung nativer Android-Anwendungen [20].

Eine Grundfunktion, welche die Basisanwendung zur Verfügung stellt, ist ein Log-In zur Sicherung der persönlichen Daten und einer Identifizierung des Nutzers. Da CareNetmobile

eine mobile Erweiterung von CareNet darstellt, werden keine gesonderten Konten für die Nutzung der mobilen Anwendung angelegt, sondern es werden die schon existierenden Konten in CareNet verwendet. Folglich müssen die bei der Anmeldung eines Nutzer eingegebenen Daten an CareNet gesendet werden, wo sie überprüft werden und das Ergebnis an die mobile Anwendung zurückgesandt wird. Benötigt werden beim Login der Mandant, der Benutzername und das Passwort.

In Kapitel 3.2 wurde die Kommunikation von CareNet_{mobile} mit CareNet via REST erklärt. Aus dieser Kommunikation resultiert eine Besonderheit bei der Authentifizierung eines Nutzers mit einem zentralen System. Abbildung 4.3 zeigt die notwendigen Schritte zur Authentifizierung eines Nutzers und dem anschließenden Senden einer Anfrage. Da keine Sitzung aufgebaut werden kann, in der ein Nutzer dauerhaft angemeldet wird, müssen die kompletten Nutzerdaten bei jeder Anfrage mitgesendet werden.

Meldet sich ein Nutzer in CareNet_{mobile} an, so werden zunächst nur die eingegebenen Daten an CareNet gesendet. Als Antwort kommt entweder der HTML-Code 200 (*OK*) oder 401 (*Unauthorized*). Sind die Daten korrekt (Code = 200), werden sie im lokalen Speicher der Anwendung abgelegt, um sie für spätere Anfragen an CareNet nicht noch einmal eingeben zu müssen. Hierzu kann der von PhoneGap angebotene lokale Key-Value-Speicher genutzt werden. In diesem Speicher können Zeichenketten, referenziert über einen Schlüssel, permanent abgelegt werden und auch nach dem Schließen der hybriden mobilen Anwendung wieder abgerufen werden. Die Nutzerdaten werden hier allerdings unverschlüsselt abgelegt. Aus diesem Grund werden diese aus dem lokalen Speicher gelöscht, sobald sich der Nutzer wieder von CareNet_{mobile} abmeldet. Weitere Anfragen an CareNet sind so nicht mehr möglich.

Der Login von CareNet_{mobile} ist als eigenständige HTML-Datei in Form eines Dialogs realisiert. Wird die Anwendung gestartet, wird zwar zunächst die zentrale Datei *index.html* geladen, jedoch wird vor der Anzeige der Startseite überprüft, ob Nutzerdaten im lokalen Speicher vorhanden sind oder nicht. Sind keine Daten vorhanden, wird zunächst der Login-Dialog aufgerufen und nur wenn die hier eingegebenen Nutzerdaten korrekt sind, erfolgt eine Anzeige der Startseite. Eine reine Überprüfung der Verfügbarkeit von Benutzerdaten im lokalen Speicher reicht als Kriterium zur Anzeige von Inhalten der Startseite aus, da aufgrund der vorherigen Überprüfung ausschließlich validierte Benutzerdaten im Speicher abgelegt werden.

Neben einer Behandlung des Zustandsübergangs beim Starten der Anwendung müssen weitere Zustände im Lebenszyklus einer Android-Anwendung berücksichtigt werden. Wurde eine Anwendung einmal gestartet, kann sie nicht direkt vollständig durch einen Befehl oder die Betätigung des Home-Buttons geschlossen werden, sondern sie wird lediglich *pausiert*. Wird die Anwendung aus dem Task-Manager heraus wieder geöffnet, wird diese nicht neu gestartet, sondern an dem Punkt fortgesetzt, an der sie geschlossen wurde. Dies kann zu einem Sicherheitsproblem werden, wenn ein Nutzer die Anwendung schließt und sich nicht darüber bewusst ist, dass diese eigentlich weiterhin verfügbar ist. Bekommt nun ein Dritter das Tablet in die Hände, kann dieser im Namen der angemeldeten Person Veränderungen vornehmen oder auf sensible Daten zugreifen. Um dies zu verhindern, werden die Nutzerdaten im lokalen Speicher gelöscht, sobald die Anwendung geschlossen wird.

Wird diese wieder geöffnet, muss der Nutzer sich neu authentifizieren. Dieses Vorgehen führt zu Einbußen bei der Benutzbarkeit der Anwendung, da immer wieder neu Nutzernname und Passwort eingegeben werden muss, auch wenn nur aus Versehen der Home-Button betätigt wurde, jedoch ist dies für den Schutz der sensiblen Daten unabdinglich. Zur Reduzierung der Wahrscheinlichkeit, dass CareNet *mobile* unabsichtlich geschlossen wird, wurde der Zurück-Button deaktiviert, da dieser bei nur zwei verwendeten HTML-Seiten zu einem Verlassen der Anwendung führt.

In Abbildung 4.3 wird die Rolle der Middleware vernachlässigt, da diese nicht für das grundsätzliche Kommunikationsprinzip relevant ist. Tatsächlich übernimmt die Middleware aber einen Arbeitsschritt bei der Kommunikation. Die Nutzerdaten werden als Zeichenkette *Mandant/Benutzername:Passwort* an die Middleware übertragen. CareNet benötigt die Daten zur Authentifizierung jedoch in Form einer nach der Base Access Authentication verschlüsselten Zeichenkette. Base Access Authentication ist [[Erläuterung Base64 nach Network security: private communication in a public world, second edition](#)]. Da PHP standardisierte Methoden zur Erstellung dieser Verschlüsselung anbietet, wurde entschieden, die Verschlüsselung in die Middleware zu verlagern.

Auch wenn es sich bei der Base Access Authentication namentlich um eine Verschlüsselung handelt, ist diese jedoch kein Schutz gegen einen Zugriff auf die Daten Dritter, da das Verfahren öffentlich bekannt ist und leicht entschlüsselt werden kann. Im Prinzip werden die Benutzerdaten also im Klartext von CareNet *mobile* zu CareNet übertragen. Hört ein Dritter die Kommunikation ab, kann er die Nutzerdaten auslesen und zweckentfremden. Aus diesem Grund muss die Verbindung an sich verschlüsselt werden. Hierzu wird eine HTTPS-Verbindung sowohl von CareNet *mobile* zur Middleware als auch von der Middleware zu CareNet verwendet.

HTTPS steht für *Hypertext Transfer Protocol Secure* und stellt eine Kombination aus einem symmetrischen und einem asymmetrischen Verschlüsselungsverfahren dar, um Internetverbindungen abhörsicher zu machen. Bei einem symmetrischen Verfahren benutzen Sender und Empfänger beide den gleichen Schlüssel, um Nachrichten zu ver- und entschlüsseln. Bei einem asymmetrischen Verfahren wird eine Nachricht mit einem öffentlichen Schlüssel verschlüsselt und mit einem privaten Schlüssel, den nur der Empfänger kennt, wieder entschlüsselt. Die symmetrische Verschlüsselung ist zwar deutlich schneller, jedoch auch unsicherer, da der Schlüssel zunächst über eine unverschlüsselte Verbindung übertragen werden muss und so eventuell abgehört werden könnte. Beim HTTPS-Verfahren wird zum Aufbau einer sicheren Verbindung der öffentliche Schlüssel an den Empfänger übertragen. Dieser nutzt den Schlüssel, um eine Nachricht mit einem symmetrischen Schlüssel an den Server zu schicken. Diesen symmetrischen Schlüssel können in der folgenden Kommunikation beide Seiten verwenden. Eine langsame asymmetrische Kommunikation findet also nur beim Aufbau der Verbindung statt [21, S. 152].

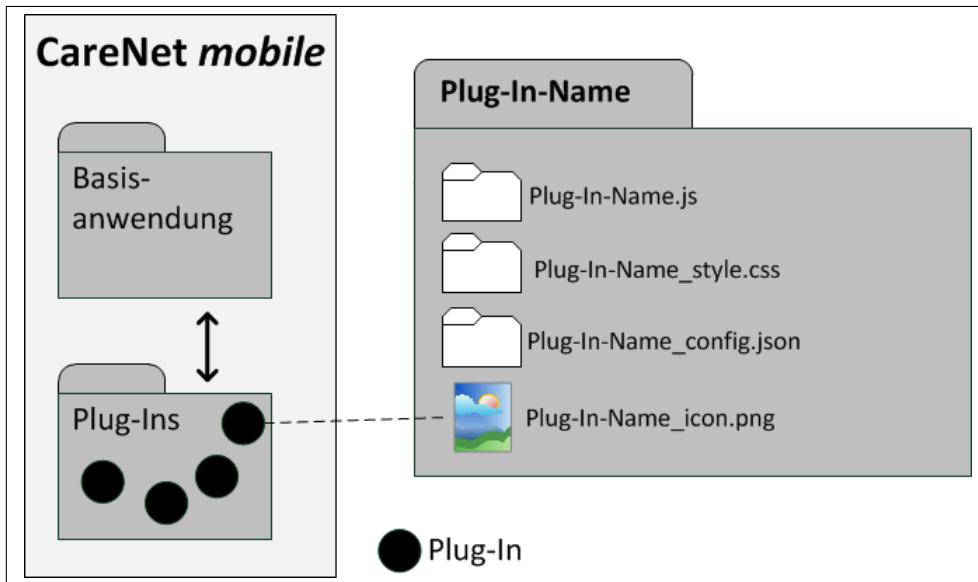


Abbildung 4.4.: Struktur eines Plug-Ins in CareNetmobile

4.2. Einbinden von Plug-Ins in die Basisanwendung

Wie in Kapitel 3.2 beschrieben, wird CareNetmobile auf Basis einer Plug-In-Architektur entwickelt. Die eigentliche Funktionalität, um die Anwendung produktiv einsetzen zu können, soll ausschließlich über Plug-Ins hinzugefügt werden, welche gekapselte Programmstücke darstellen. Plug-Ins sollen der Basisanwendung leicht hinzugefügt und wieder entfernt werden können. Außerdem soll durch die programmatische Entkopplung logisch getrennter Programmstücke eine höhere Code-Qualität erreicht werden, da so Teile des Programms verändert werden können, ohne die Gesamte Anwendung zu beeinflussen.

Die Idee des Plug-In-Musters ([22, S. 499 ff.]) ist, einer Anwendung Implementierungen zur Laufzeit durch Konfiguration hinzuzufügen zu können. Ein erneutes Kompilieren bzw. ein neuer Build-Prozess soll nicht notwendig sein. Da CareNetmobile nicht mit einer objektorientierten Sprache wie Java oder C# entwickelt wird, sondern mit JavaScript, kann diese Eigenschaft nicht vollständig umgesetzt werden. Der Quellcode eines Plug-Ins muss in CareNetmobile gekapselt in einen Ordner zum Zeitpunkt der Entwicklung eingebunden werden, bevor die Anwendung zur Installation auf dem Tablet-PC generiert wird. Damit ein Plug-In aber tatsächlich in der Anwendung bereitgestellt wird, muss es namentlich in einer zentralen Konfigurationsdatei *pluginConfig.json* aufgeführt werden. Beim Start der App werden alle hier aufgeführten Plug-Ins geladen, falls der benötigte Quellcode tatsächlich vorhanden ist. Die Konfigurationsdatei dient somit als Startpunkt zum Laden der eigentlichen Funktionalität. Zwar muss bei dem hier skizzierten Vorgehen der Quellcode der Plug-Ins noch zur Entwicklungszeit eingebunden werden, jedoch ermöglicht die Konfigurationsdatei ein Einbinden und Ausschließen von Plug-Ins zur Laufzeit, da die Konfigurationsdatei im JSON-Format durch JavaScript veränderbar ist. Die Anforderungen eines *Plug-In*-Musters werden so zumindest in Teilen umgesetzt.

Um eine unbegrenzte Anzahl Plug-Ins über ein generisches Verfahren einbinden zu können, muss eine einheitliche Struktur gegeben sein. Abbildung 4.4 zeigt die Struktur eines Plug-Ins. Es besteht aus einer Menge von Dateien, die in einem Ordner zusammengefasst

werden. Der Ordner muss den Namen des Plug-Ins tragen und die darin enthaltenen Dateien müssen der Namenskonventionen folgen, dass sie jeweils den Namen des Plug-Ins enthalten und ein beschreibendes Stichwort für den Inhalt der Datei. Zwar könnte auch schon die Dateiendung als Indikator für den Inhalt dienen, jedoch wird es durch den beschreibenden Zusatz besser lesbar.

Jedes Plug-In muss die in Abbildung 4.1 gezeigte Struktur der Anwendung nutzen und die einzelnen Div-Elemente über die in Variablen abgelegten IDs ansteuern. Um nicht bei jeder Veränderung eine neue HTML-Datei laden zu müssen, wird der Inhalt der Div-Elemente dynamisch angepasst. Hierzu ist JavaScript notwendig, folglich muss ein Plug-In eine JavaScript-Datei enthalten (*Plug-In-Name.js*), die sowohl die Methoden zum Aufbau der grafischen Oberfläche als auch die Anwendungslogik enthält. Wichtig für die grafische Darstellung von HTML-Elementen sind Cascading Style Sheets (CSS). Sie werden dazu verwendet, wiederverwendbare Formatierungen zu hinterlegen, die den HTML-Elementen zugewiesen werden können. Diese müssen, wie JavaScript auch, in einer separaten Datei zusammengefasst werden (*Plug-In-Name_style.css*).

Die Basisanwendung soll keinerlei Abhängigkeiten zu den Plug-Ins haben, jedoch verwenden Plug-Ins von der Basisanwendung bereitgestellte Methoden. Außerdem können Abhängigkeiten der Plug-Ins untereinander bestehen. Sind notwendige Methoden nicht verfügbar oder wurden diese verändert (beispielsweise die Methodensignatur), kann dies dazu führen, dass ein Plug-In nicht mehr funktioniert. Um die fehlenden Abhängigkeiten nicht erst zu entdecken, während der Nutzer eine nicht verfügbare Funktion versucht zu verwenden, müssen Abhängigkeiten in einer Konfigurationsdatei beschrieben werden. Die Datei *Plug-In-Name_config.json* enthält eine Liste aller Plug-Ins sowie deren minimal und maximal mögliche Versionsnummer, die ein Plug-In benötigt, um grundsätzlich funktionieren zu können. Außerdem enthält die Datei die Versionsnummer des Plug-Ins selbst. Vor dem Laden eines Plug-Ins wird überprüft, ob die Plug-Ins, zu denen eine Abhängigkeit besteht, grundsätzlich vorhanden sind, und ob die Version des Plug-Ins in dem angegebenen Bereich liegt (Bsp: Min-Version: 1.0, Max-Version: 1.3, Plug-In-Version: 1.1 ⇒ OK). Ist dies nicht der Fall, wird ein Plug-In nicht geladen und der Nutzer bekommt einen Hinweis angezeigt. Neben den Abhängigkeiten zu anderen Plug-Ins können außerdem Abhängigkeiten zu JavaScript-Bibliotheken wie jQuery mobile oder PhoneGap bestehen. Um hier Redundanzen zu vermeiden, die entstehen, wenn jedes Plug-In diese Bibliotheken selbst einbindet, können auch diese in der Konfigurationsdatei angegeben werden. So wird der benötigte Speicherplatz der Anwendung möglichst gering gehalten.

Die Basisanwendung bietet außer dem Login ausschließlich Funktionalitäten an, die der Nutzer nicht direkt verwenden kann, sondern die nur programmatisch von Plug-Ins genutzt werden können. Da die gesamte produktiv nutzbare Funktionalität folglich über die Plug-Ins hinzugefügt wird, dient die Startseite von CareNet *mobile* als eine Art Sprungbrett, um die hinterlegten Anwendungen zu starten. [23, S. 3 ff.] beschreibt das sogenannte *Springboard*-Muster (Sprungbrett-Muster) als häufig verwendetes und etabliertes Entwurfsmuster für die primäre Navigation in mobilen Anwendungen. Meist wird ein Gittermuster verwendet, in welchem zu den jeweiligen Anwendungen passende Abbildungen und ein textueller Bezeichner angeordnet werden. Bei dem Entwurf der grafischen Oberflä-

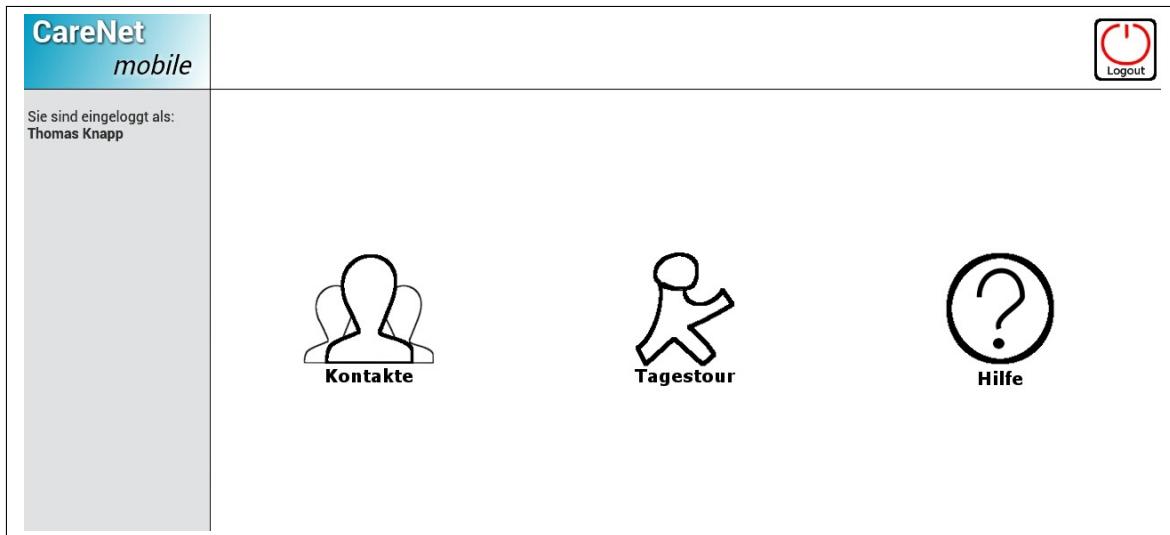
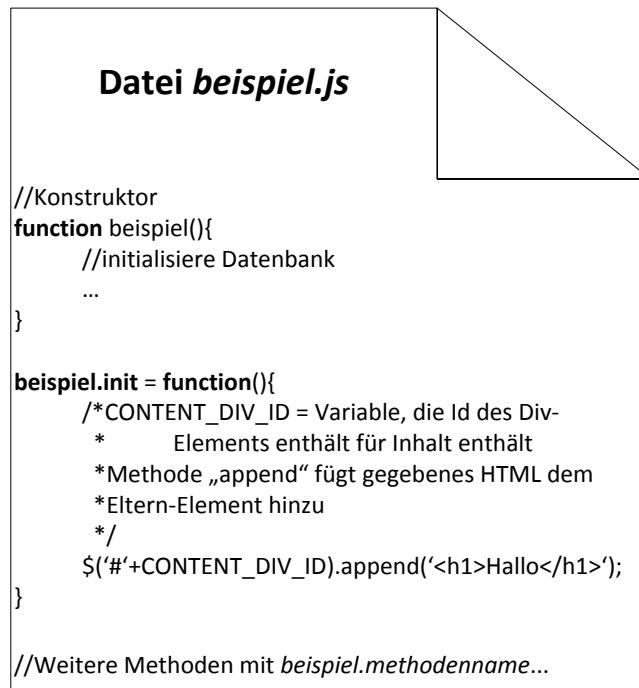


Abbildung 4.5.: Übersicht der verwendbaren der Plug-Ins

che von CareNet_{mobile} wurde dieses Muster gewählt, da es eine hohe Übersichtlichkeit auf der Startseite bietet. Abbildung 4.5 zeigt die Umsetzung in CareNet_{mobile} für fünf eingebundene Plug-Ins. Die Abstände zwischen den Symbolen wurden bewusst groß gewählt, um eine Fehlauswahl durch versehentliches Tippen auf falsche Symbole zu vermeiden. Alle Symbole werden in gleicher Größe dargestellt, um dem Nutzer zu vermitteln, dass es sich um gleichwertig gewichtete Anwendungen handelt. Es werden jeweils drei Symbole pro Reihe angezeigt. Bei mehr als drei Plug-Ins werden zusätzliche Reihen nach unten hinzugefügt. Müssen mehr Plug-Ins angezeigt werden, als Platz auf dem Bildschirm zu Verfügung steht, kann der Bereich für den Inhalt nach unten gescrollt werden. Jedes Plug-In muss ein eigenes Icon mit dem Namen *Plug-In-Name_icon.png* bereitstellen.

Zur Zeit der Abgabe dieser Arbeit war es leider noch nicht möglich, JavaScript- und CSS-Dateien zur Laufzeit einzubinden. Es muss jede Datei im Plug-In-Ordner manuell in der zentralen HTML-Datei eingebunden werden. Dies verursacht zusätzlichen Aufwand, jedoch konnte keine andere Lösung gefunden werden. Außerdem muss der Name jedes Plug-Ins in einer Konfigurationsdatei der Basisanwendung hinterlegt werden, denn es werden nur die Plug-Ins eingebunden, die hier aufgelistet wurden. Ein Nachteil der manuellen Einbindung einer JavaScript-Datei ist, dass nicht einfach eine Menge von Dateien hinterlegt werden kann, die automatisch verwendet werden. Wäre dies möglich, könnte die Anwendungslogik eines Plug-Ins auf mehrere JavaScript-Dateien aufgeteilt werden. Wird nur eine Datei eingebunden, muss jeglicher JavaScript-Code in dieser Datei zusammengefasst werden, was die Übersichtlichkeit während der Entwicklung hindert, da diese Datei mehrere tausend Zeilen Code enthalten kann.

Die Benutzung von CareNet_{mobile} soll auch ohne eine Internetverbindung möglich sein, wenn vorher alle relevanten Daten auf den Tablet-PC geladen wurden. Hierzu wird eine lokale Datenbank von der Basisanwendung bereitgestellt. Die Einzelnen Plug-Ins können über vordefinierte Methoden eigene Tabellen anlegen, um Plug-In spezifische Informationen abzuspeichern. Damit es beim ersten Öffnen eines Plug-Ins zu keiner größeren Verzö-



```

Datei Beispiel.js

//Konstruktor
function Beispiel(){
    //initialisiere Datenbank
    ...
}

Beispiel.init = function(){
    /*CONTENT_DIV_ID = Variable, die Id des Div-
     *      Elements enthält für Inhalt enthält
     *Methode „append“ fügt gegebenes HTML dem
     *Eltern-Element hinzu
     */
    $('#'+CONTENT_DIV_ID).append('<h1>Hallo</h1>');
}

//Weitere Methoden mit Beispiel.methodename...

```

Abbildung 4.6.: Grundstruktur einer Plug-In-JavaScript-Datei

gerung wegen des Anlegens benötigter Tabellen kommt, ist eine Funktion wünschenswert, mit der Plug-Ins Vorgänge anstoßen können, die schon vor dem ersten Öffnen ausgeführt werden. Um dies zu ermöglichen, muss die JavaScript-Datei die in Abbildung 4.6 gezeigten Methoden implementieren. Prinzipiell wird hier ein *Separated Interface*-Muster [22, S. 476] genutzt, bei dem generelle Methoden, die jedes Plug-In implementiert, jedoch mit unterschiedlicher Funktionalität, ausgelagert werden. Da in JavaScript die explizite Definition von Schnittstellen (Interfaces) nicht möglich ist, muss hier auf global verfügbare Methoden zurückgegriffen werden.

Benötigt wird zunächst ein Konstruktor, der nach dem Plug-In benannt wurde. Dieser wird aufgerufen, wenn ein Plug-In geladen wird, bei dem alle Abhängigkeiten erfüllt sind. Hier können beispielsweise Tabellen einer Datenbank initialisiert werden. Neben einem Konstruktor wird eine Methode zur Initialisierung der Anwendung benötigt. Diese wird aufgerufen, sobald ein Nutzer ein Plug-In öffnet. In dieser Methode muss die Anzeige der App in den Bereichen Navigation, Information und Inhalt angepasst werden. Beispielsweise könnte ein Button in die Navigationsleiste gesetzt werden oder, wie in Abbildung 4.6 gezeigt, dem Inhalts-Div der Schriftzug "Hallo" in dem HTML-Größenelement "<h1>" hinzugefügt werden. Würde die Plug-In-Struktur in Java implementiert, könnte die Methode *init()* in ein Interface gefasst werden, welches jedes Plug-in implementieren muss. Auch ein Konstruktor wäre gegeben. Dies wäre eine bessere Variante, da so sichergestellt werden kann, dass ein Programmierer die Methoden implementiert (alle Methoden eines verwendeten Interfaces müssen in Java implementiert werden). In JavaScript kann die Anforderung nur über Konventionen eingefordert werden, was zu menschlichen Fehlern durch Vergessen der Methoden führen kann.

Ähnlich wie bei der Namensgebung der Dateien müssen auch innerhalb der Dateien Konventionen eingehalten werden. Um zu verhindern, dass Methoden mit gleichen Namen oder CSS-Klassen mit gleichen Namen in verschiedenen Plug-Ins implementiert werden, die sich dann gegenseitig stören, muss jede Methode mit dem Namen des Plug-Ins beginnen (*Plug-In-Name.methode*), ebenso wie jede CSS-Klasse, jedoch hier ohne Punkt (*Plug-In-NameKlassenname*). Auch jede global verwendete Variable muss den Plug-In-Namen vorangestellt bekommen, um Konflikte zu vermeiden. Eine Nicht-Einhaltung der Konventionen kann zu einer völligen Funktionsstörung der Anwendung führen.

4.3. Das Plug-In Kontakte

Das Plug-In *Kontakte* wurde entworfen, um gezielt auf in CareNet hinterlegte Informationen über Personen und Organisationen zugreifen zu können, ohne dass diese einen Bezug zu einem aktuell im Tagesplan auszuführenden Auftrag haben. Beispielsweise soll die Telefonnummer eines Arztes abgerufen werden können, um Rückfragen zu einem Termin stellen zu können. Ein anderer Anwendungsfall wäre die Suche nach Angehörigen eines Klienten. Vorwiegend soll mit dem Plug-In auf Stammdaten wie Name, Anschrift, Alter und Kontaktmöglichkeiten zugegriffen werden können, jedoch ist im Falle von Klienten auch eine Erweiterung um Angaben zu Krankheitsverläufen, Vitaldatenstatistiken oder aktuelle Medikation denkbar. Im Rahmen dieser Arbeit wird die Anzeige jedoch auf Stammdaten reduziert.

In CareNet wird unterschieden, ob es sich bei einem Kontakt um eine Organisation oder eine Person handelt. Im Falle von Personen wird jeweils eine bestimmte Adressart wie Klient, Angehöriger oder Ehepartner (nicht abschließend) unterschieden. Bei Organisationen können einige Einrichtungen des Gesundheitssystems angegeben werden, z.B. ein Krankenhaus, ein ambulanter Pflegedienst oder auch ein Hausarzt. Angezeigt werden die Kontakte über sogenannte *Sichten* (Views). Eine Sicht stellt einen Filter dar, der jeweils nur die Kontakte einer bestimmten Gruppe anzeigt. Diese sind beispielsweise *Alle Kontakte*, *Alle Angehörigen*, *Alle Klienten* oder auch *Alle Organisationen*. Diese Filterung nach Sichten sollte auch in CareNetmobile möglich sein. Bei dem Verhältnis von Kontakten zu Sichten handelt es sich um eine n:n-Beziehung. Das bedeutet, dass ein Kontakt in mehreren Sichten enthalten sein kann und mehrere Kontakte einer Sicht zugeordnet werden. Beispielsweise wird ein Klient sowohl unter *Alle Kontakte* als auch unter *Alle Klienten* angezeigt werden. Außerdem wird es in der Regel mehr als einen Klienten geben.

Da auch die Kontaktinformationen offline verfügbar sein sollen, müssen entsprechende Tabellen zur lokalen Speicherung vorhanden sein. Als Speicher wird eine von HTML5 angebotene Sqlite-Datenbank verwendet. Diese speichert Daten persistent, d.h. auch wenn die Anwendung auf dem Tablet-PC geschlossen wird oder der Tablet-PC heruntergefahren wird, werden die Daten nicht gelöscht. Beim Start von CareNetmobile wird eine Datenbank angelegt bzw. überprüft, ob bereits eine Datenbank mit entsprechendem Namen vorhanden ist. Da die Datenbank innerhalb des von PhoneGap angebotenen Browsers/Web Views angelegt wird, kann von außen bzw. anderen auf dem Tablet-PC installierten Anwendungen nicht darauf zugegriffen werden. So entsteht kein Sicherheitsproblem durch die Speiche-

rung sensibler Daten auf dem Tablet-PC. Alle Daten werden in Tabellen gespeichert, die in unbegrenzter Zahl erstellt werden können. Abbildung 4.7 zeigt die notwendige Tabellenstruktur, um Sichten und Kontakte lokal abspeichern zu können. Die linke Tabelle enthält die IDs und Namen aller Sichten, die aus CareNet geladen wurden. In der rechten Tabelle sind alle Kontakte mit IDs und einer Reihe von Attributen enthalten. Die mittlere Tabelle verbindet beide Tabellen, indem sie durch Gegenüberstellung der IDs der Sichten und der Kontakte eine Abbildung schafft, welcher Kontakt in welcher Sicht enthalten ist. Beispielsweise ist in Abbildung 4.7 der Kontakt mit der ID C2 in den Sichten mit den IDs V1 und V2 enthalten, während die anderen Kontakte jeweils nur in einer Sicht enthalten sind. Wählt ein Nutzer eine Sicht aus, wird in der Tabelle *Mapping* nach allen Kontakt-IDs gesucht, die der ID der Sicht zugeordnet sind. Mit Hilfe der gesammelten IDs werden die eigentlichen Kontaktinformationen aus der Tabelle *Contacts* abgefragt, welche nun dargestellt werden können.

Die Darstellung soll sowohl die Sichten als auch die darin enthaltenen Kontakte auf einen Blick enthalten. Für die Auflistung der verfügbaren Sichten wurde eine einzeilige Tabelle ohne Kopfzeile gewählt. Die zugehörigen Kontakte einer Sicht werden als Kacheln bzw. als Galerie ([23, S. 17]) dargestellt. Abbildung 4.8 zeigt einen Ausschnitt aus der Darstellung der Kontakte in CareNetmobile . In dem orangefarbenen Balken ist vermerkt, ob es sich um eine Person oder eine Organisation handelt, im grauen Kasten darunter werden Vorname und Name sowie Wohnort aufgeführt. Es könnten beispielsweise auch die Telefonnummer, die E-Mail-Adresse oder ein Foto einer Person angezeigt werden, jedoch wurden aus Gründen der Übersichtlichkeit diese Informationen gewählt.

Jedes Element in der Tabelle der Sichten weist nach oben und unten einen gewissen Abstand vom Text zum Rand auf, um genug Fläche für eine Auswahl mit dem Finger zu bieten und somit die Wahrscheinlichkeit neben die gewollte Schaltfläche zu tippen zu verringern. Das gleiche gilt für die Kacheln zur Auflistung der Kontakte. Neben der Informationsfunktion soll auch hier eine komfortable Bedienung ermöglicht werden.

Wählt ein Nutzer einen Kontakt aus, werden alle relevanten Informationen in einem separaten Fenster dargestellt. Momentan findet lediglich eine Unterscheidung bei der Darstellung von Personen und Organisationen statt, jedoch ist eine Differenzierung nach Adressarten geplant.

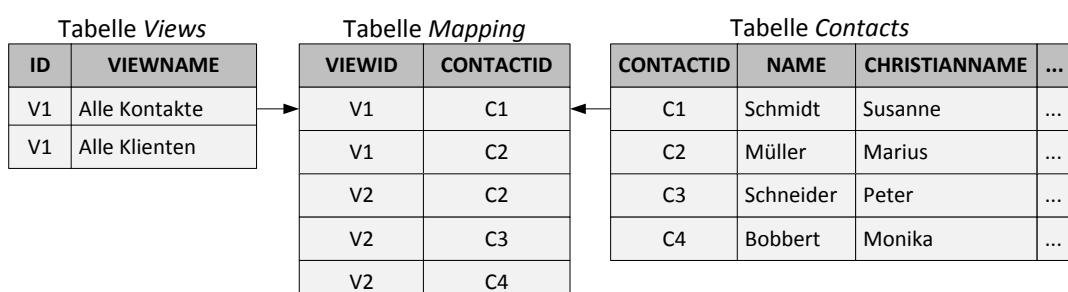


Abbildung 4.7.: Tabellen in der lokalen Datenbank des Plug-Ins *Kontakte*

 															
<p>Sie sind eingeloggt als: Thomas Knapp</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #0070C0; color: white; padding: 5px;">Kontaktgruppe</th><th style="background-color: #0070C0; color: white; padding: 5px;">Alle Klienten</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;">Alle Generalbevollmächtigte/ges. best. Betreuer</td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Alle stationären Dienste</td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Alle Organisationen</td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Alle Benutzer</td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Alle Kommunen</td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px; background-color: #333; color: white;">Alle Klienten</td><td style="padding: 5px;"></td></tr> </tbody> </table>	Kontaktgruppe	Alle Klienten	Alle Generalbevollmächtigte/ges. best. Betreuer		Alle stationären Dienste		Alle Organisationen		Alle Benutzer		Alle Kommunen		Alle Klienten	
Kontaktgruppe	Alle Klienten														
Alle Generalbevollmächtigte/ges. best. Betreuer															
Alle stationären Dienste															
Alle Organisationen															
Alle Benutzer															
Alle Kommunen															
Alle Klienten															

Abbildung 4.8.: Ausschnitt der Darstellung von Kontakten innerhalb einer Sicht

Zum Zeitpunkt der Fertigstellung dieser Arbeit hatte das Plug-In nur eine Funktion zum Anzeigen von Kontakten. Eine Editierfunktion bestehender Kontakte oder die Neuanlage von Kontakten ist nicht möglich.

Die in Abbildung 4.7 gezeigten Tabellen werden im Konstruktor des Plug-Ins erzeugt bzw. wird überprüft, ob sie vorhanden sind. Wird das Plug-In geöffnet und es sind keine Kontaktdateien vorhanden, wird angenommen, dass das Plug-In zum ersten Mal geöffnet wird und es wird automatisch ein Versuch unternommen, Kontaktinformationen von CareNet zu laden. Sind bereits Daten vorhanden, werden diese lediglich angezeigt. Über einen Button in der Navigationsleiste kann der Nutzer auch manuell einen Aktualisierungsvorgang anstoßen. Ist dieser abgeschlossen, werden die verfügbaren Views angezeigt.

Da Sichten, Kontakte und das Mapping in verschiedenen Tabellen abgespeichert werden, muss sichergestellt werden, dass alle Daten vollständig geladen wurden, bevor dem Nutzer die Möglichkeit gegeben wird, Sichten auszuwählen. Sind die Kontakte zu einer Sicht noch nicht vollständig geladen und der Nutzer möchte einen dieser Kontakte anzeigen lassen, kommt es zu einem Fehler, versucht wird, auf einen leeren Datensatz zuzugreifen. Das Laden der Daten von CareNet erfolgt über REST-basierte AJAX-Aufrufe. AJAX steht für Asynchronous JavaScript and XML. Wie der Name schon sagt, handelt es sich hierbei um *asynchrone* Webservice-Aufrufe, was bedeutet, dass der Aufruf getätigt wird, ohne dass bekannt ist, wann das Ergebnis zurück kommt. Da die Ausführung des übrigen JavaScript-Codes nicht automatisch auf das Ergebnis wartet, auch wenn dieses benötigt wird, muss manuell ein "Warte-Verfahren" implementiert werden. Hierzu wird das Entwurfsmuster *Command* (Kommando) verwendet. Die Idee ist, dass ausführbarer Code bzw. im Falle von JavaScript eine ganze Methode in ein Objekt gekapselt wird, welche der asynchronen Methode als Parameter übergeben wird. Ist die Ausführung der asynchronen Methode beendet, kann diese den übergebenen Code ausführen, der in der Regel Logik zum Verarbeiten der angefragten Daten enthält. Die als Parameter übergebene Methode wird auch *Callback-Methode* (Rückruf-Methode) genannt.

Vor dem Abfragen der Views und der Kontaktdaten wird für beide Datensätze ein True-False-Marker (Wahr-Falsch-Marker) angelegt, der besagt, ob bereits alle Daten übertragen und in die Datenbank eingepflegt wurden. Initial enthält der Marker den Wert *false* (falsch). Wird die Callback-Methode aufgerufen, wird der entsprechende Marker auf *true* gesetzt, um anzuzeigen, dass der asynchrone Methodenaufruf beendet wurde und anschließend überprüft, ob dies für alle Methoden zum Abrufen der Kontaktdaten gilt. Stehen alle Marker auf *true*, wird die Anzeige der Daten angestoßen.

Da Aufrufe an die lokale Sqlite-Datenbank ebenfalls asynchron verlaufen, findet hier auch das Command-Muster Anwendung.

4.4. Das Plug-In *Touren*

Das Plug-In *Touren* bildet den funktionellen Kern von CareNet*mobile*. Hier werden alle Informationen und Funktionen, die zur Dokumentation der übertragenen ärztlichen Tätigkeiten notwendig sind, gekapselt zur Verfügung gestellt. Touren müssen hier übersichtlich dargestellt werden, auszuführende Leistungen pro Klient müssen abrufbar sein und Zusatzinformationen zum Patienten müssen verfügbar sein, ohne das Plug-In verlassen zu müssen. Die gesamte Funktionalität basiert also auf dem Vorhandensein von Daten, die von CareNet bereitgestellt werden. Das Plug-In *Touren* bietet keinerlei Funktionen an, die ohne Touren- oder Patientendaten verwendbar sind. Um aber auch ohne eine Internetverbindung mit dem Plug-In arbeiten zu können (vgl. Anforderungen in Kapitel 3.1), bilden eine Reihe von Tabellen in der lokalen Sqlite-Datenbank das Rückgrat des Plug-Ins.

Das Datenmodell in CareNet ist so aufgebaut, dass alle Elemente wie Touren, Stationen oder auch Pflegeleistungen eigene Datenobjekte sind. Eine Verbindung zwischen zwei Datenobjekten wird in einem *Dossier*, das jeweils einem bestimmten Datenobjekt zugeordnet wird, festgehalten. Es besteht aus einer Auflistung von Objektpaaren, welche eine Verbindung haben, von denen immer ein Objekt das dem Dossier zugeordnete ist. Zu jedem Objekt ist die entsprechende ID, der Objekttyp (z.B. TOUR) und die Art der Verbindung zum Partnerobjekt angegeben. Art der Verbindung kann beispielsweise "BIDIRECTIONAL" sein, was bedeutet, dass beide Objekte mit einer ungerichteten Kante verbunden sind, d.h. beide wissen von der Verbindung mit dem jeweils anderen Objekt. Alternativ könnte es auch eine gerichtete Kante sein, bei der nur ein Datenobjekt von der Verbindung weiß.

Zur Darstellung einer Tour in CareNet*mobile* wird eine Tour, eine Menge von Stationen und für jede Station eine Menge von Pflegeleistungen benötigt. Stationen werden in CareNet als Termine (*APPOINTMENT*) angesehen, da es sich im Prinzip um ein Treffen mit dem Klienten zu einem bestimmten Zeitpunkt handelt. Abbildung 4.9 zeigt im oberen Teil den Zusammenhang zwischen den benötigten Datenobjekten Tour, Appointment und Service in Baumform. Die bidirektionale Verbindung zwischen den Datenobjekten ist durch ungerichtete Pfeile markiert. Diese Konstellation von Datenobjekten erfordert drei Tabellen in der lokalen Datenbank, die in Abbildung 4.9 im unteren Teil dargestellt sind. Eine zur Speicherung des Datenobjektes *TOUR*, eine für die zugeordneten Termine und eine Dritte für die Menge der zu erbringenden Leistungen (*SERVICES*). Die Tabellen werden

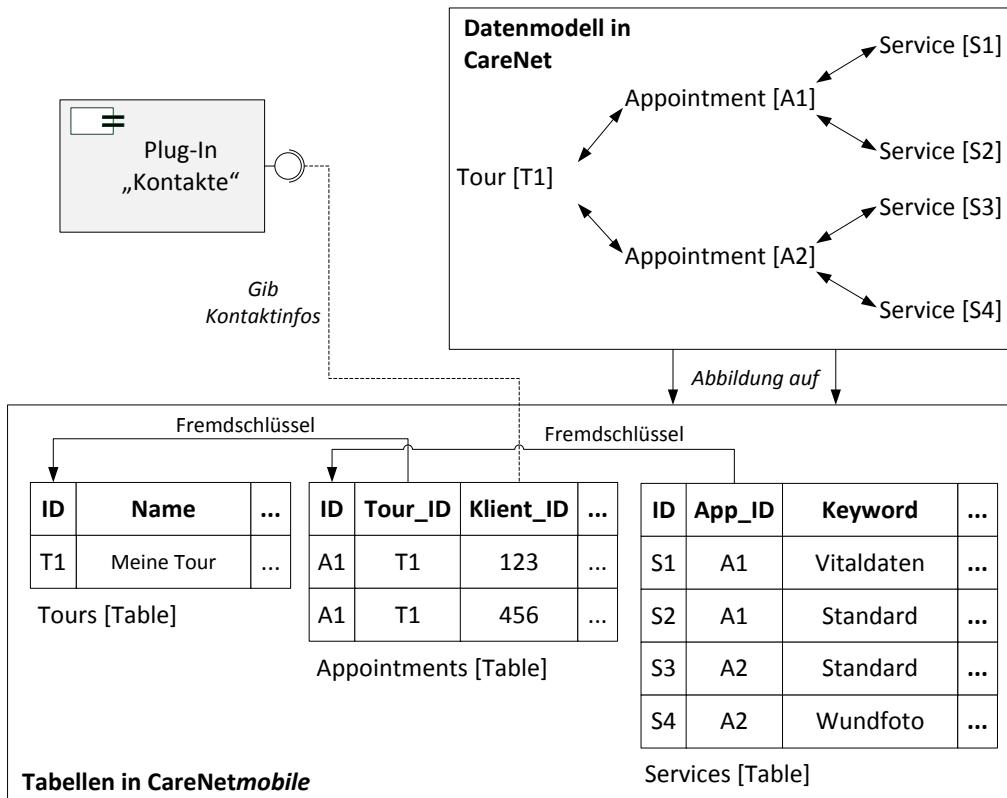


Abbildung 4.9.: Datenmodell und zugehörige Datenbanken im Plug-In Touren

im Konstruktor des Plug-Ins angelegt bzw. werden diese aus dem lokalen Speicher geladen, falls sie schon zu einem früheren Zeitpunkt angelegt wurden. Die Verbindung zwischen den Datenobjekten, welche in CareNet in den genannten Dossiers beschrieben wird, wird in CareNetmobile über Fremdschlüsseleinträge aufgelöst. Bei Fremdschlüsseleinträgen wird die ID eines in einer anderen Tabelle liegenden Datenobjektes gespeichert, um dort das richtige Objekt identifizieren zu können. Da Pflegeleistungen eine Verfeinerung einer Station/eines Termins darstellen und Stationen einer Tour untergeordnet sind, besteht absteigend eine 1:n-Beziehung zwischen den Datenobjekten. Aus diesem Grund wird jeweils dem Datenobjekt, welches nur *einem* anderen Objekt zugeordnet wird, der Fremdschlüssel gegeben. Beispielsweise kann dem Tabelleneintrag des Service mit der ID *S1* in der rechten Tabelle in Abbildung 4.9 entnommen werden, dass er dem Termin/der Station mit der ID *A1* zugeordnet ist, während von der Station nicht auf die zugeordneten Services geschlossen werden kann. Die bidirektionale Verbindung aus dem Datenmodell von CareNet wurde für die lokale Datenbank nicht übernommen, da eine Verbindung der Daten mit Fremdschlüsseln einfacher zu realisieren ist, da ohnehin nur eine 1:n-Beziehung abgebildet werden muss. Würde gefordert, von einem Appointment direkt auf die Menge der zugewiesenen Services schließen zu können, müsste hierzu entweder eine weitere Tabelle, ähnlich der in Abbildung 4.7 gezeigten, angelegt werden, oder es müssten die IDs der Services als Zeichenkette konateniert im Appointment-Datenobjekt abgelegt werden, was einen erhöhten Aufwand beim Lesen der IDs bedeuten würde.

Die REST-Schnittstelle von CareNet bietet nur das Laden von allen oder einzelnen Objekten eines Typs an, die der angemeldeten Person freigegeben sind. Eine MFP bekommt beispielsweise nur Informationen angezeigt, die sie für die Ausführung Ihrer Arbeit benötigt oder die explizit für sie freigegeben wurden. Eine Differenzierung einer Menge von Datenobjekten nach zeitlichen Kriterien in der Abfrage ist nicht möglich. Soll nun also eine Tour eines bestimmten Tages vom Server geladen werden, müssen die Daten aller Touren abgefragt werden, damit auf dem Tablet-PC eine Filterung der Daten erfolgen kann. Ein Problem mit diesem Vorgehen kann entstehen, wenn nur ein Internetzugang mit begrenztem Datenvolumen zur Verfügung steht und dieses durch das Laden von Daten aufgebraucht wird, die eigentlich nicht benötigt werden. Die Folge wären entstehende Mehrkosten oder eine eingeschränkte Nutzbarkeit der Internetverbindung, wenn die Übertragungsgeschwindigkeit wegen Überschreitung des Datenvolumens vom Anbieter gedrosselt wird. Da CareNet *mobile* auch bei ambulanten Diensten eingesetzt werden soll, die während eines Besuchs bei einem Klienten in der Regel nur auf Internetverbindungen über Mobilfunkanbieter zurückgreifen können, kann dies ein Problem darstellen. Eine Reduzierung des Problems auf Einzelfälle kann dadurch erreicht werden, dass die benötigten Daten zu Beginn der Schicht über eine WLAN-Verbindung mit unbegrenztem Datenvolumen geladen werden und nur noch gezielte Aktualisierungen über das mobile Netzwerk durchgeführt werden. Eine weitere Möglichkeit wäre die Anpassung der REST-Schnittstelle, sodass kontextsensitive Daten abgefragt werden können, jedoch bedeutet dies umfangreiche Veränderungen am zentralen System, von denen zum momentanen Zeitpunkt abgesehen wird. Zwar ist es jetzt schon möglich, statt des gesamten Datensatzes nur eine Liste von Quasi-Signaturen bestehend aus einem Bezeichner, der ID und der URI zum Dossier herunterzuladen, jedoch enthält diese nicht die Informationen über die zeitliche Einordnung des Datenobjektes. Hiermit kann folglich keine Reduzierung durch Differenzierung nach Zeitstempel erreicht werden. In Abbildung 4.9 ist eine Verbindung zwischen den Terminen und dem Plug-In *Kontakte* vermerkt. Diese Verbindung existiert, da jeder Termin einem bestimmten Klienten zugeordnet ist, von dem im Datenobjekt *Appointment* aber nur die ID bekannt ist. Um Redundanzen zu vermeiden, lädt nicht das Touren-Plug-In selbst die Daten des Nutzer von CareNet, sondern es wird eine Schnittstelle des Plug-Ins *Kontakte* genutzt, um an die Nutzerdaten zu kommen, da diese hier bereits alle vorhanden sind. Damit es zu keinem Bruch in der Verarbeitung kommt, muss die Abhängigkeit vom Touren-Plug-In zum Kontakte-Plug-In inklusive benötigter Version in der Konfigurationsdatei des Plug-Ins *Touren* vermerkt werden.

Wird das Plug-In geöffnet, wird zunächst überprüft, ob Daten in den Tabellen vorhanden sind. Sind keine Daten vorhanden, wird angenommen, dass das Plug-In zum ersten Mal nach der Installation der Anwendung oder der Löschung der Daten geöffnet wurde und der Vorgang zum Laden der Daten vom Server wird automatisch angestoßen. Sind Daten vorhanden, kann aber keine Tour für das aktuelle Datum gefunden werden, wird der Nutzer gefragt, ob er nach aktuellen Informationen suchen lassen möchte. Momentan ist noch keine Funktionalität für die Auswahl einer Tour an einem bestimmten Datum vorgesehen,

da in der Regel nur die Daten für die aktuelle Schicht relevant sind.

Das Laden der Daten erfolgt sequentiell. Zuerst werden alle vorhandenen Services geladen und in die lokale Datenbank eingepflegt. Das Feld für die ID der zugehörigen Station, der Fremdschlüssel, wird zunächst leer gelassen. Anschließend werden alle Touren abgefragt und lokal gespeichert. Hier können alle Felder gefüllt werden, da keine Abhängigkeiten gesetzt werden. Als letztes werden die vorhandenen Termine/Stationen geladen. Diese werden in die lokale Datenbank geschrieben, das Feld für den Schlüssel zur zugehörigen Tour bleibt auch hier zunächst frei. Da ein Termin zwischen Tour und Services steht, enthält dessen Dossier wegen der bidirektionalen Verbindungen der Datenobjekte alle notwendigen Informationen, um die Fremdschlüsselbeziehungen in den lokalen Tabellen aufzulösen. Folglich wird jedes Objektpaar im Dossier eines Termins angeschaut und überprüft, ob es entweder die Verbindung zu einer Tour, einem Service oder einem Klienten beschreibt. Alle anderen Dossiereinträge sind für das Füllen der in Abbildung 4.9 gezeigten Tabellen irrelevant. Wird beispielsweise eine Verbindung zu einem Service gefunden, wird dieser in der lokalen Datenbank gesucht und der Fremdschlüssel zum Termin gesetzt. Wurde das gesamte Dossier gelesen, sollten, unter der Annahme, dass es von CareNet korrekt aufgebaut wurde, alle Fremdschlüsselverbindungen in der lokalen Datenbank gesetzt sein und der vollständige Datenbaum somit lokal verfügbar sein.

Das Vorgehen, zuerst die Services und Touren vollständig zu laden, bevor die Termine abgefragt werden und die entsprechenden Verbindungen gesetzt werden, stellt sicher, dass nicht versucht wird, auf ein lokales Datenobjekt zuzugreifen, welches noch nicht vollständig geladen wurde. Da sowohl die Serverzugriffe als auch die Zugriffe auf die lokale Datenbank asynchron ablaufen, könnten Fehlzugriffe theoretisch vorkommen. Zur Vermeidung eines solchen Fehlers wird, wie auch im Plug-In Kontakte (vgl. Abschnitt 4.3), das Kommand-Muster verwendet, um feststellen zu können, wann alle Daten vollständig geladen wurden.

The screenshot shows the CareNet mobile application interface. At the top, there is a header bar with the CareNet mobile logo, two buttons labeled "Senden Daten" and "Aktualisiere Touren", and a "Startseite" button with a house icon. Below the header, a message says "Sie sind eingeloggt als: Thomas Knapp". The main content area is titled "Ihre heutigen Stationen:" and lists three stations:

Platznr.	Name	Postleitzahl	Aktionen
1	Herr Master Arbeit	76131 Karlsruhe	65 Offen
2	Frau Erna Ernst	76131 Karlsruhe	12 Offen
3	Herr Mark Müller	76133 Karlsruhe	97 Offen

Abbildung 4.10.: Übersicht der abzuarbeitenden Stationen einer Tour

Wurde keine Tour für das aktuelle Datum gefunden und der Nutzer entscheidet sich gegen eine Aktualisierung oder ist diese aufgrund einer fehlenden Internetverbindung nicht möglich, so erscheint ein Hinweis, dass keine Stationen angezeigt werden können. Der Nutzer kann in diesem Fall das Touren-Plug-In nicht verwenden.

Wurde eine Tour gefunden, werden alle abzuarbeitenden Stationen dieser Schicht aufgelistet. Abbildung 4.10 zeigt die Darstellung in der Anwendung. Da jede Station unmittelbar einer Person zugeordnet werden kann, werden Anrede, Vor- und Nachname sowie der Ort des Besuchs angezeigt. Neben den Stammdaten befindet sich ein Schlüsselsymbol jeweils in Verbindung mit einer Zahl. Da die Klienten häufig immobil sind und nicht mehr selbst die Tür öffnen können, besitzt der Pflege- oder Krankendienst in der Regel Schlüssel, um sich ohne Hilfe Dritter Zutritt zu den Aufenthaltsräumen des Klienten verschaffen können. Die Schlüssel werden häufig an einem Bund zusammengefasst und nummeriert. Um mit der Suche nach dem richtigen Schlüssel keine Zeit zu verlieren, wird die Nummer des einem Klienten zugeordneten Schlüssels direkt in der Übersicht der Stationen angezeigt. Handelt es sich nicht um einen ambulanten Dienst, sondern um eine stationäre Behandlung, kann an dieser Stelle die Nummer des Zimmers angezeigt werden. Für jede Station wird außerdem der aktuelle Status der Arbeit angezeigt. So kann festgestellt werden, ob alle Leistungen an einer Station erbracht wurden, ohne die Leistungen im Einzelnen zu betrachten. In Abbildung 4.10 zeigt das rote Kreuz und die Bezeichnung "Offen" an, dass noch abzuarbeitende Tätigkeiten vorhanden sind.

Die Navigationsleiste enthält in dieser Ansicht drei Buttons. Auf der linken Seite befindet sich ein Button zum Senden der dokumentierten Leistungen an CareNet sowie ein Button zum Abfragen aktueller Toureninformationen. Der rechte Button navigiert zurück auf die Startseite (vgl. Abbildung 4.5).



Abbildung 4.11.: Übersicht der zu erbringenden Dienste einer Station

Wählt ein Nutzer durch Berührungen der Nummerierung im blauen Kreis eine Station aus, werden die zu erbringenden Leistungen der entsprechenden Station aufgelistet. Abbildung 4.11 zeigt für einen Klienten drei zu erbringende Leistungen, von denen jede durch ein Stichwort und ein Statussymbol gekennzeichnet wird. Damit eine MFP immer weiß, für welchen Klienten sie gerade die Dokumentation durchführt, wird auf der linken Seite im Informationsteil der Name des Klienten angezeigt. Falls den Stammdaten des Klienten ein Bild hinzugefügt wurde, ist denkbar, dass dieses zusätzlich zum Namen angezeigt wird. Jede aufgelistete Leistung ist als ein aufklappbares Element realisiert. Durch Antippen

des Listenelements klappt ein Bereich aus, in dem spezifische Informationen und Schaltflächen zur Bearbeitung des Status vorhanden sind. Zur Umsetzung der Anforderungen der Dokumentation ärztlich übertragener Leistungen wurden im Wesentlichen drei Arten von Leistungen identifiziert, bei denen unterschiedliche Schaltflächen benötigt werden:

Standardleistung Dieser Leistungstyp deckt alle Tätigkeiten ab, bei denen keine speziellen Daten dokumentiert werden müssen, sondern nur bestätigt werden muss, ob diese erbracht wurde oder nicht.

Wunddokumentation Im Falle der Wunddokumentation müssen Bilder von Wunden aufgenommen werden. Hierzu sind Schaltflächen zum Starten der geräteinternen Kamera notwendig, außerdem muss der Leistung bzw. dem Klienten eine Bilddatei zugeordnet werden, was zusätzliche Hintergrundfunktionalität erfordert.

Vitalwertmessug Bei der Vitaldatenerfassung müssen zwei Werte für den Blutdruck (sy-tolisch, oberer Wert, und diastolisch, unterer Wert) und ein Wert für den Puls dokumentierbar sein. Dies erfordert drei Schaltflächen für die Eingabe der Werte und eine weitere, um den Speichervorgang anzustoßen.

In CareNet muss derjenige, der eine Tour für eine MFP zusammenstellt, jeder Leistung einen der genannten Typen über ein Drop-Down-Menü zuweisen, damit diese in CareNetmobile identifiziert werden kann und die grafische Oberfläche und die damit verbundene Funktionalität entsprechend angepasst werden kann.

Ihre auszuführenden Tätigkeiten:

Standardleistung

Beschreibung: Tropf anhängen

Nicht durchgeführt
Erledigt

Warum kann die Tätigkeit nicht durchgeführt werden?

Der Patient war nicht anzutreffen.

Abbildung 4.12.: Darstellung einer Standardleistung

In Kapitel 3.1 wurde gefordert, dass es bei jeder zu erbringenden Leistung die Möglichkeit bestehen muss, diese mit Begründung als nicht durchgeführt zu markieren. Abbildung 4.12 zeigt die Umsetzung dieser Anforderung. Dargestellt ist eine Standardleistung, der neben einer Beschreibung der Tätigkeit ein Ankreuzfeld "Nicht durchgeführt" und der Button "Erledigt" zur Verfügung stehen. In der Abbildung wurde ein Haken im Ankreuzfeld gesetzt, sodass die Leistung nun als nicht erbracht gilt. Der Button "Erledigt" wird in diesem Fall ausgeblendet. Er wird erst dann wieder aktiviert, wenn der Haken im Ankreuzfeld entfernt wird. Darunter befindet sich ein Textfeld, in dem eine Begründung eingetragen werden kann, wieso die Leistung nicht erbracht wurde. Es kann im Quellcode konfiguriert

werden, wie viele Zeichen eine Begründung mindestens lang sein muss, um als ausreichend angesehen zu werden. Im Beispiel in Abbildung 4.12 wurde die Mindestlänge auf 10 Zeichen gesetzt, welche der Satz "Der Patient war nicht anzutreffen", erfüllt. Der Status der Leistung ist dann weder erfüllt, noch offen, da diese grundsätzlich bearbeitet wurde, jedoch nicht das gewünschte Ergebnis erreicht werden konnte. Aus diesem Grund wurde ein dritter Status "eingeschränkt erfüllt" eingeführt, der diesen Zustand beschreibt und durch eine orangefarbene Tilde repräsentiert wird (siehe rechte Seite des grauen Balkens). Eine gesamte Station gilt als "eingeschränkt erfüllt", wenn alle Leistungen entweder den genannten Zwischenzustand haben oder vollständig erbracht wurden. In diesem Fall wird für eine gesamte Station das gleiche Symbol wie bei einer Leistung verwendet. Für den Zustand "eingeschränkt erfüllt" gibt es keine eigene Repräsentation in der lokalen Datenbank, sondern er wird aus der Attributkombination *Erledigt = falsch* und *Länge Begründung > Mindestlänge* abgeleitet. Wird der Haken im Ankreuzfeld entfernt, wird auch das Textfeld für die Begründung ausgeblendet.

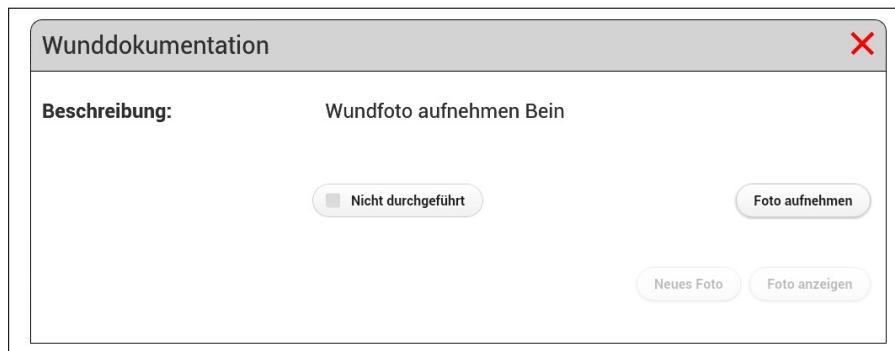


Abbildung 4.13.: Darstellung der Schaltflächen zur Wunddokumentation

Die Schaltflächen zur Wunddokumentation sind in Abbildung 4.13 zu sehen. Sie unterscheiden sich von der Standardleistung dadurch, dass anstatt einem Button "Erledigt" ein Button "Foto aufnehmen" verfügbar ist, mit dem die geräteinterne Kamerafunktion aufgerufen wird. Wurde ein Foto erfolgreich aufgenommen, wird der "Foto aufnehmen"-Button ausgeblendet und die Buttons "Foto anzeigen" und "Neues Foto" eingeblendet. Ersterer öffnet ein Fenster, in dem das aufgenommene Foto noch einmal angesehen werden kann, letzterer ermöglicht es, ein neues Foto aufzunehmen, welches durch das alte Foto ersetzt wird. Möchte der Nutzer im Nachhinein das Foto löschen und die Tätigkeit als "Nicht durchgeführt" markieren, kann er analog zur Standardleistung den entsprechenden Haken setzen. Es öffnet sich ein Fenster, in welchem der Nutzer bestätigen muss, dass er das Bild tatsächlich löschen möchte, und wenn er dies bestätigt, wird auch hier ein Textfeld zum Eintragen einer Begründung eingeblendet. Ein separater Button zum Löschen des Bildes ist nicht vorgesehen, da dies zu dem Status "Offen" führen würde, welcher nach einer Bearbeitung nicht erwünscht ist. Eine MFP ist angehalten, jede Leistung entweder durchzuführen oder zu begründen, warum sie Leistung nicht durchgeführt werden konnte. Wird ein Wundfoto mithilfe der nativen Kamerafunktion über eine in PhoneGap implementierte Funktion aufgenommen, wird es persistent im Speicher der Anwendung abgelegt und der Pfad zu der Datei zurückgeliefert. Um die Datei einer Leistung zuordnen zu können, wird

der Pfad als Attribut dem entsprechenden Datenobjekt in der lokalen Sqlite-Datenbank abgespeichert.

The screenshot shows a mobile application interface for vital value measurement. The title bar says 'Vitalwertmessung' with a green checkmark icon. Below it, the description 'Beschreibung: Blutdruck, Puls' is displayed. The 'Blutdruck' section contains two sliders: one for systolic pressure (mmHG sys) set at 124 and another for diastolic pressure (mmHG dia) set at 71. The 'Puls' section contains a slider for heart rate (bpm) set at 70. At the bottom left is a button labeled 'Nicht durchgeführt' (Not performed), and at the bottom right is a button labeled 'Speichern' (Save).

Abbildung 4.14.: Darstellung der Schaltflächen zur Erfassung von Blutdruck und Puls

Die Umsetzung der Erfassung von Blutdruck und Puls (Vitaldaten) ist in Abbildung 4.14 zu sehen. Für jeden einzugebenden Wert ist ein Schieberegler vorhanden, mit dem ein Wert aus einem Intervall möglicher Werte ausgewählt werden kann, sowie ein Textfeld, in dem der Wert dargestellt wird. Möchte ein Nutzer den Schieberegler nicht benutzen, ist es durch Berührung des Textfeldes möglich, über die virtuelle Tastatur einen Wert einzugeben. Das Setzen von Intervallen hat den Effekt, dass Werte, die naturgemäß nicht möglich sind, wie z.B. ein Puls von 500 Schlägen po Minute, von vorn herein ausgeschlossen werden. So wird eine einfache implizite Plausibilitätsprüfung implementiert, die ein Vertippen mit Werten außerhalb des möglichen Bereiches ausschließt. Falscheingaben innerhalb des zulässigen Wertebereiches können so jedoch nicht ausgeschlossen werden. Nach der Eingabe muss der Nutzer den Speichervorgang manuell über den Button "Speichern" anstoßen. In Abbildung 4.14 wurde dies bereits getan und die Leistung gilt somit als erbracht (zu erkennen am grünen Haken). Eine automatische Erfassung mittels über Bluetooth angegeschlossene Messgeräte, wie in Kapitel 3.1 angedacht, konnte aufgrund der begrenzten Zeit nicht implementiert werden, ist jedoch grundsätzlich technisch möglich.

Ein Messwert (bestehend aus Blutdruck *und* Puls) ist in CareNet als eigener Datentyp dargestellt, der einem Kontakt zugewiesen wird. Auch in CareNet_{mobile} wird diese Teilung übernommen und die Werte werden nicht direkt im Datenobjekt einer Leistung gespeichert. Stößt der Nutzer den Speichervorgang an, wird ein neues Messwert-Datenobjekt in einer separaten Tabelle angelegt. Als ID wird die ID der Leistung/des Service verwendet, der die Erfassung der Messwerte "in Auftrag" gegeben hat, da so eine Fremdschlüsselbeziehung zwischen Leistung und Messwert hergestellt werden kann. Da es sich hier um eine 1:1-Beziehung handelt, können trotz Wiederverwendung der ID keine Inkonsistenzen entstehen. Die Verbindung vom Messwert zum Klienten wird mit in das Messwert-Datenobjekt integriert, indem dessen ID als eigenes Attribut mitgespeichert wird.

Der Vorteil einer lokalen separaten Liste von Messwerten ist die bessere Auswertbarkeit zu statistischen Zwecken. Soll beispielsweise für einen Patienten der Verlauf des Blutdrucks

der letzten vier Wochen erstellt werden, ist dies mit einer Messwerttabelle mit enthaltener Klienten-ID durch *eine* Anfrage an alle Datensätze eingeschränkt auf einen bestimmten Klienten und ein Zeitfenster möglich. Wären die Messwerte in den Leistungen enthalten, die nur über die Station einem Klienten zugeordnet werden können, müsste zur Überprüfung, ob ein Service einen relevanten Messwert enthält, jedes Mal eine zusätzliche Anfrage an die Datenbank gemacht werden, um den Klienten zu prüfen. Dies kostet bei einer größeren Datenmenge Zeit und schränkt somit die Nutzbarkeit ein, die durch die Verwendung einer *separaten* Tabelle eingespart werden kann.

CareNet mobile		Aktuelle Behandlung	Klientenakte
Sie sind eingeloggt als: Thomas Knapp		Name	Herr Master Arbeit
Sie behandeln gerade: Master Arbeit		Adresse	Schönfeldstraße 8 76131 Karlsruhe
		Telefon (privat)	06221 - 000
		E-Mail	thesis@email.de
		Alter	79
		Pflegestufe	Beantragt
		Vitaldatenstatistik	Zukünftige Termine
		Vorgänge	

Abbildung 4.15.: Darstellung der Klientenakte im Plug-In *Touren*

Befindet sich eine MFP bei der Dokumentation von Leistungen, kann es nötig sein, auf die Daten eines Klienten zuzugreifen, um Zusatzinformationen zu einer Behandlung zu bekommen. Um dies möglichst komfortabel und zeitsparend zu gestalten, wurde eine direkte Schnittstelle zum Plug-In *Kontakte* implementiert, die es ermöglicht, aus einem anderen Plug-In heraus Informationen zu einem Kontakt nachzufragen, ohne diesen redundant zu speichern. Im Plug-In *Touren* wird die Navigation zu der so genannten *Klientenakte* als Reiter (auch *Tab*) realisiert (vgl. Abbildung 4.15). Durch Berühren des Reiters werden entweder die Schaltflächen zur Dokumentation der aktuellen Behandlung oder die Klientendaten angezeigt. Die Umsetzung der Navigation als Reiter soll dem Nutzer vermitteln, dass die beiden Elemente einem Kontext angehören. Außerdem ist diese Art der Seitenstruktur, bei der scheinbar mehrere Seiten übereinander liegen, mit einer hohen Wahrscheinlichkeit den meisten potenziellen Nutzern von CareNetmobile bekannt, da sie häufig in Browsern zur Darstellung mehrerer Internetseiten verwendet wird. Das Ausnutzen eines bekannten Bedienkonzeptes soll die intuitive Bedienung der mobilen Anwendung fördern.

Die in Abbildung 4.15 dargestellten Informationen haben nur beispielhaften Charakter. Welche Daten tatsächlich über einen Klienten verfügbar sein sollten, konnte zum Zeitpunkt der Implementierung nicht ermittelt werden. Außerdem ist zu vermuten, dass, abhängig vom jeweiligen Einsatzgebiet (z.B. ambulante oder stationäre Dienste), ohnehin eine individuelle Anpassung in Zusammenarbeit mit dem Endnutzer erfolgen müsste. Wichtig ist

zu zeigen, dass Klienteninformationen grundsätzlich verfügbar gemacht werden können. Die orangefarbene Gruppe von Buttons unter den gezeigten Informationen steht für eine mögliche Navigation zu weiteren Funktionen wie der Anzeige von Vitaldatenstatistiken (Verlauf von Blutdruck und Puls), einer Auflistung zukünftiger, den Klienten betreffende Termine oder auch durchgeführte Tätigkeiten, die unter "Vorgänge" zusammengefasst werden. Momentan ist noch keine dieser Funktionen implementiert, jedoch sollte ein Eindruck dessen vermittelt werden, wie ein Zugriff auf über die Stammdaten hinausgehende Informationen ermöglicht werden könnte, ohne den Grundsatz der Übersichtlichkeit der Anwendung zu verletzen.

Nachdem Leistungen auf dem mobilen Gerät dokumentiert wurden, müssen die Daten an CareNet übermittelt werden, damit der Arzt, der die Leistungen in Auftrag gegeben hat, deren Ausführung zeitnah kontrollieren und gegebenenfalls eingreifen kann. Da die dokumentierten Informationen ausschließlich in den Leistungen erfasst wurden und keinerlei Änderungen an den Datenobjekten *Tour* oder *Station* vorgenommen wurden, müssen lediglich die Leistungen wieder an CareNet übertragen werden. Um die Menge der übertragenen Daten gering zu halten, sollen nur die Leistungen übertragen werden, die auch verändert wurden. Hierzu wurde einem Datenobjekt *Service* ein Attribut *EDITED* hinzugefügt, welches einen Wahrheitswert (wahr oder falsch) enthält, um anzuzeigen, ob eine Leistung angepasst wurde oder nicht. Sollte der Zustand einer Leistung logisch hergeleitet werden, müsste ein Vergleich mit dem ursprünglichen Zustand aller Attribute erfolgen. Dies wäre entweder über eine lokale Kopie des "Originals" jeder Leistung möglich, oder aber es müssten alle Leistungen vor der Übertragung noch einmal von Server abgefragt werden, um diese vergleichen zu können. Die erste Variante sorgt jedoch für redundante Speicherbelegung und die zweite Variante erhöht die zu übertragene Datenmenge, was nicht wünschenswert ist.

Wird also eine veränderte Leistung gefunden (*EDITED = true*), muss außerdem überprüft werden, ob diese mit einem Messwert oder einem Wundfoto verbunden ist. Ist die Leistung mit einem Messwert verbunden, muss dieser vor der Aktualisierung der eigentlichen Leistung zu CareNet übertragen werden. War der Vorgang erfolgreich, muss das Datenobjekt in CareNet über ein Dossier sowohl mit der Leistung als auch mit dem entsprechenden Klienten verbunden werden. Das Setzen des Dossiers erfolgt über einen separaten REST-Aufruf, für den die von CareNet erzeugten IDs des Klienten, der Leistung und des Messwertes benötigt werden. Die IDs der ersten beiden Objekte wurden bereits beim Abfragen der Toureninformationen und der Klientendaten von CareNet übertragen, jedoch wird die ID für den lokal erzeugten Messwert erst dann von CareNet generiert, wenn der Wert auch übertragen wurde. Aus diesem Grund muss die ID des Messwertes aus der Antwort auf die Anfrage zur Übertragung zu CareNet extrahiert werden und ebenfalls lokal gespeichert werden. Da die lokale ID bisher die ID der zugehörigen Leistung war und ein ersetzen dieser zu einem Bruch der Fremdschlüsselbeziehung zur Leistung führen würde, wurde dem Datenobjekt *Messwert* ein weiteres Attribut für die von CareNet erzeugte ID hinzugefügt. Die ID wird neben dem Setzen des Dossiers auch für eine eventuelle Löschanfrage benötigt, falls der Nutzer lokal einen bereits übertragenen Messwert löscht, der dann auch in CareNet gelöscht werden muss.

Analog zur Behandlung eines Messwertes muss auch ein aufgenommenes Wundfoto zunächst übertragen werden und anschließend die entsprechenden Dossiers gesetzt werden. Da sich aber die REST-Methoden zur Übertragung unterscheiden, müssen beide Fälle gesondert behandelt werden.

Ein Nutzer von CareNetmobile hat die Möglichkeit, Informationen zu Touren manuell durch betätigen des Buttons "Aktualisiere Touren" (vgl. Abbildung 4.10) zu aktualisieren. Hierbei muss zunächst überprüft werden, ob bereits Dokumentation stattgefunden hat, die durch eine Aktualisierung überschrieben werden könnten. Werden veränderte Leistungen gefunden, die noch nicht übertragen wurden, wird der Nutzer gefragt, ob er die bereits dokumentierten Leistungen verwerfen möchte, oder ob die Daten vor der Aktualisierung zunächst übertragen werden sollen. Entscheidet er sich gegen die Übertragung, werden alle lokalen Datenbanken geleert und die aktuellen Toureninformationen neu geladen, abgespeichert und dargestellt.

5. Evaluation

Im folgenden Kapitel wird die bisher entwickelte Anwendung zur Unterstützung der Dokumentation von delegierten ärztlichen Tätigkeiten evaluiert. In Abschnitt 5.1 wird CareNet_{mobile} zunächst an Qualitätskriterien der ISO/IEC 25000 gemessen, die auf die Software-Produktqualität abzielt. In Abschnitt 5.2 wird das Ergebnis einer Anforderungsevaluation im Rahmen einer Schulung von Angehörigen der Alten- und Krankenpflegeberufe vorgestellt.

5.1. Evaluation der technischen Umsetzung der mobilen Anwendung

Zur Evaluation der entwickelten Anwendung wird auf den von der International Standards Organization (ISO) und der International Electrotechnical Commission (IEC) entwickelten Standard 25000 zurückgegriffen, der den Namen "Software product Quality Requirements and Evaluation (SQuaRE)" trägt [7]. Der Standard ersetzt die beiden älteren Standards ISO/IEC 14598 (Software Product Evaluation) und ISO/IEC 9126 (Software Product Quality), da diese ohnehin komplementären Standards die gleichen normativen und funktionellen Wurzeln bildeten, jedoch Inkonsistenzen in den vorgeschlagenen Lebenszyklen auftraten. Der Standard ISO/IEC 25000 soll Entwicklern, Käufern und unabhängigen Gutachtern von Software-Produkten eine Menge von Kriterien zur Bewertung von Software in verschiedenen Entwicklungsphasen bieten. Von der Anforderungsanalyse über die Implementierung bis hin zum fertigen Software-Produkt werden Methoden vorgeschlagen, mit denen eine Koordination zwischen Messung und Evaluation von definierten Charakteristika vorgenommen werden kann. Definiert wird ein Software-Produkt Lebenszyklus, in welchem die Software-Produktausprägung in drei Phasen erfasst wird. Diese sind die *Qualität während der Benutzung* (Quality in Use), die *externe Qualität* (external Quality) und die *interne Qualität* (internal Quality). Erstere beschreibt die Anforderungen an die Software aus Sicht des Endnutzers beim finalen Endprodukt. Die externe Qualität wird aus den Endnutzeranforderungen abgeleitet und als Ziel für die technische Verifikation und

Validierung der Software genutzt. Die interne Qualität wird wiederum zum Teil aus den Anforderungen an die externe Qualität abgeleitet, definiert werden aber auch die Eigenschaften von Software-Zwischenprodukten auf dem Weg zum Endprodukt. Die internen Qualitätskriterien können auch auf Nebenprodukte der Entwicklung wie die Dokumentation oder Anleitungen angewandt werden. Die Evaluation von CareNet *mobile* erfolgt nur auf Basis der internen und externen Qualitätskriterien, da es während der begrenzten Bearbeitungszeit nicht möglich war, eine marktreife Anwendung zu entwickeln, sondern lediglich einen Prototypen. Außerdem fehlen zu einer vollständigen Evaluation tatsächliche Endnutzer, die für die Beurteilung der Qualität während der Benutzung unerlässlich sind.



Abbildung 5.1.: Interne und externe Charakteristika von Software-Produktqualität

Abbildung 5.1 zeigt die Charakteristika und Subcharakteristika für die Bewertung der internen und externen Qualität. Aufgrund der zeitlichen Rahmenbedingungen kann nicht in größtmöglicher Detailtiefe auf jedes Charakteristikum eingegangen werden, jedoch wird jedes Element dahingehend bewertet, mit welcher Priorität es in der Implementierung von CareNet *mobile* berücksichtigt wurde oder warum diese Gewichtung erfolgte.

Wartbarkeit

Änderbarkeit Die technischen Anforderungen an CareNet *mobile* zielten vor allem auf eine leichte Erweiterbarkeit und Änderbarkeit wegen veränderter Rahmenbedingungen ab. Gefordert wurde ein modularer Aufbau, der in einer Plug-In-Architektur umgesetzt wurde. Eine leichte Änderbarkeit bzgl. Hinzufügen und Entfernen von Plug-Ins wurde folglich adressiert, jedoch sind Veränderungen *innerhalb* eines Plug-Ins oder der Basisanwendung schwieriger umzusetzen, da durch die Verwendung von nur einer JavaScript-Datei pro Modul nur eine verhältnismäßig unübersichtliche Quellcode-

Basis zur Verfügung steht. Durch ausführliche Kommentierung des Quellcodes wurde versucht, die fehlende Struktur auszugleichen.

Analysierbarkeit Die grundsätzliche Architektur ist aufgrund seines modularen Aufbaus gut analysierbar, da die Zusammenhänge und Abhängigkeiten über die Konfigurationsdateien nachvollziehbar sind. Sollen allerdings Aufrufketten von Funktionen verfolgt werden, muss dies in der Regel manuell gemacht werden, da hier keine Unterstützung durch Software-Werkzeuge vergleichbar beispielsweise mit der Unterstützung, die Eclipse [18] bei der Java-Entwicklung bietet, vorhanden ist. Eine Analyse ist zwar immer noch möglich, jedoch ist die Gefahr größer, dass hierbei wegen der fehlenden Automatisierung Fehler passieren.

Stabilität Die Stabilität der Anwendung stand bisher nicht im Fokus der Entwicklung. Zwar wurden Instabilitäten behoben, wenn sie entdeckt wurden, jedoch konnte aufgrund des zeitlichen Rahmens keine ausführliche Analyse bezüglich möglicher instabiler Abläufe durchgeführt werden.

Prüfbarkeit CareNet*mobile* besteht aufgrund seines hybriden Charakters aus einer Menge verschiedener Technologien. Der native Container besteht aus der jeweils zugrunde liegenden Programmiersprache, während die hauptsächliche Funktionalität mit JavaScript implementiert wird, die Darstellung aber vollständig über HTML und CSS läuft. Außerdem bestehen Abhängigkeiten zu Bibliotheken wie jQuery, jQuery mobile und PhoneGap. Theoretisch können in allen Elementen Fehler auftreten, was eine komplett, im Idealfall automatisierte Überprüfung sehr schwierig macht. Zwar gibt es JavaScript Test-Frameworks wie Jasmine [24] oder QUnit (für jQuery) [25], jedoch können hiermit hauptsächlich Fehler im eigenen Quellcode entdeckt werden. Fehler in den extern entwickelten Bibliotheken sind so nur schwer zu identifizieren. Bisher wurden aus Zeitgründen keine automatischen Tests implementiert. Die Überprüfung der grafischen Oberfläche bestehend aus HTML und CSS, die einen signifikanten Anteil der Entwicklungszeit beansprucht hat, kann ohnehin nur manuell durch Blickprüfung erfolgen. Die Prüfbarkeit von CareNet*mobile* ist daher als schwierig einzustufen.

Übertragbarkeit

Anpassbarkeit Die Anpassbarkeit bezieht sich bei CareNet*mobile* vor allem auf die Möglichkeit, die Anwendung auf verschiedenen Plattformen (Android, iOS, Windows Phone etc.) bereitzustellen. Durch die Verwendung von PhoneGap wurde diese Art der Anpassbarkeit implizit mit in die Architektur aufgenommen. Dadurch, dass die Funktionalität von CareNet*mobile* ausschließlich über JavaScript realisiert wurde, was zwischen den Plattformen portierbar ist, wurde nicht nur zum Entwurfszeitpunkt, sondern auch während der Implementierung ein besonderes Augenmerk auf die Erhaltung der von PhoneGap ermöglichten Anpassbarkeit an verschiedene Plattformen gelegt.

Koexistenz Da CareNet*mobile* gekapselt als Anwendung für eine jeweilige Plattform bereitgestellt wird, ist es problemlos mit anderen Anwendungen auf einem Endgerät

kombinierbar. Auch könnten verschiedene Versionen der gleichen Anwendung auf dem Endgerät installiert sein, wenn verschiedene Signaturen der Anwendungen verwendet werden.

Installierbarkeit Zur Installation werden die von der jeweiligen Plattform unterstützen Verfahren verwendet, die in der Regel einfach und intuitiv sind, da sie für eine breite Masse von Nutzern verwendbar sein müssen. Da CareNetmobile aber nicht für den Einsatz in der breiten Masse gedacht ist, sondern gezielt an Kunden weitergegeben wird und somit nur eine verhältnismäßig geringe Anzahl an Installationen durchgeführt werden wird, spielt die Installierbarkeit nur eine untergeordnete Rolle.

Austauschbarkeit Ähnlich einfach wie die Installation einer Anwendung auf einer der gängigen Plattformen lässt sich über einen auf der Plattform vorhandenen Update-Mechanismus oder eine Deinstallation der alten Anwendung mit anschließender Neuinstallation die Anwendung austauschen. Die Entwurfsentscheidung Tablet-PCs zu verwenden implizierte folglich eine Sicherstellung einfacher Installierbarkeit und Austauschbarkeit.

Effizienz

Zeitverhalten Bei CareNetmobile handelt es sich nicht um eine Anwendung, bei der das Zeitverhalten ein entscheidender Faktor ist. Zwar soll durch den Einsatz der Anwendung Zeit gespart werden, jedoch kommt es hier nicht auf Millisekunden an. Die einzige zeitliche Limitierung ist die Geduld des Nutzers während der Bedienung, welche je nach Individuum sehr unterschiedlich ausfallen kann. Solange keine sekundenlangen Verzögerungen beim Laden von Daten oder Ansichten auftreten, welche bisher nicht festgestellt werden konnten, muss das Zeitverhalten nicht näher betrachtet werden.

Ressourcenverbrauch Die bisher betrachteten Tablet-PCs, welche für den Einsatz von CareNetmobile in Frage kommen, sind in der Regel mit Ressourcen für die Nutzung von Multimediainhalten ausgestattet. Aufgrund dessen, dass zur Dokumentation von delegierten ärztlichen Leistungen lediglich einige Datenbankoperationen, die Anzeige von HTML-Inhalten und einige Animationen benötigt werden, sollten die Ressourcen der Zielgeräte ausreichend sein. Da der Speicher der Geräte im Gigabyte-Bereich (1 Gigabyte = 1024 Megabyte) liegt und die Anwendung zum Zeitpunkt der Abgabe weniger als 2 Megabyte groß war, muss dem Ressourcenverbrauch keine besondere Aufmerksamkeit geschenkt werden.

Funktionalität

Angemessenheit Im Vergleich mit den Anforderungen ist die bisher implementierte Funktionalität zwar eine gute Basis, jedoch fehlen noch wichtige Funktionalitäten wie das Hinzufügen durchgeföhrter Tätigkeiten vor Ort beim Klienten oder eine vollständige Klienteninformation. Aus diesem Grund ist die Funktionalität von CareNetmobile noch nicht dem Einsatzziel der Dokumentation ärztlich übertragener Tätigkeiten anzusehen.

Richtigkeit Da noch einige Anforderungen offen sind, die für einen produktiven Einsatz der Software notwendig sind, kann zum jetzigen Zeitpunkt die Richtigkeit der Anwendung nicht überprüft werden.

Interoperabilität CareNetmobile wurde dafür entwickelt, Daten von CareNet abzurufen, veränderbar zu machen und diese zurückzusenden. Zwar wird eine allgemeine REST-Schnittstelle verwendet, jedoch ist die Behandlung der abgefragten Daten stark an das Datenmodell von CareNet angepasst. CareNetmobile kann also nicht ohne Weiteres mit einem anders aufgebauten Basissystem verwendet werden, sondern ist an CareNet gebunden. Allerdings ist es möglich, über eine Konfigurationsdatei verschiedene Instanzen von CareNet anzusteuern (über die Änderung der URL zum Server). Es ist allerdings eine Erweiterung auf andere Produkte der nubedian GmbH denkbar (vgl. [10]), da diese ein ähnliches Datenmodell und die gleiche Schnittstelle verwenden. Eventuell würde hierzu die Entwicklung weiterer Plug-Ins notwendig.

Sicherheit Die Sicherheit in CareNetmobile muss vor allem für die verwendeten persönlichen Daten hergestellt werden. Dies wird lokal auf dem Endgerät über einen Log-In-Mechanismus realisiert, mit dem sichergestellt wird, dass nur autorisierte Personen Zugriff auf die Daten von Klienten und sonstigen Kontakten bekommen. Sobald die Anwendung verlassen wird, sie aber trotzdem noch im Hintergrund läuft und wieder aufgerufen werden kann, werden die Nutzerdaten aus dem lokalen Speicher entfernt, um nicht einem Dritten, der das Endgerät in die Hände bekommt, einen Zugang über die hinterlegten Daten zu ermöglichen. Dadurch, dass CareNetmobile vom Betriebssystem einen eigenen Speicherbereich zugewiesen bekommt, ist ein Zugriff auf gespeicherte Inhalte von anderen Anwendungen aus nicht möglich. Die Übertragung der Daten zu CareNet wird über eine HTTPS-Verbindung gesichert.

Konformität ???

Zuverlässigkeit

Reife Bis zum Ende der Bearbeitungszeit dieser Master-Arbeit konnte lediglich ein Prototyp einer mobilen Anwendung zur Dokumentation ärztlicher Leistungen entwickelt werden. Aus diesem Grund ist die Anwendung noch einige Entwicklungs- und Testschritte von einer marktreifen Anwendung entfernt.

Fehlertoleranz Zur Implementierung einer Fehlertoleranz wurden in erster Linie die von den JavaScript-Frameworks gegebenen Mechanismen benutzt. Beispielsweise werden Datenbankanfragen in Transaktionen gekapselt, damit eine direkte Fehlerbehandlung erfolgen kann, wenn eine Anfrage fehlschlägt. Auch bei AJAX-Anfragen an CareNet wird jeweils überprüft, ob die Anfrage erfolgreich war und wenn nicht, werden entsprechende Maßnahmen eingeleitet.

Um sicherzustellen, dass Daten an CareNet vollständig übermittelt werden, wird jedes Element (z.B. ein Messwert oder ein Dossier) einzeln übertragen, um bei einem Fehler genau zu wissen, welche Operationen noch einmal ausgeführt werden muss. Die Fehlertoleranz ist allerdings noch nicht für alle möglichen Anwendungsfälle implementiert. Es besteht also noch Nachbesserungsbedarf.

Robustheit Zum Zeitpunkt der Abgabe der Arbeit wurden noch keine ausführlichen Tests bezüglich Robustheit der Anwendung durchgeführt. Zwar kann durch die Einbettung der Anwendung in den Kontext der jeweiligen Plattform von einer gewissen Robustheit ausgegangen werden, da hier Mechanismen zum Lebenszyklusmanagement implementiert sind und auch PhoneGap als Basis ständig weiterentwickelt wird, jedoch ist mit Instabilitäten innerhalb der Anwendung aufgrund des Prototypenstatus zu rechnen.

Wiederherstellbarkeit Stürzt die Anwendung oder das gesamte Endgerät ab, sind nicht gleich alle Daten verloren. Eingaben wie die Bestätigung einer Standardleistung (vgl. Kapitel 4.4) oder die Eintragung einer Begründung, warum eine Leistung nicht erbracht werden konnte, werden direkt bei der Eingabe in der lokalen Sqlite-Datenbank gespeichert. Für Wundfotos und Messwerte muss der Speichervorgang explizit angestoßen werden. Da die lokale Datenbank persistent ist, können gespeicherte Daten auch nach einem Neustart des Tablet-PCs oder der Anwendung wiederhergestellt werden. Wird die Anwendung oder der zugehörige Speicherbereich allerdings gelöscht, bevor die Daten an CareNet übertragen wurden, sind sie nicht wiederherstellbar.

Benutzbarkeit

Verständlichkeit, Erlernbarkeit, Bedienbarkeit, Attraktivität Die Benutzbarkeit des Prototyps von CareNet*mobile* wurde in einer Schulung mit einigen möglichen Endnutzern evaluiert. Die Ergebnisse der Evaluation werden in Abschnitt 5.2 vorgestellt.

5.2. Evaluation von Nutzbarkeit, Funktionsumfang und Akzeptanz anhand einer Zielgruppenschulung

Am 19.01.2013, nach etwa drei Monaten Entwicklungszeit, fand eine ganztägige EDV-Anwenderschulung im Rahmen der Ausbildung zur Medizinischen Fachpflegekraft am Forschungszentrum Informatik (FZI) statt. Die sieben Teilnehmerinnen waren ausgebildete Alten- und Krankenpflegerinnen mit mehrjähriger Berufserfahrung. Der erste Teil des Tages wurde vorwiegend zu Schulungszwecken genutzt, während der zweite Teil des Tages der Evaluation der zu Beginn der Arbeit auf Basis von Literatur ermittelten Anforderungen an eine mobile Anwendung diente. Ziel des Schulungsteils der Veranstaltung war einerseits eine grundlegende Einführung in die Nutzung von CareNet, andererseits sollten die Teilnehmerinnen den Umgang mit Tablet-PCs erlernen und für die hiermit gegebenen technischen Möglichkeiten begeistert werden. Im zweiten Teil zur Evaluation wurde der Fokus auf die Ermittlung der momentanen Abläufe bei der Dokumentation pflegerischer Leistungen gelegt, auftretende Probleme sollten identifiziert werden und daraus Anforderungen an die Dokumentation auf einem Tablet-PC abgeleitet werden.

5.2.1. Vorwissen der Teilnehmer und vorbereitende Maßnahmen

Zu Beginn der Schulung musste zunächst der EDV-Wissensstand ermittelt werden, um festzustellen, welche Kenntnisse vorausgesetzt werden können, ohne die Teilnehmer zu überfordern. Ziel war es, zu keinem Zeitpunkt wegen zu schneller Vorgehensweise Teilnehmer zu verlieren oder Langeweile durch zu langsames Vorgehen aufkommen zu lassen. Hierzu wurde im Gespräch mit der Gruppe ermittelt, ob Computer überhaupt Teil des täglichen beruflichen und privaten Lebens seien und, falls ja, welche Einstellung die Teilnehmer gegenüber Computern haben. Ein Stimmungsbild bezüglich der Einstellung ist als Grundlage einer Anwenderschulung besonders wichtig, da so frühzeitig generelle Kritikpunkte an sowie Widerstände gegen EDV identifiziert werden können und gezielt dagegen gearbeitet werden kann. Werden versteckte Widerstände nicht erkannt, kann dies den Erfolg einer Schulung gefährden.

Die Teilnehmer gaben an, sowohl privat als auch beruflich mit Computern zu arbeiten. Hauptsächlich würden das Internet (verschiedene Browser) und Office-Anwendungen benutzt, sowie im Beruf die zur Dokumentation von Tätigkeiten verwendete Software. Neben stationären PCs wurde auch die Verwendung von Mobiltelefonen, Smartphones und Tablet-PCs abgefragt. Mobiltelefone benutzten alle Teilnehmer, ein Smartphone besaßen jedoch nur zwei Teilnehmerinnen und Tablet-PCs wurden lediglich einmal von zwei Teilnehmerinnen verwendet.

Zur Fundierung der in der Diskussion gewonnenen Eindrücke wurde von den Teilnehmern ein Fragebogen zur Ermittlung der Einstellung gegenüber elektronischer Datenverarbeitung auf Basis des Computer Attitude Scale (CAS) [26] ausgefüllt. Da der CAS im Jahre 1986 entwickelt wurde, als die Nutzung von Computer in Unternehmen und vereinzelt auch in Privathaushalten gerade erst begann, konnte nur ein Teil der Fragen verwendet werden. In der Master-Arbeit von Mathias Schmon [27] wurden die Fragen bereits in einer ersten Version übersetzt und überarbeitet, was als Grundlage für den in dieser Arbeit verwendeten Fragebogen diente. Die letztendliche Auswahl der Fragen zielte vor allem darauf ab, wie sich die Teilnehmer bei der Arbeit mit Computern fühlen, ob sie sie beispielsweise als Bedrohung ansehen oder ob Computer eher als nützliches Hilfsmittel angesehen werden. Außerdem wurde gefragt, ob die Teilnehmer meinen, ein grundsätzliches Verständnis für die Arbeit mit Computern zu haben.

Die Ergebnisse des Fragebogens sind in Abbildung 5.2 aufgelistet. Sechs von sieben Teilnehmern sagten aus, dass sie sich bei der Arbeit mit Computern weder unwohl fühlen (1), noch eingeschüchtert werden (2). Hieraus kann geschlossen werden, dass grundsätzlich keine Scheu gegenüber Computern besteht. Jedoch waren sich die Teilnehmer bei der Frage, ob Computer sie verunsichern, nicht einig (drei stimmten zu, vier nicht), was offenbart, dass trotz täglicher Nutzung immer noch gewisse Bedenken vorhanden sind. Dies steht in gewissem Widerspruch zu der Frage, ob Computer verunsichernd wirken (6), da Unsicherheit häufig zu Unwohlsein führt. Auch bei der Frage, ob ein übermäßiger Einsatz von Computern schädigende Wirkung haben könnte (3), herrschte keine Einigkeit. Obwohl vier von sieben Teilnehmern der These nicht zustimmten, kann aus der Diversität der Antworten geschlossen werden, dass eine flächendeckende Einführung von Computern kritisch

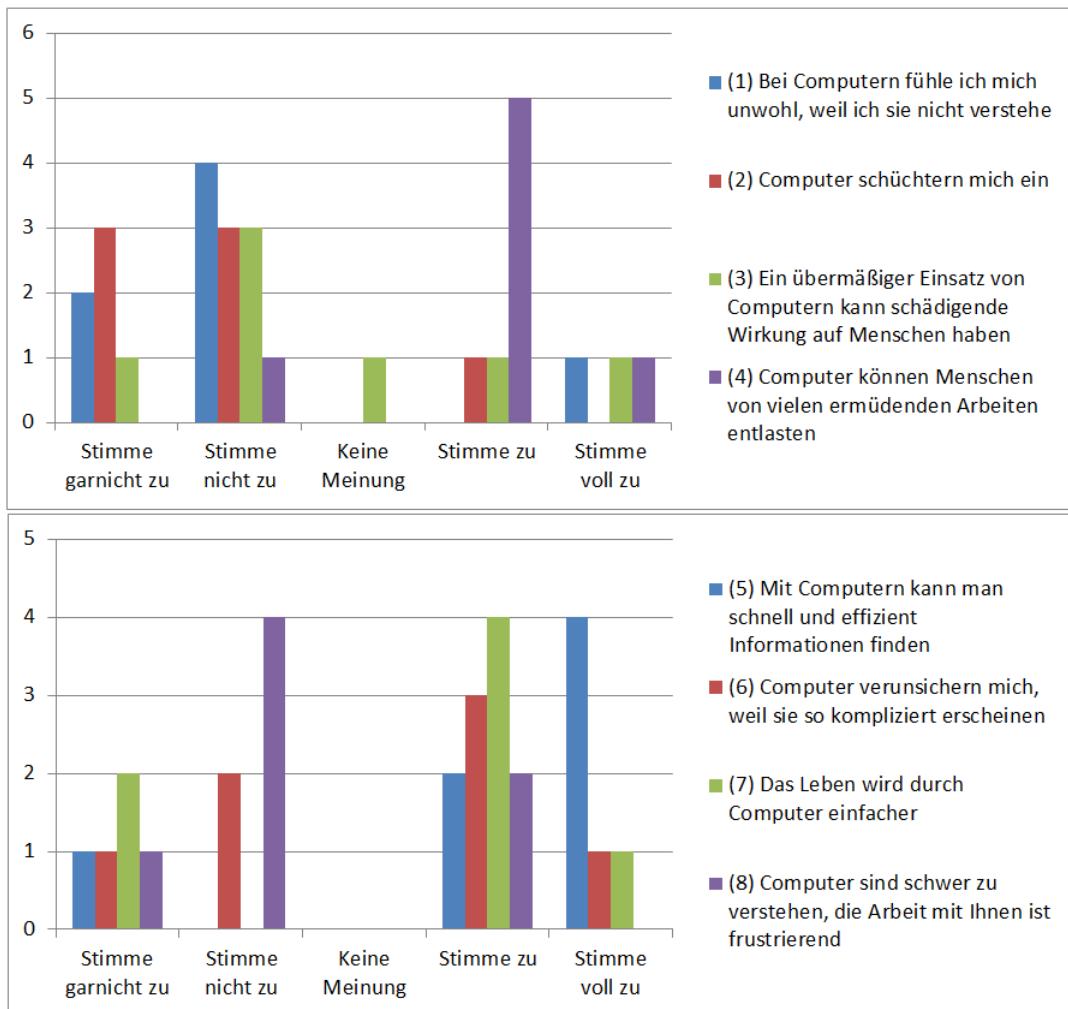


Abbildung 5.2.: Ergebnisse des Fragebogens zur Einstellung gegenüber Computern

gesehen wird. Die Fragen zum Nutzen von Computern bei der Ausführung der täglichen Arbeit fielen grundsätzlich positiv aus. Zwar stimmten jeweils ein bzw. zwei Teilnehmerinnen den Thesen nicht zu, dass Computer ermüdende Aufgaben übernehmen können (4), man mit ihnen schnell Informationen finden kann (5) und sie das Leben grundsätzlich vereinfachen (7), jedoch lässt sich daraus ablesen, dass Computer als Hilfsmittel und nicht als Hürde angesehen werden. Trotz dieser positiven Sicht auf die Verwendung gaben fünf von sieben Teilnehmern an, dass Computer an sich schwer zu verstehen sind und die Arbeit mit ihnen zu Frustration führen kann (8). Dies ist vermutlich auf die Tatsache zurückzuführen, dass Computer eine derartige Komplexität besitzen, dass ein Laie oft nicht in der Lage ist, auftretende Fehler zu beheben. Die Antwort auf die Frage passen zu Berichten der Teilnehmer, dass einer oder mehrere der zur Arbeit vorhandenen Computer häufig ausfällt oder zu sonstigen Problemen führt.

Die Ergebnisse des Fragebogens sowie der in den Gesprächen gewonnene Eindruck deckten sich weitgehend. Für die Schulung wurde aus dem gewonnenen Eindruck geschlossen, dass keine grundsätzlichen Widerstände gegen den Einsatz von Computern überwunden werden müssen, dass jedoch gerade im Hinblick auf die Einführung einer neuen Technologie Überzeugungsarbeit geleistet werden muss. Als Ziel der Schulung bzw. der Anforderungs-

ermittlung sollte die Erleichterung der Arbeit kommuniziert werden, damit nicht der Eindruck entsteht, es sollte nur eine bessere Kontrolle der geleisteten Arbeit möglich werden. Denn wenn die Einführung elektronischer Dokumentation als Überwachung wahrgenommen wird, könnten sich hierdurch Widerstände formieren.

Für die Schulungen in CareCM und dem Umgang mit Tablet-PCs wurden die Teilnehmer in Zweierteams aufgeteilt. Da es sich um eine ungerade Anzahl an Teilnehmern handelte, wurde ein Team mit einem Studenten des FZI ergänzt, der ohnehin aus Interesse an der Schulung teilnahm, jedoch nicht in die Evaluation mit einbezogen wurde. Durch die Teambildung sollte eine gegenseitige Unterstützung der Teilnehmer gefördert werden und eine kommunikativere Arbeitsumgebung geschaffen werden. Durch unterschiedliches Vorwissen konnten sich die Teilnehmer mit ihren Fähigkeiten im Team bei der Mitarbeit im Workshop gegenseitig ergänzen.

Die Schulung für CareCM war vor allem dafür gedacht, den Teilnehmern einen Eindruck von einem möglichen zentralen Verwaltungssystem zu geben. Es wurden verschiedene Funktionalitäten wie das Anlegen von Kontakten und Terminen sowie deren Verknüpfung vermittelt, die Struktur der grafischen Oberfläche erklärt und das Einsatzgebiet von CareCM erläutert. Von den Teilnehmern wurden häufig Vergleiche zu ihren momentan verwendeten Verwaltungssystemen gezogen. Da diese nach Aussagen der Teilnehmer ähnlich aufgebaut sind, kamen die meisten schnell mit der Benutzung von CareCM zurecht. Trotz der Aufgaben (siehe Anhang [REFERENZ]) zur Festigung des gelernten, konnte in der Kürze der Zeit nur ein oberflächlicher Eindruck von CareCM vermittelt werden. Weil der Schulungsteil zu CareCM ohnehin in erster Linie zur Erfüllung der Schulungsrichtlinien zur Weiterbildung zur MFP in das Programm eingefügt wurde, stand nicht mehr Zeit zur Schulung zu Verfügung. Da CareCM nur *ähnlich* zu dem zentralen System CareNet ist, war den Teilnehmern schwer die Sinnhaftigkeit für den Schulungsteil zu vermitteln. Es ist zu vermuten, dass am Ende des Tages kaum noch ein Teilnehmer in der Lage gewesen wäre, die am Vormittag gestellten Aufgaben zu CareCM zu lösen, da der danach durchgeführte Teil der Schulung keinen Bezug hierzu hatte. Wahrscheinlich ist daher, dass dieser Teil der Schulung tatsächlich nur seinen Pflichtcharakter erfüllte.

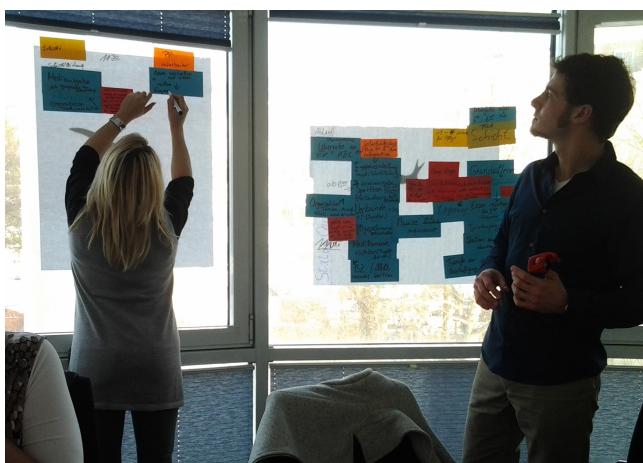
Zur Schulung im Umgang mit Tablet-PCs wurde jedem Zweierteam ein Tablet zur Verfügung gestellt (Samsung Galaxy Tab I oder II). Die Geräte lagen schon zu Beginn der Schulung aus und es konnte beobachtet werden, dass einige der Teilnehmer schon vor dem eigentlichen Schulungsteil die Geräte in Augenschein nahmen. Ziel dieses Teils sollte neben dem reinen Erlernen der Funktionalität sein, die Alten- und Pflegekräfte für die neue Technologie zu begeistern und Neugier zu wecken. Hierfür wurden zunächst allgemeine Eigenschaften eines Tablet-PCs wie die Bedienung durch Berührung, vorhandene Hardware wie eine oder mehrere eingebaute Kameras, den Beschleunigungssensor oder den GPS-Sensor und die damit verbundenen Möglichkeiten erklärt. Es erfolgte eine generelle Abgrenzung von Tablet-PCs zu herkömmlichen Desktop- und Laptop-PCs sowie Erklärungen zu den verschiedenen Betriebssystemen wie Android und dem bekannten iOS

von Apple. Das Interesse der Teilnehmer war spürbar, da die Erklärungen zur Hardware und Ausstattung der Geräte häufig durch Fragen zu Preisen, erhältlichen Varianten und Einsatzmöglichkeiten unterbrochen wurden.

Nach einer grundlegenden Erklärung der Eigenschaften wurden verschiedene Anwendungen (Apps) vorgeführt. Die Teilnehmer waren aufgefordert, schon während der Erklärung selbst das Gesehene auszuprobieren. Vorgestellt wurde der Internet Browser, die Kamera-Funktion, Google Maps, die YouTube-Anwendung und der Google Playstore. Bei der Auswahl der vorgestellten Anwendungen wurde bewusst ein Fokus auf Unterhaltung und tägliche Anwendbarkeit gelegt, da nach Meinung des Autors nur auf diesem spielerischen Wege in der Kürze der Zeit eine Begeisterung für die Geräte hervorgerufen werden konnte. Eine direkte Vorstellung von Anwendungen zur Unterstützung der täglichen Arbeit hätte diese Wirkung nicht erzielen können. Dass das Ziel der Schaffung von Interesse und Begeisterung erreicht wurde, konnte am Verhalten der Teilnehmer abgelesen werden, da bei den anschließend gestellten Aufgaben erkennbar war, dass deutlich mehr mit dem Gerät ausprobiert wurde als gefordert. Die Interaktion im Team und zwischen den Teams sorgte für eine aufgeheiteerte Stimmung im Schulungsraum.

5.2.2. Ermittlung des Status Quo - Ablauf und Schlussfolgerungen

Der zweite Teil der Anwenderschulung sollte in erster Linie dazu genutzt werden, um Wissen über den momentanen Arbeitslauf einer Pflegekraft zu gewinnen und daraus Anforderungen an eine mobile Anwendung zur Dokumentation von erbrachten Tätigkeiten abzuleiten. Der eigentliche Fokus der Master-Arbeit liegt zwar auf der Implementierung eines Systems zur Dokumentation *ärztlich übertragener* Tätigkeiten, jedoch kann für diese Tätigkeiten kein Status Quo ermittelt werden, da die Teilnehmer der Schulung die ersten Alten- und Pflegekräfte sind, die zu einer Medizinische Fachpflegekraft ausgebildet werden. Es konnte jedoch on den Teilnehmerinnen in Erfahrung gebracht werden, dass einige übertragbare Tätigkeiten wie die Blutabnahme oder die Gabe von Medikamenten schon jetzt von examinierten Krankenschwestern im Rahmen der Pflege ausgeführt werden dürfen. Außerdem ähneln sich die zu erwartende Arbeitsstruktur einer MFP und die einer Alten- und Krankenpflegerin dahingehend, dass in einem bestimmten Zeitraum (meist einer Arbeitsschicht), eine Reihe von Klienten mit bestimmten Dienstleistungen versorgt werden müssen.



Durch eine Analyse des momentanen Ablaufes in Zusammenarbeit mit den ausführenden Kräften sollten Stärken und Schwächen mit besonderem Augenmerk auf der Dokumentation identifiziert werden. Zunächst wurde hierzu der Tagesablauf der Pflegekräfte rekonstruiert und in einem Diagramm für bessere Über-

sichtlichkeit festgehalten. Ein Eindruck der Ermittlung des Status Quo kann in Abbildung 5.3 gewonnen werden. Das erstellte Diagramm zeigt prinzipiell eine Sequenz von Tätigkeiten (blau), denen Probleme auf roten Karten zugeordnet wurden. Der genaue Ablauf des Tages ist nicht entscheidend und wird hier nicht reproduziert, sondern lediglich die dadurch gewonnen Erkenntnisse zusammengefasst. Ziel des Diagramms war es herauszufinden, an welchen Stellen in welchem Umfang dokumentiert werden muss, welche Fehlerquellen sich im momentanen Ablauf ergeben und welche Aspekte verbessert werden könnten, um den Ablauf im Bezug auf die Dokumentation zu vereinfachen.

Schnell kristallisierte sich heraus, dass der wichtigste und kritischste Faktor während der Arbeit der Pflegekräfte *Zeit* ist. In der Konsequenz bedeutet dies, dass jede Maßnahme, die mehr Zeit verschafft, positiv zu bewerten ist und jede Verzögerung des Ablaufs zu vermeiden ist.

Eine wichtige Erkenntnis ist, dass es nicht eine Reihe von Tätigkeiten gibt, die während *eines* Besuchs beim Patienten abgearbeitet werden, sondern dass während einer Schicht, zumindest im stationären Bereich, mehrere Besuche beim Patienten mit verschiedenen Aufgaben durchgeführt werden. Alle Tätigkeiten werden dem Personal in einer so genannten *Tagesstruktur* mitgeteilt, die im Laufe der Schicht abgearbeitet werden müssen. Es lassen sich bestimmte Tätigkeitsblöcke identifizieren, die in einer bestimmten zeitlichen Reihenfolge liegen wie z.B. die Grundpflege vor dem Essen reichen, anschließend Transfer der Bewohner zu individuellen Beschäftigungsprogrammen. Daneben gibt es eine Reihe von Tätigkeiten, die nicht zu einem bestimmten Zeitpunkt erledigt werden müssen, sondern jeweils dann, wenn Bedarf besteht. Hierzu gehören beispielsweise die Lagerung der immobilen Bewohner sowie Toilettengänge. Dokumentiert werden muss alles, das entweder für die spätere Abrechnung mit der Krankenkasse benötigt wird oder medizinisch relevant ist. Medizinische Relevanz haben z.B. Toilettengänge, da der Stuhlgang ständig kontrolliert werden muss. Konnte bei einem Patienten für eine gewisse Zeit kein Stuhlgang festgestellt werden, wird hieraus abgeleitet, dass dieser an einer Verstopfung leidet und es werden abführende Medikamente verabreicht.

In einer Schicht sind für ca. 40 Bewohner drei bis vier Pflegekräfte sowie eine Schichtleitung zuständig. Die Pflegekräfte sind in der Regel nur für Pflegetätigkeiten zuständig, während die Schichtleitung, meist eine examinierte Krankenschwester, neben der Pflege der Bewohner außerdem eine Reihe organisatorischer Tätigkeiten sowie Arzt- und Angehörigenkommunikation erledigt. Um dieses Maß an Arbeit zu schaffen, ist jedoch die Anzahl zu pflegender Bewohner auf vier begrenzt, wobei die reinen Pflegekräfte etwa acht Bewohner versorgen.

Nach Aussagen der Teilnehmer ist es wahrscheinlich, dass ausgebildete MFPs nur noch ärztlich übertragene Tätigkeiten durchführen und keine Pflege mehr übernehmen. Dies bedeutet, dass die Durchführung und Dokumentation der übertragenen Tätigkeiten nicht zwischen den Pflegetätigkeiten erfolgt, sondern in einem gesonderten Tagesablauf. Nimmt

man an, dass dieser Tagesablauf an den einer Schichtleitung angelehnt ist, wird auch hier jeder Klient mehrfach besucht, da beispielsweise die Gabe von Medikamenten mehrfach pro Schicht erfolgt.

Die Dokumentation der Tätigkeiten wird direkt elektronisch durchgeführt, d.h. ohne die Ergebnisse auf Papier oder einem anderen Medium übergangsweise festzuhalten. Dokumentiert werden muss mit der dafür vorgesehenen Software, die über drei Computer verwendbar ist. Nur einer davon ist ein Laptop, die Dokumentation muss also in der Regel im Dienstzimmer an den Desktop-PCs erfolgen. Da es nur drei PCs sind, die von vier bis fünf Mitarbeitern inklusive Schichtleitung für die Dokumentation aller Belange von 40 Patienten genutzt werden müssen, kommt es hier zwangsläufig zu Engpässen. Wenn dann auch noch ein oder mehrere PCs defekt sind, verschärft sich die Situation. Die Folge sind unproduktive Wartezeiten, um die notwendige Dokumentation durchführen zu können, die von der Zeit, die für die Durchführung der Tätigkeiten gedacht war, abgeht.

Jede Pflegekraft versorgt in der Regel Patienten, die in nah beieinander liegenden Räumen untergebracht sind. So sollen die Wege zwischen den Zimmern kurz gehalten werden und somit Zeit eingespart werden. Da alle Pflegekräfte von den ihnen zugewiesenen Zimmern häufig zur Dokumentation ins Dienstzimmer müssen, wäre es wünschenswert, dass auch hier die Wege kurz gehalten werden. Bei 40 Bewohnern auf einer Etage ist es jedoch nicht möglich, allen Mitarbeitern Zimmer direkt neben oder zumindest in unmittelbarer Nähe des Computers zur Dokumentation zuzuweisen. Einige Mitarbeiter müssen also jedes Mal relativ lange Wege auf sich nehmen. Eigentlich sind die Pflegekräfte angehalten, jede Tätigkeit unverzüglich zu dokumentieren, aber aufgrund der zurückzulegenden Wege werden meist mehrere Tätigkeiten durchgeführt, bevor die Dokumentation erfolgt, um die Wegezeit zu sparen. Zwischen Durchführung und dem Einpflegen des gemachten in das Computersystem werden alle Tätigkeiten also zunächst im Kopf "dokumentiert". Besonders morgens, bei der Durchführung der Grundpflege, die für acht Bewohner in etwa zwei Stunden Zeit in Anspruch nehmen darf, haben die Mitarbeiter keine Kapazitäten mehr frei, um die Dokumentation nach jedem Bewohner durchzuführen. Es kommen so enorm viele Informationen über Tätigkeiten zusammen, die sich eine Pflegekraft merken muss. Hier kann es leicht passieren, dass Tätigkeiten, die nicht routinemäßig durchgeführt werden oder sonstige Besonderheiten vergessen oder falsch dokumentiert werden. Dies kann einerseits wirtschaftliche Konsequenzen haben, wenn Fehler bei abzurechnenden Tätigkeiten gemacht werden, andererseits aber auch zu gesundheitlichen Schäden führen, wenn hier relevante Informationen vergessen werden.

Neben dem reinen Zeitverlust durch die Wege zum Computer und zurück treten häufig "Störfaktoren" auf dem Weg aus, die zu weiterem Zeitverlust führen. Denn sobald eine Pflegekraft ein Zimmer verlässt, kommt es häufig vor, dass sie von anderen Bewohnern um etwas gebeten wird oder von Angehörigen in ein Gespräch verwickelt wird. Zwar sind dies Dinge, die neben der reinen Dienstleistung ein wichtiger Bestandteil des Alten- und Pflegeberufs ist, da dies die menschliche Ebene adressiert, jedoch bedeuten derartige Gespräche und Zusatzleistungen rein rational gesehen einen kritischen Zeitverlust. Müssten weniger Wege ins Dienstzimmer angetreten werden, könnte das Potential für Ablenkungen in Zusammenhang mit der Dokumentation gesenkt werden und somit effektiv Zeit einge-

spart werden.

Neben den genannten Problemen ist die Dokumentation an sich nach Aussage der Schulungsteilnehmer relativ aufwendig, da häufig nicht nur eine Tätigkeit bestätigt werden muss, sondern zusätzlich Daten eingepflegt werden müssen. Eine effizientere Gestaltung wäre auch hier zur Einsparung der Dokumentationszeit wünschenswert.

Im Zusammenhang mit Arztvisiten wurde als Problem genannt, dass ein Arzt die Pflegekraft häufig nach der aktuellen Medikation oder sonstigen den Bewohner betreffenden Informationen fragt. Da bei acht zu betreuenden Bewohnern nicht alle Informationen im Kopf behalten werden können und gerade bei Medikamenten exakte Informationen verfügbar sein sollten, muss auch hier ein Zusatzweg in das Dienstzimmer in Kauf genommen werden, um die Informationen am zentralen Computer abzurufen.

Werden bei einer Arztvisite neue Medikamente verschrieben, Dosierungen geändert oder bestehende abgesetzt, wird dies häufig zunächst handschriftlich direkt beim Bewohner dokumentiert. Handschriftliche Dokumentation bei derartig wichtigen Werten wie Menge oder Name eines Medikaments kann schlimme Folgen haben, wenn die Anweisungen nicht gelesen werden können oder missverstanden werden. Wegen Zeitdrucks entstehen so vermeidbare Fehler. Eine andere Möglichkeit der Dokumentation, bei der Kommunikationsfehler durch unleserliche Schrift vermieden werden kann, ist hier sinnvoll.

Zur Aufgabe der Pflegekräfte und auch der MFPs gehört die Wunddokumentation. Momentan läuft es so ab, dass mit einer separat mitgeführten Kamera die Wunden fotografiert werden und die Bilder später am Computer der Akte des Patienten hinzugefügt werden. Zur Identifikation wird neben die Wunde ein Papierstreifen mit Name, Datum und der Größe der Wunde gelegt. So kann zumindest, wenn das Bild angeschaut wird, kein Zuordnungsfehler passieren. Um die Bilddatei der Patientenakte hinzufügen zu können, muss allerdings die Datei direkt ausgewählt werden. Hierzu muss sich der Mitarbeiter die Nummer des entsprechenden Bildes merken und diese korrekt auswählen. Klickt er beim Auswahlvorgang daneben oder wählt aus einem anderen Grund das falsche Bild aus, kommt es zu einem Durcheinander bei den Wundfotos. Laut Teilnehmern der Schulung kommt dies zwar selten vor, jedoch reicht die Möglichkeit der Verwechslung aus, um Verbesserungspotential zu bieten.

Im Falle der Messung von Vitaldaten wie Blutdruck und Puls müssen genaue Werte festgehalten werden. Aus diesem Grund werden die Werte vor Ort beim Bewohner auf Papier festgehalten und anschließend am Computer übertragen. Auch hier kann es ein Problem mit unsleserlichen Werten geben oder es werden schlichtweg Werte durch Vertippen oder zu großer Hektik bei der Übertragung falsch in das System eingepflegt. Zur Verbesserung dieses Vorgangs müsste das Zwischenmedium abgeschafft werden oder sogar vollständig von der manuellen Dokumentation abgesehen werden.

Der bisher beschriebene Status Quo der Dokumentation von Pflegeleistungen und vereinzelt auch ärztlich übertragbaren Tätigkeiten hat einige Punkte mit Verbesserungspotential offenbart. Die Befragung hat die Annahme bestätigt, dass eine Untersuchung sinnvoll ist, in wie fern eine stärkere Unterstützung durch IT den Dokumentationsvorgang effizienter

Identifiziertes Problem	Kritischer Faktor
Häufiger Wechsel zwischen Patientenzimmer und Dienstzimmer zwecks Dokumentation	Zeit
Drei Computer zur Dokumentation bei 4 - 5 Mitarbeitern führen zu Engpässen und Wartezeiten	Zeit
Auf dem Weg von Dienstzimmer zum Patient und zurück treten häufig "Störfaktoren" auf (Gespräche mit Angehörigen, Fragen durch Patienten, Telefon)	Zeit
Während der Arztvisite ist das Bereitstellen von Informationen zum Patienten wie die aktuelle Medikation meist mit einem Gang ins Dienstzimmer verbunden, da die Pflegekräfte nicht alle Daten im Kopf haben können und die Daten auch nicht vor Ort im Zimmer abgerufen werden können.	Zeit
Die Dokumentation dauert wegen fehlender Autovervollständigung oder vordefinierten Antworten relativ lange.	Zeit
Es werden nicht für jeden Bewohner direkt die durchgeführten Tätigkeiten dokumentiert, da dies zusätzliche Wege zum Computer und eventuelle Wartezeiten bedeuten würde. Die Dokumentation erfolgt also nicht zeitnah und für mehrere Bewohner auf einmal. Da die Informationen auch nicht übergangsweise auf Papier festgehalten werden (außer die Viltaldata-Messwerte), können hier bei der Dokumentation leicht Fehler durch vergessene oder falsche Informationen auftreten.	Qualität
Wundfotos könnten zur Patientenakte falsch zugeordnet werden, da sie mit einer externen Kamera aufgenommen wurden und erst später manuell der Akte hinzugefügt werden.	Qualität
Übertragungsfehler bei auf Papier zwischennotierten Messwerten	Qualität
Anordnungen des Arztes wie beispielsweise eine geänderte Medikation werden oft vor Ort beim Patienten von Hand dokumentiert, um diese später in das Computersystem einzupflegen. Wegen Unleserlichkeit können hier Fehler mit gesundheitsschädlichen oder sogar lebensbedrohlichen Folgen passieren.	Qualität

Tabelle 5.1.: Liste der identifizierten Probleme bei der Dokumentation von Pflegeleistungen in der Residenz Bad Friedrichshall

und qualitativ hochwertiger machen kann. In Tabelle 5.1 werden die identifizierten Probleme noch einmal kurz zusammengefasst und die damit verbundenen kritischen Faktoren benannt.

5.2.3. Anforderungen an CareNetmobile aus Sicht der Zielnutzergruppe

Im folgenden Abschnitt werden die in der Diskussion des Status Quo identifizierten Anforderungen an eine mobile Lösung zur Unterstützung der Dokumentation aufgelistet und bewertet. Die Bewertung erfolgt danach, ob eine Anforderung bereits vorher antizipiert wurde und falls ja, wie sie umgesetzt wurde. Falls es sich um eine neue Anforderung handelt, wird überprüft, ob sie überhaupt umsetzbar ist und ob die Umsetzung für den Erfolg der Arbeit notwendig ist. Bei der Erhebung der Anforderungen muss darauf hingewiesen werden, dass den Teilnehmern vorher die Rahmenbedingungen wie die Nutzung eines Tablet-PCs und die damit verbundenen Möglichkeiten genannt wurden. Aus diesem Grund wurden manche Anforderungen genannt, die nicht zwingend aus dem momentanen Tagesablauf hervorgehen, sondern mehr genannt wurden, weil es möglich gemacht werden kann. Es bestimmte also teilweise das Angebot die Nachfrage und nicht umgekehrt.

1. *Die Anwendung muss übersichtlich strukturiert und intuitiv bedienbar sein.*

Neben den fachlichen Anforderungen an die Anwendung wurde vor allem auf eine intuitive Bedienung und Übersichtlichkeit geachtet, da dies als kritischer Faktor für die Akzeptanz der Anwendung angesehen wird. Mit der in Abschnitt 4.1 beschriebenen grafischen Oberfläche soll den meist wenig technik-affinen Nutzern das Erlernen der Bedienung erleichtert werden.

2. *Die Tagesstruktur muss komplett pro Bewohner abrufbar sein.*

Unter *Tagesstruktur* verstanden die Teilnehmer der Schulung einen Plan, der alle zu versorgenden Patienten und die auszuführenden Tätigkeiten auflistet. Bisher wurde angenommen, dass die MFPs eine Art Tourenplan bekommen, nachdem sie arbeiten. Es wurde also bisher vor allem von einer zumindest teilweisen ambulanten Erbringung von Dienstleistungen ausgegangen. Da die Konzepte einer Tagesstruktur und eines Tourenplans aber prinzipiell gleich sind, muss nur entsprechend dem Kontext (stationäre oder ambulante Dienste) die Namensgebung angepasst werden, um den Nutzern möglichst viele bekannte Elemente in der neuen Technologie zu bieten.

3. *Die Dokumentation ausgeführter Tätigkeiten muss im Zimmer/vor Ort möglich sein.*

Diese Anforderung wurde implizit dadurch erfüllt, dass in der Aufgabenstellung die Nutzung von Tablet-PCs gefordert war. Durch die mobile Verwendbarkeit ist eine Dokumentation im Zimmer möglich.

4. *Die Patientenakte sollte auf dem Endgerät vollständig abrufbar sein.*

Dass Informationen zum Patienten abrufbar sein müssen, wurde zu Beginn angenommen. Neben den Grundinformationen zum Finden des Patienten wie

Name und Adresse sind auch Informationen zu Angehörigen, zuständigen Ärzten und aktueller Medikation gewünscht. Da CareNet diese Informationen bereits anbietet, wurde zumindest für die Stammdaten (Name, Adresse, Kontaktmöglichkeit) eine Repräsentation in den Plug-Ins "Kontakte" und Touren vorbereitet. Eine vollständige Implementierung aller möglichen Daten erfolgt im Rahmen dieser Arbeit nicht, da eine Vorbereitung zum Einfügen weiterer Daten das Ziel grundsätzlich erfüllt.

5. Die Patientenakte sollte vor Ort beim Patienten anpassbar sein.

Von den Teilnehmern wurde gewünscht, dass die Daten des Patienten auf dem Tablet-PC editierbar sein sollten. Da dies aber auch in CareNet möglich ist und anzunehmen ist, dass nicht bei jedem Besuch beim Patienten die Daten angepasst werden müssen, wurde entschieden, dass eine Implementierung dieser Anforderung zum jetzigen Zeitpunkt keinen Mehrwert bringt und deswegen nicht umgesetzt wird. Sollte die Anwendung später tatsächlich zum Einsatz kommen, ist die Umsetzung einer Editierfunktion aber sicherlich zu empfehlen, da sie den Nutzungskomfort steigert.

6. Die Patientenakte sollte Bilder vom Patienten enthalten, um Verwechslungen bei gleichem Namen (z.B. bei einem Ehepaar) im gleichen Zimmer zu vermeiden.

Die Verwendung von Bildern in der Patientenakte und in der Tagesstruktur wurde zu Beginn der Arbeit bereits diskutiert. Die Anforderung wurde allerdings verworfen, da Bilder der Patienten zusätzliche Anforderungen an den Datenschutz in CareNet gestellt hätten, was in Anbetracht der Bearbeitungszeit der Master-Arbeit nicht zu realisieren gewesen wäre.

7. Möglichkeit bei jeder Tätigkeit anzugeben, dass diese nicht ausgeführt wurde und warum.

Die Teilnehmer der Schulung gaben an, dass es immer wieder sein kann, dass aus irgend einem Grund eine geforderte Leistung nicht erbracht werden kann. Damit die Aufgaben des Tages trotzdem als erfüllt angesehen werden können, muss eine Tätigkeit mit Begründung als "nicht erbracht" markiert werden können.

8. Zeitlich nicht planbare und plötzlich zu erbringende Tätigkeiten wie die Hilfe beim Toilettengang müssen der Tagesstruktur am Tablet-PC hinzufügbar sein.

Die Ergänzung von Dienstleistungen, die von der geplanten Tagesstruktur abweichen, wurde bereits im ersten Anforderungskatalog aufgenommen, da angenommen wurde, dass die Arbeit mit Menschen immer unvorhergesehene Tätigkeiten nötig machen kann. Aus Zeitgründen konnte diese wichtige Funktionalität allerdings nicht mehr im Rahmen dieser Arbeit implementiert werden.

9. Vitaldaten (Blutdruck, Puls, Schmerzmanagement, Gewicht) müssen am Tablet-PC erfassbar sein.

Die Anforderung der Erfassung von Vitaldaten musste bereits im VitaBit Projekt umgesetzt werden und wurde aus diesem Grund auch in den Anforderungskatalog von CareNetmobile aufgenommen. Allerdings beschränken sich die erfassbaren Daten bisher auf Blutdruck und Puls. Eine Erfassung von Gewicht und ein Schmerzmanagement ist jedoch möglich, wird aber im Rahmen dieser Arbeit nicht mehr implementiert.

10. *Statistiken über alle Vitaldaten wären hilfreich (Bsp: Verlauf des Gewichts seit Einlieferung).*

Eine genaue Zustandsänderung bei einer Person lässt sich meist nur mit Hilfe von Vergleichswerten vergangener Erhebungen identifizieren. Aus dieser Annahme wurde bei der initialen Anforderungsevaluation abgeleitet, dass es hilfreich sein könnte, vorangegangene Messungen von Vitalparametern in einer übersichtlichen Statistik abfragen zu können. Die Implementierung einer solchen Funktionalität war zunächst geplant, musste jedoch wegen der Kürze der Zeit aus der Agenda genommen werden. Für einen produktiven Einsatz von CareNetmobile wird diese Funktionalität allerdings als notwendig angesehen, um einen klaren Vorteil gegenüber manueller Dokumentation und Auswertung der erhobenen Daten zu bieten.

11. *Automatische Plausibilitätsprüfung von eingegebenen Messwerten (Z.B. Puls > 250 ⇒ Warnung).*

Automatische Plausibilitätsprüfungen sind eine zentrale Möglichkeit zur Steigerung der Qualität bei EDV-gestützter Dokumentation. Diese Art der Überprüfung wurde von Anfang an als wünschenswert angesehen, jedoch ist dies ein Merkmal, das erst implementiert werden kann, wenn grundsätzlich die Dokumentation funktioniert. Da die Umsetzung der gesamten Architektur und der Basisfunktionalitäten schon die gesamte Bearbeitungszeit in Anspruch genommen hat, wurde eine Plausibilitätsprüfung nur in den Anforderungen für den Fall einer Weiterführung des Projektes aufgenommen. Je nach Komplexität der Werte ist die Implementierung zwar sehr aufwendig, aber möglich.

12. *Warnungen, wenn gemessene Werte kritisch sind oder wenn dokumentierte Tätigkeiten auf mögliche Gefahren für den Bewohner/Patienten hindeuten (beispielsweise, wenn für einen Patienten schon länger kein Stuhlgang mehr dokumentiert wurde).*

Diese Anforderung ist ähnlich zu bewerten wie die Plausibilitätsprüfung, da sie prinzipiell die gleichen Voraussetzungen haben. Auch hier wird eine Logik über die grundlegende Dokumentation hinaus benötigt, welche erst zu einem sehr "reifen" Zeitpunkt der Entwicklung implementiert werden kann. Dieser konnte während dieser Arbeit noch nicht erreicht werden. Logik, die aus den dokumentierten Daten Schlüsse zieht, sollte unbedingtes Ziel der Entwicklung sein.

13. *Wundfotos sollten mit der internen Kamera des Tablet-PCs aufnehmbar sein und direkt dem Patienten zugeordnet werden.*

Diese Anforderung wurde nicht aktiv von den Schulungsteilnehmern geäußert, sondern als mögliche Funktionalität aufgrund der Eigenschaften eines Tablet-PCs vorgeschlagen. Der Vorschlag wurde positiv aufgenommen und als möglicherweise komfortabel und nützlich bewertet. Einerseits aus dem Grund, weil keine zusätzlich Kamera mehr mitgeführt werden muss, und andererseits weil der Arbeitsschritt der manuellen Zuordnung wegfällt und somit Zeit gespart wird. Daneben fällt auch die Fehlerquelle der manuellen Zuordnung der Bilder zur Patientenakte weg. Diese Funktion wurde in der aktuellen Version der Anwendung implementiert.

14. *Die Pflegestandards des Hauses sollten auf dem Tablet-PC abrufbar sein.*

Diese Anforderung hat rein informativen Charakter und soll als Hilfestellung dienen, wenn unklar ist, wie in einem bestimmten Fall vorgegangen werden soll. Da sich die Teilnehmer über die Notwendigkeit dieser Funktion nicht einig waren, da die meisten Pflegekräfte bereits über langjährige Erfahrung verfügen und die Vorgänge ohnehin kennen, wurde die Anforderung nicht aufgenommen. Außerdem liefert sie keinen Mehrwert zur Dokumentation, welche das Kernthema der Evaluation ist.

15. *Spracherkennung zur Eingabe von Freitext.*

Eine Spracherkennung zur Eingabe von Freitext wurde häufig gewünscht, da die Eingabe von Freitext über die virtuelle Tastatur als zu langwierig angesehen wurde. Zwar bieten die neueren Betriebssysteme von Android und iOS die Möglichkeit der Spracheingabe, jedoch ist hierzu eine Internetverbindung notwendig, was nicht immer gewährleisten werden kann. Der initiale Anforderungskatalog enthielt die Anforderung, dass CareNet mobile auch ohne Internetverbindung funktionieren muss, was mit einer internetabhängigen Spracherkennung nicht zu vereinbaren ist. Die Anforderung wurde folglich nicht aufgenommen.

16. *Abrufen des Tropfenplans auf dem Tablet-PC.*

Der Tropfenplan ist eine Auflistung aller benötigten flüssigen Medikamente einer Schicht. Er wird dazu verwendet, die Medikamente in einem Arbeitsschritt an der Medikamentenausgabe zusammenzustellen. Prinzipiell könnte diese Anforderung mit vertretbarem Aufwand implementiert werden, jedoch fehlt hierfür die Voraussetzung in CareNet, da die flüssigen Medikamente nicht gesondert in der Patientenakte verfügbar sind. Weil der Tropfenplan keine Relevanz für die Dokumentation hat, sondern nur für die Vorbereitung der Tätigkeiten benötigt wird, wird die Anforderung zwar in die möglicherweise später umzusetzenden Liste von Anforderungen aufgenommen, in dieser Arbeit aber nicht weiter beachtet.

17. *Notrufknopf zum Absetzen eines Telefonanrufs für die ambulante Dienste.*

In einem vorangegangenen Projekt wurde ein Notrufknopf zum schnellen Absetzen von Notrufen eingebaut. Auf die Frage, ob dies auch eine wünschens-

werte Funktionalität für CareNetmobile sei, sagten die Teilnehmer, dass dies eigentlich nur im ambulanten Bereich sinnvoll wäre, da in der Residenz ohnehin Telefone schnell verfügbar wären. Da der Fokus momentan auf einem Einsatz im stationären Bereich liegt, wird kein Notrufknopf implementiert. Außerdem setzt das voraus, dass der Tablet-PC Telefonanrufe absetzen kann. Das verteuert die Hardware und macht weitere wirtschaftliche Untersuchungen erforderlich.

18. *"Rote Liste" auf Tablet-PC abrufbar.*

Die "Rote Liste" stellt eine Sammlung aller zugelassenen Medikamente dar, in der Wirkung und Wechselwirkungen mit anderen Medikamenten beschrieben werden. Einige Teilnehmer wünschten sich, dass diese Liste auf dem Tablet-PC abrufbar sei. Da es sich hierbei aber eigentlich um ein Buch eines Anbieters handelt, dürfen die Inhalte des Buchs nicht ohne Lizenzzahlungen in der Anwendung verwendet werden. Zudem ist unklar, ob die Verwendung überhaupt erlaubt würde, da die Implementierung einer App zum Zugriff auf die Inhalte eigentlich Aufgabe des Verlages wäre. Da diese Funktionalität auch keine Relevanz für die Dokumentation hat, wird sie nicht in den Anforderungskatalog aufgenommen.

19. *Tablet-PCs mit Glasoberfläche, um diese leicht reinigen zu können.*

Hygiene ist im medizinischen und pflegerischen Bereich ein wichtiger Aspekt. Da die Tablet-PCs mit in die Zimmer oder Wohnungen der Bewohner/Klienten genommen werden, müssen diese nach jedem Besuch desinfiziert werden. Die Oberfläche sollte deswegen aus Glas sein, da die Desinfektionsmittel eine Kunststoffoberfläche bei häufigem Kontakt möglicherweise schädigen. Die Anforderung war zu Beginn der Arbeit zwar nicht bekannt, wurde aber erfüllt, da die Implementierung testweise auf Samsung Galaxy Tabs I und II erfolgte, welche eine Glasoberfläche besitzen. Die Anforderung wurde in den Gesamtkatalog aufgenommen.

5.2.4. Ergebnisse des Abschlussfragebogens

Nachdem die Anforderungen an CareNetmobile mit den Teilnehmern evaluiert wurden, wurde der Stand der Entwicklung der Anwendung vorgestellt. Fertiggestellt war zu diesem Zeitpunkt die Einstiegsseite mit der Übersicht über die verfügbaren Plug-Ins, das Plug-In Kontakte und im Plug-In Touren konnte bereits eine einfache Tagestruktur abgerufen werden, sprich pro Bewohner Tätigkeiten, die nur Bestätigt werden müssen. Zudem konnte jede Tätigkeit mit Begründung als "nicht durchgeführt" markiert werden. Die Dokumentation von Wunden per Foto und die Erfassung von Vitaldaten war noch nicht möglich. Es konnte jedoch zu jedem Patienten eine erste Version der Patientenakte abgerufen werden. Nach der Vorführung, bei der jedes Team auf dem eigenen Tablet-PC die Anwendung ausprobieren konnte, wurde ein Fragebogen verteilt, mit dem die Benutzbarkeit, der Lernaufwand zur Nutzung der Anwendung und der erwartete Nutzen durch den Einsatz bei

der täglichen Arbeit abgefragt wurde. Der verwendete Fragebogen ist im Anhang ([[REFERENZ](#)]) zu finden. Als Basis für die verwendeten Fragen wurde auf den bereits validierten Fragebogen des UTAUT-Modells [28] zurückgegriffen. UTAUT steht für *Unified Theory of Acceptance and Use of Technology* und stellt eine Rahmenwerk zur Bewertung des Nutzens und der Akzeptanz durch den Endnutzer dar. Da die Fragen im Original auf englisch sind, mussten diese zunächst ins Deutsche überführt werden. [27, S. 50 ff.] hat dies zur Evaluation der Software CareCM der nubedian GmbH, auf der CareNet aufbaut, bereits getan. Die hier verwendeten Fragen sind an diese Übersetzung angelehnt und wurden an die Belange der Bewertung von CareNetmobile angepasst.

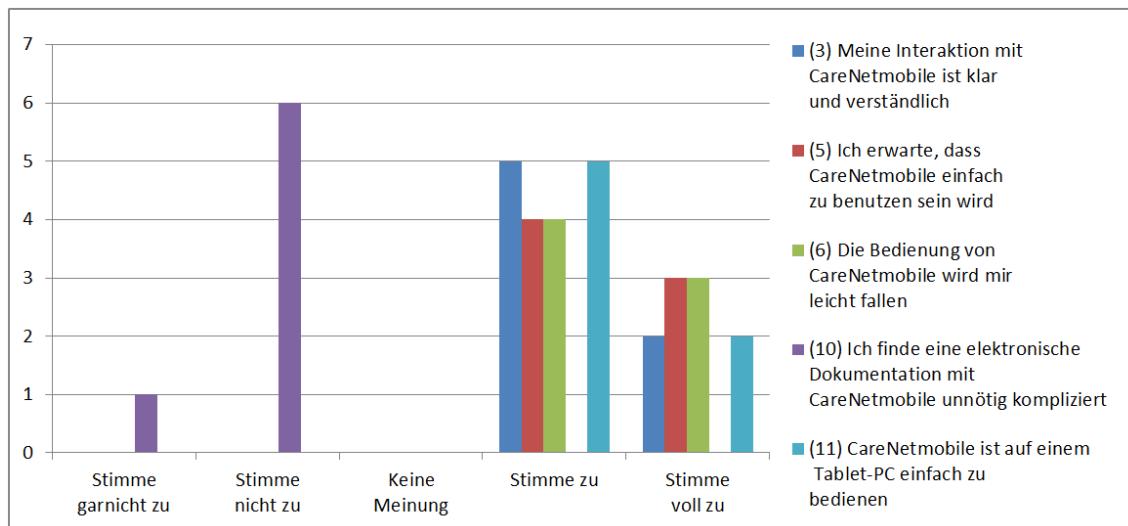


Abbildung 5.4.: Ergebnisse der Fragen zur Benutzbarkeit von CareNetmobile .

Die Ergebnisse der Fragen zur Benutzbarkeit von CareNetmobile sind in Abbildung 5.4 aufgeführt. Den Fragen, ob die Anwendung leicht zu bedienen sein wird (5, 6 und 11), ist von allen Teilnehmern zugestimmt worden. Der Inhalt der drei Fragen überschneidet sich aus dem Grund, um festzustellen, ob die Teilnehmer nur zufällig angekreuzt haben. Da alle Fragen von allen Teilnehmern mit gleicher positiver Tendenz angekreuzt wurden, ist anzunehmen, dass die Fragen verstanden und konsistent beantwortet wurden. Auch der Frage, ob die Interaktion mit CareNetmobile (3) verständlich ist, wurde zugestimmt. Dass "nur" fünf von sieben Teilnehmern zugestimmt und nicht voll zugestimmt, lässt darauf schließen, dass gewisse Anpassungen der grafischen Oberfläche notwendig sind, um eine volle Zufriedenheit zu erreichen. Die Gegenfrage, ob eine elektronische Dokumentation mit CareNetmobile unnötig kompliziert sei, wurde konsequenter Weise im Vergleich zu den anderen Antworten verneint. Bei dieser Frage stimmte nur eine Teilnehmerin voll zu, was vermutlich auf den im ersten Fragebogen (vgl. Abbildung 5.2) festgestellten Respekt gegenüber einer neuen Technologie zurückzuführen ist.

Abbildung 5.5 zeigt das Ergebnis der Fragen zum erwarteten Lernaufwand zur Nutzung von CareNetmobile . Die Fragen, ob die Teilnehmer selbst den Umgang mit der Anwendung leicht lernen können (4) und wie sie dies bei anderen einschätzen (13), wurde positiv beantwortet. Die Angabe, dass sechs von sieben Teilnehmern meinen, das grundsätzliche Wissen zu haben, um CareNetmobile zu benutzen (7), spricht dafür, dass die Bedienung der

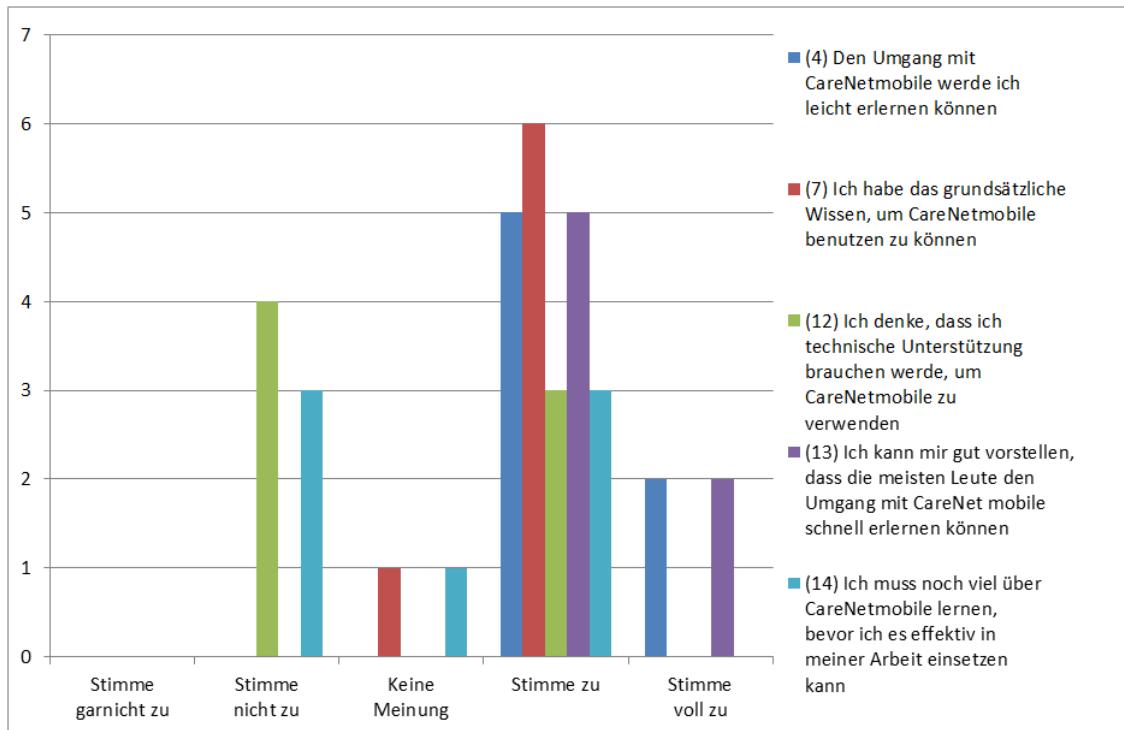


Abbildung 5.5.: Ergebnisse der Fragen zum benötigten Vorwissen und Lernaufwand zur Nutzung von CareNetmobile .

Anwendung als ähnlich intuitiv angesehen wird wie die Bedienung sonstiger Anwendungen des Tablets, da bisher keine spezielle CareNetmobile Schulung erfolgt ist. Bei der Frage nach zusätzlicher technischer Unterstützung (12) waren sich die Teilnehmer nicht einig. Ähnlich unentschieden fielen die Antworten bei der These aus, dass die Teilnehmer vor einem effektiven Einsatz noch viel zusätzlich lernen müssten (14). Es ist zu vermuten, dass eine Aussage, dass keine technische Unterstützung mehr nötig wäre, eher ein Ausdruck von technischem Selbstvertrauen ist als das tatsächlich vorhandene Wissen CareNetmobile zu verwenden. Erwartet worden war hier, dass eine deutliche Mehrheit der Teilnehmer ausstehenden Schulungsbedarf sieht und Unterstützung für Notwendig hält. Dies wurde erwartet, da erstens keine echte Schulung statt gefunden hat und außerdem nur ein Teil der geforderten Funktionalität implementiert war.

Die Fragen zu Nutzen und Akzeptanz von CareNetmobile sind sehr positiv ausgefallen. Alle Teilnehmerinnen sagten aus, dass die Anwendung grundsätzlich hilfreich sein könnte (1) und dass durch sie die Arbeit effizienter erledigt werden und somit Zeit eingespart werden könnte (2). Die Teilnehmerinnen sagten sogar durchweg aus, dass sie die Anwendung gerne regelmäßig nutzen würden, was ein sehr gutes Ergebnis des Schulungstages ist, da diese Aussage dafür spricht, dass eventuell vorhandene Widerstände und Vorbehalte, die zu Beginn identifiziert wurden (vgl. 5.2.1, erfolgreich überwunden wurden. Obwohl zunächst gewisse Ängste aufgrund der geringen Technikaffinität gegenüber der neuen Technologie bestanden, lehnte keine Teilnehmerin die Nutzung grundsätzlich ab.

Lediglich die Frage nach der Freiwilligkeit der Verwendung von CareNetmobile wurde aus der Bewertung des Fragebogens herausgenommen. Eigentlich war diese Frage dafür gedacht

herauszufinden, ob die Alten- und Krankenpfleger durch Rückmeldung einen Einfluss auf die Art der Dokumentation haben, jedoch zeigen die widersprüchlichen Antworten, dass keine Einigkeit herrscht, obwohl die meisten Teilnehmer in der gleichen Einrichtung mit den gleichen Vorgesetzten arbeiten. Aus dem Ergebnis wurde geschlossen, dass die Frage missverständlich gestellt war und gegen die gängige Organisationsstruktur in den Einrichtungen sprach, die meist hierarchisch geprägt ist. Aus diesem Grund wurde die Frage nicht berücksichtigt.

Aufgrund der Ergebnisse des Abschlussfragebogens und des gewonnenen Eindrucks kann die Schulung als grundsätzlich erfolgreich bewertet werden. Es konnte einerseits eine Reihe von Anforderungen gesammelt werden, andererseits konnte die Akzeptanz der bisher entwickelten Idee an einer ersten Endnutzergruppe getestet werden. Das Ziel, bei den Teilnehmern eine gewisse Begeisterung für die Tablet-Technologie zu wecken und daraus mit einer positiven Stimmung in die Anforderungsevaluation zu starten ist gelungen. Es war sehr erfreulich zu erfahren, dass die initialen Ideen und Anforderungen nicht völlig in die falsche Richtung gingen, sondern im Grunde den Nerv und die Notwendigkeiten der Teilnehmer trafen. Die Evaluation der Anforderungen in Abschnitt 5.2.3 hat gezeigt, dass nur wenige in der Schulung genannte Anforderungen neu waren und dass an bestehenden Anforderungen nur geringe Änderungen vorgenommen werden mussten. Da einige der neuen Anforderungen nichts mit dem eigentlichen Ziel der Dokumentation der erbrachten Leistungen zu tun hat, wie z.B. die Verfügbarkeit eines Tropfenplans oder eine Spracherkennung, wurden die meisten neuen Anforderungen verworfen oder für eine eventuelle Weiterentwicklung zurückgestellt. Die übrigen Anforderungen zusammen mit den technischen Voraussetzungen aus Kapitel 3 ergeben eine gute Basis zur Entwicklung einer Anwendung, die Chancen hat, bei einer Weiterentwicklung zur Marktreife vom zuständigen Personal akzeptiert zu werden und einen echten Mehrwert bei der Dokumentation ärztliche übertragener Tätigkeiten zu liefern.

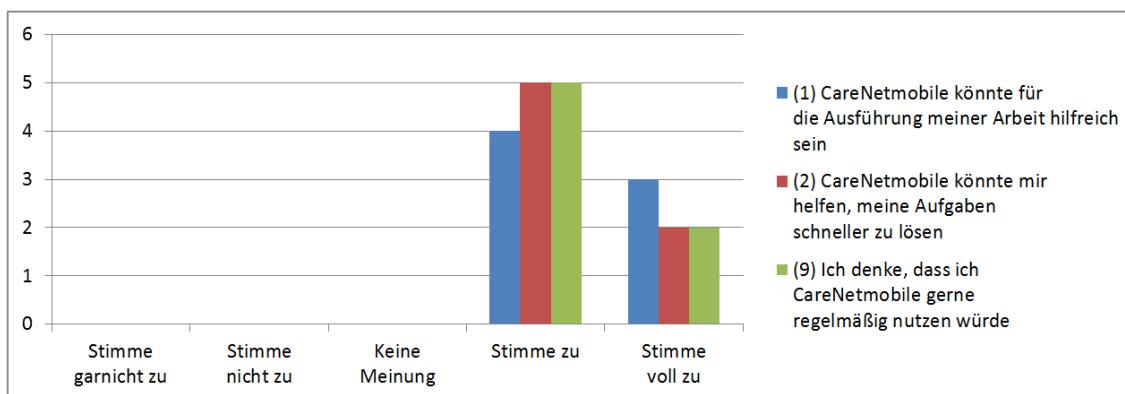


Abbildung 5.6.: Ergebnisse der Fragen zu potentiellem Nutzen und Akzeptanz von CareNetmobile .

6. Fazit

Zu Beginn dieser Master-Arbeit war das Ziel die Erstellung einer lauffähigen mobilen Anwendung, die Angehörige der alten- und Pflegeberufe darin unterstützen sollte, an sie delegierte ärztliche Tätigkeiten zu dokumentieren. Die Daten sollten an das zentrale System *CareNet* übermittelt werden, sodass der Arzt, der die Tätigkeiten in Auftrag gegeben hat, eine zeitnahe Kontrolle durchführen konnte.

Dieses Ziel konnte in der zur Verfügung stehenden Bearbeitungszeit nicht in vollem Umfang erreicht werden, jedoch konnten die wesentlichen Anforderungen umgesetzt werden. Eine Kernanforderung war, die Anwendung durch einen modularen Aufbau flexibel und an geänderte Rahmenbedingungen anpassbar zu gestalten. Technologisch war die einzige Vorgabe, dass moderne Tablet-PCs eingesetzt werden sollten. Dies erforderte einen zusätzlichen Evaluationsprozess der am besten geeigneten Technologie zur Erstellung einer mobilen Anwendung. Der Vergleich von nativ entwickelten Anwendungen, WebApps und hybriden Anwendungen ergab, dass der hybride Ansatz trotz eventueller die Performanz betreffender Nachteile die beste Basistechnologie sei. Das entscheidende Argument war die potenzielle Portierbarkeit der Anwendung auf verschiedene Plattformen (Android, iOS, Windows Phone), da die endgültig zu verwendende Hardware noch nicht fest stand.

Zur Realisierung der Anpassbarkeit der Anwendung wurde eine Plug-In-Architektur gewählt, bei der jegliche für den Endnutzer zu verwendende Funktionalität in Plug-Ins gekapselt werden sollte. Die Architektur konnte auf Basis von JavaScript, HTML und CSS umgesetzt werden, jedoch nahm der Entwurf und die Implementierung mehr Zeit in Anspruch, als zunächst geschätzt wurde. Zwar konnten trotz dieser Verzögerung die wesentlichen Funktionen umgesetzt werden, jedoch mussten manche wünschenswerte Eigenschaften aus Zeitgründen zurückgestellt werden. Implementiert werden konnte ein Plug-In zum Abrufen von Kontaktdaten aus CareNet und ein Plug-In zur Darstellung einer Tour für eine Medizinische Fachpflegekraft, die die einzelnen Stationen (Klienten) sowie die zu erbringenden Leistungen enthielt. Bei den Leistungen wurden auf Basis der Anforderungsanalyse drei Leistungstypen identifiziert: Standardleistungen, die nur eine Bestätigung benötigen, eine Wunddokumentation, mit der eine Bilddatei verknüpft werden muss und eine Vitalwerter-

fassung, mit der Messwerte von Blutdruck und Puls verbunden werden müssen. Standardleistungen und Wunddokumentation konnte den Anforderungen entsprechend umgesetzt werden, jedoch sollte die Vitalwerterfassung durch den Anschluss von Messgeräten über Bluetooth automatisiert werden, was nicht mehr realisiert werden konnte. Die Erfassung von Vitalwerten ist momentan nur über manuelle Eingabe möglich. Eine weitere geplante Funktionalität, die aus Zeitgründen nicht mehr implementiert werden konnte, ist das Hinzufügen ungeplanter Leistungen wie Toilettengänge zu einer Station.

Die bisher entwickelte mobile Anwendung bildet die notwendige Grundfunktionalität ab (bis auf die vorher genannten Punkte), um in CareNet zentral definierte Leistungen zu dokumentieren. Während der Anforderungsanalyse konnten allerdings noch einige Erweiterungsmöglichkeiten identifiziert werden, die CareNet *mobile* attraktiver machen würden. Hierzu zählen

- Navigation zum Klienten per GPS, entweder in die Anwendung integriert oder Nutzung externer Navigationsprogramme
- Plausibilitätsprüfung von eingegebenen Werten
- Warnungen bei eingegebenen kritischen Werten oder wenn Daten eines Klienten auf einen kritischen Zustand schließen lassen (beispielsweise wenn eine bestimmte Zeit kein Toilettengang dokumentiert wurde)
- Nutzung der per GPS ermittelten Position, um eine Behandlung auf dem Tablet-PC zu schließen, sobald sich die behandelnde MFP vom Einsatzort entfernt
- Autovervollständigung und Drop-Down-Listen zur Beschleunigung der Dokumentation von Standardfällen
- Integration einer Telefonfunktion für Notrufe und Rückfragen (Umsetzbarkeit hardwareabhängig)

Die Realisierung von CareNet *mobile* auf Basis einer Plug-In-Architektur bietet eine Vielzahl von Erweiterungsmöglichkeiten. Neben einer Erweiterung der Funktionen zur Dokumentation könnten auch völlig andere Anwendungsfelder von CareNet oder verwandten Produkten der nubedian GmbH mobil verfügbar gemacht werden, da diese über die gleich REST-Schnittstelle kommunizieren. Durch die Kapselung von Funktionalität innerhalb eines Plug-Ins könnten beispielsweise Vorgänge, Termine oder sonstige Daten mit einer individuellen grafischen Oberfläche dargestellt werden, ohne Implementierungen anderer Plug-Ins zu beeinflussen. Da die Plug-Ins über eine Konfigurationsdatei der Basisanwendung hinzugefügt werden, ist außerdem die Möglichkeit gegeben, einen Autorisierungsmechanismus zu implementieren, mit dem bestimmten Personen in Abhängigkeit von ihren Rechten bestimmte Plug-Ins angezeigt werden. Der Vorteil wäre, dass die mobile Anwendung zur Laufzeit konfiguriert werden könnte, ohne diese neu kompilieren zu müssen, da die Informationen zur Autorisierung über die REST-Schnittstelle bezogen werden könnten. Da alle verwendeten Nachrichten und Beschriftungen zentral in variablen festgehalten sind und nicht direkt in die einzelnen Elemente geschrieben sind, wäre sogar ein Modell zur Sprachkonfiguration denkbar, indem alle Beschriftungs- und Nachrichtenvariablen

beim Laden der Anwendung mit Daten aus einer Datei, die in der Anwendung hinterlegt werden kann, gefüllt werden.

Die Plattformunabhängigkeit von CareNet*mobile* und die damit verbundene Notwendigkeit, jegliche Funktionalität in JavaScript zu implementieren, bringt allerdings auch umfangreiche Nachteile mit sich. In Kapitel 4.1 wurde bereits das Problem erläutert, dass nahezu alles in JavaScript global verfügbar ist und damit keine Einschränkung auf manche Quellcode-Bereiche wie in einer objektorientierten Sprache möglich ist. Für die Entwicklung bedeutet dies eine Reihe von Fehlerquellen aufgrund von doppelten Methoden- oder Variablennamen, die nicht automatisch gefunden werden können und damit die Suche nach der sprichwörtlichen Nadel im Heuhaufen verursachen können. Die an die Entwickler gestellten Anforderungen bzgl. Disziplin bei der Einhaltung von Konventionen sind deutlich höher als bei einer stark werkzeuggestützten Entwicklung. Auch die Fehlerfindung ist auf Grund geeigneter Standardwerkzeuge vor allem manueller Natur. Zwar gibt es verschiedene Ansätze, die Fehlerfindung in JavaScript zu automatisieren, jedoch hat sich noch kein Standard-Werkzeug etabliert. Soll beispielsweise die für PhoneGap entwickelte Anwendung in einem WebKit-basierten Browser wie Google Chrome mit den dort vorhandenen Werkzeugen überprüft werden, müssen alle Anfragen an die Plattform, wie z.B. der Aufruf der Kamera, von einer zusätzlichen JavaScript-Bibliothek emuliert werden. Diese Emulatoren sind allerdings auch nicht fehlerfrei und deswegen können dort wieder externe Fehler auftreten, die nicht der eigentlich zu testenden Anwendung zuzuschreiben sind. Eine weitere Möglichkeit ist die Nutzung des "WEb INspector REmote" (Weinre) [29], der den Quellcode einer PhoneGap-Anwendung im Browser mit einem Schlüsselwort untersuchbar macht. Der Nachteil ist allerdings, dass bei Nutzung des Anbieterservers der gesamte Quellcode dem Anbieter zur Verfügung gestellt wird, was signifikante Sicherheitslücken verursachen kann. Es ist zwar möglich, einen eigenen Server mit der entsprechenden Software aufzusetzen, jedoch sorgt dies für deutlichen Zusatzaufwand, der sich bei Betrachtung der Mächtigkeit des Analysewerkzeuges nach Einschätzung des Autors kaum lohnt.

Positiv an der Aufgabenstellung der vorliegenden Arbeit war, dass ein gesamter Software-Entwicklungsprozess gestaltet werden konnte. Alle grundlegenden Stationen einer Entwicklung von der Anforderungsanalyse über eine prototypische Entwicklung bis hin zur technischen Evaluation und einer Evaluation mit möglichen Endnutzern konnten durchlaufen werden. Speziell die Zielgruppenschulung mit den Angehörigen der Alten- und Krankenpflegeberufe mit positiver Rückmeldung war eine schöne Bestätigung, dass die entworfene Anwendung nicht völlig an den praktischen Anforderungen vorbei ging. Mit den dort gewonnenen Eindrücken kann darauf geschlossen werden, dass eine technische Unterstützung der Dokumentation von Pflegetätigkeiten und delegierten ärztlichen Leistungen durchaus erwünscht ist, wenn sie denn anwenderfreundlich und auf die Bedürfnisse der Nutzer abgestimmt ist. Die grundsätzliche Akzeptanz ist vorhanden und kann durch gezielte Schulungen an intuitiv bedienbarer Hardware weiter gefördert und gefestigt werden. Probleme könnten allerdings auf technischer Ebene auftreten, da CareNet*mobile* momentan stark auf eine Anbindung an CareNet ausgelegt ist und noch keine Schnittstelle zu anderen Verwaltungssystemen implementiert ist. Für eine Verbreitung der mobilen Lösung wäre

dies jedoch unbedingt notwendig, da die Wahrscheinlichkeit dafür, dass eine soziale Einrichtung die finanziellen Mittel aufbringen möchte und aufbringen kann, um die gesamte vorhandene IT-Infrastruktur zu ersetzen, ist als gering einzuschätzen. Die Implementierung weiterer Schnittstellen wurde durch die Einführung der in Kapitel 3.2 erwähnten Middleware vorbereitet.

Die Unterstützung der betreuenden Mitarbeiter war sehr gut und das Arbeitsklima sehr freundlich und motivierend. Die gewährten Freiheiten bei der Umsetzung und der Arbeitsweise sorgten für einen weitreichenden Gestaltungsspielraum, der für einen freien Entwurf der Anwendung förderlich war. Der Ablauf der gesamten Arbeit sowie das erarbeitete Ergebnis kann aus Sicht des Autors als zufriedenstellend betrachtet werden.

7. Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des Karlsruher Instituts für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den 31. 03. 2012

Thomas Knapp

Anhang

A. Kernanforderungen an eine mobile Anwendung zur Dokumentation ärztlich übertragener Tätigkeiten

- Wie wurden die Anforderungen erhoben?
 - Richtlinie
 - Gespräche mit Raymond
 - Gespräche Corinna
 - Gesunder Menschenverstand
 - Wissen über Zielgruppe
 - Artikel Ärzteblatt
 - Diplomarbeit Christina Hardt
 - Projekt VitaBit (Schlüsselnummer, Tourenplan)
 - ...
- Kernanforderungen (Prio A)
 - Funktionalität
 - * Informationen über Klienten müssen verfügbar sein (Patientenakte) [CHECK]
 - * Persönliche Daten → Schutz durch Login-Funktion [CHECK]
 - * Dokumentation von Tätigkeiten aus Richtlinie muss möglich sein [CHECK]
 - * ambulante und stationäre Pflege abdecken! [CHECK]
 - * Dokumentierte Tätigkeiten müssen an zentrales System übertragbar sein [CHECK]
 - * Ablaufplan jeweils für eine MFP pro Tag/Schicht [CHECK]
 - * Übersicht, ob jeweilige Station schon vollständig abgearbeitet [CHECK]
 - * Es muss eine Tätigkeit als *nicht durchgeführt* markiert werden können + Begründung, falls etwas dazwischenkommt [CHECK]
 - * Einmal dokumentierte Werte sollten korrigierbar sein [CHECK]

- * Nachvollziehbarkeit von Änderungen [CHECK]
- * Daten ”offline“ auf dem Endgerät verfügbar, d.h. keine zwangsläufige Internetverbindung [CHECK]
- * Dokumentation von Wunden (Fotos) [CHECK]
- * Erfassung von Vitalwerten (Blutdruck, Puls), evtl. mittels automatischer Erfassung [CHECK]
- * Schlüsselnummer für Hausbesuche bzw. Zimmernummer bei stationären Behandlungen (aus VitaBIT) [CHECK]
- * Klar und übersichtlich strukturierte Anwendung, intuitive Bedienung (am besten Touch, Endnutzerwissen) [CHECK]
- * Für ambulanten Dienst eventuell Navigation mittels GPS [CHECK]
- * Telefongespräche? [CHECK]
- * ...
- Abgeleitete technische Anforderungen
 - * sozialer Kontext → günstige Hardware [CHECK]
 - * **Hardware:** mobiles Endgerät muss internetfähig sein (am besten über Mobilfunknetze für ambulanten Dienst), eine Kamera haben (Wunddokumentation), am besten Touch-Bedienung, darf trotzdem nicht so teuer sein, evtl. GPS, falls automatische Erfassung evtl. Bluetooth → Android-Tablet [CHECK]
 - * ...
- ...
- Wünschenswerte Eigenschaften, um Nutzungskomfort zu steigern (Prio B) [NEIN]
- Anforderungen, die über die reine Dokumentation oder dessen Unterstützung hinausgehen (Prio C, werden in der Arbeit nicht behandelt) [NEIN]

B. Liste grundsätzliche Entwurfsentscheidungen CareNetmobile

- Zentrales System: CareNet, basierend auf VitaBit [CHECK]
- Kommunikation mit REST; Was ist REST? Prinzipien? [CHECK]
- **CareNet mobile**
 - Name [CHECK]
 - Warum überhaupt Tablet? (Vgl. mit Diplomarbeit der anderen Studentin auf Nokia-Handy) [CHECK]
 - Nativ - WebApp - Hybrid [CHECK]

- PhoneGap, jQuery mobile [CHECK]
- Falls doch jemand Apple will, nach Mglk. suchen, wie das evtl. auch bedienst werden kann [CHECK]
- Architektur (Plug-In-Struktur) [CHECK]
- Änderungen an zu dokumentierenden Tätigkeiten sind zu erwarten, also möglichst flexible Architektur [CHECK]
 - Warum Plug-In? Vorteile? [CHECK]
- **Middleware**
 - Schnittstellenfunktion nochmal erwähnen [CHECK]
 - Technologie (PHP) [CHECK]
 - Wo wird Middleware positioniert? Welcher Server? (konzeptionell) [NEIN]

C. Basisanwendung

- Grundsätzlich: Plug-In-Architektur angelehnt an Eclipse-Bsp [CHECK]
- Bestandteile: Grundanwendung (Basis), Plug-Ins [CHECK]
- Aufgaben Grundanwendung: Sicherheit (Login, Kommunikation), Verwaltung von Plug-Ins, Bereitstellen einer GUI-Struktur, zentrale Verwaltung lokaler Datenbanken (Methoden) [CHECK]
- Programmereigenschaften von JavaScript ggü. Java [CHECK]
- Zentrale Verwaltung von IDs und sonstigen Strings [CHECK]
- Plug-Ins sollten keinen Einfluss auf Grundanwendung haben (Kapselung, aber komplette Trennung nicht möglich) [CHECK]
- Struktur GUI
- Grundstruktur: jQuery mobile (Seitenaufbau, alles mit Divs, CSS, wie Webseite) [MMHHH..., NÜTZLICH?]
- Login
 - als Dialog (vorgeschaltete Seite) [CHECK]
 - Mandant, Nutzernname, PW [CHECK]
 - Wg. REST-Prinzip: Nur einmalige Überprüfung, ob Daten korrekt sind, kein Aufbau einer dauerhaften Verbindung [CHECK]
 - Speicherung der Daten im lokalen Speicher im Klartext (HTML5/PhoneGap Key-Value-Speicher) [CHECK]
 - Senden der Daten bei jeder Anfrage (Beispielanfrage) [CHECK]

- Authentifizierungstring: mandant/nutzername:passwort [CHECK]
- Middleware verschlüsselt String mit Base64 und sendet den an CareNet (kann einfach entschlüsselt werden) [TEIL-CHECK]
- Daten werden so unverschlüsselt gesendet, d.h. Verbindung an sich muss gesichert werden (https) [CHECK]
- Bei Logout werden Daten einfach aus lokalem Speicher gelöscht [CHECK]
- Sicherheitsproblem bei eingeschleuster Spyware (JS) über Browser? Nein, wg. Unabhängigkeit von PhoneGap-Browser/WebView und installierten anderen Browsern [MMMHHH...]

D. Plug-In-Struktur

- Plug-Struktur:
 - Plug-In = Menge von Dateien [CHECK]
 - Alle zugehörigen Dateien gekapselt in einem Ordner [CHECK]
 - Namenskonventionen wg. JS global bzgl. Dateien [CHECK]
 - Namenskonventionen bzgl. Methoden, Variablen, CSS-Klassen [CHECK]
 - Konfiguration eingebundener Plug-Ins (JSON) [CHECK]
 - Icons (Darstellung in Dreierreihen + Grafik) [CHECK]
 - CSS-Datei [CHECK]
 - Beschreibung der Abhängigkeiten zu Bibliotheken wie jQuery, jQuery mobile, PhoneGap (Redundanzen vermeiden) [CHECK]
 - Grafik zu Plug-In-Struktur [CHECK]
 - JS-Datei (Problem der dynamischen Einbindung) [CHECK]
 - Konstruktor, Methode zur Initialisierung [CHECK]
- Probleme (Einbinden von JS und CSS Dateien) [CHECK]

E. Plug-In Kontakte

Für alle Plug-Ins feste Beschreibungsstruktur:

- **Begründung für Plug-In (aus welchen Anforderungen geht Plug-In hervor)**
 - Gezielt Infos über Klienten, Ärzte, Angehörige herausfinden [CHECK]
 - Vorwiegend Anzeige von Stammdaten [CHECK]
 - Unterscheidung der Anzeige nach Kontaktart (Klient, Angehöriger, Arzt, Organisation) [CHECK]

- Aktualisieren von Kontakten [CHECK]
- **Aufbau/Navigationsstruktur**
 - in CareNet: verschiedene Sichten auf Kontakte (soll auch im Mobilteil verfügbar sein) [CHECK]
 - Views und Kontakte über Zeiger miteinander verbunden [CHECK]
 - n:n-Beziehung [CHECK]
 - drei Tabellen: eine mit Views, eine mit Kontaktinfos, eine für Mapping [CHECK]
 - Erstellen der Tabellen bei Initialisierung [CHECK]
 - Füllen der Tabellen beim Öffnen des Plug-Ins, falls keine Daten vorhanden sind [CHECK]
 - Ansonsten Anzeige der vorhandenen Daten, manuelle Aktualisierung notwendig [CHECK]
 - Wie wird Laden der Infos vollzogen, sodass es zu keinen Brüchen kommt (View versucht Kontakte anzuzeigen, die noch nicht geladen wurden)

Literaturverzeichnis

- [1] S. B. Pressestelle, "Bevölkerung Deutschlands bis 2050 - 11. koordinierte Bevölkerungsvorausberechnung," November 2006.
- [2] D. T. Kopetsch, "Dem deutschen Gesundheitswesen gehen die Ärzte aus! Studie zur Altersstruktur-und Arztdahlentwicklung," Bundesärztekammer und Kassenärztliche Bundesvereinigung, Studie 5, August 2010.
- [3] "Richtlinie des Gemeinsamen Bundesausschusses über die Festlegung ärztlicher Tätigkeiten zur Übertragung auf Berufsangehörige der Alten- und Krankenpflege zur selbständigen Ausübung von Heilkunde im Rahmen von Modellvorhaben nach § 63 Abs. 3c SGB V." p. 1128, März 2012.
- [4] N. van den Berg, C. Meinke, R. Heymann, T. Fiß, E. Suckert, C. Pöller, A. Dreier, H. Rogalski, T. Karopka, R. Oppermann, and W. Hoffmann, "AGnES: Hausarztsunterstützung durch qualifizierte Praxismitarbeiter; Evaluation der Modellprojekte: Qualität und Akzeptanz," *Deutsches Ärzteblatt*, vol. Heft 1-2, no. 106, pp. 3–9, Januar 2009.
- [5] I. für hausärztliche Fortbildung im Deutschen Hausärzteverband (IhF) e.V., "Fortbildungscurriculum "Versorgungsassistentin in der Hausarztpraxis-VERAH"," Januar 2008.
- [6] I. für hausärztliche Fortbildung im Deutschen Hausärzteverband (IhF) e. V., "Versorgungsassistentin in der Hausarztpraxis - VERAH® [Informationen für Praxisinhaber]," 2008.
- [7] *ISO/IEC 25000 Software engineering - Software product Quality Requirements and Evaluation SQuaRE - Guide to SQuaRE*, International Standards Organization Std. 1, Rev. 1, 8 2005.
- [8] Forschungsprojekt VitaBIT, "Pflegeservice von morgen," *Deutsches Ärzteblatt*, vol. 106, no. 50, pp. 3,4, 2009.
- [9] D. Klingbeil, "Mehr Zeit für die Pflege der Kunden," *Häusliche Pflege*, vol. 11, pp. 32–34, November 2010.
- [10] nubedian GmbH. (2013, Januar). [Online]. Available: <http://www.nubedian.de/>
- [11] (2013, 01) Residenz Bad Friedrichshall. [Online]. Available: <http://www.pflegedienste-hn.drk.de/>

- [12] C. Hardt, "Entwicklung eines prototypischen Softwaresystems zur Unterstützung der Kommunikation und Dokumentation in der stationären Pflege," Diplomarbeit, Hochschule Karlsruhe Technik und Wirtschaft; Universität Karlsruhe (TH), April 2010.
- [13] R. Görlitz, L. Müller, and A. Rashid, "An individual patient-oriented stroke manager architecture," in *Proceedings of the eHealth2012*, Schreier, Hayn, Hörbst, and Ammenwerth, Eds., Wien, Österreich, Mai 2012.
- [14] M. zur Muehlen, J. V. Nickerson, and K. D. Swenson, "Developing web services choreography standards - the case REST vs. SOAP," *Decision Support Systems*, vol. 40, no. 1, pp. 9 – 29, 2005, <ce:title>Web services and process management</ce:title>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167923604000612>
- [15] (2013, Januar) Phonegap homepage. Adobe Systems Inc. [Online]. Available: <http://www.phonegap.com/>
- [16] (2013) jquery mobile homepage. The jQuery Foundation. [Online]. Available: <http://jquerymobile.com/>
- [17] (2013, Januar) Sencha touch homepage. Sencha Inc. [Online]. Available: <http://www.sencha.com/products/touch>
- [18] (2013, Januar) Architektur Eclipse. The Eclipse Foundation. [Online]. Available: <http://help.eclipse.org/juno/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Farch.htm>
- [19] (2013, Januar) PHP Homepage. The PHP Group. [Online]. Available: <http://www.php.net>
- [20] (2013, Januar) Verwaltung von String Ressourcen bei der nativen Android-Anwendungsentwicklung. Google Inc. [Online]. Available: <http://developer.android.com/guide/topics/resources/string-resource.html>
- [21] K. Janowicz, *Sicherheit im Internet : [Gefahren einschätzen, Risiken minimieren: fundiertes Hintergrundwissen rund ums Internet; effektiver Schutz für den PC: behandelt Windows XP und Vista; mit zahlreichen Sicherheitstools auf CD-ROM]*, 3rd ed., ser. oreillys basics. Beijing: O'Reilly, 2007, früher mit der Nummer 9783897214255. [Online]. Available: <http://swbplus.bsz-bw.de/bsz26744186xcov.htm>; <http://www.gbv.de/dms/ilmenau/toc/527923656.PDF>
- [22] M. Fowler, *Patterns of enterprise application architecture*, ser. The Addison-Wesley signature series. Boston, Mass.: Addison-Wesley, 2003.
- [23] T. Neil, *Mobile Design Pattern Gallery : [UI Patterns for iOS, Android and more]*, 1st ed. Beijing: O'Reilly, 2012. [Online]. Available: http://deposit.d-nb.de/cgi-bin/dokserv?id=3962922&prov=M&dok_var=1&dok_ext=htm; <http://swbplus.bsz-bw.de/bsz363523030cov.htm>
- [24] (2013, April) Jasmine - JavaScript Unit Test Framework. Pivotal Labs. [Online]. Available: <http://pivotal.github.io/jasmine/>

- [25] (2013, April) QUnit - A JavaScript Unit Test. jQuery Foundation. [Online]. Available: <http://qunitjs.com/>
- [26] G. S. Nickell and J. N. Pinto, "The Computer Attitude Scale," *Computers in Human Behavior*, vol. 2, pp. 301–306, 1986.
- [27] M. Schmon, "Herausforderungen einer nutzerorientierten Spezifikation, prototypische Entwicklung und Evaluation einer Case Management Software für die Pflege," Master's thesis, Hochschule Reutlingen, 2011.
- [28] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly*, vol. 27, no. 3, pp. pp. 425–478, 2003. [Online]. Available: <http://www.jstor.org/stable/30036540>
- [29] P. Mueller. (2013, April) Weinre - WEb INspector REmote. [Online]. Available: <http://people.apache.org/~pmuellr/weinre/docs/latest/>