

# Anforderungs- spezifikationen

HEALTH VISITOR WEB APP

SUTER RAPHAEL, AUDERSET FLORIAN, HÄUSLER REMO,  
RUDOLF MATHIAS, FLÜHMANN TOBIAS

## DOKUMENTENPROTOKOLL

Version	Datum	Name	Änderungen
0.1	01.04.2016	alle	Initiale Version
0.2	01.04.2016	Florian Auderset	Vorwort, Einleitung erfassen
0.3	02.04.2016	Mathias Rudolf	Use Case Diagramm Medikamente, UC-01 erstellt
0.4	03.04.2016	Tobias Flühmann	System Architecture, System Requirements
0.5	03.04.2016	Raphael Suter	Kapitel Testing verfasst und detaillierter Use Case UC-02 dokumentiert
0.6	03.04.2016	Tobias Flühmann	Diverse Grafiken angepasst
0.7	05.04.2016	Florian Auderset	System Model- und Evolution.
0.8	05.04.2016	Remo Häusler	Refactoring FR / NFR
0.9	05.04.2016	Raphael Suter	Grafik System Model, Typo und kleinere Anpassungen
0.10	05.04.2016	Tobias Flühmann	System Evolution erweitert
1.0	05.04.2016	alle	Finale Version

## INHALTSVERZEICHNIS

<b>DOKUMENTENPROTOKOLL</b>	<b>2</b>
<b>INHALTSVERZEICHNIS</b>	<b>3</b>
<b>1 VORWORT</b>	<b>5</b>
<b>2 EINLEITUNG</b>	<b>6</b>
<b>3 GLOSSAR</b>	<b>7</b>
<b>4 BENUTZERANFORDERUNGEN</b>	<b>8</b>
<b>4.1 USE CASE DIAGRAMME</b>	<b>8</b>
4.1.1 ALARMIERUNG	9
4.1.2 RAPPORTIERUNG	10
4.1.3 AKTUELLER TERMIN	11
4.1.4 MEDIKAMENTE FÜR DEN TAG	11
4.1.5 TERMINE FÜR DEN TAG	12
<b>4.2 DETAILLIERTE USE CASES</b>	<b>13</b>
4.2.1 MEDIKAMENTE BESTÄTIGEN	13
4.2.2 RAPPORTIERUNG	14
<b>5 SYSTEMARCHITEKTUR</b>	<b>16</b>
<b>5.1 HARDWAREKOMPONENTEN</b>	<b>16</b>
5.1.1 WEB APPLICATION FIREWALL (WAF)	16
5.1.2 WEBSERVER	16
5.1.3 APPLICATION SERVER	16
<b>6 SYSTEMANFORDERUNGSSPEZIFIKATION</b>	<b>17</b>
<b>6.1 FUNCTIONAL REQUIREMENTS</b>	<b>17</b>
6.1.1 OUT OF SCOPE	17
<b>6.2 NON-FUNCTIONAL REQUIREMENTS</b>	<b>18</b>
6.2.1 SYSTEMVERFÜGBARKEIT	18
6.2.2 PERFORMANCE	18
6.2.3 SCHNITTSTELLEN	18
6.2.4 SICHERHEIT	19
6.2.5 USABILITY	20
<b>7 SYSTEM MODELS</b>	<b>21</b>
<b>7.1 BENUTZERVERWALTUNGS MODUL</b>	<b>22</b>
<b>7.2 STAMMDATEN MODUL</b>	<b>22</b>
<b>7.3 PATIENTENDOSSIER MODUL</b>	<b>22</b>
<b>7.4 TERMINPLANER MODUL</b>	<b>22</b>
<b>7.5 HEALTH VISITORS MODUL</b>	<b>22</b>

<b>8</b>	<b>SYSTEM EVOLUTION</b>	<b>23</b>
8.1	ALLGEMEIN	23
8.2	NEUE BENUTZERGRUPPE PATIENT	23
8.3	NATIVE APPLIKATION	23
8.4	DYNAMISCHE TERMINZUORDNUNG	23
<b>9</b>	<b>TESTING</b>	<b>24</b>
<b>10</b>	<b>INDEX</b>	<b>25</b>
10.1	ABBILDUNGSVERZEICHNIS	25

## 1 VORWORT

Dieses Dokument beschreibt, welche Ansprüche an die Health Visitors Web App gestellt werden. Im Weiteren wird diese nur noch als System bezeichnet. Es definiert Benutzer- und Systemanforderungen und spezifiziert diese.

Das Dokument richtet sich an den Endbenutzer, die Health Visitors, zur Überprüfung der Benutzeranforderungen. Zudem richtet sich das Dokument an das Management des Kunden, zur Überprüfung der Systemanforderungen und zum planen des Entwicklungsprozesses. Es richtet sich ebenfalls an die System Engineers als Referenz und Gesamtübersicht, welche Funktionen das System am Ende übernehmen soll. Aus dem Dokument können fürs Testing die nötigen Tests abgeleitet werden, sofern nicht weiter vom Kunden spezifiziert. Am Schluss richtet sich das Dokument an die Systemwartungsverantwortlichen, welche daraus die Zusammenhänge des Systems, dessen Subsystemen und eventueller externer Systeme rauslesen können.

## 2 EINLEITUNG

Die Health Visitors dokumentieren, die Besuche und Patientendossiers in verschiedenen Systemen. Die geführten Interviews zeigen sogar, dass gewisse Betriebe bereits teils digitalisiert arbeiten, andere noch alles von Hand in Papierdossiers dokumentieren. Die Bedürfnisse der Health Visitors zeigen, dass sie gerne eine komplett digitalisierte Lösung bevorzugen, welche Ihnen Hilfestellung bei den verschiedenen Tätigkeiten bietet.

Ziel ist es, ein System zu erschaffen, welches die Arbeitsabläufe der Health Visitors vereinfacht und effizienter macht. So soll der Health Visitor am Morgen eine Terminübersicht der anstehenden Besuche erhalten. Eine Packlisten-funktion überfliegt die Termine und gibt dem Health Visitor eine Auflistung der Medikamente und sonstiger Materialien, welche vorbereitet und eingepackt werden müssen. Das System soll dem Health Visitor ermöglichen die Zeit des Besuchs festzuhalten. Der Health Visitor erhält zu jedem Termin genaue Informationen zum Besuch. Diese Informationen beinhalten Gesundheitszustand des Patienten, Medikation, Krankheitsgeschichte, sowie Notfallkontakte. Der Health Visitor kann über das System die Besuche rapportieren. Die Rapporte können geschrieben oder um Zeit zu sparen gesprochen aufgegeben werden. Dazu kommen die Erfassung der verabreichten Medikation und neuer Termine.

Die Detailansicht des Besuchs soll zudem eine Checkliste aufzeigen, welche dem Health Visitor als Gedankenstütze während des Besuchs dient.

Die Health Visitors müssen bei besuchen oft externe Stellen hinzuziehen (Bsp. Psychiater, Krankenwagen, Notfallkontakte). Das System bietet dem Health Visitor im Patientendossier an, die entsprechenden Personen anzurufen, oder eine E-Mail Nachricht zu senden. Das erspart das mühselige Raussuchen der entsprechenden Telefonnummern und E-Mail Adressen.

Das System bezieht die Termine aus einer externen Datenquelle. Die digitale Terminverwaltung ermöglicht es Termine viel schneller und flexibler zu vereinbaren. Das System zeigt an, falls sich Termine ändern, schlägt Ausweichdaten vor und ermöglicht es neue Termine direkt vor Ort beim Patienten zu vereinbaren.

Mit der Einführung des Systems wird der Datenfluss, sowie die Qualität und Kontrolle der Patientendossiers verbessert. Es ermöglicht die Erfassung der Ressourcen pro Patient, was der Verrechnung der Dienstleistungen dient. Zudem verbessert das System die Qualität der Dienstleistung durch bessere Information und Planung des Health Visitors.

### 3 GLOSSAR

Das Glossar beinhaltet eine Auflistung der verwendeten Fremdwörter und Begriffe, sowie die dazugehörige Begriffserklärung.

Begriff	Erklärung
<b>ERP</b>	Enterprise Resource Planning
<b>Integrationsumgebung</b>	Eine Systeminstallation für die (Weiter-)Entwicklung des Systems. Die Integrationsumgebung ist beim Entwicklerteam installiert und beinhaltet nur anonymisierte Testdaten.
<b>native Applikation</b>	Eine Applikation, welche für ein spezifisches Betriebssystem geschrieben wurde. Grosse Verbreitung finden iOS (Apple Geräte), Android (verschiedene Hersteller, u.a. SAMSUNG).
<b>Notifications</b>	Benachrichtigungen (meist auf mobilen Geräten)
<b>Produktionsumgebung</b>	Die produktive Installation des Systems beim Kunden, auf welchem die Mitarbeiter tagtäglich arbeiten. Dies ist das Endprodukt, welches der Kunde effektiv bestellt hat.
<b>Sprint</b>	In der agilen Softwareentwicklung ein Entwicklungsdurchlauf, in dem neue Funktionen durch die Entwickler erstellt, implementiert und getestet werden.
<b>State of the Art</b>	Auf dem neusten Stand der Technik.
<b>Storyboard</b>	Eine Zusammenführung von mehreren Use Cases, die benötigt werden um eine bestimmte Tätigkeit eines Users während seiner Arbeit mit dem System zu vollführen.
<b>System</b>	Health Visitors Web App
<b>Testumgebung</b>	Eine Systeminstallation, die beim Kunden installiert ist und ein Abbild der Produktionsumgebung ist. Diese Umgebung wird für Tests durch den Kunden verwendet. Sämtliche Anpassungen werden immer zuerst auf der Testumgebung geprüft und abgenommen, bevor sie in die Produktionsumgebung implementiert werden.
<b>Use Case</b>	Beschreibung eines spezifischen Anwendungsfalls, der alle möglichen Szenarien, die eintreten können, abbildet
<b>User</b>	Endanwender. Der Mitarbeiter, der das System produktiv benutzt.
<b>WAF</b>	Web Application Firewall

## 4 BENUTZERANFORDERUNGEN

### **Zwingend notwendige Anforderungen:**

1. Medikamente für den Tag
2. Termine heute
3. Alarm
4. Aktueller Termin
5. Schriftlicher Report & Termin Anfang/Ende Bestätigung

### **Optionale Anforderungen:**

1. Termine (alle)
2. Notfallkontakte Patient
3. Patientenakte (Details)
4. Voice Report
5. Navigation
6. Medikamente und Material bestellen

### **Anforderungen, die mit dieser Applikation nicht abgedeckt werden:**

1. Krankheit für Mitarbeiter erfassen (Admin)

## 4.1 USE CASE DIAGRAMME

Die Use Case Diagramme sind unterteilt in die nächstgrösseren Anwendungsbereiche und wurden mit folgendem Tool [http://yuml.me/diagram/scruffy/Use Case/draw](http://yuml.me/diagram/scruffy/Use%20Case/draw) erstellt.



#### 4.1.1 ALARMIERUNG

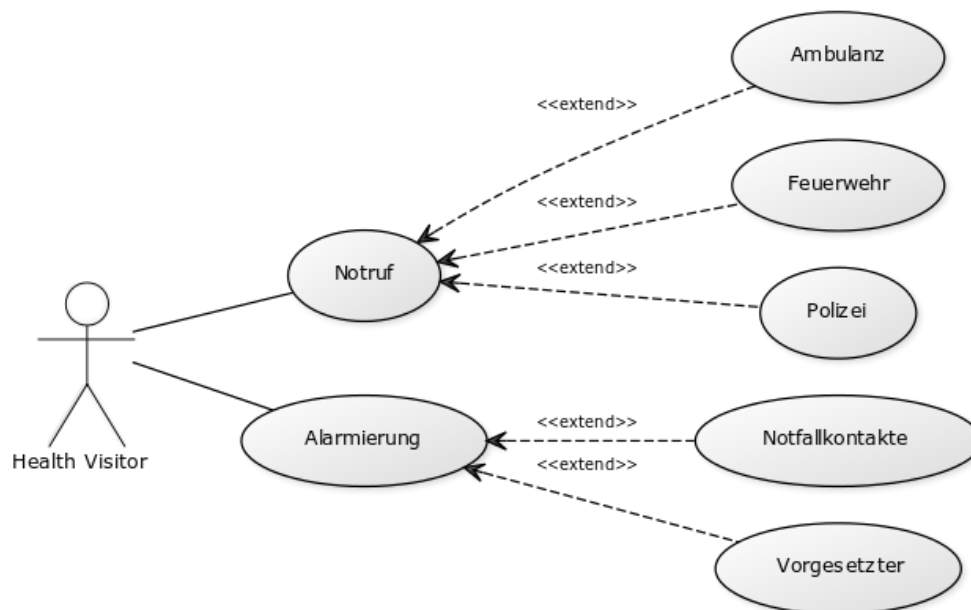


Abbildung 1: Use Case "Alarmierung"

##### Kurzbeschreibung:

Die Alarmierung beinhaltet unter anderem die Alarmierung der Notfallkontakte oder interne Kontakte, sowie der Notruf bei allgemeinen Stellen (Polizei, Feuerwehr, Ambulanz).

Der Health Visitor erhält im Patientendossier einen Bereich „Alarmierung“, in welchem alle für den Patienten relevanten Nummern hinterlegt sind. Mit einem Tap / Klick auf den Kontakt wird der Kontakt angerufen (Voraussetzung Telefonfunktion steht auf der Hardware zur Verfügung).

#### 4.1.2 RAPPORTIERUNG

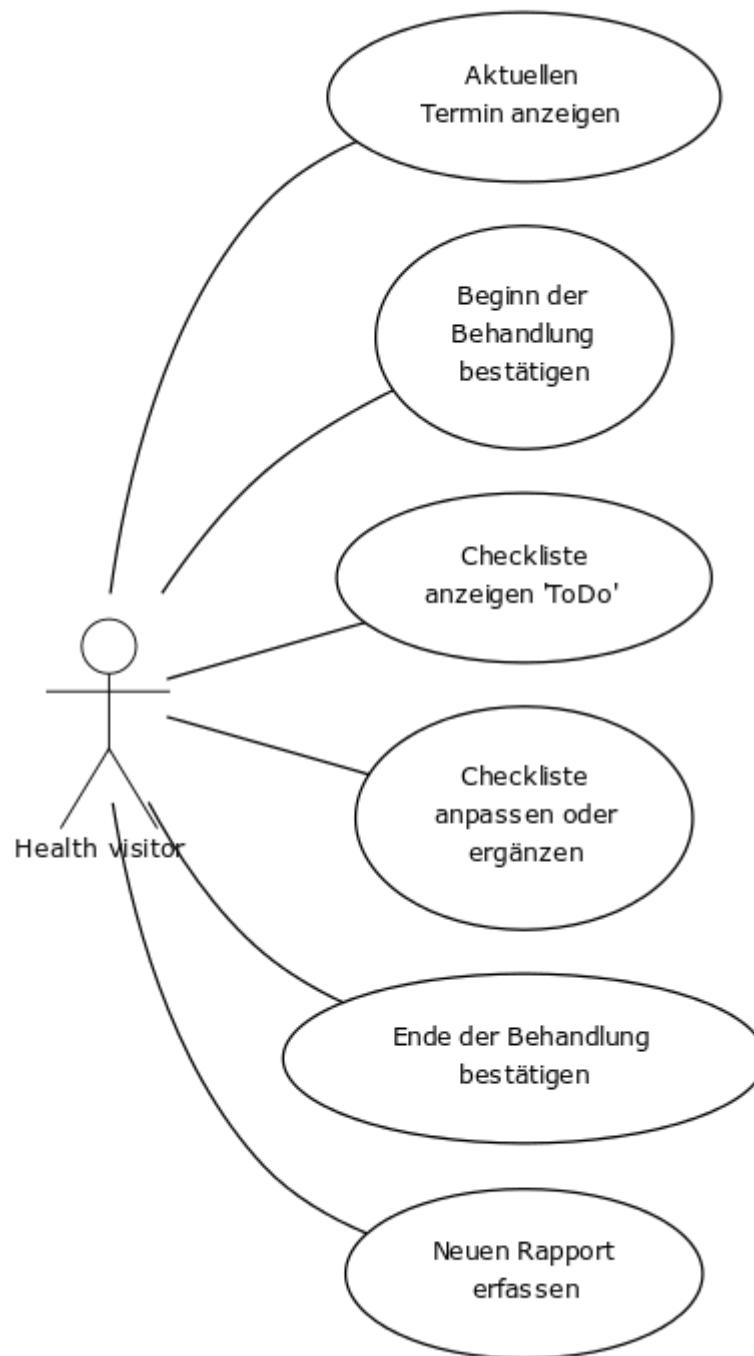


Abbildung 2: Use Case "Rapportierung"

##### Kurzbeschreibung:

Die Rapportierung bietet dem Health Visitor an seine Ankunft zu bestätigen, danach wird die Checkliste freigegeben. Der Health Visitor erledigt seine Arbeiten und notiert die Erledigung auf der Checkliste. Am Schluss des Besuchs kann er allfällige wichtige Informationen zum Besuch entweder schriftlich erfassen oder via Diktierfunktion aufzeichnen. Bevor er den Patienten verlässt kann er das Ende der Behandlung bestätigen, die Zeit wird entsprechend im System erfasst.

#### 4.1.3 AKTUELLER TERMIN

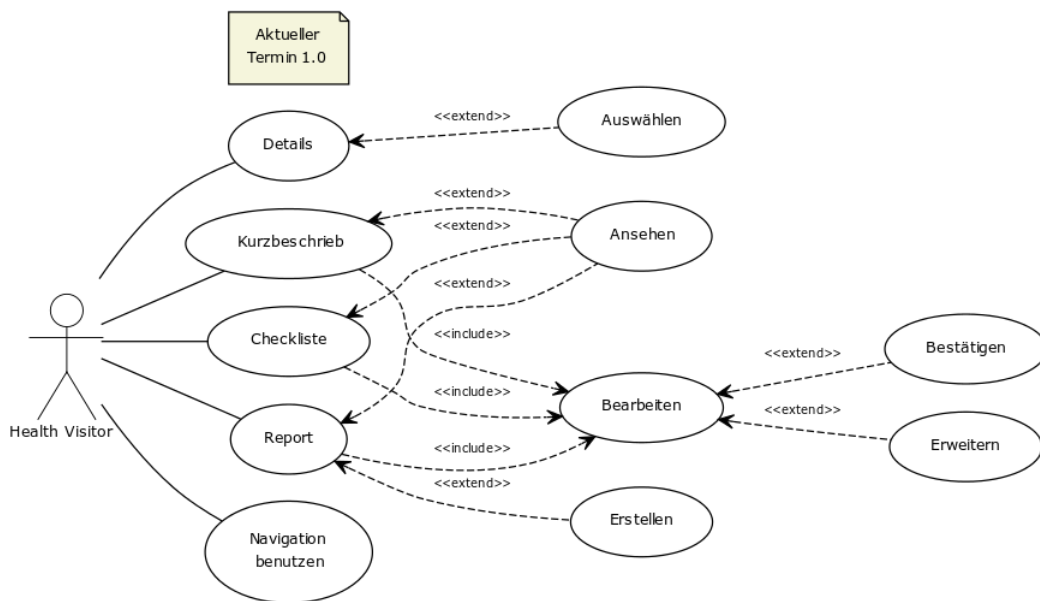


Abbildung 3: Use Case "Aktueller Termin"

##### Kurzbeschreibung:

Der Health Visitor kommt vom Hauptmenü aus mit einem Klick auf seinen aktuellen Termin. Dort findet er alle nötigen Informationen über seinen nächsten Patienten. Er sieht eine Checkliste mit allen notwendigen Tätigkeiten, die bei diesem entsprechenden Patienten gemacht werden müssen. Ebenfalls kann er sich dort alte Rapporte und einen Kurzbeschreibung sowie die persönlichen Notfalladressen des Patienten einsehen. Dadurch weiss der Health Visitor bereits vor dem Besuch des Kunden, was zu tun ist und was bisher passiert ist.

#### 4.1.4 MEDIKAMENTE FÜR DEN TAG

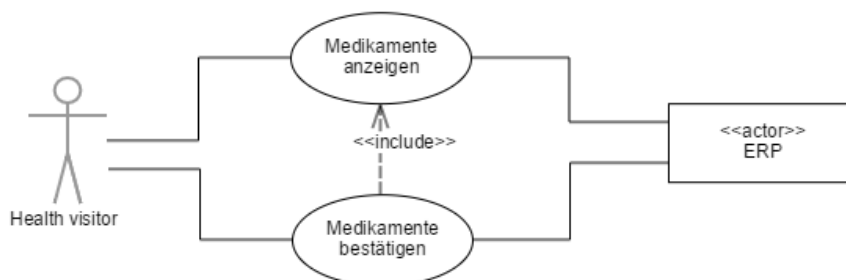


Abbildung 4: Use Case "Medikamente"

##### Kurzbeschreibung:

Der Healt Visitor kann am Morgen vor seinem Einsatz die benötigten Medikamente für die zu besuchenden Patienten anzeigen lassen. Medikamente die eingepackt sind, können bestätigt werden.

#### 4.1.5 TERMINE FÜR DEN TAG

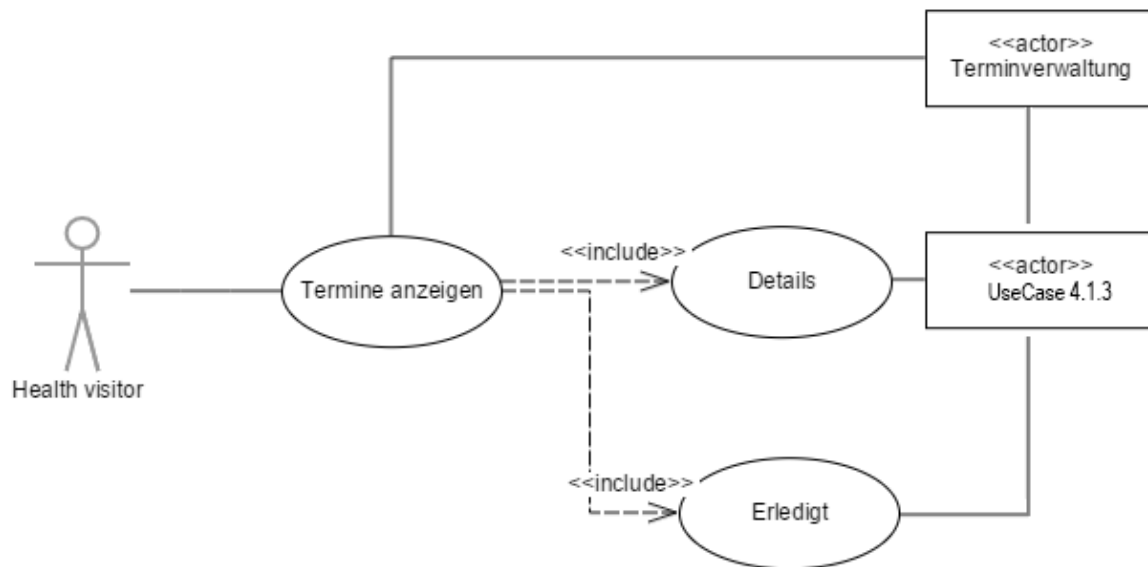


Abbildung 5: Use Case "Termine"

##### Kurzbeschreibung:

Der Health Visitor kann seine Termine für den Tag einsehen. Er kann die Details zu einem Termin einsehen. Er kann einen Termin als erledigt markieren.

## 4.2 DETAILLIERTE USE CASES

### 4.2.1 MEDIKAMENTE BESTÄTIGEN

Nr. und Name:	UC-01: Medikamente bestätigen
Szenario:	Der Health Visitor geht morgens ins Büro und schaut auf dem System nach, welche Medikamente er für den aktuellen Tag benötigt und nimmt diese anhand der angegebenen Mengen aus dem Medikamentenschrank. Anschliessend bestätigt er auf dem System, welche Medikamente und in welcher Anzahl er diese aus dem Medikamentenschrank entfernt hat.
Kurzbeschreibung:	Der Benutzer bestätigt alle Medikamente, die er aus dem Lager nimmt
Beteiligt Akteure:	Health Visitor, ERP
Auslöser / Vorbedingung:	Die Liste der benötigten Medikamente wurde vorher abgefragt
Ergebnisse / Nachbedingung:	Der Bestand der Medikamente wurde im ERP angepasst

#### Ablauf:

Nr.	Wer	Was
1.0	Health Visitor	Bestätigt alle entnommenen Medikamente
1.1	System	Sendet die Bestätigung an das ERP
1.2	ERP	Prüft den Medikamentenbestand
1.3	ERP	Aktualisiert den Medikamentenbestand und sendet Bestätigung an System
1.4	System	Sendet Bestätigung an den Benutzer

#### Ausnahmen, Varianten:

Nr.	Wer	Was
1.1		Keine Verbindung zu ERP
1.1.1	System	Fehlermeldung an den Benutzer
1.1.2	Health Visitor	Administrator oder Support informieren
1.2		Medikamentenbestand zu gering
1.2.1	ERP	Meldung an System, dass Bestand zu gering ist
1.2.2	System	Meldung an Benutzer, dass Bestand zu gering ist
1.2.3	Health Visitor	Überprüft seine Eingabe, bei Fehler weiter mit 1.0
1.2.4	Health Visitor	Administrator oder Support informieren

#### 4.2.2 RAPPORTIERUNG

Nr. und Name:	UC-02: Rapportierung
Szenario:	Der Health Visitor besucht aufgrund seiner Termine einen Patienten und nimmt vor Ort die entsprechende Behandlung vor. Nach Abschluss der Behandlung muss der Health Visitor einen Rapport über den Besuch beim Patienten erfassen. Dieser wird im Patientendossier abgelegt und steht danach im System den berechtigten Personen zur Verfügung. Dieser Rapport wird auch für die Abrechnung mit der Krankenkasse benötigt und muss deshalb zwingend nach jedem Besuch erfasst werden. Das System unterstützt den Health Visitor bei der Erstellung des Rapports, damit alle Rapports gewissen Mindeststandards entsprechen.
Kurzbeschreibung:	Der Health Visitor erstellt aufgrund seines Behandlungstermins einen Rapport über die Vorkommnisse und getätigten Arbeiten bei seinem Patienten.
Beteiligt Akteure:	Health Visitor
Auslöser / Vorbedingung:	Der Health Visitor hat seine Behandlung beim Patienten beendet. Vorbedingung: Beim Beginn des Behandlungstermins hat der Health Visitor im System beim aktuellen Termin seine Ankunft bestätigt.
Ergebnisse / Nachbedingung:	Vollständiger Rapport über den Behandlungstermin beim Patienten befindet sich im System.

##### Ablauf:

Nr.	Wer	Was
2.0	Health Visitor	Aktuellen Termin öffnen
2.1	Health Visitor	Ende des Termins durch Klick auf „Ende bestätigen“ festhalten
2.2	Health Visitor	Erledigte Arbeiten in der Checkliste abhaken.
2.3	Health Visitor	Weitere vorgenommene Arbeiten in der Checkliste ergänzen
2.4	Health Visitor	Durch Klick auf „Neuer Rapport erfassen“ die Erstellung des Rapports beginnen.
2.5	Health Visitor	Bereits ausgefüllte Daten überprüfen (Zeit Beginn, Zeit Ende, Checkliste)
2.6	Health Visitor	Verabreichte Medikamente ergänzen
2.7	Health Visitor	Abgegebene Medikamente ergänzen
2.8	Health Visitor	Feld „Bericht“ mit den Geschehnissen ergänzen.
2.9	Health Visitor	Rapport mit Klick auf „Rapport abschliessen“ speichern und im System ablegen.

##### Ausnahmen, Varianten:

Nr.	Wer	Was
2.0	Health Visitor	Falscher Termin wird angezeigt
2.0.1	Health Visitor	Zurück auf Hauptmenü
2.0.2	Health Visitor	Klick auf Termine
2.0.3	Health Visitor	Auswählen des gewünschten Termins. Weiter bei 2.1
2.5	Health Visitor	Keine Daten sichtbar
2.5.1	Health Visitor	Daten manuell ergänzen. Weiter mit 2.6

2.9		Rapport konnte nicht online gespeichert werden.
2.9.1	Health Visitor	Meldung an Benutzer, dass Rapport nicht im System abgelegt wurde.
2.9.2	Health Visitor	Internetverbindung überprüfen. Falls ja, weiter mit 2.9.2.1 sonst 2.9.3
2.9.2.1	Health Visitor	Meldung an Support durch Benutzer
2.9.2.2	Health Visitor	Daten auf Gerät zwischenspeichern.
2.9.2.3	Health Visitor	Auf Antwort von Support warten und nach deren Vorgaben weitermachen.
2.9.3	System	Daten auf Gerät zwischenspeichern
2.9.4	System	Bei verfügbarer Verbindung automatisch durch das System hochladen.
2.9.5	Health Visitor	Meldung an Benutzer, dass Daten jetzt im System abgelegt sind.

## 5 SYSTEMARCHITEKTUR

Unser System soll grundlegend aus mehreren Komponenten bestehen. Die Hardwarekomponenten werden im folgenden Abschnitt beschrieben.

### 5.1 HARDWAREKOMPONENTEN

#### 5.1.1 WEB APPLICATION FIREWALL (WAF)

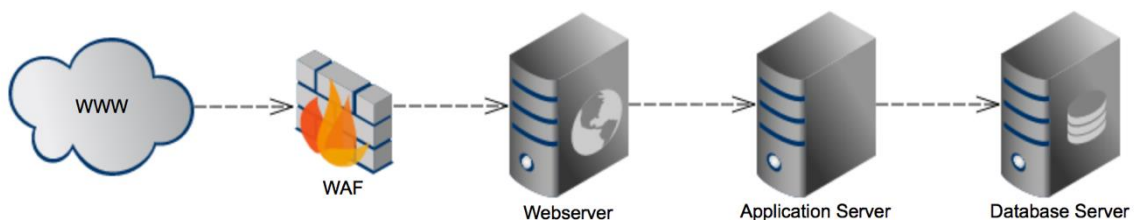
Um das Risiko für Angriffe oder Spoofing aus dem Internet auf ein Minimum zu beschränken wird eine Web Application Firewall vor das System geschaltet um den Zugriff genau beschränken zu können. Angriffe sollen so frühzeitig erkannt werden können und abgewendet werden.

#### 5.1.2 WEBSERVER

Als nächste Instanz wird ein Webserver eingerichtet. Dieser Webserver wird Anfragen des Clients auf dem Port 443 (https) beantworten. Der Webserver muss so konfiguriert werden, damit er keine unnötigen Daten an den Client weiterschickt. Als unnötige Daten werden beispielsweise Serverversion, Patchlevel usw. angeschaut. Der Webserver hat die Aufgabe den Request an ein Socket des Application Servers weiterzuleiten.

#### 5.1.3 APPLICATION SERVER

Als Application Server wird eine Distribution aus dem Java Bereich eingesetzt werden, die aber in einem weiteren dafür vorgesehenen Dokument noch genauer definiert werden wird.





## 6 SYSTEMANFORDERUNGSSPEZIFIKATION

### 6.1 FUNCTIONAL REQUIREMENTS

ID	Name	Beschreibung
<b>FR-01</b>	Information	Ist eine Aktion abgeschlossen, soll der Benutzer darüber informiert werden.
<b>FR-02</b>	Alarmierung	Das System soll eine Funktion liefern die Notfallkontakte direkt anrufen zu können, sofern das Endgerät das Unterstützt.
<b>FR-03</b>	Alarmierung II	Das System soll den Namen, Beziehung zum Patient und die Telefonnummer anzeigen.
<b>FR-04</b>	Termin	Jeder Termin soll eine Kurzzusammenfassung des Patienten haben.
<b>FR-05</b>	Termin II	Es soll ein Button zur Verfügung stehen mit dem der Beginn oder das Ende des Termins bestätigt werden kann.

Medikamentierung:

#### 6.1.1 OUT OF SCOPE

ID	Name	Beschreibung
<b>FR-06</b>	Terminverwaltung	Das System verwaltet nicht selber die Termine
<b>FR-07</b>	Medikamenteninformationen	Das System verwaltet keine Informationen über Medikamente
<b>FR-08</b>	Personalverwaltung	Das System muss nicht das Personal verwalten.

## 6.2 NON-FUNCTIONAL REQUIREMENTS

### 6.2.1 SYSTEMVERFÜGBARKEIT

ID	Name	Beschreibung
<b>NFR-01.1</b>	Anforderung	Wir stellen die Anforderung an unser System 7*24 online zu sein Wenn in Zukunft eine höhere Verfügbarkeit erreicht werden soll, muss die Architektur erweitert werden.
<b>NFR-01.2</b>	Downtime	Hier sollte eine Systemverfügbarkeit von 98.809% erreicht werden was einer Downtime von 120 Minuten pro Woche entspricht. Die 120 Minuten pro Woche sollten für Entwicklungs- und Wartungsarbeiten verwendet werden.

### 6.2.2 PERFORMANCE

ID	Name	Beschreibung
<b>NFR-02.1</b>	Antwortzeiten	Um einen optimalen Service für den Kunden gewährleisten zu können und die Kundenakzeptanz hoch zu halten, ist das Ziel die Responsetime des Systems im Mittel auf maximal 350 Millisekunden zu halten. Ein Maximum von einer Sekunde sollte nicht überschritten werden. Die Messung dieser Anforderungen soll nur auf einem Client direkt am System gemessen werden, da auf andere Faktoren wie Netzabdeckung usw. nicht Einfluss genommen werden kann.
<b>NFR-02.2</b>	Transaktionszeiten	Entsprechend den Antwortzeiten müssen auch die Datenbanktransaktionszeiten klein sein. Maximal 50% der Antwortzeit sollte zur Transaktion verwendet werden.

### 6.2.3 SCHNITTSTELLEN

ID	Name	Beschreibung
<b>NFR-03.1</b>	Medikamentenzulieferer	Das System muss eine Schnittstelle zum Medikamentenzulieferer benutzen können. Medikamentenbestellungen (UC 01).
<b>NFR-03.2</b>	Terminverwaltung	Die Termine werden von einer auf Terminverwaltung spezialisierter Software verwaltet.

## 6.2.4 SICHERHEIT

ID	Name	Beschreibung
<b>NFR-04.1</b>	Authentifizierung	Um das System bestmöglich zu schützen muss eine 2 Faktor Authentifizierung gemacht werden. Hier kommen verschiedene Lösungen in Frage. Denkbar wäre „Google Authenticator“ einzubinden, so dass sich der Benutzer auf seinem Endgerät mit dem Usernamen und Passwort einloggen muss, Authentifiziert wird und als dritten Schritt noch den jede Minute aktualisierten Code aus dem Google Authenticator nachtragen muss. Google Authenticator ist nicht an eine funktionierende Internetverbindung gebunden, somit könnte sich der Health Visitor auch in einem Keller ohne Empfang nur mit einer LAN Verbindung noch einloggen.
<b>NFR-04.2</b>	Autorisierung	<p>Zugriffsrechte auf die Applikation können durch geschultes Personal vergeben werden. Hierzu soll ein Rollenkonzept mit mindestens folgenden Rollen eingeführt werden:</p> <ol style="list-style-type: none"> <li>1. Guest (Standardrolle beim Einloggen, sieht nur kurze Systemeinführung)</li> <li>2. Health Visitor (Standardrolle für Berechtigte)</li> <li>3. Leader (Standardrolle für geschultes Personal)</li> <li>4. Admin (Systemadministratoren)</li> </ol> <p>Weitere Rollen können problemlos eingeführt werden, da es mehrere Ausprägungen von Health Visitors geben kann.</p>
<b>NFR-04.3</b>	Accounting	Während der User eingeloggt ist, sollen seine Aktivitäten geloggt werden. Dies betrifft besonders Aktionen die Änderungen in den Patientendaten und deren Medikation zur Folge haben. Diese Daten sind sehr heikel und müssen verfolgbar sein.
<b>NFR-04.4</b>	Zugriffskontrolle	Das System soll nur autorisiertem Personal zur Verfügung stehen und nicht der Öffentlichkeit zugänglich sein. Der User muss sich mit seinem Benutzernamen und Kennwort authentifizieren, danach mittels einem aktuell erstellten Token die Anmeldung bestätigen. Zudem werden auf dem Webserver nur Verbindungen von bekannten Geräten zugelassen (MAC-Adressfilterung).
<b>NFR-04.5</b>	Datenübertragung	Daten werden nur verschlüsselt übermittelt.
<b>NFR-04.6</b>	Zwischenspeicher	Daten werden nur temporär im Client gespeichert. Diese Daten sind zu verschlüsseln.

## 6.2.5 USABILITY

ID	Name	Beschreibung
<b>NFR-03.1</b>	Anforderung	Aus den Interviews hat sich ergeben, dass eine elektronische Hilfe nur nützlich ist, wenn diese auch eine gewisse Nutzerfreundlichkeit aufweist. Die Benutzer wollen nicht ein Tool welches sie noch lange lernen müssen bevor sie wissen wo sich die Funktionen verstecken. Unser Ziel ist es also die Akzeptanz bei den Benutzern von Anfang an hochzuhalten.
<b>NFR-03.2</b>	Hierarchie-Tiefe	Um die Anforderung erreichen zu können wollen wir die wichtigsten Funktionen der beschriebenen Use Cases innerhalb von 3 Klicks erreichbar machen. Als Startpunkt gilt hier entweder die Startseite des Use Cases oder die Startseite der Applikation.

## 7 SYSTEM MODELS

Das System ist ein Teil eines grösseren Systems, welches unter anderem auch Module für Ärzte oder für Reporting und Management enthält. Die Liste der unten aufgeführten Module ist nicht abschliessend und kann durch weitere Module anderer Gruppen ergänzt werden.

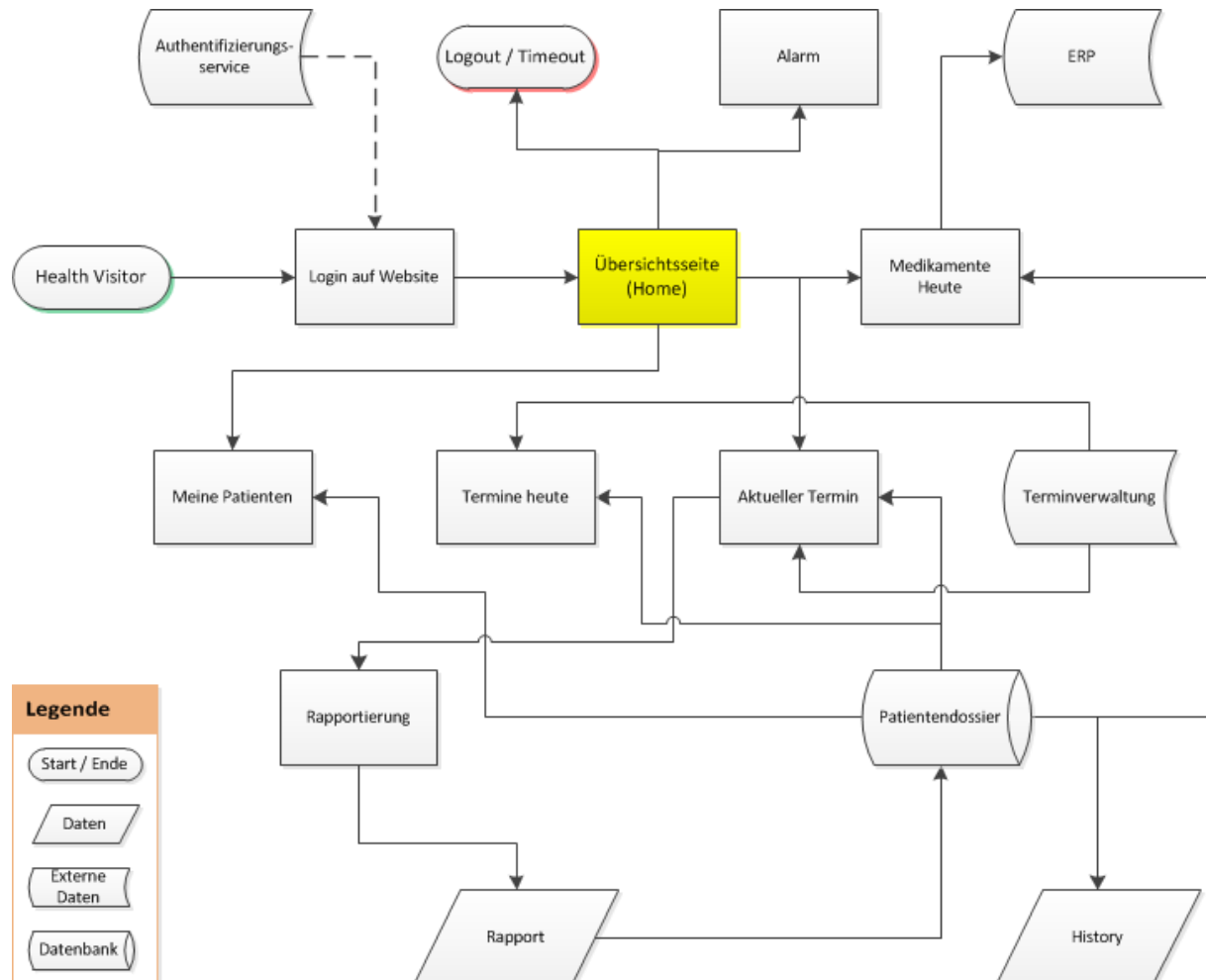


Abbildung 5: Systemmodell

Das oben abgebildete Systemmodell zeigt den Inhalt des Health Visitor Moduls.

## 7.1 BENUTZERVERWALTUNGS MODUL

Das Modul verwaltet die Benutzer des Systems. Es regelt die Zugriffsberechtigung einzelner Benutzer und Benutzergruppen. Die Benutzerverwaltung beinhaltet ebenfalls die Login- und Authentifizierungsmechanismen.

## 7.2 STAMMDATEN MODUL

Beinhaltet alle Daten zum Patienten. Die Daten können folgende Punkte umfassen:

- Personalien
- Versicherungsinformationen
- Buchhalterische Informationen (z.B. abgelegte Rechnungen)

## 7.3 PATIENTENDOSSIER MODUL

Das Patientendossier beinhaltet alle von den behandelnden Ärzten hinzugefügten Berichte, sowie seine detaillierte Krankheitsgeschichte.

## 7.4 TERMINPLANER MODUL

Das Modul bietet an Termine eines Patienten zu erfassen und diese Termine direkt einem behandelnden Arzt, oder Health Visitor zuzuweisen.

## 7.5 HEALTH VISITORS MODUL

Das Modul bietet eine Terminübersicht für die Health Visitors an. Zudem können die Termine im Detail angesehen werden. Die detaillierten Termine beinhalten das Patientendossier, Checklisten, sowie eine Rapportierungsfunktion. Unter anderem wird auch eine Notruf-Funktion angeboten.

## 8 SYSTEM EVOLUTION

### 8.1 ALLGEMEIN

Das System soll «State of the Art» umgesetzt werden. Das ermöglicht eine gute Wartbarkeit und für eine längere Betriebsdauer des Systems. Da die Umsetzung mit Scrum erfolgt, ist es möglich dynamisch immer wieder Neuerungen zu deployen und auch dynamisch auf den Kunden zu reagieren.

### 8.2 NEUE BENUTZERGRUPPE PATIENT

Anfangs arbeiten nur die Health Visitors mit dem System. In späteren Versionen kann man auch dem Patienten die Möglichkeit / Berechtigung geben, damit er z.B. seine Medikation und Termine selbst eintragen kann. Der Zugriff müsste entsprechend eingeschränkt werden. Der Health Visitor kann dann die vom Patienten erfassten Daten periodisch auswerten und die Ergebnisse dem behandelnden Arzt mitteilen.

### 8.3 NATIVE APPLIKATION

Eine native Applikation bietet viel mehr Möglichkeiten um auf die Hardware zuzugreifen. So kann die Applikation direkt auf Kamera oder Mikrofon des Gerätes zugreifen. Die nativen Applikationen können auch Daten lokal abspeichern, was dem Health Visitor ermöglicht auch offline Kundenbesuche zu machen und Daten später wieder in die Datenbank zu synchronisieren.

Weitere Vorteile können auch Notifications sein, welche der Health Visitor auf seinem Gerät erhält.

### 8.4 DYNAMISCHE TERMINZUORDNUNG

In Zukunft ist es vorstellbar, dynamisch auf Mitarbeiterabwesenheiten zu reagieren. Der Mitarbeiter soll sich hierzu abmelden können, und das System soll die Termine sofern möglich auf die restlichen Mitarbeiter optimal verteilen.

## 9 TESTING

Getestet wird das System in verschiedenen Phasen und unterschiedlicher tiefe.

Jedes Element und jede Funktion die ein Entwickler implementiert wird von ihm direkt getestet. Diese Tests beschränken sich auf einfache „Blackbox“-Tests. Es werden nur die wichtigsten Werte überprüft und grob auf die Performance geschaut.

Nach der Implementierung der jeweiligen Use Cases, werden diese durch das Entwicklerteam auf deren Integrität getestet und ob die Funktion den User Requirements entspricht.

Die nachfolgenden Tests werden durch den Kunden auf dem Integrationssystem vorgenommen, welches das Entwicklerteam dem Kunden für die Tests zur Verfügung stellt. Auf dem Integrationssystem wird mit fiktiven und anonymisierten Daten gearbeitet und dient einzig und allein dem Zweck der Funktionstests.

Danach werden diese Use Cases nach Abschluss des jeweiligen Sprints durch den vom Kunden bestimmten zuständigen Mitarbeiter getestet. Dieser achtet speziell darauf, dass das gesamte Systemverhalten demjenigen entspricht, welches der Kunde wünscht. Hierfür verwendet der zuständige Mitarbeiter die in diesem Dokument beschriebenen Use Cases. Falls er noch weitere Use Cases während den Tests findet, dokumentiert er diese und gibt diese an das Entwicklerteam weiter. Der zuständige Mitarbeiter dokumentiert seine Tests in einem dafür vorgesehenen Testprotokoll und gibt sämtliche Fehler, Fehlverhalten und allenfalls geänderte Systemanforderungen an das Entwicklerteam weiter.

Auf Basis dieser Testergebnisse kann das Entwicklerteam die Qualität des Systems am Ende erhöhen.

Wird ein Storyboard abgeschlossen, so wird zuerst wie oben beschrieben der jeweilige Use Case getestet und danach nach demselben Schema ebenfalls das Storyboard. Das fertige Storyboard wird zusätzlich zu den beschriebenen Prüfpunkten auch vertieft auf die Performance und sämtliche Schnittstellen überprüft. Das Ergebnis wird wiederum in ein dafür vorgesehenes Testprotokoll einfließen, welches dem Entwicklerteam ausgefüllt zur Verfügung steht.

Sobald das System für den Versionsstatus 1.0 bereitsteht, wird das Gesamtsystem getestet. Zuerst wird das Entwicklerteam die Tests auf die Grundfunktionalitäten vornehmen. Sprich, funktioniert die Speicherung von Daten, ist ein Login auf die Applikation möglich etc.

Anschliessend wird das System durch den zuständigen Mitarbeiter des Kunden getestet. Hierbei sind sämtliche Use Cases mit allen möglichen Sonderfällen und Ausnahmesituationen aber auch die Standardabläufe zu testen. Hierfür dienen die beschriebenen Use Cases aus diesem Dokument und die nachgereichten Use Cases aus den Tests der Use Cases durch den Mitarbeiter des Kunden. Sämtliche Ergebnisse werden in einem Abnahmeprotokoll für die Integrationsumgebung dokumentiert. Mögliche Fehler und Fehlverhalten, die eine Einführung des Systems verhindern könnten, sind umgehend durch das Entwicklerteam zu korrigieren. Kleinere Fehler können auch nach der Abnahme noch korrigiert werden. Sobald das System bereit zum Release ist und das Abnahmeprotokoll durch den Kunden unterschrieben wurde, wird die definitive Testumgebung durch das Entwicklerteam in Zusammenarbeit mit der IT des Kunden eingerichtet.



Auf dem Testsystem werden nochmals sämtliche Abnahmetests vorgenommen wie auf der Integrationsumgebung. Zusätzlich werden auch Last- und Funktionstests durch die Mitarbeitenden des Kunden vorgenommen. Alle diese Tests werden ebenfalls in einem Abnahmeprotokoll für die Testumgebung festgehalten und durch den Testverantwortlichen Mitarbeiter des Kunden unterschrieben.

Nach Absprache mit dem Kunden wird das produktive System zu einem angekündigten Zeitpunkt 1:1 der Testumgebung durch das Entwicklerteam in Zusammenarbeit mit der IT des Kunden aufgesetzt. Der verantwortliche Mitarbeiter für die Tests des Kunden wird auf der Produktionsumgebung nochmals die Abnahmetests vornehmen und unterschreibt das Abnahmeprotokoll. Danach sind die Tests abgeschlossen und das System vollständig beim Kunden im Einsatz.

Für weitere Entwicklungen/Releases am System wird das in diesem Kapitel festgehaltene Verfahren eingesetzt.

## 10 INDEX

### 10.1 ABBILDUNGSVERZEICHNIS

Abbildung 1: Use Case "Alarmierung" .....	9
Abbildung 2: Use Case "Rapportierung" .....	10
Abbildung 3: Use Case "Aktueller Termin" .....	11
Abbildung 4: Use Case "Medikamente" .....	11
Abbildung 5: Systemmodell .....	21