

Software Engineering and Design

Versionierung

#	Beschreibung	Name	Datum
0.1	Initiales Dokument	Dario Furigo	09.04.2018
0.2	Einleitung und Management Summary erstellt	Dario Furigo	11.04.2018
0.3	User requirements definition & System Evolution erstellt	Beat Schärz	11.04.2018
0.4	System requirements specification	Ohran Mujkic	13.04.2018
0.5	Testing	Janick A. Lüdi	14.04.2018
0.6	System Architecture	Nik Arm	15.04.2018
0.7	Appendices	Janick A. Lüdi	16.04.2018

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
1. Management Summary.....	4
2. Einleitung.....	5
Zielgruppe.....	5
Projektmitarbeiter.....	5
Funktionsübersicht.....	5
Patienten-Tagebuch	5
Angehörigen-Tagebuch	5
Zielvereinbarungen	6
Anbindung an Umsysteme	6
3. Requirements Specification Document.....	7
User requirements definition.....	7
Patienten-Tagebuch	7
Angehörigen-Tagebuch	7
Zielvereinbarungen	7
Non-functional requirements.....	7
System architecture	8
System requirements specification.....	9
Functional requirements.....	9
Non-Functional requirements.....	10
System models	11
System evolution.....	11
Testing	11
Komponententests.....	11
Integrationstests	11
Systemtests	12
Abnahmetest	12
4. Anhang.....	13
Datenmodell.....	13
Hardwareanforderungen	14
5. Glossar	15
6. Index.....	15

1. Management Summary

Dieses Dokument beinhaltet eine Zusammenfassung aller Anforderungen des Kunden und eine Übersicht über die Hauptfunktionen. Zudem ist ersichtlich welche technischen Voraussetzungen vorhanden sein müssen, damit das Patientenmanagementsystem erfolgreich umgesetzt werden kann.

Die Sprache dieses Dokumentes ist bewusst nicht-technisch gehalten, um den Ansprüchen der Zielgruppe gerecht zu werden.

2. Einleitung

Zielgruppe

Dieses Dokument bedient primär folgende Zielgruppen:

- Medizinisches Personal (Ärzte/Therapeuten)
- Leitende Angestellte von Spitälern und Kliniken
- Zentraler Einkauf von Spitälern und Kliniken
- Angehörige von Patienten mit Sozialer Phobie
- Entwickler/Architekten des PMS

Projektmitarbeiter

Name	Github Username	Rolle
Andreas Erb	Res13	Student
Beat Schärz	Pidu2	Student
Nik Arm	projektins	Student
Dario Furigo	VirtSyntaxError	Student
Ivo Kozina	kozina	Student
Janick Lüdi	janexander	Student
Ohran Mujkic	mujko1	Student

Tabelle 1: Projektteam Mitglieder

Funktionsübersicht

Nachfolgend werden die drei Hauptfunktionen der Software detailliert beschrieben.

Patienten-Tagebuch

Hierbei handelt es sich um ein Tagebuch zwischen Patienten und deren Ärzten oder Therapeuten. Angehörige können Einträge (mit Einverständnis des Patienten) auch lesen, aber keine eigenen Einträge und Anpassungen vornehmen.

Das Tagebuch beinhaltet sowohl Instruktionen als auch Medikamentenverschreibungen.

Angehörigen-Tagebuch

Das Angehörigen-Tagebuch sorgt für einen einfachen Informationsfluss zwischen Angehörigen und Ärzten/Therapeuten. Angehörige tragen hier z.B. alltägliche Beobachtungen über den Patienten ein um dem Arzt/Therapeuten Informationen über Fortschritte und Veränderungen im Verhalten einfach zur Verfügung zu stellen.

Ärzte/Therapeuten haben hier die Möglichkeit Angehörigen Tipps zu geben und sie zu Unterstützen.

Der Patient hat auf diese Einträge keinen Zugriff.

Zielvereinbarungen

Dieses Feature ermöglicht dem behandelnden Arzt oder Therapeuten, sowie auch den näheren Angehörigen die einfache Erfassung definierter Ziele für den Patienten.

Laufend kann der aktuelle Status des Ziels angepasst werden. Ebenfalls gibt es die Möglichkeit ein Ziel zu favorisieren und dem Patienten dazu eine Mitteilung zu senden.

Der Patient selbst darf keine Ziele eintragen, anpassen oder löschen. Er hat aber jederzeit lesenden Zugriff auf diese.

Die Ziele der Angehörigen können durch den Arzt geprüft und angepasst werden, da nur er die benötigten Kompetenzen besitzt.

Anbindung an Umsysteme

Eine Anbindung an Umsysteme ist nicht vorgesehen, einzig die anderen Teile dieser Software, die in eigenen Projekten realisiert werden, sind als Schnittstellen vorhanden.

3. Requirements Specification Document

User requirements definition

Insgesamt gibt es drei grosse Services, welche wir mit dem System unseren Usern zur Verfügung stellen.

Patienten-Tagebuch

Das Tagebuch richtet sich sowohl an Ärzte, Patienten als auch an die Angehörigen.

Änderungen im Tagebuch dürfen allerdings nur Patienten und Ärzte vornehmen, Angehörige dürfen mit Einverständnis des Patienten die Einträge aber lesen.

Inhalt des Tagebuchs sind einerseits Instruktionen welche der Arzt dem Patienten gibt und andererseits eventuell anfällige Medikamente und deren Dosierung.

Dem Patienten (und Angehörigen) stehen so jederzeit die gegebenen Instruktionen und die Medikamenteninformationen zur Verfügung. Der Arzt kann sich mit dem Tagebuch auf die Behandlung vorbereiten und den Fortschritt des Patienten erkennen.

Angehörigen-Tagebuch

Dieses Tagebuch richtet sich einzig an die Angehörigen und Ärzte. Patienten haben hier keinen Zugriff darauf.

Das Angehörigen-Tagebuch dient vor allem dazu einen einfachen Informationsfluss zwischen den Angehörigen und Ärzten zu gewährleisten. Beispiele für Einträge wären hier erkannte Verhaltensänderungen des Patienten, Fortschritte oder einfach alltägliche Beobachtungen.

Auch Ärzte können Einträge vornehmen um die Angehörigen im Alltag mit Ratschlag zur Seite zu stehen.

Zielvereinbarungen

Mit dem Zielvereinbarungssystem wird den Usern, in diesem Fall den Ärzten, Patienten und Angehörigen, eine einfache Möglichkeit zur Verfügung gestellt, Ziele mit dem Patienten zu erfassen und deren Status aktuell zu halten.

So kann z.B. der Status eines Ziels laufend angepasst oder auch einzelne Ziele priorisiert und dem Patienten sogar eine dazugehörige Mitteilung gesendet werden.

Der Patient hat hier keine Möglichkeit Änderungen vorzunehmen, kann aber jederzeit lesend aufs System zugreifen.

Ziele setzen dürfen sowohl Angehörige als auch Ärzte. Dies wird aber klar getrennt, da die Meinung des Arztes natürlich ausschlaggebender ist.

Non-functional requirements

System

Beim System wird es sich um eine Web-App handeln.

Verfügbarkeit

Diese soll jederzeit, bis auf eventuell anfallige Wartungsarbeiten, zur Verfügung stehen.

Usability

Das System soll einfach zu bedienen sein, damit keine Schulung für die Benutzer notwendig ist.

Space

Da es sich bei den Informationen im System im Prinzip nur um Text handeln wird, wird es nicht sehr viel Platz in Anspruch nehmen. Bei vielen Usern kann sich über längere Zeit natürlich trotzdem eine grosse Menge an Informationen sammeln, weswegen in diesem Fall über eine Bereinigung älterer Einträge oder Archivierung derer auf ein externes System nachgedacht werden kann.

Law

Der Datenschutz soll sich am schweizerischen Datenschutzgesetztes über vertrauliche persönliche Informationen orientieren.

Weiter soll das System auch den Grundlagen des schweizerischen Gesundheitsgesetzes entsprechen.

Development

Als Programmiersprache soll Java verwendet werden, da jedes Projektmitglied bereits Erfahrungen mit dieser Programmiersprache gemacht hat.

System architecture

Die Applikation wird als Webapplikation umgesetzt. Wir unterscheiden zwischen Backend und Frontend der Applikation.

Backend

Die Applikation wird auf einem Server in der Cloud installiert.

Die Datenverarbeitung und auswertung findet auf dem Server statt. Die Daten werden in einer Datenbank gespeichert auf welche die Applikation zugriff hat. Die Programm Logik wird in Java umgesetzt. Die Wartung der Applikation und des Servers wird von uns übernommen. Dementsprechend hat der Client kein Aufwand im bereich Wartung und Backup.

Frontend

Die Webapplikation ist via Webbrowser aufrufbar.

Nach einem erfolgreichem Login werden hier die Daten angezeigt und können bearbeitet werden. Es wird keine lokale Installation benötigt, der Client braucht nur einen modernen Webbrowser. Dies hat den vorteil, dass auch mobile Geräte auf die Webapplikation zugreifen können. Voraussetzung ist ein Responsive Design der Webapplikation, damit auf allen Geräten auch alles optimal dargestellt wird.

System requirements specification

Functional requirements

Development

Die Software wird mit Java Version 8, welche unter einer GNU Lizenz steht entwickelt. Als Framework wird Vaadin verwendet, ein Tool für „rich internet applications“. Die Lizenz für Vaadin ist eine Apache License 2.0.

Data

Die Applikation soll in der Lage sein mit einem Datenbanksystem zu kommunizieren. Hierbei wird auf eine SQL-Datenbank gesetzt. Die Daten sollen im System erstellt, bearbeitet, abgerufen sowie gelöscht werden können.

Responsive Design

Um eine gute UX zu gewährleisten soll die App responsive sein. Das bedeutet in erster Linie, dass auf allen Gerätgrößen eine optimale Nutzung der App ermöglicht werden soll.

Roles

Es soll ein Berechtigungssystem geben, welches von bestimmten Benutzer angepasst werden. Ausserdem kürzt oder erweitert dieses Berechtigungskonzept die funktionalen Möglichkeiten des Benutzers. Ein nicht-bevormundeter Patient (Soziale Phobie) soll die Möglichkeit erhalten die Berechtigungen der Angehörigen einzuschränken.

Patienten-Tagebuch

Die Software ermöglicht einen einfachen Austausch zwischen dem Klienten und dem Health Professional. Der Patient kann jeden Tag max. einen Tagebuch-Eintrag erfassen und diesen mit seinem Betreuer sowie den Angehörigen teilen. Falls der Patient nicht bevormundet ist, kann er den Angehörigen Leserechte gewähren und wieder entziehen. Durch eine Kommentar-Funktion soll ähnlich wie bei einem Forum-Thread die Möglichkeit bestehen, eine Diskussion zu führen.

Die Einträge sind, dank dem Berechtigungskonzept, nur vom Health Professional und Erkrankten bearbeitbar. Die Speicherung der Einträge erfolgt über die Datenbank.

Damit jeden Tag ein Eintrag erfasst wird, werden die Angehörigen per Mail informiert.

Angehörigen-Tagebuch

Die Software ermöglicht einen einfachen Austausch zwischen den Angehörigen und dem Health Professional. Die Angehörigen können jeden Tag max. einen Tagebuch-Eintrag erfassen und mit dem Health Professional teilen. Durch eine Kommentar-Funktion soll ähnlich wie bei einem Forum-Thread die Möglichkeit bestehen eine Diskussion zu führen.

Die Einträge sind, dank dem Berechtigungskonzept, nur vom Health Professional und den Angehörigen bearbeitbar. Die Speicherung der Einträge erfolgt über die Datenbank.

Damit jeden Tag ein Eintrag erfasst wird, werden die Angehörigen per Mail informiert.

Zielsetzung

Das Berechtigungskonzept ermöglicht es, dass Angehörige sowie Health Professionals in der Lage sind Ziele für den Erkrankten zu setzen. Die Erkrankten haben auf diese Maskenansicht nur Leserechte.

Die Ziele sind auf der Datenbank gespeichert und werden von dort aus auch aufgerufen.

Damit der Erkrankte das Ziel nicht vergisst, wird er anhand eines Reminder's (Mail) auf das aktuelle Ziel aufmerksam gemacht.

Die Ziele der Angehörigen können durch die Health Professionals angepasst werden, da nur diese die benötigten Kompetenzen besitzen.

Non-Functional requirements

Performance

Die Applikation sollte eine schnelle Reaktionszeit haben. Dabei sollten die Daten auch möglichst schnell geladen werden, so dass für den Benutzer kaum Wartezeiten entstehen.

Privacy

Sicherheitslücken, besonders betreffend Personendaten, dürfen in der Applikation nicht vorhanden sein. Die Software muss alle Regelungen des Datenschutzgesetzes einhalten.

Design

Für alle Benutzergruppen sollte das Layout der Applikation ansprechend sein. Die Verwendung einer solchen Software sollte selbsterklärend sowie einfach sein.

System models

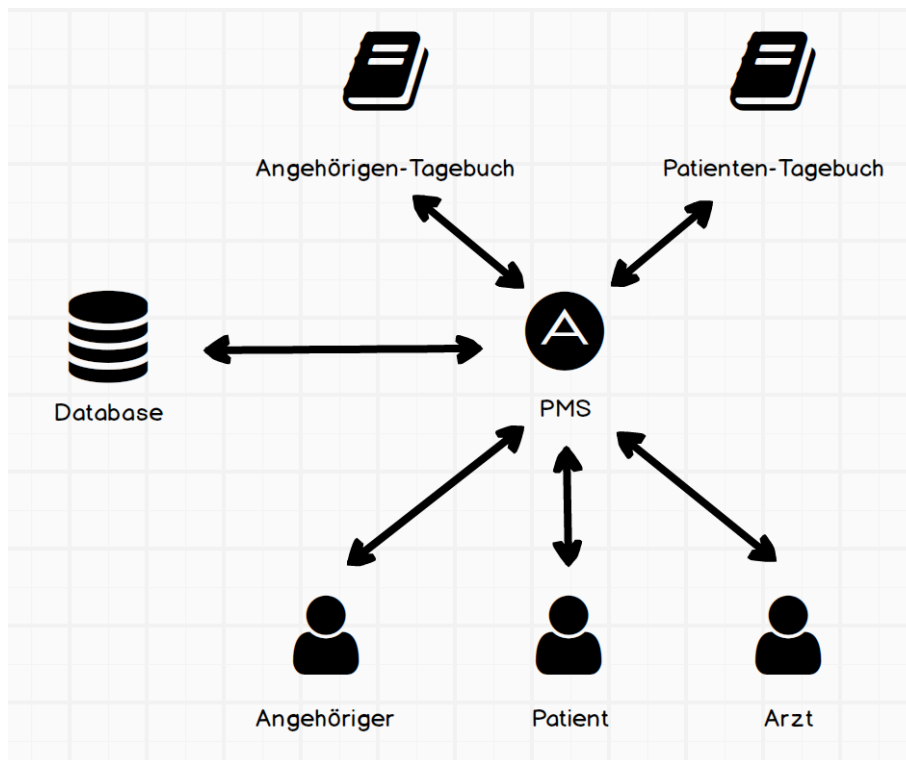


Abbildung 1: System model

System evolution

Für die System-Evolution kann man sich an den 8 Lehmans Gesetzen orientieren.

Die beiden wichtigsten Teile welche man diesen entnehmen kann sind, dass das System kontinuierlich an Neuerungen angepasst werden muss, da es ansonsten an Effektivität verliert und dass verhindert werden sollte, dass Änderungen der Software zu Unübersichtlichkeit und extremer Komplexität führen.

Die Software soll von Anfang an so aufgebaut werden, dass das Hinzufügen neuer Funktionalitäten auf Grund von Userwünschen kein Problem darstellt.

Testing

Komponententests

Bereits während der Entwicklungsphase wird die Software regelmässig durch die Entwickler auf deren Funktionsfähigkeit getestet. Mittels Komponententests («Units-Tests») werden die einzelnen Einheiten der Software getestet. Mögliche Fehler werden mit «Debugging» ausfindig gemacht und unmittelbar behoben.

Integrationstests

Abhängig von der gewählten Systemarchitektur werden die Schnittstellen der einzelnen Systemkomponenten mithilfe eines geeigneten Tools getestet. Es werden verschiedene Schnittstellenabfragen gemacht, sowohl «Happy Cases» wie auch «Negativ Cases», und die Rückgabewerte mit den zu erwartenden Ergebnissen verglichen.

Systemtests

Am Ende der Entwicklungsphase bzw. bereits während dieser, werden die einzelnen funktionalen Anforderungen durch manuelles sowie automatisiertes GUI-Testing geprüft. Zu deren Durchführung werden auf einer voraussichtlich nicht-produktiven Umgebung Testdaten generiert.

Abnahmetest

Nach Beendigung der Entwicklungsphase wird die Software auf einer produktiven Umgebung von einer End-User-Nahen Testgruppe / Person zusammen mit dem Projektteam auf Herz und Nieren geprüft. Es handelt sich hierbei um eine Art «Black-Box-Testing», da die miteinbezogenen externe/-n Testperson/-en über keinerlei Hintergrundwissen zum «Code» verfügen. Es werden hierfür ebenfalls Testdaten («Echtdaten») generiert.

4. Anhang

Datenmodell

Zur Umsetzung der Applikation und insbesondere der Datenbank wird ein Datenmodell benötigt. Nachfolgend ein grobes Entity-Relationship-Diagramm zu den Hauptfunktionalitäten der Anwendung.

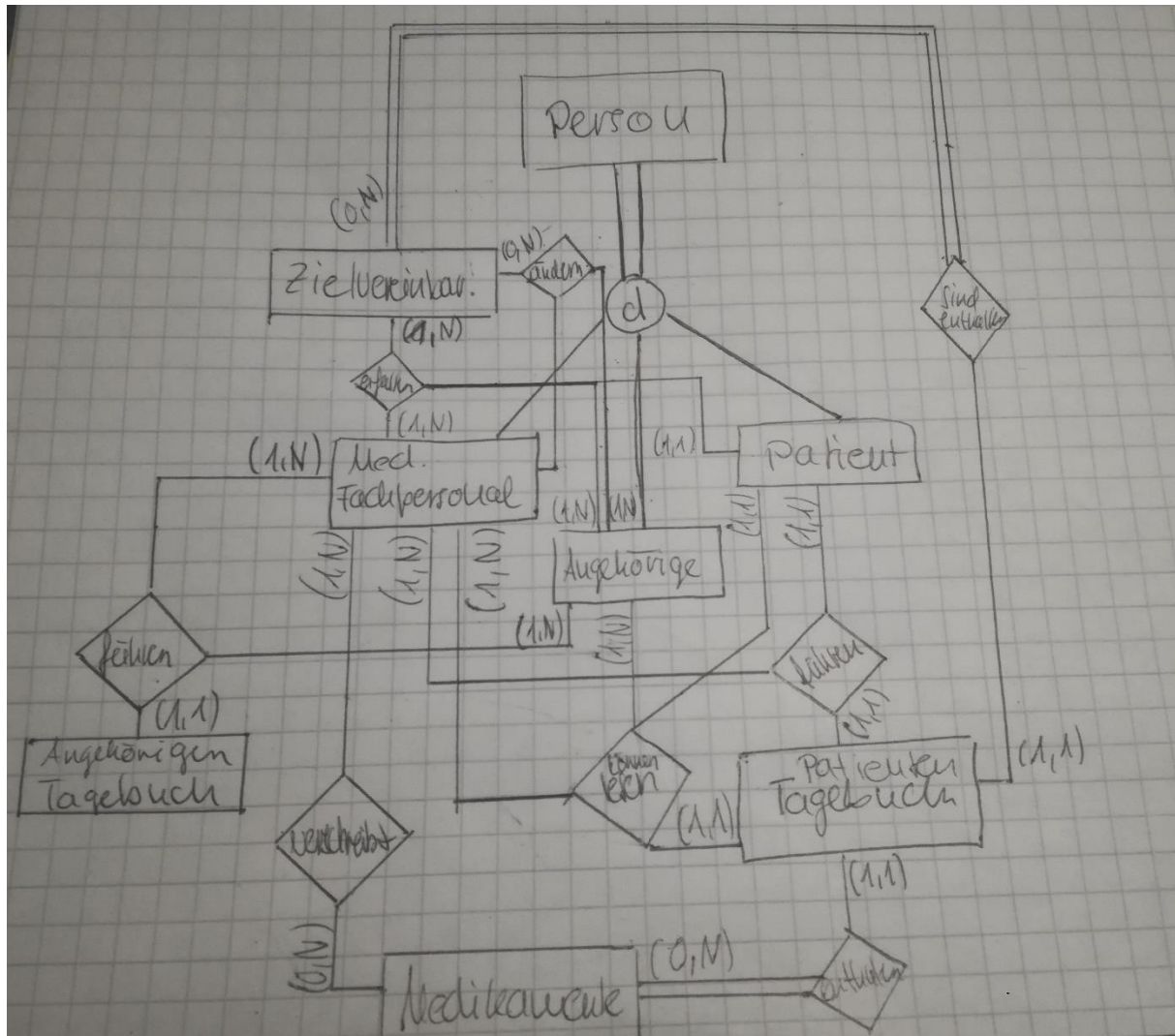


Abbildung 2: Entity-Relationship-Diagramm

Hardwareanforderungen

In der nachfolgenden Tabelle sind die grobe Anforderungen der Systems aufgelistet:

Hardwarekomponente	Anforderung
Cloud Server (Backend)	<ul style="list-style-type: none">• Open Source OS (free)• Speicherkapazität, Rechenleistung etc. wurden nicht spezifiziert vom Auftraggeber• Wartbar durch Zugriff innerhalb des BFH Netzes (evtl. auch ausserhalb)
Clientseitig (Frontend)	<ul style="list-style-type: none">• Desktop PC / Laptop mit gängigen OS's und Browsern (auch für die Entwicklung der Applikation)• Evtl. responsives Design für mobile Endgeräte mit gängigen OS

Tabelle 2: Hardwareanforderungen

5. Glossar

Begriff	Definition
PMS	patient management System / Patientenmanagementsystem
MHC	Mental Health Care / Psychische Gesundheitsfürsorge
RBAC	Role Based Access Control / Rollenbasierte Zugangskontrolle

Tabelle 3: Glossar

6. Index

Verzeichnis der enthaltenen Abbildungen:

Abbildung 1: System model.....	11
Abbildung 2: Entity-Relationship-Diagramm.....	13