

Software Engineering

CS2 Task 2

Requirements

App für Ärzte welche Patienten mit
Zwangsstörungen behandeln

Modul

BTI7081 - Software Engineering and Design

Durchführung

FS2019 Klasse p

Team

Red

Blaser Steve
Haug Sophie
Hutzli Marc
Kramer Ueli
Meyer Cyrill
Schwärzler Sascha

2019-04-08

Preface	3
Versionierung	3
Introduction	3
Glossary	3
User Requirements Definition	4
Use-Case Diagram	4
Use-Cases	5
Use-Case Scenarios	5
02 - Erfassung und Anpassung von Patienten	5
05 - Gruppensitzungen erfassen	7
Activity Diagrams	9
Functional Requirements	10
System Architecture	11
System Requirements	12
Functional System Requirements	12
Non-Functional System Requirements	14
System Model	15
System Evolution	16
Testing	16
Four-Eyes Code Reviews	16
Unit Tests	16
Integration Tests	16
Manuelle UI Tests	17
Appendices	17
Voraussetzungen Client	17
Voraussetzungen Server	17
Hardwarevoraussetzungen	17
Use-Case-Diagramm Extended	18

Preface

Dieses Dokument dient dem Projektteam zur Analyse der Anforderungen für das PMS. Zusammen mit dem Kunden wird man prüfen ob die Benutzeranforderungen richtig verstanden wurden.

Versionierung

Version	Status	Beschreibung	Autoren	Datum
0.1	Entwurf	Initiales Dokument erstellt	Ueli Kramer Cyrill Meyer Marc Hutzli	01.04.19
1.0	Freigegeben	Alle erarbeiteten Modelle und Analysen wurden zusammengeführt.	Ueli Kramer Cyrill Meyer Marc Hutzli Sophie Haug Sascha Schwärzler Steve Blaser	08.04.19

Introduction

Beim Patienten Management System (MHC-PMS) handelt es sich um eine Applikation zur Verwaltung von Patienten mit Zwangsstörungen. Der Arzt nutzt dieses System zur Verwaltung von Terminen und Patienten, sowie zur Protokollierung der gehaltenen Sitzungen.

Glossary

Begriff	Definition
PMS	Patienten Management System
MHC	Mental Health Care
responsive	An die Eigenschaften des jeweiligen Endgerätes zugeschnittenes Web Layout und Design.
DB	Datenbank
Branch	Entwicklungszweig in einem Versionsverwaltungssystem wie GIT. Ermöglicht die Abzweigung und parallele Weiterentwicklung.

User Requirements Definition

Use-Case Diagram

Aufgrund der mehreren Iterationsschritte und der Diskussionen hat das Use-Case Diagramm mehrmals geändert. Zwecks der Übersichtlichkeit und Verständlichkeit sind im nachfolgenden Diagramm lediglich die 5 wichtigen Use-Cases aufgelistet, welche auch umgesetzt werden und in direkter Verbindung mit den Aktoren stehen. Im [Appendices](#) ist ausserdem die etwas umfassendere Version [Use-Case-Diagramm Extended](#). Dies kann nützlich sein, um die Gedankengänge zu einem späteren Zeitpunkt nachvollziehen zu können. Es ist absichtlich im Anhang, da die Spezifikation zu weit ginge und zu stark einschränken würde. Wichtig ist primär die Umsetzung der nachfolgenden Use-Cases. Die Abhängigkeit des Patienten zu "Verlauf von Expositionstherapie verfolgen" ist noch etwas unklar. Es sind Szenarien möglich, dass der Therapeut dem Patienten den Verlauf aufzeigt oder der Patient selber die Möglichkeit hat, per Computer oder Smartphone Einsicht zu nehmen. Um überhaupt einen Verlauf verfolgen zu können, ist es wichtig möglichst viele Daten regelmässig zu erfassen. Die Erstellung einer Smartphone App ist aber nicht Bestandteil dieser Arbeit und die Erfassung der Daten wird weggelassen. Stattdessen werden die Beispieldaten manuell, ohne grafische Oberfläche, von uns erzeugt und Wert auf den Verlauf gelegt.

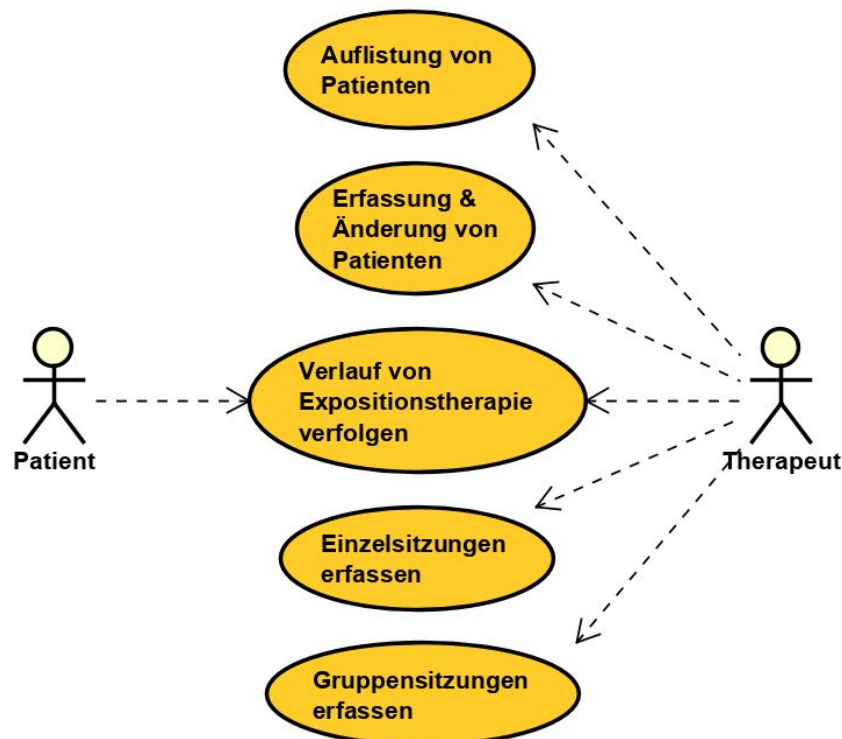


Abbildung 1: Use-Case Diagram

Use-Cases

Nr.	Beschreibung
01	Auflistung von Patienten
02	Erfassung & Anpassung von Patienten
03	Verlauf von Expositionstherapie verfolgen
04	Einzelsitzungen erfassen
05	Gruppensitzungen erfassen

Use-Case Scenarios

02 - Erfassung und Anpassung von Patienten

Nr. und Name	02 - Erfassung und Anpassung von Patienten
Szenario	Ein neuer Patient meldet sich beim Therapeuten für die Behandlung seiner Zwangsstörung oder Patientendaten eines bestehenden Patienten ändern sich und müssen angepasst werden.
Kurzbeschreibung	Der Arzt wählt auf der Patientenübersicht den Button fürs Erfassen eines Patienten oder wählt einen bestehenden Patienten aus. Anschliessend erfasst er alle Daten oder passt Daten an und speichert die gemachten Änderungen durch den Klick auf den Speichern Button.
Beteiligte Akteure	Arzt, System
Auslöser / Vorbedingung	<ul style="list-style-type: none"> - Neuer Patient angenommen - Patientendaten sind inkorrekt - Patientendaten haben sich geändert und sind noch nicht im System übernommen
Ergebnisse / Nachbedingung	<ul style="list-style-type: none"> - Der neue Patient ist mit allen Patientendaten erfasst - Die Patientendaten sind korrigiert / angepasst und im System gespeichert

Ablauf:

Nr.	Wer	Was
2.0	Arzt	Optionen: <ul style="list-style-type: none"> - Neuer Patient erfassen - Bestehenden Patient bearbeiten
2.1	System	Variante: Lädt Daten
2.2	System	Stellt die Patientendaten in einem Formular dar
2.3	Arzt	Füllt die Patientendaten aus
2.4	Arzt	Speicherung auslösen
2.5	System	Prüft die Pflichtfelder und Plausibilisierung der Benutzereingaben
2.6	System	Ausnahme: Daten nicht i.O.
2.7	System	Aufbereitung der Eingaben für die Speicherung
2.8	System	Speicherung der Daten in die Datenbank
2.9	System	Laden der Bearbeitungsseite des soeben gespeicherten Patienten

Ausnahmen, Varianten:

Nr.	Wer	Was
2.1		Lädt Daten
2.1.1	System	Bestehender Patient: Bereits gespeicherte Daten werden geladen
2.1.2	System	Neuer Patient: neuer Patient wird mit Standard Werten vorbereitet
2.6		Daten nicht i.O.
2.6.1	System	Fehlermeldung mit Angabe der nicht korrekten Daten, Abbruch des Vorganges
2.6.2	Arzt	Korrektur der Daten
2.6.3	Arzt	Erneut Speicherung auslösen => 2.4

05 - Gruppensitzungen erfassen

Nr. und Name	05 - Gruppensitzungen erfassen
Szenario	Eine Gruppensitzung mit mind. zwei verschiedenen Patienten wurde durchgeführt und muss im System erfasst werden für die Verrechnung und Dokumentation.
Kurzbeschreibung	Der Arzt wählt die Aktion "Erfassung einer Gruppensitzung" und wählt dort alle betroffenen Patienten aus. Er kann Sitzungsnotizen hinzufügen und bei jeder Notiz die Sichtbarkeit wählen. Zudem kann gewählt werden ob die Notiz für alle Patienten gilt oder nur für einzelne Patienten. Beim Speichern werden die Notizen in den dazugehörigen Patientenakten abgelegt.
Beteiligte Akteure	Arzt, System
Auslöser / Vorbedingung	<ul style="list-style-type: none"> - Gruppensitzung abgeschlossen und noch nicht im System erfasst
Ergebnisse / Nachbedingung	<ul style="list-style-type: none"> - Die Gruppensitzung ist erfasst - Die erfassten Notizen sind bei den Patienten ersichtlich

Ablauf:

Nr.	Wer	Was
5.0	Arzt	Option: Neue Gruppensitzung erfassen
5.1	System	Erfassungsmaske geladen
5.2	Arzt	Erfasst alle Angaben zur Gruppensitzung
5.3	Arzt	Speicherung auslösen
5.4	System	Prüft die Pflichtfelder und Plausibilisierung der Benutzereingaben
5.5	System	Ausnahme: Daten nicht i.O.
5.6	System	Speichert die Gruppensitzung in der Datenbank

5.7	System	Speichert die Notizen bei den Patientendossiers in der Datenbank
5.8	System	Laden der Therapieübersichtseite mit einer Erfolgsmeldung

Ausnahmen, Varianten:

Nr.	Wer	Was
5.5		Daten nicht i.O.
5.5.1	System	Fehlermeldung mit Angabe der nicht korrekten Daten, Abbruch des Vorganges
5.5.2	Arzt	Korrektur der Daten
5.5.3	Arzt	Erneut Speicherung auslösen => 5.3

Activity Diagrams

02-Erfassung und Anpassung von Patienten

04-Einzelsitzungen erfassen

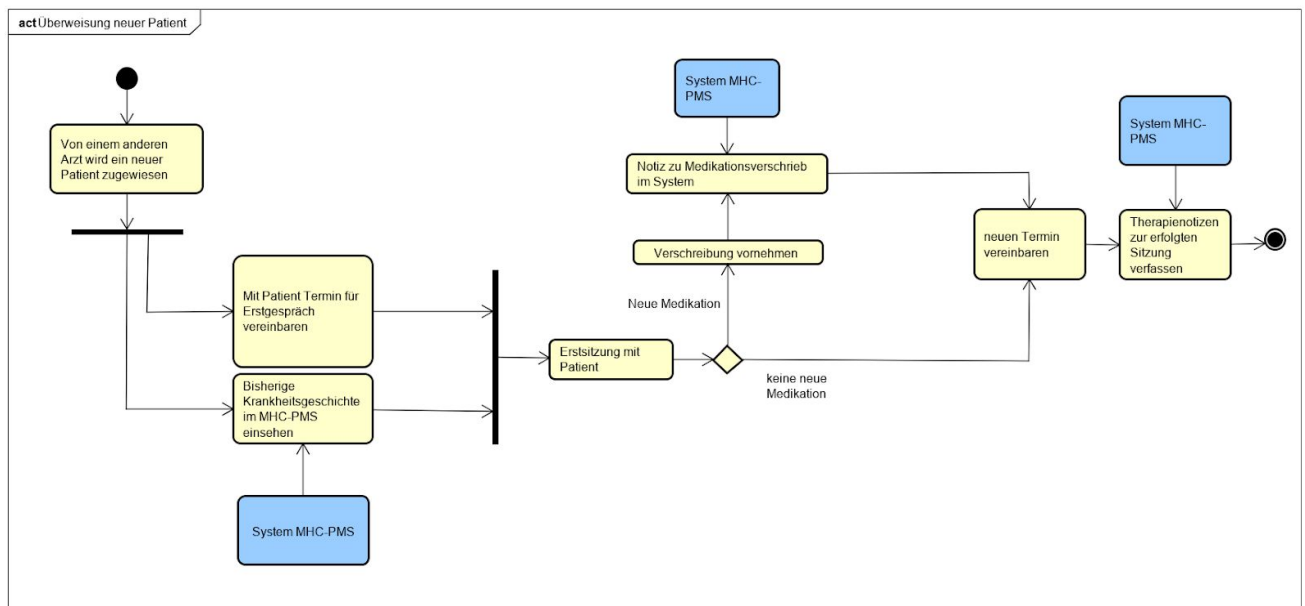


Abbildung 2: Activity Diagram zu Use-Cases 02 und 04

03-Verlauf von Expositionstherapie verfolgen

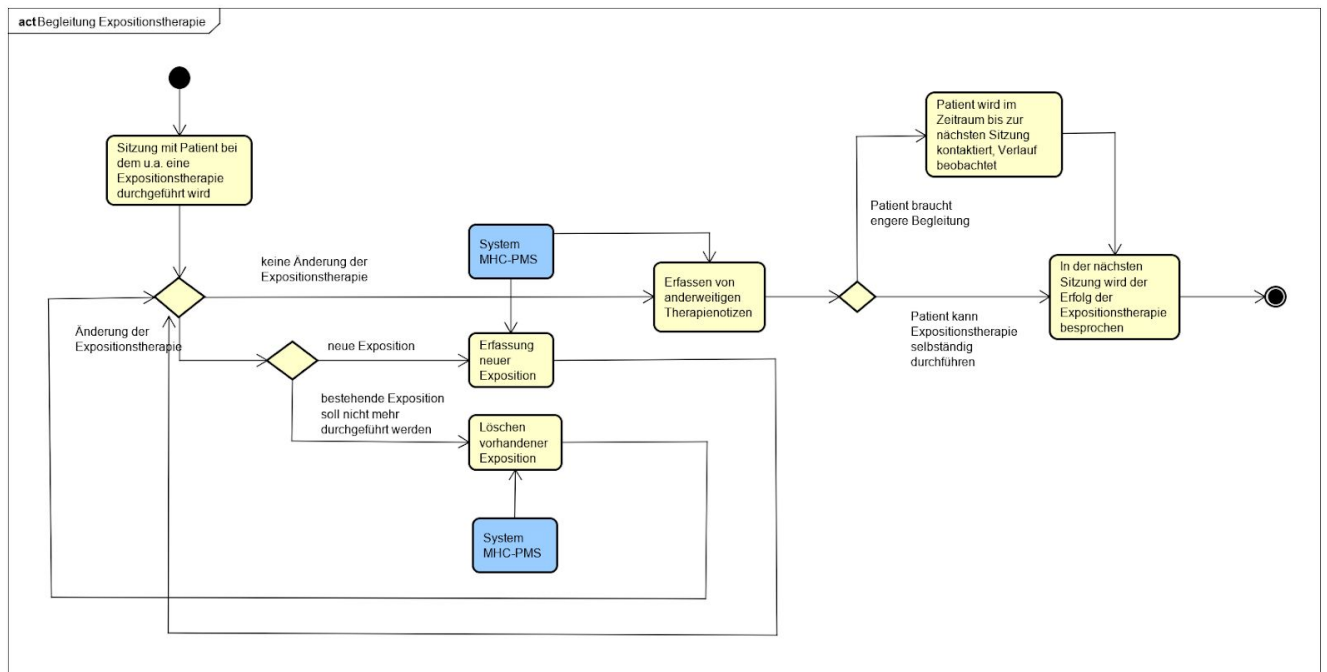


Abbildung 3: Activity Diagram zu Use-Case 03

Functional Requirements

Nr.	Anforderung	Basierend auf Use-Case
01	Der Arzt muss eine Liste über alle Patienten sehen können.	01 - Auflistung von Patienten
02	Der Arzt muss anhand verschiedener Patientendaten in der Liste suchen können.	01 - Auflistung von Patienten
03	Der Fortschritt der Expositionstherapie soll verfolgt werden können. Dazu gehört die Analyse von Daten abhängig von der Zeit.	01 - Auflistung von Patienten 02 - Erfassung und Anpassung von Patienten 04 - Einzelsitzungen erfassen 05 - Gruppensitzungen erfassen
04	Der Arzt kann einen neuen Patienten erfassen.	02 - Erfassung und Anpassung von Patienten
05	Der Arzt kann Daten eines bestehenden Patienten bearbeiten.	02 - Erfassung und Anpassung von Patienten
06	Der Arzt kann anhand eines Patienten und Datumszeitraum nach Therapiesitzungen suchen.	03 - Verlauf von Expositionstherapie verfolgen
07	Der Arzt kann eine Einzelsitzung zu einem Patientendossier hinzufügen mit Notizen und deren Sichtbarkeit.	04 - Einzelsitzungen erfassen
08	Der Arzt kann in einer einzigen Ansicht eine Gruppensitzung mit mehreren Patienten erfassen und entsprechend Notizen mit deren Sichtbarkeit hinzufügen.	05 - Gruppensitzungen erfassen

System Architecture

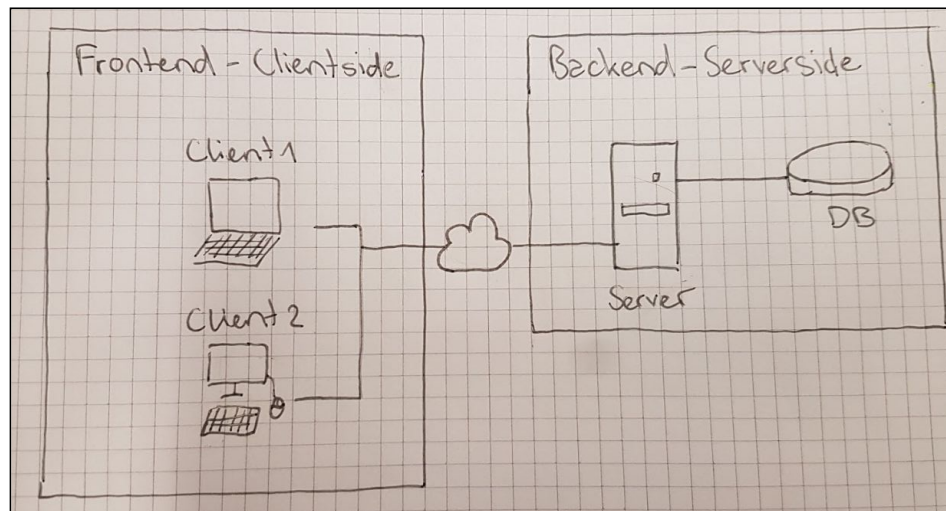


Abbildung 4: Systemarchitektur

Die Systemlandschaft beinhaltet eine Frontend-Clientside sowohl auch eine Backend-Serverside. Der Benutzer interagiert mit der Maus und Tastatur mit dem Frontend. Anfragen werden übers Frontend an den Server im Backend geschickt, welcher diese auswertet und entsprechend Daten aus der Datenbank zurückschickt.

Datenbank (DB)

Die Datenbank wird zur Speicherung aller Informationen (Patienten, Benutzer, Sitzungen) verwendet. Die Datenbank unterliegt hohen Sicherheitsanforderungen, da die gespeicherten Daten als besonders schützenswert gelten.

Server

Der Server ist ein Java Web-Server, der die Schnittstelle von Benutzer (Client) und Daten (Datenbank) schafft. Zudem ist der Server für die Darstellung und somit Auslieferung von HTML, CSS und statischen Dateien verantwortlich.

Clients

Die Clients benötigen für den Zugriff nur einen Webbrowser in der aktuellsten Version und eine aufrechte Internetverbindung.

Migration

Wir gehen davon aus, dass die bestehende Softwarelösung komplett abgelöst wird und die Daten vollständig in die neue Datenbank migriert werden. Dies wird den Aufbau der Applikation massiv vereinfachen. Im produktiven Umfeld wäre dies sicherlich ein Punkt, welcher detailliert geprüft werden müsste.

System Requirements

Functional System Requirements

Nr.	Beschreibung	Use Cases
01	Suchen nach Vor- / Nachname, Strasse, PLZ, Ort, Alter.	01 - Auflistung von Patienten
02	Sortieren nach Vorname oder Nachname.	01 - Auflistung von Patienten
03	Durch das Klicken auf einen Patienten öffnet sich die Maske zum Anpassen eines Patienten.	01 - Auflistung von Patienten 02 - Erfassung und Anpassung von Patienten
04	Ein Patient enthält folgende Attribute: <ul style="list-style-type: none"> - Geschlecht (Pflichtfeld, Standard: m) - Vor- und Nachname (Pflichtfelder) - Strasse, PLZ, Ort (Pflichtfelder) - Alter (Pflichtfeld) - Sozialversicherungsnummer (Optional) - Vormund (Optional) - Allergien (Optional) 	02 - Erfassung und Anpassung von Patienten
05	Die Eingaben werden auf Plausibilität geprüft: <ul style="list-style-type: none"> - Geschlecht: "m" oder "w" - Vor- und Nachname: Buchstaben - Alter: zwischen 0 - 200 - Sozialversicherungsnummer: gem. Kapitel 1.2.1.2 des Dokuments: https://www.admin.ch/opc/de/federal-gazette/2006/501.pdf 	02 - Erfassung und Anpassung von Patienten
06	Verlauf von Expositionstherapie verfolgen nach Datum / Patientennamen (Vor- / Nachname).	03 - Verlauf von Expositionstherapie verfolgen
07	Die Therapiesitzungen werden in einer Liste angezeigt mit den dazugehörigen Daten (Datum, Ort, Notiz, Erfasser).	03 - Verlauf von Expositionstherapie verfolgen
08	Zur Erfassung einer Einzelsitzung gehören folgende Attribute: <ul style="list-style-type: none"> - Patient (Pflichtfeld, Standard: aktiver Patient) - Datum (Pflichtfeld, Standard: aktueller Tag) - Dauer (Optional) 	04 - Einzelsitzungen erfassen

	<ul style="list-style-type: none"> - Ort (Optional) - Notiz (Optional) - Notiz Sichtbarkeit (Pflichtfeld, Standard: public) <ul style="list-style-type: none"> - Öffentliche Notizen (public) - Vertrauliche Notizen (private) 	
09	Geschützte Notizen können andere Fachkräfte einsehen.	02 - Erfassung und Anpassung von Patienten 03 - Verlauf von Expositionstherapie verfolgen 04 - Einzelsitzungen erfassen 05 - Gruppensitzungen erfassen
10	Vertrauliche Notizen können nur vom Erfasser gesehen werden. Diese werden mit *vertraulich* gekennzeichnet.	02 - Erfassung und Anpassung von Patienten 03 - Verlauf von Expositionstherapie verfolgen 04 - Einzelsitzungen erfassen 05 - Gruppensitzungen erfassen
11	Beim Speichern wird geprüft ob alle Pflichtfelder ausgefüllt sind.	02 - Erfassung und Anpassung von Patienten 04 - Einzelsitzungen erfassen 05 - Gruppensitzungen erfassen
12	Die Eingaben werden auf Plausibilität geprüft: <ul style="list-style-type: none"> - Patient: Existierender Patient - Datum: Gültiges Datum (DD.MM.YYYY, vor dem aktuellen Tag) - Dauer: > 0 Min. - Ort: Buchstaben - Notiz: Keine Prüfung - Notiz Sichtbarkeit: "public" oder "private" 	04 - Einzelsitzungen erfassen
13	Zur Erfassung einer Gruppensitzung gehören folgende Attribute: <ul style="list-style-type: none"> - Patienten (Pflichtfeld) - Datum, Dauer, Ort (analog Einzelsitzung) - Notiz pro Patient oder für die ganze Gruppe (Optional) - Notiz Sichtbarkeit pro Notiz (Pflichtfeld, Standard: public) <ul style="list-style-type: none"> - Öffentliche Notizen (public) - Vertrauliche Notizen (private) 	05 - Gruppensitzungen erfassen
14	Bei Patienten erscheint ein Button zum Hinzufügen eines Patienten zur	05 - Gruppensitzungen erfassen

	Gruppensitzung. Der Klick darauf öffnet ein Fenster mit der Liste aller Patienten und deren Suche.	
--	---	--

Non-Functional System Requirements

Nr.	Type	Beschreibung
01	Performance	Jede Anfrage an den Webserver darf maximal 2 Sekunden betragen. Der Webserver empfängt Daten von der Datenbank. Die Datenbank muss demnach auch eine gute Performance aufweisen.
02	Security	Daten werden in einer Datenbank dauerhaft gespeichert. Bei einem Absturz dürfen die Daten nicht verloren gehen. Ein späterer Zugriff auf die Daten zeigt denselben Stand wie vorher.
03	Security	Ein Benutzer kann nur auf Patientendaten zugreifen, wenn dieser eingeloggt ist. Jeder unautorisierte Zugriff wird verhindert. Einzelne Datensätze (z.B. Notizen zu Sitzungen) können als vertraulich markiert werden und sind nur für den Erfasser sichtbar.
04	Regulatory	Das System erfüllt die rechtlichen Vorschriften und die Vorschriften von Swissmedic. (Out of Scope)
05	Usability	Ältere (Ü40) Psychotherapeuten können die Applikation ohne Probleme bedienen. Die Applikation stellt die regulären Arbeitsabläufe logisch dar und unterstützt die Fachpersonen und hindert nicht.
06	Development	Alle Software Änderungen werden nach dem Vier-Augen Prinzip geprüft. Jeder Pull Request wird noch von einer zweiten, anderen Person als der Entwickler, geprüft und angenommen. Erst dann wird ein Pull Request in den aktuellen Entwicklungszweig übernommen.
07	Security	Das System umfasst Benutzer mit verschiedenen Rollen. Anhand der Rollen können verschiedene Zugriffsrechte vergeben werden. (Out of Scope)

System Model

Es wird eine klare Schnittstelle definiert zwischen dem Client und dem Server. Die sicherheitsrelevanten Operationen wie Autorisierung, Validierung werden allesamt auf dem Server bewältigt. Client seitig befindet sich ein Event Handling, welches die grafischen Elemente kontrolliert.

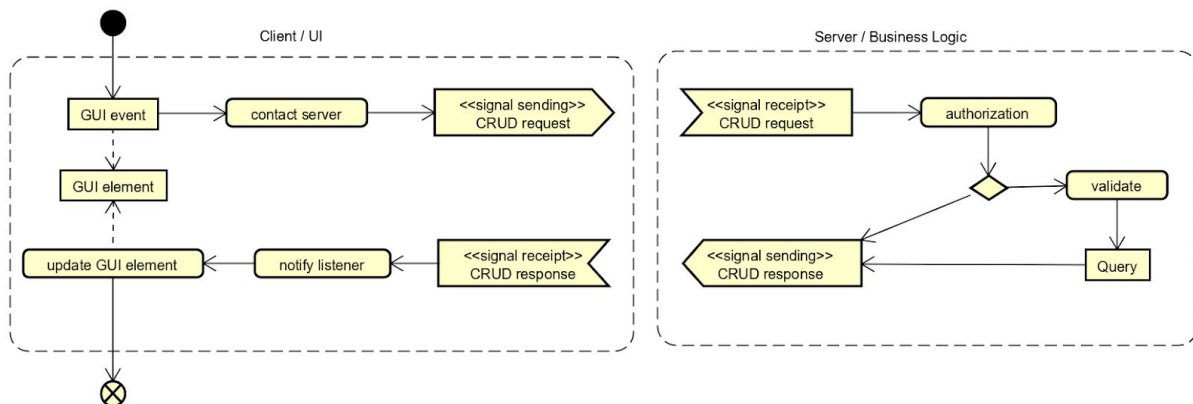


Abbildung 5: Activity Diagram zum Event Handling

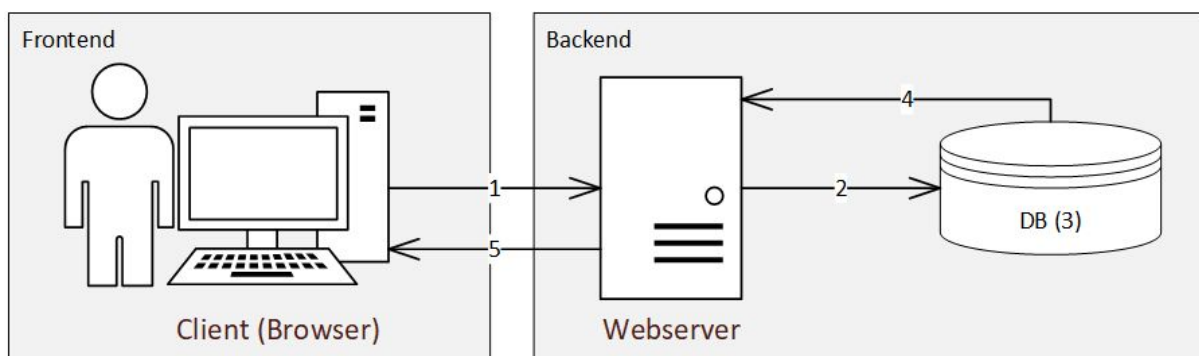


Abbildung 6: System Model

Nr.	Beschreibung
01	Der Arzt setzt auf seinem Client einen Request an den Server.
02	Der Server wertet die Anforderung aus und schickt nach entsprechend ausgeführter Business Logik eine Query an die Datenbank.
03	Die Datenbank führt das erhaltene Query aus und liefert die Daten zurück an den Server.
04	Der Server verarbeitet die erhaltenen Daten der Datenbank und schickt sie in entsprechender Form zurück auf den Client.
05	Der Client zeigt dem Arzt das Resultat der Anfrage.

System Evolution

Das MHC-PMS beinhaltet überwiegend Patientendaten sowie Informationen zu Behandlungen und Medikation. In der ersten Phase wird dafür ein Webserver mit einer Datenbank verwendet.

Die Weiterentwicklung beinhaltet regelmässige Releases, weshalb auch ein Update System implementiert wird. Durch die zunehmende Datenmenge muss zu einem späteren Zeitpunkt womöglich eine Skalierung in Betracht gezogen werden.

Die System Architektur mittels Website (Frontend) und Datenverarbeitung (Backend) bietet eine dynamische Lösung, die auf Software und Hardwareänderungen gut eingehen kann.

Testing

Um die Qualität der Software hoch zu halten und die nicht-funktionalen Anforderungen einzuhalten, werden einige Werkzeuge eingesetzt:

- Four-Eyes Code Reviews
- Unit Tests
- Integration Tests
- Manuelle UI Tests

Four-Eyes Code Reviews

Jede Weiterentwicklung wird in einem eigens dafür angelegten Branch entwickelt. Nach Abschluss der Entwicklungen erstellt der Entwickler einen Pull-Request, welcher von mindestens einer anderen Person eingehend angeschaut und geprüft wird. Wird der Pull-Request akzeptiert, kann dieser in den Haupt-Entwicklungszweig übernommen werden.

Unit Tests

Um eine möglichst hohe Code Coverage zu erlangen, werden Unit Tests geschrieben, welche automatisiert durchlaufen werden und so bereits einen grossen Teil der Applikation abdecken. Bei einem Pull-Request müssen alle Tests erfolgreich durchlaufen werden.

Integration Tests

Vor dem Release in der Produktion, werden die Entwicklungen in einem Testsystem getestet. Es werden ausschliesslich Black-Box Tests durchgeführt. Bei diesen Black-Box Tests schaut der Tester nur auf die Eingabe und die erwartete Ausgabe ohne die Kenntnis über den Code, der dahinter steckt.

Manuelle UI Tests

Diese Tests sind schwierig zu automatisieren und werden deshalb von Hand durchgeführt. Es werden in unserem Fall jetzt keine ausführlichen Testprotokolle erstellt, sondern lediglich die zentralen Elemente getestet. Z.B. Erstellung einer Person, Ändern einer Gruppensitzung. Vor dem Vertrieb (Abgabe des finalen Codes) müssen alle Punkte einmal komplett getestet werden. Die Prüfung des Verhalten bei ungültigen oder leeren Eingabewerten in den Masken wird nur durchgeführt, insofern diese Eingaben die Sicherheit gefährden.

Appendices

Voraussetzungen Client

Die Anforderungen an den Client sind minimal, da die Daten zentral in einer gesicherten Datenbank auf dem Server abgelegt werden und auch vom Server aufbereitet werden. Für die Nutzung auf dem Client (PC, Notebook, Tablet, Phone) wird lediglich ein aktueller Browser wie Chrome, Firefox oder Safari sowie eine aufrechte Internetverbindung vorausgesetzt.

Voraussetzungen Server

Die Hardwareanforderungen des Servers richten sich nach der Anzahl Benutzer. Daher ist es besonders wichtig, dass der Server erweiterbar ist.

Für die erste Phase des Projekt muss der Server in der Lage sein einen Webserver (zB Apache), Java sowie eine Datenbank (z.B. MariaDB oder MySQL) zu hosten.

Allenfalls bestehende Datenbanken alter Kundensysteme werden migriert. Daher gibt es von den Stakeholdern keine Voraussetzungen an das verwendete Datenbankmodell oder -Architektur.

Hardwarevoraussetzungen

	Minimal	Empfohlen
Prozessor	1 Core à 1 GHz	2 Cores à 1.5 GHz oder äquivalent
RAM	1 GB DDR3	4 GB DDR4
Speicherplatz	10 GB freier Speicherplatz	30 GB freier Speicherplatz
Netzwerk	Internetanbindung	Internetanbindung up/down 50Mbps/50Mbps
Betriebssystem	Supported stable release Windows oder Linux	CentOs 7, Ubuntu Server 18.04.2 LTS, Windows Server 2016 oder äquivalent

Use-Case-Diagramm Extended

Während des Erarbeitungsprozess wurden diverse Überlegungen zur Architektur und Umsetzung gemacht, wobei versucht wurde, diese im Use-Case Diagramm wiederzugeben.:

- Filterung der Daten serverseitig um die sensiblen/vertraulichen Daten zu schützen.
- Gemeinsamkeiten von Einzelsitzungen und Gruppensitzungen
- Lösung für die Integration der Expositionstherapie mit anderen Komponenten
- Abhängigkeiten aufzeigen um später die Arbeiten gliedern zu können

Nebst diesen Punkten, ist auch das erweiterte Use-Case Diagramm nicht vollständig und viele Use-Cases wurden absichtlich weggelassen. Darunter fallen feinere Unterteilungen und eher schwache oder offensichtliche Abhängigkeiten.:

- Unterteilung in "Erfassung von Patienten" und "Erfassung von Therapeuten" abhängig von der "Erfassung von Personen".
- "Ändern von Gruppensitzungen", "Auflisten von Gruppensitzungen", "Löschen von Personen", ... Use-Cases welche durchaus zur Applikation gehören könnten, aber für die Hauptfunktionen und den Prototypen eine untergeordnete Rolle spielen.
- "Verschreibung von Medikamenten", "Vereinbaren von Terminen", ... Use-Cases welche absichtlich weggelassen wurden, da diese nicht mit unserer Applikation umgesetzt werden.

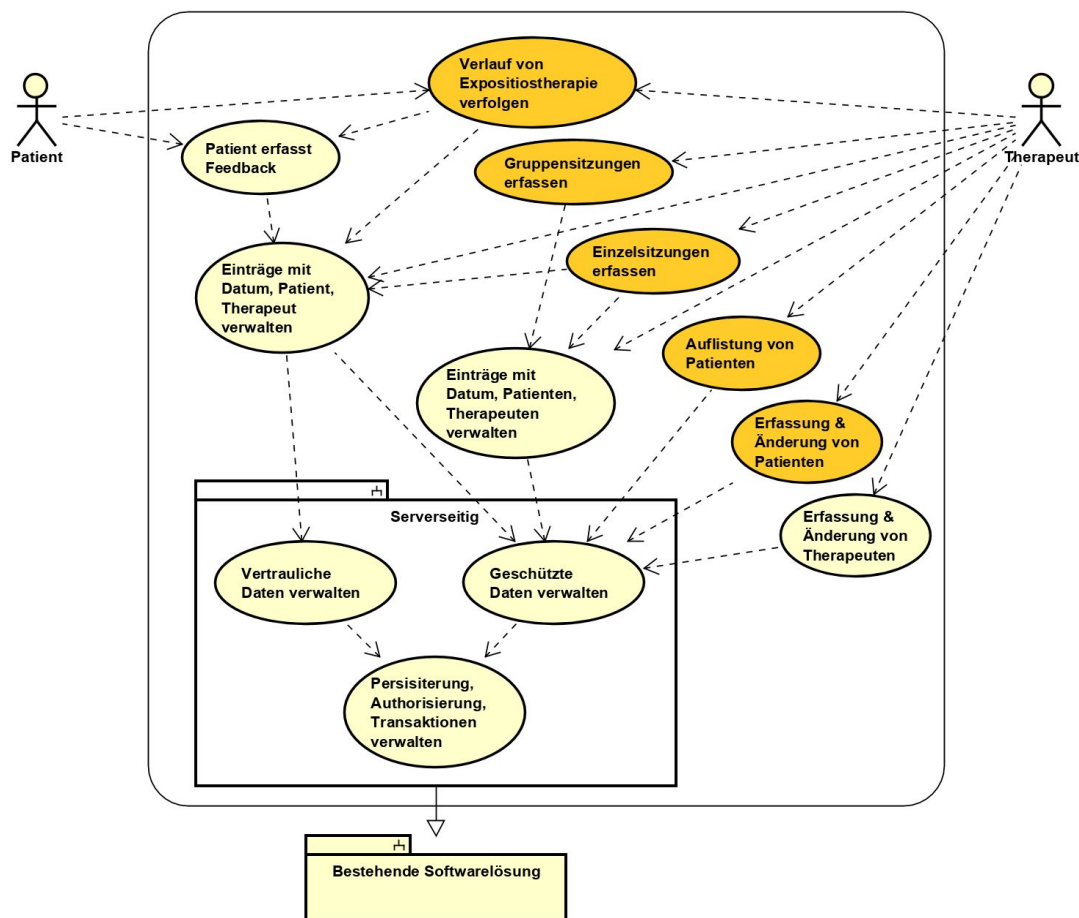


Abbildung 7: Erweitertes Use Case Diagramm