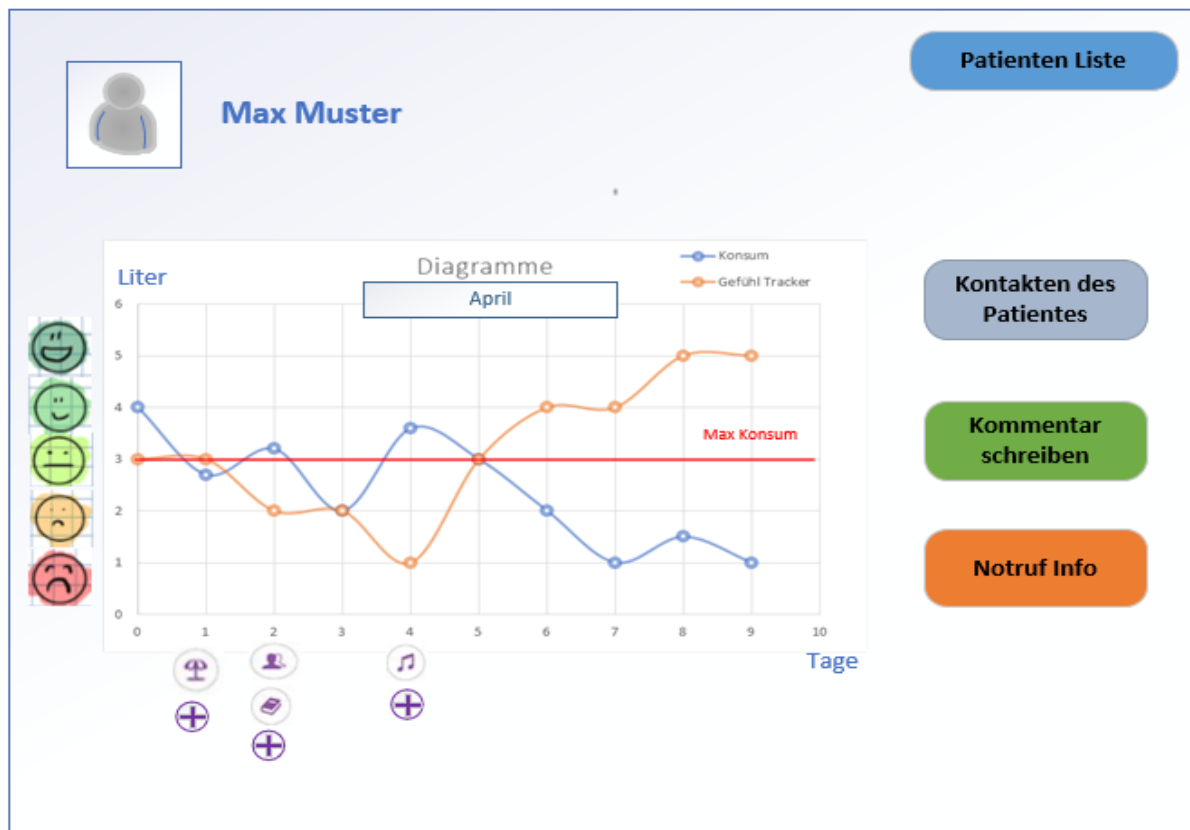


Requirements Specification



Sucht – Tagebuch

Team red

Task 2

Inhalt

Vorwort	3
Versionsverlauf.....	3
Einleitung.....	4
Glossar	4
Nutzeranforderungen - User Requirements Definition	5
Functional System Requirements.....	5
Non-Functional System Requirements.....	7
Übersicht – Use Case Diagram	7
Use Case – Szenarios	8
Use Case Scenario: Notruf auslösen/annehmen.....	8
Use Case Scenario: Tagebucheintrag erstellen	11
Systemarchitektur	12
System Modell.....	12
Weiterentwicklung des Systems	13
Testing	14
Anhang	15

Vorwort

Dieses Dokument richtet sich an alle Beteiligten des Software Engineering Projekts **«Sucht-Tagebuch»**, welches im Rahmen des Moduls «Software Engineering and Design – BTI7081/20» der Berner Fachhochschule durchgeführt wird. Es spezifiziert die Benutzer- und Systemanforderungen und soll sowohl für die Entwickler der Applikation wie auch für deren Benutzer bzw. den Kunden verständlich sein.

Die Mitglieder des Entwicklungs-Teams sind:

Natalya Dénervaud, Dmytriy Pelts, Kaurisanker Kirupananthan, Julian Rodriguez Ugolini und Remo Meyer.

Versionsverlauf

Version	Autor	Änderungen	Datum
0.1	Alle	Initiales Dokument erstellt	31.03.2020
0.2	Alle	Erster Entwurf	02.04.2020
0.3	Alle	Zweiter Entwurf	07.04.2020
1.0	Alle	Definitive Version	08.04.2020

Einleitung

Das Ziel der Applikation «Sucht-Tagebuch» ist es, Ärztinnen und Ärzte bei der Behandlung von suchtkranken Menschen zu unterstützen. Sie richtet sich primär an Ärztinnen und Ärzte, welche Patienten ambulant in einer Klinik betreuen. Diese haben viele verschiedene Patienten, welche sie in unterschiedlichen zeitlichen Abständen behandeln. Das Sucht-Tagebuch soll es ihnen erleichtern, sich unkompliziert und zeitnah einen Überblick über den Behandlungsverlauf und den Zustand ihrer Patienten zu verschaffen.

Die Patienten erfassen täglich auf ihrem Handy ihren Gefühlszustand (Konsumdruck und Motivation) und ggf. ihr Konsumverhalten und ergänzen dies bei Bedarf mit kurzen Kommentaren oder Angaben zu ihren Tagesaktivitäten (insbesondere was sie getan haben, um nicht zu konsumieren).

Da sich die Patienten freiwillig in der Klinik anmelden, um ihre Sucht behandeln zu lassen, gehen wir davon aus, dass sie bereit sind, die App zu nutzen und ihr Tagebuch ehrlich zu führen. Die Bedienung der Patienten-Applikation soll sehr einfach gehalten werden, so dass kein grosser Zeitaufwand entsteht und alle Patienten in der Lage sind, sie zu benutzen.

Die so erfassten Daten werden den Ärztinnen und Ärzten auf einer Website übersichtlich graphisch dargestellt. Sie haben somit die Möglichkeit, täglich (bzw. nach Bedarf) den Zustand der Patienten zu verfolgen und sind dadurch besser über deren aktuellen Zustand informiert. Falls sie mit dem Patienten ein Konsumlimit vereinbart haben können sie dieses in der App eintragen und mit dem tatsächlichen Konsum des Patienten vergleichen. Zusätzlich können die Ärztinnen und Ärzte sich Notizen zu ihren Beobachtungen und Fragen für die nächste Sitzung aufschreiben.

Als zusätzliches Feature wird ein «Alarm Knopf» eingebaut. Diesen können die Patienten drücken, wenn sie Hilfe benötigen. Dabei können sie zwischen drei Notsituationen wählen: Notruf, dringender Alarm und schneller Termin. Bei einem Notruf soll sich der Patient mit dem Notruf 112 verbunden werden, beim dringenden Alarm, soll sich schnellstmöglich jemand telefonisch bei ihm melden, der ihn psychisch unterstützen kann, und beim schnellen Termin soll sich bald jemand bei ihm melden, um einen Termin abzumachen. Bei den letzteren zwei soll der Arzt benachrichtigt werden.

Glossar

Framework	Das Framework stellt ein Programmiergerüst dar und umfasst Bibliotheken, Komponenten und Laufzeitumgebungen und stellt die Designgrundstruktur für die Entwicklung zur Verfügung.
Vaadin	Vaadin ist ein freies Webframework, der als Framework für die Entwicklung von Geschäftsanwendungen für den Browser bekannt und auch bringt die Applikationen auf Smartphones und Tablets zu laufen.
GUI	(von englisch <i>graphical user interface</i>) Grafische Benutzeroberfläche von Spielen und Programmen
Use Case	Use Cases dokumentieren die Funktionalität eines geplanten oder existierenden Systems auf Basis von einfachen Modellen. In einem Use Case – auch Anwendungsfall genannt – wird das nach aussen sichtbare Verhalten eines Systems aus Sicht der Nutzer beschrieben.

Nutzeranforderungen - User Requirements Definition

Functional System Requirements

FR – Functional Requirement; UC – Use Case

Nr.	Name	User	Beschreibung	UC
FR 1	Login	App	Sowohl Patient wie auch Arzt können sich anmelden und sehen die für sie bestimmte Ansicht.	
FR 2	Patient erstellen	Arzt	Ärzte müssen Patienten kreieren können, damit diese die App nutzen können	
FR 2.1	Kontaktdaten erfassen und ändern	Arzt	Der Arzt soll für Notfälle die Kontaktdaten des Patienten erfassen und ändern können	3
FR 2.2	Kommentar zu Patienten erfassen und ändern	Arzt	Der Arzt soll einen Kommentar zu jedem Patient haben, den er erfassen und ändern kann	3
FR 2.3	Aktivitäten zum Erfassen festlegen	Arzt	Der Arzt soll festlegen welche Aktivitäten gefragt werden, ob sie getätigt wurden	
FR 2.4	Vermeidungsstrategien erfassen	Arzt	Der Arzt soll festlegen können, was für Vermeidungsstrategien der Patient anwenden kann, wenn der Konsumdruck hoch ist	
FR 2.5	Bedingungen automatischer Alarm festlegen	Arzt	Der Arzt soll individuell einstellen können, unter welchen Bedingungen einen automatischen Alarm erhalten soll	4
FR 3	Übersicht über alle Patienten	Arzt	Der Arzt braucht eine Liste aller Patienten	1
FR 3.1	Patient suchen	Arzt	Der Arzt soll in der Liste der Patienten einen Patienten suchen können	1
FR 3.2	Übersicht Verlauf	Arzt	Der Arzt hat bei allen Patienten eine Übersichtliche Grafik mit den wichtigsten Parameter Konsum, Befindlichkeit, Konsumdruck und Motivation	2
FR 3.3	Tagesansicht einsehen	Arzt	Wenn auf einen Tag geklickt wird, sollen alle erfassten Aktivitäten und Kommentare zu sehen sein.	2
FR 4	Alarmsystem	App	Die Applikation soll ein abgestuftes Alarmsystem haben	5
FR 4.1	Notfall auslösen	Patient	Der Patient soll einen Notfallalarm auslösen können und mit 112 in Kontakt gebracht werden	7
FR 4.2	Dringender Alarm auslösen	Patient	Der Patient soll einen dringenden Alarm auslösen können, um einen Anruf des Arztes oder des psychiatrischen Notfalldienstes erhalten	7
FR 4.3	Dringender Alarm erhalten	Arzt	Wenn der Patient bei Zusammenbrüchen einen dringenden Alarm auslöst, soll der Arzt benachrichtigt werden, um zurückrufen zu können oder den psychiatrischen Notdienst zu benachrichtigen	5,6
FR 4.4	Schneller Termin auslösen	Patient	Der Patient soll einen schnellen Termin auslösen können und so einen schnellstmöglichen Termin erhalten	7

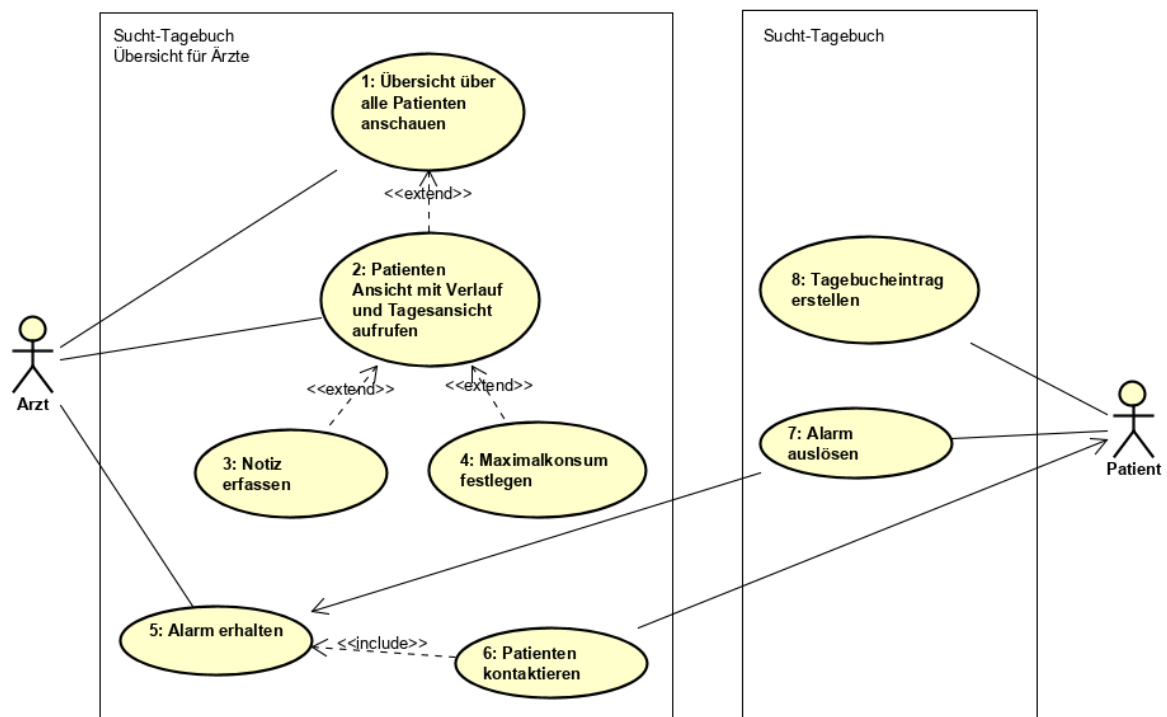
FR 4.5	Schneller Termin erhalten	Arzt	Wenn der Patient einen baldigen Termin wünscht, soll der Arzt benachrichtigt werden, damit er seinen Assistenten einen Termin abmachen lässt	5
FR 4.6	Automatischer Alarm auslösen	App	Das System soll, wenn die festgelegten Bedingungen erfüllt sind einen automatischen Alarm auslösen	7
FR 4.7	Automatischer Alarm erhalten	Arzt	Der Arzt soll benachrichtigt werden, wenn die Bedingungen für einen automatischen Alarm erfüllt werden	5
FR 5	Notifikation zum Erfassen erhalten	Patient	Der Patient soll daran erinnert werden, den täglichen Eintrag zu erfassen	8
FR 5.1	Täglichen Eintrag erfassen	Patient	Der Patient soll seinen täglichen Eintrag erfassen	8
FR 5.3	Konsum erfassen	Patient	Der Patient soll seinen Konsum erfassen	8
FR 5.4	Konsumdruck erfassen	Patient	Der Patient soll seinen Konsumdruck erfassen	8
FR 5.5	Motivation erfassen	Patient	Der Patient soll seine Motivation erfassen	8
FR 5.6	Aktivitäten erfassen	Patient	Der Patient soll seine Aktivitäten erfassen	8
FR 5.7	Aktivitäten kreieren	Patient	Der Patient soll Aktivitäten kreieren können	
FR 5.8	Kommentar erfassen	Patient	Der Patient soll nach einem Kommentar und Veränderungen gefragt werden und als Test erfassen	8
FR 5.9	Fragen für Konsultation notieren	Patient	Der Patient soll Fragen für die nächste Konsultation erfassen	8
FR 5.10	Eintrag speichern	App	Die App soll den Eintrag speichern und ablegen	8
FR 6	Vermeidungstipps erhalten	Patient	Wenn der Konsumdruck für den Patienten hoch ist, soll er Vermeidungsstrategien als Tipps erhalten	8
FR 6.1	Vermeidungstipps ansehen	Patient	Der Patient soll einfach auf die Vermeidungsstrategien zugreifen können, im Fall, dass der Konsumdruck hoch ist	
FR 6.2	Vermeidungsstrategien erfassen	Patient	Der Patient soll Vermeidungsstrategien erfassen, die er anwenden kann, wenn der Konsumdruck hoch ist	
FR 7	Übersicht Verlauf	Patient	Der Patient soll eine Übersicht über den Verlauf seiner Sucht haben	
FR 7.1	Tagesansicht einsehen	Patient	Wenn auf einen Tag geklickt wird, sollen alle erfassten Aktivitäten und Kommentare zu sehen sein.	
FR 8	Webapplikation	App	Die Applikation soll Webbasiert sein	
FR 8.1	Responsive design	App	Das Design soll responsiv sein und sowohl auf Computern wie auch auf Smartphones funktionieren	
FR 8.2	Persistenz	App	Die Daten müssen gespeichert werden und nicht verloren gehen	

Non-Functional System Requirements

NR – Non-Functional Requirement; UC – Use Case

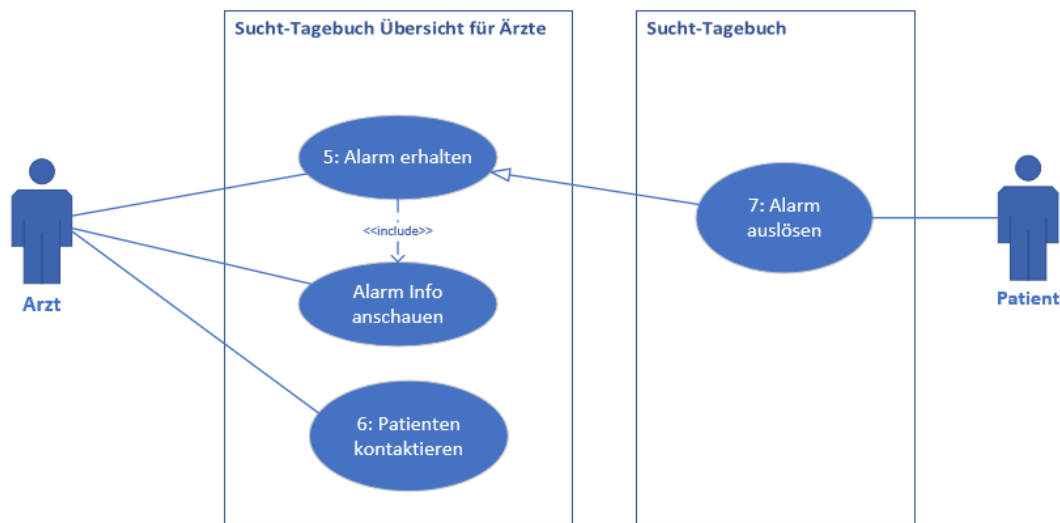
Nr	Name	User	Beschreibung	UC
NR 1	Bedienung	Patient	Patienten können die Applikation nach kurzer Erklärung ohne Probleme bedienen.	
NR 2	Bedienung	Arzt	Ärzte können die Applikation nach kurzer Schulung problemlos bedienen.	
NR 3	Datenschutz	System	Alle sehen nur die für sie bestimmten Daten	
NR 4	Performance	System	Alle Ansichten sind in nützlicher Frist geladen	
NR 5	Zuverlässigkeit	System	Das System ist immer erreichbar und verliert durch abstürze keine Daten	

Übersicht – Use Case Diagram



Use Case – Szenarios

Use Case Scenario: Notruf auslösen/annehmen



Use Case Nr. 5

Nr. und Name:	5: Alarm erhalten
Szenario:	Arzt bekommt eine Alarm-Meldung von dem Patienten.
Kurzbeschreibung:	Arzt bekommt eine Alarm-Meldung mit der Beschreibung, was für ein Alarm ausgelöst wurde. Nach Bedarf sucht der Arzt nach dem Patienten-Kontakt, um den Patienten anzurufen oder mit dem Patienten wird einen Termin vereinbart.
Beteiligt Akteure:	Arzt, Patient
Auslöser / Vorbedingung:	Die Applikation läuft im Hintergrund.
Ergebnisse / Nachbedingung:	Arzt bekommt einen Alarm wie eine Push-Notifikation mit der Beschreibung des Alarms und dann wird Kontakt mit dem Patienten aufgenommen.

Ablauf:

Nr.	Wer	Was
5.1	System	Die Applikation «Sucht - Tagebuch» läuft im Hintergrund.
5.2	System	Die Applikation bekommt eine Meldung von dem Smartphone des Patienten.
5.3	System	Zeigt eine Alarm-Meldung als eine Push-Notifikation mit den Angaben folgender Daten: Datum, Zeit und Beschrieb des Alarms.
5.4	Arzt	Sieht die Alarm-Meldung und bestätigt mit Option «Gelesen».
5.5	Arzt	Arzt öffnet die Applikation «Such-Tagebuch».
5.6	Arzt	Wählt die Option «Kontakte des Patienten».
5.7	Arzt	Sucht die Telefonnummer des Patienten, um ihn anzurufen.

Ausnahmen, Varianten:

Nr.	Wer	Was
5.4	Arzt	Bekommt eine Alarm-Meldung und wählt die Option «App öffnen».
5.5	System	Die Applikation «Such-Tagebuch» wird geöffnet mit der Ansicht des Patienten, der einen Alarm ausgelöst hat.

Ausnahmen, Varianten:

Nr.	Wer	Was
5.4	Arzt	Bekommt eine Alarm-Meldung mit der Information über den «Notruf 112» und wählt die Option «App öffnen».
5.5	System	Die Applikation «Such-Tagebuch» wird geöffnet mit der Ansicht des Patienten, der einen Alarm ausgelöst hat.
5.6	Arzt	Wählt die Option «Kontakte des Patienten».
5.7	Arzt	Sucht nach dem Kontakt des Angehörigen, um diese Person über den Notfall zu informieren.

Use Case Nr. 6

Nr. und Name:	6: Patienten kontaktieren
Szenario:	Arzt schaut die Kontaktdaten des Patienten an und ruft ihn an
Kurzbeschreibung:	Arzt schaut die Information über den Zustand des Patienten durch die grafische Darstellung. Arzt will einen zusätzlichen Termin vereinbaren. Er sucht mit Hilfe der Option «Kontakte des Patienten», um den Patienten anzurufen.
Beteiligt Akteure:	Arzt
Auslöser / Vorbedingung:	Die Applikation läuft im Hintergrund.
Ergebnisse / Nachbedingung:	Arzt bekommt die Kontaktinformation des Patienten.

Ablauf:

Nr.	Wer	Was
6.1	Arzt	Arzt öffnet die Applikation «Sucht - Tagebuch».
6.2	Arzt	Wählt die Option «Kontakte des Patienten».
6.3	Arzt	Sucht die Telefonnummer des Patienten, um ihn anzurufen.

Use Case Nr. 7

Nr. und Name:	7: Alarm auslösen
Szenario:	Patient meldet einen Alarm.
Kurzbeschreibung:	Zustand des Patienten wird verschlechtert und er wird Hilfe vom Arzt brauchen. Mit Hilfe der Applikation «Sucht-Tagebuch» kann der Patient einen Alarm bei dem Arzt auslösen, um seine Schwierigkeiten mit dem Sucht-Problemen via Applikation zu melden.
Beteiligt Akteure:	Patient
Auslöser / Vorbedingung:	Patient öffnet die Applikation.
Ergebnisse / Nachbedingung:	Patient schickt eine Notifikation an dem Arzt via Applikation.

Ablauf:

Nr.	Wer	Was
7.1	Patient	Öffnet die Applikation und wählt eine aus den vorgegebenen Optionen: <ul style="list-style-type: none"> - Notruf 112 - Dringender Fall - Schneller Termin
7.2	Patient	Betätigt die Taste «Schneller Termin».
7.3	System	Erfasst die Information über den Alarm.
7.4	System	System schickt die Push-Notifikation an die Applikation des Arztes.

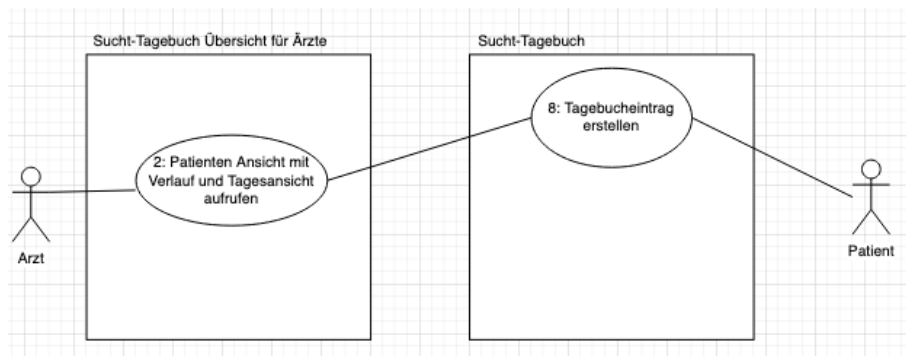
Ausnahmen, Varianten:

Nr.	Wer	Was
7.2	Patient	Betätigt die Taste «Notruf 112».
7.3	System	Erfasst die Information über den Alarm.
7.4	System	Öffnet eine im Smartphone vorhandene Telefon-Applikation und fügt die Notruf-Nummer «112» im Nummer-Block ein.
7.5	Patient	Betätigt die «Anruf» Taste.

Ausnahmen, Varianten:

Nr.	Wer	Was
7.2	Patient	Betätigt die Taste «Dringender Fall».
7.3	System	Erfasst die Information über den Alarm.
7.4	System	System schickt die Push-Notifikation an die Applikation des Arztes.

Use Case Scenario: Tagebucheintrag erstellen



Use Case Nr. 8

Nr. und Name:	8: Tagebucheintrag erstellen
Szenario:	Patient möchte den täglichen Eintrag auf dem App erfassen.
Kurzbeschreibung:	Im Eintrag werden seine Motivation, sein Konsumverhalten und seine Aktivitäten erfasst. Optional können neue Aktivitäten, Kommentare und Notizen für die nächste Konsultation erfasst werden.
Beteiligt Akteure:	Patient, Arzt
Auslöser / Vorbedingung:	Patient öffnet das Sucht-Tagebuch App und möchte einen Eintrag erfassen.
Ergebnisse / Nachbedingung:	Eintrag wird erfasst und wird auf der Website für Ärzte ersichtlich sein. Mit diesen Informationen kann der Arzt den nächsten Termin besser planen.

Ablauf:

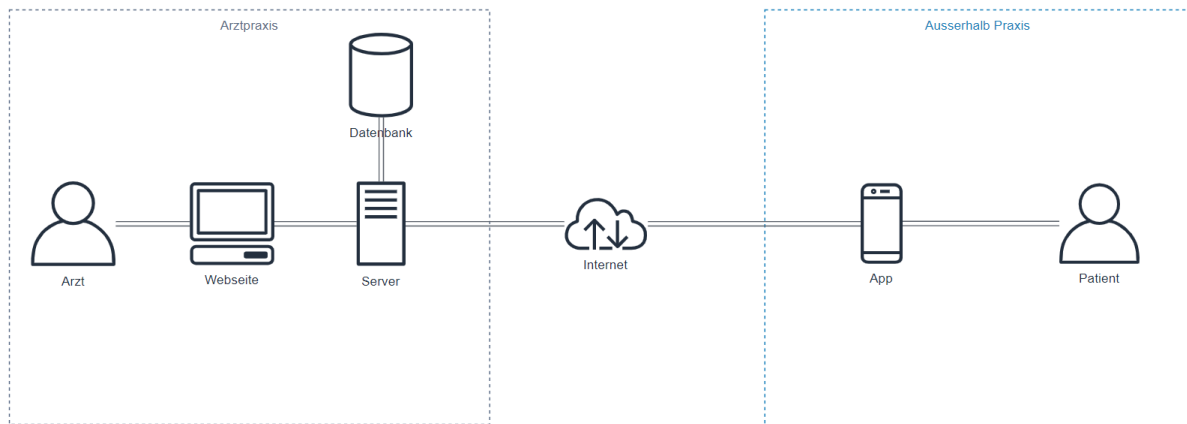
Nr.	Wer	Was
8.0	Patient	Öffnet die Applikation und tätigt die Taste «Tagebucheintrag erfassen».
8.1	System	System zeigt das entsprechende Fenster an.
8.2	Patient	Konsum erfassen (wie viel war der tägliche Konsum?).
8.3	Patient	Konsumdruck erfassen (wie gross war das Verlangen?).
8.4	Patient	Motivation erfassen (wie gross war die Motivation, um die Therapie fortzusetzen?).
8.5	Patient	Aktivitäten erfassen (welche Aktivitäten wurden heute unternommen?).
8.6	Patient	Kommentar erfassen (optional).
8.7	Patient	Frage für nächste Konsultation notieren (optional).
8.8	Patient	Eintrag mit der Taste «Eintrag speichern» abschliessen und speichern
8.9	System	System speichert den Eintrag und leitet die Informationen an die Webseite weiter und alle Einträge können durch den Arzt eingesehen werden.

Ausnahmen, Varianten:

Nr.	Wer	Was
8.5.1	Patient	Aktivitäten kreieren (Patient kann laufend neue Aktivitäten hinzufügen).
8.8.1	Patient	Ist im Nachhinein etwas in den Sinn gekommen und möchte sein Eintrag ändern.

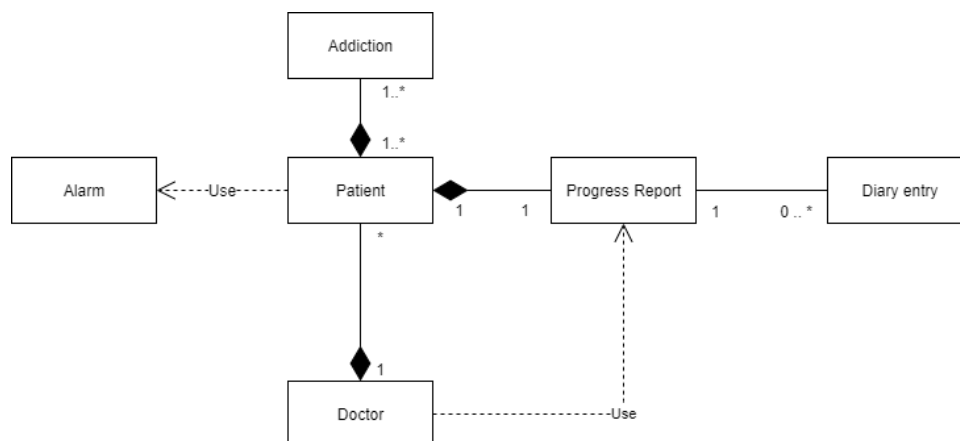
Systemarchitektur

Der Kunde verwendet eine App auf seinem Mobiltelefon. Diese verschickt die Eingaben übers Internet über das Internet an die Praxis.



Der Server, der in der Praxis des Arztes steht, empfängt die Daten des Benutzers übers Internet und speichert diese in einer Datenbank. Sollte der Arzt nun die Daten einsehen wollen, werden ihm diese auf einem GUI auf einem Computer in der Praxis vom Server zur Verfügung gestellt.

System Modell



Patient	Surname, Name, E-Mail, Password, Address, Addictions, Phone number, Relatives phone number, Progress report.
Doctor	Surname, Name, E-Mail, Password, Phone number, patients.
Alarm	Types
Progress Report	Diary entries
Diary entry	Motivation, Consumption pressure, Consumption, Activity, Date
Addiction	Types

Der Patient, der eine oder mehrere Süchte haben kann, wird von einem Arzt behandelt. Dieser wiederum kann mehrere Patienten behandeln. Der Patient soll täglich einen Tagebucheintrag ausfüllen. Dieser wird in einem Verlaufsbericht zusammengefasst. Diesen Bericht kann der Arzt einsehen. Zudem kann der Patient einen Alarm auslösen, der mehrere Optionen bietet.

Weiterentwicklung des Systems

Es gibt verschiedene Features, die je nach Feedback in Zukunft interessant sein könnten:

Medikamenteneinnahme bestätigen

Es könnte interessant sein, über dieses Tool allfällige Medikamenteneinnahmen mit einem Foto der Medikamentenpackung bestätigen zu lassen und entsprechend einen Alarm auszulösen, wenn es länger ausgesetzt wurde.

Kalender importieren

Wenn die Patienten ihre Kalender elektronisch führen, könnte man diese auch mit unserer Applikation verbinden und gewisse Tätigkeiten automatisiert erfassen. Dafür wäre es sinnvoll das Erfassen zukünftiger Aktivitäten nicht komplett zu verunmöglichen.

Mehr Grafiken

Wir erstellen Grafiken, die eine Übersicht über die Stimmung und den Konsum bieten. Es könnte das Bedürfnis aufkommen individuelle Grafiken mit Aktivitäten oder Mehrfachkonsum anzeigen zu lassen.

Zielvereinbarungen

Das Festlegen von Zielen und eine Möglichkeit diese Ziele über Aktivitäten oder über den Konsum zu überprüfen und den Fortschritt zu verfolgen könnte nützlich sein.

Passwortzurücksetzung

Das Zurücksetzen von Passwörtern über E-Mail müsste bestimmt mal implementiert werden.

Unterschiedliche Designs

Verschiedene Designs des User interface und ein Dark Mode könnten mal implementiert werden.

Ausdrucken

Das Gesundheitspersonal wäre je nach dem froh, um eine Möglichkeit die Daten übersichtlich auszudrucken, da viele noch mit Papier arbeiten oder die Daten ablegen möchten.

Exportieren/Schnittstelle

Je nach dem, wäre es nützlich die Daten, über eine Schnittstelle exportieren zu können oder über ein API von einem anderen Programm (z.B. elektronische Patientendossiers) abfragen zu können.

Testing

Das Ziel ist es die Funktionalität des Codes sicherzustellen, indem alle Fehler gefunden und behoben werden. Wir werden folgende Testverfahren anwenden:

Komponententest / Unit Tests

Diese Tests haben das Ziel die Funktionalität innerhalb einzelner abgrenzbaren Teile der Software (in unserem Fall einzelne Klassen oder Methoden) auf Ihre Funktionalitäten und Korrektheit zu überprüfen. Durch den Entwickler selbst durchgeführten Tests ist der Nachweis der technischen Lauffähigkeit.

Integrationstest

Mit Unit-Tests überprüft man Komponenten unabhängig von anderen und garantiert die funktionale Korrektheit. Integrationstests stellen anschliessend sicher, dass das Zusammenspiel und die Kommunikation der Komponenten untereinander fehlerfrei ablaufen.

Applikationstest

Hier ist das Ziel, dass wir das gesamte System überprüfen und sicherstellen, ob unsere Applikation und Datenbank auf spezifische Anfragen korrekt reagiert.

Akzeptanztests

User Stories werden im Rahmen der agilen Softwareentwicklung zusammen mit Akzeptanztests zur Spezifikation von Anforderungen eingesetzt. Dabei wird jede User Story auf eine Story-Card geschrieben. Mithilfe von Akzeptanztests wird geprüft, ob unsere Software die funktionalen Erwartungen und Anforderungen im Gebrauch erfüllt.

Anhang

Die Applikation wird mit dem Vaadin-Framework erstellt und läuft serverseitig. Sowohl Ärzte wie auch Patienten sollen über ein Smartphone, Tablet oder Computer darauf zugreifen können.

Bedingung dafür ist, dass das Gerät einen modernen Browser nutzt. Die Daten werden auch auf dem Server gespeichert.