

Software Engineering and Design - Team Red

Software Requirements Specification Document

Task 2: Requirements

Thomas Baumann
Ismael Riedo

Frédéric Lehmann
Fridolin Zurlinden
Tobias Weissert

Severin Thalmann
Roland Roccaro

April 16, 2018

Contents

1	Preface	3
1.1	Leserschaft	3
1.2	Version	3
1.3	Versionsbegründungen	3
1.4	Versionsänderungen	3
2	Introduction	4
2.1	Kurzbeschreibung der Funktionen des Systems	4
2.2	Wie wird es mit anderen Systemen zusammen arbeiten	4
3	Glossary	5
4	User requirements definition	6
4.1	Use Cases Diagramm	6
4.2	Use Cases	7
4.2.1	Use Case 1	7
4.2.2	Use Case 2	7
5	System architecture	9
5.1	Übersicht der System Architektur	9
6	System requirements specification	10
6.1	Data Flow	10
6.2	Schnittstellen	10
7	System models	11
7.1	Modell einer Datenbank	11
7.2	Modell des Data-Flow	12
8	System evolution	13
8.1	Was wir für unsere Anwendung beachten müssen im Design	14
9	Testing	15
9.1	Was wird getestet?	15
9.2	Wie wird es getestet?	15
10	Appendices	16
10.1	Hardware requirements	16
10.2	Database and application requirements	16
11	Index	17

1 Preface

1.1 Leserschaft

Dieses Dokument richtet sich an die Kunden des Team Red, Kurs Software Engineering and Design Frühlingssemester 2018. Es dient als Grundbeschreibung der Applikation welche durch das Team entwickelt wird.

1.2 Version

Table 1.1: Version

Versionsnummer	Autor	Beschreibung
1.0	Roccaro	Grundfassung, Bassierend auf Design Thinking

1.3 Versionsbegründungen

Table 1.2: Versionsbegründungen

Versionsnummer	Autor	Beschreibung
1.0	Roccaro	Grundfassung des Spezifikationsdokuments, Fassung zur Ansicht der Spezifikation durch Kunden.

1.4 Versionsänderungen

Table 1.3: Versionsänderungen

Versionsnummer	Autor	Beschreibung
1.0	Roccaro	Grundfassung des Spezifikationsdokuments, Fassung zur Ansicht der Spezifikation durch Kunden.

2 Introduction

Das System wird benötigt im Umgang mit Angststörungen bzw. Sozialen Phobien. Aktuell existieren keine Lösungen für Patienten in ambulanter Behandlung die an Angststörungen leiden. Unsere Applikation wird den Patienten bei der Genesung unterstützen und im mehr Lebensqualität geben. Auch soll es dem Patienten ermöglichen durch weniger Angst wieder ein aktives Mitglied der Gesellschaft zu werden.

2.1 Kurzbeschreibung der Funktionen des Systems

Unsere Applikation bietet über Gameification einen Anreiz sich jeden Tag schrittweise an ein Ziel anzunähern. Die Applikation bietet die Möglichkeit, dass der Patient zusammen mit dem Therapeuten Challenges erstellt. Diese Challenges sind Aufgaben welche der Patient anschliessend gestellt bekommt. Nach der Ausführung einer Challenge wird der Patient nach einer Beschreibung des Erlebten gefragt. Diese Beschreibungen sind für den Therapeuten zugänglich, später während den Therapiesitzungen können sie besprochen werden.

2.2 Wie wird es mit anderen Systemen zusammen arbeiten

Die Applikation wird selbst laufen bzw. keinen Zugang zu anderen Systemen benötigen. Es sind keine Schnittstellen zu anderen Systemen für den Datenabgleich vorgesehen.

Die Applikation kommuniziert nicht mit anderen Systemen. Die Verbindungen welche projektiert werden, sind Emails oder Push Nachrichten an den Therapeuten, welche von der Applikation ausgelöst werden.

3 Glossary

Challenges Is a mark up language specially suited for scientific documents. 4

Cross Platform Is a mark up language specially suited for scientific documents. 9

Gameification Is a mark up language specially suited for scientific documents. 4

GDPR compliant Is a mark up language specially suited for scientific documents. 9

mobile first Is a mark up language specially suited for scientific documents. 9

responsive Design Is a mark up language specially suited for scientific documents. 9

Vaadin Is a mark up language specially suited for scientific documents. 9

4 User requirements definition

4.1 Use Cases Diagramm

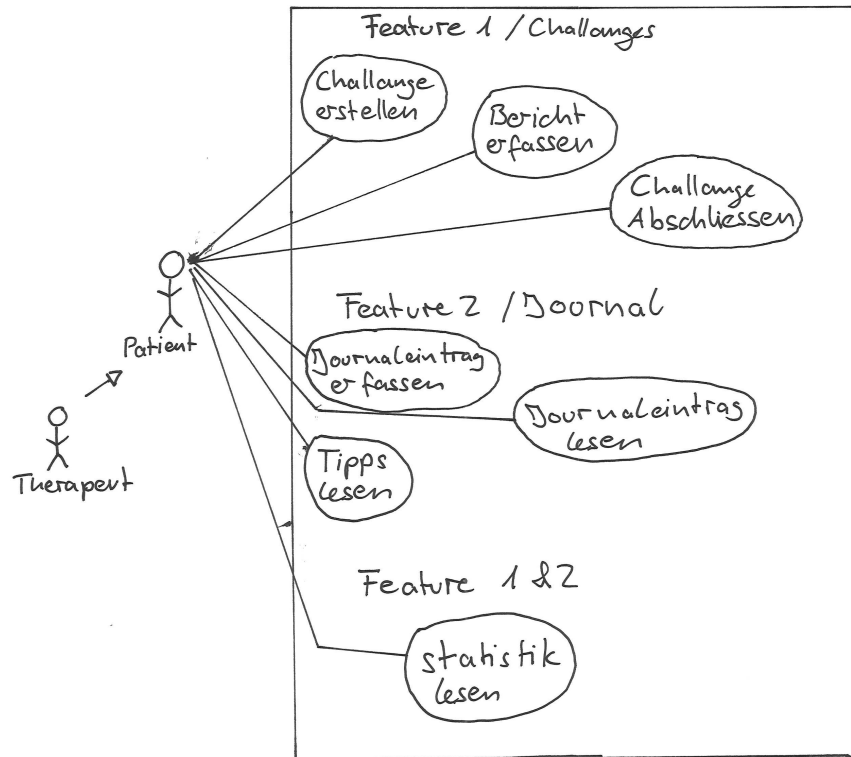


Figure 4.1: Use Case Diagramm Feature 1

4.2 Use Cases

4.2.1 Use Case 1

Table 4.1: Use Case 1

Bezeichnung	Beschreibung
Name	Challenge erstellen
Nummer	1
Kurzbeschreibung	Der Patient erfasst eine neue Challenge.
Beteiligte Akteure	Benutzer
Auslöser / Vorbedingung	keine
Ergebnisse / Nachbedingung	Die Challenge erscheint im richtigen Level mit dem richtigen Status. Sie wird für die Statistik berücksichtigt.

Ablauf

Table 4.2: Ablauf Use Case 1

Nr.	Akteur	Aktion
1.0	Benutzer	Der Patient betätigt den Hinzufügen Button.
1.1	Benutzer	Über die Erstellungsmaske werden die Details für die Challenge abgefüllt. Danach wird die Challenge gespeichert.
1.2	System	Überprüfung der eingegebenen Daten.
1.3	System	Die Challenge wird dem dazugehörigen Level angefügt und auf dem UI angezeigt.

4.2.2 Use Case 2

Table 4.3: Use Case 2

Bezeichnung	Beschreibung
Name	Neuer Journaleintrag
Nummer	2
Kurzbeschreibung	Der Patient erfasst einen neuen Journaleintrag.
Beteiligte Akteure	Benutzer
Auslöser / Vorbedingung	keine
Ergebnisse / Nachbedingung	Der Journaleintrag wird erstellt und ist in der Übersicht der Journaleinträge ersichtlich. Der neue Journaleintrag wird für die Statistik berücksichtigt.

Ablauf

Table 4.4: Ablauf Use Case 2

Nr.	Akteur	Aktion
1.0	Benutzer	Der Benutzer wählt den Hinzufügen Button.
1.1	Benutzer	Über die Erstellungsmaske werden die Details für den Journaleintrag eingefügt. Danach wird er gespeichert.
1.2	System	Überprüfung der eingegebenen Daten.
1.3	System	Der Eintrag wird den bestehenden Einträgen angefügt und im UI dargestellt.

5 System architecture

Bei der Applikation handelt es sich um eine Webapplikation, sie soll also Cross Platform funktionieren. Einzige Systemanforderung an den Client ist ein moderner Browser. Durch den steigenden Trend an mobilen Geräten soll die Applikation sowohl auf Smartphones wie auch Tablets funktionieren, dies geschieht durch ein responsive Design und einer mobile first Entwicklung. Die Business Logik der Applikation soll in Java geschrieben werden, da sich das Team bereits ein breites Wissen in dieser Technologie erarbeitet hat. Genauere Spezifikationen an das Backend und die Datenbank werden im Kapitel Appendices spezifiziert.

5.1 Übersicht der System Architektur

Die Webapplikation wird mit Vaadin umgesetzt. Dabei wird ein hoher Wert auf Simplität seitens Client gesetzt. Sämtliche Daten werden auf der Datenbank des Backends abgelegt. Auf der Clientseite werden keine Installationen durchgeführt.

Alle Daten werden ausschliesslich von der Applikation verwendet und nicht Drittanbietern zur Verfügung gestellt. Die Applikation geht mit den Personendaten voll GDPR compliant um.

Der Server selbst wird in einer Cloud positioniert, die Backups sowie fundamentale Sicherheitsmassnahmen werden durch den Hoster des Servers durchgeführt.

6 System requirements specification

6.1 Data Flow

Da es sich um eine Webapplikation handelt werden keine Daten auf dem Gerät gespeichert, sondern werden jeweils direkt vom Server gelesen und geschrieben. Dadurch haben alle Geräte jeweils den aktuellen Stand der Daten. Beim speichern von neuen Daten werden diese direkt auf Ihre Korrektheit überprüft und der Benutzer im Fehlerfall informiert.

6.2 Schnittstellen

Aktuell sind keine Schnittstellen an Fremdsysteme vorgesehen, die Applikation wird sowohl von Patienten wie auch vom Arzt Standalone eingesetzt. Von einer Integration in ein anderes System oder zu Schnittstellen kann zukünftig nicht ausgeschlossen werden. Als mögliches Beispiel könnte ein Arzt direkt in seinem Klinik Informationssystem den Status aller Challenges seiner Patienten prüfen.

7 System models

7.1 Modell einer Datenbank

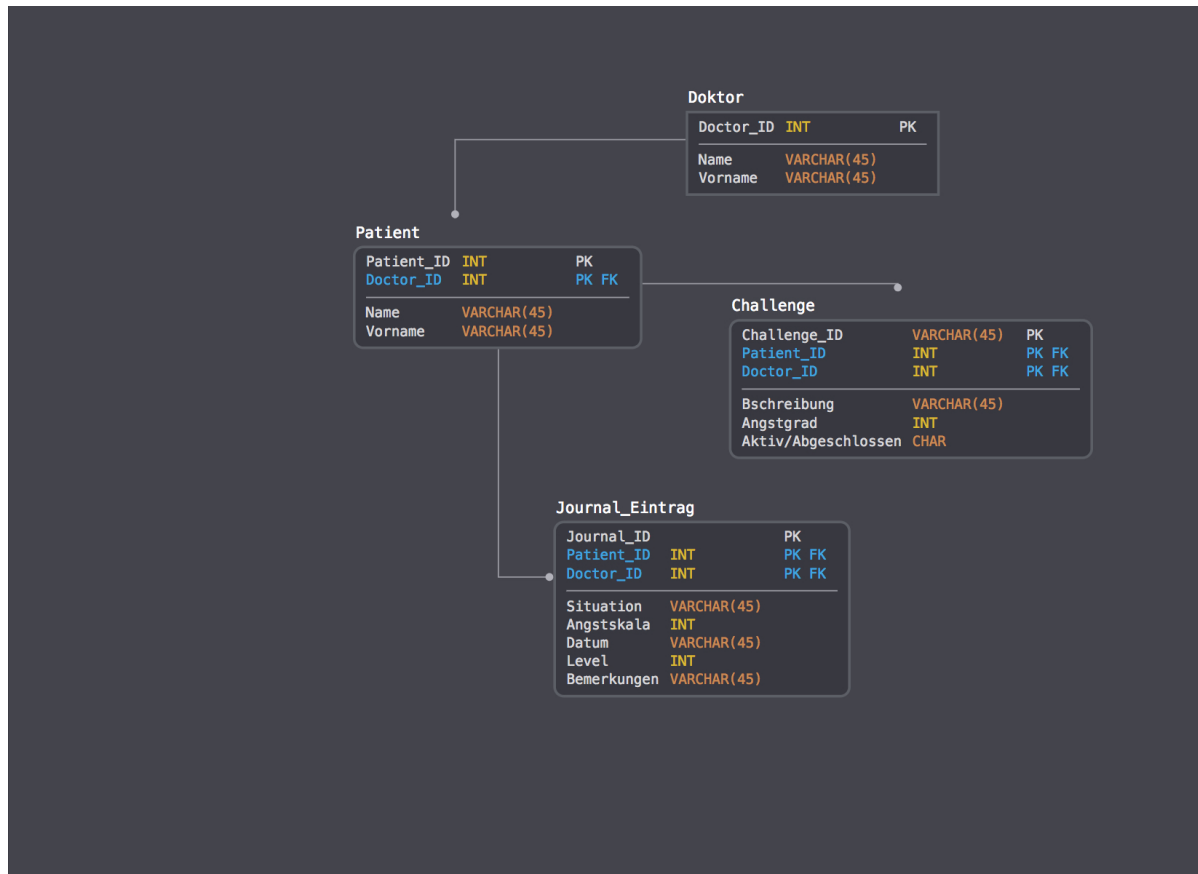


Figure 7.1: Database Schema

7.2 Modell des Data-Flow

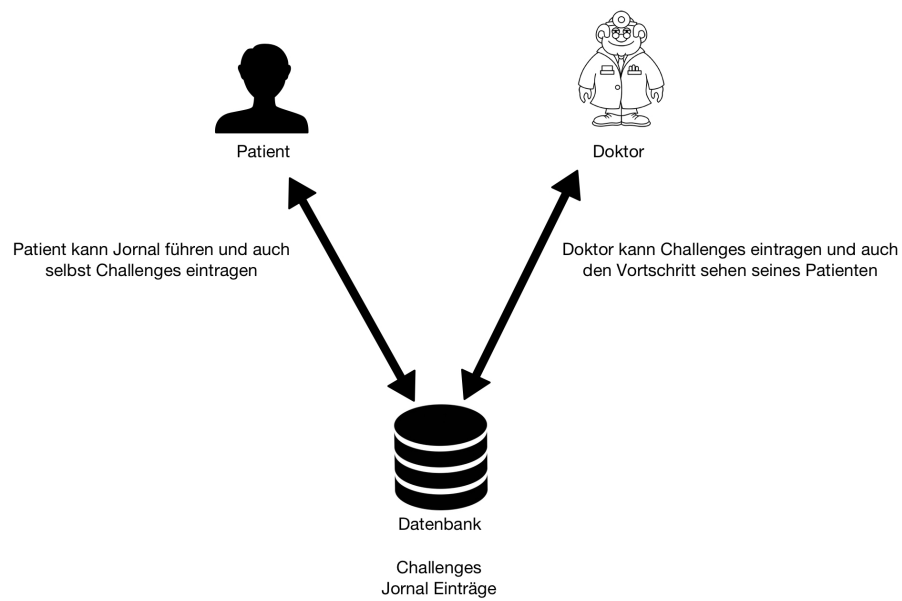


Figure 7.2: Data-Flow

8 System evolution

Software Evolution in Zusammenhang mit diesen acht Gesetzen von Lehmans.

Die Gesetze beschreiben ein Gleichgewicht zwischen den Kräften, die neue Entwicklungen vorantreiben, und den Kräften, die den Fortschritt verlangsamen. In den letzten Jahrzehnten wurden die Gesetze mehrfach überarbeitet und erweitert.

Table 8.1: Lehmans Law

Law	Description
Continuing Change	An E-type system must be continually adapted or it becomes progressively less satisfactory.
Self Regulation	E-type system evolution processes are self-regulating with the distribution of product and process measures close to normal.
Conservation of Organisational Stability	The average effective global activity rate in an evolving E-type system is invariant over the product's lifetime.
Conservation of Familiarity	As an E-type system evolves, all associated with it, developers, sales personnel and users, for example, must maintain mastery of its content and behaviour to achieve satisfactory evolution. Excessive growth diminishes that mastery. Hence the average incremental growth remains invariant as the system evolves.
Continuing Growth	The functional content of an E-type system must be continually increased to maintain user satisfaction over its lifetime.
Declining Quality	The quality of an E-type system will appear to be declining unless it is rigorously maintained and adapted to operational environment changes.
Feedback System	E-type evolution processes constitute multi-level, multi-loop, multi-agent feedback systems and must be treated as such to achieve significant improvement over any reasonable base.

An E-program is written to perform some real-world activity; how it should behave is strongly linked to the environment in which it runs, and such a program needs to adapt to varying requirements and circumstances in that environment.

8.1 Was wir für unsere Anwendung beachten müssen im Design

- Der Aufbau der Datenbank. Skalierbarkeit, Erweiterbarkeit und der Zugriff.
- Die Darstellung der App muss auf verschiedensten Geräten funktionieren. Falls Browser App, Darstellung auf Smartphone oder Tablet beachten.
- Verschiedene Betriebssysteme. Mac und Windows, Linux ausgenommen.
- Die Erweiterbarkeit des Journals falls vom Patienten oder Doktor erwünscht. Z.B. neue Punkte die er ins Journal aufnehmen will. Die Challenges erweitern, etc.
- Sprachsteuerung mit Sprache zu Text Integration.

9 Testing

Um sicherstellen zu können, dass unsere Applikation einwandfrei funktioniert, muss getestet werden.

9.1 Was wird getestet?

Unser verschiedenen Futures werden zuerst einem umfangreichen funktionalen Test unterzogen. Dabei wird zuerst ein Unit testing und ein gesamter System Test durchgeführt. Weiter werden wir Sicherheitstests und zum Abschluss Interoperabilitätstests durchführen.

In einem zweiten Schritt werden Nicht-funktionalen Tests durchgeführt. Dabei wird die Benutzbarkeit, Effizienz sowie Zuverlässigkeit des Systems getestet. Gerade für die Usability-Tests werden wir externe Testpersonen brauchen. Dazu würden sich unsere Interviewpartner gut eignen.

9.2 Wie wird es getestet?

Zuerst werden aufgrund des Anforderungskatalogs verschiedene Testfälle entworfen.

Verschiedene Testuser (interne und externe) bekommen ein Testprotokoll mit diesen Testfällen und den zu erwartenden Resultaten.

Für die internen Test werden wir ein White-Box Testverfahren einsetzen. Für die externen werden wir ein Black-Box Verfahren vorbereiten.

10 Appendices

10.1 Hardware requirements

Server:

Die wichtigste Anforderung an die Hardware ist die Erweiterbarkeit. Sie muss zu jederzeit einfach und schnell ausgebaut werden kann. Am Anfang, bei wenig Traffic auf dem Webserver wird die Hardware nur eine nebensächliche Rolle spielen. Bei steigenden Anfragen, muss die Hardware aber dementsprechend erweitert werden.

Client:

Grundsätzlich kann auf unsere Applikation von jedem internetfähigen Gerät zugegriffen werden. Es ist aber empfohlen nicht mit veralteten Browsern die Applikation zu nutzen. Dabei kann nicht garantiert werden, dass alle Funktionen einwandfrei funktionieren.

Die Datenbank und der Applikation werden auf einem Server in einer Cloud gehostet, die hier genannten Werte sind für ein initiales Setup gedacht.

Bei intensiver Nutzung und je nach Grösse der Datenbank und der Applikation müssen die Werte entsprechend angepasst werden.

Das initiale Setup sieht die regelmässige Verwendung der App für einen Patientenstamm mit zehn Patienten und zwei Therapeuten vor, welche das System aktiv nutzen.

Falls diese Werte überstiegen werden muss das System entsprechend skaliert werden.

10.2 Database and application requirements

- Memory: 1 GB
- Processor speed: 1 GHz
- Processor architecture: x64
- Hard Disk: 10 GB
- Internet bandwidth (included per month): 5 GB

Für Details zum Datenmodell siehe 7.1.

11 Index

List of Figures

4.1	Use Case Diagramm Feature 1	6
7.1	Database Schema	11
7.2	Data-Flow	12

List of Tables

1.1	Version	3
1.2	Versionsbegründungen	3
1.3	Versionsänderungen	3
4.1	Use Case 1	7
4.2	Ablauf Use Case 1	7
4.3	Use Case 2	7
4.4	Ablauf Use Case 2	8
8.1	Lehmans Law	13