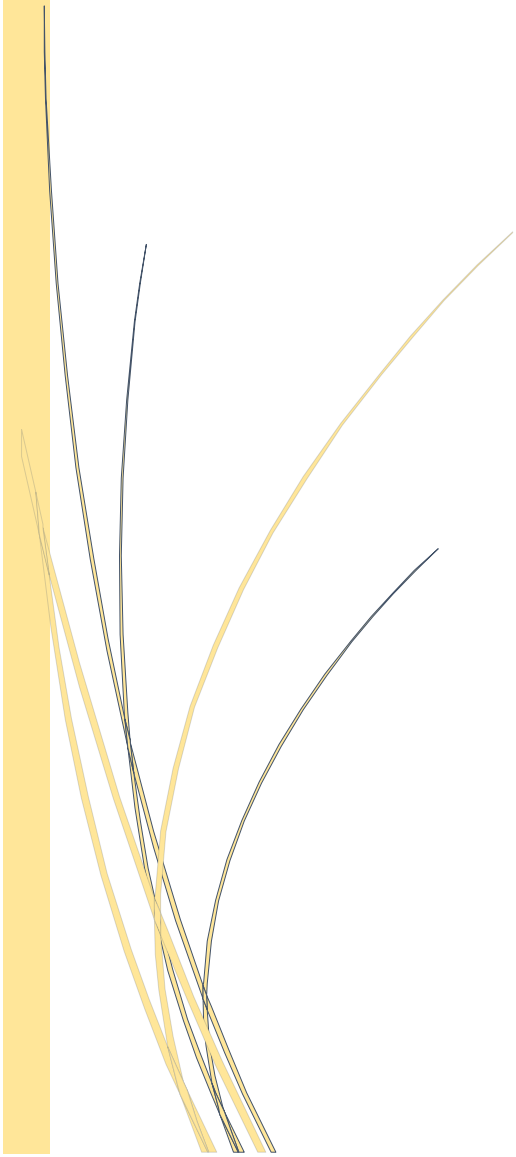
11.4.2018

Task 02

Team Yellow



Christian Haldi, Marc Häsler, Philipp Köfer, Nicola Michaelis, Stefan Schranz, Kevin von Allmen, Fabian Zurbuchen

Inhalt

1. Preface

Dieses Dokument richtet sich an die an dem Projekt beteiligten Systemingenieure, Endbenutzer, Systemadministratoren und Manager auf Kundenseite.

1.1 History

Version	Author	Anpassung	Datum
V001.00	Alle	Eröffnung Dokument	11.04.2018
V001.01	Alle	Dokument ergänzt	13.04.2018

https://github.com/PremiumBurger/ch.bfh.bti7081.s2017.green/blob/master/teamgreen-pms/src/main/resources/doc/cs01/task04/CS1%20Task4.pdf

2. Introduction

Initial	Name	Rolle
haldc4	Christian Haldi	Student
haesler	Marc Häsler	Student
Kofep1	Phillip Köfer	Student
michn2	Nicola Michaelis	Student
joedoe	Stefan Schranz	Student
Vonak1	Kevin von Allmen	Student
ezurbf	Fabian Zurbuchen	Student
Vgj1	Jürgen Vogel	Dozent
UrsKuenzler	Urs Künzler	Dozent

Das vorliegende Dokument beschreibt die Anforderungen an die durch Team Yellow zu implementierende Software zur Verwaltung von Patienten mit sozialer Phobie durch eine freischaffende Spitex-Pflegeperson.

Die momentane Situation im Bereich der Patientenverwaltung besteht aus vielen individuellen Teilsystemen. Ein Informationsaustausch zwischen den Systemen ist sehr umständlich.

Das System soll als Software für freischaffende Spitex-Angestellte dienen. In diesem Dokument wird der Fokus auf die Rolle des Health Visitors (Spitex-Angestellte) gelegt.

3. Glossary

Abkürzung	Definition
HMS	**H**ealth **M**anagement **S**ystem
DMZ	**D**emilitarized **Z**one
UML	**U**nified **M**odeling **L**anguage
HTTPS	**H**yper**t**ext **T**ransfer **P**rotocol **S**ecure

4. User requirements definition

4.1 Functional User Requirements

#	Requirement
1	Spitex MitarbeiterIn kann Patienten erfassen
2	Spitex MitarbeiterIn kann Patienten suchen
3	Spitex MitarbeiterIn kann Patienten mutieren
4	Spitex MitarbeiterIn kann Therapie/Behandlung erfassen
5	Spitex MitarbeiterIn kann Therapie/Behandlung durchsuchen
6	Spitex MitarbeiterIn kann Therapie/Behandlung mutieren
7	Spitex MitarbeiterIn kann Ärtzte, Versicherungen, etc verwalten
8	Spitex MitarbeiterIn kann Angehörige, Ärtzte, Versicherungen, etc mit dem Patienten verknüpfen
9	Spitex MitarbeiterIn hat Einsicht in die Medikamente und kann sie vor Ort verabreichen
10	Spitex MitarbeiterIn kann Medikamente für Patientienten inklusive Dosierung im System registrieren
11	Spitex MitarbeiterIn kann Journal erfassen
12	Spitex MitarbeiterIn kann Journal durchsuchen
13	Spitex MitarbeiterIn kann Journal mutieren
14	Spitex MitarbeiterIn kann weiteres Vorgehen erfassen
15	Spitex MitarbeiterIn kann weiteres Vorgehen durchsuchen
16	Spitex MitarbeiterIn kann weiteres Vorgehen mutieren
17	Spitex MitarbeiterIn kann Reaktion auf Behandlung erfassen
18	Spitex MitarbeiterIn kann Reaktion auf Behandlung durchsuchen
19	Spitex MitarbeiterIn kann Reaktion auf Behandlung mutieren
20	Spitex MitarbeiterIn kann Krankheitsverlauf auswerten
21	Spitex MitarbeiterIn kann Krankheitsverlauf drucken
22	Spitex MitarbeiterIn kann Patientenblatt ausdrucken

4.2 Non-Functional User Requirements

#	Requirement
1	Die Daten sind vor unberechtigtem Zugriff geschützt
2	Die Web Applikation ist immer verfügbar (24/7)
3	Die Web Applikation ist performant und hat nur kurze Ladezeiten
4	Die Web Applikation ist benutzerfreundlich
5	Der Support hat eine kurze Reaktionszeit (1 Arbeitstag)
6	Die Web Applikation hat eine Ausfallsicherheit von 98.5%
7	Die Web Applikation ist für zukünftige Anforderungen erweiterbar
8	Die Web Applikation wird laufend weiterentwickelt

4.3 Use-Case Szenarios

4.3.1 Szenario: Eintrag in Krankheitsverlauf

Nr. und Name	01 – Krankheitsverlauf
Szenario	Eintrag in Krankheitsverlauf
Kurze Beschreibung	Die Spitexperson macht einen neuen Eintrag in den Krankheitsverlauf eines Patienten. Dieser Eintrag wird im System abgespeichert.
Beteiligte	Spitexangestellte
Start-Event	Patient muss im System erfasst sein
Resultat	Eintrag im Krankheitsverlauf vorhanden

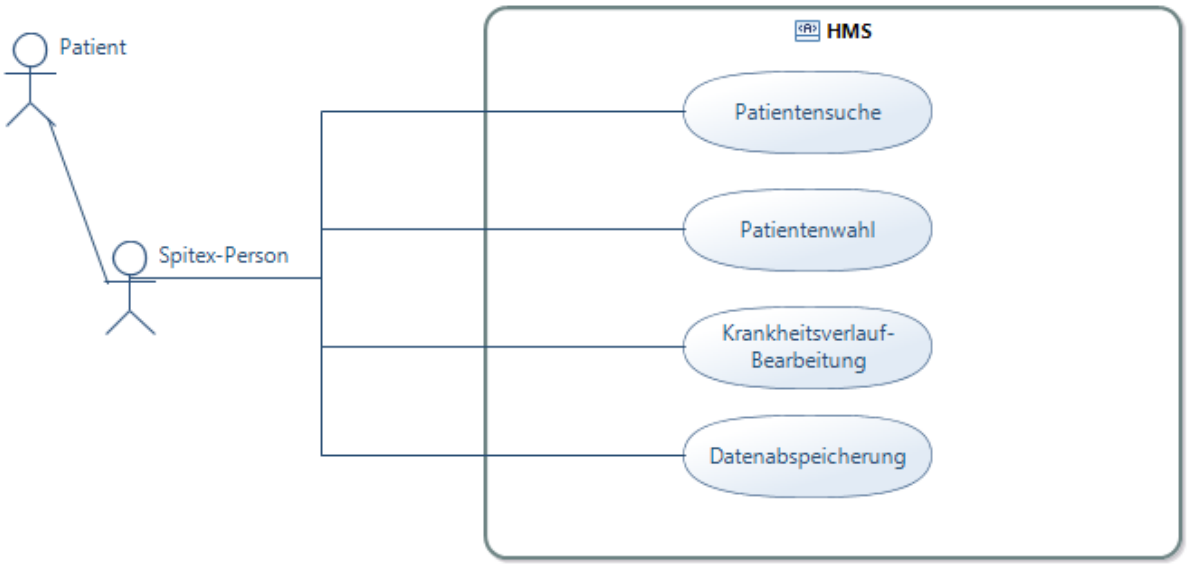
Schritte

Nummer	Beteiligter	Beschreibung
1.0	Spitex-Person	Sucht nach Patient im System
2.0	System	Zeigt eine Liste von Patienten
3.0	Spitex-Person	Wählt gewünschten Patienten
4.0	System	Zeigt Patienten-Info
5.0	Spitex-Person	Navigiert in den Bereich «Krankheitsverlauf»
6.0	Spitex-Person	Füllt das Formular aus
7.0	System	Validiert die Eingabe
8.0	System	Speichert die Eingabe im System

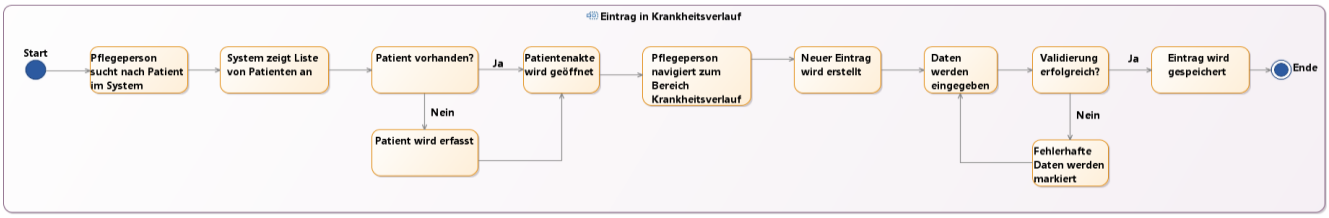
Exceptions

Nummer	Beteiligter	Beschreibung
2.0	System	Patient kann nicht gefunden werden
2.1	System	Zeigt an -> Patient nicht vorhanden
7.0	System	Validierung schlägt fehl
7.1	System	Falsche Eingaben werden markiert
7.2	Spitex-Person	Korrigiert Eingabe
7.3	System	Validiert Eingabe

Use-Case-Diagramm



Aktivitätsdiagramm



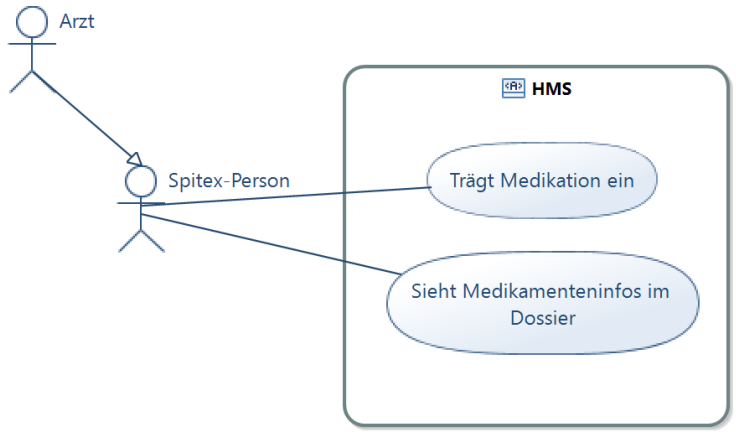
4.3.2 Szenario Medikament inklusive Dosierung erfassen

Nr. und Name	02 – Medikamentenerfassung
Szenario	Dem Patienten wird ein neues Medikament verschrieben
Kurze Beschreibung	Der Arzt verschreibt dem Patienten ein neues Mediakment. Die Spitexperson erfasst dieses zusammen mit der Dosierung im System.
Beteiligte	Patient, Spitex-Angestellte, Arzt
Start-Event	Patient benötigt Medikament, Arzt hat Medikament verschrieben
Resultat	Patient erhält das Medikament in der richtigen Dosierung

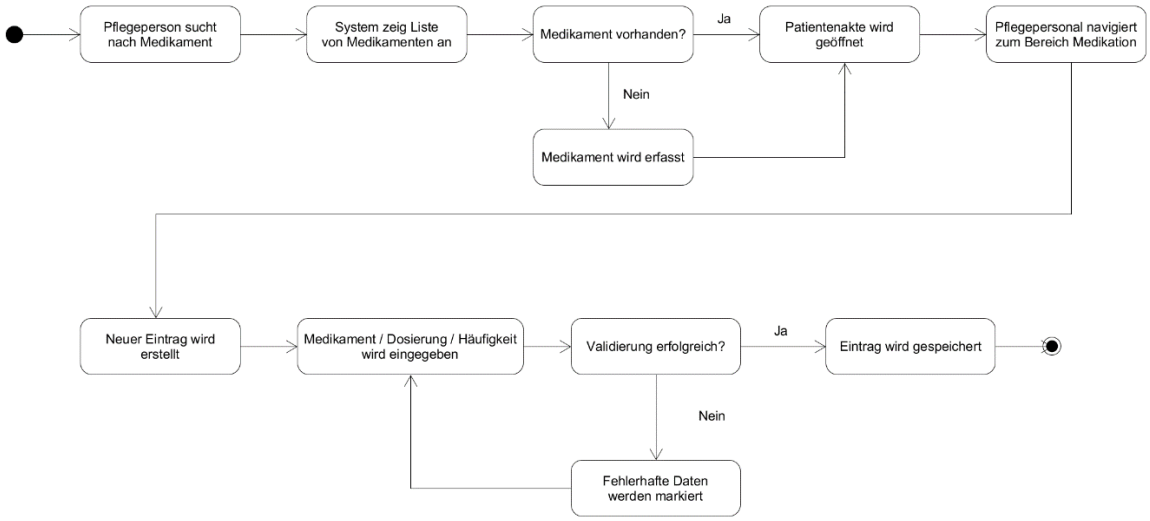
Schritte

Nummer	Beteiligter	Beschreibung
1.0	Patient	Patient konsultiert Arzt
2.0	Arzt	Arzt verschreibt Medikament
3.0	Spitex-Person	Spitex-Person erhält Rezept und trägt Medikament ein.
4.0	Spitex-Person	Trägt Medikation ein (Medikament / Dosierung / Häufigkeit).
5.0	Spitex-Person	Sieht Medikamententinfo in Dossier
6.0	Spitex-Person	Händigt Medikamente aus
7.0	Patient	Konsumiert Medikament

Use-Case-Diagramm



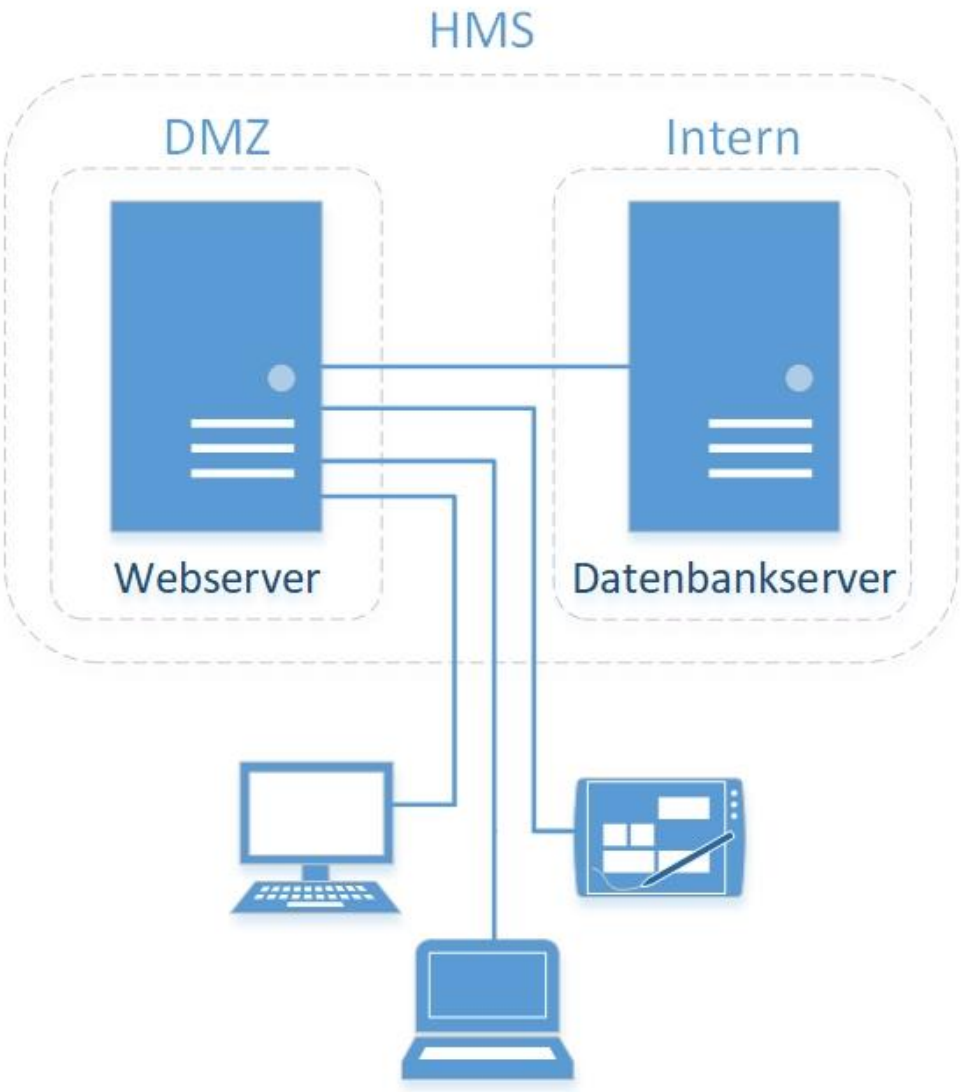
Aktivitätsdiagramm



Exceptions

Nummer	Beteiligter	Beschreibung
3.0	System	Patient kann nicht gefunden werden
3.1	System	Zeigt an -> Patient nicht vorhanden
4.0	System	Validierung schlägt fehl
4.1	System	Falsche Eingaben werden markiert
4.2	Spitex-Person	Korrigiert Eingabe
4.3	System	Validiert Eingabe

5. System architecture



Unsere Webapplikation greift via dem Protokoll HTTPS auf unseren Webserver zu. Auf die Datenbank greift der Webserver direkt zu. Der Client hat keinen Zugriff auf die Datenbank. Die Webapplikation kann von Smartphones, Tablets, Notebooks und Desktops verwendet werden.

6. System requirements

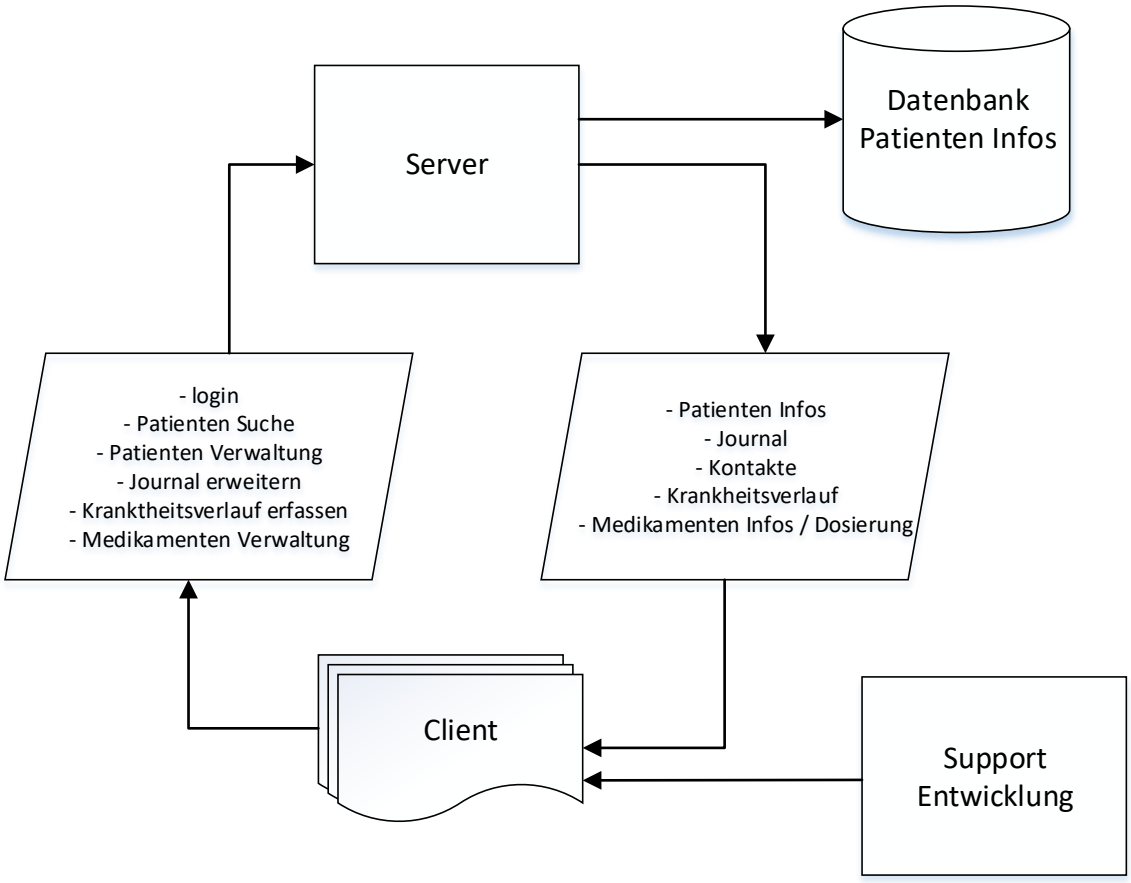
6.1 Functional System Requirements

#	Requirement
1	Daten zusätzlich sichern (Backup)
2	Datenbank als Speicherort der Daten
3	Webserver, damit das Programm über einen Webbrowser verwendet werden kann
4	Sichere Übertragung (HTTPS) zwischen Client und Server
5	Authentifizierung am System

6.2 Non functional System Requirements

#	Requirement
1	Neue Daten sollen auch offline erfasst werden können
2	Die Patientendaten sollen ausreichend geschützt sein gegen Angriffe von aussen
3	Skalierbare Infrastruktur
4	System muss jederzeit verfügbar sein
5	Das System soll flexibel anpassbar und erweiterbar sein für Änderungen
6	Funktionsfähig in allen aktuellen Browsern
7	Applikation auch auf mobilen Geräten (Responsive) brauchbar

7. System models



8. System evolution

Die voraussichtliche Entwicklung der Software für die Anwendergruppe „Ambulantes psychiatrisches Pflegepersonal“ stellt die aktuelle Situation dar. Es ist aber durchaus denkbar, dass in Zukunft die Anwendergruppe erweitert und/oder vergrössert wird. Eine gemeinsame Nutzung der Software durch das Pflegepersonal, den Patienten und dessen Angehörigen, dem Psychiater und dem Arzt könnte in Zukunft also notwendig sein. Ein weiteres Zukunftsszenario kann die Umsetzung von Schnittstellen zu anderen Dienstleitern sein. Mit dem Datenaustausch kann der administrative Aufwand minimiert werden.

Auch sollte die Software für die Behandlung von anderen psychischen Erkrankungen erweiterbar sein.

Auch Hardwaretechnisch kann sich in Zukunft noch einiges ändern. Die Entwicklung einer kompatiblen, plattformunabhängigen Software muss daher gewährleistet sein.

9. Testing

9.1 Komponenten Tests

Der Code wird mit Testmethoden wie z.B. Unittests und durch den Entwickler getestet damit eine hohe Softwarequalität erreicht wird. Weiter soll durch Codereviews ein hoher Standard bezüglich des Codes erreicht werden.

9.2 Integrations Tests

Die "Sprints" sollen jeweils mit einem vollständig funktionierenden Programm abgeschlossen werden. Um dies zu erreichen wird am Schluss eines jeden Sprints durch Integrationstest geprüft ob alle Komponenten gemäss den geplanten Abläufen zusammen funktionieren und die korrekten Ergebnisse liefern.

9.3 System Tests

Ebenfalls am Ende der Sprints werden die Ergebnisse gegen die Spezifikation geprüft. Somit wird klargestellt, dass die Requirements umgesetzt wurden und die Funktionalität gegeben ist. Zusätzlich werden Load Test durchgeführt um sicher zu sein, dass bei voller Auslastung sich das System korrekt verhält.

9.4 Abnahme Test

Am Ende des Projekts werden Abnahmetests zusammen mit den Kunden gemacht.

10. Appendices

Systemvoraussetzung

User:

- Geräte (Computer, Tablet, Handy) mit einer Internetverbindung.
- Browser z.B. Safari, Firefox, Chrome

Server:

- Zuverlässige und schnelle Internetverbindung
- Firewall für die DMZ
- Genügend Speicherplatz für die Daten
- Genügend Serverleistung (Arbeitsspeicher, Prozessor), damit man keine lange Wartezeiten hat