**Team**

Blau

Requirements MHC-PMS

Klinischer Apotheker

Inhaltsverzeichnis

Einführung 2

User Requirements 2

Medikamente bestellen 2

Medikamente Suchen 2

Medikamente herausgeben 2

Erfahrungsbericht erfassen 3

System Models 4

Medikament 4

Hersteller 4

Patient 4

Erfahrungsbericht 4

Systemarchitektur 5

System requirements 6

Use Case Diagram 6

Use Case 1: Medikament herausgeben 6

Use Case 2: Medikament suchen 7

Non-Functional Requirements 7

Testing 8

# Einführung

Dieses Dokument spezifiziert alle Anforderungen an die Mobile Applikation MHC-PMS für klinische Apotheker. Die Applikation hilft dem Apotheker, sein Lager zu verwalten, Medikamente and Krankenschwestern und Patienten herauszugeben und Erfahrungsberichte für Medikamente zu erfassen.

# User Requirements

## Medikamente bestellen

Um das Lager aufzufüllen, muss der Apotheker von verschiedenen Lieferanten Medikamente bestellen. Damit er nicht alle benötigte Medikamente aus der Datenbank heraussuchen und bestellen muss, gibt es in der Applikation eine Funktion zum automatischen bestellen von Medikamenten bei verschiedenen Lieferanten. Es gibt zwei verschiedene Arten von Medikamentenbedarf:

* Täglicher Bedarf von Standard-Medikamenten wie z.B. Morphin
* Bedarf für spezielle Medikamente die nicht regelmässig gebraucht werden, z.B. Krebsmedikamente

Der tägliche Bedarf ist direkt beim Medikament hinterlegt, hier kann die zu bestellende Menge einfach durch Abgleich mit dem Lagerbestand bestimmt werden. Der Bedarf für Spezialmedikamente kann aus der Patientendatenbank generiert werden. Dort ist bei jedem Patient hinterlegt, welche Medikamente er in der nächsten Zeit einnehmen muss.

Aus dieser Bedarfsliste wird nun für jeden Lieferant eine Bestellliste erstellt. Der Apotheker muss die Bestellliste kontrollieren, wenn sie in Ordnung ist wird sie an die Lieferanten verschickt. Dazu wird aus der Liste pro Lieferant ein PDF generiert, und per E-Mail an ihn versendet.

## Medikamente Suchen

Der Apotheker kann nach Medikamenten suchen, wenn er z.B. ein Generika für ein teures Original Medikament sucht. Als Datenquelle dient der pharmINDEX der Firma e-mediat, in der 200’000 Medikamente verzeichnet sind.

Die Suche kann mit folgenden Angaben gefiltert werden:

* Wirkstoff
* Hersteller
* Original/Generika
* Im Lager vorhanden

## Medikamente herausgeben

* Ein Medikamentenbearf wird mittels Patientennamen und Geburtstag eindeutig im Datenbestand gefunden.
* Der Barcode eines Medikaments kann mit Hilfe der Kamera des mobilen Endgerätes gescannt werden.
* Mit dieser Information wird der Medikamentenbestand automatisch verifiziert. Zudem kann das gescannte Medikament so direkt aus dem Lager ausgetragen werden.

## Erfahrungsbericht erfassen

Erfahrungsberichte werden vor allem für Medizinische Studien verwendet, können aber auch für normal eingesetzte Medikamente verwendet werden, um Erfahrung mit z.B. verschiedenen Patientengruppen zu hinterlegen.

# 

# System Models

In diesem Kapitel werden die verwendeten Datenmodelle spezifiziert.

## Medikament

* Name
* Wirkstoff
* Tagesbedarf (Für Standard-Medikamente wie Morphin)
* Aktueller Lagerbestand
* Hersteller
* Lieferant
* Barcode

## Hersteller

* Name
* Kontaktperson
* E-Mail Adressen für Bestellung

## Patient

Der Patient wird nicht in der Apotheker Datenbank gespeichert, sondern in einem externen System, auf das die Applikation zugreifen kann. Folgende Daten werden benötigt:

* Vor- & Nachname
* Geburtsdatum
* Verschriebene Medikamente
  + Welches Medikament?
  + Wann muss es eingenommen werden?
  + Welche Menge?

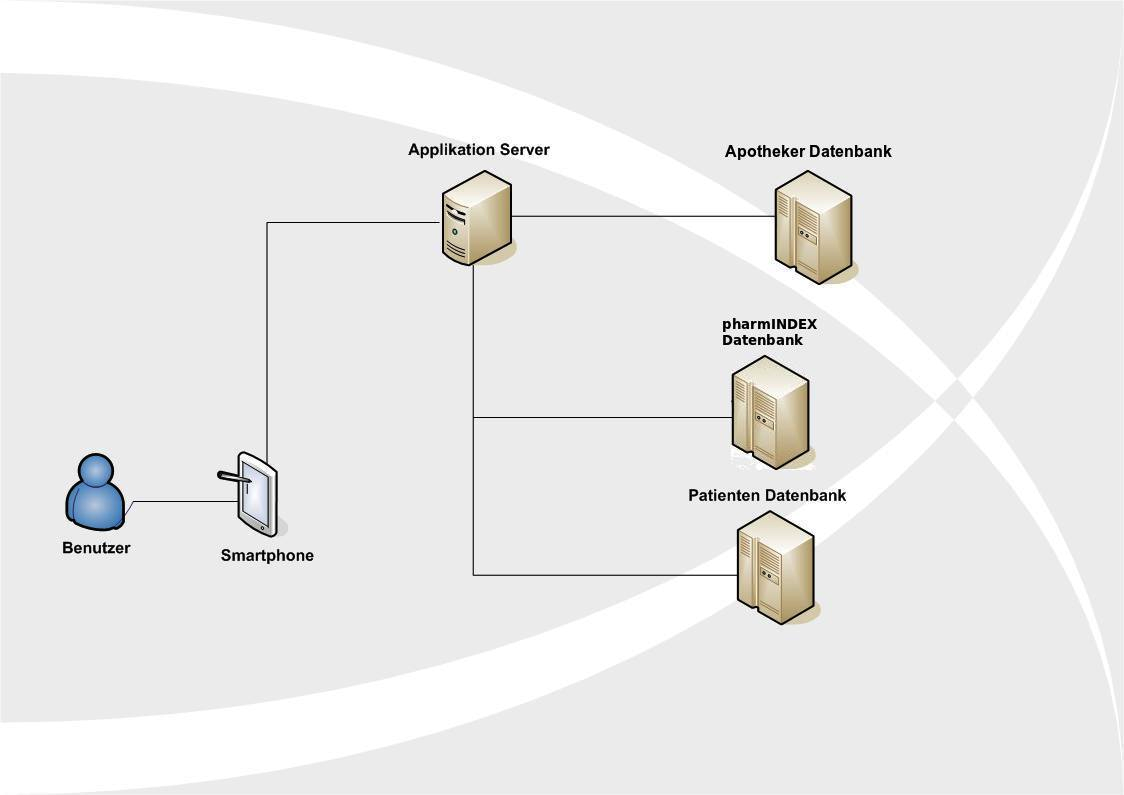
## Erfahrungsbericht

* Patientenkennung
* Eingenommenen Medikamente
* Messwerte (Blutdruck etc.)
* Effektivität
* Nebenwirkung
* Textfeld für Bericht

# 

# 

# Systemarchitektur



Die mobile Applikation kommuniziert mit unserem Applikations-Server. Dieser ist mit drei verschiedenen Datenbanken verbunden.

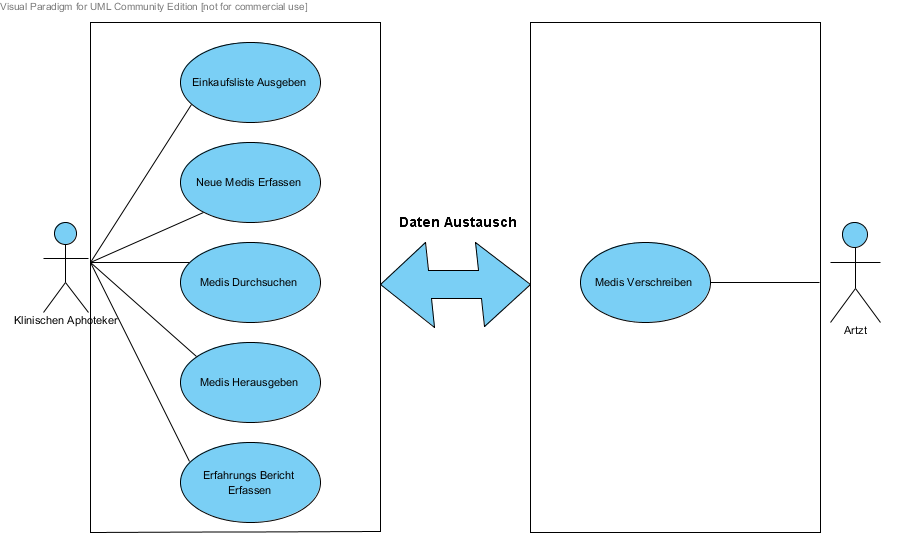
**Apotheker Datenbank:** Lagerbestände / Erfahrungsberichte

**pharmINDEX Datenbank:** offizielle Schweizer Medikamentendatenbank

**Patientendatenbank**: Information über Patient. Relevant für uns ist, welche Medikamente wurden dem Patienten verschrieben.

# System requirements

## Use Case Diagram



## Use Case 1: Medikament herausgeben

|  |  |
| --- | --- |
| Nr. und Name: | Medikament herausgeben (1) |
| Szenario: |  |
| Kurzbeschreibung: | Medikament an Patient/Krankenschwester herausgeben |
| Beteiligt Akteure: | Apotheker, Patient oder Krankenschwester |
| Auslöser /  Vorbe­dingung: | Verschriebene Medikamente sind im System erfasst |
| Ergebnisse /  Nach­bedingung: | Patient/Krankenschwester hat die richtigen Medikamente erhalten.  Medikamente wurden im Lagersystem abgebucht |

Ablauf:

|  |  |  |
| --- | --- | --- |
| Nr. | Wer | Was |
| 1 | Apotheker | Patient der die Medikamente erhalten soll im System suchen |
| 2 | Apotheker | Information über benötigte Medikamente abfragen |
| 3 | Apotheker | Medikamente im Lager holen und einscannen |
| 4 | Apotheker | Medikamente im Lagersystem abbuchen |

## Use Case 2: Medikament suchen

|  |  |
| --- | --- |
| Nr. und Name: | Medikament suchen (2) |
| Szenario: | Apotheker muss ein alternatives Medikament mit dem selben Wirkstoff suchen |
| Kurzbeschreibung: | Apotheker such Medikamente im System nach bestimmten Kriterien |
| Beteiligt Akteure: | Apotheker |
| Auslöser /  Vorbe­dingung: | Medikamente sind mit allen wichtigen Daten erfasst |
| Ergebnisse /  Nach­bedingung: | Apotheker hat Informationen zum Medikament gefunden |

Ablauf:

|  |  |  |
| --- | --- | --- |
| Nr. | Wer | Was |
| 1 | Apotheker | App öffnen |
| 2 | Apotheker | Menüpunkt Medikament suchen wählen |
| 3 | Apotheker | Name des Medikamentes im Suchfeld eingeben |
| 4 | Apotheker | Suche starten |
| 5 | Apotheker | Richtiges Medikament auswählen |
| 6 | Apotheker | Wirkstoff anwählen, es öffnet sich eine Liste mit Medikamenten mit dem selben Wirkstoff |

## Non-Functional Requirements

Sicherheit und Stabilität

* Benutzer muss sich einloggen.
* Verbindung zwischen der mobilen Applikation und dem Server sind verschlüsselt.

# 

# Testing

Um die korrekte Funktion der Applikation zu gewährleisten, werden vier Arten von Tests durchgeführt.

**Modultests**

Während dem Entwicklungsprozess der Software werden Modultest zu den kritischen Komponenten geschrieben, welche automatisch ausgeführt werden. Wird ein Fehler erkannt, wird ein Testfall geschrieben der den Fehler im Nachhinein reproduzieren kann. Somit ist sichergestellt das der Fehler vor einer Einführung behoben wird.

**Funktionale Tests**

Für jede Anforderung an die Applikation werden Testszenarios erfasst. Die Szenarios beinhalten neben dem Sollverhalten auch die Gegentests der Funktionalität. Die funktionalen Tests werden während der Entwicklung durch die Entwickler sowie vor der Abnahme, auch durch den Endbenutzer in Form einer Endprüfung durchgeführt.

**Sicherheitstests**

Die Top-Ten der häufigsten Sicherheitslücken bei mobilen Applikationen sind:

1. Insecure Data Storage

2. Weak Server Side Controls

3. Insufficient Transport Layer Protection

4. Client Side Injection

5. Poor Authorization and Authentication

6. Improper Session Handling

7. Security Decisions Via Untrusted Inputs

8. Side Channel Data Leakage

9. Broken Cryptography

10. Sensitive Information Disclosure

Die Funktionalitäten werden nach Best-Practices entwickelt um keine Sicherheitslücken aktiv einzubauen. In der Testphase wird die Applikation auf die erwähnten Sicherheitslücken geprüft um diese allenfalls zu schliessen.

**Usability Tests**

Neben der korrekten Funktionalität stellen Benutzerfreundlichkeit und Softwareergonomie einen zentralen Erfolgsfaktor der Software dar. In Entwicklungsprojekten, insbesondere bei mobilen Applikationen, werden Usability-Aspekte bereits bei der Realisierung berücksichtigt. Bei Software mit einem grossen Benutzerkreis und/oder komplexen Abläufen führen wir bereits in der Entwicklungsphase Usability-Tests durch und in der Abnahmephase zusätzliche Usability-Tests mit effektiven Benutzern. Solche Tests ermöglichen es, den Schulungs- und Supportaufwand zu reduzieren, sowie die Akzeptanz und die Produktivität bei den Benutzern zu steigern.