



## Software Requirements Specification

**Autoren:**

Michel Utz  
Sabine Zumstein  
Alexander Nussbaum  
Kevin Suter  
Simon Schaad  
Dominique Halter

Bern, im April 2016

## Dokumentenhistorie

Version	Datum	Verantwortliche/r	Änderung(en) / Bemerkung(en)
0.1	30.03.2016	Dominique Halter	Initiale Version (Erstellung der Vorlage)
0.2	31.03.2016	Dominique Halter	Ergänzungen des Vorwortes, der Einleitung und der Testfälle
0.3	01.04.2016	Dominique Halter	Use Cases hinzugefügt, Kapitel System Architektur hinzugefügt
0.4	02.04.2016	Alle Beteiligten	Überarbeitung der Use Cases, Use Case Diagramm hinzugefügt
0.5	02.04.2016	Alle Beteiligten	Systemmodelle ergänzt
0.6	03.04.2016	Kevin Suter	Komponentendiagramm eingefügt, Systemanforderungen ergänzt.
0.7	03.04.2016	Michel Utz	Datenflussdiagramm eingefügt
0.8	04.04.2016	Dominique Halter	Anpassungen Aufbau des Dokumentes und Formatierungen vereinheitlichen.
1.0	04.04.2016	Dominique Halter	Rechtschreibkorrektur, finale Version

## Vorwort

### Zweck des Dokumentes

Die Software-Anforderungen enthalten die Festlegung der Software-Anforderungen als Funktionale-, Nichtfunktionale Anforderungen, Use-Case Modelle und Use-Case Beschreibungen und der Randbedingungen (Abgrenzungen und Abhängigkeiten) aus Benutzer-Sicht für die Applikation „swissMD“. Die Erhebung der Anforderungen und die Erstellung des vorliegenden Dokumentes werden anhand des Standards 29148, herausgegeben von der IEEE, vorgenommen. Kapitel 4.3 des Standards definiert folgende acht Charakteristika einer guten Software Requirements Spezifikation:

- Korrektheit
- Eindeutigkeit
- Konsistent
- Vollständig
- Bewertbar nach Wichtigkeit
- Verifizierbar
- Modifizierbar
- Verfolgbar

Die Kapitel des vorliegenden Dokumentes wurden ebenfalls teilweise dem Standard 29148 entnommen und, wo notwendig, angepasst beziehungsweise ergänzt.

Die Applikation wird im Rahmen des Moduls BTI7081 *Software Engineering and Design* an der Berner Fachhochschule entwickelt.

### Zielgruppe

Das Dokument richtet sich an den Kunden und seine Endbenutzer zur Spezifizierung und Validierung der Anforderungen. Es soll den Psychiatern und Ärzten helfen zu verstehen, wozu die Software genau eingesetzt werden kann und wo allfällig Probleme auftreten können. Im Weiteren richtet sich dieses Dokument intern an die Software-Entwickler als Referenz und zum Verständnis welche Funktionen die Applikation bieten soll, an die System-Architekten als Referenz der Systemvoraussetzungen und an die Testpersonen (intern und extern) als Referenz.

### Sprachliche Gleichstellung

Aus Gründen der besseren Lesbarkeit wird im vorliegenden Dokument auf die gleichzeitige Nennung der männlichen und weiblichen Form verzichtet. Es sind selbstverständlich immer beide Geschlechter gemeint.

## Einleitung

### Ausgangslage

Die regionale Gesundheitsbehörde hat uns beauftragt, ein Patientenmanagement-System (PMS) für den Einsatz im psychiatrischen Umfeld zu entwickeln. Das PMS soll als zentrale Applikation das Patientenmanagement der Patienten, welche unter psychischen Problemen leiden, unterstützen. Dabei sollen sowohl die medizinische als auch die soziale Betreuung berücksichtigt werden. Die meisten psychisch kranken Patienten benötigen keine stationären Aufenthalte in einer Klinik, sondern besuchen regelmässig Fachkliniken oder Praxen wo sie einen Psychiater treffen, welcher ihre Probleme und Diagnosen detailliert kennt. Die Kliniken sind nicht zwingend in einem Spital untergebracht, sie können ebenso in Arztpraxen, lokalen Gesundheitszentren oder Gemeindezentren untergebracht sein. Die regionale Gesundheitsbehörde arbeitet mit einer Reihe von Fachkliniken und Praxen zusammen, in denen die Patienten ihre Termine wahrnehmen können. Dies vereinfacht es den Patienten, ohne längere Anfahrtswege die notwendige Hilfe und Unterstützung in Anspruch zu nehmen. Die Patienten müssen dadurch auch nicht immer die gleiche Klinik oder Praxis konsultieren. Einige der Kliniken bieten auch einen sogenannten Walk-In an. Der Patient kann dabei ohne vorgängigen Termin eine Behandlung in Anspruch nehmen oder seine benötigten Medikamente beziehen. Dies bringt für die psychisch kranken Patienten eine grosse Erleichterung, da diese oftmals schlecht organisiert sind oder Termine vergessen. Auch dem versehentlichen oder absichtlichen Verlust von Medikamenten und den dazugehörigen Rezepten wird dadurch Abhilfe geboten.

Die Applikation soll das Fachpersonal (Psychiater, Ärzte, Pflegefachpersonen, Administration, Management, Spitex) als Endbenutzer bei Ihrer täglichen Arbeit unterstützen. Im Wesentlichen umfasst dies die Bereiche Termin-Management, Medikamenten-Management, Gefährdungs-Management, Diagnosen-Management, Notfall-Management und Krankengeschichte. Die Benutzer der Applikation umfassen sowohl jüngere, als auch ältere Personen, welche möglicherweise weniger technikaffin sind. Die Applikation umfasst eine Datenbank mit sämtlichen Patientendaten und die Applikation selbst. Das Produkt soll plattformunabhängig und internationalisierbar sein.

Das Team Yellow entwickelt die Applikation ausgerichtet auf das ärztliche Personal, darunter fallen Psychiater, Psychologen und Ärzte. Die weiteren Teile oder Module der Applikation für die Patienten und das paramedizinische Personal werden nicht abgedeckt.

Sämtliche gesetzlichen Vorgaben betreffend Datenschutz und Gesundheitsgesetz müssen zwingend eingehalten werden. Besonderes Augenmerk legen wir hierbei auf den Schutz gefährdeter Personen vor sich selbst und auf den Schutz der Gesellschaft vor gefährlichen Personen.

## Glossar

Begriff	Erklärung
AMBV	Verordnung über die Bewilligung im Arzneimittelbereich.
Application Server	Computerprogramm, welches eine Laufumgebung für ein System zur Verfügung stellt, wird meist zentral betrieben.
Behandlung	Eine Behandlung umfasst sämtliche Tätigkeiten, welche ein Arzt und/oder eine Pflegefachperson für einen Patienten erbringt.
Datenbank	Ein System zur elektronischen Datenverwaltung, die Daten werden dabei in einem relationalen Datenbankmodell gespeichert und zur Verfügung gestellt.
Diagnose	Die Diagnose beschreibt die Erkennung von Krankheiten mittels verschiedenster Untersuchungen.
DSG	Bundesgesetz über den Datenschutz.
Fürsorgerische Unterbringung	Wenn Personen mit bestimmten Schwächezuständen nicht anders behandelt oder betreut werden können, dürfen sie in einer geeigneten Einrichtung untergebracht werden. Die Massnahmen gemäss Art. 426 ff ZGB umfassen die Fürsorgerische Unterbringung (FU) oder die Zurückbehaltung einer freiwillig eingetretenen Person in einer geeigneten Einrichtung.
Gefährdung	Die Gefährdung beschreibt die Gefahr, welche von einem Patienten zum Nachteil von sich selbst oder Drittpersonen ausgeht.
GUI	<b>G</b> raphical <b>U</b> ser <b>I</b> nterface. Benutzeroberflächen, dessen Inhalte auf dem Bildschirm pixelgenau positioniert werden.
History	Unter History wird die Krankheitsgeschichte eines Patienten verstanden. Sie umfasst sämtliche Angaben über die in der Vergangenheit gestellten Diagnosen, die verordneten Medikationen und Sitzungsnotizen.
HL7	“Health Level Seven” hat seinen Ursprung 1987 in den USA genommen und beschreibt einen international anerkannten Standard im medizinischen Umfeld zur Datenübermittlung. Aktuell wird der Standard in der Version 3.0 (XML) angeboten. Die Schweiz gehört zu den Mitgliedstaaten der HL7.org Dachorganisation.
HMG	Bundesgesetz über Arzneimittel und Medizinprodukte
HTTPS	HTTPS ( <b>h</b> ypertext <b>t</b> ransfer <b>p</b> rotocol <b>s</b> ecure) ist ein Abkömmling von HTTP, bei dem die Daten von einer Webseite verschlüsselt übertragen werden. Dabei wird die Verschlüsselungs-"Methode" SSL angewandt.
IaaS	Infrastructure as a Service ist im Cloud Computing die Bereitstellung von virtualisierter IT-Infrastruktur über öffentliche oder private Netzwerke, meist über das Internet.
ICD	<b>I</b> nternational <b>S</b> tatistical <b>C</b> lassification of <b>D</b> iseases and <b>R</b> elated <b>H</b> ealth <b>P</b> roblems. Das wichtigste, weltweite anerkannte Diagnoseklassifikationssystem der Medizin welches von der WHO herausgegeben wird.
IEEE	<b>I</b> nstitute of <b>E</b> lectrical and <b>E</b> lectronic <b>E</b> ngineers. Ein weltweiter

	Berufsverband von Ingenieuren mit Sitz in New York (USA), der nebst der Herausgabe diverser Fachzeitschriften hauptsächlich in der Standardisierung von Techniken, Hardware und Software tätig ist.
Klinik	Eine Klinik ist eine Einrichtung im Gesundheitswesen welche ambulante und/oder stationäre Behandlungsmöglichkeiten für Patienten anbietet.
KVG	Bundesgesetz über die Krankenversicherung
Medikament	Ein Medikament ist Teil einer Medikation. Es kann entweder durch den Patienten selbst bezogen (Medikamente Kategorien C und D gemäss Swissmedic) oder durch einen Arzt/Apotheker verschrieben werden (Medikamente Kategorien A und B).
Patient	Ein Patient ist ein Kunde des Psychiaters, des Arztes oder einer Klinik
PC	Ein <b>P</b> ersonal <b>C</b> omputer ist ein Mikrocomputer, der im Gegensatz zu einem Minirechner oder Großrechner von nur einem einzigen Benutzer persönlich bedient und genutzt wird.
PMS	<b>P</b> atient <b>M</b> anagement <b>S</b> ystem, ein System zur zentralen Verwaltung und Verarbeitung von Patienten- und Falldaten.
Responsive Design	Beim Responsive Webdesign handelt es sich um einen gestalterischen und technischen Ansatz zur Erstellung von Websites, so dass diese Websites auf Eigenschaften des jeweils benutzten Endgeräts reagieren können.
SQL	Die <b>S</b> tructured <b>Q</b> uery <b>L</b> anguage ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen.
Swissmedic	Zum Schutz von Mensch und Tier gewährleistet Swissmedic, das Schweizerische Heilmittelinstitut, dass nur qualitativ hochstehende, sichere und wirksame Heilmittel in der Schweiz in Verkehr gebracht werden.
Tablet	Ein Tablet oder Tabletcomputer, ist ein tragbarer, flacher Computer in besonders leichter Ausführung mit einem Touchscreen, aber, anders als bei Notebooks, ohne ausklappbare mechanische Tastatur.
Use Case	Ein Use Case (Anwendungsfall) beschreibt alle möglichen Szenarien, die eintreten können, sobald ein Benutzer versucht mit Hilfe des Systems ein bestimmtes Ziel zu erreichen.
VAM	Verordnung über die Arzneimittel
Webbrowser	Der Browser ist ein Programm zur grafischen Darstellung der Inhalte des World Wide Web (WWW), welches neben HTTP noch andere Dienste wie FTP unterstützt.
WHO	<b>W</b> orld <b>H</b> ealth <b>O</b> rganization. Die WHO ist eine Sonderorganisation der Vereinten Nationen mit Sitz in Genf, CH.

## Inhaltsverzeichnis

1	Benutzeranforderungen .....	1
1.1	Funktionelle Anforderungen .....	1
1.2	Nicht-Funktionelle Anforderungen .....	2
1.3	Use-Case Diagramme .....	3
1.4	Use Case Szenarien .....	4
1.4.1	Patient Beistand zuweisen .....	4
1.4.2	Medikation anpassen .....	5
2	System-Architektur .....	6
2.1	Überblick .....	6
2.2	Data Layer (Data Access Logic) .....	7
2.3	Backend (Business Logic) .....	7
2.4	Web (Presentation Logic) .....	7
2.5	User .....	8
2.6	Umsysteme .....	8
2.7	Infrastruktur .....	8
2.8	Authentifikation .....	8
2.9	Security .....	8
3	Systemanforderungen .....	9
3.1	Funktionelle Anforderungen .....	9
3.2	Nicht-Funktionelle Anforderungen .....	10
3.2.1	Produktanforderungen .....	10
3.2.2	Organisatorische Anforderungen .....	10
3.2.3	Externe Anforderungen .....	10
4	Systemmodelle .....	11
4.1	Komponentenmodell .....	11
4.2	Schnittstellen innerhalb des Application Server .....	12
4.3	Datenflussmodell .....	13
5	Weiterentwicklung des Systems .....	14
6	Testfälle .....	15
6.1	White-Box-Test .....	15
6.2	Black-Box-Test .....	16
	Abbildungsverzeichnis .....	17
	Tabellenverzeichnis .....	17
	Referenzen .....	17

# 1 Benutzeranforderungen

## 1.1 Funktionelle Anforderungen

Bereich	Funktionen
Krankengeschichte	<ul style="list-style-type: none"> <li>Auf der View können folgende Dinge angezeigt werden                             <ul style="list-style-type: none"> <li>Patient auswählen (Dropdown inkl Suchfunktion)</li> <li>Allgemeine Grunddaten</li> <li>Medizinische Grunddaten (inkl Daten der Sozialdienste)</li> <li>Die Krankheiten des gewählten Patienten</li> <li>Übersicht der Medikamente (aktive und inaktive), inkl Einnahme-Regelmässigkeit und Gültigkeit/Laufzeit des Rezeptes</li> <li>Übersicht über alle erfassten Sitzungen</li> </ul> </li> <li>Alle Boxen müssen per Klick auf das Minus-Symbol einzeilig gemacht werden können. Ist eine Box minimiert, erscheint anstatt dem Minus-Symbol eine Plus-Symbol, über welches man per Klick die Box wieder auf die normale Grösse bringt</li> <li>Inaktive Medikamente können mit einer Checkbox angezeigt oder versteckt werden</li> <li>Die Anzeige muss druckbar sein</li> <li>Alle Boxen bis auf die Medikations-Box müssen per Klick auf das Bleistift-Symbol (jeweils oben rechts in der Box) editierbar sein</li> <li>Medikationen müssen per Klick auf sich selber editierbar sein</li> </ul>
Sitzungsnotizen	<ul style="list-style-type: none"> <li>Auf der View können folgende Dinge angezeigt werden                             <ul style="list-style-type: none"> <li>aktuell verschriebene Medikamente</li> <li>In der Vergangenheit verschriebene, nun inaktive Medikamente</li> <li>Der aktuelle Patient</li> <li>Die Krankheiten des gewählten Patienten</li> <li>Die Gültigkeit und Laufzeit des Rezeptes</li> </ul> </li> <li>Inaktive Medikamente können mit einer Checkbox angezeigt oder versteckt werden</li> <li>Die Anzeige muss druckbar sein</li> <li>Es müssen neue Medikationen erfasst werden können</li> <li>Medikationen müssen entfernt werden können</li> </ul>

Tabelle 1 Funktionelle Benutzeranforderungen



## **1.2 Nicht-Funktionelle Anforderungen**

- Sämtliche gesetzliche Vorgaben werden eingehalten. Insbesondere die Vorgaben hinsichtlich Datenschutz und Gefährdung.
- Die Applikation kann während einer Sitzung (Behandlung) oder eines Meetings auf dem Tablet benutzt werden.
- Die Applikation kann auf einem Desktop-Computer oder Notebook verwendet werden.
- Das GUI soll einfach und intuitiv aufgebaut sein.
- Die Sprache der Software ist Deutsch, es soll möglich sein zu einem späteren Zeitpunkt weitere Sprache zu integrieren (nicht Bestandteil dieses Projektes).
- Ein Psychiater kann nur die Daten seiner eigenen Patienten bearbeiten oder abändern.
- Es ist nur Psychiatern gestattet Medikamente zu verordnen oder die Dosierung einer bestehenden Medikation zu ändern. Die weiteren Benutzergruppen haben nur Lesezugriff auf diese Informationen.
- Die Authentisierung an der Applikation geschieht mittels verschlüsseltem Benutzername und Passwort.
- Die Applikation ist plattformunabhängig.

### 1.3 Use-Case Diagramme

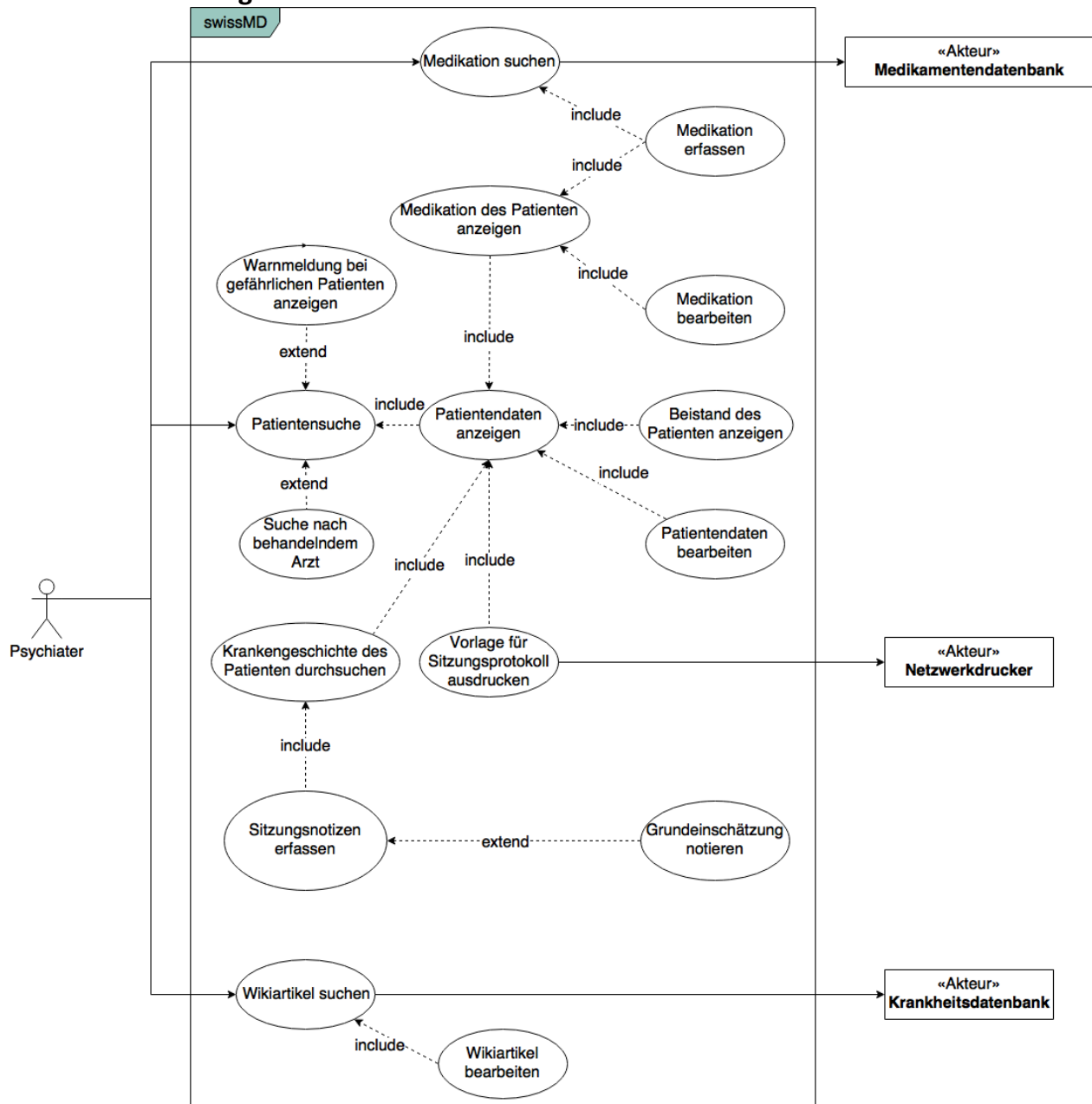


Abbildung 1 Use-Case-Diagramm Übersicht

## 1.4 Use Case Szenarien

### 1.4.1 Patient Beistand zuweisen

Nr. und Name:	2 Patient Beistand zuweisen
Szenario:	Der Patient wird für unmündig erklärt. Der Psychiater möchte den nun dem Patienten zugeteilten Beistand im System speichern
Kurzbeschreibung:	In der Krankengeschichte sind die medizinischen Grunddaten in einer Box aufgeführt. In dieser Box gibt es ein Feld „Beistand“
Beteiligt Akteure:	Psychiater und Applikation
Auslöser / Vorbedingung:	Psychiater möchte den neuen Beistand im System erfassen
Ergebnisse / Nachbedingung:	Psychiater sieht nach erfolgreichem Erfassen des Beistandes den Eintrag in der Box „Medizinische Grunddaten“

#### Ablauf:

Nr.	Wer	Was
1.0	Psychiater	Klickt auf den Abschnitt Krankengeschichte
1.1	Applikation	Zeigt den Abschnitt Krankengeschichte an
1.2	Psychiater	Wählt gewünschten Patienten aus
1.3	Applikation	Zeigt den Abschnitt Krankengeschichte für den Patienten an
1.4	Psychiater	Klickt auf „bearbeiten“ (Bleistift-Symbol) in der Box „Medizinische Grunddaten“
1.5	Applikation	Formular mit den medizinischen Grunddaten wird geöffnet
1.6	Psychiater	Gibt im Feld „Beistand“ die gewünschten Daten ein und klickt speichern (Disketten-Symbol oder Ähnliches)
1.7	Applikation	Speichert und zeigt wieder die nun aktualisierte Krankengeschichte an

#### Ausnahmen, Varianten:

Nr.	Wer	Was
Alle	Applikation	Die Applikation stürzt ab oder reagiert nicht mehr. → Die bereits erfassten Daten wurden zwischengespeichert und es ist nur ein vernachlässigbarer Teil verloren. Somit kann der Therapeut in der Medizinische Grunddaten-Bearbeiten-View fortfahren.

Tabelle 2 Use Case Form 1 Patient Beistand zuweisen

**1.4.2 Medikation anpassen**

Nr. und Name:	1 Medikation anpassen
Szenario:	Psychiater überprüft und ändert Medikation
Kurzbeschreibung:	Im Sitzungsprotokoll ist die Medikamenten-History verlinkt oder eingebettet. Der Button „Neues Medikament verschreiben“ führt zur Maske für das Ausstellen von einem neuen Rezept.
Beteiligt Akteure:	Psychiater und Applikation
Auslöser / Vorbedingung:	Psychiater möchte die Medikation des Patienten ansehen und verändern.
Ergebnisse / Nachbedingung:	Psychiater wird auf Seite geleitet, wo ein neues Rezept ausgestellt werden kann.

**Ablauf:**

Nr.	Wer	Was
1.0	Psychiater	Klickt auf den Abschnitt Medikation
1.1	Applikation	Zeigt den Abschnitt Medikation für den gewählten Patienten an
1.2	Psychiater	Überprüft die History und entscheidet, dass ein anderes Medikament nötig ist. Klickt auf Gültigkeitsdatum des Medikaments, das nicht mehr benötigt wird
1.3	Applikation	Zeigt kleinen Kalender an (Data Chooser)
1.4	Psychiater	Wählt Datum, ab dem das aktuelle Medikament nicht mehr gültig ist
1.5	Applikation	Kalender verschwindet und angeklicktes Datum wird neu im Feld des Endtermins angezeigt
1.6	Psychiater	Klickt auf „Neues Medikament verschreiben“ (Plus-Zeichen oder Ähnliches)
1.7	Applikation	Wechselt zu Abschnitt / Maske „Neues Medikament verschreiben“

**Ausnahmen, Varianten:**

Nr.	Wer	Was
Alle	Applikation	Die Applikation stürzt ab oder reagiert nicht mehr. → Die bereits erfassten Daten wurden zwischengespeichert und es ist nur ein vernachlässigbarer Teil verloren. Somit kann der Therapeut in der Medikation-View mit der Bearbeitung fortfahren.
1.2	Psychiater	Der Anwender entscheidet, keine Medikamente zu verschreiben. → Die Applikation bietet ihm eine Möglichkeit, die View zu verlassen und dorthin zurückzukehren, wo er herkam.

Tabelle 3 Use Case Form 2 Medikation anpassen

## 2 System-Architektur

### 2.1 Überblick

Die Systemarchitektur unserer Applikation “swissMD” besteht im Wesentlichen aus den in Abbildung 2 ersichtlichen Einzelkomponenten. Das vorliegende Kapitel umfasst eine kurze Beschreibung der einzelnen Komponenten. Mit dem gewählten Client-Server-Modell werden unter anderem die domänenspezifischen Anforderungen an Sicherheit und Datenschutz gewährleistet. Die zentrale Speicherung der Daten erlaubt es, diese laufend auf gesetzliche Konformität zu prüfen und die Zugriffsbeschränkungen für alle Benutzer zentral zu verwalten.

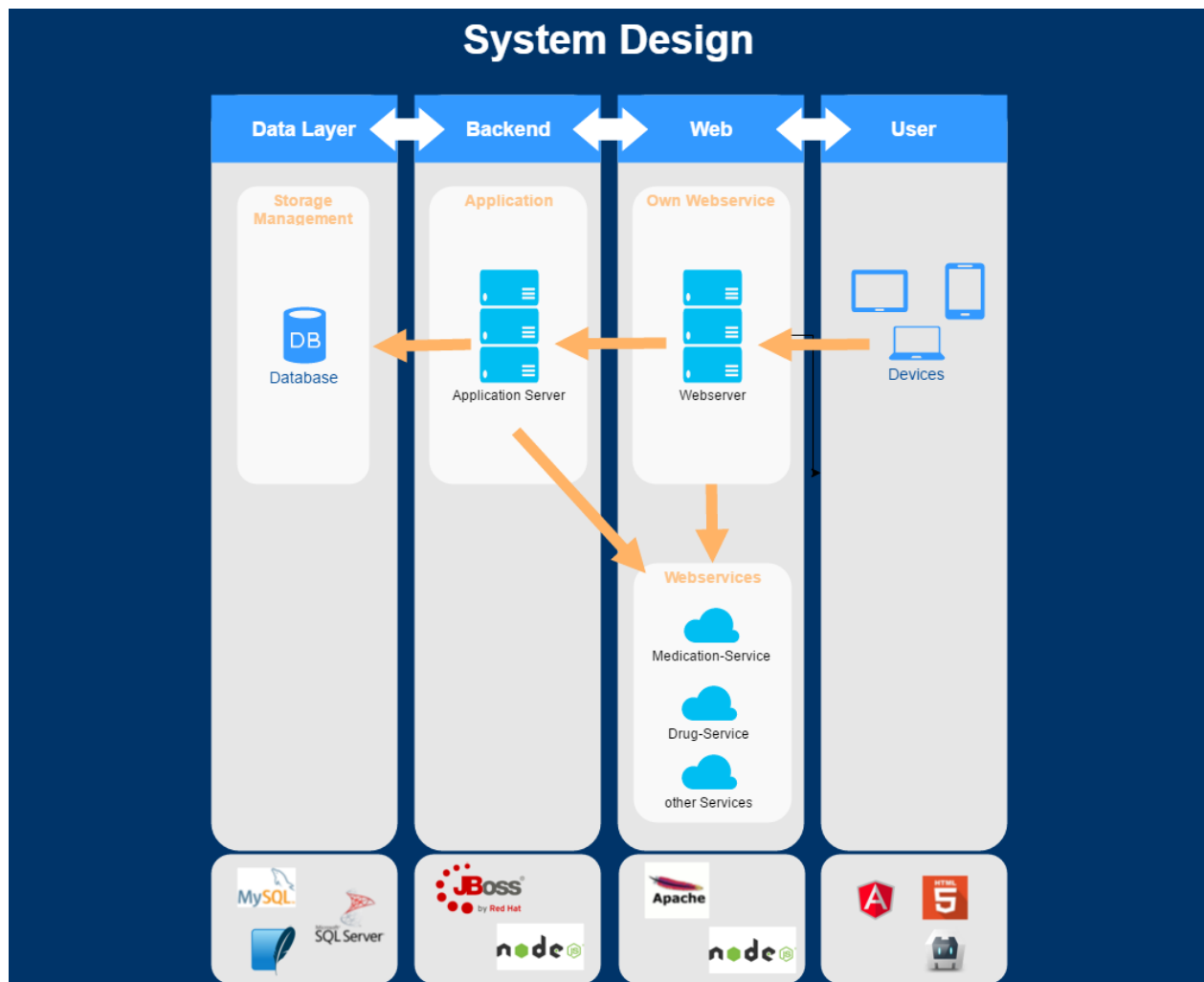


Abbildung 2 Überblick System-Architektur

## 2.2 Data Layer (Data Access Logic)

Auf der Ebene Data Layer setzen wir einen Datenbankserver zur Speicherung der Daten unserer Applikation ein. An den Datenbankserver werden besonders hohe Anforderungen gestellt, da die gespeicherten Patientendaten in jedem Falle datenschutzrechtlich von grosser Relevanz sind. Sie gehören zur Kategorie der besonders schützenswerten Personendaten (Art. 3 Datenschutzgesetz DSG).

Der Datenbankserver besteht aus folgenden zwei Komponenten:

- Datenbankmanagementsystem (DBMS)
- Datenbank

Das *DBMS* repräsentiert die eingesetzte Datenbanksoftware, sie legt das Datenbankmodell fest und verwaltet die logisch zusammengehörigen Datenbestände der Datenbank. Das DBMS steuert sämtliche Prozesse der Datenbank und stellt weitere Dienstleistungen zur Verfügung.

Unter *Datenbank* oder auch Datenbasis versteht man eine logisch zusammengehörige Ansammlung von Daten. Die Daten werden dabei nach einheitlichen Regeln gespeichert (persistente Datenhaltung) und meist in Tabellen oder als Objekte abgebildet. Nebst den eigentlichen Daten wird in der Datenbank auch das Data Dictionary gespeichert. In diesem Dictionary werden die Metainformationen des DBMS zur Speicherung und Verwaltung der Anwendungsdaten abgelegt. Diese Informationen betreffen typischerweise Metadaten welche die Datenbasis definieren, zum Beispiel die Tabellenstruktur.

Der Datenbankserver stellt eine Query-Language zur Verfügung, über welche der Application Server (siehe 2.3 Backend) die Daten abfragen und speichern kann. Die Query-Language hängt vom eingesetzten Datenbanksystem ab, typischerweise wird hierzu jedoch SQL eingesetzt. Als Datenbanksystem kann typischerweise Microsoft SQL, MySQL oder eine Oracle Datenbank zum Einsatz kommen.

## 2.3 Backend (Business Logic)

Das Backend, oder auch Application Server genannt, übernimmt die Logik der Applikation. Der Application Server steht für ein Rahmenwerk, welches die Entwicklung von Webanwendungen erleichtert und einen Server bereitstellt. Das Backend befindet sich dabei zwischen der Datenhaltung (Datenbank) und der Präsentation (Web). Der Application Server stellt über definierte Schnittstellen spezielle Dienste zur Verfügung, wie beispielsweise Transaktionen, Authentifizierung oder den Zugriff auf die Verzeichnisdienste wie das Datenbanksystem.

## 2.4 Web (Presentation Logic)

Die Ebene Web kann grob in zwei Bereiche unterteilt werden: Der eigene Webserver reagiert auf die Clientanfragen und liefert die entsprechenden Inhalte zurück. Diese Inhalte werden aus den Daten der Datenbank oder externen Webservices aufgebaut. Der eigene Webserver übernimmt des Weiteren die Funktion der Lastverteilung. Um die beiden Ebenen Backend (siehe 2.3) und Web abzudecken bietet sich gegeben falls der Einsatz eines Web Application Containers an. Ein solches System bietet eine Kombination aus Web- und Application Server.

Unter externe Webservices fallen externe Daten, welche bezogen werden. Diese Umsysteme sind unter 2.6 näher beschreiben.

## 2.5 User

Der Endbenutzer greift von seinem Gerät aus via einem Webbrowser auf die Applikation zu. Als Geräte können alle Arten der gegenwärtig auf dem Markt verfügbaren Modelle zum Einsatz kommen. Typischerweise wird die Applikation auf einem Personal Computer (PC), einem Laptop oder einem Tablet genutzt. Der Webbrowser übernimmt die grafische Darstellung der Webseiten und der Dokumente. Die Applikation unterstützt die gängigen, auf dem Markt verfügbaren, Webbrowser wie zum Beispiel Firefox, Google Chrome oder Microsoft Internet Explorer. Es wird davon ausgegangen, dass die Nutzung der Applikation normalerweise stationär stattfindet. Daher ist ein ständiger Internetzugang vorhanden und die Applikation wird keinen Offline-Modus zur Verfügung stellen.

## 2.6 Umsysteme

Eine Anbindung an bestehende Umsysteme ist durch das Client-Server-Modell ebenfalls möglich. Dadurch wird es möglich auf die Daten weiterer Umsysteme zuzugreifen und deren Funktionalität mit zu nutzen. Es soll eine möglichst umfassende Integration in die bestehende Umgebung angestrebt werden. Dazu sollen auch Schnittstellen für andere Systeme zur Verfügung gestellt werden. Typischerweise wird für Schnittstellen im medizinischen Umfeld das Datenformat HL7 genutzt. Das HL7-Format ist international anerkannt und wird von nahezu jedem Hersteller implementiert. Konkret sollten unter anderem folgende Umsysteme eingebunden werden können:

- Medikamentendatenbank
- Medizinische Datenbanken (zum Beispiel ICD10-Datenbank)
- Weitere Patientensysteme (zum Beispiel SAP oder KONAS)

## 2.7 Infrastruktur

Um die Kosten für den Aufbau und Betrieb der Applikation so tief wie möglich zu halten wird eine virtuelle Infrastruktur angestrebt. Diese virtuelle Infrastruktur soll entweder durch den Kunden selbst angeboten werden, wobei hier die Anforderungen der Applikation hinsichtlich Performance, Security und Kompatibilität zu beachten sind. Alternativ bietet sich auch die Möglichkeit, die Infrastruktur als Infrastructure-as-a-Service (IaaS) bei einem Cloud-Provider wie zum Beispiel Amazon Web Service (AWS), Google Compute Engine, Microsoft Azure oder IBM SmartCloud zu beziehen.

## 2.8 Authentifikation

Auf Seiten der Kunden wird zwar partiell ein Directory-Service (zum Beispiel Microsoft Active Directory) zur Verfügung stehen. Da die Applikation jedoch gemäss Anforderungen des Kunden regional in verschiedenen Kliniken und Praxen eingesetzt werden soll, ist eine eigene Benutzerverwaltung und Authentifizierung vorgesehen. Eine Anbindung der unterschiedlichen Directory-Services ist aufgrund der Komplexität nicht sinnvoll.

## 2.9 Security

Die Kommunikation zwischen dem Client und dem Webserver soll verschlüsselt über HTTPS erfolgen. Die Applikationsinterne Kommunikation zwischen Webserver, Application Server und dem Datenbanksystem wird nur wo notwendig und technisch realisierbar verschlüsselt.

Für das Datenbanksystem wird ein regelmässiges und automatisiertes Backup angestrebt. Die meisten Datenbanksysteme bieten diese Funktionalität bereits an.

Die Applikation wird als Standalone-Lösung aufgebaut, eine Redundanz in Form eines Backups ist nicht vorgesehen.

### 3 Systemanforderungen

#### 3.1 Funktionelle Anforderungen

Bereich	Funktionen
Patientendaten	<ul style="list-style-type: none"> <li>• Technische Möglichkeit die Patientendaten auszudrucken (z.B. Anbindung an bestehenden Druckserver)</li> <li>• Während der Suche nach Dokumenten oder Stichworten in der Patientenhistorie soll die Ergebnisliste während der Eingabe aktualisiert werden</li> <li>• In der Datenbank müssen vordefinierte Gefährdungskategorien zur Einstufung der Patienten geführt werden</li> <li>• Es soll möglich sein, medizinische oder weitere Daten von externen Systemen in den Stammdaten des Patienten anzuzeigen (z.B. Laborwerte von Bluttests, Daten der Sozialdienste). Diese Daten sollen über eine HL7-Schnittstelle ausgetauscht werden.</li> <li>• Es soll möglich sein, jedem Patient seine bisherigen Sitzungen zuzuordnen und diese anzuzeigen</li> </ul>
Sitzung	<ul style="list-style-type: none"> <li>• Jeder Sitzung müssen ein Datum, ein Arzt sowie Sitzungsnotizen beigelegt werden können</li> </ul>
Sitzungsnotizen	<ul style="list-style-type: none"> <li>• Wenn während einer Sitzung Medikamente verordnet werden, soll die Ergebnisliste während der Eingabe aktualisiert werden. Diese Real-Time-Suche muss performant aufgebaut sein, da die Medikamentendatenbank sehr viele Einträge umfasst</li> <li>• Die Behandlungen inklusive Diagnosen und Medikamenten können mit einem Termin verknüpft werden</li> <li>• Möglichkeit, ein eigene Vorlage mit vordefinierten Fragen für die Sitzungsnotizen zu erstellen</li> </ul>
Medikamente	<ul style="list-style-type: none"> <li>• Externe Medikamentendatenbank abfragen</li> <li>• Jede Behandlung hat 0 bis unendlich viele Medikamentenverschreibungen</li> <li>• Zu einer Verschreibung gehören das Medikament, die Dosis wie auch die Einnahmehäufigkeit und –dauer</li> <li>• Das Layout für ein korrektes Medikamentenrezept muss hinterlegt sein. Das Layout muss den Vorgaben in den geltenden Gesetzen (HMG, AMBV, VAM, KVG) entsprechen.</li> </ul>
Krankheiten	<ul style="list-style-type: none"> <li>• Externe Datenbank der Krankheiten abfragen</li> <li>• Einem Patienten muss eine Krankheit zugewiesen werden können mit Kommentar zur Diagnose</li> </ul>
Personensuche	<ul style="list-style-type: none"> <li>• Suche nach Patienten oder Ärzten in der Datenbank mit Wildcards</li> <li>• Tabellarische Darstellung der Resultate</li> <li>• Möglichkeit, von der Personensuche direkt auf die Patientendaten eines einzelnen Patienten zu gelangen</li> </ul>
Wiki	<ul style="list-style-type: none"> <li>• Es muss ein Datenbestand von Wikieinträgen eingelesen und angezeigt werden können</li> <li>• Möglichkeit, einen Wikieintrag zu bearbeiten bzw. zu kommentieren</li> </ul>

Tabelle 4 Funktionelle Systemanforderungen



## **3.2 Nicht-Funktionelle Anforderungen**

### **3.2.1 Produktanforderungen**

- Das Login muss die Sicherheitsanforderungen (siehe 2.8 und 2.9) erfüllen
- Die Vorgaben der geltenden Datenschutzgesetze müssen eingehalten werden
- Der Psychiater darf nur die Daten seiner eigenen Patienten bearbeiten
- Benutzerhandbuch auf Deutsch
- Schulung der Endbenutzer
- Die Datenbank benötigt genug Speicherplatz, um die Daten von 50 Ärzten und 100 Patienten fassen zu können
- Einheitliches Design des GUI, das Master Design muss überall eingehalten werden
- Das GUI soll in einem Responsive Design erstellt sein
- Umsetzung des objektorientierten Ansatzes
- Die Applikation sollte für mobile Geräte optimiert sein

### **3.2.2 Organisatorische Anforderungen**

- Versionsverwaltung mit Git
- Sourcecodeverwaltung mit Github
- Eclipse als Entwicklungsumgebung
- Fertigstellung im FS2016

### **3.2.3 Externe Anforderungen**

- Das Datenschutzgesetz bezüglich Schutz von Personen- und Gesundheitsdaten muss eingehalten werden
- Gemäss Gesundheitsgesetz muss es die Software ermöglichen, Patienten, die eine unmittelbare Gefahr für sich oder die Umwelt darstellen, in einer geeigneten Einrichtung zurückzubehalten

## 4 Systemmodelle

### 4.1 Komponentenmodell

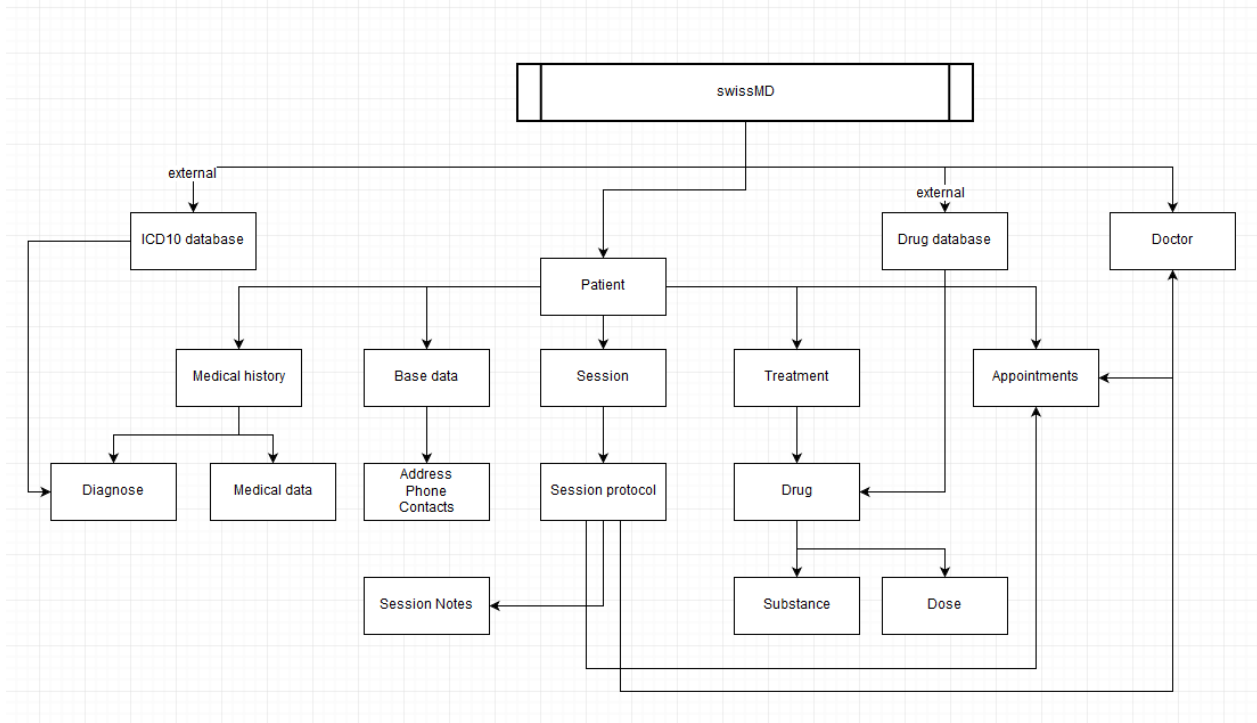


Abbildung 3 Komponentenmodell schematisch

## 4.2 Schnittstellen innerhalb des Application Server

Wie unter 2.3 beschrieben findet innerhalb des Applikations-Servers die Logik der Applikation statt. Die Logik steuert dabei die Kommunikation zwischen dem End-User und der Datenbank. Auf die Applikation bezogen bedeutet dies, dass die aus der Datenbank extrahierten Daten zur Anwendung kommen. Daraus resultieren Schnittstellen und Interaktionen innerhalb des Application Servers.

Konkret sind zum jetzigen Zeitpunkt folgende Interaktionen denkbar und vorgesehen:

- **Behandlungen**
  - Verschiedene Krankheiten und Medikamente können verschiedenen Patienten zugeordnet werden.
  - Der Psychiater kann Behandlungen ausführen und dokumentieren sowie Folgetermine erfassen
  - Abhängigkeiten:
    - Patientenstamm
    - Terminverwaltung
    - Krankheiten (nach ICD10)
    - Therapien/Therapieschemata
    - Suche
    - Medikamente
- **Suche**
  - Der Psychiater sucht anhand eines Textmusters nach Patienten, Krankheiten, Therapien, Therapieschema oder Medikamenten.
  - Abhängigkeiten:
    - Patientenstamm
    - Behandlungen/Sitzungen
    - Krankheiten (nach ICD10)
    - Therapien
    - Therapieschemata
    - Medikamente
- **Medikamente**
  - Die Verabreichung und die Dosierung der Medikamente wird durch den behandelnden Psychiater verwaltet
  - Abhängigkeiten:
    - Medikamente
    - Patientenstamm
- **Gefährdung**
  - Für sich selbst oder die Gesellschaft gefährliche Patienten werden gemäss den Vorgaben über eine FU durch den Psychiater in eine Klinik eingewiesen
  - Abhängigkeiten:
    - Patientenstamm
    - Kliniken
- **Sitzungen**
  - Sitzungsprotokoll
    - Datum
    - Name des behandelnden Arztes
    - Sitzungsnotizen

### 4.3 Datenflussmodell

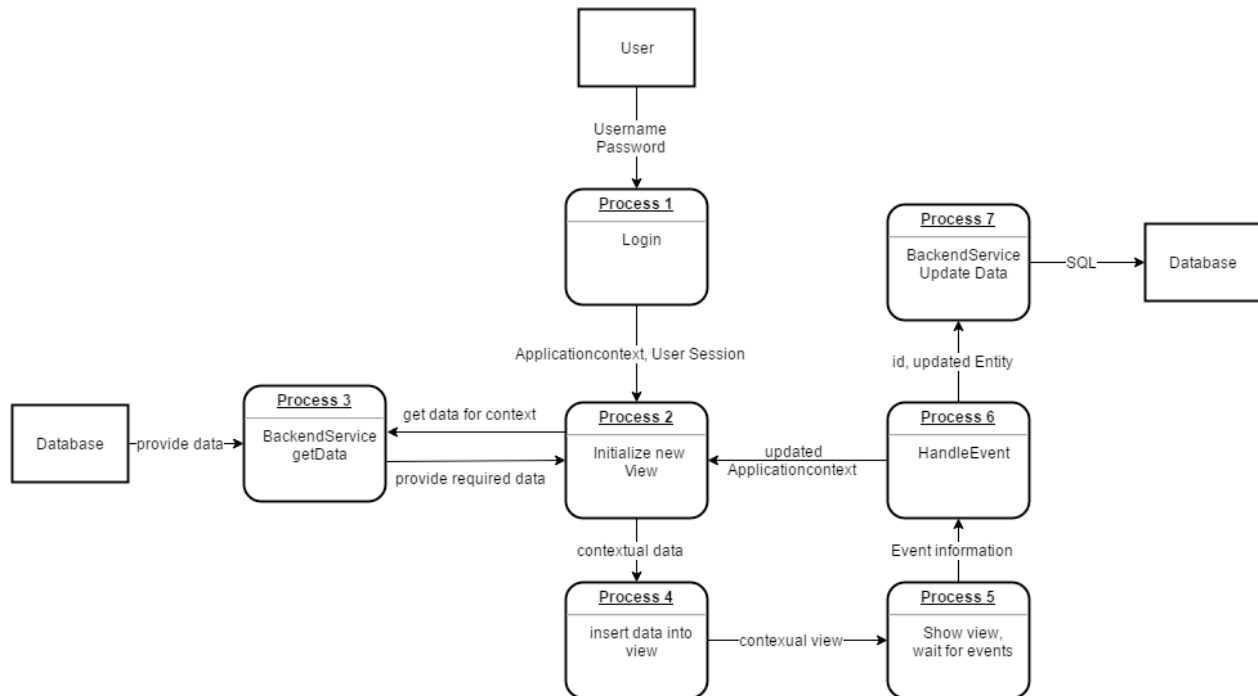


Abbildung 4 Datenflussmodell innerhalb der Applikation

## 5 Weiterentwicklung des Systems

Die Applikation "swissMD" soll modulbasiert entwickelt werden. Dies ermöglicht eine einfache Weiterentwicklung der Applikation zu einem späteren Zeitpunkt. Für die gesamte Applikation streben wir als Entwicklungsteam eine lange Lebensdauer an. Die Qualität der Software soll mit kleinen Iterationen während künftiger Softwareupdates, und mit einer qualitativ hochstehenden und umfassenden Dokumentation, hoch gehalten werden. Die Wartung und Behebung von Bugs wird über die gesamte Lebensdauer der Applikation angeboten und ist sichergestellt.

Um zukünftige Anforderungen der Endbenutzer mit möglichst geringem Aufwand umzusetzen, wird die Softwarearchitektur flexibel geplant. Die Benutzeroberfläche soll, genauso wie die zugrunde liegenden Datenstrukturen, erweiterbar sein um auf zukünftige Anforderungen rasch reagieren zu können. Besonderes Augenmerk muss hierbei auf die Datenbank gelegt werden. Die Datenbank der Applikation soll von Beginn weg diese Anforderungen erfüllen. Dazu werden die Datenschemen (Tabellenstruktur) spätestens zur Produktivsetzung der Applikation in der dritten Normalform vorliegen. Dadurch sind vermeidbare Redundanzen eliminiert und die Datenbank kann auch zu einem späteren Zeitpunkt ohne grossen Aufwand angepasst werden.

Die folgenden Punkte könnten zu einem späteren Zeitpunkt, nach Veröffentlichung der Version 1.0 verbessert werden:

- Anbindung weiterer Schnittstellen der Kliniken und Praxen:  
Die Applikation würde dadurch noch individueller einsetzbar und der Funktionsumfang würde erweitert. Dem Endbenutzer ergeben sich dadurch noch vielfältigere Möglichkeiten zur Unterstützung der täglichen Arbeit.
- Performanceverbesserungen
- Implementierung der Internationalisierung um weitere Sprachen für die Benutzeroberfläche anzubieten

## 6 Testfälle

Für das Testing setzen wir auf einen Mix aus White-Box-Tests und Black-Box-Tests. Mit diesen beiden Testverfahren bietet sich uns die Möglichkeit, die Qualität der Applikation sowohl während der Entwicklungsphase wie auch am Schluss, vor der Übergabe an den Kunden, sicher zu stellen. Die Methoden des Black-Box-Testes alleine sind nicht ausreichend, da die Spezifikationen eine höhere Abstraktion besitzen als die eigentliche Implementierung.

### 6.1 White-Box-Test

Die Testfälle im White-Box-Test gehen von der Struktur des Testobjektes aus. Die Testfälle werden von den Software-Entwicklern beschrieben und die Testüberdeckung wird an Hand des Codes bewertet. Das Ziel dieser Tests ist es, dass alle Anweisungen, Bedingungen und Pfade mindesten einmal ausgeführt werden. Mit Hilfe dieser Tests wird die Codequalität sichergestellt. Die Methoden eines White-Box-Testes teilen sich auf in die folgenden Verfahren:

- Kontrollflussorientiertes Verfahren
- Datenflussorientiertes Verfahren

Die Applikation „swissMD“ wird mit Hilfe des Kontrollflussorientierten Verfahrens geprüft. Diese Tests werden mit Hilfe von sogenannten Kontrollflussgraphen beschrieben. Ein Kontrollflussgraph ist ein gerichteter Graph mit genau einem Start- und Endknoten. Rechteckige Knoten stellen hierbei Anweisungen dar und rautenförmige Knoten repräsentieren Bedingungen. Folgendes Beispiel zeigt auf wie ein Code anhand eines Kontrollflussgraphen beschrieben wird:

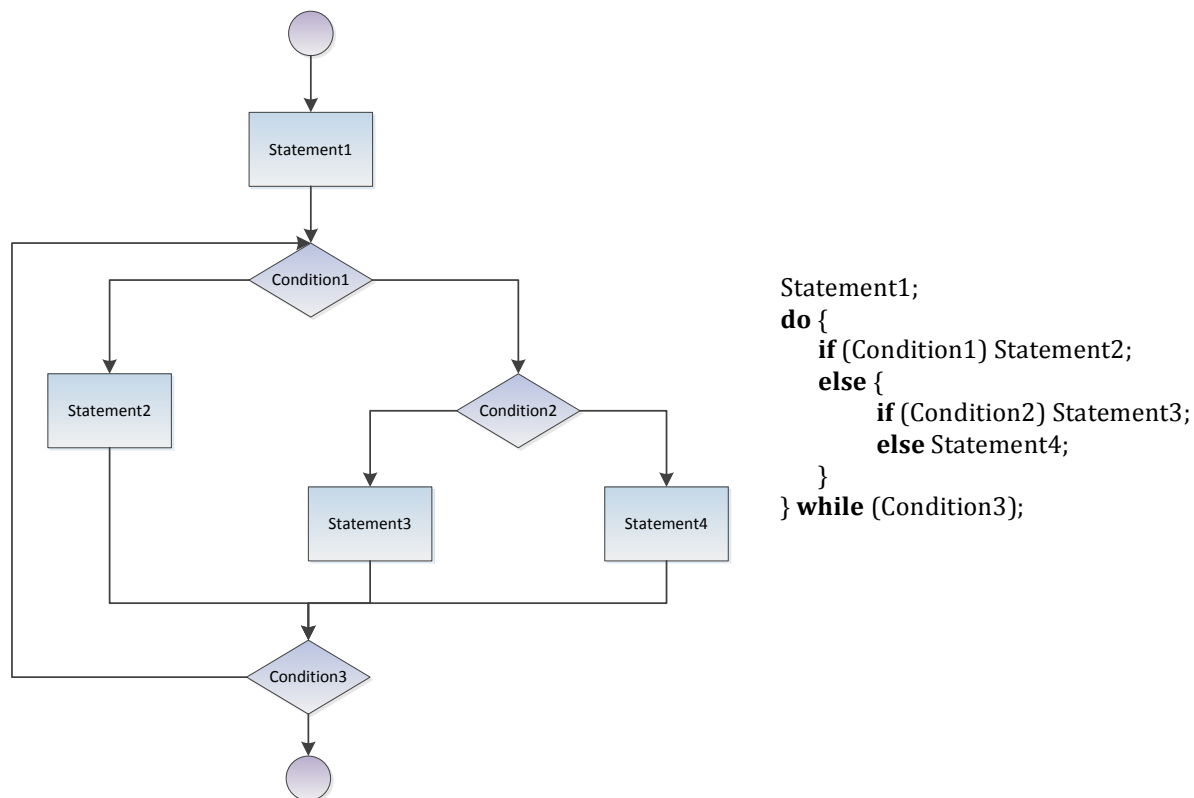


Abbildung 5 Beispiel eines Kontrollflussgraphen

**Vorteile:**

- Die Codequalität wird auf einem hohen Level gehalten
- Es ist sichergestellt dass auf dem Github-Verzeichnis nur ausführbarer Code vorhanden ist

**Nachteile:**

- Mehraufwand

## **6.2 Black-Box-Test**

Im Gegensatz zu den oben beschriebenen White-Box-Tests gehen die Testfälle in einem Black-Box-Test von den Spezifikationen aus. Die Interna der Applikation sind dabei nicht bekannt. Dazu wird die Applikation aus Sicht des Anwenders angeschaut und geprüft. Zum Testen besteht dabei kein Zugriff auf den Code oder die Datenbank.

Damit die erstellte Applikation sämtlichen definierten Anforderungen entspricht, werden die Anwendungsfälle (Use Cases, siehe 1.4) von einer oder mehreren Testpersonen durchgespielt. Die Testperson ist dabei von Vorteil eine am Projekt unbeteiligte Person. Diese Personen kennen das Produkt und insbesondere seine Feinheiten noch nicht. Die Testpersonen führen ein Testprotokoll zur Nachvollziehbarkeit und für die anschließende Fehlerbehebung.

Für die Black-Box-Tests werden wir folgende Fehlerklassen zur Verfügung stellen:

- Fehlende oder inkorrekte Funktionen
- Schnittstellenfehler
- Fehler in den Datenstrukturen
- Performancefehler oder –Beeinträchtigungen
- Fehler bei der Initialisierung und Terminierung von Abläufen in der Applikation

**Vorteile:**

- Feedback von unabhängigen Testpersonen

**Nachteile:**

- Betreuungsaufwand für unabhängige Testpersonen benötigt Zeit
- Das Auffinden von festgestellten Fehlfunktionen kann sich unter Umständen schwierig gestalten

## Abbildungsverzeichnis

Abbildung 1 Use-Case-Diagramm Übersicht.....	3
Abbildung 2 Überblick System-Architektur.....	6
Abbildung 3 Komponentenmodell schematisch .....	11
Abbildung 4 Datenflussmodell innerhalb der Applikation.....	13
Abbildung 5 Beispiel eines Kontrollflussgraphen.....	15

## Tabellenverzeichnis

Tabelle 1 Funktionelle Benutzeranforderungen .....	1
Tabelle 2 Use Case Form 1 Patient Beistand zuweisen.....	4
Tabelle 3 Use Case Form 2 Medikation anpassen .....	5
Tabelle 4 Funktionelle Systemanforderungen .....	9

## Referenzen

Bundesversammlung der Schweizerischen Eidgenossenschaft. 2014. *Bundesgesetz über Arzneimittel und Medizinprodukte (Heilmittelgesetz, HMG)*. SR 812.21

Bundesversammlung der Schweizerischen Eidgenossenschaft. 2014. *Bundesgesetz über den Datenschutz (DSG)*. SR 235.1

Bundesversammlung der Schweizerischen Eidgenossenschaft. 2016. *Bundesgesetz über die Krankenversicherung (KVG)*. SR 832.10

Der Schweizerische Bundesrat. 2016. *Verordnung über die Arzneimittel (Arzneimittelverordnung, VAM)*. SR 812.212.21

Der Schweizerische Bundesrat. 2016. *Verordnung über die Bewilligung im Arzneimittelbereich (Arzneimittel-Bewilligungsverordnung, AMBV)*. SR 812.212.1

Der Schweizerische Bundesrat. 2015. *Verordnung über die Invalidenversicherung (IVV)*. SR 831.201

ISO/IEC/IEEE. 2011. *Systems and software engineering - Requirements engineering*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission/ Institute of Electrical and Electronics Engineers (IEEE), (IEC), ISO/IEC/IEEE 29148.