

# Medikamententrackingsystem für die Flugrettung

**Studien-/Bachelorarbeit HS 2015**

Version: 1.1

**Autor:** **Germann Fabienne**  
**Muggli Sandro**  
**Laib Fabio**  
fgermann@hsr.ch  
E-Mail: smuggling@hsr.ch  
flaib@hsr.ch  
Matrikelnummer: 12153938  
12154365  
12154472

**Dozent:** Prof. Dr. Christian  
Bermes  
Prof. Dr. Farhad D.  
Mehta  
**Themengebiet:** Automation und  
Robotik  
Softwareentwick-  
lung  
**Studiengang:** Maschinen-  
bau/Innovation  
Informatik

**Erstellt am: 22. September 2015**

**Letzte Änderung am: 16. Dezember 2015**

## Abstract

### Ausgangslage

Die REGA Flugrettung Schweiz führt in Ihren Einsatzhelikoptern einen Standardsatz Notfallmedikamente mit. Einige dieser Medikamente sind sehr teuer und haben eine verhältnismässig kurze Haltbarkeitsdauer. Gleichzeitig müssen diese speziellen Medikamente immer mitgeführt werden, kommen aber nur relativ selten zum Einsatz. Ein Ablauf der Haltbarkeit dieser Medikamente vor ihrem Einsatz bei der REGA ist deshalb wahrscheinlich. Bis heute wird das Ablaufdatum der Medikamente mittels manuell geführten Listen kontrolliert. Dies ist sehr zeitaufwendig. Mit diesen Listen wird nur kontrolliert, ob das Medikament noch haltbar ist oder nicht. Abgelaufene Medikamenten müssen entsorgt werden, was einen Verlust für die REGA bedeutet.

### Ziel

Die REGA möchte die Medikamente in Zukunft zuverlässig markieren, scannen, nachverfolgen und vor Ablauf der Haltbarkeitsdauer an den Hersteller retournieren. Dieser Vergütet zum einen die Medikamente der REGA zurück und zum anderen gibt er die ausrangierten Medikamente an grosse Apotheken und Spitäler weiter, welche diese Medikamente täglich brauchen. Somit kann die REGA Verschwendung vermeiden und Kosten sparen. Deshalb ist das Ziel dieser Arbeit, ein Medikamententrackingsystem im Prototypstatus zu erstellen, welches die Medikamente überwacht und die Benutzer vor dem Ablauf informiert, sodass die Medikamente rechtzeitig ausgetauscht werden können. Da es sich um Programmier- und Konstruktionsarbeiten handelt, wird die Arbeit interdisziplinär durchgeführt. Es sind die Studiengänge Maschinenbau/Innovation und Informatik involviert.

### Methode / Vorgehen

Da die ganze Arbeit interdisziplinär ist, wurde nach dem V-Modell vorgegangen. Das ganze Projekt wurde deshalb in folgende Phasen unterteilt:

- **Analyse** → Aufgabenstellung Klären, Morphologischer Kasten erstellen, Varianten ausarbeiten und entscheiden, Festhalten der Softwareanforderungen, Erstellung Use Cases und Domain Model.
- **Design** → Programmstruktur designen, Konstruktion ausarbeiten, Ausarbeitung der Applikationsarchitektur, Erstellung ERM und Technologieevaluation.
- **Realisierung** → Prototyp fertigen/montieren, Programmierarbeiten durchführen
- **Integrationstest** → Applikationsprototyp verknüpfen und testen, Bugfixing.
- **Abnahmetest** → Abnahme Gesamtprototyp durch REGA.

### Wesentliche Ergebnisse

Das Ergebnis dieser gemeinsamen Arbeit ist eine IOS-Applikation und ein stabiler Unterbau für das iPad und den Scanner. Die Applikation empfängt die Barcodes von dem Bluetooth Scanner und erkennt dadurch das Medikament. Medikamente können so eingelagert und den verschiedenen Rettungsmodulen zugewiesen werden. Vordefinierte Benutzer werden vor Ablauf einer Frist per Email informiert, dadurch ist ein frühzeitiges Erkennen von ablaufenden Medikamenten möglich. Das System ist so aufgebaut, dass es ausbaufähig ist und auf die Bedürfnisse der REGA angepasst werden kann. So könnten in Zukunft der Lagerbestand überprüft, Bestellungen der Medikamente automatisch ausgeführt und bei Opiaten die Patientendaten hinterlegt werden.



Abbildung 1: Medikamententrackingsystem

## Änderungsnachweis

Version	Änderungsgrund	Kürzel	Datum
1.0	Erstellung des Dokumentes	STUD	17.09.15
1.1	Korrekturen	STUD	14.12.15

## Inhalt

<b>Abstract</b> .....	<b>2</b>
<b>1 Verzeichnisse</b> .....	<b>5</b>
1.1 Abbildungen .....	5
1.2 Tabellen.....	6
1.3 Begriffe und Definitionen .....	7
<b>2 Einleitung</b> .....	<b>8</b>
2.1 Einführung ins Thema.....	8
2.2 Systemgrenze.....	9
2.3 Auszüge aus dem Pflichtenheft.....	10
<b>3 Analyse</b> .....	<b>13</b>
3.1 Analyse der Aufgabe.....	13
3.1.1 Stand der Technik Codes und Scanner .....	14
3.1.2 Marktübersicht bestehender Lösungen .....	18
3.1.3 Patentrecherche.....	21
3.2 Analyse der Softwareanforderungen .....	21
3.2.1 Use Cases.....	21
3.2.2 Nicht funktionale Anforderungen .....	26
3.2.3 Domain Model.....	27
3.2.4 Zustandsdiagramm .....	28
3.2.5 Wireframes.....	29
3.3 Analyse der Konstruktionsanforderungen .....	31
3.3.1 Funktionsstruktur .....	32
3.3.2 Lösungen der Teilfunktionen .....	33
3.3.3 Morphologischer Kasten und Vorentscheidung .....	47
3.4 Definitiver Variantenentscheid mit der REGA .....	58
3.5 Fazit Analyse.....	61
<b>4 Design</b> .....	<b>62</b>
4.1 Software .....	62
4.1.1 Application Architecture .....	62
4.1.2 Technologieevaluierung.....	64
4.1.3 Verwendete Technologien .....	66
4.1.4 Entity-Relationship-Modell .....	67
4.2 Unterbau.....	68
4.2.1 Basismodell.....	68
4.2.2 Berechnungen.....	68

4.2.3	Konstruktion .....	80
4.2.4	Herstellung und Kosten .....	83
4.2.5	Ergebnis/ Fazit .....	85
4.3	Fazit Design .....	85
<b>5</b>	<b>Realisierung.....</b>	<b>86</b>
5.1	Software .....	86
5.1.1	Generell .....	86
5.1.2	Server .....	86
5.1.3	Client .....	89
5.2	Unterbau .....	94
5.2.1	Montage .....	94
5.3	FMEA .....	97
5.4	Installations- und Bedienungsanleitung .....	99
5.5	Fazit Realisierung .....	99
<b>6</b>	<b>Integrationstest .....</b>	<b>100</b>
6.1	Erster Integrationstest .....	100
6.2	Defect Log Integrationstest .....	103
6.3	Zweiter Integrationstest .....	103
6.4	Fazit .....	104
<b>7</b>	<b>Abnahmetest .....</b>	<b>105</b>
7.1	Defect Log Abnahmetest .....	106
7.2	Fazit .....	106
<b>8</b>	<b>Projektplanung .....</b>	<b>107</b>
8.1	Das V-Modell .....	107
8.2	Gantt-Diagramm .....	107
8.3	Projektphasen .....	108
8.4	Meilensteine .....	108
8.4.1	Offizielle Meilensteine .....	109
8.4.2	Projektinterne Meilensteine .....	109
8.5	Meetings .....	110
<b>9</b>	<b>Zukunftsausblick und Optimierungen .....</b>	<b>111</b>
9.1	Software .....	111
9.2	Unterbau .....	112
9.3	Fazit Zukunftsausblicke und Optimierungen .....	113
<b>10</b>	<b>Schlussfolgerungen, Ausblicke und Empfehlung .....</b>	<b>114</b>
10.1	Zusammenfassung .....	114
<b>11</b>	<b>Literatur und Quellenverzeichnis .....</b>	<b>115</b>
11.1	Literaturverzeichnis .....	115
<b>12</b>	<b>Erklärung zur Urheberschaft .....</b>	<b>116</b>
<b>Anhang .....</b>		<b>117</b>
<b>Autoren .....</b>		<b>118</b>

# 1 Verzeichnisse

## 1.1 Abbildungen

Abbildung 1: Medikamententrackingsystem.....	2
Abbildung 2: Systemgrenze.....	9
Abbildung 3: Organisation .....	12
Abbildung 4: Barcode.....	14
Abbildung 5: OCR Code .....	14
Abbildung 6: DataMatrix .....	14
Abbildung 7: Lesetechnologie Barcode.....	15
Abbildung 8: Lesestift.....	15
Abbildung 9: Lese pistole.....	16
Abbildung 10: Schlitzleser .....	16
Abbildung 11: Strichcodekamera.....	16
Abbildung 12: Barcodeleser Spital .....	19
Abbildung 13: Funktionen ScanModul.....	20
Abbildung 14: Use Case Diagramm .....	22
Abbildung 15: Domain Model.....	27
Abbildung 16: Zustandsdiagramm.....	28
Abbildung 17: Wireframe Startscreen .....	29
Abbildung 18: Wireframe Login .....	29
Abbildung 19: Wireframe Vorauswahl .....	29
Abbildung 20: Wireframe Lager auffüllen .....	30
Abbildung 21: Wireframe Rettungsmodul auffüllen.....	30
Abbildung 22: Wireframe System verwalten .....	30
Abbildung 23: Wireframe Medikamente verwalten .....	31
Abbildung 24: Wireframe Opiat .....	31
Abbildung 25: Funktionsstruktur .....	32
Abbildung 26: Blechkonstruktion .....	34
Abbildung 27: Profilkonstruktion .....	35
Abbildung 28: Profilrahmen an der Wand .....	35
Abbildung 29: Box.....	35
Abbildung 30: Morphologischer Kasten Unterbau .....	48
Abbildung 31: Kartonmodell Blechkonstruktion.....	48
Abbildung 32: Kartonmodell Profilrahmen.....	49
Abbildung 33: Kartonmodell Box .....	50
Abbildung 34: Morphologischer Kasten System .....	53
Abbildung 35: Morphologischer Kasten mit definitiver Variante, Unterbau .....	58
Abbildung 36: Morphologischer Kasten mit definitiver Variante, System.....	59
Abbildung 37: Application Architecture.....	63
Abbildung 38: Entity-Relationship-Modell.....	67
Abbildung 39: Basismodell .....	68
Abbildung 40: Koordinatensystem.....	69
Abbildung 41: Berechnung Fall 1 .....	71

Abbildung 42: Berechnungen Fall 2 .....	72
Abbildung 43: Berechnungen Fall 3 .....	73
Abbildung 44: Berechnungen Sicherheit bezogen auf die Grundfläche .....	74
Abbildung 45: Berechnung Sicherheit bezogen auf die Kraft .....	76
Abbildung 46: Berechnung max. Kippkraft nach vorne .....	79
Abbildung 47: Prototyp Unterbau .....	80
Abbildung 48: Modularität Medikamententrackingsystem .....	82
Abbildung 49: Klassendiagramm Data Access Objects.....	87
Abbildung 50: Starten des Mailjobs .....	88
Abbildung 51: www-Folder.....	89
Abbildung 52: Auszug aus app.js .....	90
Abbildung 53: Auszug aus app.js .....	90
Abbildung 54: Auszug aus controller.js .....	90
Abbildung 55: Drug Factory .....	91
Abbildung 56: Auszug aus index.html .....	91
Abbildung 57: Dependency Injection .....	92
Abbildung 58: Steckdose vor der Bearbeitung.....	94
Abbildung 59: Steckdose nach der Bearbeitung .....	94
Abbildung 60: Öffnung für das Kabel .....	95
Abbildung 61: Kabelentlastung / Reduktion .....	95
Abbildung 62: Befestigung Steckdose.....	95
Abbildung 63: Befestigung Moosgummi.....	96
Abbildung 64: Befestigung Moosgummi Laschen.....	96
Abbildung 65: Befestigung Moosgummi Boden .....	97
Abbildung 66: Bedeutung Risiko- Prioritätszahl .....	98
Abbildung 67: V-Modell.....	107
Abbildung 68: Gantt-Diagramm .....	108
Abbildung 69: Prototyp Befestigung Scanner .....	112
Abbildung 70: Optimierung Befestigung Scanner .....	112
Abbildung 71: Prototyp Befestigung Ladekabel .....	112
Abbildung 72: Optimierung Befestigung Ladekabel .....	112

## 1.2 Tabellen

Tabelle 1: Allgemeine, kommerzielle und wirtschaftliche Musskriterien.....	10
Tabelle 2: Technische Optimierungskriterien.....	11
Tabelle 3: Allgemeine, kommerzielle und wirtschaftliche Optimierungskriterien.....	11
Tabelle 4: Nice to have .....	11
Tabelle 5: Scancodes .....	14
Tabelle 6: Mobile Lesegeräte .....	16
Tabelle 7: Stationäre Lesegeräte .....	16
Tabelle 8: Übersicht Patentrecherche .....	21
Tabelle 9: Swot- Analyse Variante rot, Unterbau .....	49
Tabelle 10: Swot- Analyse Variante blau, Unterbau .....	50
Tabelle 11: Swot- Analyse Variante grün, Unterbau .....	51
Tabelle 12: Kriterien und Gewichtung Entscheidung Unterbau .....	51

Tabelle 13: Entscheidung Unterbau .....	52
Tabelle 14: Swot- Analyse Variante Rot, System .....	54
Tabelle 15: Swot- Analyse Variante Blau, System.....	55
Tabelle 16: Swot- Analyse Variante grün, System.....	55
Tabelle 17: Kriterien und Gewichtung Entscheidung System.....	56
Tabelle 18: Entscheidung System .....	57
Tabelle 19: Swot- Analyse Endvariante, System .....	61
Tabelle 20: Entscheidungskriterien .....	64
Tabelle 21: Auswertung Varianten .....	65
Tabelle 22: Standsicherheit .....	70
Tabelle 23: Allgemeine Berechnungen .....	70
Tabelle 24: Berechnung Fall 1 .....	72
Tabelle 25: Berechnungen Fall 2.....	73
Tabelle 26: Berechnungen Fall 3.....	74
Tabelle 27: Berechnung Sicherheit bezogen auf die Grundfläche .....	75
Tabelle 28: Berechnungen Sicherheit bezogen auf die Kraft .....	76
Tabelle 29: FM Max für die neue Grundfläche.....	78
Tabelle 30: Kostenübersicht aller Komponenten .....	83
Tabelle 31: Kosten Variante 1 .....	84
Tabelle 32: Kosten Variante 2 .....	84
Tabelle 33: Kosten Variante 3 .....	84
Tabelle 34: Auszüge aus der FMEA.....	98
Tabelle 35: Ergebnis Test Use Cases.....	101
Tabelle 36: Ergebnis Test Pflichtenheft (Muss- Kriterien) .....	102
Tabelle 37: Ergebnis Test Pflichtenheft (Optimierungs- Kriterien) .....	102
Tabelle 38: Defect Log Integrationstest.....	103
Tabelle 39: Ergebnis Use Cases Test 2.....	104
Tabelle 40: Ergebnis Abnahmetest .....	105
Tabelle 41: Defect Log Integrationstest.....	106

### 1.3 Begriffe und Definitionen

**Rettungsmodul**      Damit sind alle möglichen Medikamentenkoffer aus dem REGA Etat gemeint (z.B. Grundmodul, Airwaymodul, Heliset, etc).

---

## 2 Einleitung

---

### 2.1 Einführung ins Thema

---

- REGA allgemein** Die REGA (schweizerische Rettungsflugwache) besitzt in der gesamten Schweiz 12 Basen. Das Logistikzentrum befindet sich am Flughafen Kloten.  
Alle Basen sind gleich aufgebaut und somit ist auch die Ausrüstung der Helikopter überall gleich. Dies bedeutet, dass alle Rettungsmodule den gleichen Stock an Medikamenten aufweisen. Zurzeit werden die Medikamente vom Logistikzentrum eingekauft, zwischengelagert und nach Bedarf an die einzelnen Basen weiter verteilt.
- Ausgangslage/  
Problem** In den Einsatzhelikoptern führt die REGA einen Standardsatz Notfallmedikamente mit. Einige Medikamente sind sehr teuer und haben eine verhältnismässig kurze Haltbarkeitsdauer. Jedoch müssen diese speziellen Medikamente immer mitgeführt werden, obwohl sie selten bis nie gebraucht werden. Ein Ablauf der Haltbarkeit dieser Medikamente vor ihrem Einsatz bei der REGA ist deshalb wahrscheinlich, da die Haltbarkeit schon bei der Zwischenlagerung im Logistikzentrum stark reduziert wird.  
Abgelaufene Medikamente müssen manuell aussortiert und entsorgt werden. Dies bedeutet enorme Verluste für die REGA.
- Lösung in der Vergangenheit** Bis jetzt wurde das Ablaufdatum der Medikamente mühsam mittels manuell geführten Listen kontrolliert. Diese waren sehr zeitaufwendig. Mit diesen Listen konnte nur die Haltbarkeit überprüft werden, nicht auch noch der Bestand, etc.
- Erforderliche Massnahmen** Die REGA möchte die Medikamente in Zukunft zuverlässig markieren, scannen, nachverfolgen und vor Ablauf der Haltbarkeitsdauer an den Hersteller retournieren. Dieser vergütet zum einen die Medikamente der REGA zurück und zum anderen gibt er die ausrangierten Medikamente an grosse Drogerien/ Spitäler weiter, welche diese Medikamente täglich brauchen. Somit kann die REGA Verschwendung vermeiden und Kosten sparen.
- Langfristige Strategie** Da die REGA langfristig plant, möchte sie in Zukunft nicht nur die Haltbarkeit der Medikamente überprüfen, sondern es sollen auch die Bestände überprüft und automatisch nachbestellt werden. Zudem soll es möglich sein, nicht nur Medikamente zu überprüfen, sondern alle medizinischen Geräte mit Ablaufdatum oder Wartungsdatum. Bei Opiaten wäre es von Nutzen, wenn die Kontaktdaten des Empfängers direkt in das System eingetragen werden könnten.
- Aufgabe** Ziel der Arbeit ist der Aufbau und erfolgreiche Betrieb eines Funktionsmusters für ein System, das Medikamente mit Ihren Verfallsdaten markiert, scannt, verfolgt und bei Bedarf aussortiert.



## 2.2 Systemgrenze

**Systemgrenze** Die Systemgrenze dieses Projektes sieht wie folgt aus:

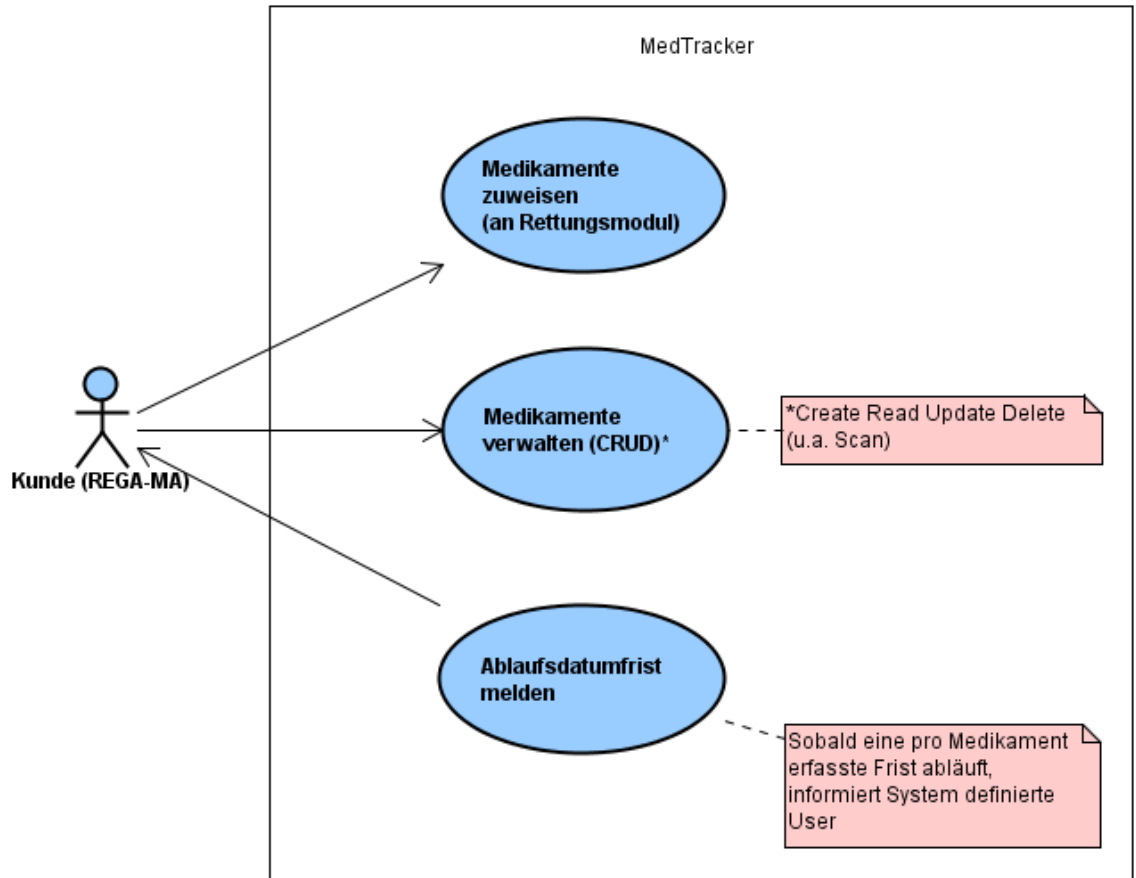


Abbildung 2: Systemgrenze

Der Benutzer scannt die Medikamente ein und weist das Medikament falls notwendig einem Rettungsmodul zu.

Das System gibt dem Benutzer vor dem Ablauf der Haltbarkeit eine Meldung aus.

## 2.3 Auszüge aus dem Pflichtenheft

**Einleitung** Das Pflichtenheft dient als Basis des Projektes. Da für das Projekt vor allem die Anforderungsliste entscheidend ist, ist diese unten angefügt.  
*Ganzes Pflichtenheft im Anhang ersichtlich.*

**Musskriterien** **Technische Kriterien (Angaben zum Produkt)**

Nr.	Kriterium
TM1	System scannt Medikamente (QR-, Barcode, Datamatrix)* ein
TM2	System gibt Meldung, wenn Medikament abläuft
TM3	System erkennt, welches Medikament in welchem Koffer (Rettungsmodul) ist
TM4	System scannt Haltbarkeit und Medikamentenname* ein
TM5	Kompatibel für alle Basen
TM6	Effiziente Bedienung durch REGA-Mitarbeiter (kein Mehraufwand zur aktuellen Lösung)

Tabelle 1: Technische Musskriterien

\*welche Informationen aus den Codes auslesbar sind und welche Art von Codes verwendet wird, ist zum Zeitpunkt der Erstellung dieses Dokuments noch in Abklärung.

**Allgemeine, kommerzielle und wirtschaftliche Kriterien**

Nr.	Kriterium
KM1	Bedienung muss durch REGA Personal auszuführen sein
KM2	Abgabe Arbeit Freitag 18. Dezember 2015, 16.00
KM3	Abgabe Kurzfassung Freitag 18. Dezember 2015, 16.00
KM4	Präsentation bei der REGA am 18. Januar 2016

Tabelle 1: Allgemeine, kommerzielle und wirtschaftliche Musskriterien

**Optimierungs-  
kriterien**

**Technische Kriterien (Angaben zum Produkt)**

Nr.	Kriterium
TO1	Einfache Bedienoberfläche (ohne Schulung)
TO2	System zeigt die Bestände der Medikamente an
TO3	System führt die Bestellung aus
TO4	Einfache Inbetriebnahme und Betrieb (durch Mitarbeiter möglich)
TO5	Einfache Fertigung (Standardverfahren bevorzugen)
T06	System läuft auf einem iPad
T07	Hygiene muss gewährleistet sein (Desinfektionsmittel geeignet)
T08	Kleines, kompaktes System (max. 1m x 1m x 1m)
T09	System läuft zuverlässig (90%)

**Tabelle 2: Technische Optimierungskriterien**

**Allgemeine, kommerzielle und wirtschaftliche Kriterien**

Nr.	Kriterium
KO1	Kostengünstig (innerhalb Kostendach)
KO2	Standardkomponenten verwenden

**Tabelle 3: Allgemeine, kommerzielle und wirtschaftliche Optimierungskriterien**

**Nice to have**

Nr.	Kriterium
N1	Bei Opiaten können die Kontaktdaten des Verbrauchers eingegeben werden
N2	Zugriff auf alle Basen (Informationen über alle Basen ersichtlich)
N3	Auch für andere medizinische Geräte geeignet

**Tabelle 4: Nice to have**

## Organisation

Da diese Arbeit interdisziplinär ist, hat auch die Projektorganisation Einfluss auf den Erfolg der Arbeit. Die Organisation sieht wie folgt aus:

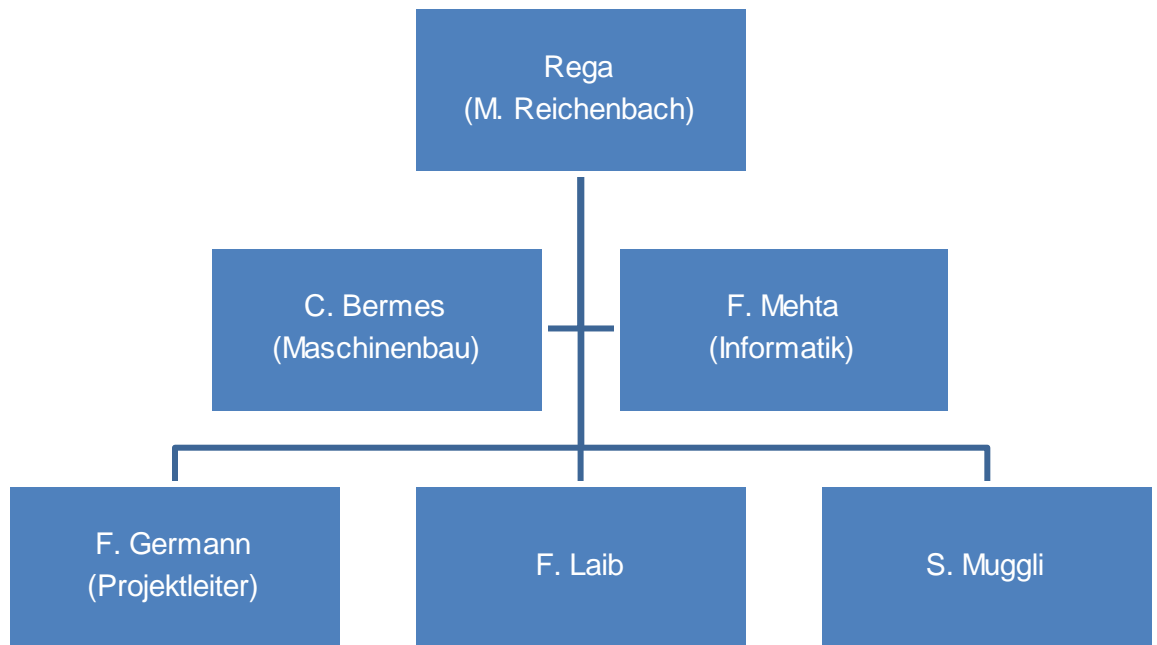


Abbildung 3: Organisation

**Auftraggeber:** REGA, Markus Reichenbach, Basisleiter Mollis, Ressortleiter Helikopter Verfahren HCM Standards

**Betreuer Maschinenbau:** Christian Bermes

**Betreuer Informatik:** Farhad Mehta

**Studentin Maschinenbau:** Fabienne Germann, Projektleiterin

**Studenten Informatik:** Fabio Laib, Sandro Muggli

**Projektplanung:** Die gesamte Planung und das Arbeitsjournal des Projektes werden auf Redmine gemacht, das auf dem HSR internen Server läuft. Jedes Teammitglied kann dort neue Aufgaben hinzufügen, seine Stunden verbuchen und beendete Aufgaben schliessen. Jede zweite Woche werden im Team zusammen die Aufgaben der nächsten zwei Wochen definiert und auf Redmine eingetragen. Von jeder Sitzung gibt es ein Sitzungsprotokoll, das ebenfalls auf Redmine bei der betreffenden Sitzung verlinkt ist. Da die Dozenten ebenfalls leserecht auf Redmine haben, können sie somit alle Sitzungsprotokolle einsehen und der Stand der Arbeit mitverfolgen.

**Datenmanagement:** Das Datenmanagement läuft über Switchdrive. Dadurch ist gewährleistet, dass jedes Mitglied immer und überall Zugriff auf alle Daten des Projektes hat.

## 3 Analyse

---

**Beschreibung** Im ersten Teil des Kapitels Analyse wird die Aufgabe genauer untersucht. Anhand der Antworten werden dann im zweiten Teil des Kapitels zum einen die Konstruktionsanforderungen und zum anderen die Softwareanforderungen definiert. Die Aufgabe wurde in einzelne Teilfunktionen unterteilt. Die verschiedenen Lösungsansätze für die einzelnen Teilfunktionen werden in diesem Kapitel genauer beschrieben. Durch diese ist der Morphologische Kasten mit seinen Varianten entstanden. Die einzelnen Varianten werden genauer beschrieben und bewertet. Zum Schluss ist dokumentiert, wie die REGA bei der Zwischenpräsentation auf die Lösungsvorsätze reagierte.

### 3.1 Analyse der Aufgabe

---

**Beschreibung** Zur Analyse gehört den Stand der Technik zu überprüfen. Welche Codes kommen für die Aufgabe in Frage? Wie werden diese Codes eingelesen? Was gibt es für Lesegeräte?

Da nicht nur die REGA mit Medikamenten arbeitet, wird in anderen Unternehmen nachgefragt, wie sie das Arbeiten mit den Medikamenten bewerkstelligen. Ziel ist es, gute Teillösungen in das neue System einfließen zu lassen.

Zum Schluss der Analyse wird noch eine Patentrecherche durchgeführt, damit danach das Endprodukt sicher nicht in Konflikt mit Patenten steht.

### 3.1.1 Stand der Technik Codes und Scanner

#### Einleitung

Hier wird nun untersucht, welche Codes geeignet sind, und wie die einzelnen Codes aufgebaut sind und gelesen werden. Für die Überwachung der Haltbarkeit sind vor allem der Name und das Haltbarkeitsdatum eines Medikamentes von Nöten. Diese sollen möglichst einfach in das System eingelesen werden können. Die Medikamentenverpackungen und Ampullen geben das zu verwendete Scansystem vor, denn diese müssen danach eingescannt werden können. Für die Medikamente der REGA kommen diese verschiedene Scansysteme in Frage:



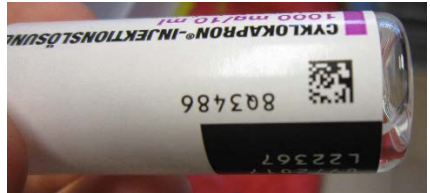
<p><b>Barcode:</b> Bei allen Medikamentenschachteln (Schachtel mit mehreren Ampullen) ist ein Barcode vorhanden. Jedoch ist dieser Barcode nicht auf allen Ampullen vorhanden.</p>	 <b>Abbildung 4: Barcode</b>
<p><b>OCR (optische Texterkennung):</b> Bei allen Verpackungen und bei allen Ampullen stehen das Datum und der Name des Medikamentes in Textform.</p>	 <b>Abbildung 5: OCR Code</b>
<p><b>DataMatrix:</b> Ab 2016/2017 sollen gemäss swissmedic alle Medikamente einheitlich gekennzeichnet werden, durch die sogenannte DataMatrix.</p>	 <b>Abbildung 6: DataMatrix</b>
<p><b>Neue Kennzeichnung:</b> Eine weitere Möglichkeit wäre, alle Medikamente neu auf der REGA Basis zu kennzeichnen und in einer Datenbank zu erfassen.</p>	

Tabelle 5: Scancodes

Auf die einzelnen Scansysteme wird nun etwas genauer eingegangen.

#### 3.1.1.1 Barcode/Strichcode

##### Beschreibung

Weil das Übertragungsmedium bei Barcodes Licht ist, gehören sie in die Gruppe der optischen Codierungen. Es wird unterschieden zwischen Zweibreitencodes und Mehrbreitencodes. Die Zeichen der Zweibreitencodes bestehen aus Elementen mit zwei verschiedenen Breiten und sind somit einfach herstellbar. Jedoch besitzen sie eine tiefe Informationsdichte. Bei den Mehrbreitencodes bestehen die Strichcode aus Elementen mit mehr als zwei Breiten. Sie sind aufwendiger herzustellen, haben aber eine höhere Informationsdichte. [1]

## Erfassung von Strichcodes

Für das Lesen von Barcodes kommen verschiedene Lesegeräte wie Lesestifte, Lesepistolen, Scanner und Kameras in Frage. Sie werden nach folgenden Kriterien unterschieden: [1]

- Mobiler oder stationärer Einsatz
- Abstand Lesegerät zum Barcode
- Winkel, unter denen die Barcodes fehlerfrei eingelesen werden müssen
- Immer gleiche Ausrichtung der Barcodes?

## Lesetechnologie

Zum Lesen streichen Lichtbündel über den Barcode. Der Durchmesser des Lichtbündels ist nicht dicker als ein schmales Barcodeelement.

Das Lesen erfolgt durch die Reflektion der Lichtstrahlen. Das zurückgeworfene Licht der schwarzen Streifen ist dabei weniger intensiv als das von einer weissen Lücke. Diese werden an ein Fotoelement gesendet, welche je nach Stärke des reflektierten Lichtbündels mehr oder weniger Spannung erzeugt. Diese Spannung wird in einem Verstärker noch verstärkt. Um an die Daten zu kommen, müssen die Signale vom Verstärker noch von einem Decoderbaustein entschlüsselt werden. [1]

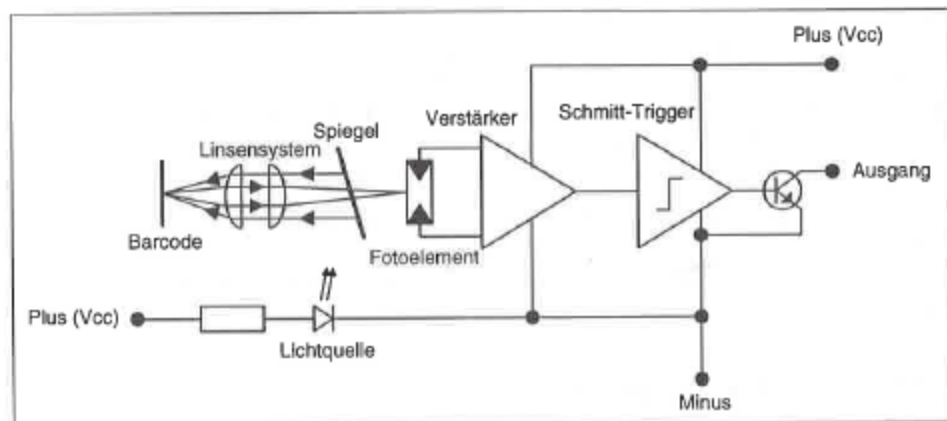


Abbildung 7: Lesetechnologie Barcode

## Lesegeräte

Es wird zwischen mobilen oder stationären Lesegeräten unterschieden.

### Mobile Lesegeräte:

#### Lesestifte:

- Geringer Abstand zum Barcode
- Konstante Geschwindigkeit nötig → sonst nicht lesbar
- Kein Absetzen des Stiftes möglich → nicht lesbar
- Farbige Barcodes lesbar
- Lesestifte mit verschiedenen Spitzen möglich [1]

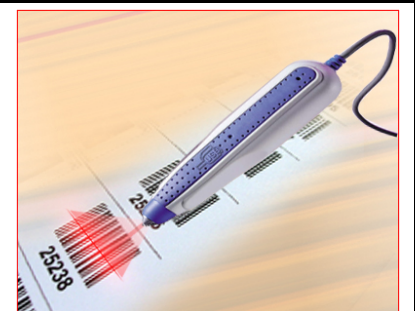


Abbildung 8: Lesestift


<p><i>Lesepistolen:</i></p> <ul style="list-style-type: none"> <li>• Barcodes aus grösserer Entfernung lesbar</li> <li>• Berührungsloses Abtasten des Barcodefeldes → keine Abnutzungserscheinungen [1]</li> </ul>	 <p><b>Abbildung 9: Lesepistole</b></p>
--	--

Tabelle 6: Mobile Lesegeräte

**Stationäre Lesegeräte:**


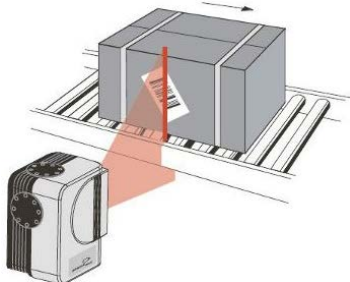
<p><i>Schlitzleser:</i></p> <ul style="list-style-type: none"> <li>• Barcode wird manuell durch den Schlitz gezogen</li> <li>• Abstand zwischen Barcode und Lesegerät immer gleich</li> <li>• Eher selten im Einsatz [1]</li> </ul>	 <p><b>Abbildung 10: Schlitzleser</b></p>
<p><i>Laserscanner:</i></p> <ul style="list-style-type: none"> <li>• Ausnahmslos für die vollautomatische Erfassung</li> <li>• Hervorragend zur Erfassung von Codes mit hoher Dichte (High-Density-Codes) [1]</li> </ul>	
<p><i>Strichcodekamera:</i></p> <ul style="list-style-type: none"> <li>• Kann Strichcode unabhängig dessen Ausrichtung erfassen</li> <li>• Unabhängige Objektform</li> <li>• Abstand zum Strichcode möglich [1]</li> </ul>	 <p><b>Abbildung 11: Strichcodekamera</b></p>

Tabelle 7: Stationäre Lesegeräte

**Probleme**

Bei der Verwendung von Barcodescannern für die REGA gibt es aber noch einige Probleme, die gelöst werden müssen.

Einerseits ist es nicht möglich, das Verfallsdatum zu hinterlegen, da dieses nicht konstant ist. Dies müsste man manuell über das System eingeben oder auswählen.

Zudem müsste man auf die Herstellerdatenbank Zugriff haben, damit man an die im Code hinterlegten Daten kommt, oder man legt eine eigene Datenbank an. Dort müsste jedes Medikament aufgenommen werden, was sehr zeitaufwendig ist.

Ein weiteres Problem ist, dass zwar auf allen Schachteln aber nicht auf jeder Ampulle ein Strichcode vorhanden ist. Man könnte die Ampullen manuell mit Barcodes bestücken oder über den Strichcode auf der Verpackung eine Ampulle einscannen. [1]



### 3.1.1.2 OCR

<b>Beschreibung</b>	OCR ist die automatische Texterkennung innerhalb von Bildern. Mit OCR Scannern können ganze Textstellen eingescannt und digitalisiert werden. [2]
<b>Erfassung von OCR Code</b>	Zur Erfassung von Texten gibt es spezielle Scanner, die über das Textfeld gezogen werden müssen. [2]
<b>Probleme</b>	Der Scanner für die Texterkennung muss mit einer Berührung über den Text gezogen werden. Da die Ampullen rund sind, kann das zu Problemen führen. Somit kann es sein, dass mehrere Male über den Text gefahren werden muss, bis er eingescannt wird (Zuverlässigkeit sinkt). Zudem ist es sehr schwer, ein Programm zu entwickeln, das aus dem gescannten Text die richtigen Informationen herausliest. [2]

### 3.1.1.3 DataMatrix

<b>Beschreibung</b>	<p>Gemäss swissmedic soll im Jahr 2016/2017 die Beschriftung der Verpackungen und Medikamente einheitlich werden. Dies soll mit einer sogenannten „GS1 DataMatrix“ erfolgen. Die DataMatrix ist ein 2-dimensionales Symbol (Barcode), welches die Kennzeichnung der Produkte im Gesundheitswesen ermöglicht. Zum einen ist es möglich die Global Trade Item Number, das Verfallsdatum und die Batch Nummer zu erfassen. Es gibt aber mehr als 150 mögliche Applikationen, die je nach Anwendung verwendet werden können.</p> <p>Folgende Vorteile bringt die DataMatrix:</p> <ul style="list-style-type: none"><li>• Einheitliche und weit verbreitete Codierungen</li><li>• Erfassung von Produkten für die Pflegedokumentation</li><li>• Erfassung und Prüfung des Verfalldatums</li><li>• Erleichterter Wareneingang</li><li>• Lagerverwaltung der Produkte</li><li>• Unterstützung der Rückverfolgbarkeit</li><li>• Unterstützung der elektronischen Bestell-, Liefer- und Rechnungsstellungsprozesse</li></ul>
<b>Erfassung von DataMatrix</b>	Das Einlesen eines QR-Codes erfolgt analog eines Barcodes. Die Lesegeräte besitzen auch hier eine Kamera und einen Decoder.
<b>Lesegeräte</b>	Gleiche Geräte wie beim Barcode, müssen aber 2D fähig sein.
<b>Probleme</b>	<p>Die DataMatrix ist in der Schweiz noch nicht auf allen Medikamenten. Auch hier ist wieder das Problem, wie man an die Herstellerdaten kommt.</p> <p><i>Das DataMatrix Datenblatt, aus dem diese Informationen stammen, ist im Anhang.</i></p>

---

### 3.1.1.4 Neue Kennzeichnung

<b>Beschreibung</b>	Heute gibt es bereits verschiedene Internetseiten, auf denen man selber Codes erstellen und die gewünschten Daten hinterlegen kann. Es gibt auch eine Seite für die Erstellung von DataMatrix Codes. Man könnte für jedes Medikament, das in die Basis kommt einen solchen DataMatrix Code erstellen. Dort könnte man dann den Namen und das Ablaufdatum hinterlegen, ihn ausdrucken und auf das Medikament kleben.
<b>Probleme</b>	Das Ganze ist sehr aufwendig und zeitintensiv. Der Vorteil einer schnellen Scanstation geht verloren.

### 3.1.1.5 Fazit

<b>Fazit</b>	Es kommen nur vier verschiedene Codierungen in Frage, weil nur diese auf den heutigen Medikamenten zu finden sind. Der Barcode und die DataMatrix eignen sich für dieses Projekt sehr gut. Die Texterkennung und eine neue Kennzeichnung sind eher weniger geeignet, da es ein Mehraufwand für die Benutzer wäre.
--------------	---

---

## 3.1.2 Marktübersicht bestehender Lösungen

<b>Einleitung</b>	Zur Analyse zählt auch abzuklären, wie dieses Problem an anderen Orten gelöst wird und was für bestehende Lösungen schon im Markt sind.
-------------------	---

### 3.1.2.1 Lösungen in anderen Unternehmen

<b>Beschreibung</b>	Um herauszufinden wie in anderen Unternehmen das Ablaufen der Medikamente überprüft wird, wurden verschiedene Personen aus dem Umfeld der Gruppenmitglieder befragt. Alle arbeiten an verschiedenen Orten, müssen aber bei der Arbeit dasselbe Problem lösen.
<b>Spitex Gossau</b>	Bei der Spitex in Gossau werden alle Waren (auch die Medikamente) selbst gelabelt. Bei Gebrauch können sie eingescannt werden und mit dem Programm am PC den einzelnen Personen zugewiesen werden. Das Programm gibt aber nur den Produktnamen und den Preis bekannt (wie beim Einkaufen in der Migros/Coop). Das Ablaufdatum muss manuell durch die Mitarbeiterinnen überprüft werden. Die abgelaufenen Medikamente müssen entsorgt werden.
<b>Krankenhaus Wattwil</b>	Im Krankenhaus Wattwil verläuft alles noch manuell. Zum einen werden die Medikamente beim wöchentlichen Schrankputzen auf die Haltbarkeit überprüft und zum anderen beim Herausgeben der Medikamente. Die Medikamente werden nur eingescannt, um sie den Patienten zuzuordnen und in Rechnung zu stellen. Die abgelaufenen Medikamente müssen auch hier weggeworfen werden.
<b>Drogerie Peterer Flawil</b>	Bei der Drogerie Peterer in Flawil wird das Ablaufen der Medikamente mit Hilfe von Excel Listen geführt. Beim Wareneingang gibt ein Mitarbeiter alle relevanten Daten (Name, Ablaufdatum, etc) in eine Excel Liste ein. Jeden Monat wird dann aus dem Excel eine Liste, mit den Medikamenten die ablaufen, exportiert. Werden die Medikamente direkt in die Regale im Laden eingeräumt, überprüft ein Mitarbeiter alle Medikamente auf die Haltbarkeit und trägt diese in eine Liste im PC ein. Beides ist sehr aufwendig und kostet viel Geld. Auch hier werden die abgelaufenen Medikamente entsorgt.

- Galexis (Hauptlieferant der REGA Medikamente)** Es wurde extra noch Galexis angefragt, weil dieser Vollgrossist der Hauptlieferant der REGA ist. Auch bei ihnen müssen die Verfalldaten der Medikamente überprüft werden, um die Lagerführung nach FEFO (first expiry-first out) zu gewährleisten. Die Daten werden von Hand erfasst oder mittels Schnittstelle vom Hersteller an Galexis übermittelt. Sie scannen für die eindeutige Identifizierung des Artikels den Pharmacode (Strichcode).
- Fazit** In keinem Unternehmen läuft die Überprüfung der Haltbarkeit automatisch ab. Es funktioniert immer noch alles von Hand, was sehr zeitaufwendig ist. Alle haben grosses Interesse an unserer Lösung gezeigt, was bedeutet, dass das Medikamententrackingsystem auch in anderen Unternehmen zum Einsatz kommen könnte.

### 3.1.2.2 Lösungen auf dem Markt

**Beschreibung** Auch im Internet wurde recherchiert, ob es ähnliche Systeme schon auf dem Markt gibt.

**Barcodeleser Spital** Die Firma „mobit“ hat mehrere Scanner im Angebot, die speziell für den Einsatz im Krankenhaus entwickelt wurden. Sie alle sind desinfizierbar und können die Barcodes auf einer Vielzahl von Medizinprodukten lesen. Da sie kabellos sind, ermöglichen sie die automatische Datenerfassung direkt am Behandlungsort. Leider konnte nicht herausgefunden werden, ob mit diesen Scanner auch die Haltbarkeit der Medikamente überprüft werden kann. [3]



Abbildung 12: Barcodeleser Spital

**Scanmodul von Datalogic** Die Firma „Datalogic“ bietet verschiedene Scanmodule für das Gesundheitswesen an. Zum einen gibt es einen Handscanner mit Display, der im Operationssaal und anderen Bereichen sterile Geräte wie Katheter und Instrumente verfolgt, damit diese nur vor Ablauf ihrer Haltbarkeit verwendet werden. Die Überprüfung der Haltbarkeit der Medikamente sieht dieses System nicht vor. Auch diese Scanner bestehen aus speziell entwickelten Werkstoffen die Infektionen zu vermeiden helfen und selbst aggressiven Desinfektionsmitteln standhalten. [4]

**ScanModul** Die Firma ScanModul bietet ein System an, bei der die Artikel mit Barcodes und RFID ausgestattet werden. Im RFID ist auch das Ablaufdatum enthalten. Da RFID über mehrere Meter gescannt werden kann, gewährleisten stationäre Scanner im ganzen Gebäude die Überwachung der Artikel. Somit weiss das System immer, welcher Artikel sich wo im Gebäude befindet. [5]

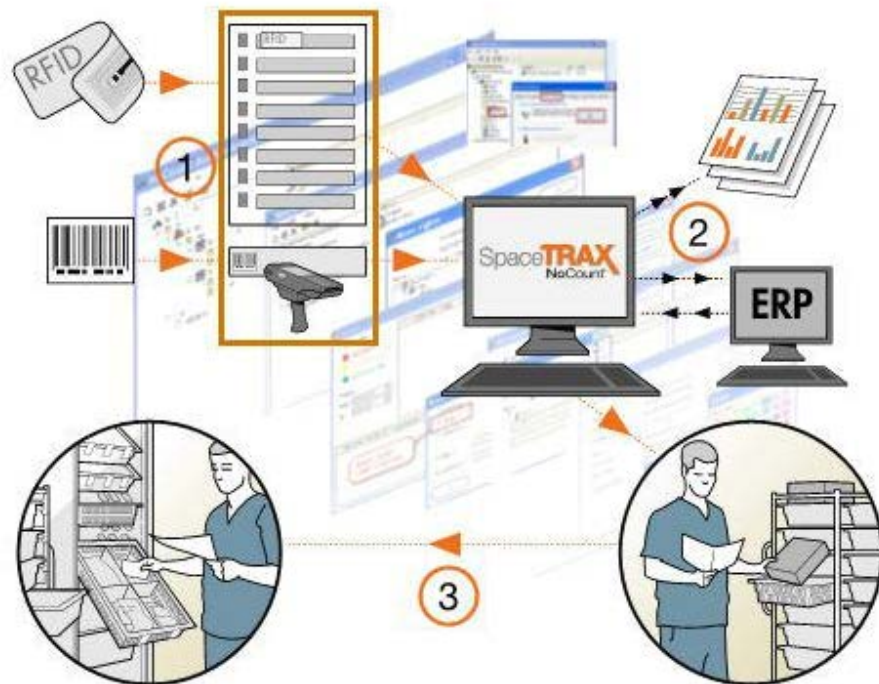


Abbildung 13: Funktionen ScanModul

## Fazit

Das System der Firma ScanModul kommt unserem Projekt am nächsten. Sie lösen die Überprüfung des Ablaufdatums mit RFID. Dies ist bei uns leider nicht möglich, da die Medikamente keine RFID besitzen und deren Einsatz deshalb zu aufwendig wäre. Ansonsten gibt es auf dem Markt keine ähnlichen Lösungen.

### 3.1.3 Patentrecherche

**Beschreibung** Um keine Schwierigkeiten mit Patenten zu bekommen, wurde eine Patentrecherche durchgeführt. Die Recherche wurde auf der Seite des Europäischen Patentamtes durchgeführt.

**Ergebnis** Es wurden folgende Suchbegriffe beim Europäischen Patentamt eingegeben:

Stichwort	Treffer	relevante Treffer
Scanner	16	0
Trackingsystem	0	0
Barcode	27	0
QR-Code	23	0
OCR	70	0
DataMatrix	3	0
Trackingverfahren	0	0
Scansysteme	0	0
Verfolgungssystem	0	0
Markierverfahren	0	0

**Tabelle 8: Übersicht Patentrecherche**

Bei einigen Treffer war das Patent so alt, dass es nicht mehr gültig ist. Andere Patente beziehen sich auf den Aufbau des Scanners, und da als Teil dieser Arbeit ein Scanner ausgesucht und nicht gebaut werden muss, sind diese Patente für diese Arbeit nicht relevant.

**Fazit** Obwohl es viele Patente zu diesem Thema gibt, wurden speziell für diese Arbeit keine relevanten Patente gefunden. Somit können bei der Lösungssuche der Teilfunktionen alle Ideen in Erwägung gezogen werden.

## 3.2 Analyse der Softwareanforderungen

**Beschreibung** Das folgende Kapitel beschreibt die Softwareanforderungen, welche die Applikation erfüllen muss. Es beinhaltet die Use Cases, nicht funktionale Anforderungen, das Domain Model und ein Zustandsdiagramm.

### 3.2.1 Use Cases

**Beschreibung** Das folgende Use Case Diagramm zeigt eine Übersicht der zu implementierenden Use Cases und deren Akteure. Darauffolgend werden die Use Cases einzeln beschrieben.

## Use Case Diagramm

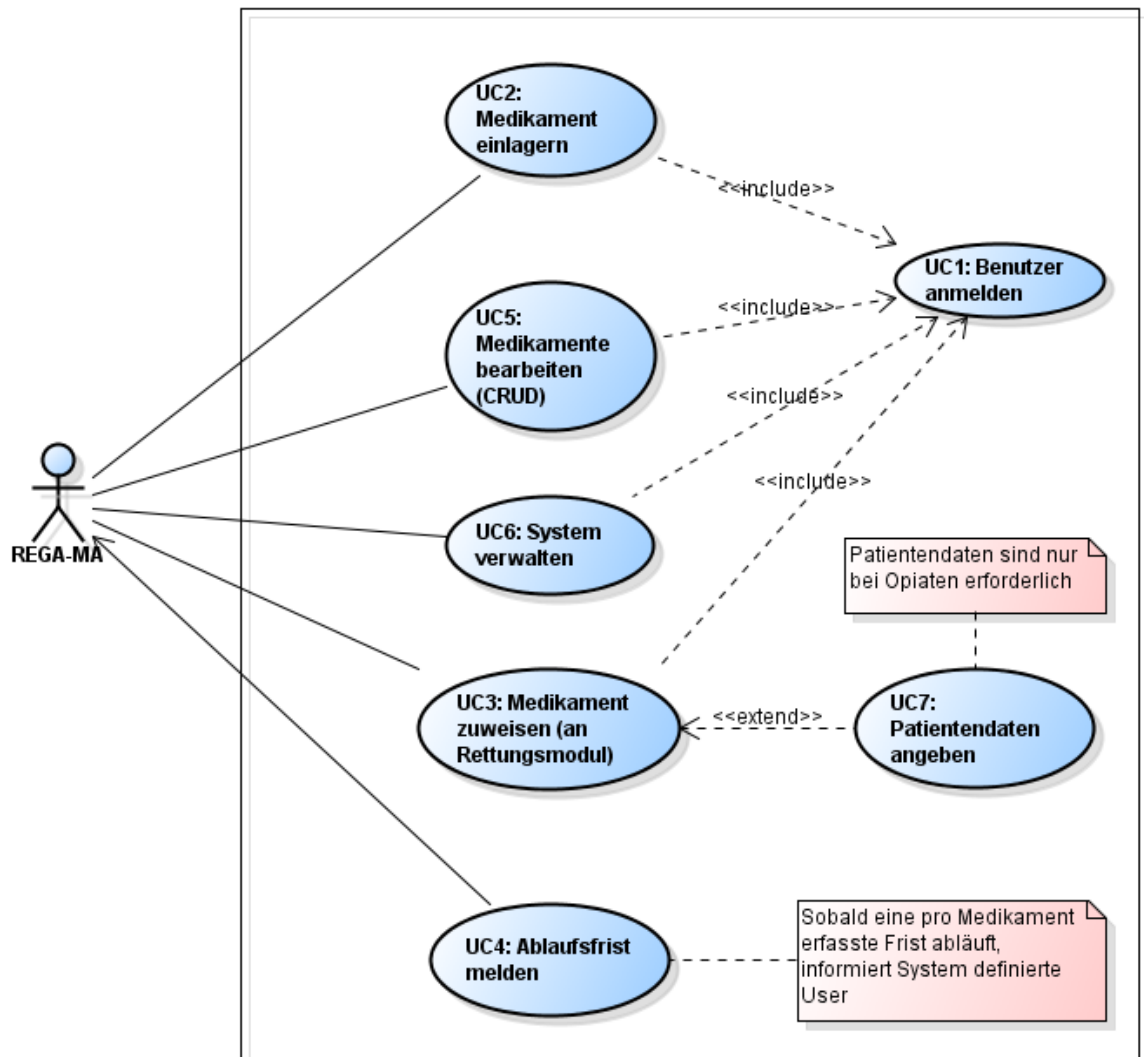


Abbildung 14: Use Case Diagramm

### 3.2.1.1 Akteure

Akteure	Name	Beschreibung
	REGA-MA (Benutzer)	Ist ein Mitarbeiter der REGA, welcher auf einer der Basen arbeitet. Dieser besitzt die Rechte Medikamente einzulagern und zuzuweisen. REGA Mitarbeiter mit Super Admin Rechten können zusätzlich dazu Medikamente bearbeiten und das System verwalten.

### 3.2.1.2 UC1: Benutzer anmelden

UC1	Mapped Requirement (aus Pflichtenheft)	-
	Primary Actor	REGA-MA
	Pre Conditions	REGA-MA ist als Benutzer im System erfasst (durch Super Admin aufgeschaltet, siehe UC6)
	Post Conditions	Benutzer ist im System angemeldet
	Main Scenario	<ol style="list-style-type: none"> <li>1. Benutzer meldet sich mit seinem Benutzernamen und Passwort im System an</li> <li>2. System lässt den Benutzer auf das System zugreifen.</li> </ol>

### 3.2.1.3 UC2: Medikament einlagern

UC2	Mapped Requirement (aus Pflichtenheft)	TM1/TM4
	Primary Actor	REGA-MA
	Pre Conditions	Benutzer im System angemeldet
	Post Conditions	Medikament erfolgreich im Lager erfasst
	Main Scenario	<ol style="list-style-type: none"> <li>1. Benutzer wählt im System aus, dass er ein Medikament einlagern möchte.</li> <li>2. Benutzer scannt einen Medikamentencode.</li> <li>3. System zeigt den Namen des Medikamentes an.</li> <li>4. Benutzer erfasst die Anzahl des einzulagernden Medikamentes im System. Solange das Ablaufdatum nicht im Medikamentencode enthalten ist, erfasst er zudem das Ablaufdatum.</li> <li>5. Benutzer bestätigt Einlagerung.</li> <li>6. System speichert Einträge.</li> </ol> <p>Benutzer wiederholt Schritte 1-6 beliebig oft</p>
	Extensions	<ol style="list-style-type: none"> <li>2a. Medikamentencode nicht im System vorhanden:             <ol style="list-style-type: none"> <li>1. System verhindert, dass Medikament eingelagert werden kann.</li> </ol> </li> </ol>

### 3.2.1.4 UC3: Medikament zuweisen (an Rettungsmodul)

UC3	Mapped Requirement (aus Pflichtenheft)	TM3
	Primary Actor	REGA-MA
	Pre Conditions	Benutzer im System angemeldet
	Post Conditions	Medikament erfolgreich einem Rettungsmodul zugeteilt
	Main Scenario	<ol style="list-style-type: none"> <li>1. Benutzer wählt im System aus, dass er ein Medikament an ein Rettungsmodul zuweisen möchte.</li> <li>2. Benutzer scannt einen Medikamentencode.</li> <li>3. System zeigt den Namen des Medikamentes an.</li> <li>4. Benutzer wählt das gewünschte Rettungsmodul aus.</li> <li>5. Benutzer bestätigt Zuweisung.</li> <li>6. System speichert Zuweisung.</li> </ol> <p>Benutzer wiederholt Schritte 1-6 beliebig oft</p>
	Extensions	<ol style="list-style-type: none"> <li>2a. Gescanntes Medikament hat Ablaufdatum erreicht oder überschritten:             <ol style="list-style-type: none"> <li>1. System erlaubt keine Zuweisung des Medikamentes an ein Rettungsmodul</li> <li>2. System warten auf weiteren Scanvorgang (Schritt 2)</li> </ol> </li> <li>4a. Medikament bereits im Rettungsmodul vorhanden:             <ol style="list-style-type: none"> <li>1. System kennzeichnet Medikament im Rettungsmodul als gebraucht                 <ol style="list-style-type: none"> <li>*a. Mehrere gleiche Medikamente im Rettungsmodul enthalten:                     <ol style="list-style-type: none"> <li>1. Benutzer gibt an, um welches Exemplar es sich handelt</li> </ol> </li> </ol> </li> </ol> </li> </ol>

### 3.2.1.5 UC4: Ablauffrist melden

UC4	Mapped Requirement (aus Pflichtenheft)	TM2
	Primary Actor	REGA-MA
	Pre Conditions	Definierte Frist eines Medikamentes abgelaufen
	Post Conditions	Definierte User via Mail über abgelaufene Frist informiert
	Main Scenario	<ol style="list-style-type: none"> <li>1. Das System erkennt, dass die Ablauffrist eines Medikamentes abgelaufen ist.</li> <li>2. Das System versendet eine Email an im System konfigurierte User.</li> </ol>



### 3.2.1.6 UC5: Medikamente bearbeiten (CRUD)

UC5	Mapped Requirement (aus Pflichtenheft)	TM3/TO2
	Primary Actor	REGA-MA
	Pre Conditions	Benutzer als Super Administrator im System angemeldet
	Post Conditions	Änderungen im System erfolgreich nachvollzogen
	Main Scenario	<ol style="list-style-type: none"> <li>Benutzer erfasst Medikamententypen im System. Nebst Namen und Hersteller kann der Benutzer auch die Ablauffrist (siehe UC4) eines Medikamententyps definieren. Zudem kann er festlegen, ob das Ablaufdatum manuell überschrieben werden darf (dies ist bei gewissen Medikamententypen betrieblich notwendig).</li> <li>System speichert Änderungen.</li> </ol> Benutzer wiederholt Schritte 1-2 beliebig oft

### 3.2.1.7 UC6 System verwalten

UC6	Mapped Requirement (aus Pflichtenheft)	TM2
	Primary Actor	REGA-MA
	Pre Conditions	Benutzer als Super Administrator im System angemeldet
	Post Conditions	Änderungen im System erfolgreich nachvollzogen
	Main Scenario	<ol style="list-style-type: none"> <li>Benutzer erfasst Benutzer im System. Er kann zudem definieren, welche Benutzer respektive Emailadressen beim Erreichen einer Ablauffrist (siehe UC4) eines Medikamentes informiert werden sollen.</li> <li>System speichert Änderungen.</li> </ol> Benutzer wiederholt Schritte 1-2 beliebig oft

### 3.2.1.8 UC7: Patientendaten angeben (optionale Anforderung)

**Anmerkung** Diese Anforderung ist optional und hat daher geringe Priorität. Sie wird nur bei genügend Zeit umgesetzt.

UC7	Mapped Requirement (aus Pflichtenheft)	N1
	Primary Actor	REGA-MA
	Pre Conditions	Benutzer im System angemeldet
	Post Conditions	Patientendaten im System erfolgreich gespeichert
	Main Scenario	<ol style="list-style-type: none"> <li>System erkennt beim Zuweisen eines Medikaments (siehe UC3), dass es sich um ein Opiat handelt.</li> <li>Benutzer muss Patientendaten erfassen, welcher das Opiat erhalten hat.</li> <li>System speichert Patientendaten.</li> </ol>

### 3.2.2 Nicht funktionale Anforderungen

**Anmerkung** Folgend aufgeführt sind die in der Applikation einzuhaltenden nicht funktionalen Anforderungen (NFRs).

NFRs	Mapped Requirement	Name	Beschreibung
	TM5	Kompatibilität	Die Applikation kann auf allen 12 REGA-Basen eingesetzt werden ohne Änderungen am Source Code durchführen zu müssen.
	TM6	Effiziente Bedienung	Arbeitsprozesse „Medikament einlagern“ (UC2) und „Medikament zuweisen“ (UC3) müssen mindestens gleich schnell ablaufen wie der bisherige manuelle Prozess.
	TO1	Einfache Bedienoberfläche	Das Frontend soll so aufgebaut sein, dass die REGA-Mitarbeiter ohne Schulung das System verstehen.
	TO4	Einfache in Betriebnahme und Betrieb	Die Applikation soll durch REGA-Mitarbeiter auf dem Tablet in Betrieb genommen werden können.
	TO6	iPad	Die Arbeitsprozesse „Medikament einlagern“ (UC2) und „Medikament zuweisen“ (UC3) müssen auf einem iPad durchführbar sein.
	TO9	Zuverlässigkeit	Das System hat eine Uptime von mindestens 90%.

### 3.2.3 Domain Model

**Beschreibung** Das folgende Domainmodell zeigt eine Übersicht über die in der Problemdomäne vorhandenen Objekte. Zu jedem dieser Objekte folgt im Anschluss eine kurze Erklärung.

**Domain Model**

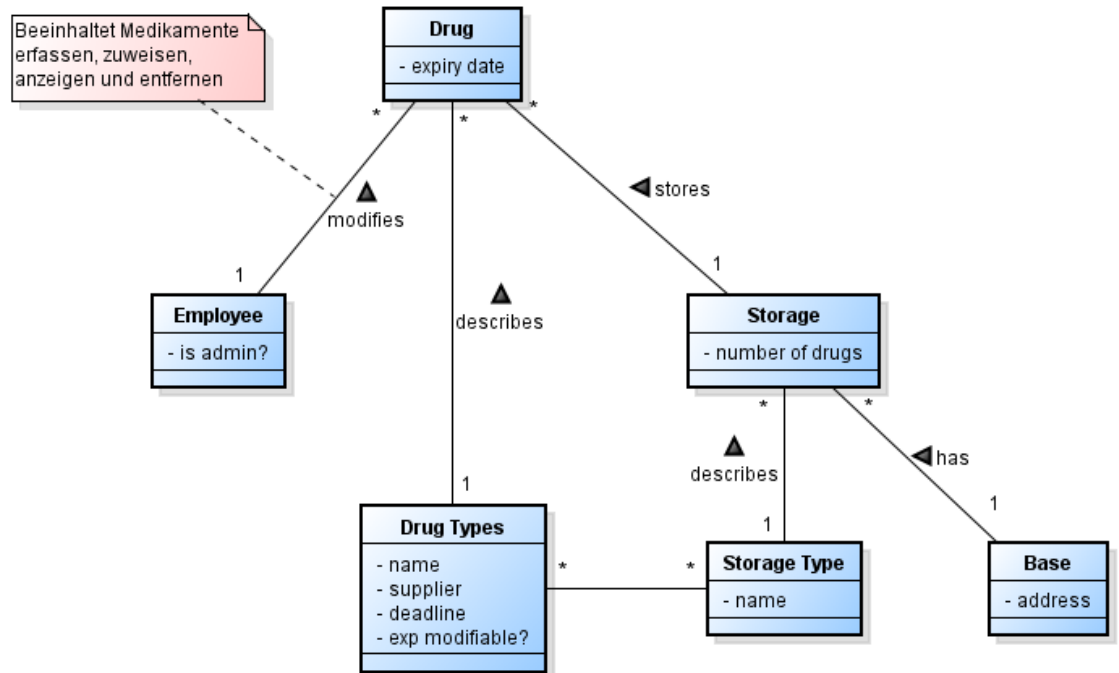


Abbildung 15: Domain Model

- Base (Basis)** Entspricht einer von 12 vorhandenen REGA-Basen. Jede Basis hat eine Adresse (address).
- Storage Type (Lagertyp)** Steht für die verschiedenen Lagertypen (z.B. das Lager selbst oder die verschiedenen Rettungsmodule). Jeder Lagertyp hat einen Namen (name).
- Storage (Lagerort)** Repräsentiert den Lagerort der Medikamente. Dies kann das Lager selbst oder ein Rettungsmodule sein. Lagerorte haben eine bestimmte Anzahl Medikamente (number of drugs).
- Drug Types (Medikamententyp)** Steht für die verschiedenen Medikamententypen die die REGA im Einsatz hat. Name (name), Lieferant (supplier), Ablauffrist (deadline) und ein Flag, ob das Ablaufdatum überschrieben werden darf (exp modifiable), sind für ein Medikamententyp definiert.
- Drug (Medikament)** Entspricht dem konkreten Medikament (ein Stück). Jedes Medikament besitzt ein Ablaufdatum.(expiry date).
- Employee (Mitarbeiter)** Widerspiegelt einen REGA-Mitarbeiter der im System aufgeschaltet ist. Es wird festgehalten ob der Mitarbeiter Super Admin Rechte (is admin) hat.

### 3.2.4 Zustandsdiagramm

**Beschreibung** Die folgende Abbildung zeigt das Zustandsdiagramm. Es stellt die einzelnen Zustände des Systems dar und zeigt wie unter den Zuständen gewechselt werden kann.

#### Zustandsdiagramm

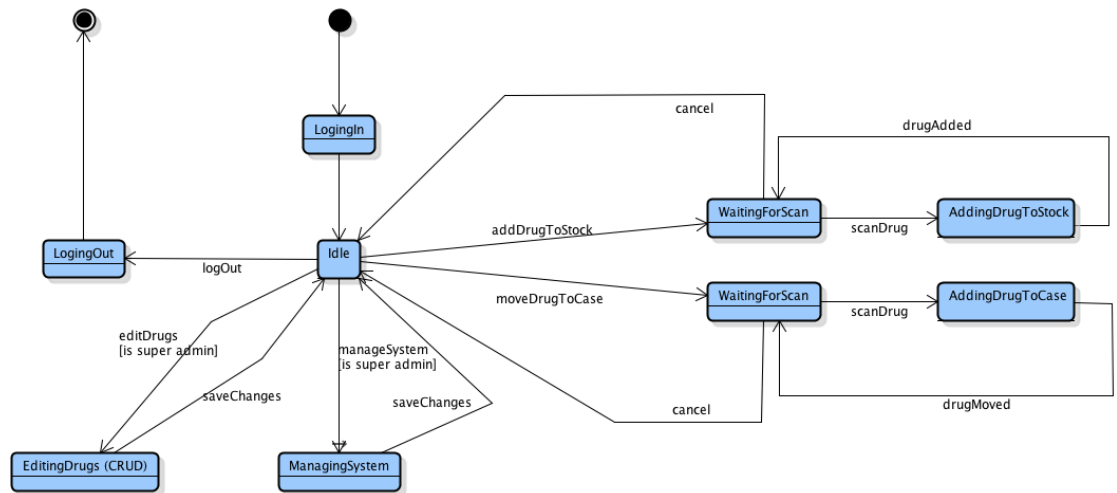


Abbildung 16: Zustandsdiagramm

- LoggingIn** Zustand tritt beim Einloggen ins System ein.
- Idle** Das System wartet auf eine Interaktion.
- WaitingForScan** Zustand tritt ein durch **addDrugToStock**. Wenn nicht via **cancel** auf den Idle Zustand gewechselt wird, dann können weitere Medikamente dem Lager hinzugefügt werden
- AddingDrug-ToStock** Zustand tritt durch **scanDrug** ein. Das Medikament wird dem Lager hinzugefügt.
- WaitingForScan** Zustand tritt ein durch **moveDrugToCase**. Wenn nicht via **cancel** auf den Idle Zustand gewechselt wird, dann können weitere Medikamente den Rettungsmodulen zugewiesen werden.
- MovingDrugToCase** Zustand tritt durch **moveDrug** ein. Das Medikament wird einem Rettungsmodul zugewiesen.
- ManagingSystem** Zustand tritt ein durch **manageSystem**, wobei der Benutzer als Super Admin hinterlegt sein muss.
- Editing-Drugs(CRUD)** Zustand tritt ein durch **editDrugs**, wobei der Benutzer als Super Admin hinterlegt sein muss.
- LoggingOut** Zustand tritt beim Ausloggen ein.

## 3.2.5 Wireframes

### Beschreibung

Die folgenden Abbildungen dienen als Modell zu Präsentationszwecken und um dem Kunden einen Einblick in die mögliche Benutzeroberfläche zu geben.

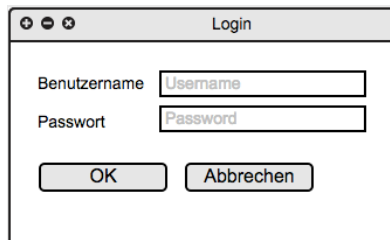
### Startscreen



Die Benutzung des Systems ist nur mit einem registrierten Benutzer möglich.

Abbildung 17: Wireframe Startscreen

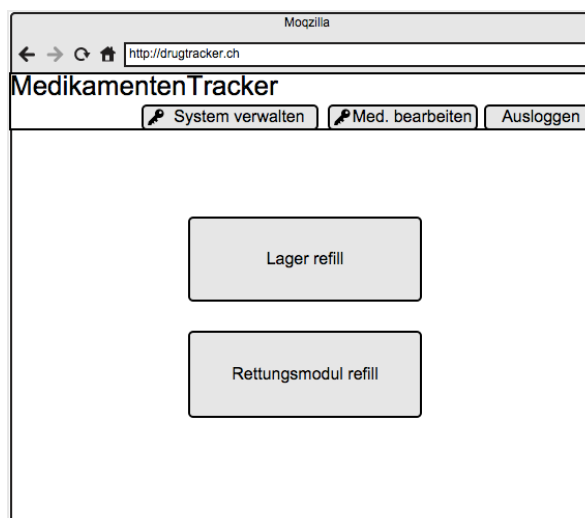
### Einloggen (UC1: Benutzer anmelden)



Über ein Login Formular kann sich ein Benutzer anmelden.

Abbildung 18: Wireframe Login

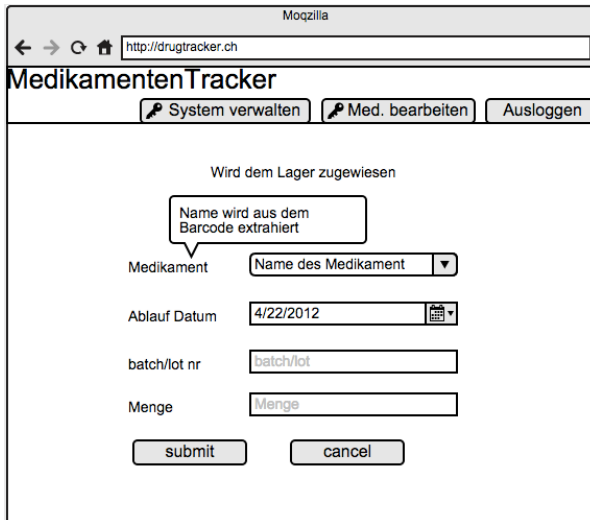
### Vorauswahl



Hier wird zwischen Lager oder Rettungsmodul auffüllen unterschieden.

Abbildung 19: Wireframe Vorauswahl

**Lager auffüllen  
(UC2: Medikament  
einlagern)**



MedikamentenTracker

System verwalten | Med. bearbeiten | Ausloggen

Wird dem Lager zugewiesen

Name wird aus dem Barcode extrahiert

Medikament: Name des Medikament

Ablauf Datum: 4/22/2012

batch/lot nr: batch/lot

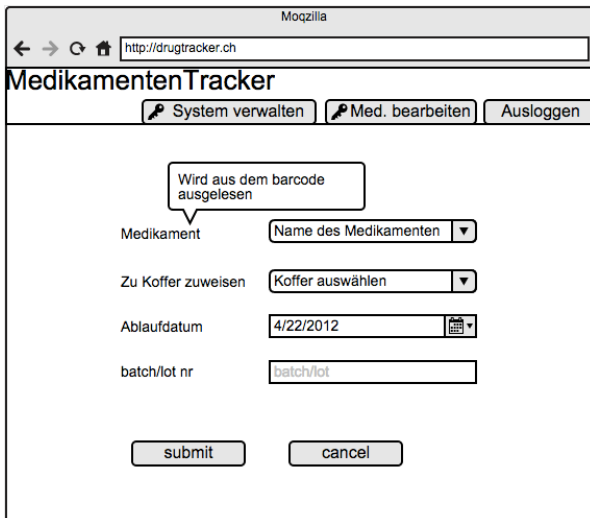
Menge: Menge

submit | cancel

Durch scannen des Barcodes wird das Medikament erkannt. Danach können weitere Eingaben getätigt und zum Schluss die Daten ans System übermittelt werden.

Abbildung 20: Wireframe Lager auffüllen

**Rettingsmodul  
auffüllen (UC3:  
Medikament zuwei-  
sen)**



MedikamentenTracker

System verwalten | Med. bearbeiten | Ausloggen

Wird aus dem barcode ausgelesen

Medikament: Name des Medikamenten

Zu Koffer zuweisen: Koffer auswählen

Ablaufdatum: 4/22/2012

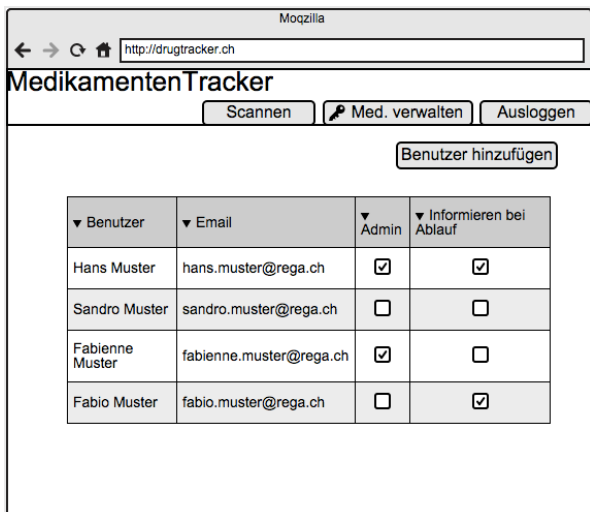
batch/lot nr: batch/lot

submit | cancel

Durch scannen des Barcodes wird das Medikament erkannt. Danach wird das gewünschte Rettungsmodule ausgewählt. Nach Eingabe des Ablaufdatums können die Daten ans System übermittelt werden.

Abbildung 21: Wireframe Rettungsmodule auffüllen

**System verwalten  
(UC6: System ver-  
walten)**



MedikamentenTracker

Scannen | Med. verwalten | Ausloggen

Benutzer hinzufügen

Benutzer	Email	Admin	Informieren bei Ablauf
Hans Muster	hans.muster@rega.ch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sandro Muster	sandro.muster@rega.ch	<input type="checkbox"/>	<input type="checkbox"/>
Fabienne Muster	fabienne.muster@rega.ch	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fabio Muster	fabio.muster@rega.ch	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Die Verwaltung des Systems steht nur dem Super Admin zur Verfügung. Hier können Benutzer erfasst und mit den nötigen Rechten versehen werden. Auch können Benutzer der Benachrichtigungsgruppe hinzugefügt werden.

Abbildung 22: Wireframe System verwalten

**Medikamente verwalten (UC5: Medikamente verwalten)**

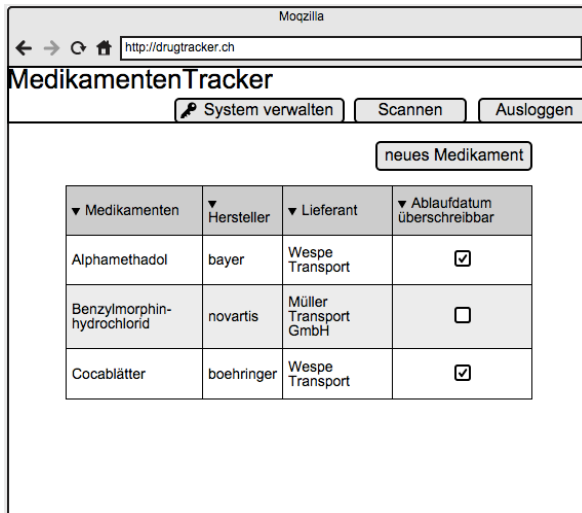


Abbildung 23: Wireframe Medikamente verwalten

Die Verwaltung der Medikamente steht nur dem Super Admin zur Verfügung. Hier können neue Medikamente erfasst und bestehende bearbeitet werden. Auch können Medikamente markiert werden, wenn ihr Ablaufdatum überschreibbar sein soll.

**Opiat (UC7: Patientendaten eingeben)**

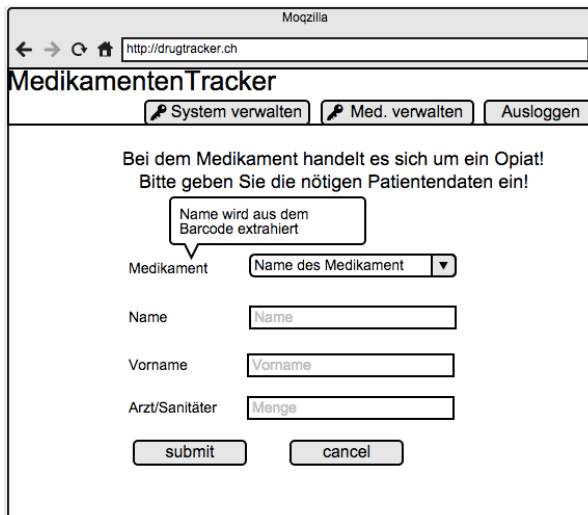


Abbildung 24: Wireframe Opiat

Wenn Opiate nachgefüllt werden, muss davon ausgegangen werden, dass eines oder mehrere verwendet wurden. In dem Fall wird ein zusätzliches Fenster geöffnet, welches um die Patientendaten bittet. Dies ist von Gesetzes wegen nötig.

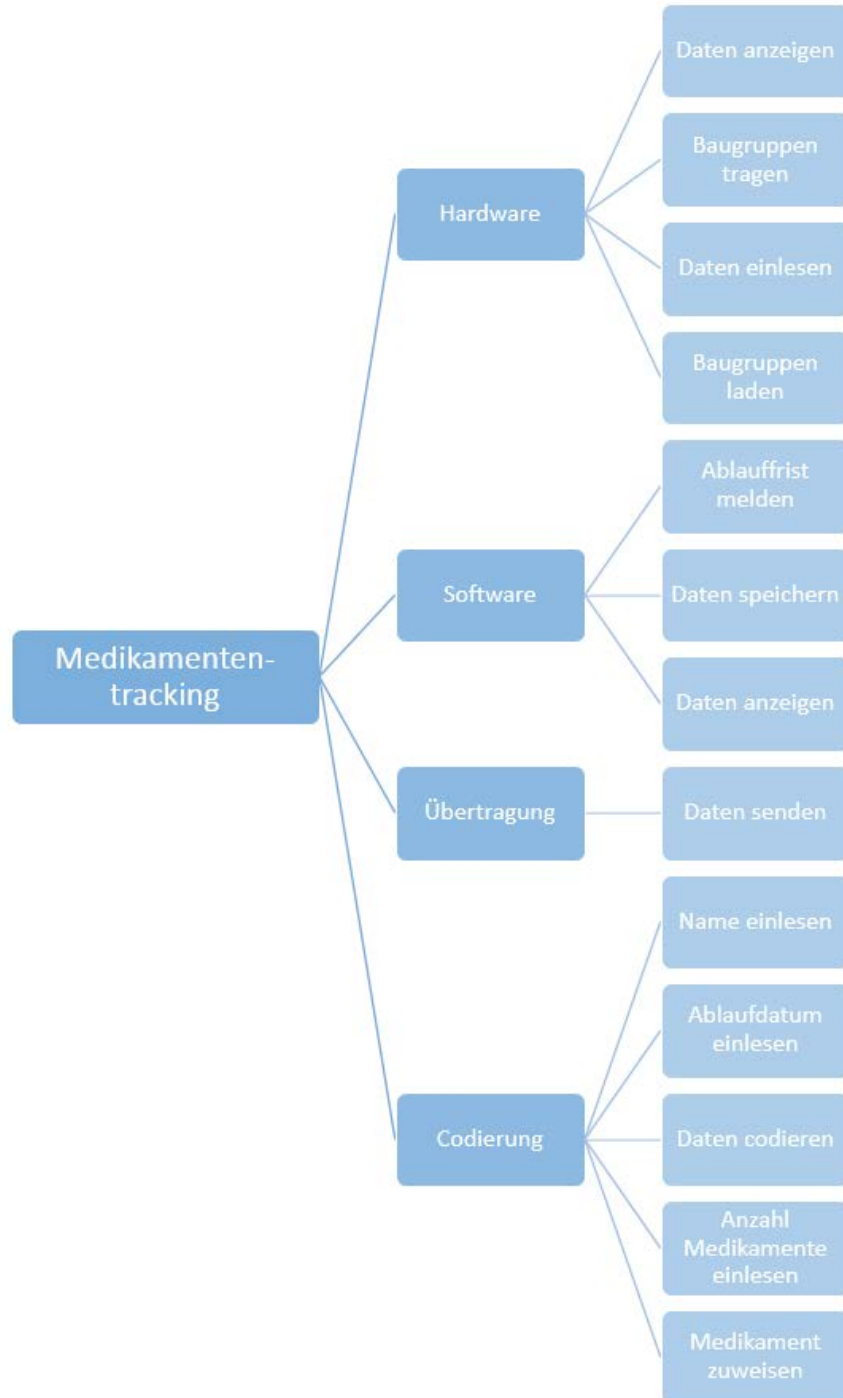
### 3.3 Analyse der Konstruktionsanforderungen

**Beschreibung**

Der Konstruktionsteil wird in mehrere Teilfunktionen unterteilt. Zu jeder dieser Teilfunktion werden anschliessend mehrere Lösungsvorschläge beschrieben, die zusammen den Morphologischen Kasten mit drei verschiedenen Varianten bilden. Mit Hilfe der Nutzwertanalyse wird aus diesen drei Varianten die am besten geeignete ausgewählt und danach designed.

### 3.3.1 Funktionsstruktur

**Funktionsstruktur** Das ganze System wurde in mehrere Teilfunktionen aufgeteilt, die in der Funktionsstruktur abgebildet werden. Für die Unterteilung wurden auch die Funktionen der Software berücksichtigt, da diese auch in den Morphologischen Kasten gehören. Nur somit ist gewährleistet, dass am Schluss eine Variante entsteht, bei der die Konstruktion und die Software zusammen harmonisieren.



**Abbildung 25: Funktionsstruktur**



---

### 3.3.2 Lösungen der Teilfunktionen

---

**Einleitung** Für die einzelnen Teilfunktionen aus der Funktionsstruktur wurden verschiedene Lösungen gesucht, die danach zusammen den Morphologischen Kasten bilden. Die verschiedenen Lösungen der Teilfunktionen werden nun hier etwas genauer beschrieben.

#### 3.3.2.1 Daten anzeigen (Display)

**Beschreibung** Das Display zeigt alle notwendigen Daten wie z.B: Name des Medikamentes, Verbleib, Ablauf und Bestand an. Es wird vom Benutzer bedient.

**iPad**

**Beschreibung:**

Der Tagesablauf der REGA wird heute mit einem iPad via Touchscreen geregelt.

**Vorteil:**

Die REGA arbeitet bereit mit iPads und somit können alle Benutzer das iPad schon bedienen. Es ist möglich das Scansystem wie eine App zu gestalten.

**Nachteil:**

Der Scanner muss für das iPad geeignet sein.

**Entscheidung:**

Dieser Lösungsansatz wird weiter verfolgt.

**PC**

**Beschreibung:**

Ein PC dient als Display.

**Vorteil:**

Die Applikation muss nicht als mobile App geschrieben werden. Alle Benutzer können den PC bedienen.

**Nachteil:**

Herr Reichenbach möchte keinen PC im Medikamentenraum. Somit wäre der Bildschirm nicht im selben Raum.

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Smartphone**

**Beschreibung:**

Das Smartphone dient als Bedienoberfläche

**Vorteil:**

Das Programm kann als mobile App gestaltet werden. Das Gerät ist klein und kompakt.

**Nachteil:**

Es müssen mehrere gleiche Smartphones vorhanden sein und der Scanner muss für das Smartphone geeignet sein.

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Touchscreen**

**Beschreibung:**

Ein Touchscreen dient als Bedienoberfläche.

**Vorteil:**

Der Touchscreen kann genau für diese Anwendung konfiguriert werden.

**Nachteil:**

Der Benutzer muss die Bedienung lernen und der Scanner muss für den Screen geeignet sein.

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Display auf Scanner**

**Beschreibung:**

Im Scanner ist direkt ein Bildschirm integriert.

**Vorteil:**

Es braucht nur ein Gerät, den Scanner. Den Bildschirm hat man automatisch immer dabei.

**Nachteil:**

Entweder ist der Bildschirm sehr klein oder der ganze Scanner sehr gross. Der Bildschirm muss über einen Touchscreen gesteuert werden können ansonsten braucht es zusätzlich eine Tastatur auf dem Scanner, was ihn wiederum vergrössert.

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

### 3.3.2.2 Baugruppen tragen (Unterbau)

**Beschreibung**

Der Unterbau umfasst das ganze System. Zum einen muss das Display und der Scanner aber auch die Stromversorgung (falls vorhanden) darin untergebracht werden.

**Blechkonstruktion**

**Beschreibung:**

Ein einfaches Blechteil befestigt den Display und den Scanner.

**Vorteil:**

Die Fertigung ist einfach und es ist auch platzsparend. Das Blechteil ist einfach zu transportieren, da eher geringes Gewicht. Somit entsteht ein mobiler Arbeitsplatz.

**Nachteil:**

Die Konstruktion muss aus Metall sein.

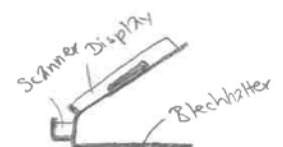


Abbildung 26: Blechkonstruktion

**Entscheidung:**

Dieser Lösungsansatz wird weiter verfolgt.

**Profilkonstruktion Beschreibung:**

Eine Profilkonstruktion dient als Unterbau.

**Vorteil:**

Die Standardprofile sind sehr kostengünstig.

**Nachteil:**

Der Unterbau braucht viel Platz und besteht aus vielen Einzelteilen.

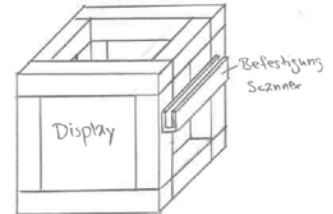


Abbildung 27: Profilkonstruktion

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Profilrahmen an der Wand Beschreibung:**

Ein einfacher Profilrahmen befestigt das Display und den Scanner an die Wand.

**Vorteil:**

Dadurch dass es an der Wand ist, ist es sehr platzsparend. Die Standardprofile sind sehr kostengünstig.

**Nachteil:**

Es müssen Löcher in die Wand gemacht werden. Zudem ist der Rahmen nicht mobil.

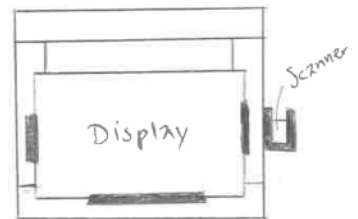


Abbildung 28: Profilrahmen an der Wand

**Entscheidung:**

Dieser Lösungsansatz wird weiter verfolgt.

**Box Beschreibung:**

Aus Blechen wird eine Box hergestellt, in der alle wichtigen Komponenten Platz haben.

**Vorteil:**

Der Unterbau ist geschlossen und man kann alle Komponenten gut integrieren, da er viel Platz bietet.

**Nachteil:**

Die Box braucht viel Platz und ist nicht mobil.

**Entscheidung:**

Dieser Lösungsansatz wird weiter verfolgt.



Abbildung 29: Box

### 3.3.2.3 Daten einlesen (Scanner)

**Beschreibung** Der Scanner ist Hauptbestandteil des Systems. Es gibt ihn in verschiedenen Ausführungen.

**Handscanner** **Beschreibung:**  
Beim Handscanner wird der Scanner durch den Benutzer an den Code geführt.

**Vorteil:**  
Das Medikament muss nicht genau positioniert werden für den Scanvorgang. Es muss auch nicht bei der Scanstation gescannt werden.

**Nachteil:**  
Der Scanner ist lose. Entweder er muss bei Nichtgebrauch im System positioniert werden oder er liegt lose herum.

**Entscheidung:**  
Dieser Lösungsansatz wird weiter verfolgt.

**Scanner stationär** **Beschreibung:**  
Der Scanner ist fix montiert und der Benutzer hält das Medikament vor den Scanner.

**Vorteil:**  
Der Scanner ist fix im System integriert.

**Nachteil:**  
Die Medikamente müssen für das Scannen genau positioniert werden. Ausserdem ist das Scannen nur bei der Scanstation möglich.

**Entscheidung:**  
Dieser Lösungsansatz wird weiter verfolgt.

**Kamera auf Gerät** **Beschreibung:**  
Es wird die Kamera, die im Display enthalten ist, für das Scannen der Daten verwendet. (z.B. beim iPad)

**Vorteil:**  
Es ist nur ein Gerät notwendig (Display und Scanner in einem)

**Nachteil:**  
Die Daten müssen aus einem Bild entnommen werden.

**Entscheidung:**  
Dieser Lösungsansatz wird nicht weiter verfolgt.

**Scanprogramm auf Beschreibung:**

**Gerät**

Es gibt einige Apps die verschiedene Codes einscannen und deren Inhalt anzeigen können. (z.B. für iPad)

**Vorteil:**

Es hat nur ein Gerät (Display und Scanner in einem)

**Nachteil:**

Die Daten müssen aus der mobile App entnommen werden können.

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

### 3.3.2.4 Baugruppen laden (Stromversorgung)

**Beschreibung**

Der Scanner und das Display brauchen Strom oder deren Akkus müssen mit der Zeit aufgeladen werden.

**Ohne Stromversorgung-  
Beschreibung:**

**gung**

Am System gibt es keine Möglichkeit, das Display und den Scanner zu laden.

**Vorteil:**

Es sind keine zusätzlichen Kabel vorhanden und durch das Wegfallen der Steckdosen kann der Unterbau kleiner gestaltet werden.

**Nachteil:**

Das Display und der Scanner müssen separat geladen werden. Während dem Laden kann das System nicht betrieben werden. Man müsste Ladezeiten definieren. (z.B. über Nacht.)

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

**Mit Stromversorgung-  
Beschreibung:**

**gung**

Das Display und der Scanner können direkt am System geladen werden.

**Vorteil:**

Der Scanner und das Display können während dem Betrieb geladen werden und somit ist das System rund um die Uhr betriebsbereit.

**Nachteil:**

Es sind zusätzliche Kabel für die Stromversorgung vorhanden. Zudem muss der Unterbau so konstruiert werden, dass kein Strom auf die Umgebung abgegeben wird.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

### 3.3.2.5 Ablauffrist melden

**Beschreibung** Bevor ein Medikament abläuft (definierte Frist), muss eine Meldung an definierte Benutzer gesendet werden, damit das Medikament ausgetauscht werden kann.

#### Via SMS

**Beschreibung:**

Definierte Benutzer (Mobilnummern) werden mit einem SMS über das baldige Ablaufen eines Medikamentes informiert.

**Vorteil:**

Der Benutzer wird aktiv informiert und muss die Informationen nicht explizit anfordern.

**Nachteil:**

Das Handy muss auf eingehende Nachrichten überprüft werden. Zudem muss es Empfang haben und eingestellt sein. Ein weiterer Nachteil ist, dass jeder Benutzer eine andere Nummer hat. Beim SMS ist die Nachrichtenlänge auf 160 Zeichen begrenzt.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

#### Via Mail

**Beschreibung:**

Definierte Benutzer (Emailadressen) werden mit einem Mail über das baldige Ablaufen eines Medikamentes informiert.

**Vorteil:**

Der Benutzer wird aktiv informiert und muss die Informationen nicht explizit anfordern. Es kann eine allgemeine Mailadresse verwendet werden, worauf alle der Basis Zugriff haben.

**Nachteil:**

Es muss überprüft werden, ob eine Mail gekommen ist.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

#### Meldung im Programm

**Beschreibung:**

Sobald das Programm gestartet wird, gibt es eine Meldung, falls ein Medikament bald abläuft.

**Vorteil:**

Es braucht kein zusätzliches Gerät (Telefon, Computer) für die Meldung.

**Nachteil:**

Benutzer wird erst über das Ablaufen eines Medikamentes informiert, wenn er das Programm aufruft.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

### 3.3.2.6 Daten speichern

<b>Beschreibung</b>	Die Applikationssoftware kann mittels verschiedenen Technologien umgesetzt werden. Je nach verwendeten Technologien sieht auch die Datenspeicherung anders aus.
<b>Server</b>	<p><b>Beschreibung:</b> Die Applikation wird als Webanwendung umgesetzt. Auf sie wird über einen internetfähigen Browser (z.B. Safari) zugegriffen. Die Daten werden auf einem Webserver zentral gespeichert.</p> <p><b>Vorteil:</b> Alle internetfähigen Geräte können via Webbrowser auf die Applikation zugreifen.</p> <p><b>Nachteil:</b> Es wird eine permanente Internetverbindung benötigt. Fehlt diese, ist die Applikation nicht aufrufbar und kann nicht verwendet werden.</p> <p><b>Entscheid:</b> Dieser Lösungsansatz wird weiter verfolgt.</p>
<b>Tablet</b>	<p><b>Beschreibung:</b> Es wird eine Mobile Applikation für das anzuzeigende Gerät (z.B. iPad) entwickelt. Diese läuft lokal auf jedem anzeigenden Gerät (Display). Die Daten werden lokal auf jedem Gerät gespeichert.</p> <p><b>Vorteil:</b> Die Applikation kann begrenzt auch offline verwendet werden.</p> <p><b>Nachteil:</b> Die Lösung ist nicht technologieunabhängig, d.h. bei Wechsel auf ein Display eines anderen Herstellers, ist die Applikation nicht mehr verwendbar.</p> <p><b>Entscheid:</b> Dieser Lösungsansatz wird weiter verfolgt.</p>
<b>Tablet und Server</b>	<p><b>Beschreibung:</b> Die Applikation läuft auf dem anzeigenden Gerät, ähnlich der nativen Mobile Applikation. Allerdings ist diese vom Betriebssystem unabhängig. Die Daten werden lokal auf jedem Gerät und auch auf einem Webserver gespeichert. Die lokalen Daten werden regelmässig mit den Daten auf dem Webserver synchronisiert.</p> <p><b>Vorteil:</b> Die Applikation kann auch offline verwendet werden und ist technologieneutral.</p> <p><b>Nachteil:</b> Umsetzungsaufwand etwas höher.</p> <p><b>Entscheid:</b> Dieser Lösungsansatz wird weiter verfolgt.</p>

### 3.3.2.7 Daten anzeigen (Software)

**Beschreibung** Die Applikationssoftware kann mittels verschiedenen Technologien umgesetzt werden. Die Anzeige der Daten unterscheidet sich insofern, als dass verschiedene Anzeigetechnologien verwendet werden.

**Browser** **Beschreibung:**  
Die Applikation wird als Webanwendung umgesetzt. Auf sie wird über einen internetfähigen Browser (z.B. Safari) zugegriffen. Die Daten werden im Browser angezeigt.

**Vorteil:**

Alle internetfähigen Geräte können via Webbrowser auf die Applikation zugreifen.

**Nachteil:**

Es wird eine permanente Internetverbindung benötigt. Fehlt diese, ist die Applikation nicht aufrufbar und kann nicht verwendet werden.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

**Mobile Applikation** **Beschreibung:**  
Es wird eine Mobile Applikation für das anzuzeigende Gerät (z.B. iPad) entwickelt. Diese läuft lokal auf jedem anzeigenden Gerät (Display). Die Daten werden in der Mobile Applikation angezeigt.

**Vorteil:**

Die Applikation kann begrenzt auch offline verwendet werden.

**Nachteil:**

Die Lösung ist nicht technologieunabhängig, d.h. bei Wechsel auf ein Display eines anderen Herstellers, ist die Applikation nicht mehr verwendbar.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

**Mobile und  
Browser**

**Beschreibung:**

Die Applikation läuft auf dem anzeigenden Gerät (Display), ähnlich der nativen Mobile Applikation. Allerdings ist diese vom Betriebssystem unabhängig. Die Daten werden in der Mobile Applikation angezeigt. Zusätzlich werden die Daten auch im Browser angezeigt.

**Vorteil:**

Die Applikation kann auch offline verwendet werden und ist technologieneutral.

**Nachteil:**

Umsetzungsaufwand etwas höher.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.



### 3.3.2.8 Daten vom Scanner zum Display senden

- Beschreibung** Die Daten die vom Scanner eingelesen werden, müssen an das Display mit dem Programm gesendet werden, damit die Daten ausgewertet werden können. Es gibt verschiedene Varianten, wie die Daten an das Display gesendet werden können.
- Via Kabel**
- Beschreibung:**  
Viele Scanner können mit einem Kabel an ein Display angeschlossen werden, um die Daten somit sicher und zuverlässig zu senden.
- Vorteil:**  
Das Kabel ist ein bewährtes System um Daten sicher und zuverlässig zu übermitteln. Es ist zudem zuverlässig.
- Nachteil:**  
Das Problem ist, dass das Kabel beim Display angeschlossen werden muss. (meistens USB Anschluss). Zudem ist der Scanner durch das Kabel an das Display gebunden.
- Entscheid:**  
Dieser Lösungsansatz wird weiter verfolgt.
- Via Bluetooth**
- Beschreibung:**  
Einige Scanner können die Daten via Bluetooth an das Display senden.
- Vorteil:**  
Durch Bluetooth ist der Scanner nicht an das Display gebunden. Somit können auch Codes gescannt werden, obwohl das Display nicht in unmittelbarer Nähe ist.
- Nachteil:**  
Eine Bluetooth Verbindung muss immer vorhanden sein. Es besteht die Gefahr, dass andere Geräte die Bluetooth-Verbindung stören.
- Entscheid:**  
Dieser Lösungsansatz wird weiter verfolgt.
- Direkt mit Display scannen**
- Beschreibung:**  
Eine Möglichkeit besteht darin, die Daten direkt mit der Kamera oder mit einem Programm auf dem Display einzuscannen. Somit wären die Daten bereits auf dem anzeigenden Gerät (Display) vorhanden.
- Vorteil:**  
Keine Verbindung nötig (kein Kabel, kein Bluetooth)
- Nachteil:**  
Es muss gewährleistet sein, dass die Daten aus einem Foto oder aus dem Scan Programm herausgelesen werden können. (sehr aufwendig)

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

### 3.3.2.9 Name einlesen

**Beschreib** Der Name des Medikamentes muss eingelesen werden, damit klar ist, um welches Medikament es sich handelt.

**Via Code einlesen** **Beschreibung:**

Der Name ist direkt im Code hinterlegt und muss dann nur noch durch den Scanner eingelesen werden.

**Vorteil:**

Es braucht keine Eingabe des Namens durch den Benutzer. Somit können auch der Scanner und das Display einfacher gestaltet werden. (es braucht kein Eingabefenster/-tastatur)

**Nachteil:**

Man muss auf die hinterlegten Daten der Hersteller zugreifen können.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

**Auf Scanner eingeben** **Beschreibung:**

Der Name wird für jedes Medikament einzeln über eine Tastatur am Scanner eingegeben.

**Vorteil:**

Es ist kein Zugriff auf die Herstellerdatenbank nötig.

**Nachteil:**

Der Vorteil der Verwendung eines Scanners (schnell Daten einscannen) geht durch die manuelle Eingabe verloren. Der Scanner muss eine Tastatur oder Eingabemöglichkeit haben.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Auf Display eingeben** **Beschreibung:**

Der Name wird über ein Eingabe- oder Auswahlfenster direkt im Display eingegeben.

**Vorteil:**

Es ist kein Zugriff auf die Herstellerdatenbank nötig.

**Nachteil:**

Der Vorteil der Verwendung eines Scanners (schnell Daten einscannen) geht durch die manuelle Eingabe verloren. Die Bedienoberfläche des Displays muss zusätzlich mit der Eingabe des Ablaufdatums erweitert werden.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

### 3.3.2.10 Ablaufdatum einlesen

**Beschreibung** Um zu erfassen, wann ein Medikament abläuft, muss das Ablaufdatum jedes einzelnen Medikamentes eingelesen werden.

**Via Code einlesen** **Beschreibung:**

Das Datum kann direkt im Code hinterlegt werden und muss dann nur noch durch den Scanner eingelesen werden.

**Vorteil:**

Es braucht keine Eingabe des Ablaufdatums durch den Benutzer. Somit können auch der Scanner und das Display einfacher gestaltet werden. (es braucht kein Eingabefenster/-tastatur)

**Nachteil:**

Es muss ein Code verwendet werden, der das Datum speichern kann. Zudem muss man auf die hinterlegten Daten der Hersteller zugreifen können.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt

**Auf Scanner eingeben** **Beschreibung:**

Über eine Tastatur auf dem Scanner kann das Ablaufdatum für jedes Medikament einzeln manuell eingegeben werden.

**Vorteil:**

Es ist kein Zugriff auf die Herstellerdatenbank nötig.

**Nachteil:**

Der Vorteil der Verwendung eines Scanners (schnell Daten einscannen) geht durch die manuelle Eingabe verloren. Der Scanner muss eine Tastatur oder Eingabemöglichkeit haben.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Via Display**

**Beschreibung:**

Das Ablaufdatum kann auch über eine Eingabe- oder Auswahlfenster direkt im Display eingegeben werden.

**Vorteil:**

Es ist kein Zugriff auf die Herstellerdatenbank nötig.

**Nachteil:**

Der Vorteil der Verwendung eines Scanners (schnell Daten einscannen) geht durch die manuelle Eingabe verloren. Die Bedienoberfläche des Displays müsste zusätzlich mit der Eingabe des Ablaufdatums erweitert werden.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

### 3.3.2.11 Daten codieren

**Beschreibung** Die Medikamentenverpackungen sind mit verschiedenen Codes ausgestattet. Somit gibt es verschiedene Möglichkeiten, welchen Code für das Einlesen der Daten verwendet werden soll.

**DataMatrix** **Beschreibung:**  
Die DataMatrix ist ein speziell für Medikamente entwickelter Code, der sowohl den Namen als auch das Ablaufdatum und die Batchnummer speichern kann.

**Vorteil:**

Im Code sind der Name und das Ablaufdatum speicherbar. Zudem sollen ab dem Jahr 2016 alle Verpackungen standardmässig mit diesem Code ausgestattet werden. Da der Code ein 2D Code ist, kann ein normaler Scanner verwendet werden.

**Nachteil:**

Der DataMatrix Code wird erst ab dem Jahr 2016 standardmässig eingeführt. Zudem muss der Zugriff auf die Daten, die der Hersteller hinterlegt hat, gewährleistet sein.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

**Barcode** **Beschreibung:**  
Der Barcode sieht man heute fast auf allen Produkten beim Einkaufen, aber auch auf Medikamenten. Mit dem Zahlencode der hinterlegt ist, kann man über eine Datenbank diverse Informationen über das Produkt erfassen. (den Namen, den Verkaufspreis,..)

**Vorteil:**

Der Barcode ist heute auf fast allen Medikamenten enthalten. Zudem kann für das Einlesen des Barcodes ein Standardscanner verwendet werden.

**Nachteil:**

In den aufgedruckten Barcodes auf den Medikamenten ist nur der Name des Medikamentes hinterlegt. Da das Ablaufdatum immer ändert, aber der Barcode für das gleiche Medikament immer gleich bleibt, ist es nicht möglich, das Ablaufdatum zu hinterlegen.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

**OCR (Text)**

**Beschreibung:**

Das Einscannen von Daten geht nicht nur über Codes, sondern die Daten können auch aus gewöhnlichen Texten ausgelesen werden.

**Vorteil:**

Auf jeder Verpackung sind das Ablaufdatum und der Name des Medikamentes in Form eines Textes abgedruckt.

**Nachteil:**

Zum einen braucht es einen speziellen, aufwendigen Scanner und zum anderen ist es nicht ganz einfach ein Programm zu entwickeln, welches die richtigen Daten aus dem Text herausliest.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Selber Labeln**

**Beschreibung:**

Es wäre möglich, selbst die Information im Code zu hinterlegen und jedes Medikament bei der Ankunft damit zu bestücken.

**Vorteil:**

Bei den selbst entworfenen Codes kann das Ablaufdatum und der Name des Medikamentes hinterlegt werden. Zudem kann ein Code verwendet werden, der einfach zu scannen ist (z.B. DataMatrix)

**Nachteil:**

Es muss eine Labeling Maschine vorhanden sein. Ausserdem braucht es eine eigene Datenbank. Das Ganze ist sehr zeitaufwendig und somit geht der Vorteil der Verwendung eines Scann Systems komplett verloren.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**3.3.2.12 Anzahl Medikamente einlesen**

**Beschreibung**

Werden an die Basis Medikamente geliefert, kommen immer vom gleichen Medikament mehrere Ampullen ins Lager. Diese müssen alle durch das System erfasst werden, damit der Bestand der Medikamente stimmt.

**Jedes Medikament  
einzeln scannen**

**Beschreibung:**

Kommt eine Verpackung mit mehreren Ampullen an die Basis, werden die Ampullen ausgepackt und jede Ampulle einzeln eingescannt.

**Vorteil:**

Der Scanner und das Display müssen kein Eingabefenster haben. Zudem muss der Benutzer die Anzahl nicht eingeben.

**Nachteil:**

Dadurch, dass jede Ampulle separat eingescannt werden muss, braucht es viel mehr Zeit, als wenn man die Verpackung einscannet und die Anzahl manuell eingibt.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

Auf Display eingeben

**Beschreibung:**

Eine Verpackung mit mehreren Ampullen wird eingescannt und die Gesamtanzahl direkt auf dem Display eingegeben.

**Vorteil:**

Dadurch, dass nur eine Ampulle oder Verpackung eingescannt werden muss, wird viel Zeit eingespart.

**Nachteil:**

Das Display muss ein Eingabefenster besitzen.

**Entscheid:**

Dieser Lösungsansatz wird weiter verfolgt.

Auf Scanner eingeben

**Beschreibung:**

Eine Verpackung mit mehreren Medikamenten wird eingescannt und die Gesamtanzahl direkt über den Scanner eingegeben.

**Vorteil:**

Dadurch, dass nur eine Ampulle oder Verpackung eingescannt werden muss, wird viel Zeit eingespart.

**Nachteil:**

Der Scanner muss eine Tastatur oder ein Eingabefenster besitzen.

**Entscheid:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

### 3.3.2.13 Medikament zuweisen

**Beschreibung**

Dem Benutzer muss nicht nur mitgeteilt werden, wann ein Medikament abläuft sondern es muss auch ersichtlich sein, wo sich das Medikament befindet. (Lager, Medikamentenkoffer)

Auf Scanner eingeben

**Beschreibung:**

Mit der Tastatur/dem Display auf dem Scanner kann man auswählen wo sich das Medikament befindet.

**Vorteil:**

Beim Einscannen hat man den Scanner schon in der Hand und man braucht somit kein weiteres Gerät.

**Nachteil:**

Es braucht eine Tastatur oder ein Display auf dem Scanner.

**Entscheidung:**

Dieser Lösungsansatz wird nicht weiter verfolgt.

**Auf Display eingeben**

**Beschreibung:**

Man kann den Verbleib des Medikamentes auf dem Display eingeben.

**Vorteil:**

Das Display ist für Eingaben geeignet. Man kann ein einfaches Auswahlfenster machen.

**Nachteil:**

Das Display muss beim Einscannen immer vor Ort sein. Vorteil kabelloser Scanner geht verloren.

**Entscheidung:**

Dieser Lösungsansatz wird weiter verfolgt.

**Einscannen**

**Beschreibung:**

Auch das Lager und die Medikamententaschen sind mit Codes ausgestattet. Für den Verbleib scannt man einfach diesen Code ein.

**Vorteil:**

Es braucht keine Eingabe des Benutzers. Es ist sehr schnell.

**Nachteil:**

Das Lager und die Medikamententaschen müssen gelabelt werden und sie müssen immer vorhanden sein beim Scannen.

**Entscheidung:**

Dieser Lösungsansatz wird weiter verfolgt.

### 3.3.3 Morphologischer Kasten und Vorentscheidung

---

**Beschreibung**

Aus den Lösungsansätzen der Teilfunktionen wurde der Morphologische Kasten erstellt. Da der Unterbau auf alle anderen Teilfunktionen angepasst werden kann, wurde für ihn ein eigener Morphologischer Kasten erstellt. Somit sind nun ein Morphologischer Kasten für das System und einer für den Unterbau entstanden.

### 3.3.3.1 Morphologischer Kasten des Unterbaus

#### Einleitung

Bei den Lösungsansätzen für den Unterbau wurde darauf geachtet, dass dieser einfach und mit Standardkomponenten herzustellen ist. Deshalb sind vor allem Blech- oder Profilösungen entstanden. Drei Unterbaulösungen wurden genauer angeschaut und bewertet. Aus diesen wurde dann in einer Vorentscheidung die für uns beste Lösung erkoren. Um das ganze etwas anschaulicher zu machen, bastelte das Team für alle drei Varianten ein massstäbliches Kartonmodell.

Alle drei Lösungen wurden am 5.10.2015 M. Reichenbach von der REGA präsentiert.

#### Morphologischer Kasten des Unterbaus

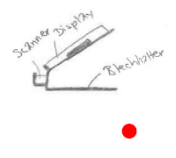
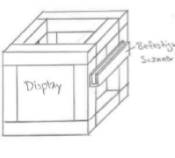
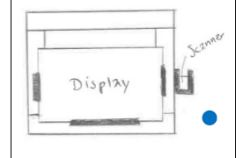
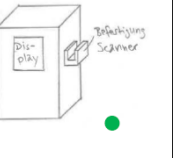
Baugruppen tragen	Blechkonstruktion	Profilkonstruktion	Profilrahmen	Box	Kein Unterbau
					

Abbildung 30: Morphologischer Kasten Unterbau

#### 3.3.3.1.1 Variante Blechkonstruktion (rot)

#### Beschreibung

Die Blechkonstruktion ist einfach herzustellen und braucht von allen Varianten am wenigsten Platz. Es sieht auch designetechnisch am besten aus und würde optimal in die Umgebung der REGA passen. Durch die kleine Geometrie und das geringe Gewicht, kann die Blechkonstruktion einfach transportiert werden. Dies garantiert ein mobiler Arbeitsplatz. Den Scanner kann ebenfalls an der Blechkonstruktion befestigt werden.

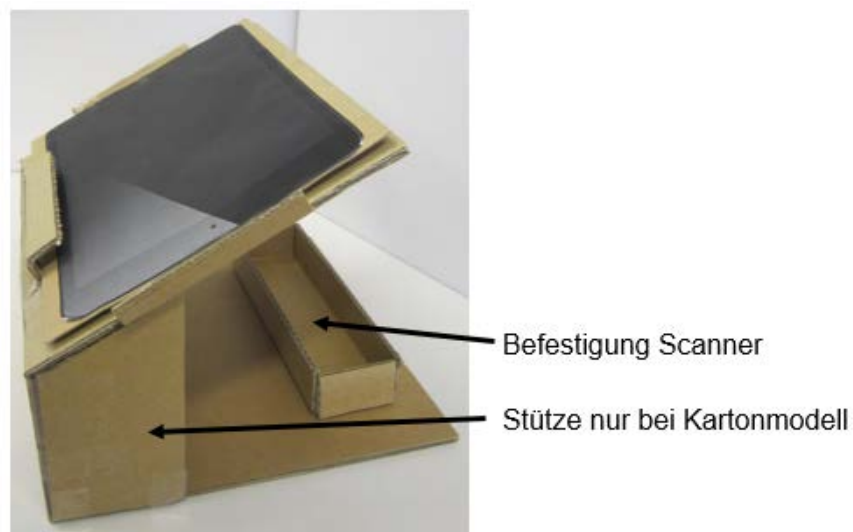


Abbildung 31: Kartonmodell Blechkonstruktion



**Beurteilung  
 (Swot- Analyse)**

<b>Stärken:</b> <ul style="list-style-type: none"> <li>• Platzsparend</li> <li>• Design</li> <li>• Display ist schräg befestigt (angenehmer zum bedienen)</li> <li>• Einfacher Aufbau der Scanstation (1 Blechteil)</li> </ul>	<b>Schwächen:</b> <ul style="list-style-type: none"> <li>• Offenes System (Stromanschluss sichtbar)</li> </ul>
<b>Chancen:</b> <ul style="list-style-type: none"> <li>• Mobile Scanstation</li> </ul>	<b>Risiken:</b> <ul style="list-style-type: none"> <li>• Scharfe Blechkanten</li> </ul>

Tabelle 9: Swot- Analyse Variante rot, Unterbau

3.3.3.1.2 Variante Profilrahmen (blau)

**Beschreibung**

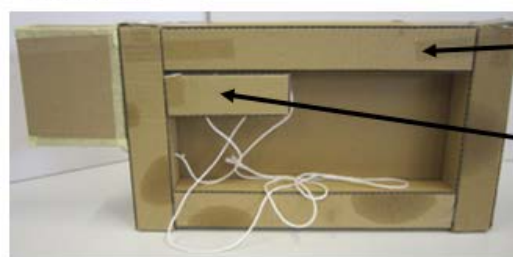
Aus vier Rechteckprofilen wird ein Rahmen zusammengebaut. Dieser wird auf einer Seite mit einer Blechabdeckung abgeschlossen. Auf dieser Blechabdeckung kann das Display befestigt werden. Hinter der Blechabdeckung hat es genügend Platz, um die Stromversorgung unterzubringen. Der ganze Rahmen wird an die Wand gehängt. Somit ist das System sehr platzsparend. Den Scanner kann auf der Aussenseite des Rahmens befestigt werden.

Von vorne:



Stationärer Scanner

Von hinten:



Profilrahmen mit Blechabdeckung

Steckdose

Abbildung 32: Kartonmodell Profilrahmen

**Beurteilung  
 (Swot- Analyse)**

<b>Stärken:</b> <ul style="list-style-type: none"> <li>• platzsparend</li> </ul>	<b>Schwächen:</b> <ul style="list-style-type: none"> <li>• Display ist parallel zur Wand befestigt (Es kann mehr reflektieren, nicht ergonomisch)</li> <li>• Fixe Scanstation</li> </ul>
<b>Chancen:</b> <ul style="list-style-type: none"> <li>• Befestigung an der Wand</li> </ul>	<b>Risiken:</b> <ul style="list-style-type: none"> <li>• Befestigung an der Wand, so dass alle Mitarbeiter es angenehm bedienen können (auf welcher Höhe)</li> </ul>

Tabelle 10: Swot- Analyse Variante blau, Unterbau

3.3.3.1.3 Variante Box (grün)

**Beschreibung**

Der Unterbau besteht aus einer Box, die aus Blechen zusammengeschnitten oder zusammengebaut ist. Aussen an der Box hat es Platz für das Display und den Scanner. Die Box besitzt eine Türe, damit man auf das Innere der Box Zugriff hat. Dort kann man die Stromversorgung unterbringen.

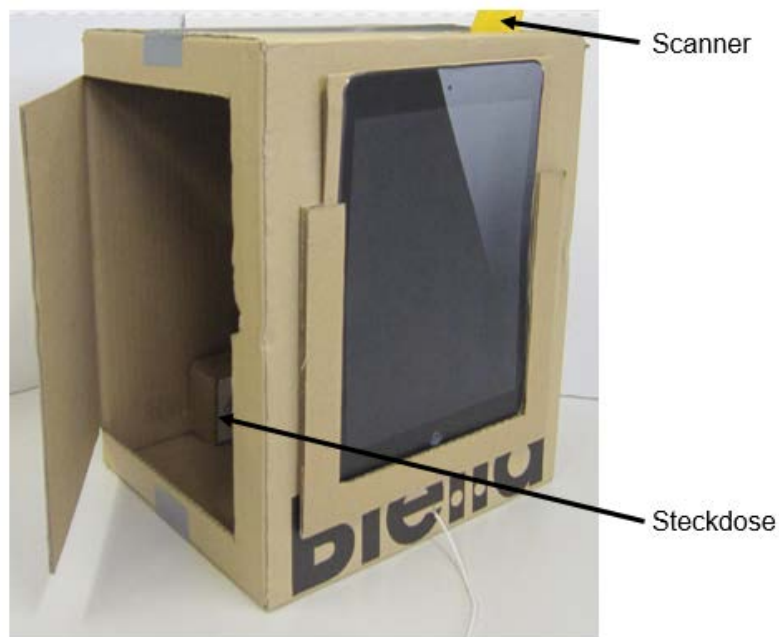


Abbildung 33: Kartonmodell Box

**Beurteilung  
(Swot- Analyse)**

<p><b>Stärken:</b></p> <ul style="list-style-type: none"> <li>• Alles in einer Box untergebracht</li> </ul>	<p><b>Schwächen:</b></p> <ul style="list-style-type: none"> <li>• Display ist parallel zur Wand befestigt (Es kann mehr reflektieren, unangenehm zum bedienen)</li> <li>• Braucht viel Platz</li> <li>• Aufbau der Scanstation aufwendig (viele Teile)</li> </ul>
<p><b>Chancen:</b></p> <ul style="list-style-type: none"> <li>• Kompaktes Design</li> <li>• Keine Kabel sichtbar</li> </ul>	<p><b>Risiken:</b></p> <ul style="list-style-type: none"> <li>• Durch grosse Geometrie eher hohes Gewicht (nicht mobil)</li> </ul>

Tabelle 11: Swot- Analyse Variante grün, Unterbau

3.3.3.1.4 Vorentscheidung

**Beschreibung** Die Vorentscheidung wurde mit einer Nutzwertanalyse getroffen. Dazu wurde die Vorlage der Maschinenteknik verwendet.

**Kriterien und Gewichtung** Die Entscheidungskriterien entsprechen den Optimierungskriterien aus dem Pflichtenheft. Es wurden aber nur die Kriterien genommen, die für den Unterbau entscheidend sind. Die einzelnen Kriterien wurden miteinander verglichen und gewichtet (von 1-10).

Kriterien	Gewichtung
T01 Einfache Bedienoberfläche	10.0
T04 Einfache Inbetriebnahme und Betrieb (durch Mitarbeiter)	10.0
T05 Einfache Fertigung (Standardverfahren bevorzugen)	3.8
T07 Hygiene muss gewährleistet sein (Desinfektionsmittelbeständig)	3.1
T08 Kleines, kompaktes System (max. 1m x 1m x1m)	7.7
K01 Kostengünstig	5.4
K02 Standardkomponenten verwenden	1.1

Tabelle 12: Kriterien und Gewichtung Entscheidung Unterbau

Wie aus der Tabelle zu entnehmen ist, haben wir die „einfache Bedienoberfläche“ und die „einfache Inbetriebnahme und Betrieb“ mit 10 am höchsten Gewichtet. Dies ist so, weil die Bedienung und der Betrieb durch den Mitarbeiter etwas alltägliches sein wird, und dies deshalb so angenehm wie möglich sein sollte.

**Entscheidung** Die Nutzwertanalyse sieht wie folgt aus:

Nr	Optimierungskriterien	G	Variante Rot		Variante Grün		Variante Blau	
			E	G x E	E	G x E	E	G x E
TO1	Einfache Bedienoberfläche (ohne Schulung)	10.0	10	100	7	70	5	50
TO4	Einfache Inbetriebnahme und Betrieb (durch Mitarbeiter)	10.0	10	100	7	70	5	50
TO5	Einfache Fertigung (Standardverfahren bevorzugen)	3.8	8	31	4	15	6	23
T07	Hygiene muss gewährleistet sein (Desinfektionsmittel)	3.1	5	15	5	15	5	15
T08	Kleines, kompaktes System (max. 1m x 1m x 1m)	7.7	10	77	4	31	7	54
KO1	Kostengünstig	5.4	8	43	4	22	7	38
K02	Standardkomponenten verwenden	3.1	5	15	5	15	5	15
		0.0		0		0		0
		0.0		0		0		0
		0.0		0		0		0
		0.0		0		0		0
		0.0		0		0		0
<b>Max Punktzahl</b>			<b>381</b>		<b>238</b>		<b>245</b>	

Tabelle 13: Entscheidung Unterbau

Aus der Nutzwertanalyse geht hervor, dass die Blechkonstruktion (rot) mit Abstand am besten für diese Aufgabe geeignet ist.

Dies wurde der REGA am 5.10.2015 auch so präsentiert.

### 3.3.3.2 Morphologischer Kasten des Systems

**Einleitung**

Der Morphologische Kasten des Systems beinhaltet alle anderen Teilfunktionen und deren Lösungsansätze. Bei der Bildung der drei Varianten wurden die ersten Wünsche der REGA schon berücksichtigt. Deshalb nutzen alle Varianten das iPad um die Daten anzuzeigen. Auch hier wurden wieder alle drei Varianten genauer angeschaut und bewertet und daraus eine Vorentscheidung getroffen.

Alle drei Lösungen wurden am 5.10.2015 M. Reichenbach von der REGA präsentiert.

## Morphologischer Kasten System





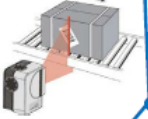





















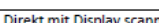





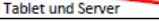


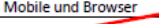


Daten codieren	Datamatrix 	Barcode 	Text 	Selber Label	
Daten einlesen	Handscanner 	Scanner stationär 	Kamera auf Gerät 	Scanprogramm auf Gerät 	
Daten anzeigen	Ipad 	Pc 	Smartphone 	Touchscreen 	Display auf Scanner 
Ablaufdatum einlesen	Via Code einlesen 	auf Scanner eingeben 	auf Display eingeben 		
Anzahl Medikamente einlesen	Jedes Medikament einzeln scannen 	auf Scanner eingeben 	auf Display eingeben 		
Name einlesen	Via Code einlesen 	auf Scanner eingeben 	auf Display eingeben 		
Medikament zuweisen	Codierung auf Medikamentenkoffer/ Lagerschublade. Codierung einscannen 	auf Scanner eingeben 	auf Display eingeben 		
Daten senden	Kabel 	bluetooth 	Direkt mit Display scannen (Keine Datenübertragung, nur Daten auslesen) 		
Ablauffrist melden	SMS 	Mail 	Meldung im Programm 		
Daten speichern	Server 	Tablet 	Tablet und Server 		
Daten anzeigen	Browser 	Mobile Applikation 	Mobile und Browser 		
Baugruppen laden	Ohne Stromversorgung für Scanner/ Display 	Mit Stromversorgung für Scanner/ Display 			

Abbildung 34: Morphologischer Kasten System

### 3.3.3.2.1 Variante Rot

**Beschreibung** Bei der roten Variante wird die DataMatrix eingescannt. Somit können der Name und das Ablaufdatum direkt aus dem Code entnommen werden. Die Daten werden dabei mit einem Handscanner eingescannt und via Bluetooth an das iPad (Display) gesendet. Falls vom gleichen Medikament mehrere eingescannt werden müssen, kann die Anzahl direkt auf dem iPad in der Applikation eingegeben werden. So muss nur ein Medikament gescannt werden. Die Zuweisung der Medikamente an das entsprechende Rettungsmodul kann ebenfalls manuell auf dem iPad eingegeben werden. Falls eine Medikamentenfrist abläuft, erhalten definierte Mitarbeiter eine Mail. Eine Stromversorgung um das Laden des iPads oder des Scanners zu gewährleisten, ist nicht vorgesehen. Die Applikation läuft auf dem iPad, ist allerdings vom mobilen Betriebssystem unabhängig. Die Daten werden lokal auf jedem Gerät gespeichert und auch auf einem Webserver. Die lokalen Daten werden regelmässig mit den Daten auf dem Webserver synchronisiert.

**Beurteilung  
(Swot- Analyse)**

<p><b>Stärken:</b></p> <ul style="list-style-type: none"> <li>• Software ist plattformunabhängig</li> <li>• Offline Nutzung möglich</li> <li>• Bearbeitung/Einsicht auch von PC Station möglich</li> <li>• Sichere Datenspeicherung, da Daten auf Server und lokal auf Gerät gespeichert.</li> </ul>	<p><b>Schwächen:</b></p> <ul style="list-style-type: none"> <li>• Keine Stromversorgung für Scanner/ iPad vorhanden</li> <li>• Manuelle Zuweisung der Medikamente</li> </ul>
<p><b>Chancen:</b></p> <ul style="list-style-type: none"> <li>• Auch für andere medizinische Geräte geeignet</li> <li>• Nutzung von DataMatrix ist zukunftsorientiert</li> </ul>	<p><b>Risiken:</b></p> <ul style="list-style-type: none"> <li>• DataMatrix wird trotz Ankündigung nicht eingeführt</li> </ul>

Tabelle 14: Swot- Analyse Variante Rot, System

### 3.3.3.2.2 Variante blau

**Beschreibung** Der Name des Medikamentes wird bei der blauen Variante mit dem Barcode eingelesen. Da der Barcode verwendet wird, muss das Ablaufdatum manuell auf dem iPad eingegeben werden. Die Daten werden mit einem stationären Scanner eingescannt. Auch hier kann bei mehreren gleichen Medikamenten die Anzahl wieder auf dem iPad eingegeben werden. Die Medikamentenzuweisung erfolgt ebenfalls manuell über das iPad. Da der Scanner stationär ist, bietet es sich an, die Daten via Kabel an das iPad zu senden. Eine Stromversorgung für den Scanner und das iPad ist vorgesehen, damit der stationäre Scanner für das Laden nicht immer demontiert werden muss. Falls eine Medikamentenfrist abläuft, werden die entsprechenden Mitarbeiter durch eine SMS informiert. Es wird eine Mobile Applikation für das iPad entwickelt. Diese läuft lokal auf jedem Gerät. Auch die Daten werden lokal auf jedem Gerät gespeichert. Es findet keine Kommunikation zwischen den iPads statt.

**Beurteilung  
(Swot- Analyse)**

<p><b>Stärken:</b></p> <ul style="list-style-type: none"> <li>• Stromversorgung für Scanner/ iPad ist vorhanden</li> <li>• Kabelübertragung ist weniger störungsanfällig</li> <li>• Offline Nutzung möglich</li> </ul>	<p><b>Schwächen:</b></p> <ul style="list-style-type: none"> <li>• Scanner ist stationär</li> <li>• SMS hat beschränkte Zeichenanzahl (dadurch weniger Informationsgehalt)</li> <li>• Manuelle Datumseingabe (fehleranfällig)</li> <li>• Barcode hat wenig Informationsgehalt</li> <li>• Bedienung der Applikation nur über Tablet möglich</li> <li>• Kein Basisübergreifender Zugriff möglich, da alle Daten lokal auf dem iPad gespeichert sind</li> </ul>
<p><b>Chancen:</b></p> <ul style="list-style-type: none"> <li>• Kabelübertragung ist schneller</li> </ul>	<p><b>Risiken:</b></p> <ul style="list-style-type: none"> <li>• Wechsel des Tablet Herstellers nicht möglich (Applikation läuft z.B. nicht auf Android)</li> <li>• Barcode wird abgelöst</li> </ul>

Tabelle 15: Swot- Analyse Variante Blau, System

3.3.3.2.3 Variante grün

**Beschreibung**

Auch bei der grünen Variante werden der Name und das Ablaufdatum des Medikamentes über die DataMatrix erfasst. Ein Handscanner sendet die Daten via Bluetooth an das iPad. Die Anzahl der zu erfassenden Medikamente kann auch hier wieder im iPad eingegeben werden. Das Zuweisen der Medikamente erfolgt durch einen Scanvorgang. Auf allen Rettungsmodulen befindet sich ein eindeutiger Code, welcher eingescannt wird, und so die Medikamente diesen Rettungsmodulen zugewiesen werden. Rückt das Ablaufdatum eines Medikamentes näher, werden die Mitarbeiter durch eine Mail benachrichtigt. Die Stromversorgung für das iPad und den Scanner sind vorgesehen. Die Applikation wird als Webanwendung umgesetzt. Auf sie wird über einen internetfähigen Browser (z.B. Safari) zugegriffen. Die Daten werden auf einem Webserver zentral gespeichert.

**Beurteilung  
(Swot- Analyse)**

<p><b>Stärken:</b></p> <ul style="list-style-type: none"> <li>• Stromversorgung für Scanner/ iPad ist vorhanden</li> <li>• Medikamentenzuweisung durch einscannen (ohne manuelle Benutzereingabe, weniger fehleranfällig)</li> </ul>	<p><b>Schwächen:</b></p> <ul style="list-style-type: none"> <li>• Labeling der Rettungsmodule</li> </ul>
<p><b>Chancen:</b></p> <ul style="list-style-type: none"> <li>• Auch für andere medizinische Geräte geeignet</li> <li>• Nutzung von DataMatrix ist zukunftsorientiert</li> </ul>	<p><b>Risiken:</b></p> <ul style="list-style-type: none"> <li>• Applikation bei fehlender Internetverbindung nicht benutzbar</li> <li>• DataMatrix wird trotz Ankündigung nicht eingeführt</li> </ul>

Tabelle 16: Swot- Analyse Variante grün, System

### 3.3.3.2.4 Vorentscheidung

**Beschreibung** Die Vorentscheidung wurde wieder mit Hilfe der Nutzwertanalyse getroffen.

**Kriterien und Gewichtung** Die Entscheidungskriterien entsprechen auch hier den Optimierungskriterien aus dem Pflichtenheft. Es wurden aber nur die Kriterien genommen, die für das System entscheidend sind. Die einzelnen Kriterien wurden miteinander verglichen und gewichtet (von 1-10).

Kriterien	Gewichtung
T01 Einfache Bedienoberfläche	6.9
T02 System zeigt die Bestände der Medikamente an	8.1
T03 System führt die Bestellung aus	3.1
T04 Einfache Inbetriebnahme und Betrieb (durch Mitarbeiter)	6.3
T06 System läuft auf einem iPad	5.0
T07 Hygiene muss gewährleistet sein (Desinfektionsmittelbeständig)	1.3
T09 System läuft zuverlässig (90%)	10.0
K01 Kostengünstig	4.4

Tabelle 17: Kriterien und Gewichtung Entscheidung System

Da sich die REGA auf dieses System verlassen muss, spielt die Zuverlässigkeit eine sehr wichtige Rolle. Deshalb wurde die Zuverlässigkeit am stärksten gewichtet. Eine weitere wichtige Aufgabe des Systems ist es, die Bestände der Medikamente anzeigen zu können. Deshalb wurde dieses Kriterium am zweit höchsten Gewichtet.



**Entscheidung** Die Nutzwertanalyse sieht wie folgt aus:

Nr	Optimierungskriterien	G	Variante Rot		Variante Grün		Variante Blau	
			E	G x E	E	G x E	E	G x E
TO1	Einfache Bedienoberfläche (ohne Schulung)	6.9	8	55	8	55	2	14
TO2	System zeigt die Bestände der Medikamente an	8.1	10	81	10	81	8	65
TO3	System führt die Bestellung aus	3.1	6	19	6	19	6	19
TO4	Einfache Inbetriebnahme und Betrieb (durch Mitarbe	6.3	5	31	8	50	3	19
T06	System läuft auf einem Ipad	5.0	10	50	10	50	10	50
T07	Hygiene muss gewährleistet sein (Desinfektionsmitt	1.3	5	6	5	6	5	6
T09	System läuft zuverlässig (90%)	10.0	10	100	4	40	7	70
KO1	Kostengünstig	4.4	5	22	8	35	5	22
		0.0		0		0		0
		0.0		0		0		0
		0.0		0		0		0
		0.0		0		0		0
<b>Max Punktzahl</b>				<b>364</b>		<b>336</b>		<b>265</b>

**Tabelle 18: Entscheidung System**

Aus der Nutzwertanalyse geht hervor, dass die Variante rot am besten geeignet ist. Hier war die Entscheidung zwischen rot und grün aber knapp. Dies wurde der REGA am 5.10.2015 auch so präsentiert.

### 3.4 Definitiver Variantenentscheid mit der REGA

**Beschreibung** Am 5. Oktober 2015 präsentierte unser Team die Morphologischen Kästen und die daraus getroffene Entscheidung Markus Reichenbach von der REGA. Der Entscheid für die weiterführende Variante wurde zusammen mit dem Auftraggeber getroffen.

**Morphologischer Kasten des Unterbaus** Beim Unterbau war er unserer Meinung und fand auch die Blechkonstruktion am besten.

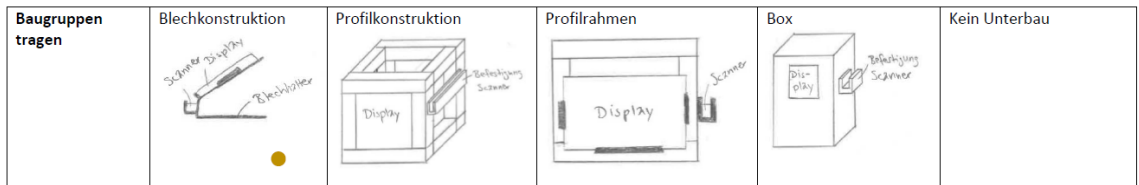




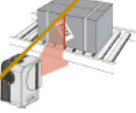

















Abbildung 35: Morphologischer Kasten mit definitiver Variante, Unterbau

**Morphologischer Kasten System** Auch beim System war er fast überall unserer Meinung. Jedoch wurde die rote Variante wie folgt angepasst:

- **Daten codieren:** Da auf einigen Medikamenten die DataMatrix vorhanden ist, aber auf anderen der Barcode, muss das System beide Codes einscannen können.
- **Daten einlesen:** Die Daten werden von einem Handscanner eingescannt.
- **Daten anzeigen:** Das iPad dient als Display
- **Ablaufdatum einlesen:** Da momentan sowohl beim Barcode als auch bei der DataMatrix nur der Name hinterlegt ist, muss für den Prototyp das Ablaufdatum noch manuell auf dem iPad eingegeben werden können. Später wäre das Ziel, dass alle Medikamente die DataMatrix besitzen und dort der Name und das Ablaufdatum hinterlegt sind und diese eingescannt werden.
- **Anzahl Medikamente einlesen:** Falls mehrere gleiche Medikamente eingescannt werden müssen, kann die Anzahl auf dem iPad eingeben werden.
- **Name einlesen:** Der Name wird durch das Scannen des Codes eingelesen. Dafür muss eine Datenbank aufgebaut werden, welche den Code auf den Medikamentennamen mappt.
- **Medikament zuweisen:** Das Medikament wird mit einer Drop down Liste auf dem iPad dem Rettungsmodul zugewiesen.
- **Daten senden:** Die Daten werden via Bluetooth vom Scanner an das iPad gesendet
- **Ablauffrist melden:** Falls ein Medikament abläuft, werden die entsprechenden Mitarbeiter per Mail informiert. Zusätzlich wird diese Information auch im Programm angezeigt. Eine Übersicht über die Medikamente die in naher Zukunft ablaufen wäre wünschenswert.
- **Daten speichern:** Die Daten werden zentral auf einem Server gespeichert. Zudem werden sie auch lokal auf dem Tablet zwischengespeichert und mit dem Server synchronisiert.
- **Daten anzeigen:** Die Medikamenteneinlagerung und -Zuweisung werden auf dem Tablet in der Mobilien Applikation durchgeführt. Die Teile Medikamente verwalten und System verwalten werden via Webbrowser durchführbar sein.
- **Stromversorgung:** Für den Scanner wird eine Stromversorgung im Unterbau zwingend gewünscht. Eine Stromversorgung für das iPad ist momentan nicht nötig, sollte aber optional möglich sein.

Somit sieht der Morphologische Kasten des Systems mit der definitiven Variante wie folgt aus:

Daten codieren	Datamatrix 	Barcode 	Text 	Selber Labeln	
Daten einlesen	Handscanner 	Scanner stationär 	Kamera auf Gerät 	Scanprogramm auf Gerät 	
Daten anzeigen	Ipad 	Pc 	Smartphone 	Touchscreen 	Display auf Scanner 
Ablaufdatum einlesen	Via Code einlesen	auf Scanner eingeben 	auf Display eingeben 		
Anzahl Medikamente einlesen	Jedes Medikament einzeln scannen	auf Scanner eingeben 	auf Display eingeben 		
Name einlesen	Via Code einlesen	auf Scanner eingeben 	auf Display eingeben 		
Medikament zuweisen	Codierung auf Medikamentenkoffer/Lagerschublade. Codierung einscannen	auf Scanner eingeben 	auf Display eingeben 		
Daten senden	Kabel	Bluetooth	Direkt mit Display scannen (Keine Datenübertragung, nur Daten auslesen)		
Ablauffrist melden	SMS	Mail	Meldung im Programm		
Daten speichern	Server	Tablet	Tablet und Server		
Daten anzeigen	Browser	Mobile Applikation	Mobile und Browser		
Baugruppen laden	Ohne Stromversorgung für Scanner/ Display	Mit Stromversorgung für Scanner/ Display			

**Abbildung 36: Morphologischer Kasten mit definitiver Variante, System**

## Anforderungen durch REGA

Bei dem Gespräch mit der REGA wurden noch andere Eckpunkte des Systems besprochen:

- Der Scanbetrieb muss eingeschränkt (Medikamente zuweisen und einlagern) auch im offline Modus möglich sein. Bei fehlender Internetverbindung werden die gescannten Daten auf dem Tablet zwischengespeichert. Sobald das Internet wieder zur Verfügung steht, sollen die gescannten Daten mit dem System (Daten auf Server) synchronisiert werden.
- Es gibt zwei Medikamente (Lysthenon und Roccuronium) die gekühlt gelagert werden müssen. Werden sie jedoch einem Rettungsmodul zugewiesen, schreibt der Mitarbeiter ein neues, verkürztes Ablaufdatum auf die Ampulle. Deshalb muss bei diesen Medikamenten eine manuelle Eingabe des Ablaufdatums möglich sein.
- Es wäre wünschenswert, wenn sich die Mitarbeiter mit ihrem REGA Login in das System einloggen könnten und kein neues Login bräuchten (mit AD koppeln). Muss aber abgeklärt werden (nice to have)
- Die Ablauffrist wird einerseits per Mail gemeldet und andererseits im Programm angezeigt. Per Mail werden aber mehrere Mitarbeiter benachrichtigt. Es ist wünschenswert, dass im Mail ein Link in die Applikation führt. Dort soll dann ersichtlich sein, welche Medikamente schon ausgetauscht wurden und welche noch ausgetauscht werden sollen (doppelte Arbeit vermeiden)
- Falls mehrere gleiche Medikamente mit verschiedenen Ablaufdaten in einem Rettungsmodul sind, muss beim Auswechseln des Medikamentes gefragt werden, welches Ablaufdatum das vorhandene Medikament im Rettungsmodul hat. Somit weiss das Programm, welches Medikament gebraucht wurde. → bereits so in bevorzugter Lösungsvariante vorgesehen
- Es kann für die Zukunft die DataMatrix verwendet werden. Damit der Prototyp aber getestet werden kann, wird der Barcode eingescannt. Somit muss beim Prototyp das Ablaufdatum noch manuell eingegeben werden. Zudem braucht es eine Datenbank, welche die Nummer auf dem Barcode dem Medikamentennamen zuweist.
- Eine Meldung (ähnlich dem Ablauf eines Medikaments) für Unterschreiten der Lagermindestbestände eines Medikaments ist zurzeit noch out of Scope.

**Swot- Analyse**

	<p><b>Stärken:</b></p> <ul style="list-style-type: none"> <li>• Mobile Scanstation</li> <li>• Software ist plattformunabhängig</li> <li>• Offline Nutzung möglich</li> <li>• Bearbeitung/Einsicht auch von PC Station möglich</li> <li>• Stromversorgung für Scanner vorhanden</li> <li>• Wechsel des Tabletherstellers ohne grössere Softwareanpassungen möglich</li> <li>• Sichere Datenspeicherung, da Daten auf Server und lokal auf Gerät gespeichert.</li> </ul>	<p><b>Schwächen:</b></p> <ul style="list-style-type: none"> <li>• Manuelle Zuweisung der Medikamente</li> <li>• Manuelle Eingabe des Datums</li> </ul>
	<p><b>Chancen:</b></p> <ul style="list-style-type: none"> <li>• Auch für andere medizinische Geräte geeignet</li> <li>• Nutzung von DataMatrix ist zukunftsorientiert</li> </ul>	<p><b>Risiken:</b></p> <ul style="list-style-type: none"> <li>• DataMatrix wird trotz Ankündigung nicht eingeführt</li> <li>• Verkratzung iPad in Blechkonstruktion</li> <li>• Scharfe Blechkanten</li> </ul>

Tabelle 19: Swot- Analyse Endvariante, System

**Fazit**

Es ist eine Variante entstanden, mit der M. Reichenbach sehr zufrieden ist. Diese Variante hat viele Stärken, jedoch dürfen beim Design die Risiken nicht vergessen werden.

### 3.5 Fazit Analyse

**Fazit**

Durch die genaue Analyse der Aufgabe konnten viele Ideen gesammelt werden, die bei den Software- und Konstruktionsanforderungen mit einbezogen wurden.

Mit Hilfe des Morphologischen Kasten sind verschiedene gute Varianten entstanden, aus welchen eine Endvariante gebildet wurde. Diese Endvariante fand bei der REGA sehr guten Anklang.

In der nächsten Phase des Projektes wird nun diese Endvariante mit den Inputs der REGA designed. Somit ist gewährleistet, dass ein Produkt entsteht, das auch unserem Industriepartner gefällt.

## 4 Design

---

**Beschreibung** In der Phase Design wird die gewählte Variante aus der Phase Analyse so ausgearbeitet, dass sie am Ende realisiert werden kann.  
Beim Softwaredesign wird die Architektur der Software festgelegt und darauf aufbauend der Technologiestack definiert.  
Beim Konstruktionsdesign (Unterbau) wird der gesamte Unterbau konstruiert, berechnet und der Zubehör bestimmt. Somit kann danach in der Phase Realisierung der ganze Unterbau und der Zubehör bestellt werden.

### 4.1 Software

---

**Beschreibung** Die Software ist grob in zwei Teile aufgeteilt. Zum einen gibt es den Serverteil, auf welchem die Daten persistiert werden und der Zugriff auf diese implementiert ist und zum anderen den Clientteil, welcher das GUI und die Mehrheit der Businesslogik enthält.

#### 4.1.1 Application Architecture

---

**Beschreibung** In diesem Abschnitt werden die verschiedenen Ebenen und Komponenten der Architektur beschrieben. Die Architektur ist technologieneutral gehalten. Die verwendeten Technologien werden in einem weiteren Abschnitt beschrieben. Die Applikation ist in drei Teile aufgegliedert, zwei verschiedene Clients und einen Server Teil. Die Kommunikation zwischen Client und Server erfolgt über REST.

## Architektur

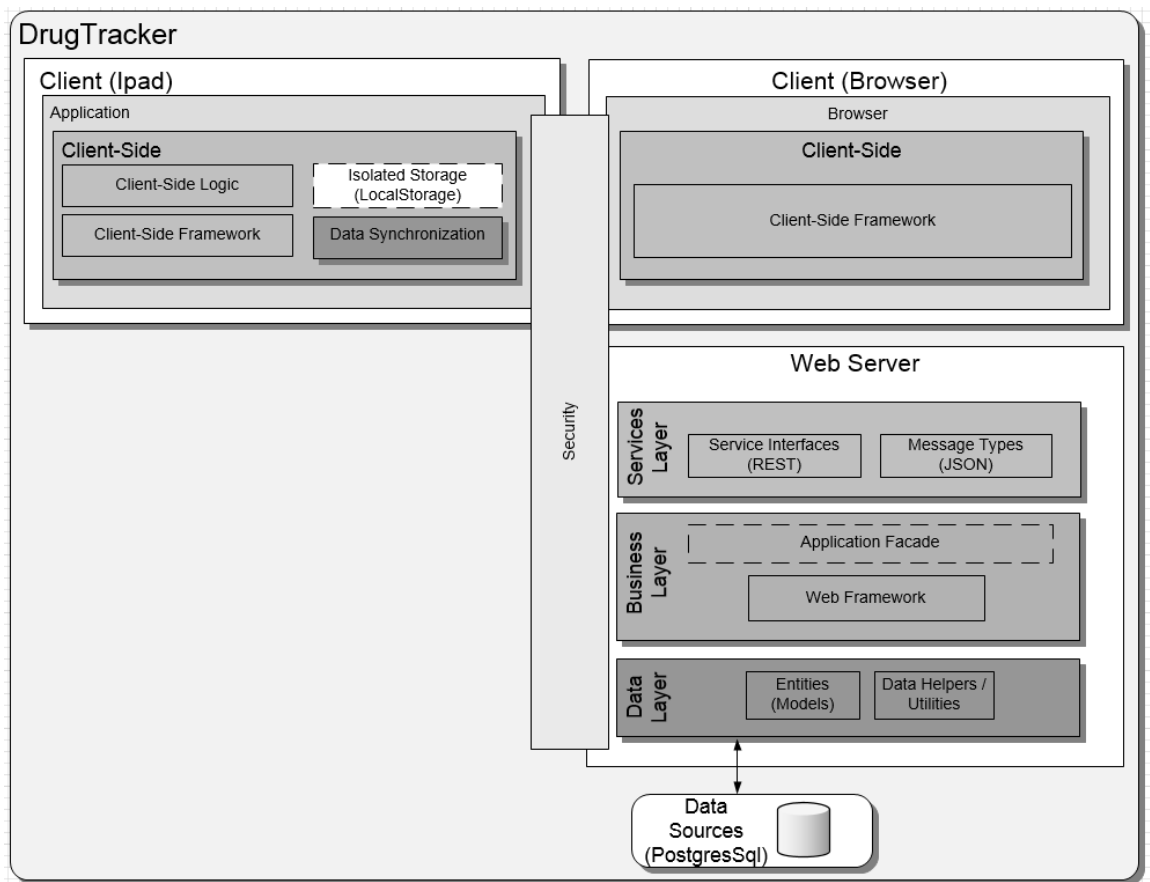


Abbildung 37: Application Architecture

### 4.1.1.1 Client (iPad)

#### Beschreibung

Der Hauptteil der Applikation wird auf einem iPad genutzt. Die Businesslogik (Client-Side Logic) für die Use Cases „Medikamente einlagern“ und „Medikamente zuweisen“ befindet sich auf dem iPad-Client. Die Applikation soll auch nutzbar sein, wenn keine Internetverbindung besteht (Offline Modus). Aus diesem Grund werden Medikamentendaten teilweise redundant zur Serverdatenbank in einen Isolated Storage auf dem iPad gespeichert. Sobald die Verbindung wieder steht, soll eine Synchronisation mit dem Backend vorgenommen werden. Damit wird sichergestellt, dass die Daten wieder konsistent sind. Im Offline Modus sind die Funktionen der Applikation eingeschränkt und reduzieren sich auf die Use Cases „Medikamente einlagern und zuweisen“. Im Offline Modus muss vom Benutzer sichergestellt werden, dass keine Medikamente ablaufen. Die Erkennung ablaufender Medikamente geschieht auf dem Server und diese kann vom Client nur über eine vorhandene Verbindung zum Server nachvollzogen werden. Auf dem Client werden UC1 (Benutzer anmelden), UC2 (Medikament einlagern), UC3 (Medikament zuweisen) umgesetzt.

### 4.1.1.2 Client (Browser)

#### Beschreibung

Zur Verwaltung von Medikamenten und Benutzern wird ein Webinterface zur Verfügung gestellt (UC5: Medikamente bearbeiten und UC6: System verwalten). Der Zugriff findet über einen Browser statt, welcher mit dem Backend kommuniziert. Aus diesem Grund befindet sich keine Business Logik auf dem Client (Browser).

### 4.1.1.3 Web Server

**Beschreibung** Der Webserver dient als Backend der Applikation. Der Webserver stellt Methoden zum Zugriff und zur Mutation von Daten auf der Serverdatenbank zur Verfügung. Auf dem Businesslayer des Webserver befindet sich auch die Logik für die Früherkennung von ablaufenden Medikamenten inklusive Mailversand.

### 4.1.2 Technologieevaluierung

**Beschreibung** Anhand verschiedener Entscheidungskriterien, werden in diesem Abschnitt die verwendeten Technologien ausgewählt.

**Erfahrungen** Die Studenten haben berufsbedingt schon Erfahrung mit Java und Javascript. Dennoch sollen neue Technologien nicht ausgelassen werden.

#### 4.1.2.1 Kriterien

**Entscheidungskriterien** Die folgende Tabelle definiert fünf Kriterien, welche zur Bewertung einer Technologie oder eines Frameworks jeweils mit 0-5 Sternen bewertet werden können. Die Spalte Gewichtung gibt an, wie wichtig das betreffende Kriterium in Bezug auf die Aufgabenstellung anzusehen ist.

Kriterium	Erläuterung	Gewichtung
Eignung	Wie gut eignet sich eine Technologie oder ein Framework für die Umsetzung der Architektur? Geschieht alles "hinter" dem Vorhang oder sind einzelne Komponenten einsehbar? Hohe Bewertung = Hohe Eignung	★★★★★
Testbarkeit	Wie gut können die mit dem Framework oder der Technologie erstellten Komponenten durch Unit Tests getestet werden? Hohe Bewertung = Hohe Testbarkeit	★★★
Produktreife	Wie gut hat sich das Framework oder die Technologie bis jetzt in der Realität beweisen können? Wie lange existiert es schon? Gibt es eine aktive Community und wird es aktiv weiterentwickelt? Hohe Bewertung = Hohe Produktreife	★★★
Ease of Use	Wie angenehm ist das initiale Erstellen, die Konfiguration und die Unterhaltung einer Applikation? Führt das Framework irgendwelchen "syntactic sugar" ein, um die Arbeit zu erleichtern? Hohe Bewertung = Hoher "Ease of use"-Faktor	★★★★★
Aktualität	Diese Arbeit befasst sich um "moderne Web-Applikationen". So sollte auch die zu verwendende Technologie gewissermassen nicht von "vorgestern" sein. Hohe Bewertung = Hohe Aktualität	★★

Tabelle 20: Entscheidungskriterien

Die Bewertung wird wie folgt vorgenommen:  
 (Bewertung \* Gewichtung)/Anzahl Kriterien = Gesamtbewertung



### 4.1.2.2 Mögliche Technologien

- Play** Ist ein Open Source Framework, welches in Scala und Java geschrieben ist, es folgt dem model-view-Controller Pattern. Es unterstützt hot-code reloading, welches die Produktivität des Entwicklers erhöht. Play verwendet keine J2EE Einschränkungen. Auch wenn Play Applikationen mit dem internen JBoss netty Web Server laufen, können sie als war File auf einen J2E Application Server deployed werden. [6]
- Spring** Das Spring Framework ist ein quelloffenes Framework für die Java Plattform. Ziel ist es die Entwicklung mit Java/J2E zu vereinfachen. Spring bietet mit einem breiten Spektrum an Funktionalität eine ganzheitliche Lösung zur Entwicklung von Anwendungen und deren Geschäftslogiken; dabei steht die Entkopplung der Applikationskomponenten im Vordergrund. [7]
- Xamarin** Mit der Programmiersprache C# können Entwickler iOS-, Android- und Windows-Apps schreiben. Der Quellcode kann für alle Plattformen gemeinsam benutzt werden. [8]
- Ionic** Ionic ist ein Open-Source-Framework zur Erstellung mobiler Web-Apps auf der Basis von HTML5. Der Fokus des Frameworks liegt auf Seiten des Front-Ends und damit bei den Nutzern. Es konzentriert sich auf Benutzerschnittstellen (UX), zur Verbesserung der intuitiven Bedienung und zur Vereinfachung der Usability der App. Dabei wird es vor allem in Verbindung mit AngularJS genutzt, wobei AngularJS die Struktur der App erstellt, während Ionic das Look&Feel, sowie die Performance der App realisiert. Ebenso unterstützt Ionic das hot-reloading. [9]
- J2EE** Java Platform, Enterprise Edition, abgekürzt Java EE oder früher J2EE, ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von in Java programmierten Anwendungen und insbesondere Web-Anwendungen. [10]

### 4.1.2.3 Auswertung und Entscheid

Auswertung	Gewichtung	Play	Spring	Xamarin	Ionic	J2EE
Eignung	★★★★★	5	4	4	4	3
Testbarkeit	★★★	4	3	5	5	4
Produktreife	★★★	4	4	4	4	5
Ease of Use	★★★★	5	4	4	5	3
Aktualität	★★	5	4	5	5	3
		<b>15.8</b>	13	14.6	<b>15.4</b>	12

Tabelle 21: Auswertung Varianten

**Entscheid** Aus der Auswertung erfolgt, dass Play und Ionic zu bevorzugen sind und für dieses Projekt verwendet wird.

---

### 4.1.3 Verwendete Technologien

---

**Beschreibung** Im folgenden Abschnitt werden die verwendeten Technologien beschrieben.

#### 4.1.3.1 Client (iPad)

**Beschreibung** Für die Entwicklung der Apple Mobile App wird auf ein Frontend-Framework zurückgegriffen. Dies vereinfacht die Entwicklung der Applikation.

**Ionic Framework** Das Ionic Framework ist im Kapitel 4.1.2.2 beschrieben.

#### 4.1.3.2 Client (Browser)

**Beschreibung** Für die Verwaltung des Systems und der Medikamente wird über einen Browser auf das System zugegriffen.

**Browser** Die Darstellung erfolgt mittels HTML5, CSS3 und JavaScript. Auf Seiten des Clients wird auf Business Logik verzichtet, diese wird vom Backend übernommen.

#### 4.1.3.3 Web-Server

**Play Framework** Play 2.0 ist ein hoch-produktives Java und Scala Web-Applikation-Framework, welches die Komponenten und APIs integriert, die für die Entwicklung von modernen Web-Applikationen benötigt werden.  
Das Framework ist stateless und völlig RESTful. Auch der modulare Aufbau vereinfacht die Entwicklung von Web-Applikationen. [11]

**JPA (EclipseLink)** Die Java Persistence API (JPA) ist eine Schnittstelle für Java-Anwendungen, die die Zuordnung und die Übertragung von Objekten zu Datenbankeinträgen vereinfacht. Sie vereinfacht die Lösung des Problems der objektrelationalen Abbildung, das darin besteht, Laufzeit-Objekte einer Java-Anwendung über eine einzelne Sitzung hinaus zu speichern (Persistenz), wobei relationale Datenbanken eingesetzt werden können, die ursprünglich nicht für objektorientierte Datenstrukturen vorgesehen sind. [12]  
Genutzt wird die JPA Referenzimplementation EclipseLink.

**PostgreSQL** Für die Datenhaltung wird das freie, objektrelationale Datenbankmanagementsystem (ORDBMS) eingesetzt.

#### 4.1.4 Entity-Relationship-Modell

**Beschreibung** Das folgende Entity-Relationship-Modell wurde aus dem in der Analyse erstellten Domain Model (Kapitel 3.2.3) abgeleitet und entspricht der Abbildung dessen auf Datenbankebene. Jede Entität entspricht dabei einer Datenbanktabelle.

**Entity-Relationship-Modell**

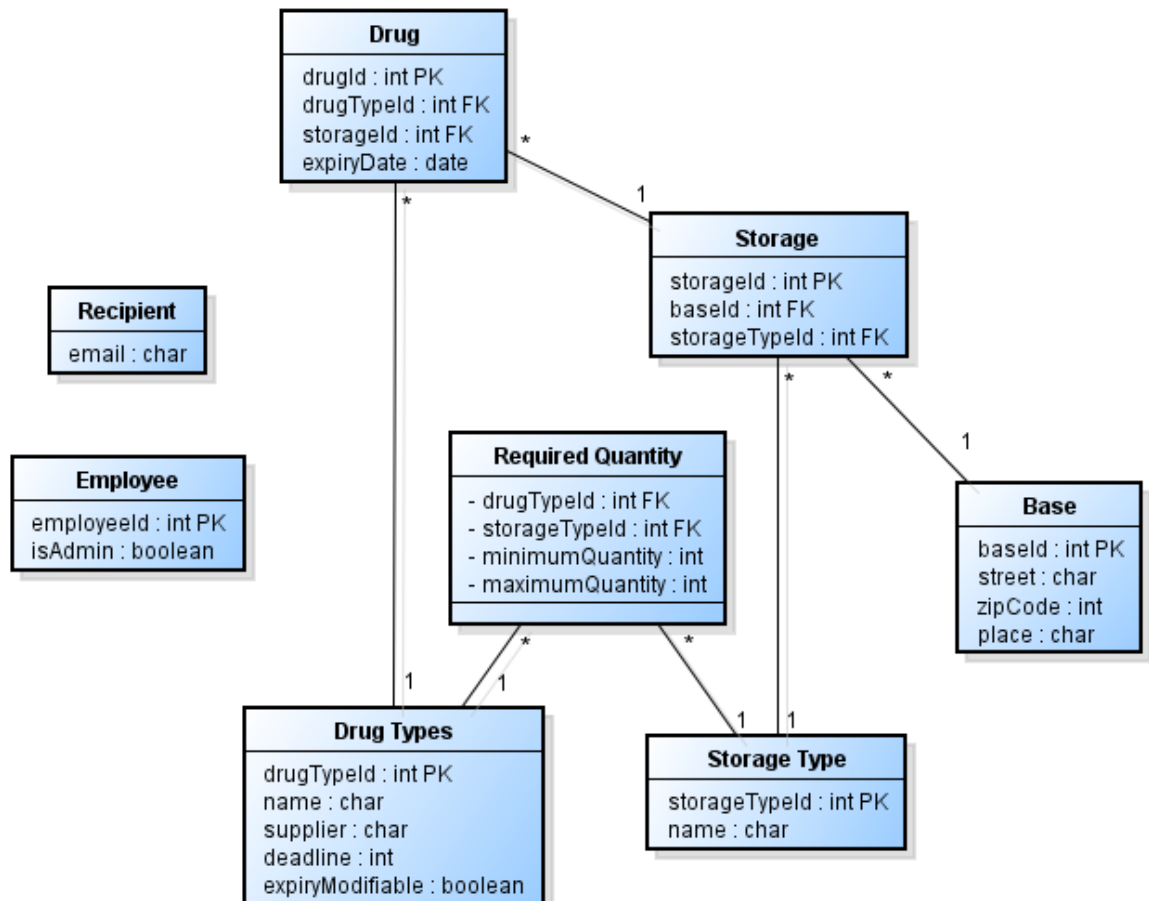


Abbildung 38: Entity-Relationship-Modell

**Entitäten/Tabellen** Die Beschreibung zu den einzelnen Entitäten/Tabellen können aus dem Domain Model entnommen werden. Lediglich die Entitäten *Required Quantity* und *Recipient* sind im Domain Model nicht vorhanden. *Required Quantity* beinhaltet die Information wie viele Exemplare eines Medikamententyps (Drug Type) pro Lagertyp (Storage Type) im Minimum und Maximum vorhanden sein müssen. Da es sich um eine n:n Beziehung handelt, war diese Zwischentabelle notwendig.

Die Tabelle Recipient definiert, wer (welche Mailadressen) über bald ablaufende oder bereits abgelaufene Medikamente informiert werden soll.

## 4.2 Unterbau

- Beschreibung** Für den Prototypen wird ein Unterbau konstruiert, der die Befestigung des iPads und des Scanners während und nach dem Gebrauch gewährleistet. Zudem kann im Unterbau auch direkt der Scanner und das iPad geladen werden.
- Die Konstruktion wird auf Basis von Berechnungen ausgeführt. Somit ist gewährleistet, dass der Unterbau für seinen Gebrauch geeignet ist.
- Für die Realisierung werden Detailzeichnungen erstellt und das passende Zubehör ausgesucht. Bevor der Unterbau in der Phase Realisierung gefertigt werden kann, ist es nötig, Offerten einzuholen.
- Der ganze Unterbau wurde für das iPad Air konstruiert, da die REGA dieses iPad verwendet. Es ist ausserdem möglich, das iPad Air 2 zu verwenden.

### 4.2.1 Basismodell

- Beschreibung** Für die Berechnungen braucht es ein erstes Modell des Unterbaus, das sogenannte Basismodell. Bei diesem Modell stimmen alle Abmasse, ausser die Länge der Grundfläche, da diese mit den Berechnungen ermittelt wird. Im Basismodell wird die Grundfläche bündig zur Deckfläche gemacht.

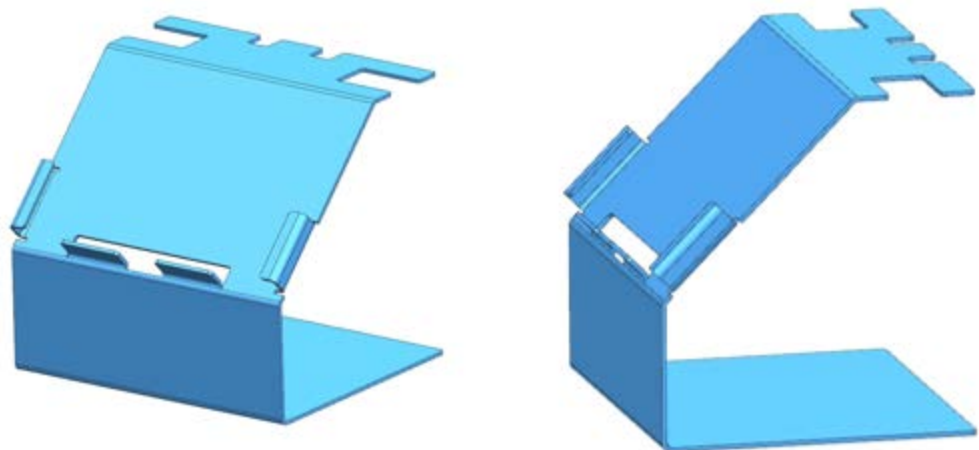


Abbildung 39: Basismodell

### 4.2.2 Berechnungen

- Beschreibung** Für die Berechnung der Länge der Grundfläche braucht es den Schwerpunkt und das Gewicht des Unterbaus. Deshalb werden in einem ersten Schritt das Gewicht und der Schwerpunkt des Basismodells berechnet und daraus die Länge der Grundfläche ermittelt.
- Die Grundfläche des Basismodells wird dementsprechend angepasst und das Modell in Endmodell umbenannt. Von diesem Endmodell wird nochmals der Schwerpunkt berechnet, aus dem danach die Sicherheit gegen das Kippen ermittelt werden kann. Zum Schluss wird noch die Haftreibung berechnet, damit man weiss, ob der Unterbau zuerst kippt oder rutscht.

#### 4.2.2.1 Schwerpunkt Ausgangslage

##### Vorgehen

Für die Berechnung des Schwerpunktes wird der ganze Unterbau in verschiedene Teilflächen unterteilt. Von jeder Teilfläche wird danach der Schwerpunkt einzeln berechnet und mit einer Tabelle aus den einzelnen Schwerpunkten der gesamte Schwerpunkt berechnet. Für die Berechnung wurde ein Excel programmiert, indem die Werte der einzelnen Flächen angepasst werden können.

*Berechnung ist im Excel im Anhang ersichtlich*

##### Koordinatensystem

Das Koordinatensystem wurde so gewählt, dass X in die Höhe, Y nach rechts und Z nach hinten zeigt. Der Ursprung liegt in X Richtung am Boden, in Y Richtung in der Mitte des Bauteils und in Z Richtung auf der Frontfläche.

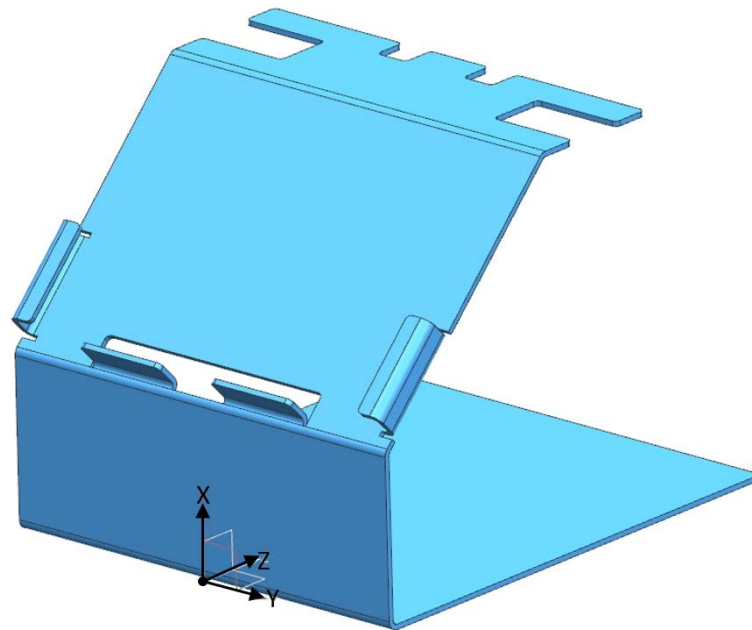


Abbildung 40: Koordinatensystem

##### Berücksichtigung

In der Schwerpunktberechnung wurden auch das iPad und der Moosgummi mit einbezogen. Die Radien, Bohrungen und die iPad Hülle wurden vernachlässigt. Der Scanner wurde ebenfalls nicht beachtet, da er nachher bei der Berechnung der Länge der Grundfläche miteinbezogen wird.

##### Modell

Für die Schwerpunktberechnung wird das Basismodell verwendet.

##### Ergebnis Handrechnung

Mit der Handrechnung im Excel wurde folgender Schwerpunkt berechnet:

Wert in X Richtung = 120.9mm

Wert in Y Richtung = 0mm

Wert in Z Richtung = 72.1mm

Gewicht: 1.6 kg

##### Ergebnis CAD

Der Schwerpunkt wurde auch mit dem CAD berechnet. Wobei aber nur der Unterbau und das iPad beachtet wurde. (Moosgummi wurde nicht beachtet)

Ergebnis Schwerpunktberechnung mit dem CAD:

Wert in X Richtung = 118mm  
Wert in Y Richtung = 0mm  
Wert in Z Richtung = 75mm

Vergleicht man die Werte aus der Handrechnung mit dem Ergebnis des CAD's fällt auf, dass die Werte in X Richtung und Z Richtung leicht voneinander abweichen. Dies ist so, weil beim CAD das Moosgummi von 2mm Dicke nicht berücksichtigt wurde. Somit ist die Handrechnung pausibler und es kann damit weiter gerechnet werden.

#### 4.2.2.2 Standsicherheit (Länge der Grundfläche)

##### Theorie

Wird das iPad auf dem Unterbau bedient, darf der Unterbau nicht wegkippen, er muss also Standsicher sein. Darum wird hier berechnet, wie lang die Grundfläche x minimal sein muss, damit das Bauteil sicher steht.

Für die Berechnung der Standsicherheit wird zum einen das Moment, das dem Bauteil zum Stehen hilft und das Moment, das das Bauteil zum Kippen bringt betrachtet.

Moment, das dem Bauteil zum Stehen hilft.	$M_s$
Moment, das das Bauteil zum Kippen bringt.	$M_k$

$M_s > M_k \rightarrow$  Bauteil steht

$M_s = M_k \rightarrow$  kritischer Punkt ,Bauteil steht

$M_s < M_k \rightarrow$  Bauteil kippt

Tabelle 22: Standsicherheit

Bei den Berechnungen der Standsicherheit wird  $M_s = M_k$  gesetzt. Somit kann der kritische Punkt berechnet werden, bei dem das Bauteil gerade noch steht.

##### Allgemeine Berechnungen

Für die Berechnung der Standsicherheit muss überlegt werden, welche Kräfte auf den Unterbau wirken. Zum einen ist das die Gewichtskraft des Unterbaus, der Scanner und die Kraft, die ein Mensch beim Bedienen des iPads aufbringt.

Die Druckkraft des Menschen wurde aus einem Handbuch (Seite 9) entnommen. Sie wirkt 45 Grad zum Koordinatensystem. Deshalb wird sie in die beiden Kräfte  $F_{M1}$  und  $F_{M2}$  aufgeteilt.

Das Handbuch Druckkraft befindet sich im Anhang

Benennung	Formelzeichen	Wert	Einheit
Gewicht Scanner	$m_s$	0.2	kg
Druckkraft Mensch	$F_M$	50	N
Gewicht Bauteil mit iPad	$m_G$	1.6	kg

Tabelle 23: Allgemeine Berechnungen

##### Berechnung Kraft Scanner, $F_s$ :

$$F_s = m_s \times g = 0.2kg \times \frac{9.81m}{s^2} = 1.962N$$

**Berechnung Kraft  $F_{M1}$  und  $F_{M2}$  ( $F_{M1}=F_{M2}$  da  $45^\circ$ ):**

$$F_{M1} = F_{M2} = \sqrt{\frac{F_M^2}{2}} = \sqrt{\frac{50^2}{2}} = 35.4N$$

**Berechnung Gewichtskraft  $F_G$ :**

$$F_G = m_G \times g = 1.6kg \times \frac{9.81m}{s^2} = 15.69N$$

**Fallunterscheidung** Die Druckkraft des Menschen kann an verschiedenen Orten angreifen, je nach dem wo der Benutzer auf das iPad drückt. Deshalb werden für die Berechnungen drei verschiedene Fälle betrachtet, damit der kritischster Fall ermittelt werden kann.

**Fall 1:** Benutzer bedient iPad ganz unten am Display

**Fall 2:** Benutzer bedient iPad in der Mitte des Displays

**Fall 3:** Benutzer bedient iPad ganz oben am Display

**Berechnung Fall 1 Kraft  $F_M$  greift bei der Bedienfläche unten an:**

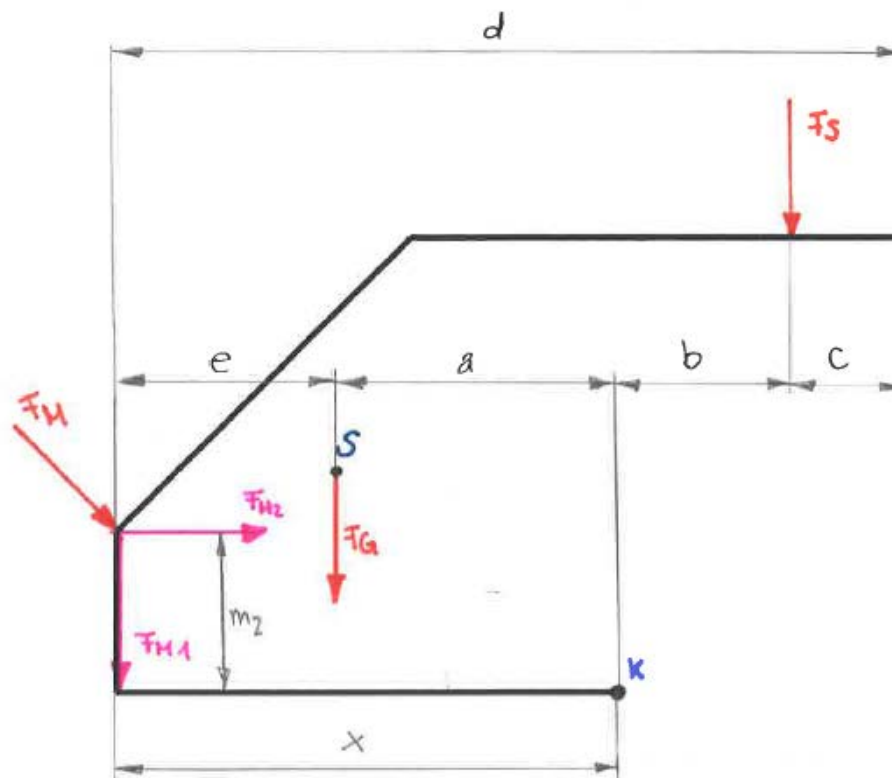


Abbildung 41: Berechnung Fall 1

Benennung	Formelzeichen	Wert	Einheit
Kraft Scanner	$F_s$	1.962	N
Druckkraft Mensch	$F_M$	50	N
Kraft	$F_{M1}, F_{M2}$	35.4	N
Gewichtskraft	$F_G$	15.69	N
Abstand d	d	0.21	m
Abstand c	c	0.04	m
Abstand e	e	0.072	m
Länge Grundfläche	x	?	m
Abstand a	a	x-e	m
Abstand b	b	d-c-x	m
Abstand m2	$m_2$	0.1	m

Tabelle 24: Berechnung Fall 1

**Berechnung x minimal, damit Bauteil noch steht ( $M_k = M_s$ ):**

$$F_s \times b + F_{M2} \times m_2 = F_G \times a + F_{M1} \times x$$

$$1.962\text{N} \times (0.21\text{m} - x - 0.04\text{m}) + 35.4\text{N} \times 0.1\text{m} = 15.69\text{N} \times (x - 0.072\text{m}) + 35.4\text{N} \times x$$

$$x = 0.094\text{m} = \mathbf{94\text{mm}}$$

**Berechnung Fall 2 Kraft  $F_M$  greift bei der Bedienfläche in der Mitte an:**

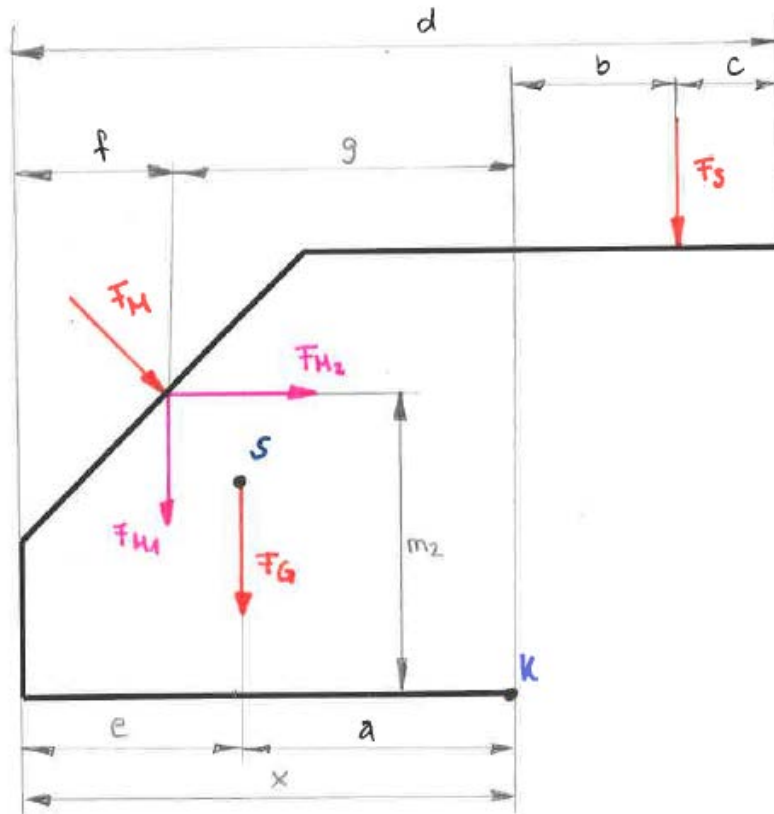


Abbildung 42: Berechnungen Fall 2



Benennung	Formelzeichen	Wert	Einheit
Kraft Scanner	$F_s$	1.962	N
Druckkraft Mensch	$F_M$	50	N
Kraft	$F_{M1}, F_{M2}$	35.4	N
Gewichtskraft	$F_G$	15.69	N
Abstand d	d	0.21	m
Abstand c	c	0.04	m
Abstand e	e	0.072	m
Länge Grundfläche	x	?	m
Abstand a	a	x-e	m
Abstand b	b	d-c-x	m
Abstand m2	$m_2$	0.165	m
Abstand f	f	0.065	m
Abstand g	g	x-f	m

Tabelle 25: Berechnungen Fall 2

**Berechnung x minimal, damit Bauteil noch steht ( $M_k=M_s$ ):**

$$F_s \times b + F_{M2} \times m_2 = F_G \times a + F_{M1} \times g$$

$$1.962N \times (0.21m - x - 0.04m) + 35.4N \times 0.165m$$

$$= 15.69N \times (x - 0.072m) + 35.4N \times (x - 0.065m)$$

$$x = 0.181m = \mathbf{181mm}$$

**Berechnung Fall 3 Kraft  $F_M$  greift bei der Bedienfläche oben an:**

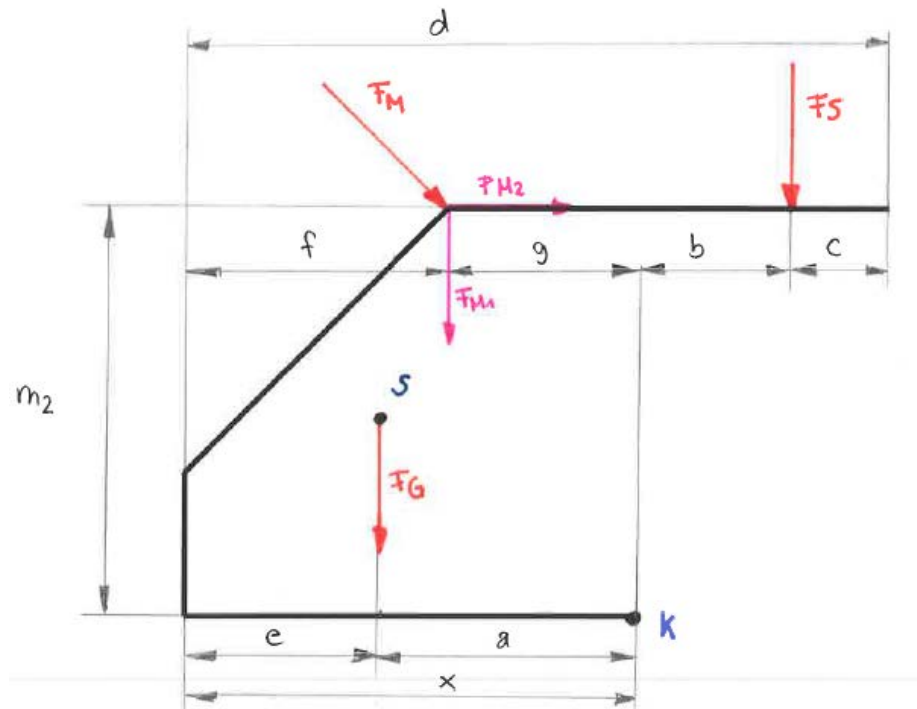


Abbildung 43: Berechnungen Fall 3

Benennung	Formelzeichen	Wert	Einheit
Kraft Scanner	$F_s$	1.962	N
Druckkraft Mensch	$F_M$	50	N
Kraft	$F_{M1}, F_{M2}$	35.4	N
Gewichtskraft	$F_G$	15.69	N
Abstand d	d	0.21	m
Abstand c	c	0.04	m
Abstand e	e	0.072	m
Länge Grundfläche	x	?	m
Abstand a	a	x-e	m
Abstand b	b	d-c-x	m
Abstand m2	$m_2$	0.23	m
Abstand f	f	0.13	m
Abstand g	g	x-f	m

Tabelle 26: Berechnungen Fall 3

**Berechnung x minimal, damit Bauteil noch steht ( $M_k=M_s$ ):**

$$F_s \times b + F_{M2} \times m_2 = F_G \times a + F_{M1} \times g$$

$$1.962N \times (0.21m - x - 0.04m) + 35.4N \times 0.23m$$

$$= 15.69N \times (x - 0.072m) + 35.4N \times (x - 0.13m)$$

$$x = 0.268m = \mathbf{268mm}$$

Wie aus den Ergebnissen hervorgeht, muss x beim Fall 3 am grössten sein. (**Kritischster Fall**). Deshalb wird ab nun mit diesem Fall 3 weiter gerechnet.

**Berechnung Sicherheit bezogen auf Grundfläche**

Zuerst wird hier berechnet, wie gross die maximale Druckkraft des Menschen ( $F_M$ ) sein darf wenn die Länge der Grundfläche x einen Sicherheitsfaktor von 1.5 hat.

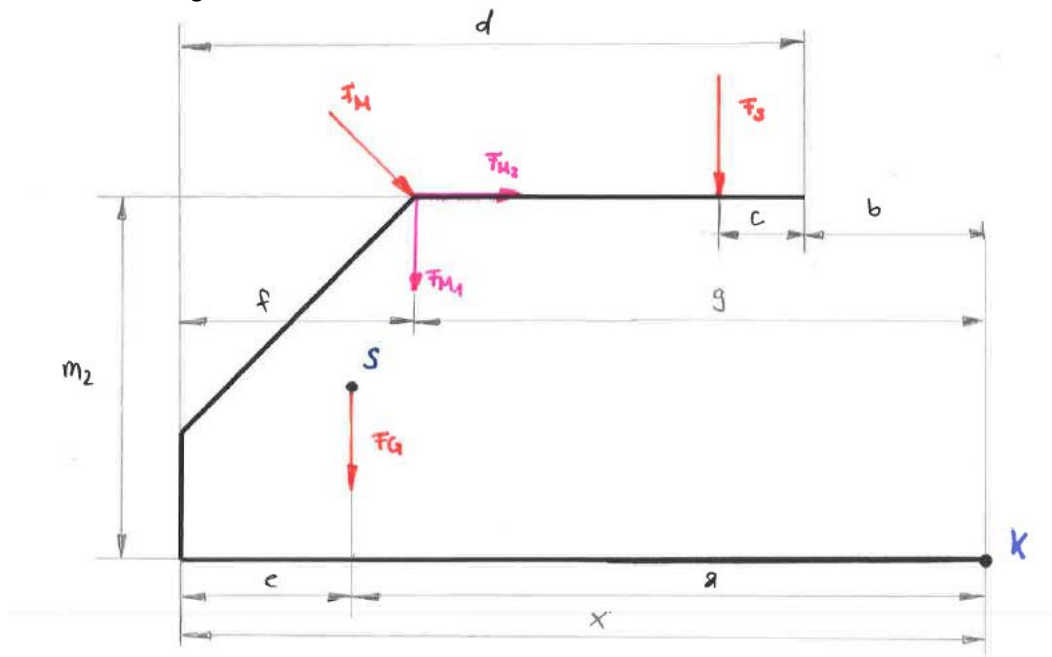


Abbildung 44: Berechnungen Sicherheit bezogen auf die Grundfläche

**Neue Länge x:**

$$x_{neu} = x_{alt} \times 1.5 = 0.402m = 402mm$$

Benennung	Formelzeichen	Wert	Einheit
Kraft Scanner	$F_s$	1.962	N
Druckkraft Mensch	$F_M$	?	N
Kraft	$F_{M1}, F_{M2}$	?	N
Gewichtskraft	$F_G$	15.69	N
Abstand d	d	0.21	m
Abstand c	c	0.04	m
Abstand e	e	0.072	m
Länge Grundfläche	x	0.402	m
Abstand a	a	x-e	m
Abstand b	b	x-d	m
Abstand m2	$m_2$	0.23	m
Abstand f	f	0.13	m
Abstand g	g	x-f	m

Tabelle 27: Berechnung Sicherheit bezogen auf die Grundfläche

**Neue Kraft  $F_{M1}/F_{M2}$ :**

$$\begin{aligned}
 F_{M2neu} \times m_2 &= F_s \times (c + b) + F_G \times a + F_{M1neu} \times g \\
 F_{M2neu} \times 0.23m &= 1.962N \times (0.04m + 0.192m) + 15.69N \times (0.402m - 0.072m) + F_{M1neu} \\
 &\quad \times (0.402 - 0.13m) \\
 F_{M2neu} &= F_{M1neu} = 134.12N
 \end{aligned}$$

**Neue Kraft  $F_M$ :**

$$F_{Mneu} = \sqrt{2 \times F_{M2}^2} = \mathbf{189.67N}$$

**Sicherheitsfaktor für die neue Kraft  $F_M$ :**

$$S = \frac{F_{Mneu}}{F_{Malt}} = \frac{189.67N}{50N} = \mathbf{3.79}$$

Für einen Menschen ist es praktisch unmöglich, eine Kraft von 189.67N nur mit dem Zeigefinger aufzubringen. Diese hohe Sicherheit ist für dieses Bauteil nicht von Nöten. Deshalb wird noch eine zweite Sicherheitsberechnung durchgeführt.

**Berechnung Sicherheit bezogen auf Kraft**

In der zweiten Sicherheitsberechnung wird berechnet, wie lang  $x$  mindestens sein muss, wenn die Kraft  $F_M$  einen Sicherheitsfaktor von 1.5 hat.

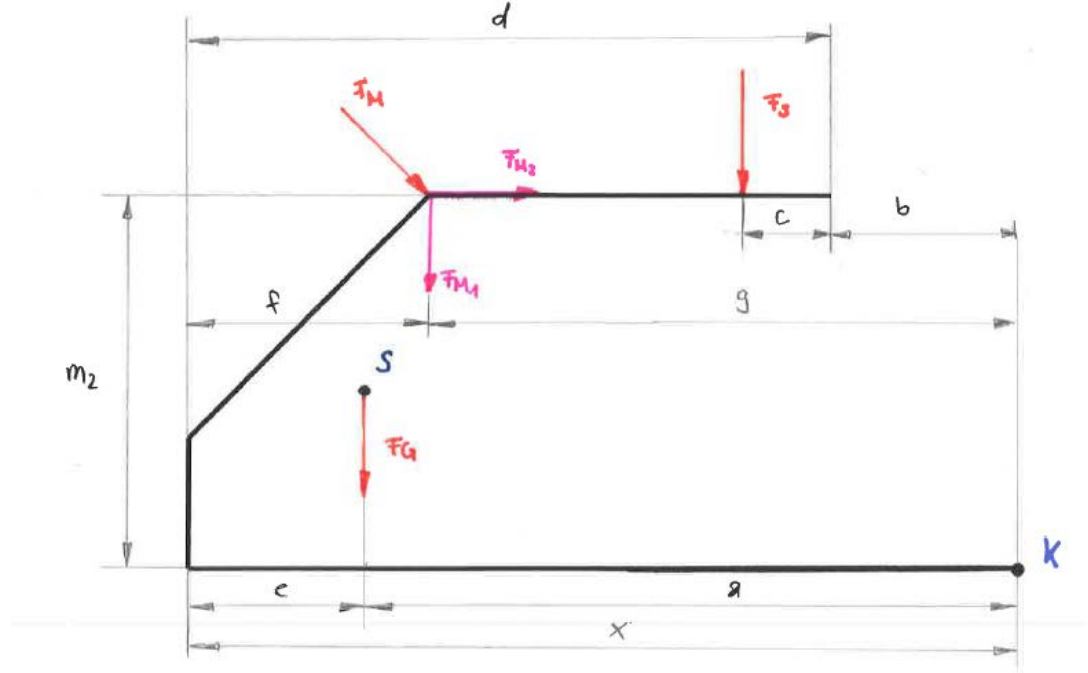


Abbildung 45: Berechnung Sicherheit bezogen auf die Kraft

Neue Kraft  $F_M$ :

$$F_{Mneu} = F_{Malt} \times 1.5 = 75N$$

Neue Kraft  $F_{M1}/F_{M2}$ :

$$F_{M1} = F_{M2} = \sqrt{\frac{F_M^2}{2}} = \sqrt{\frac{75^2}{2}} = 53.1N$$

Benennung	Formelzeichen	Wert	Einheit
Kraft Scanner	$F_s$	1.962	N
Druckkraft Mensch	$F_M$	75	N
Kraft	$F_{M1}, F_{M2}$	53.1	N
Gewichtskraft	$F_G$	15.69	N
Abstand d	d	0.21	m
Abstand c	c	0.04	m
Abstand e	e	0.072	m
Länge Grundfläche	x	?	m
Abstand a	a	x-e	m
Abstand b	b	x-d	m
Abstand m2	$m_2$	0.23	m
Abstand f	f	0.13	m
Abstand g	g	x-f	m

Tabelle 28: Berechnungen Sicherheit bezogen auf die Kraft

### Neue Länge Grundfläche x:

$$F_{M2} \times m_2 = F_S \times (c + b) + F_G \times a + F_{M1} \times g$$

$$53.1N \times 0.23m = 1.962N \times (0.04m + (x - 0.21)) + 15.69N \times (x - 0.072m) + 53.1 \times (x - 0.13m)$$

$$x = 0.2908m = \mathbf{290.8mm}$$

Diese Sicherheit reicht für den Unterbau aus.

**Erkenntnis, weiteres Vorgehen** Aus den Berechnungen oben geht hervor, dass der Fall 3, also der Mensch bedient das iPad ganz oben am Display, am kritischsten ist. Deshalb wurde auch mit diesem Fall die Sicherheit berechnet.

Die erste Sicherheitsrechnung hat eine Sicherheit, bezogen auf die Druckkraft, von 3.79. Für den Unterbau reicht auch eine kleinere Sicherheit. Bei der zweiten Sicherheitsberechnung wurde eine Sicherheit, bezogen auf die Druckkraft, von 1.5 gewählt. Diese Sicherheit reicht vollkommen aus, weshalb die Grundfläche auf ein Endmass von 295mm angepasst wurde.

### 4.2.2.3 Schwerpunkt Endbauteil

**Beschreibung** Die Grundfläche wurde durch die Erkenntnisse der Berechnungen oben von 210mm auf 295mm verlängert. Somit verschiebt sich auch der Schwerpunkt des Bauteils. Dieser wird hier neu berechnet. Zudem wird mit einer Sicherheitsberechnung überprüft, ob die gewünschte Sicherheitsfaktor von 1.5 noch vorhanden ist.

Danach wird noch berechnet, mit welcher Kraft  $F_K$  man das Bauteil tatsächlich zum kippen bringt.

Zum Schluss wird die Haftreibung berechnet. Somit weiss man, ob der Unterbau zuerst kippt oder wegrutscht.

**Neuer Schwerpunkt Handrechnung** Neuer Schwerpunkt, wenn  $x = 295\text{mm}$  statt  $210\text{mm}$ :  
 Wert in X Richtung = 110.1mm  
 Wert in Y Richtung = 0mm  
 Wert in Z Richtung = 88.7mm

**Neuer Schwerpunkt CAD** Auch hier wurde mit dem CAD die Handrechnung überprüft:  
 Wert in X Richtung = 107mm  
 Wert in Y Richtung = 0mm  
 Wert in Z Richtung = 91.4mm

Auch hier können die geringen Abweichungen dadurch entstehen, dass beim CAD der Moosgummi nicht berücksichtigt wurde.

**Schwerpunkt mit Scanner** Für den endgültigen Unterbau wurde noch der Schwerpunkt berechnet, wenn der Scanner auf der rechten Seite befestigt wird:  
 Wert in X Richtung = 229.99mm  
 Wert in Y Richtung = 6.25mm  
 Wert in Z Richtung = 171.2mm

Es ist zu sehen, dass sich der Schwerpunkt um einiges verschiebt, wenn der Scanner am Unterbau befestigt wird.

**FM Max für neuen Schwerpunkt und neue Grundfläche x** Da die Länge und der neue Schwerpunkt nun bekannt sind, kann jetzt die neue  $F_M$  Max. berechnet werden, mit dem der Benutzer das iPad bedienen darf, ohne dass der Unterbau kippt.

Benennung	Formelzeichen	Wert	Einheit
Kraft Scanner	$F_s$	1.962	N
Druckkraft Mensch	$F_M$	?	N
Kraft	$F_{M1}, F_{M2}$	?	N
Gewichtskraft	$F_G$	15.69	N
Abstand d	d	0.21	m
Abstand c	c	0.04	m
Abstand e	e	0.0887	m
Länge Grundfläche	x	0.295	m
Abstand a	a	x-e	m
Abstand b	b	x-d	m
Abstand m2	$m_2$	0.23	m
Abstand f	f	0.13	m
Abstand g	g	x-f	m

Tabelle 29: FM Max für die neue Grundfläche

**Neue Kraft  $F_{M1}/F_{M2}$ :**

$$F_{M2neu} \times m_2 = F_s \times (c + b) + F_G \times a + F_{M1neu} \times g$$

$$F_{M2neu} \times 0.23m = 1.962N \times (0.04m + 0.085m) + 15.69N \times (0.295m - 0.0887m) + F_{M1neu} \times (0.295 - 0.13m)$$

$$F_{M2neu} = F_{M1neu} = 53.57N$$

**Neue Kraft  $F_M$ :**

$$F_{Mneu} = \sqrt{2 \times F_{M2}^2} = 75.76N$$

**Sicherheitsfaktor für die neue Kraft  $F_M$ :**

$$S = \frac{F_{Mneu}}{F_{Malt}} = \frac{75.76N}{50N} = 1.52$$

Der Sicherheitsfaktor von 1.5 wird immer noch eingehalten. Der Unterbau kann so hergestellt werden.

**Berechnung max. Kippkraft nach vorne**

Zum Schluss wird noch berechnet, mit welcher Kraft  $F_K$  man von hinten an den Unterbau stossen kann, ohne dass er nach vorne kippt.

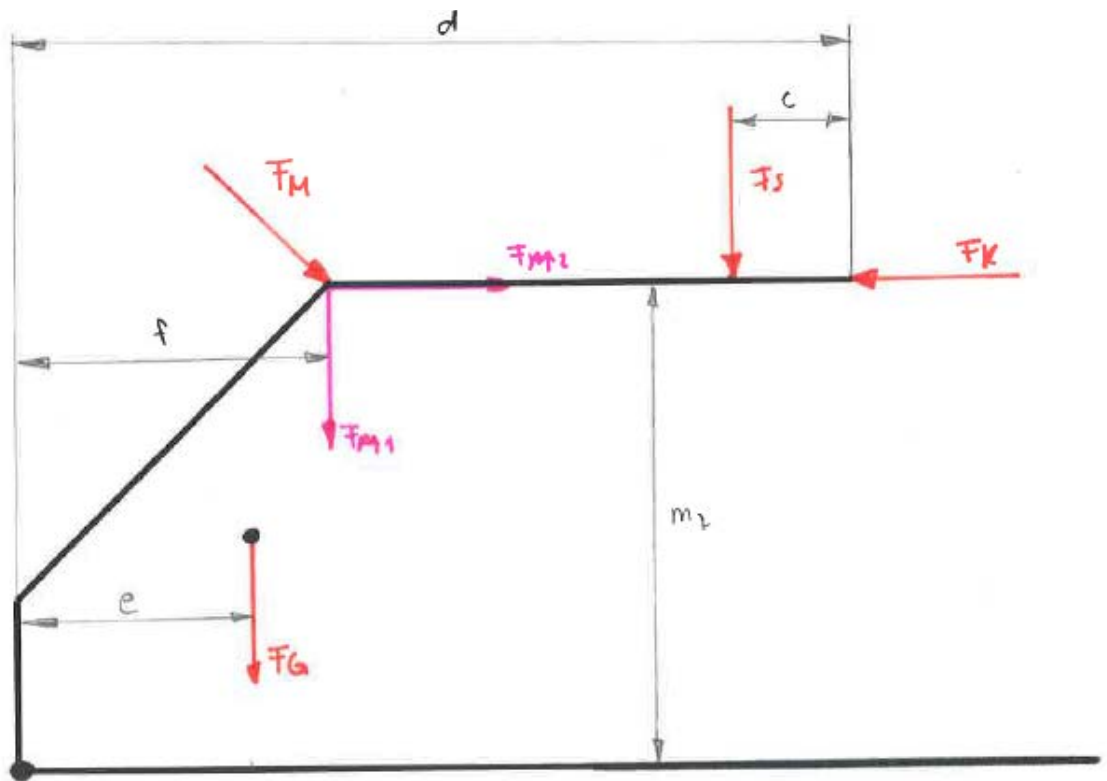


Abbildung 46: Berechnung max. Kippkraft nach vorne

**Max. Kraft  $F_K$ , damit Unterbau nicht nach vorne kippt:**

$$\begin{aligned}
 F_{M2} \times m_2 + F_S \times (d - c) + F_G \times e + F_{M1} \times f &= F_K \times m_2 \\
 35.4N \times 0.23m + 1.962N \times (0.21m - 0.04m) + 15.69N \times 0.072m + 35.4N \times 0.13m \\
 &= F_K \times 0.23m \\
 F_K &= \mathbf{61.77N}
 \end{aligned}$$

Die Stabilität reicht aus, damit ein Kippen nach vorne verhindert werden kann.

#### Haftreibung

Um herauszufinden, ob der Unterbau bei dieser Kraft  $F_K$  kippt oder wegrutscht, wird noch die Haftreibung des Unterbaus auf dem Tisch berechnet.

Für diese Berechnung braucht es den Reibungskoeffizient zwischen Stahl (Tisch) und Moosgummi (Unterbau). Da dieser nirgends im Internet zu finden war, musste er zuerst von Hand bestimmt werden. Dazu wurde auf ein Brett etwas Moosgummi geklebt und darauf ein Stück aus Stahl gelegt. Danach wurde der Winkel des Bretts so lange verändert, bis das Stahlstück leicht zu rutschen begann. Dieser Winkel wurde dann gemessen.

Gemessener Winkel:  $25^\circ$

Reibungskoeffizient:  $\mu_H = \tan(25) = 0.466$

Vergleicht man diesen Wert mit dem Wert von Leder auf Grauguss (0.4), sieht man, dass der Berechnete Reibungskoeffizient etwa stimmen kann.

**Haftreibung:**  $F_H = \mu_H \times (F_G + F_S) = 0.466 \times (1.962N + 15.69N) = \mathbf{8.225N}$

Da  $F_H < F_K$  ist, würde der Unterbau vorher auf dem Tisch rutschen, als dass er kippt.

### 4.2.3 Konstruktion

**Einleitung** Die Konstruktion wird möglich einfach gehalten, damit die Herstellung einfach, schnell und somit kostengünstig ist. Auch wird darauf geachtet, dass der Unterbau designtechnisch in die Umgebung der REGA passt.

**Beschreibung** **Befestigung iPad**  
Das iPad wird mit Hilfe von Laschen befestigt. Dadurch kann es einfach in den Unterbau gelegt und entfernt werden. Der Unterbau ist so konzipiert, dass das iPad quer in den Unterbau gelegt wird. Es ist aber auch möglich, das iPad hoch hinein zu legen, es ist dann einfach weniger stabil befestigt.

#### **Befestigung Scanner**

Der Scanner kann entweder links oder rechts in die Ausfräsung gehängt werden (je nach Links- oder Rechtshänder). Das Kabel des Scanners kann aufgewickelt in die zwei Ausfräsungen hinten befestigt werden. Somit ist das Kabel und der Scanner bei nicht Gebrauch des Trackingsystems immer im Unterbau platziert und geht so nicht verloren.

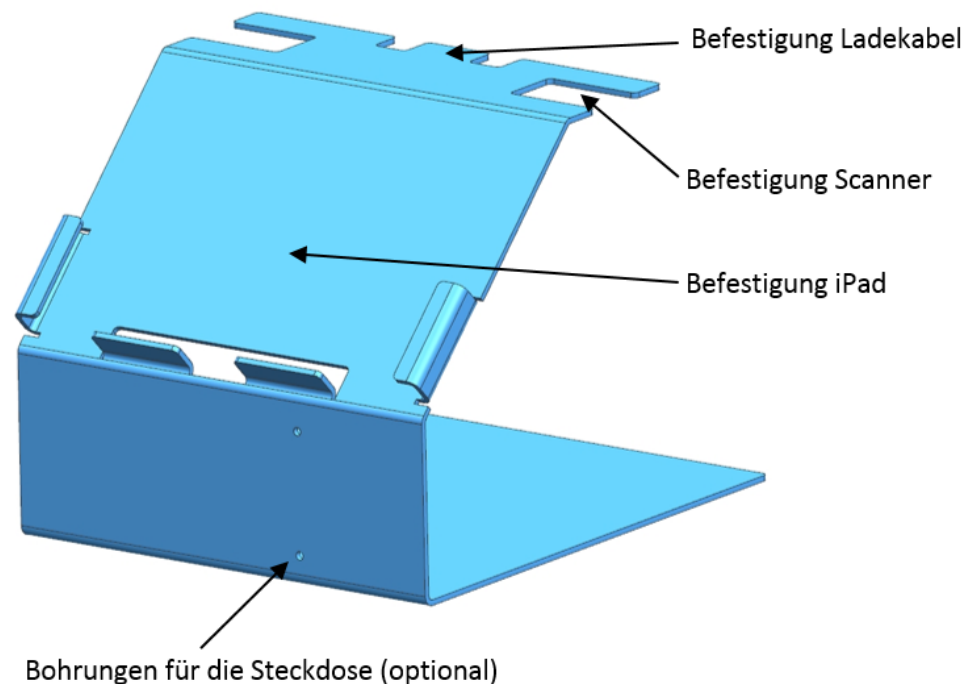


Abbildung 47: Prototyp Unterbau



## Material

Da der ganze Unterbau mit Desinfektionsmittel gereinigt wird, muss er aus rostfreiem Material sein. Dafür kam Edelstahl und Aluminium infrage. Es wurde Aluminium gewählt, da es leichter und preisgünstiger ist. Somit kann der Unterbau auch problemlos herumgetragen und anders platziert werden.

### 4.2.3.1 Zubehör

#### Moosgummi

Um das iPad, den Scanner und den Tisch nicht zu verkratzen, muss auf den Unterbau eine weichere Unterlage geklebt werden, welche auch mit Desinfektionsmittel gereinigt werden kann. Dazu wurden verschiedene Materialien auf deren Beständigkeit gegen Desinfektionsmittel getestet.

*Die Materialtests befinden sich im Anhang*

Als Unterlage wurde Moosgummi ausgewählt, da er beständig gegen Desinfektionsmittel, weich, selbstklebend und preisgünstig ist.

Wird der Unterbau für das iPad Air gebraucht, wird ein 2mm dicker Moosgummi verwendet. Sollte jedoch das iPad Air 2 zum Einsatz kommen, muss ein 3mm dicker Moosgummi verwendet werden, da das iPad Air 2 dünner ist als das iPad Air.

#### Scanner

Da auf den Verpackungen Barcodes oder Datamatrix Codes vorhanden sind, muss ein Scanner benutzt werden, der sowohl 1D als auch 2D Codes scannen kann. Damit der Benutzer beim Scannen flexibel ist, eignet sich besonders ein Scanner ohne Kabel. Der Scanner kann nun ganz einfach als externe Tastatur am iPad angeschlossen werden und sendet die Daten dem iPad via Bluetooth. Er kann auch an alle anderen Geräte, die Bluetooth besitzen, angeschlossen werden. (z.B. Laptop, PC)

Es wurde ein Scanner der Firma Datalogic ausgewählt, da dieser den gewünschten Anforderungen entsprach. Zudem war er im Vergleich zur Konkurrenz eher preisgünstig.

#### Scanner laden

Zum Scanner wird ein Ladekabel geliefert, das nur einen USB Anschluss besitzt. Mit diesem Kabel ist es möglich, den Scanner am USB-Port eines Computers zu laden. Nicht so an der Steckdose. Möchte man den Scanner an der Steckdose laden, braucht es zusätzlich ein Direktlade-Netzteil (94ACC1380) und einen Steckdosenadapter. Das Direktlade-Netzteil kann direkt bei der Firma ID-Systems AG in 2557 Studen CH bezogen werden. Der Steckdosenadapter kann in jedem handelsüblichen Elektronikgeschäft gekauft werden.

#### Steckdose

Die REGA möchte den ganzen Unterbau mit einem feuchten Lappen reinigen können. Deshalb muss die Steckdose spritzwassergeschützt sein. Es wird eine handelsübliche 3fach Steckdose aus dem Baumarkt verwendet, die extra für den Aussenbereich (spritzwassergeschützt) geeignet ist. Diese kann einfach mit 2 Schrauben an dem Unterbau platziert werden.

#### Schrauben und Muttern

Für die Befestigung der Steckdose reichen zwei M4X16 Linsenschrauben und dazu je zwei M4 Muttern und zwei Scheiben für M4. Es ist darauf zu achten, dass die Schrauben aus rostfreiem Material sind, damit auch diese für den Gebrauch von Desinfektionsmittel geeignet sind.

### 4.2.3.2 Modularität

#### Laden der Baugruppe

Der Wunsch der REGA war es, das beim Unterbau optional auch noch eine Steckdose eingebaut ist. Deshalb wurde der Unterbau so gestaltet, dass einfach die Steckdose bei Bedarf ergänzt werden kann oder nicht.

Hier wird nun der Aufbau des ganzen Medikamententrackingsystem dargestellt:

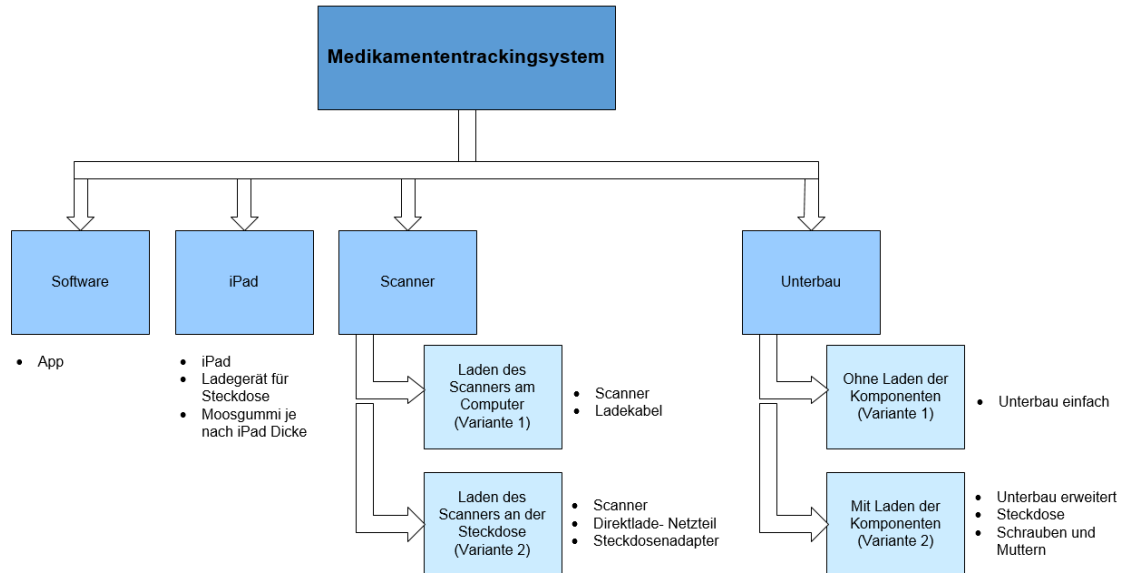


Abbildung 48: Modularität Medikamententrackingsystem

Auf diesem Bild ist zu sehen, aus welchen Baugruppen das Medikamententrackingsystem besteht, und welche Komponenten zu den jeweiligen Baugruppen gehören und dementsprechend bestellt werden müssen. Das System besteht zwingend aus der Software, dem iPad, dem Scanner und dem Unterbau.

Beim Scanner kann man dabei zwischen zwei Varianten wählen.

#### Variante 1: Laden des Scanners am Computer

Der Scanner wird mit dem USB Kabel bestellt. Somit kann er am Computer aufgeladen werden.

#### Variante 2: Laden des Scanners an der Steckdose (Unterbau)

Um den Scanner an der Steckdose laden zu können, muss zum Scanner noch die Direktlade- Netzteil und ein Steckdosenadapter dazu gekauft werden. Somit kann der Scanner direkt an der Steckdose aufgeladen werden.

Auch beim Unterbau kann dabei zwischen zwei Varianten ausgewählt werden.

#### Variante 1: Ohne Laden der Komponenten

Der Unterbau wird ohne zusätzlichen Bohrungen vorne bestellt.

### Variante 2: Laden der Komponenten

Der Unterbau wird mit den zwei zusätzlichen Bohrungen bestellt. Dazu wird eine 3fach Steckdose bestellt und auf die zwei Bohrungen montiert. Um die Steckdose zu befestigen braucht es noch je zwei Schrauben, Scheiben und Muttern. Dadurch kann nun der Scanner (nur wenn Ladestation Scanner vorhanden) und das iPad geladen werden.

Somit ist gewährleistet, dass jede Basis je nach Wunsch und Bedarf ihren Unterbau auswählen und individuell bestellen kann.

## 4.2.4 Herstellung und Kosten

**Detailzeichnungen** Zum Bestellen des Unterbaus wurden Detailzeichnungen erstellt. Es wurde je nachdem ob der Unterbau mit oder ohne Steckdose bestellt wird zwei Detailzeichnungen erstellt. Die Zeichnung d-00775\_P0005\_z01-a--1 ist für ein Unterbau ohne Steckdose geeignet und die Zeichnung d-00775\_P0006\_z01-a--1 für ein Unterbau mit Steckdose.  
*Die Detailzeichnungen befinden sich im Anhang*

**Offerten Unterbau** Für die Herstellung des Unterbaus wurden bei zwei Blechherstellern Offerten eingeholt.

### Ruoss Tech AG

Die Ruoss Tech AG offerierte uns den Unterbau zu einem Preis von **160.90 Fr.**

*Die Offerte befindet sich im Anhang*

### eMDe Blechfabrik AG

Leider bekamen wir keine Offerte auf unsere Anfrage.

Da die Ruoss Tech AG einen fairen Preis offerierte, wurden sie mit der Herstellung des Unterbaus beauftragt. Es wird der Unterbau mit Steckdose bestellt, da mit dieser Variante dem Kunden alle Möglichkeiten offen stehen.

### Kosten

Für das Projekt durfte ein Budget von 2000 Fr. nicht überschritten werden. Die Kosten der einzelnen Komponenten belaufen sich auf folgende Beträge:

Benennung	Hersteller/ Lieferant	Material	Anzahl	Kosten/Stk.	Kosten Gesamt
Scanner	Conrad.ch	Kunststoff	1	329.95	329.95
Steckdose	Bau und Hobby	Kunststoff	1	84.95	84.95
Kabel	Bau und Hobby	Kunststoff	1	5.95	5.95
Moosgummi 3mm	Staufenbiel Modellbau Schweiz GmbH	Moosgummi	1	16.88	16.88
Moosgummi 2mm	KEL- Modellbau Senn, MuttENZ	Moosgummi	1	16.87	16.87
Unterbau	Ruoss Tech AG	Alu	1	160.9	160.9
Schraube M4 X 16	HSR Werkstatt	rostfreier Stahl	2	0.5	1
Mutter M4	HSR Werkstatt	rostfreier Stahl	2	0.5	1
Scheibe M4	HSR Werkstatt	rostfreier Stahl	2	0.5	1
Direktlade- Netzteil	ID- Systems AG		1	30	30
Steckdosenadapter	m electronics Migros		1	14.8	14.8

Tabelle 30: Kostenübersicht aller Komponenten

Da der Unterbau Modular aufgebaut ist, ist der Preis pro Variante verschieden:

**Variante 1: ohne Steckdose, ohne Laden des Scanners an der Steckdose (Laden am Computer)**

Benennung	Anzahl	Kosten/Stk. [Fr.]	Kosten Gesamt [Fr.]
Scanner	1	329,95	329,95
Moosgummi 3mm	1	16,88	16,88
Moosgummi 2mm	1	16,87	16,87
Unterbau	1	160,9	160,9

Tabelle 31: Kosten Variante 1

Total Kosten Variante 1: **525.6 Fr.**

**Variante 2: mit Laden des Scanners an der Steckdose (extern)**

Benennung	Anzahl	Kosten/Stk. [Fr.]	Kosten Gesamt [Fr.]
Scanner	1	329,95	329,95
Moosgummi 3mm	1	16,88	16,88
Moosgummi 2mm	1	16,87	16,87
Unterbau	1	160,9	160,9
Direktlade- Netzteil	1	30	30
Steckdosenadapter	1	14.8	14.8

Tabelle 32: Kosten Variante 2

Total Kosten Variante 2: **569.4 Fr.**

**Variante 3: mit Steckdose, mit Laden des Scanners an der Steckdose**

Benennung	Anzahl	Kosten/Stk. [Fr.]	Kosten Gesamt [Fr.]
Scanner	1	329,95	329,95
Steckdose	1	84,95	84,95
Kabel	1	5,95	5,95
Moosgummi 3mm	1	16,88	16,88
Moosgummi 2mm	1	16,87	16,87
Unterbau	1	160,9	160,9
Schraube M4 X 16	2	0,5	1
Mutter M4	2	0,5	1
Scheibe M4	2	0,5	1
Direktlade- Netzteil	1	30	30
Steckdosenadapter	1	14.8	14.8

Tabelle 33: Kosten Variante 3

Total Kosten Variante 3: **663.3 Fr.**

**Alle drei Varianten liegen im Budget von 2000 Fr.**

**Anmerkung** Es ist zu beachten, dass nur die Materialkosten beachtet wurden. Die Konstruktions-, Montage- und Inbetriebnahme- Kosten wurden bei dieser Auflistungen nicht beachtet.

#### 4.2.5 Ergebnis/ Fazit

---

**Fazit** Durch die Berechnungen ist ein Unterbau entstanden, welcher die Anforderungen im Betrieb standhält. Es ist zu beachten, dass der Unterbau nur für das iPad Air oder das iPad Air 2 geeignet ist. Ansonsten muss die Geometrie angepasst werden. Der modulare Aufbau ermöglicht ein individueller Unterbau je nach Bedürfnis des Kunden. Auch liegen alle Varianten unter dem Maximalen Budget von 2000 Fr.

#### 4.3 Fazit Design

---

**Fazit** Die Architektur und der Umfang des Systems sind definiert und es kann mit der Umsetzung begonnen werden.  
Der Unterbau ist fertig konstruiert und kann nun bestellt werden.

---

## 5 Realisierung

---

**Beschreibung** Im Kapitel Realisierung wird das System, das in der Phase Design ausgearbeitet wurde, umgesetzt. Dies bedeutet, dass der Programmcode geschrieben und der Unterbau montiert wird. Am Ende der Phase steht der fertige Prototyp für die Integrationstests bereit.

### 5.1 Software

---

**Beschreibung** Im folgenden Kapitel werden die effektiv umgesetzten Teile der Software genauer erklärt. Es werden die wichtigsten Software Komponenten beschrieben. Hierbei wird allerdings nicht der gesamte Source Code aufgeführt sondern nur auf die Hauptbestandteile der Applikation eingegangen. Das Kapitel soll einen guten Einblick in die Strukturierung des Source Codes der zwei verschiedenen Teile (Client und Server) geben.

#### 5.1.1 Generell

---

**Unit Testing** Das Unit Testing wurde während der Realisierungsphase „händisch“ durchgeführt, d.h. ohne automatisierte Unit Tests (z.B. JUnit), wobei auf dem Serverteil ein paar wenige grundlegende Funktionen mittels Unit Tests geprüft wurden. Dies wurde bewusst so entschieden, da wir automatisierte Unit Tests, welche unserer Ansicht vor allem für das Refactoring von Code sehr nützlich sind, bei der Erstellung eines Prototypen nicht als all zu wichtig einschätzten. Zudem bieten die eingesetzten Frameworks Play (Serverteil) und Ionic (Clientteil) sehr flexible und realtime durchgeführte lokale Build und Deploys, sodass Änderungen am Sourcecode direkt lokal getestet werden können.

**Repositories** Der Sourcecode von Client und Server werden in separaten git-Repositories verwaltet. Die Datenbankskripts werden selten modifiziert und wurden deshalb direkt auf SWIT-CHdrive abgelegt. Die git-Repositories lauten:

**Client**

<https://git.hsr.ch/git/drugTracker-server>

**Server**

<https://git.hsr.ch/git/drugTracker-client>

*Der Sourcecode für Client und Server und die Datenbankskripts befinden sich im Anhang*

#### 5.1.2 Server

---

**Beschreibung** Der Server stellt Methoden zum Zugriff auf die Datenbank zur Verfügung. Die Businesslogik zur Ermittlung von bald ablaufenden Medikamenten und deren Versand als Frühwarnung via Email ist ebenfalls im Serverteil umgesetzt. Wie im Kapitel 4.1.3.3 erwähnt, wurde die Serverinfrastruktur mit Hilfe des Play Frameworks realisiert.

**controllers**

Enthält die Klasse *Application* und ist vom Typ *Controller*. *Controller* ist eine vom Play Framework zur Verfügung gestellte Klasse. Die *Application* Klasse dient als Controller für alle in der Applikation durchgeführten http-Requests (REST). Die im *routes*-File definierten Routen (für die http-requests) werden hier entgegengenommen und verarbeitet. Die Klasse enthält REST-konforme get, post- und put-Methoden für die Entity Objekte (siehe models).

**dao**

“DAO” steht für Data Access Objects. Sie kapseln den Zugriff auf die Datenbank und werden von der Klasse *Application* instanziiert. Für jede Entität (siehe models) wird ein DAO zur Verfügung gestellt.

Im Package ist auch das generische Interface *Dao* enthalten. Es bietet für alle möglichen Datenbankzugriffe (CRUD) leere, generische Methodenrumpfe an.

Diese Methoden sind in der ebenfalls generischen, abstrakten Klasse *DaoImpl* implementiert.

Auch im Package befindet sich die Klasse *DaoFabric*. Wie der Name verrät, handelt es sich dabei um eine Fabric. Sie enthält ausschliesslich statische Methoden zum Erstellen und Lesen der für jede Entityklasse vorhandenen Daos.

Das Diagramm unten beschreibt die Situation anhand vom Beispiel des *DrugDao*.

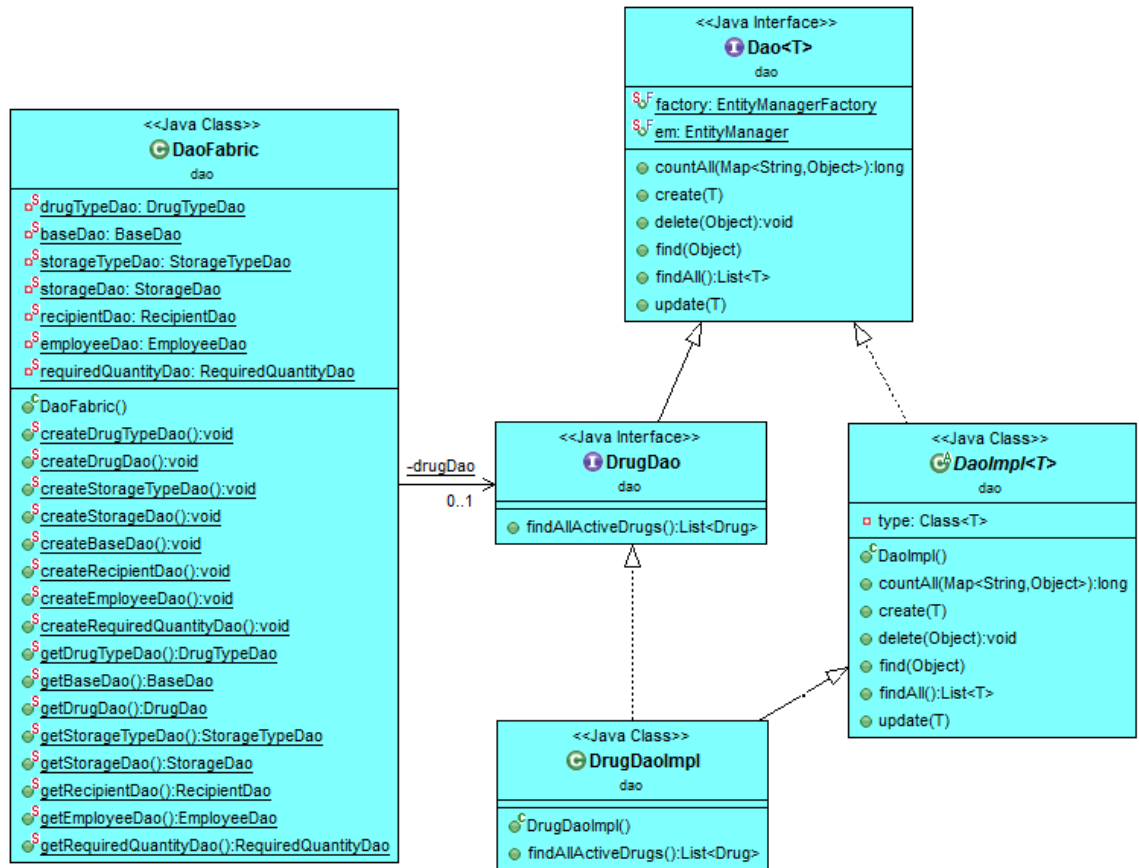


Abbildung 49: Klassendiagramm Data Access Objects

## globals

Enthält die Klasse *SettingUpApplication* die von der Play Klasse *GlobalSettings* erbt. Sie erledigt alle bei Applikationsstart durchzuführenden Aufgaben. Darunter zählt unter anderem auch das Starten eines Threads für die Früherkennung von ablaufenden Medikamenten.

Das Codefragment unten zeigt das Starten des Mailjobs als Thread.

```

// The MailJob runs every 24h hours
private static void startMailJob() {
    new Thread(() -> {
        try {
            while (true) {
                TimeUnit.HOURS.sleep(24);
                ExpiredDrugJob expiredDeadline = new ExpiredDrugJob();
                expiredDeadline.getExpiredDrugs(SelectExpiredDrugsEnum.SELECT_WITH_EXPIRED_DEADLINE);
                ExpiredDrugJob expiredExpiry = new ExpiredDrugJob();
                expiredExpiry.getExpiredDrugs(SelectExpiredDrugsEnum.SELECT_WITH_EXPIRED_EXPIRY_DATE);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }).start();
}

```

Abbildung 50: Starten des Mailjobs

## jobs

Beinhaltet die Klassen *ExpiredDrugJob* und *SendMail*. Erstere ist für das Ermitteln von abgelaufenen oder bald ablaufenden Medikamenten zuständig. Letztere verschickt die betroffenen Medikamente als Mail an definierte Benutzer.

## models

Im Package enthalten sind alle Entity Klassen. Sie werden für das objekt-relational Mapping (ORM) eingesetzt. Dabei wird die Referenzimplementation von JPA, eclipseLink eingesetzt. Das ORM ist direkt mittels annotations in den Klassen definiert.

## conf

Enthält das File *persistence.xml*, welches alle für die Datenbankverbindung notwendigen Konfigurationen enthält.

Das File *application.conf* wird von Play benötigt und enthält ebenfalls diverse Einstellungen für die Applikation.

Im *routes* File sind die Routen für alle in der Applikation verwendeten http-Requests festgelegt. Das File ermöglicht einen REST-konformen Aufbau der Requests. Unten aufgeführt ist die Konfiguration der http-Requests und deren Mapping auf die entsprechenden Methoden im Controller anhand vom Beispiel der Drug-Objektes.

# Drug		
GET	<u>/drugtracker</u> /drug/drugs	controllers.Application.getDrugs()
GET	<u>/drugtracker</u> /drug/activedrugs	controllers.Application.getActiveDrugs()
POST	<u>/drugtracker</u> /drug/adddrug	controllers.Application.addDrug()
PUT	<u>/drugtracker</u> /drug/updatedrug	controllers.Application.updateDrug()

Die erste Spalte definiert den Typ des entsprechenden http-Requests. In der zweiten Spalte ist die eigentliche URL, mit welcher die Ressource angesprochen werden kann. Die Klasse respektive Methode welche bei Zugriff auf die URL aufgerufen werden soll, wird in der dritten Spalte definiert.



---

<b>Deploy der Applikation</b>	<p>Folgende Schritte sind notwendig um den Serverteil auf einem beliebigen Server zu deployen:</p> <ol style="list-style-type: none"><li>1. Den Sourcecode vom git-Repository clonen.</li><li>2. In der Konsole in das entsprechende lokale Verzeichnis wechseln.</li><li>3. Die Eingabe von <code>'activator dist'</code> erzeugt eine binary Version der Applikation, sodass sie ohne Abhängigkeiten zum Playframework auf dem Server deployed werden kann.</li><li>4. Binary-File (ZIP!) auf den Server kopieren und entzippen.</li><li>5. Im Verzeichnis mittels <code>'drugTrackerServer -Dhttp.port=portNr Dhttp.address=iPadress'</code> den Play-Server starten.</li></ol>
<b>Starten der Applikation (lokal)</b>	<ol style="list-style-type: none"><li>1. Kopieren des Source-Codes (Clone des git-Repository)</li><li>2. Im lokalen Verzeichnis mittels <code>'activator run'</code> den lokalen Server starten.</li></ol>

### 5.1.3 Client

---

**Beschreibung** Der Client ist das Front-End zum ganzen System. Dieses wurde mit Hilfe von Ionic realisiert und als Hybrid-App auf ein iPad deployed. Der Benutzer kann über die App auf das System zugreifen und Medikamente einlagern und zuweisen. Untenstehend folgt eine Erklärung der Struktur des Codes, nachfolgend einige Screens und Erklärungen zu der App.

**www Folder** In diesem Folder sind die wichtigsten Files enthalten. Dieser ist wie folgt gegliedert:

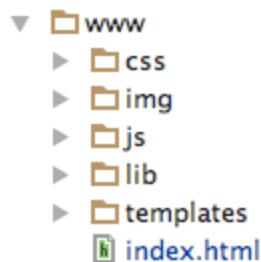


Abbildung 51: www-Folder

## app.js

In diesem File wird die Applikation definiert mit ihren Abhängigkeiten.

```
angular.module('starter', ['ionic', 'ngIOS9UIWebViewPatch', 'starter.controllers', 'starter.services'])
```

Abbildung 52: Auszug aus app.js

Der erste Parameter ist der Name dieses Angular Moduls, das nachfolgende Array beinhaltet die Dependencies.

Ionic verwendet AngularUI Router, welches sich dem Konzept von Zuständen bedient. Dadurch lassen sich verschiedene Controller den einzelnen Zuständen zuordnen.

```
.config(function($stateProvider, $urlRouterProvider) {  
  
  $stateProvider  
    .state('tab', {  
      url: '/tab',  
      abstract: true,  
      cache: false,  
      templateUrl: 'templates/tabs.html'  
    })  
  
    //Each tab has its own nav history stack:  
    .state('tab.admin', {  
      url: '/admin',  
      views: {  
        'tab-admin': {  
          templateUrl: 'templates/tab-admin.html',  
          controller: 'AdminCtrl'  
        }  
      }  
    })  
  })  
})
```

Abbildung 53: Auszug aus app.js

## controller.js

Enthält das AngularModul `starter.controllers` welches wiederum alle controller für die einzelnen Screens enthält.

```
angular.module('starter.controllers', [])  
  
  .controller('AdminCtrl', function ($scope, Storage, Drugs, $ionicPopup) {
```

Abbildung 54: Auszug aus controller.js

Auf den Controller werden die einzelnen Dependencies mitgegeben. Dadurch wird verhindert, dass jedes Mal alles geladen werden muss. Durch Hinzufügen von entry-Listener auf einzelne Controller, wird gewährleistet, dass diese bei erneutem Öffnen wieder ausgeführt werden. Sonst werden Controller nur beim Starten der App ausgeführt. Die Controller beinhalten die Business Logik.

Durch binden einzelner Objekte an die `$scope` Variable lassen sich von den Views auf diese Objekte zugreifen(siehe Dependency Injection)

services.js

Dieses Angular Modul dient zur Instanziierung einzelner Services, so zum Beispiel einer Factory für Drugs und eine für Storages. Durch Verwendung des 'Revealing Module Pattern' werden Daten und ihr Verhalten sauber gekapselt.

Ein Ausschnitt von der 'Drug-Factory':

```
angular.module('starter.services', [])  
  
.factory('Drugs', function($http) {  
  
  var loadDrugs = function() {  
    return $http({  
      url: 'http://152.96.56.31:40005/drugtracker/drug/drugs',  
      method: "GET",  
      dataType: "json",  
      headers: {'Content-Type': 'application/json'}  
    }).then(function(value){  
      localStorage.setItem('drugs', JSON.stringify(value.data));  
      return value;  
    }, function(errResult){  
      return errResult;  
    });  
  };  
  
  return {  
    getDrugTypes: getDrugTypes,  
    getDrugs: getDrugs,  
    loadDrugTypes: loadDrugTypes,  
    loadDrugs: loadDrugs  
  };  
});
```

Abbildung 55: Drug Factory

index.html

Das index.html ist wie ein normales html File aufgebaut, enthält zusätzlich noch die Angular Direktive im Body-Tag.

```
<body ng-app="starter">  
  <!--  
    The nav bar that will be updated as we navigate between views.  
  -->  
  <ion-nav-bar class="bar-stable">  
    <ion-nav-back-button>  
  </ion-nav-back-button>  
  </ion-nav-bar>  
  
  <!--  
    The views will be rendered in the <ion-nav-view> directive below  
    Templates are in the /templates folder (but you could also  
    have templates inline in this html file if you'd like).  
  -->  
  <ion-nav-view>  
  
  </ion-nav-view>  
</body>
```

Abbildung 56: Auszug aus index.html

**Dependency  
 Injection  
 (Angular Example)**

Dependency Injection ist ein Software Design Pattern, welches den Komponenten ermöglicht ihre Abhängigkeiten mit zugegeben, anstelle dass diese hart codiert werden.

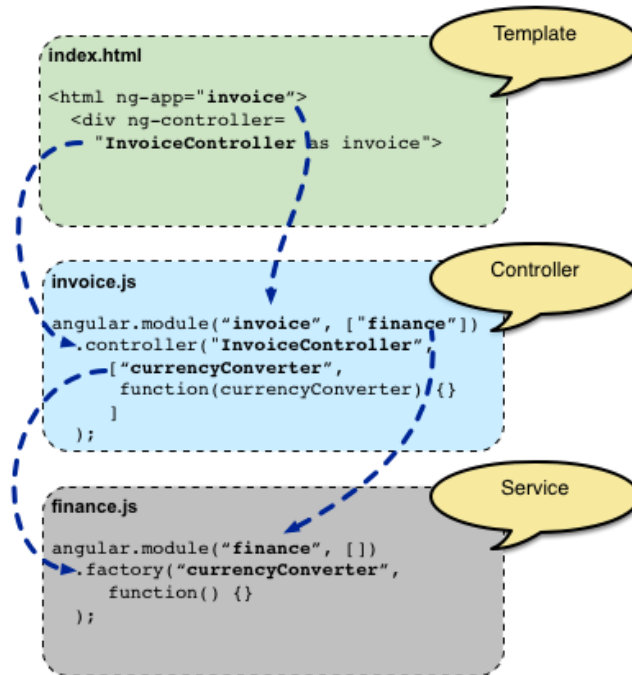


Abbildung 57: Dependency Injection

**Hinzufügen ver-  
 schiedener Platt-  
 formen**

Das Ionicframework unterstützt ios, Android oder Windowsphone. Durch Ausführen von 'ionic platform add [device]' kann die gewünschte Plattform hinzugefügt werden.

**Builden/Deploy der  
 Applikation**

Durch Ausführen des Befehls 'ionic build ios' wird aus dem www-Folder ein Xcode-Project erstellt. Dieses kann durch Öffnen des Projekts in Xcode auf ein i-Pad/iPhone deployed werden.

**Starten der Appli-  
kation (lokal)**

Folgende Voraussetzungen müssen gegeben sein:

- nodeJS muss installiert sein
- bower installieren `'npm install -g bower'`
- ionic installieren `'npm install -g ionic'`

Danach kann die Applikation wie folgt installiert werden:

1. Kopieren des Source-Codes (Clone des git-Repository)
2. Wechsel in den root Folder
3. `'npm install'` ausführen
4. `'bower install'` ausführen
5. Folgende Ordner erstellen: (`mkdir`)
  - a. `platforms`
  - b. `plugin`
6. `'ionic platform add ios'` Durch diesen Befehl wird die ios-Plattform hinzugefügt
7. `'ionic plugin add ionic-plugin-keyboard'` Dadurch kann das Device-Keyboard gesteuert werden.
8. Mittels `'ionic serve'` kann das Ganze im Browser angeschaut werden, es handelt sich hierbei um einen Live-Reload.
9. Durch Ausführen des Befehls `'ionic build ios'` kann das Ganze für die ios Plattform gebuildet werden.

## 5.2 Unterbau

**Beschreibung** Der Unterbau wird in dieser Phase gefertigt (extern) und danach montiert. Die Montage wird hier genauer beschrieben.

### 5.2.1 Montage

**Beschreibung** Die Montage des Unterbaus ist nicht sehr aufwendig, da nur die Steckdose und der Moosgummi zu befestigen sind.

**Nachbearbeitung Steckdose** Die Steckdose musste vor der Montage noch nachbearbeitet werden:

Vor der Bearbeitung:



Abbildung 58: Steckdose vor der Bearbeitung

Nach der Bearbeitung:

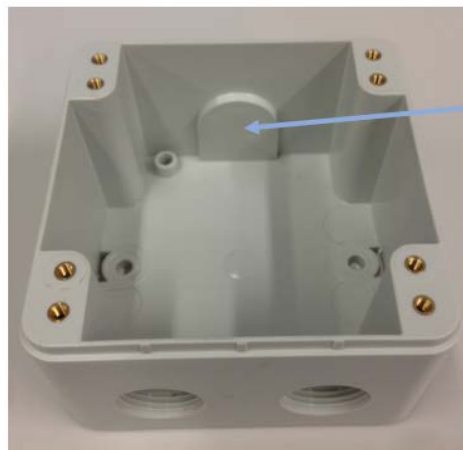


Abbildung 59: Steckdose nach der Bearbeitung

Bei der Bearbeitung wird der Boden der Steckdose eben abgefräst, damit danach die Steckdose ohne Probleme mit Schrauben, Scheiben und Muttern befestigt werden kann.

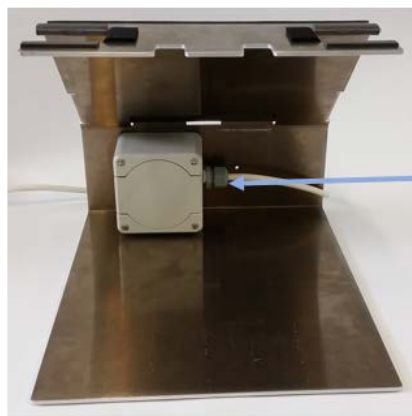
**Montage Steckdose** Es ist sicherzustellen, dass die Montage der Steckdose von einer **Fachperson** gemacht wird.

Zuerst muss eine Öffnung der Steckdose mit dem Schraubenzieher geöffnet werden. Danach schraubt man eine Kabelentlastung und Reduktion auf die Öffnung, die verhindert, dass das Kabel rutscht. Das Kabel wird durch diese Kabelentlastung gezogen.



Öffnung für Kabel (durch Schraubenzieher öffnen)

Abbildung 60: Öffnung für das Kabel



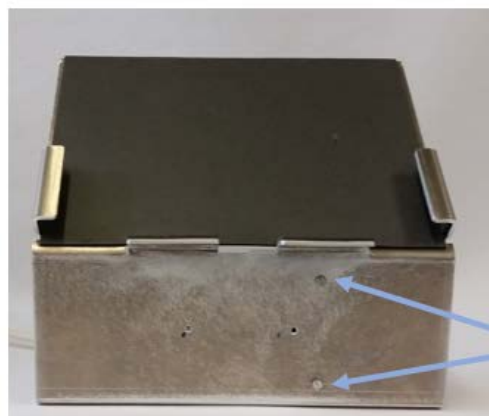
Kabelentlastung / Reduktion

Abbildung 61: Kabelentlastung / Reduktion

Die drei Drähte aus dem Kabel werden am vorgesehenen Ort (durch Farbe gekennzeichnet) mit dem Schraubenzieher befestigt.

Durch zwei Schrauben von aussen und je einer Scheibe und Mutter innen, wird die Steckdose am Unterbau befestigt. Es ist zu beachten, dass die Schrauben mit einem Anziehdrehmoment von 2.6 Nm angezogen werden, damit sich die Steckdose im Betrieb nicht löst.

Das Datenblatt „Anziehdrehmomente“ ist im Anhang ersichtlich



Befestigung Steckdose durch Schrauben

Abbildung 62: Befestigung Steckdose

### Montage Moosgummi

Damit das iPad, der Scanner und der Tisch nicht verkratzt werden, muss der Unterbau noch mit Moosgummi beklebt werden. Dazu muss der Moosgummi wie folgt zugeschnitten und auf den Unterbau geklebt werden:

1. Moosgummi Grundfläche: Dicke 2mm, 180mm X 250mm , Anzahl: 1
2. Lasche 1: Dicke 2mm, 9mm X 55mm , Anzahl: 2
3. Lasche 2: Dicke 2mm, 10mm X 55 mm , Anzahl: 2
4. Lasche unten: Dicke 2mm, 20mm X 55mm , Anzahl: 2
5. Boden: Dicke 2mm, 30mm X 30mm , Anzahl: 6
6. Scanner 1: Dicke 3mm, 30mm X 45mm , Anzahl:2
7. Scanner 2: Dicke 2mm, 50mm X 30mm , Anzahl: 4

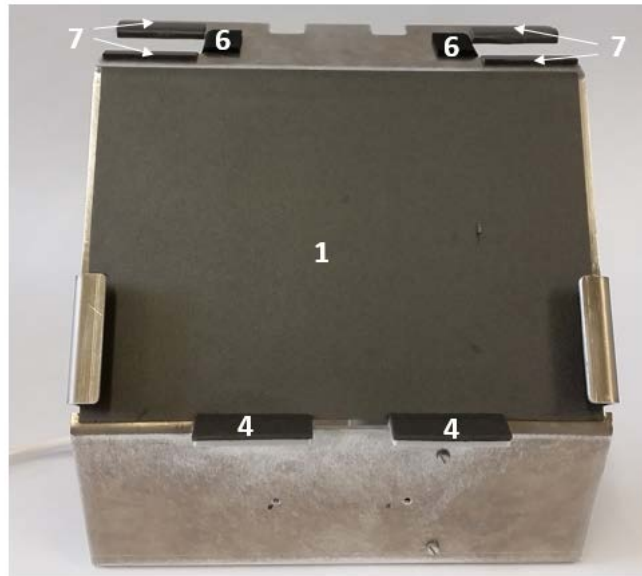


Abbildung 63: Befestigung Moosgummi



Abbildung 64. Befestigung Moosgummi Laschen





Abbildung 65: Befestigung Moosgummi Boden

**Installation Scanner für verschiedene Geräte**

Da der Scanner für mehrere Geräte gebraucht wird, muss er jedes Mal von neuem mit dem Gerät verbunden werden. Dazu ist es notwendig, dass Zahlencodes eingescannt werden. Diese werden einfacher halber auf die Frontfläche des Unterbaus aufgeklebt, damit sie immer griffbereit sind.

### 5.3 FMEA

**Beschreibung**

Um das Medikamententrackingsystem sicher zu machen, wurde mit einer FMEA die Risiken beurteilt und das System dementsprechend angepasst.

*Die FMEA für das System befindet sich im Anhang*

Für das System wurden diverse potentielle Fehler, deren Folgen und Ursache aufgeschrieben. Diese wurde je nach Auftreten, Bedeutung und Entdeckung beurteilt und bewertet. Somit wurde für jeden Fehler die Risiko- Prioritätszahl ermittelt. Diese zeigt an, wie gross das Risiko diese Fehlers ist:

RPZ* ... Risiko-Prioritätszahl	
hoch	<= 1000
mittel	<= 250
gering	<= 125
kein	= 1

Abbildung 66: Bedeutung Risiko- Prioritätszahl

Viele Risiken sind gemäss Risiko- Prioritätszahl gering. Es gibt aber trotzdem einzelne, bei denen das Risiko hoch ist. Diese sind nun unten genauer dargestellt.

**Hohe Risiken**

Fehlerort	Fehler	Fehlerfolge	RPZ	Getroffene Massnahmen
Unterbau	Steckdose defekt	iPad und Scanner können nicht mehr geladen werden	216	keine
Scanner	Akku leer	Scanner an die Steckdose gebunden	324	Ladezyklus in der Bedienungsanleitung festgehalten
Scanner	Scanner defekt	Einscannen/ Verwalten der Medis nicht möglich	324	Befestigung Scanner am Unterbau überarbeitet
Scanner	Verbindung zum iPad nicht möglich (kein Bluetooth)	Einscannen/ Verwalten der Medis nicht möglich	486	Konfiguration Codes auf Frontfläche des Unterbaus, in der Bedienungsanleitung beschreiben
Server	kein Netz vorhanden	kann keine neuen Daten vom Client beziehen	270	keine
Client	kein Wlan vorhanden	kann keine neuen Daten vom Server beziehen	270	Darstellung des Konktivitätsstatus im System
Client	App kann nicht installiert werden	Einscannen/ Verwalten der Medis nicht möglich	225	Absprache mit interner Informatik der REGA

Tabelle 34: Auszüge aus der FMEA

Die meisten Risiken konnte durch die getroffenen Massnahmen um einiges reduziert werden. Für zwei Fehler konnten leider keine geeigneten Massnahmen getroffen werden. Dort bleibt das Risiko weiterhin hoch.

## 5.4 Installations- und Bedienungsanleitung

---

**Beschreibung** Für das gesamte System ist eine Installations- und Bedienanleitung erstellt worden, in der genau beschrieben wird, wie der Scanner, die Software und die Hardware in Betrieb genommen werden. Zudem wird beschrieben, wie der Unterbau gewartet werden muss und wie die ganze Software funktioniert und zu bedienen ist.  
*Die gesamte Installations- und Bedienanleitung befindet sich im Anhang*

## 5.5 Fazit Realisierung

---

**Fazit** Der Unterbau, die Hardware und die Software sind fertig und es kann mit dem Testen begonnen werden.

## 6 Integrationstest

### 6.1 Erster Integrationstest

**Beschreibung** Der Integrationstest wird vom Projektteam durchgeführt. Er dient als Vorbereitung zum Abnahmetest und prüft das Zusammenspiel der einzelnen Softwarekomponenten (deren Integration in das Gesamtsystem).  
 Zum einen wird überprüft, ob die im Kapitel Analyse definierten Use Cases eingehalten werden und zum anderen, ob die Kriterien aus dem Pflichtenheft erfüllt sind.  
*Das ausführliche Testprotokoll befindet sich im Anhang*

**Ergebnis Test Use Cases** Die Übersicht zeigt, welche Use Cases erfüllt sind und welche noch nicht.

ID	Titel	erfüllt	Bemerkung
UC1	Benutzer anmelden	teils	Ausloggen nicht möglich
UC2.1	Scanbefehl kommt	ja	
UC2.2	Medikamentencode einscannen	ja	
UC2.3	System zeigt den Namen des Medikamentes an	ja	
UC2.4	Anzahl und Ablaufdatum erfassen	ja	
UC2.5	Benutzer bestätigt Einlagerung	ja	
UC2.6	System speichert die Einträge	ja	
UC2.7	Medikamentencode noch nicht im System vorhanden	ja	
UC2.8	Unbekanntes Medikament einlagern	teils	Falsche Meldung erscheint
UC2.9	Medikament mit abgelaufenem Datum einlagern	nein	Medikament kann normal eingelagert werden.
UC3.1	Befehl Medikamente zuweisen auswählen	ja	
UC3.2	Medikamentencode einscannen	ja	
UC3.3	System zeigt den Namen des Medikamentes an	ja	
UC3.4	Rettungsmodul auswählen	ja	
UC3.5	Benutzer bestätigt Zuweisung	ja	
UC3.6	System speichert die Zuweisung	ja	
UC3.7	Gescanntes Medikament läuft demnächst ab	nein	Medikament kann normal zugewiesen

			werden
UC3.8	Gescanntes Medikament hat Ablaufdatum erreicht oder überschritten	nein	Medikament kann normal zugewiesen werden
UC3.9	Mehrere gleiche Medikamente im Rettungsmodul enthalten	teils	Falls vergessen geht, ein Medikament auszutauschen, sind statt z.B. nur zwei, drei Medikamente im Modul
UC3.10	Medikament im Rettungsmodul hat abgelaufene Frist oder das Ablaufdatum erreicht	ja	
UC4.1	System erkennt Ablauffrist	ja	
UC4.2	Meldung per Mail	ja	
UC4.3	Übersicht Medikamente austauschen	ja	
UC5.1	Medikament bearbeiten	ja	
UC6.1	System verwalten	ja	
UC7.1	Patientendaten angeben (optional)	nein	Kann Opiat nicht kennzeichnen, kann keine Patientendaten eingeben
UC8.1	Bestellungen ausführen (optional)	nein	Kann keine Bestellung ausführen

Tabelle 35: Ergebnis Test Use Cases

**Ergebnis Test  
 Pflichtenheft  
 (Muss-Kriterien)**

Diese Übersicht zeigt, ob die Muss- Kriterien aus dem Pflichtenheft erfüllt sind oder nicht.

Nr.	Titel	erfüllt	Bemerkung
TM1	System scannt Medikamente (QR-, Barcode, Datamatrix)* ein	Ja	
TM2	System gibt Meldung, wenn Medikament abläuft	Ja	
TM3	System erkennt, welches Medikament in welchem Koffer (Rettungsmodul) ist	Ja	
TM4	System scannt Haltbarkeit und Medikamentenname ein	Ja	
TM5	Kompatibel für alle Basen	Ja	
TM6	Effiziente Bedienung durch REGA-Mitarbeiter (kein Mehraufwand zur aktuellen Lösung)	Ja	

Tabelle 36: Ergebnis Test Pflichtenheft (Muss- Kriterien)

**Ergebnis Test  
 Pflichtenheft (Optimierungs-  
 Kriterien)**

Diese Übersicht zeigt, ob die Optimierungs- Kriterien aus dem Pflichtenheft erfüllt sind.

Nr.	Titel	erfüllt	Bemerkung
TO1	Einfache Bedienoberfläche (ohne Schulung)	Ja	
TO2	System zeigt die Bestände der Medikamente an	Nein	Aus Zeitgründen nicht möglich
TO3	System führt die Bestellung aus	Nein	Aus Zeitgründen nicht möglich
TO4	Einfache Inbetriebnahme und Betrieb (durch Mitarbeiter möglich)	Ja	
TO5	Einfache Fertigung (Standardverfahren bevorzugen)	Ja	
T06	System läuft auf einem iPad	Ja	
T07	Hygiene muss gewährleistet sein (Desinfektionsmittel geeignet)	Ja	
T08	Kleines, kompaktes System (max. 1m x 1m x 1m)	Ja	
T09	System läuft zuverlässig (90%)	Ja	
KO1	Kostengünstig (innerhalb Kostendach)	Ja	
KO2	Standardkomponenten verwenden	Ja	

Tabelle 37: Ergebnis Test Pflichtenheft (Optimierungs- Kriterien)

**Fazit**

Es konnten beim ersten Test nicht alle Use Cases erfüllt werden. Somit wird die Software noch angepasst, bis alle erfüllt sind (Siehe Defect Log Integrationstest). Danach wird ein zweiter Integrationstest durchgeführt.

Die Muss- Kriterien aus dem Pflichtenheft sind alle erfüllt. Es konnten auch fast alle Optimierungs- Kriterien erfüllt werden, was bedeutet, dass das System den Wünschen der REGA entspricht.

## 6.2 Defect Log Integrationstest

**Defect Log Integrationstest**

Die folgende Tabelle enthält die gefundenen Defects. Dabei wurde jeder Defect vom Projektteam priorisiert. Die Defects im Status „Behoben“ konnten vom Informatikteam noch vor dem Abnahmetest gefixed werden. Diejenigen im Status „Verschoben“ können in einem späteren Zeitpunkt umgesetzt werden und sind nicht mehr Teil dieser Arbeit. *Der gesamte Defect Log ist im Anhang ersichtlich*

Defect Log Integrationstest		
Beschreibung	Priorität	Status
Benutzer kann Medikament auch zuweisen, ohne dass er ein zu ersetzendes ausgewählt hat	Hoch	Behoben
Wenn Medikament ersetzt wird, stimmt die Anzahl Medikamente im Modul nicht mehr (+1)	Hoch	Behoben
Ausloggen nicht möglich	Mittel	Behoben
Medikament mit Abgelaufenem Datum kann eingelagert/zugewiesen werden	Mittel	Behoben
Refresh nachdem Medikamente entfernt wurden funktioniert nicht richtig	Mittel	Verschoben
Wenn Medikament dem Kühlschrank zugewiesen wird, wird die gleiche Bestätigung ausgegeben („Medikament dem Lager hinzugefügt“)	Tief	Behoben
Scrollen sollte möglich sein. Etwas mühsam zum Eingeben. Zudem fehlt Bestätigung wenn angelegt	Tief	Behoben
Statusinfo immer auf 1 gesetzt beim Ersetzen. Sollte aber je nach Situation auf 2 gesetzt werden	Tief	Verschoben

Tabelle 38: Defect Log Integrationstest

## 6.3 Zweiter Integrationstest

**Beschreibung**

Da beim ersten Integrationstest nicht alle Use Cases erfüllt wurden, wurde die Software noch angepasst. Beim zweiten Integrationstest werden nur noch die nicht erfüllten Use Cases aus dem ersten Test getestet.

**Ergebnis Use Cases**

Beim zweiten Integrationstest konnten folgende Ergebnisse erzielt werden.

ID	Titel	erfüllt	Bemerkung
UC1	Benutzer anmelden	Ja	
UC2.8	Unbekanntes Medikament einlagern	Ja	
UC2.9	Medikament mit abgelaufenem Datum einlagern	Ja	
UC3.7	Gescanntes Medikament läuft demnächst ab	Ja	
UC3.8	Gescanntes Medikament hat Ablaufdatum erreicht oder überschritten	Ja	
UC3.9	Mehrere gleiche Medikamente im Rettungsmodul enthalten	Ja	
UC7.1	Patientendaten angeben (optional)	nein	Kann Opiat nicht kennzeichnen, kann keine Patientendaten eingeben
UC8.1	Bestellungen ausführen (optional)	nein	Kann keine Bestellung ausführen

Tabelle 39: Ergebnis Use Cases Test 2

**Fazit**

Beim zweiten Integrationstest wurden alle Use Cases erfüllt. Die Use Cases die optional sind, konnten aus Zeitgründen nicht mehr erfüllt werden.

**6.4 Fazit**

**Beschreibung**

Das System erfüllt alle Use Cases und Muss- Kriterien aus dem Pflichtenheft. Zudem konnten auch einige Optimierungskriterien umgesetzt werden. Somit kann das System der REGA präsentiert und mit dem Auftraggeber der Abnahmetest durchgeführt werden.



## 7 Abnahmetest

**Beschreibung** Der Abnahmetest wurde vor Ort auf der REGA-Basis in Kloten gemeinsam mit dem Auftraggeber durchgeführt. Der Auftraggeber war sehr zufrieden mit der Softwarequalität und hat entsprechend sein „Sign-Off“ für den Prototypen gegeben.

**Ergebnis** Der Auftraggeber hat alle Punkte abgenommen:

ID	Titel	Abgenommen	Datum
A1	Benutzer anmelden/ abmelden	X	30.11.2015
A2	Medikament einlagern	X	30.11.2015
A3	Medikament zuweisen	X	30.11.2015
A4	Administratorrechte	X	30.11.2015
A5	Ablauffrist melden	X	30.11.2015

Tabelle 40: Ergebnis Abnahmetest

Zudem hat er im Protokoll zu den Ergebnissen noch folgendes erwähnt:

„Die Vorgaben wurden korrekt abgelesen.

Das Produkt daraus ist gelungen und hat die Erwartungen bestens Erfüllt.

Das Produkt ist ausbaufähig und lässt viele zusätzliche Möglichkeiten für einen Ausbau zu.

Die Anwendung ist intuitiv und praxisorientiert.

Insgesamt eine gelungenes Produkt von dem ich mir in der Praxis viel verspreche“

*Das ausführliche Abnahmeprotokoll vom Auftraggeber befindet sich im Anhang.*

**Fazit** Der Auftraggeber war sehr zufrieden mit dem Ergebnis der Arbeit. Der Abnahmetest und das Projekt sind gelungen. Es wurden aber trotzdem noch ein paar Kleinigkeiten gefunden, die verbessert werden können. Diese sind in der Tabelle Defect Log Abnahmetest beschrieben.

## 7.1 Defect Log Abnahmetest

**Defect Log Abnahmetest** Die folgende Tabelle enthält die gefundenen Defects. Dabei wurde jeder Defect vom Projektteam anhand des Feedbacks des Auftraggebers priorisiert. Die Defects im Status „Behoben“ konnten vom Informatikteam noch vor dem Projektabschluss gefixed werden. Diejenigen im Status „Verschoben“ können in einem späteren Zeitpunkt umgesetzt werden und sind nicht mehr Teil dieser Arbeit.  
*Der gesamte Defect Log ist im Anhang ersichtlich*

Defect Log Abnahmetest		
Beschreibung	Priorität	Status
Die auszuwählenden Medikamente nach Datum sortieren	Hoch	Behoben
Bezeichnung der Lager von „blau/grün“ in „1/2“ ändern	Mittel	Behoben
Scrollen beim Adminbereich sollte möglich sein	Mittel	Behoben
Wenn ein Medikament aus Lager ausgewählt wird, alle mit gleichem Datum unter einer Zeile zusammenfassen (Anzahl)	Mittel	Verschoben
Bei Zuweisen eines Medikaments aus dem Kühlschrank muss das Ablaufdatum auf den Tag genau einstellbar sein	Mittel	Verschoben
Bei den abgelaufenen Medikamenten alle gleichen Medikamente mit gleichem Datum zusammenfassen	Mittel	Verschoben
Wie lösche ich Medikamente, Mailempfänger und Benutzer aus dem System?	Tief	Verschoben
Kühlschrank Medikamente sollten nur in den Kühlschrank eingelagert werden können	Tief	Verschoben
Bluetooth ein/aus automatisch	Tief	Verschoben

Tabelle 41: Defect Log Integrationstest

## 7.2 Fazit

**Fazit** Durch den Defect Log Abnahmetest konnte die Software noch auf Wunsch des Auftraggebers angepasst werden. Einige Punkte konnten leider aus Zeitgründen nicht mehr umgesetzt werden. Trotzdem ist unser Auftraggeber mehr als zufrieden und somit das Projekt gelungen.

## 8 Projektplanung

- Beschreibung** Das Projektvorgehen orientiert sich am V-Modell. Ein Gantt-Diagramm dient der Übersicht. Zum Schluss werden die Meilensteine aufgelistet und das generelle Vorgehen im Projekt beschrieben. Die umfangreiche und detaillierte Planung befindet sich auf Redmine und ist nicht Bestandteil dieser Dokumentation.
- Redmine** Für die gesamte Projektplanung, die Aufwanderfassung sowie das Task-Management wird Redmine eingesetzt.

### 8.1 Das V-Modell

- V-Modell** Vorgabe der Betreuer war die Projektabwicklung mittels V-Modell. Dies ist im Bereich Maschinenbau weit verbreitet und Standard. Es entspricht weitgehend dem Wasserfallmodell, welches im Bereich Informatik eher bekannt ist. Die Grafik unten zeigt die einzelnen Phasen des V-Modells.

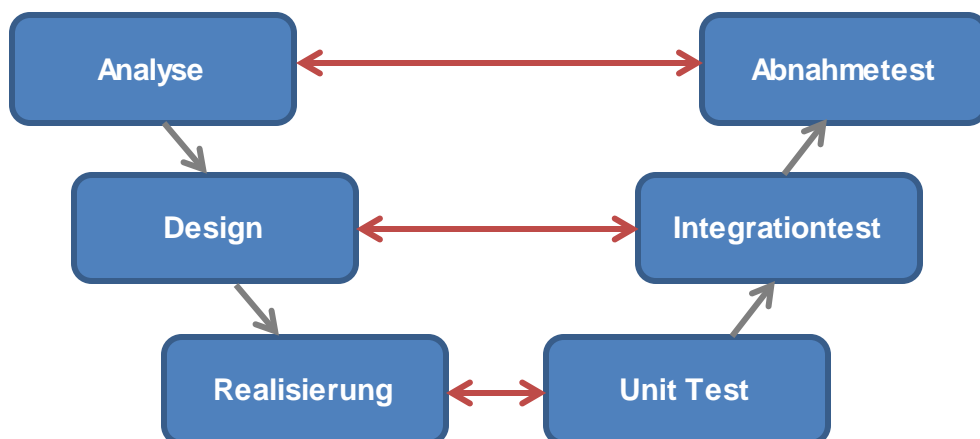


Abbildung 67: V-Modell

### 8.2 Gantt-Diagramm

- Gantt-Diagramm** Das gesamte Projekt wurde in die sechs Phasen des V-Modells aufgeteilt. Wobei zusätzlich noch die Phasen „Setup“ und „Projektabschluss“ hinzugefügt wurden. Die Setup-Phase dient dem Projektsetup. Darunter fallen unter anderem die Erstellung der Projektplanung und das Festlegen der Infrastruktur. Im Projektabschluss werden noch die letzten Projektarbeiten abgeschlossen, dazu gehört z.B. die Vervollständigung der gesamten Projektdokumentation.

Das Diagramm unten ist ein Auszug aus dem Gantt-Diagramm von Redmine. Der Übersicht halber wurden die Arbeitspakete entfernt. Die vollständige Projektplanung kann auf Redmine eingesehen werden.

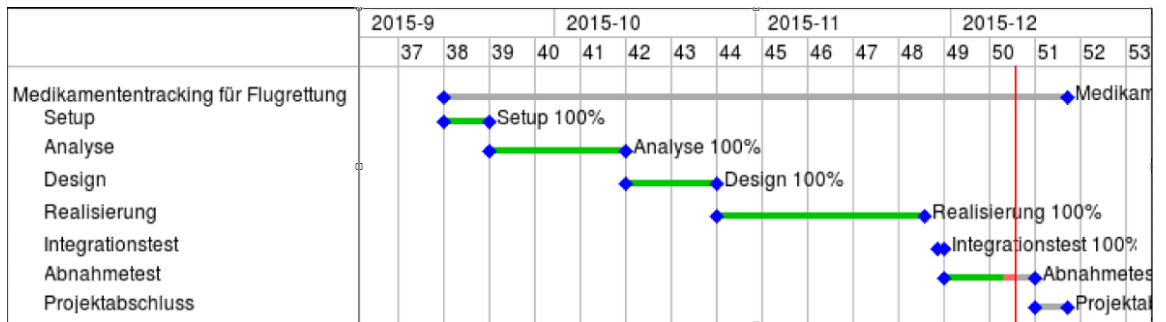


Abbildung 68: Gantt-Diagramm

## 8.3 Projektphasen

<b>Setup</b>	Grobe Projektplanung durchführen und Aufbau der Projektinfrastruktur wie: Projektmanagementtool aufsetzen (Redmine), Repositories anlegen (git hsr), Projektablage definieren und vorbereiten (Switchdrive)
<b>Analyse</b>	Aufgabenstellung Klären, Informationen beschaffen, Morphologischer Kasten erstellen, Varianten ausarbeiten und entscheiden. Festhalten der Softwareanforderungen, Erstellung Use Cases und Domain Model.
<b>Design</b>	Programmstruktur designen, Konstruktion ausarbeiten. Ausarbeitung der Applikationsarchitektur, Erstellung ERM und Technologieevaluation.
<b>Realisierung/Unit Testing</b>	Prototyp fertigen/montieren. Programmierarbeiten durchführen inklusive Unit Testing.
<b>Integrationstest</b>	Applikationsprototyp verknüpfen und testen. Bugfixing.
<b>Abnahmetest</b>	Abnahme Gesamtprototyp (Hard- und Software) durch REGA, kleinere Anpassungen falls notwendig.
<b>Projektabschluss</b>	Dokumente für Projektabschluss erstellen.
<b>Dokumentation</b>	Dokumentation der Arbeitsergebnisse.

## 8.4 Meilensteine

<b>Beschreibung</b>	Im Folgenden werden die offiziellen Meilensteine, die mit dem Auftraggeber vereinbart wurden und die projektinternen Meilensteine, aufgelistet.
---------------------	---

---

### 8.4.1 Offizielle Meilensteine

---

- MO1: 05.10.2015** Lasten- und Pflichtenheft unterschrieben durch Auftraggeber M. Reichenbach.
- MO2: 05.10.2015** Präsentation Varianten und Variantenentscheid mit M. Reichenbach.
- MO3: 30.11.2015** Abnahme Prototyp durch M. Reichenbach.
- MO4: 07.12.2015** Zweiter Abnahmetermin (wurde nicht benötigt).

---

### 8.4.2 Projektinterne Meilensteine

---

- M1: Abschluss Setup (20.09.2015)** Abschluss des Projektsetups.
- M2: Ende Analyse/Scope Freeze (11.10.2015)** Analysetätigkeiten abgeschlossen. Scope definiert und gefreezed.
- M3: Ende Design (25.10.2015)** Designaktivitäten abgeschlossen. Architektur festgelegt.
- M4: Ende Realisierung/Code Freeze (26.11.2015)** Realisierungstätigkeiten abgeschlossen. Keine grossen Codeanpassungen mehr, nur Bugfixes und kleinere, hoch priorisierte Wünsche vom Auftraggeber. Beginn der Testphasen.
- M4: Ende Integrationstest (29.11.2015)** Abschluss des Integrationstests.
- M5: Ende Abnahmetest (Deployment) (13.12.2015)** Abschluss des Abnahmetests. Produkt in produktiven Betrieb überführen.
- M6: Projektabschluss (18.12.2015)** Projektarbeiten abgeschlossen.

## 8.5 Meetings

---

- Weekly (Betreuer)** Während der gesamten Projektdauer findet jeweils am Montag um 9 Uhr ein wöchentliches Statusmeeting statt. Die Agenda wird jeweils bis spätestens 24 Stunden vor dem Meeting auf dem Redmine-Wiki hochgeladen. Das Protokoll des Meetings wird immer bis Ende der entsprechenden Woche ebenfalls auf dem Redmine-Wiki veröffentlicht.  
*Der Link zum Redmine befindet sich im Anhang*
- Weekly (Intern)** Ebenfalls wöchentlich finden projektinterne Besprechungen statt. Diese werden allerdings nur in Ausnahmefällen dokumentiert.
- Auftraggeber** Es finden drei Termine mit dem Auftraggeber statt, welche schon früh in der Projektphase definiert wurden (auf Wunsch des Auftraggebers). Auch die Protokolle dieser werden auf dem Redmine-Wiki veröffentlicht.

## 9 Zukunftsausblick und Optimierungen

**Beschreibung** Beim Design und Umsetzung des Prototypes wurde grossen Wert auf eine zukünftige Ausbaufähigkeit gelegt. Dies war auch explizit der Wunsch des Auftraggebers. Während der gesamten Projektdauer wurden die Mussziele des Auftraggebers stets priorisiert. Ein lauffähiger Prototyp, welcher diese Grundbedürfnisse abdeckt, stand im Vordergrund. Das Kapitel soll bei einer möglichen zukünftigen Weiterführung des Projektes, einen guten Einstiegspunkt bieten. Auch beim Unterbau ist noch Verbesserungspotenzial vorhanden, welches in diesem Kapitel ebenfalls beschrieben wird.

### 9.1 Software

- Beschreibung** Untenstehend wird ein Ausblick in die Zukunft gegeben und mögliche Optimierungen aufgelistet. Funktionen welche in diesem Umfang der Arbeit nicht umgesetzt werden konnten, werden auch aufgelistet und sollten bei allfälligen Weiterführung des Projektes unbedingt beachtet werden. Auch werden Defects des Prototypen angegeben, welche im Umfang dieser Arbeit nicht behoben werden konnten.
- CRUD (Jetzt)** Die Software unterstützt nur das Erfassen neuer Benutzer, Mailempfänger und Medikamententypen auf dem Tablet (nicht wie geplant im Browser). Bestehende können nur über manuelle Zugriff auf die Datenbank entfernt oder modifiziert werden.
- CRUD (Zukunft)** Die Applikation soll alle vier Funktionen von CRUD anbieten(Create, Read, Update und Delete), so dass auf allen Tabellen vollumfänglich gearbeitet werden kann.
- Security (Jetzt)** Die Sicherheit wurde aussen vor gelassen. Da es sich bei der Software um einen Prototypen handelt wurde dies vernachlässigt.
- Security (Zukunft)** Gewährleistung einer verschlüsselten Verbindung zwischen Client und Server. Auch müssen die Passwörter gehashed werden. Ein besonderes Augenmerk soll auch auf die Datenbank gelegt werden.
- Ablaufdatum überschreiben (Jetzt)** Das Ablaufdatum kann nicht manuell überschrieben werden. Die Datenbank ist bereits so ausgelegt, dass das Medikament als ‚expiryModifiable‘ markiert werden kann.
- Ablaufdatum überschreiben (Zukunft)** Bei gewissen Medikamenten muss es möglich sein das herstellergegebene Ablaufdatum zu überschreiben und zwar in diesem Augenblick, wenn das Medikament aus dem Kühlschrank genommen wird und somit die Kühlkette unterbrochen wird.
- Opiate (Jetzt)** Opiate werden nicht erkannt und können nicht speziell gekennzeichnet werden.
- Opiate (Zukunft)** Opiate müssen in der Datenbank als solche markiert werden. Das Betäubungsmittelgesetz gibt vor, dass Opiate zu jedem Zeitpunkt nach verfolgbar sein müssen. Aus diesem Grund muss das Medikament auf der Datenbank um ein Flag erweitert werden (Bsp. isOpiate). Bei dem Zuweisen eines Medikaments zu einem Rettungsmodul muss der Zuständige Mitarbeiter die Personalien des Patienten eintragen.

- Offline (Jetzt)** Die Offlinefunktionalität ist zurzeit nicht gewährleistet, jedoch werden die GET-Requests bereits in den localStorage gespeichert.
  
- Offline (Zukunft)** Die Flugrettung Schweiz wünscht eine offline Nutzung der Applikation für Notsituationen. Auch wenn kein Netz vorhanden ist, muss der Betrieb weiterhin gewährleistet sein.

## 9.2 Unterbau

- Optimierungen** Der Unterbau erfüllt seinen Zweck sehr gut, er könnte aber trotzdem noch ein wenig optimiert werden.
  
- Befestigung Scanner** Die Befestigung des Scanners kann so verbessert werden, in dem noch zusätzlich zwei Laschen nach oben gebogen werden. So kann der Scanner sicher in den Unterbau eingehängt werden. Dadurch hält er sicher und es besteht keine Gefahr, dass er beim Transport des Unterbaus herunterfällt (Massnahme aus FMEA). Da die Aussparung für den Scanner beim Prototypen recht knapp berechnet ist, wäre es empfehlenswert, die Aussparung etwas breiter zu machen. Auch in Zukunft ist es ratsam, die Aussparung mit Moosgummi zu überziehen, damit der Scanner nicht verkratzt.

**Prototyp:**

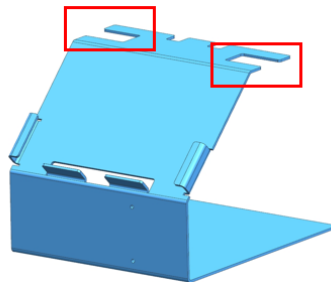


Abbildung 69: Prototyp Befestigung Scanner

**Optimierte Version:**

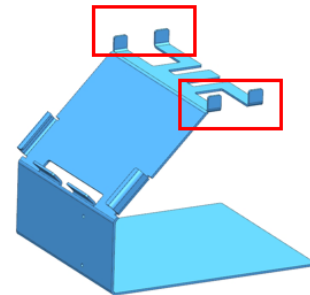


Abbildung 70: Optimierung Befestigung Scanner

- Aufhängung Ladekabel Scanner** Die Aussparungen für das Ladekabel des Scanners sind etwas zu wenig lang. Somit kann es vorkommen, dass das Kabel vom Unterbau fällt. Deshalb müssen die Aussparungen in Zukunft verlängert werden.

**Prototyp:**

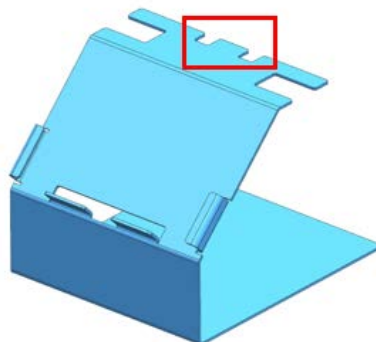


Abbildung 71: Prototyp Befestigung Ladekabel

**Optimierte Version:**

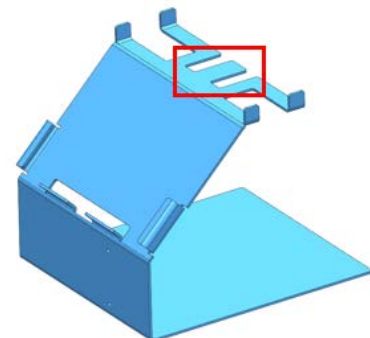


Abbildung 72: Optimierung Befestigung Ladekabel



**Detailzeichnungen** Von der optimierten Varianten wurden ebenfalls Detailzeichnungen erstellt. Auch bei der optimierten Variante gibt es zwei Varianten, mit oder ohne Bohrungen für die Steckdose. *Die Detailzeichnungen befinden sich im Anhang.*

### 9.3 Fazit Zukunftsausblicke und Optimierungen

---

**Fazit** Der Prototyp kann mit wenigen Optimierungen noch verbessert werden.  
Es ist eine Software entstanden, die in Zukunft noch vielseitig ergänzt werden kann.  
Somit ist der Auftraggeber an nichts gebunden und kann all seine Wünsche umsetzen.

## 10 Schlussfolgerungen, Ausblicke und Empfehlung

---

### 10.1 Zusammenfassung

---

<b>Ergebnisse</b>	Das Ergebnis der interdisziplinären Arbeit ist ein lauffähiges Gesamtsystem umgesetzt als Prototyp. Es beinhaltet die Software auf dem iPad und einen stabilen Unterbau inklusive Scanner. Der Prototyp ist einsatzbereit und ist so aufgebaut, dass er in Zukunft problemlos und einfach erweitert werden kann. Die Software empfängt die Barcodes vom Handscanner via Bluetooth und erkennt dadurch das gescannte Medikament. Medikamente können eingelagert und verschiedenen Rettungsmodulen zugewiesen werden. Vordefinierte Benutzer werden vor Ablauf einer Frist per Email informiert, dadurch ist ein frühzeitiges Erkennen von ablaufenden Medikamenten möglich.
<b>Bewertung</b>	Das Ziel dieser Arbeit, ein Prototyp zu entwickeln, der Medikamente auf die Haltbarkeit überprüft, ist gelungen. Der Auftraggeber war sehr zufrieden mit dem Resultat. Mit dieser Arbeit wurde gezeigt, dass es möglich ist, das Verfolgen von Medikamenten von A bis Z zu übernehmen und den Umgang mit Medikamenten zu vereinfachen. Durch das System können die Prozesse im Umgang mit Medikamenten in der Flugrettung vereinfacht werden.
<b>Empfehlungen</b>	Wie im Kapitel Optimierungen erwähnt, wird empfohlen, den Unterbau noch entsprechend anzupassen. Bezüglich der Software ist es sicher von Nöten, die Sicherheitsaspekte mit einzubauen, damit das System in Zukunft sicher und geschützt läuft. Je nach dem für was das Medikamentensystem alles eingesetzt werden soll, muss es noch dementsprechend erweitert und ergänzt werden.
<b>Ausblicke</b>	Wird die Software noch um einige Optionen ergänzt, wird es in Zukunft möglich sein, dass die REGA die gesamte Handhabung der Medikamente mit dem Medikamententrackingsystem durchführt. Es ist wahrscheinlich, dass nicht nur Medikamente den Lagern bzw. Rettungsmodulen zugewiesen und auf deren Ablaufdatum überprüft werden, sondern dass auch Bestellungen automatisch darüber abgewickelt werden. Zudem kann das Medikamententrackingsystem auch den Umgang mit Opiaten vereinfachen.

## 11 Literatur und Quellenverzeichnis

---

**Herkunft der Vorlage** Das Dokument wurde auf der Basis einer Vorlage für Technische Berichte erstellt. Die Vorlage ist ein Element des „Werkzeugkastens Technische Berichte“ der Hochschule für Technik Rapperswil. Sie orientiert sich an Prinzipien des Strukturierten Schreibens.

### 11.1 Literaturverzeichnis

---

- 1] Jesse Rosenbaum, Barcode, Berlin: Verlag Technik, 2000
- 2] „Texterkennung“ [Online]. Available:  
<https://de.wikipedia.org/wiki/Texterkennung>. [Zugriff Anfang Oktober 2015]
- 3] „mobit“ [Online]. Available:  
<http://www.mobit.ch/de/produkte/barcode-scanner/healthcare-barcode-scanner/>. [Zugriff Anfang Oktober 2015]
- 4] „Datalogic“ [Online]. Available:  
<http://www.datalogic.com/deu/lösungen/gesundheitswesen/krankenhäuser-labore-apotheken-so-4.html>. [Zugriff Anfang Oktober 2015]
- 5] „Scanmodul“ [Online]. Available:  
[http://www.scanmodul.com/#!nocount-software/ct7z\\_](http://www.scanmodul.com/#!nocount-software/ct7z_). [Zugriff Anfang Oktober 2015]
- 6] „Play Framework“ [Online]. Available:  
[https://en.wikipedia.org/wiki/Play\\_framework](https://en.wikipedia.org/wiki/Play_framework). [Zugriff Ende Oktober 2015]
- 7] „Spring Framework“ [Online] Available:  
[https://en.wikipedia.org/wiki/Spring\\_Framework](https://en.wikipedia.org/wiki/Spring_Framework). [Zugriff Ende Oktober 2015]
- 8] „Xamarin“ [Online]. Available:  
<https://xamarin.com>. [Zugriff Ende Oktober 2015]
- 9] „ionic“ [Online]. Available:  
<https://ionicframework.com>. [Zugriff Ende Oktober 2015]
- 10] „Java Plattform, Enterprise Edition“ [Online]. Available:  
[https://en.wikipedia.org/wiki/Java\\_Plattform,\\_Enterprise\\_Edition](https://en.wikipedia.org/wiki/Java_Plattform,_Enterprise_Edition). [Zugriff Ende Oktober 2015]
- 11] „Play“ [Online]. Available:  
<https://www.playframework.com/>. [Zugriff Ende Oktober 2015]
- 12] „Java Persistence API“ [Online]. Available:  
[https://de.wikipedia.org/wiki/Java\\_Persistence\\_API](https://de.wikipedia.org/wiki/Java_Persistence_API) [Zugriff Ende Oktober 2015]

## 12 Erklärung zur Urheberschaft

---

**Erklärung** Wir erklären hiermit, dass wir die vorliegende Arbeit ohne Hilfe Dritter angefertigt haben. Wir haben nur die Hilfsmittel benutzt, die wir angegeben haben. Gedanken, die wir aus fremden Quellen direkt oder indirekt übernommen haben, sind kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

**Ort/Datum** Rapperswil, 16. Dezember 2015

**Unterschrift**

Fabienne Germann

Sandro Muggli

Fabio Laib

---

## Anhang

---

- 2) Einleitung
  - a) Aufgabenstellung
  - b) Lastenheft [G]
  - c) Pflichtenheft [G]
  
- 3) Analyse
  - a) Datenblatt DataMatrix
  - b) Nutzwertanalysen [G]
  
- 4) Design
  - a) Berechnungen Schwerpunkt Basismodell [M]
  - b) Berechnungen Schwerpunkt Endmodell [M]
  - c) Materialtest [M]
  - d) EPDM- Datenblatt
  - e) Datenblatt Druckkraft
  - f) Diverse CAD Modelle [M]
  - g) Detailzeichnungen Endmodell [M]
  - h) Datenblatt Scanner
  
- 5) Realisieren
  - a) Offerte Unterbau
  - b) REGA Etat
  - c) FMEA [G]
  - d) Datenblatt Anziehdrehmoment
  - e) Installations- und Bedienanleitung [G]
  - f) Anleitung Scanner mit neuem Gerät verbinden [M]
  - g) Barcodekarten Medikamente [M]
  - h) Source Code Server [I]
  - i) Source Code Client [I]
  - j) Datenbankskripts [I]
  
- 6) Integrationstest
  - a) Testprotokoll Integrationstest 1 [M]
  - b) Testprotokoll Integrationstest 2 [M]
  - c) Defect Log Integrationstest [I]
  
- 7) Abnahmetest
  - a) Testprotokoll Abnahmetest [G]
  - b) Defect Log Abnahmetest [I]
  
- 8) Projektplanung
  - a) Redmine [G]
  - b) Redmine Excel Auszug [G]
  - c) Protokolle [G]
  
- 9) Zukunftsausblick und Optimierungen
  - a) CAD Modelle optimierter Unterbau [M]
  - b) Detailzeichnungen optimierter Unterbau [M]

---

## Autoren

---

Im Folgenden werden die Kapitel dem Verfasser zugeordnet. [I] steht für Informatik, [M] für Maschinenbau und [G] für beide Studienrichtungen.

<b>Abstract</b>	[G]
<b>2. Einleitung</b>	
2.1. Einführung ins Thema	[G]
2.2. Systemgrenze	[G]
2.3. Auszüge aus dem Pflichtenheft	[G]
<b>3. Analyse</b>	
3.1. Analyse der Aufgabe	
3.1.1. Stand der Technik Codes und Scanner	[G]
3.1.2. Marktübersicht bestehender Lösungen	[M]
3.1.3. Patentrecherche	[M]
3.2. Analyse der Softwareanforderungen	
3.2.1. Use Cases	[I]
3.2.2. Nicht funktionale Anforderungen	[I]
3.2.3. Domain Model	[I]
3.2.4. Zustandsdiagramm	[I]
3.2.5. Wireframes	[I]
3.3. Analyse der Konstruktionsanforderungen	
3.3.1. Funktionsstruktur	[G]
3.3.2. Lösungen der Teilfunktionen	[M]
3.3.3. Morphologischer Kasten und Vorentscheidung	[G]
3.4. Definitiver Variantenentscheid mit der REGA	[G]
3.5. Fazit Analyse	[G]
<b>4. Design</b>	
4.1. Software	
4.1.1. Application Architecture	[I]
4.1.2. Technologieevaluierung	[I]
4.1.3. Verwendete Technologien	[I]
4.1.4. Entity-Relationship-Modell	[I]
4.2. Unterbau	
4.2.1. Basismodell	[M]
4.2.2. Berechnungen	[M]
4.2.3. Konstruktion	[M]
4.2.4. Herstellung und Kosten	[M]
4.2.5. Ergebnis/ Fazit	[M]
4.3. Fazit Design	[G]
<b>5. Realisierung</b>	
5.1. Software	
5.1.1. Generell	[I]
5.1.2. Server	[I]
5.1.3. Client	[I]
5.2. Unterbau	
5.2.1. Montage	[M]
5.3. FMEA	[G]
5.4. Installations- und Bedienungsanleitung	[G]
5.5. Fazit Realisierung	[G]

<b>6. Integrationstest</b>	
6.1. Erster Integrationstest	[G]
6.2. Defect Log Integrationstest	[G]
6.3. Zweiter Integrationstest	[G]
6.4. Fazit	[G]
<b>7. Abnahmetest</b>	
7.1. Defect Log Abnahmetest	[G]
7.2. Fazit	[G]
<b>8. Projektplanung</b>	
8.1. Das V-Modell	[G]
8.2. Gantt-Diagramm	[G]
8.3. Projektphasen	[G]
8.4. Meilensteine	[G]
8.4.1. Offizielle Meilensteine	[G]
8.4.2. Projektinterne Meilensteine	[G]
8.5. Meetings	[G]
<b>9. Zukunftsausblick und Optimierungen</b>	
9.1. Software	[I]
9.2. Unterbau	[M]
9.3. Fazit Zukunftsausblicke und Optimierungen	[G]
<b>10. Schlussfolgerungen, Ausblicke und Empfehlung</b>	
10.1. Zusammenfassung	[G]