

Wegoptimierung für Gabelstapler bei Bico Matratzen

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2015

Autor(en): Philippe Naegeli und Raffael Ioannone
Betreuer: Dr. Daniel Keller
Projektpartner: Hilding Anders Switzerland AG, Schänis

Eigenständigkeitserklärung

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt sind oder mit dem Betreuer schriftlich vereinbart wurden,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben,
- dass wir keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt haben.

Rapperswil, 18. Dezember 2015



Raffael Ioannone



Philippe Naegeli

Inhaltsverzeichnis

1	Management Summary	5
1.1	Ausgangslage	5
1.2	Vorgehen	5
1.3	Ergebnisse	5
1.4	Weiteres Vorgehen	6
2	Einführung und Aufgabenstellung	7
2.1	Einführung	7
2.2	Aufgabenstellung	8
2.2.1	Beteiligte Parteien	8
2.2.2	Einführung	8
2.2.3	Erwartete Resultate	8
2.2.4	Hinweise	9
2.2.5	Termine	9
3	Requirements	10
3.1	Ist-Analyse	10
3.1.1	Übersicht	10
3.1.2	Produktionsprozess	12
3.1.3	Listen	13
3.1.4	Reporting	16
3.1.5	Bachelorarbeit	17
3.1.6	Optimierungsmöglichkeiten	18
3.2	Anforderungsspezifikation	20
3.2.1	User Stories	20
3.2.2	Use Cases	21
3.2.3	Mockups	26
3.3	Nichtfunktionale Anforderungen	29
3.3.1	Funktionalität	29
3.3.2	Zuverlässigkeit	29
3.3.3	Benutzbarkeit	29
3.3.4	Effizienz	29
3.3.5	Wartbarkeit	29

3.4	Domainanalyse	31
3.4.1	Domain-Modell	32
3.4.2	Übersetzungstabelle	34
4	Software Architecture Documentation	35
4.1	Kontextdiagramm	35
4.2	Verteilungsdiagramm	36
4.3	Schichtendiagramm	37
4.3.1	Server	37
4.3.2	Client	38
4.4	Komponenten	39
4.4.1	Server	39
4.4.2	Mobile Client	49
4.5	Kommunikation zwischen Server und Client	51
4.6	User Interface	51
4.7	Datenbank	53
4.8	Routen	55
5	Projektplanung	56
5.1	Qualitätssicherung	56
5.1.1	Continuos Integration	56
5.2	Projektablauf	57
6	Projektnachverfolgung	58
6.1	Einschränkungen	58
6.1.1	Back Button Android	58
6.1.2	Datenbankverbindung	58
6.1.3	Offene Use Cases	58
6.2	Metriken	59
6.3	Geleistete Zeit	59
6.4	Ausblick	60
Anhang A	Deployment	61
A.1	Export VBA	61
A.2	REST Service	62
A.3	Android App	63
A.3.1	Installation	63
A.3.2	Anpassung	63
Anhang B	Eigene Reviews	65
B.1	Philippe Naegeli	65
B.2	Raffael Ioannone	66
Anhang C	Risikoanalyse	67

Abstract

Die Firma Hilding Anders AG produziert in der Schweiz Matratzen unter den Marken BICO und HAPPY. Die Produktion arbeitet dabei mit Losgrösse 1. Für die Organisation der Fertigung werden mehrere Excel-Listen ausgedruckt und an die einzelnen Mitarbeiter verteilt.

Im Rahmen dieser Arbeit wurde für eine der Listen ein Prototyp einer Android-Applikation hergestellt, mit welchem es möglich ist die Daten auf einem Tablet zu betrachten. Dabei handelt es sich um die Liste der Gabelstaplerfahrer. Die Fahrer suchen die Kerne der Matratzen im Lager zusammen und gruppieren diese in einer vorgegebenen Reihenfolge auf den Paletten.

Nach dem Login in der Applikation wird dem Staplerfahrer eine Palette zugewiesen, welche er dann abarbeiten kann. Sobald er die Palette fertiggestellt hat, wird ihm automatisch eine neue Palette zugewiesen. Die Suche nach den Matratzenkernen wird dem Staplerfahrer mithilfe des Lagerstandortes und einem Bild des gesuchten Kerns erleichtert.

Für den Prototypen wurde eine native Android-App und ein dazugehöriger Server entwickelt. Der Server besitzt eine REST-Schnittstelle mit welcher die Produktionsdaten der Hilding Anders Switzerland AG an das Tablet übermittelt werden.

Kapitel 1

Management Summary

1.1 Ausgangslage

Die Hilding Anders Switzerland AG ist in der Schweiz, unter den Namen BICO und HAPPY, für die Produktion von qualitativ hochwertigen Matratzen bekannt. Durchschnittlich werden 400 Matratzen am Tag hergestellt, wobei jede davon durch Handarbeit entsteht. Die Produktionsdauer einer Matratze dauert maximal 2 Tage, was eine gute Produktionsplanung benötigt.

Die Produktionsplanung wird momentan in einem Excel-Dokument durchgeführt. Für jeden Arbeitsschritt wird dazu eine Liste mit den Bestellungen und zusätzlichen Angaben generiert. Diese werden dann an die verschiedenen Arbeitsstationen verteilt.

1.2 Vorgehen

Zuerst wurden mögliche Optimierungsmöglichkeiten eruiert. Mit dem Auftraggeber entschied man sich für das Ablösen der Papierliste der Staplerfahrer durch eine Android-App. Zusätzlich sollte das Finden der Matratzenkerne durch Bilder erleichtert werden. Die Applikation wurde mit Hilfe von agilen Planungsmethoden entwickelt und regelmässig dem Kunden präsentiert um ein optimales Ergebnis zu erhalten.

1.3 Ergebnisse

Während dieser Arbeit wurde eine Android-Applikation und das dazugehörige Back-End erstellt. Mit der App ist es möglich Matratzenkern-Listen anzusehen, abzarbeiten und neue Listen zu importieren:

Nr.	Art. Nummer	Name	Kern	Auftr. Nr.		
1	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
2	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
3	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
4	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
5	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
6	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
7	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
8	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
9	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️
10	H1347.22.7230	AirRelax 90x200 DJ Premio	HS20704.22	1700597361	X	🖼️

Abbildung 1.1: Screenshot Android

Zusätzlich wurde das bestehende Excel-Dokument so erweitert, dass die notwendigen Daten für die Listen möglichst einfach dem Back-End übermittelt werden können.

1.4 Weiteres Vorgehen

Es wurden während dieser Arbeit erst die Liste der Gabelstaplerfahrer auf eine elektronische Version migriert. Die restlichen Listen sowie zahlreiche weitere Optimierungen (siehe Kapitel 3.1.6) bieten nach wie vor eine Vielzahl an Erweiterungsmöglichkeiten.

Zudem wurde während des Projektes ersichtlich, dass eine direkte Verbindung zur produktiven Datenbank der Hilding Anders Switzerland AG nicht ohne weiteres möglich ist. Deshalb werden die Daten mithilfe eines Makros im Excel in die Datenbank des Back-Ends geschrieben. In Zukunft ist eine direkte Verbindung wünschenswert.

Kapitel 2

Einführung und Aufgabenstellung

In diesem Kapitel soll mit der Einführung ein Überblick über die Projektdokumentation aufgezeigt und mit der Aufgabenstellung der offizielle Auftrag dieses Projektes definiert werden.

2.1 Einführung

Dieses Dokument ist in die vier Teile Requirements, Software Architecture Documentation, Projektplanung und Projektnachverfolgung aufgeteilt.

In den **Requirements** wird die aktuelle Umgebung, mögliche Verbesserungsvorschläge und die Anforderungen des Kunden beschrieben.

Das Kapitel **Software Architecture Documentation** beschreibt die Details der erstellten Lösung. Hier werden die wichtigsten Bausteine des entwickelten Codes und wichtige Architekturentscheidungen beschrieben.

Projektplanung beschreibt die Vorgänge während der Studienarbeit und enthält Informationen über diverse Qualitätsmassnahmen und den Projektablauf

Im Kapitel **Projektnachverfolgung** werden die hergestellten Artefakte nochmals beschrieben. Zudem finden sich hier Informationen über offene Punkte in der neuen Lösung, sowie Anleitungen und ein Ausblick in die Zukunft.

2.2 Aufgabenstellung

2.2.1 Beteiligte Parteien

Beim Auftraggeber dieser Studienarbeit handelt es sich um den Industriepartner Hilding Anders Switzerland AG.

Ansprechpartner Auftraggeber:

- Arthur Schmucki, arthur.schmucki@hildinganders.com
- Pascal Kuster, pascal.kuster@hildinganders.com

Betreuer:

- Dr. Daniel Keller, daniel.keller@hsr.ch

Studenten:

- Philippe Naegeli, philippe.naegeli@hsr.ch
- Raffael Ioannone, raffael.ioannone@hsr.ch

2.2.2 Einführung

Die Hilding Anders Groupe ist der führende Bettenhersteller in Europa und Asien. Zur Hilding Anders Switzerland AG gehören die bekannten Marken BICO und HAPPY. Der Sitz befindet sich in Schänis im Kanton St. Gallen und produziert wird noch von Hand. Jede Matratze wird einzeln auf Mass gefertigt. Dies hat weg- und kommunikationsintensive Produktionsprozesse zur Folge. Heute wird die Produktion mit einem Excel-Dokument, welches am Morgen ausgedruckt wird, organisiert. Diese Lösung entspricht nicht mehr dem Stand der Technik und erlaubt es nicht flexibel und schnell auf Ereignisse zu reagieren.

2.2.3 Erwartete Resultate

Die Abläufe und damit die Produktion sollen vereinfacht werden. Dazu werden die folgenden Punkte umgesetzt:

- In einem ersten Schritt wird die Papier-Liste mit den Aufträgen für die Gabelstaplerfahrer durch eine Tablet-Lösung ersetzt. Längerfristig - voraussichtlich erst nach der Studienarbeit - wird angestrebt, alle bestehenden Papierlisten in der Produktion zu ersetzen.
- Der Staplerfahrer erhält Unterstützung beim Aufladen mit der Reihenfolge der Matratzenkerne.
- (Optional) Die Suche der Matratzenkerne wird durch eine Visualisierung des Aussehens vom gesuchten Kern vereinfacht.

2.2.4 Hinweise

Es besteht bereits ein Prototyp einer Lösung, welche die bestehenden Papierlisten ersetzen soll. Diese wurde in einer BA-Arbeit ein Semester zuvor erarbeitet. Diese kann als Fundament für die jetzige Arbeit verwendet werden.

2.2.5 Termine

Die Termine zu dieser Arbeit können der folgenden Seite entnommen werden:
<https://www.hsr.ch/Termine-Bachelor-und-Studiena.5142.0.html>

Kapitel 3

Requirements

3.1 Ist-Analyse

Die Ist-Analyse beschreibt, wie bis anhin in der Hilding Anders Switzerland AG gearbeitet wurde. Auf die einzelnen Produktionsschritte wird nur oberflächlich eingegangen. Diese sind ansonsten genauer in der Bachelorarbeit von Marcel Loop und Philipp Koster beschrieben. Zudem wurde analysiert, was in der Bachelorarbeit bereits erstellt wurde und was wir davon verwenden können.

Anschliessend wurden mögliche Optimierungsmöglichkeiten aufgestellt, welche auch für spätere Arbeiten berücksichtigt werden können.

3.1.1 Übersicht

Das bestehende System hat als zentrale Komponente ein Excel-File, welches mit VBA-Code erweitert wurde. Über den Code wird auf die Datenbank auf dem AS400 zugegriffen. Diese Datenbank wird durch das ERP-System Movex mit Informationen versorgt. Mit Hilfe des Scripts werden jeden Morgen die Kundenaufträge abgerufen, welche in zwei Tagen versendet werden müssen. Nach dem Daten-Import werden die Informationen mithilfe von anderen Funktionen im VBA-Code aufbereitet. Dadurch werden die einzelnen Listen, die in der Produktion benötigt werden, generiert. Zudem werden mit dem VBA-Code ein Reporting-Sheet ausgefüllt, die Etiketten für die Matratzen gedruckt und eine Access-Datenbank, welche in der späteren Produktion verwendet wird, mit den Kundenaufträgen versorgt. Zusammengefasst sieht man das Zusammenspiel all dieser Komponenten in der Abbildung 3.1.

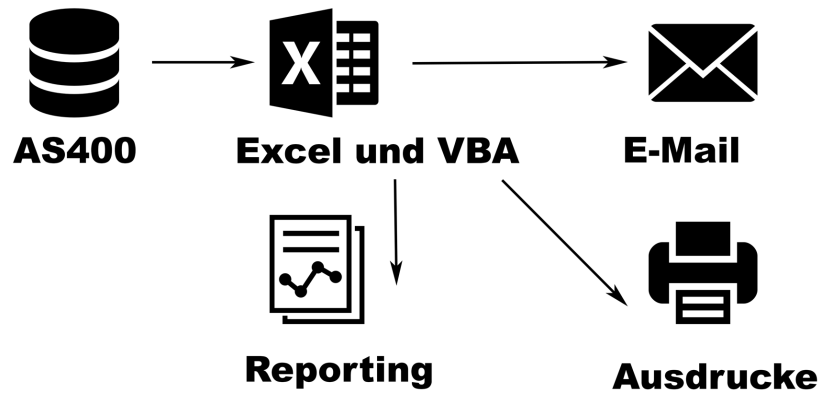


Abbildung 3.1: Ausgangslage, Bilderquelle: [3]

3.1.2 Produktionsprozess

Der Produktionsprozess wird in die folgenden Schritte unterteilt:

- Zusammenstellen der Kern-Paletten
- Reissverschlüsse zuschneiden
- Border zuschneiden
- Überzüge steppen
- Überzüge , Border und Reissverschlüsse zusammennähen
- Matratzenkerne, Etiketten und Überzüge zusammenfügen

Eine grafische Veranschaulichung dieser Schritte sieht man in der Abbildung 3.2:

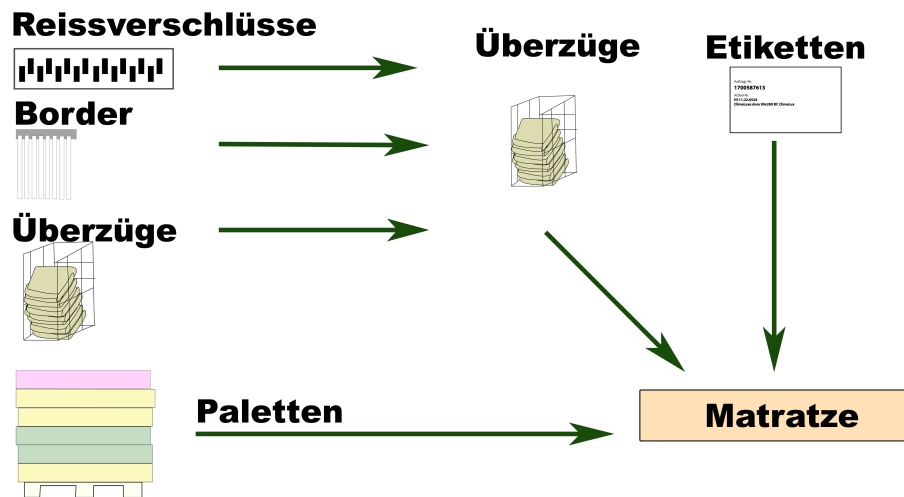


Abbildung 3.2: Übersicht Produktionsprozess

Lager: Die Produktion einer neuen Matratze beginnt im Lager einen Tag vor dem Zusammenfügen des Matratzenkerns mit dem Überzug. Die Gruppierung und Sortierung der einzelnen Kerne wird dem Gabelstaplerfahrer durch eine Excel-Liste vorgegeben. Dies ist wichtig, damit die Kerne in der selben Reihenfolge wie die Überzüge aufgeteilt und gestapelt sind.

Die Paletten werden so gruppiert, dass der Stapel nicht zu hoch wird. Um eine gute Stabilität zu gewährleisten, berücksichtigt die Sortierung, dass die grossen Kerne auf der Palette weiter unten zu finden sein müssen. Kerne welche der Kunde mit einer speziellen Grösse bestellt hat, sind in der Excel-Liste farblich gekennzeichnet und werden vor dem Aufladen auf den Stapel auf die gewünschte Grösse zugeschnitten.

Ist eine Palette abgeschlossen, wird diese in der Nähe der Warenliftes deponiert. Am nächsten Tag werden diese Paletten, sobald sie benötigt werden, einen Stock höher zum Zusammenfügen gebracht. Bevor dies geschieht, muss jedoch zuerst der Überzug und die Etikette hergestellt werden.

Weitere Schritte: Das Zuschneiden der Reissverschlüsse und der Border, sowie das Steppen/Zuschneiden der Überzüge wird jeweils parallel erledigt. Anschliessend werden diese in einem neuen Produktionsschritt zusammengenäht.

Als letzten Produktionsschritt werden die vom Vortag vorbereiteten Matratzenkerne mit den Überzügen zusammengebracht. Dabei wird die Matratze noch mit der Etikette versehen und transportbereit verpackt. Zum Schluss werden die Bestellungen der Kunden mittels Lastwagen ausgeliefert.

3.1.3 Listen

Durch das VBA-Script werden die folgenden Listen generiert:

- Gesamtliste
- Kernliste
- Einzugsliste
- Reissverschlussliste

Die Kernliste wird hauptsächlich im Lager verwendet. Die Reissverschlussliste wird beim Zuschneiden der Reissverschlüsse und der Border gebraucht. Die Gesamtliste wird für das Steppen, das Zuschneiden und das Sortieren der Überzüge benötigt. Die Einzugsliste wird beim Zusammenfügen der Überzüge und der Matratzenkerne verwendet.

Jede Liste enthält eine Nummer, welche pro Excel-Dokument definiert wird. Die Nummer ist wichtig, da es pro Tag mehrere Dokumente geben kann. Sie setzt sich aus dem Datum sowie einer Laufnummer zusammen.

3.1.3.1 Kernliste

Kernliste		Hüllen Kerne Baby HV	Pl.Nr	20151001002	3V4				
Pal.Nr	Art. Nr	Name	Auftr. Nr	Enddat.	PANR	Kern Nr.	Kern	Linie	
1	10	HC1107.22.6037	Curem Heaven Forte S200 90x200 DJ Cur em	1700598451	01.10.15	3751075	HS20689.22	Kern Curem Heaven Forte 19 2.0	HA
1	11	HC1107.22.6037	Curem Heaven Forte S200 90x200 DJ Cur em	1700598451	01.10.15	3751075	HS20689.22	Kern Curem Heaven Forte 19 2.0	HA
1	12	HC1106.24.6037	Curem Heaven S200 100x200 DJ Curem	1700595160	01.10.15	3751074	HS20212.24	Kern Curem Heaven 19	HA
1	13	HC1107.24.6037	Curem Heaven Forte S200 100x200 DJ Cur	1700595160	01.10.15	3751076	HS20689.24	Kern Curem Heaven Forte 19 2.0	HA
1	10	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597361	01.10.15	3751001	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	11	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597388	01.10.15	3751002	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	12	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597392	01.10.15	3751003	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	13	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598597	01.10.15	3751004	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	14	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	3751005	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	15	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	3751005	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	16	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	3751005	HS20704.22	Kern AirRelax 90x200x20 cm	HB
1	17	H1347.28.7230	AirRelax 180x200 DJ Premio	1700598629	01.10.15	3751012	HS20704.28	Kern AirRelax 180x200x20 cm	HB
1	18	H1347.28.7230	AirRelax 180x200 DJ Premio	1700598629	01.10.15	3751012	HS20704.28	Kern AirRelax 180x200x20 cm	HB
2	20	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597361	01.10.15	3751006	HS20704.26	Kern AirRelax 140x200x20 cm	HB
2	21	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597363	01.10.15	3751007	HS20704.26	Kern AirRelax 140x200x20 cm	HB
2	22	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597392	01.10.15	3751008	HS20704.26	Kern AirRelax 140x200x20 cm	HB
2	23	H1347.27.7230	AirRelax 160x200 DJ Premio	1700597353	01.10.15	3751009	HS20704.27	Kern AirRelax 160x200x20 cm	HB
2	24	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	3751010	HS20704.27	Kern AirRelax 160x200x20 cm	HB
2	25	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	3751010	HS20704.27	Kern AirRelax 160x200x20 cm	HB
2	26	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	3751011	HS20704.27	Kern AirRelax 160x200x20 cm	HB
2	27	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	3751011	HS20704.27	Kern AirRelax 160x200x20 cm	HB

Abbildung 3.3: Auszug Kernliste

Die Kernliste wird hauptsächlich von den Staplerfahrer verwendet, welche anhand dieser Liste die einzelnen Paletten für den kommenden Tag bereitstellen. Die Tabelle setzt sich aus folgenden Spalten zusammen:

Spalte	Beschreibung
Pal.	Die Nummerierung der Paletten beginnt bei jeder Kernliste und Produktionslinie bei 1.
Nr	Die Position des Kerns auf der Palette
Art. Nr	Die Artikelnummer der Matratze
Name	Der Name der Matratze
Auftr. Nr	Die Auftragsnummer der Bestellung aus dem Movex
Enddat.	Das Datum an dem die Bestellung ausgeliefert werden muss
PANR	Eine weitere Nummer aus dem Movex
Kern Nr	Die Kernnummer inklusive der codierten Grösse
Kern	Der Kernname inklusive der Grösse
Linie	Die Produktionslinie für die Aufteilung auf die Arbeitsstationen

Tabelle 3.1: Informationen aus Kernliste

Ein Gabelstaplerfahrer benötigt hauptsächlich die folgenden Spalten:

- Kern
- Kern Nr
- Auftr. Nr

3.1.3.2 Gesamtliste

Gesamtliste		Handmuster/Kissen HM	Pl.Nr	20151001002	3V4	EmailListe	
Pal. Nr	Art. Nr	Name	Auftr. Nr	Enddat.	Bezugsnr.	Kern	Linie
1	10	HC1107.22.6037	Curem Heaven Forte S200 90x200 DJ Cur. em	1700598451	01.10.15	6037 Kern Curem Heaven Forte 19 2.0	HA
1	11	HC1107.22.6037	Curem Heaven Forte S200 90x200 DJ Cur. em	1700598451	01.10.15	6037 Kern Curem Heaven Forte 19 2.0	HA
1	12	HC1106.24.6037	Curem Heaven S200 100x200 DJ Curem	1700595160	01.10.15	6037 Kern Curem Heaven 19	HA
1	13	HC1107.24.6037	Curem Heaven Forte S200 100x200 DJ Cur	1700595160	01.10.15	6037 Kern Curem Heaven Forte 19 2.0	HA
1	10	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597361	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	11	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597368	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	12	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597392	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	13	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598597	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	14	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	15	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	16	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	7230 Kern AirRelax 90x200x20 cm	HB
1	17	H1347.28.7230	AirRelax 180x200 DJ Premio	1700598629	01.10.15	7230 Kern AirRelax 180x200x20 cm	HB
1	18	H1347.28.7230	AirRelax 180x200 DJ Premio	1700598629	01.10.15	7230 Kern AirRelax 180x200x20 cm	HB
2	20	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597361	01.10.15	7230 Kern AirRelax 140x200x20 cm	HB
2	21	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597363	01.10.15	7230 Kern AirRelax 140x200x20 cm	HB
2	22	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597392	01.10.15	7230 Kern AirRelax 140x200x20 cm	HB
2	23	H1347.27.7230	AirRelax 160x200 DJ Premio	1700597353	01.10.15	7230 Kern AirRelax 160x200x20 cm	HB
2	24	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	7230 Kern AirRelax 160x200x20 cm	HB
2	25	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	7230 Kern AirRelax 160x200x20 cm	HB
2	26	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	7230 Kern AirRelax 160x200x20 cm	HB
2	27	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	7230 Kern AirRelax 160x200x20 cm	HB
3	30	H1146.21.7219	>> Topper Sky 80x200 896 Probiotex W1500w	1700599012	01.10.15	7219 Kern Topper EvoPore 200x90x4	HB
3	31	H1146.21.7219	>> Topper Sky 80x200 896 Probiotex W1500w	1700599012	01.10.15	7219 Kern Topper EvoPore 200x90x4	HB
3	32	H1349.22.7231	>> ErgoRelax 90x190 DJ Selection	1700597383	01.10.15	7229 Kern ErgoRelax 90x200x22 cm	HB
3	33	H1346.12.7229	>> ErgoRelax 90x190 DJ Selection	1700598608	01.10.15	7229 Kern ErgoRelax 90x200x16 cm	HB
3	34	H1346.12.7229	>> ErgoRelax 90x190 DJ Selection	1700598608	01.10.15	7229 Kern ErgoRelax 90x200x16 cm	HB
3	35	H1363.23.7232	>> ClimaStar 90x200 DJ Punti weiss	1700598637	01.10.15	7232 Kern ClimaStar 90x200x21.6 cm	HB
3	36	H1363.23.7232	>> ClimaStar 90x200 DJ Punti weiss	1700598637	01.10.15	7232 Kern ClimaStar 90x200x21.6 cm	HB
3	37	H1146.26.7219	>> Topper Sky 140x200 896 Probiotex W1500w	1700598168	01.10.15	7219 Kern Topper EvoPore 200x160x4	HB
3	38	H1349.27.7231	>> ClimaRelax 160x200 DJ Punti weiss	1700598615	01.10.15	7231 Kern ClimaRelax 160x200x22 cm	HB
1	10	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15	0	HM
1	11	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15	0	HM
1	12	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15	0	HM
1	13	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15	0	HM

Abbildung 3.4: Auszug Gesamtliste

Die Gesamtliste enthält neben den Angaben zur Bestellungen und zum Kern auch Informationen zum Überzug.

3.1.3.3 Einzugsliste

Einzugsliste		Hüllen Kerne Baby HV	Pl.Nr	20151001002	3V4	EmailListe	
Pal. Nr	Art. Nr	Name	Auftr. Nr	Enddat.	Kern Nr.	Kern	Linie
1	10	HC1107.22.6037	Curem Heaven Forte S200 90x200 DJ Cur. em	1700598451	01.10.15	HS20689.22 Kern Curem Heaven Forte 19 2.0	HA
1	11	HC1107.22.6037	Curem Heaven Forte S200 90x200 DJ Cur. em	1700598451	01.10.15	HS20689.22 Kern Curem Heaven Forte 19 2.0	HA
1	12	HC1106.24.6037	Curem Heaven S200 100x200 DJ Curem	1700595160	01.10.15	HS20112.24 Kern Curem Heaven 19	HA
1	13	HC1107.24.6037	Curem Heaven Forte S200 100x200 DJ Cur	1700595160	01.10.15	HS20689.24 Kern Curem Heaven Forte 19 2.0	HA
1	10	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597361	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	11	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597368	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	12	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597392	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	13	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598597	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	14	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	15	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	16	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	HS20704.22 Kern AirRelax 90x200x20 cm	HB
1	17	H1347.28.7230	AirRelax 180x200 DJ Premio	1700598629	01.10.15	HS20704.28 Kern AirRelax 180x200x20 cm	HB
1	18	H1347.28.7230	AirRelax 180x200 DJ Premio	1700598629	01.10.15	HS20704.28 Kern AirRelax 180x200x20 cm	HB
2	20	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597361	01.10.15	HS20704.26 Kern AirRelax 140x200x20 cm	HB
2	21	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597363	01.10.15	HS20704.26 Kern AirRelax 140x200x20 cm	HB
2	22	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597392	01.10.15	HS20704.26 Kern AirRelax 140x200x20 cm	HB
2	23	H1347.27.7230	AirRelax 160x200 DJ Premio	1700597353	01.10.15	HS20704.27 Kern AirRelax 160x200x20 cm	HB
2	24	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	HS20704.27 Kern AirRelax 160x200x20 cm	HB
2	25	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	HS20704.27 Kern AirRelax 160x200x20 cm	HB
2	26	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	HS20704.27 Kern AirRelax 160x200x20 cm	HB
2	27	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	HS20704.27 Kern AirRelax 160x200x20 cm	HB
3	30	H1146.21.7219	>> Topper Sky 80x200 896 Probiotex W1500w	1700599012	01.10.15	HS20195.22 Kern Topper EvoPore 200x90x4	HB
3	31	H1146.21.7219	>> Topper Sky 80x200 896 Probiotex W1500w	1700599012	01.10.15	HS20195.22 Kern Topper EvoPore 200x90x4	HB
3	32	H1349.22.7231	>> ClimaRelax 90x200 DJ Punti weiss	1700597383	01.10.15	HS20702.22 Kern ClimaRelax 90x200x22 cm	HB
3	33	H1346.12.7229	>> ErgoRelax 90x190 DJ Selection	1700598608	01.10.15	HS20705.22 Kern ErgoRelax 90x200x16 cm	HB
3	34	H1346.12.7229	>> ErgoRelax 90x190 DJ Selection	1700598608	01.10.15	HS20705.22 Kern ErgoRelax 90x200x16 cm	HB
3	35	H1363.23.7232	>> ClimaStar 90x200 DJ Punti weiss	1700598637	01.10.15	HS20708.22 Kern ClimaStar 90x200x21.6 cm	HB
3	36	H1363.23.7232	>> ClimaStar 90x200 DJ Punti weiss	1700598637	01.10.15	HS20708.22 Kern ClimaStar 90x200x21.6 cm	HB
3	37	H1146.26.7219	>> Topper Sky 140x200 896 Probiotex W1500w	1700598168	01.10.15	HS20195.27 Kern Topper EvoPore 200x160x4	HB
3	38	H1349.27.7231	>> ClimaRelax 160x200 DJ Punti weiss	1700598615	01.10.15	HS20702.27 Kern ClimaRelax 160x200x22 cm	HB
1	10	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15		HM
1	11	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15		HM
1	12	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15		HM
1	13	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15		HM

Abbildung 3.5: Auszug Einzugsliste

Die Einzugsliste ähnelt sehr der Kernliste und wird vor allem beim Zusammenführen des Überzugs mit dem Kern verwendet.

3.1.3.4 Reissverschlussliste

Reissverschluss		Hüllen Kerne Baby HV		Pl.Nr	20151001002	3V4 EmailListe		
Pal. Nr	Art. Nr	Name	Auftr. Nr	Enddat.	RV Nr.	RV-Bezeichnung	Linie	
1	10	HC1107.22.8037	Curem Heaven Forte S200 90x200 DJ Cur. em	1700598451	01.10.15			HA
1	11	HC1107.22.8037	Curem Heaven Forte S200 90x200 DJ Cur. em	1700598451	01.10.15			HA
1	12	HC1106.24.6037	Curem Heaven S200 100x200 DJ Curem	1700595160	01.10.15			HA
1	13	HC1107.24.6037	Curem Heaven Forte S200 100x200 DJ Cur	1700595160	01.10.15			HA
1	10	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597361	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	11	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597361	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	12	H1347.22.7230	AirRelax 90x200 DJ Premio	1700597362	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	13	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598597	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	14	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	15	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	16	H1347.22.7230	AirRelax 90x200 DJ Premio	1700598619	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
1	17	H1347.28.7230	AirRelax 160x200 DJ Premio	1700598629	01.10.15	HS31081	RV S60 tb.m. 160x200cm r.weiss	HB
1	18	H1347.28.7230	AirRelax 160x200 DJ Premio	1700598629	01.10.15	HS31081	RV S60 tb.m. 160x200cm r.weiss	HB
2	20	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597361	01.10.15	HS31077	RV S60 tb.m. 140x200cm r.weiss	HB
2	21	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597363	01.10.15	HS31077	RV S60 tb.m. 140x200cm r.weiss	HB
2	22	H1347.26.7230	AirRelax 140x200 DJ Premio	1700597392	01.10.15	HS31077	RV S60 tb.m. 140x200cm r.weiss	HB
2	23	H1347.27.7230	AirRelax 160x200 DJ Premio	1700597353	01.10.15	HS31079	RV S60 tb.m. 160x200cm r.weiss	HB
2	24	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	HS31079	RV S60 tb.m. 160x200cm r.weiss	HB
2	25	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598618	01.10.15	HS31079	RV S60 tb.m. 160x200cm r.weiss	HB
2	26	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	HS31079	RV S60 tb.m. 160x200cm r.weiss	HB
2	27	H1347.27.7230	AirRelax 160x200 DJ Premio	1700598662	01.10.15	HS31079	RV S60 tb.m. 160x200cm r.weiss	HB
3	30	H1146.21.7219	>> Topper Sky 80x200 896 Probiotex WI500w	1700599012	01.10.15	HS31052	RV S60 tb.m. 80x200cm r.weiss	HB
3	31	H1146.21.7219	>> Topper Sky 80x200 896 Probiotex WI500w	1700599012	01.10.15	HS31052	RV S60 tb.m. 80x200cm r.weiss	HB
3	32	H1349.27.7231	ClimaRelax 90x200 DJ Punti weiss	1700597363	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
3	33	H1346.12.7229	>> ErgoRelax 90x190 DJ Selection	1700598608	01.10.15	HS31054	RV S60 tb.m. 90x190cm r.weiss	HB
3	34	H1346.12.7229	>> ErgoRelax 90x190 DJ Selection	1700598608	01.10.15	HS31054	RV S60 tb.m. 90x190cm r.weiss	HB
3	35	H1363.22.7232	ClimaStar 90x200 DJ Punti weiss	1700598637	01.10.15	HS31053	RV S60 tb.m. 90x200cm r.weiss	HB
3	36	H1363.22.7232	ClimaStar 90x200 DJ Punti weiss	1700598637	01.10.15	HS31072	RV S60 tb.m. 90x200cm r.weiss	HB
3	37	H1146.28.7219	>> Topper Sky 140x200 896 Probiotex WI500w	1700598168	01.10.15	HS31077	RV S60 tb.m. 140x200cm r.weiss	HB
3	38	H1349.27.7231	ClimaRelax 160x200 DJ Punti weiss	1700598615	01.10.15	HS31079	RV S60 tb.m. 160x200cm r.weiss	HB
1	10	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15			HM
1	11	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15			HM
1	12	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15			HM
1	13	H4138	Ausstellungskissen 40x50 cm blau mit Logo	1700598083	01.10.15			HM

Abbildung 3.6: Auszug Reissverschlussliste

Anhand der Reissverschlussliste werden bei den Arbeitsstationen die Reissverschlüsse und die Border für die Matratzen zugeschnitten.

3.1.4 Reporting

Tagesproduktion Einziehen		Tag: 01.10.15																																				
Pl. Nr.	20151001002	Liefertag: 05.10.15																																				
30.09.2015 - 14:13:27																																						
Art	Bezeichnung																																					
HA	Abnehmbar	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
HE	Einzug	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
HS	1 RV	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
HR	2 RV	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
HT	CL Finesse	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
HV	Hüllen/Kerne	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16																					
HB	Sonder	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																						
MI	Migros	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
IN	Interio	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
HO		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		

Abbildung 3.7: Auszug Reporting

Neben den einzelnen Listen wird im Excel-Dokument ein Reporting über die produzierten Matratzen erstellt.

3.1.5 Bachelorarbeit

Im Frühlingssemester wurde von Marcel Loop und Philipp Koster die Bachelorarbeit „Prozessoptimierung mit mobilen Devices“ [5] durchgeführt. Ziel dieser Arbeit war es die Analyse der bestehenden Excel-Listen und die Herstellung eines Prototypen, mit welchem die verschiedenen Listen auf einem Tablet angezeigt werden können.

3.1.6 Optimierungsmöglichkeiten

In diesem Kapitel werden alle potenziellen Optimierungsmöglichkeiten in der Matratzenproduktion bei der Hilding Anders Switzerland AG beschrieben:

3.1.6.1 Papierliste

Als erste Optimierungsmöglichkeit bietet sich die digitale Anzeige der Listen aus der Excel-Datei an. Mit einer mobilen Applikation kann vor allem die Kommunikation, welche für das Aufteilen der Paletten unter den Staplerfahrer benötigt wird, vereinfacht werden.

3.1.6.2 Matratzenkern-Stapel

Optimierung im Bereich der Matratzenkern-Stapel bieten sich darin an, dass zusätzlich zu den aktuellen Anforderungen beim Stapeln der Lagerort der Kerne mit einberechnet wird.

3.1.6.3 Kern-Findung

Der Standort der einzelnen Kerne im Lager ist aktuell nicht wirklich bekannt. Ein neuer Staplerfahrer hat deshalb grosse Probleme sich im grossen Lager zurechtzufinden. Mittels Bilder der Kerntypen und dem zugewiesenen Lagerort kann dieses Problem merklich verbessert werden.

3.1.6.4 Wege der Gabelstapler

Momentan werden die Wege der Gabelstapler nur bedingt analysiert. Sobald die Lagerorte der Matratzenkerne bekannt sind, können optimale Routen, für das Erstellen der Paletten, berechnet werden.

3.1.6.5 Priorisierung und Stornierung

Gibt es eine Änderung (Priorisierung bzw. Stornierung) zu einer laufenden Bestellung, muss zwischen den verschiedenen Arbeitsstationen telefoniert werden, was zum einen viel Zeit braucht und zum andern sehr umständlich ist. Mittels einer mobilen Applikation, welche die weiteren Listen aktualisiert, können die Kommunikationswege erheblich verkürzt werden.

3.1.6.6 Rollenwechsel

Beim Zuschneiden der Überzüge müssen die Stoff-Rollen ständig gewechselt werden. Dies ist mühsam und kostet viel Zeit. Eine Optimierungsmöglichkeit wäre, alle Überzüge welche die selbe Rolle benötigen, nacheinander zuzuschneiden. Damit kann einiges an Zeit eingespart werden.

3.1.6.7 Platzierung der Kerne

Aktuell werden die am häufigsten verwendeten Kerne an gut erreichbaren Positionen gelagert. Besser wäre es, die Kerne so zu lagern, dass für die Erstellung der Palette ein möglichst kleiner Weg zurückgelegt werden muss.

3.1.6.8 Lastwagenroute

Für eine weitere Optimierungsmöglichkeit bieten sich die Routen der Lastwagen für die Auslieferung der Bestellungen an. Hier könnte man zum Beispiel die Reihenfolge beim Beladen des Lasters verbessern.

3.1.6.9 Matratzen Tracking

Um die Kernfindung zusätzlich optimieren zu können würde sich ein Tracking von den einzelnen Matratzen mittels einer Technologie wie RFID bzw. NFC. Mit dieser Lösung können die Matratzenkerne im Lager einfach aufgespürt werden und somit können auch die Distanzen für eine Palette viel besser berechnet werden.

3.2 Anforderungsspezifikation

3.2.1 User Stories

3.2.1.1 Umsetzung Papierliste

Als Gabelstaplerfahrer, möchte ich die täglichen Kernlisten digital auf einem Tablet ansehen und abschliessen können, damit ich flexibler bin und den Status der anderen Bestandteile der Matratze einfacher abklären kann.

3.2.1.2 Optimierung Matratzenkern Stapel

Als Staplerfahrer, möchte ich die Kerne, welche ich aufladen muss, bereits in der korrekten Reihenfolge in der App sehen, damit mein Stapel stabil ist.

3.2.1.3 Optimierung Kernfindung

Als Gabelstaplerfahrer, der nach einem bestimmten Matratzenkern sucht, möchte ich ein Bild des Kerns in der App sehen, damit ich weiss wie dieser aussieht und ich ihn schneller finden kann.

3.2.2 Use Cases

Die Use Cases werden mit folgender Priorität durchgeführt:

Use Case	Geplant	Priorität
UC01: Log in	Ja	1
UC02: Kernliste generieren	Ja	1
UC03: Palette anfordern	Ja	1
UC04: Palette abschliessen	Ja	1
UC05: App synchronisieren	Ja	1
UC06: Status aller Kernlisten abfragen	Ja	1
UC07: Ansehen des Sektors	Optional	2
UC08: Ansehen des Kernbildes	Optional	2
UC09: Kern abschliessen	Optional	3
UC10: Kern stornieren	Optional	3
UC11: Position eines Kernes ändern	Optional	4

Tabelle 3.2: Priorisierung Use Cases

3.2.2.1 Aktoren

Die verschiedenen Use Cases werden von den vier verschiedenen Aktoren Benutzer, Gabelstaplerfahrer, Meister und Movex initiiert oder abgearbeitet:

Benutzer: Einem Benutzer ist es möglich, sich über die Android Applikation anzumelden und dort diverse Informationen anzusehen. Meister und Gabelstaplerfahrer sind ebenfalls Benutzer.

Gabelstaplerfahrer: Der Gabelstaplerfahrer kann sich aktuelle Kernlisten ansehen und sich neue Palette zuweisen lassen.

Meister: Dem Meister ist es möglich neue Daten zu importieren. Ausserdem ist er auch ein Gabelstaplerfahrer.

Movex: Das Movex System ist dafür zuständig entsprechende Anfragen, zum Importieren der Produktionsdaten, zu beantworten.

3.2.2.2 Use Case Diagramm

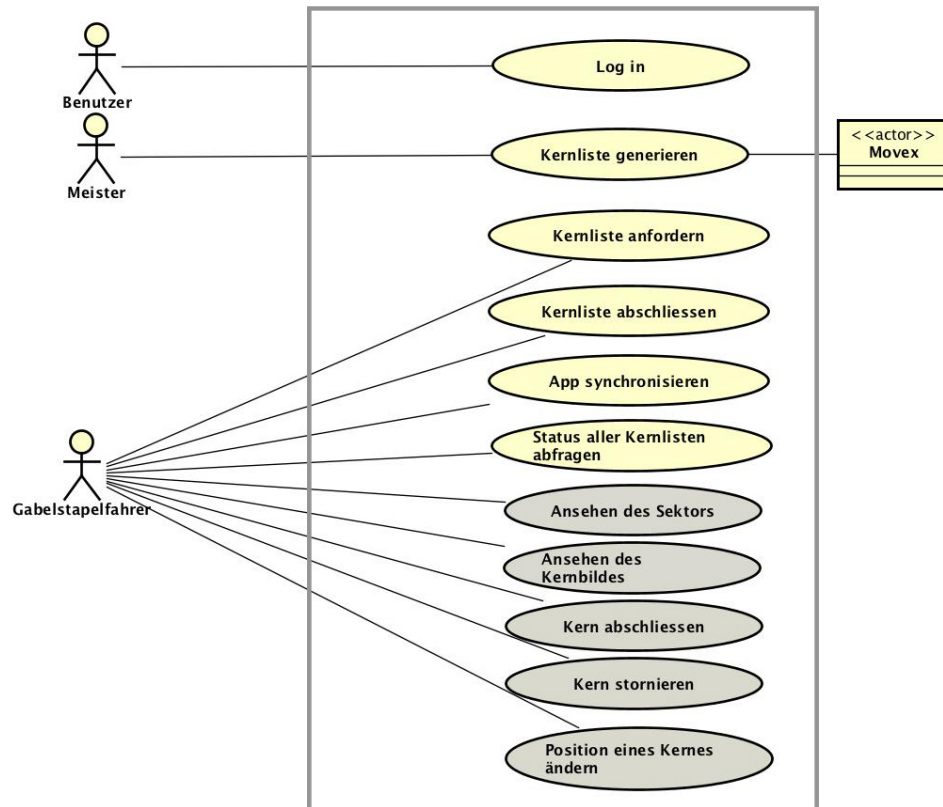


Abbildung 3.8: Use Case Diagramm

3.2.2.3 Use Cases Brief

UC01: Log in

Jeder Benutzer kann sich mit seinen eigenen Credentials in der Android Applikation an- und abmelden.

UC02: Kernliste generieren

Der Meister kann in der Android App neue Daten importieren und daraus neue Kernlisten generieren.

UC03: Palette anfordern

Der Gabelstaplerfahrer kann eine neue Palette zur Bearbeitung anfordern.

UC04: Palette abschliessen

Der Gabelstaplerfahrer kann eine Palette abschliessen.

UC05: App synchronisieren

Der Gabelstaplerfahrer kann seine lokalen Änderungen mit dem Server synchronisieren.

UC06: Status aller Kernlisten abfragen

Der Gabelstaplerfahrer kann Kernlisten und deren Status ansehen.

3.2.2.4 Casual Use Cases

In diesem Kapitel werden die Use Cases, welche mit der Priorität 1 definiert wurden, genauer beschrieben.

3.2.2.4.1 UC01: Log in

Primary Actor: Benutzer

Preconditions:

1. Benutzer im System vorhanden
2. Benutzer noch nicht eingeloggt

Main Success Scenario:

1. Benutzer öffnet Applikation
2. Login Maske erscheint
3. Benutzer loggt sich mit seinen Credentials ein
4. Übersichtsseite wird in App angezeigt

Extensions (or Alternative Flows):

2a: Benutzer ist bereits eingeloggt. Weiter bei Punkt 4

3a: Falsche Credentials: Weiter mit Punkt 2

3.2.2.4.2 UC02: Kernliste generieren

Primary Actor: Meister

Preconditions:

1. Meister ist eingeloggt
2. Aufträge für heutigen Tag sind geladen

Main Success Scenario:

1. Meister öffnet Applikation
2. Übersichtsseite erscheint
3. Meister importiert neue Daten
4. System generiert, gemäss den Kriterien, die Kernlisten
5. Kernlisten sind in Systemübersicht sichtbar

3.2.2.4.3 UC03: Palette anfordern

Primary Actor: Benutzer

Preconditions:

1. Benutzer ist eingeloggt

Main Success Scenario:

1. Benutzer öffnet Applikation
2. Benutzer öffnet Übersicht der Kernlisten
3. Benutzer wählt eine bestimmte Kernliste
4. Benutzer fordert neue Palette an

Extensions (or Alternative Flows):

2a: Keine Kernliste anzuzeigen: Schluss

4a: Benutzer hat bereits eine Palette zugewiesen: Schluss

4b: Keine Paletten mehr verfügbar: Schluss

3.2.2.4.4 UC04: Palette abschliessen

Primary Actor: Benutzer

Preconditions:

1. Benutzer ist eingeloggt

Main Success Scenario:

1. Benutzer öffnet Applikation
2. Benutzer öffnet Übersicht der Kernlisten
3. Benutzer klickt auf eine Kernliste
4. Benutzer wählt zugewiesene Palette aus
5. Benutzer beendet Palette
6. Neue Palette wird zugewiesen

Extensions (or Alternative Flows):

2a: Keine Kernliste anzuzeigen: Schluss

6a: Keine Palette mehr verfügbar: Schluss

3.2.2.4.5 UC05: App synchronisieren

Primary Actor: Benutzer

Preconditions:

1. Benutzer ist eingeloggt

Main Success Scenario:

1. Benutzer öffnet Applikation

2. Benutzer klickt auf Synchronisieren
3. Daten werden vom Server geladen

Extensions (or Alternative Flows):

2a: Keine Verbindung zum Server: Schluss

3.2.2.4.6 UC06: Status aller Kernlisten abfragen

Primary Actor: Benutzer

Preconditions:

1. Benutzer ist eingeloggt

Main Success Scenario:

1. Benutzer öffnet Applikation
2. Benutzer öffnet Übersicht der Kernlisten
3. Übersicht erscheint

Extensions (or Alternative Flows):

2a: Keine Kernliste anzuzeigen: Schluss

3.2.3 Mockups

Öffnet der Benutzer zum ersten Mal die Android Applikation, erscheint der Bildschirm im folgendem Design:

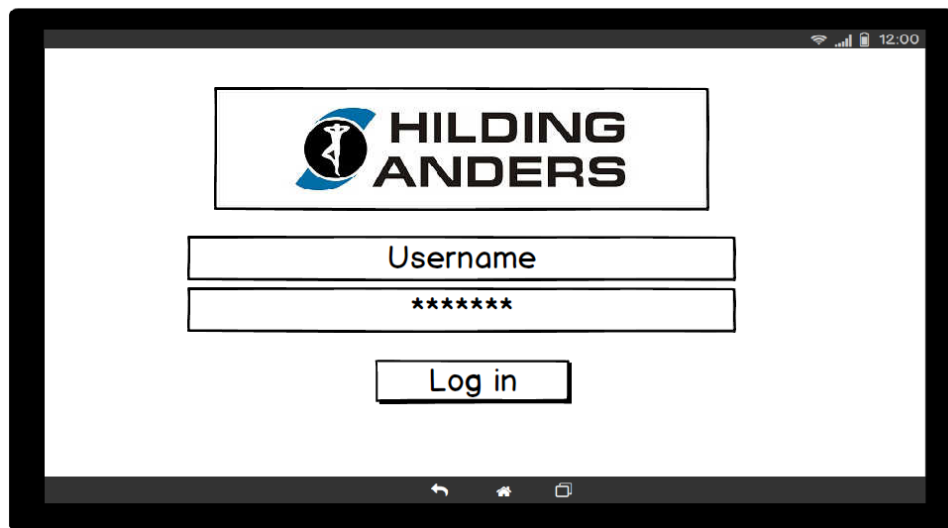


Abbildung 3.9: Mockup Login

Hat sich der Benutzer erfolgreich eingeloggt erscheint die folgende Übersicht:

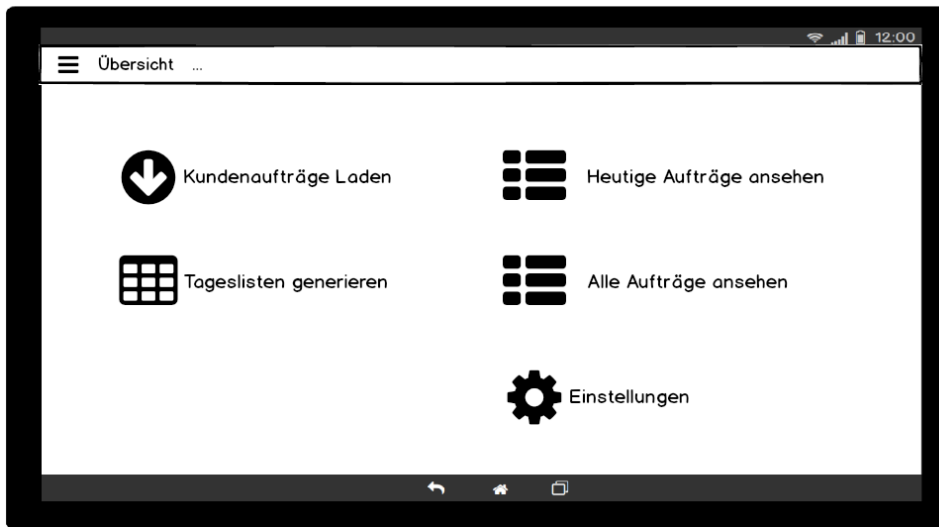


Abbildung 3.10: Mockup Einstiegsseite

Klickt der Benutzer auf „Heutige Aufträge ansehen“ oder auf „Alle Aufträge ansehen“, erscheint eine Ansicht wie in Abbildung 3.11.

Mit einem Klick auf das Übernehmen-Icon, kann der Benutzer den entsprechenden Auftrag sich zuweisen.

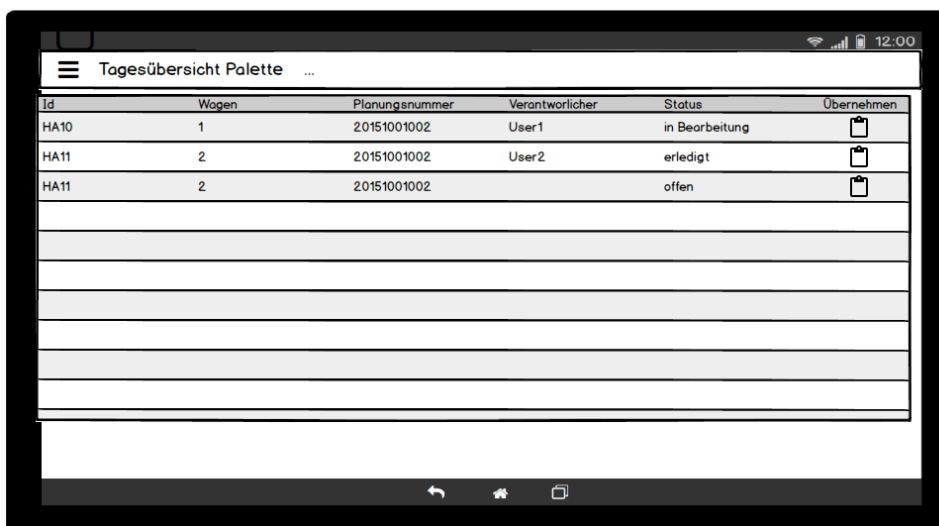
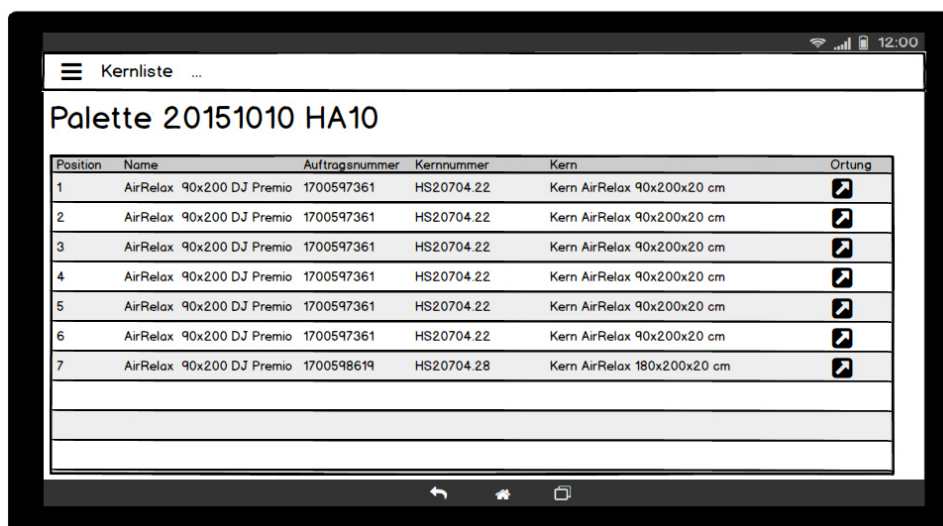


Abbildung 3.11: Mockup Übersicht Kernliste

Mit einem Klick auf ein Element der Liste in Abbildung 3.12, kann die Detailansicht der Kernliste geöffnet werden:



Position	Name	Auftragsnummer	Kernnummer	Kern	Ortung
1	AirRelax 90x200 DJ Premio	1700597361	HS20704.22	Kern AirRelax 90x200x20 cm	
2	AirRelax 90x200 DJ Premio	1700597361	HS20704.22	Kern AirRelax 90x200x20 cm	
3	AirRelax 90x200 DJ Premio	1700597361	HS20704.22	Kern AirRelax 90x200x20 cm	
4	AirRelax 90x200 DJ Premio	1700597361	HS20704.22	Kern AirRelax 90x200x20 cm	
5	AirRelax 90x200 DJ Premio	1700597361	HS20704.22	Kern AirRelax 90x200x20 cm	
6	AirRelax 90x200 DJ Premio	1700597361	HS20704.22	Kern AirRelax 90x200x20 cm	
7	AirRelax 90x200 DJ Premio	1700598619	HS20704.28	Kern AirRelax 180x200x20 cm	

Abbildung 3.12: Mockup Detailansicht Kernliste

Hier werden gleiche Kerne auch gleich eingefärbt, damit dies dem Benutzer sofort auffällt. Für eine bessere Ortung gibt es in der letzten Spalte einen Button, um mehr Informationen über den Kern zu erhalten. Dabei öffnet sich folgendes Popup:

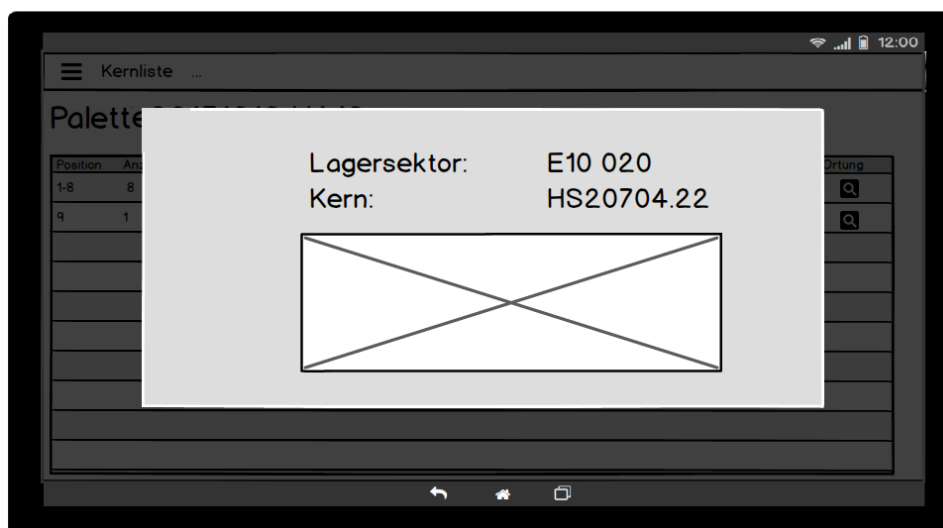


Abbildung 3.13: Mockup Kern Information

Mit einem Klick auf das Popup, lässt sich dieses wieder schliessen.

3.3 Nichtfunktionale Anforderungen

Die folgenden Qualitätsmerkmale wurden aufgrund von ISO 9126 erstellt.

3.3.1 Funktionalität

3.3.1.1 Richtigkeit

Die einzelnen Arbeitsstationen arbeiten in einer vordefinierten Reihenfolge. Daraus folgt, dass die Reihenfolge der Elemente in der Kernliste exakt mit den anderen Listen übereinstimmen muss.

3.3.1.2 Interoperabilität

Die Aufträge werden jeweils direkt aus dem AS400 importiert. Aus diesem Grund muss das System fähig sein, mit den bestehenden Komponenten zu interagieren.

3.3.2 Zuverlässigkeit

3.3.2.1 Fehlertoleranz

Die Android-Applikation muss auch mit einem temporären Netzwerkunterbruch umgehen können. Insbesondere beim Abarbeiten der Kernliste kann es zum Verlust des Netzwerksignals kommen.

3.3.3 Benutzbarkeit

3.3.3.1 Verständlichkeit

Die Applikation soll für den Benutzer selbsterklärend sein. In der Navigation soll jeder Unterpunkt mit maximal 4 Schritten erreicht werden.

3.3.4 Effizienz

3.3.4.1 Zeitverhalten

Das Importieren der aktuellen Aufträge soll maximal 3 Minuten in Anspruch nehmen. Anschliessend sollen die Staplerfahrer problemlos arbeiten können.

3.3.5 Wartbarkeit

3.3.5.1 Modifizierbarkeit

Der Algorithmus für das Berechnen der Kernlisten kann sich von Zeit zu Zeit ändern. Deshalb müssen die Parameter dafür ohne grosse Programmierkenntnisse angepasst werden können.

3.3.5.2 Analysierbarkeit

Alle getätigten Aktionen sollen auch im Nachhinein nachvollziehbar sein, um eventuelle Probleme lösen zu können. Dazu muss ersichtlich sein, wann welcher User welche Aktion ausgeführt hat.

3.4 Domainanalyse

In der Abbildung 3.14 ist das Domain-Modell der Produktion ersichtlich. Dieses Domain-Modell umfasst auch die vorgeschlagenen potenziellen Optimierungsmöglichkeiten aus der Ist-Analyse. Für dieses Projekt wurden die grün und orange eingefärbten Domain-Klassen umgesetzt oder abstrahiert. Die restlichen Elemente haben auf das aktuelle Projekt keinen Einfluss.

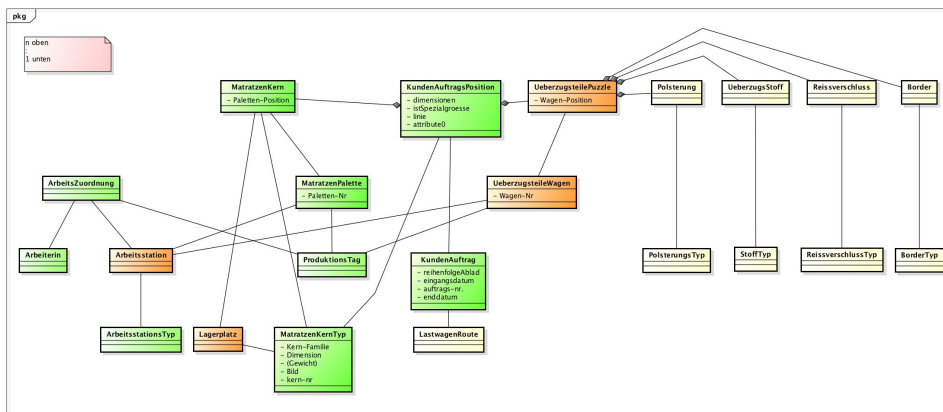


Abbildung 3.14: Domain-Modell

3.4.1 Domain-Modell

Aus dem definierten Scope ergibt sich für diese Arbeit folgendes Domain-Modell (die orangenen Domain-Klassen werden als Abstraktion implementiert):

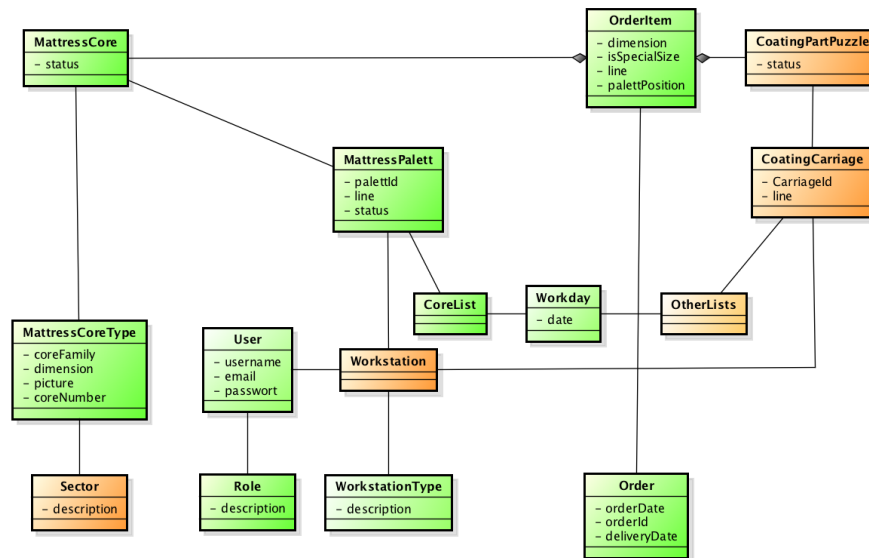


Abbildung 3.15: Domain-Modell Projekt

In den folgenden Unterkapitel werden alle Domain-Klassen beschrieben:

3.4.1.1 Benutzer

Der Hauptzweck der Klasse Benutzer ist den Login zu ermöglichen und die Paletten einem Arbeiter zuweisen zu können.

3.4.1.2 Rolle

Um den einzelnen User unterschiedliche Berechtigungen zuweisen zu können, wird die Klasse Rolle eingeführt.

3.4.1.3 Sektor

Für die Unterteilung des Lagers werden Sektoren als logische Gruppierung eingeführt. Dabei wird jedem Kerntyp ein fixer Sektor zugewiesen. Anhand dieser Sektoren kann dem Staplerfahrer mitgeteilt werden, wo sich ein bestimmter Kern befindet.

3.4.1.4 Kundenauftrag

Kundenaufträge werden beim Importieren erstellt und dienen als Container für die einzelnen Auftragspositionen.

3.4.1.5 Auftragsposition

Eine Auftragsposition ist meist äquivalent zu einer Matratze. Die Auftragsposition dient als Container für alle Bestandteile der Matratze.

3.4.1.6 Matratzenkern

Ein Matratzenkern wird einem Kerntyp zugeordnet und bekommt auf einer Palette jeweils eine Position zugewiesen.

3.4.1.7 Kerntyp

Ein Kerntyp gehört zu einer Kernfamilie und unterscheidet sich von den anderen Elementen der Familie lediglich in den Dimensionen.

3.4.1.8 Kernpalette

Eine Palette enthält eine Anzahl von Kernen (zwischen 1 und 10) und wird pro Tag und Produktionslinie eindeutig nummeriert.

3.4.1.9 Arbeitsstation

Um den Usern eine bestimmte Produktionslinie und Art der Tätigkeit zuzuweisen, gibt es die Arbeitsstationen.

3.4.1.10 Arbeitsstationentyp

Um die Arbeitsstationen in der Art der Tätigkeiten, die dort ausgeführt werden, unterscheiden zu können, gehört jede Station zu einem bestimmten Arbeitsstationentypen.

3.4.1.11 Produktionstag

Jede Kernliste gehört zu einem bestimmten Tag. Dieser wird anhand des Produktionstags auf die Liste gemappt.

3.4.1.12 Überzugsteile-Wagen und Überzugsteile-Puzzle

Diese beiden Domain-Klassen müssen abstrahiert werden, damit es bei den Überzügen keine Durchmischung der Reihenfolge gibt.

3.4.2 Übersetzungstabelle

In der Tabelle 3.3 wird für jede Domain-Klasse die englische Übersetzung, welche verwendet wurde, angegeben. Dies minimiert Missverständnisse beim Programmieren.

deutscher Begriff	englischer Begriff
Benutzer	User
Rolle	Role
Sektor	Sector
Kundenauftrag	Order
Auftragsposition	OrderItem
Matratzenkern	MattressCore
Kerntyp	MattressCoreType
Kernpalette	MattressCorePallet
Arbeitsstation	Workstation
Arbeitsstationtyp	WorkstationType
Überzugsteile-Wagen	CoatingCart
Überzugsteile-Puzzle	CoatingPuzzle

Tabelle 3.3: Tabelle Übersetzungen Domain-Klassen

Kapitel 4

Software Architecture Documentation

4.1 Kontextdiagramm

Das Kontextdiagramm soll die externen Abhängigkeiten zum entwickelten System abgrenzen:

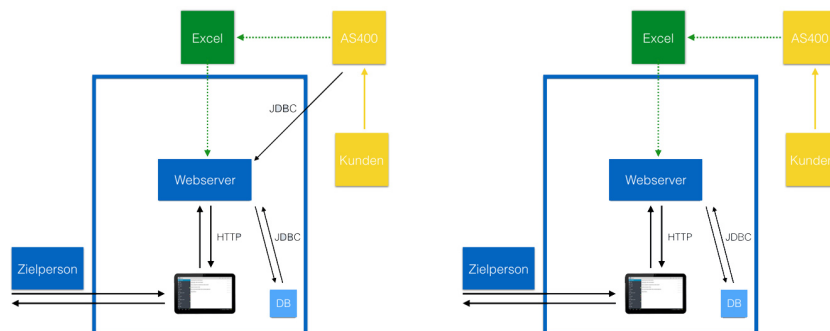


Abbildung 4.1: Kontextdiagramme

Die beiden Versionen des Kontextdiagramms unterscheiden sich nur bezüglich der Datenbankverbindung. Ursprünglich war geplant direkt über eine JDBC-Verbindung auf das AS400 zuzugreifen. Im Laufe der Studienarbeit stellte sich aber heraus, dass ein solche Verbindung nicht bewerkstelligt werden kann.

Nun werden die produktiven Daten mit Hilfe eines VBA-Makro in die Datenbank gespielt. Nachteil: Diese Aktion muss immer manuell ausgeführt werden.

4.2 Verteilungsdiagramm

Mit dem Verteilungsdiagramm wird eine grobe Übersicht über das ganze System gezeigt:

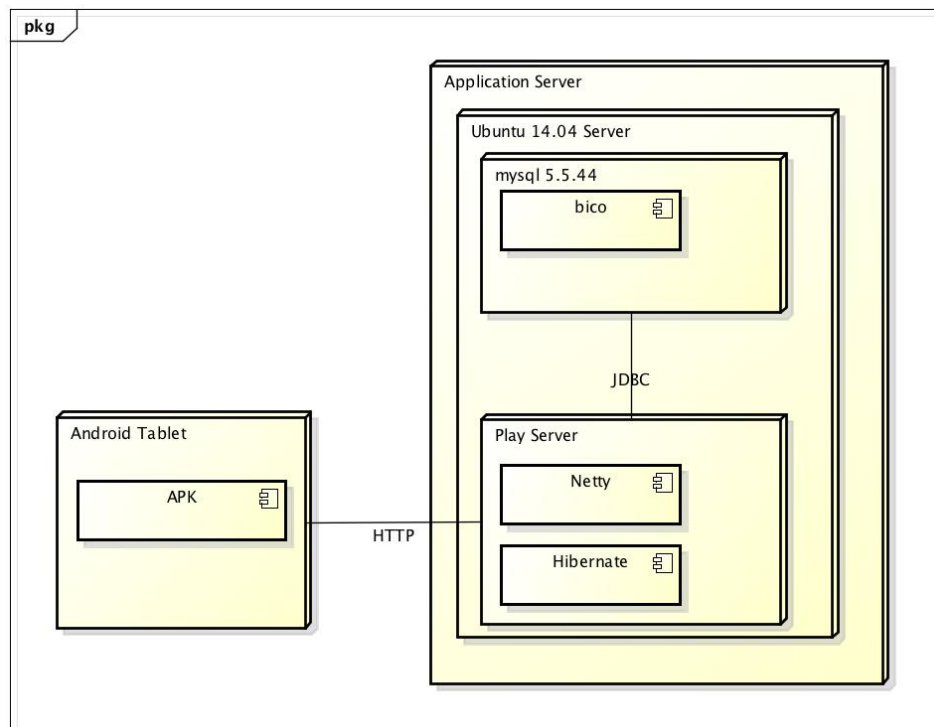


Abbildung 4.2: Verteilungsdiagramm

Von einer App aus, die auf einem Android-Tablet ausgeführt wird, wird auf den Webserver zugegriffen. Der Server beantwortet darauf den Request. Dabei verwendet er die Informationen aus einer lokalen Datenbank, welche zuvor mit den Daten aus den Excel-Dateien befüllt wurde.

4.3 Schichtendiagramm

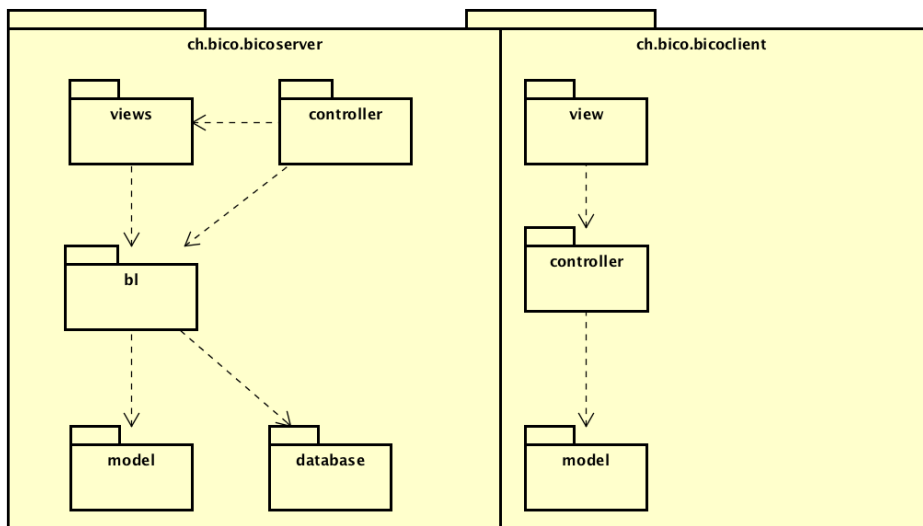


Abbildung 4.3: Schichtendiagramm

4.3.1 Server

views: Das Package enthält HTML Templates, für die Darstellung von formatierten Daten. Die Templates werden durch die Controller aufgerufen.

controller: Im „controller“-Package werden alle implementierten Routen erstellt. Die Logik zur Generierung der Antworten wurde bei längerem Code ins Package „bi“ ausgelagert.

bi: Hier befinden sich unter anderem die Funktionalitäten für den Import der Daten, sowie die Logik welche für das Erstellen der Paletten benötigt wird.

model: Dieses Package beinhaltet alle Entities, welche für die Persistierung der Daten mit Hibernate verwendet werden.

database: In diesem Package befinden sich die Klassen, die für die Interaktionen mit der Datenbank notwendig sind. Um die Abfragen generisch gestalten zu können, wurden mehrere Hilfsmethoden für das Abfragen der gespeicherten Entities erstellt.

4.3.2 Client

view: Das „view“-Package enthält die Android Fragments und Activities, welche für die grafische Benutzeroberfläche benötigt werden.

controller: Inhalt dieses Packages bilden die Klassen, die für den Verbindungsaufbau zum Server verwendet werden. Ausserdem finden sich hier Adapter und weitere Hilfsklassen.

model: Damit die vom Server empfangenen Objekte geparkt werden können, müssen die Entity-Klassen auch auf dem Client existieren. Die Klassen sind den Server-Entities sehr ähnlich, jedoch enthalten sie keine Annotationen und nur die Methoden, welche für die Anzeige benötigt werden.

4.4 Komponenten

4.4.1 Server

Der Server basiert auf dem Play Framework [6]. Dabei werden folgende Tools verwendet:

Jetty Webserver: Damit das Play Framework eine REST-Schnittstelle anbieten kann, wird Jetty als Webserver verwendet. Standardmässig läuft dieser auf TCP Port 9000. Aufgrund der Einschränkungen der HSR-Firewall, muss er aber auf Port 40001 gestartet werden, da ansonsten alle Verbindungen blockiert werden.

Hibernate: Als OR-Mapper wird Hibernate [1] verwendet. Hibernate kann im Play Framework über eine Datei „hibernate.cfg.xml“ im Root Verzeichnis konfiguriert werden. Dabei wird unter anderem festgelegt, welche Klassen als Entities dienen:

```
<mapping class="model.CoreList"/>
<mapping class="model.MattressCore"/>
<mapping class="model.MattressCorePicture"/>
<mapping class="model.MattressCoreType"/>
<mapping class="model.MattressPallet"/>
<mapping class="model.OrderItem"/>
<mapping class="model.Order"/>
<mapping class="model.ProductionLine"/>
<mapping class="model.Sector"/>
<mapping class="model.User"/>
```

Abbildung 4.4: hibernate.cfg.xml

Mit JPA-Annotationen werden die Entities direkt in den Entity-Klassen konfiguriert:

```
@Id
@GeneratedValue
private Long id;

private String mattressPalletId;

private String user;

@OneToMany(fetch = FetchType.EAGER, mappedBy = "mattressPallet", cascade = CascadeType.ALL)
@JsonManagedReference
private List<MattressCore> mattressCoreList = new ArrayList<>();

@OneToOne
private ProductionLine productionLine;

@ManyToOne
@JsonBackReference
private CoreList coreList;

public MattressPallet() {
}
```

Abbildung 4.5: JPA-Annotationen in den Entities

Logback: Seit der Version 2.0 verwendet das Play Framework „Logback“ als Logging-Framework. Einstellungen bezüglich dem Logging können in der Datei conf/application.conf definiert werden. Aktuell werden INFO-Meldungen des Play Frameworks und DEBUG-Meldungen der Applikation geloggt.

Typesafe Activator: Mithilfe des Build-Tool „Typesafe Activator“ können im Play Framework die Build-Prozesse durchgeführt werden. Dabei werden alle Abhängigkeiten in der „build.sbt“-Datei definiert:

```
libraryDependencies += Seq(
  javaJdbc,
  cache,
  javaws,
  javaJpa.exclude(
    "org.hibernate.javax.persistence", "hibernate-jpa-2.0-api",
    "org.hibernate" % "hibernate-entitymanager" % "4.3.11.Final",
    "mysql" % "mysql-connector-java" % "5.1.36",
    "org.scoverage" %% "scalac-scoverage-plugin" % "1.1.1",
    "com.fasterxml.jackson.dataformat" % "jackson-dataformat-csv" % "2.6.3",
    "org.hibernate" % "hibernate-c3p0" % "4.3.11.Final",
    "org.hibernate" % "hibernate-core" % "4.3.11.Final",
    "junit" % "junit" % "4.11" % "test",
    "org.easytesting" % "fest-assert-core" % "2.0M8"
  )
)
```

Abbildung 4.6: build.sbt

Jackson: Für das Parsing von CSV Files wurde das Utility von Jackson verwendet. Mit diesem kann einfach aus einer CSV-Datei eine Liste von Key-Value Objekten erstellen werden. [2].

4.4.1.1 views Package

Dieses Package enthält ein Template um eine Liste von stornierten Matratzenkerne anzeigen zu können. Dies ist nötig, da diese Ansicht nicht in der App implementiert wurde und es sich doch um wichtige Informationen für die Planung handelt. In der Abbildung 4.7 ist eine solche Liste dargestellt.

Elemente mit Status Cancelled: 1

Artikelnummer	Kernname	Kern	Auftragsnummer	Kernliste	Pallet	Status	Bestelldatum	
H1161.22.6581	VitaLuxe medium 90x200 DJ VitaLuxe	HS20210.22	1700597499	20150922002	HR7	CANCELLED	20150922	Ereignet



Abbildung 4.7: Stornierte Matratzenkerne

4.4.1.2 controller Package

In diesem Package sind, in drei Klassen aufgeteilt, alle HTTP-Controller zu finden. „Android“ enthält alle Controller, welche innerhalb der Android-App aufgerufen werden. „Application“ implementiert die Controller, die für die Ansicht der stornierten Kerne notwendig sind. Die Klasse „Importer“ stellt diverse Controller zum Importieren von Daten zur Verfügung.

Die wichtigsten Controller sind in der folgenden Tabelle aufgelistet:

Methode	Beschreibung
importData	Importiert alle Einträge aus der „movex“-Tabelle und generiert daraus die dazugehörigen Kernlisten
getCoreLists	Gibt eine Liste aller Namen von Kernlisten zurück
getCoreList	Retourniert eine bestimmte Kernliste
getNextPallet	Gibt die nächste verfügbare Palette zurück
getPicture	Liefert das Bild des Kerns Base64 encoded als String zurück
getPictures	Retourniert eine Map mit allen Kernbildern (auch Base64 encoded)
postCancelMattressCore	Setzt einen gewissen Kern auf den Status „CANCELLED“
postFinishMattressPallet	Setzt alle Kerne einer Palette auf den Status „DONE“
postLogin	Überprüft die Anmeldeinformationen und gibt bei einem positiven Entscheid einen Token zurück
postLogout	Setzt den Token des Users auf null
cancelledMattressCores	Gibt im Browser eine Tabelle mit allen Kernen aus, die den Status auf „CANCELLED“ gesetzt haben
clearMattressCores	Ändert den Status eines Kerns von „CANCELLED“ auf „DONE“

Tabelle 4.1: Controller Methoden

4.4.1.3 bl Package

Dieses Package enthält die ganze Business Logik und wird von den Controllern aufgerufen. Es finden sich Methoden für das Erstellen der Paletten, das Importieren von Daten sowie für das Authentifizieren der User.

LoginAuthenticator: Diese Klassen enthält Methoden, die sicherstellen, dass nur User mit genügend Berechtigungen auf die REST Routen zugreifen:

```
public class LoginAuthenticator extends Security.Authenticator {  
  
    @Override  
    public String getUsername(Http.Context ctx) {  
        String[] authTokenHeaderValues = ctx.request().headers().get("Authorization");  
        if (authTokenHeaderValues != null && authTokenHeaderValues[0] != null) {  
            String token = authTokenHeaderValues[0];  
            User user = getUserByToken(token);  
            if (user != null) {  
                return user.getEmail();  
            }  
        }  
        return null;  
    }  
  
    @Override  
    public Result onUnauthorized(Http.Context ctx) { return super.onUnauthorized(ctx); }  
}
```

Abbildung 4.8: LoginAuthenticator

StaticDataImporter: Diese Klasse ermöglicht das Importieren von Kern-Bilder und Sektoren. Dazu wird im Filesystem nach den entsprechenden Daten gesucht. Aktuell wurden die folgenden Orte definiert:

Methode	Beschreibung
Bilder	db/corePictures
Sektoren	db/sectorData.csv

Tabelle 4.2: Speicherort der zu importierenden Daten

DatabaseImporter: Diese Klasse ist dafür zuständig die Daten aus der Tabelle „palletSort“ abzuarbeiten. Ziel ist es, für die bereits aus der „movex“-Tabelle importierten Daten die Paletten anhand der Daten aus der Tabelle „palletSort“ zu erstellen.

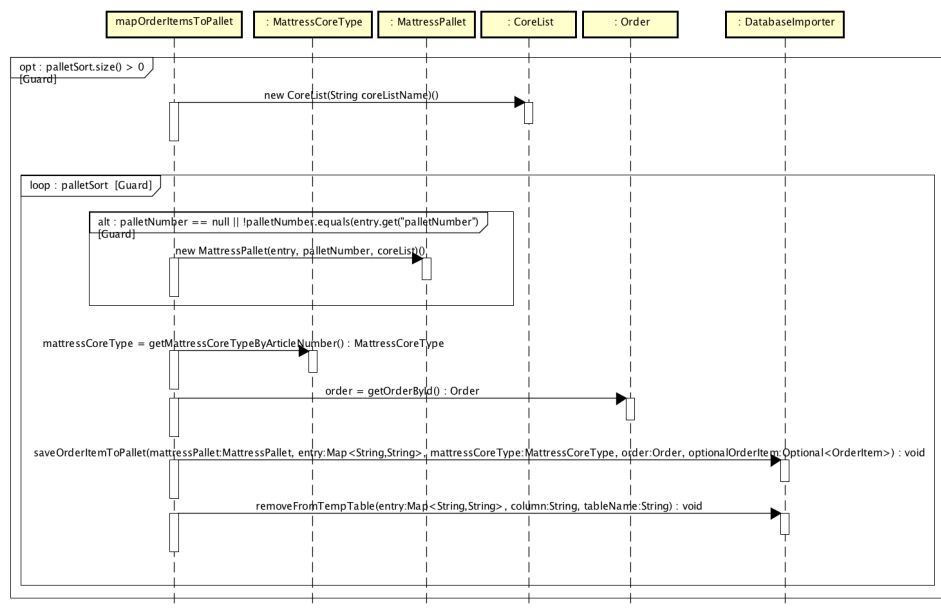


Abbildung 4.9: Ablauf OrderImporter

OrderImporter: Mit dieser Klasse wird eine `Map<List<String,String>>` entgegen genommen und erstellt daraus die Bestellungen der einzelnen Entities.

PalletFilter: Die Methode „nextPalletByPalletNumber“ sucht nach Paletten mit dem Status „OPEN“, die einem User zugewiesen werden können. Die Reihenfolge basiert dabei auf der „palletNumber“, sodass die Produktlinien parallel verarbeitet werden.

Komparatoren: Da bei gewissen Sortierungen ein Lambda zu unübersichtlich geworden wäre, wurden auch Komparatoren erstellt. Eine Liste dieser findet man in der Tabelle 4.3.

Komparatoren	Beschreibung
MattressCoreComparator	Dieser Komparator wird verwendet, um die Kerne sortiert nach der Position auf einer Palette zurückzugeben.
MattressPalletComparator	Dieser Komparator wird verwendet, um eine Liste von Paletten basierend auf deren Name (z.B. „HA01“) zu sortieren.

Tabelle 4.3: Verwendete Komparatoren

PalletGenerator: Diese Klasse ist zuständig für das Erstellen der Matratzenkerne und das Aufteilen der Kerne auf die verschiedenen Paletten.



Abbildung 4.10: Sequenzdiagramm PalletGenerator

4.4.1.4 model Package

Diese Package enthält alle Informationen, welche für die Persistierung der Entities benötigt werden. Die Models wurden mit folgenden Annotationen konfiguriert:

Annotation	Beschreibung
@Entity(name =)	Definiert Tabellennamen
@Id	Setzt Spalte in MySQL als PRIMARY KEY
@GeneratedValue	sorgt dafür, dass das AUTO_INCREMENT Attribut von MySQL verwendet wird
@JsonManagedReference	Variable ist Teil einer zirkulären Referenz
@JsonBackReference	Variable wird mittels Json nicht serialisiert
@OneToMany	Erstellt 1:N Referenz zwischen zwei Entities
@ManyToOne	Erstellt N:1 Referenz zwischen zwei Entities
@OneToOne	Erstellt 1:1 Referenz zwischen zwei Entities
@LazyCollection	Sorgt dafür, dass ManyToOne Collections als LazyCollections initiiert werden
@Column	Weitere Konfiguration für MySQL Spalte (wie Grösse des Feldes)

Tabelle 4.4: Übersicht JPA Annotationen

Die Klassen „Role“ und „Status“ sind als Enums implementiert.

Mögliche Werte für „Rolle“ sind:

- FOREMAN
- WORKER

Werte für den Status sind:

- RUNNING
- OPEN
- DONE
- CANCELLED

Status der einzelnen Entities: Der Status wird bloss in der „MattressCore“-Klasse definiert. Der Status einer Palette oder einer Kernliste wird dann auf Basis dieser Klasse wie folgt berechnet:

```
public Status getStatus() {
    if (user == null) {
        return Status.OPEN;
    } else {
        Status temp = Status.CANCELLED;
        for (MattressCore c : mattressCoreList) {
            switch (c.getStatus()) {
                case RUNNING:
                    return Status.RUNNING;
                case DONE:
                    if (temp.equals(Status.CANCELLED)) {
                        temp = Status.DONE;
                    }
                    break;
                default:
                    break;
            }
        }
        return temp;
    }
}
```

Abbildung 4.11: Berechnung des Status einer Palette

4.4.1.5 database Package

Dieses Package enthält mit **HibernateUtil** eine Klasse mit zahlreichen Hilfsmethoden für das Speichern, Aktualisieren und Abfragen der Entities in der Datenbank. Die Annotation **@Transactional** kennzeichnet Methoden, welche auf die Datenbank zugreifen und werden nur in diesem Package verwendet.

getSession: Diese Methode gibt eine Session zurück. Falls keine laufende Session existiert, wird eine neue mit Hilfe der „SessionFactory“-Klasse generiert. So wird sichergestellt, dass der Connection Pool der Datenbank nicht mit unnötigen Sessions überfüllt wird.

mergeEntity: Diese Methode verwendet die „saveOrUpdate“-Methode von Hibernate erstellt eine neue Entity, wenn sie noch nicht existiert. Ansonsten werden die aktuellen Daten der Entity aktualisiert.

getEntityList: Die Methode „getEntityList“ gibt eine Liste von Entities zurück. Beim Aufruf muss die Klasse des gesuchten Resultats mitgegeben werden.

getUniqueEntityWithCriteria/getEntityListWithCriteria: Diese beiden Methoden verhalten sich ähnlich wie „getEntityList“. Jedoch werden hier die Resultate gefiltert. „getUniqueEntityWithCriteria“ geben eine einzige Entity und „getEntityListWithCriteria“ eine gefilterte Liste von Entities zurück.

executeSQLQuery: Diese Methode wird verwendet um einzelne SQL-Statements (INSERT, DROP, UPDATE) via Hibernate abzusetzen. Der Hauptzweck ist das Entfernen der Daten aus den Tabellen „movex“ und „palletSort“, nachdem deren Inhalt für den Import verwendet wurde.

4.4.2 Mobile Client

Für die Android-Applikation wurden die folgenden Tools verwendet:

Gradle: Gradle ist ein Open Source Automatisierungs-Tool. Damit können zum Beispiel Abhängigkeiten zentral in der Datei „build.gradle“ definiert werden. Im Projekt wurde Gradle in der Version 23 verwendet, da gewisse grafische Elemente dies benötigen haben.

GSON: GSON ist ein Parser von Google, der für die Umwandlung von JSON-Strings in Objekte verwendet wurde.

Beim Design der Applikation wurden so weit wie möglich die Google Design Guidelines [4] umgesetzt. An gewissen Orten musste auf das Material Design aufgrund von Kundenwünschen verzichtet werden. Deshalb skaliert die Applikation bei einer zu kleinen Auflösung nicht sauber mit.

4.4.2.1 view Package

LoginActivity: Diese Activity erscheint beim Start der Applikation. In dieser Ansicht kann sich der User mit seinem Username und Passwort beim Server authentisieren. Wenn dieser Ablauf positiv war, wird ein Token im Hintergrund abgespeichert. Beim erneuten Öffnen der App muss sich der User dadurch nicht erneut anmelden.

NavigationDrawerActivity: Die „NavigationDrawerActivity“ dient als Container für die Fragments, welche die eigentlichen Daten beinhalten. Die Activity fügt aber das Design Element des Navigation Drawer ein. Dieses Element stammt aus dem Material Design von Google und soll eine neue Art der Navigation bieten. Die folgenden Fragments sind bei der Verwendung der App für den User sichtbar:

- CorelistFragment
- CorelistsFragment
- ImportFragment
- OverviewFragment
- PalletlistFragment

Das Fragment „LogoutFragment“ ist nie sichtbar, wird aber für den Logout des Users benötigt.

4.4.2.2 controller Package

adapter: Die Klassen in diesem Package werden für das Design Element „RecyclerView“ benötigt. Die Adapter sorgen für das Mapping der Daten auf die einzelnen Textfelder.

asynctask: Die Klassen in diesem Package werden für die asynchrone Kommunikation mit dem Back-End benötigt.

FileService: Diese Helferklasse vereinfacht das Erstellen, Lesen und Löschen von Files auf dem Android-Tablet.

HttpService: Diese Helferklasse wird für die HTTP-Kommunikation zwischen dem Server und Client benötigt. Sie erlaubt dabei POST und GET Nachrichten zu versenden.

4.4.2.3 model Package

In diesem Package befinden sich alle Klassen, die für das Parsing, der vom Server empfangenen Entities, benötigt werden. Die Klassen ähneln denen vom Server, sind jedoch nicht identisch. Es werden zum Beispiel keine JPA-Annotationen benötigt und die Klassen enthalten bloss die Methoden, die für das Anzeigen und Parsing benötigt werden.

4.5 Kommunikation zwischen Server und Client

Bei einem Login wird auf der Serverseite ein Token generiert. Dieser muss nun für die zukünftigen Anfragen zur Verifizierung des Users mitgesendet werden. Der Token wird im HTTP-Header mitgesendet. Daraus ergibt sich der folgende Ablauf zwischen dem Server und Client:

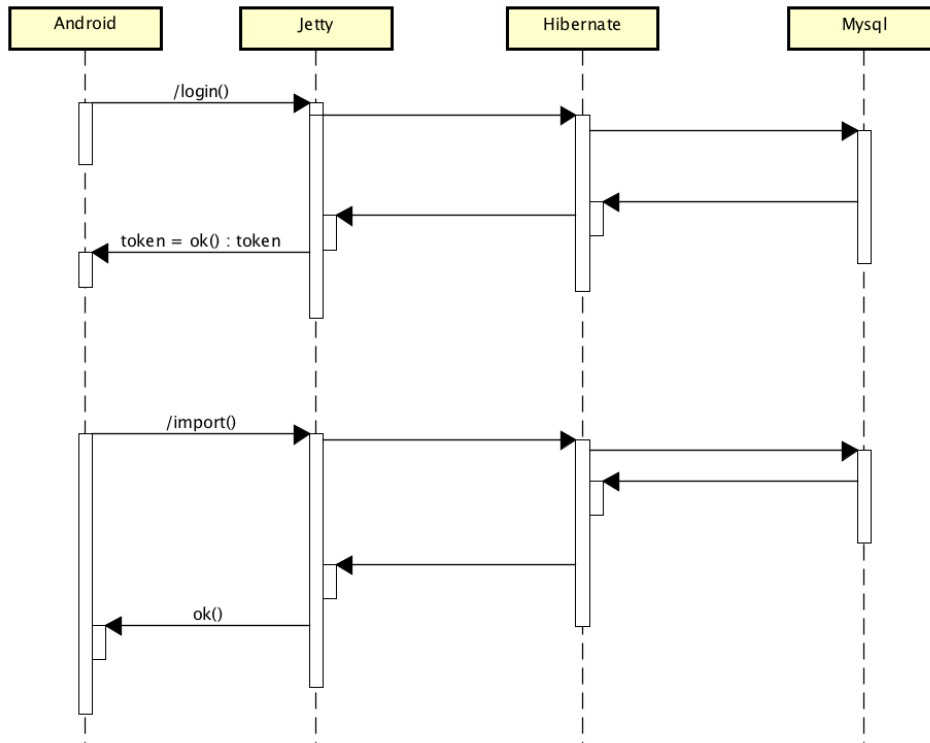


Abbildung 4.12: Ablauf eines Requests zwischen Android und Server

Die Requests werden von der App an den Webserver (Jetty) gesendet. Dort wird der Request verarbeitet. Wo nötig, werden mittels Hibernate weitere Requests auf die Datenbank abgesetzt. Anschliessend wird der Android App der Request beantwortet.

4.6 User Interface

Für das GUI wurden die folgenden drei Farben verwendet: #81b2e9, #4b92db und #fcd3bc. Alle Farben wurden vom Online-Auftritt der Hilding Anders Switzerland AG übernommen.

In der Abbildung 4.13 ist der Ablauf zwischen den einzelnen Ansichten ersichtlich.

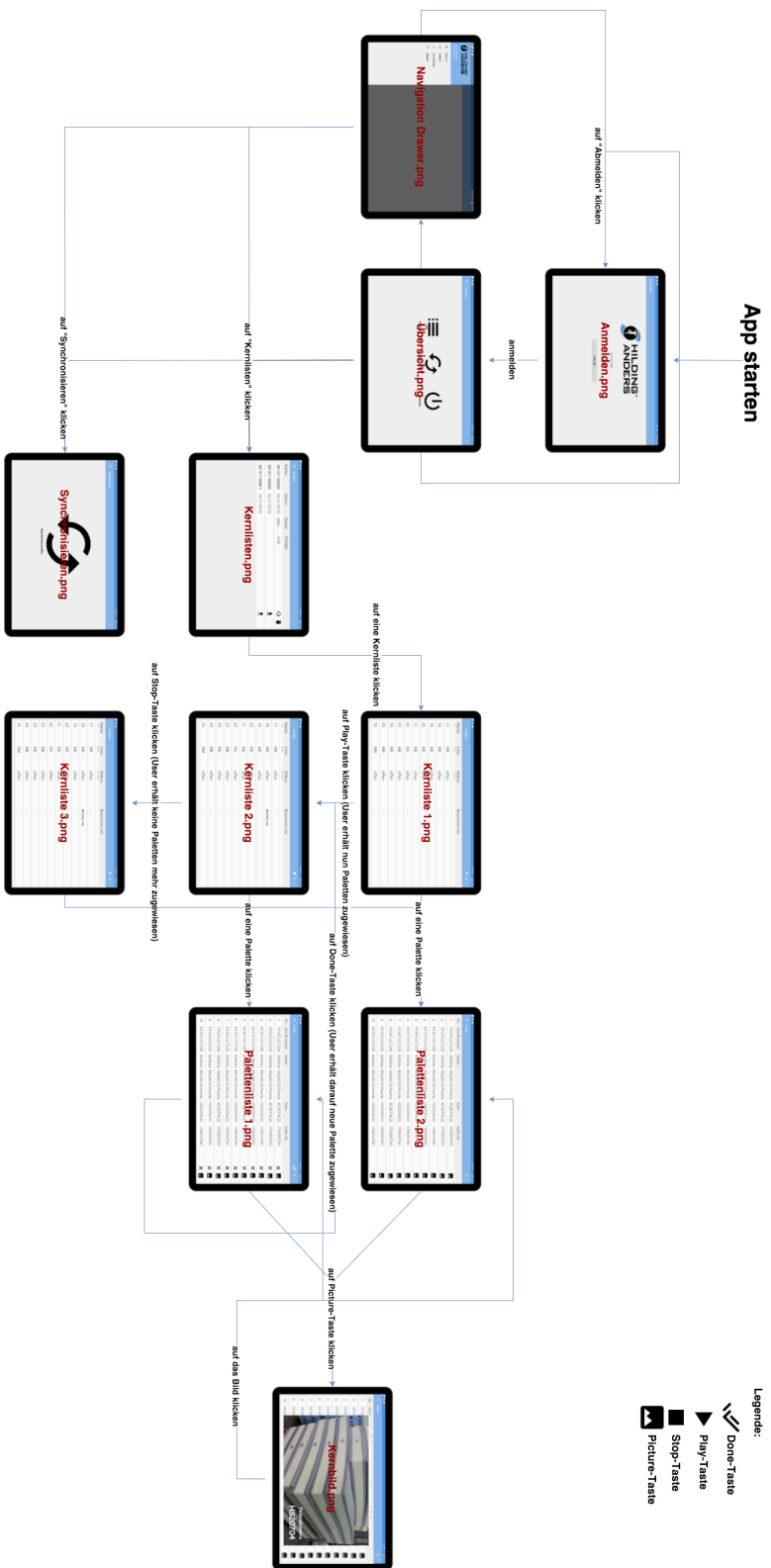


Abbildung 4.13: Ablauf Android

4.7 Datenbank

Um die verschiedenen Entities persistieren zu können, wird folgendes Datenbankschema verwendet:

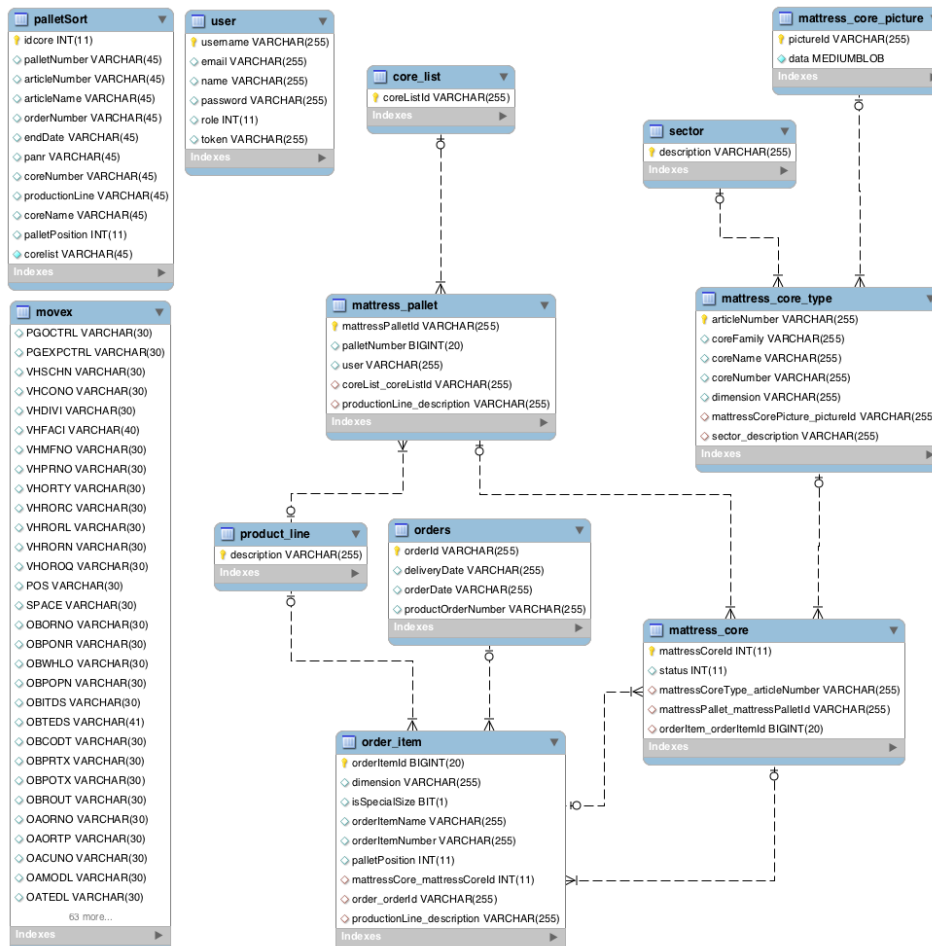


Abbildung 4.14: Datenbankschema

Ein grosser Teil der Datenbanktabellen setzt sich aus den Entities zusammen. Ausnahmen bilden die beiden Tabellen „movex“ und „palletSort“, welche für die Importfunktionalität aus dem Excel zuständig sind und nicht von Hibernate gemanagt werden.

Die Tabelle „movex“ enthält die Daten genau so, wie sie aus dem AS400 exportiert werden und stellt damit sicher, dass der Zugriff auf die produktiven Daten möglich ist. In der Tabelle „palletSort“ wird die Aufteilung der Kerne, auf die unterschiedlichen Paletten, aus dem Excel übernommen.

Folgende Tabelle zeigt auf, wie die Datenbanktabellen mit den Entities verknüpft sind:

Datenbank Tabelle	Entity
user	User
sector	Sector
orders	Order
order_item	OrderItem
mattress_core	MattressCore
mattress_core_type	MattressCoreType
mattress_pallet	MattressCorePallet
corelist	CoreList
production_line	ProductionLine
mattress_core_picture	MattressCorePicture

Tabelle 4.5: Beziehung Datenbank zu Entities

4.8 Routen

Auf dem Application-Server wurden die folgenden Routen definiert:

HTTP Verb	Route	Beschreibung
POST	/login	Führt einen Login durch.
POST	/logout	Führt einen Logout durch.
POST	/import	Importiert Daten aus Movex und generiert entsprechende Kernlisten
GET	/csvload	Lädt Movex Daten aus dem lokalen CSV
GET	/createPallets	Erstellt Paletten aus den zur Verfügung stehenden Bestellungen
GET	/cancelledMattressCores	Generiert die Ansicht der stornierten Kerne
POST	/clearMattressCore/:coreId	Methode zum Abschliessen stornierter Kerne
GET	/coreLists	Gibt eine Liste aller Kernlisten-Namen zurück.
GET	/coreList/:list	Gibt eine bestimmte Kernliste zurück.
GET	/nextPallet/:list	Liefert die nächste Palette für einen Benutzer
GET	/picture/:core	Liefert das Bild eines bestimmten Kerntyps.
GET	/pictures	Gibt die Bilder aller Kerntypen zurück.
POST	/finishPallet	Schliesst eine Palette ab.
POST	/cancelMattressCore	Setzt den Status eines Kernes auf CANCELLED.

Tabelle 4.6: Server-Routen

Kapitel 5

Projektplanung

5.1 Qualitätssicherung

5.1.1 Continuous Integration

Zur Umsetzung von Continuous Integration wurde Travis CI verwendet. Dabei wird bei jedem Push auf den Master Branch ein neuer Build angestoßen.



Build Status	Commit Message	Commit Author	Build Number	Build Status	Duration	Time Ago
Success	master fix coverage data	Philippe Naegeli committed	# 61 passed	Success	8 min 21 sec	about 3 hours ago
Success	master add post request	Philippe Naegeli committed	# 60 passed	Success	7 min 1 sec	about 5 hours ago
Success	master copy check style to hsr-server	Philippe Naegeli committed	# 59 passed	Success	6 min 10 sec	about 14 hours ago
Success	master add findbugs to build	Philippe Naegeli committed	# 58 passed	Success	6 min 40 sec	about 14 hours ago
Failure	master add pmd to build	Philippe Naegeli committed	# 57 canceled	Failure	7 min 5 sec	about 14 hours ago
Success	master add google check style config	Philippe Naegeli committed	# 56 passed	Success	6 min 28 sec	about 15 hours ago

Abbildung 5.1: Übersicht Travis CI

Während des Builds werden die Plugins „Coverage Data“, „Checkstyle“ und „Findbugs“ aufgerufen. Somit können im Nachhinein Trends der Testabdeckung eingesehen werden.

Ist ein Build erfolgreich, wird von Travis CI zusätzlich ein Deployment auf den produktiven Server gestartet.

5.2 Projektablauf

Das Projekt wurde mittels agilem Projektmanagement realisiert. Dafür wurden im Laufe des Projektes total 7 Sprints à 2 Wochen durchgeführt. Folgende Tabelle beschreibt den Zweck der einzelnen Sprints und welche Artefakte während dieser Zeit erzielt wurden:

Beschreibung	Startdatum	Beschreibung
Sprint 1	14.09.2015	Das Ziel dieser ersten Phase war das Verstehen der aktuellen Situation und das definieren des Projektauftrages. Zusätzlich musste die Vorgängerarbeit zum Laufen gebracht werden.
Sprint 2	28.09.2015	Das Ziel dieses Sprints war es, den Auftrag komplett zu verstehen, die Anforderungen genauer zu spezifizieren und die Fragmente für End of Elaboration zu erstellen. Während dieser zwei Wochen wurde auch ein Prototyp erstellt, mit welchem die Kommunikation zwischen allen Schichten sichergestellt wurde.
Sprint 3	12.10.2015	Inhalt dieses Sprint war es, den Prototypen weiter auszubauen und auf der Android Applikation bereits erste Inhalte darstellen zu können.
Sprint 4	26.10.2015	Ziel war das Anzeige der ersten Kernlisten und die Login Funktionalität.
Sprint 5	09.11.2015	In diesem Sprint bestand das Ziel aus den folgenden Punkten: Anpassen des User Interfaces, Übernehmen einer Kernliste und Optimieren des Algorithmus. Zudem wurde ein Interface erstellt, damit ein Export vom Excel in eine Datenbank gemacht werden kann.
Sprint 6	23.11.2015	In diesem Sprint wurde der Fokus auf die Implementation der letzten Funktionen gelegt. Dazu gehören das Abhaken von Kernlisten und das Anzeigen von Kern-Bilder und Sektoren. Zudem fanden die Dreharbeiten für den Werbefilm der HSR statt.
Sprint 7	7.12.2015	Das Ziel dieses Sprints war die Durchführung der letzten Tests, die Übergabe des Produktes und das Fertigstellen der Dokumentation.

Tabelle 5.1: Sprintplanung

Kapitel 6

Projektnachverfolgung

6.1 Einschränkungen

6.1.1 Back Button Android

Problembeschreibung: Wird unter Android das Fragment ein zweites Mal aufgerufen, so verschwindet der Zurück-Button in der Toolbar.

Workaround: Physischer Zurück-Button des Tablets kann verwendet werden.

6.1.2 Datenbankverbindung

Problembeschreibung: Eine direkte Datenbankverbindung zum AS400 von BICO war nicht möglich.

Workaround: Mittels eines VBA-Scripts können die Movex-Daten in die Datenbank geladen werden.

6.1.3 Offene Use Cases

Der Use Case „UC05: App synchronisieren“ wurde nicht so umgesetzt, da keine lokalen Daten existieren, die synchronisiert werden müssten. Es wird aber vorausgesetzt, dass der Benutzer beim Laden der Listen, sowie beim Abschliessen einer Palette eine aktive Verbindung zum Server hat.

Zudem wurde der optionale Use Case „UC11: Position eines Kernes ändern“ aus zeitlichen Gründen nicht implementiert.

6.2 Metriken

	Server	Client
Anzahl Klassen	25	45
Anzahl Packages	6	4
Zeilen Java Code	1699	2260
Zeilen XML Code	0	977

Tabelle 6.1: Metriken Tabelle

6.3 Geleistete Zeit

Wie in der Tabelle 6.2 zu sehen ist, wurde während dieser Studienarbeit mehr geleistet, als die erforderlichen 240 Stunden pro Student:

Woche	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
Philippe Naegeli	6.5	17.5	10.9	21.3	19.7	18.25	12.75	26	26.25	22.25	36	34	40.25	32.5	324.15
Raffael Ioannone	3	16	9	19.5	11	25	15	29.25	32.9	18.75	36.75	32.25	36	40.5	324.9
Total	9.5	33.5	19.9	40.8	30.7	43.25	27.75	55.25	59.15	41	72.75	66.25	76.25	40	616.05

Tabelle 6.2: Geleistete Zeit

Über die komplette Projektdauer visualisiert, sieht die erfasste Zeit folgendermassen aus (Soll-Zeit ist 17.14h pro Student):

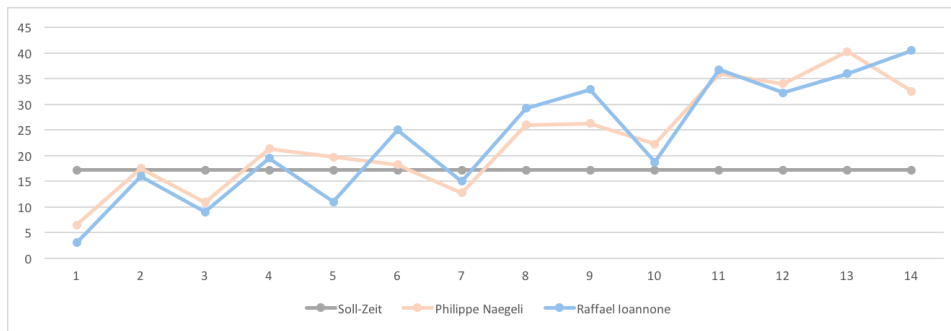


Abbildung 6.1: Geleistete Arbeit über die komplette Projektdauer

6.4 Ausblick

Während dieser Studienarbeit wurde die Kernliste auf das Tablet migriert. Als nächster Schritt empfiehlt sich die Migration der restlichen Listen. Dafür wäre eine bessere Anbindung an die produktive Datenbank sinnvoll.

Zudem gibt es für die ganzen CRUD-Cases keine richtige Administrationsoberfläche. Diese könnte in Form einer Webapplikation umgesetzt werden. Zudem könnte in dieser, die bestehende Access-App, integriert werden.

Anhang A

Deployment

A.1 Export VBA

Voraussetzungen:

- Microsoft Active X Data Objects 2.8
- ODBC 3.51 Treiber für MySQL (32Bit)
- Microsoft Excel

Die Makros zum Speichern der Daten, verwenden die Active X Data Objects. Deswegen kann das Makro nur auf einem Windows Rechner ausgeführt werden.

Im Modul „dbConnect“ sind 3 Methoden zu finden. In „ConnectDB“ können die Parameter für die Datenbank angegeben werden. Wichtig ist, dass die beiden Tabellen „movex“ und „palletSort“ in dieser Datenbank bereits existieren. Die beiden Subfunktionen „doInsertPalletPosition“ und „doInsertMovexData“ rufen für die INSERT Statements „ConnectDB“ für die Verbindung zur Datenbank auf.

Beschreibung:

Es wird über die beiden Arbeitsblätter MOVEX und Kernliste iteriert und die Daten in die Tabellen „movex“ bzw. „palletSort“ mittels INSERT-Statements geschrieben.

A.2 REST Service

Voraussetzungen:

- MySQL
- Java 8

Webserver: Der Webserver

(05_Implementation/Server/bico-server/bico-server-1.0-SNAPSHOT.zip) muss bloss entpackt und ausgeführt werden. Das Executable für das Starten des Servers lautet „bin/bico-server“. Unter Windows kann alternativ auch die Datei „bico-server.bat“ verwendet werden.

Die Konfigurationen des Webserver finden sich alle im File „conf/application.conf“:

Keyword	Beschreibung
db.default.url	Verbindung zur lokalen JDBC Verbindung: jdbc:mysql://localhost/bico
db.default.username	Username für oben stehende Verbindung. Gleicher Username wie im File „02_create_main_user.sql“ zu finden ist.
db.default.password	Passwort für oben stehende Verbindung
http.port	(Optional) Standardmässig läuft der Webserver auf Port 9000 und kann mit diesem Eintrag angepasst werden.

Tabelle A.1: Informationen zu application.conf

Datenbank: Um das Deployment des REST-Services möglichst einfach zu gestalten, wurden folgende SQL-Skripts für die Generierung der Datenbankstruktur erstellt. Diese sind unter „05_Implementation/Server/datenbank/scripts“ zu finden.

SQL Script	Beschreibung
01_create_db.sql	Erstellt die Datenbank mit dem Namen bico.
02_create_main_user.sql	Erstellt einen User bico, mit dem der Server auf die Datenbank zugreifen kann.
03_create_tables.sql	Erstellt die zahlreichen Datenbanktabellen.
04_create_app_users.sql	Beispiel für das Erstellen von Usern, die sich in der Android App anmelden können. Detailliertere Informationen zu den Usern sind in der Tabelle A.3 zu finden.

Tabelle A.2: SQL-Skripte für das Deployment

User: Das User-Management der Android App wird über die Tabelle „user“ gesteuert. Neue User können momentan über SQL INSERT Befehle hinzugefügt werden. In der Datei „04_create_app_users.sql“ findet sich ein Beispiel:

Spalte	Beschreibung
username	Der Username wird vom User für den Login verwendet. Dieser sollte daher nicht allzu lang sein jedoch eindeutig.
email	E-Mail Adresse des Users
name	Name des Users, welcher in den Kernlisten angezeigt werden soll
password	Passwort für das Login in der App
role	Momentan gibt es die zwei Rollen 0 (Arbeiter) und 1 (Meister). In die Tabelle muss jeweils der Integer-Wert eingetragen werden.
token	Ist leer ausser der User ist gerade eingeloggt.

Tabelle A.3: Users für App-Login

A.3 Android App

Voraussetzungen:

- Android 19: Kitkat

A.3.1 Installation

Die Android-App wird als APK-Datei paketiert und befindet sich unter *„05_implementation/AndroidApp/bico-client.apk. Um die Datei auf ein Android – Tablet zu kopieren, wird am besten ein Cloud – Provider wie Google Drive oder Dropbox verwendet.*

Wird die Datei via E-Mail übertragen, kann es zu Problemen kommen.

Auf dem Tablet muss die Datei nur noch ausgeführt werden. Womöglich muss zuerst in den Einstellungen noch der Punkt „Installationen aus Unbekannten Quellen“ erlaubt werden.

A.3.2 Anpassung

Müssen die Verbindungsparameter angepasst werden, so muss das APK entpackt und die Datei „res/values/strings.xml“ verändert werden. Dafür befinden sich im Verzeichnis „redeploy“ zwei Skripte. Das erste Skript erlaubt das APK zu entpacken. Mit dem zweiten Skript wird ein neues APK erstellt. Das neue APK wird unter dem Namen „bico-client-updated.apk“ ersichtlich sein. Dabei wird es automatisch neu signiert.

Die Adresse und der HTTP-Port für die Server-Verbindung, können in der Datei „res/values/strings.xml“ verändert werden. Dabei müssen die Werte mit den Namen „server“ und „port“ angepasst werden.

Anhang B

Eigene Reviews

B.1 Philippe Naegeli

Die Studienarbeit habe ich als sehr interessant und herausfordernd empfunden. Ich habe es sehr geschätzt, dass sich Arthur Schmucki und Pascal Kuster zu Beginn der Studienarbeit sich extrem Mühe für eine detaillierte Einführung in die Produktion von BICO gegeben haben. Zudem fand ich es extrem positiv, dass wir unsere eigenen Ideen in das Projekt einbringen konnten.

So waren wir zum Beispiel bei der Architektur nur an das Android Tablet gebunden, den Rest durften wir selbst entscheiden. Da wir bereits beim Softwareprojekt 2 im letzten Semester eine Android App mit einem separaten Backend umgesetzt hatten, konnten wir uns für die gleichen Systemkomponenten entscheiden.

Es war gut, dass sich die Arbeit so einfach in die zwei Themen Server und Client aufteilen liess. So konnten wir uns die Arbeit während der Implementationsphase wunderbar aufteilen. Generell war das Arbeitsklima mit Raffael Ioannone immer sehr positiv. Wir konnten uns sehr gut ergänzen und ich denke, wir haben schlussendlich keine schlechte Arbeit abgeliefert.

Extrem spannend fand Ich auch, dass wir während dieser Studienarbeit auch an einen Film für den Studiengang Informatik mitmachen konnten. Es war eine extrem neuwertige Erfahrung und vor allem mal was neues vor und nicht immer hinter der Kamera zu stehen.

Verbesserungspotenzial liegt sicherlich darin, dass wir in der Startphase des Projektes enger zusammen Arbeiten erledigen und wo es Sinn macht Pair Programming durchführen.

B.2 Raffael Ioannone

Das Thema war von Anfang an spannend und lehrreich. Ich konnte einiges an Erfahrungen sammeln, im Bezug auf das Erfassen von Requirements beim Kunden. Es wurde mir klar wie wichtig Mockups zu Beginn eines Projekts sein können.

Schön war, dass wir von Dr. Keller und der Hilding Anders Switzerland AG sehr viele Freiheiten bekommen haben. Insbesondere Arthur Schmucki hat sich sehr viel Zeit genommen unsere Fragen ausführlich zu beantworten. Aufgrund dieses Feedbacks konnten wir uns von Mal zu Mal verbessern.

Die Dreharbeiten für den Werbefilm der HSR habe ich als spannende Abwechslung empfunden.

Ich wurde beim Programmieren der Android-App oft mit unerwarteten Fehlermeldungen konfrontiert und musste erst ein geeignetes Vorgehen entwickeln, um solche Probleme anzugehen. Dies hat zum Teil auch das Analysieren der Google APIs bedeutet, wo ich einige Beispiele für schönen aber auch schlechten Code sehen konnte.

Für zukünftige Arbeiten wäre es sicher sinnvoll, wenn wir früher und intensiver mit den Endanwendern der App (Staplerfahrer) Tests durchführen und diese Erkenntnisse in den Code einfließen lassen.

Ich freue mich darauf das Projekt in unserer Bachelorarbeit weiterführen zu dürfen.

Anhang C

Risikoanalyse

Die Tabelle C.1 wurde während des Requirement Engineerings erstellt und widerspiegelt den Status zu Projektbeginn.

Nr	Titel	Beschreibung	max. Schaden [h]	Eintrittswahrscheinlichkeit	Gewichteter Schaden	Vorbeugung	Verhalten beim Eintreten
R1	Wiederverwendbarkeit der Vorgängerarbeit	Wir können keinen Code von der Vorgängerarbeit übernehmen aufgrund von verschiedenen Frameworks und Java-Versionen.	25	30%	7.5	Falls möglich die gleichen Softwarekomponenten verwenden, welche die Verwendung des Codes möglich macht.	An Standard-Libraries von Java und Android halten. Dadurch auf in Zukunft kompatibel und supported. Zudem sollte einiges an Hilfsmaterial online verfügbar sein.
R2	Data-Import	Der Datenimport von der Datenbank oder dem Excel-Sheet stellt sich als schwieriger heraus, als ursprünglich gedacht.	10	30%	3	Versuchen das SQL aus den VBA Scripts zu verwenden oder den Import-Mechanismus aus der Vorgängerarbeit.	Informationen über die DB einfordern, neue SQL-Statements schreiben. Online nach Excel-Importmechanismen suchen.
R3	User Interaction Problem	Den Gabelstapler-Fahrer ist un-user Interface zu kompliziert oder zu wenig intuitiv. Deshalb verzichten sie auf die Verwendung der Lösung.	20	20%	4	Kritik umsetzen aus der Vorgängerarbeit. Immer wieder mit den Kunden anschauen, überprüfen und nachbessern.	Redesign des Layouts von Android.
R4	Gruppierungs-Algorithmus	Der Algorithmus für die Gruppierung der Kerne kommt nicht an die Qualität der bestehenden Excel-Lösung hin. Der Aufwand für die Gabelstapler-Fahrer erhöht sich dadurch.	17	35%	5.95	Bestehenden Code genau analysieren. Einfach anpassbares Konstrukt entwerfen. Tolerante Gruppierung anwenden.	Spezialfälle verallgemeinern. Mit Auftraggeber über Optimierungsmöglichkeiten diskutieren.
R6	Sortier-Algorithmus	Die Sortierung ist nicht optimal oder es gibt zuviele Spezialfälle die zu beachten sind. Der Stapel könnte dadurch instabil werden.	17	60%	10.2	Genaue Kriterien vom Kunden erfassen. Von Endanwender überprüfen lassen.	Sich zuerst auf Grundkriterien fokussieren (Grösse und Familie) und zusätzliche Kriterien als optional deklarieren.
R7	WLAN-Empfang	Der Empfang im Lager ist nicht durchgehend stabil. Es kommt zu Verbindungsunterbrüchen, in denen die App nicht korrekt bedient werden kann.	25	80%	20	Kommunikations-Schnittstellen Fehlertolerant designen. Zwischenergebnisse Persistieren. Asynchrone Kommunikation.	Evt. Nach Messaging Lösung für Android schauen.
Summe			114		50.65		

Tabelle C.1: Risikoanalyse zu Projektbeginn

Abbildungsverzeichnis

1.1	Screenshot Android	6
3.1	Ausgangslage, Bilderquelle: [3]	11
3.2	Übersicht Produktionsprozess	12
3.3	Auszug Kernliste	14
3.4	Auszug Gesamtliste	15
3.5	Auszug Einzugsliste	15
3.6	Auszug Reissverschlussliste	16
3.7	Auszug Reporting	16
3.8	Use Case Diagramm	22
3.9	Mockup Login	26
3.10	Mockup Einstiegsseite	27
3.11	Mockup Übersicht Kernliste	27
3.12	Mockup Detailansicht Kernliste	28
3.13	Mockup Kern Information	28
3.14	Domain-Modell	31
3.15	Domain-Modell Projekt	32
4.1	Kontextdiagramme	35
4.2	Verteilungsdiagramm	36
4.3	Schichtendiagramm	37
4.4	hibernate.cfg.xml	39
4.5	JPA-Annotationen in den Entities	40
4.6	build.sbt	40
4.7	Stornierte Matratzenkerne	41
4.8	LoginAuthenticator	43
4.9	Ablauf OrderImporter	44
4.10	Sequenzdiagramm PalletGenerator	45
4.11	Berechnung des Status einer Palette	47
4.12	Ablauf eines Requests zwischen Android und Server	51
4.13	Ablauf Android	52
4.14	Datenbankschema	53
5.1	Übersicht Travis CI	56

6.1 Geleistete Arbeit über die komplette Projektdauer 59

Glossar

APK Fileformat, welches unter Android für die Verteilung bzw. Installation von Software verwendet wird

AS400 Server der Firma IBM: Auch unter dem Namen iServer bekannt

CSV Tabellenähnliches Dateiformat bei welchem die Felder via Komma abgetrennt sind

ERP System für die Planung von Ressourcen

HTML Textbasierte Sprache zur Anzeige von Text und Bildern

HTTP Netzwerkprotokoll für die Versendung von Daten

Kitkat Version 4.4 des Android Betriebssystems

Makros Ausführbare Programme in VBA

MySQL Datenbankprovider

NFC Technologie zum Austausch von Daten per Funktechnik

OR-Mapper Bildet Objektrelationelle Elemente auf Datenbanken ab

RFID Technologie von elektronischen Wellen zur drahtlosen Lokalisieren von Objekten

SQL Structured Query Language für die Abfrage von Datenbanken

TCP Protokoll der Transportschicht

VBA Visual Basic for Application: Sprache welches ermöglicht automatisieren in Microsoft Office Produkten vorzunehmen

Literaturverzeichnis

- [1] RedHat Community. Hibernate documentation. <http://hibernate.org/orm/>, 2015. [Online; accessed 30-September-2015].
- [2] FasterXML. BWorld Robot Control Software. <https://github.com/FasterXML/jackson-dataformat-csv>, 2015. [Online; accessed 19-October-2015].
- [3] freepik. Bilder für Grafik. <http://www.flaticon.com/>, 2015. [Online; accessed 17-Dezember-2015].
- [4] Google. Google Material Design Guidelines. <https://www.google.com/design/spec/material-design/introduction.html>, 2015. [Online; accessed 16-Dezember-2015].
- [5] Marcel Loop and Philipp Koster. Prozessoptimierung mit mobilen devices, 2015.
- [6] Typesafe. Play 2.4.x documentation. <https://www.playframework.com/documentation/2.4.x/Home>, 2015. [Online; accessed 19-September-2015].