

Bachelorarbeit

IoT Konsignationslagermanagement

Iwan Epp & Oswald Föhn

6. Juni 2019



Bachelorarbeit an der Hochschule Luzern – Informatik

Titel: IoT Konsignationslagermanagement

Studentin / Student 1: Iwan Epp

Studentin / Student 2: Oswald Föhn

Studiengang: BSc Informatik

Abschlussjahr: 2019

Betreuungsperson: Prof. Dr. René Meier

Expertin / Experte: Roman Bachmann

Codierung / Klassifizierung der Arbeit:

- A: Einsicht (Normalfall)**
- B: Rücksprache** (Dauer: Jahr / Jahre)
- C: Sperre** (Dauer: Jahr / Jahre)

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich/wir die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt haben, alle verwendeten Quellen, Literatur und andere Hilfsmittel angegeben haben, wörtlich oder inhaltlich entnommene Stellen als solche kenntlich gemacht haben, das Vertraulichkeitsinteresse des Auftraggebers wahren und die Urheberrechtsbestimmungen der Fachhochschule Zentralschweiz (siehe Markblatt «Studentische Arbeiten» auf MyCampus) respektieren werden.

Ort / Datum, Unterschrift _____

Ort / Datum, Unterschrift _____

Abgabe der Arbeit auf der Portfolio Datenbank:

Bestätigungsvisum Student/in

Ich bestätige, dass ich die Bachelorarbeit korrekt gemäss Merkblatt auf der Portfolio Datenbank abgelegt habe. Die Verantwortlichkeit sowie die Berechtigungen habe ich abgegeben, so dass ich keine Änderungen mehr vornehmen kann oder weitere Dateien hochladen kann.

Ort / Datum, Unterschrift _____

Ort / Datum, Unterschrift _____

Verdankung

Wir möchten uns bei allen bedanken, die uns bei dieser Bachelor-Diplomarbeit unterstützt haben. Allen voran Flavio Balsemin (B. Braun Medical AG), er hat uns die Möglichkeit gegeben, dieses spannende Projekt mit der B. Braun und ihm durchzuführen. Er stand uns immer mit Rat und Tat zur Seite und hat uns sehr viel Spielraum für unsere eigenen Lösungsansätze gelassen.

Prof. Dr. René Meier, er unterstütze uns in allen projektspezifischen Belangen, sei es, um uns präventiv vor Fehlern zu warnen oder uns bei der Strukturierung von Inhalten zu unterstützen. Des Weiteren möchten wir uns bei Micha Burger (Swisscom) für die tatkräftige Unterstützung zu Beginn des Projekts bei der LoRa Evaluation und dem super Tipp betreffend des Pycom LoPy bedanken.

Iwan Epp und Oswald Föhn,
Rotkreuz, 06.06.2019

Eingangsvisum (durch das Sekretariat auszufüllen):

Rotkreuz, den _____ Visum: _____

Hinweis: Die Bachelorarbeit wurde von keinem Dozierenden nachbearbeitet. Veröffentlichungen (auch auszugsweise) sind ohne das Einverständnis der Studiengangleitung der Hochschule Luzern – Informatik nicht erlaubt.

Copyright © 2019 Hochschule Luzern – Informatik

Alle Rechte vorbehalten. Kein Teil dieser Arbeit darf ohne die schriftliche Genehmigung der Studiengangleitung der Hochschule Luzern – Informatik in irgendeiner Form reproduziert oder in eine von Maschinen verwendete Sprache übertragen werden.

I. Abstract (DE)

Die Firma B. Braun Medical AG hat ein verwaltungstechnisches Problem bei sich festgestellt. Die Bewirtschaftung ihrer dezentralen Warenlager in Spitälern (Konsignationslager) ist für Aussendienstmitarbeiter sowie den Spitalangestellten aufwändig und erfordert viel Handarbeit (führen von Listen und Excel Tabellen).

Ziel der Bachelor-Diplomarbeit ist es, den Prozess der Lagerbewirtschaftung zu analysieren und mittels eines Prototyps zu zeigen, wie eine elegante IoT-Lösung aussehen könnte, welche den Anwendern die Arbeit erleichtert. Mittels eigens zusammengestellter Hardware und darauf abgestimmter Software sollen die Konsignationslager mit einer zentralen Verwaltung vernetzt werden. Diese wird über Änderungen des Lagerbestands informiert und stellt die Verwaltungsfunktionalität zur Verfügung.

Diese Arbeit zeigt die Entwicklung eines smarten LoRa-Geräts, welches mit einer einfach bedienbaren Benutzeroberfläche allen Anforderungen der Aussendienstmitarbeiter sowie des Spitalpersonals gerecht wird und damit die Bewirtschaftung der Konsignationslager markant vereinfacht. Es wird aufgezeigt, wie in den Zeiten von Industrie 4.0 Lösungen entwickelt werden können, um manuelle Aufgaben effizient und ohne grossen Aufwand zu erledigen.

II. Abstract (EN)

The company B. Braun Medical AG has identified an administrative problem. The management of their decentralized warehouses in hospitals (consignment stores) is too complex for the company's field staff and hospital employees. It requires a high degree of manual work (keeping lists and Excel files).

The aim of the bachelor thesis is to analyse the warehouse management process. Furthermore, a prototype will be used to show how a more efficient and smart IoT solution could look like, which would make work easier for the users. The consignment warehouses are to be networked with a central administration by means of specially compiled hardware and software. The latter will be informed about changes in the warehouse stock and will provide the management functionality.

This work shows the development of a smart LoRa device with an easy-to-use user interface that meets all the requirements of field staff and hospital staff and thus significantly simplifies the management of consignment stocks. It will be shown how solutions can be developed in the age of industry 4.0 in order to carry out manual tasks efficiently and without great additional effort.

III. Inhaltsverzeichnis

1.	Projektbeschreibung	1
1.1.	Einleitung und Ausgangslage	1
1.2.	Erwartete Ziele und Resultate	1
2.	Planung	2
2.1.	Vorgehensmodell.....	2
2.2.	Projektorganisation.....	2
2.2.1.	Aufteilung der Arbeiten	2
2.3.	Rahmenplan.....	3
2.3.1.	Meilensteine	3
2.4.	Risikomanagement.....	4
2.4.1.	Massnahmen zur Verminderung der Risiken.....	5
2.5.	Projektunterstützung.....	5
2.5.1.	Budget.....	5
2.5.2.	Zeit.....	6
2.5.3.	Tools.....	7
3.	Konzeption	8
3.1.	Begrifflichkeiten.....	8
3.2.	Anforderungen	8
3.2.1.	Akteure	9
3.2.2.	Anwendungsfälle.....	11
3.2.3.	Systemanforderungen.....	22
3.2.4.	Funktionale Anforderungen.....	24
3.2.5.	Zuweisung der Anforderungen	26
3.3.	Technologien.....	28
3.3.1.	LoRa / LoRaWAN.....	28
3.3.2.	Python.....	29
3.3.3.	Micropython.....	29
3.3.4.	C#.....	29
3.3.5.	Angular	29
3.3.6.	Apache HTTP Server.....	30
3.3.7.	Internet Information Services (IIS)	30
3.3.8.	Microsoft SQL Server	30
3.4.	Lösungskonzept	30
3.4.1.	E-Konsignationslager	31
3.4.2.	Lagerverwaltung	32
3.4.3.	Kommunikation	34
3.5.	Design	35
3.5.1.	Vorgaben	35
3.5.2.	Eigene Designelemente	38
4.	Realisierung	42

4.1.	Architektur Backend Lagerinterface.....	43
4.1.1.	Datenpersistenz.....	44
4.1.2.	API-Schnittstelle.....	45
4.1.3.	Kommunikation mit LoRa Modul.....	50
4.1.4.	Konfiguration.....	53
4.1.5.	Konfiguration für Fehleranalyse.....	55
4.2.	Architektur Backend Lagerverwaltung.....	55
4.2.1.	Datenpersistenz.....	56
4.2.2.	API-Schnittstelle.....	58
4.2.3.	Konfiguration.....	59
4.3.	Architektur Frontend Lagerinterface und Lagerverwaltung	60
4.4.	Kommunikation	61
4.4.1.	Verbindung.....	61
4.4.2.	Zuverlässigkeit	62
4.4.3.	Datensicherheit.....	62
4.4.4.	Datenschutz.....	62
4.4.5.	Nachrichten	62
4.4.6.	Datenkonvertierung	64
4.5.	Lagerbestand.....	65
4.5.1.	Ändern.....	65
4.5.2.	Mutationen übertragen.....	67
4.5.3.	Überwachen	68
4.6.	Benutzerverwaltung	69
4.6.1.	Privilegien.....	69
4.7.	Betrieb E-Konsignationslager.....	70
4.7.1.	Verbindungen.....	71
4.7.2.	Systemstatus.....	71
4.8.	Artikelstamm.....	74
4.8.1.	Importieren des Artikelstamms	75
4.8.2.	Artikelstamm synchronisieren.....	75
4.9.	Chargen	76
4.10.	Kunden.....	77
5.	Inbetriebnahme	78
5.1.	Lagerverwaltung.....	78
5.2.	Swisscom LPN	78
5.3.	Lagerinterface.....	78
6.	Validierung.....	79
6.1.	Tests während der Entwicklung.....	79
6.2.	Definition von realen Bedingungen.....	80
6.3.	Überprüfung der Anwendungsfälle	80
6.4.	Überprüfung der Anforderungen	80
6.5.	Gesamt-Test des Systems.....	80

7.	Evaluation.....	81
7.1.	Stärken.....	81
7.1.1.	Bedienung.....	81
7.1.2.	Inbetriebnahme.....	81
7.1.3.	Abdeckung.....	81
7.1.4.	Lagerbewirtschaftung.....	81
7.1.5.	Robustheit.....	81
7.2.	Schwächen.....	81
7.2.1.	Anzeige.....	81
7.2.2.	Wartbarkeit.....	81
7.3.	Limitierungen.....	82
7.3.1.	Verbindung.....	82
7.3.2.	Abhängigkeit von der Swisscom.....	82
8.	Abschluss und Ausblick.....	83
8.1.	Lessons Learned.....	83
9.	Anhänge.....	84
9.1.	Anhang 1: Aufgabenstellung.....	84
9.2.	Anhang 2: Anforderungsliste.....	84
9.3.	Anhang 3: Inbetriebnahme Lagerverwaltung.....	84
9.4.	Anhang 4: Inbetriebnahme Swisscom LPN.....	84
9.5.	Anhang 5: Inbetriebnahme Lagerinterface.....	84
9.6.	Anhang 6: Langzeittest.....	84
10.	Verzeichnisse.....	85
10.1.	Abbildungsverzeichnis.....	85
10.2.	Tabellenverzeichnis.....	86
11.	Literaturverzeichnis.....	88

IV. Glossar

Begriff	Erläuterung
Charge	Die Menge aller Einheiten (Produkte), welche innerhalb eines Produktionsprozesses unter denselben Bedingungen hergestellt werden.
Deliverables	Erwartete Ergebnisse eines Meilensteins
E-Konsignationslager	Konsignationslager inklusive Lagerinterface.
IoT	Internet of Things
Konsignationslager	Dezentrales Warenlager in der Nähe des Kunden. Der Inhalt des Lagers verbleibt im Eigentum des Herstellers, bis der Kunde Waren aus dem Lager entnimmt.
Lagerinterface	Soft- und Hardware zur Verwaltung und Überwachung der Lagerbestände.
Lagerverwaltung	Zentrale Verwaltungssoftware aller Konsignationslager.
LoPy	Erweiterbares Microcontroller-Entwicklungsboard
LoRa	Long Range, Funktechnologie. Detaillierte Erklärung siehe Kapitel 3.3.1: LoRa / LoRaWAN
Mutation	Änderung des Lagerbestands ausgelöst durch eine Benutzerinteraktion.
Provider	Dienstleister für die drahtlose Kommunikation zwischen E-Konsignationslager und Lagerverwaltung.
Raspberry Pi	Linux-basierter Microcomputer
Smallest Sellable Unit	Bezeichnet die kleinste Verkaufseinheit eines Produkts. Ein 12er Pack Chirurgenmesser wird zum Beispiel als ein Produkt gehandhabt.

Tabelle 1: Glossar

1. Projektbeschreibung

1.1. Einleitung und Ausgangslage

Im vergangenen Semester wurde im Rahmen des Wirtschaftsprojekts ein Proof of Concept erstellt, welches mit einem Hardware-Prototyp den automatisierten Betrieb eines Konsignationslagers aufzeigen sollte.

Diese Bachelorarbeit befasst sich mit der Weiterentwicklung des Prototyps und der gesamten Softwarelösung. Die Lösung soll stabiler und umfangreicher werden, als die des Wirtschaftsprojekts. Insbesondere sollen die Probleme der LoRa-Verbindung aus dem Wirtschaftsprojekt eliminiert werden. Die Diagnose-Möglichkeiten für den Betrieb der Konsignationslager soll zudem erhöht werden. Zusätzlich wird der ursprünglich vorgeschlagene Barcode-Typ durch den GS1 Barcode ersetzt, welcher zusätzliche Informationen wie Ablaufdatum und Chargennummer erhält. Des Weiteren sollen die grafischen Benutzerschnittstellen optisch an das Design der B. Braun Medical AG angepasst werden.

Im Anhang 1: Aufgabenstellung ist die Original-Aufgabenstellung zu finden.

- Anmerkungen in Bezug auf das vorausgegangene Wirtschaftsprojekt sind speziell mit diesem Zeichen ➤ gekennzeichnet. In Abbildungen sind diese blau markiert.

1.2. Erwartete Ziele und Resultate

Im Rahmen der Bachelor-Diplomarbeit (BDA) sollen die folgenden Ergebnisse erarbeitet werden.

- **Prozess und Anwendungsfälle:** Der Prozess der Lagerbewirtschaftung soll ausgearbeitet werden. Zusätzlich sollen die Anwendungsfälle dieses Prozesses definiert werden.
- **Lösungskonzept:** Erstellung eines Lösungskonzepts, unter Verwendung des IoT Ansatzes, für die Zukunft der Konsignationslagerbewirtschaftung basierend auf dem erarbeiteten Prozess.
- **Prototypenentwicklung:** Implementierung eines lauffähigen und robusten Prototyps, welcher anhand der Anforderungen und der Funktionen entwickelt und getestet wurde.
- **Design-Mockup:** Der B. Braun Brand Guide soll für die Entwicklung vom Frontend verwendet werden. Designvorschläge müssen durch den Wirtschaftspartner validiert werden.
- **Robustheit Prototyp:** Für die Evaluierung sollen möglichst reale Bedingungen berücksichtigt werden.
 - Wie dies bereits im Wirtschaftsprojekt der Fall war, werden zwei Hardwareprototypen erstellt. Dies hat den Vorteil, dass jeder Studierende unabhängig entwickeln kann.

In diesem Projekt nicht inbegriffen sind die effektive Einführung des neuen Prozesses, die Einbindung der neuen Lagerbestandsverwaltung ins B. Braun ERP-System und die Anbindung an die bisherigen Prozesse der Rechnungsstellung.

2. Planung

2.1. Vorgehensmodell

Als Vorgehensmodell wurde das V-Modell gewählt, weil die Phasen des Modells sehr gut mit dem Projektauftrag übereinstimmen. In den ersten drei Wochen des Projekts werden zusammen mit dem Auftraggeber die Anforderungen und Use-Cases definiert und abgesprochen (Systemanforderungsanalyse). Aufgrund der Anforderungen wird ein Lösungskonzept erstellt (System-Architektur / System-Entwurf). Im weiteren Verlauf des Projekts werden diese schrittweise umgesetzt (Software-Architektur / Software-Entwurf). Zur Qualitätssicherung werden automatisierte Unit- und Integrationstests, sowie manuelle Gesamtsystemtests eingesetzt (System-Integration). Die Abnahme der Anforderungen erfolgt durch eine Vorführung der Funktionalität beim Auftraggeber (Abnahme und Nutzung).

2.2. Projektorganisation

Die nachfolgende Abbildung zeigt die Projektorganisation.

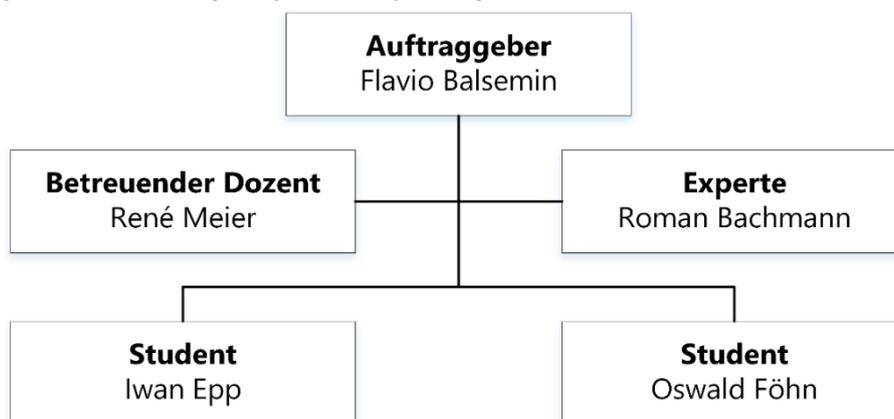


Abbildung 1: Projektorganisation

2.2.1. Aufteilung der Arbeiten

Um Parallelität im Projekt zu gewährleisten, wurde für bestimmte Arbeiten ein Hauptverantwortlicher festgelegt.

Hauptverantwortlicher	Arbeit
Iwan Epp	Anforderungen Lösungskonzept Entwicklung Lagerinterface (Backend) LoRa Kommunikation Validierung
Oswald Föhn	Terminplanung Risikomanagement Buchhaltung Design-Mockups Entwicklung Lagerinterface (Frontend) Entwicklung Lagerverwaltung (Frontend & Backend) Inbetriebnahme

Tabelle 2: Aufteilung der Arbeiten

2.3. Rahmenplan

Der nachfolgende Rahmenplan (Abbildung 2) zeigt die wichtigsten Termine und Meilensteine des Projekts auf. Für die ersten drei Wochen wurde eine Detailplanung erstellt, da viel Kommunikation mit dem Auftraggeber nötig war. Nachdem die Anforderungen und das Lösungskonzept besprochen wurden, wurde auf eine Detailplanung verzichtet, da der Umfang der Arbeiten mittels Priorisierung festgelegt wurde.

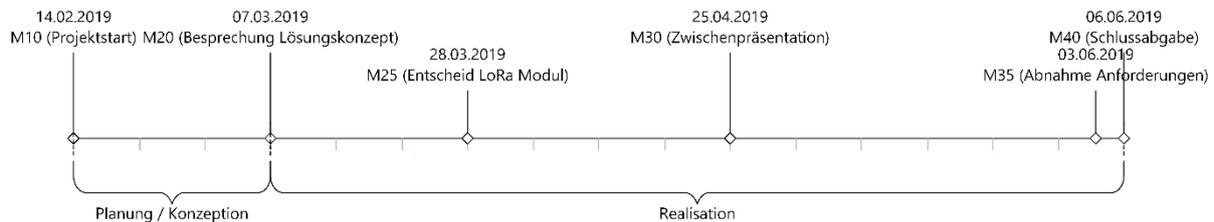


Abbildung 2: Rahmenplan

2.3.1. Meilensteine

In den folgenden Abschnitten (Tabelle 1) werden die definierten Meilensteine kurz erklärt.

Bezeichnung	Datum	Deliverables
M10 Projektstart	14.02.2019	Start der Bachelorarbeit Projektauftrag definiert und unterschrieben
M20 Besprechung Lösungskonzept	07.03.2019	Anforderungen definiert <ul style="list-style-type: none"> • Systemanforderungen • Anwendungsfälle Lösungskonzept finalisiert <ul style="list-style-type: none"> • Systemaufbau • Funktionskontrolle LoRa • Funktionskontrolle Barcodescanner • Hardwareaufbau
M25 Entscheid LoRa Modul	28.03.2019	Hardwareentscheid, welches LoRa Modul endgültig verwendet wird.
M30 Zwischenpräsentation	25.04.2019	Zwischenpräsentation vor Auftraggeber, betreuendem Dozenten und dem Experten. Erste Abnahme von Teilfunktionen mit dem Arbeitgeber. Dies wurde gemacht, um die Menge für den Meilenstein M35 zu verkleinern.
M35 Abnahme Anforderungen	03.06.2019	Abnahme der verbleibenden Anforderungen mit dem Auftraggeber.
M40 Schlussabgabe	06.06.2019	Ende der Bachelorarbeit Abgabe aller erforderlichen Dokumente

Tabelle 3: Meilensteine

2.4. Risikomanagement

Die folgenden Risiken wurden identifiziert und aufgrund der Eintritts-Wahrscheinlichkeit (W) und der Auswirkungen (A) auf das Projekt bewertet.

Nr.	Beschreibung	W	A
Projektbezogene Risiken			
1.1	Ein Projektteilnehmer kann seine Arbeit aufgrund neuer Verpflichtungen oder Verhinderungen nicht zeitgerecht durchführen.	1	4
1.2	Das Budget für Projektausgaben ist zu klein.	3	2
1.3	Nicht alle Anforderungen können erwartungsgemäss umgesetzt werden.	4	3
1.4	Verlust von Teilen der Arbeit (Code, Dokumentation)	2	4
Technologiebezogene Risiken			
2.1	Verbindung im dem LPN Netzwerk zeigt Mängel auf. Deshalb kann keine stabile Verbindung hergestellt werden.	5	3
2.2	Hardware-Komponenten haben Fehlfunktionen und/oder gehen bei der Verwendung kaputt.	2	2
2.3	Design-Mockup entspricht wiederholt nicht den Anforderungen des Kunden.	3	3

Tabelle 4: Risiken im Projekt

Die folgende Risikomatrix wurde verwendet, um die Risiken zu bewerten.

Eintritts-Wahrscheinlichkeit

6 – häufig

5 – wahrscheinlich

4 – gelegentlich

3 – vorstellbar

2 – unwahrscheinlich

1 – unvorstellbar

			2.1	
			1.3	
		1.2	2.3	
		2.2		1.4
				1.1
	1	2	3	4
	unwesentlich	gering	kritisch	katastrophal
	Auswirkung			

Tabelle 5: Risikomatrix

2.4.1. Massnahmen zur Verminderung der Risiken

Um das Ausmass der Auswirkungen oder die Wahrscheinlichkeit eines Eintritts zu reduzieren, werden die folgenden Massnahmen zum jeweiligen Risiko definiert.

Nr.	Massnahme
Projektbezogene Massnahmen	
1.2	Bei Neuanschaffungen wird zuerst eine Einheit gekauft, bevor diese für beide Prototypen angeschafft wird.
1.3	Die Anforderungen werden früh im Projekt spezifiziert, auf ihre Machbarkeit eingeordnet und geschätzt. Aufgrund der Aufwandschätzungen können gewisse Funktionen als «Optional» gekennzeichnet werden.
1.4	Folgende Applikationen werden verwendet, um das Risiko des Datenverlusts zu minimieren: Sharepoint-Server für Dokumentation Gitlab für Code Versionisierung
Technologiebezogene Massnahmen	
2.1	Schon in einem frühen Stadion des Projektes wird ein Meilenstein definiert, in welchem die Hardware und Software zur Kommunikation mit dem LPN Netzwerk festgelegt wird.
2.2	30% des Budgets werden als Reserve für defekte oder zerstörte Hardware betrachtet und können erst in den letzten 3 Wochen bezogen werden.
2.3	Grundlage für das Design wird bereits früh im Projekt geschaffen. Dazu werden direkt vom Kunden Rückmeldungen eingefordert.

Tabelle 6: Massnahmen zur Risikoverminderung

2.5. Projektunterstützung

In den folgenden Abschnitten werden die Ressourcen beschrieben, welche für die Bachelorarbeit zur Verfügung stehen.

2.5.1. Budget

Für Projektausgaben wurde vom Auftraggeber ein Budget von 1000.- CHF zur Verfügung gestellt. Mit diesem Budget müssen alle Hard- und Softwareausgaben abgedeckt werden können. Die verwendete Hardware aus dem vorangegangenen Wirtschaftsprojekt kann übernommen werden und entfällt somit aus den Budget-Plänen. Dazu gehören unter anderem:

- 2x Raspberry Pi 3 Model B+
- 2x Raspberry Pi Touch Display

Eine Auflistung der Hardware für den realisierten Prototyp findet sich im Abschnitt «Eingesetzte Hardware» des Kapitels «Lösungskonzept».

Da die Betriebsdauer des Systems über das Ende der Bachelorarbeit hinausgehen muss (AF#008 – Betriebsdauer), wurde das Serverhosting und das Swisscom LPN Konto vom Budget ausgenommen und liegt in der Verantwortung des Auftraggebers, damit es nach Projektende aufrechterhalten werden kann.

Für etwaige Softwaretools wurde soweit möglich mit den zur Verfügung stehenden Studentenlizenzen gearbeitet, um das Budget nicht zusätzlich zu belasten.

Um das Budget-Risiko zu vermindern, wird bei Neuanschaffungen zuerst nur eine Komponente für einen Prototypen beschafft und erst bei erfolgreicher Funktionskontrolle eine zweite Einheit für den anderen Prototypen gekauft.

In der nachfolgenden Tabelle sind die Projektausgaben aufgelistet.

Datum	Nr.	Lieferant	Betrag	Beschreibung
22.02.2019	1	Pycom	CHF 119.84	LoRa Modul für Test
22.02.2019	3	Pycom	CHF 29.35	LoRa Modul Versand
23.02.2019	2	ARP	CHF 285.00	Barcodescanner
03.04.2019	4	Mouser	CHF 133.34	Weiteres LoRa Modul
08.04.2019	5	Arthur Weber	CHF 4.05	Div. Schrauben
09.05.2019	6	Galaxus	CHF 42.00	Halterung
09.05.2019	7	ARP	CHF 280.00	Zweiter Barcodescanner
17.05.2019	8	Digitec	CHF 16.90	Kurze USB Kabel
17.05.2019	9	Pi-Shop	CHF 14.80	Micro USB Splitter Kabel
		Summe	CHF 925.28	
		Restbudget	CHF 74.72	

Tabelle 7: Kostenauswertung

2.5.2. Zeit

Die Bachelorarbeit schreibt einen Zeitaufwand von 360 Stunden pro Studierenden vor. Die aufgewendeten Stunden werden in einem Arbeitsjournal erfasst. Um eine Kontrolle über die geleisteten Arbeiten zu erhalten wurde ein Ampelsystem eingeführt, welches die geleistete Zeit mit einer linearen Verteilung der zu leistenden Stunden vergleicht. Im Ampelsystem bedeutet grün «Im Zeitplan», gelb bedeutet «1 – 3 Tage hinter dem Zeitplan» und rot bedeutet «Mehr als 3 Tage hinter dem Zeitplan». Zusätzlich wird der zeitliche Projektfortschritt in einem Burndown-Diagramm aufgezeichnet, um eine genauere Übersicht dazu zu erhalten. Das Diagramm zeigt den zeitlichen Verlauf der zu leistenden Arbeitsstunden.

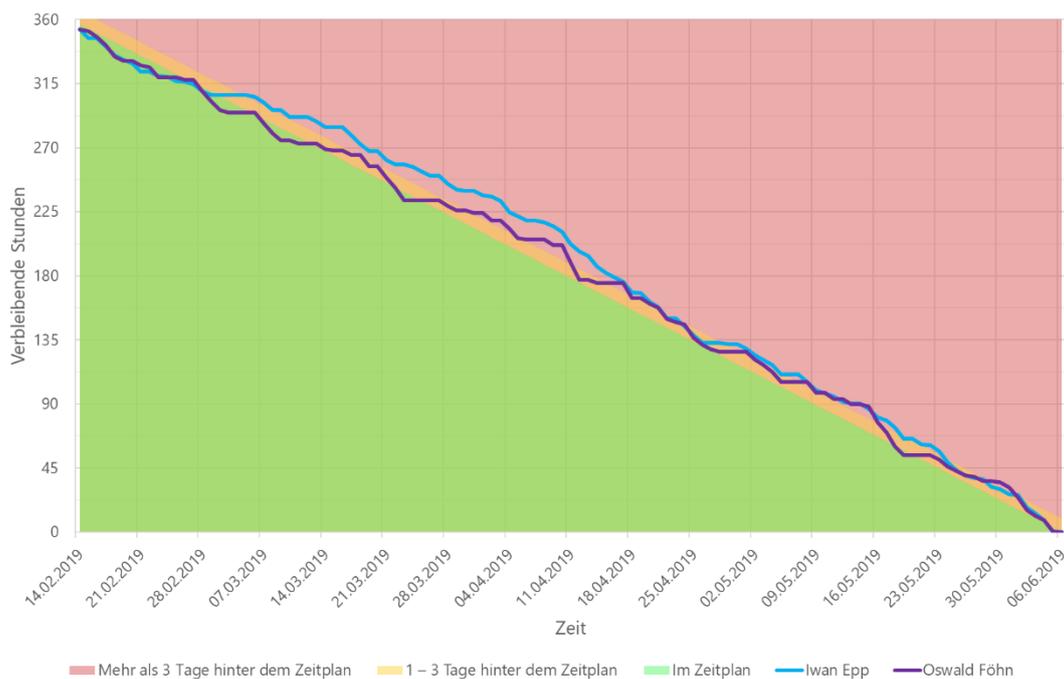


Abbildung 3: Auswertung Arbeitsjournal

2.5.3. Tools

In der folgenden Tabelle werden die verwendeten Tools aufgelistet und kurz beschrieben.

Name	Beschreibung
Microsoft Office 365 Plus Word, Excel, PowerPoint, Visio	Projektdokumentation
JetBrains PyCharm 18	Entwicklungsumgebung für Python
Atom 1.36.0	Entwicklungsumgebung für Micropython
Visual Studio 2017 Enterprise	Entwicklungsumgebung für C#
Git, Git Extensions	Versionsverwaltung Quellcode
Putty, VNC Viewer, FileZilla	Tools für Fernzugriff auf Raspberry Pi
SQL Server Management Studio 2017	Verwaltungstool für SQL Datenbank

Tabelle 8: Unterstützende Tools während der Projektarbeit

3. Konzeption

In diesem Kapitel wird das geplante Vorgehen beschrieben. Zuerst werden die Anforderungen an das zu realisierende System definiert. Danach werden die verwendeten Schlüsseltechnologien aufgelistet. Das darauffolgende Lösungskonzept zeigt auf, wie das System in Form eines Prototyps umgesetzt wird. Der Design Prozess wird im letzten Kapitel genauer erläutert.

3.1. Begrifflichkeiten

Die folgende Abbildung zeigt symbolisch die Aufteilung des Gesamtsystems. Die einzelnen Komponenten werden im Kapitel 3.4 Lösungskonzept erläutert. Das Gesamtsystem besteht aus einer zentralen Lagerverwaltung (rechts) und den E-Konsignationslager (links). Ein E-Konsignationslager besteht aus einem bestehenden Konsignationslager (Warenlager) und dem, in dieser Arbeit entwickelten, Lagerinterface.

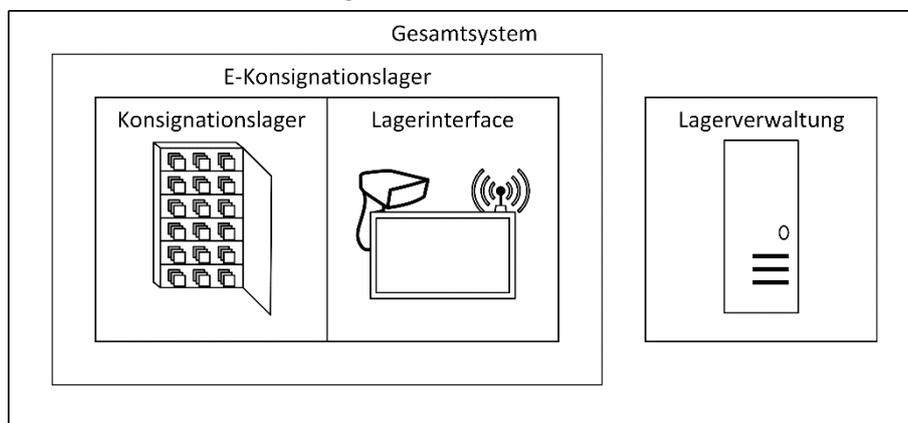


Abbildung 4: Aufteilung des Gesamtsystems

3.2. Anforderungen

- Einige der Anwendungsfälle stammen zu Teilen aus dem vorangegangenen Wirtschaftsprojekt. Sie wurden aber für diese Arbeit überarbeitet und verfeinert. Diese sind in den Anwendungsfalldiagrammen blau gekennzeichnet.

In den folgenden Abschnitten werden die Anwendungsfälle und Anforderungen an das neue System beschrieben. Zu Beginn des Projekts wurden mehrere Besprechungen mit dem Auftraggeber durchgeführt, um den definitiven Funktionsumfang des Prototyps festzulegen. Die Anforderungen werden in einem separaten Dokument geführt (siehe Anhang 2: Anforderungsliste). In diesem Dokument wird eine vereinfachte Form der Anforderungen aufgelistet, wobei der Status und die Änderungshistorie weggelassen wird.

Die realisierten Anforderungen werden zusammen mit dem Auftraggeber besprochen und abgenommen, dies ist in der originalen Anforderungsliste vermerkt.

3.2.1. Akteure

In den folgenden Anwendungsfällen werden bestimmte Akteure verwendet. Diese werden nachfolgend anhand eines Beispiels definiert.

Kunde

Name	Sarah Klinik
Beruf	Fachfrau Gesundheit (FaGe)
Arbeitgeber	Kantonsspital Luzern
Hintergrund	Sarah hat soeben die Ausbildung zur Fachfrau Gesundheit abgeschlossen. Sie ist eine sehr motivierte Person, die für ihre Abrufbarkeit von den Ärzten geschätzt wird. Bei Behandlungen besorgt sie die benötigten Materialien und bezieht diese gegebenenfalls auch aus den Konsignationslagern der B. Braun Medical AG. Beruflich wie auch privat hat sie nicht viel mit Technik und Computern zu tun. Das Arbeiten mit Menschen macht ihr sehr viel Spass.
Ansprüche ans E-Konsignationslager	Sarah will Produkte aus dem Lager beziehen. Wichtig ist ihr, dass sie Produkte einfach und schnell beziehen kann.

Tabelle 9: Beschreibung Akteur Kunde

Aussendienstmitarbeiter

Name	Hans Agent
Beruf	Aussendienstmitarbeiter
Arbeitgeber	B. Braun Medical AG
Hintergrund	Hans ist eine offene, kommunikationsfreudige Person. Er kann sich gut in Leute hineinversetzen und ist teamfähig. Deshalb hat er auch die Stelle als Aussendienstmitarbeiter bei der B. Braun Medical AG angenommen. Er arbeitet gerne mit Personen und ist stets versucht, bei Problemen zu vermitteln. Neben der Arbeit mit Menschen ist er auch technisch sehr begabt und probiert mit Freude neue Technologien aus. In letzter Zeit nervt er sich immer wieder über den mühsamen Ablauf in den Konsignationslagern, welche er betreut. Er hat bereits versucht die Bestellvorgänge etc. für sich zu optimieren, jedoch hat er immer wieder Probleme, dass die Lagerbestände nicht mit seinen Aufzeichnungen übereinstimmen.
Ansprüche ans E-Konsignationslager	Hans will zusammen mit dem E-Konsignationslager seine Arbeit vereinfachen, er will auch nicht darauf angewiesen sein, dass ihn das Spital immer anruft, um Produkte zu bestellen. Dies empfindet er als unnötigen Aufwand im Zeitalter der Automatisierungen.

Tabelle 10: Beschreibung Akteur Aussendienstmitarbeiter

Technischer Mitarbeiter

Name	Peter Geek
Beruf	Technischer Mitarbeiter
Arbeitgeber	B. Braun Medical AG
Hintergrund	Peter ist eine sehr technikaffine Persönlichkeit. Er ist ein gewissenhafter Arbeiter, der es liebt, neue Dinge auszuprobieren und mit unkonventionellen Mitteln zu arbeiten. Er hält sich akribisch auf dem aktuellsten Stand der Technik. Zuhause hat er bereits angefangen, seine Eigentumswohnung mit IoT-Geräten auszurüsten, welche via LoRa kommunizieren.
Ansprüche ans E-Konsignationslager	Peter will sicherstellen, dass die E-Konsignationslager, welche er für Kunden vorbereitet, bestmöglich funktionieren.

Tabelle 11: Beschreibung Akteur Technischer Mitarbeiter

3.2.2. Anwendungsfälle

Die nachfolgenden Diagramme beschreiben die im System vorhandenen Anwendungsfälle. Diese sind aufgeteilt in die Kategorien «Kundenbetrieb» und «Technischer Betrieb». Die Anwendungsfälle der ersten Kategorie (Abbildung 5) finden hauptsächlich vor Ort beim Kunden statt und betreffen hauptsächlich den Betrieb. Die Anwendungsfälle der zweiten Kategorie (Abbildung 6) dienen der Inbetriebnahme und der Störungsbehebung.

- Die folgenden Anwendungsfälle (blau markiert) «Lagerbestand korrigieren», «Lagerbestand anzeigen», «Waren einlagern», «Waren kaufen» «Lagerinterface konfigurieren» und «E-Konsignationslager in Betrieb nehmen» haben ihren Ursprung aus dem Wirtschaftsprojekt. Sie wurden im Rahmen dieser Arbeit komplett überarbeitet und teilweise umbenannt. In der Änderungshistorie ist dies ersichtlich.

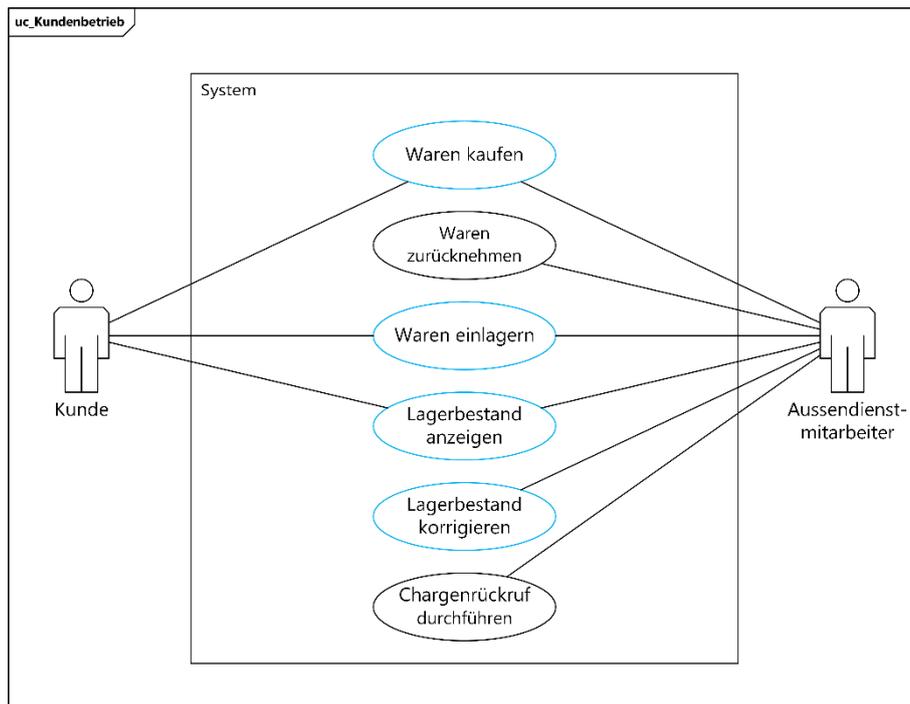


Abbildung 5: Anwendungsfälle Kundenbetrieb

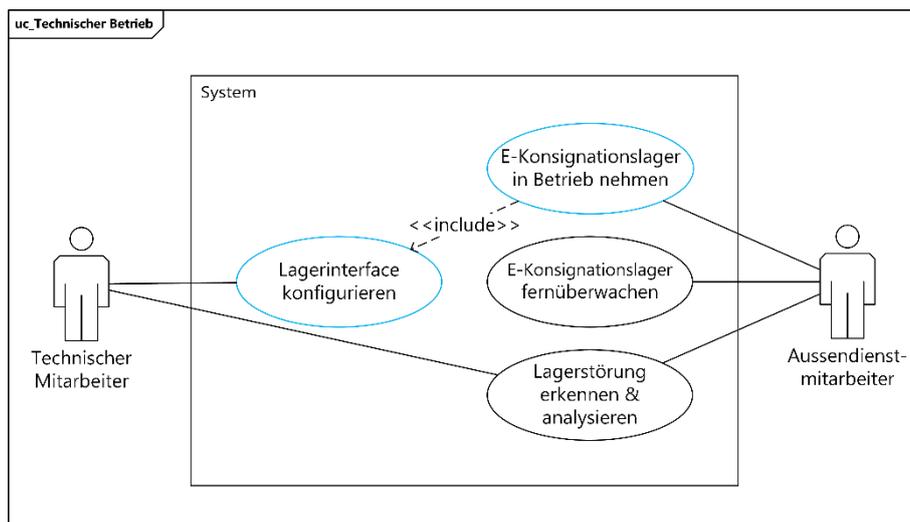


Abbildung 6: Anwendungsfälle Technischer Betrieb

Use-Case «Waren kaufen»

Anwendungsfallbeschreibung			
Name	Waren kaufen		
Kurzbeschreibung	Ein Kunde bezieht Ware aus dem E-Konsignationslager. Die entnommene Ware wird in der Lagerverwaltung abgebucht.		
Akteure	Kunde / Aussendienstmitarbeiter E-Konsignationslager, Lagerverwaltung		
Auslöser	Der Akteur möchte Waren kaufen.		
Ergebnisse	<p>Der Akteur erhält die gewünschte Ware. Ein Kauf im rechtlichen Sinne hat stattgefunden. Der Lagerbestand ist nachgeführt. Produktinformationen werden angezeigt. Der Akteur erhält eine Rückmeldung ob</p> <ul style="list-style-type: none"> - ein Chargenrückruf ausstehend ist - das Verfallsdatum erreicht ist 		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	Barcode der Ware, die gekauft wird		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. In den Betriebsmodus «Produktkauf» wechseln 2. Ware entnehmen 3. Barcode scannen 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Übernommen aus Wirtschaftsprojekt (Epp & Föhn, 2018)
	15.02.19	IE	Diverse Anpassungen
	18.02.19	OF	Erweiterung Benachrichtigung
Sonstiges	Der Akteur hat die Möglichkeit den Kauf innerhalb einer kurzen Zeit rückgängig zu machen.		

Tabelle 12: Anwendungsfall «Waren kaufen»

Use-Case «Waren zurücknehmen»

Anwendungsfallbeschreibung			
Name	Waren zurücknehmen		
Kurzbeschreibung	Ein Aussendienstmitarbeiter entnimmt Ware aus dem E-Konsignationslager. Die entnommene Ware wird in der Lagerverwaltung abgebucht.		
Akteure	Aussendienstmitarbeiter E-Konsignationslager		
Auslöser	Der Aussendienstmitarbeiter will abgelaufene oder bald ablaufende, sowie mangelhafte Produkte aus dem E-Konsignationslager entfernen.		
Ergebnisse	Die gewünschten Produkte sind im E-Konsignationslager nicht mehr vorhanden. Es hat kein Kauf stattgefunden. Der Lagerbestand ist nachgeführt. Produktinformationen werden angezeigt. Der Akteur erhält eine Rückmeldung ob <ul style="list-style-type: none"> - ein Chargenrückruf ausstehend ist - das Verfallsdatum erreicht ist 		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	Barcode der Ware, die zurückgenommen wird		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. In den Betriebsmodus «Produktrücknahme» wechseln 2. Rücknahme Grund auswählen 3. Produkt entnehmen 4. Barcode scannen 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Erste Version erstellt
	18.02.19	OF	Anpassungen Produktrücknahme
Sonstiges	Der Akteur hat die Möglichkeit die Rücknahme innerhalb einer kurzen Zeit rückgängig zu machen.		

Tabelle 13: Anwendungsfall «Waren zurücknehmen»

Use-Case «Waren einlagern»

Anwendungsfallbeschreibung			
Name	Waren einlagern		
Kurzbeschreibung	Ein Kunde / Aussendienstmitarbeiter lagert Produkte im E-Konsignationslager ein. Die eingelagerten Produkte werden in der Lagerverwaltung hinzugefügt.		
Akteure	Kunde / Aussendienstmitarbeiter E-Konsignationslager		
Auslöser	Der Kunde / Aussendienstmitarbeiter will neue Produkte einlagern.		
Ergebnisse	Die neuen Produkte sind im E-Konsignationslager eingelagert. Der Lagerbestand ist nachgeführt. Produktinformationen werden angezeigt. Der Akteur erhält eine Rückmeldung ob <ul style="list-style-type: none"> - ein Chargenrückruf ausstehend ist - das Verfallsdatum erreicht ist 		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	Barcode der Ware, die eingelagert wird		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. In den Betriebsmodus «Produkteinlagerung» wechseln 2. Barcode auf Produkt scannen 3. Waren einlagern 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Übernommen aus Wirtschaftsprojekt (Epp & Föhn, 2018)
	18.02.19	OF	Anpassung Produkteinlagerung
Sonstiges	Der Akteur hat die Möglichkeit das Einlagern innerhalb einer kurzen Zeit rückgängig zu machen.		

Tabelle 14: Anwendungsfall «Waren einlagern»

Use-Case «Lagerbestand anzeigen»

Anwendungsfallbeschreibung			
Name	Lagerbestand anzeigen		
Kurzbeschreibung	Ein Kunde / Aussendienstmitarbeiter will sich den Lagerbestand anzeigen lassen, um ihn mit dem Inhalt des Konsignationslager abzugleichen.		
Akteure	Kunde / Aussendienstmitarbeiter E-Konsignationslager		
Auslöser	Eine Kontrolle des Konsignationslagers soll durchgeführt werden.		
Ergebnisse	Der Kunde / Aussendienstmitarbeiter sieht das Inventar des E-Konsignationslagers.		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	--		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Lagerbestand anzeigen, es werden folgende Eigenschaften angezeigt: Bestände der Artikel (Anzahl der gleichen Produkte), 2. Für einen Artikel können zusätzlich alle Produkte einzeln mit den jeweiligen Eigenschaften angezeigt werden. 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	18.02.19	OF	Aus bestehendem Use-Case «Bestand kontrollieren» extrahiert (Epp & Föhn, 2018)
Sonstiges	--		

Tabelle 15: Anwendungsfall «Lagerbestand anzeigen»

Use-Case «Lagerbestand korrigieren»

Anwendungsfallbeschreibung			
Name	Lagerbestand korrigieren		
Kurzbeschreibung	Der Lagerbestand stimmt nicht mit dem effektiven Bestand überein und die Bestände sollen korrigiert werden.		
Akteure	Aussendienstmitarbeiter E-Konsignationslager		
Auslöser	Eine Differenz zwischen dem effektiven Bestand und dem Bestand des Lagerinterfaces wurde festgestellt.		
Ergebnisse	Der Lagerbestand des E-Konsignationslagers ist korrekt.		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	Produkt(e), welche nicht mehr im Lager vorhanden sind. Produkt(e), welche zusätzlich im Lager vorhanden sind.		
Essenzielle Schritte	Produkt(e), die nicht mehr im Lager vorhanden sind: <ol style="list-style-type: none"> 1. Mit Kunde abklären, was mit diesen Produkten passiert ist 2. Entweder Use-Case «Waren kaufen» oder Use-Case «Waren zurücknehmen» Produkt(e), welche nicht im E-Konsignationslager erfasst sind: <ol style="list-style-type: none"> 1. Use-Case «Waren einlagern» 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	18.02.19	OF	Aus bestehendem Use-Case «Bestand kontrollieren» extrahiert (Epp & Föhn, 2018)
Sonstiges	--		

Tabelle 16: Anwendungsfall «Lagerbestand korrigieren»

Use-Case «Chargenrückruf durchführen»

Anwendungsfallbeschreibung			
Name	Chargenrückruf durchführen		
Kurzbeschreibung	Ein Aussendienstmitarbeiter bereitet den Rückruf einer Charge vor.		
Akteure	Aussendienstmitarbeiter Lagerverwaltung		
Auslöser	Ein Aussendienstmitarbeiter erhält die Information, dass eine Charge zurückgerufen werden muss.		
Ergebnisse	Chargenrückruf erfolgreich durchgeführt		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	Chargennummer		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Der Mitarbeiter wählt die mangelhafte Charge aus 2. Der Mitarbeiter sperrt die Charge 3. Lager, welche Produkte der betroffenen Charge lagern, werden angezeigt 4. E-Konsignationslager, welche rückgerufene Produkte im Lagerbestand haben, werden informiert 5. Aussendienstmitarbeiter wechselt beim betroffenen Kunden in den Betriebsmodus «Produktrücknahme» 6. Aussendienstmitarbeiter scannt betroffene Produkte 7. Die betroffene Charge ist nicht mehr in der Lagerverwaltung vorhanden 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Erste Version erstellt
	18.02.19	OF	Ergänzung der verschiedenen Varianten
Sonstiges	--		

Tabelle 17: Anwendungsfall «Chargenrückruf durchführen»

Use-Case «Lagerinterface konfigurieren»

Anwendungsfallbeschreibung			
Name	Lagerinterface konfigurieren		
Kurzbeschreibung	Der technische Mitarbeiter richtet ein E-Konsignationslager für einen Kunden ein.		
Akteure	Technischer Mitarbeiter Lagerinterface, Lagerverwaltung		
Auslöser	Ein Kunde wird mit dem neuen E-Konsignationslager ausgerüstet.		
Ergebnisse	Die Lagerinterface ist kundenneutral eingerichtet. Das Lagerinterface ist bereit, um vom Aussendienstmitarbeiter für einen Kunden installiert zu werden.		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	Hard- und Software zur Bereitstellung des Lagerinterfaces		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Hard- und Software einrichten 2. Lagerinterface konfigurieren 3. Lagerinterface beim Provider registrieren 4. Verbindung von Lagerinterface zur Lagerverwaltung testen 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Übernommen aus Wirtschaftsprojekt (Epp & Föhn, 2018)
	15.02.19	IE	Diverse Anpassungen
Sonstiges	--		

Tabelle 18: Anwendungsfall «Lagerinterface konfigurieren»

Use-Case «E-Konsignationslager in Betrieb nehmen»

Anwendungsfallbeschreibung			
Name	E-Konsignationslager in Betrieb nehmen		
Kurzbeschreibung	Der Aussendienstmitarbeiter bereitet das Lagerinterface vor und installiert es im Konsignationslager bei einem Kunden.		
Akteure	Aussendienstmitarbeiter Lagerinterface, Konsignationslager, Lagerverwaltung		
Auslöser	Ein Kunde wird mit dem neuen E-Konsignationslager ausgerüstet.		
Ergebnisse	Das neue E-Konsignationslager ist beim Kunden installiert. Die Lagerbestände wurden im Lagerinterface sowie in der Lagerverwaltung erfasst.		
Vorbedingungen	Lagerinterface konfiguriert und getestet (siehe Use-Case «Lagerinterface konfigurieren»).		
Nachbedingungen	--		
Eingehende Daten	Artikelstamm Informationen zum Lager <ul style="list-style-type: none"> • Kunde • Stockwerkbezeichnung • Standortbezeichnung • Mindestbestand pro Artikel • Nachbestellmenge pro Artikel 		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Lagerinterface in der Lagerverwaltung mit Kunde verknüpfen 2. Artikelstamm auf das Lagerinterface laden 3. Beim Kunden das Lagerinterface im Konsignationslager des Kunden installieren 4. Produkte erfassen (gemäss Use-Case «Waren einlagern») 5. Mindestbestände und Nachbestellmenge in der Lagerverwaltung festlegen 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Übernommen aus Wirtschaftsprojekt (Epp & Föhn, 2018)
	18.02.19	OF	Ergänzt mit Artikelstamm
Sonstiges	--		

Tabelle 19: Anwendungsfall «E-Konsignationslager in Betrieb nehmen»

Use-Case «E-Konsignationslager fernüberwachen»

Anwendungsfallbeschreibung			
Name	E-Konsignationslager fernüberwachen		
Kurzbeschreibung	Ein Aussendienstmitarbeiter überwacht das E-Konsignationslager in der Lagerverwaltung.		
Akteure	Aussendienstmitarbeiter Lagerverwaltung		
Auslöser	Der Aussendienstmitarbeiter will den Zustand eines E-Konsignationslagers von seinem Arbeitsplatz aus überprüfen.		
Ergebnisse	Der Aussendienstmitarbeiter hat einen Überblick über <ul style="list-style-type: none"> • die Lagerbestände im E-Konsignationslager • offene Chargenrückrufe • den Verbindungszustand des Lagerinterfaces um weitere Schritte einzuleiten.		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	--		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Kunde auswählen 2. Lagerbestände anzeigen (inkl. eventueller Unterschreitung der Mindestbestände und Chargenrückrufe) 3. Verbindungszustände/Verläufe anzeigen 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Erste Version erstellt
	18.02.19	OF	Ergänzung von Detailinformationen
Sonstiges	--		

Tabelle 20: Anwendungsfall «E-Konsignationslager fernüberwachen»

Use-Case «Lagerstörung erkennen & analysieren»

Anwendungsfallbeschreibung			
Name	Lagerstörung erkennen & analysieren		
Kurzbeschreibung	Ein technischer Mitarbeiter erhält eine Störungsmeldung eines E-Konsignationslager und will diese Störung beheben.		
Akteure	Technischer Mitarbeiter, Aussendienstmitarbeiter Lagerverwaltung, E-Konsignationslager		
Auslöser	Ein Aussendienstmitarbeiter / technischer Mitarbeiter sieht in der Lagerverwaltung ein Verbindungsproblem bei einem E-Konsignationslager.		
Ergebnisse	Der technische Mitarbeiter hat einen Überblick über die Störung eines E-Konsignationslager eines Kunden. Der Aussendienstmitarbeiter / technischer Mitarbeiter hat die Voraussetzungen, um die Störung zu beheben.		
Vorbedingungen	--		
Nachbedingungen	--		
Eingehende Daten	--		
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Überblick über den Zustand des E-Konsignationslager des Kunden erhalten 2. Informationen anzeigen, welche für die Störung nützlich sind 3. Möglichkeit im E-Konsignationslager beim Kunden: <ul style="list-style-type: none"> • Anzeigen des Systemstatus • Neustart des Lagerinterfaces 		
Offene Punkte	--		
Änderungshistorie	Wann	Wer	Was
	15.02.19	IE	Erste Version erstellt
	18.02.19	OF	Ergänzung Analysemethoden
Sonstiges	--		

Tabelle 21: Anwendungsfall «Lagerstörung erkennen & analysieren»

3.2.3. Systemanforderungen

Allgemein

AF#001 – Verteiltes System

Die E-Konsignationslager und die Lagerverwaltung müssen als verteiltes System realisiert werden und über einen sicheren Kommunikationskanal miteinander kommunizieren können.

AF#002 – Skalierbarkeit

Die Lagerverwaltung muss mehrere Kunden gleichzeitig verwalten können. Ein Kunde kann mehrere E-Konsignationslager besitzen. Ein Lagerinterface muss nicht zwingend einem Kunden zugewiesen sein.

AF#003 – Datensicherheit

Ein unerwarteter Hardwareausfall oder ein Neustart eines Teilsystems dürfen nicht zu Datenverlust führen. Jedes Teilsystem muss seine Daten regelmässig vor einem Betriebsausfall sichern.

AF#004 – Datenintegrität

Es muss sichergestellt sein, dass die Daten des E-Konsignationslagers mit den Daten der Lagerverwaltung automatisch abgeglichen werden. Dateninkonsistenzen müssen vom System erkannt werden und anhand der Daten des E-Konsignationslagers aufgelöst werden.

AF#005 – Benutzerverwaltung

Die folgenden Rollen müssen im System unterschieden werden:

Kunde

Der Kunde hat beschränkte Rechte und darf nicht alle Funktionen ausführen können. Er ist der Standardbenutzer und braucht keine Authentifizierung.

Aussendienstmitarbeiter

Der Aussendienstmitarbeiter darf alle Funktionen ausführen, benötigt dazu jedoch eine Authentifizierung.

AF#006 – Nachverfolgbarkeit

Veränderungen des Lagerbestands (Mutationen) müssen rückverfolgbar aufgezeichnet werden. Die folgenden Informationen müssen gespeichert werden.

- Zeitstempel der Mutation
- Grund der Mutation
 - Kauf
 - Chargenrückruf
 - Verfallsdatum überschritten
 - Verfallsdatum bald erreicht
 - Kulanz
 - Sonstiges
- Angemeldete Rolle

AF#007 – Automatisiertes Startverhalten

Durch einen Neustart muss der Betriebszustand ohne zusätzliche Benutzerinteraktion wiederhergestellt werden.

AF#008 – Betriebsdauer

Der Betrieb des Systems muss bis Ende Januar 2020¹ sichergestellt sein.

¹ Auf Anfrage der B. Braun Medical AG

Kommunikation

AF#011 – Funktechnologie

Das E-Konsignationslager muss eine drahtlose Technologie verwenden, um mit der Lagerverwaltung zu kommunizieren und muss unabhängig von der bestehenden Netzwerkinfrastruktur des Kunden funktionieren.

AF#012 – Optimierte Datenübertragung

Die Kommunikation zwischen den E-Konsignationslagern und der Lagerverwaltung soll auf ein mögliches, betriebsnotwendiges Minimum reduziert werden.

AF#013 – Verbindungsqualität

Die Lagerverwaltung muss eine Übersicht über die Qualität der Verbindungen zu den E-Konsignationslagern aufzeigen. Detaillierte Qualitätsinformationen können vom Provider bezogen werden.

AF#014 – Umgang mit Kommunikationsunterbrüchen

Es darf davon ausgegangen werden, dass die Kommunikation zwischen den E-Konsignationslagern und der Lagerverwaltung grundsätzlich hergestellt werden kann. Bei kurzfristigen Unterbrüchen muss das System sicherstellen, dass keine Nachrichten verloren gehen.

Benutzerinteraktionen

AF#021 – Barcodescanner

Der verwendete Barcodescanner muss eine automatische Scanfunktion verfügen. Das Verwenden des Barcodescanners muss ohne Betätigung einer Taste funktionieren.

AF#022 – Produktidentifikation

Die Produkte im E-Konsignationslager müssen über den GS1 Barcode auf der Verpackung identifiziert werden können. Die folgenden Application Identifier müssen erkannt werden können (GS1 Switzerland, 2019):

- Global Trade Item Number (01, numerisch)
- Losnummer / Chargennummer (10, alphanumerisch)
- Verfallsdatum im Format «JJMMTT» (17, numerisch)

Produkte mit einem Barcode dürfen als Smallest Sellable Unit (SSU) betrachtet werden.

AF#023 – Anzeige

Das E-Konsignationslager muss über eine grafische Anzeige verfügen, um Informationen abbilden zu können.

AF#024 – Corporate Design

Die Benutzerschnittstellen müssen dem Corporate Design der B. Braun AG² entsprechen.

Artikelstamm

AF#031 – Artikelstamm führen

Zusätzliche Informationen über Artikel, welche nicht über den Barcode identifizierbar sind, müssen separat geführt und zentral verwaltet werden.

AF#032 – Artikelname führen

Im Artikelstamm muss der Name des Artikels geführt werden.

AF#033 – Artikelgrafik führen

Im Artikelstamm muss eine Grafik des Artikels geführt werden.

² <https://www.bbraun-brand.com/bbraun-brand-guide.html>

3.2.4. Funktionale Anforderungen

Allgemein

AF#101 – Lagerbestand anzeigen

Der komplette Lagerbestand muss angezeigt werden können. Folgende Informationen müssen pro Produkt angezeigt werden:

- Artikelname
- Global Trade Item Number
- Verfallsdatum
- Lagerbestand

AF#102 – Produktkauf ohne Scannen

Falls der Barcode auf der Produktverpackung nicht lesbar ist, muss die Möglichkeit bestehen, das Produkt ohne Barcodescanner zu beziehen.

AF#104 – Verfallsdatum überschritten beim Produktkauf

Beim Kauf eines Produkts muss der Benutzer darauf hingewiesen werden, falls das Verfallsdatum überschritten ist.

AF#105 – Artikelstamm abgleichen

Die E-Konsignationslager müssen mit dem originalen Artikelstamm abgeglichen werden können.

AF#106 – Lagerinterface vorkonfigurieren

Das Lagerinterface muss eine Verbindungskontrolle mit der Lagerverwaltung durchführen können, ohne dass das Lagerinterface einem Kunden zugewiesen ist.

In der Lagerverwaltung muss das Ergebnis dieser Verbindungskontrolle angezeigt werden.

AF#107 – Produktrücknahme ohne Scannen

Falls der Barcode auf der Produktverpackung nicht lesbar ist, muss die Möglichkeit bestehen, das Produkt ohne Barcodescanner zurückzunehmen.

Betriebsmodi

AF#111 – Betriebsmodi Lagerinterface

Das Lagerinterface muss mehrere Betriebsmodi unterstützen. Durch das Ändern des Betriebsmodus wird das Verhalten beim Scannen eines Barcodes verändert.

AF#112 – Betriebsmodus «Produktkauf»

Beim Scannen eines Produkts im Betriebsmodus «Produktkauf» muss der Lagerbestand reduziert werden. Ein Kauf im rechtlichen Sinn hat stattgefunden.

AF#113 – Betriebsmodus «Produkteinlagerung»

Beim Scannen eines Produkts im Betriebsmodus «Produkteinlagerung» muss der Lagerbestand erhöht werden.

AF#114 – Betriebsmodus «Produktrücknahme»

Beim Scannen eines Produkts im Betriebsmodus «Produktrücknahme» muss der Lagerbestand reduziert werden. Es hat kein Kauf stattgefunden. Der Grund für die Produktrücknahme muss festgelegt sein.

AF#115 – Betriebsmodus «Produktanzeige»

Beim Scannen eines Produkts im Betriebsmodus «Produktanzeige» muss eine Produktübersicht angezeigt werden. Folgende Informationen müssen angezeigt werden

- Global Trade Item Number
- Chargennummer
- Verfallsdatum
- Informationen zu Produkt im Artikelstamm

Chargen

AF#121 – Chargen anzeigen

Alle Chargen eines E-Konsignationslagers müssen angezeigt werden können. Folgende Informationen müssen angezeigt werden:

- Chargennummer
- Vorhandene Produkte im Lager

AF#122 – Chargen markieren

Eine Charge muss in der Lagerverwaltung markiert werden können. Betroffene Kunden und E-Konsignationslager dieser Charge müssen angezeigt werden können.

AF#123 – Gesperrte Chargen synchronisieren

Die Lagerverwaltung muss den E-Konsignationslagern Informationen über gesperrte Chargen anbieten.

AF#124 – Chargenrückruf ausstehend beim Produktkauf

Beim Kauf eines Produkts muss der Benutzer darauf hingewiesen werden, falls die entsprechende Charge gesperrt wurde.

Systemzustand

AF#131 – Regelmässige Verbindungskontrolle

Das Lagerinterface muss im Schnitt alle zwei Stunden ein Signal an die Lagerverwaltung schicken, damit diese eine Übersicht über den Verbindungsstatus aller Geräte bilden kann.

AF#132 – Manuelle Verbindungskontrolle

Die automatisch durchgeführte Verbindungskontrolle muss auch manuell gestartet werden können.

AF#133 – Vereinfachter Systemstatus anzeigen

Das Lagerinterface muss eine leicht verständliche Anzeige des aktuellen Systemzustands beinhalten. Die angezeigten Informationen müssen für eine erste Fehleranalyse ausreichen.

AF#134 – Erweiterter Systemstatus anzeigen

Das Lagerinterface muss eine aussagekräftige Anzeige des aktuellen Systemzustands beinhalten. Die angezeigten Informationen müssen für eine komplette Fehleranalyse ausreichen.

AF#135 – Automatischer Neustart

Kann das Lagerinterface über einen Zeitraum von 10 Stunden keine Verbindung zur Lagerverwaltung herstellen, so muss sich das Lagerinterface um 1:00 Uhr morgens automatisch neustarten. Der automatische Neustart muss fünf Minuten im Voraus angezeigt werden und durch den Benutzer abgebrochen werden können.

AF#136 – Fern Neustart

Durch die Lagerverwaltung muss ein E-Konsignationslager neugestartet werden können. Der Neustart muss zwei Minuten im Voraus angezeigt werden und durch den Benutzer abgebrochen werden können.

Überwachung

AF#141 – Benachrichtigungssystem

Die Lagerverwaltung muss ein E-Mail an einen verantwortlichen Mitarbeiter der B. Braun AG schicken können. Pro E-Konsignationslager wird eine Kontaktperson zugewiesen.

AF#142 – Mindestbestand erfassen

Pro Artikel im E-Konsignationslager muss ein Mindestbestand angegeben werden können. Unterschreitet der Lagerbestand eines Artikels eine definierte Mindestanzahl, so muss dies optisch ersichtlich sein.

AF#143 – Mindestbestandsbenachrichtigung

Unterschreitet der Lagerbestand eines Artikels eine definierte Mindestanzahl, so muss der verantwortliche Mitarbeiter benachrichtigt werden.

3.2.5. Zuweisung der Anforderungen

Die nachfolgende Tabelle zeigt den Zusammenhang der Anforderungen zu den Teilsystemen auf. Ist eine Anforderung an ein Teilsystem zugewiesen, bedeutet dies, dass dieses Teilsystem eine Funktion zur Erfüllung der Anforderung liefern muss. Beim Lagerinterface gibt es eine Unterteilung für die Rollen Kunde (K) und Aussendienstmitarbeiter (A). Ist ein Kreuz nur in einer der beiden Spalten gesetzt, so darf diese Funktion nur durch diejenige Rolle ausgeführt werden können. Der technische Mitarbeiter agiert als Aussendienstmitarbeiter auf dem Lagerinterface. Auf der Lagerverwaltung existiert keine Rollenunterteilung.

Anforderung	Lagerinterface		Lagerverwaltung
	K	A	
AF#001 – Verteiltes System	X		X
AF#002 – Skalierbarkeit			X
AF#003 – Datensicherheit	X		X
AF#004 – Datenintegrität	X		X
AF#005 – Benutzerverwaltung	X		
AF#006 – Nachverfolgbarkeit	X		X
AF#007 – Automatisiertes Startverhalten	X		X
AF#008 – Betriebsdauer	X		X
AF#011 – Funktechnologie	X		
AF#012 – Optimierte Datenübertragung	X		
AF#013 – Verbindungsqualität			X
AF#014 – Umgang mit Kommunikationsunterbrüchen	X		X
AF#021 – Barcodescanner	X		
AF#022 – Produktidentifikation	X		
AF#023 – Anzeige	X		

Anforderung	Lagerinterface		Lagerverwaltung
	K	A	
AF#024 – Corporate Design	X		X
AF#031 – Artikelstamm führen	X		X
AF#032 – Artikelname führen	X		X
AF#033 – Artikelgrafik führen	X		X
AF#101 – Lagerbestand anzeigen	X	X	X
AF#102 – Produktkauf ohne Scannen	X	X	
AF#104 – Verfallsdatum überschritten beim Produktkauf	X	X	
AF#105 – Artikelstamm abgleichen		X	X
AF#106 – Lagerinterface vorkonfigurieren		X	X
AF#107 – Produktrücknahme ohne Scannen		X	
AF#111 – Betriebsmodi Lagerinterface	X	X	
AF#112 – Betriebsmodus «Produktkauf»	X	X	
AF#113 – Betriebsmodus «Produkteinlagerung»	X	X	
AF#114 – Betriebsmodus «Produktrücknahme»		X	
AF#115 – Betriebsmodus «Produktanzeige»	X	X	
AF#121 – Chargen anzeigen	X	X	X
AF#122 – Chargen markieren			X
AF#123 – Gesperrte Chargen synchronisieren	X	X	X
AF#124 – Chargenrückruf ausstehend beim Produktkauf	X	X	
AF#131 – Regelmässige Verbindungskontrolle	X	X	X
AF#132 – Manuelle Verbindungskontrolle	X	X	
AF#133 – Vereinfachter Systemstatus anzeigen	X		
AF#134 – Erweiterter Systemstatus anzeigen		X	
AF#135 – Automatischer Neustart	X	X	
AF#136 – Fern Neustart	X	X	X
AF#141 – Benachrichtigungssystem			X
AF#142 – Mindestbestand erfassen			X
AF#143 – Mindestbestandsbenachrichtigung			X

Tabelle 22: Zuweisung der Anforderungen zu den Teilsystemen

3.3. Technologien

Die nachfolgende Abbildung zeigt die wichtigsten verwendeten Technologien, aufgeteilt nach Teilsystem. In den folgenden Abschnitten werden diese kurz erläutert.

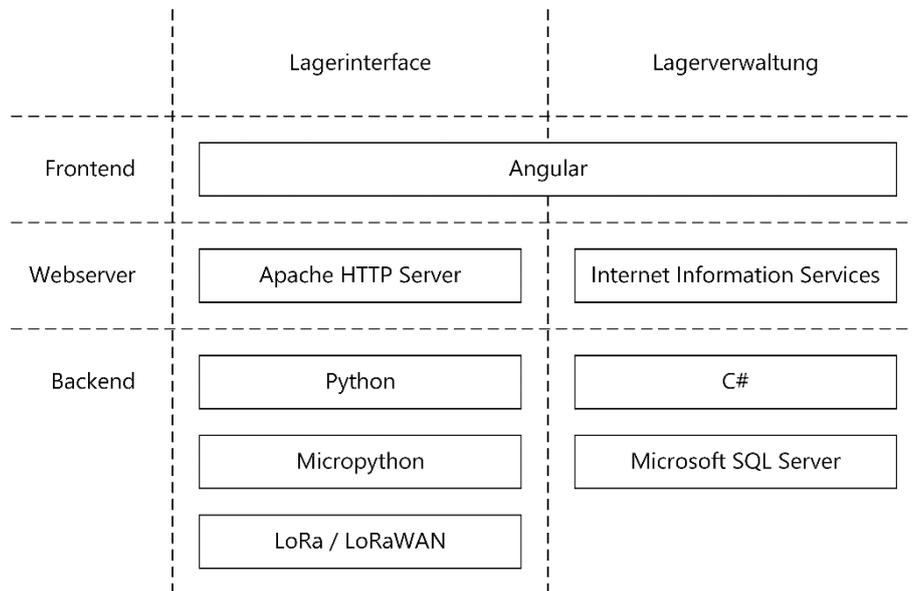


Abbildung 7: Technologie Stack des gesamten Systems

3.3.1. LoRa / LoRaWAN

LoRa, kurz für Long Range, ist eine Funktechnologie, welche ausgerichtet ist für IoT-Geräte. LoRa ist der physikalische Layer, wobei LoRaWAN, kurz für Long Range Wide Area Network, die Vermittlungs- und Sicherungsschicht übernimmt. Das LoRaWAN-Protokoll verbindet Endgeräte über sogenannte LoRaWAN-Gateways mit einem Applikationsserver. Um Daten zum Gateway zu senden, muss jedem Gerät eine eindeutige Geräte Adresse zugewiesen werden. Dazu gibt es zwei Verfahren, Over-The-Air-Activation (OTAA) und Activation By Personalization (ABP). Beim OTAA-Verfahren, wird infolge einer Join-Request die Geräteadresse durch das Gateway zugewiesen. Bei ABP ist die Geräteadresse fix auf dem Endgerät hinterlegt und der Join-Request kann weggelassen werden (LoRa Alliance™ [I], 2019).

Das LoRaWAN-Protokoll ist aufgeteilt in die drei Klassen, A, B und C, wobei diese vom Endgerät festgelegt werden. Klasse A erlaubt die bidirektionale Kommunikation zwischen Endgerät und Server, wobei die Kommunikation vom Endgerät gestartet werden muss. Geräte der Klasse B bieten zusätzlich Datenempfangsfenster an, so dass der Server zu regelmässigen Zeiten Datenpakete an das Endgerät übermitteln kann. Geräte der Klassen A und B sind auf energieeffizienz ausgelegt, meistens batteriebetrieben, da diese nicht ständig die Funkantenne verwenden. Klasse C Geräte unterstützen eine vollwertige bidirektionale Kommunikation (LoRa Alliance™ [I], 2019). Um alle Anforderungen (insbesondere AF#123 – Gesperrte Chargen synchronisieren und AF#136 – Fern Neustart) erfüllen zu können, muss ein Gerät der Klasse C verwendet werden.

Die maximal garantierte Datenlänge für das europäische Frequenzband 868 beträgt 51 Bytes (LoRa Alliance™ [II], 2019). Dies muss bei der LoRa-Kommunikation berücksichtigt werden.

Die folgende Abbildung zeigt die geschichtete Aufteilung der LoRaWAN-Spezifikation mit den drei Klassen, sowie der Aufteilung der Frequenzbänder.

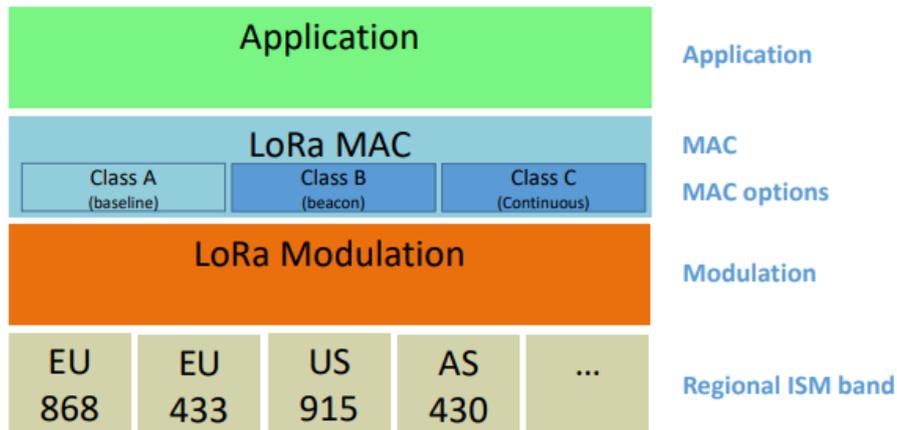


Abbildung 8: LoRa Übersicht (LoRa Alliance™ [I], 2019)

3.3.2. Python

Python³ ist eine objektorientierte Programmiersprache. In diesem Projekt wird Python, in der Version 3.7, für die Umsetzung des Lagerinterface Backend verwendet. Entwickelt wird mit der Entwicklungsumgebung PyCharm⁴ von JetBrains.

3.3.3. Micropython

Micropython⁵ ist eine Implementation von Python mit reduziertem Funktionsumfang, welche für Microcontroller optimiert ist. Für die Entwicklung wird Atom⁶ mit dem Pymakr Plugin⁷ verwendet (Pycom [I], 2019).

3.3.4. C#

C-Sharp⁸ ist eine objektorientierte Programmiersprache entwickelt von Microsoft. In diesem Projekt wird das Sprachlevel 7 verwendet, um das Backend der Lagerverwaltung umzusetzen. Für die Entwicklung wird die Entwicklungsumgebung Visual Studio 2017⁹ verwendet.

3.3.5. Angular

Angular 7¹⁰ ist eine Typescript-basierte Programmiersprache, um Web-Applikationen zu erstellen. Mittels dieser Technologie wird das Frontend des Lagerinterfaces, wie auch der Lagerverwaltung umgesetzt. Für die Entwicklung wird die Entwicklungsumgebung Visual Studio Code¹¹ verwendet.

³ <https://www.python.org/>

⁴ <https://www.jetbrains.com/pycharm/>

⁵ <https://micropython.org/>

⁶ <https://atom.io/>

⁷ <https://pycom.io/solutions/software/pymakr/>

⁸ <https://docs.microsoft.com/en-us/dotnet/csharp/>

⁹ <https://visualstudio.microsoft.com>

¹⁰ <https://angular.io/>

¹¹ <https://code.visualstudio.com/>

3.3.6. Apache HTTP Server

Für die Bereitstellung des Frontend auf dem Lagerinterface wird der Apache HTTP Server verwendet. Dieser stellt den Webserver, sowie einen Proxy bereit, mittels welchen das Frontend des Lagerinterfaces mit dem Backend kommunizieren kann.

3.3.7. Internet Information Services (IIS)

Im Bereich der Lagerverwaltung wird für das Hosting des Backend sowie des Frontend der Dienst Internet Information Services (kurz IIS)¹² in der Version 10 von Microsoft verwendet. Dieser stellt zudem sicher, dass die Applikationen bei einem Neustart des Servers automatisch wieder in Betrieb genommen werden.

3.3.8. Microsoft SQL Server

Die Lagerverwaltung soll Daten mittels des Microsoft SQL Server¹³ persistieren. Damit können Daten in strukturierter Form abgelegt und normalisiert werden. Die Version 2017 (14.0.2002) wird verwendet.

3.4. Lösungskonzept

- Das folgende Lösungskonzept ist eine überarbeitete Variante der im Wirtschaftsprojekt vorgeschlagenen Version. Dort wurden zwei Lösungsvarianten für das Lagerinterface beschrieben, wobei in diesem Projekt nur noch die Variante mit dem Barcodescanner weiterverfolgt wird.

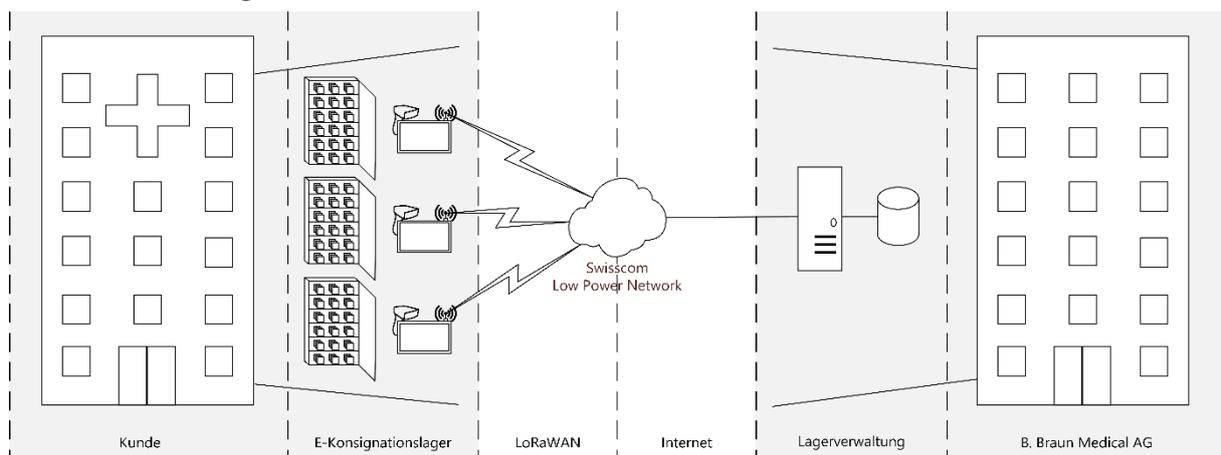


Abbildung 9: Systemübersicht

Das Gesamtsystem besteht aus zwei unterschiedlichen Teilsystemen, namentlich den E-Konsignationslagern (links) und der Lagerverwaltung (rechts). Die Kommunikation zwischen E-Konsignationslager und Lagerverwaltung wird über das Swisscom Low Power Network (LPN) abgewickelt.

Die gesamte Software ist aufgeteilt in eine Clientapplikation, ein LoPy-Modul und eine Serverapplikation (Abbildung 8). Die Clientapplikation wird auf den Lagerinterfaces in den E-Konsignationslagern verteilt. Das Lagerinterface ist zuständig für die Interaktionen mit dem Benutzer des E-Konsignationslagers und unterhält den lokalen Lagerbestand. Bei einer Mutation wird diese zuerst im Lagerinterface verarbeitet und via LoPy-Modul an die Serverapplikation weitergeleitet. Der Server unterhält eine Kopie aller Lagerbestände der E-Konsignationslager, um diese zentral auswerten zu können.

¹² <https://www.iis.net/>

¹³ <https://www.microsoft.com/en-us/sql-server/sql-server-2017>

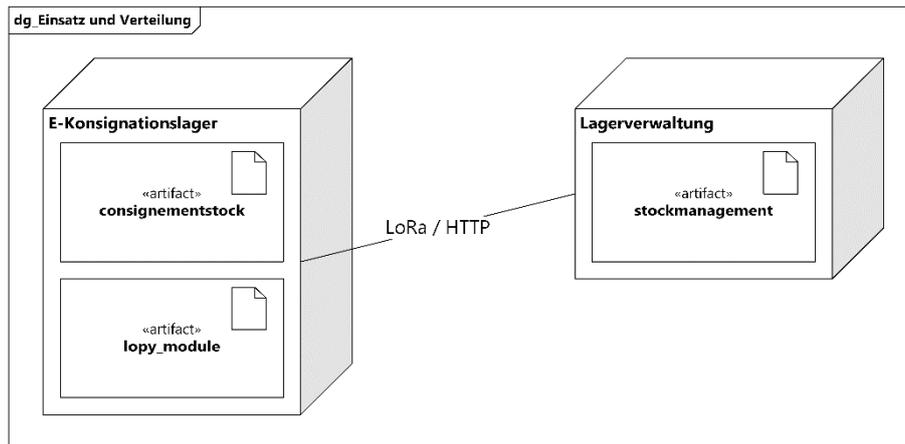


Abbildung 10: Softwareeinsatz und -verteilung

3.4.1. E-Konsignationslager

Die E-Konsignationslager werden beim Kunden betrieben. Dieser Begriff wird definiert durch ein Konsignationslager und dem sogenannten Lagerinterface. Das Lagerinterface besteht aus einem Raspberry Pi mit Touchscreen, einem LoRa-Modul, sowie einem Barcodescanner. Zusammen mit der darauf installierten Software kann das E-Konsignationslager vor Ort verwaltet werden. Auf dem Touchscreen werden Informationen zum Lagerbestand und Systemstatus angezeigt. Zusätzlich kann darauf der Betriebsmodus gewechselt werden, um die Reaktion beim Scannen eines Produkts zu verändern. Der Benutzer verwendet den Barcodescanner, um einfach und schnell Produkte ein- beziehungsweise auszulagern. Aussendienstmitarbeiter können sich im E-Konsignationslager anmelden, um auf zusätzliche Funktionen zuzugreifen.

Die nachfolgende Abbildung zeigt den Systemaufbau des Lagerinterfaces.

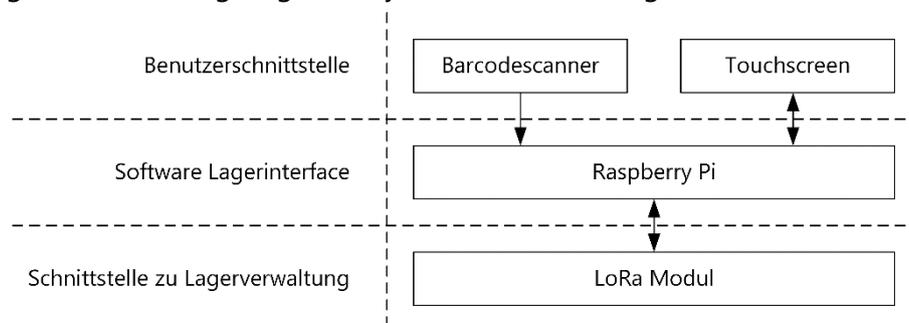


Abbildung 11: Übersicht Systemaufbau Lagerinterface

Eingesetzte Hardware

Für das Lagerinterface wird folgende Hardware benötigt:

- Raspberry Pi 3 Model B+¹⁴
- Raspberry Pi Touch Display¹⁵
- Datalogic Magellan 1100i Scanner Kit (Barcodescanner)¹⁶
- LoPy 4¹⁷

¹⁴ <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

¹⁵ <https://www.raspberrypi.org/products/raspberry-pi-touch-display/>

¹⁶ <https://www.datalogic.com/deu/einzelhandel-transport-logistik/kassenscanner/magellan-1100i-pd-157.html>

¹⁷ <https://pycom.io/product/lopy4-multipack/>

Barcodescanner

Für das Scannen von Barcodes wird ein Barcodescanner mit Näherungssensor verwendet. Dieser aktiviert sich automatisch, wenn sich ein Gegenstand genug nahe am Scanner befindet und führt dann ebenfalls automatisch einen Scan durch. Der gelesene Barcode wird via USB übertragen und auf dem Raspberry ausgewertet.

LoRa Modul

- Das LoRa Modul aus dem Wirtschaftsprojekt wurde verworfen und durch das LoPy 4 ersetzt. Der «LoRa Hat» von Dragino¹⁸ hatte Probleme beim Verbinden mit dem LoRa-Netzwerk und konnte keine stabile Klasse C Verbindung garantieren.

Das LoPy (Abbildung 12) von Pycom ist ein externer Microcontroller mit LoRa Antenne. Er kann via USB mit dem Raspberry Pi verbunden werden. Auf das LoPy kann ein eigenes Micropython Skript geladen werden, welches die Kommunikation zwischen dem Raspberry und dem LoRa-Netzwerk regelt (Pycom [II], 2019). Für die serielle Kommunikation steht eine UART Schnittstelle zur Verfügung (Pycom [III], 2019).



Abbildung 12: Verwendetes LoRa Modul (LoPy)

3.4.2. Lagerverwaltung

Die Lagerverwaltung ist die Verwaltungsplattform für die E-Konsignationslager. Sie bietet eine Übersicht über alle E-Konsignationslager und deren Kunden. Die Lagerverwaltung wird hauptsächlich für die Aussendienstmitarbeiter entwickelt, so dass diese die Lagerbestände ihrer Kunden und den Systemzustand des E-Konsignationslager aus der Ferne überwachen können.

¹⁸ https://wiki.dragino.com/index.php?title=Lora/GPS_HAT

Die Bereitstellung der Lagerverwaltung wird mittels eines virtuellen Windows Server 2016 sichergestellt. Dieser Server wird in der Azure Cloud mit der Unterstützung der Firma BitHawk AG bereitgestellt. Zusätzlich zum Server wird ein Routing erstellt, um einen Zugang aus dem Internet zu erhalten. Dies ist nötig, um vom Swisscom LPN Portal Meldungen zu empfangen (Port 8080) und die Erreichbarkeit des Frontend (Port 80) zu erstellen. Für die Bereitstellung sowie Kostenabrechnung des Servers ist der Auftraggeber zuständig.

Die komplette Serverapplikation wird über die Microsoft Internet Information Services (IIS) gehostet. Damit integriert sich die Software nahtlos in die Microsoft Umgebung. Als Datenpersistenzschicht wird ein Microsoft SQL Server verwendet.

Die nachfolgende Abbildung liefert eine Übersicht über die Abhängigkeiten zwischen den Netzwerkschnittstellen der Lagerverwaltung.

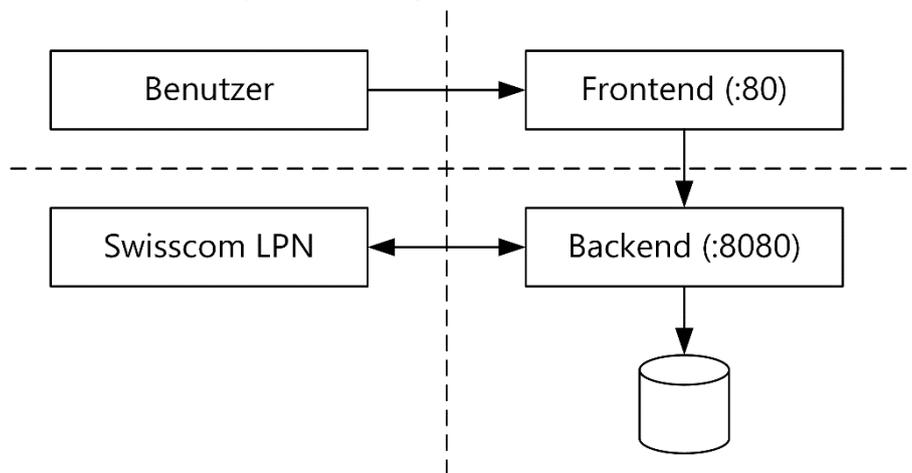


Abbildung 13: Übersicht Hard- und Software Lagerverwaltung

Serverumgebung

Die Server-Umgebung wurde im Detail wie folgt aufgebaut.

- Die Firewall hat lediglich den Port 80/8080 geöffnet. Zusätzlich für Wartungsaufgaben den Port 3389 für RDP¹⁹.
- Auf dem Applikationsserver wird das Frontend sowie das Backend bereitgestellt. Zusätzlich wird ein Proxy auf dem Frontend bereitgestellt, um Anfragen an das Backend weiterzuleiten.

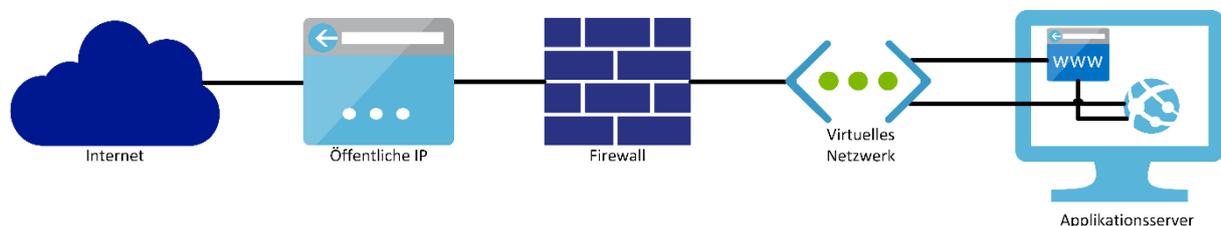


Abbildung 14: Lösungskonzept Hosting der Lagerverwaltung²⁰

¹⁹ Remotedesktop-Verbindung (sollte im produktiven Einsatz vermieden werden, eine bessere Lösung wäre VPN)

²⁰ Quelle Icons: <https://www.microsoft.com/en-us/download/details.aspx?id=41937>

3.4.3. Kommunikation

Die Kommunikation zwischen dem E-Konsignationslager und der Lagerverwaltung wird durch das Swisscom LPN gewährleistet. Die Swisscom deckt mit ihrem Low Power Network 96.8% der Schweizer Bevölkerung ab (Swisscom [I], 2019). Über das LPN können einerseits LoRa Geräte (E-Konsignationslager) und andererseits ein Applikationsserver (Lagerverwaltung) registriert werden. Datenpakete, welche über das LoRaWAN Protokoll verschickt werden (Uplink), werden über einen Swisscom Server in eine HTTP²¹-Anfrage umgewandelt und an den definierten Server weitergeleitet. Der umgekehrte Weg, Daten vom Applikationsserver zu einem einzelnen LoRa Gerät (Downlink), ist ebenfalls möglich (Swisscom [II], 2019).

Die folgende Abbildung beschreibt die Kommunikation zwischen E-Konsignationslager und Lagerverwaltung. Der Uplink (grün) und der Downlink (blau) sind farblich gekennzeichnet.

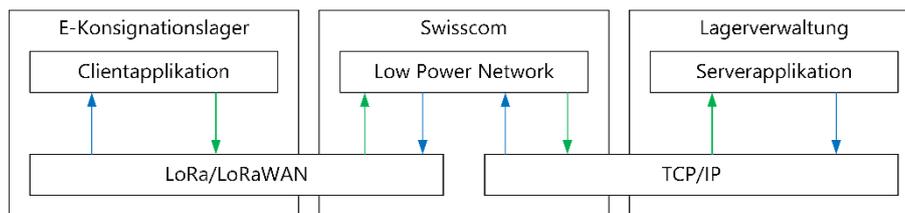


Abbildung 15: Aufbau des Protokolls

Protokoll

Die Nachrichten, welche unter den Systemen ausgetauscht werden, basieren auf einem eigens erstellten proprietären Protokoll. Die nachfolgende Abbildung zeigt den Grundaufbau dieser Nachrichten. Jeder Nachrichtentyp hat einen eindeutigen Code. Die Daten der Nachricht werden ASCII-codiert übermittelt und können für jeden Nachrichtentyp unterschiedlich lang sein.

Message type	Message data
1 Byte	n Bytes

Abbildung 16: Aufbau des system-eigenen Protokolls

In der folgenden Tabelle werden die verschiedenen Nachrichten aufgelistet und beschrieben.

Nachricht	Kurzbeschreibung
Bestandsmutation	Veränderung des Lagerbestands im E-Konsignationslager, welche der Lagerverwaltung mitgeteilt wird.
Heartbeat	Verbindungskontrolle des E-Konsignationslager mit der Lagerverwaltung.
Mutationsanfrage	Aufforderung der Lagerverwaltung, dass eine oder mehrere Bestandsmutationen erneut gesendet werden müssen.
Neustart	Aufforderung der Lagerverwaltung, dass sich das E-Konsignationslager neustarten soll.
Chargensperrung	Nachricht von der Lagerverwaltung an ein E-Konsignationslager, dass eine Charge gesperrt wurde.

Tabelle 23: Nachrichten

²¹ Hypertext Transfer Protocol

3.5. Design

3.5.1. Vorgaben

Als Design-Grundlage wird der B. Braun Brand-Guide verwendet. Dieser definiert die Richtlinien des Auftraggebers. Folgende Design-Elemente aus der Definition «B. Braun Process Apps»²² wurden verwendet.

Farben

In der folgenden Abbildung werden Farben für den spezifischen Bereich festgelegt. Diese sollen in den jeweiligen Bereichen der Benutzeroberfläche verwendet werden.

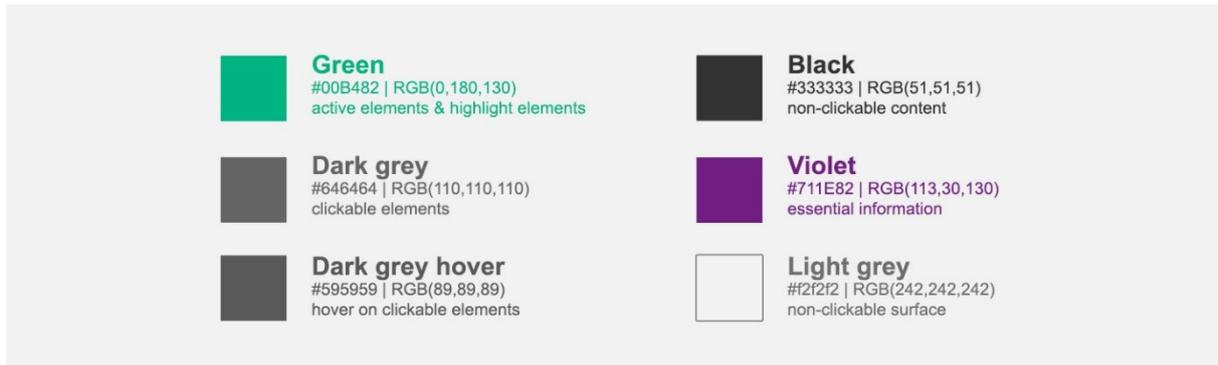


Abbildung 17: Farben im Design des B. Braun Brand Guides (B. Braun AG, 2019)

Layout

Das Basislayout einer Applikation soll anhand der folgenden Abbildung umgesetzt werden. Dabei werden die Farben der Kopfzeile sowie der Fusszeile mit speziellen Farben gekennzeichnet.

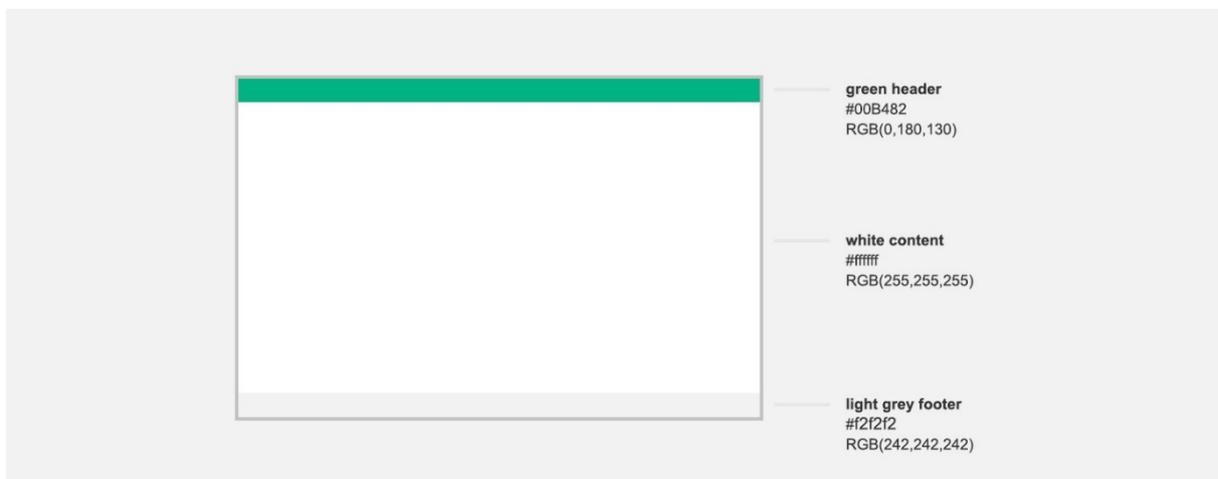


Abbildung 18: Layout des B. Braun Brand Guides (B. Braun AG, 2019)

Zusätzlich im Layout enthalten sind die Definitionen von Infoboxen (Abbildung 19). In diesen können seitenübergreifende Informationen abgelegt werden. Des Weiteren wird die Form des Inhalts mittels einer Tabelle oder Übersichten definiert.

²² <https://www.bbraun-brand.com/bbraun-brand-guide/en/design/digital/apps1.html>

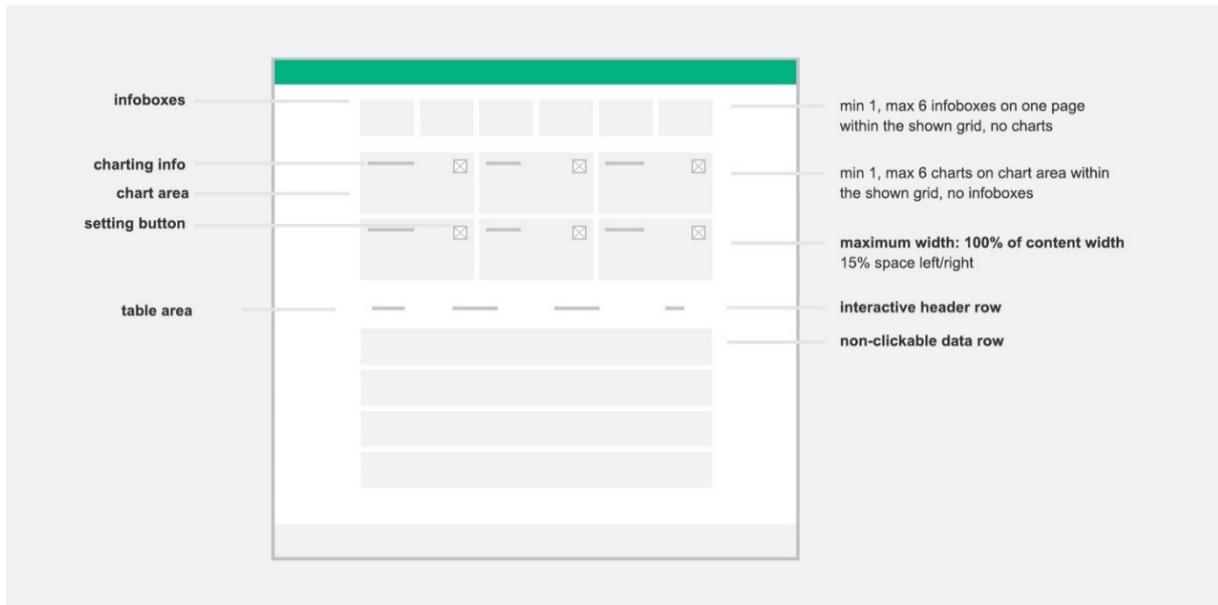


Abbildung 19: Layout einer Seite mit Infoboxen und Inhalt (B. Braun AG, 2019)

Komponenten

Folgende Komponenten werden durch den B. Braun Brand Guide explizit definiert. Falls zusätzliche Bestandteile nötig sind, werden diese an dem Design angepasst.

Kopfzeile

Die Kopfzeile soll in der Farbe «Green» Gestaltet werden. Auf der linken Seite soll ein «zurück»-Link sowie der Seitentitel stehen. Auf der rechten Seite können bei Bedarf mittels eines «...»-Buttons erweiterte Aktionen eingefügt werden.

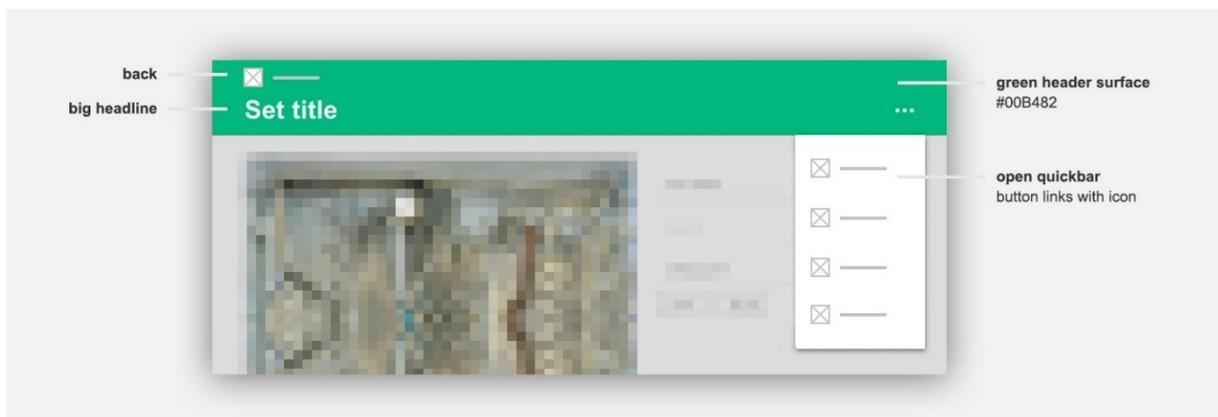


Abbildung 20: Kopfzeilendefinition des B. Braun Brand Guides (B. Braun AG, 2019)

Fusszeile

Die Fusszeile soll mittels der Farbe «Light Grey» gekennzeichnet werden. Inhalte sollen mittig angeordnet werden und Links sollen Bild und Text enthalten.

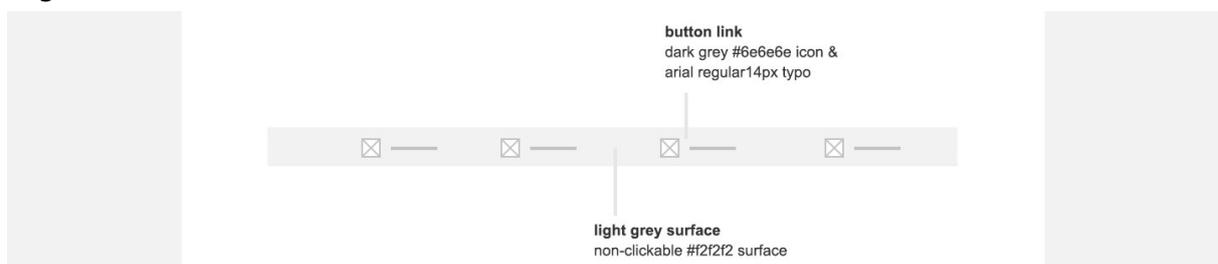


Abbildung 21: Definition Fusszeile des B. Braun Brand Guides (B. Braun AG, 2019)

Tabbar

Mittels einer Tabbar soll einfach zwischen verschiedenen Ansichten einer Seite navigiert werden können. Das aktive Element soll in der Farbe «Green» geschrieben/unterstrichen werden.

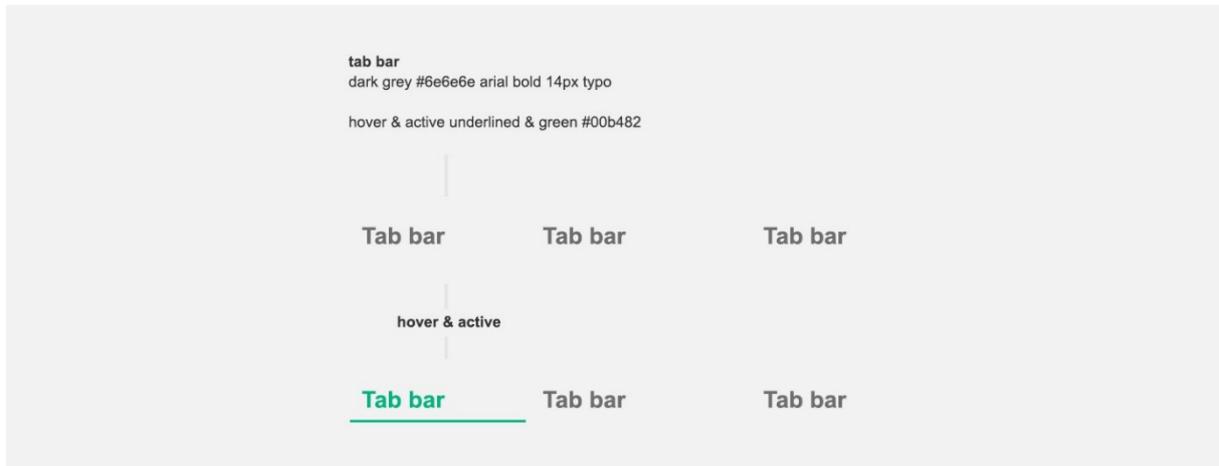


Abbildung 22: Definition Tabbar des B. Braun Brand Guides (B. Braun AG, 2019)

Button

Innerhalb der Applikationen sollen für Buttons die Vorlage «call to action» verwendet werden. Wobei die Schriftfarbe «Weiss» und die Hintergrundfarbe «Violet» ist.

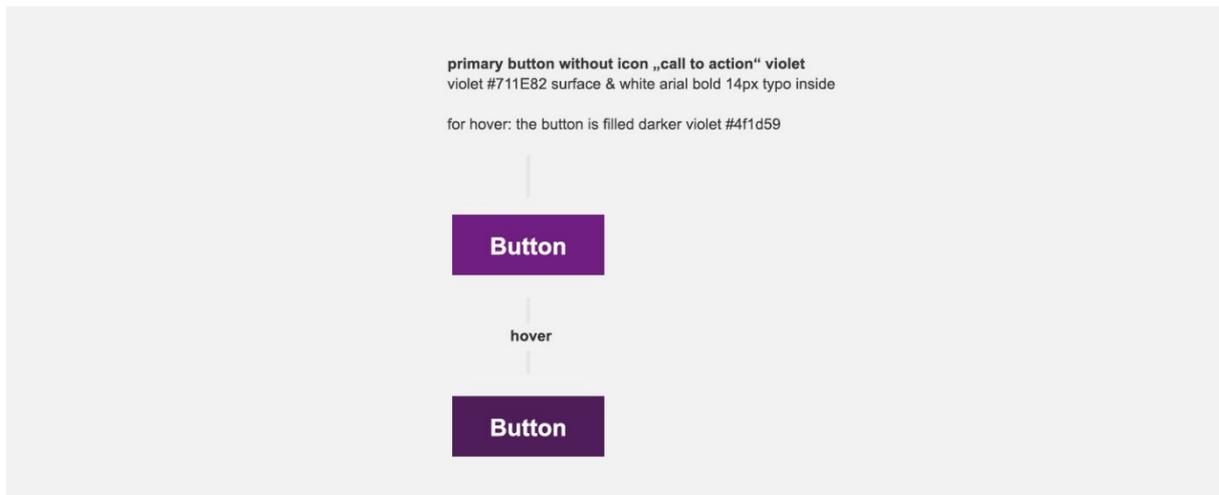


Abbildung 23: Definition Button des B. Braun Brand Guides (B. Braun AG, 2019)

3.5.2. Eigene Designelemente

Basis-Layout – Lagerverwaltung

Für die Lagerverwaltung wurde ein grundlegender Seitenaufbau festgelegt. Derselbe Seitenaufbau wurde zusätzlich als Basis für das Lagerinterface verwendet. Folgende Design-Elemente sind im Seitenaufbau enthalten.

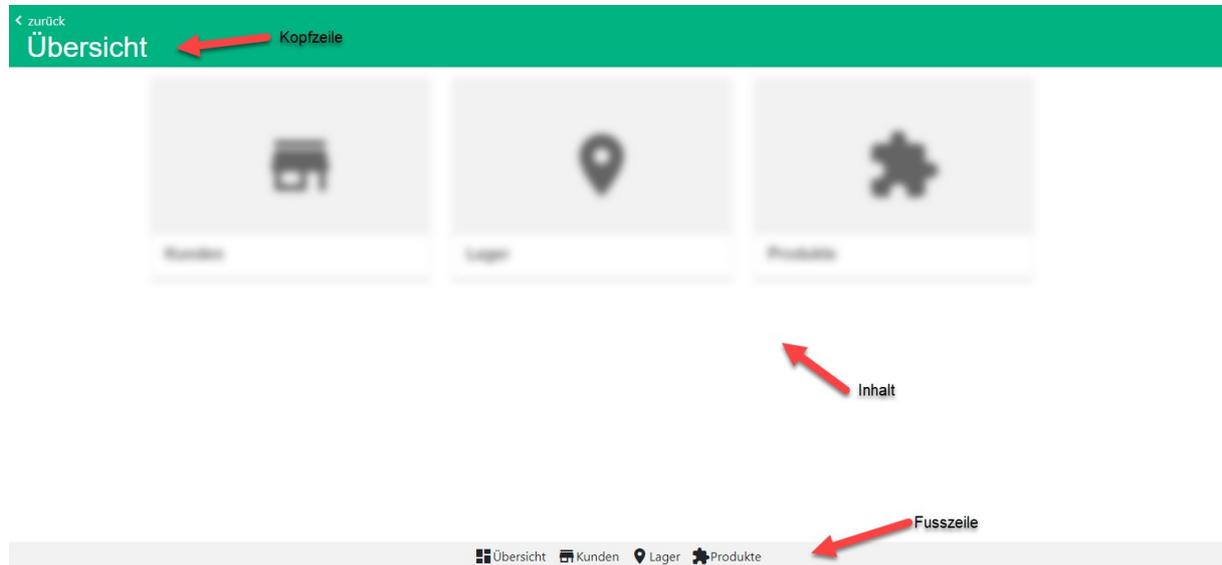


Abbildung 24: Grundlegender Seitenaufbau der Lagerverwaltung

Name	Beschreibung
Kopfzeile	In der Kopfzeile gibt es die folgenden Elemente: <ul style="list-style-type: none"> • «Zurück»: Reagiert analog dem Zurück-Button des Browsers. • Titel: Bezeichnung der aktuellen Seite.
Inhalt	Darstellung des Inhalts aufgrund der aktuellen URL
Fusszeile	Link zur Startseite und zu allen Einstiegsseiten der Bereiche.

Tabelle 24: Teile des Basis-Layouts des Frontend

Basis-Layout – Lagerinterface

Zusätzlich zum Basis-Layout der Lagerverwaltung beinhaltet das Lagerinterface Darstellungen zum aktuell verwendeten Betriebsmodus mittels Ribbons. Diese Eigenschaft fließt in das Basis-Layout, da sie jederzeit sichtbar sein soll.

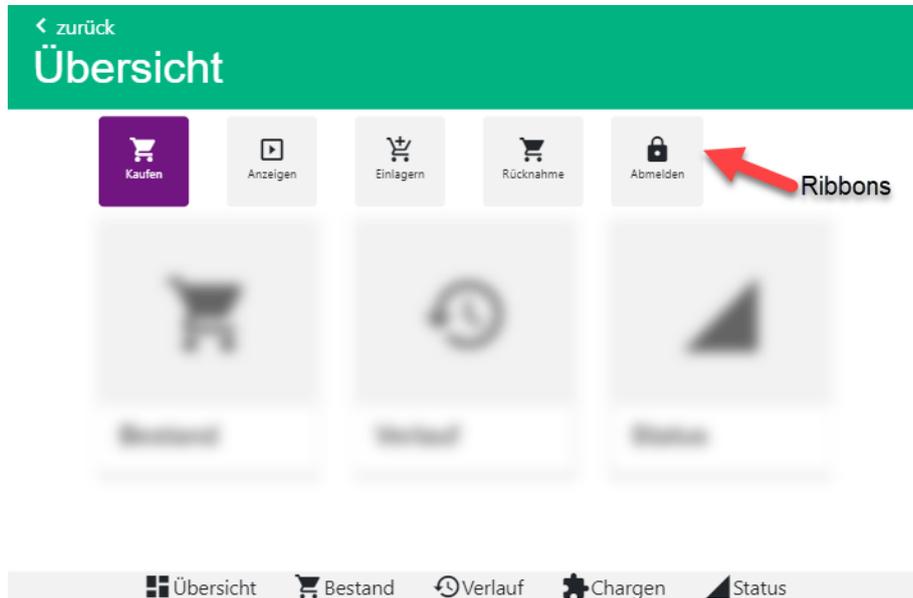


Abbildung 25: Darstellung des Betriebsmodus im Basis-Layout des Lagerinterfaces

Übersichts-Kacheln

Mittels einer Übersicht soll der Einstieg gestaltet werden. Jede dieser Kacheln besitzt jeweils einen Titel sowie ein Bild in Form eines Icons. Es sollen maximal drei Übersichts-Kacheln nebeneinander dargestellt werden.

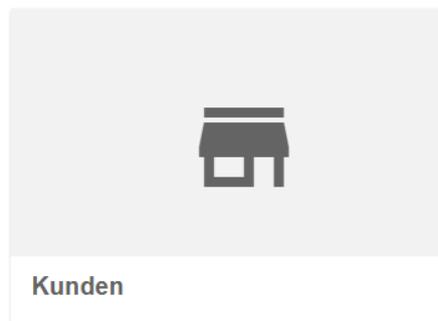


Abbildung 26: Designelement Kachel

Tabbar

Bei der Aufteilung von Eigenschaften einer Ansicht soll eine Tabbar verwendet werden. Die URL soll den selektierten Tab ablegen, sodass bei einem Reload der Seite der selektierte Tab nicht verloren geht. Die Position der Tabbar ist immer direkt unter der Kopfzeile und sieht wie folgt aus.



Abbildung 27: Designelement Tabbar

Detail-Edit

Um eine Entität zu bearbeiten, sollen Detail-Ansichten zur Verfügung gestellt werden. Um Aktionen vorzunehmen sollen Buttons (Speichern, Neu laden, Löschen) unten rechts eingefügt werden. Falls ein Bild der Entität hinterlegt werden kann, soll dieses auf der linken Seite dargestellt sein.

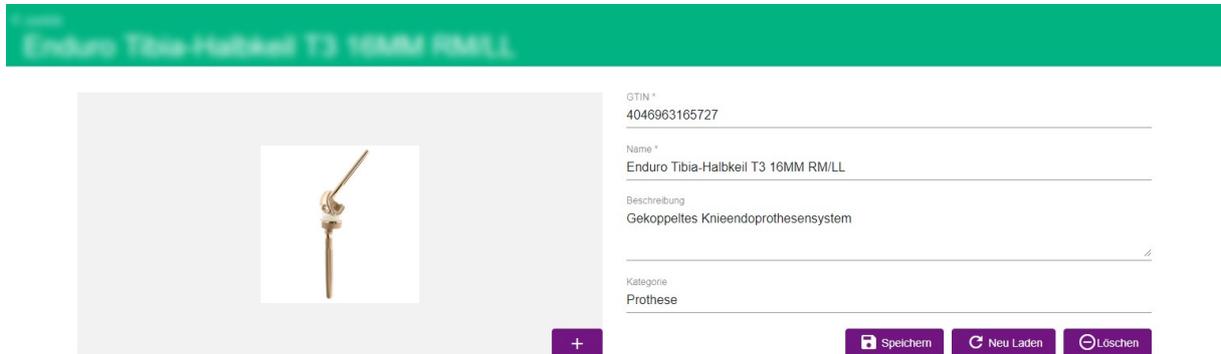


Abbildung 28: Designelement Detail-Edit mit Bilder Upload

Tabelle

Um Listen von Elementen darstellen zu können, sollen Tabellen verwendet werden. Falls das Eingabegerät eine Tastatur besitzt, soll zusätzlich ein Filter eingeblendet werden, welcher mittels Volltext-Suche in der Tabelle Suchabfragen tätigen kann.



Abbildung 29: Designelement Tabelle mit Filter

Im Lagerinterface soll ein Paging bei den Tabellen eingebaut werden, damit nicht gescrollt werden muss.

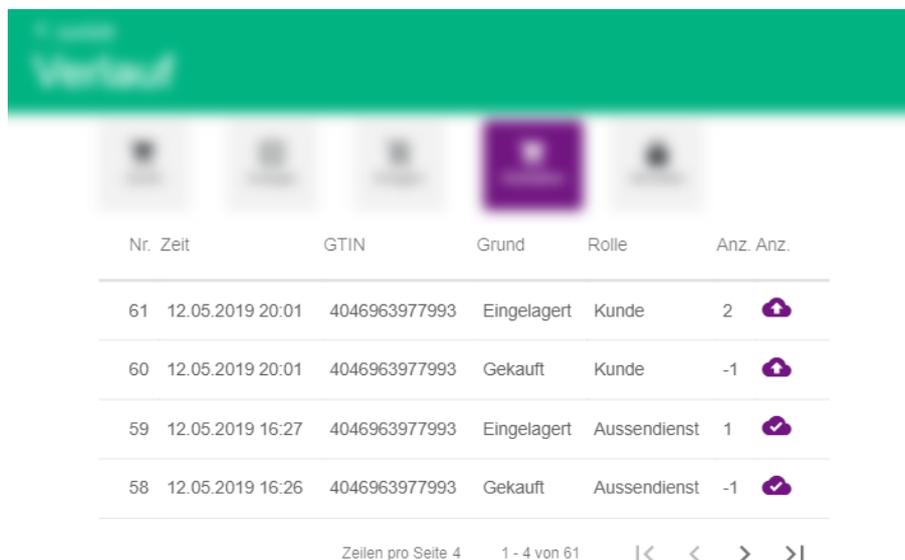


Abbildung 30: Designelement Tabelle ohne Filter, mit Paging

Dialog/Snackbar

Wichtige Benachrichtigungen für den Benutzer sollen mittels Snackbar oder Dialog dargestellt werden. Sobald ein Benutzer Eingaben machen muss, soll ein Dialog verwendet werden.

Darstellungsteile eines Dialogs:

- Titel
- Inhalt
- Aktionen



The image shows a dialog box with a green header containing the title 'Chargensperrung'. Below the header, there is a label 'Chargennummer' followed by the text '16K12809'. Underneath is a horizontal line. Below that is a label 'Bemerkung' followed by a horizontal line and a double-slash symbol '//'. At the bottom of the dialog, there are two buttons: 'Sperren' and 'Abbrechen'.

Abbildung 31: Designelement Dialog

Snackbars sollen angezeigt werden, falls Benachrichtigungen oder Fehler auftreten. Diese werden unten auf der Seite angezeigt und verschwinden nach angemessener Zeit wieder.



Abbildung 32: Designelement Snackbar

4. Realisierung

In diesem Kapitel wird die Realisierung des Projekts anhand des Lösungskonzepts (siehe Kapitel 3 Konzeption) erläutert. Als erstes werden die Architekturen der beiden Teilsysteme Lagerinterface und Lagerverwaltung und die Kommunikation beschrieben. Danach erfolgt kapitelweise die Umsetzung der geforderten Teilfunktionen. Am Anfang der Kapitel ist jeweils ein Verweis auf die zu erfüllende(n) Anforderung(en), durch die Aufzählung mit dem Zeichen  zu finden.

Abbildung 33 zeigt das realisierte E-Konsignationslager. Auf der linken Seite ist der Barcodescanner ersichtlich, welcher via USB-Kabel mit dem Raspberry Pi verbunden ist. Auf dem Touch Display sieht man das Frontend des Lagerinterfaces. Vom LoPy ist nur die Antenne (rechts vom Display) ersichtlich. In den beiden Boxen auf der rechten Seite werden die Artikel des E-Konsignationslagers abgelegt.



Abbildung 33: E-Konsignationslager

Die folgende Abbildung zeigt die Einstiegsseite der Lagerverwaltung.

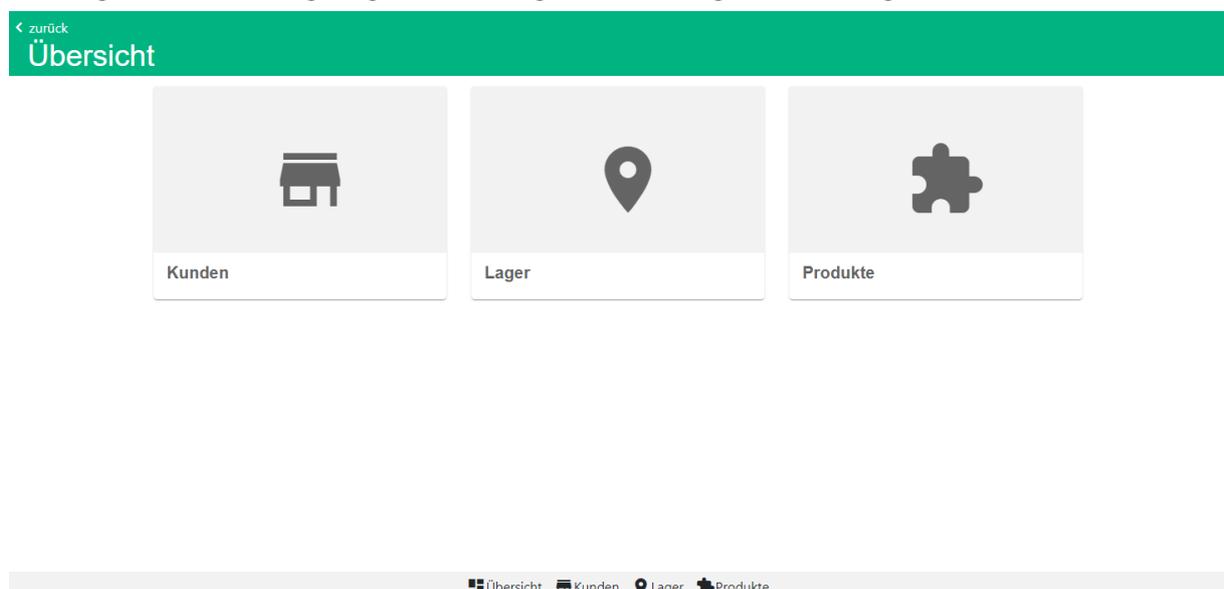


Abbildung 34: Benutzer-Interface der Lagerverwaltung

4.1. Architektur Backend Lagerinterface

Das Lagerinterface, bestehend aus Barcodescanner, Lora Modul und Touch Display, befindet sich direkt bei den Kunden vor Ort. Über das Lagerinterface können die Kunden einfach und unkompliziert die Lagerbestände verwalten. Das Lagerinterface wurde anhand des Lösungskonzepts (3.4.1 E-Konsignationslager) umgesetzt.

- 📄 AF#001 – Verteiltes System
- 📄 AF#003 – Datensicherheit
- 📄 AF#007 – Automatisiertes Startverhalten

Das folgende Diagramm zeigt die Abhängigkeiten im Backend. Die Abhängigkeiten zum Package «stockcore» wurden auf Grund der Übersichtlichkeit weggelassen und stattdessen mit einem Sternchen markiert.

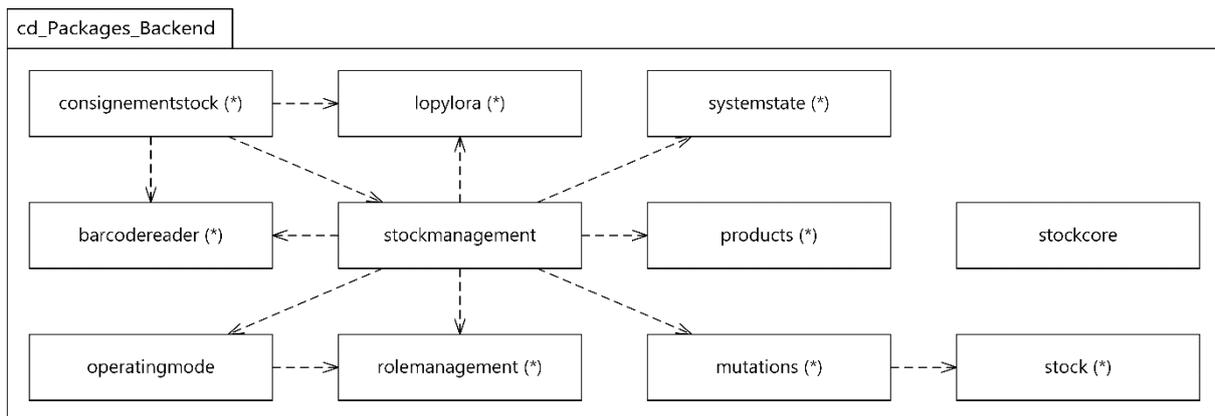


Abbildung 35: Paket-Diagramm des Lagerinterface

Paket	Beschreibung
consignmentstock	Bietet dem Frontend die Schnittstelle zum Backend an und enthält die startbare Backend-Applikation.
lopylora	Bietet Zugriff auf das Lora Modul.
systemstate	Kontrolliert den Verbindungsstatus.
barcodereader	Bietet Zugriff auf den Barcodescanner.
stockmanagement	Lokale Lagerverwaltung auf dem Lagerinterface, welche alle Pakete zusammenführt und Daten für die Backend-Applikation aufbereitet.
products	Enthält Produktinformationen für Anzeige.
stockcore	Enthält Entitäts- und Hilfsklassen, welche von anderen Paketen verwendet werden können.
operatingmode	Regelt die verschiedenen Betriebsmodi.
rolemangement	Ist zuständig für die Zugriffsberechtigungen.
mutations	Verwaltet die Bestandsmutationen.
stock	Unterhält den Lagerbestand.

Tabelle 25: Beschreibung der Pakete der Lagerverwaltung

4.1.1. Datenpersistenz

Für die Datenpersistierung verwendet das Lagerinterface das Filesystem des Raspberry Pi. Jede Softwarekomponente, welche Daten persistieren muss, regelt die Persistenz innerhalb des Pakets. Das heisst, dass es keinen globalen Persistenz-Layer im Lagerinterface gibt. Für die Persistenz wird trotzdem ein einheitliches Format, JSON-Dateien (Javascript Object Notation) verwendet. Dazu wird die Standardbibliothek «json»²³ von Python verwendet.

In den folgenden Kapiteln werden die persistierten Daten beschrieben.

Lagerbestand

Das Softwarepaket «stock» unterhält den aktuellen Lagerbestand. Pro Artikel werden der dazugehörige Barcode und der aktuelle Bestand gespeichert. Bestände, die kleiner oder gleich 0 sind, werden nicht gespeichert. Der folgende Ausschnitt zeigt ein Beispieleintrag für einen Artikel, welcher zweimal im Lager vorhanden ist.

```
[
  {
    "barcode": "0101234567890121170101011001234567890123456789",
    "inventory": 2
  }
]
```

Mutationen

Im Softwarepaket «mutations» werden die Lagerbestandsveränderungen gespeichert. Dazu pflegt das Softwarepaket zwei Dateien mit demselben Format. In einer Datei werden die ausstehenden Mutationen gespeichert – Mutationen, welche noch nicht an die Lagerverwaltung gesendet wurden – und in der anderen Datei die verarbeiteten Mutationen. Wenn eine Mutation verarbeitet wurde, wird diese von der einen Datei in die andere vorschoben. Das folgende Beispiel zeigt das Format einer Mutation. In dieser wurden zwei Artikel vom Standard Benutzer eingelagert.

```
[
  {
    "amount": 2,
    "barcode": "0104046963977993172012311001234567890123456789",
    "number": 1,
    "origin": "2019-05-15T17:04:20.450123",
    "reason": "STORE",
    "role": "DEFAULT_ROLE"
  }
]
```

Produktinformationen

Zusätzliche Informationen zu Produkten, welche nicht im Barcode enthalten sind, werden im Softwarepaket «products» verwaltet. Produktinformationen werden über den GTIN identifiziert. Im folgenden Beispiel ist das Format für einen solchen Eintrag abgebildet. Aus Übersichtsgründen wurde der Inhalt des Properties «image» nicht dargestellt. An dieser Stelle würden die Binärdaten des Bildes abgelegt.

²³ <https://docs.python.org/3/library/json.html>

```
[
  {
    "category": "Prothese",
    "description": "Gekoppeltes Knieendoprothesensystem",
    "gtin": 4046963165727,
    "image": "Aus übersichtsgründen nicht dargestellt",
    "name": "Enduro Tibia-Halbkeil T3 16MM RM/LL"
  }
]
```

Chargenrückruf

Chargenrückrufe werden im Softwarepaket «stockmanagement» verwaltet. Dort werden gesperrte Chargen gespeichert. Das folgende Beispiel zeigt einen Eintrag für eine zurückgerufene Charge.

```
[
  "01234567890123456789"
]
```

4.1.2. API-Schnittstelle

Die Schnittstelle zwischen dem Frontend und dem Backend erfolgt über eine REST-Schnittstelle, die das Softwarepaket «consignmentstock» zur Verfügung stellt. Diese wird mit dem Python Framework Flask²⁴ umgesetzt und ist über folgenden Aufruf erreichbar.

```
http://localhost/api
```

Mittels sogenannten Blueprints werden die verschiedenen Schnittstellen separiert. In der folgenden Tabelle sind diese Schnittstellen zusammengefasst.

Ziel	Beschreibung
/history	Historie der Bestandsmutationen
/items	Informationen zum Lagerbestand
/modes	Zugriff zu den Betriebsmodi
/roles	Zugriff zur Benutzerverwaltung
/restart_notification	Benachrichtigung zu ausstehenden Neustarts
/scan_notifications	Benachrichtigung beim Scannen eines Produkts
/system_state	Informationen zum Systemstatus

Tabelle 26: Verwendete Blueprints des Lagerinterface Backend

Event-basierte Benachrichtigungen werden via Eventstream²⁵ umgesetzt. Dies erlaubt der Serverapplikation (Lagerinterface) das Frontend zu benachrichtigen. Diese Variante ist effizienter, als wenn das Frontend regelmässig Werte abfragt.

²⁴ <http://flask.pocoo.org/>

²⁵ https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events/Using_server-sent_events

Historie der Bestandsmutationen

Ziel	Methode	Beschreibung
/	GET	Gibt eine Auflistung aller Mutationen zurück. Beispiel Rückgabewert: <pre>[{ "amount": -1, "barcode": { "expiry": "2020-12-31", "gtin": 4046963977993, "lot_id": "01234567890123456789" }, "is_processed": true, "number": 8, "origin": "2019-05-19T18:07:32.133343", "reason": "BUY", "role": "DEFAULT_ROLE" }]</pre>

Tabelle 27: Historie der Bestandsmutationen

Informationen zum Lagerbestand

Ziel	Methode	Beschreibung
/	GET	Gibt den Lagerbestand zurück. Beispiel Rückgabewert: <pre>[{ "barcode": { "expiry": "2020-12-31", "gtin": 4046963977993, "lot_id": "01234567890123456789" }, "expired": false, "expires": false, "inventory": 1, "lot_called_back": false, "product": { "category": "Verankerungsschraube", "description": "", "gtin": 4046963977993, "image": "Aus übersichtsgründen nicht dargestellt " }, "name": "Verankerungsschraube D6,5MM" }]</pre>

		<pre>L16MM SW3,5" } }]</pre>
/manual_mutation	POST	<p>Führt eine manuelle Mutation durch anhand des mitgegebenen Barcodes aus.</p> <p>Beispiel für Daten:</p> <pre>{ "gtin": "4046963977993", "expiry": "2020-12-31", "lot_id": "01234567890123456789" }</pre>
/abort_mutation	POST	Bricht die letzte Mutation ab, sofern diese noch nicht verarbeitet ist.

Tabelle 28: Informationen zum Lagerbestand

Zugriff zu den Betriebsmodi

Ziel	Methode	Beschreibung
/	GET	<p>Gibt den aktuellen Betriebsmodus zurück.</p> <p>Beispiel Rückgabewert:</p> <pre>{ "mode": "BUY" }</pre>
/	POST	<p>Ändert den Betriebsmodus</p> <p>Beispiel für Daten:</p> <pre>{ "mode": "BUY" }</pre>

Tabelle 29: Zugriff zu den Betriebsmodi

Zugriff zur Benutzerverwaltung

Ziel	Methode	Beschreibung
/	GET	<p>Gibt die Rolle des aktuellen Benutzers zurück.</p> <p>Beispiel Rückgabewert:</p> <pre>{ "role": "DEFAULT_ROLE" }</pre>

/	POST	Login für Benutzer Beispiel für Daten:
		<pre>{ "role": "DEFAULT_ROLE" "pin_code": "GEHEIM" }</pre>

Tabelle 30: Zugriff zur Benutzerverwaltung

Benachrichtigung zu ausstehenden Neustarts

Ziel	Methode	Beschreibung
/local_event	GET	Eventstream für Benachrichtigung, wenn Neustart anstehend ist. Beispiel Event:
		<pre>data: {"pending": "true", "time_till_restart": "00:00:27"}\r\n</pre>
/local	GET	Informationen zum anstehenden Neustart. Beispiel Rückgabewert:
		<pre>{ "pending": "true", "time_till_restart": "00:00:27" }</pre>
/local	POST	Abbruch des Neustarts
/remote_event	GET	Eventstream für Benachrichtigung, wenn Fern-Neustart anstehend ist. Beispiel Event:
		<pre>data: {"pending": "true", "time_till_restart": "00:00:27"}\r\n</pre>
/remote	GET	Informationen zum anstehenden Fern-Neustart Beispiel Rückgabewert:
		<pre>{ "pending": "true", "time_till_restart": "00:00:27" }</pre>
/remote	POST	Abbruch des Fern-Neustarts

Tabelle 31: Benachrichtigung zu ausstehenden Neustarts

Benachrichtigung beim Scannen eines Produkts

Ziel	Methode	Beschreibung
/scan_event	GET	<p>Eventstream für Benachrichtigung, wenn ein Produkt gescannt wird</p> <p>Beispiel Format Event:</p> <pre>data: {"mode": "BUY", "item": "Siehe Informationen zum Lagerbestand" }\r\n</pre>

Tabelle 32: Benachrichtigung beim Scannen eines Produkts

Informationen zum Systemstatus

Ziel	Methode	Beschreibung
/	GET	<p>Gibt Informationen zum aktuellen Systemstatus zurück. Werte in Grün werden nur zurückgegeben, wenn der Aussendienstmitarbeiter angemeldet ist.</p> <p>Beispiel Rückgabewert:</p> <pre>{ "BarcodeReader": { "Device name": "Datalogic Scanning, Inc. Point of Sale Handable Scanner", "LastBarcode": { "expiry": "2021-12-31", "gtin": 4046963165727, "lot_id": "ABCDEFGHJKLMNOPQRST" }, "LastRead": "2019-05-19T18:58:57.143800" }, "Connection": { "ConnectionHistory": { "2019-05-19T18:23:25.424701": true, "2019-05-19T18:23:55.427438": true, }, "CurrentIp": "192.168.1.106", "LastConnection": "2019-05-19T19:00:55.554754" }, "LoRa": { "Connected": true, "DeviceAddress": "70B3D54993738B5A", "History": ["No history"] }, "PlannedRestart": "", "User": {</pre>

		<pre>"LastLogin": "2019-05-19T19:01:08.643209", "SignedIn": "PRIVILEGED_ROLE" } }</pre>
/connection_test	POST	Startet eine manuelle Verbindungskontrolle mit dem LoRa-Netzwerk.
/product_import	GET	Startet den Import des Artikelstamms und gibt die Anzahl importierter Produkte zurück. Beispiel Rückgabewert:
		<pre>{ "products": 502 }</pre>

Tabelle 33: Informationen zum Systemstatus

4.1.3. Kommunikation mit LoRa Modul

Auf dem LoPy läuft separat zum Raspberry Pi ein Programm, welches für die LoRa-Kommunikation zuständig ist. Das Programm wertet eingehende Kommandos von der seriellen Schnittstelle aus und führt diese dann aus.

Das LoPy bietet eventbasierte Benachrichtigungen an, wenn über die LoRa-Antenne Daten empfangen werden. Dazu stellt der Hersteller eine LoRa-Komponente zur Verfügung, welcher intern das LoRaWAN-Protokoll implementiert.

Daten senden

Um Daten über LoRa zu übertragen, kann das Raspberry Pi über die serielle UART Schnittstelle die Daten verschicken. Das LoPy wertet die empfangenen Daten aus und schickt einen Status zurück, ob die Daten gesendet wurden oder nicht.

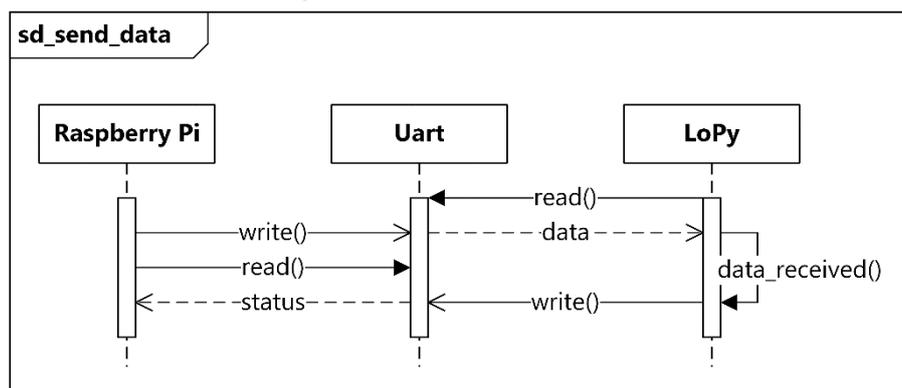


Abbildung 36: Daten über LoRa verschicken

Um Daten zu senden, muss folgendes auf die serielle Schnittstelle geschrieben werden:

```
send [Daten, die übertragen werden müssen]
```

Intern werden auf dem LoPy innerhalb einer Endlosschleife die seriellen Daten ausgewertet und an die LoRa-Komponente, welche vom Hersteller zur Verfügung gestellt wird, verschickt.

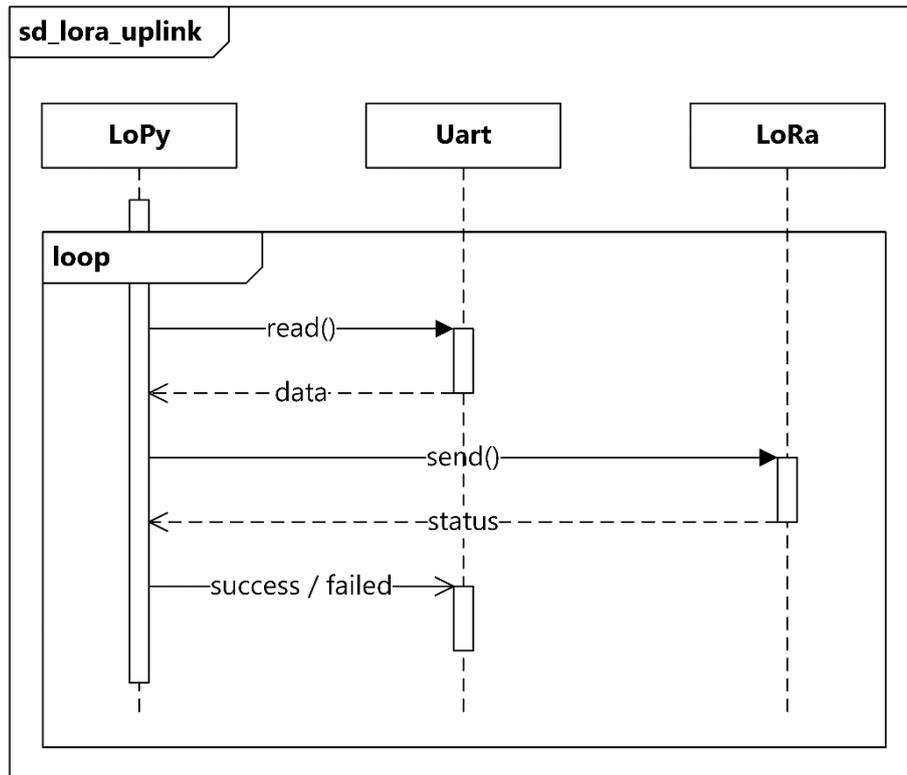


Abbildung 37: Lebenszyklus Daten senden

Daten empfangen

Die LoRa-Komponente bietet die Möglichkeit ein Callback zu hinterlegen. Dieser wird jedes Mal aufgerufen, wenn die LoRa-Antenne Daten empfängt. Das LoPy legt danach die Daten auf die serielle Schnittstelle.

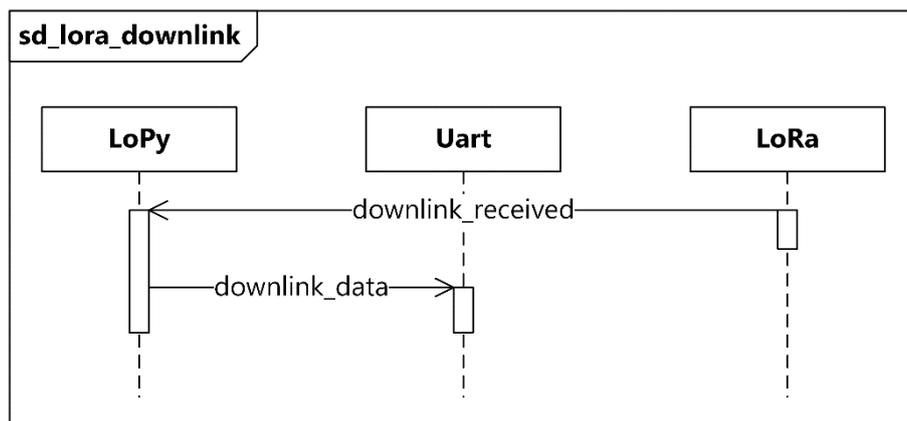


Abbildung 38: Daten über LoRa empfangen

Daten, welche empfangen werden, haben folgendes Format:

Data=[Empfange Daten]

Das Raspberry Pi ist dafür verantwortlich, die serielle Schnittstelle auszulesen und die Daten zu interpretieren.

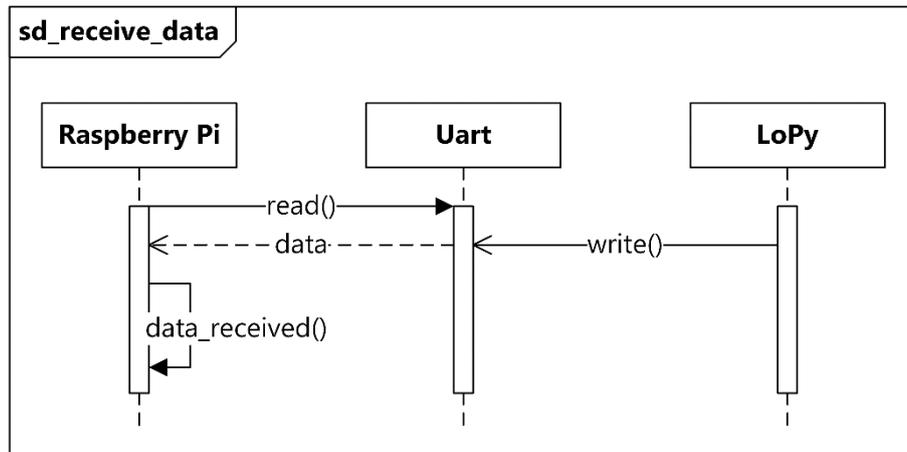


Abbildung 39: Empfangene Daten vom LoPy lesen

Status abfragen

Das Programm auf dem LoPy bietet zusätzlich die Möglichkeit einen Status abzufragen. Dieser enthält die eindeutige Geräteadresse, welche zur Inbetriebnahme und zur Anzeige verwendet wird, sowie den Verbindungsstatus mit dem Swisscom LPN.

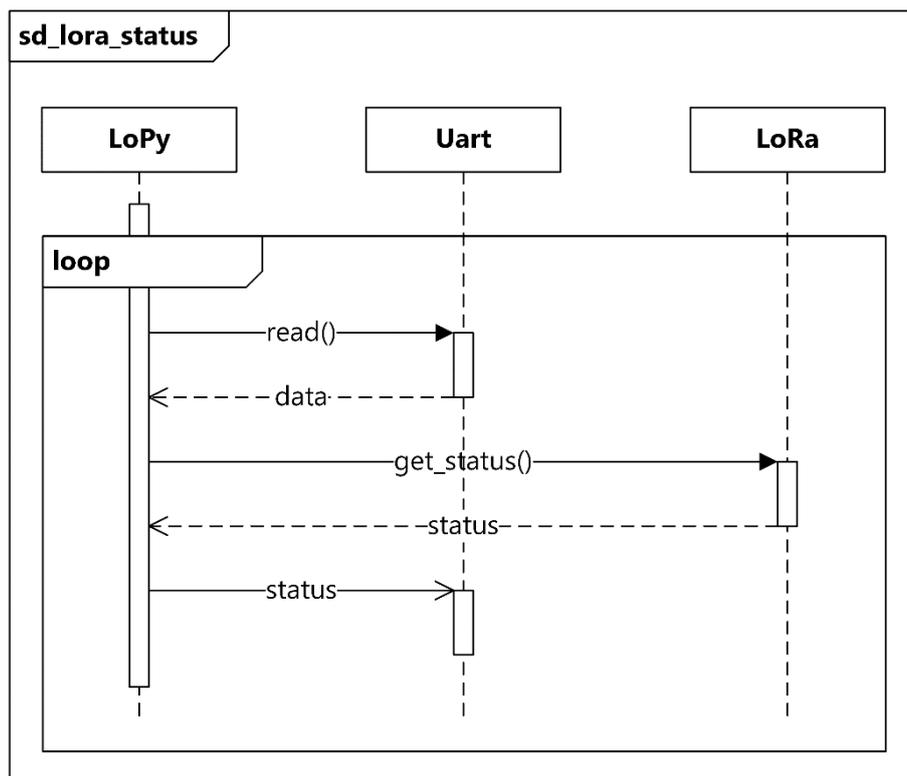


Abbildung 40: Status abfragen

Um den Status abzufragen muss folgendes Kommando gesendet werden:

`status`

4.1.4. Konfiguration

Die Hauptapplikation kann über eine Konfigurationsdatei extern konfiguriert werden. Der folgende Ausschnitt zeigt die Standardkonfiguration im System. Die Konfiguration ist unterteilt in mehrere Sektionen. Das Format der Konfigurationsdatei basiert auf dem Microsoft Windows INI Format²⁶. Die verschiedenen Parameter werden noch genauer erklärt.

[DEFAULT]

MutationHandlerInterval = 00:10:00
MutationPipeInterval = 00:00:10
PendingMutationsFile = pending_mutations.json
ProcessedMutationsFile = processed_mutations.json
ProductsFile = products.json
StockFile = stock.json
LotCallbackFile = lot_callback.json

[barcode_reader]

DeviceName = Datalogic Scanning, Inc. Point of Sale Handable Scanner

[lora]

DeviceName = /dev/ttyACM0

[system_state]

AutomaticRestartTime = 01:00:00
ConnectionMonitorTime = 10:00:00
ConnectionTestInterval = 02:00:00
WarnTimeBeforeLocalRestart = 00:05:00
WarnTimeBeforeRemoteRestart = 00:02:00

Standard Sektion

Einstellung	Bedeutung
MutationHandlerInterval	Legt das Zeitintervall fest, nachdem die Mutationen vom Konsignationslager an die Lagerverwaltung übertragen werden (siehe 4.5.2 Mutationen übertragen).
MutationPipeInterval	Wenn ein Produkt fälschlicherweise gekauft wurde, hat der Benutzer die Möglichkeit die Mutation während einer bestimmten Zeit abzurechnen. Die Zeit, welche dem Benutzer zum Abrechnen bleibt, kann über diese Einstellung festgelegt werden.
PendingMutationsFile	Diese Einstellung legt fest, in welcher Datei die pendenten Mutationen persistiert werden.
ProcessedMutationsFile	Diese Einstellung legt fest, in welcher Datei die verarbeiteten Mutationen persistiert werden.
ProductsFile	Diese Einstellung legt fest, in welcher Datei die zusätzlichen Produktinformationen abgelegt sind.

²⁶ <https://docs.python.org/3/library/configparser.html>

StockFile	Diese Einstellung legt fest, in welcher Datei der Lagerbestand gespeichert wird.
LotCallbackFile	Diese Einstellung legt fest, in welcher Datei die zurückgerufenen Chargen gespeichert sind.

Tabelle 34: Einstellungen Standard Sektion

Barcodescanner Sektion

Einstellung	Bedeutung
DeviceName	Gibt den Namen des Barcodescanners an. Dieser Name wird verwendet, um auf den Barcodescanner zuzugreifen.

Tabelle 35: Einstellungen Barcodescanner Sektion

LoRa Sektion

Einstellung	Bedeutung
DeviceName	In dieser Einstellung wird der Gerätenamen des LoRa Moduls angegeben

Tabelle 36: Einstellungen Lora Sektion

Systemstatus Sektion

Einstellung	Bedeutung
AutomaticRestartTime	Diese Einstellung legt fest, um welche Uhrzeit das Gerät neugestartet wird, wenn keine LoRa-Verbindung besteht. Die Einstellung muss auf 1:00 Uhr eingestellt werden (siehe AF#135 – Automatischer Neustart).
ConnectionMonitorTime	Gibt die Zeit an, wie lange keine LoRa-Verbindung vorhanden sein muss, bis das Gerät neugestartet wird. Die Einstellung muss auf 10 Stunden eingestellt werden (AF#135 – Automatischer Neustart).
ConnectionTestInterval	Mit dieser Einstellung wird festgelegt, nach welcher Zeit eine automatische Verbindungskontrolle durchgeführt wird.
WarnTimeBeforeLocalRestart	Diese Einstellung legt fest, wie viel Zeit dem Benutzer bleibt, bis das Gerät neugestartet wird, wenn keine LoRa-Verbindung besteht. Die Einstellung muss auf 5 Minuten eingestellt werden (siehe AF#135 – Automatischer Neustart).
WarnTimeBeforeRemoteRestart	Diese Einstellung legt fest, wie viel Zeit dem Benutzer bleibt, bis das Gerät neugestartet wird, wenn dies von der Lagerverwaltung gewünscht wurde.

Die Einstellung muss auf 2 Minuten eingestellt werden (AF#136 – Fern Neustart)

Tabelle 37: Einstellungen Systemstatus Sektion

4.1.5. Konfiguration für Fehleranalyse

Die erstellten Softwarepakete verwenden zum Loggen die Standardbibliothek «logging»²⁷ von Python. Diese bietet die Möglichkeit über eine separate Konfigurationsdatei die Ausgabe der verwendeten Logger einzustellen. Im Lagerinterface ist der Logger so eingestellt, dass dieser pro Tag eine Log-Datei erstellt und diese über einen Zeitraum von 30 Tagen aufbewahrt. Dies soll den Aussendienstmitarbeiter helfen, eine genaue Fehleranalyse vor Ort durchzuführen. Nachfolgend wird die verwendete Konfiguration gezeigt.

```
[loggers]
keys=root

[handlers]
keys=consoleHandler,fileHandler

[formatters]
keys=sampleFormatter

[logger_root]
level=DEBUG
handlers=consoleHandler,fileHandler

[handler_consoleHandler]
class=StreamHandler
level=DEBUG
formatter=sampleFormatter
args=(sys.stdout,)

[handler_fileHandler]
class=handlers.TimedRotatingFileHandler
level=DEBUG
formatter=sampleFormatter
args=('var/log/locastra.consignmentstock.log','D',1,30,None,False,True)

[formatter_sampleFormatter]
format=%(asctime)s - %(name)s - %(lineno)d - %(levelname)s - %(message)s
```

4.2. Architektur Backend Lagerverwaltung

Durch die Lagerverwaltung können die Daten aller E-Konsignationslager verwaltet werden. Aussendienstmitarbeiter können hier Informationen über E-Konsignationslager beziehen, ohne dass sie vor Ort beim Kunden sind oder mit diesem in Kontakt stehen müssen. Technische Mitarbeiter können die technischen Details der Lagerinterfaces überwachen.

²⁷ <https://docs.python.org/3/library/logging.html>

Anhand des Lösungskonzepts (3.4.1 E-Konsignationslager) wurden folgende Anforderungen umgesetzt.

- 📄 AF#001 – Verteiltes System
- 📄 AF#002 – Skalierbarkeit
- 📄 AF#003 – Datensicherheit
- 📄 AF#007 – Automatisiertes Startverhalten
- 📄 AF#008 – Betriebsdauer

Die folgenden Softwarekomponenten werden in der Lagerverwaltung verwendet. Zusätzlich wird die API-Schnittstelle vom Swisscom LPN Portal verwendet, um LoRa Meldungen abzusetzen und zu erhalten.

Die umgesetzten Software-Komponenten werden mittels IIS zur Verfügung gestellt. Dieser startet die beiden Applikationen (Frontend, Backend) automatisch bei einem Update und stellt sicher, dass die Zustände der Applikationen bei einem System-Neustart wiederhergestellt werden.

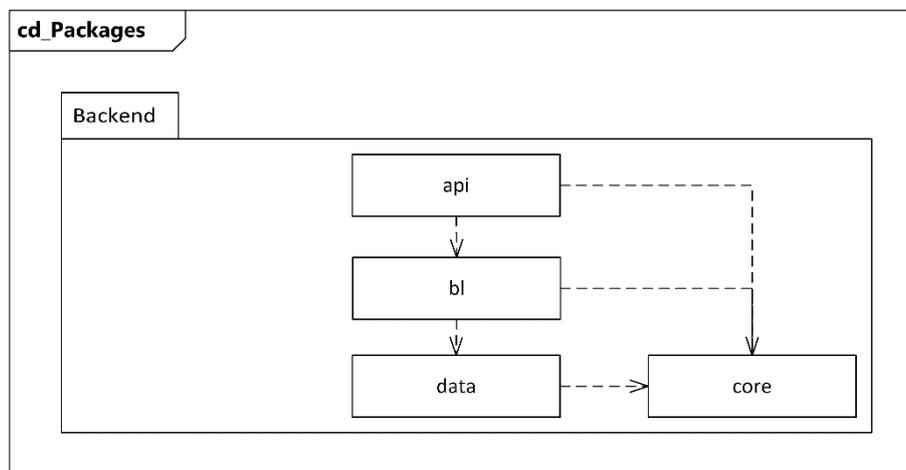


Abbildung 41: Paket-Diagramm der Lagerverwaltung

Paket	Beschreibung
api	Bereitstellung der REST-Schnittstelle via HTTP mittels OData, dient zusätzlich als Schnittstelle zum Empfang von Meldungen via LPN.
bl	Business-Logik der Lagerverwaltung.
core	Kern-Funktionalitäten, Hilfsmethoden, etc.
data	Verbindung zur Datenbank mittels Entity-Framework.

Tabelle 38: Beschreibung der Pakete der Lagerverwaltung

4.2.1. Datenpersistenz

Die folgenden Entitäten werden in der Lagerverwaltung verwendet. Um diese zu persistieren wird der Microsoft SQL Server verwendet, welcher bei einem System-Neustart automatisch startet. Für die Verwendung in der Business-Schicht werden sie mittels Microsoft Entity-Framework²⁸ aufbereitet.

²⁸ <https://docs.microsoft.com/en-us/ef/>

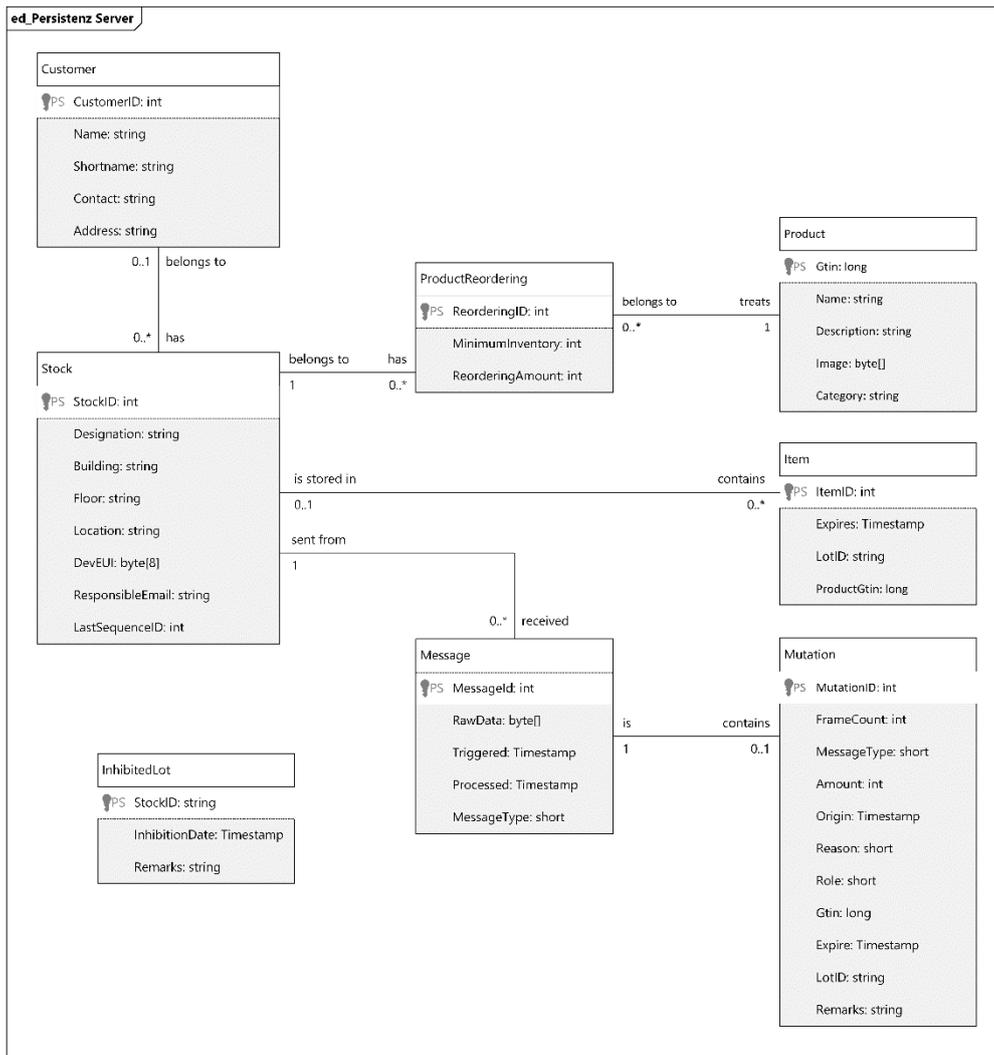


Abbildung 42: Entitäten-Diagramm Lagerverwaltung

Entität	Beschreibung
Customer	Kunden mit relevanten Informationen für die Lagerverwaltung.
Stock	E-Konsignationslager beim Kunden.
Product	Artikelstamm aller Produkte der B. Braun Medical AG, welche in Konsignationslagern bezogen werden können.
ProductReordering	Festlegung der Mindestbestände und der Nachbestellmengen für Lager und Produkte.
Item	Eingelagerte Artikel in den E-Konsignationslagern.
Message	Nachrichten, welche via LPN empfangen wurden.
Mutation	Mutationen, welche im E-Konsignationslager vorgenommen wurden.
InhibitedLot	Gesperrte Chargen.

Tabelle 39: Beschreibung der Entitäten der Lagerverwaltung

4.2.2. API-Schnittstelle

Um die Daten der Lagerverwaltung bereit zu stellen, wurde das Protokoll OData²⁹ eingesetzt. Mit dem OData Package³⁰, welches durch Microsoft bereitgestellt wird, können Entitäten aus dem Entity-Framework direkt für eine REST-Schnittstelle aufbereitet werden. Zusätzlich können Query-Optionen mitgegeben werden, um nur die relevanten Daten vom Backend abzufragen.

Format für die OData URL für eine Abfrage auf ein Entity-Set.

http://Service-Root/Entity-Set?Query-Operation

Service-Root: Stellt die Root-Adresse des OData-Services dar.

Entity-Set: Entitäten eines Typs.

Query-Operation: Operationen, welche auf einem Entity-Set oder einer Entität ausgeführt werden können.

Format für die Abfrage einer spezifischen Entität

http://Service-Root/Entity-Set(Identity)?Query-Operation

Identity: Identität einer Entität (z.B. GTIN eines Produkts)

Mittels HTTP-Methoden können Selektionen oder Mutationen auf der Schnittstelle durchgeführt werden.

Ziel	Methode	Beschreibung
/Entity-Set	GET	Lesen einer Entitätsmenge.
	POST	Hinzufügen einer neuen Entität.
/Entity-Set(Identity)	GET	Lesen einer Entität.
	PUT	Ersetzen einer Entität mit derselben Identität.
	PATCH	Ändern einer Entität.
	DELETE	Löschen einer Entität.

Tabelle 40: Beschreibung der HTTP-Methoden bei der OData API

Mittels Query-Operationen werden eine Entität oder Entitätsmenge zusätzlich verarbeitet. Folgende Operationen werden in der Lagerverwaltung implementiert.

Operation	Beschreibung
\$count	Die Entitäten der Selektion werden gezählt. Es wird nur diese Anzahl zurückgegeben.
\$filter	Gewisse Eigenschaften der Entitäten können im Aufruf gefiltert werden.
\$orderBy	Es können Eigenschaften zur Sortierung mitgegeben werden.
\$select	Nur gewisse Eigenschaften der gefundenen Entitäten sollen zurückgegeben werden.

Tabelle 41: Beschreibung der Query-Operationen der OData API

²⁹ <https://www.odata.org/>

³⁰ <https://www.nuget.org/packages/Microsoft.AspNet.OData/>

Für die Implementierung der einzelnen Funktionalitäten der Lagerverwaltung wurden folgende Entity-Sets definiert. Des Weiteren werden für spezifische Entitäten Zusatzfunktionen bereitgestellt.

EntitySet	Zusatzfunktion	Entität/Beschreibung
Products		Produkte
	<i>Import</i>	<i>Import des Artikelstamms</i>
Customers		Kunden
	<i>Stocks</i>	<i>Lager, welche dem Kunden zugeordnet sind.</i>
Stocks		E-Konsignationslager
	<i>ConnectionState</i>	<i>Verbindungsstatus eines Lagers</i>
	<i>Mutations</i>	<i>Mutationen eines gewissen Lagers</i>
	<i>RestartDevice</i>	<i>LoRa Gerät neu starten</i>
Items		Artikel, welche in den Lagern verfügbar sind
	<i>Lots</i>	<i>Chargen im Umlauf</i>
Messages		Meldungen
Mutations		Mutationen
ProductReordering		Nachbestellungs-Konfigurationen
InhibitedLots		Gesperrte Chargen
Lora ³¹		LoRa Meldungen empfangen (Endpoint, hinterlegt bei der Swisscom)

Tabelle 42: Entitäten der API Schnittstelle der Lagerverwaltung

4.2.3. Konfiguration

Konfigurationen der Lagerverwaltung können in der Datei «web.config» im Standard .Net Framework Konfigurationsformat³² vorgenommen werden. Die folgenden Elemente beinhalten die wichtigsten Einstellungen.

```
<connectionStrings>
  <add name="consignmentManagementEntities" connectionString="..."
  providerName="System.Data.EntityClient" />
</connectionStrings>
...
<Hslu.ConsignmentStock.BL.Properties.Settings>
  <setting name="LoraRequestsPath" serializeAs="String">
    <value>...</value>
  </setting>
  <setting name="SmtpServerAddress" serializeAs="String">
    <value>...</value>
  </setting>
  <setting name="SmtpMail" serializeAs="String">
    <value>...</value>
  </setting>
</Hslu.ConsignmentStock.BL.Properties.Settings>
```

³¹ Virtuelle Entität: Entität ohne Datenbankeintrag

³² <https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/index>

```

</setting>
<setting name="SmtpPassword" serializeAs="String">
  <value>...</value>
</setting>
<setting name="SmtpServerPort" serializeAs="String">
  <value>...</value>
</setting>
</Hslu.ConsignmentStock.BL.Properties.Settings>
</applicationSettings>

```

Einstellung	Bedeutung
ConnectionString	Verbindungseinstellungen zum Datenbank-Server ³³
LoraRequestsPath	Pfad, um eingehende LoRa Nachrichten zu speichern. Falls leer gelassen, werden Nachrichten nicht gespeichert.
SmtpServerAddress	SMTP Server für Benachrichtigungen
SmtpServerPort	Port des SMTP Servers
SmtpMail	E-Mail-Adresse, von der aus Benachrichtigungen versendet werden
SmtpPassword	Passwort für das Login beim SMTP Server

Abbildung 43 Konfiguration Lagerverwaltung

4.3. Architektur Frontend Lagerinterface und Lagerverwaltung

Der Aufbau des Frontend des Lagerinterfaces und der Lagerverwaltung ist derselbe. Entsprechend wurden gleiche Komponenten und Elemente verwendet. Folgende Anforderungen werden mit dem Frontend umgesetzt:

- 📄 AF#023 – Anzeige
- 📄 AF#024 – Corporate Design

Das Frontend wurde mit Angular implementiert. Es soll als Schnittstelle gegenüber den Mitarbeitern der B. Braun dienen.

³³ <https://www.connectionstrings.com/sql-server/>

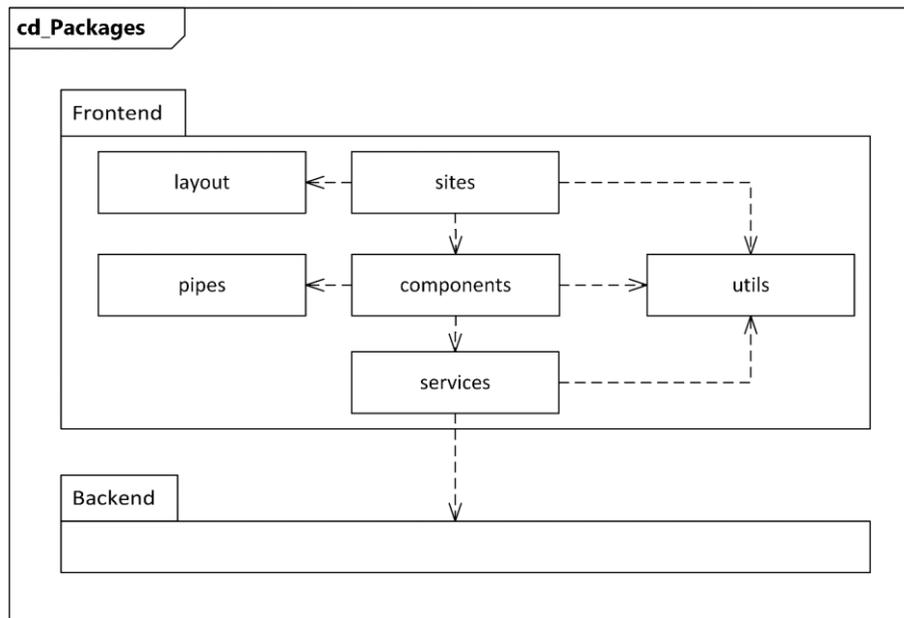


Abbildung 44: Paket-Diagramm des Frontend

Paket	Beschreibung
layout	Basis-Layout basierend auf dem Design-Konzept (3.5 Design)
sites	Einzelne Seiten des Frontend
components	Komponenten wie Tabellen, Dialoge, etc.
pipes	Umwandlungen für die Oberfläche (z.B. Übersetzung)
utils	Allgemeine Hilfsmethoden
services	Services, um via HTTP auf das Backend zuzugreifen.

Tabelle 43: Beschreibung der Pakete des Frontend

4.4. Kommunikation

- 📄 AF#011 – Funktechnologie
- 📄 AF#012 – Optimierte Datenübertragung

In diesem Kapitel wird das Kommunikationsprotokoll zwischen dem Lagerinterface und der Lagerverwaltung beschrieben. Das proprietäre Protokoll hat folgende Eigenschaften: Verbindungslos, nicht-zuverlässig, ungesichert und ungeschützt.

4.4.1. Verbindung

Das gesamte Protokoll ist verbindungslos, weil keine direkte End-zu-End-Verbindung zwischen Lagerinterface und Lagerverwaltung aufgebaut wird. Zwischen den einzelnen Verbindungsteilnehmern ist die Kommunikation verbindungsorientiert.

Die Verbindung zwischen Lagerinterface und Swisscom LPN wird durch den Join-Mechanismus des LoRaWAN-Protokolls stetig aufrechterhalten. Beim Datenversand werden bestätigte Nachrichten eingesetzt, das heisst, dass der Empfänger eine Bestätigung zurückmeldet (LoRa Alliance™ [1], 2019). Die Verbindung zwischen Swisscom LPN und

Applikationsserver läuft über HTTP, welche auf TCP³⁴ aufbaut und ist per Definition verbindungsorientiert.

4.4.2. Zuverlässigkeit

Die Zuverlässigkeit des Protokolls beim Uplink ist auf Grund des Swisscom LPN grundsätzlich gegeben. Die Swisscom garantiert die Übermittlung von Endgerät zu Applikationsserver, sofern der Applikationsserver verfügbar ist. Beim Downlink ist die Zuverlässigkeit nicht gegeben. Die Swisscom erwähnt explizit, dass beim Downlink das erneute Senden von verlorenen Paketen Aufgabe des Applikationsservers ist (Swisscom [III], 2019). Dies bedeutet, dass nicht sichergestellt ist, dass alle Nachrichten ankommen. Aus diesem Grund wurde der Synchronisationsmechanismus eingeführt. Das Lagerinterface sendet in der Heartbeat-Nachricht zusätzlich die letzte, verschickte Mutationsnummer. Die Lagerverwaltung erhält dadurch die Möglichkeit eine Inkonsistenz zu erkennen und fehlende Mutationen nachzufragen und nachträglich zu verarbeiten.

4.4.3. Datensicherheit

Bezüglich der Datensicherheit gilt speziell zu erwähnen, dass das Swisscom LPN die Downlink-Nachrichten nicht auf ihren Ursprung überprüft. Die Downlink API stellt eine URL zur Verfügung, welche von überall aufgerufen werden kann. Sobald also die eindeutige Gerätenummer (DEVEUI) bekannt ist, kann jeder Datenpakete an das Lagerinterface senden (Swisscom [III], 2019).

4.4.4. Datenschutz

Auf eine komplette Verschlüsselung der Kommunikation wird in dieser Arbeit verzichtet, da diese nicht gefordert wurde. Beim Join-Mechanismus wird die Join-Request verschlüsselt. Dies ist beim OTAA-Verfahren so vorgegeben und verhindert, dass sich das Lagerinterface mit einem anderen Gateway verbindet (z.B. Gateways vom «The Things Network») (LoRa Alliance™ [I], 2019). Die effektive Datenübertragung über LoRa, sowie die Weiterleitung vom Swisscom LPN zur Lagerverwaltung (HTTP) ist jedoch unverschlüsselt.

4.4.5. Nachrichten

Die folgende Tabelle liefert eine Übersicht über die verschiedenen Nachrichten im System. In den folgenden Abschnitten werden die jeweiligen Nutzdaten jedes Nachrichtentyps aufgelistet und erklärt. Die Definitionen und der Grundaufbau der Nachrichten sind im Kapitel 3.4.3 Kommunikation unter Protokoll definiert.

Nachricht	Nachrichtentyp	Richtung	Länge ³⁵
Bestandsmutation	0	Uplink	46 Byte
Heartbeat	1	Uplink	5 Byte
Mutationsanfrage	2	Downlink	1 Byte + 4 * n Byte n := [1, 12]
Neustart	3	Downlink	1 Byte
Chargensperrung	4	Downlink	22 Byte

Tabelle 44: Nachrichtentypen des Systems

³⁴ Transmission Control Protocol

³⁵ Inklusive 1 Byte für den Nachrichtentyp

Bestandsmutation

Nachrichtentyp: Bestandsmutation 1 Byte: 0x00
Mutationsnummer 4 Bytes
Änderung des Lagerbestands 2 Bytes
Zeitpunkt der Mutation 8 Bytes
Grund der Mutation 1 Bytes
Angemeldeter Benutzer 1 Bytes
GTIN 6 Bytes
Ablaufdatum 3 Bytes
Chargennummer 20 Bytes

Abbildung 45: Paket Bestandsmutation

Mutationsnummer (4 Bytes)

- Automatisch inkrementierende und eindeutige Laufnummer pro Lager

Änderung des Lagerbestands (2 Bytes)

- Positiv: Wurde dem Lager hinzugefügt
- Negativ: Wurde dem Lager entfernt

Zeitpunkt der Mutation (8 Bytes)

- Datum und Uhrzeit (Lokalzeit)
- Anzahl Sekunden seit dem 1. Januar 1970 (Unixzeit)

Grund der Mutation (1 Byte)

- 0 => für Kauf
- 1 => für Einlagerung
- 2 => für Rücknahme wegen Chargensperrung
- 3 => Rücknahme, weil bald abläuft
- 4 => Rücknahme, weil abgelaufen
- 5 => Kulanz
- 6 => Sonstiges

Angemeldeter Benutzer (1 Byte)

- 0 => Kunde
- 1 => Aussendienstmitarbeiter

GTIN (6 Bytes)

- GTIN des Barcodes

Ablaufdatum (3 Bytes)

- Ablaufdatum des Barcodes
- Zahl im Format «YMMDD»

Chargennummer (20 Bytes)

- Chargennummer des Barcodes

Heartbeat

Nachrichtentyp: Heartbeat 1 Byte: 0x01
Mutationsnummer 4 Bytes

Abbildung 46: Paket Heartbeat

Mutationsnummer (4 Bytes)

- Letzte verschickte Mutationsnummer

Mutationsanfrage

Nachrichtentyp: Mutationsanfrage 1 Byte: 0x02
Mutationsnummer 1 4 Bytes
...
Mutationsnummer n 4 Bytes

Abbildung 47: Paket Mutationsanfrage

Mutationsnummern (je 4 Bytes)

- Angefragte Mutationsnummern (maximal 12)

Neustart

Nachrichtentyp: Neustart 1 Byte: 0x03

Abbildung 48: Paket Neustart

Keine Nutzdaten (0 Byte)

Chargensperrung

Nachrichtentyp: Chargensperrung 1 Byte: 0x04
Rückruf 1 Byte
Chargennummer 20 Bytes

Abbildung 49: Paket Chargensperrung

Rückruf (1 Byte)

- 0 => Chargenrückruf wurde rückgängig gemacht
- 1 => Charge wurde zurückgerufen

Chargennummer (20 Bytes)

- Chargennummer des Barcodes

4.4.6. Datenkonvertierung

Die übermittelten Nutzdaten müssen für die Übertragung in Bytes umgewandelt werden. Wenn immer möglich, werden die einzelnen Nutzdaten in eine Ganzzahl umgewandelt. Ganzzahlen werden vorzeichenbehaftet und mit dem MSB zuerst codiert. Bei der Chargennummer (Alphanumerischer Wert) ist dies nicht möglich. Diese wird via ASCII-Encoding umgewandelt.

4.5. Lagerbestand

Um den Lagerbestand aller Konsignationslager zu betrachten, sind in der Lagerverwaltung sowie dem Lagerinterface Implementierungen vorhanden. Diese beziehen sich auf die folgende Anforderung:

- AF#101 – Lagerbestand anzeigen

Für jedes E-Konsignationslager kann der Bestand in der Lagerverwaltung überwacht werden. Im Tab «Bestand» des gewünschten Lagers werden die vorhandenen Artikel angezeigt.

Anz.	Produkt	GTIN	Min. Ablaufdatum	Chargen
6	Verankerungsschraube D6,5MM L16MM SW3,5	4046963977993	10.10.2020	16K12809, 19K12809
6	Sequent Please Neo PTCA-Katheter 3,0 x 15	4046963816803	10.10.2020	16K12809, 19K12809
6	BIOLOX DELTA PROTHESENKOPF 12/14 32MM L	4046963190477	10.10.2020	16K12809, 19K12809
6	BIOLOX DELTA EINSATZ C 28MM SYM.	4046963973186	15.05.2019	16K12809, 18K12809

Abbildung 50: Lagerbestand in der Lagerverwaltung

Vor Ort, im Konsignationslager, werden auf dem Lagerinterface unter «Bestand» die Artikel dargestellt.

Anz.	Name	GTIN	Min. Ablaufdatum
6	Verankerungsschraube D6,5MM L16MM SW3,5	4046963977993	10.10.2020
6	Sequent Please Neo PTCA-Katheter 3,0 x 15	4046963816803	10.10.2020
6	BIOLOX DELTA PROTHESENKOPF 12/14 32MM L	4046963190477	10.10.2020
6	BIOLOX DELTA EINSATZ C 28MM SYM.	4046963973186	10.07.2018

Abbildung 51: Lagerbestand im Lagerinterface

4.5.1. Ändern

Der Lagerbestand im Konsignationslager kann entweder mit dem Barcodescanner oder direkt auf der Anzeige geändert werden.

- AF#006 – Nachverfolgbarkeit
- AF#021 – Barcodescanner
- AF#022 – Produktidentifikation
- AF#102 – Produktkauf ohne Scannen

- 📄 AF#104 – Verfallsdatum überschritten beim Produktkauf
- 📄 AF#107 – Produktrücknahme ohne Scannen

Die Verwaltung des Lagerbestands im E-Konsignationslager ist die eigentliche Hauptfunktion des Lagerinterfaces. Produkte, welche im E-Konsignationslager vorhanden sind, müssen im Lagerinterface eingetragen werden. Bei der Herausnahme eines Produkts muss dieses wieder ausgetragen werden. Dies geschieht über den Barcode des Produkts, welches dieses identifiziert.

Betriebsmodi

Das Lagerinterface verfügt über mehrere Betriebsmodi, welche über die Ribbons geändert werden können.

- 📄 AF#111 – Betriebsmodi Lagerinterface
- 📄 AF#112 – Betriebsmodus «Produktkauf»
- 📄 AF#113 – Betriebsmodus «Produkteinlagerung»
- 📄 AF#114 – Betriebsmodus «Produktrücknahme»
- 📄 AF#115 – Betriebsmodus «Produktanzeige»

Die Betriebsmodi verändern die Funktion des Barcodescanners. Um einen nicht vorhandenen Artikel im Lagerface einzutragen, muss der Artikel im Betriebsmodus «Produkteinlagerung» gescannt werden. Dies führt intern dazu, dass der Lagerbestand um eins erhöht wird. Möchte man den Artikel wieder aus dem Lager entfernen, geschieht dies über einen der Betriebsmodi «Produktkauf» oder «Produktrücknahme». Sollte der Barcode nicht lesbar sein, kann auf der Anzeige der entsprechende Artikel ausgewählt und mit der Angabe des Ablaufdatums und der Chargennummer manuell entfernt werden. Um Informationen über ein Produkt zu erhalten, ohne es einzulagern oder zu entfernen, steht der Betriebsmodus «Produktanzeige» zur Verfügung, welcher beim Scannen Informationen zum Artikel auf dem Display anzeigt.

Das folgende Sequenzdiagramm zeigt den internen Ablauf vom Scannen eines Produkts auf und welche Pakete beteiligt sind. Der Benutzer scannt einen Artikel mit dem Barcodescanner. Dieser leitet den Event einerseits weiter an die Hauptapplikation, welches den Benutzer informiert, andererseits an die lokale Lagerverwaltung. Mit dem Barcode, dem Betriebsmodus und dem aktuellen Benutzer wird die Mutation gebildet und dem Softwarepaket «mutations» weitergeleitet. Diese passt den Lagerbestand an und legt die Mutation ab.

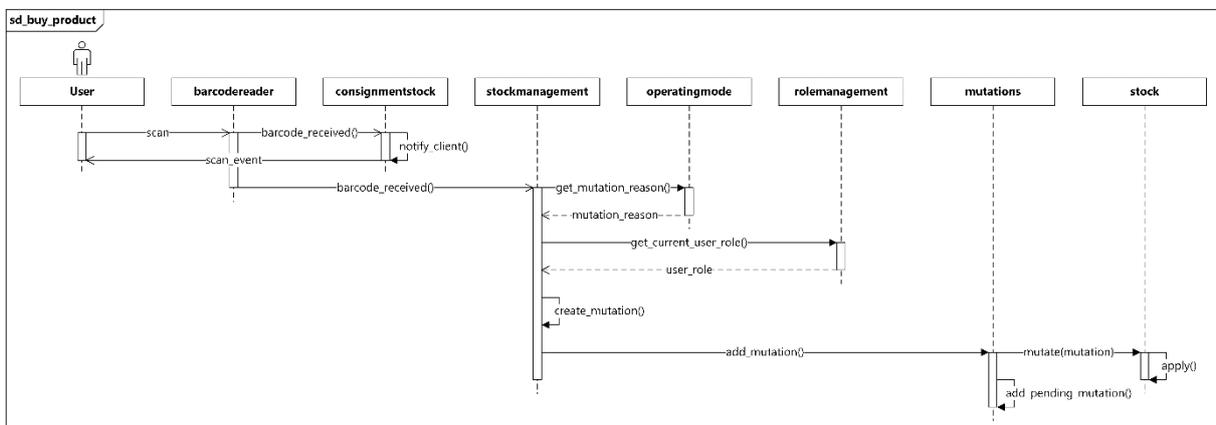


Abbildung 52: Ablauf beim Produktkauf

4.5.2. Mutationen übertragen

Änderungen des Lagerbestands im E-Konsignationslager müssen konsistent an die Lagerverwaltung weitergeleitet werden.

- 📄 AF#004 – Datenintegrität
- 📄 AF#014 – Umgang mit Kommunikationsunterbrüchen

Wie im Kapitel 4.1.1 Datenpersistenz beschrieben, sorgt das Lagerinterface dafür, dass die lokalen Mutationen konsistent gespeichert werden. Mutationen werden jedoch nicht direkt beim Scannen verschickt, sondern über einen Zeitraum gespeichert und erst dann verschickt. Dies hat den Vorteil, dass wenn z.B. mehrere Artikel mit demselben Barcode gekauft werden, diese einmalig übertragen werden können und nicht für jedes Produkt einzeln eine Nachricht gesendet werden muss. Voraussetzung dafür ist, dass der Barcode im selben Betriebsmodus und durch den selben Benutzer (siehe 4.6 Benutzerverwaltung) gescannt wurde, um die Nachverfolgbarkeit nicht zu verletzen (siehe AF#006 – Nachverfolgbarkeit). Wenn keine Verbindung mit dem LoRa-Netzwerk besteht, im Moment der geplanten Nachrichtenübertragung, dann werden diese auf dem Filesystem zwischengespeichert und beim nächsten Mal erneut übertragen.

Beim Senden einer Mutation gibt es zwei mögliche Ausgänge. Die Nachricht wird verschickt und via LoRaWAN-Protokoll bestätigt, dass diese angekommen ist oder die Nachricht kann nicht gesendet werden. Die Bestätigung ist jedoch nicht aussagekräftig, ob die Mutation effektiv in der Lagerverwaltung angekommen ist (siehe 4.4.2 Zuverlässigkeit). Deshalb bietet das Lagerinterface eine Schnittstelle an, auf der die Lagerverwaltung verlorene Mutationen anfragen kann. Das Lagerinterface sendet daraufhin die angefragten Mutationen an die Lagerverwaltung zurück. Das nachfolgende Sequenzdiagramm zeigt den Fall, dass eine Mutation nicht angekommen ist und die Lagerverwaltung diese nach einer Zeit automatisch nachfragt.

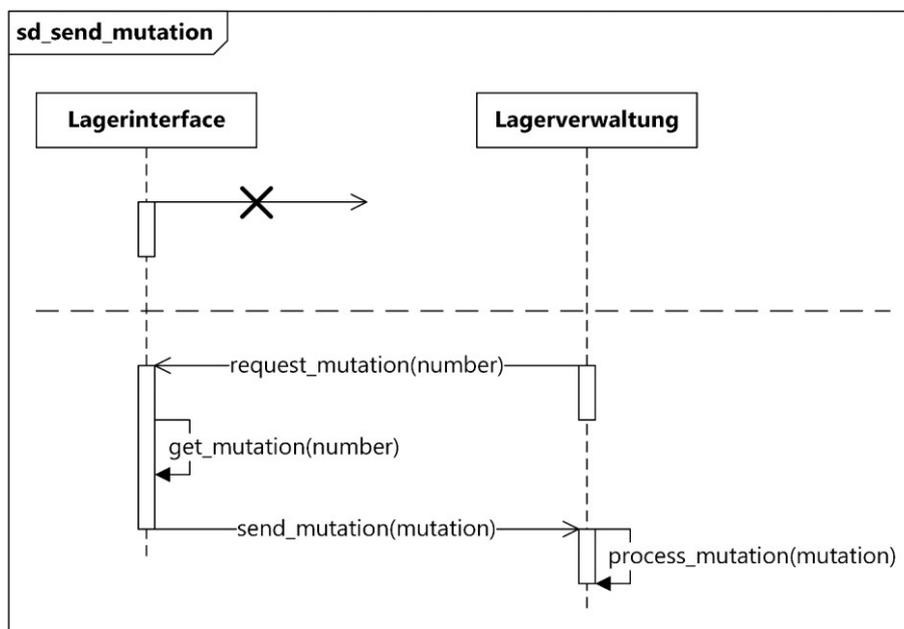


Abbildung 53: Verlorene Mutationen nachfragen

4.5.3. Überwachen

Um die Lagerbestände zu bewirtschaften, gibt es in der Lagerverwaltung eine zentrale Überwachung. Diese kann verwendet werden, um auf kleine Restbestände innerhalb der Lager hinzuweisen und die Aussendienstmitarbeiter auf eine Nachbestellung aufmerksam zu machen.

Mindestbestand erfassen

📄 AF#142 – Mindestbestand erfassen

Zu Produkten, welche im Lagerbestand vorhanden sind, können Mindestbestände und Nachbestellmengen erfasst werden. Diese Mindestbestände werden gespeichert und bei eingehenden Mutationen geprüft (siehe Benachrichtigung).

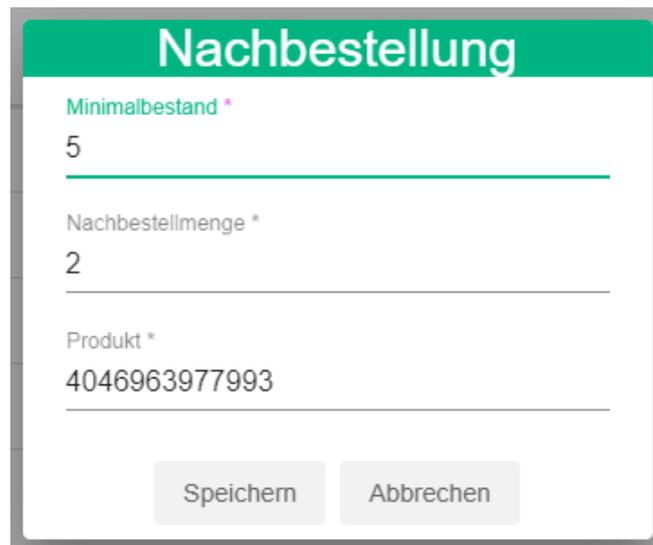
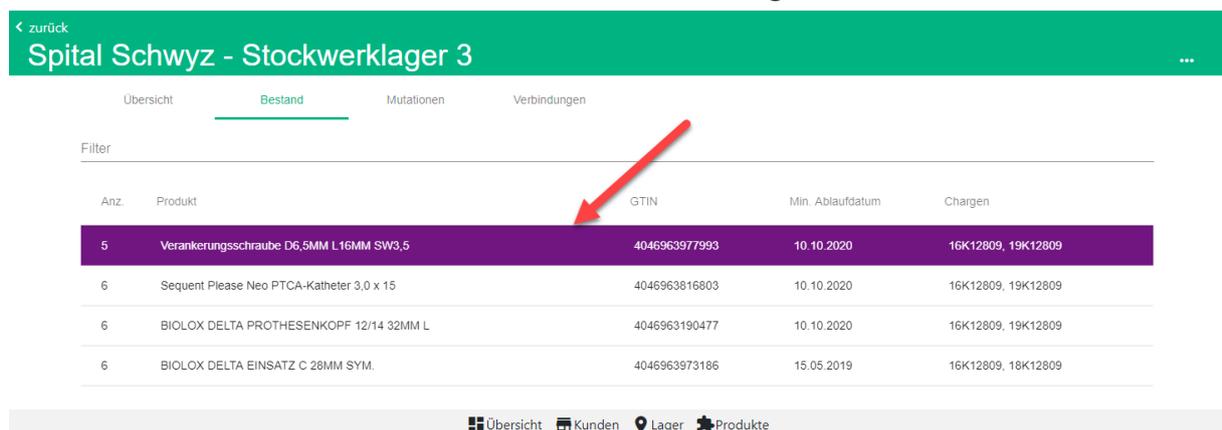


Abbildung 54: Nachbestellung für ein Produkt im E-Konsignationslager festlegen

Falls ein Mindestbestand erreicht ist, wird dies im Bestand gekennzeichnet.



Anz.	Produkt	GTIN	Min. Ablaufdatum	Chargen
5	Verankerungsschraube D6,5MM L16MM SW3,5	4046963977993	10.10.2020	16K12809, 19K12809
6	Sequent Please Neo PTCA-Katheter 3,0 x 15	4046963816803	10.10.2020	16K12809, 19K12809
6	BIOLOX DELTA PROTHESENKOPF 12/14 32MM L	4046963190477	10.10.2020	16K12809, 19K12809
6	BIOLOX DELTA EINSATZ C 28MM SYM.	4046963973186	15.05.2019	16K12809, 18K12809

Abbildung 55: Bestand eines E-Konsignationslagers mit erreichtem Mindestbestand für ein Produkt

Benachrichtigung

- 📄 AF#141 – Benachrichtigungssystem
- 📄 AF#143 – Mindestbestandsbenachrichtigung

Beim Erhalt einer Mutation wird überprüft, ob der Mindestbestand erreicht wurde. Bei Erreichung wird eine E-Mail zum, im Lager hinterlegten, Verantwortlichen gesendet.



Abbildung 56: Erhaltene E-Mail bei Erreichung des Mindestbestands

Falls ein Mindestbestand erfasst ist, jedoch keine Benachrichtigung erfolgen soll, muss die Nachbestellmenge auf 0 gestellt sein.

4.6. Benutzerverwaltung

Im Lagerinterface dürfen gewisse Aktionen oder Informationen nur von autorisiertem Personal durchgeführt, beziehungsweise angezeigt werden. Um dies zu lösen, wurde eine Benutzerverwaltung implementiert, welche die Privilegien des Benutzers überprüft.

- 📄 AF#005 – Benutzerverwaltung

Die Benutzerverwaltung unterscheidet zwischen den Rollen Kunde (Standard Benutzer) und Aussendienstmitarbeiter (Privilegiertes Benutzer). Beim Starten des Lagerinterfaces ist standardmässig immer der Standard Benutzer aktiv. Der Aussendienstmitarbeiter kann sich über einen Pin-Code anmelden, um Zugriff auf erweiterte Funktionen zu erhalten.

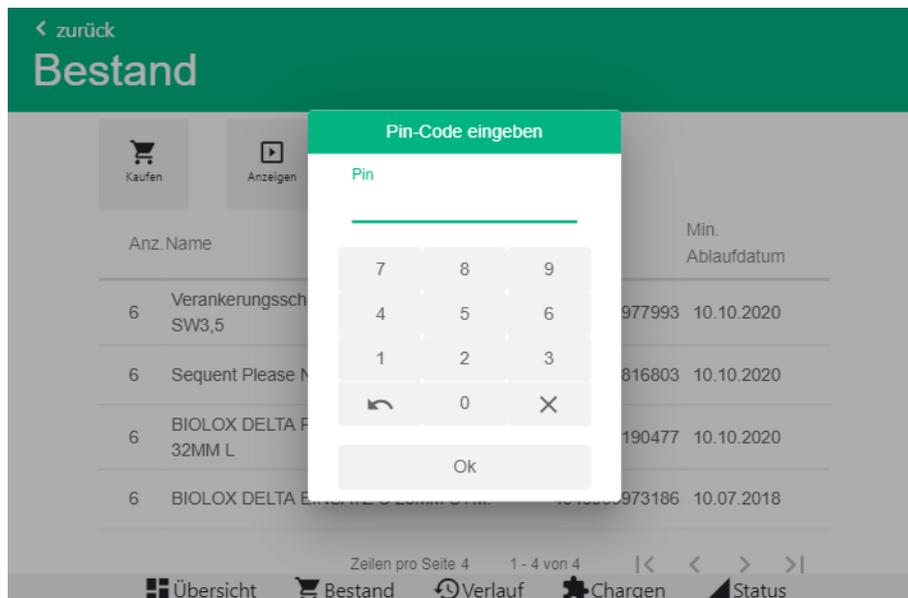


Abbildung 57: Login für Aussendienstmitarbeiter

4.6.1. Privilegien

Welche Rolle was darf und was nicht, ist im Kapitel 3.2.5 Zuweisung der Anforderungen zu entnehmen. Für Anforderungen, welche nur von einer Rolle durchgeführt werden darf, wurde

ein Privileg erstellt, welches beim Ausführen überprüft wird. Anforderungen, welche für beide Rollen gelten, werden nicht überprüft.

In der folgenden Tabelle werden die Privilegien für die zwei Benutzer aufgezeigt.

Privileg	Standard Benutzer	Privilegierter Benutzer
Artikelstamm abgleichen	×	✓
Produkte kaufen	✓	✓
Produkte einlagern	✓	✓
Produkte zurücknehmen	×	✓
Produktinformationen anzeigen	✓	✓
Einfacher Systemstatus anzeigen	✓	×
Erweiterter Systemstatus anzeigen	×	✓

Tabelle 45: Benutzerprivilegien

4.7. Betrieb E-Konsignationslager

Um die E-Konsignationslager zu betreiben, sind einige Mittel umgesetzt worden. Diese unterstützen die Technischen Mitarbeiter sowie die Aussendienstmitarbeiter. Es wurden diverse Anforderungen im Rahmen des Betriebs berücksichtigt und umgesetzt.

AF#106 – Lagerinterface vorkonfigurieren

Mittels einer Übersicht kann sich ein Mitarbeiter der B. Braun direkt die ersten Informationen zum Zustand aller E-Konsignationslager einholen. Neue Lagerinterfaces, welche zum ersten Mal eine Verbindung zur Lagerverwaltung aufbauen, werden automatisch erfasst und in die Übersicht eingebunden.

Bezeichnung	Kunde	Gebäude	Stockwerk	Standort	DevEUI	Vrb. Status
Kantonsspital Uri - OP1	Kantonsspital Uri	Hauptgebäude	5. OG	Im Gang, linke Seite vor OP1	70b3d54993738b5a	✓
Spital Schwyz - Stockwerklager 3	Spital Schwyz	4. Trakt	3. Stock	Vor Zimmer 312	70b3d5499e3421a0	✓

Abbildung 58: Übersicht aller E-Konsignationslager, welche verwaltet werden

In der Detailansicht (Abbildung 59) jedes Lagers können die Eigenschaften der Lager bearbeitet werden, um beispielsweise den Standort anzupassen, damit es leichter auffindbar ist.

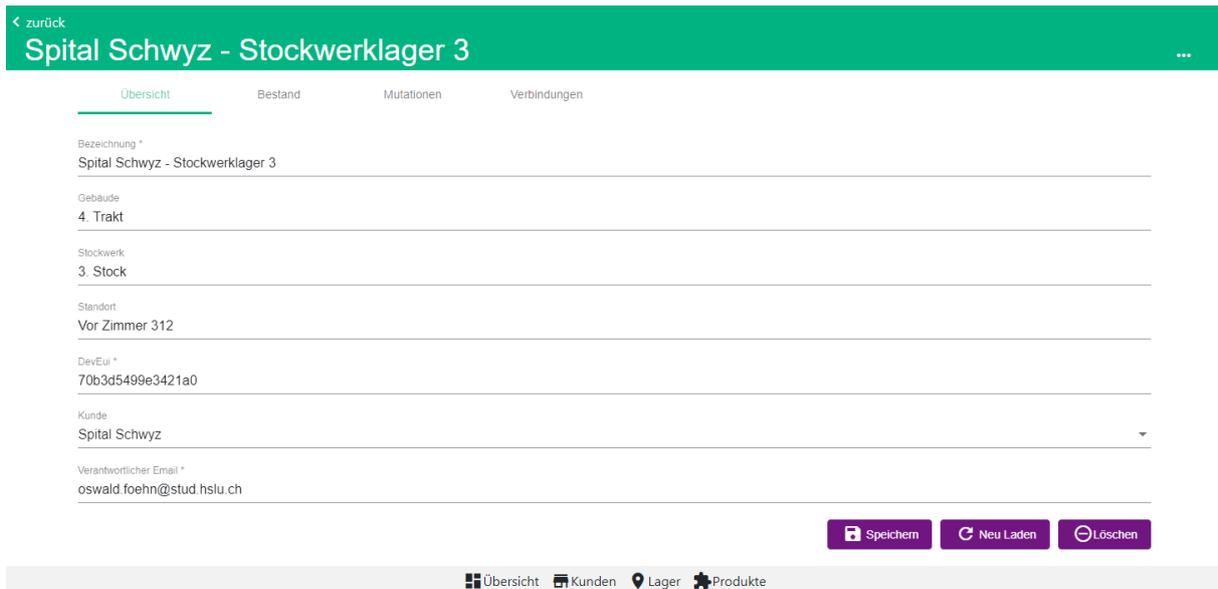


Abbildung 59: Eigenschaften eines E-Konsignationslager

4.7.1. Verbindungen

- 📄 AF#013 – Verbindungsqualität
- 📄 AF#131 – Regelmässige Verbindungskontrolle
- 📄 AF#132 – Manuelle Verbindungskontrolle

Bei der Auflistung der E-Konsignationslager in der Lagerverwaltung wird zusätzlich eine Aussage über die Verbindungsqualität des E-Konsignationslagers gemacht. Die folgenden Status-Bilder lassen den Rückschluss auf die Verbindungsqualität zu.

Vrb. Status	Name	Beschreibung
	OK	In den letzten 24 Stunden gab es mindestens eine Verbindung.
	Warnung	In den letzten zwei Tagen gab es mindestens eine Verbindung.
	Fehler	In den letzten zwei Tagen gab es keine Verbindung.

Tabelle 46: Verbindungsstatus der E-Konsignationslager

Damit die Lagerverwaltung diese Übersicht bilden kann, muss diese regelmässig durch das Lagerinterface benachrichtigt werden. Dies geschieht beim Starten des Geräts und danach automatisch in regelmässigen Zeitintervallen. Beim Senden dieser Heartbeat-Nachricht wird zusätzlich die letzte Mutationsnummer mitgeschickt, dies erlaubt der Lagerverwaltung zu überprüfen, ob ihre Lagerbestände noch konsistent sind (siehe 4.5.2 Mutationen übertragen).

4.7.2. Systemstatus

Der Benutzer soll vor Ort eine Übersicht des Status erhalten können. Diese Übersicht soll bei einer allfälligen Fehlersuche eine zentrale Rolle einnehmen. Die folgenden Anforderungen wurden entsprechend definiert und umgesetzt.

- 📄 AF#133 – Vereinfachter Systemstatus anzeigen
- 📄 AF#134 – Erweiterter Systemstatus anzeigen

Im Tab «Verbindungen» werden alle Verbindungen vom Lager zur Verwaltung aufgezeigt. Dadurch kann ein Aussendienstmitarbeiter den Verbindungsstatus prüfen, in dem er zum

Beispiel die Regelmässigkeit der Heartbeats prüft. Zusätzlich zur Auflistung kann pro Verbindung die komplette Meldung der Swiscom angezeigt werden.

The screenshot shows a mobile application interface for 'Spital Schwyz - Stockwerklager 3'. The top navigation bar is green with a back arrow and the title. Below the bar, there are four tabs: 'Übersicht', 'Bestand', 'Mutationen', and 'Verbindungen'. The 'Verbindungen' tab is active, displaying a table of connection records. To the right of the table, a detailed view of a heartbeat message is shown in a scrollable area.

ID	Verarbeitet	Abgesetzt	Nachrichtentyp
34	10.05.2019 10:45:56	10.05.2019 10:45:56	Heartbeat
33	10.05.2019 10:05:02	10.05.2019 10:05:00	Mutation
32	10.05.2019 09:51:11	10.05.2019 09:51:10	Mutation
31	10.05.2019 09:51:09	10.05.2019 09:51:09	Mutation
30	10.05.2019 09:51:02	10.05.2019 09:51:02	Mutation
29	10.05.2019 09:51:00	10.05.2019 09:51:00	Mutation
28	10.05.2019 09:50:56	10.05.2019 09:50:56	Mutation
27	10.05.2019 09:50:52	10.05.2019 09:50:52	Mutation
26	10.05.2019 09:50:48	10.05.2019 09:50:47	Mutation
25	10.05.2019 09:50:44	10.05.2019 09:50:43	Mutation

```

{
  "DevEUI_Luplink": {
    "ADRBit": 0,
    "Channel": "L69",
    "CustomerData": {
      "air": {
        "prot": "LORA(Generic)",
        "ver": "11"
      },
      "type": null
    },
    "CustomerID": "100010519",
    "DevAddr": "08593CD4",
    "DevEUI": "70B3D5499E3421A0",
    "DevLnCh": 1,
    "FCntDn": 0,
    "FCntUp": 0,
    "FFPort": 2,
    "InstanceID": 0,
    "LSeq": 0,
    "LSeqID": "00000401",
    "LSeqLAT": 47.003328,
    "LSeqLON": 8.813342,
    "LSeqRSSI": -90,
    "LSeqSNR": -3,
    "LSeqID": "0B030318",
    "LSeq": {
      "Chain": 0,
      "LSeqSP": -94.764351,
      "LSeqRSSI": -90,
      "LSeqSNR": -3,
      "LSeqID": "0B030318"
    }
  }
}
  
```

Abbildung 60: Anzeige der Meldungen eines E-Konsignationslagers

Detaillierte Informationen zum Zustand des E-Konsignationslagers lassen sich vor Ort anzeigen. Für den Kunden gibt es dazu eine vereinfachte Anzeige, welche nur die nötigsten Informationen anzeigt.

The screenshot shows a mobile application interface for 'Systemstatus'. The top navigation bar is green with a back arrow and the title. Below the bar, there are five icons: 'Kaufen', 'Anzeigen', 'Einlagern', 'Rücknahme', and 'Abmelden'. The 'Einlagern' icon is highlighted in purple. Below the icons, there are three sections: 'LoRa', 'Barcode Leser', and 'Benutzer', each with a table of information.

LoRa	
Verbunden	Ja
Adresse	70B3D5499E3421A0
Letzte Verbindung	10:40 23.05.19

Barcode Leser	
Zuletzt gelesen	09:44 23.05.19
GTIN	4046964107863

Benutzer	
Aktueller Benutzer	Aussendienst

Abbildung 61: Vereinfachter Systemstatus

Der Aussendienstmitarbeiter kann auf einen erweiterten Systemstatus (Abbildung 62) zugreifen, welcher ihm zusätzliche Informationen anzeigt. So kann er sich beispielweise die letzten Ergebnisse der Verbindungskontrollen, sowie Informationen zum letzten Barcode Scan Event anzeigen lassen.

```

Erweitert
{
  "BarcodeReader": {
    "Device name": "Datalogic Scanning, Inc. Point of Sale Handable Scanner
- /dev/input/event1",
    "LastBarcode": {
      "expiry": "2022-02-01",
      "gtin": 4046964107863,
      "lot_id": "000COROFLEXISAR20X09"
    },
    "LastRead": "2019-05-23T09:44:10.637292"
  },
  "Connection": {
    "ConnectionHistory": {
      "2019-05-23T09:40:39.776664": true
    },
    "CurrentIp": "192.168.1.15",
    "LastConnection": "2019-05-23T10:40:40.405297"
  },
  "LoRa": {
    "Connected": true,
    "DeviceAddress": "70B3D5499E3421A0",
    "History": []
  },
  "PlannedRestart": "",
  "User": {
    "LastLogin": "2019-05-23T10:47:15.403182",
    "SignedIn": "PRIVILEGED_ROLE"
  }
}

```

Abbildung 62: Erweiterter Systemstatus

Automatischer Neustart

AF#135 – Automatischer Neustart

Das Lagerinterface überwacht intern, ob eine Verbindung mit der Lagerverwaltung besteht. Dazu wertet es die Rückmeldungen der regelmässigen Verbindungskontrolle aus. Wenn über einen bestimmten Zeitraum keine Nachrichten abgesetzt werden können, initiiert das Lagerinterface einen automatischen Neustart, welcher in der Nacht durchgeführt wird. Dies aus dem Grund, da erwartungsgemäss das E-Konsignationslager in der Nacht weniger verwendet wird und dies den Kunden weniger beeinträchtigt.

Fern Neustart

AF#136 – Fern Neustart

Falls ein Kunde ein Problem mit einem Lagerinterface meldet oder ein Aussendienstmitarbeiter Probleme erkennt, kann das Lagerinterface direkt aus der Lagerverwaltung neu gestartet werden.

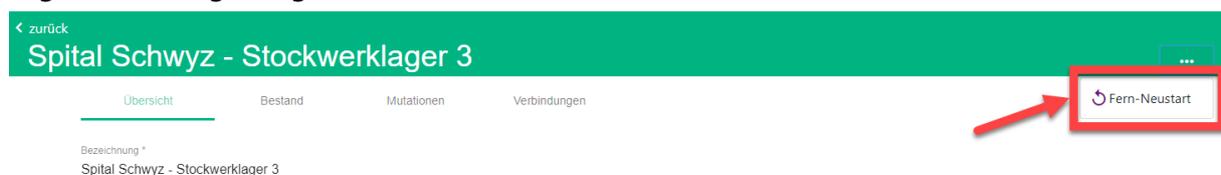


Abbildung 63: Aktion Fern-Neustart eines E-Konsignationslagers

Bei diesem Vorgang wird via dem LPN eine Nachricht an das Lagerinterface abgesetzt, um den Neustart durchzuführen. Falls das Problem des Lagerinterfaces an der LPN Verbindung liegt, bleibt nur ein manueller Neustart am Gerät übrig.

4.8. Artikelstamm

Um die Produkte der B. Braun Medical AG in der Lagerverwaltung zu administrieren, werden Möglichkeiten zur einfachen Verwaltung bereitgestellt. Die folgenden Anforderungen betreffen die Implementation dieses Teils.

- 📄 AF#031 – Artikelstamm führen
- 📄 AF#032 – Artikelname führen
- 📄 AF#033 – Artikelgrafik führen
- 📄 AF#105 – Artikelstamm abgleichen

Produkte sollen in der Benutzeroberfläche bearbeitet werden können. Mittels einer Tabelle werden alle Produkte dargestellt. Falls Produkte gesucht werden, kann der Freitext-Filter verwendet werden.

GTIN	Name	Beschreibung	Kategorie
0	Produkt nicht in Artikelstamm	Produkt nicht in Artikelstamm	Produkt nicht in Artikelstamm
4038653352258	TIBIA-VERSCHLUSSSCHRAUBE D14MM		
4046963165512	ENDURO FEMURKOMPONENTE ZEMENTIERT F1L	Gekoppeltes Knieendoprothesensystem	Prothese
4046963165529	ENDURO FEMURKOMPONENTE ZEMENTIERT F2L	Gekoppeltes Knieendoprothesensystem	Prothese
4046963165536	ENDURO FEMURKOMPONENTE ZEMENTIERT F3L	Gekoppeltes Knieendoprothesensystem	Prothese
4046963165550	ENDURO FEMURKOMPONENTE ZEMENTIERT F1R	Gekoppeltes Knieendoprothesensystem	Prothese
4046963165567	ENDURO FEMURKOMPONENTE ZEMENTIERT F2R	Gekoppeltes Knieendoprothesensystem	Prothese
4046963165574	ENDURO FEMURKOMPONENTE ZEMENTIERT F3R	Gekoppeltes Knieendoprothesensystem	Prothese
4046963165581	ENDURO TIBIAKOMP.OFFSET ZEMENTIERT T1	Gekoppeltes Knieendoprothesensystem	Prothese

Abbildung 64: Übersicht über alle Produkte der E-Konsignationslager

Zum Bearbeiten eines Produktes, kann mittels Selektion die Detail-Ansicht (Abbildung 65) geöffnet werden. Auf dieser Seite können alle Eigenschaften (ausser der GTIN) bearbeitet werden. Falls ein Bild hinzugefügt oder geändert werden soll, kann mittels des «+» Buttons ein Bild hochgeladen werden.

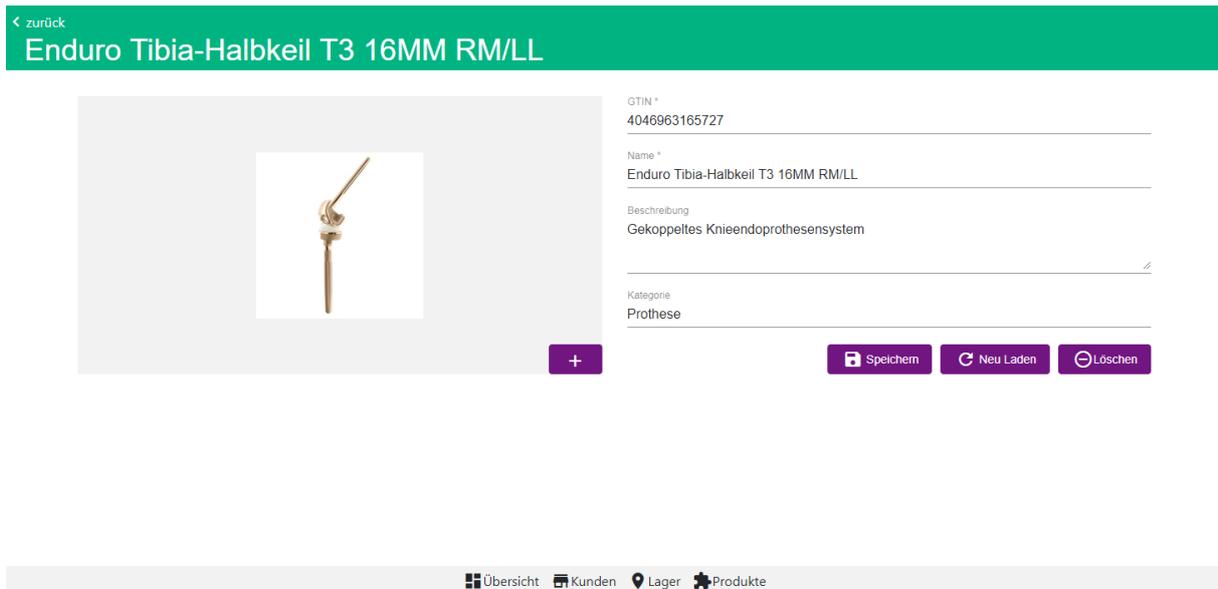


Abbildung 65: Detail-Ansicht eines Produkts

4.8.1. Importieren des Artikelstamms

Um neue Produkte zu erfassen, ist eine Schnittstelle erstellt worden, in welcher mittels CSV-Datei viele Produkte in einer Operation importiert werden können.

Das CSV muss folgende Spalten beinhalten (Reihenfolge der Spalten nicht zwingend). Falls zusätzliche Spalten in der Datei vorhanden sind, werden diese ignoriert.

- GTIN
- Artikelbezeichnung
- Beschreibung



Abbildung 66: Import von neuen Produkten mittels CSV

4.8.2. Artikelstamm synchronisieren

Um Produkt-Eigenschaften mit den Lagerinterfaces zu synchronisieren, muss die Lagerverwaltung eine Schnittstelle bereitstellen, damit via HTTP-Anfrage der komplette Stamm abgefragt werden kann. Dies wird mittels der OData Schnittstelle zur Verfügung gestellt. Mit folgender URL können die Ressourcen abgefragt werden.

<http://bbraunba20191-1.westeurope.cloudapp.azure.com/api/Products>

Der privilegierte Benutzer kann, sofern er mit dem Internet verbunden ist, eine Synchronisierung starten. Das Lagerinterface parsst die Daten des Artikelstamms und wandelt sie ins korrekte Format um.

4.9. Chargen

- 📄 AF#121 – Chargen anzeigen
- 📄 AF#122 – Chargen markieren
- 📄 AF#123 – Gesperrte Chargen synchronisieren
- 📄 AF#124 – Chargenrückruf ausstehend beim Produktkauf

In der Lagerverwaltung kann eine Übersicht über alle Chargen angezeigt werden, welche in allen E-Konsignationslagern eines Kunden vorhanden sind, um beispielsweise im Falle eines Chargenrückrufs zu eruieren, welche E-Konsignationslager von einer Rückrufaktion betroffen sind. Falls eine Charge bereits zurückgerufen wurde und trotzdem Artikel in den Lagern vorhanden sind, wird dies entsprechend gekennzeichnet.

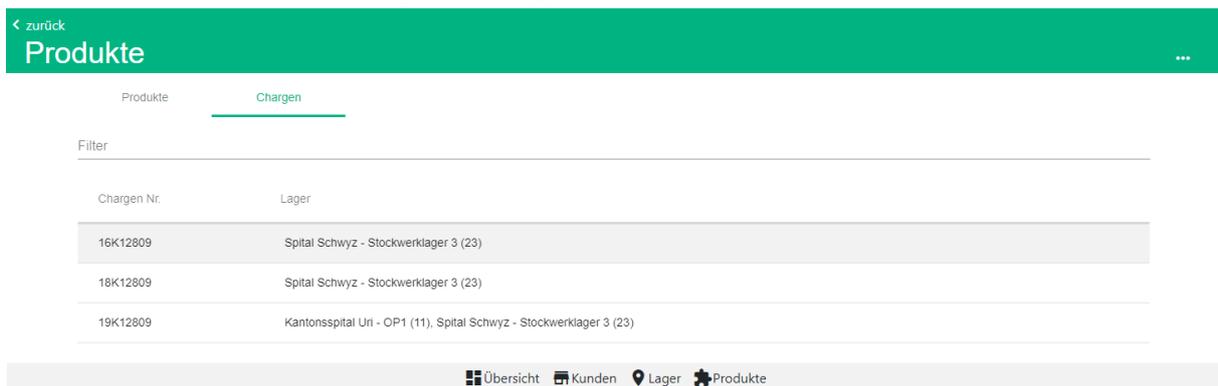


Abbildung 67: Sich im Umlauf befindende Chargen

Wird nun eine Charge zurückgerufen, kann diese in der Lagerverwaltung selektiert und entsprechend markiert werden. Bei einer Markierung kann zusätzlich eine Bemerkung erfasst werden, in welcher ein Grund für einen Rückruf vermerkt werden kann.

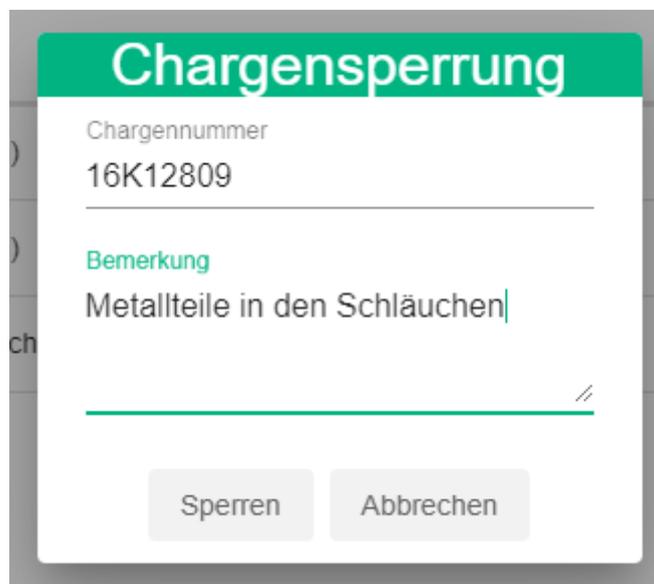


Abbildung 68: Sperrung einer Charge

Bei der Sperrung einer Charge werden alle Lagerinterfaces mit Artikeln dieser Charge benachrichtigt und melden dies bei einem allfälligen Bezug des Artikels. Das nachfolgende Sequenzdiagramm zeigt detailliert die Sperrung einer Charge, welche auf zwei Lagerinterfaces vorhanden ist (Lagerinterface X & Z).

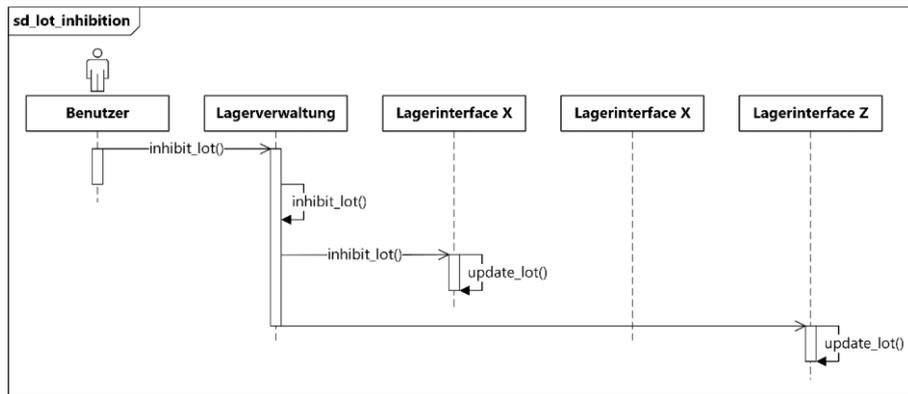


Abbildung 69: Verteilung einer Chargensperrung

4.10. Kunden

Es gibt eine einfache Kundenverwaltung, in welcher Kunden erstellt, angezeigt, bearbeitet und gelöscht werden können.

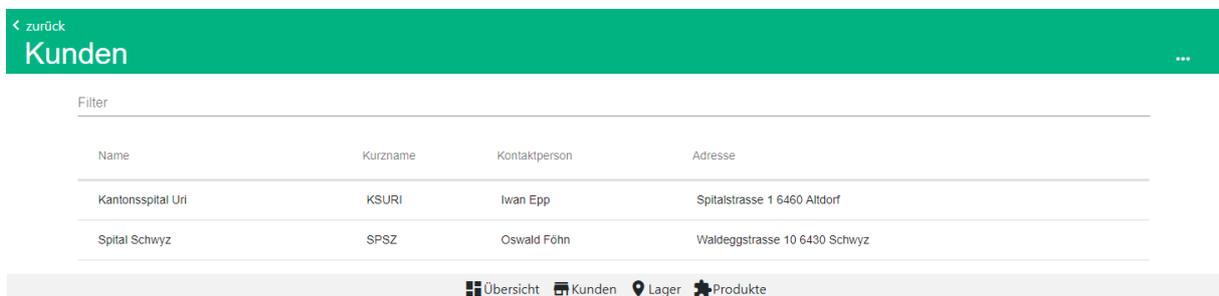


Abbildung 70: Übersicht über alle Kunden mit E-Konsignationslager

Ein neuer Kunde kann erstellt werden, indem in der Kunden-Übersicht die Aktion «Neu» gewählt wird. Bei dieser Aktion wird der Kunde erstellt und auf die Detail-Ansicht navigiert.

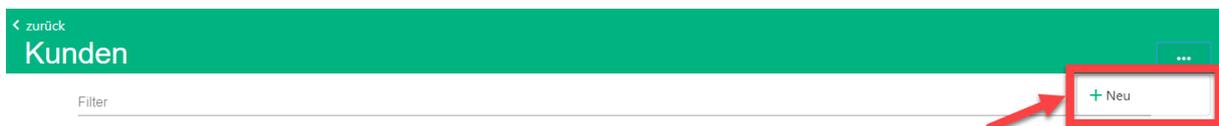


Abbildung 71: Neuen Kunden erstellen mittels Aktion

Um Kunden zu bearbeiten, kann mittels Selektion in der Tabelle auf die Eigenschaft des Kunden navigiert werden, in welcher alle Eigenschaften des Kunden bearbeitet werden können.



Abbildung 72: Detail-Ansicht eines Kunden

5. Inbetriebnahme

Um das Gesamtsystem bereitzustellen, muss zuerst die Lagerverwaltung in Betrieb genommen werden. Als nächstes muss die Kommunikation mit den E-Konsignationslager mittels einer Registrierung der Lagerverwaltung im Swisscom Device Manager initialisiert werden. Der letzte Schritt ist die Installation der Lagerinterfaces inklusive der Einbindung in das Swisscom LPN.

5.1. Lagerverwaltung

Um die Lagerverwaltung zu installieren, sind folgende Schritte nötig:

- Grundinstallation Betriebssystem, Datenbank-Server
- Installation IIS (Internet Information Service)
- Konfiguration Frontend/Backend
- Deploy der Applikation

Die detaillierte Anleitung befindet sich im Anhang 3: Inbetriebnahme Lagerverwaltung.

5.2. Swisscom LPN

Die API der Lagerverwaltung muss bei der Swisscom hinterlegt werden, damit Meldungen der Lagerinterfaces der Lagerverwaltung mitgeteilt werden.

Genauere Beschreibungen befinden sich im Anhang 4: Inbetriebnahme Swisscom LPN.

5.3. Lagerinterface

Damit Lagerinterfaces in den Konsignationslagern in Betrieb genommen werden können, braucht es einige Schritte als Vorbereitung:

- Hardware Installation
- Grundinstallation des Raspberry Pi inkl.
 - Python 3.7
 - Apache
- Deploy der Applikationsteile
 - LoPy 4
 - Backend (inkl. Konfiguration)
 - Frontend
- Registrierung im Swisscom Device Manager
- Konfiguration es Lagerinterfaces in der Lagerverwaltung

Eine Beschreibung aller Teilschritte befindet sich im Anhang 5: Inbetriebnahme Lagerinterface.

6. Validierung

Um die Funktionalität des Systems sicherzustellen, wurden in jeder Phase des Projekts Validierungen durchgeführt. Diese Validierungen helfen auch bei Änderungen im Projekt die Funktionalität der bereits abgeschlossenen Teilsysteme weiterhin sicherzustellen.

6.1. Tests während der Entwicklung

Während der gesamten Projektdauer, insbesondere bei der Implementierung, wurden Funktionalitäten mittels Unit- und Integrationstests überprüft. Dadurch wird die Richtigkeit der Teilfunktionen sichergestellt. Die Überprüfung kann jederzeit wiederholt werden. Auf eine Integration in eine CI/CD Buildchain wird verzichtet, da pro Teilsystem jeweils ein Entwickler für die korrekte Umsetzung verantwortlich ist. Dadurch wurde bezweckt, dass die Verantwortung pro Teilsystem zugeteilt ist. Mittels Ausführung der automatischen Tests (z.B. vor dem Check-In) kann der Entwickler die Richtigkeit sicherstellen und jederzeit neu überprüfen.

Um die Tests durchzuführen, wurden die bestehenden Komponenten der Entwicklungsumgebungen verwendet.

Eine Auswertung der automatischen Tests beim Teilsystem «Backend Lagerinterface» ergab folgende Resultate (siehe Abbildung 73: Messung der Codeabdeckung des Backend des Lagerinterfaces). Für das Paket «consignmentstock» wurden keine automatisierten Tests erstellt, da das Softwarepaket nur die Webschnittstelle zur Verfügung stellt und für die Funktionalität auf Funktionen anderer Softwarepakete zugreift.

Element	Statistics, %
.idea	
locastra.barcode reader	55% files, 58% lines covered
locastra.consignmentstock	0% files, not covered
locastra.lopylora	95% files, 98% lines covered
locastra.mutations	88% files, 100% lines covered
locastra.operatingmode	85% files, 100% lines covered
locastra.products	88% files, 100% lines covered
locastra.rolemanagement	90% files, 100% lines covered
locastra.stock	83% files, 100% lines covered
locastra.stockcore	95% files, 99% lines covered
locastra.stockmanagement	96% files, 99% lines covered
locastra.systemstate	94% files, 100% lines covered

Abbildung 73: Messung der Codeabdeckung des Backend des Lagerinterfaces

Die Auswertung der automatischen Tests beim Teilsystem «Backend Lagerverwaltung» zeigt, dass die Abdeckung insgesamt zwar weniger hoch ist, diese jedoch im zentralen Bereich (Business-Layer) erhöht ist. Zusätzlich wurde bei der C#-Implementierung das Tool «SonarLint» verwendet, welches mittels statischer Code-Analysen eine weitere Qualitätsmessung aufstellt.

Hierarchy	Covered (% Blocks)	Covered (Blocks)	Not Covered (% Blocks)	Not Covered (Blocks)
	84.51%	2466	15.49%	452
consignmentstockdata.dll	82.09%	165	17.91%	36
consignmentstockcore.dll	47.14%	33	52.86%	37
consignmentstockbltest.dll	98.28%	741	1.72%	13
consignmentstockbl.dll	89.91%	936	10.09%	105
consignmentstockapitest.dll	95.71%	424	4.29%	19
consignmentstockapi.dll	40.83%	167	59.17%	242

Abbildung 74: Messung der Codeabdeckung des Backend der Lagerverwaltung

6.2. Definition von realen Bedingungen

Gemäss der Aufgabenstellung (siehe Anhang 1: Aufgabenstellung) muss der entwickelte Prototyp unter realen Bedingungen getestet werden. Bezüglich der Umgebung des Standorts des Lagerinterfaces wurden folgende Annahmen getroffen.

Reale Bedingungen implizieren:

- Keine Internetverbindung
- Zugriff nur über Benutzeroberfläche

Explizit ausgeschlossene Bedingungen (in Absprache mit dem Auftraggeber):

- Keine LoRa Verbindung am Standort
- Hardware muss nicht nach Spitalstandards gereinigt werden können.

Diese Bedingungen wurden beim Abschlusstest berücksichtigt. Es kann damit sichergestellt werden, dass das System unter diesen realen Bedingungen einen Test erfolgreich besteht und unter diesen Bedingungen produktiv eingesetzt werden kann.

6.3. Überprüfung der Anwendungsfälle

Nach der Erarbeitung der Anforderungen wurden die Anwendungsfälle erneut durchgespielt. Jeder Schritt, der in den Anwendungsfällen festgelegt wurde, musste mit einer Anforderung abgedeckt sein. Durch die Validierung der Anforderungen wird die Richtigkeit der Anwendungsfälle impliziert.

6.4. Überprüfung der Anforderungen

Das Vorgehen der Überprüfung der Anforderungen wurde wie folgt gewählt. Zuerst wurde die Funktionalität durch beide Studierende gemeinsam sichergestellt. Danach wurde mittels einer Demonstration vor dem Kunden die Korrektheit überprüft. Falls beide Testreihen erfolgreich durchschritten sind, werden die Anforderungen in der Tabelle (siehe Anhang 2: Anforderungsliste) als «abgenommen» gekennzeichnet. Zusätzlich wurden die Anforderungen im Rahmen der «Gesamt-Test des Systems» zusätzlich in ihrer Funktionalität getestet.

Im Rahmen der Bachelor-Diplomarbeit konnten alle Anforderungen, auch Optionale, implementiert und durch den Kunden abgenommen werden.

6.5. Gesamt-Test des Systems

Nach internem Abschluss aller Anforderungen wurde das Gesamtsystem getestet. Ein spezielles Augenmerk galt dabei den Anwendungsfällen, da es für den Benutzer wichtig ist, diese durchführen zu können.

Zusätzlich wird ein Langzeittest durchgeführt, welcher die Funktionalität des Systems über einen längeren Zeitpunkt überprüfen soll (siehe Anhang 6: Langzeittest). Die folgenden Punkte werden im Langzeittest abgedeckt.

- Testdauer 7 Tage
- Mindestens 15 Barcodescans pro Tag
- 1 Chargenrückruf durch Aussendienstmitarbeiter
- Unterbrüche der LoRa-Verbindung
- Unerwartete Stromausfälle

7. Evaluation

Aufgrund der Validierung, insbesondere des Langzeit-Tests (siehe Kapitel 6.5 Gesamt-Test des Systems) wurden Aussagen getroffen, welche für eine Evaluation des Systems einen wertvollen Beitrag leisten.

7.1. Stärken

7.1.1. Bedienung

Die Bedienung des Lagerinterfaces ist simpel und intuitiv gehalten. Es gibt wenig Bedienmöglichkeiten für den Kunden, was einerseits zu weniger Fehlbedienungen führt und andererseits den Schulungsaufwand für die Kunden minimiert.

7.1.2. Inbetriebnahme

Eine Erstinbetriebnahme kann fast komplett intern durchgeführt werden. Die einzigen Punkte die vor Ort gemacht werden müssen, ist die manuelle Verbindungskontrolle und die erstmalige Lagerbestandserfassung.

7.1.3. Abdeckung

Durch die Verwendung des Swisscom LoRaWAN kann eine grosse Abdeckung erreicht werden. Über dieses Netz können 96.8% der Bevölkerung erreicht werden (Swisscom [1], 2019). Das E-Konsignationslager funktioniert jedoch nur in diesem abgedeckten Bereich.

7.1.4. Lagerbewirtschaftung

Mit Hilfe der Lagerverwaltung wird der Prozess der Lagerbewirtschaftung grundsätzlich verbessert. Die Aussendienstmitarbeiter können sich einfach und schnell einen Überblick über ihre Kunden und Lager verschaffen. Es besteht die Möglichkeit das System zu erweitern und Schritt für Schritt den kompletten Prozess mehr und mehr zu optimieren. Ein Beispiel dafür wäre die automatische Nachbestellung, welche momentan nur mit einer E-Mail-Benachrichtigung umgesetzt ist.

7.1.5. Robustheit

Die Robustheit des Systems konnte besonders im Langzeittest genauer untersucht werden. Es zeigt sich das Gesamtsystem bei Strom- und Verbindungsunterbrüchen selbständig wieder erholt und Datenbestände abgleicht («eventual consistency»).

7.2. Schwächen

7.2.1. Anzeige

Um den gesamten Lagerbestand im E-Konsignationslager anzuzeigen, ist die Anzeige zu klein. Der Ansatz, den Lagerbestand mittels Paging anzuzeigen, löst das Problem nicht abschliessend. Eine grössere Anzeige, eventuell sogar mit Suchfunktion vor Ort, würde das Spitalpersonal essenziell unterstützen.

7.2.2. Wartbarkeit

Da die E-Konsignationslager mit dem Internet verbunden sind, müssen Software-Aktualisierungen und detaillierte Fehleranalysen vor Ort durchgeführt werden. Diesem Problem kann man, sofern der Lösungsansatz gleichbleibt, nur mit der Erhöhung der Robustheit des Systems entgegenwirken. Die Störungsbehebung vor Ort ihrerseits können mit der jetzigen Lösung nur mit dem entsprechenden Know-how durchgeführt werden (siehe Kapitel 3.3 Technologien und 4.1.5 Konfiguration für Fehleranalyse).

7.3. Limitierungen

7.3.1. Verbindung

Es können nicht mehr als acht Mutationen im selben Abarbeitungszyklus übertragen werden. Wenn mehr Mutationen übertragen werden, kann es sein, dass diese nicht übermittelt werden können und diese somit erst im nächsten Zyklus verschickt werden. Es können maximal 30 Mutationen pro Minute durchgeführt werden. Dieser Wert wurde hochgerechnet von ungefähr zwei Sekunden pro Übertragung (siehe Anhang 6: Langzeittest). Dieser Wert gilt nur, sofern durch den «Connectivity Plan» der Swisscom keine weiteren Einschränkungen vorliegen (Swisscom [III], 2019).

7.3.2. Abhängigkeit von der Swisscom

Die Swisscom ist momentan einer der grössten Provider für IoT Lösungen in der Schweiz. Dies birgt eine grosse Abhängigkeit zur Swisscom und ihrer LPN Lösung. Gerade im Bereich Sicherheit ist seitens der Swisscom noch Optimierungspotential vorhanden, da die Downlink API öffentlich und ohne Authentifizierung zugänglich ist (siehe auch 4.4.3 Datensicherheit). Betreffend Skalierbarkeit und Wartbarkeit bietet die Swisscom jedoch schon einige Funktionalität an, zum Beispiel das Konfigurieren mehrerer Applikationsserver oder die «DX Api», welche jedoch im Rahmen dieser Arbeit nicht untersucht wurden (Swisscom [III], 2019).

8. Abschluss und Ausblick

Durch die Erstellung dieses Systems wurde der B. Braun Medical AG eine Grundlage geschaffen, mit welcher sie nun den produktiven Einsatz in ihren Konsignationslagern eruieren können. Geplant ist, die Prototypen in der Niederlassung Sempach den Mitarbeitern zum Testen zur Verfügung zu stellen. Durch diese Tests sollen wertvolle Rückmeldungen gewonnen werden.

Zusätzlich müssen für einen produktiven Einsatz der E-Konsignationslager weitere Umsetzungen bezüglich der Sicherheit gemacht werden. Beispielsweise darf die Lagerverwaltung nicht ohne weiteres aus dem Internet verwendet werden können. Eine weitere Sicherheitsmassnahme sollte sein, dass der Versand einer Meldung an ein Lagerinterface nur von einer authentifizierten Quelle aus geschehen darf (dies muss mit der Swisscom geklärt werden).

Durch den «heissen Draht» mit unserem Auftraggeber war es möglich, schnell auf Änderungen zu reagieren. Dabei konnten wir bei Änderungsvorschlägen rasch und agil Änderungen an den Anforderungen vornehmen und für eine kurze Sichtkontrolle mittels eines Videoanrufs oder eines Telefonats Kontakt aufnehmen. Dies hat uns erlaubt, effizient zu arbeiten.

8.1. Lessons Learned

Expertenmeinungen einholen: Aus dem Wirtschaftsprojekt hatten wir LoRa Probleme mitgenommen. Unsere Komponente/Software hat nur teilweise funktioniert. Nach zwei E-Mails mit der Swisscom haben wir eine neue Hardware-Komponente inkl. robuster LoRaWAN Implementierung gefunden. Das hätte uns früher einfallen können und hätte uns einige mühsame Arbeitstage erspart.

Weniger Planung bedeutet mehr Zeit, um zu arbeiten: Wir haben uns am Anfang einen groben Plan gemacht, wann wir welchen Teil der Anforderungen umgesetzt haben wollten. Dies erlaubte es uns, schnell mit der Implementierung zu beginnen. Zusätzlich konnten wir agil auf Änderungen reagieren und uns schnell anpassen, ohne dass wir den ganzen Plan erneut anpassen mussten.

Bessere Kenntnisse der Programmiersprache steigert die Effizienz exponentiell: Im Rahmen des Projekts benutzten wir drei Programmiersprachen. Die Implementierungen in C# gingen sehr schnell, dank der Erfahrung, die wir bereits hatten. Bei Python und Angular 7 sah dies schon bedeutend anders aus. Während der Implementierung lernten wir neue Sprachfeatures kennen, welche uns das Leben von Beginn an erleichtert hätten. Es war spannend zu sehen, wie sich der Code während der Projektarbeit verändert hat, insbesondere in der Qualität.

Mittel der Dokumentation: Während des gesamten Projekts haben wir Microsoft Word eingesetzt, um Dokumente zu erstellen. Wir mussten feststellen, dass es scheinbar nicht möglich ist, Dokumente effizient zu bearbeiten, insbesondere wenn wir gemeinsam an selben Dokumenten arbeiten wollten. Zusätzlich ist die Formatierung sehr zeitintensiv und fehleranfällig. In weiteren Projekten werden wir nach einem alternativen Tool für die Dokumentation suchen müssen.

9. Anhänge

9.1. Anhang 1: Aufgabenstellung

 **01_Anhänge/01_Aufgabenstellung IoT Konsignationslagermanagement.pdf**

9.2. Anhang 2: Anforderungsliste

 **01_Anhänge/02_Anforderungsliste.pdf**

9.3. Anhang 3: Inbetriebnahme Lagerverwaltung

 **01_Anhänge/03_Inbetriebnahme Lagerverwaltung.pdf**

9.4. Anhang 4: Inbetriebnahme Swisscom LPN

 **01_Anhänge/04_Inbetriebnahme Swisscom LPN.pdf**

9.5. Anhang 5: Inbetriebnahme Lagerinterface

 **01_Anhänge/05_Inbetriebnahme Lagerinterface.pdf**

9.6. Anhang 6: Langzeittest

 **01_Anhänge/06_Langzeittest.pdf**

10. Verzeichnisse

10.1. Abbildungsverzeichnis

Abbildung 1: Projektorganisation.....	2
Abbildung 2: Rahmenplan.....	3
Abbildung 3: Auswertung Arbeitsjournal.....	6
Abbildung 4: Aufteilung des Gesamtsystems.....	8
Abbildung 5: Anwendungsfälle Kundenbetrieb.....	11
Abbildung 6: Anwendungsfälle Technischer Betrieb.....	11
Abbildung 7: Technologie Stack des gesamten Systems.....	28
Abbildung 8: LoRa Übersicht (LoRa Alliance™ [1], 2019).....	29
Abbildung 9: Systemübersicht.....	30
Abbildung 10: Softwareeinsatz und -verteilung.....	31
Abbildung 11: Übersicht Systemaufbau Lagerinterface.....	31
Abbildung 12: Verwendetes LoRa Modul (LoPy).....	32
Abbildung 13: Übersicht Hard- und Software Lagerverwaltung.....	33
Abbildung 14: Lösungskonzept Hosting der Lagerverwaltung.....	33
Abbildung 15: Aufbau des Protokolls.....	34
Abbildung 16: Aufbau des system-eigenen Protokolls.....	34
Abbildung 17: Farben im Design des B. Braun Brand Guides (B. Braun AG, 2019).....	35
Abbildung 18: Layout des B. Braun Brand Guides (B. Braun AG, 2019).....	35
Abbildung 19: Layout einer Seite mit Infoboxen und Inhalt (B. Braun AG, 2019).....	36
Abbildung 20: Kopfzeilendefinition des B. Braun Brand Guides (B. Braun AG, 2019).....	36
Abbildung 21: Definition Fusszeile des B. Braun Brand Guides (B. Braun AG, 2019).....	36
Abbildung 22: Definition Tabbar des B. Braun Brand Guides (B. Braun AG, 2019).....	37
Abbildung 23: Definition Button des B. Braun Brand Guides (B. Braun AG, 2019).....	37
Abbildung 24: Grundlegender Seitenaufbau der Lagerverwaltung.....	38
Abbildung 25: Darstellung des Betriebsmodus im Basis-Layout des Lagerinterfaces.....	39
Abbildung 26: Designelement Kachel.....	39
Abbildung 27: Designelement Tabbar.....	39
Abbildung 28: Designelement Detail-Edit mit Bilder Upload.....	40
Abbildung 29: Designelement Tabelle mit Filter.....	40
Abbildung 30: Designelement Tabelle ohne Filter, mit Paging.....	40
Abbildung 31: Designelement Dialog.....	41
Abbildung 32: Designelement Snackbar.....	41
Abbildung 33: E-Konsignationslager.....	42
Abbildung 34: Benutzer-Interface der Lagerverwaltung.....	42
Abbildung 35: Paket-Diagramm des Lagerinterface.....	43
Abbildung 36: Daten über LoRa verschicken.....	50
Abbildung 37: Lebenszyklus Daten senden.....	51
Abbildung 38: Daten über LoRa empfangen.....	51
Abbildung 39: Empfangene Daten vom LoPy lesen.....	52
Abbildung 40: Status abfragen.....	52
Abbildung 41: Paket-Diagramm der Lagerverwaltung.....	56
Abbildung 42: Entitäten-Diagramm Lagerverwaltung.....	57
Abbildung 43 Konfiguration Lagerverwaltung.....	60
Abbildung 44: Paket-Diagramm des Frontend.....	61
Abbildung 45: Paket Bestandsmutation.....	63
Abbildung 46: Paket Heartbeat.....	64
Abbildung 47: Paket Mutationsanfrage.....	64
Abbildung 48: Paket Neustart.....	64
Abbildung 49: Paket Chargensperrung.....	64

Abbildung 50: Lagerbestand in der Lagerverwaltung	65
Abbildung 51: Lagerbestand im Lagerinterface.....	65
Abbildung 52: Ablauf beim Produktkauf.....	66
Abbildung 53: Verlorene Mutationen nachfragen.....	67
Abbildung 54: Nachbestellung für ein Produkt im E-Konsignationslager festlegen	68
Abbildung 55: Bestand eines E-Konsignationslagers mit erreichtem Mindestbestand für ein Produkt ..	68
Abbildung 56: Erhaltene E-Mail bei Erreichung des Mindestbestands	69
Abbildung 57: Login für Aussendienstmitarbeiter.....	69
Abbildung 58: Übersicht aller E-Konsignationslager, welche verwaltet werden	70
Abbildung 59: Eigenschaften eines E-Konsignationslager	71
Abbildung 60: Anzeige der Meldungen eines E-Konsignationslager	72
Abbildung 61: Vereinfachter Systemstatus.....	72
Abbildung 62: Erweiterter Systemstatus.....	73
Abbildung 63: Aktion Fern-Neustart eines E-Konsignationslagers.....	73
Abbildung 64: Übersicht über alle Produkte der E-Konsignationslager	74
Abbildung 65: Detail-Ansicht eines Produkts.....	75
Abbildung 66: Import von neuen Produkten mittels CSV	75
Abbildung 67: Sich im Umlauf befindende Chargen	76
Abbildung 68: Sperrung einer Charge.....	76
Abbildung 69: Verteilung einer Chargensperrung	77
Abbildung 70: Übersicht über alle Kunden mit E-Konsignationslager	77
Abbildung 71: Neuen Kunden erstellen mittels Aktion	77
Abbildung 72: Detail-Ansicht eines Kunden	77
Abbildung 73: Messung der Codeabdeckung des Backend des Lagerinterfaces	79
Abbildung 74: Messung der Codeabdeckung des Backend der Lagerverwaltung.....	79

10.2. Tabellenverzeichnis

Tabelle 1: Glossar	V
Tabelle 2: Aufteilung der Arbeiten	2
Tabelle 3: Meilensteine	3
Tabelle 4: Risiken im Projekt.....	4
Tabelle 5: Risikomatrix	4
Tabelle 6: Massnahmen zur Risikoverminderung.....	5
Tabelle 7: Kostenauswertung	6
Tabelle 8: Unterstützende Tools während der Projektarbeit	7
Tabelle 9: Beschreibung Akteur Kunde	9
Tabelle 10: Beschreibung Akteur Aussendienstmitarbeiter	9
Tabelle 11: Beschreibung Akteur Technischer Mitarbeiter	10
Tabelle 12: Anwendungsfall «Waren kaufen».....	12
Tabelle 13: Anwendungsfall «Waren zurücknehmen»	13
Tabelle 14: Anwendungsfall «Waren einlagern»	14
Tabelle 15: Anwendungsfall «Lagerbestand anzeigen».....	15
Tabelle 16: Anwendungsfall «Lagerbestand korrigieren».....	16
Tabelle 17: Anwendungsfall «Chargenrückruf durchführen»	17
Tabelle 18: Anwendungsfall «Lagerinterface konfigurieren»	18
Tabelle 19: Anwendungsfall «E-Konsignationslager in Betrieb nehmen»	19
Tabelle 20: Anwendungsfall «E-Konsignationslager fernüberwachen»	20
Tabelle 21: Anwendungsfall «Lagerstörung erkennen & analysieren»	21
Tabelle 22: Zuweisung der Anforderungen zu den Teilsystemen.....	27
Tabelle 23: Nachrichten	34
Tabelle 24: Teile des Basis-Layouts des Frontend.....	38
Tabelle 25: Beschreibung der Pakete der Lagerverwaltung.....	43
Tabelle 26: Verwendete Blueprints des Lagerinterface Backend	45

Tabelle 27: Historie der Bestandsmutationen.....	46
Tabelle 28: Informationen zum Lagerbestand.....	47
Tabelle 29: Zugriff zu den Betriebsmodi.....	47
Tabelle 30: Zugriff zur Benutzerverwaltung.....	48
Tabelle 31: Benachrichtigung zu ausstehenden Neustarts.....	48
Tabelle 32: Benachrichtigung beim Scannen eines Produkts.....	49
Tabelle 33: Informationen zum Systemstatus.....	50
Tabelle 34: Einstellungen Standard Sektion.....	54
Tabelle 35: Einstellungen Barcodescanner Sektion.....	54
Tabelle 36: Einstellungen Lora Sektion.....	54
Tabelle 37: Einstellungen Systemstatus Sektion.....	55
Tabelle 38: Beschreibung der Pakete der Lagerverwaltung.....	56
Tabelle 39: Beschreibung der Entitäten der Lagerverwaltung.....	57
Tabelle 40: Beschreibung der HTTP-Methoden bei der OData API.....	58
Tabelle 41: Beschreibung der Query-Operationen der OData API.....	58
Tabelle 42: Entitäten der API Schnittstelle der Lagerverwaltung.....	59
Tabelle 43: Beschreibung der Pakete des Frontend.....	61
Tabelle 44: Nachrichtentypen des Systems.....	62
Tabelle 45: Benutzerprivilegien.....	70
Tabelle 46: Verbindungsstatus der E-Konsignationslager.....	71

11. Literaturverzeichnis

- B. Braun AG. (3. Mai 2019). *B. Braun Brand Guide*. Abgerufen am 3. Mai 2019 von B. Braun Brand Guide: <https://www.bbraun-brand.com/bbraun-brand-guide>
- Epp, I., & Föhn, O. (2018). *WIPRO: B. Braun is in love with LoRa / IoT Konsignationslagermanagement*. Rotkreuz. Abgerufen am 23. Mai 2019
- GS1 Switzerland. (1. Mai 2019). *Application Identifier*. Abgerufen am 1. Mai 2019 von GS1 Application Identifier: <https://www.gs1.ch/home/themen/gs1-standards/capture/application-identifiers>
- LoRa Alliance™ [I]. (2. Mai 2019). *LoRaWAN™ Specification v1.0 | LoRa Alliance™*. Abgerufen am 2. Mai 2019 von LoRaWAN™ Specification v1.0: <https://lora-alliance.org/resource-hub/lorawantm-specification-v10>
- LoRa Alliance™ [II]. (2. Mai 2019). *LoRaWAN™ Regional Parameters v1.0 | LoRa Alliance™*. Abgerufen am 2. Mai 2019 von LoRaWAN™ Regional Parameters v1.0: <https://lora-alliance.org/resource-hub/lorawantm-regional-parameters-v10>
- Pycom [I]. (2. Mai 2019). *Atom - Pycom*. Abgerufen am 2. Mai 2019 von Atom: <https://docs.pycom.io/pymakr/installation/atom.html>
- Pycom [II]. (2. Mai 2019). *Introduction to MicroPython - Pycom*. Abgerufen am 2. Mai 2019 von Introduction to MicroPython: <https://docs.pycom.io/gettingstarted/programming/micropython.html>
- Pycom [III]. (2. Mai 2019). *LoPy 4 - GitBook*. Abgerufen am 2. Mai 2019 von LoPy 4: <https://docs.pycom.io/gettingstarted/connection/lopy4.html#third>
- Swisscom [I]. (28. Februar 2019). *Unsere Abdeckung - Swisscom*. Abgerufen am 28. Februar 2019 von Swisscom: <http://lpn.swisscom.ch/d/unsere-abdeckung/>
- Swisscom [II]. (28. Februar 2019). *Unser Angebot - Swisscom*. Abgerufen am 28. Februar 2019 von Swisscom: <http://lpn.swisscom.ch/d/unser-angebot/>
- Swisscom [III]. (5. Mai 2019). *Developer Guide for CMP LPN - Swisscom*. Abgerufen am 5. Mai 2019 von Developer Guide for CMP LPN: https://developer.lpn.swisscom.ch/sites/default/files/2019-04/Swisscom_LPN%20Portal_V2.0_DeveloperGuide_0.pdf