

HOCHSCHULE LUZERN

BACHELORARBEIT

# JTI Pickup Station

*Oliver Werlen*

Betreut durch Rene Meier  
xz

22. März 2021

Page

## Bachelorarbeit an der Hochschule Luzern – Informatik

**Titel:** JTI Pickup Station

**Studentin/Student:** Oliver Werlen

**Studentin/Student:**

**Studiengang:** BSc Informatik

**Jahr:** 2021

**Betreuungsperson:**

**Expertin/Experte:**

**Auftraggeberin/Auftraggeber:**

**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 Merkblatt «Studentische Arbeiten» auf MyCampus) respektieren werden.

Ort / Datum, Unterschrift \_\_\_\_\_

Ort / Datum, Unterschrift \_\_\_\_\_

**Abgabe der Arbeit auf der Portfolio Datenbank:**

Bestätigungsvisum Studentin/Student

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**

xxx

**Ausschliesslich bei Abgabe in gedruckter Form:  
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.

## **Zusammenfassung**

## Inhaltsverzeichnis

<b>1 Problem und Vision</b>	<b>4</b>
1.1 Problem . . . . .	4
1.1.1 Versandkosten/Mindestbestellwert . . . . .	4
1.1.2 Dauer bis Ware bei Endkonsumenten . . . . .	4
1.1.3 Angebot nur in begrenztem Zeitraum möglich . . . . .	4
1.2 Vision . . . . .	4
<b>2 Stand der Technik</b>	<b>5</b>
2.1 MyPost 24 . . . . .	5
2.1.1 Allgemein . . . . .	5
2.2 avec . . . . .	6
2.2.1 avec now . . . . .	6
2.2.2 avec Box . . . . .	6
2.2.3 Ablauf Tabakkauf . . . . .	7
2.3 Starbucks Progressive Web App (PWA) . . . . .	8
2.3.1 Standortsuche . . . . .	8
2.3.2 Fazit . . . . .	9
2.4 Fazit . . . . .	9
<b>3 Ideen und Konzepte</b>	<b>10</b>
3.1 Systemarchitektur . . . . .	10
3.2 Frameworks . . . . .	11
3.2.1 Spring Boot . . . . .	11
3.2.2 Angular . . . . .	11
3.3 weitere Technologien . . . . .	11
3.3.1 Docker . . . . .	11
3.3.2 MariaDB . . . . .	11
<b>4 Methoden</b>	<b>13</b>
4.1 SoDa . . . . .	13
4.2 CI/CD . . . . .	13
<b>5 Realisierung</b>	<b>14</b>
5.1 Initialisierungsphase . . . . .	14
5.1.1 Kick-Off Meeting . . . . .	14
5.1.2 Erstellen des Projektmanagementplans . . . . .	14
5.1.3 Problem und Vision . . . . .	14
5.1.4 Requirement Engineering . . . . .	14
5.1.5 Stand der Technik . . . . .	14
5.1.6 Meilenstein Abschluss Initialisierungsphase . . . . .	14
5.2 Konzeptionsphase . . . . .	15
5.2.1 Sprint 1 . . . . .	15
5.2.2 Sprintreview Sprint 1 . . . . .	16
5.2.3 Sprint 2 . . . . .	17
5.2.4 Sprintreview Sprint 2 . . . . .	21
5.2.5 Sprint 3 . . . . .	21
<b>6 Evaluation und Validation</b>	<b>22</b>
<b>7 Ausblick</b>	<b>23</b>
<b>8 Verzeichnisse</b>	<b>24</b>

<b>A Testprotokolle</b>	<b>30</b>
A.1 Testprotokolle Schülermodus . . . . .	30
<b>B Projektmanagementplan</b>	<b>31</b>
B.1 Projektorganisation . . . . .	31
B.1.1 Organisationsplan, Rollen, Zuständigkeiten . . . . .	31
B.1.2 Projektstrukturplan . . . . .	32
B.2 Projektführung . . . . .	33
B.2.1 Rahmenplan . . . . .	33
B.2.2 Meilensteine . . . . .	34
B.2.3 Risikomanagement . . . . .	36
B.2.4 Definition of done . . . . .	38
B.3 Projektunterstützung . . . . .	39
B.3.1 Tools für Entwicklung, Test und Abnahme . . . . .	39
B.3.2 Konfigurationsmanagement . . . . .	39
B.4 Teststrategie und Drehbuch . . . . .	40
B.4.1 Teststrategie . . . . .	40
B.4.2 Testdrehbuch . . . . .	40
B.5 Bemerkungen . . . . .	40
<b>C System-Spezifikation</b>	<b>41</b>
C.1 Systemübersicht . . . . .	41
C.1.1 Systemarchitektur . . . . .	41
C.1.2 Kontextdiagramm . . . . .	42
C.2 Architektur und Designentscheide . . . . .	43
C.2.1 Modelle und Sichten . . . . .	43
C.2.2 Daten (Mengengerüst und Strukturen) . . . . .	43
C.2.3 Entwurfsentscheide . . . . .	43
C.3 Schnittstellen . . . . .	45
C.3.1 Externe Schnittstellen . . . . .	45
C.3.2 Wichtige interne Schnittstellen . . . . .	45
C.3.3 Benutzerschnittstellen . . . . .	45
C.4 Environment-Anforderungen . . . . .	45
C.4.1 Hardware . . . . .	45
C.4.2 Software . . . . .	45
<b>D Software Requirements Specification</b>	<b>46</b>
D.1 Zweck . . . . .	46
D.1.1 Zielgruppe . . . . .	46
D.1.2 Produktumfang . . . . .	46
D.1.3 Definitionen . . . . .	46
D.1.4 Systemübersicht . . . . .	46
D.1.5 Abhängigkeiten . . . . .	46
D.2 Spezifische Anforderungen . . . . .	47
D.2.1 Funktionale Anforderungen . . . . .	47
D.2.2 Nicht funktionale Anforderungen . . . . .	47
D.3 Bemerkungen . . . . .	48
D.4 Unterschriften . . . . .	48
<b>E Sitzungsprotokolle</b>	<b>49</b>
E.1 23.02.2021 . . . . .	49
E.1.1 Ordnungsaufruf . . . . .	49
E.1.2 Teilnehmer . . . . .	49
E.1.3 Genehmigung des Protokolls . . . . .	49
E.1.4 Ankündigungen . . . . .	49

E.1.5	besprochene Punkte . . . . .	49
E.1.6	Tagesordnung der nächsten Sitzung . . . . .	50
E.1.7	Unterschriften . . . . .	50
E.2	04.03.2021 . . . . .	51
E.2.1	Ordnungsauftrag . . . . .	51
E.2.2	Teilnehmer . . . . .	51
E.2.3	Genehmigung des Protokolls . . . . .	51
E.2.4	Ankündigungen . . . . .	51
E.2.5	besprochene Punkte . . . . .	51
E.2.6	Tagesordnung der nächsten Sitzung . . . . .	52
E.2.7	Unterschriften . . . . .	52
E.3	11.03.2021 . . . . .	52
E.3.1	Ordnungsauftrag . . . . .	52
E.3.2	Teilnehmer . . . . .	52
E.3.3	Genehmigung des Protokolls . . . . .	52
E.3.4	Ankündigungen . . . . .	52
E.3.5	besprochene Punkte . . . . .	52
E.3.6	Tagesordnung der nächsten Sitzung . . . . .	53
E.3.7	Unterschriften . . . . .	53
E.4	18.03.2021 . . . . .	53
E.4.1	Ordnungsauftrag . . . . .	53
E.4.2	Teilnehmer . . . . .	53
E.4.3	Genehmigung des Protokolls . . . . .	53
E.4.4	Ankündigungen . . . . .	53
E.4.5	besprochene Punkte . . . . .	54
E.4.6	Tagesordnung der nächsten Sitzung . . . . .	54
E.4.7	Unterschriften . . . . .	54
<b>F</b>	<b>Rahmenpläne</b>	<b>55</b>
<b>G</b>	<b>Originale Aufgabenstellung</b>	<b>56</b>
<b>H</b>	<b>Wireframes mit Balsamiq Mockup</b>	<b>56</b>
<b>I</b>	<b>API Documentation mit Swagger</b>	<b>56</b>

# 1 Problem und Vision

## 1.1 Problem

Der Onlinekauf ist beim Zigarettenkauf ein sehr selten genutzter Absatzweg. Dabei sind vor allem die nachfolgenden Punkte verantwortlich für die seltene Nutzung dieses Angebots.

- Versandkosten/Mindestbestellwert
- Dauer bis Ware beim Endkonsumenten
- Angebot nur in begrenztem Zeitraum möglich

### 1.1.1 Versandkosten/Mindestbestellwert

Bei diversen Onlineshops kommen bei zu geringer Bestellmenge erhebliche Versandkosten hinzu. So kostet der Versand per Paket in der Regel 9 Franken. Bei Kioskolino ist der Versand ab einem Bestellwert von 139.- Fr. Portofrei. [Genf, o.D.] Bei Coop ist die Liefergebühr sogar noch höher. Sie beträgt 17.90 Fr. Die Versandkosten nehmen mit zunehmendem Bestellwert ab. Ab 500.- Fr. ist der Versand kostenlos. [Coop, 2020]

### 1.1.2 Dauer bis Ware bei Endkonsumenten

Bei der Bestellung bei Kioskolino ist die Ware innerhalb von 1-3 Werktagen beim Konsumenten. [Genf, o.D.] Für die meisten Kunden dauert dies zu lange. Coop verspricht die Lieferung am selben Tag. Dazu können bei der Bestellung verschiedene Zeitfenster ausgewählt werden, die Verfügbarkeit ist dabei von der Region abhängig. Die Ware muss aber frühzeitig bestellt werden, um die Lieferung am gleichen Tag garantieren zu können. Zudem bietet Coop auch die Möglichkeit, die Produkte direkt in der Filiale abzuholen. [Coop, 2020]

Das Problem ist aber mit beiden Anbieter identisch. Es muss die Ware sehr früh bestellt werden. Zudem dauert die Lieferung immer zwischen 4 Stunden bis zu 3 Tagen. Eine Lieferung am Sonntag ist dabei nicht möglich, Samstags wird nur am Nachmittag geliefert.

### 1.1.3 Angebot nur in begrenztem Zeitraum möglich

Die bestellte Ware wird nur zu bestimmten Zeiten ausgeliefert. So ist eine Lieferung an Sonn- und Feiertagen nicht möglich.

## 1.2 Vision

Durch die JTI Pick-Up Station ist es dem Kunden möglich, seine Ware bequem im Onlineshop zu bestellen und anschliessend direkt und ohne Wartezeit an der gewünschten Pick-Up Station abzuholen.

Die Artikel werden durch den Kunden an der gewählten Pick-Up Station bereitgestellt. Durch das Vorzeigen der Bestellbestätigung durch den Kunden wird der Artikel freigegeben und steht zur Abholung bereit.

Ein Mindestbestellwert muss nicht erreicht werden. Zudem werden keine zusätzlichen Gebühren verlangt.

Die Applikation soll dabei durch eine einfache und intuitive Bedienung eine optimale Benutzerexperience bieten. Die Bestellung soll schnell und einfach ablaufen. Die Abholung soll anschliessend in kurzer Zeit abgewickelt werden. Durch die Umsetzung als PWA ist die Applikation auch ohne aktive Internetverbindung nutzbar.

Durch das Verwenden eines bereits etablierten Altersverifikationsanbieters können zudem die rechtlichen Bedingungen erfüllt werden. Der Bezahlvorgang wird ebenfalls durch einen etablierten Anbieter durchgeführt. Dies garantiert eine sichere und zuverlässige Bezahlabwicklung.

## 2 Stand der Technik

### 2.1 MyPost 24

#### 2.1.1 Allgemein

Mit der Pick Post und My Post 24 können Briefe und Pakete an die genannte Pick-Up Station gesendet werden. Es besteht auch das Angebot, Pakete von einer Pick-Up Station zu versenden. Die Auswahl einer Pick-Up Station geschieht dabei mit der Angabe der entsprechenden Station. Der Dienst lässt sich somit in jedem Onlineshop nutzen.

Die Abholung der Artikel muss innerhalb von 10 Tagen geschehen. Sobald die Artikel zur Abholung bereit sind, erhält der Kunde wahlweise eine Bestätigungs-mail oder ein SMS. Mit dem darin enthaltenen Abholcode lässt sich die Ware abholen. [Post-CH-AG, 2015]

Im Grossraum Luzern befinden sich acht My Post 24-Abholstellen. [**myPost24Stations**] Das Design der Abholstelle sieht dabei konventionellen Briefkästen der Post sehr ähnlich.



Abbildung 1: My Post 24-Abholstelle,  
Quelle: AG, 2021

Die Lösung der Post behebt dabei aber nicht die in Kapitel 1 beschriebenen Probleme. Das Angebot richtet sich hauptsächlich an Personen, welche bei der Lieferung der Post nicht zuhause sind. Durch das Angebot kann so ein Abholen an der Poststelle vermieden werden. Die Lieferzeit sowie die Lieferkosten bleiben dabei aber vorhanden. Der Bestell- und Bezahlvorgang wird beim jeweiligen Onlineshop durchgeführt.

## 2.2 avec

### 2.2.1 avec now

"— Dein Online Lieferservice von avec — Mit avec now haben wir rund um das beliebte Angebot unseres Convenience-Formats avec einen Online-Store lanciert. Zur Auswahl steht ein breites Convenience-Sortiment. Die bestellten Waren werden direkt in unseren Stores zusammengestellt und so schnell wie möglich ausgeliefert." [Valora, 2021e]

Auf der Website von avec now wird dabei eine Lieferzeit von 60 Minuten angepriesen. Der Mindestbestellwert beträgt dabei 20.- Fr. [Valora, 2021a] Das Angebot gilt dabei für einen grossen Teil des Sortiments, darunter auch Tabakwaren. Das Angebot befindet sich dabei noch in der Pilotphase und wird nur im Raum Zürich angeboten. [Valora, 2021f]

### 2.2.2 avec Box

Der Anbieter geht bei diesem Angebot einen neuen Weg. Das gesamte Einkaufen wird mittels App durchgeführt. Mittels dieser können Produkte dem Warenkorb hinzugefügt und so bezogen werden. In einer ersten Phase sind zu Stosszeiten Mitarbeiter präsent, um die Kunden beim Einkauf zu unterstützen.

Um Tabakprodukte zu beziehen, befindet sich in der App eine integrierte Altersverifikation. Die Tabakwaren werden dabei im Store via Touchscreen ausgewählt. Durch die App wird eine Altersverifikation durchgeführt. [Valora, 2021b]

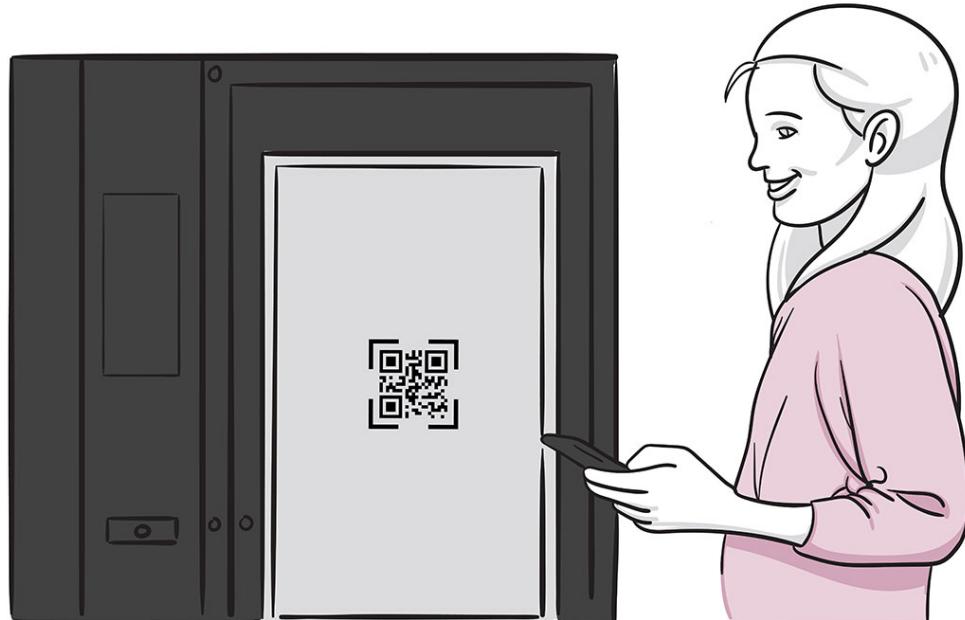


Abbildung 2: Tabakkauf avec box,  
Quelle: Valora, 2021b

Nach momentanem Stand steht die avec box am Campus der ETH Zürich. Die Box ist dabei von Montag-Sonntag jeweils von 6:00 - 22:00 in Betrieb. Es ist ein Rollout in weitere Regionen der Schweiz vorgesehen. [Valora, 2021d]

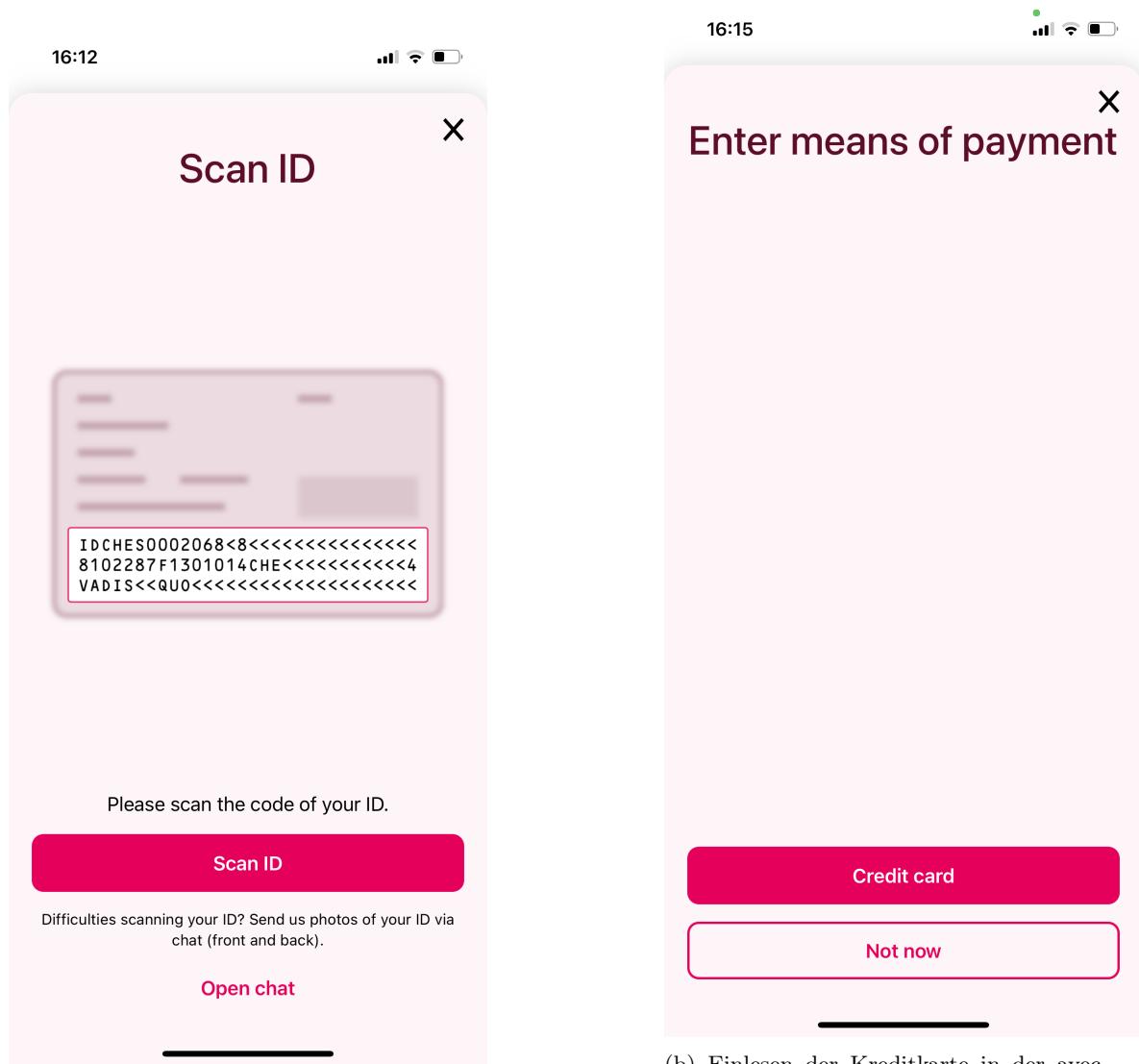
Das Angebot von avec löst einige der genannten Probleme. So kann die Ware direkt bezogen werden.

Es fallen keine Versandkosten an. Zur Zeit ist die avec box nur an einem Standort verfügbar, was die Verfügbarkeit erheblich einschränkt. Zudem ist sie nur zwischen 6:00-22:00 in Betrieb.

### 2.2.3 Ablauf Tabakkauf

Die avec box ist sehr ähnlich zur JTI Pick-Up Station. Aus diesem Grund wird der Registrierungsvorgang nachfolgend genauer betrachtet.

**Registrierung** Das Anlegen eines avec-Kontos verläuft dabei analog zur Erstellung von anderen Accounts. Mittels Handynummer und Passwort wird der Account angelegt. Zur Verifikation wird ein Bestätigungscode an die Nummer gesendet. Dieser muss eingegeben werden. Anschliessend wird die Alterverifikation durchgeführt. Hierzu muss die Identitätskarte mit der Kamera eingelesen werden. In einem nächsten Schritt kann die Kreditkarte hinterlegt werden. Der Bezahlldienst wird dabei von Datatrans bereitgestellt. Anschliessend ist die Registrierung abgeschlossen und der Einkauf in der avec box könnte beginnen.



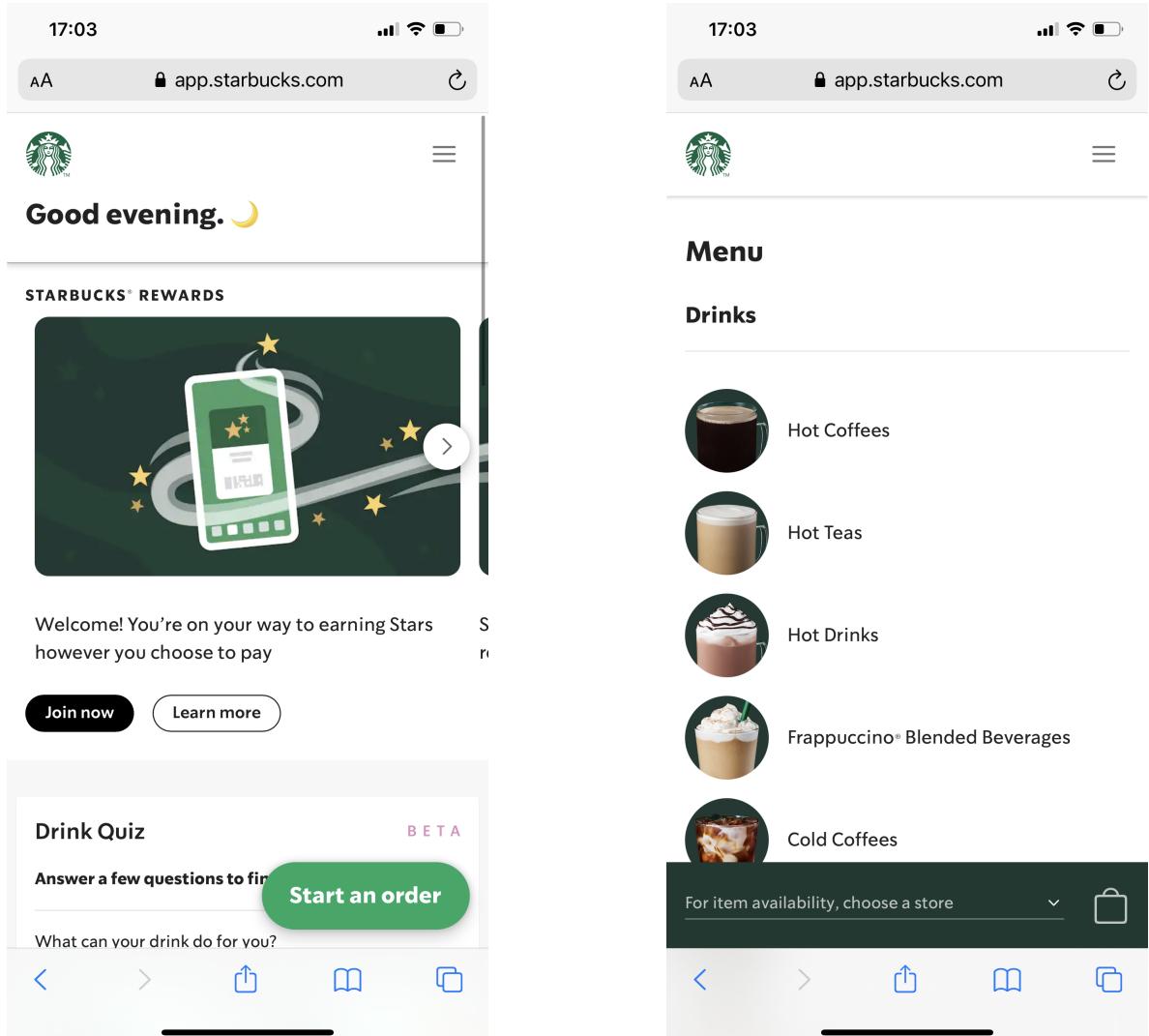
(a) Einlesen der Identitätskarte,  
Quelle: Valora, 2021c

(b) Einlesen der Kreditkarte in der avec App,  
Quelle: Valora, 2021c

Leider wird in der Applikation keine Auskunft über den Anbieter der Alterverifikation gegeben. Das Vorgehen ist dabei sehr intuitiv und schnell.

### 2.3 Starbucks PWA

In diversen Quellen wird die PWA von Starbucks immer als eine der besten PWA's genannt. [] Die Applikation ermöglicht es dem Nutzer, die angebotenen Produkte zu bestellen und diese anschließend im Store abzuholen. Der Anwendungszweck ist somit ähnlich zur JTI-Pick-Up Station. Sie ist dabei sehr nahe an einer nativen App, wodurch dem Benutzer die Bedienung sehr leicht fällt. Die Applikation reagiert sehr schnell, es sind keine Ladezeiten zu bemerken. Zudem ist die PWA auch offline nutzbar. Hierbei kommt es zwar zu Einschränkungen in der Nutzung, jedoch lässt sie sich weiterhin bedienen.



#### 2.3.1 Standortsuche

Die Applikation bietet auch ein Feature, um die nächstgelegene Starbucksfiliale anzuzeigen. Hierbei wird auf die aktuelle Position des Nutzers zugegriffen.

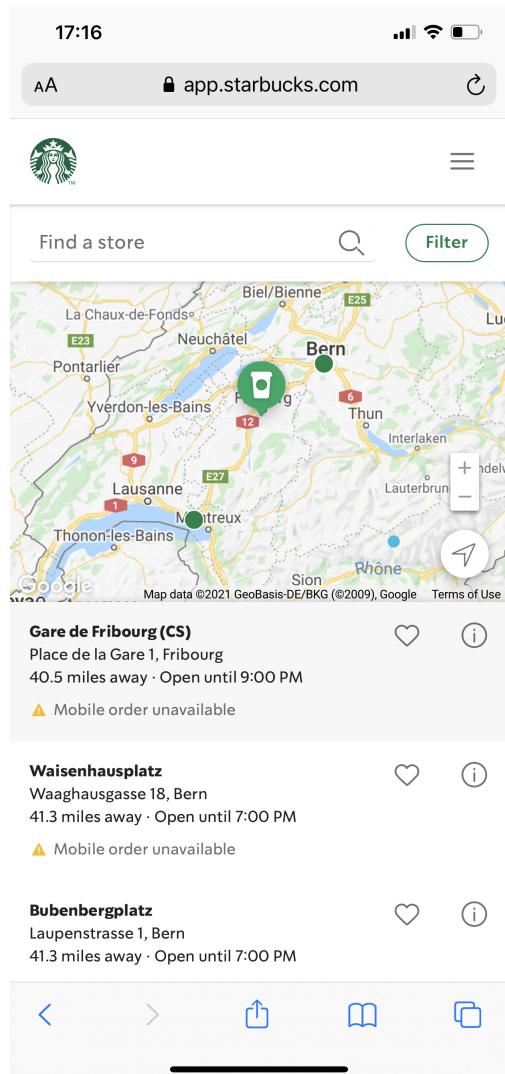


Abbildung 5: Standort in der StarbucksPWA,  
Quelle: Starbucks, 2021c

### 2.3.2 Fazit

Die Applikation von Starbucks zeigt auf, was eine PWA leisten kann. Sie dient als Vorbild für die Applikation der JTI-Pick-Up Station.

### 2.4 Fazit

Es existieren diverse Produkte, welche einen ähnlichen Ansatz verfolgen wie dieses Projekt. Besonders hervorzuheben ist dabei die avec box. 2.2.2 Der Betreiber verfolgt hier einen ähnlichen Ansatz. Der Kaufvorgang bei Tabakwaren unterscheidet sich dabei kaum von dem in diesem Projekt umzusetzenden. Durch die Analyse des dort verwendeten Vorgehens konnte ein guter Überblick gewonnen werden. Zudem konnte auch gesehen werden, wie die Integration der Altersverifikation umgesetzt wurde. Dieses Wissen ist für die spätere, eigene Umsetzung sehr wichtig.

Die Applikation von Starbucks liefert einen sehr guten Überblick über die Möglichkeiten von PWA's. Besonders Designtechnisch ist diese Anwendung sehr wertvoll.

Die anderen analysierten Angebote lieferten keinen Mehrwert für das Projekt, da sie die gestellte Problematik nur bedingt oder gar nicht lösen.

### 3 Ideen und Konzepte

#### 3.1 Systemarchitektur

Als Systemarchitektur stand die Erweiterbarkeit im Vordergrund. Allerdings sollte die Applikation durch die verwendete Architektur nicht unmöglich komplex werden. Aus diesem Grund wurde bewusst gegen eine Microservicearchitektur entschieden. Die Umsetzung der Applikation mit Microservices würde zwar zu einer besseren Verteilbarkeit und Skalierung führen, der Aufwand der Umsetzung würde jedoch erheblich steigen.

Aus diesem Grund wurde auf eine Schichtenarchitektur gesetzt. Die klassische logische drei Schichten Architektur wurde dabei noch weiter verfeinert. Final wurde eine 6-Schichten-Architektur entworfen. Die Architektur wird zusätzlich in 3 physische Tier aufgeteilt.

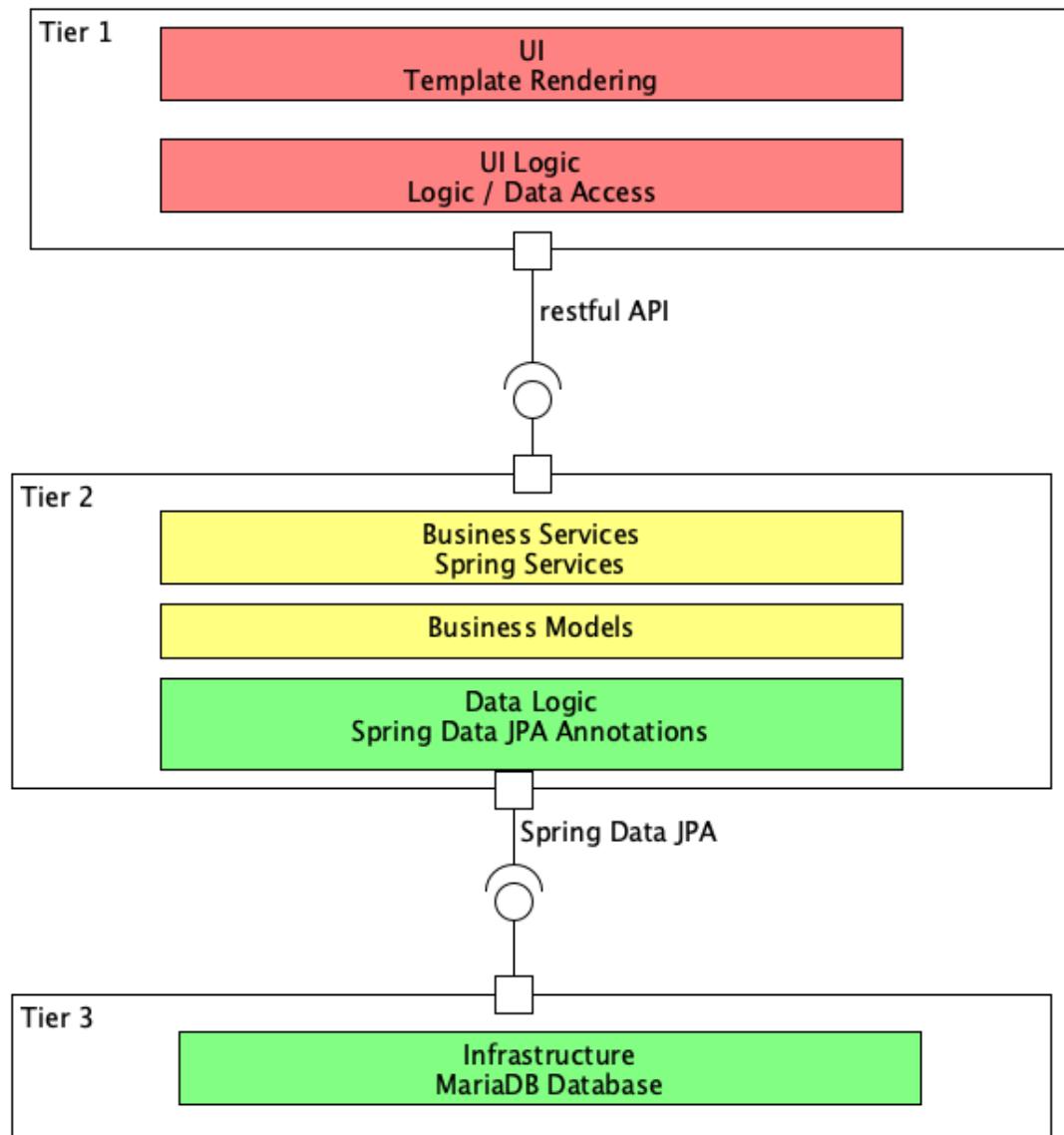


Abbildung 6: Architektur,  
Quelle: Autor

Die Architektur ermöglicht dabei eine sehr gute Erweiterbarkeit. Zudem kann die Software verteilt werden und unabhängig voneinander Skalieren. Durch die Aufteilung in 6 Tier wird zudem vermieden, dass die einzelnen Tiers zu breit werden. Durch das Verwenden von Frameworks kann zudem die Komplexität gering gehalten werden. Die Schnittstellen innerhalb der einzelnen Layer sind klar vorgegeben. Durch die Kommunikation via REST, bzw. die Spring Data JPA sind die Komponenten untereinander austauschbar. Die Kopplung ist sehr gering.

## 3.2 Frameworks

### 3.2.1 Spring Boot

Spring Boot baut auf dem Spring Framework auf. Dieses bietet sich an, um performante Enterprise-Applikationen mit Java zu erstellen. Spring Boot kann dabei sehr einfach erweitert werden, sodass Spring Security, Spring Web oder Spring Data eingesetzt werden kann.

Spring Boot kommt dabei mit einem Tomcat Server. Dies verringert den Konfigurationsaufwand. Es ist kein Einpacken in ein .war nötig und kein zusätzliches Deployen. Zudem wird das Debugging erheblich erleichtert.

Es übernimmt dabei einen grossen Teil der Beans-Konfiguration von Spring [Waldmann, 2020].

Spring Boot bietet dabei den grossen Vorteil, dass bereits Erfahrung in der Anwendung vorhanden ist. Es ist somit keine Einarbeitungszeit nötig, gängige Fehler können vermieden werden. Aus diesem Grund fiel die Entscheidung gegen Frameworks wie NodeJS.

### 3.2.2 Angular

Angular ist ein Application Design Framework, um effiziente Single Page Applikationen zu erstellen. Es basiert auf TypeScript [Böhm, 2017]. Zudem bietet Angular die Möglichkeit, als PWA genutzt zu werden [Google, 2021b]. Angular wurde dabei von Google entwickelt und bietet mit dem Material UI bereits viele Elemente, welche von nativen Android-Apps bekannt sind [Google, 2021a].

Wie auch bei Spring ist auch bei Angular bereits Projekterfahrung vorhanden. Ein Einarbeiten ist nicht mehr nötig, die gängigsten Funktionen sind bereits bekannt.

In vorhergehenden Projekten wurde das Frontend mit unterschiedlichen Technologien umgesetzt. Einerseits kam hier Plain JavaScript zum Einsatz, andererseits wurden auch Technologien wie React genutzt. Allerdings überzeugte Angular von all diesen am Meisten. Besonders durch die Verwendung von TypeScript wird dieses Framework hier genutzt.

## 3.3 weitere Technologien

### 3.3.1 Docker

Docker ist eine Containertechnologie. Sie erlaubt die Erstellung und den Betrieb von Linux-Containern. Die Container sind dabei sehr leichtgewichtig und modular.

Die einzelnen Teile der Applikation werden in verschiedenen Containern betrieben [RedHat, 2021]. Die Anwendung und Funktion von Docker wird in diversen Modulen an der Hochschule Luzern gelehrt. In vorhergegangenen Projekten wurde Docker bereits im selben Kontext wie hier genutzt.

### 3.3.2 MariaDB

MariaDB ist ein OpenSource Datenbankmanagementsystem. Es ist durch die Abspaltung von MySQL entstanden. Es handelt sich dabei um ein Relationale Datenbanksystem. Aus Lizenzgründen wird in diesem Projekt MariaDB und nicht MySQL genutzt. Alternativ wäre auch ein Einsatz von PostgreSQL möglich. Der Unterschied zwischen PostgreSQL und MariaDB ist dabei nur marginal. Bei dieser Applikation wurde lediglich aus Erfahrung auf MariaDB gesetzt [DB-Engines, 2021].

Durch die Verwendung von Spring Data wäre es möglich, die Datenbank im Verlauf des Projektes zu wechseln.

## 4 Methoden

### 4.1 SoDa

Das Projektmanagement wird mit dem hybriden Projektvorgehen SoDa der Hochschule Luzern durchgeführt. SoDa wurde dabei bereits schon in vorhergehenden Softwareentwicklungsmodulen eingesetzt und hat sich hier bewährt. Die Userstories werden dabei auch im GitLab erfasst.

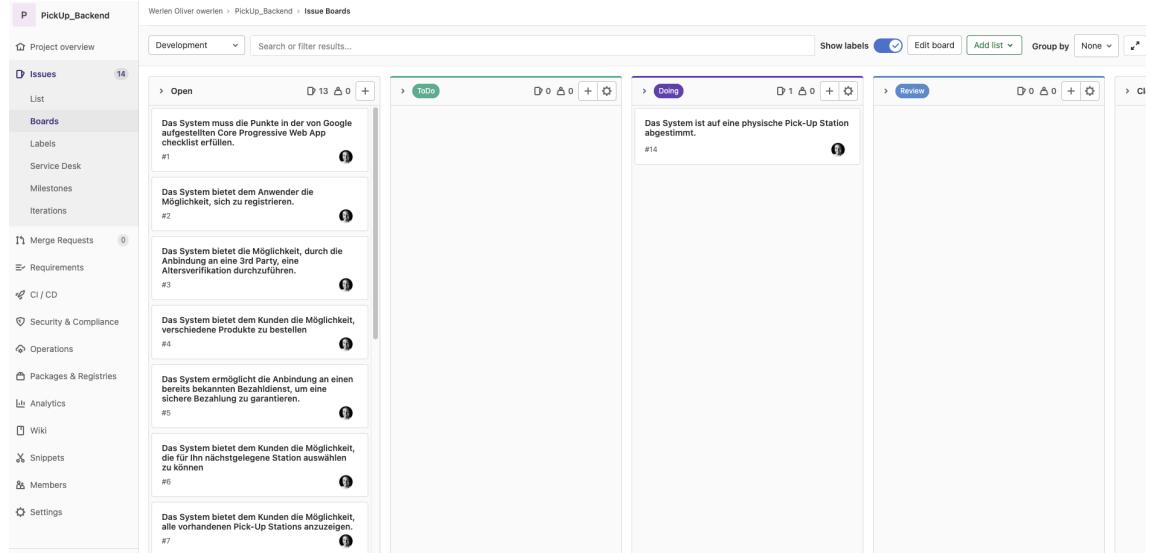


Abbildung 7: GitLab Board,  
Quelle: Autor

Für die Sprint Planung werden einzelne User Stories verschoben. Beim Sprint Review werden die Stories in "reviewing" mit den definierten Akzeptanzkriterien mit den Resultaten verglichen. Basierend darauf wird entschieden, ob an der User Story noch weiter gearbeitet werden, das heisst zurück zu "doing" oder die Story in "done" verschoben werden kann. Bei Beginn des nächsten Sprints wird das Vorgehen wiederholt.

Die einzelnen Items sind dabei priorisiert. Elemente mit einer hohen Einstufung werden dabei bei der Bearbeitung vorgezogen.

### 4.2 CI/CD

## 5 Realisierung

### 5.1 Initialisierungsphase

#### 5.1.1 Kick-Off Meeting

Das Kick-Off Meeting fand am 23.02.2021 als Zoom Meeting statt. Das Sitzungsprotokoll dazu ist im Kapitel E.1 zu finden.

Es waren bei diesem Meeting alle Projektbeteiligten anwesend. In erster Linie wurde von Herr Meier das genaue Vorgehen bei der Bachelorarbeit vorgestellt. Anschliessend stellte der Auftraggeber das Projekt genauer vor und zeigte dabei seine Erwartungen auf. Die Aufgabenstellung wurde finalisiert und von allen Projektbeteiligten akzeptiert. Zudem schlug der Betreuer vor, im zwei bis drei Wochenrhythmus ein Meeting abzuhalten. Im Anschluss wurde ein Termin für das erste Meeting vereinbart.

#### 5.1.2 Erstellen des Projektmanagementplans

Gemäss Software Development Agile (SoDa) wurde in einem ersten Projektschritt der Projektmanagementplan Berstellt. In diesem wurde auch gleich der Rahmenplan erarbeitet. In diesem sind unter anderem die Meilensteine des Projekts dargestellt. Der Projektstrukturplan erleichterte dabei das Finden. Sie wurden in einem nächsten Schritt genauer spezifiziert und die Deliverables für ein erfolgreiches Erreichen des Meilensteins definiert. Durch den Projektstrukturplan konnten die einzelnen Teilbereiche des Projekts aufgelistet werden. In der folgenden Risikoanalyse wurden die Risiken und entsprechenden Gegenmassnahmen erarbeitet.

Als letztes wurde die Projektunterstützung genauer spezifiziert. Dabei wurden die zu verwendenden Tools sowie die Elemente der Konfigurationseinheit festgelegt.

Als letzter Teil des Projektmanagements wurde die Teststrategie und die Testdrehbücher formuliert. Die Testdrehbücher werden während des Projekts vorlaufend formuliert.

#### 5.1.3 Problem und Vision

Das Kapitel 1 wurde zum Beginn der Initialisierungsphase bearbeitet. Die Hauptprobleme konnten dabei sehr schnell gefunden werden. Die Vision des Projektes konnte mithilfe der Aussagen des Auftraggebers im Kick-Off Meeting E.1 sehr gut beschrieben werden.

#### 5.1.4 Requirement Engineering

Beim Requirements Engineering diente die IEEE Spezifikation 29148-2018 als Grundlage [Doran, 2018]. Um die Requirements zu finden, wurde in einem ersten Schritt eine Analyse der Aufgabenstellung durchgeführt. Ergänzt wurden diese durch Befragungen des Auftraggebers. Durch die so erlangten Informationen konnten die Anforderungen formuliert werden. In einem letzten Schritt wurden diese mit dem Auftraggeber besprochen. Hierbei wurden noch einige Anpassungen gemacht.

#### 5.1.5 Stand der Technik

In diesem Kapitel wurde eine Analyse der bestehenden, vergleichbaren Lösungen durchgeführt. Dabei wurde vor allem das Angebot der Post und von Valflora genauer betrachtet. Die bekannte PWA von Starbucks zeigte dabei sehr gut deren Möglichkeiten auf.

#### 5.1.6 Meilenstein Abschluss Initialisierungsphase

##### Meilensteinbericht

**Termin Meilenstein 2** Der Meilenstein 2 ist am 07.03.2021 abgeschlossen und somit pünktlich fertiggestellt worden.

**Beschreibung Meilenstein 2** Die Beschreibung des Meilensteins ist im Abschnitt B.2.2 ersichtlich.

**Meilensteinziele/Vorgaben** Das übergeordnete Ziel dieses Meilensteins ist die Fertigstellung der Initialisierungsphase. Hierzu war die Auslieferung der nachfolgenden Artefakte notwendig:

- Projektmanagementplan
- Systemspezifikation
- Anforderungsliste

Zusätzlich wurden bereits die Kapitel 1 und 2 fertiggestellt.

**Meilensteinzielerreichung** Es konnten alle geforderten Artefakte geliefert werden. Die Artefakte wurden bereits mit der Betreuungsperson im Meeting E.3 besprochen und konnten abgenommen werden. Der Meilenstein wurde erfolgreich erreicht.

**Fazit** Es wurden alle Artefakte erarbeitet. Der Meilenstein wurde somit erreicht und es kann weiter nach Plan gearbeitet werden.

## 5.2 Konzeptionsphase

### 5.2.1 Sprint 1

User Story	Number
Das System ist auf eine physische Pick-Up Station abgestimmt.	F.2

Tabelle 1: Userstories Sprint 1,  
Quelle: Autor

**Spezifikation der Schnittstelle zur Abholung** Um eine Abholung der Produkte zu bekommen, braucht es eine Kommunikation zwischen der PWA und der Pick-Up Station. Um diese Schnittstelle genauer spezifizieren zu können, war es in einem ersten Schritt nötig, eine entsprechende Übertragungstechnologie festzulegen. Bei der Auswahl war dabei die Kompatibilität mit verschiedenen Geräten und Browsern ausschlaggebend.

**NFC** Als eine erste Idee wurde Near Field Communication (NFC) analysiert. NFC eignet sich dabei ideal für die Übertragung von geringen Datenmenge. Dabei funktioniert es bis zu einer Distanz von 10cm. Die Umsetzung wäre dabei sehr einfach mittels einem Raspberry Pi umsetzbar. Einfache NFC-Leser gibt es dabei schon für wenig Geld zu kaufen. Dabei könnte die gesamte Übertragung von der Informatik übernommen werden, eine weitere Schnittstelle zwischen Elektrotechnik und Informatik könnte dabei vermieden werden.

Die Technologie ist dabei sehr robust. Ein Überkleben oder Zerkratzen des Lesers hat keinen Einfluss auf die Funktionalität [congstar, 2021]. Der Abholvorgang wäre sehr einfach und schnell, da nur ein kurzer Kontakt mit dem Smartphone bereits ausreicht.

**Kompatibilität** Apple schränkt den Zugriff von Webpages auf Standort- und Hardwaredienste unter iOS-Geräten sehr stark ein. Offiziell begründet wurde dies durch die Einschränkung von footprinting und des damit verbundenen Nutzertrackings. Für dieses Projekt ausschlaggebend ist dabei vor allem das Entfernen von NFC- und Bluetooth-Support. Somit ist es auch PWA's nicht mehr möglich, unter iOS auf die genannten Funktionen zuzugreifen [Apple, 2021].

Die Web-NFC-API wird zusätzlich bislang nur von Google Chrome unterstützt. Dies würde die Nutzbarkeit der Applikation sehr stark einschränken [Mozilla, 2021]. Aus diesem Grund eignet sich NFC nicht für die Verwendung in diesem Projekt.

**QR-Code** Die Restriktionen von Apple schränken die geeigneten Technologien sehr stark ein. Es bleibt nur noch der Zugriff auf die Kamera, um mit der Pick-Up Station zu kommunizieren. Ursprünglich war dies vom Auftraggeber nicht gewünscht. Das Verwenden einer anderen Technologie würde aber viel zu viele Einschränkungen führen, sodass die PWA mit vielen Geräten unbrauchbar wäre.

Aus diesem Grund musste ein Konzept entwickelt werden, um die Warenausgabe mittels QR-Code auszulösen. Dabei sind zwei Ansätze möglich:

- Fixer QR-Code auf Pick-Up Station, wird von PWA eingelesen.
- Variabler QR-Code in PWA, wird von Pick-Up Station eingelesen.

Aus hardwaretechnischer Sicht ist die Umsetzung der ersten Lösung bedeutend einfacher Umzusetzen, bietet aber auch Fehlerpotential. Ein Überkleben des QR-Codes auf der Station würde die gesamte Station unbrauchbar machen. Eine Abholung der Bestellungen wäre nicht mehr möglich. Die zweite Variante ist hardwaretechnisch anspruchsvoller. Auf der Pick-Up Station muss ein optischer Leser verbaut werden. Mit diesem kann der QR-Code aus der PWA eingelesen und die Bestellung ausgegeben werden.



Abbildung 8: Beispielanwendung QR-Code auf Gerät,  
Quelle: tagmotion, 2018

Eine Grafik zum zweiten Lösungsansatz ist dabei im Kapitel 2.2.2 zu finden. Die genaue Wahl wird im Meeting mit dem Auftraggeber besprochen und abgesegnet.

### 5.2.2 Sprintreview Sprint 1

Im Sprint 1 wurde eine genauere Analyse der Schnittstelle zur Produktabholung durchgeführt. Dabei müssen im Meeting von dieser Woche die gefundenen Lösungen mit dem Auftraggeber besprochen werden. Die Userstory kann erst zu Beginn des nächsten Sprints abgeschlossen werden.

## Meilensteinbericht

**Termin Meilenstein 3** Der Meilenstein 3 ist am 18.03.2021 abgeschlossen und somit mit drei Tagen Verspätung fertiggestellt worden.

**Beschreibung Meilenstein 2** Die Beschreibung des Meilensteins ist im Abschnitt B.2.2 ersichtlich.

**Meilensteinziele/Vorgaben** Das übergeordnete Ziel dieses Meilensteins ist die Fertigstellung der Initialisierungsphase. Hierzu war die Auslieferung der nachfolgenden Artefakte notwendig:

- Projektmanagementplan
- Systemspezifikation
- Anforderungsliste

Zusätzlich wurden bereits die Kapitel 1 und 2 fertiggestellt.

**Meilensteinzielerreichung** Es konnten alle geforderten Artefakte geliefert werden. Die Artefakte wurden bereits mit der Betreuungsperson im Meeting E.3 besprochen und konnten abgenommen werden. Der Meilenstein wurde erfolgreich erreicht.

**Fazit** Es wurden alle Artefakte erarbeitet. Der Meilenstein wurde somit erreicht und es kann weiter nach Plan gearbeitet werden.

### 5.2.3 Sprint 2

User Story	Number
Das System ist auf eine physische Pick-Up Station abgestimmt.	F.2
Das System bietet dem Kunden die Möglichkeit, verschiedene Produkte zu bestellen.	F.6
Das System muss über eine CI/CD-Pipeline verfügen.	L.6

Tabelle 2: Userstories Sprint 2,  
Quelle: Autor

**CI/CD Pipeline** Nach Absprache mit dem Auftraggeber im Meeting wurde entschieden, dass der Prototyp im EnterpriseLab laufen soll. Daraufhin wurde eine Maschine beantragt. Es handelt sich hierbei um ein Ubuntu 16.07 LTS. Die Applikationen sollen dabei als Docker Container deployed werden.

Zuerst wurde geplant, die Pipeline wie im offiziellen Tutorial des Enterpriselabs zu erstellen. Auf Anfrage wurde jedoch ein anderes Vorgehen empfohlen. Nachfolgend wird die Antwort zitiert.

"Wenn es dein Ziel ist eine Spring Boot Applikation zu builden und dann auf der VM zu deployen dann würde den Container auf den Shared Runner unserer GitLab Instanz builden lassen und in die Container Registry deines Projekts pushen. Für die Deploy Stage der CI/CD Pipeline kannst du deine VM als privaten GitLab Runner registrieren und so ohne SSH login den Container von der Registry pullen und laufen lassen. Die SSL Termination mit Lets Encrypt würde ich mit einem separaten nginx Container lösen der reverse proxy spielt. Dieser kann dann einfach laufen und muss für Änderungen an der Spring Boot Applikation auch nie modifiziert werden.

Der Vorteil im Vergleich zur Docker Übung ist, dass hier alle Hosts von der GitLab CI/CD Pipeline kontrolliert werden. In der Übung ist der docker-cloud-exercise Host abgekapselt und pullt mit Watchtower einfach blind das neuste Image von einer Registry. Dieser Aufbau macht IMO mehr Sinn wenn man einfach Container Images von dritten konsumiert, aber ist weniger elegant wenn man selbst Kontrolle über die Source und CI/CD Pipelines hat." [von Uslar, 2021]

In diesem Auszug aus der Email vom Enterpriselabmitarbeiter Cyril von Uslar sind sehr viele Informationen enthalten. Es war ein mehrmaliges Durchlesen nötig, um sich darunter etwas vorstellen zu können. Anschliessend wurde die Pipelineerstellung in die wesentlichen Punkte aufgeteilt um umgesetzt.

- Builden auf dem Shared Runner
- Pushen in die Container Registry des Projekts
- VM als privaten Runner registrieren und deployen
- nginx server als reverse Proxy für SSL Termination

**Builden auf dem Shared Runner** In diesem Projekt wird die Pipeline für zwei Projekte aufgesetzt. In einem ersten Schritt wurde dies nur für die Spring Applikation durchgeführt. Das Vorgehen unterscheidet sich dabei nur im Buildprozess. Um die Spring Applikation zu Builden, wurde das Dockerfile identisch zum Spring Boot Docker-Tutorial aufgebaut [VMWare, 2021]

```
FROM maven:3.6.3-jdk-11-slim
ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

Entgegen dem Tutorial, in welchem noch die Java Version 11 genutzt wird, kommt hier direkt Java 16 zum Einsatz.

Anschliessend wurde das .gitlab-ci.yml File erstellt. Es wurde hier eine Anleitung aus dem Gitlab Blog genutzt. Diese ist jedoch nicht mehr ganz aktuell. Das Deployment wird hier zudem auf ein Kubernetes Cluster durchgeführt. Es musste daher individuell angepasst werden. Es konnte somit nur der Build-Teil übernommen werden. [Lenzo, 2016]

Gitlab stellt zum Builden von Docker-Images bereits mehrere Shared-Runner zur Verfügung.

## Shared runners

These runners are shared across this GitLab instance.

The same shared runner executes code from multiple projects, unless you configure autoscaling with [MaxBuilds](#) set to 1 (which it is on GitLab.com).

[Disable shared runners](#) for this project

### Available shared runners: 5

#### ● e1f64da7

gitlabrunner-02

#37

[docker](#) [runner02](#) [test-runner](#) [ubuntu 18.04](#)

#### ● piYF7yKo

gitlabrunner-06

#56

[docker](#) [runner06](#) [ubuntu 18.04](#)

#### ● 8bU3YzuD

gitlabrunner-01

#62

[BigBuild](#) [docker](#) [runner01](#) [ubuntu 18.04](#)

#### ● cf1ce3b4

gitlabrunner-03

#49

[docker](#) [runner03](#) [ubuntu 18.04](#)

#### ● ydSz8fXG

gitlabrunner-05

#55

[docker](#) [runner05](#) [ubuntu 18.04](#)

Abbildung 9: Verfügbare Shared-Runner von GitLab,  
Quelle: Autoren

Der jeweilige Runner wird dabei mittels eines "fair usage algorithms" zugewiesen. Dabei ist die Anzahl der momentan auszuführenden Jobs pro Projekt entscheidend für die Wahl. Das Projekt mit den am wenigsten laufenden Jobs kommt zuerst. [GitLab, 2021] Zudemachtet der runner auf die verwendeten Tags. In diesem expliziten Beispiel sind mehrere Runner mit dem Docker-Tag versehen. Daher kann nicht sicher gesagt werden, welcher Runner den Docker Build durchführt.

**Builden in die Registry des Projekt** Um das erstellte Image in die Registry des Projekts zu pushen, werden von GitLab bereits die einzelnen Befehle vorgegeben. Diese können in die package-Stage integriert werden.

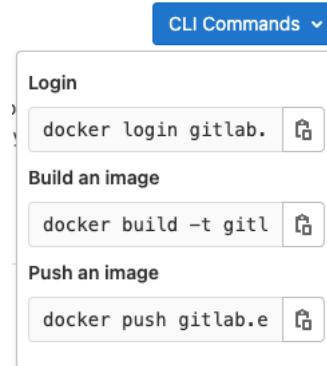


Abbildung 10: CLI Commands GitLab Container Registry,  
Quelle: Autoren

Das entsprechende Image wird so in die interne Container Registry des GitLab Projekts gepushed. Dadurch kann auf das Verwenden eines Drittddienstes wie DockerHub verzichtet werden. Beim Build werden immer zwei Images in die Registry geschrieben. Einerseits eines mit dem Tag "latest" und eines mit dem Commit-Hash als Tag. Beim Deployment wird dann immer das Image mit dem aktuellen Commit-Hash verwendet. Dieser Mechanismus dient dazu, dass alle Versionen immer vorhanden sind. "Latest" weist dabei immer auf die aktuelle Version.

**VM als privaten Runner registrieren und deployen** Um das Deployment ohne Watchtower durchführen zu können, wurde auf die Virtuelle Maschine als Docker-Runner hinzugefügt. Es handelt sich somit dabei um einen specific Runner. GitLab Runner kann dabei einfach mittels Paketmanager auf Ubuntu installiert werden. Anschliessend musste dieser noch konfiguriert werden. Dazu musste der Token sowie die URL von GitLab dem Runner hinzugefügt. Der Runner ist anschliessend im Runner Bereich des Projekts zu finden. Zudem wurde anschliessend die Option "Lock to current project" entfernt, sodass dieser Runner auch direkt für das Angular Projekt genutzt werden kann. Um ein SSH-Login zu vermeiden, wurde der Executor dabei als Shell-Executor definiert. Bei einem SSH-Executor wäre vorgängig die Verbindung via SSH nötig gewesen. Anschliessend können die entsprechenden Befehle für das Deployen in das gitlab-ci.yml integriert werden. Dabei war es zunächst ein Problem, dass die Berechtigung fehlte. Erst nach dem Login mittels GitLab CI-Token konnte der Container erfolgreich gepulled und anschliessend gestartet werden.

**Build und Deployment des Frontends** Um mit dem nächsten Schritt weiterzufahren, musste zuerst das Deployment des Angular Projekts konfiguriert werden. Hierbei liegt der Hauptunterschied in dem Build-Prozess. Dabei ist es wichtig, die beim Build entstehenden Artifakte in GitLab zu speichern. Nur so kann auf ein erneutes Builden beim Erstellen des Docker Containers verzichtet werden. Zudem werden die Node Module in den Cache gespeichert. Die nachfolgenden Befehle sind identisch zum Vorgehen bei Spring.

**nginx als Reverse Proxy** Um den nginx-Server gegen Zugriffe von Aussen zu schützen, wurde auf einen Reverse Proxy gesetzt. Der Reverse Proxy stellt dabei die einzige Verbindung ins öffentliche Netz dar. Zudem übernimmt er die Zertifikatsverwaltung des Frontends. Alternativ wäre auch ein Caching möglich, auf die Umsetzung wurde für diesen Anwendungsfall jedoch verzichtet. [KnowHow, 2020]

Die Umsetzung wurde dabei analog zum Tutorial von Alexander Bohndorf durchgeführt. [Bohndorf, o.D.] Dabei wurde der nginx-proxy von jwilder sowie die damit kompatible letsencrypt companion verwendet. Die Umsetzung wurde dabei mittels docker-compose Files durchgeführt.

**Abschliessende Bemerkungen** Das Erstellen der CI/CD Pipelines des Projekts verlief grösstenteils problemlos. Durch die Grundlage des Enterpriselabs und der sehr guten Dokumentation von GitLab konnte dies sehr schnell umgesetzt werden. Durch das Hinzufügen des Reverse Proxies konnte die Sicherheit des Systems massiv erhöht werden. Zusätzlich wurde so auch gleich die Auslieferung via HTTPS hinzugefügt sowie für ein immer gültiges Zertifikat gesorgt. Beim Backend wurde auf eine zusätzliche Test-Stage verzichtet. Die gesamten Tests werden bereits im Maven-Build ausgeführt. Beim Frontendprojekt wurde dazu eine eigene Stage für e2e Tests festgelegt. Die fertigen Konfigurationen sind im GitLab Projekt enthalten und können dort eingesehen werden.

**Bestellen von Produkten** Um eine Bestellung durchführen zu können, wurde in einem ersten Schritt das Article Object definiert. Basierend auf diesem wurde das passende Data Transfer Object (DTO) definiert. Um das Mapping zwischen DTOs und Objekt zu vereinfachen, wurde der Object Mapper modelmapper genutzt. Zudem wurde direkt mit Hypertext as the engine of application state (HATEOAS) gearbeitet. Der Controller stellt dabei die gewohnten CRUD-Operationen zur Verfügung.

**Abstimmung auf physische Pick Up Station** Im Meeting von dieser Woche wurden dem Auftraggeber die beiden in Kapitel 5.2.1 erarbeiteten Lösungen vorgestellt. Der Entscheid fiel dabei zugunsten der ersten Alternative. Dabei befindet sich der QR-Code fix auf der Station. Diese ist deutlich robuster und einfacher umzusetzen.

Die Schnittstellen zwischen Elektrotechnik und Informatik sind essentiell für die Funktionalität des Endprodukts. Aus diesem Grund wurde in diesem Sprint ein Meeting zwischen den Projektbeteiligten einberufen. Besonders das Senden der Ausgabeanforderung führte dabei zu Problemen. Hierbei soll auf einen Busy-Waiting Ansatz verzichtet werden. Jedoch soll die Lösung auch sehr energiesparend und effizient umsetzbar sein. Da beide Beteiligten keine Erfahrung im Internet of Things (IoT)-Umfeld besitzen, wurde hier der Betreuer Michael Handschuh um Hilfe gebeten.

#### 5.2.4 Sprintreview Sprint 2

In Sprint 2 konnten nicht alle User Stories vollständig erfüllt werden. Es werden daher zwei von drei User Stories im nächsten Sprint weiter bearbeitet. Die Umsetzung der CI/CD Pipeline ist hingegen abgeschlossen. Dies war auch die Hauptarbeit in diesem Sprint. Für die Spezifizierung der Schnittstelle wird beim Meeting mit dem Betreuer eine geeignete Lösung erarbeitet. Bei dem Bestellprozess ist bereits die Abfrage von Produkten an der API möglich. In einem nächsten Schritt wird das passende Frontend entworfen und umgesetzt.

#### 5.2.5 Sprint 3

**User Stories** In diesem Sprint wurden keine neuen User Stories zum Sprint Backlog hinzugefügt. Es wird weiterhin an den beiden verbleibenden User Stories gearbeitet.

## 6 Evaluation und Validation

## 7 Ausblick

## 8 Verzeichnisse

### Abbildungsverzeichnis

1	My Post 24-Abholstelle . . . . .	5
2	Tabakkauf avec box . . . . .	6
5	Standort in der Starbucks PWA . . . . .	9
6	Architektur . . . . .	10
7	GitLab Board . . . . .	13
8	Beispielanwendung QR-Code auf Gerät . . . . .	16
9	Verfügbare Shared-Runner von GitLab . . . . .	19
10	CLI Commands GitLab Container Registry . . . . .	20
11	Organigramm . . . . .	31
12	Projektstrukturplan . . . . .	32
13	SoDa Rahmenplan . . . . .	33
14	Risikomatrix . . . . .	36
15	RisikomatrixNach . . . . .	38
16	Systemarchitektur . . . . .	41
17	Kontextdiagramm . . . . .	42
18	Datenbankschema . . . . .	43
19	SoDa Rahmenplan Version 1 . . . . .	55

## Glossar

**Github** grösste Versionskontrollplattform [[Github](#)]. 25

**GitLab** Versionskontrollsyste aufbauend auf git, Alternative zu Github. 13

**React** Java Script Library zum Erstellen von User Interfaces [Facebook, 2021]. 11

**SoDa** Hybrides Projektmanagementvorgehen der Hochschule Luzern. 1, 13

**User Story** Element in SoDa, werden aus Epics im Format Als Rolle möchte ich Ziel/Wunsch, um Nutzen erstellt [[userStory](#)]. 13

## Abkürzungsverzeichnis

<b>SoDa</b> Software Development Agile.....	14
<b>HATEOAS</b> Hypermedia as the Engine of Application State .....	21
<b>HATEOAS</b> Hypertext as the engine of application state.....	21
<b>DTO</b> Date Transfer Object.....	21
<b>JTI</b> Japan Tobacco International.....	46
<b>PWA</b> Progressive Web App .....	1
<b>NFC</b> Near Field Communication .....	15
<b>DTO</b> Data Transfer Object.....	21
<b>HATEOAS</b> Hypertext as the engine of application state .....	21
<b>IoT</b> Internet of Things .....	21

## Tabellenverzeichnis

1	Userstories Sprint 1 . . . . .	15
2	Userstories Sprint 2 . . . . .	17
3	Testprotokoll Test 1, Quelle: Autoren . . . . .	30
4	Meilensteine, Quelle: Autoren . . . . .	35
5	Risikoanalyse, Quelle: Autoren . . . . .	36
6	Risikoanalyse nach Massnahmen, Quelle: Autoren . . . . .	37
7	Entwicklungstools, Quelle: Autoren . . . . .	39
8	Testtools, Quelle: Autoren . . . . .	39
9	Konfigurationseinheit Release 1, Quelle: Autoren . . . . .	39
10	Test Lernmodus Frage anzeigen, Quelle: Autoren . . . . .	40
11	Funktionale Anforderungen, Quelle: Autoren . . . . .	47
12	Nicht Funktionale Anforderungen, Quelle: Autoren . . . . .	47
13	Sitzungsprotokoll, Quelle: Autoren . . . . .	49
14	Sitzungsprotokoll, Quelle: Autoren . . . . .	51
15	Sitzungsprotokoll, Quelle: Autoren . . . . .	52
16	Sitzungsprotokoll, Quelle: Autoren . . . . .	53

## Literatur

- AG, D. S. P. (2021). Versenden und empfangen rund um die Uhr. Zugriff unter <https://www.post.ch/de/empfangen/empfangsorte/pickpost-my-post-24/my-post-24>. (03.03.2021)
- Apple. (2021). Tracking Prevention in WebKit. Zugriff unter <https://webkit.org/tracking-prevention/>. (12.03.2021)
- Böhm, R. (2017). Was sind Angular und Angular JS. Zugriff unter <https://angular.de/artikel/was-ist-angular/>. (14.03.2021)
- Bohndorf, A. (o.D.). DOCKER-COMPOSE SETUP MIT NGINX REVERSE PROXY. Zugriff unter <https://sitegeist.de/blog/typo3-blog/docker-compose-setup-mit-nginx-reverse-proxy.html>. (22.03.2021)
- congstar. (2021). NFC-Near Field Communication. Zugriff unter <https://www.congstar.de/handys-technik-news-trends/nfc/>. (12.03.2021)
- Coop. (2020). Lieferbedingungen. Zugriff unter <https://www.coop.ch/de/wie-wir-liefern.html>. (03.03.2021)
- Doran, T. (2018). IEEE/ISO/IEC 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering. Zugriff unter <https://standards.ieee.org/standard/29148-2018.html>. (27.02.2021)
- DB-Engines. (2021). Vergleich der Systemeigenschaften MariaDB vs. PostgreSQL vs. MySQL. Zugriff unter <https://db-engines.com/de/system/MariaDB%3BPostgreSQL>. (14.03.2021)
- Facebook. (2021). React. Zugriff unter <https://reactjs.org/>. (14.03.2021)
- Genf, K. (o.D.). Versand und Zahlungsbedingungen. Zugriff unter <https://www.kioskino.ch/de-versand-und-zahlungsbedingungen>. (03.03.2021)
- GitLab. (2021). Configuring runners in GitLab. Zugriff unter <https://docs.gitlab.com/ee/ci/runners/>. (18.03.2021)
- Google. (2021a). Angular Material. Zugriff unter <https://material.angular.io/>. (14.03.2021)
- Google. (2021b). Angular Service Worker Introduction. Zugriff unter <https://angular.io/guide/service-worker-intro>. (14.03.2021)
- HSLU. (o.D. a). Artefakte und Downloads Planungs- und Entwurfsdokumente. Zugriff unter <https://www.hslu.ch/de-ch/informatik/studium/soda/artefakte-und-downloads/>. (02.03.2021)
- HSLU. (o.D. b). Planung und Vorgehen «Lieber ungefähr richtig, als genau falsch». Zugriff unter <https://www.hslu.ch/de-ch/informatik/studium/soda/planung/>. (02.03.2021)
- KnowHow. (2020). Reverse-Proxy-Server-Kernkomponente in Sicherheitsarchitekturen. Zugriff unter <https://www.ionos.de/digitalguide/server/knowhow/was-ist-ein-reverse-proxy/>. (22.03.2021)
- Krüger, N. (2018). How to Write a Software Requirements Specification (SRS Document). Zugriff unter <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>. (27.02.2021)
- Lenzo, M. (2016). Continuous delivery of a Spring Boot application with GitLab CI and Kubernetes. Zugriff unter <https://about.gitlab.com/blog/2016/12/14/continuous-delivery-of-a-spring-boot-application-with-gitlab-ci-and-kubernetes/>. (18.03.2021)
- Mozilla. (2021). Web NFC API. Zugriff unter [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_NFC\\_API#browser\\_compatibility](https://developer.mozilla.org/en-US/docs/Web/API/Web_NFC_API#browser_compatibility). (12.03.2021)
- Post-CH-AG. (2015). Allgemeine Geschäftsbedingungen PickPost und My Post 24. Zugriff unter <https://www.post.ch/de/empfangen/empfangsorte/pickpost-my-post-24/my-post-24/weitere-informationen>
- RedHat. (2021). Docker - Funktionsweise, Vorteile, Einschränkungen. Zugriff unter <https://www.redhat.com/de/topics/containers/what-is-docker>. (14.03.2021)
- Sam Richard, P. L. (2020). What makes a good Progressive Web App? Zugriff unter <https://web.dev/pwa-checklist/>. (24.02.2021)
- Starbucks. (2021a). app.starbucks. Zugriff unter <https://app.starbucks.com>. (09.03.2021)
- Starbucks. (2021b). app.starbucks. Zugriff unter <https://app.starbucks.com/menu>. (09.03.2021)
- Starbucks. (2021c). app.starbucks. Zugriff unter <https://app.starbucks.com/store-locator?map=46.670396,7.455289,10z>. (09.03.2021)

- tagmotion. (2018). NFC vs. QR-Code. Zugriff unter <https://www.tagmotion.de/nfc-vs-qr-code/>. (12.03.2021)
- Valora. (2021a). Avec Now. Zugriff unter <https://www.avecnow.ch>. (09.03.2021)
- Valora. (2021b). Bewährtes und Neues nur für dich. Zugriff unter <https://avec.ch/de/avecbox/>. (09.03.2021)
- Valora. (2021c). Scan ID. Zugriff unter Kontoerstellung. (09.03.2021)
- Valora. (2021d). Standorte und Öffnungszeiten. Zugriff unter <https://avec.ch/de/avecbox/>. (09.03.2021)
- Valora. (2021e). Über avec now. Zugriff unter <https://www.avecnow.ch/pages/uber-avec-now>. (09.03.2021)
- Valora. (2021f). Versand. Zugriff unter <https://www.avecnow.ch/policies/shipping-policy>. (09.03.2021)
- VMWare. (2021). Spring Boot with Docker. Zugriff unter <https://spring.io/guides/gs/spring-boot-docker/>. (18.03.2021)
- von Uslar, C. (2021). private email communication.
- Waldmann, S. (2020). Spring oder Spring Boot, das ist hier die Frage. Zugriff unter <https://blog.doubleslash.de/spring-vs-spring-boot/>. (14.03.2021)

## A Testprotokolle

### A.1 Testprotokolle Schülermodus

<b>Test Nr.</b>	1
<b>Beschreibung</b>	Durch diesen Test wird die Lernfunktion sowie die Lernstatistik für Lernende manuell getestet.
<b>Randbedingungen</b>	<ul style="list-style-type: none"> <li>• Die Testperson besitzt einen Account bzw. besitzt den Zugriff auf die Applikation.</li> <li>• Die Testperson kann selbstständig auf die Fragen zugreifen.</li> </ul>
<b>erwartete Resultate</b>	<ul style="list-style-type: none"> <li>• Wenn die Testperson ein Quiz startet, wird diesem eine Frage angezeigt.</li> <li>• Der Testperson wird neben der Frage auch die Antwortmöglichkeiten angezeigt.</li> </ul>
<b>Testperson</b>	Frederico Fischer
<b>Datum</b>	01.10.2020
<b>Durchführung</b>	<ol style="list-style-type: none"> <li>1. Die Testperson gibt die URL ein.</li> <li>2. Die Testperson drückt auf den Reiter Übungen.</li> <li>3. Auf dem Bildschirm erscheint die Frage mit den möglichen Antworten.</li> </ol>
<b>erhaltenes Resultat</b>	<ol style="list-style-type: none"> <li>1. Die Testperson hat eine Frage angezeigt bekommen.</li> <li>2. Die Testperson hatte eine Übersicht über alle möglichen Antworten.</li> </ol>
<b>Test bestanden</b>	Ja

Tabelle 3: Testprotokoll Test 1, Quelle: Autoren

## B Projektmanagementplan

### B.1 Projektorganisation

#### B.1.1 Organisationsplan, Rollen, Zuständigkeiten

In nachfolgendem Diagramm sind alle Projektbeteiligten aufgeführt. Die Projektmitglieder von der Hochschule Luzern Technik und Architektur unterstehen dabei in diesem Projekt keiner hier genannten Person. Sie haben aus Ihrem Projekt entsprechend eigene Projektorganisationen.

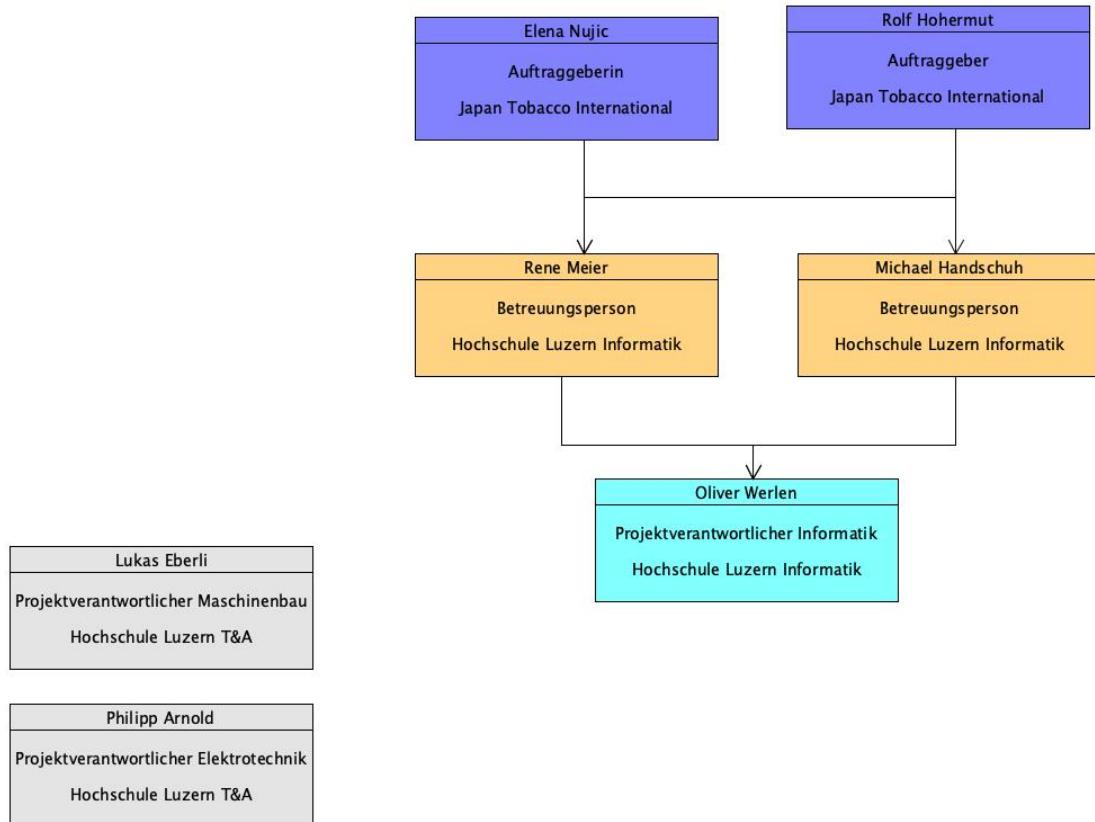


Abbildung 11: Organigramm, Quelle: Autoren

**Rollen** Im Projekt wird nach dem hybriden Projektmanagementvorgehen SoDa gearbeitet. Es werden die hier genutzten Rollen beibehalten.

- Projektleiter/in
- Product Owner
- Scrum Master
- Scrum Team

[HSLU, o.D. b] Da es jedoch in diesem Projekt nur einen aktiven Projektmitarbeiter gibt, werden alle Rollen von Oliver Werlen übernommen.

### B.1.2 Projektstrukturplan

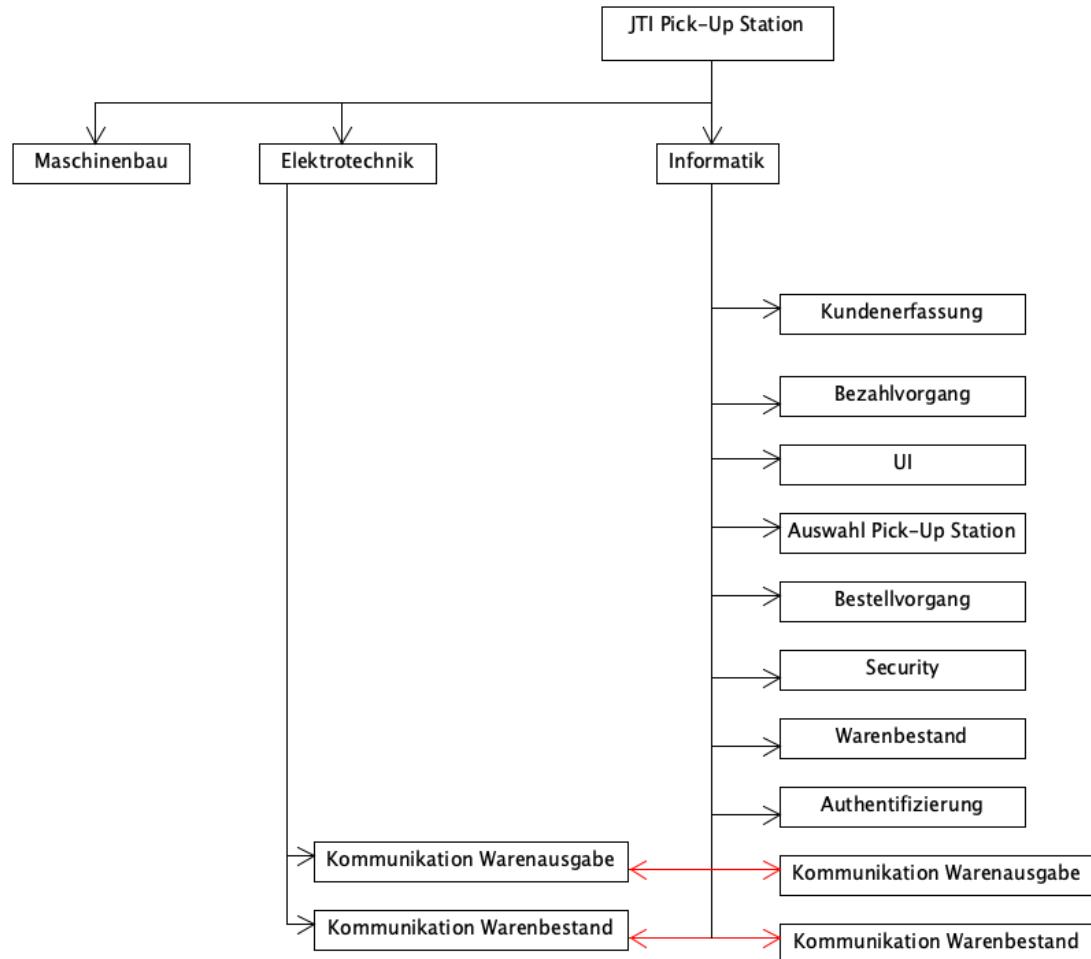


Abbildung 12: Projektstrukturplan,  
Quelle: Autoren

**Beschreibung** Im obigen Projektstrukturplan in Abbildung 12 werden die wichtigsten Teilbereiche der Applikation aufgelistet. Dabei wird der Fokus auf den Informatikteil gelegt. Es werden einzig die Schnittstellen zur Elektrotechnik berücksichtigt. Diese wurden rot eingezzeichnet. Die Teilbereiche beziehen sich dabei hauptsächlich auf die in D erarbeiteten Anforderungen.

## B.2 Projektführung

### B.2.1 Rahmenplan

Im untenstehenden Rahmenplan wird mittels Zeitstrahl eine Grobplanung dargestellt.

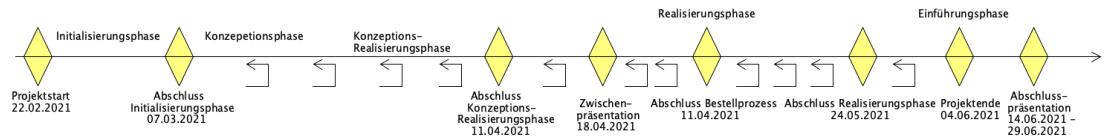


Abbildung 13: Rahmenplan,  
Quelle: Autoren

Der Rahmenplan wurde zu Beginn des Projekts grob dargestellt. Im Verlauf des Projekts kann dieser bei Bedarf angepasst werden. Die einzelnen Versionen des Rahmenplans sind im Anhang F zu finden.

**B.2.2 Meilensteine**

Wie in Abbildung 13 zu sehen gibt es insgesamt sieben Meilensteine. Diese werden in folgender Tabelle beschrieben sowie die nötigen Deliverables aufgezeigt.

<b>Meilenstein</b>	<b>Beschreibung</b>	<b>Deliverables</b>
Projektstart	Das Kick-Off Meeting mit allen Projektteilnehmern wurde durchgeführt und das Projekt freigegeben.	finale Aufgabenstellung
Abschluss Initialisierungsphase	In der Initialisierungsphase wurden alle zum erfolgreichen Start benötigten Unterlagen erstellt. Die Anforderungen wurden von allen Projektmitgliedern akzeptiert.	Projektmanagementplan, Systemspezifikation, Anforderungsliste
Abschluss Konzeptions-Realisierungsphase	Es sind GUI Prototypen vorhanden und vom Auftraggeber abgenommen worden. Die CI/CD Pipeline ist erfolgreich umgesetzt worden. Es wurde eine Spezifikation der Schnittstelle zwischen Software und Hardware erstellt. Zudem wurde das Kapitel 2 abgeschlossen.	abgenommene GUI-Prototypen, CI/CD Pipeline, Schnittstellenspezifikation, Kapitel SStand der Technik" <b>Release 1</b>
Abschluss Bestellprozess	Der Bestellprozess Artikelauswahl, Artikel in Warenkorb, Artikel Bezahlensowie die Kundenregistrierung sind umgesetzt und getestet.	Testprotokolle zu Abschluss Bestellprozess, Demo Bestellprozess, Release Bestellprozess <b>Release 2</b>
Zwischenpräsentation	Die Zwischenpräsentation ist durchgeführt worden.	Zwischenpräsentation im Anhang
Abschluss Realisierungsphase	Die noch fehlenden Anforderungen aus dem vorherigen Meilenstein sind hier abzuliefern. Es handelt sich dabei um die Auswahl sowie die Abholung an einer Pick-Up Station. Zudem ist die Abfrage des Warenbestandes Teil dieses Meilensteins.	Testprotokolle zu Abholung, Testprotokolle Auswahl, Integration alte Daten, Demo verschiedene Features <b>Release 3</b>
Start Einführung	Der Auftraggeber erhält eine Einführung in die Software	Sitzungsprotokoll zum Ende der Einführungsphase
Projektende	Der Auftraggeber erhält eine Einführung in die Software	Fertige Projektdokumentation, Abgeschlossene Testprotokolle <b>Release 4</b>
Abschlusspräsentation	Die Abschlusspräsentation ist durchgeführt worden.	-

Tabelle 4: Meilensteine, Quelle: Autoren

### B.2.3 Risikomanagement

Beim Risikomanagement werden die wichtigsten Risiken für das Projekt ermittelt und passende Gegenmassnahmen ausgearbeitet.

Risiko	Eintrittswahrsch.	Schaden
Falsche Zeiteinschätzung	70	80
Requirements nehmen zu / Requirements ändern sich	60	60
Entwicklerausfall	40	70
Unklare Schnittstellenspez.	40	70
Vernachlässigung Designprozess	20	60
Fehlende technische Kompetenz	20	90
Veränderung im Projektteam	10	70

Tabelle 5: Risikoanalyse, Quelle: Autoren

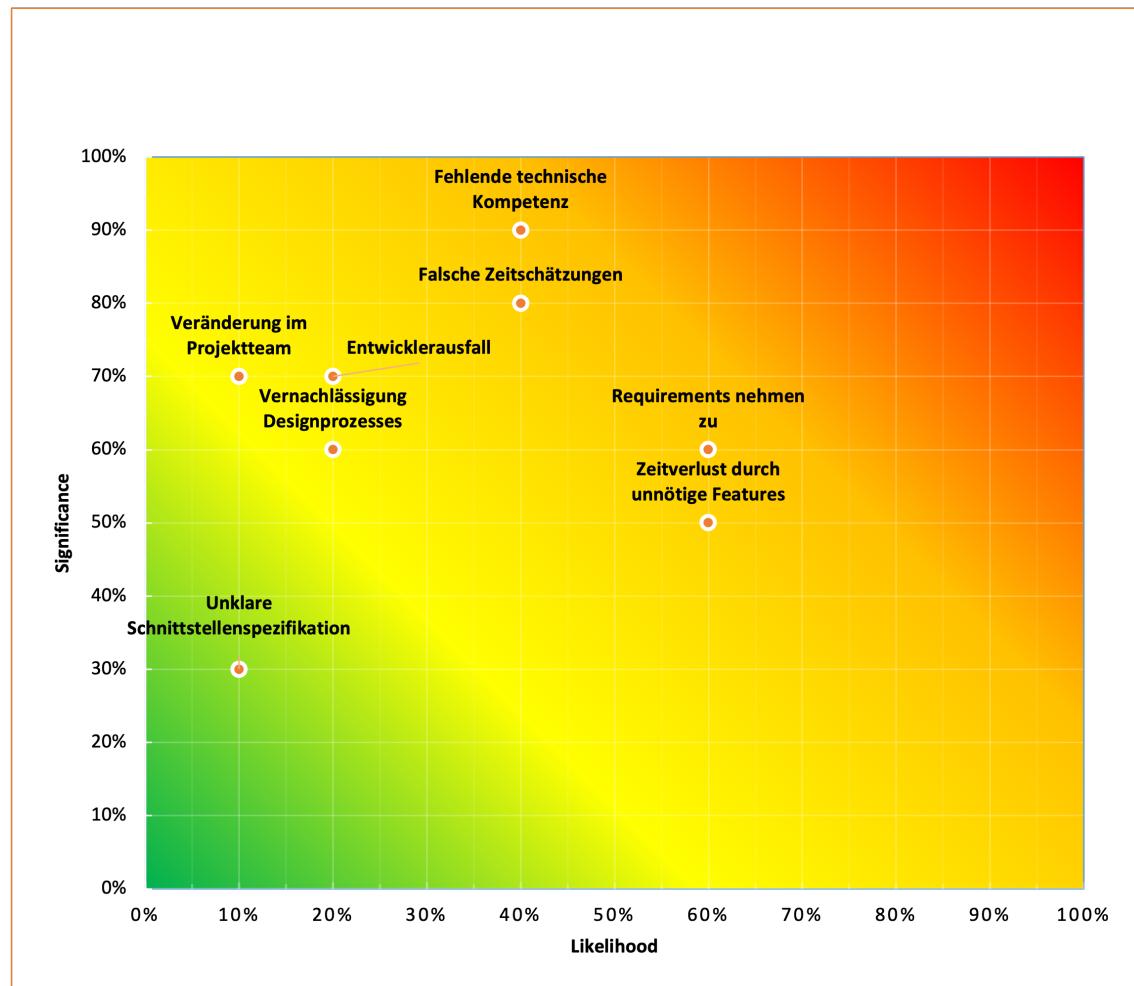


Abbildung 14: Risikomatrix,  
Quelle: Autoren

**Beschreibung** Basierend auf der Risikomatrix in Abbildung 14 müssen für die Risiken im rechten oberen Viertel Gegenmassnahmen erarbeitet werden. In diesem Viertel liegt allerdings nur das Risiko "Requirements nehmen zu". Aus diesem Grund werden hier noch weitere Risiken bearbeitet.

- Requirements nehmen zu
- Zeitverlust durch unnötige Features
- Falsche Zeiteinschätzung
- Fehlende technische Kompetenz

### Gegenmassnahmen

**Requirements nehmen zu** Um eine Veränderung der Requirements während des Projekt zu vermeiden, werden die Requirements in ständigem Kontakt mit den Auftraggebern erarbeitet und von diesen abgenommen.

**Zeitverlust durch unnötige Features** Um dies zu verhindern werden die entsprechenden User Stories definiert. Es werden dabei nur die Requirements berücksichtigt, welche beim Requirements Engineering erarbeitet und vom Auftraggeber abgenommen wurden.

**Falsche Zeiteinschätzung** Um eine bessere Zeiteinschätzung zu erlangen, wird auf das Wissen aus vorherigen Projekten zurückgegriffen. Basierend darauf kann die Planung genauer durchgeführt werden.

**Fehlende technische Kompetenz** Es werden Technologien verwendet, welche bereits bekannt sind. Zudem finden diese in vielen Projekten Anwendung, sodass auf das Wissen von erfahrenen Entwicklern zurückgegriffen werden kann.

Risiko	Eintrittswahrsch.	Schaden
Requirements nehmen zu / Requirements ändern sich	20	10
Zeitverlust unnötige Features	30	40
Falsche Zeiteinschätzung	30	80
Fehlende technische Kompetenz	20	20

Tabelle 6: Risikoanalyse nach Massnahmen, Quelle: Autoren

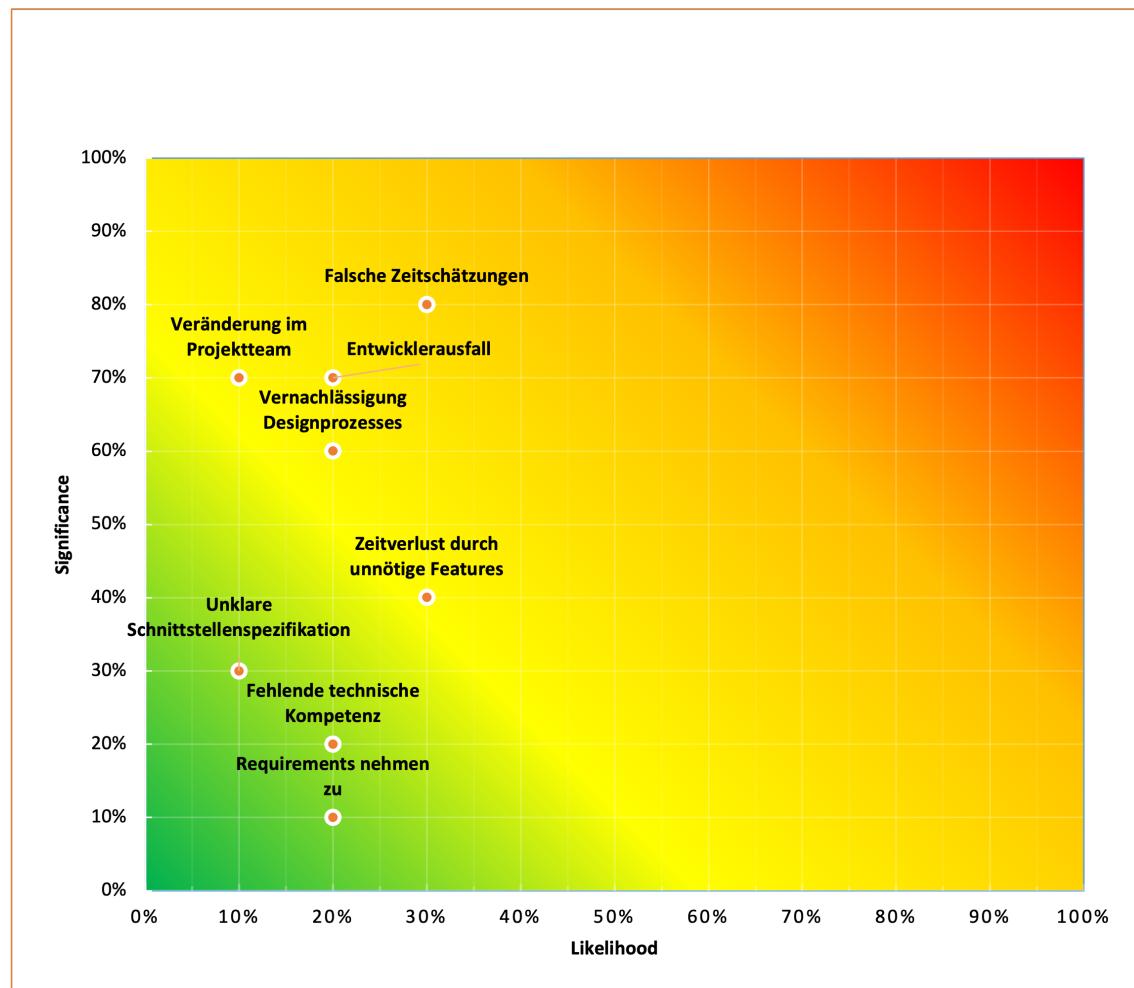


Abbildung 15: Risikomatrix nach Massnahmen,  
Quelle: Autoren

#### B.2.4 Definition of done

In jedem Sprint müssen die nachfolgenden Punkte zwingend erreicht werden, um ein potenziell auslieferbares Produkt zu erhalten:

- Review durchgeführt
- Akzeptanzkriterien erfüllt
- Unit Tests Grün
- CI/CD ohne Fehler
- keine kritischen Bugs
- Clean Code Guidelines eingehalten
- Dokumentation aktuell

### B.3 Projektunterstützung

#### B.3.1 Tools für Entwicklung, Test und Abnahme

**Entwicklungstools** Bei der Entwicklung des Projekts kommen folgende Programme zum Einsatz:

Typ	Tool	Version
IDE	InteliJ Ultimate	2020.1
IDE	Visual Studio Code	1.53.2
Versionsverwaltung	Git	2.27.0

Tabelle 7: Entwicklungstools, Quelle: Autoren

**Testtools** Beim Testing kommen folgende Tools zum Einsatz

Typ	Tool	Version
Unit Testing	JUnit	5.6.2
API Testing	Postman	7.36.0

Tabelle 8: Testtools, Quelle: Autoren

#### B.3.2 Konfigurationsmanagement

**Konfigurationseinheit** Bei diesem Projekt besteht eine Konfigurationseinheit aus mehreren Teilen. Dabei werden diese bei jedem Release aufgeführt. Zusätzlich dazu kommen noch die Reports der Automatisierten Tests, falls vorhanden auch der Systemtests.

- API
- Datenbank
- Webapplikation
- Dokument

Typ	Version
API	1.0.0
Datenbank	x
Webapplikation	x
Dokumentation	x

Tabelle 9: Konfigurationseinheit Release 1, Quelle: Autoren

#### Release 1

**Testprotokolle** Die gesamten Testprotokolle sind im Anhang A zu finden.

## B.4 Teststrategie und Drehbuch

### B.4.1 Teststrategie

Es handelt sich hier um die Entwicklung eines Prototypen. Die Priorität wird entsprechend gesetzt.

- Funktionalität vor Design
- Sicherheit vor Tempo
- Funktionalität vor Testing

Es wurde sich dabei bewusst auf diese Priorisierungen beschränkt. Für das Testing wird in erster Linie auf Automated Tests gesetzt. Hinzu kommen die Abnahmetests zu den jeweiligen Meilensteine. Diese werden durch ausgewählte Testpersonen ausgeführt.

**Automated Testing der REST-Schnittstelle** Zum Testen der REST-Schnittstelle wird dabei in erster Linie immer Postman genutzt. Es wird auf die darin enthaltene "automated Testing" Funktion gesetzt.

### B.4.2 Testdrehbuch

Wie oben genannt wird hauptsächlich auf Automated Testing gesetzt. Daher werden nur sehr wenige manuelle Tests durchgeführt. Die Tests gehen mit den gleichnamigen Meilensteinen einher. Nachfolgend werden diese inklusive den erhaltenen Resultate beschrieben.

Test Lernmodus Frage anzeigen	
Test Nr.	1
Beschreibung	Durch diesen Test wird die Lernfunktion sowie die Lernstatistik für Lernende manuell getestet.
Randbedingungen	Die Testperson hat einen bereits eingerichteten Account mit den für sie relevanten Fragen.
erwartete Resultate	Der Nutzer bekommt eine Frage inklusive den möglichen Antworten angezeigt.
Testperson	Oliver Werlen
Datum	01.10.2020
Testprotokoll	3

Tabelle 10: Test Lernmodus Frage anzeigen, Quelle: Autoren

## B.5 Bemerkungen

Zur Erstellung des Projektmanagementplans wurde die Vorlage der Hochschule Luzern verwendet.  
[HSLU, o.D. a]

## C System-Spezifikation

### C.1 Systemübersicht

#### C.1.1 Systemarchitektur



Abbildung 16: Systemarchitektur,  
Quelle: Autoren

### C.1.2 Kontextdiagramm

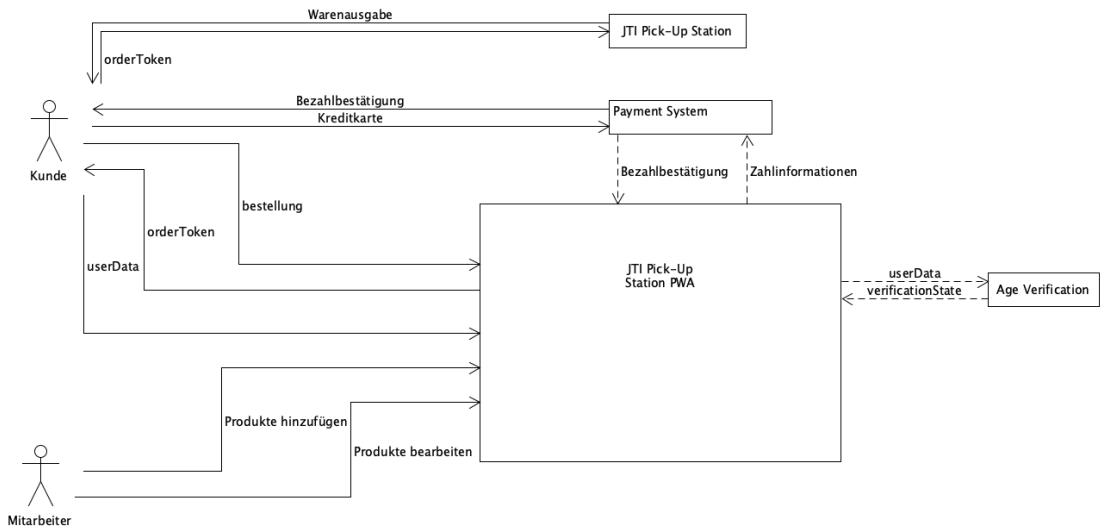


Abbildung 17: Kontextdiagramm,  
Quelle: Autoren

## C.2 Architektur und Designentscheide

Es wurde bei diesem Projekt auf eine REST-Architektur gesetzt. Die Web-Applikation wird als PWA umgesetzt.

### C.2.1 Modelle und Sichten

In diesem Projekt wird zwischen zwei verschiedenen Sichten unterschieden:

- **Kunde** Es handelt sich dabei um die Person, welche in der PWA Produkte bestellt und diese anschliessend abholt.
- **Administrator** Dem Administrator ist es möglich, Produkt hinzuzufügen, zu verändern oder auch zu löschen.
- **Programmierer:** Dieser konzipiert und realisiert die Applikation gemäss den Anforderungen des Auftraggebers.

### C.2.2 Daten (Mengengerüst und Strukturen)

**Datenbankschema** Das Datenbankschema wurde mittels Reverse Engineering mit MySQL Workbench erstellt und ist in der Abbildung 18 ersichtlich.

Abbildung 18: Datenbankschema, Quelle: Autoren

### C.2.3 Entwurfsentscheide

**Frontend**

**Technologien**

**Projektstruktur**

**Backend**

**Spring Boot** Für die Backendentwicklung wurde Spring Boot in der Version 2.3.4 genutzt.

**Datenbank**

**Konfigurationen**

**Frontend**

**Backend**

### C.3 Schnittstellen

#### C.3.1 Externe Schnittstellen

REST API

#### C.3.2 Wichtige interne Schnittstellen

Schnittstelle

Steckbrief

Einsatz, Abläufe, Voraussetzungen und Zusicherung

Aufbau und Konfiguration

Fehlerbehandlung

Qualitätsmerkmale

Entwurfsentscheidungen

Beispielverwendung

#### C.3.3 Benutzerschnittstellen

### C.4 Environment-Anforderungen

#### C.4.1 Hardware

Folgende Hardware wurde für diese Applikation verwendet und kann als ausreichend betrachtet werden:

#### C.4.2 Software

- Peter

## D Software Requirements Specification

### D.1 Zweck

Der Auftraggeber will durch die JTI Pickup Station einen neuen Absatzkanal zum Vertrieb seiner Produkte an Endkunden erstellen. Durch das Erstellen einer Softwarelösung soll es möglich sein, Kunden inklusive einer Altersverifikation zu erfassen. Auch ist die Umsetzung einer Kaufabwicklung sowie die Auswahl einer Pick-Up Station Teil dieses Projekts. Die Applikation wird dabei als PWA umgesetzt. Die Software soll dabei mit der physischen Pick-Up Station kompatibel sein. Die Umsetzung von diesen ist Teil von zwei weiteren Bachelorarbeiten an der Hochschule Luzern. "Verweis auf Aufgabenstellung"

#### D.1.1 Zielgruppe

Zur Zielgruppe dieser Software gehören Kunden und Kundinnen von Japan Tobacco International (JTI) auf der ganzen Welt. Die Software in Kombination mit den physischen Pick-Up Stations soll international eingesetzt werden.

#### D.1.2 Produktumfang

Der Umfang der Software beginnt bei der Registrierung der Nutzer. Hierbei wird eine bereits vorhandene Alterverifikation eingesetzt, um dies gesetzeskonform Umsetzen zu können. Im Onlineshop werden die verfügbaren Produkte von JTI gelistet. Der Nutzer kann diese Auswählen, anschliessend werden ihm alle Pick-Up Stations, in denen das Produkt verfügbar ist, angezeigt. Der Käufer kann die von ihm gewünschte Station auswählen. Anschliessend wird die Bezahlung per Kreditkarte durchgeführt. In anderen Projekten wurde dabei von JTI bereits ein bekannter Anbieter genutzt. In dieser Software wird darauf zurückgegriffen. Nach erfolgreicher Bezahlung wird ein Code auf dem Gerät des Nutzers gespeichert. Mit diesem kann an der gewünschten Pick-Up Station das bestellte Produkt abgeholt werden.

#### D.1.3 Definitionen

**Risiken** Das Risikomanagement wird im Projektmanagementplan in Kapitel B.2.3 detailliert aufgeführt.

#### D.1.4 Systemübersicht

Die Systemübersicht ist in der Systemspezifikation im Kapitel C zu finden.

#### D.1.5 Abhängigkeiten

Die Erfüllung der Requirements hängt von diversen Faktoren ab. Wesentlich dabei ist die Abhängigkeit von den Bachelorarbeiten der Studierenden an der Hochschule Luzern Technik und Architektur. Die hier vorhandenen Abhängigkeiten werden während der Realisierung möglichst minimiert.

## D.2 Spezifische Anforderungen

### D.2.1 Funktionale Anforderungen

F.1	Das System muss die Punkte in der von Google aufgestellten Core Progressive Web App checklist erfüllen. Sam Richard, 2020	Muss
F.2	Das System ist auf eine physische Pick-Up Station abgestimmt.	Muss
F.4	Das System bietet dem Anwender die Möglichkeit, sich zu registrieren.	Muss
F.5	Das System bietet die Möglichkeit, durch die Anbindung an eine 3rd Party, eine Altersverifikation durchzuführen.	Muss
F.6	Das System bietet dem Kunden die Möglichkeit, verschiedene Produkte zu bestellen.	Kann
F.7	Das System ermöglicht die Anbindung an einen bereits bekannten Bezahlungsdienst, um eine sichere Bezahlung zu garantieren.	Muss
F.9	Das System bietet dem Kunden die Möglichkeit, die für ihn nächstgelegene Station auswählen zu können.	Muss
F.10	Das System bietet dem Kunden die Möglichkeit, alle vorhandenen Pick-Up Stations anzuzeigen.	Muss
F.11	Das System bietet dem Kunden die Möglichkeit, seine beliebtesten Produkte direkt zu bestellen.	Kann
F.12	Das System bietet dem Dienstleister die Möglichkeit, einen aktuellen Warenbestand zu erhalten.	Kann
F.13	Das System bietet dem Dienstleister die Möglichkeit, bei zu geringem Warenbestand eine Benachrichtigung zu senden.	Muss
F.14	Das System bietet dem Betreiber die Möglichkeit, Artikel hinzuzufügen und Artikel zu bearbeiten	Kann

Tabelle 11: Funktionale Anforderungen, Quelle: Autoren

### D.2.2 Nicht funktionale Anforderungen

ID	Anforderung	Muss/Kann
L.1	Das System soll dem Kunden die Möglichkeit bieten, eine Bestellung mit 5 Klicks zu platzieren	Kann
L.2	Das System bietet die Möglichkeit, International eingesetzt zu werden.	Kann
L.3	Das System muss via HTTPS kommunizieren.	Muss
L.4	Das System muss durch einen modernen und sicheren Authentifizierungsmechanismus geschützt sein.	Muss
L.5	Das System bietet die Möglichkeit, durch die Verwendung von bewährten Programmervorgehen von einem externen Fachmann verstanden zu werden	Muss
L.6	Das System muss über eine CI/CD-Pipeline verfügen	Muss

Tabelle 12: Nicht Funktionale Anforderungen, Quelle: Autoren

### **D.3 Bemerkungen**

Als Grundstruktur für die SRS wurde eine Vorlage von Perforce genutzt. [Krüger, 2018] Als Basis dazu diente die IEEE Spezifikation 29148-2018. [Doran, 2018]

### **D.4 Unterschriften**

Mit der Unterschrift gilt die Software Requirement Specification als bestätigt.

Ort, Datum: \_\_\_\_\_

Visum: \_\_\_\_\_

## E Sitzungsprotokolle

Auf den nachfolgenden Seiten sind alle Protokolle von den durchgeführten Sitzungen ersichtlich.

### E.1 23.02.2021

Kick Off Meeting

#### E.1.1 Ordnungsaufruf

Eine Besprechung aller Projektbeteiligten fand online als Zoom-Meeting am 23.02.2021 um 15:00 Uhr statt.

#### E.1.2 Teilnehmer

Anwesende Mitglieder	Nicht anwesende Mitglieder
Meier Rene, Betreuungsperson Handschuh Michael, Betreuungsperson Oliver Werlen, Projektleiter Philipp Arnold, Elektrotechnik Elena Nujic, Auftraggeber Rolf Hohermut, Auftraggeber	-

Tabelle 13: Sitzungsprotokoll, Quelle: Autoren

#### E.1.3 Genehmigung des Protokolls

Es handelt sich hierbei um die erste Sitzung in diesem Projekt. Es ist noch kein Protokoll vorhanden.

#### E.1.4 Ankündigungen

Es handelt sich hierbei um das Kickoff Meeting. Als Ziel wird die Finalisierung der Aufgabenstellung genannt.

#### E.1.5 besprochene Punkte

##### Aufgabenstellung

- Register, Altersüberprüfung -> Bereits vorhanden, Shop (Brands, Produkt), Warenkorb, Bezahlung mit Kreditkarte, Beim Kauf definieren, wo abgeholt werden soll, reserviert in PickUp Station, An PickUp-> Mittels QR Code, Bestellung ausgeben
- Zahlung mit Twint, Kreditkarte, keine Nachnahme -> data trans, payrex
- Lösungen bereits in Onlineshop
- Postautomat als Beispiel
- Automat muss wissen, welche Artikel er noch hat
- Preis muss Variabel sein, Gratis Paket
- Website als Informationsquelle

### **Sitzungen**

- Abhängig von Projektphase, alle 2-3 Wochen
- Zwischenpräsentation von 20 Min, Resultate vorstellen, Fragen
- Schlusspräsentation 20-25 Minuten

### **Fragen zur Dokumentation**

- Benutzen von vorhandenen Texten aus WiPro?
- Auftrag Start WiPro auch in BAA?

### **E.1.6 Tagesordnung der nächsten Sitzung**

- Besprechung Anforderungen
- Rahmenplan, erste Doku Teile

### **E.1.7 Unterschriften**

Mit der Unterschrift gilt das Sitzungsprotokoll als bestätigt.

Ort, Datum: \_\_\_\_\_

Visum: \_\_\_\_\_

## E.2 04.03.2021

### E.2.1 Ordnungsaufruf

Eine Besprechung mit den Auftraggebern fand online als Microsoft-Teams-Meeting am 04.03.2021 um 15:00 Uhr statt.

### E.2.2 Teilnehmer

Anwesende Mitglieder	Nicht anwesende Mitglieder
Oliver Werlen, Projektleiter Philipp Arnold, Elektrotechnik Elena Nujic, Auftraggeber Rolf Hohermut, Auftraggeber	Meier Rene, Betreuungsperson Handschuh Michael, Betreuungsperson

Tabelle 14: Sitzungsprotokoll, Quelle: Autoren

### E.2.3 Genehmigung des Protokolls

Es handelt sich hierbei um die erste offiziell zu protokollierende Sitzung in diesem Projekt. Auf eine Genehmigung des Protokolls zum Kick-Off Meeting E.1 wird daher verzichtet.

### E.2.4 Ankündigungen

In diesem Meeting werden offene Punkte besprochen, welche in der Initialisierungsphase aufgetaucht sind.

### E.2.5 besprochene Punkte

#### Deployment

- Wo soll die Applikation laufen? Enterpriselab für Entwicklung ausreichend, evt. Deployment auf Umgebung von JTI
- Vorgabe vom Auftraggeber?

#### vorhandene Anbieter

- Anbieter für Altersverifikation? Winston-Camel Registrieren -> Swisscom, Sunrise
- Neue Lösung mittels -> Jumio
- Anbieter für Bezahlvorgang? Datatrans, Payment von Six Payment
- Kontakt zu Experten von JTI

#### Requirements

- Durchgehen, finalisieren, erweitern
- F.13 Im Kiosk im Aussenbereich -> Nachfüllanfrage an Kioskbetreiber -> Muss Features
- Ändern und hinzufügen von Produkten

### Probleme

- Probleme bei aktuellem Absatzkanal -> Wichtigsten Punkte gefunden, AVEC mit Ihrem Produkt reinnehmen.

### Kommunikation der Pick-Up Station

- Internet und Strom vorhanden
- Einlesen des Abholcodes mit RFID

### E.2.6 Tagesordnung der nächsten Sitzung

Die nächste Sitzung wird in zwei Wochen angesetzt.

### E.2.7 Unterschriften

Mit der Unterschrift gilt das Sitzungsprotokoll als bestätigt.

Ort, Datum: \_\_\_\_\_ Visum: \_\_\_\_\_

## E.3 11.03.2021

### E.3.1 Ordnungsaufruf

Eine Besprechung mit der Betreuungsperson fand online als Zoom-Meeting am 04.03.2021 um 13:00 Uhr statt.

### E.3.2 Teilnehmer

Anwesende Mitglieder	Nicht anwesende Mitglieder
Handschuh Michael, Betreuungsperson	Meier Rene, Betreuungsperson Oliver Werlen, Projektleiter Philipp Arnold, Elektrotechnik Elena Nujic, Auftraggeber Rolf Hohermut, Auftraggeber

Tabelle 15: Sitzungsprotokoll, Quelle: Autoren

### E.3.3 Genehmigung des Protokolls

Es handelt sich hierbei um die erste offiziell zu protokollierende Sitzung in diesem Projekt. Auf eine Genehmigung des Protokolls zum Kick-Off Meeting E.1 wird daher verzichtet.

### E.3.4 Ankündigungen

In diesem Meeting werden offene Punkte besprochen, welche in der Initialisierungsphase aufgetaucht sind.

### E.3.5 besprochene Punkte

#### Deployment

- Wo soll die Applikation laufen? Enterpriselab für Entwicklung ausreichend, evtl. Deployment auf Umgebung von JTI
- Vorgabe vom Auftraggeber?

### Allgemeine Punkte

- Schnittstelle spezifizieren mit ET und Maschinen evt. als Meilenstein
- Meilenstein Zwischenpräsentation
- Meilenstein Abschlusspräsentation
- Wieso NFC und nicht QR-Code
- Datenschutz bei jumio ->
- Schnittstelle 3rd Party Systeme in Risikoanalyse
- Datenschutz bei 3rd Party
- Doku im github Michael Handschuh freigeben

### E.3.6 Tagesordnung der nächsten Sitzung

Die nächste Sitzung mit dem Betreuer wird in 2 Wochen stattfinden.

### E.3.7 Unterschriften

Mit der Unterschrift gilt das Sitzungsprotokoll als bestätigt.

Ort, Datum: \_\_\_\_\_

Visum: \_\_\_\_\_

## E.4 18.03.2021

### E.4.1 Ordnungsaufruf

Eine Besprechung mit den Auftraggebern fand online als Microsoft-Teams-Meeting am 18.03.2021 um 13:00 Uhr statt.

### E.4.2 Teilnehmer

Anwesende Mitglieder	Nicht anwesende Mitglieder
Oliver Werlen, Projektleiter Philipp Arnold, Elektrotechnik Elena Nujic, Auftraggeber Rolf Hohermut, Auftraggeber	Meier Rene, Betreuungsperson Handschuh Michael, Betreuungsperson

Tabelle 16: Sitzungsprotokoll, Quelle: Autoren

### E.4.3 Genehmigung des Protokolls

Das Protokoll der letzten Sitzung wurde von allen Anwesenden bestätigt

### E.4.4 Ankündigungen

In diesem Meeting wird der Abschluss der Initialisierungsphase besprochen. Zusätzlich werden die offenen Punkte zu 5.2.1 besprochen und finalisiert.

**E.4.5 besprochene Punkte****Spezifikation Schnittstelle Abholung**

- Problem mit NFC/Bluetooth
- Vorstellen der Alternativen
- Besprechung Alternativen -> Alternative 1 wird bevorzugt, QR-Code auf Maschine
- Wenn keine Bestellung, dann auf Webpage
- Produkte, Liste von Produkten wird gesendet, jtiproducts.ch sind alle Produkte vorhanden
- Keine Policy von JTI vorhanden, jtiproducts als Referenz
- Statusmeldungen an Nachfüller

**Schnittstelle Produktbestand**

- Finalisierung mit Philipp Arnold individuell
- Übertragung mittels JSON Format

**E.4.6 Tagesordnung der nächsten Sitzung**

Die nächste Sitzung mit den Auftraggebern findet in zwei Wochen statt.

**E.4.7 Unterschriften**

Mit der Unterschrift gilt das Sitzungsprotokoll als bestätigt.

Ort, Datum: \_\_\_\_\_

Visum: \_\_\_\_\_

## F Rahmenpläne

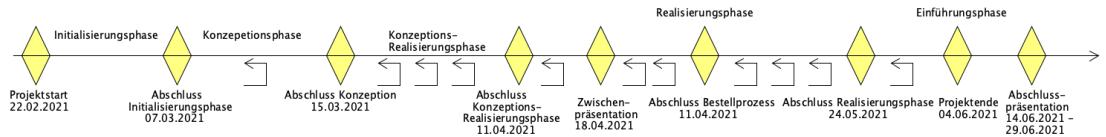


Abbildung 19: Rahmenplan Version 1,  
Quelle: Autoren

**G Originale Aufgabenstellung**

**H Wireframes mit Balsamiq Mockup**

**I API Documentation mit Swagger**