

HOCHSCHULE LUZERN

BACHELORARBEIT

JTI Pickup Station

Oliver Werlen

Betreut durch Rene Meier
xz

2. März 2021

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

1 Problem und Vision	3
2 Stand der Technik	4
3 Ideen und Konzepte	5
4 Methoden	6
5 Realisierung	7
6 Evaluation und Validation	8
7 Ausblick	9
8 Verzeichnisse	10
A Testprotokolle	15
A.1 Testprotokolle Schülermodus	15
B Projektmanagementplan	16
B.1 Projektorganisation	16
B.1.1 Organisationsplan, Rollen, Zuständigkeiten	16
B.1.2 Projektstrukturplan	17
B.2 Projektführung	18
B.2.1 Rahmenplan	18
B.2.2 Meilensteine	19
B.2.3 Risikomanagement	20
B.2.4 Definition of done	22
B.3 Projektunterstützung	23
B.3.1 Tools für Entwicklung, Test und Abnahme	23
B.3.2 Konfigurationsmanagement	23
B.4 Teststrategie und Drehbuch	24
B.4.1 Teststrategie	24
B.4.2 Testdrehbuch	24
B.5 Bemerkungen	24
C System-Spezifikation	25
C.1 Systemübersicht	25
C.1.1 Systemarchitektur	25
C.1.2 Kontextdiagramm	26
C.2 Architektur und Designentscheide	27
C.2.1 Modelle und Sichten	27
C.2.2 Daten (Mengengerüst und Strukturen)	27
C.2.3 Entwurfsentscheide	27
C.3 Schnittstellen	30
C.3.1 Externe Schnittstellen	30
C.3.2 Wichtige interne Schnittstellen	30
C.3.3 Benutzerschnittstellen	32
C.4 Environment-Anforderungen	36
C.4.1 Hardware	36
C.4.2 Software	36

D Software Requirements Specification	37
D.1 Zweck	37
D.1.1 Zielgruppe	37
D.1.2 Produktumfang	37
D.1.3 Definitionen	37
D.1.4 Systemübersicht	37
D.1.5 Abhängigkeiten	37
D.2 Spezifische Anforderungen	38
D.2.1 Funktionale Anforderungen	38
D.2.2 Nicht funktionale Anforderungen	38
D.3 Bemerkungen	38
D.4 Unterschriften	39
E Sitzungsprotokolle	40
E.1 23.02.2021	40
E.1.1 Ordnungsaufruf	40
E.1.2 Teilnehmer	40
E.1.3 Genehmigung des Protokolls	40
E.1.4 Ankündigungen	40
E.1.5 besprochene Punkte	40
E.1.6 Tagesordnung der nächsten Sitzung	41
E.1.7 Unterschriften	41
F Originale Aufgabenstellung	42
G Wireframes mit Balsamiq Mockup	42
H API Documentation mit Swagger	42

1 Problem und Vision

2 Stand der Technik

?? Here comes some text

3 Ideen und Konzepte

4 Methoden

5 Realisierung

6 Evaluation und Validation

7 Ausblick

8 Verzeichnisse

Abbildungsverzeichnis

1	Organigramm	16
2	Projektstrukturplan	17
3	SoDa Rahmenplan	18
4	Risikomatrix	20
5	RisikomatrixNach	22
6	Systemarchitektur	25
7	Kontextdiagramm	26
8	Datenbankschema	27
9	Projektstruktur vom Frontend	27
10	Komponentendiagramm des Frontends	28
11	Externe Packages für das Frontend	28
12	Klassendiagramm des CsvParsers	30
13	Klassendiagramm DTOStrategy Question	32
14	Packagediagram Backend	32
15	Lehrersicht der Applikation	34

Abkürzungsverzeichnis

SoDa Software Development Agile.....	16
JTI Japan Tobacco International.....	37
PWA Progressive Web App.....	37

Tabellenverzeichnis

1	Testprotokoll Test 1, Quelle: Autoren	15
2	Meilensteine, Quelle: Autoren	19
3	Risikoanalyse, Quelle: Autoren	20
4	Risikoanalyse nach Massnahmen, Quelle: Autoren	21
5	Entwicklungstools, Quelle: Autoren	23
6	Testtools, Quelle: Autoren	23
7	Konfigurationseinheit Release 1, Quelle: Autoren	23
8	Test Lernmodus Frage anzeigen, Quelle: Autoren	24
9	Funktionale Anforderungen, Quelle: Autoren	38
10	Nicht Funktionale Anforderungen, Quelle: Autoren	38
11	Sitzungsprotokoll, Quelle: Autoren	40

Literatur

- 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)
- 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)
- 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)
- Sam Richard, P. L. (2020). What makes a good Progressive Web App? Zugriff unter <https://web.dev/pwa-checklist/>. (24.02.2021)

A Testprotokolle

A.1 Testprotokolle Schülermodus

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

Tabelle 1: 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.

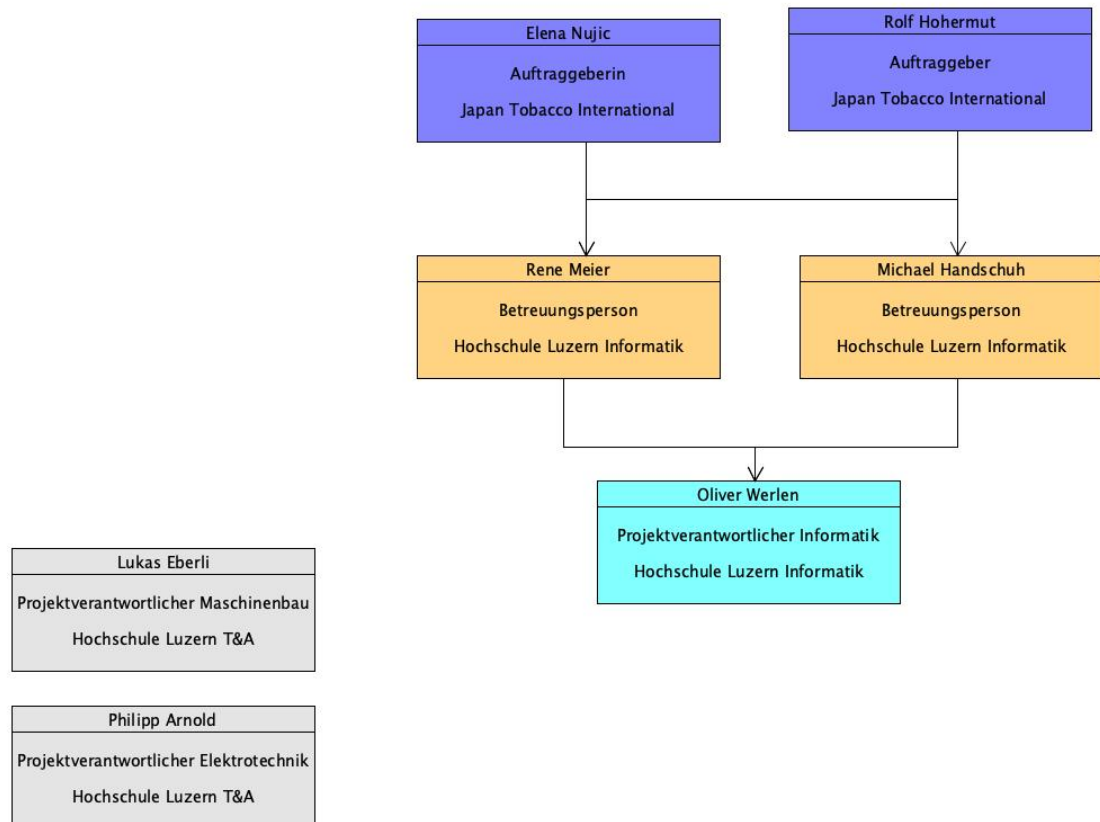


Abbildung 1: Organigramm, Quelle: Autoren

Rollen Im Projekt wird nach dem hybriden Projektmanagementvorgehen Software Development Agile (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

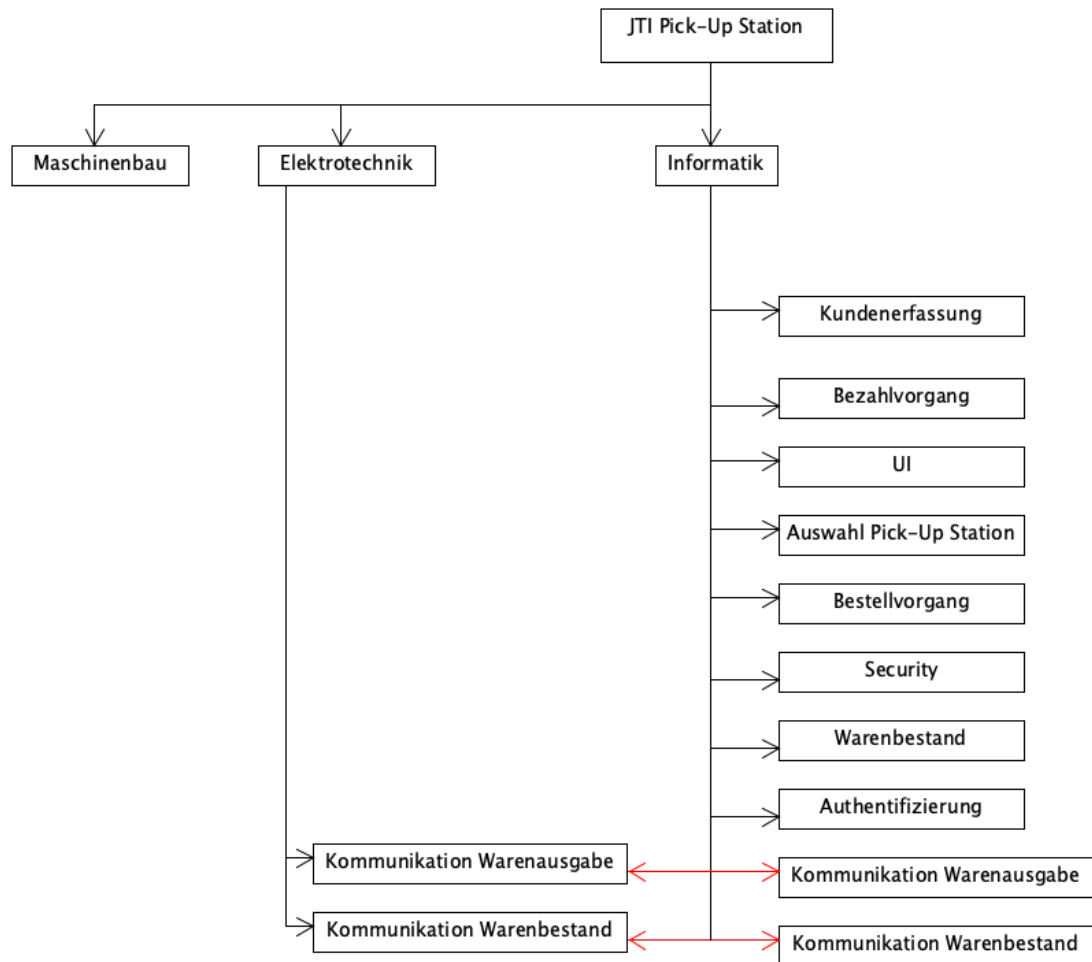


Abbildung 2: Projektstrukturplan, Quelle: Autoren

Beschreibung Im obigen Projektstrukturplan in Abbildung 2 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 eingezeichnet. 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.

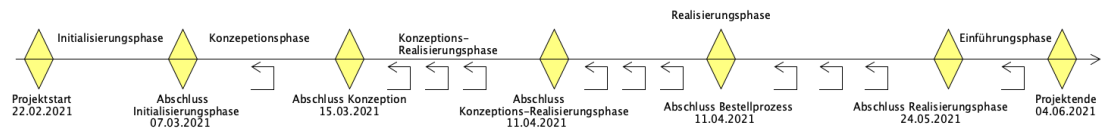


Abbildung 3: Rahmenplan, Quelle: Autoren

Der Rahmenplan wurde zu Beginn des Projekts grob dargestellt. Im Verlauf des Projekts kann dieser bei Bedarf angepasst werden.

B.2.2 Meilensteine

Wie in Abbildung 3 zu sehen gibt es insgesamt sieben Meilensteine. Diese werden in folgender Tabelle beschrieben sowie die nötigen Deliverables aufgezeigt.

Meilenstein	Beschreibung	Deliverables
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 ?? abgeschlossen.	abgenommene GUI-Prototypen, CI/CD Pipeline, Schnittstellenspezifikation, Kapitel SStand der Technik" Release 1
Abschluss Bestellprozess	Der Bestellprozess Artikel-auswahl, Artikel in Warenkorb, Artikel Bezahlen sowie die Kundenregistrierung sind umgesetzt und getestet.	Testprotokolle zu Abschluss Bestellprozess, Demo Bestellprozess, Release Bestellprozess Release 2
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 Release 3
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 Release 4

Tabelle 2: 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 techni- sche Kompetenz	20	90
Veränderung im Projektteam	10	70

Tabelle 3: Risikoanalyse, Quelle: Autoren

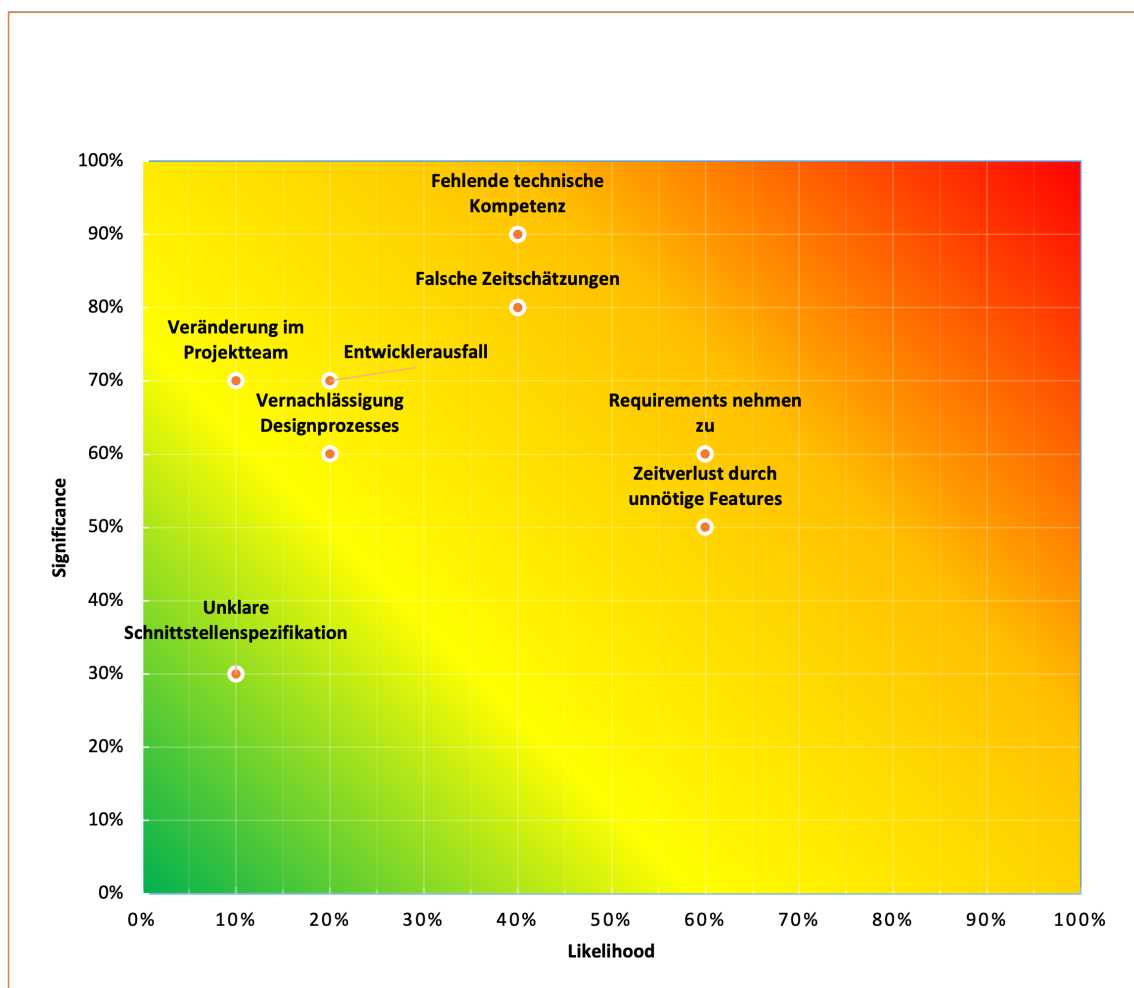


Abbildung 4: Risikomatrix, Quelle: Autoren

Beschreibung Basierend auf der Risikomatrix in Abbildung 4 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 4: Risikoanalyse nach Massnahmen, Quelle: Autoren

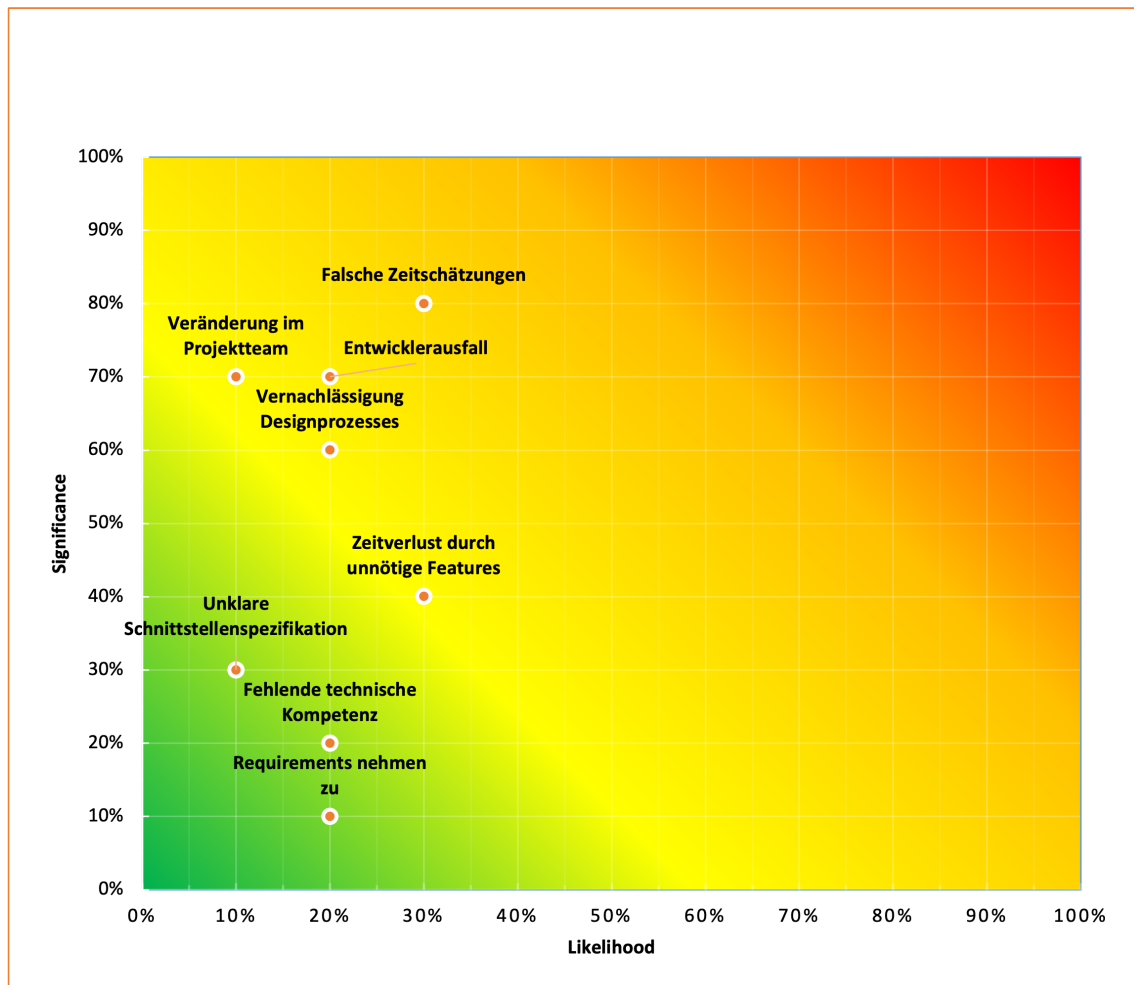


Abbildung 5: Risikomatrix nach Massnahmen, Quelle: Autoren

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 5: 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 6: 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 7: 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 wird bei diesem Projekt hauptsächlich auf Automated Testing gesetzt. Unit Tests werden dabei Integration Tests vorgezogen. Hierzu wird auf das bewährte JUnit Framework gesetzt. Es wird dabei das Test-First-Prinzip verwendet.

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	1

Tabelle 8: 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

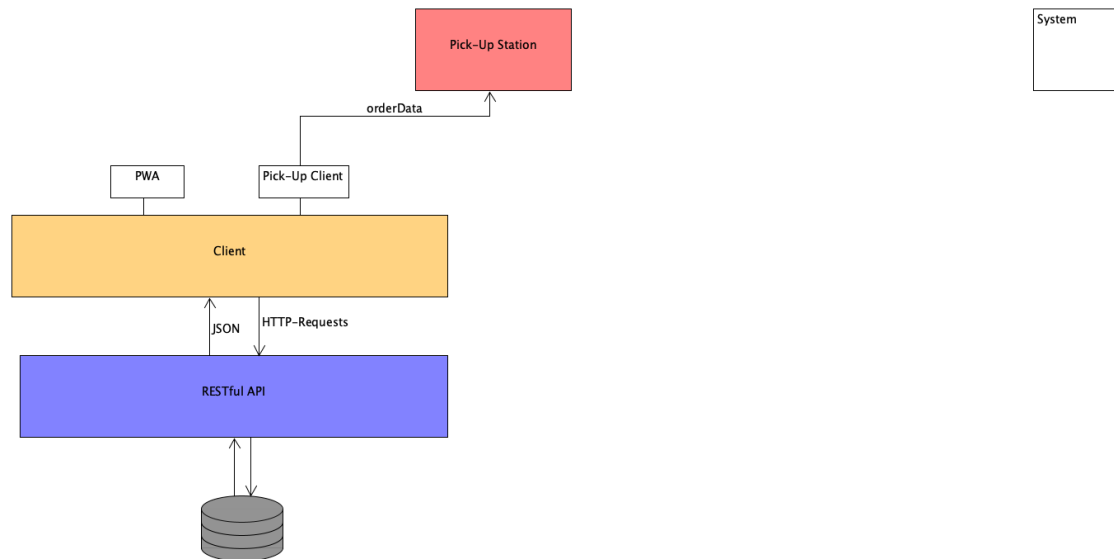


Abbildung 6: Systemarchitektur, Quelle: Autoren

C.1.2 Kontextdiagramm

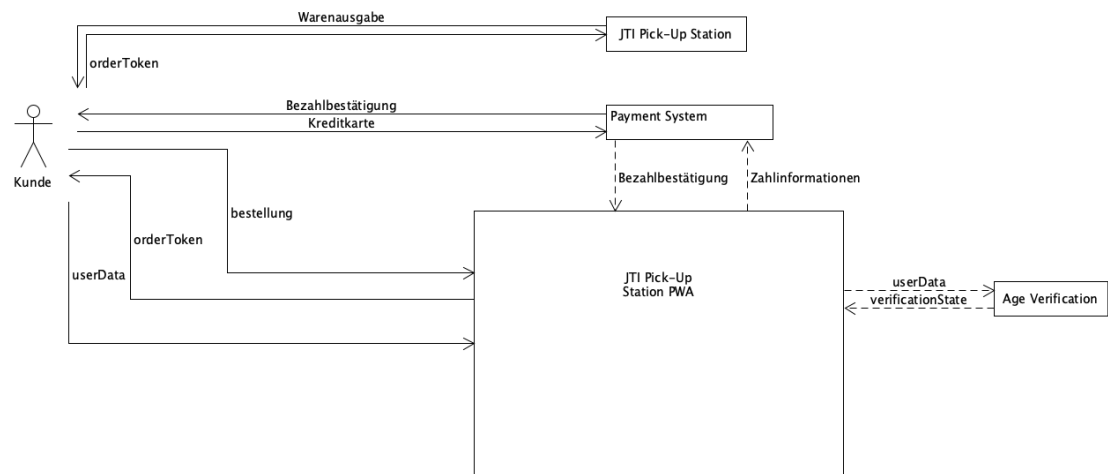


Abbildung 7: Kontextdiagramm, Quelle: Autoren

C.2 Architektur und Designentscheide

Da es sich bei dieser Applikation um eine Webapplikation handelt, wurde diese mit einer Client-Server-Architektur aufgebaut. Im Rahmen dieses Projektes musste das Frontend lediglich für den Browser realisiert werden. Damit diese Applikation bezüglich einer Weiterentwicklung bspw. auch mit einer App verwendbar sein soll, hat man sich aus Architekturgründen für eine REST-Schnittstelle entschieden.

C.2.1 Modelle und Sichten

In diesem Projekt wird zwischen vier verschiedenen Sichten unterschieden:

- **Schüler:** Dabei handelt es sich um einen Nutzer der Applikation mit eingeschränkten Rechten.
- **Lehrer:** Der Lehrer hat dieselben Rechte wie der Schüler, ist jedoch zusätzlich berechtigt weitere administrativen Tätigkeiten auf dem System durchzuführen.
- **Administrator:** Dieser kann Konfigurationen an der Applikation sowie am Server vornehmen. Dazu gehören bspw. die Benutzerverwaltung oder administrative Tätigkeiten an der Datenbank sowie Applikation.
- **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 erstellt und ist in der Abbildung 8 ersichtlich.

Abbildung 8: Datenbankschema, Quelle: Autoren

C.2.3 Entwurfsentscheide

Frontend

Technologien Das Frontend wurde mit React aufgebaut, wobei als Hilfsmittel **create-react-app** für die Entwicklung verwendet wurde. Betrieben wird die Applikation auf einem NGINX-Webserver mit der Version 1.16.0, welcher durch die **create-react-app** mitgeliefert wird.

Projektstruktur Die Projektstruktur ist wie folgt gegliedert:

- **Components:** Darin befinden sich alle Komponenten welche für die Applikation aufgebaut wurden.
 - **Frontpage:** Komponenten welche ohne Authentifizierung zugänglich sind.
 - **MainContent:** Komponenten die nur durch eine erfolgreiche Authentifizierung zugänglich sind.
 - * **Admin:** Komponenten für die Lehrer als auch Administratoren (bspw. Formulare, Im- und Exports etc.)
 - * **Exam:** Komponenten welche für die Prüfungen zuständig sind.
 - * **MainComponents:** Komponenten welche alle durch den Schüler zugänglich und die Mainfeatures ausmachen wie bspw. das Quiz, die Statistiken etc.
 - * **Navigation:** Der Inhalt der Sidebar als auch diverse Navigationshilfsmittel.

Abbildung 9: Projektstruktur vom Frontend

- * Utils: Diverse Hilfsmittel
- * MainContent: Dabei handelt es sich um die Hauptkomponente bei der bspw. der Router eingebaut ist und somit anhand der URL die entsprechende Komponenten rendert.
- Quiz: Hierbei handelt es sich um die Quiz-Komponente, welche für die Übungen verwendet wird.
- CSS: Darin sind alle Stylesheets welche nicht in den Komponenten direkt eingebaut wurden.
- images: Hier befinden sich alle Bilder welche für das Frontend verwendet wurden.
- Redux: Hier befinden sich alle Redux-Komponenten, welche alle Reducers, Actions und den Store beinhalten.
- AuthService. Hier wurden alle Services zusammengefasst, welche einerseits mit der Authentifizierung als auch mit der API vom Backend etwas zu tun haben.

In der Abbildung ?? ist ein vereinfachtes Komponentendiagramm ersichtlich, welches die selbst entwickelten Komponenten aufweist.

Abbildung 10: Komponentendiagramm des Frontends, Quelle: Autoren

Externe Packages In der Abbildung 11 sind alle externen Packages aufgelistet, welche für dieses Projekt verwendet wurden.

Abbildung 11: Externe Packages für das Frontend, Quelle: Autoren

Backend

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

Datenbank Als Datenbank wurde während der Entwicklung MariaDB in der Version 10.5 genutzt. Für die Auslieferung der Applikation wurde auf Postgre SQL gewechselt.

Konfigurationen

Frontend An folgenden Dateien können Konfigurationen am Frontend vorgenommen werden:

- nginx.config: Dort kann Webserver-Konfigurationen für den nginx vorgenommen werden. Falls ein anderer Webserver verwendet wird, ist diese Konfiguration überflüssig.
- package.json: Darin werden alle Abhängigkeiten und externen Packages verwaltet inklusive deren Versionierung.
- Dockerfile: Hier wird der Container-Build deklariert wobei hier auch diverse Einstellungen wie bspw. das Einbinden eines Zertifikats oder weitere Konfigurationen am Nginx vorgenommen werden können.

Backend Die Konfigurationen im Backend wurden mittels des application.properties-File gemacht.

```
spring.jpa.hibernate.ddl-auto=update

#for development with mariadb
spring.datasource.url=jdbc:mariadb://mariadb:3306/db_electrolernapp?
createDatabaseIfNotExist=true
spring.datasource.url=jdbc:mariadb://localhost:3306/db_electrolernapp?
spring.datasource.username=root
spring.datasource.password=electrolernapp2020

#for development with postgresql
spring.datasource.url=jdbc:postgresql://localhost:5432/db_electrolernapp?
createDatabaseIfNotExist=true
spring.datasource.username=elektro_app_user
spring.datasource.password=j^`k&hKbm8A!n2"]
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
spring.hateoas.use-hal-as-default-json-media-type=true
spring.jackson.time-zone: Europe/Paris
spring.http.multipart.max-request-size=50Mb

# CSV Configurations
spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=10MB

# App Properties
bezkoder.app.jwtSecret= bezKoderSecretKey
bezkoder.app.jwtExpirationMs= 86400000

server.contextPath=/api/v1
springfox.documentation.swagger.v2.path=/api-docs

#Mail properties
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=electrolernapp@gmail.com
spring.mail.password=electrolernapp2020
spring.mail.smtp.auth=true
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

C.3 Schnittstellen

C.3.1 Externe Schnittstellen

REST API Die REST-Schnittstelle wurde mit dem Tool „Swagger“ erstellt. Die Dokumentation ist im Anhang im Kapitel H aufzufinden.

C.3.2 Wichtige interne Schnittstellen

CsvService

Steckbrief Die Schnittstelle „CsvService“ dient dazu, Entities über eine CSV-Datei in die Datenbank zu migrieren.

Interaktionen

Abbildung 12: Klassendiagramm des CsvParsers, Quelle: Autoren

Operationen und Datenstrukturen

Einsatz, Abläufe, Voraussetzungen und Zusicherung

- Bei der Implementation einer Klasse mit der Verwendung dieser Schnittstelle ist es jeweils notwendig, die Repository-Klasse der betroffenen Entities, welche vom JpaRepository erben, zu injecten.
- Der Header der CSV-Datei muss von vornherein klar definiert werden und ist nicht dynamisch anpassbar.

Aufbau und Konfiguration Keine zusätzlichen Informationen.

Fehlerbehandlung Zur Fehlerbehandlung werden Runtime Exceptions geworfen, wenn eine CSV-Datei nicht eingelesen werden kann. Es können aber auch ResourceNotFoundExceptions auftreten, wenn Entities Abhängigkeiten zu anderen Entities haben, die jedoch nicht gefunden werden. Wichtig ist, dass die Exceptions auf jeden Fall geworfen werden, damit der CsvController den Clients eine entsprechende Rückmeldung geben kann und den Prozess beendet.

Qualitätsmerkmale Keine zusätzlichen Informationen.

Entwurfsentscheidungen

- Die Schnittstelle besitzt eine default-Methode, um den Typ der Datei zu überprüfen.
- Diese Schnittstelle wurde darauf ausgelegt, lediglich mit CSV-Dateien zu arbeiten. Für andere Dateitypen sollte eine neue Schnittstelle definiert werden.

Beispielverwendung

```
//Injection im CsvController
@Autowired
private CsvCategoryService csvCategoryService;

//Einsatz des CsvControllers
if (csvCategoryService.hasCSVFormat(categoryFile)) {
    try {
```

```
        csvCategoryService.saveNewEntities(categoryFile);

        categoryMessage = "Uploaded the file successfully: " +
            categoryFile.getOriginalFilename();
        messageList.add(new MessageResponse(categoryMessage));
    } catch (Exception e) {
        categoryMessage = "Could not upload the file: " +
            categoryFile.getOriginalFilename() +
            "! " + e.getMessage();
        LOG.error(categoryMessage);
        e.printStackTrace();
        messageList.add(new MessageResponse(categoryMessage));
        return ResponseEntity.status(HttpStatus.EXPECTATION_FAILED)
            .body(messageList);
    }
}
```

DTOParser

Steckbrief Als interne Schnittstelle kommt die DTOParserStrategy bei der Übersetzung von Data Transfer Objects zu Entities zum Einsatz. Diese befindet sich in der Version 1.0.0.

Interaktionen

Abbildung 13: Klassendiagramm DTOStrategy Question, Quelle: Autoren

Operationen und Datenstrukturen

Einsatz, Abläufe, Voraussetzungen und Zusicherung

- Bevor die Schnittstelle verwendet werden kann, muss der dazugehörige Service mittels Auto-wired in die betreffende Klassen injected werden.
- Die eingegebene Id muss mit einem der Datenbankobjekte übereinstimmen.

Aufbau und Konfiguration Keine zusätzlichen Informationen.

Fehlerbehandlung Zur Fehlerbehandlung werden Runtime Exceptions geworfen, wenn ein Element beim Parsen nicht verfügbar ist.

Qualitätsmerkmale Keine zusätzlichen Informationen.

Entwurfsentscheidungen

- Die Schnittstelle wurde mit Generics umgesetzt, um sie für alle Parser verwenden zu können.
- Mit den hier vorgegebenen Interfaces können beliebige Typen mitgegeben und auch zurückgegeben werden.

Abbildung 14: Packagediagram Backend, Quelle: Autoren

Beispielverwendung

```
//Parsen von Listen
dtoParserQuestion.generateDTOsFromObjects(questions);

//Parsen von einzelнем Element
dtoParserQuestion.generateDTOFromObject(question.getId());

//Parsen von Objekt zu DTO
Question question = dtoParserQuestion.generateObjectFromDTO(questionDTO);
```

C.3.3 Benutzerschnittstellen

Als Benutzerschnittstelle fungiert hierbei lediglich das Frontend. Hierfür werden zwei Bereiche unterschieden:

- Öffentlicher Bereich: In diesem Bereich erscheint der Benutzer beim Aufruf der Website und ist für jedermann zugänglich.
- Privater Bereich: In diesem Bereich kann sich ein Benutzer lediglich befinden, wenn dieser sich erfolgreich authentifiziert hat.

Öffentlicher Bereich In diesem Bereich steht den Benutzern folgende drei Formulare zur Verfügung:

- Login: Hier kann sich ein Benutzer mit seinem Benutzernamen sowie seinem Passwort einloggen. Von hier aus kann dieser auf den Reiter „Passwort zurücksetzen“ wechseln.
- Passwort zurücksetzen: Hier kann der Benutzer seine E-Mail-Adresse eingeben um sein Passwort zurücksetzen zu lassen.
- Passwort neu setzen: Hier muss der Benutzer das neue Passwort zwei Mal eingeben. Dieses Formular kann lediglich mit einem gültigen Token im Backend abgearbeitet werden.

Privater Bereich In diesem Bereich wird zwischen der Schülersicht sowie der Lehrersicht unterschieden. Der Lehrer besitzt dieselbe Sicht wie der Schüler. Es stehen ihm aber auch noch zusätzliche Features zur Verfügung. Die Sichten beziehen sich dabei auf die Abbildung 15.

Schülersicht Dem Schüler stehende folgende Sichten zur Verfügung:

1. Home: Hier kann der Benutzer jederzeit auf die Startseite der Applikation navigieren.
2. Übungen: Darin kann sich der Benutzer durch die Kategorien sowie Übungssets navigieren und ein solches starten.
3. Prüfungen: In der Prüfungsübersicht kann der Benutzer seine anstehenden als auch absolvierten Prüfungen betrachten.
4. Statistiken: Hier kann der Benutzer seine eigenen Statistiken in Bezug auf alle Kategorien als auch auf die einzelnen Übungssets betrachten.
12. Profil: Hier kann man das eigene Profil betrachten, welches aus dem Benutzernamen und der E-Mail-Adresse besteht.

Abbildung 15: Lehrersicht der Applikation, Quelle: Autoren

Lehrersicht Während dem Lehrer dieselben Sichten wie des Schülers zur Verfügung stehen, besitzt dieser weitere welche zur Administration dienen.

5. Benutzerverwaltung: Bei der Benutzerverwaltung besteht die Möglichkeit entweder einen einzelnen Benutzer über ein Registrierungsformular zu erstellen oder anhand einer CSV-Datei mehrere Benutzer zu importieren.
6. Prüfung erstellen: Hier kann anhand eines Formulars eine Prüfung erstellt werden, in dem man mehrere Fragen in eine Liste zusammenfügt.
7. Fragen erstellen: In diesem Formular lassen sich neue Fragen erstellen und einem Übungsset zuweisen.
8. Klasse erstellen: Hier kann man eine Klasse erstellen, indem man diese einer Bildungsinstitution zuweist und entsprechende Schüler hinzufügt.
9. Klasse bearbeiten: Dieses Formular sieht genau gleich aus wie das bei der Erstellung der Klasse, wohingegen hier über die Wahl der Bildungsinstitution und der entsprechenden Klasse diese danach bearbeitet werden kann.
10. Bildungsinstitution erstellen: Hiermit wird eine neue Bildungsinstitution erstellt.
11. Daten exportieren: Hier lassen sich entsprechende Daten exportieren. Aktuell stehen Frage,, Benutzer, Prüfungen und Prüfungsergebnisse als Export zur Verfügung.

C.4 Environment-Anforderungen

Da keine ausgiebige Evaluation der Hardware- als auch Softwareplattform vorgenommen wurde, wird hier auf die Umgebung eingegangen, auf der die Applikation aufgebaut wurde und somit bestätigt werden kann, dass anhand dieser Anforderungen ein Betrieb der Applikation möglich ist.

C.4.1 Hardware

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

- CPU: Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
- RAM: 4GB

C.4.2 Software

Die ganze Applikation läuft auf einer virtuellen Maschine, auf der folgendes Betriebssystem inklusive Packages installiert sind:

- Ubuntu v. 20.04.01
- Docker v. 19.03.8

Des Weiteren musste ein SSH-Key mittels dem „ssh-keygen“ generiert werden, damit das Deployment sichergestellt werden konnte. Die SSH-Packages sind jedoch im Ubuntu bereits integriert.

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 Progressive Web App (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 Bezahlndienst, um eine sichere Bezahlung zu garantieren.	Muss
F.8	Das System bietet dem Kunden die Möglichkeit, verschiedene Produkte zu bestellen.	Kann
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	Kann

Tabelle 9: 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 Programmierverfahren von einem externen Fachmann verstanden zu werden	Muss

Tabelle 10: 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 11: 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: _____

F **Originale Aufgabenstellung**

G **Wireframes mit Balsamiq Mockup**

H **API Documentation mit Swagger**