

HOCHSCHULE LUZERN

BACHELORARBEIT

# JTI Pickup Station

*Oliver Werlen*

Betreut durch Rene Meier  
xz

23. Februar 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:  
Eingangvisum 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>3</b>
<b>2 Stand der Technik</b>	<b>4</b>
<b>3 Ideen und Konzepte</b>	<b>5</b>
<b>4 Methoden</b>	<b>6</b>
<b>5 Realisierung</b>	<b>7</b>
<b>6 Evaluation und Validation</b>	<b>8</b>
<b>7 Ausblick</b>	<b>9</b>
<b>8 Verzeichnisse</b>	<b>10</b>
<b>A Testprotokolle</b>	<b>15</b>
A.1 Testprotokolle Schülermodus . . . . .	15
<b>B Projektmanagementplan</b>	<b>16</b>
B.1 Projektorganisation . . . . .	16
B.1.1 Organisationsplan, Rollen, Zuständigkeiten . . . . .	16
B.1.2 Projektstrukturplan . . . . .	16
B.2 Projektführung . . . . .	17
B.2.1 Rahmenplan . . . . .	17
B.2.2 Meilensteine . . . . .	18
B.2.3 Risikomanagement . . . . .	19
B.2.4 Definition of done . . . . .	20
B.3 Projektunterstützung . . . . .	21
B.3.1 Tools für Entwicklung, Test und Abnahme . . . . .	21
B.3.2 Konfigurationsmanagement . . . . .	21
B.4 Teststrategie und Drehbuch . . . . .	22
B.4.1 Teststrategie . . . . .	22
B.4.2 Testdrehbuch . . . . .	22
<b>C System-Spezifikation</b>	<b>27</b>
C.1 Systemübersicht . . . . .	27
C.1.1 Systemarchitektur . . . . .	27
C.1.2 Kontextdiagramm . . . . .	28
C.2 Architektur und Designentscheide . . . . .	29
C.2.1 Modelle und Sichten . . . . .	29
C.2.2 Daten (Mengengerüst und Strukturen) . . . . .	29
C.2.3 Entwurfsentscheide . . . . .	29
C.3 Schnittstellen . . . . .	32
C.3.1 Externe Schnittstellen . . . . .	32
C.3.2 Wichtige interne Schnittstellen . . . . .	32
C.3.3 Benutzerschnittstellen . . . . .	34
C.4 Environment-Anforderungen . . . . .	38
C.4.1 Hardware . . . . .	38
C.4.2 Software . . . . .	38

<b>D Anforderungen</b>	<b>39</b>
D.1 Einführung . . . . .	39
D.2 Einzelanforderungen . . . . .	39
D.2.1 Funktionen . . . . .	39
D.2.2 Anforderungen an die Benutzbarkeit . . . . .	39
D.2.3 Anforderungen an die Leistungsfähigkeit . . . . .	39
D.2.4 Weitere Qualitätsmerkmale . . . . .	39
D.2.5 Wartungs- und Supportinformationen . . . . .	39
D.2.6 Weitere Anforderungen . . . . .	40
<b>E Sitzungsprotokolle</b>	<b>41</b>
E.1 14.09.2020 . . . . .	41
E.1.1 Ordnungsaufruf . . . . .	41
E.1.2 Teilnehmer . . . . .	41
E.1.3 Genehmigung des Protokolls . . . . .	41
E.1.4 Ankündigungen . . . . .	41
E.1.5 besprochene Punkte . . . . .	41
E.1.6 Tagesordnung der nächsten Sitzung . . . . .	42
E.1.7 Unterschriften . . . . .	42
<b>F Originale Aufgabenstellung</b>	<b>43</b>
<b>G Wireframes mit Balsamiq Mockup</b>	<b>43</b>
<b>H API Documentation mit Swagger</b>	<b>43</b>

## 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 . . . . .	16
3	SoDa Rahmenplan . . . . .	17
4	SoDa Rahmenplan überarbeitet . . . . .	17
5	Risikomatrix . . . . .	19
6	RisikomatrixNach . . . . .	20
7	Systemarchitektur . . . . .	27
8	Kontextdiagramm . . . . .	28
9	Datenbankschema . . . . .	29
10	Projektstruktur vom Frontend . . . . .	29
11	Komponentendiagramm des Frontends . . . . .	30
12	Externe Packages für das Frontend . . . . .	30
13	Klassendiagramm des CsvParsers . . . . .	32
14	Klassendiagramm DTOStrategy Question . . . . .	34
15	Packagediagram Backend . . . . .	34
16	Lehrersicht der Applikation . . . . .	36



## Abkürzungsverzeichnis



## Tabellenverzeichnis

1	Testprotokoll Test 1, Quelle: Autoren . . . . .	15
2	Meilensteine, Quelle: Autoren . . . . .	18
3	Risikoanalyse, Quelle: Autoren . . . . .	19
4	Risikoanalyse nach Massnahmen, Quelle: Autoren . . . . .	20
5	Entwicklungstools, Quelle: Autoren . . . . .	21
6	Testtools, Quelle: Autoren . . . . .	21
7	Konfigurationseinheit Release 1, Quelle: Autoren . . . . .	21
8	Test Lernmodus Frage anzeigen, Quelle: Autoren . . . . .	22
9	Test Lernmodus Frage anzeigen, Quelle: Autoren . . . . .	22
10	Test Lernmodus Frage korrigieren, Quelle: Autoren . . . . .	23
11	Test Lernmodus Statistik anzeigen, Quelle: Autoren . . . . .	23
12	Test Registrierung, Quelle: Autoren . . . . .	23
13	Test Einloggen, Quelle: Autoren . . . . .	23
14	Test Zugriff geschützte Ressource, Quelle: Autoren . . . . .	24
15	Test Aufgabenstellung hinzufügen, Quelle: Autoren . . . . .	24
16	Test Prüfung erstellen, Quelle: Autoren . . . . .	24
17	Test Analysieren und bearbeiten einer abgeschlossenen Prüfung, Quelle: Autoren .	25
18	Test Absolvieren einer Prüfung nur während bestimmtem Zeitpunkt, Quelle: Autoren	25
19	Test Absolvieren einer Prüfung ohne Berechtigungen, Quelle: Autoren . . . . .	25
20	Test Migration von Daten, Quelle: Autoren . . . . .	26
21	Anforderungen, Quelle: Autoren . . . . .	39
22	Anforderungen, Quelle: Autoren . . . . .	39
23	Anforderungen, Quelle: Autoren . . . . .	39
24	Anforderungen, Quelle: Autoren . . . . .	39
25	Anforderungen, Quelle: Autoren . . . . .	39
26	Anforderungen, Quelle: Autoren . . . . .	40
27	Sitzungsprotokoll, Quelle: Autoren . . . . .	41

## Literatur

Facebook. (o.D.). react quiz component. Zugriff unter <https://github.com/facebook/create-react-app>. (10.09.2020)

## 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 1: Testprotokoll Test 1, Quelle: Autoren

## B Projektmanagementplan

### B.1 Projektorganisation

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

Abbildung 1: OrganigrammQuelle: Autoren

**Rollen**

#### B.1.2 Projektstrukturplan

Abbildung 2: Projektstrukturplan,Quelle: Autoren

**Beschreibung** Im Projektstrukturplan in Abbildung 2 wird das Projekt in 3 grobe Teile geteilt. Sie sind dabei angepasst an die Grobplanung und decken die Anforderungen ab.

## **B.2 Projektführung**

### **B.2.1 Rahmenplan**

Im untenstehenden Rahmenplan wird mittels Zeitstrahl eine Grobplanung dargestellt.

Abbildung 3: Rahmenplan,Quelle: Autoren

Nach 7 absolvierten Sprints wurde der Rahmenplan angepasst.

Abbildung 4: Rahmenplan überarbeitet,Quelle: Autoren

### 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	Bei Meilenstein eins wird das Kickoff-Meeting mit allen Projektteilnehmern durchgeführt.	finale Aufgabenstellung
Start Umsetzung	Bei Meilenstein zwei wird vom klassischen Projektmanagement zum agilen Projektmanagement übergegangen. Dazu muss die Initialisierungsphase abgeschlossen sein	Projektmanagementplan, Systemspezifikation, Anforderungsliste
Abschluss Systemkontext	Zu diesem Zeitpunkt ist alles bereit, um mit der Entwicklung zu beginnen. Es wurden bereits erste GUI Entwürfe erarbeitet sowie die Systemarchitektur definiert.	CI/CD Umgebung eingerichtet, GUI-Prototyp, UML-Diagramme <b>Release 1</b>
Abschluss Schülermodus	Der Entwicklung des Schülermodus ist abgeschlossen. Es können vorgegebene Fragen beantwortet werden sowie eine Statistik zu bisher Gelerntem eingesehen werden.	Testprotokolle zu Schülermodus, Demo Schülermodus, Release Schülermodus <b>Release 2</b>
Abschluss Lehrermodus	Der Entwicklung des Lehrermodus ist abgeschlossen. Es können Fragen und Prüfungen erstellt und verteilt werden. Zusätzlich sind die Fragen aus der alten Applikation integriert.	Testprotokolle zu Lehrermodus, Testprotokolle Prüfungsmodus, Integration alte Daten, Demo verschiedene Modis, Release Lehrermodus <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>

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	20	70
Unklare Spezifikationen	10	30
Vernachlässigung Designprozess	20	60
Zeitverlust unnötige Features	60	50
Fehlende techni- sche Kenntnisse	40	90

Tabelle 3: Risikoanalyse, Quelle: Autoren

Abbildung 5: Risikomatrix, Quelle: Autoren

**Beschreibung** Basierend auf der Risikomatrix in Abbildung 5 müssen für die Risiken im rechten oberen Viertel Gegenmassnahmen erarbeitet werden. Dabei handelt es sich um die folgenden Risiken:

- Falsche Zeiteinschätzung
- Requirements nehmen zu/Veränderung der Requirements
- Zeitverlust durch unnötige Features
- Fehlende technische Kenntnisse

#### Gegenmassnahmen

**Falsche Zeiteinschätzung** Um das Risiko einer falschen Zeiteinschätzung zu minimieren wird bei der Planung auf bestehende, erfolgreich abgeschlossene Projekte zurückgegriffen. Basierend auf diesen wird die Zeitplanung durchgeführt.

**Requirements nehmen zu/Veränderung der Requirements** Die Requirements werden fortlaufend im Product Backlog überprüft. Die Aufgabenstellung dient dabei als Basis. Mittels vom Product Owner abgesegneten Akzeptanzkriterien ist der Umfang klar abgegrenzt.

**Zeitverlust durch unnötige Features** Durch den Product Backlog sowie den Sprint Backlog mit klaren Beschreibungen sowie Akzeptanzkriterien der Issues sind dem Entwickler zu jedem Zeitpunkt die zu bearbeitenden Punkte klar.

**Fehlende technische Kenntnisse** Beim Projekt wird auf vielgenutzte Technologien mit grosser Community gesetzt.

Risiko	Eintrittswahrsch.	Schaden
Falsche Zeiteinschätzung	30	80
Requirements nehmen zu / Requirements ändern sich	20	10
Zeitverlust unnötige Features	30	50
Fehlende technische Kenntnisse	40	40

Tabelle 4: Risikoanalyse nach Massnahmen, Quelle: Autoren

Abbildung 6: 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	Webstorms	2020.2
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	1.0.0
Webapplikation	1.0.0
Dokumentation	0.9

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 Unirest sowie JUnit verwendet.

### 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	Frederico Fischer
Datum	01.10.2020
Testprotokoll	1

Tabelle 8: Test Lernmodus Frage anzeigen, Quelle: Autoren

Test Lernmodus Frage beantworten	
Test Nr.	2
Beschreibung	Durch diesen Test wird die Beantwortung von Fragen getestet.
Randbedingungen	Die Testperson hat einen bereits eingerichteten Account mit für sie relevanten Fragen. Der Test Nr. 1 ist erfolgreich verlaufen
erwartete Resultate	Der Nutzer kann eine Fragen beantworten und gelangt direkt zur nächsten.
Testperson	Frederico Fischer
Datum	01.10.2020
Testprotokoll	??

Tabelle 9: Test Lernmodus Frage anzeigen, Quelle: Autoren

Test Lernmodus Frage korrigieren	
Test Nr.	3
Beschreibung	Durch diesen Test wird die Korrektur von Fragen getestet.
Randbedingungen	Die Testperson hat einen bereits eingerichteten Account mit für sie relevanten Fragen. Die Tests Nr. 1 und 2 sind erfolgreich verlaufen
erwartete Resultate	Bei der Beantwortung der Frage wird dem Nutzer angezeigt, ob die angewählte Lösung korrekt war. Falls nicht, wird die richtige Lösung angezeigt.
Testperson	Frederico Fischer
Datum	01.10.2020
Testprotokoll	??

Tabelle 10: Test Lernmodus Frage korrigieren, Quelle: Autoren

Test Lernmodus Statistik zu User anzeigen	
Test Nr.	4
Beschreibung	Durch diesen Test wird die Anzeige einer Userstatistik getestet.
Randbedingungen	Es sind bereits 3 Statistiken zu unterschiedlichen Fragen in der Datenbank vorhanden.
erwartete Resultate	Dem Nutzer wird seine Nutzerstatistik bezogen auf die beantworteten Fragen angezeigt.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 11: Test Lernmodus Statistik anzeigen, Quelle: Autoren

Test Registrierung	
Test Nr.	5
Beschreibung	Durch diesen Test wird die Registrierung eines neuen Nutzers getestet
Randbedingungen	Ein Nutzer mit diesem Benutzernamen ist noch nicht erstellt worden.
erwartete Resultate	Auf dem Bildschirm wird die Meldung User created successfully angezeigt.
Testperson	Frederico Fischer
Datum	01.10.2020
Testprotokoll	??

Tabelle 12: Test Registrierung, Quelle: Autoren

Test Login	
Test Nr.	6
Beschreibung	Durch diesen Test wird das Login eines Nutzers getestet
Randbedingungen	Ein Nutzer mit dem Benutzernamen «peterpan» und dem Passwort «peterpan» ist erfolgreich erstellt worden.
erwartete Resultate	Der Nutzer wird erfolgreich angemeldet. Im Userprofil wird der korrekte Nutzer angezeigt.
Testperson	Frederico Fischer
Datum	01.10.2020
Testprotokoll	??

Tabelle 13: Test Einloggen, Quelle: Autoren

Test Zugriff geschützte Ressource	
Test Nr.	7
Beschreibung	Durch diesen Test wird der Zugriff auf eine geschützte Ressource getestet.
Randbedingungen	Ein Nutzer mit dem Benutzernamen "peterpan" und dem Passwort "peterpanist" erfolgreich angemeldet worden.
erwartete Resultate	Der Nutzer greift auf die Lernfunktion zu. Er kann mit einer Lernsession beginnen.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 14: Test Zugriff geschützte Ressource, Quelle: Autoren

Test Aufgabenstellung hinzufügen	
Test Nr.	8
Beschreibung	Durch diesen Test wird das Hinzufügen einer neuen Aufgabenstellung im Multiple Choice Format getestet.
Randbedingungen	Es wird ein bereits eingerichteter Lehreraccount, bestehende CategorySets und Categories, sowie die vorhandenen Bilder zur Verfügung gestellt.
erwartete Resultate	Die Frage wird erstellt und dem gewünschten CategorySet hinzugefügt.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 15: Test Aufgabenstellung hinzufügen, Quelle: Autoren

Test Prüfung aus bestehenden Fragen erstellen	
Test Nr.	9
Beschreibung	Durch diesen Test wird das Erstellen und direkte Freigeben einer Prüfung getestet.
Randbedingungen	Es wird ein bereits eingerichteter Lehreraccount, bestehende CategorySets und Categories, Schulklassen mit Schülern, sowie die vorhandenen Bilder zur Verfügung gestellt.
erwartete Resultate	Die Prüfung kann mit den gewünschten Fragen und den gewählten Klassen erstellt werden.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 16: Test Prüfung erstellen, Quelle: Autoren

Test Analysieren und bearbeiten einer abgeschlossenen Prüfung	
Test Nr.	10
Beschreibung	Durch diesen Test wird das nachträgliche Bearbeiten von Prüfungsergebnissen durch einen Lehrer getestet.
Randbedingungen	Es wird ein bereits eingerichteter Lehreraccount mit einer von Schülern absolvierten Prüfung zur Verfügung gestellt.
erwartete Resultate	Der Lehrer kann die Antworten von einzelnen Usern dessen Fragen bearbeiten und so die erreichte Punktezahl korrigieren.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 17: Test Analysieren und bearbeiten einer abgeschlossenen Prüfung, Quelle: Autoren

Test Absolvieren einer Prüfung nur während bestimmtem Zeitraum	
Test Nr.	11
Beschreibung	Durch diesen Test wird das Absolvieren einer Prüfung ausserhalb eines gültigen Zeitraums getestet.
Randbedingungen	Es ist eine Prüfung mit einem Bearbeitungszeitraum vom 1.1.2020 - 2.1.2020 erstellt worden. Die Prüfung wurde der Klasse des Users freigegeben.
erwartete Resultate	Der Schüler kann die Prüfung nicht starten.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 18: Test Absolvieren einer Prüfung nur während bestimmtem Zeitpunkt, Quelle: Autoren

Test Absolvieren einer Prüfung ohne Berechtigung	
Test Nr.	12
Beschreibung	Durch diesen Test wird das Absolvieren einer Prüfung für einen nicht autorisierten Nutzer getestet.
Randbedingungen	Es ist eine Prüfung mit einem gültigen Bearbeitungszeitraum erstellt worden. Die Testperson besitzt einen Schüleraccount, für den die entsprechende Prüfung nicht zugewiesen wurde.
erwartete Resultate	Die Testperson kann die Prüfung nicht starten.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 19: Test Absolvieren einer Prüfung ohne Berechtigungen, Quelle: Autoren

Test Migration von Daten	
Test Nr.	13
Beschreibung	Durch diesen Test wird die Migration der Daten in die neue Applikation getestet.
Randbedingungen	Die bestehende Datenbank wurde gelöscht und die Applikation neu gestartet. Die csv-Dateien werden mit den zur Verfügung gestellten Postman Kommandos an die API gesendet.
erwartete Resultate	Die Daten werden erfolgreich in die Applikation migriert. Bei der Migration der Fragen ist nach Auftreten einer Fehlermeldung ein erneutes Senden nötig. In der Datenbank befinden sich rund 3000 Fragen.
Testperson	Frederico Fischer
Datum	01.12.2020
Testprotokoll	??

Tabelle 20: Test Migration von Daten, Quelle: Autoren

## **C System-Spezifikation**

### **C.1 Systemübersicht**

#### **C.1.1 Systemarchitektur**

Abbildung 7: Systemarchitektur, Quelle: Autoren

### **C.1.2 Kontextdiagramm**

Abbildung 8: 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 9 ersichtlich.

Abbildung 9: Datenbankschema, Quelle: Autoren

### C.2.3 Entwurfsentscheide

#### Frontend

**Technologien** Das Frontend wurde mit React aufgebaut, wobei als Hilfsmittel Facebook, o.D. für die Entwicklung verwendet wurde. Betrieben wird die Applikation auf einem NGINX-Webserver mit der Version 1.16.0, welcher durch die Facebook, o.D. 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 10: 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 11: Komponentendiagramm des Frontends, Quelle: Autoren

**Externe Packages** In der Abbildung 12 sind alle externen Packages aufgelistet, welche für dieses Projekt verwendet wurden.

Abbildung 12: 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 13: 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 14: 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 15: 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 16.

**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 16: 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 Anforderungen

### D.1 Einführung

### D.2 Einzelanforderungen

#### D.2.1 Funktionen

ID	Anforderung	Muss/Kann
F.1	Als Benutzer kann ich gewünschte Übungen markieren um diese selektiv nochmals absolvieren zu können.	Muss
F.2	Als Lehrer kann ich eine Prüfung mit selektierten Fragen erstellen.	Muss

Tabelle 21: Anforderungen, Quelle: Autoren

#### D.2.2 Anforderungen an die Benutzbarkeit

ID	Anforderung	Muss/Kann
B.1	Die Applikation soll ein modernes, innovatives und zeitgemässes User Interface bieten.	Ver

Tabelle 22: Anforderungen, Quelle: Autoren

#### D.2.3 Anforderungen an die Leistungsfähigkeit

ID	Anforderung	Muss/Kann
L.1	Das System muss jederzeit von mindestens 100 bis 120 Benutzern gleichzeitig verwendet werden können.	Muss

Tabelle 23: Anforderungen, Quelle: Autoren

#### D.2.4 Weitere Qualitätsmerkmale

ID	Anforderung	Muss/Kann
Q.1	Als Benutzer muss ich mich bei vorhandener Internetverbindung jederzeit im System einloggen können.	Muss

Tabelle 24: Anforderungen, Quelle: Autoren

#### D.2.5 Wartungs- und Supportinformationen

ID	Anforderung	Muss/Kann
S.1	Die Applikation muss so dokumentiert werden, dass eine spätere Weiterentwicklung anhand dieser problemlos möglich ist.	Muss

Tabelle 25: Anforderungen, Quelle: Autoren

**D.2.6 Weitere Anforderungen**

<b>ID</b>	<b>Anforderung</b>	<b>Muss/Kann</b>
W.1	Die Applikation soll in Form einer Webapplikation realisiert werden, die auf einem Webserver als eigenständige Applikation betrieben werden kann.	Muss
W.2	In der Applikation muss das Logo der Hochschule Luzern sowie der Studiengang Gebäudetechnik ersichtlich sein.	Muss

Tabelle 26: Anforderungen, Quelle: Autoren

## E Sitzungsprotokolle

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

### E.1 14.09.2020

#### E.1.1 Ordnungsaufruf

Eine Besprechung aller Projektbeteiligten fand online als Zoom-Meeting am 14.09.2020 um 11:00 Uhr statt.

#### E.1.2 Teilnehmer

Anwesende Mitglieder	Nicht anwesende Mitglieder
Jordan Sucur, Betreuungsperson Roger Buser, Auftraggeber Oliver Werlen, Projektleiter Frederico Fischer, Scrum Master	-

Tabelle 27: 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

Gemäss dem Zeitplan wird in zwei Wochen die Systemkontextphase abgeschlossen. Dabei erfolgt der erste Release eines GUI-Prototypen.

#### E.1.5 besprochene Punkte

##### Registrierung

- Lehrer bestimmt, wer im Kurs ist und kann Teilnehmer nachträglich bearbeiten
- Umsetzung ist dabei frei

##### Anforderungen

- Anforderungen müssen einheitlicher formuliert werden. Alle als Epics oder Userstories.
- Anforderungen müssen strukturiert werden, basierend auf dem Strukturplan.
- Kategorie für "weitere Anforderungen" definieren.
- In der Software sollen Fragen nicht nur importiert, sondern auch exportiert werden. Format ist dabei frei.
- Als Sponsor soll die Hochschule Luzern inklusive dem Studiengang Gebäudetechnik gut sichtbar sein.
- Antworten sollen im Übungsmodus direkt auf Korrektheit überprüft werden können. Das Ergebnis ist direkt sichtbar. Sollte dies vom Schüler nicht gewünscht werden, kann eine Übungsprüfung absolviert werden.
- Anstatt eines Handbuchs wird ein interaktives Tutorial zur Verfügung gestellt. Ein Handbuch ist nicht nötig.

- Spezifizieren wie man Applikation erweitern müsste, falls doch mehr als 120 Nutzern vorhanden sind.
- Applikation muss parallel nutzbar sein (als Anforderung spezifizieren).
- Quizes als Kann-Funktion erweitern

### **Stand der Praxis in Dokumentation**

- Beschreiben, wieso alte Software nicht mehr Nutzbar ist.
- Was könnte man besser machen?
- Gibt es noch andere vergleichbare Applikationen auf dem Markt?

### **Kapitelaufteilung in Dokumentation**

- Projektmanagementplan so belassen, Änderungen notieren und jeweils auf den Anhang verweisen

### **Anderes**

- Blockdiagramme: Datenfluss spezifizieren

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

- Analyse GUI-Prototyp
- UML-Diagramme besprechen
- Besprechung CI/CD

### **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**