SE Project
Documentation

OSTShare

Semester: Spring 2023

SE PROJECT

Version: 1.0
Date: 2023-06-01 17:37:07Z
Git Version: review-5-36-gfa4df63

Project Team: Fabian Freitag
Fabio Elvedi
Josip di Benedetto
Kevin Pfister
Nicolas Gattlen

Project Advisor: Timon Erhart

School of Computer Science
OST Eastern Switzerland University of Applied Sciences

Contents

Part I

Management Summary

Management Summary

0.1 Projektidee

Die Idee des Projektes ist es eine einheitliche Lernplattform für Schüler und Lehrer zu erstellen. Die Lehrer bekommen dabei die Möglichkeit ein Forum für Ihr Fach zu erstellen. In diesem Forum können sie dann ihre Lernunterlagen für die Schüler bereit stellen. Ebenfalls können wichtige Kursankündigungen via dem Forum mitgeteilt werden und es müssen nicht extra E-Mails geschrieben werden.Die Schüler bekommen gleichzeitig die Möglichkeit Ihre selbsterstellten Unterlagen im gleichen Forum hochzuladen um diese dann mit Ihren Mitstudenten zu teilen. Damit lösen wir das Problem, dass Schüler nur eine Plattform bedienen müssen um an alle Informationen und Unterlagen Ihres Kurses zu kommen. Das Gleiche Prinzip gilt für die Lehrer bezüglich der Distribution Ihrer Unterlagen

0.2 Problemlösung

Um eine einheitliche Lernplattform zu schaffen musste man ein Medium wählen, bei welchem jeder User einen schnellen Zugriff hat. Man hat sich deshalb dazu entschieden eine Webseite zu erstellen. Um sicherzustellen dass innerhalb des Teams die gleichen Vorstellungen bestehen, wurden am Anfang die Voraussetzungen für eine solche Plattform besprochen. Dabei war es dem Team wichtig die Kernfunktionen zu bestimmen ohne gross ins Detail zu gehen, diese sollen dann im Verlaufe des Projektes näher besprochen werden. Basierend dieser Kernfunktionen hat man sich innerhalb des Teams dafür entschieden, die Webseite mit React, einem Web Framework, zu erstellen. Nach näherer Ausarbeitung unserer Voraussetzungen (Funktional und nicht-Funktional) wurde dann das Team in 2 separate Teams unterteilt. Das Frontend Team war dabei für das Aussehen der Webseite verantwortlich und sorgte dafür, dass die Webseite trotz vielen Funktionen übersichtlich bleibt. Das Backend Team war für die Umsetzung der technischen Voraussetzungen verantwortlich, ihr Job bestand darin dass Technische Funktionen wie Datenbankimplementationen oder die Instandhaltung der Webseite funktionieren. Nach der Erstellung der Webseite war es wichtig, diese an Studenten zu testen. Dazu wurden Schüler aus verschiedenen Studiengängen befragt die einzelnen Funktionen zu testen und gleichzeitig ein Feedback zur Intuivität der Webseite zu geben.

0.3 Erfolg des Projektes

Das Team konnte innerhalb kurzer Zeit eine funktionierende Plattform aufbauen welche alle wichtigen Funktionen und Voraussetzungen erfüllt. Leider konnte jedoch die Funktion, dass Lehrer Änderungen und updates ankündigen können, nicht erfüllt werden. Dies ist jedoch nicht so schlimm, da die Priorität dieser Funktion als niedrig eingestuft wurde. Die Rückmeldungen der Test-User ergaben, dass die erstellte Plattform übersichtlich ist und alle wichtigen Funktionen abdeckt. Ob das Problem, welches wir lösen wollten, jedoch wirklich gelöst ist, ist schwierig zu sagen. Da das Produkt neben den Tests noch keine echt-Welt Anwendung gefunden hat. Um hierbei eine definitive Antwort geben zu können, müsste man es Lehrern oder Dozenten geben, welche die Plattform über längere Zeit mit Ihren Schülern testen.

0.4 Fortsetzung des Projektes

Das Projekt konnte viele Funktionen implementieren und wichtige Voraussetzungen abdecken. Jedoch gibt es noch viele weitere Funktionen welche man implementieren könnte um das Leben der Schüler und Lehrer leichter zu gestalten. Dazu gehören unter anderem ein Chat oder eine Mobile Version. Es ist schwierig zu sagen ob das Projekt weitergeführt wird, da der Ursprung des Projektes darin lag ein Projekt für das Modul SE Projects abzugeben. Jedoch wurde die Webseite so programmiert, dass die Webseite einfach skaliert werden kann und allfällige Veränderungen einfach hinzugefügt werden können. Schlussendlich kann gesagt werden, dass jedes Teammitglied sich selber entscheiden muss, ob es am Projekt weiterarbeiten möchte oder nicht.

Part II

Product Documentation

Chapter 1

Requirements

1.1 Funktionale Anforderungen

1.1.1 Akteure und Ziele

Akteure und Ziele	
Akteure	Ziele
Studierende	Teilen von Zusammenfassungen / Vorlesungsnotizen Dokumente bewerten Vorlesungsunterlagen ansehen und herunterladen Fragen zu Lektionsinhalten stellen / beantworten
Lehrperson	Hochladen und teilen von Vorlesungsunterlagen Kurse und Lektionen erstellen Fragen zu Lektionsinhalten stellen / beantworten

Table 1.1: Akteure und Ziele

1.1.2 Use cases

Für alle Funktionalen Anforderungen gilt die Voraussetzung, dass der Benutzer angemeldet ist. Anforderungen mit Studierenden als Akteuren können von Studierenden und Lehrpersonen ausgeführt werden. Die Anforderungen mit Lehrpersonen als Akteuren können jedoch nur von Lehrpersonen ausgeführt werden.

ID	FR-1
Title	Zusammenfassungen teilen
Beschreibung	Studierende können ihre Zusammenfassungen und Vorlesungsnotizen hochladen und mit anderen teilen.
Akteur	Student
Voraussetzung	Kurs mit Lektionen ist erstellt
Priorität	Hoch

Table 1.2: FR-1

ID	FR-2
Title	Zusammenfassungen bewerten
Beschreibung	Benutzer können Zusammenfassungen von anderen Studierenden bewerten
Akteur	Student, Lehrperson
Voraussetzung	Zusammenfassungen existieren für einen Kurs oder Lektion
Priorität	Mittel

Table 1.3: FR-2

ID	FR-3
Title	Fragen und Antworten
Beschreibung	Studierende können ihre Fragen zu den Kursinhalten stellen und von anderen Studierenden oder Dozenten beantworten lassen
Akteur	Student, Lehrperson
Voraussetzung	Kurs existiert
Priorität	Niedrig

Table 1.4: FR-3

ID	FR-4
Title	Kursankündigungen
Beschreibung	Lehrpersonen können Ankündigungen, Änderungen oder Updates zu den Kursinhalten auf der Plattform ankündigen
Akteur	Lehrperson
Voraussetzung	Kurs existiert
Priorität	Niedrig

Table 1.5: FR-4

ID	FR-5
Title	Vorlesungsunterlagen teilen
Beschreibung	Lehrpersonen können ihre Vorlesungsunterlagen hochladen und mit den Studierenden teilen
Akteur	Lehrperson
Voraussetzung	Kurs mit Lektionen existiert
Priorität	Hoch

Table 1.6: FR-5

ID	FR-6
Title	Kursmodellierung
Beschreibung	Lehrpersonen können Kurs mit Lektionen auf der Lernplattform abbilden
Akteur	Lehrperson
Voraussetzung	Lehrperson-Konto wurde autorisierung
Priorität	Hoch

Table 1.7: FR-6

1.1.3 Status Funktionale Anforderungen

ID	Geplant	Erledigt
FR-1	Review 3	i.O
FR-2	Review 5	i.O
FR-3	Review 5	i.O
FR-4	Review 5 (Tiefe Priorität)	n.i.O
FR-5	Review 3	i.O
FR-6	Review 5	i.O

Table 1.8: Status Funktionale Anforderungen

1.1.4 Use case diagram

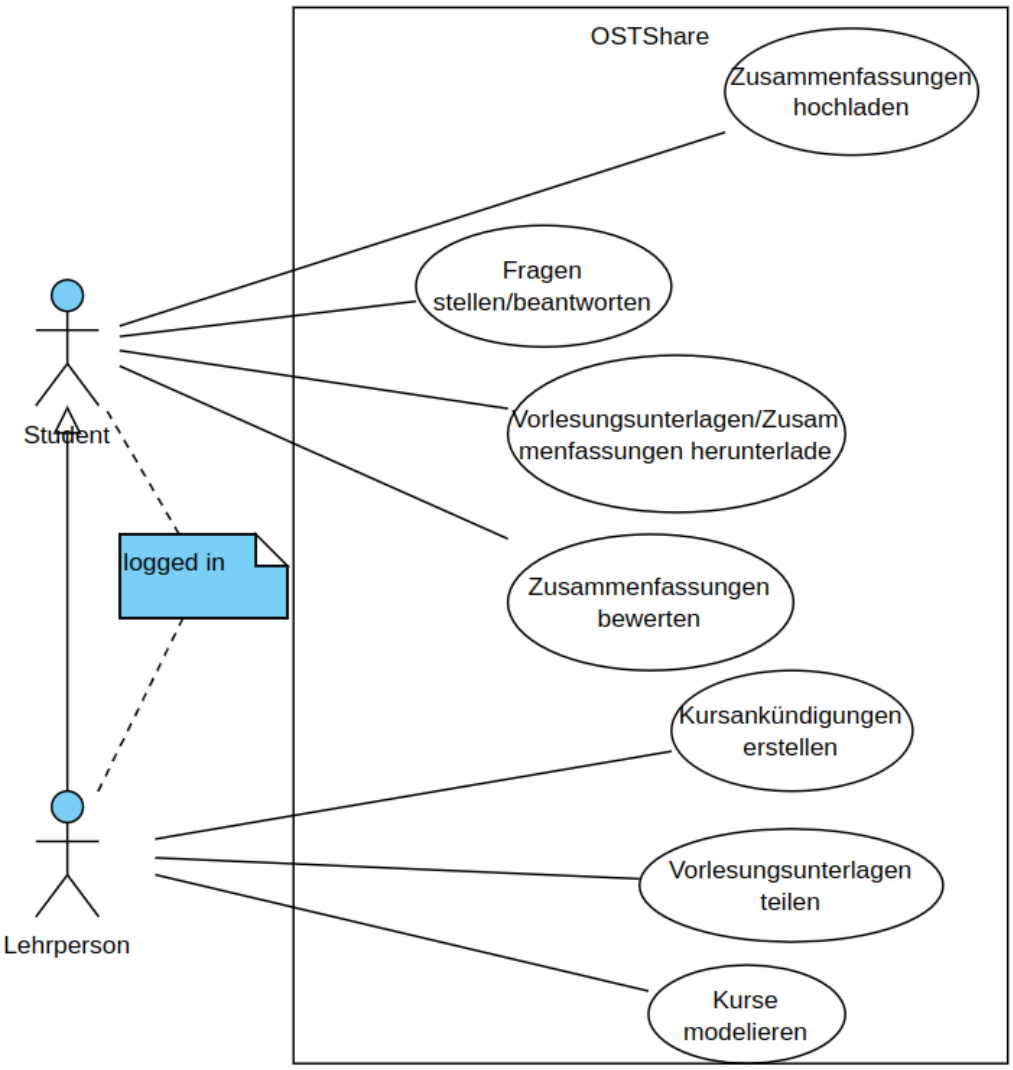

Figure 1.1: Use Case Diagram

1.2 Nicht funktionale Anforderungen

ID	NFR-1
Thema (Subject)	Verantwortlichkeit (Accountability) der Files, welche hochgeladen werden
Voraussetzung (Requirement)	Security - Accountability
Wichtigkeit	Hoch
Beschreibung	• Sicherstellen dass User nur Ihre eigenen Files löschen können. • Dem User ist stets bewusst wer das File hochgeladen hat
Überprüfbarkeit	Ein Login einführen welches die User authentifiziert

Table 1.9: NFR-1

ID	NFR-2
Thema (Subject)	Sicherung der Passwörter
Voraussetzung (Requirement)	Security - Confidentiality
Wichtigkeit	Hoch
Beschreibung	• Die Passwörter sollen nicht in plain Text gespeichert werden. Passwörter dürfen nicht eingesehen werden.
Überprüfbarkeit	Die Passwörter müssen vor dem abspeichern gehasht werden

Table 1.10: NFR-2

ID	NFR-3
Thema (Subject)	Selbsterklärendes User Interface
Voraussetzung (Requirement)	Usability - Appropriateness
Wichtigkeit	Mittel
Beschreibung	• Users können das User Interface benutzen, ohne eine Erklärung zu bekommen
Überprüfbarkeit	Eine Umfrage mit 5 Personen bei denen 80 Prozent der Teilnehmer sagen, dass es einfach zu bedienen ist

Table 1.11: NFR-3

ID	NFR-4
Thema (Subject)	Limit der Dateiengrösse
Voraussetzung (Requirement)	Performanz Effizienz - Ressourcenverhalten
Wichtigkeit	Hoch
Beschreibung	• Sicherstellen dass die hochgeladenen Dateien nicht grösser als 10 Mb sind.
Überprüfbarkeit	Beim Hochladen der Dateien wird geprüft ob die Datei grösser ist als 10 Mb

Table 1.12: NFR-4

1.2.1 Status Nicht-Funktionale Anforderungen

ID	Geplant	Erledigt
NFR-1	Review 5	i.O
NFR-2	Review 3	i.O
NFR-3	Review 6	i.O
NFR-4	Review 5	i.O

Table 1.13: Status Nicht-Funktionale Anforderungen

1.3 Minimum Viable Product

Das Ziel dieses Projektes ist es, eine Filesharing Plattform zu erstellen. Für ein MVP muss der grundlegende Workflow einer Filesharing Plattform funktionieren. Das bedeutet:

- Der Benutzer kann sich mit seinem Account anmelden
- Bereits erstellte Kurse werden aufgelistet
- Innerhalb eines Kurses werden alle vorhandenen Dokumente aufgelistet
- Der Benutzer kann alle Files herunterladen
- Der Benutzer kann neue Dokumente zu einem Kurs hochladen
- Der Benutzer kann seine Dokumente löschen

Chapter 2

Domain Analysis

Das UML Diagram basiert auf den Requirements und visualisiert die Problem Domäne.

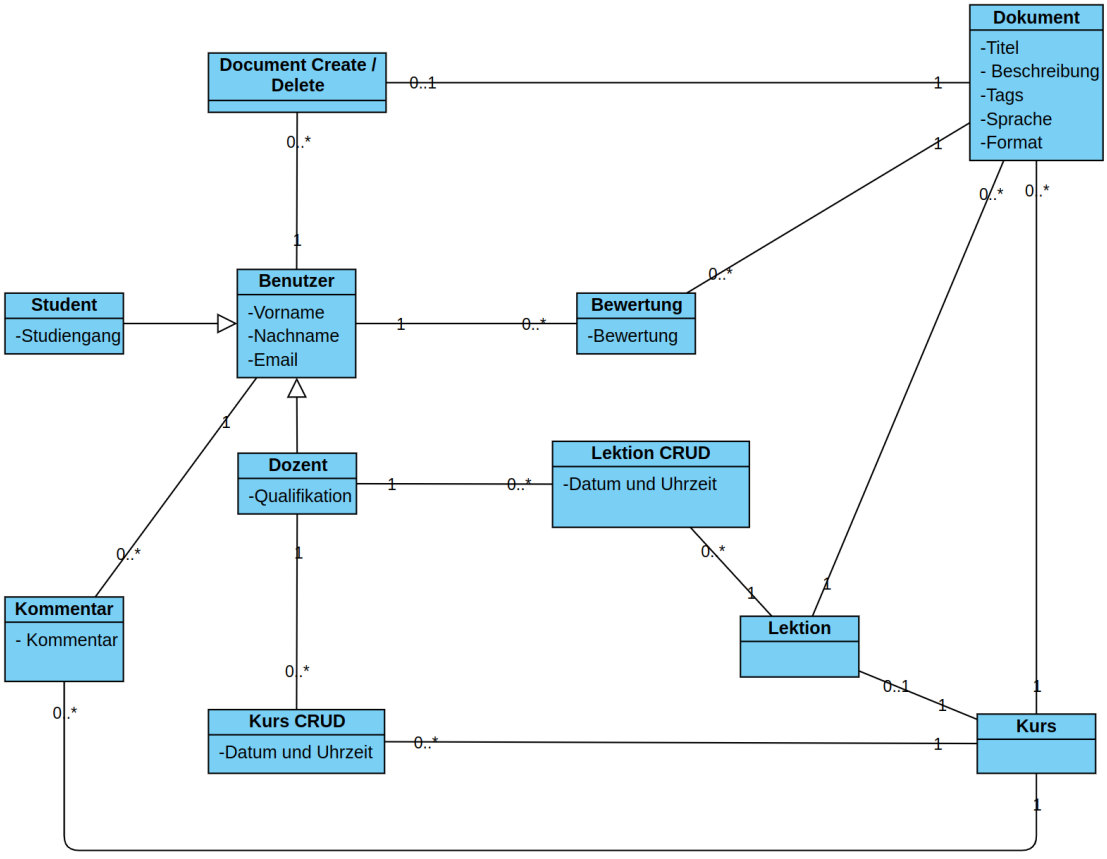

Figure 2.1: Problem Domäne

- Benutzerkonten wie Studenten und Lehrpersonen werden vom OSTShare Team direkt in der Datenbank hinterlegt

Chapter 3

Architecture

3.1 Layer und Tiers

Das Projekt wird in drei Layers un zwei Tiers unterteilt. Die drei Layers sind: Presentation Layer, Business-logic Layer und Data-access Layer. Zusätzlich wird es zwei Tiers geben: Client (React) und Server (Node Express) welche in einer monolitischen Architektur aufgebaut sind.

3.1.1 Presentation Layer

Der Presentaton-Layer ist dazu da, die Informationen für den Benutzer zu visualisieren und die Inputs des Benutzers für den Business-logic-Layer bereitzustellen. Dieser Layer wird im Client Tier verwendet. Konkret hat der Presentation Layer die folgenden Aufgaben:

- Das User Interface darstellen und sicherstellen, dass der Benutzer damit interagieren kann.
- Nutzereingaben entgegennehmen und so verarbeiten, dass der Business-logic Layer damit weiterarbeiten kann.
- Sicherheit bringen, indem die Nutzereingaben bereinigt werden.

3.1.2 Business-logic Layer

Der Business-logic Layer ist dafür da, die Daten, welche vom Presentation Layer kommen, weiterzuverarbeiten und sowohl mit dem Presentation als auch mit dem Data-access Layer zu kommunizieren. Er muss sicherstellen, dass die Applikation wirklich das macht, was sie soll. Dieser Layer wird ausschließlich vom Server Tier verwendet. Konkret hat der Business-logic Layer die folgenden Aufgaben:

- Business rules implementieren, also sicherstellen, dass die Applikation so arbeitet wie geplant.

- Daten vom Presentation Layer so verarbeiten und anreichern, dass der Data-access Layer damit weiterarbeiten kann und das auch in umegekehrter Richtung.

3.1.3 Data-access Layer

Der Data-access Layer ist dafür verantwortlich, dass die Daten persistent gespeichert werden und der Business-logic Layer Zugriff darauf hat. Dieser Layer wird ausschließlich vom Server Tier verwendet. Konkret hat der Data-access Layer folgende Aufgaben:

- Kommunikation mit dem Datenbank-Server herstellen und die Kommunikation mit ihm aufrecht halten.
- Datenzugriffsmethoden implementieren, damit der Business-logic Layer einfachen Zugriff auf die Datenbank hat.
- Datenvalidierung implementieren, damit die Daten in der Datenbank konsistent sind und mit dem definierten Datenmodell übereinstimmen.

3.2 C4 Modell

Das C4 Modell wird laufend aktuell gehalten, sobald nennenswerte Änderungen vorgenommen werden.

Context

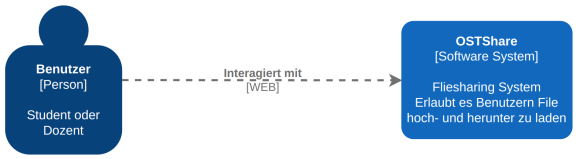

Figure 3.1: C4 Context Diagramm

Container

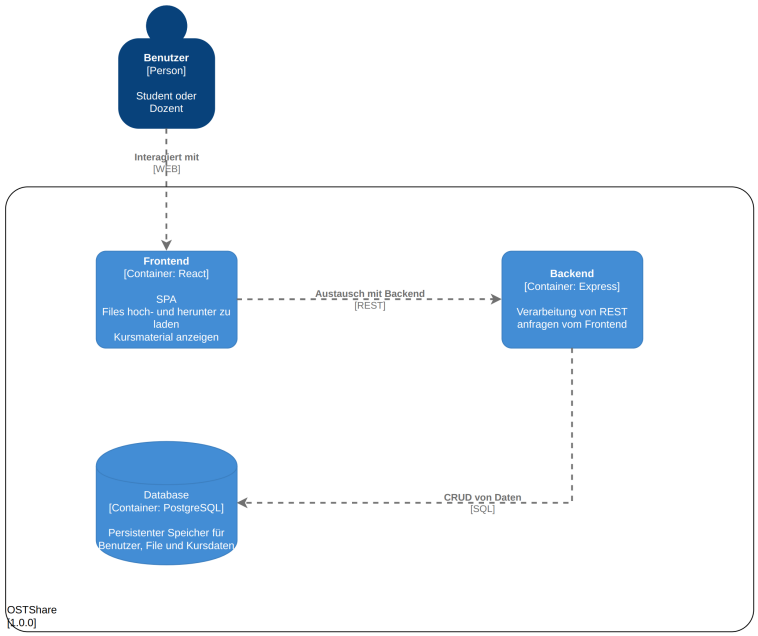

Figure 3.2: C4 Container Diagramm

Auf dem Server für das Deployment werden wie im obigen Diagramm zu sehen sind, zwei Docker Container mit dem NodeJS Server und ein Docker Container mit der PostgreSQL Datenbank verwendet um die Applikation zu betreiben.

Components

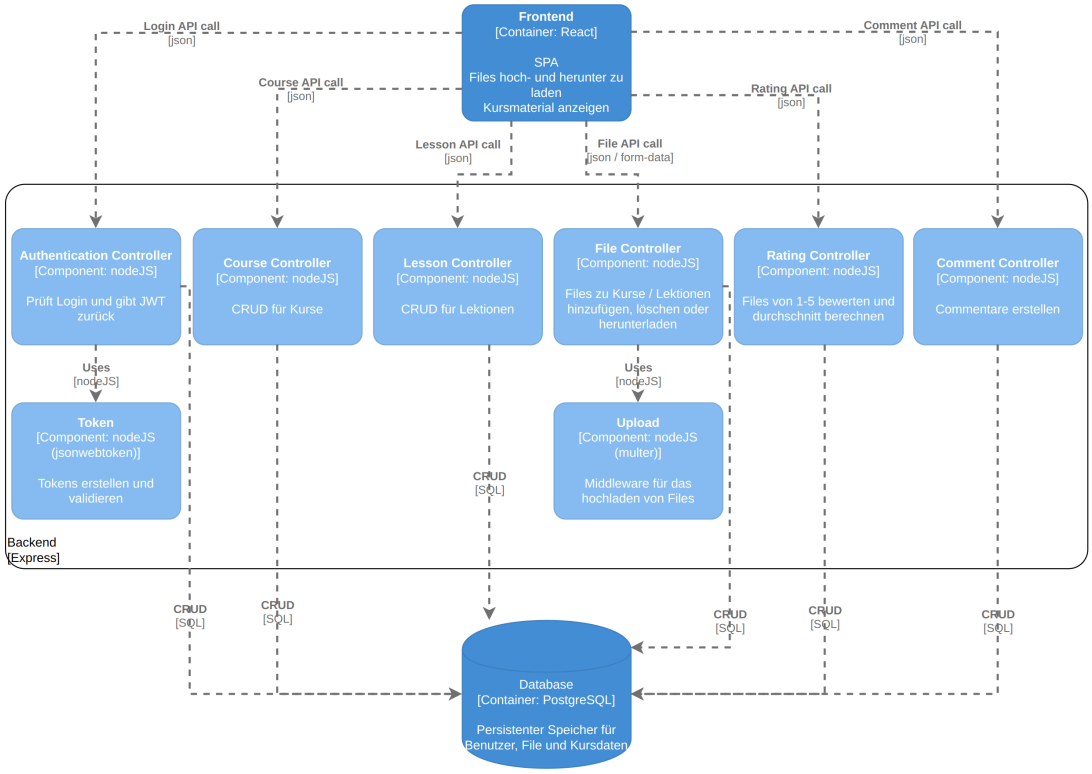

Figure 3.3: C4 Component Backend Diagramm

Für den Datenaustausch zwischen dem Backend und der Datenbank werden Repository Klassen verwendet. Diese Klassen sind dafür zuständig, die Datenbankabfragen zu erstellen und auszuführen. Die Datenbankabfragen werden mit dem ORM Sequelize erstellt und ausgeführt, welches die Datenbankabfragen in SQL Queries umwandelt.

ERD

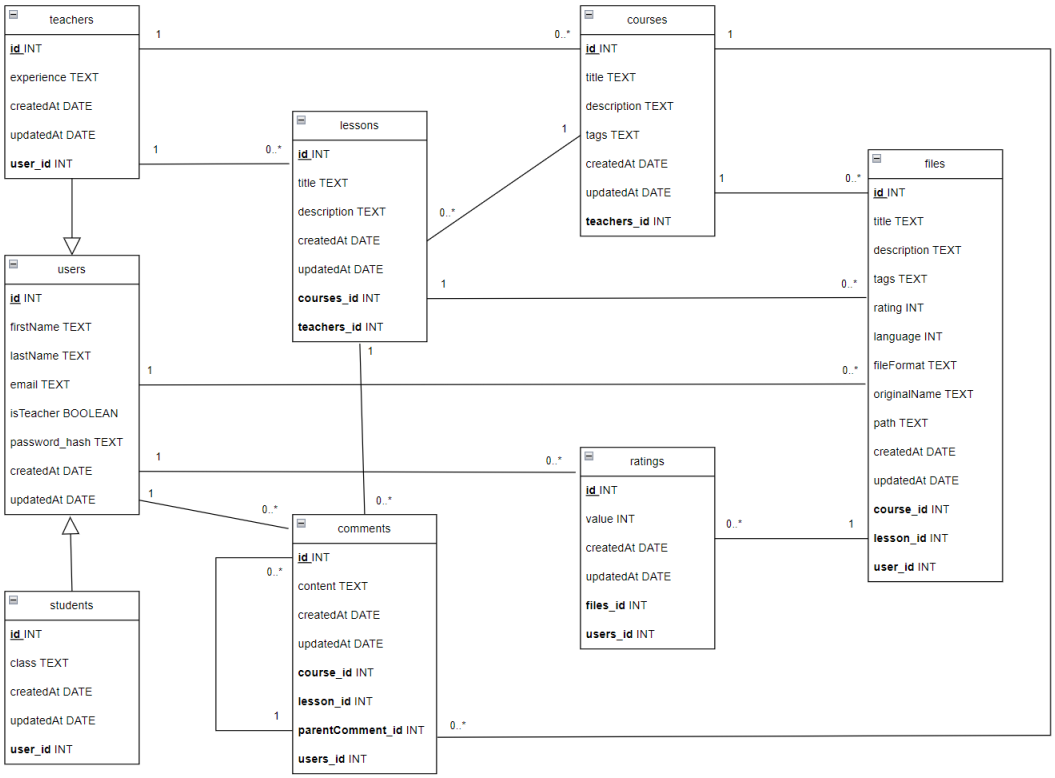

Figure 3.4: C4 ERD

3.3 Technologien

3.3.1 Frontend

Das Frontend wird in React implementiert. Das Projektteam hat bisher wenig Erfahrung mit der Frontend Entwicklung. Deshalb ist es wichtig, für das Frontend eine Technologie zu verwenden, die sehr gut dokumentiert ist, zu welcher schnell viel know-how gesammelt werden kann und leicht zu verwenden ist. React erfüllt diese Voraussetzungen.

3.3.2 Backend

Das Backend wird ein NodeJS Server mit einer RESTful API sein. NodeJS ist dem ganzen Projektteam bereits ein wenig bekannt, wodurch es eine offensichtliche Option war. Ausserdem wirbt NodeJS mit einigen Punkten, die für dieses Projekt wertvoll erscheinen. NodeJS ist relativ simpel und sehr gut dokumentiert. Es sind jede Menge open-source Packages verfügbar, die die Entwicklung stark vereinfachen sollten, da sie "Standardfunktionen" bereits zur verfügung stehen.

3.3.3 Datenbank

Als Datenbank wird eine PostgreSQL Datenbank verwendet. Da die Datenbank Module an der OST anhand von PostgreSQL unterrichtet werden, sollte hier bereits ein entsprechndes know-how vorhanden sein. Entsprechend war die Wahl der Datenbank relativ offensichtlich. Es wurden keine anderen Datenbank Technologien evaluiert, da entschieden wurde, dass der Aufwand für das lernen einer neuen Datenbank Technologie den Rahmen des Projektes sprengen würde und PostgreSQL die Anforderungen für das Projekt erfüllen.

3.4 Deployment

Für das Deployment wurde ein Server bei DigitalOcean gemietet. Auf diesem Server wurde Docker installiert und die Docker Container werden deployed. Die Docker Container sind über einen Nginx Reverse Proxy erreichbar. Dieser leitet die Anfragen an den entsprechenden Docker Container weiter. Zusätzlich wurde mit Let's Encrypt ein SSL Zertifikat erstellt, damit die Verbindung verschlüsselt ist. Dafür wurde die Domäne 'ostshare.online' gekauft und auf den Server weitergeleitet.

3.5 Skalierbarkeit

Die einzelnen Komponenten (Frontend und Backend) werden in Docker Containern deployed. Dadurch ist es möglich, die einzelnen Komponenten auf mehrere Server zu verteilen. Dies könnte mit einem Kubernetes Cluster oder ähnlichen Tools realisiert werden. Jedoch ist die Skalierbarkeit nicht Teil des Projektes und wird deshalb nicht weiter verfolgt.

3.6 Erweiterbarkeit

Die Applikation soll so erweiterbar sein, dass es möglich ist, weitere Module hinzuzufügen. Dies soll möglich sein, ohne dass die bestehenden Module angepasst werden müssen.

3.6.1 React

Im React Frontend ist dies relativ einfach möglich, da die Module in React Components umgesetzt werden. Diese können einfach hinzugefügt werden.

3.6.2 Express

Im Express Backend ist dies etwas komplizierter, da die Datenbank Tabellen angepasst werden müssen. Im Development-environemnt wird dies automatisiert durchgeführt. Auf dem Server werden Datenbankmigrationen manuell durchgeführt. Dies ist nicht optimal, jedoch ist es für dieses Projekt ausreichend. Zusätzlich sind die Definitionen der Routes so aufgeteilt, dass es möglich ist, neue Module hinzuzufügen, ohne dass bestehender Code angepasst werden muss.

```
const routes = require('express').Router();
const authenticationRoutes = require('./authenticationRoutes');
const fileRoutes = require('./filesRoutes');
const courseRoutes = require('./courseRoutes');
const lessonRoutes = require('./lessonRoutes');
const commentRoutes = require('./commentRoutes');
const ratingRoutes = require('./ratingRoutes');

routes.use('/', authenticationRoutes);
routes.use('/', fileRoutes);
routes.use('/', courseRoutes);
routes.use('/', lessonRoutes);
routes.use('/', commentRoutes);
routes.use('/', ratingRoutes);

// catch all
routes.use('*', (req, res) => {
    res.status(404).json({ message: 'route does not exists' })
})

module.exports = routes;
```

Figure 3.5: API Routes

Chapter 4

Quality Measures

4.1 CI/CD

4.1.1 Dokumentation Repository

Die Dokumentation wird mit GitLab versioniert. Das Repository ist auf zwei Branches aufgeteilt. Die Branch Staging wird für tägliche Änderungen verwendet. Die Branch Main wird für die Abgaben der Reviews verwendet. Beim mergen auf der Staging Branch wird jeweils das main.pdf und diff.pdf generiert. Beim Mergen auf der Main Branch werden die gleichen CI/CD schritte wie auf der Staging Branch durchgeführt. Änderungen an der Dokumentation werden über Feature-Branches durchgeführt. Beim Merge-Request der Feature-Branches wird der neue, oder geänderte Inhalt vom Reporter des Tasks überprüft

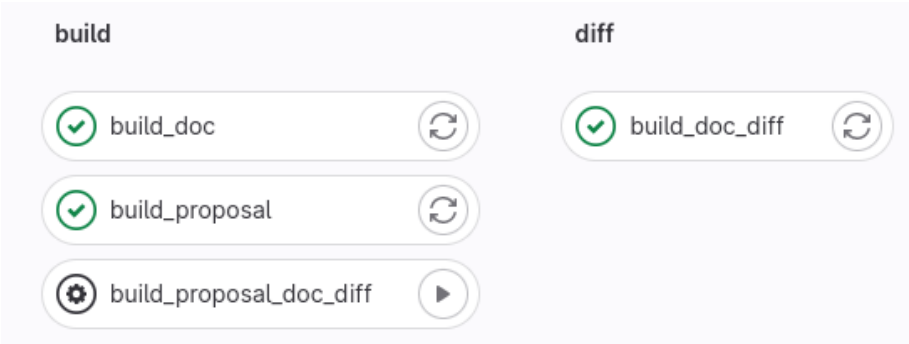

Figure 4.1: CI/CD Dokumentation

4.1.2 Product Repository

Das Product Repository ist auf zwei Branches aufgeteilt. Die Main Branch wird für stabile releases der Applikation verwendet. Die Staging Branches werden für tägliche Änderungen verwendet. Neue Funktionen werden über Feature-Branches implementiert.

Die daraus folgenden Merge-Requests werden jeweils von einer Person des entsprechenden Teams überprüft (Frontend/Backend/Testing). Die CI/CD Pipeline dockerisiert eine PostgreSQL Datenbank und zwei Node Server. Auf den Node Containers wird der Express Server und React Client ausgeführt. Die Codebase wird mit ESLint in der CI/CD Pipeline geprüft. Anschliessend werden die Unit und Integration Tests durchgeführt. Zusätzlich bietet die Pipeline auch eine Stage für das Deployment an. Das Deployment wird mit Docker Compose auf einem remote Server durchgeführt. Diese Stage wird nicht automatisch durchgeführt, sondern muss manuell gestartet werden.

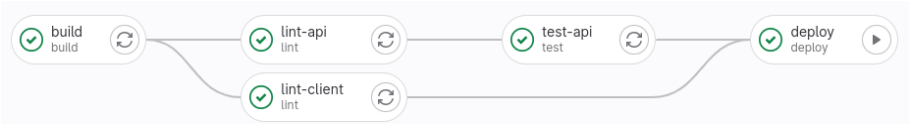

Figure 4.2: CI/CD Code

4.2 Teststrategie

4.2.1 Unit Testing

Unit-Tests für das Node Express Backend werden automatisch in einer GitLab CI/CD Pipline ausgeführt oder manuell über ein makefile. Da das React Frontend wenig bis keine Logik enthält, wird dies mit den End To End Tests abgedeckt.

4.2.2 Integration Testing

Im Backend werden zusätzlich Integration Tests für die einzelnen "Routes" (API Interfaces) erstellt. Wie auch die Unit-Tests werden diese automatisch von der CI/CD Pipline ausgeführt oder manuell über ein makefile.

4.2.3 End To End Testing

End To End Tests werden mit Hilfe von Testcafe automatisch durchgeführt. Die Tests werden einmal pro Woche oder bei Bedarf durchgeführt. Sie werden nicht in die CI/CD Pipeline aufgenommen. Die Tests werden sowohl in Chrome als auch in Firefox ausgeführt und die jeweiligen ergebnisse in einem Textfile gespeichert. Im Anhang findet sich die Auswertung der Tests der aktuellen Woche.

4.2.4 End User Tests

Zum Ende der Construction Phase, also in der Woche vom 17.05.2023 sollen End User Tests durchgeführt werden. Dabei sollen mindestens fünf Studierende ausserhalb des Projektteams das Fertige Produkt testen und darüber berichten, ob sie es sich vorstellen können, mit dem Tool zu arbeiten und falls nicht sollen Verbesserungsvorschläge gebracht werden. Dabei wird auch das NFR-3 "Selbsterklärendes User Interface" überprüft. Die

Tests finden am Ende der Construction Phase statt, damit noch gelegenheit besteht, Feedback in das finale Produkt mit einzubeziehen. Dies ist besonders für das NFR-3 wichtig, da das Entwicklerteam dieses nicht sinnvoll selbst testen kann.
Folgende Punkte sollen bei den End User Tests aus der Sicht Student und Lehrperson getestet werden:

Situation Student	
Aufgabe	Beschreibung
Login	Dem Benutzer wird ein Login zur Verfügung gestellt. Er soll sich damit einloggen.
Kursnavigation	Dem Benutzer wird ein bestimmter Kurs angegeben, zu dem er navigieren soll.
Lektionsnavigation	Dem Benutzer wird eine bestimmte Lektion angegeben, zu der er navigieren soll.
Datei Upload	Dem Benutzer wird eine Datei vorgegeben, die er zur Lektion, zu der er navigiert hat, hochladen soll.
Datei Download	Der Benutzer soll eine vorhandene Datei, die in einer vorgegebenen Lektion hinterlegt ist, herunterladen.
Datei löschen	Der Benutzer soll die Datei, die er zuvor hochgeladen hat, löschen.
Datei bewerten	Der Benutzer soll eine beliebige Datei bewerten.
Frage stellen	Der Benutzer soll zu einem Kurs eine Frage stellen.
Frage beantworten	Der Benutzer soll zu einem Kurs eine Frage beantworten.

Situation Lehrperson	
Aufgabe	Beschreibung
Kursankündigung erstellen	Der Benutzer soll zu einem Kurs eine Ankündigung erstellen.
Kurs erstellen	Der Benutzer soll einen neuen Kurs erstellen.
Lektion erstellen	Der Benutzer soll im erstellten Kurs eine Lektion erstellen.
Vorlesungsunterlagen Teilen	Dem Benutzer wird eine Datei zur Verfügung gestellt, welche er als Vorlesungsunterlage teilen soll.

Auswertung der End User Tests

An den End User Tests haben sechs Studierende teilgenommen. Davon studieren zwei Informatik, zwei Elektrotechnik, einer Maschinenbau und einer Energie und Umwelt. Durch die Verteilung der Teilnehmenden auf die verschiedenen Studiengänge, kann davon ausgegangen werden, dass das Ergbnis representativ für den Allgemeinen Nutzen für das Studium ist. Die Tester haben die vorbereitete Liste durchgearbeitet und danach nach eigenem ermessen die Applikation weiter getestet. Von den sechs Testern haben alle die Frage "Ist die Benutzeroberfläche Intuitiv?" mit ja geantwortet. Das heisst das Ziel des NFR-3, nach dem mindestens 80% der Tester das UI als intuitiv bewerten sollen, ist absolut erfüllt.
Bei der Frage nach Feedback haben sich folgende Punkte ergeben:

- Der Details Button für die Navigation zum Kurs ist überflüssig, man sollte direkt beim klicken auf den Kurs auf diesen weitergeleitet werden.
- Die Aufteilung ob eine Datei in einem Kurs oder in einer Lektion ist, könnte besser dargestellt werden.
- Eine Antwort auf einen Kommentar sollte in einem Textfeld und nicht in einem Input sein, so wäre es angenehmer, die Antwort zu schreiben.
- Der (fehlende) Abstand zwischen bestimmten Elementen sieht nicht immer schön aus.
- Die Buttons "Logout" und "Courses" sollten erst angezeigt werden, wenn der Benutzer eingeloggt ist.
- Es könnte besser ersichtlich gemacht werden, wer der Besitzer/Ersteller eines Kurses ist.
- Die Sprachauswahl beim Fileupload sollte in der jeweiligen Sprache sein, also z.B. Deutsch, English, Francais.
- Man könnte das OST-Logo in die Titelbar einbringen.
- An gewissen stellen des Workflows wird die Seite neu geladen, was teilweise nervig ist.
- Die Tags werden nicht angezeigt.

Aufgrund des späten Durchführungszeitpunkt der End User Tests werden diese Vorschläge zur kenntnis genommen, allerdings nicht während dem Projekt verbessert.

4.3 Organisatorisch

4.3.1 Definition of Done

Ein Feature oder eine User Story gilt als "done", wenn folgenden Kriterien erfüllt sind:

- Der Code wurde geschrieben und alle Änderungen wurden in Git dokumentiert.
- Der Code wurde von einem anderen Entwickler überprüft und gegebenenfalls Feedback wurde eingearbeitet.
- Neue Tests wurden geschrieben, welche die Grundfunktionalität des Features sinnvoll überprüfen.
- Alle Tests wurden erfolgreich ausgeführt und die Codeabdeckung ist angemessen.
- Alle notwendigen Dokumentationen wurden aktualisiert

4.4 Tools zur Qualitätskontrolle

4.4.1 Linter

Es werden zwei separate ESLint-Konfigurationen eingerichtet, eins für das Frontend und eins für das Backend Team, da React und Node.js unterschiedliche Anforderungen und Konventionen haben. Für React wird der Styleguide von Airbnb ausgewählt, da er speziell für die Anforderungen von React-Komponenten und JSX geschrieben wurde. Für Node.js hingegen wird der Styleguide von ESLint Recommendation verwendet, weil es auf die Bedürfnisse von Server-Code und Modulen abgestimmt ist.

Zusätzlich wird eine Lint-Stage in der GitLab Pipeline eingerichtet, welcher automatisch eine Lint-Überprüfung ausführt. Wenn dabei Warnungen oder Fehler gefunden werden, wird ein Report generiert, der alle Details auflistet und den Entwicklern die Möglichkeit gibt, diese Probleme schnell und einfach zu beheben.

4.4.2 Qualitätssicherung von Code

Um die Qualität des Codes sicherzustellen, wird das Tool SonarQube verwendet, welches in der Cloud in einem Docker-Container läuft. Die Analyse wird nur auf der Staging Branch ausgeführt.

Einmal pro Woche wird der SonarQube-Report von einer verantwortlichen Person sorgfältig analysiert. Diese Person überprüft den Report auf mögliche Probleme wie Code Smells, Bugs und Duplikate. Basierend auf dem Report werden Tickets im Jira System erstellt und den entsprechenden Teams zugewiesen.

Die zuständige Person bespricht die Tickets entweder mit dem Frontend oder Backend Team. Dies kommt darauf an wer für den betroffenen Code verantwortlich ist. Das Team wird dann gebeten, die notwendigen Änderungen vorzunehmen, um den Code zu verbessern und die Qualität zu steigern.

4.4.3 Report Qualitätssicherung Code

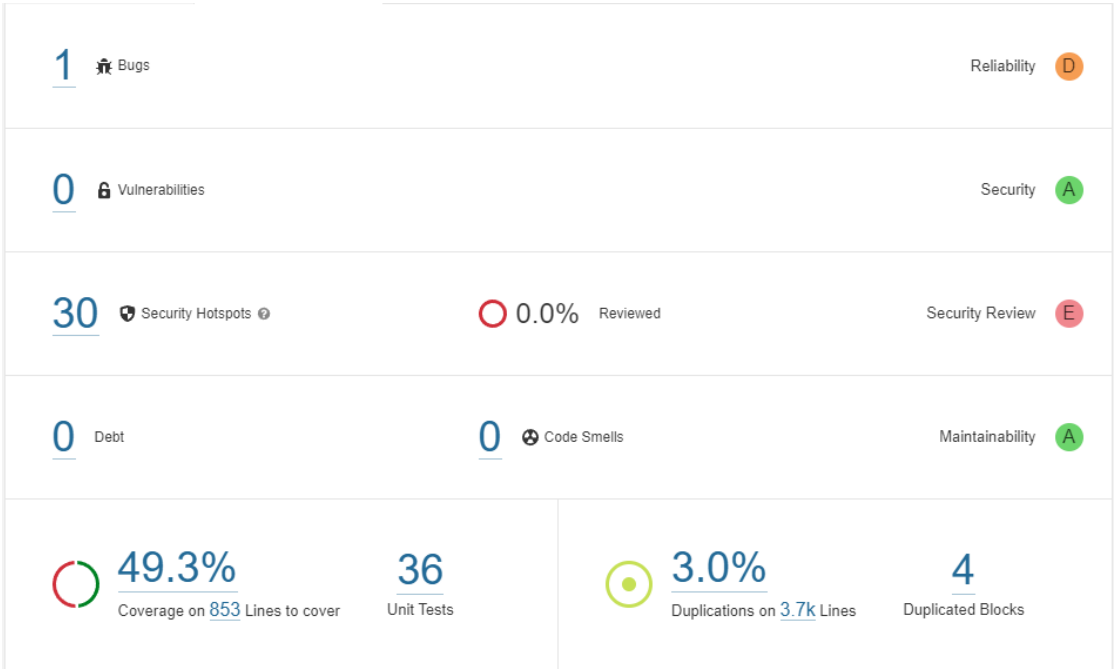

Figure 4.3: Report Qualitätssicherung Code

Die Testabdeckung von SonarQube liegt bei 50% da nur die Tests im Backend analysiert werden. Das Frontend wird anhand End to End Tests getestet, welche nicht von SonarQube analysiert werden können.

Part III

Project Documentation

Chapter 5

Initial Project Proposal

Project name: OSTShare

Team Members

1. Fabian Freitag (fabian.freitag@ost.ch)
2. Fabio Elvedi (fabio.elvedi@ost.ch)
3. Josip di Benedetto (josip.dibenedetto@ost.ch)
4. Kevin Pfister (kevin.Pfister@ost.ch)
5. Nicolas Gattlen (nicolas.gattlen@ost.ch)

Availabilities

Time slot	Mon	Tue	Wed	Thu	Fri
08h00-09h00	-	-	XR	-	-
09h00-10h00	-	-	XR	-	-
10h00-11h00	-	-	XR	-	-
11h00-12h00	-	-	XR	-	-
12h00-13h00	-	-	XR	-	-
13h00-14h00	-	-	XR	-	-
14h00-15h00	-	-	-	-	-
15h00-16h00	-	-	-	-	-
16h00-17h00	-	-	-	-	-
17h00-18h00	-	-	-	-	-
18h00-19h00	-	-	-	-	-

Vision

Wir wollen eine Lernplattform bauen, die es Studierenden und dozierenden erlaubt, ihre Informationen zu Modulen an der OST miteinander zu teilen. Studierende können ihre Zusammenfassungen und Vorlesungsnotizen hochladen und mit anderen teilen. Dozierende

können die Platform nutzen, um ihre Vorlesungsunterlagen zu verteilen. Bis jetzt wurden die Unterlagen über verschiedene Kanäle verteilt, unser Ziel ist es, den Prozess zu vereinfachen und es zu ermöglichen, dass alles an einem Ort ist.

Proposed Realisation

Das Backend könnte zum Beispiel ein NodeJS Server mit einer REST API für das Frontend sein. Das Frontend könnten wir in React implementieren. Ausserdem könnten wir PostgreSQL für unsere Datenbank verwenden.

Chapter 6

Project Plan

6.1 Organisation

6.1.1 Ressourcen

Hier werden die Ressourcen, welche uns für dieses Projekt zur Verfügung stehen, beschrieben. Dazu gehören der Zeitliche Rahmen, die Kosten und die Personas.

Zeitlicher Rahmen

Das Projekt startete am 20. Februar 2023 und dauert bis zum 2. Juni 2023. Während dieser Zeitspanne finden Projekt Reviews mit dem Projektbetreuer Timon Erhart statt. Bei den Reviews müssen die Fortschritte vorgestellt werden und das weitere Vorgehen wird besprochen. Der Aufwand des Projektes soll dabei 120 Stunden pro Person betragen.

Kosten

Da es sich hierbei um ein Schulprojekt handelt, gibt es weder Personal noch Ressourcenkosten. Die benötigte Soft-/Hardware wird sowohl von den Teilnehmenden, als auch von der Schule zur Verfügung gestellt.

Personas

In diesem Abschnitt werden die Mitarbeiter des Projektes vorgestellt.

Name:	Kevin Pfister
Studium:	Informatikstudium im 6. Semester
Arbeitserfahrung:	Arbeitet bei der Hostpoint im Customer Care Support. Ansonsten Quereinsteiger
Name:	Nicolas Gattlen
Studium:	Informatikstudium im 6. Semester
Arbeitserfahrung:	Lehre in der Systemtechnik. Mehrere Jahre Arbeitserfahrung in der Systemtechnik + Cybersecurity.
Name:	Fabian Freitag
Studium:	Informatikstudium im 6. Semester
Arbeitserfahrung:	Lehre als Elektroniker. Ansonsten Quereinsteiger.
Name:	Josip di Benedetto
Studium:	Informatikstudium im 6. Semester
Arbeitserfahrung:	Lehre als Systemtechniker. Arbeitet bei der Hostpoint im Customer Care Support.
Name:	Fabio Elvedi
Studium:	Informatikstudium im 6. Semester
Arbeitserfahrung:	Lehre als Betriebsinformatiker. Arbeitet bei der FH Graubünden.

6.1.2 Arbeitsweise

Das Projekt wird nach der Agilen Arbeitsweise durchgeführt. Aufgrund der unterschiedlichen Stundenpläne kann kein klar definiertes Agiles Framework eingehalten werden, SCRUM+ wird aber als Orientierungshilfe verwendet.

6.1.3 Rollenverteilung

Projektrollen

Projektrollen		
Title	Verantwortlich	Verantwortlichkeiten
Project Manager	Kevin, stv. Nicolas	Sicherstellen, dass das Projektteam die Projektziele erreicht und das Projekt abschliesst. Projekt Plan regelmässig aktualisieren.
Software Architect	Fabio, stv. Kevin	Bereitstellung von Architektur-Entwürfen für das DEV-Team
Frontend-Developer	Kevin, stv. Josip	Verantwortlich für die Entwicklung neuer benutzerorientierter Funktionen und die Bestimmung der Struktur und des Designs von Webseiten
Backend-Developer	Nicolas, stv. Fabio	Verantwortlich für die serverseitige Webanwendungslogik sowie die Integration des Frontend-Teils
Tester	Fabian, stv. Fabio	Tests durchführen und protokollieren, Ergebnisse auswerten und gefundene Probleme dokumentieren
Protokollierer	Josip, stv. Fabio	Team-Meetings und Reviews protokollieren

Table 6.1: Projektrollen in OSTShare

6.1.4 Meetings

Es findet einmal wöchentlich ein Meeting statt, um den Projektstatus zu besprechen. Während den wöchentlichen Meetings werden jeweils die getätigten Arbeiten, aufgetretene Probleme und anstehende Aufgaben besprochen. Die anstehenden Aufgaben werden dabei gemäss den Stärken aufgeteilt. Um Aufgaben zu erstellen und sie gerecht zu verteilen wird die Checkliste für das SE Project verwendet. Je weiter das Projekt voranschreitet, desto flexibler müssen die Aufgaben erstellt werden (je nach Status des Projektplans). Sollten es notwendig sein, wird ein zweites Meeting am Ende der Woche geplant, um Fragen und Schwierigkeiten zu klären.

6.2 Longtermplan

6.2.1 Roadmap

Die Projektplanung wird anhand von RUP gemacht. Dabei wird das Projekt in Phasen unterteilt, welche in Iterationen (Sprints) abgearbeitet werden. Zur Orientierung der Projektplanung wurde der Musterplan aus der SEProj-Introduction Vorlesung verwendet. Das bedeutet, dass die Längen der Phasen sich prozentual an den Richtwert des Musterplans orientieren.

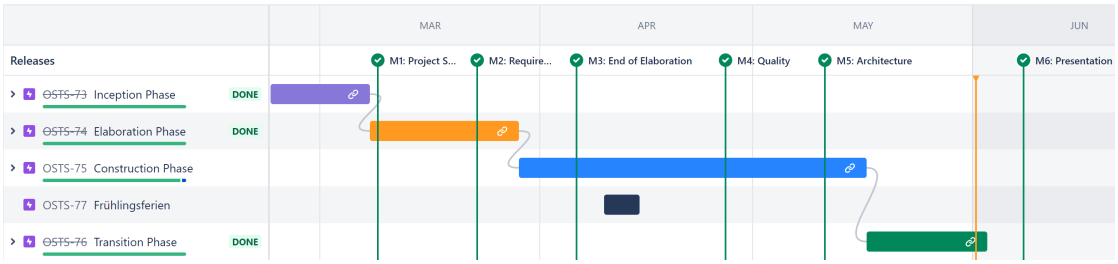

Figure 6.1: Roadmap aus Jira

Meilenstein	Enddatum
M1: Project Setup	08.03.2023
M2: Requirements	22.03.2023
M3: End of Elaboration	05.04.2023
M4: Quality	26.04.2023
M5: Architecture	10.05.2023
M6: Presentation	07.06.2023

Table 6.2: Die Meilensteine wie in der Roadmap gezeigt mit Daten

Phase	Startdatum	Enddatum	Dauer (Tage)
Inception	22.02.2023	07.03.2023	14
Elaboration	08.03.2023	28.03.2023	21
Construction	29.03.2023	16.05.2023	42
Transition	17.05.2023	02.06.2023	17

Table 6.3: Die Daten und Dauer der einzelnen Phasen

Phasen

In diesem Abschnitt werden die einzelnen Phasen beschrieben. Dabei werden die Hauptaufgaben der jeweiligen Phase definiert. Zum Schluss jeder Phase werden Meilensteine gesetzt und kontrolliert. Diese sind dazu da, dem Team den momentanen Fortschritt zu zeigen und ihm einen Überblick zu verschaffen, ob es im Zeitplan ist.

Inception (2 Wochen)

In der Inception Phase soll alles organisatorische soweit bearbeitet werden, dass die Arbeit danach ohne grossen Unterbrüche für die Projektplanung durchgeführt werden kann. Ausserdem sollte der Projektplan sowohl kurzfristig wie auch langfristig erstellt sein. Für die Inception- und Elaboration-Phase soll ein Short-Term-Plan bereit sein. Dieser soll wo nötig von Phase zu Phase angepasst werden. Zuletzt soll eine klare Vision des Projektes definiert werden, bei der die Kernfeatures des Produktes bestimmt sind.

Elaboration (3 Wochen)

In dieser Phase sollen die Funktionalen-, Nicht-Funktionalen-Requirements und das Domain Model erarbeitet werden. Ausserdem soll eine erste Skizze für das User Interface erstellt werden. Das Team soll sich in dieser Phase auch bereits über mögliche Technologien informieren und isch mit diesen auseinander setzen.

Construction (7 Wochen)

Während der Construction Phase werden die definierten Features erarbeitet. Es werden zuerst die Kernfeatures für das MVP fertig gestellt, danach werden zusätzliche Features hinzugefügt. Parallel zur technischen Implementation sollen alle Funktionen dokumentiert werden. Am Ende der Construction Phase wird das nahezu fertige Produkt an End Usern getestet. Anschlissend werden Feedback Punkte, die als extrem wichtig eingestuft werden implementiert.

Transition (2 Wochen)

In der Transition Phase wird das Endprodukt und die Dokumentation für die Abgabe vorbereitet und die Präsentation wird geplant.

6.3 Short Term Plan

Nachfolgend werden die Short-Term Pläne für die einzelnen Meilensteine beschrieben. Es ist zu beachten, dass nicht weiter als für den nächsten Meilenstein definiert wird. Ausserdem weist der Short-Term Plan viel mehr Abweichungen auf, als der Long-Term Plan. Für jeden Meilenstein wird das jeweilige Arbeitspaket definiert.

6.3.1 Inception Phase

Git Repository	Git Repository ist vorbereitet und alle User haben zugriff
Jira	Jira ist vorbereitet, User sind registriert, Issues sind erfasst
Longtermplan	Longtermplan grafisch erstellt und alle Meilensteine definiert
Shorttermplan	Arbeitspakete definiert
Risikoanalyse	Projektspezifische Risiken sind ermittelt und in Matrix dargestellt

6.3.2 Elaboration

Feedback von R1	Dokumentation wird anhand von Feedback überarbeitet
Functional Requirements	FR sind definiert
Non Functional Requirements	NFR sind definiert
Domain Model	Domain Model aufgezeichnet
UI Skizze	Skizze für UI zeichnen

6.3.3 End of Elaboration

Feedback von R2	Dokumentation/Plan wird anhand von Feedback überarbeitet
End of Elaboration	Planung ist abgeschlossen und korrekt
Initial Architecture	Erste Architektur ist geplant
Prototype	Technischer Durchstich erreicht, Prototyp funktioniert

6.3.4 Quality

Feedback von R3	Dokumentation wird anhand von Feedback überarbeitet
Quality Measures	Schritte zur Qualitätssicherung sind klar umgesetzt
Testing	Testkonzept vorhanden

6.3.5 Architecture

Feedback von R4	Dokumentation wird anhand von Feedback überarbeitet
Technical	Architektur baut auf ursprünglicher Architektur auf, ist korrekt und erweiterbar
Documentation	Dokumentation ist korrekt und up to date
MVP	Minimal Viable Product ist fertig
End User Tests	End User Tests wurden durchgeführt

6.3.6 Presentation

Feedback von R5	Dokumentation wird anhand von Feedback überarbeitet
Personal Reports	Persönliche Rückblicke sind geschrieben
Product	Kursankündigung ist fertig
Dokumentation	Dokumentation ist fertig, korrekt und aufgeräumt
Repositories	Code und Dokumentations Repositories sind aufgeräumt
Presentation	Präsentation ist vorbereitet

6.4 Issue Management

Das Ziel ist es ein Tool auszuwählen, welches einfach zu bedienen ist und bei dem sich alle Funktionen und Methoden auf deiner Plattform befinden. Ausserdem soll das Projekt eine definierte Struktur aufweisen. Das Team hat sich für Jira entschieden, da hierbei alle zu brauchenden Use-Cases abgedeckt werden und als benutzerfreundlich abgestempelt werden können.

6.4.1 Issues

In Jira gibt es verschiedene Typen von Issues, darunter Story, Bug, Epic, Task und Subtask. Nachfolgend eine Beschreibung wie diese Typen in diesem Projekt verwendet werden.

Typen von Issues	Beschreibung
Epic	Epics werden als Phasen des Longtermplans dargestellt
Story	Storys werden verwendet, sobald sich eine Anforderung oder eine Funktion auf eine spezifische Benutzeraktion oder ein Benutzerziel fokussiert.
Task	Tasks werden verwendet, wenn es sich um allgemeine Aufgaben handelt.
Subtask	Falls eine Story oder ein Task weiter unterteilt werden muss, kann ein Subtask erstellt werden.
Bugs	Sollten während der Entwicklung Bugs auftreten, können diese als separate Bugs in Jira erstellt werden.

Table 6.4: Beschreibung der verschiedenen Typen von Issues

Die Issues in Jira werden hierarchisch strukturiert. In diesem Projekt werden Stories, Task und Bugs einem Epic zugeordnet. Subtasks hingegen werden nur zu Stories und Tasks zugeteilt.

6.4.2 Components

Jeder Issue wird einem Component zugeordnet, der dazu dient Aufgaben zu gruppieren. In anderen Worten kann ein Component als Tag betrachtet werden. Nachfolgend sind die Components aufgeführt, die definiert wurden. Die Liste wird regelmässig aktualisiert. Zusätzlich werden die Components für die Zeitauswertung benötigt.

Component	Description	Component lead	Default assignee	Issues	
Project Management	Component for Project Management		Project default	34 issues	•••
Meetings	Component for Meetings		Project default	2 issues	•••
Infrastruktur	Component for Infrastruktur		Project default	10 issues	•••
Frontend	Component for Frontend		Project default	1 issue	•••
Backend	Component for Backend		Project default	1 issue	•••

Figure 6.2: Definierte Components

6.4.3 OSTShare Board

Das OSTSharing Board ist so strukturiert, dass hauptsächlich drei Features verwenden werden: Roadmap, Backlog und Kanban Board. Die zwei Features Backlog und Kanban Board werden in ihrem jeweiligen Abschnitt im Detail erklärt.

Backlog

Das Backlog wird dazu verwendet, ein Übersicht über die offenen Aufgaben zu geben und zu zeigen wer für sie verantwortlich ist. Damit RUP-Iterationen (Shorttermplan) in der Roadmap richtig dargestellt werden, werden sogenannte Versions erstellt und die Issues / Tasks den jeweiligen Iteration zugewiesen. Diese Versions sind die Meilensteine.

Kanban Board

Das Kanban Board dient als Hilfestellung, um den Zustand und den Fortschritt eines Issues / Tasks zu überwachen. Sobald sich der Zustand eines Issues / Tasks ändert, werden automatisch alle per E-Mail benachrichtigt

6.5 Risk Assessment

Risiken			
Risiko	Milderung	Warscheinlichkeit	Konsequenz
Minimale Erfahrung mit React	Einige Teammitglieder besuchen das Modul Web Engineering 2, wo es unter anderem um React geht. Dort werden die Basics gelernt, und alle Teammitglieder haben Erfahrung mit NodeJS, was bei React hilfreich sein kann	Mittel	Mehr Zeit für die Umsetzung des Frontends erforderlich, andere Teammitglieder müssen einspringen und können sich nicht auf Ihre Arbeit konzentrieren
Wenig Erfahrung mit Git	Prozess um Anpassungen am Repository vorzunehmen wird mit einer Anleitung dokumentiert. Merge-Request werden von anderen Teammitglieder überprüft und genehmigt	Niedrig	Änderungen können überschrieben werden. Dadurch wird viel Zeit in das Beheben von falsch gelösten Konflikten investiert
Unsaubere Definition der API Schnittstelle	Bevor der Implementierung eines neues Features werden die entsprechenden API Schnittstellen klar definiert, damit das Frontend und Backend Team parallel arbeiten können	Mittel	React Abrufe auf der API lieferen unerwartete oder keine Daten

Table 6.5: Risiken

6.5.1 Risk Matrix

Auswirkung / Wahrsch.	Schwach - 1	Minimal - 2	Moderat - 3	Hoch - 4	Schwer - 5
Sehr hoch - 4	(leer)	(leer)	(leer)	(leer)	(leer)
Hoch - 3	(leer)	(leer)	(leer)	(leer)	(leer)
Mittel - 2	(leer)	Wenig Erfahrung mit Git (4)	Unsaubere Definition der API Schnittstelle (6)	Minimale Erfahrung mit React (9)	(leer)
Niedrig - 1	(leer)		(leer)	(leer)	(leer)

Table 6.6: Risk Matrix

6.5.2 Risk mitigation

03.04.2023

- Minimale Erfahrung mit React (von hoch/hoch auf mittel/hoch)
  - Frontend Team konnte durch eigenständiges Lernen die Basics von React erlernen
- geringen Git Erfahrung (von hoch/moderat auf mittel/moderat)
  - Eine Anleitung wurde erstellt, welche erklärt wie das Git benutzt werden soll
- unsaubere Definition der API Schnittstelle (von hoch/minimal auf mittel/minimal)
  - Mithilfe von Postman wurde eine API Schnittstellendonumentation erstellt

07.05.2023

- Minimale Erfahrung mit React (von mittel/hoch auf mittel)
  - Die Kenntnisse bezüglich des React Frameworks konnten erweitert werden und die Fortschritte werden von Woche zu Woche grösser.
- geringen Git Erfahrung (von mittel/moderat auf gering)
  - Durch das Einhalten von Deadlines und steigender Erfahrung ist es dem Team möglich Merge Conflicts zu vermeiden oder schnell zu beseitigen.
- unsaubere Definition der API Schnittstelle (keine Änderungenen)
  - Mithilfe von Postman wurde eine API Schnittstellendonumentation erstellt

Chapter 7

Time Tracking Report

7.1 TimeTracker

Die Zeiterfassung wird durch eine Third Party App namens TimeTracker verwaltet, die mit Jira kompatibel ist. Anschliessend werden die Daten von Jira exportiert und mit einem Python Script ein grafischer Zeitreport erstellt. Jeder Zeitreport zeigt den Ist- und Sollzustand.

7.2 Auswetung

Die Zeiterfassung wird vor den Reviews aktuallisiert und jeweils als Screenshot in diesem Kapitel aufgeführt

7.2.1 Gesamtprojekt

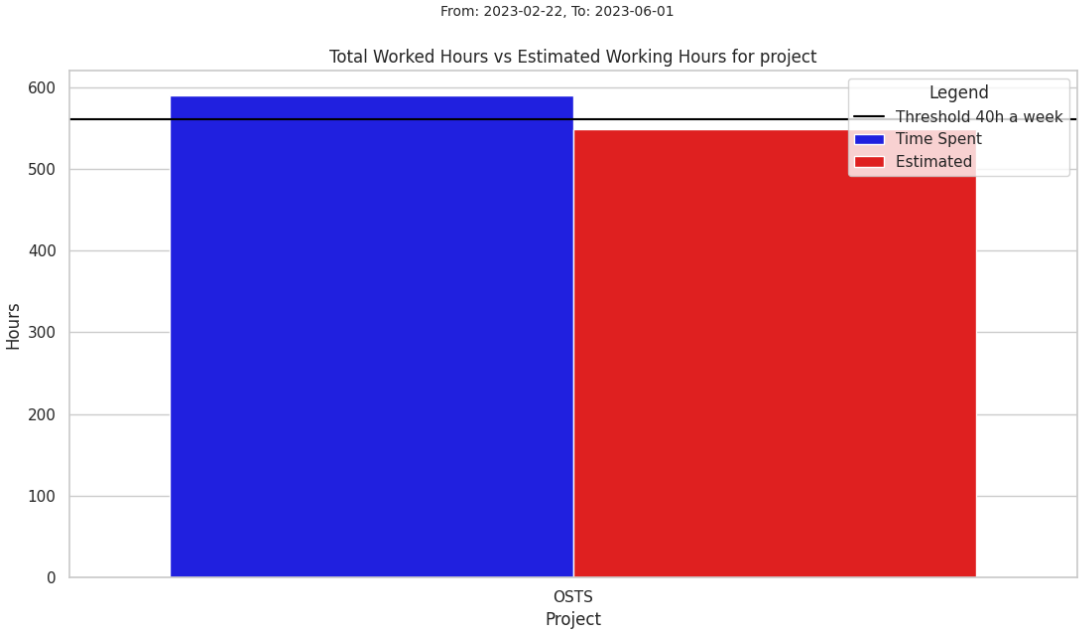

Figure 7.1: Time Report Gesamtprojekt

7.2.2 Pro Phase

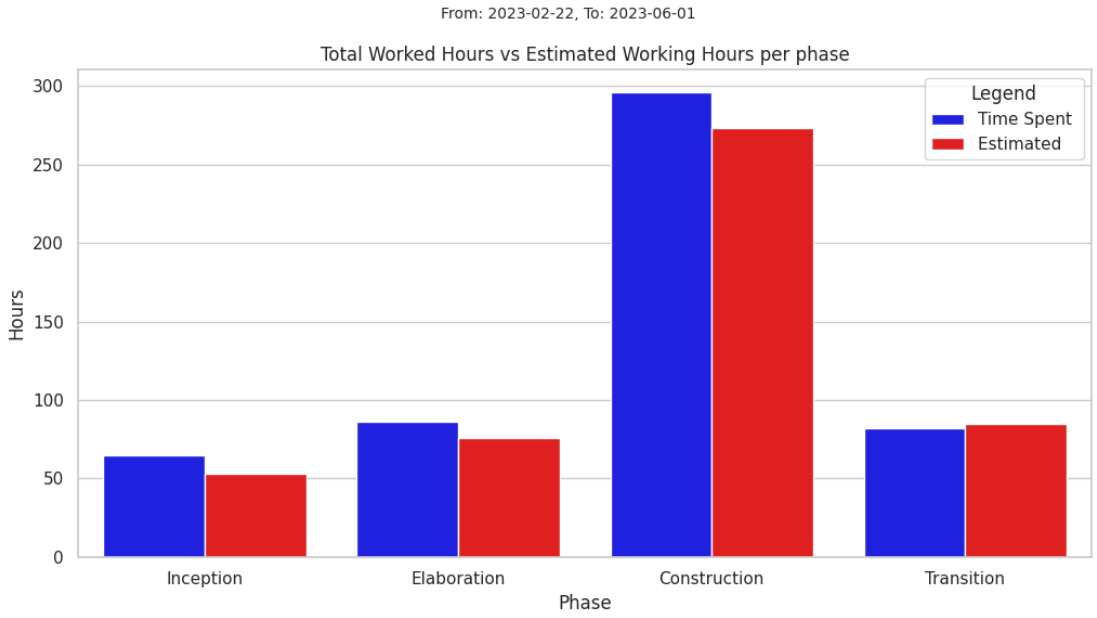

Figure 7.2: Time Report pro Phase

7.2.3 Pro Person

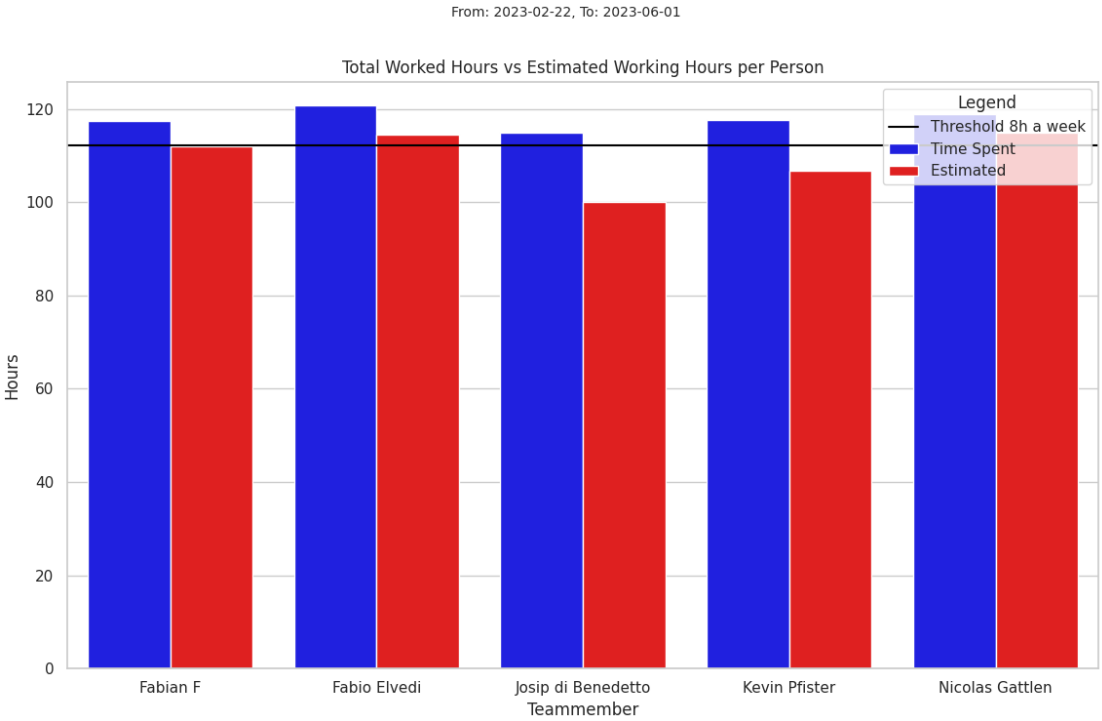

Figure 7.3: Time Report pro Person

Chapter 8

Personal Reports

8.1 Personal Report: *Fabio Elvedi*

Zu Begin des Projektes war das Zusammenarbeiten sehr schwierig und ineffizient. Das lag daran, dass wir uns nicht persönlich kannten und die Kommunikation nicht immer optimal war. Nachdem wir uns jedoch besser kennengelernt haben und uns besser organisieren konnten, lief die Zusammenarbeit viel besser. Schwierig war es auch, den Issue Tracker (JIRA) zu verwenden, da wir nicht daran gewöhnt waren. Am Anfang fühlte sich JIRA mehr nach einer TODO Liste an, als nach einem Issue Tracker, es war schwierig denn aktuellen Stand des Projektes zu sehen. Jedoch haben wir uns nach einiger Zeit daran gewöhnt und es hat gut funktioniert. Wir haben uns gut aufgeteilt und jeder hat seinen Teil zum Projekt beigetragen. Ich war vor allem um die Programmierung und das Produkt verantwortlich. Ich habe mich sehr gefreut, dass wir das Projekt schlussendlich gut gemeistert haben und dass wir ein gutes Produkt abliefern konnten. Zusätzlich fand ich es auch gut, dass Timon bei den Reviews immer sehr konstruktive Kritik gegeben hat, welche uns geholfen hat, das Projekt zu verbessern.

8.2 Personal Report: *Fabian Freitag*

Im grossen und ganzen bin ich der Meinung, dass die Durchführung des Projektes gut funktioniert hat. Zu beginn mussten wir lernen, wie wir am besten zusammenarbeiten, entsprechend ineffizient waren unsere Sitzungen. Wir hatten auch ein bisschen mühe, die Dokumentation richtig zu schreiben. Nach einer Weile haben wir dann aber als Team gut zusammen gearbeitet und dann sind wir auch gut vorwärts gekommen. Gegen Ende der Construction Phase hatte ich nochmal das Gefühl, dass wir nicht wirklich vorwärts kommen, dann waren aber alle auf einen Schlag mit ihren Tasks fertig und das Produkt hat nochmals einen enormen Sprung gemacht, so dass wir pünktlich mit den von uns als wichtig eingetuften FRs fertig wurden. Ich denke an disem Punkt ist bei uns noch eher Verbesserungspotenzial vorhanden. Ich war mir öfters nicht sicher, wie Weit die anderen Teammitglieder mit ihren Tasks sind, vielleicht wäre es besser, wenn wir unseren fortschritt öfters auf das Repo pushen, damit wir sehen, dass etwas läuft.

Ich persönlich habe mich sehr gefreut, als ich die End User Test mit Studierenden aus anderen Studiengängen, die nichts mit dem Projekt zu tun hatten, durchgeführt habe und gesehen habe, wie sie mit unserem Produkt gearbetet haben. Dabei haben wir nur kleine Verbesserungen als Feedback bekommen.

8.3 Personal Report: *Nicolas Gattlen*

Während unseres Projekts war ich für die Backend-Entwicklung zuständig und arbeitete eng mit einem Teamkollegen zusammen. Unsere Zusammenarbeit verlief effizient, und wir konnten die Aufgaben fair untereinander aufteilen. Die Zusammenarbeit mit den anderen Teammitgliedern war ebenfalls angenehm und produktiv. Zu Beginn des Projekts stiessen wir auf einige Herausforderungen, insbesondere mit Git Merge-Konflikten. Wir konnten diese Probleme bewältigen, in dem wir alle Merge-Konflikte lösten und eine hilfreiche Anleitung erstellten, die allen half, solche Konflikte zu vermeiden. Ein weiterer Aspekt, den wir verbessern konnten, war die Effizienz unserer Meetings. Anfangs dauerten unsere Besprechungen oft zu lange, sodass wir Zeit verloren, die wir für die eigentliche Projektarbeit hätten nutzen können. Als Lösung entschieden wir uns dafür, die Meetings nicht unnötig in die Länge zu ziehen. Diese Anpassung ermöglichte es uns, uns wieder zu fokussieren und unsere Zeit effektiver zu nutzen. Die größte Herausforderung für mich bestand darin, die Pipeline in GitLab einzurichten. Wir hatten Schwierigkeiten beim Starten der Container und investierten viel Zeit in die Automatisierung der Tests. Doch durch gemeinsames Problemlösen konnten wir diese Herausforderungen meistern und die Pipeline erfolgreich konfigurieren. Das größte Highlight des Projekts war die Möglichkeit, unseren eigenen Prototypen zu entwickeln und das gesamte Projekt zu simulieren. Diese Erfahrung ermöglichte es mir, meine Kenntnisse in React, Node.js, GitLab Pipeline und Docker Compose zu erweitern. Insgesamt war das Projekt eine wertvolle Erfahrung, die meine Fähigkeiten weiterentwickelt hat. Ich schätzte die gute Zusammenarbeit im Team und die Möglichkeit, mich in verschiedenen Technologien weiterzubilden.

8.4 Personal Report: *Kevin Pfister*

Ich empfand die Zusammenarbeit zu beginn eher schwierig. Jeder hatte eine andere Vision und mann musste öfters genau abklären was der andere meint. Den Vorgegebenen Workflow mit den einzelnen Phasen sah ich zu Beginn als Mühsam und ineffizient an. Ich erkannte jedoch während der Arbeit wie wichtig es war alles zu dokumentieren. Denn aus unserer Dokumentation entstanden schliesslich die Tickets im Jira und jeder wusste was er zu tun hatte. Ich bekam während des Projektes immer mehr den Durchblick und wusste stets woran ich arbeite und weshalb meine Arbeit wichtig war. Ebenfalls konnte ich meine Programmierkenntnisse im Verlaufe des Semesters stark verbessern. Ich merkte wie ich am Anfang des Projektes viel Mühe mit React hatte. Diese Probleme wurden jedoch während des Projektes immer weniger und ich merkte wie meine Produktivität stieg. Was ich ebenfalls gemerkt habe, ist dass ich mich im Bereich der Organisa-

tion noch verbessern muss, es geschah mir mehr als nur einmal dass ich Änderungen an unserem Projekt vorgenommen habe und diese anschliessend nicht gepusht habe, dies führte dann zu mühsamen Merge requests, welche einiges an Zeit beansprucht haben. Alles in allem sehe ich jedoch das Projekt als gelungen und als einen guten Einblick in die Software-Entwicklung.

8.5 Personal Report: *Josip Di Benedetto*

Ich war für die Entwicklung des Frontends zuständig. Während des Projekts hatten wir vor allem zu Beginn ein wenig Schwierigkeiten, mit dem Projekt loszulegen. Wir hatten zwar eine "Vision", wie das Endprodukt aussehen sollte, aber die meisten von uns haben noch nie im Team eine Software entwickelt, und wir haben uns persönlich auch nicht gekannt. Das hat die Zusammenarbeit ein wenig schwierig gemacht. Im weiteren Projektverlauf wurde die Zusammenarbeit effizienter, und wir konnten uns relativ schnell einigen. Eine grosse Herausforderung für mich (und wahrscheinlich auch für einige andere) war die kollaborative Zusammenarbeit mit Git. Ich habe persönlich nicht wenig Zeit aufgewendet, um Git Konflikte zu lösen. Insgesamt verlief das Projekt ohne grössere Schwierigkeiten nach Plan. Etwas, mit dem ich mich höchstwahrscheinlich nie anfreunden werde, ist Jira. Trotzdem war die kollaborative Zusammenarbeit mit Jira eine wertvolle Erfahrung für mich, denn diesem System werde ich vermutlich auch in der Arbeitswelt nicht ausweichen können. Deshalb finde ich es toll, dass ich ein wenig Erfahrung damit sammeln konnte. Das gilt natürlich auf für andere Technologien, welche wir verwendet haben, wie bspw. React, Latex, Git und Docker. Die Arbeit mit React hat mir gezeigt, wie moderne Web-Applikationen entwickelt werden können, und mich auch dazu animiert ein eigenes kleines Projekt damit zu starten. Verbesserungspotenzial sehe ich persönlich bei den Meetings. Diese wurden vor allem am Anfang etwas in die Länge gezogen, aber gegen Ende des Projekts fand ich das Zeitmass sehr vernünftig. Mein Fazit ist, dass ich meine Web-Dev Skills gut verbessern konnte und auch einigermassen vernünftig mit Git und Docker umgehen kann. Ich hatte bereits Erfahrung mit Docker, aber dieses Projekt hat mein Wissen etwas aufgefrischt. Ich freue mich auf zukünftige Projekte wie die SA und BA und bin sicher, dass ich bei diesen Arbeiten auf mein gewonnenes Wissen in diesem Projekt zurückgreifen kann.

Chapter 9

Meeting Minutes

9.1 Review Meetings

9.1.1 Review 0

Datum und Zeit	Mittwoch 22 Februar 2023 09:00
Ort	Gebäude 1
Teilnehmer	T.Erhart, F.Freitag, F.Elvedi, J.Benedetto, K.Pfister, N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
Ferien	In der Woche 17.05.2023 Projektberater weg	**T.E**
Git	Commit der bewerteteten Abgabe wird entsprechend markieren	**F.E**
Technischer Durchschnitt	Datenbankanfrage funktioniert. Techniken die wir einsetzen, müssen mal funktionieren. Danach kann der Durchschnitt (Tunnel - "Hello World") ausgebaut werden. Technik sollte bis Review 3 funktionieren	**Alle**
Risiko-Analyse	Risikoanalyse laufend neu bewerten	**J.B**
Agil	Projektplanung sollte agilen Ansatz verfolgen	**Alle**
Review 1	08.03.2023 Bewertung "Initial project setup" (Am Dienstagmorgen senden, einen Tag davor)	**Alle**

9.1.2 Review 1

Datum und Zeit	Mittwoch 08 März 2023 09:00
Ort	Teams
Teilnehmer	T.Erhart, F.Freitag, F.Elvedi, J.Benedetto, N.Gattlen
Entschuldigungen	K.Pfister

Thema	**Notizen**	**Verantwortlich**
Kapitel Struktur	Macht an bestimmten Orten keinen Sinn. Beispielsweise "Rollenverteilung" unter "Long term plan"	**Alle**
Scrum	Wir arbeiten nicht zu 100 prozent nach Scrum (bspw. kein festgelegter Product Owner). Stattdessen sollte man beschreiben, man arbeite agil. Alles zu Scrum / agiler Arbeitsweise unter ein Kapitel	**J.B**
Lang- / Kurzzeitplanung	Für Langzeitplanung orientiert man sich an RUP, für Kurzzeitplanung an agilen Methoden	**Alle**
Elaboration Phase	Mehr Zeit einrechnen, als vorgesehen ist	**F.E**
Meeting Planung	Sich nicht zu stark fixieren, bspw. keine bestimmte Wochentage und Uhrzeiten für Meetings festlegen. Es reicht, wenn man schreibt, man werde sich wöchentlich treffen	**J.B**
Langzeit Planung	GANTT Diagramm in Jira abbilden. Start- und Enddatum für Phasen definieren. Meilensteine sollten fixes Datum haben. Kapitel "Jira" umbenennen, bspw. Issues Management oder Time Management	**N.G**
Risk Assessment	Risiken sind zu generisch. Man sollte mehr projektspezifische Risiken auflisten	**J.B**
Arbeitszeit	In Jira sollte man eine kurve sehen, welche visuell das Verhältnis zwischen geplanter und aufgewendeter Zeit aufzeigt	**Alle**

9.1.3 Review 2

Datum und Zeit	Mittwoch 22 März 2023 09:00
Ort	Teams
Teilnehmer	T.Erhart, F.Freitag, F.Elvedi, J.Benedetto, K. Pfister N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
Akteure	Dozenten können Fragen stellen wie auch beantworten	**F.F**
Use Case Diagramm	Es sollte ersichtlich sein, wenn ein Benutzer eingeloggt ist. Vererbung verwenden, damit weniger Linien sich überschneiden	**F.F**
NFR	Anderes Wort für Gegenmassnahmen verwenden, NFR 5 nochmals anschauen und evtl. zu anderen hinzu nehmen	**F.E**
Domain Model	CRUD-Klassen statt Transaktionsklassen verwenden, Domain Model gemäss Besprechung anpassen	**J.B**
Zeit- management	Überprüfen und ggf. anpassen. Vor der Abgabe kontrollieren	**N.G**
Meilensteine	M4 und M5 eine Woche nach hinten schieben, Datum bei M6 anpassen	**N.G**
Risk Assessment	Auswirkung x Warscheinlichkeit bei Risk Matrix einfügen	**J.B**
Abgabe	Abgabe ist neu am Montag Mittag	**Alle**

9.1.4 Review 3

Datum und Zeit	Mittwoch 05 April 2023 09:00
Ort	Teams
Teilnehmer	T.Erhart, F.Freitag, F.Elvedi, J.Benedetto, K. Pfister N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
API Schnittstelle	Als Anhang in Doku	**F.E**
C4 Modell	Deployment sollte vorhanden / sichtbar sein	**F.F**
Tools zur Quialitätskon	Linting sollte von Pipeline geprüft werden	**N.G**
Metrics	Überlegen, wie wir an die Metrics kommen	**Alle**
Short Term Plan	Milestones mit Phasen vermischt, muss angepasst werden	**F.F**
Zeiplanung	Grosser Gap bei der Inception Phase	**N.G**
MVP	Man sollte einfacher sehen, welche Requirements erfüllt werden müssen	**F.E**
Code Repo	Staging Branch als default setzen	**N.G**

9.1.5 Review 4

Datum und Zeit	Mittwoch 26 April 2023 09:00
Ort	Teams
Teilnehmer	T.Erhart, F.Freitag, F.Elvedi, J.Benedetto, K. Pfister N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
Definition of Done	Noch nicht nach dod gemacht	**Alle**
Short term plan	Aktualisieren	**F.F**
Domain Model anpassen	Kommentarklasse erstellen	**J.B**
C4 Model	Aktualisieren, code UML weglassen, db lassen	**F.E**
End to end tests	Sollten früher sein, Auswertung in Anhang	**F.F**
Code Tests	Aktuell halten	**N.G, F.E**
Zeit- erfassung	Prüfen	**Alle**
Test- abdeckung	Sinnvoll messen	**N.G, F.E**

9.1.6 Review 5

Datum und Zeit	Mittwoch 10 Mai 2023 09:00
Ort	Teams
Teilnehmer	T.Erhart, F.Freitag, F.Elvedi, J.Benedetto, K. Pfister
Entschuldigungen	N. Gattlen

Thema	**Notizen**	**Verantwortlich**
Browsertests	Dokumentieren	**F.E**
Tests	Alles testen und protokollieren	**N.G**
Beispiel 9.5	Eventuell mit openapi schöner gestalten	**F.E**
Präsentation	Jeder sollte etwas sagen, Folien, wenig Text, kurze Demo	**Alle**
Personal Reports	Muss erstellt werden, ca. 0.5 Seiten pro Person	**Alle**
Management Summary	Ca. 1-2 Seiten, Fliesstext und 1-2 Diagramme	**F.E**
Zeit- erfassung	Prüfen	**Alle**

9.2 Team Meetings

9.2.1 Meeting 22.02.2023

Datum und Zeit	Mittwoch 22 Februar 2023 09:45
Ort	Gebäude 1
Teilnehmer	F.Freitag, F.Elvedi, J.Benedetto, K.Pfister, N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
JIRA	Jira als Projektplanungssoftware	**N.G**
R1:P1	RUP sollte bis 01.03.2023 09:00 vorhanden sein (Phasen, Iterationen)	**F.F**
R1:P2	Überlegen, welche Rollen es im Team gibt	**F.E**
R1:P3	Protokollierung der Sitzungen, Ziele der Besprechung festlegen	**J.B**
R1:P4	Groben Zeitplan (Phasen, Meilensteine) für das Projekt festlegen	**K.P**
R1:P5	Grobe Zeitschätzungen für das gesamte Projekt (basierend auf grobem Zeitplan)	**K.P**
R1:P6	Kurzzeitplan für nächste Iteration erstellen	**K.P**
R1:P7	Zeitschätzungen für nächste Iterationen festlegen	**K.P**
R1:P8	Offensichtliche Risiken identifizieren, Warscheinlichkeit und Schweregrad abschätzen	**J.B**
R1:P8	Risiko Assessment laufend an das Projekt anpassen	**J.B**

9.2.2 Meeting 01.03.2023

Datum und Zeit	Mittwoch 01 März 2023 09:00
Ort	Gebäude 1
Teilnehmer	F.Freitag, F.Elvedi, J.Benedetto, K.Pfister, N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
JIRA	Demonstration JIRA von Nicolas	**N.G**
R1:T1	Es sollen zwei Repositories erstellt werden, um Code und Dokumentation zu trennen	**F.E**
R1:P1	SCRUM plannung wurde reviewd und akzeptiert, backuptermin am Freitag streichen	**F.F**
R1:P2	Rollen wurde verteilt	**F.E**
R1:P3	Meeting Prozess müssen genaur definiert werden	**J.B**
R1:P4	Resourcen, Phasen und Meilensteine wurde reviewed und akzeptiert, TODO iteration bei SEP2 übung nachfragen	**Alle**
R1:P5	Wurde reviewed und akzeptiert	**K.P**
R1:P6	Wurde reviewed und akzeptiert, muss noch in JIRA nachgetragen werden	**K.P**
R1:P7	Wurde reviewed und akzeptiert, muss noch in JIRA nachgetragen werden	**K.P**
R1:P8	Wurde reviewed und akzeptiert	**K.P**
R1:P9	Wurde reviewed und akzeptiert, Matrix für Risiken erstellen	**J.B**

9.2.3 Meeting 15.03.2023

Datum und Zeit	Mittwoch 15 März 2023 09:00
Ort	MS Teams
Teilnehmer	F.Freitag, F.Elvedi, J.Benedetto, K.Pfister, N.Gattlen
Entschuldigungen	Keine

Thema	**Notizen**	**Verantwortlich**
R2:F2	Timon bzgl. Login Use-Case gefragt und mit Team angeschaut	**F.E**
Gitlab	Dev Environment mit Tests ist bereit	**F.E**
Allgemein	Josip hat versehentlich Dokument-Änderungen von Fabian und Nicolas überschrieben	**J.B**
Quality Measures	Sollten bis zum nächsten Meeting definiert sein	**Alle**
Short Term Plan	Für Review 3 planen	**F.F**
Allgemein	Auf Samstag Mitternacht sollten alle Issues abgeschlossen sein, damit Kevin am Sonntag das Dokument kontrollieren kann	**Alle**
Langzeit-Planung	Nicolas hat Python Script geschrieben, welches Soll- und Ist-Zeitaufwand visualisiert	**N.G**
Langzeit-Planung	GANTT Diagram besprochen	**N.G**
FR und NFR	Jeder soll es bis Samstag anschauen und dem Ersteller Feedback geben	**F.E**
Domain Model	Fertig machen und in Dokumentation hochladen	**J.B**
User Interface	UI Skizze erstellen und am Freitag dem Team präsentieren	**J.B**

Bibliography

Part IV

Appendix

appendix

9.3 UI Sketch

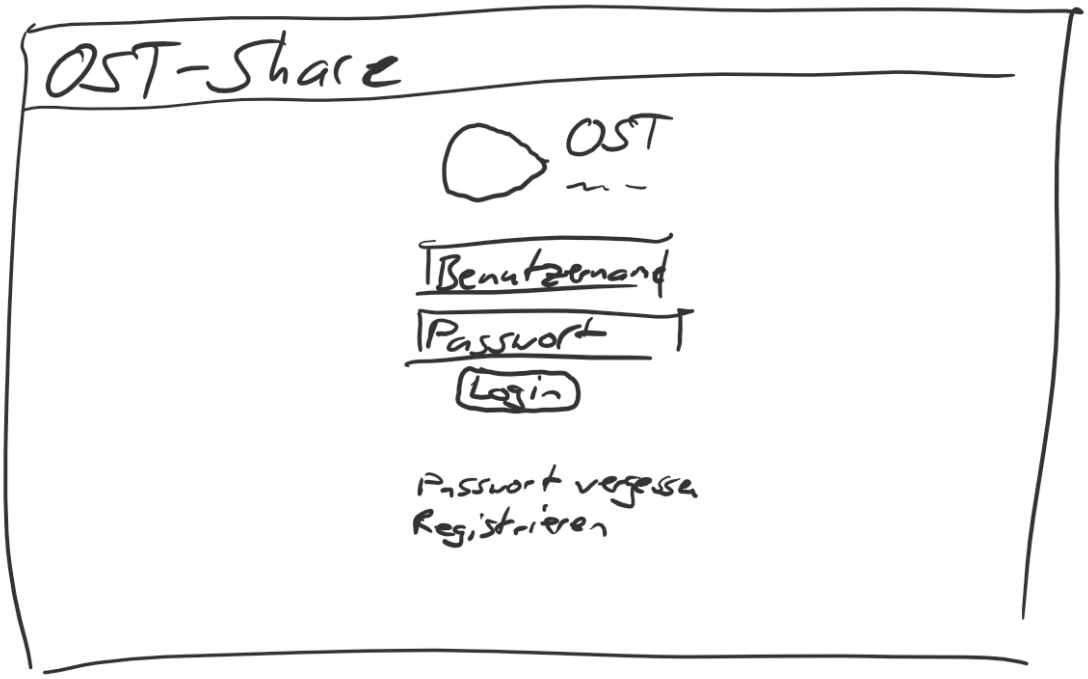

Figure 9.1: Skizze der Login Seite

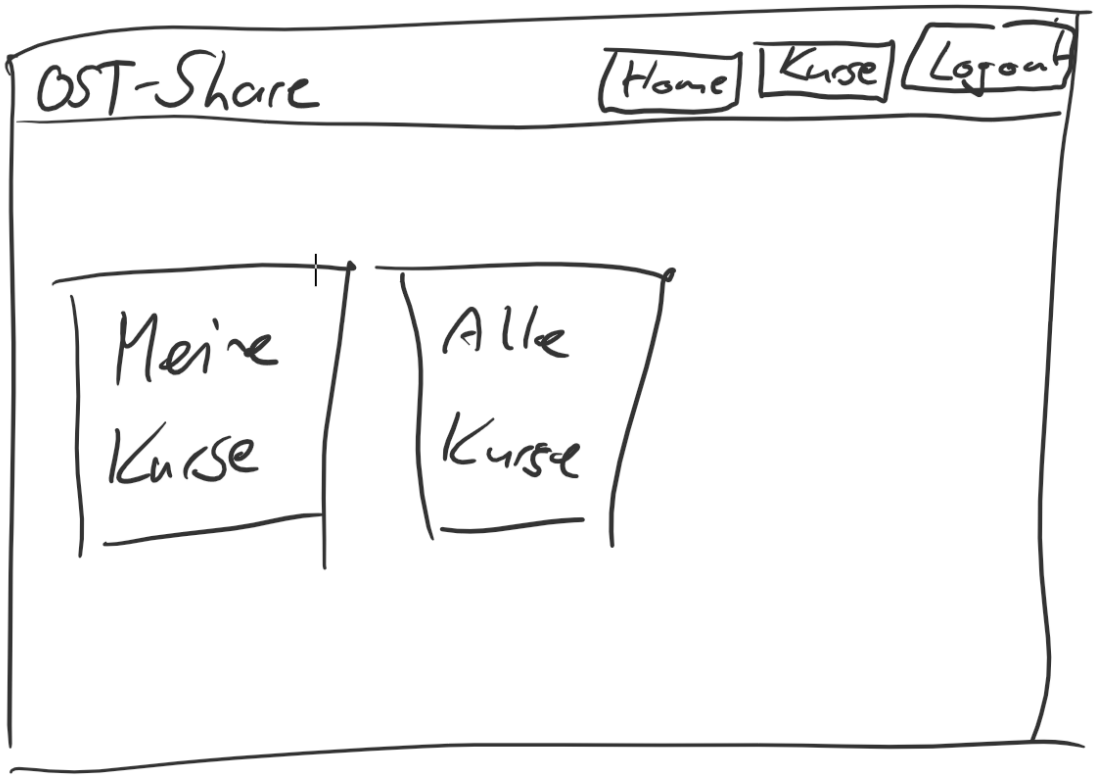

Figure 9.2: Skizze der Startseite, nach dem Login

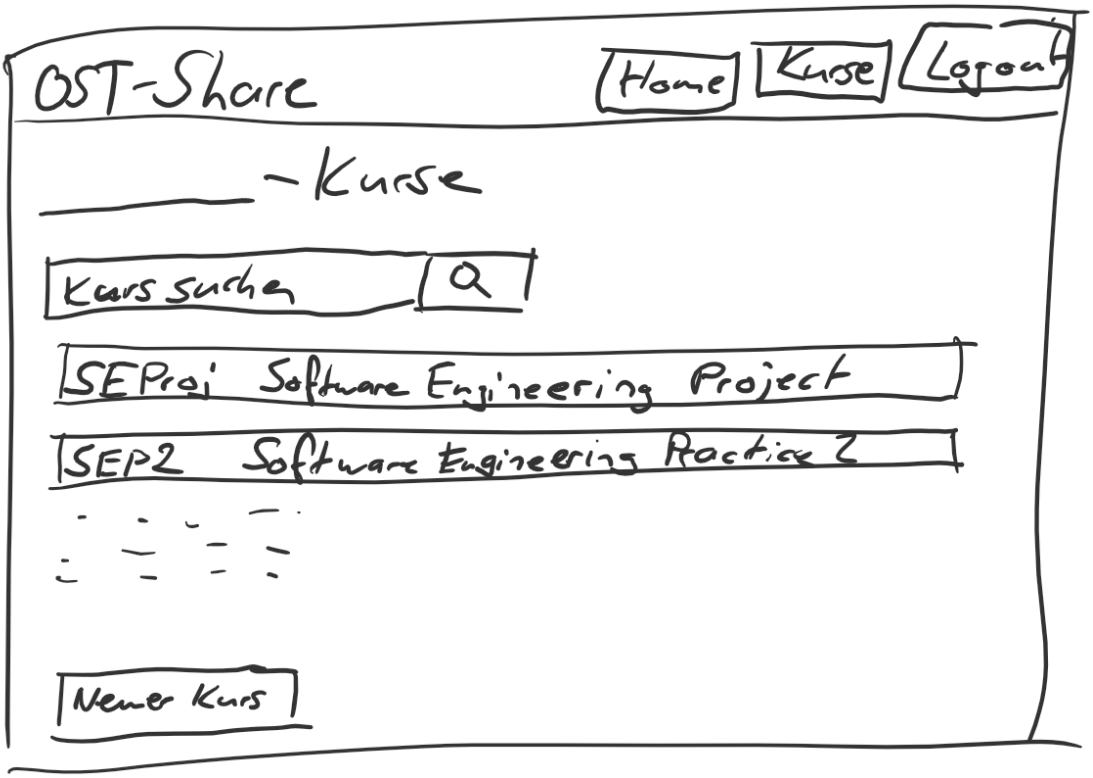

Figure 9.3: Skizze der Auflistung der Kurse

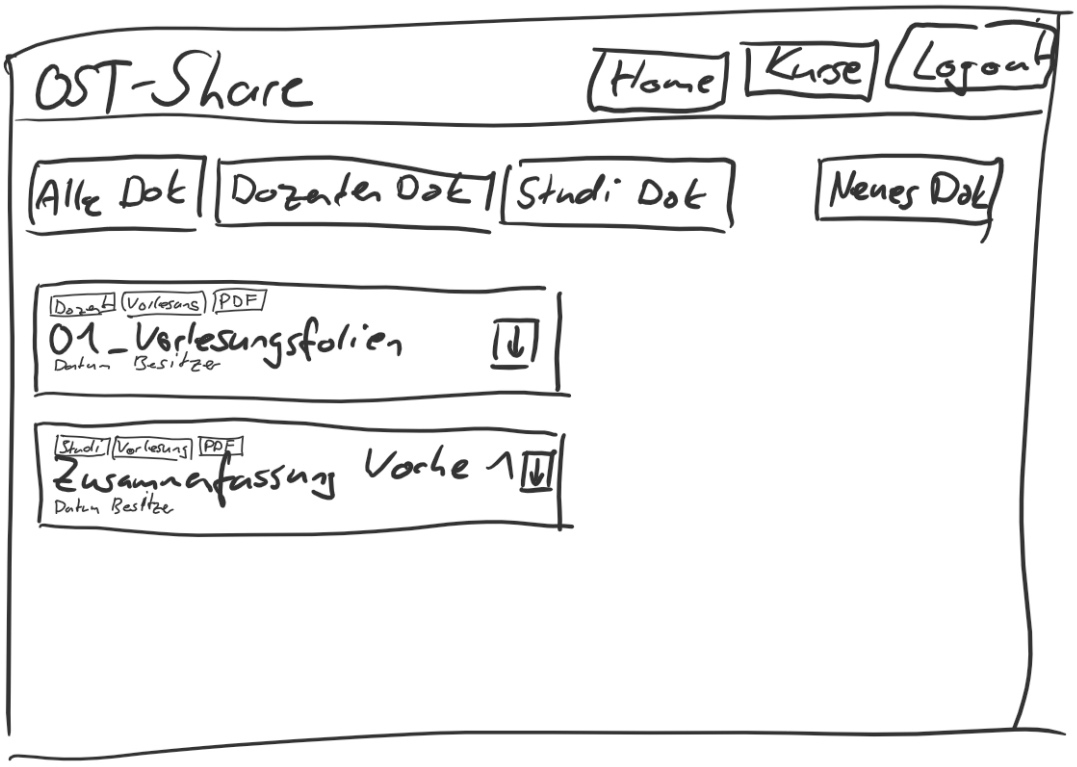

Figure 9.4: Skizze der Auflistung der Dateien innerhal beines Kurses

9.4 Testprotokoll

Titel des Test					
Test Nr.	**Beschreibung der Auszuführenden Aktion**	**Beschreibung des zu erwartenden Ergebnis/ zu Überprüfende Punkte**	**Tatsächliches Ergebnis**	**OK/NOK**	**Anmerkungen**
00_Bsp	Benutzer gibt gültigen Benutzername und korrektes Passwort in die entsprechenden Felder ein und drückt "Login"	Der Benutzer wird eingeloggt und auf die Übersichtsseite weitergeleitet. Oben rechts in der Navigationsleiste ist ersichtlich, dass der Benutzer nun eingeloggt ist.	Main Page wird angezeigt, Nutzername wird angezeigt	OK	-
1					
2					

Figure 9.5: Beispiel des Testprotokolles

9.5 API Dokumentation

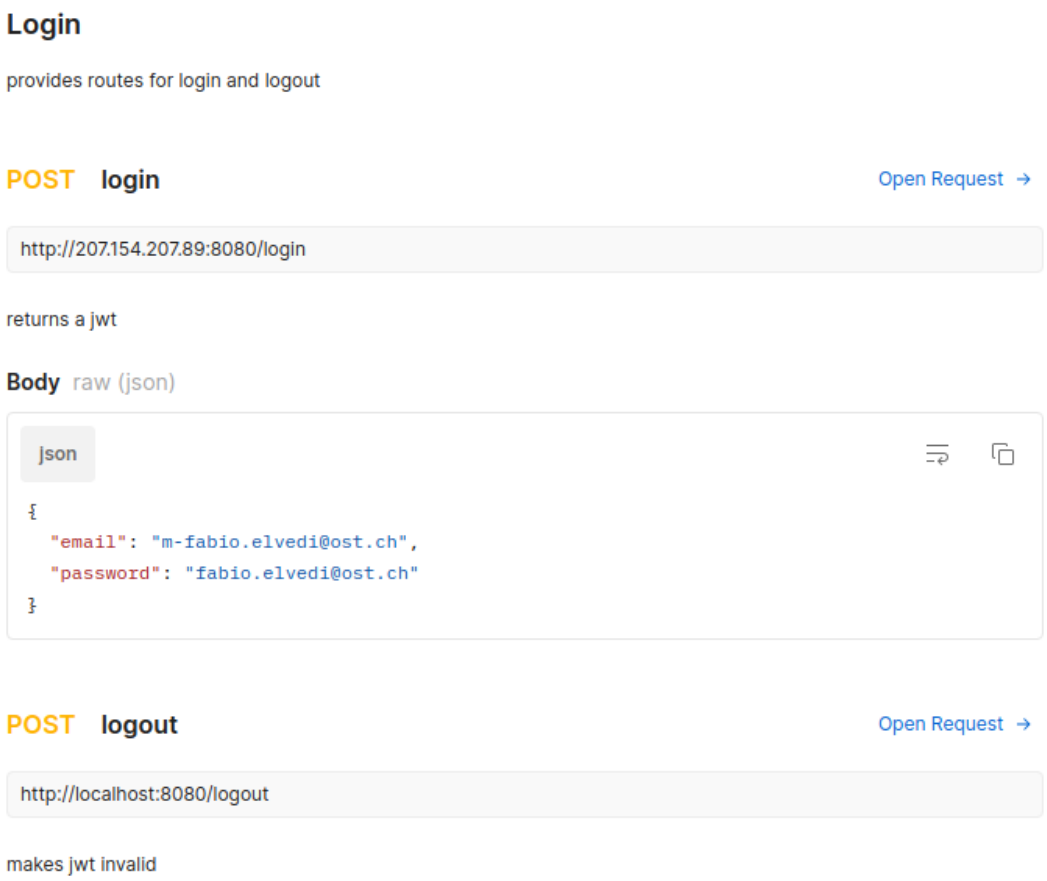

Figure 9.6: API Login

Course

manage courses

POST create

Open Request →

http://localhost:8080/courses/create

creates a course

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoifSwiaWF0IjoxNjgxODkzNjQxLCJleHAiOjE2ODE4OTcyNDF9.-CLmxnETECn75kGy6ZHuHCyi60IRRZlS2_qcGqlUP9g

Body raw (json)

```
json
{
  "title": "test course",
  "description": "test description",
  "tags": "test tag"
}
```

GET list

Open Request →

http://localhost:8080/courses

lists all courses

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoifSwiaWF0IjoxNjgxODkzNjQxLCJleHAiOjE2ODE4OTcyNDF9.-CLmxnETECn75kGy6ZHuHCyi60IRRZlS2_qcGqlUP9g

Figure 9.7: API Course 1

GET **view** Open Request →

http://localhost:8080/courses/view/1

view a single course

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoifSwiaWF0IjoxNjgxODkzNjQxLCJleHAiOjE2ODE4OTcyNDF9.-CLmxnETECn75kGy6ZHuHCyi60IRRZlS2_qcGqlUP9g

DEL **delete** Open Request →

http://localhost:8080/courses/delete/1

delete a single course

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoifSwiaWF0IjoxNjgxODkzNjQxLCJleHAiOjE2ODE4OTcyNDF9.-CLmxnETECn75kGy6ZHuHCyi60IRRZlS2_qcGqlUP9g

Figure 9.8: API Course 2

POST **update** Open Request →

http://localhost:8080/courses/update

update a single course

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA4OjQwOjMyLjE1OFoifSwiaWF0IjoxNjgxODkzNjQxLCJleHAiOjE2ODE4OTcyNDF9.-CLmxnETECn75kGy6ZHuHCyi60IRRZIS2_qcGqlUP9g

Body raw (json)

json

```
{
  "id": "1",
  "title": "test course updated",
  "description": "test description",
  "tags": "test tag"
}
```

Figure 9.9: API Course 3

Lessons

manage lessons

POST create

Open Request →

http://localhost:8080/lessons/create

create lesson

Request Headers

x-auth-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5
hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhY
mIvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1
azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmL
mdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA
0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5
OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxODk4MjY2LCJleHAiOjE2ODE5MDE4NjZ
9.qOBqwX8dM557FpiJMF7dibuOP-xKiLQbgGRqjWQioa0

Body raw (json)

```
json
```


```
{
  "courseId": 2,
  "title": "test lesson 22",
  "description": "test description"
}
```


GET list all

Open Request →

http://localhost:8080/lessons/all

list all lessons

Request Headers

x-auth-token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5
hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhY
mIvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1
azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmL
mdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA
0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5
OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxOTc4NjIzLCJleHAiOjE2ODE5ODIyMjN9.r
USSaIb8gF_WSBRHzJmVeO1mWszPcpvJlNo5EpeIYSs

Figure 9.10: API Lesson 1

DEL **delete** Open Request →

http://localhost:8080/lessons/delete/3

delete a single lesson

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxODk4MjY2LCJleHAiOjE2ODE5MDE4NjZ9.qOBqwX8dM557FpiJMF7dibuOP-xKiLQbgGRqjWQioa0

POST **update** Open Request →

http://localhost:8080/lessons/update

update a single lesson

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxODk4MjY2LCJleHAiOjE2ODE5MDE4NjZ9.qOBqwX8dM557FpiJMF7dibuOP-xKiLQbgGRqjWQioa0

Body raw (json)

json


```
{
  "id": "1",
  "title": "test lesson 1 updated",
  "description": "test description",
  "tags": "test tag"
}
```


Figure 9.11: API Lesson 3

Q&A

comments for questions / answers

POST create

Open Request →

http://localhost:8080/comments/create

create a comment

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxOTc4NjIzLCJleHAiOjE2ODE5ODIyMjN9.rUSSaIb8gF_WSBRHzJmVeO1mWszPcpvJlNo5EpeIYSs

Body raw (json)

```
json
```

```
{
  "content": "test comment lesson response",
  "courseId": null,
  "lessonId": 1,
  "previouseCommentId": 2
}
```

Figure 9.12: API comment 1

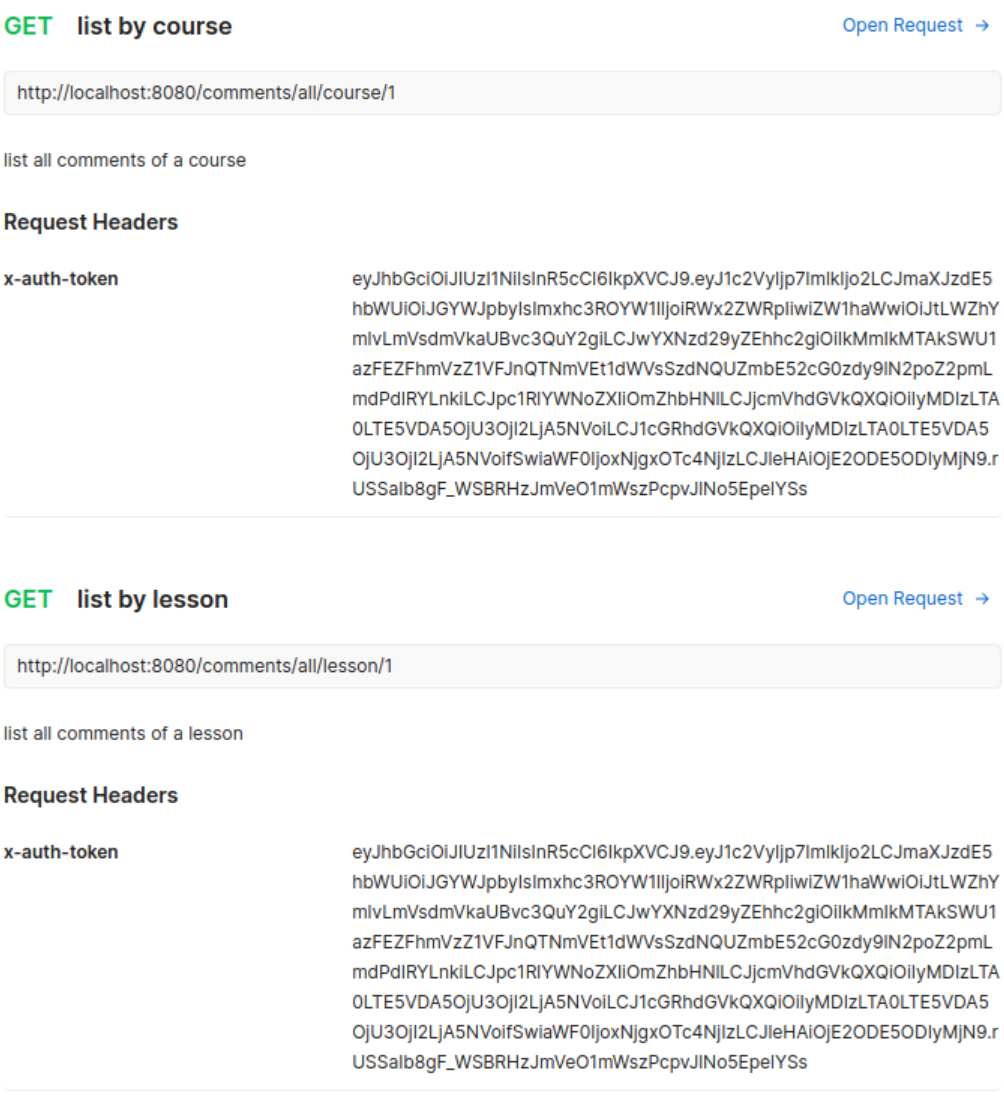

Figure 9.13: API comment 2

Files

upload files to courses / lessons

rating

rate files from 1-5

GET rate

Open Request →

```
http://localhost:8080/rating
```

rate a file

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpliwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxOTc4NjIzLCJleHAiOjE2ODE5ODIyMjN9.rUSSaIb8gF_WSBRHzJmVeO1mWszPcpvJlNo5EpeIYSs

GET show rating

Open Request →

```
http://localhost:8080/rating/get/1
```

show raitings

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpliwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxOTc4NjIzLCJleHAiOjE2ODE5ODIyMjN9.rUSSaIb8gF_WSBRHzJmVeO1mWszPcpvJlNo5EpeIYSs

Figure 9.14: API rating

POST **upload** Open Request →

http://localhost:8080/files/upload

upload a file

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoxLCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpliwiZW1haWwiOiJmYWJpby5lbHZlZGlAb3N0LmNoIiwicGFzc3dvcmRIYXNoIjoiJDJiJDEwJElINWsxRGRYZlc2dVRSZ0EzZlRLdXVlbEs3TUFGZmxOdnBtM3cvZTdqaGdqZi5nT3ZUWC55IiwiaXNUZWFjaGVyIjpmYWxzZSwiY3JlYXRlZEF0IjoiMjAyMy0wNC0xOFQyMDo1NTowNi40NjRaIiwidXBkYXRlZEF0IjoiMjAyMy0wNC0xOFQyMDo1NTowNi40NjRaIn0sImlhdCI6MTY4MTg4MDExOCwiZXhwIjoxNjgxODgzNzE4fQ.QpjeXuo8Jg603YUS-LESnY8YKq1fR1xLKs2454WqNeE

Body form-data

file	/home/fabio/Dropbox/01_School/01_OST/02_Modlus/6_semester/02_SEP2/Documents/Excercises/07-1 Clean Code.pdf
title	test file
description	test description
tags	test tag
language	1

GET **list** Open Request →

http://localhost:8080/files

list files

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoxLCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpliwiZW1haWwiOiJmYWJpby5lbHZlZGlAb3N0LmNoIiwicGFzc3dvcmRIYXNoIjoiJDJiJDEwJElINWsxRGRYZlc2dVRSZ0EzZlRLdXVlbEs3TUFGZmxOdnBtM3cvZTdqaGdqZi5nT3ZUWC55IiwiaXNUZWFjaGVyIjpmYWxzZSwiY3JlYXRlZEF0IjoiMjAyMy0wNC0xOFQyMDo1NTowNi40NjRaIiwidXBkYXRlZEF0IjoiMjAyMy0wNC0xOFQyMDo1NTowNi40NjRaIn0sImlhdCI6MTY4MTg4MDY1MCwiZXhwIjoxNjgxODg0MjUwfQ.TgNKfSQ595Ln6l6HLKipZ9Kqmy77-zqnyvsDTPRku4E

Figure 9.15: API file 1

GET **downlaod** Open Request →

http://localhost:8080/files/download/2

download file

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoxLCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJmYWJpby5lbHZlZGlAb3N0LmNoIiwicGFzc3dvcmRIYXNoIjoiJDJiJDEwJElINWsxRGRYZlc2dVRSZ0EzZlRLdXVlbEs3TUFGZmxOdnBtM3cvZTdqaGdqZi5nT3ZUWC55IiwiaXNUZWFjaGVyIjpmYWxzZSwiY3JlYXRlZEF0IjoiMjAyMy0wNC0xOFQyMDo1NTowNi40NjRaIiwidXBkYXRlZEF0IjoiMjAyMy0wNC0xOFQyMDo1NTowNi40NjRaIn0sImlhdCI6MTY4MTg4MDI1NywiZXhwIjoxNjgxODgzODU3fQ.z3uWa4dvDf2pZlcVgiLV42mcB4-xODfSg_tcrDYNrxY

DEL **delete** Open Request →

http://localhost:8080/files/delete/2

delete a file

Request Headers

x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjo2LCJmaXJzdE5hbWUiOiJGYWJpbyIsImxhc3ROYW1lIjoiRWx2ZWRpIiwiZW1haWwiOiJtLWZhYmlvLmVsdmVkaUBvc3QuY2giLCJwYXNzd29yZEhhc2giOiIkMmIkMTAkSWU1azFEZFhmVzZ1VFJnQTNmVEt1dWVsSzdNQUZmbE52cG0zdy9lN2poZ2pmLmdPdlRYLnkiLCJpc1RlYWNoZXIiOmZhbHNlLCJjcmVhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoiLCJ1cGRhdGVkQXQiOiIyMDIzLTA0LTE5VDA5OjU3OjI2LjA5NVoifSwiaWF0IjoxNjgxODk4MjY2LCJleHAiOjE2ODE5MDE4NjZ9.qOBqwX8dM557FpiJMF7dibuOP-xKiLQbgGRqjWQioa0

Figure 9.16: API file 2

9.6 End To End Test Log

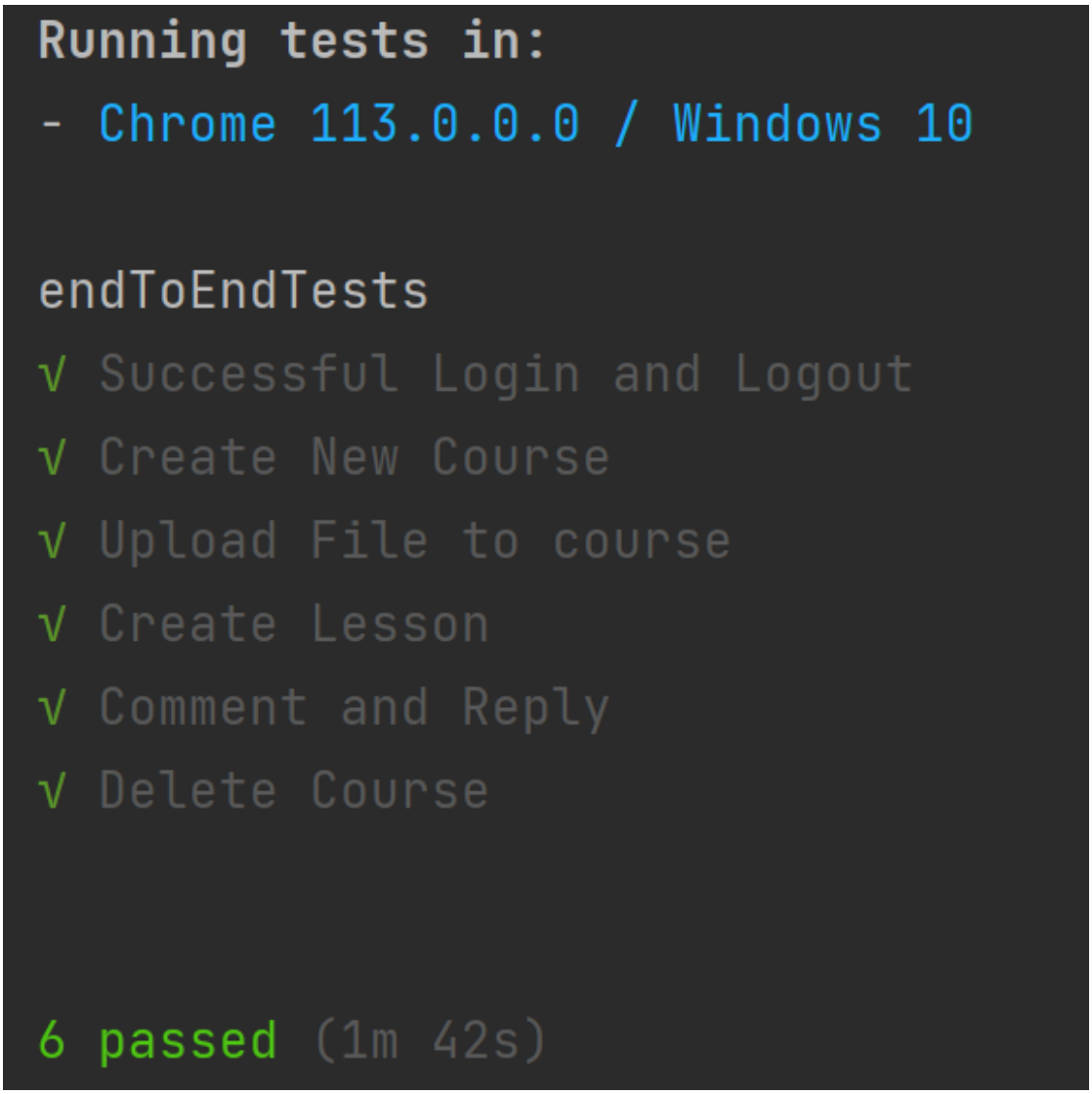

Figure 9.17: End To End Tests Log