



MEDIAVERSE

Software-Entwicklungspraktikum (SEP)

Sommersemester 2023

Angebot

Auftraggeber

Technische Universität Braunschweig

Institut für Robotik und Prozessinformatik - IRP

Mühlenpfordtstraße 23

38106 Braunschweig

Betreuer: Dr. Bertold Bongardt, Heiko Donat, Sven Tittel, Christopher Lösch

Auftragnehmer:

Name	E-Mail-Adresse
Fahd Ferjani	f.ferjani@tu-braunschweig.de
Yassine Kechiche	y.kechiche@tu-braunschweig.de
Quynh Tran	quynh.tran@tu-braunschweig.de
Michèle Eger	m.eger@tu-braunschweig.de
Subing Shen	subing.shen@tu-braunschweig.de
Oussema Ben Smida	o.ben-smida@tu-braunschweig.de
Yigit Kemal Çağlar	y.caglar@tu-braunschweig.de
Theodore Zebua	t.zebua@tu-braunschweig.de

Braunschweig, 26. April 2023

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Quynh, Michèle	...
1.1	Quynh, Michèle	...
1.2	Quynh, Michèle	...
2	Quynh, Michèle	...
3	Yassine, Fahd	...
3.1	Yassine, Fahd	...
3.2	Yassine, Fahd, Michèle	...
4	Yassine, Fahd	...
4.1	Yassine, Fahd	...
4.2	Yassine, Fahd	...
4.3	Michèle	...
5	Yigit, Oussema	...
5.1	Yigit, Oussema	...
5.2	Yigit, Oussema	...
5.3	Yigit, Oussema	...
6	Subing, Quynh	...
6.1	Subing, Quynh	...
6.2	Subing, Quynh	...
6.3	Subing, Quynh	...
7	Theodore	...

Inhaltsverzeichnis

1	Einleitung	5
1.1	Ziel	5
1.2	Motivation	6
2	Formale Grundlagen	7
3	Projektablauf	8
3.1	Meilensteine	8
3.2	Geplanter Ablauf	8
4	Projektumfang	10
4.1	Lieferumfang	10
4.2	Kostenplan	10
4.3	Funktionaler Umfang	11
5	Entwicklungsrichtlinien	12
5.1	Konfigurationsmanagement	12
5.2	Design- und Programmierrichtlinien	13
5.3	Verwendete Software	14
6	Projektorganisation	15
6.1	Schnittstelle zum Auftraggeber	15
6.2	Schnittstelle zu anderen Projekten	15
6.3	Interne Kommunikation	15
7	Glossar	16

Abbildungsverzeichnis

3.1	Gantt-Diagramm	9
5.1	Screenshot Ordnerstruktur	12

1 Einleitung

Aus unserem Projekt soll die webbasierte Software 'Mediaverse' entstehen, die Ihnen eine effiziente Nutzung sowie Verwaltung digitaler und analoger Medien ermöglichen soll. Sie werden in den folgenden Kapitel erfahren, was man sich genau unter 'Mediaverse' vorstellen kann und wie wir unsere Vision verwirklichen werden.

1.1 Ziel

Priorität und Kern unseres Projekts liegt in seiner geschickten Einfachheit als hybride Mediathek, um Ihnen eine benutzerfreundliche Anwendung zu bieten, die Sie schnell und sicher zu Ihrem Ziel führt.

Durch die einfache Erweiterbarkeit von Ressourcen und Funktionalitäten gibt es Ihnen die Möglichkeit 'Mediaverse' in unterschiedlichen Arbeitsgruppen sowie Projekten einzusetzen und zusätzlich es nach Ihren Bedürfnissen anpassen zu können.

Sie können mithilfe von 'Mediaverse' Ihre gesamten Ressourcen, seien es digitale Medien jeglicher Art, Bücher, Dokumente, technische Geräte wie Rechner, verwalten und es nach Ihren Wünschen anpassen.

Folgende Funktionen von 'Mediaverse' können Sie sich unter anderem vorstellen:

- Die Verwaltung von Rechnern und technischen Geräten bzgl. Belegung und Verfügbarkeit.
- Bücher aus der Bibliothek finden, nach Ausleihbarkeit abfragen und gegebenenfalls ausleihen.
- Wissenschaftliche Text-Dokumente finden und hinzufügen / löschen je nach Benutzergruppe.
- Sich einen Überblick über gesamte Ressourcen und Ergebnisse zu schaffen.

1.2 Motivation

Die Vielfalt der Medien wird in der heutigen Gesellschaft immer größer, früher wurden meist Bücher ausgeliehen, aber heutzutage kommen digitale Medien jeglicher Art und ganze Hardwarekomponenten dazu.

Um den Überblick bei der präsenten Informationsüberflutung zu behalten, kann Ihnen unsere hybride Mediathek 'Mediaverse' Unterstützung bieten. In unserer webbasierten Software, die im Rahmen eines Softwareentwicklungspraktikums entwickelt wird, können Sie verschiedenste Medien speichern und verwalten, ohne den Überblick zu verlieren oder lange nach etwas suchen zu müssen. Die Webseite ist von jedem Ort mit einer Internetverbindung erreichbar und somit sind digitale Medien, wie Texte und digitalisierte Bücher als Download zu jeder Zeit möglich. Dies gilt für analoge Medien hinsichtlich der Verfügbarkeit und Belegung, die man mit einer Internetverbindung einfach prüfen und zusätzlich verwalten kann.

'Mediaverse' bietet die Möglichkeit einen Ort zu erschaffen, der Wissen, Austausch und Zusammenarbeit in effizienter Art und Weise ermöglicht.

2 Formale Grundlagen

Mediaverse ist eine webbasierte Applikation, dessen Code auf zwei Subsystemen basiert. Diese sind einmal das Frontend und zum anderen das Backend. Die beiden Teile werden mithilfe einer API miteinander verbunden.

Das Frontend, welches die Benutzeroberfläche darstellt, wird mit JavaScript geschrieben und mit dem JavaScript-Framework 'Svelte' verwaltet. Für das Backend wird Python mithilfe des Frameworks 'Django' genutzt. Die dabei verwendete Entwicklungsumgebung wird Visual Studio Code sein.

Am Ende wird die Anwendung auf Englisch benutzbar sein und kann auf mobilen Endgeräten wie Tablet oder auf dem Laptop bzw. PC verwendet werden.

3 Projektablauf

Am 13. April 2023 erfolgte der Projektstart mit einem Kick-off-Meeting, das den Startschuss für das Projekt gab. Die Meilensteine und Fristen sind in der Tabelle und Abbildung 3.1 dargestellt und zeigen, dass rechtzeitige Uploads relevanter Dateien in das GitLab-Repository unerlässlich sind, um alle Verpflichtungen zu erfüllen.

3.1 Meilensteine

In diesem Unterkapitel werden alle Meilensteine mit eigenen Namen und Fristen erwähnt.

Nummer	Meilenstein	Dokumente	Abgabetermin
1	Projektstart	-	13.04.23
2	Angebot	Angebot	26.04.23
3	Pflichtenheft	Pflichtenheft	17.05.23
4	Abnahmetestspezifikation	Abnahmetestspezifikation	17.05.23
5	Zwischenpräsentation	-	26.05.23
6	Fachentwurf	Fachentwurf	07.06.23
7	Technischer Entwurf	Technischer Entwurf	28.06.23
8	Testdokumentation	Testdokumentation	12.07.23
9	TDSE	-	20.07.23

3.2 Geplanter Ablauf

Dieses Gantt-Diagramm stellt die Meilensteine dar, die im vorherigen Unterkapitel erwähnt wurden.



Abbildung 3.1: Gantt-Diagramm

4 Projektumfang

Dieses Kapitel dient dazu, den Projektumfang zu definieren. Dazu gehört eine Übersicht über Lieferumfang, Kostenplan und Funktionsumfang der Software.

4.1 Lieferumfang

Der Lieferumfang unseres Projekts umfasst eine webbasierte Anwendung, die als hybride Mediathek fungiert und grundlegende Funktionen wie die Suche und Ausleihe von Ressourcen, den Download digitaler Kopien und das Verwalten von Favoritenlisten ermöglicht. Wir stellen zudem den vollständigen Quellcode der Anwendung sowie eine umfangreiche Dokumentation zur Verfügung, die aus verschiedenen Teilen besteht. Das Angebot enthält eine Beschreibung des Projekts und der angebotenen Leistungen, während der technische Entwurf die technische Umsetzung des Projekts beschreibt. Der fachliche Entwurf umfasst die funktionalen Anforderungen an die Anwendung. Die Abnahmespezifikation definiert, welche Kriterien das System erfüllen muss, um abgenommen zu werden, und die Testdokumentation beschreibt die durchgeführten Tests und deren Ergebnisse. Das Pflichtenheft enthält die Anforderungen an die Anwendung und ihre Funktionen. Die technischen Details der Implementierung umfassen Informationen über die verwendeten Technologien und deren Integration. Die Systemanforderungen beschreiben, welche Hardware- und Software-Komponenten für den Betrieb des Systems erforderlich sind. Die Benutzerhandbücher erläutern die Benutzung der Anwendung für Endnutzer und Administratoren.

4.2 Kostenplan

Die Anzahl der Personen, die für die Entwicklung dieses Projekts verantwortlich sind, beträgt 8. Jedes Mitglied wird ungefähr 200 Stunden arbeiten, wobei der Arbeitslohn für einen Entwickler bei 100 Euro pro Stunde liegt. Die Gesamtkosten belaufen sich somit auf 160.000€

4.3 Funktionaler Umfang

Es wird drei Benutzergruppen geben, den Benutzer, den Mediathekar und den Administrator. Benutzer müssen sich vorerst registrieren und können sich erst dann mit dem Benutzernamen und Passwort anmelden. Sie können Abfragen für Bücher vornehmen, indem sie nach dem Titel, Autor und/oder Jahr suchen. Auch eine Abfrage zu Stichworten und Kategorien ist möglich, wenn kein bestimmtes Buch oder Dokument gesucht wird, sondern Dokumente zu einem bestimmten Thema. Die Bücher und anderen Medien können sie ausleihen und müssen diese nach einer bestimmten Zeit zurückgeben. Mit einem QR-Code oder einer anderen Methode, wie DENT werden die physikalischen Ressourcen eingelesen, was den Vorgang der Ausleihe vereinfacht. Digitale Medien, wie zum Beispiel E-Books können als eine Kopie downgeloadet werden. Der Benutzer hat auch die Möglichkeit den Bibtex-Eintrag eines Buchs der Bibliothek herunterzuladen. Jeder Benutzer kann eine Favoritenliste erstellen und beliebig viele Bücher und andere Medien dort abspeichern.

Der Mediathekar verwaltet die Ressourcen, indem man neue Bücher, Dokumente und andere Medien, sowie auch Hardware anlegen, editieren und löschen.

Administratoren verwalten Benutzer und Mediathekare, sie können diese neu anlegen und neue Rechte vergeben, diese wieder ändern und auch wieder löschen. Sie generieren Berichte aller Aktionen, machen Versionskontrollen der textbasierten Daten und passen das Datenschema an. Außerdem können sie neue Ressourcentypen erstellen.

5 Entwicklungsrichtlinien

Im Folgenden werden Details zu Richtlinien für die Entwicklung wie Software und Frameworks, die verwendet werden, Regeln für das Hochladen eines Dokuments in das Repository und anderes näher erläutert.

5.1 Konfigurationsmanagement

Das Team wird ein zentrales Repository für das Projekt verwenden, das die folgende Ordnerstruktur aufweist:

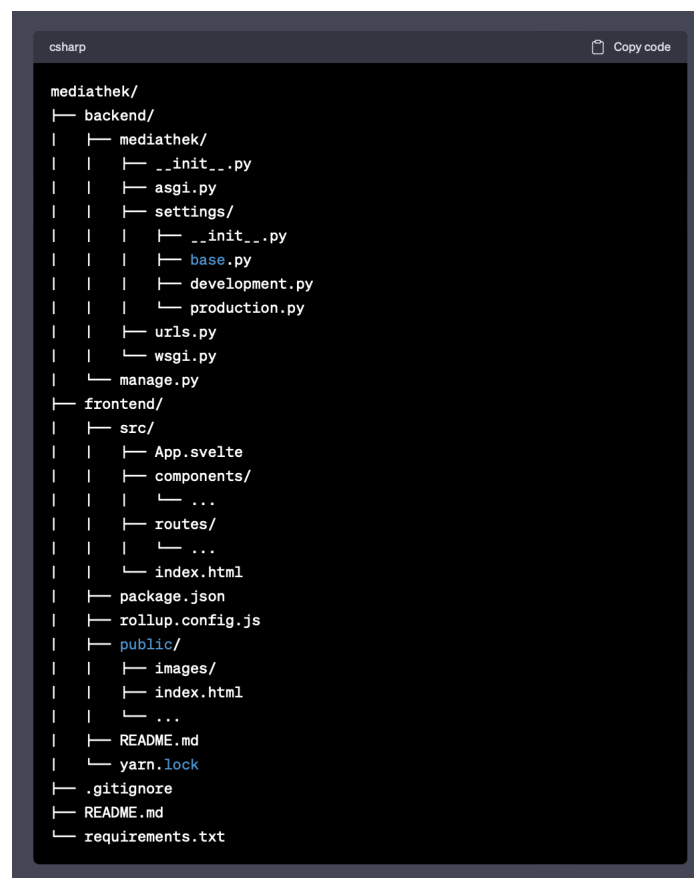


Abbildung 5.1: Screenshot Ordnerstruktur

In dieser Beispiel-Ordnerstruktur haben wir den Backend- und Frontend-Code in getrennte Verzeichnisse aufgeteilt, um den Code übersichtlich und einfach zu verwalten.

Das Verzeichnis `backend/` enthält das Django-Projekt und die Anwendungsdateien, wobei sich das Hauptprojekt im Unterverzeichnis `mediathek/` befindet. Das Unterverzeichnis `settings/` enthält die Einstellungsdateien für die verschiedenen Umgebungen, z. B. Entwicklung und Produktion.

Das Verzeichnis `frontend/` enthält den Svelte-Frontend-Code, wobei sich die Hauptkomponente von Svelte in `src/App.svelte` befindet. Das Verzeichnis `public/` enthält die statischen Assets für das Frontend, wie z.B. Bilder und die Hauptdatei `index.html`.

Das Wurzelverzeichnis enthält die projektweiten Dateien, wie z. B. `.gitignore`, `README.md` und `requirements.txt`.

Diese Struktur wird dazu beitragen, die Projektdateien zu organisieren und sie leicht zugänglich zu machen.

Beim Hinzufügen von Dateien zum Repository gelten ebenfalls bestimmte Regeln. Erstens werden sinnvolle Dateinamen verwendet, um das Auffinden und Identifizieren von Dateien zu erleichtern. Zweitens sollen Commits früh und oft durchgeführt werden, um sicherzustellen, dass Änderungen regelmäßig gesichert werden und die Zusammenarbeit erleichtert wird. Drittens sollen keine großen Binärdateien übertragen werden, die das Repository verlangsamen könnten. Viertens sollen Abhängigkeiten sorgfältig aktualisiert werden, um Konflikte zu vermeiden und Kompatibilität zu gewährleisten. Zuletzt soll kein fehlerhafter Code in das Repository übertragen werden, um Fehler und Probleme für andere Teammitglieder zu vermeiden.

5.2 Design- und Programmierrichtlinien

Für das Frontend und das Backend des Projekts hat sich das Team für spezifische Style Guides entschieden. Für das Frontend wird der Airbnb JavaScript Style Guide verwendet, der eine umfassende Reihe von Richtlinien für das Schreiben von sauberem und wartbarem JavaScript-Code enthält. Dieser Style Guide deckt ein breites Spektrum an Themen ab, darunter Namenskonventionen, Variablendeklarationen, Abstände und Formatierungen. Durch die Verwendung dieses Styleguides kann das Team sicherstellen, dass der Code für das Frontend konsistent, leicht zu lesen und einfach zu warten ist.

Für das Backend wird das Team den Python Style Guide (PEP 8) verwenden, den offiziellen Style Guide für Python-Code. Dieser Style Guide deckt ein breites Spektrum an Themen ab, darunter Namenskonventionen, Funktions- und Variablendeklarationen, Leerzeichen und Dokumentation.

Durch die Einhaltung von PEP 8 kann das Team sicherstellen, dass sein Backend-Code konsistent und leicht lesbar ist und den Best Practices der Python-Community entspricht.

Durch die Verwendung dieser Styleguides kann das Team sicherstellen, dass der Code leicht zu verstehen und zu pflegen ist und dass er den Best Practices sowohl im Frontend- als auch im Backend-Bereich entspricht.

5.3 Verwendete Software

Für die Entwicklung des Projekts hat sich das Team nicht für eine bestimmte integrierte Entwicklungsumgebung (IDE) entschieden. Allerdings wurde festgelegt, dass Visual Studio Code die gewünschte IDE sein wird. Das Team ist der Ansicht, dass Visual Studio Code eine benutzerfreundliche Oberfläche und eine breite Palette von Funktionen bietet, die für das Projekt von Vorteil sein werden.

Für das Backend setzt das Team auf das Django-Framework, ein High-Level-Python-Webframework, das eine schnelle Entwicklung und ein sauberes, pragmatisches Design fördert. Für das Frontend wurde das Framework Svelte gewählt, ein modernes JavaScript-Framework, das sich auf Leistung und Benutzerfreundlichkeit konzentriert.

Es ist auch wichtig, dass es bei der Entwicklung von Webanwendungen verschiedene Arten von Tests durchgeführt werden, um sicherzustellen, dass die Anwendung stabil und zuverlässig ist. Dazu sollten wir sowohl das Frontend als auch das Backend separat testen und einzelne Funktionen getrennt voneinander testen, um sicherzustellen, dass sie ordnungsgemäß funktionieren.

Des Weiteren sollen verschiedene Datensätze bei unseren Tests verwendet werden, um zu gewährleisten, dass die Anwendung unter verschiedenen Bedingungen und mit unterschiedlichen Eingaben zuverlässig funktioniert.

Für die Erstellung von Diagrammen schlägt das Team die Verwendung von Software wie draw.io, dia und lucidchart vor. Diese Tools sind beliebt und einfach zu bedienen. Durch den Einsatz dieser Tools hofft das Team, klare und umfassende Diagramme zu erstellen, die den Entwicklungsprozess unterstützen.

6 Projektorganisation

In diesem Kapitel wird besprochen, welche Schnittstellen es geben wird mit dem Auftraggeber sowie zu anderen Projekten. Des Weiteren wird erklärt, wie das Team intern miteinander kommuniziert.

6.1 Schnittstelle zum Auftraggeber

Regelmäßige Treffen finden wöchentlich am Donnerstag statt, um den Entwicklungsfortschritt, bestehende Probleme sowie weitere Pläne auszutauschen.

TU-Mail und Matrix-Chat werden auch für die asynchrone Kommunikation benutzt.

6.2 Schnittstelle zu anderen Projekten

Zurzeit ist keine Zusammenarbeit mit anderen Teams geplant.

Mögliche Kooperationsabsichten werden per TU-Mail ausgetauscht. Sobald die Zusammenarbeit bestätigt ist, wird das Projekt auf GitLab erstellt sowie in der Chatgruppe auf Matrix-Chat mitgeteilt.

6.3 Interne Kommunikation

Direkt nach dem Treffen mit dem Auftraggeber (Betreuer), wird sich das Team für kurze Zeit versammeln, falls etwas diskutiert werden muss.

Es gibt außerdem ein wöchentliches Online-Treffen über Big Blue Button für die interne Kommunikation. Wie das Treffen mit Auftraggeber (Betreuer), werden der Entwicklungsfortschritt, bestehende Probleme und weitere Pläne besprochen.

Feedback-Sessions werden flexibel nach der Diskussion geplant, bevor etwas, beispielsweise dieses Angebotsheft, abgegeben wird.

7 Glossar

Analoge Medien:

Medien in gedruckter form, z.B.: Bücher, Zeitungen, Zeitschriften, Plakate etc.

API:

Bedeutet „Application Programming Interface“ und beschreibt eine Schnittstelle, die verschiedene Anwendungen miteinander verbindet zum Datenaustausch bzw. zur Kommunikation.

Binärdatei:

Computerdatei, die keine Textdatei ist, die also nicht nur Repräsentationen „druckbarer“ Zeichen enthält, sondern die beliebige Bitmuster enthalten kann.

Bibtex-Eintrag:

Ein BibTeX ist ein Programm zur Erstellung von Literaturangaben und -verzeichnissen in TeX- oder LaTeX-Dokumenten.

Digitale Medien:

Elektronische Medien, die mit digitalen Codes arbeiten.

Django:

Ein kostenloses und open-source, Python-basiertes High-Level-Webframework zum Erstellen von Webanwendungen, das dem Modell-Vorlage-Ansichten-Architekturmuster folgt.

Hybride Mediathek:

Strategische Nutzung traditioneller und neuer Medien, um Botschaften effektiver zu kommunizieren.

Ident-Code:

Ein Code, den ein Informationssystem erkennt, um das Subjekt zu identifizieren. Eine Benutzer-ID bspw. ist ein typischer Identifikationscode.

JavaScript:

Eine high-level, dynamische, objektorientierte Programmiersprache, die neben HTML und CSS eine der Kerntechnologien zum Erstellen interaktiver Webseiten ist.

Python:

Eine high-level, allgemeine Programmiersprache, die für eine Vielzahl von Anwendungen verwendet wird, darunter Webentwicklung, Datenanalyse, künstliche Intelligenz, wissenschaftliches Rechnen, usw.

QR-Code:

Kurz für „Quick Response“-Code, ist eine Art zweidimensionaler Barcode, der aus schwarzen und weißen Quadraten besteht, die in einem quadratischen Raster auf weißem Hintergrund angeordnet sind, und kann eine Vielzahl von Informationen wie Website-URLs, Kontaktinformationen, Produktinformationen, usw. speichern.

Svelte:

Eine JavaScript-Softwarebibliothek, die ein Framework zum Anzeigen von Benutzeroberflächenkomponenten von Websites bereitstellt.

Visual Studio Code:

Ein Code-Editor, der von Microsoft entwickelt wurde und für die Arbeit mit mehreren Programmiersprachen entwickelt wurde und eine Reihe von Funktionen wie Debugging, Code-Refactoring, Versionskontrolle und mehr bietet.

Webbasierte Applikation:

Ein Anwendungsprogramm, das auf einem entfernten Server gespeichert ist und über das Internet über eine Browserschnittstelle bereitgestellt wird.