HTL Leonding
next level

HÖHERE TECHNISCHE
BUNDESLEHRANSTALT
LEONDING

Unser Thema ist Mediahub

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für IT-Medientechnik

Eingereicht von:
Markus Tran
Abdulkerim Cufurovic

Betreuer:
Robert Reder

Leonding, 4. April 2025

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, 4. April 2025 Markus Tran & Abdulkerim Cufurovic

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation und Problemstellung der Abo-Verwaltung

Text ...

1.2 Zielsetzung der Diplomarbeit

Text ...

1.3 Nutzen der Anwendung für Endanwender

Text ...

1.4 Aufbau der Diplomarbeit

Text ...

2 Ausgangssituation und Analyse

2.1 Aktuelle Herausforderungen bei Streaming-Abonnements

Text ...

2.2 Probleme durch parallele Abos und doppelte Abbuchungen

Text ...

2.3 Bestehende Lösungen und deren Schwächen

Text ...

2.4 Abgrenzung zu Standard-Webprojekten

Text ...

3 Fachlicher Hintergrund Abo- & Mediatheken-Management

3.1 Zentrale Verwaltung von Abonnements und Laufzeiten

Text ...

3.2 Darstellung von Verfügbarkeiten von Filmen und Serien

Text ...

3.3 Mehrwert einer zentralen Applikation für Nutzer

Text ...

3.4 Datenschutz- und Sicherheitsanforderungen

Text ...

4 Technologische Grundlagen

4.1 Angular

Angular wurde als zentrales Frontend-Framework für die Umsetzung der Webanwendung eingesetzt. [1] Ziel war die Entwicklung einer strukturierten, wartbaren und erweiterbaren Benutzeroberfläche, die eine zentrale Verwaltung von Mediatheken-Abonnements ermöglicht. Durch den komponentenbasierten Aufbau eignet sich Angular besonders für komplexe Anwendungen mit klar getrennten Funktionsbereichen [2].

4.1.1 Rolle von Angular innerhalb der Anwendung

Angular übernimmt die vollständige Darstellung und Interaktion der Benutzeroberfläche. Sämtliche Funktionen zur Verwaltung von Abonnements, zur Anzeige von Laufzeiten sowie zur Darstellung der Verfügbarkeit von Filmen und Serien werden im Frontend realisiert. Die Trennung von Frontend und Backend ermöglicht eine klare Verantwortungsaufteilung zwischen Präsentations- und Geschäftslogik.

4.1.2 Architektur als Single-Page-Application

Die Anwendung wurde als Single-Page-Application (SPA) umgesetzt, bei der Inhalte nach dem initialen Laden dynamisch aktualisiert werden, ohne dass ein vollständiger Seitenreload erforderlich ist [3]. Dieses Konzept verbessert die Benutzererfahrung erheblich, da Navigationsvorgänge zwischen verschiedenen Bereichen verzögerungsfrei erfolgen.

Für eine Abo-Verwaltungsplattform ist dieses Verhalten besonders relevant, da häufig zwischen unterschiedlichen Ansichten gewechselt wird, um Laufzeiten zu vergleichen oder Inhalte zu prüfen.

4.1.3 Begründung der Wahl von Angular gegenüber React

Die Entscheidung für Angular fiel aufgrund des ganzheitlichen Framework-Ansatzes. Angular stellt im Gegensatz zu React ein vollständiges Framework bereit, das unter anderem Routing, Formularverarbeitung und HTTP-Kommunikation standartmäßig integriert [4]. Dies ermöglicht eine einheitliche Projektstruktur und erleichtert die langfristige Wartung der Anwendung.

4.1.4 Komponentenstruktur der Abo-Verwaltung

Die Anwendung ist in logisch getrennte Komponenten unterteilt. Zentrale Komponenten sind unter anderem:

- Übersicht der aktiven Abonnements
- Detailansicht einzelner Abos mit Laufzeitinformationen
- Ansicht zur Anzeige von Filmen und deren Streaming-Anbietern
- Benutzer- und Kontoverwaltung

Diese Struktur unterstützt eine klare Trennung der Verantwortlichkeiten und erleichtert zukünftige Erweiterungen der Anwendung.

4.1.5 Data Binding und Services zur Zustandsverwaltung

Angular stellt mit Data Binding und Services bewährte Mechanismen zur Verwaltung des Anwendungszustands bereit [2]. Benutzereingaben werden direkt mit dem Datenmodell verknüpft, wodurch Änderungen unmittelbar in der Benutzeroberfläche sichtbar werden.

Die Kommunikation mit dem Backend erfolgt über dedizierte Services, die sämtliche HTTP-Anfragen kapseln und eine konsistente Datenbasis sicherstellen.

4.2 Quarkus

4.2.1 Grundlagen des Frameworks

4.2.2 Vorteile im Kontext der Abo-Verwaltung

4.2.3 Aufbau der REST-Schnittstellen

4.2.4 Kommunikation mit externen APIs

4.3 Keycloak

4.3.1 Grundlagen der Authentifizierung und Autorisierung

4.3.2 Ablauf des Login- und Token-Prozesses

4.3.3 Rollen- und Benutzerverwaltung

4.3.4 Sicherheitsaspekte und Datenschutz

4.4 PostgreSQL

4.4.1 Datenmodell für Abonnements und Mediatheken

4.4.2 Strukturierung der relationalen Tabellen

4.4.3 Vorteile relationaler Datenbanken für dieses Projekt

4.4.4 Evaluierung möglicher Alternativen

4.5 Docker

4.5.1 Grundlagen der Containerisierung

4.5.2 Einsatz von Docker im Entwicklungsprozess

4.5.3 Vereinheitlichung der Entwicklungsumgebung

5 Architektur und Systemdesign

5.1 Gesamtarchitektur der Anwendung

Text ...

5.2 Kommunikation zwischen Frontend und Backend

Text ...

5.3 API-Integration externer Movie-Datenbanken

Text ...

5.4 Sicherheits- und Datenschutzkonzept

Text ...

6 Umsetzung der Anwendung

6.1 Implementierung der Abo-Verwaltung

Text ...

6.2 Darstellung von Laufzeiten und Kündigungszeitpunkten

Text ...

6.3 Verknüpfung von Filmen mit Streaming-Anbieter

Text ...

6.4 Account- und Passwortverwaltung

Text ...

6.4.1 Analyse bestehender Lösungen (Bitwarden, 1Password)

Text ...

6.4.2 Sicherheits- und Datenschutzbetrachtung

Text ...

6.4.3 Entscheidung: Eigenlösung vs. Integration

Text ...

7 Projektverlauf und Entscheidungen

7.1 Projektplanung und Arbeitsweise

7.2 Technische und fachliche Herausforderungen

7.3 Entscheidungsfindung im Projektteam

7.4 Einbindung von Betreuern und Feedback

7.5 Reaktion auf Fragen und Kritik aus Präsentationen

8 Ergebnisse und Evaluation

8.1 Erfüllung der gesetzten Ziele

8.2 Bewertung des Nutzens für Endanwender

8.3 Reflexion der eigenen Leistung

9 Ausblick

9.1 Mögliche Erweiterungen der Anwendung

9.2 Zukunftspotenzial der Plattform

10 Zusammenfassung

11 Quellen-, Literatur- und Abbildungsverzeichnis

11.1 Quellen

11.2 Literatur

11.3 Abbildungsverzeichnis

Literaturverzeichnis

[1] Angular Team, „Angular Documentation," https://angular.io/docs, 2024, Zugriff am 06.01.2026.

[2] ——, „Angular Architecture Overview," https://angular.io/guide/architecture, 2024, Zugriff am 06.01.2026.

[3] Mozilla Developer Network, „Single-page applications," https://developer.mozilla.org/en-US/docs/Glossary/SPA, 2024, Zugriff am 06.01.2026.

[4] A. Freeman, *Pro Angular*. Apress, 2019.

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis