

M294 & M295 Projektdokumentation

Sasha Fellmann

Film Watchlist

Inhalt

Projektidee	2
Anforderungsanalyse	2
Klassendiagramm	3
Storyboard und Screen Mockups	4
REST-Schnittstellen	6
Testplan	7
Durchführung der Tests	7
Installationsanleitung	9
Hilfestellungen	10

Projektidee

Meine Anwendung ist eine Film-, beziehungsweise Serien-bibliothek, mit der man seine eigene Watchlist kurieren kann.

Die Anwendung ermöglicht die Erfassung von verschiedenen Filmen, Serien, Episoden und deren Details. Darunter den Namen, Erscheinungsdatum, Genre und Regisseur für Filme/Serien. Diese Daten werden auf dem Home Screen in einer Tabelle übersichtlich dargestellt. Durch die einzelnen Filme, kommt man dann zu den dazugehörigen Episoden, welche Namen und Releasedatum zeigen.

Filme und deren Episoden können alle auch einzeln eingesehen werden. Fertig geschaute Filme können von der Liste mitsamt dazugehörigen Episoden entfernt werden.

Im Frontend nutze ich React. Um die Anfragen ins backend zu schicken verwende ich Axios. Das Backend habe ich mit dem Spring-Boot-Framework kreiert. Das ganze habe ich in VSCode (Frontend) und in IntelliJ(Backend) benutzt um meine Anwendung zu erstellen. Die MySQLWorkbench habe ich benutzt um die Tabellen einzusehen und zuletzt Postman um Anfragen zu testen, bevor das Frontend bestand.

Anforderungsanalyse

Einsehen von Filmen

- Alle sich auf der Datenbank befindenden Filme werden auf der dazugehörigen Seite angezeigt.

Hinzufügen von Filmen

- Der User kann Filme hinzufügen.
- Der User muss den Filmenamen, das Erscheinungsdatum, das Genre und den Regisseur angeben.
- Die Filme werden auf der Datenbank gespeichert und auf der dazugehörigen Seite angezeigt.

Bearbeiten von Filmen

- Der User kann die hinzugefügten Filme nach Belieben bearbeiten.
- Filmname, Erscheinungsdatum, Genre und Regisseur müssen immer noch gegeben sein.
- Die Veränderungen werden auf der Datenbank gespeichert und auf der dazugehörigen Seite angezeigt.

Löschen von Filmen

- Der User kann die hinzugefügten Filme Löschen (bzw. von der Watchliste entfernen).
- Die gelöschten Filme werden von der Datenbank und der dazugehörigen Seite entfernt.

Einsehen von Episoden

- Alle zu einem Film gehörenden Episoden können eingesehen werden.

Hinzufügen von Episoden

- Der User kann Episoden hinzufügen.
- Der User muss den Episodennamen und das Releasedatum angeben.
- Die Filme werden auf der Datenbank gespeichert und auf der dazugehörigen Seite angezeigt.

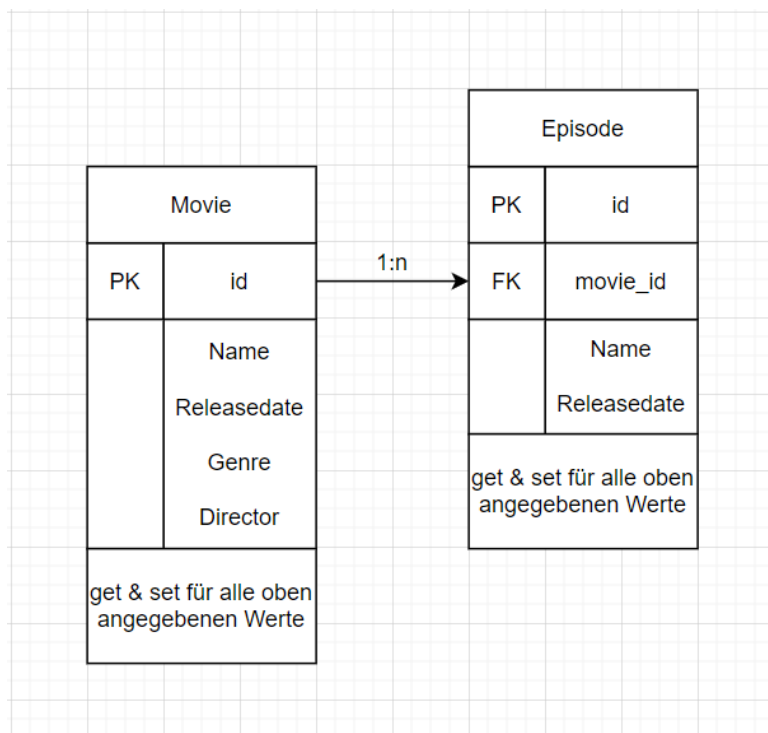
Bearbeiten von Episoden

- Der User kann die hinzugefügten Episoden nach Belieben bearbeiten.
- Episodename und Releasedatum müssen immer noch gegeben sein.
- Die Veränderungen werden auf der Datenbank gespeichert und auf der dazugehörigen Seite angezeigt.

Löschen von Episoden

- Der User kann die hinzugefügten Episoden Löschen (bzw. von der Watchliste entfernen).
- Die gelöschten Episoden werden von der Datenbank und der dazugehörigen Seite entfernt.

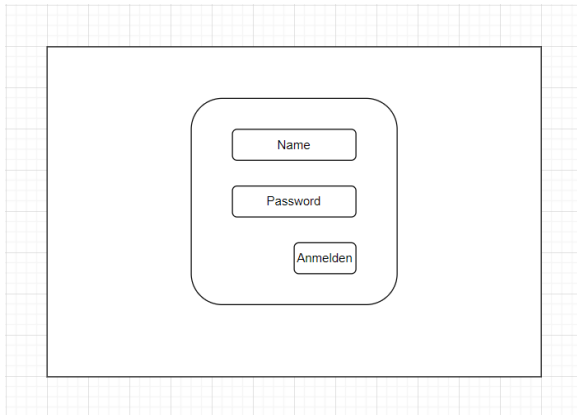
Klassendiagramm



Beide der oben gezeigten Klassen haben einen Controller (MovieController und EpisodeController, der dazu da ist zu bestimmen was mit den Filmen und Episoden passieren soll (z.B. ob ein neuer Film hinzugefügt wird, oder ein alter gelöscht wird). Beide haben auch Repositories (MovieRepository und EpisodeRepository) die wissen wo alle Daten zu finden sind und die Befehle der Controller ausführen. Zuletzt gibt es noch die beiden Klassen selbst (Movie und Episode) wo die Infos zu den Filmen und deren Episoden bestimmt werden.

Storyboard und Screen Mockups

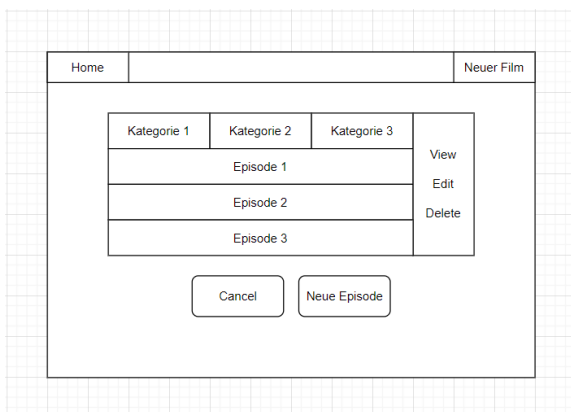
Login



A login screen mockup on a grid background. It features a central rounded rectangle containing three input fields: 'Name', 'Password', and an 'Anmelden' button.

Die erste Seite, die der User sieht ist die Login Seite, wo er seinen Benutzernamen und sein Passwort eingeben kann. Nach der Anmeldung wird der User zu der Home Seite weitergeleitet.

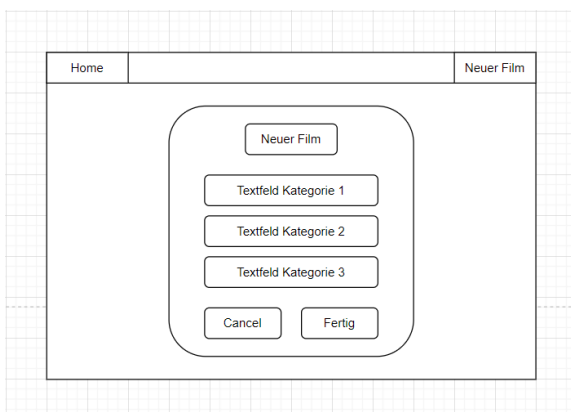
Filme



A home screen mockup on a grid background. It has a navigation bar at the top with 'Home' and 'Neuer Film'. Below it is a table with columns 'Kategorie 1', 'Kategorie 2', 'Kategorie 3', and a vertical column for actions. The table contains three rows for 'Episode 1', 'Episode 2', and 'Episode 3'. Below the table are 'Cancel' and 'Neue Episode' buttons.

Kategorie 1	Kategorie 2	Kategorie 3	
Episode 1			View
Episode 2			Edit
Episode 3			Delete

Auf dem Home screen sieht man alle vom User angelegten Filme in einer Tabelle aufgelistet. Auf der rechten Seite der Tabelle sind die jeweiligen View, Edit und Delete Buttons, mit denen man die Filme bearbeiten kann. Daneben gibt es noch einen Button, mit dem man zu den zum Film dazugehörigen Episoden kommt. In der Navigationsleiste am oberen Bildschirm Rand findet man den Add Movie Knopf, mit dem man auf die dazugehörige Seite weitergeleitet wird.



An 'Add Movie' screen mockup on a grid background. It has a navigation bar at the top with 'Home' and 'Neuer Film'. Below it is a form with a 'Neuer Film' button, three text fields labeled 'Textfeld Kategorie 1', 'Textfeld Kategorie 2', and 'Textfeld Kategorie 3', and 'Cancel' and 'Fertig' buttons at the bottom.

Auf der Add Movie Seite gibt der User alle Daten eines Films, den er hinzufügen möchte ein und drückt danach auf Fertig. Somit wird der Film in die Datenbank eingefügt und taucht nachher auf der Home Seite auf. Mit dem Home oder dem Cancel Button kann der User zum Home Screen zurückkehren. Der Fertig Button leitet einem nach erfolgreicher Prüfung der Daten auch auf den Homescreen zurück.

Wenn der User auf dem Home Screen auf den Edit Button neben einem bestimmten Filmes klickt, kommt er auf die zum Film gehörende Edit Seite. Hier kann der User alle vorhin getätigten Eingaben überarbeiten. Alle eingaben werden in der Datenbank und im Home Screen wiedergegeben.

Wenn der User auf dem Home Screen auf den View Button neben einem bestimmten Filmes klickt, kommt er auf die zum Film gehörende Seite. Hier kann der User alle zu einem spezifischen Film gehörenden Daten (Auch die id) ansehen.

Episoden

Kategorie 1	Kategorie 2	Kategorie 3	
Film 1			View Edit Delete
Film 2			
Film 3			
Film 4			
Film 5			

Wenn der User auf dem Home Screen auf den Episodes Button klickt, kommt er auf die zum Film gehörenden Episoden. Auf der rechten Seite der Tabelle sind die jeweiligen View, Edit und Delete Buttons, mit denen man die Episoden bearbeiten kann.

Auf der Add Episode Seite gibt der User alle Daten einer Episode, die er hinzufügen möchte ein und drückt danach auf Fertig. Somit wird die Episode in die Datenbank eingefügt und taucht nachher auf der Episodes Seite auf. Mit dem Home oder dem Cancel Button kann der User zum Home Screen zurückkehren. Der Fertig Button leitet einem nach erfolgreicher Prüfung der Daten auch auf den Homescreen zurück.

Wenn der User im Episode Screen auf den Edit Button neben einer bestimmten Episode klickt, kommt er auf die zu der Episode gehörende Edit Seite. Hier kann der User alle vorhin getätigten Eingaben überarbeiten. Alle eingaben werden in der Datenbank und im Episode Screen wiedergegeben.

Wenn der User auf dem Episode Screen auf den View Button neben einer bestimmten Episode klickt, kommt er auf die zu der Episode gehörende Seite. Hier kann der User alle zu einer spezifischen Episode gehörenden Daten (Auch die id) ansehen.

REST-Schnittstellen

Filme

- Abrufen aller Filme (GET /movies)
- Abrufen eines bestimmten Films mit der Id (GET /movie/{id})
- Hinzufügen eines neuen Films (POST /movie)
- Aktualisieren eines vorhandenen Films (PUT /movie/{movieid})
- Löschen eines Films (DELETE /movie/{movieid})

Episoden

- Abrufen aller Episoden (GET /episodes)
- Abrufen einer bestimmten Episode mit der Id (GET /episode/{id})
- Hinzufügen einer neuen Episode (POST /episode)
- Aktualisieren einer vorhandenen Episoden (PUT /episode/{episodeid})
- Löschen einer Episode (DELETE /episode/{episodeid})

Testplan

Filme

1. Wird ein neu erstellter Film richtig in die Datenbank aufgenommen?
2. Werden Veränderungen an den Details (Über den Edit Screen) in der Datenbank?
3. Wird ein gelöschter Film von der Datenbank entfernt?

Episoden

1. Wird eine neu erstellte Episode richtig in die Datenbank aufgenommen?
2. Werden Veränderungen an den Details (Über den Edit Screen) in der Datenbank?
3. Wird eine gelöschte Episode von der Datenbank entfernt?

Beides

1. Werden Episoden gelöscht, wenn der dazugehörige Film gelöscht wird?

Durchführung der Tests

Filme

Testfall 1	Hinzufügen eines Films
Testziel	Überprüfen, ob Filme erfolgreich in die Datenbank aufgenommen wird.
Schritte	<ol style="list-style-type: none">1. Im Home Screen auf den Add Movie Button in der Navbar Klicken.2. Alle verlangten Daten Korrekt eingeben.3. Auf Submit Button Klicken um den Film hinzuzufügen.4. Über MySQLWorkbench kontrollieren ob alle Daten korrekt in die movie Tabelle übernommen wurden.
Erwartetes Ergebnis	Alle Daten werden in die Tabelle movie in der Fullstack Datenbank aufgenommen.
Ergebnis: Erfolgreich	Alle Daten wurden in die Tabelle aufgenommen.

Testfall 2	Bearbeiten eines Films
Testziel	Überprüfen, ob Änderungen an Filmen erfolgreich in die Datenbank aufgenommen wird.
Schritte	<ol style="list-style-type: none">1. Im Home Screen auf den Edit Button eines bestimmten Films Klicken.

	<ol style="list-style-type: none"> 2. Zwei der Felder bearbeiten (Releasedate und Director). 3. Auf Submit Button Klicken um die Änderungen zu speichern. 4. Über MySQLWorkbench kontrollieren ob alle Daten korrekt in die movie Tabelle übernommen wurden.
Erwartetes Ergebnis	Alle Änderungen werden in die Tabelle movie in der Fullstack Datenbank aufgenommen.
Ergebnis: Erfolgreich	Alle Änderungen wurden in die Tabelle aufgenommen.

Testfall 3	Löschen eines Films
Testziel	Überprüfen, ob gelöschte Filme aus der Datenbank entfernt werden.
Schritte	<ol style="list-style-type: none"> 1. Im Home Screen auf den Delete Button eines bestimmten Films Klicken. 2. Über MySQLWorkbench kontrollieren ob der Film aus der Tabelle movie entfernt wurde.
Erwartetes Ergebnis	Der gewählte Film wird aus der Tabelle movie entfernt.
Ergebnis: Erfolgreich	Der gewählte Film wurde entfernt.

Episoden

Testfall 1	Bearbeiten einer Episode
Testziel	Überprüfen, ob Episoden erfolgreich in die Datenbank aufgenommen wird.
Schritte	<ol style="list-style-type: none"> 1. Im home Screen bei einem bestimmten Film auf den Episodes Button klicken. 2. Im episodes Screen auf den Add episode Button klicken. 3. Alle verlangten Daten korrekt eingeben. 4. Auf Submit Button klicken, um die Episode hinzuzufügen. 5. Über MySQLWorkbench kontrollieren ob alle Daten korrekt in die episode Tabelle übernommen wurden.
Erwartetes Ergebnis	Alle Daten werden in die Tabelle episode in der Fullstack Datenbank aufgenommen.
Ergebnis: Erfolgreich	Alle Daten wurden in die Tabelle aufgenommen.

Testfall 2	Bearbeiten einer Episode
Testziel	Überprüfen, ob Änderungen an Episoden erfolgreich in die Datenbank aufgenommen wird.
Schritte	<ol style="list-style-type: none"> 1. Im episodes Screen auf den Edit Button einer bestimmten Episode Klicken. 2. Eines der Felder bearbeiten (Releasedate). 3. Auf Submit Button Klicken um die Änderungen zu speichern. 4. Über MySQLWorkbench kontrollieren ob alle Daten korrekt in die episode Tabelle übernommen wurden.
Erwartetes Ergebnis	Alle Änderungen werden in die Tabelle episode in der Fullstack Datenbank aufgenommen.
Ergebnis: Erfolgreich	Alle Änderungen wurden in die Tabelle aufgenommen.

Testfall 3	Löschen einer Episode
Testziel	Überprüfen, ob gelöschte Episoden aus der Datenbank entfernt werden.

Schritte	<ol style="list-style-type: none"> 1. Im episodes Screen auf den Delete Button eines bestimmten Films Klicken. 2. Über MySQLWorkbench kontrollieren ob der Film aus der Tabelle movie entfernt wurde.
Erwartetes Ergebnis	Die gewählte Episode wird aus der Tabelle episode entfernt.
Ergebnis: Erfolgreich	Die gewählte Episode wurde entfernt.

Beides

Testfall 1	Löschen eines Films und dazugehörigen Episoden
Testziel	Überprüfen, ob die zu einem Film gehörenden Episoden gelöscht werden, wenn der dazugehörige Film gelöscht wird.
Schritte	<ol style="list-style-type: none"> 1. Im Home Screen auf den Delete Button eines bestimmten Films Klicken. 2. Über MySQLWorkbench kontrollieren ob die dazugehörigen Episoden aus der episode Tabelle entfernt wurde.
Erwartetes Ergebnis	Sowohl der ausgewählte Film, als auch alle dazugehörigen Episoden wurden gelöscht.
Ergebnis: Erfolgreich	Der Film und die Episoden wurden entfernt.

Installationsanleitung

Datenbank-Setup (MySQL):

1. Eine Datenbank mit dem Namen fullstack kreieren.

Backend-Setup (Spring Boot mit Maven):

1. Den backend Ordner in deiner Java-IDE (z.B. IntelliJ) öffnen.
2. Evt. User und Passwörter im application.properties file ändern. Momentan sind diese u: root, p: root
3. Die Main Datei ausführen, durch das wird ein Server gestartet, der auf Port 8080 hört und die tabellen movie und episode werden kreiert.

Frontend-Setup (ReactJS):

1. Den frontend Ordner in deiner IDE (z.B. VSCode) öffnen.
2. Im Terminal npm install ausführen, um alle notwendigen Abhängigkeiten zu installieren.
3. Im Terminal npm start ausführen, dadurch wird das Frontend gestartet und läuft nachher auf <https://localhost:3000> .

Fertig!

Hilfestellungen

Mitschüler:

Mehrere Mitschüler haben mir geholfen, am meisten waren das

- Devrim Caglar
- Burak Akbas

Internetrecherche:

- ChatGPT
- PedroTech (Youtube)
- Code With Arjun (Youtube)

Lehrer:

- Patrick Meier, Lehrer der Module M294 und M295