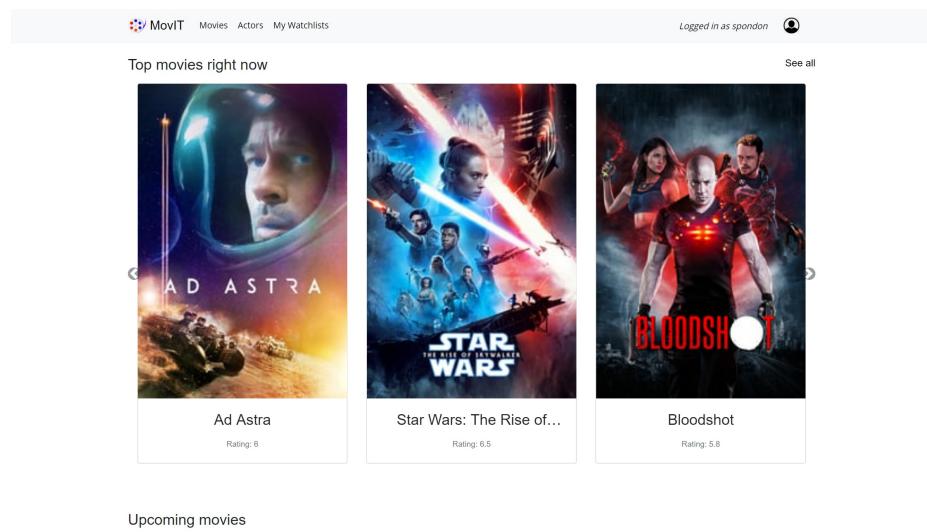


MovIT

DAT076 Webweebs - Final Report

Jesper Berglind, Mathias Lammers, Jian Shin
Spondon Siddiqui, Marcus Svensson,

2020-03-19



Upcoming movies

1 Introduction

MovIT is an application with the purpose to display information related to movies - including cast, plot summaries and reviews. A user can search for actors, movies, see information about the movie, rate them and add them to their watch list.

Without logging in a user can view movies, actors and reviews. After creating an account and logging in, the user is also able to leave reviews/comments on movies and add them to their own watch list.

Github repository: <https://github.com/SpondonSiddiqui/Webweebs>

2 Use cases

The following subsections of the report contains the different use cases of the MovIT application.

2.1 Sign up

As a user I want to create an account in order to fully make use of the website's functionality.

2.2 Log in

As a returning user I want to log in to my already existing account so that I can use the website to its fullest.

2.3 Browse movies

As a person looking for a new movie to watch I want to browse different movies to get some inspiration for what to watch.

2.4 Read about a specific movie

As a person looking for a new movie to watch I want to be able to read about a movie I am interested in so that I can better understand if the movie is worth watching.

2.5 Search for a movie

As a user I want to be able to search for an interesting movie I have heard about.

2.6 See which actors star in a movie

As a user who has seen a certain movie I want to be able to see which actors starred in that movie and read more about them.

2.7 Leave a review

As a user who has seen a certain movie I want to be able to leave a review so I can tell others what I thought about the movie.

2.8 See reviews

As a user I want to be able to see reviews of a movie I'm interested in so that I can get a grasp of what kind of movie it is and what others thought about it.

2.9 See watchlist

As a user I want to be able to see my watchlist so that I can keep track of what movies I want to watch.

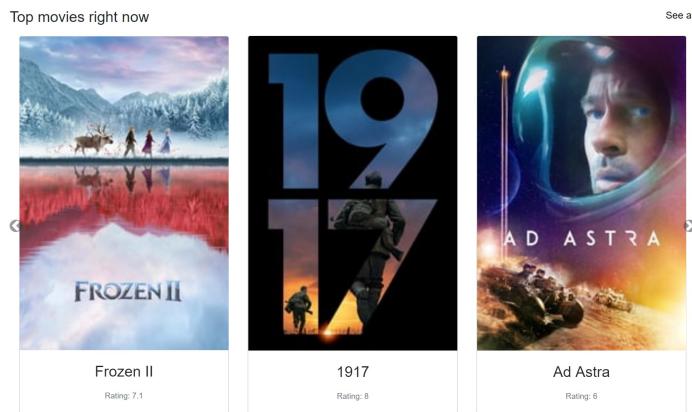
2.10 Add a movie to watchlist

As a user I want to be able to add a movie to my watchlist so I remember to watch it later.

3 User Manual

3.1 Home page

When opening up the application, the user is met with the following view.



Here, the user can view a few of the most popular movies in a carousel. This satisfies the first use case, where *A user wants to browse different movies to get some inspiration for what to watch*. If these movies are not enough, the user can click "See all" in order to view the current most popular movies.

If the user scrolls down a bit, they are met with the following view.

Upcoming movies



Downhill

Barely escaping an avalanche during a family ski vacation in the Alps, a married couple is thrown into disarray as they are forced to reevaluate their lives and how they feel about each other.

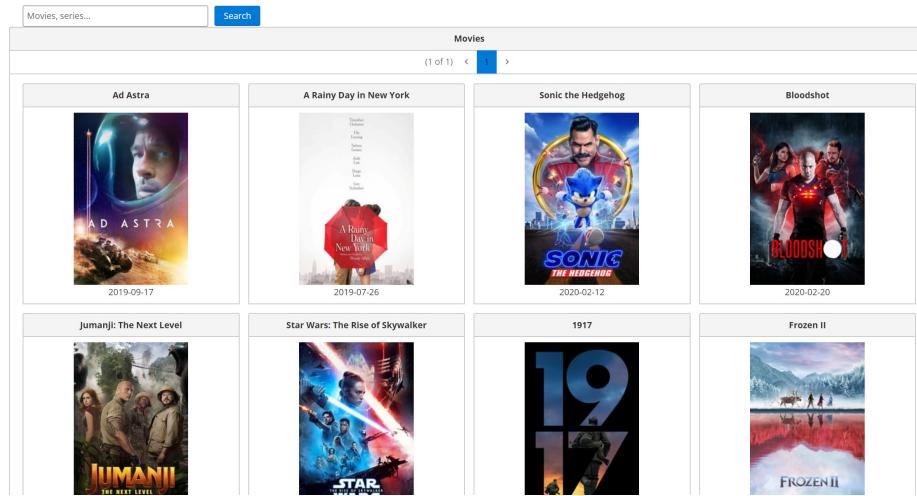


localhost:8080/lab3/detailedMovie.xhtml?Id=560391

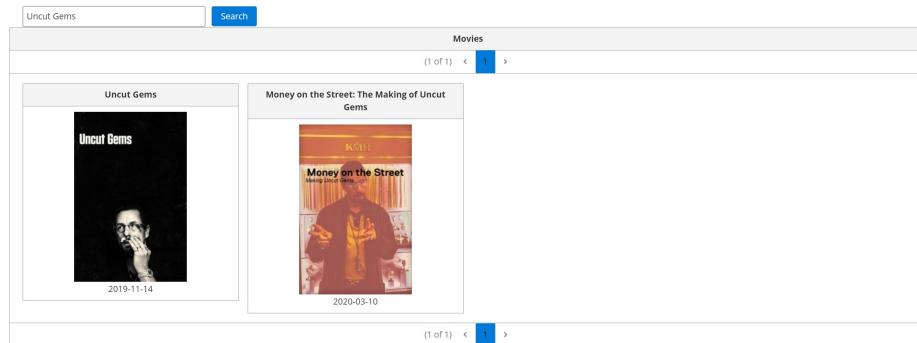
This section simply contains a few upcoming movies that the user might be interested in. If the user hovers over a card, they can read a short description of the movie, and can click on it to be brought to that movie's detailed page.

3.2 Most popular movies

If the user clicks on "See all", they are brought to the following page.



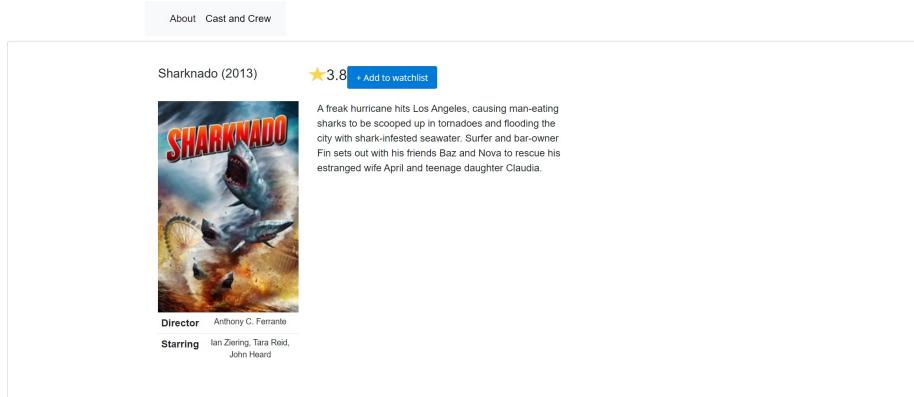
Here, the user can view more of the most popular movies. The movies are displayed in a thumbnail grid, and the user can click on a movie if they want to know more about it. If the user is looking for a specific movie, they can use the searchbar at the top, just below the navigation bar.



In this example, we can see that if the User searches for 'Uncut gems', the application returns all results relating to that name. By having this functionality, the use case *As a user I want to be able to search for an interesting movie I have heard about* is complete.

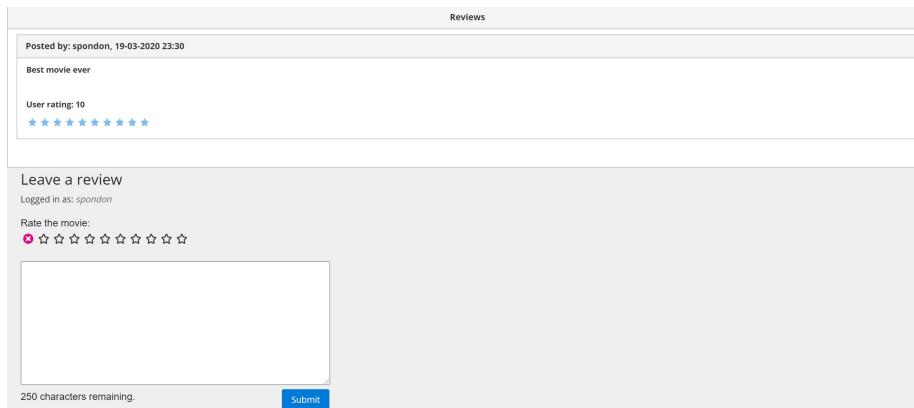
3.3 Movie details

If the user clicks on a movie, they are met with the following page.



On this page, they can see the title and a poster of the movie, when it was released, what rating it has, and a short description of it. With this page, the use case *As a person looking for a new movie to watch I want to be able to read about a movie I am interested in so that I can better understand if the movie is worth watching* is satisfied. The user can also choose to add the movie to their watchlist.

Furthermore, the user can see reviews of other people.

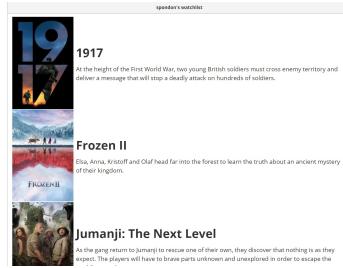


These reviews contain a star rating, a comment, which user wrote the review and when they left it. This fulfills the use case *As a user I want to be able to see reviews of a movie I'm interesting in so that I can get a grasp of what kind of movie it is and what others thought about it*. The user can also leave a review themselves, which makes it so the use case *As a user who has seen*

a certain movie I want to be able to leave a review so I can tell others what I thought about is completed.

3.4 Watchlist

The page for the users watchlist will look like this.



Here, the user can view what movies are in their watchlist, and can see the title and a short description of all movies in it. With this, the use cases *As a user I want to be able to see my watchlist so that I can keep track of what movies I want to watch.* and *As a user I want to be able to add a movie to my watchlist so I remember to watch it later.* are satisfied

3.5 Sign up

In order to fully make use of the application, the user needs to create an account by signing up. Once they have an account, they can leave reviews on a movie and add a movie to their watchlist.

A registration form titled "Register an account". It features a logo at the top left, a "Go back" link at the top right, and a large blue "Sign up" button at the bottom. The form includes fields for "Username" (with a placeholder "I") and "Password" (with a placeholder " "). Below the form is a link "Already a user? Click here to log in".

When clicking the sign up button on the navigation bar, they are brought to this page. The user can then create an account by typing in their desired

username and password and clicking sign up. If they already have an account, they can click the link at the bottom to log in. This page fulfils the use case *As a user I want to create an account in order to fully make use of the website's functionality*. Moreover, an empty watchlist is created for the user when they sign up.

3.6 Login

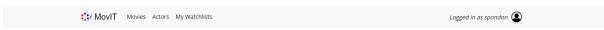
If the user already has an existing account, the user can use this page to log in to that account.

The form is a login interface. It includes a logo, a 'Go back' link, a title 'Log in to your account', two input fields for 'Username' and 'Password', a 'Log in' button, and a link to create a new account.

The user simply needs to input the same username and password as they did when creating the account. Once logged in, a link to the user's watchlist becomes available on the navigation bar. This makes it so the use case *As a returning user I want to log in to my already existing account so that I can use the website to its fullest* is satisfied.

3.7 Navigation bar

To reach these pages, the user can click on the links on the navigation bar, which can be seen at the top of every page. The "My watchlist" link is not rendered unless the user is logged in. The navigation bar also holds the log



4 Design

A substantial part of the applications functionality is tied to the TMDB(The Movie Database)-API, which handles requests for getting both movies and ac-

tors. The url-requests to the API is parsed by a static class which reads the json-responses and returns data relevant to the context. For the API to return any information every requests needs to be supplied with a personal key given when registering an account at TMDB. All this is handled by the static *JsonReader*-class, which receives requests from the DAOs. The structure and content of the responses can change depending on the context, for example: when searching for an actor using the API the response contains a field for movies starring the actor. However, this field does not exist when requesting the actors personal page. Fields containing the same information can also differ in name and total structure.

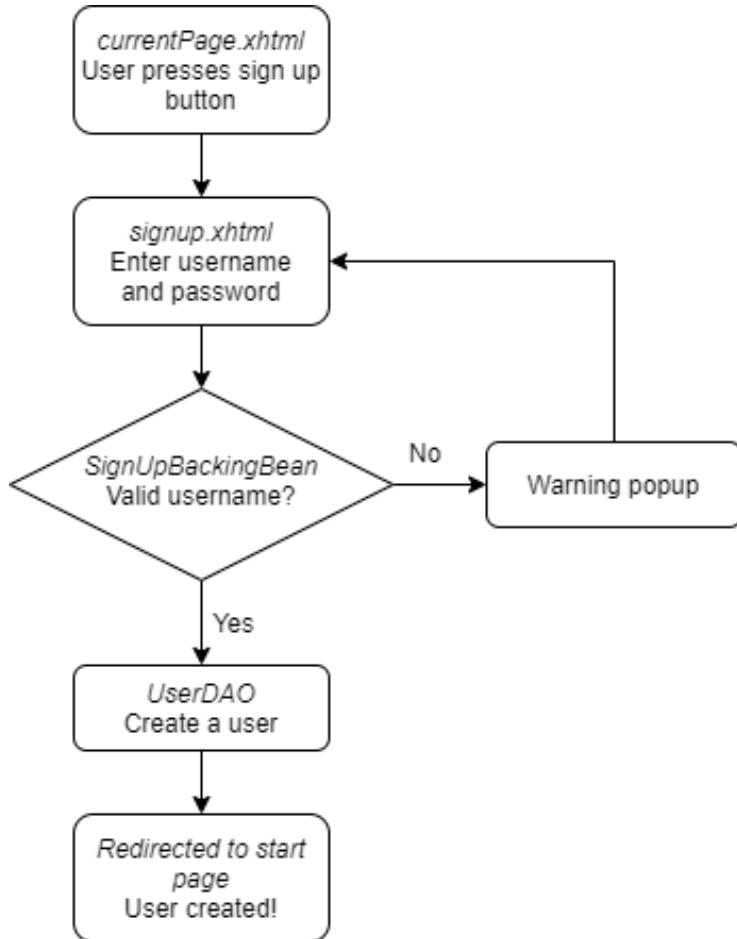
Primefaces has been used throughout the application, which is a library of UI-components. Examples of such UI-components are *dataGrid*, which is used for the thumbnail grid on the top rated page, and *commandButton*, which is used for the login and signup buttons on the navigation bar.

Most of the model can be found in the *entity* package. This package contains classes for *WebUser*, *Movie*, *Actor*, *Review* and *WatchList*. The *WebUser* class is used to represent person using the application. It contains the users username and password, which are used to log in. It also contains the user's watchlist in the form of a list of movies. The *Movie* class contains fields for the movies title, the average rating, an overview, a release date, a url to a picture, an ID, and a list of genres. These fields have names that are matched to the JSON file we receive from TMDB's API so that we can easily get the correct data for a movie. The *Actor* class is similar to the *Movie* class in the way that it pretty much only contains fields. It has fields for an actors name, their birthday, the day of their death (if deceased), a short bio, an ID, and a url to a picture. Again, these fields have names that are matched to the names given by TMDB's API for the sake of easily getting the actors' data.

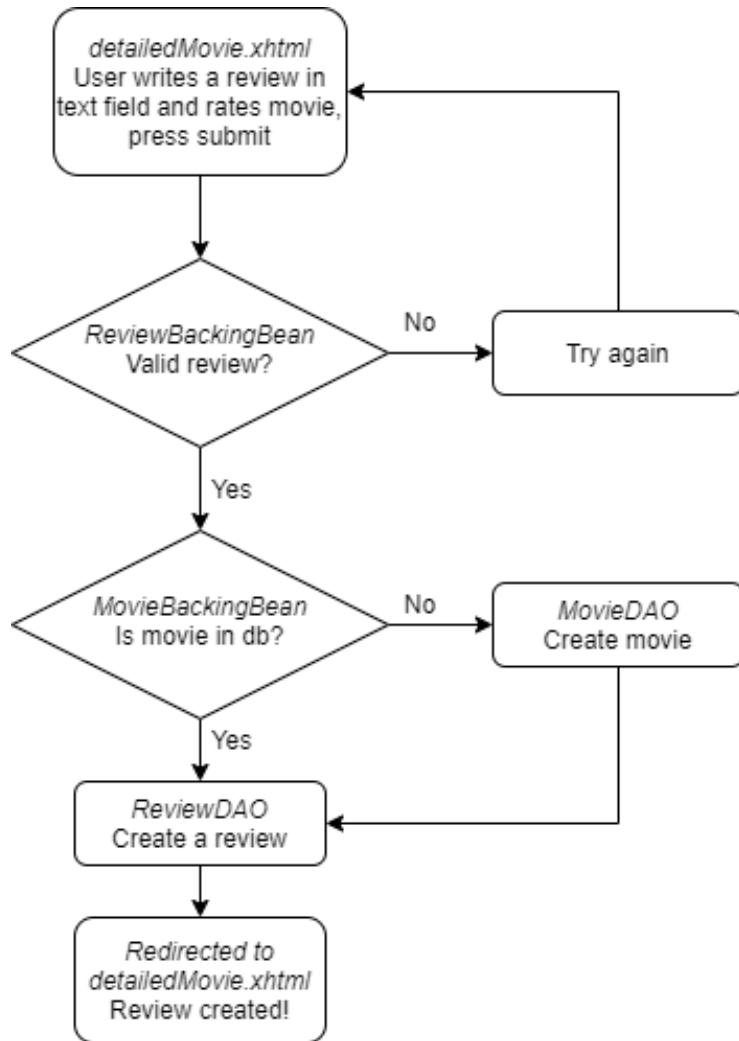
Both *Movie* and *Actor* gets their information indirectly via the *JsonReader*-class which handles requests from *ActorDAO* and *MovieDAO*.

5 Application flow

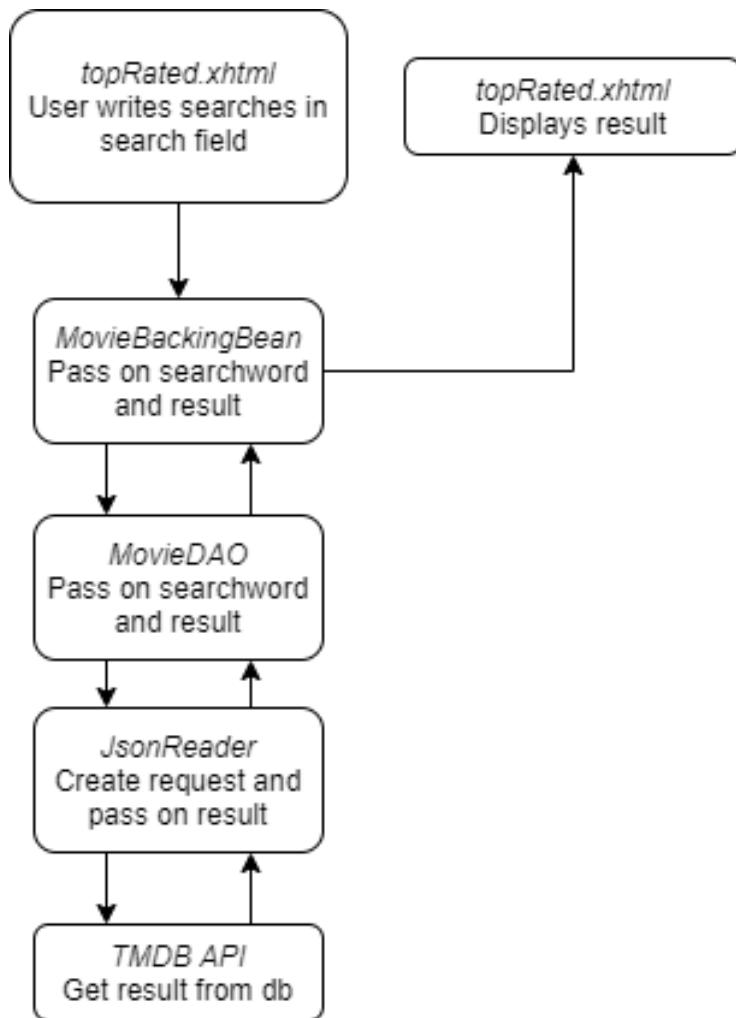
5.1 Signup Flowchart



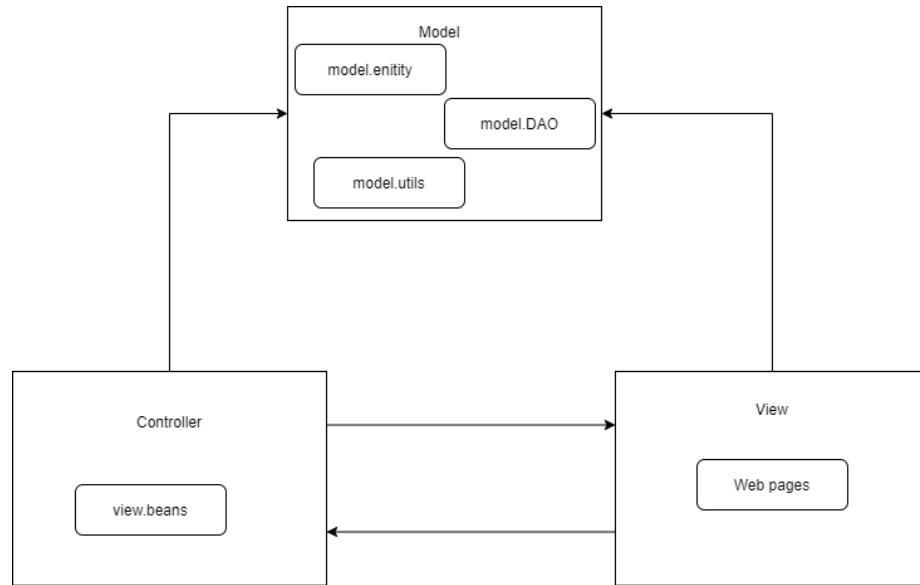
5.2 Leave Review Flowchart



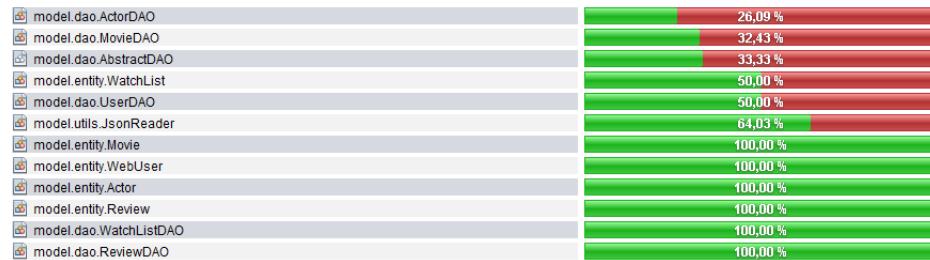
5.3 Search Review Flowchart



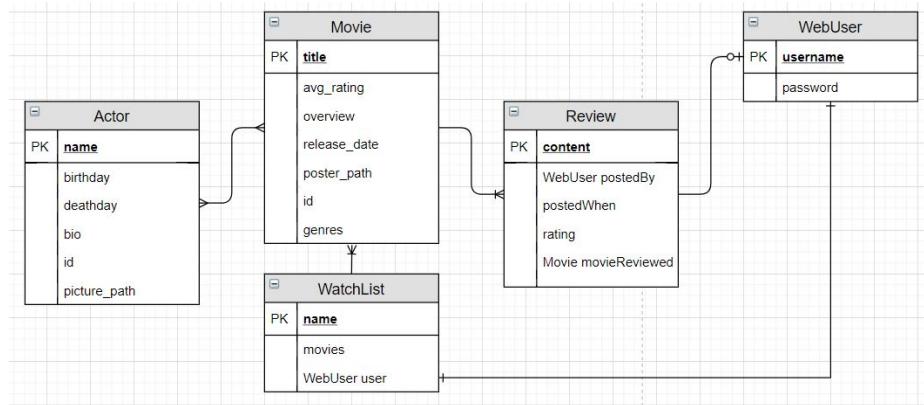
6 Package Diagram



7 Code coverage



8 Model diagram



9 Responsibilities

9.1 Jesper

Worked mostly with *JsonReader*, the DAOs, and the model. Wrote some tests as well.

9.2 Mathias

Mainly worked on the front page, creating the carousel showing trending movies and the flippable cards showing upcoming movies. Also worked together with Marcus on the functionality of being able to leave reviews, which are written to the database and read by and displayed on the corresponding movie page, as well as worked together with Jian on showing the information of a movie after clicking on it.

9.3 Jian

Mostly worked on the detailed movie page, showing the cast of a movie, showing a detailed actor page and showing the movies an actor has been in. Did most diagrams and the presentation.

9.4 Spondon

Worked with creating the logic for the watchlists as well as the views for them together with Marcus. Also worked on the log in page.

9.5 Marcus

Worked with creating a user + logging in, users being able to add reviews and rating to a movie, the navigation bar and user watchlist. Created a few tests and the test suite. We sometimes programmed in groups of two. Worked together with Mathias a bit on user reviews and with Spondon on the user watchlist.

