SOFTWARE ARCHITECTURES 2022

MOVIES REVIEWS JAVA PROJECT

ABOUT

SOAR Movie Reviews is an application that our group will develop using the java programming language. Our idea is to work on developing a website where users can create an account, read movie reviews and store their favourite movies in a personalised individual list. This website is owned by a movie review company that uploads different reviews written by film critics working inside the company that owns the application. Everyone can sign up to the website, they just need to enter a valid email address, a pseudonym and a password. Also, for users to easily search for various movies, movies are sorted by different categories matching the movie genre.

MODEL

A Model manages the data in an application

A) USER

A user is defined by the following data:

- a unique username,
- a password,
- unique email address.
- an individual list of favourites in which we keep the movies the user has set as favourites,

Users can perform the following actions:

- include movies in their favourites,
- delete them from their favourites,
- verify their password when logging in

B) FAVOURITE LIST

In a second stage , as optional features depending on time, we would like to include as user action the possibility of posting a review of the movies they have set as favourites and grade them from 0 to 5.

The favourites list is implemented a separate object defined by the movies the user has set as such. They share the same data as the movie items below.

C) MOVIES CATALOG

The movie catalog is implemented as a separate object defined by individual movies who have the following attributes:

- Title of the movie
- Photo
- Release data
- Synopsis
- Genre
- Review of the movie by our experts

ALEX BYSTRITZSKY | ELENA FERRADAL | SIMON DEMONT | YONAH BOLE

- Optional : rating average by users, comments of the users.

The user can access and filter the movies by genre, title and year of release. Optionally, they could also filter by rate.

VIEW

A) HOME PAGE

When launched, the application starts by printing the following menu, called Main Page, and waits for the user to choose an option. The different options are:

Enter:

[q] to quit the application

[1] to login

[2] to create a user account

Description:

[q] to quit the application: the user can quit the application, and the app stops.

[1] to login: the user can login, with his username and his password. If they are correct, the user home page runs.

[2] to create a user account: the user can create an account, by registering his email, his password and then a username.

If the user types anything else then ([q] [1] [2]), a warning message is printed.

B) USER HOME PAGE

After login, the application prints the following menu, called User Home Page, and waits for the user to choose an option. The different options are:

[q] to log out

[1] to see movies in the library

[2] to filter movies by categories

[3] to add movies in the list of favourites

[4] to see movies in the list of favourites

[5] to remove a movie from the list of favourites

[6] to verify their password when logging in

Description:

[q] to log out: the user can return to the main page.

[1] to see movies in the library: the user can see all movies in the catalog.

[2] to filter movies by categories: the user can filter the catalog by selecting a category.

[3] to add movies in the list of favourites: the user can add a movie to his favourite list.

[4] to see movies in the list of favourites: the user can see his favourite list.

[5] to remove a movie from the list of favourites: the user can remove any movies from his list of favourites.

[6] to verify their password: the user can check his password when logging in

If the user types anything else then ([q] [1] [2] [3] [4]), a warning message is printed.

CONTROLLER

A controller manages the communication between models and views.

LoginController. This controller holds username and password information of a user who wants to log in. If the user successfully logs in, the controller holds the current user. In addition, it manages log in and log out functions of the application.

UserController. This controller holds username, email, password, and favourites list information. It also manages creation of a new user account, including movies in favourites or not including them.

MovieFavouriteController. This controller holds films information listed in the favourites. It managed adding and removing favourites to the user's personal list.

A MOCK DATABASE

The mock database will keep the record of movies and users. This class will also manage adding and removing Users and movie objects.

EXCEPTIONS

- AlreadyExistsExeption. When a user already has an account and tries to sign in with their already registered pseudonym or/and email address.
- DoesNotExistsException. When a user wants to log in (and the username does not exist), to add a movie in their favourited list (and the title is not stored in the movie catalog), and to remove a movie from their favourite list (and the movie is not in the favourite list), this exception must be thrown.
- AlreadyFavouritedException. When a user wants to add a movie in their favourite list but the same movie is already stored in it, this exception must be thrown.