System Design Document: RateFlix

December 3, 2024

Table of Contents

CRC Cards:

Class Name: AccountView.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Provides a way for users to view their own account and edit features such as profile pictures, description, and username.**	**Collaborators:** - User.java

Class Name: LoginPage.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Provides a way for users to create an account of log in to access more services**	**Collaborators:** - User.java - RegisterPage.java

Class Name: RegisterPage.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Provides users a way to create an account if they do not have one**	**Collaborators:** - User.java - LoginPage.java

Class Name: ReviewPage.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Create a GUI of the review**	**Collaborators:** - Review.java

component - **Display all reviews made with associated users, time, and rating** - **Allow users to write and publish reviews with a rating**	

Class Name: WatchlistFrontend.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Provide a GUI of a list of movies that the user plans to watch** - **Provide ratings for these movies** - **Be able to manage this list (add/remove/alter)**	**Collaborators:** - Movie.java - Show.java - Watchlistitem.java - Movie Database API

Class Name: Movie.java	
Parent Class:N/A **Subclasses: N/A**	
Responsibilities: - **Movie object providing description of movie, title, cast, director, data** - **Setter and getter methods**	**Collaborators:** - WatchlistFrontend.java - Watchlistitem.java

Class Name: Review.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Review object provides components such as the review, rating, user, and**	**Collaborators:** - ReviewPage.java

timestamp - **Setter and getter methods** - **SQL Query methods (insert record)**	

Class Name: Show.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Provide description of shows, title, director, cast, and data** - **Setter and getter methods**	**Collaborators:** - WatchlistFrontend.java - Watchlistitem.java

Class Name: User.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **User object** - **Provide users with a username and password** - **Setter and getter methods**	**Collaborators:** - LoginPage.java - RegisterPage.java

Class Name: Watchlistitem.java	
Parent Class: N/A **Subclasses: N/A**	
Responsibilities: - **Provide movie or shows with a title, rating, and a status** - **Setter and getter methods**	**Collaborators:** - WatchlistFrontend.java

Class Name: DatabaseSetup.java
Parent Class:N/A **Subclasses: N/A**

Responsibilities: - **Java Database Connector** - **Easy access to database connection**	**Collaborators:** - LoginPage.java - RegisterPage.java - ReviewPage.java - WatchlistFrontend.java

System Interaction Description:

Operating System:

- Assumes compatibility with Windows, macOS, or Linux environments that support Java Runtime Environment (JRE).

Programming Language and Framework:

- Developed in Java using Swing for the graphical user interface.

Database:

- Uses SQL for data storage and management. Presence of a compatible relational database system such as MySQL.
- Database connection requires proper configuration such as JDBC driver, credentials, and connection URL.

Network Configuration:

- If the database is hosted remotely, assume a stable network connection and correct firewall settings to access the database server.

Java Environment:

- Requires Java Development Kit (JDK)
- End-users must have JRE installed for application execution.
- Requires Maven

Environment Configuration:

- A display environment is needed for Swing-based UI rendering.

Movie Database API

- Movie Database API key required to access movie images

System Architecture:

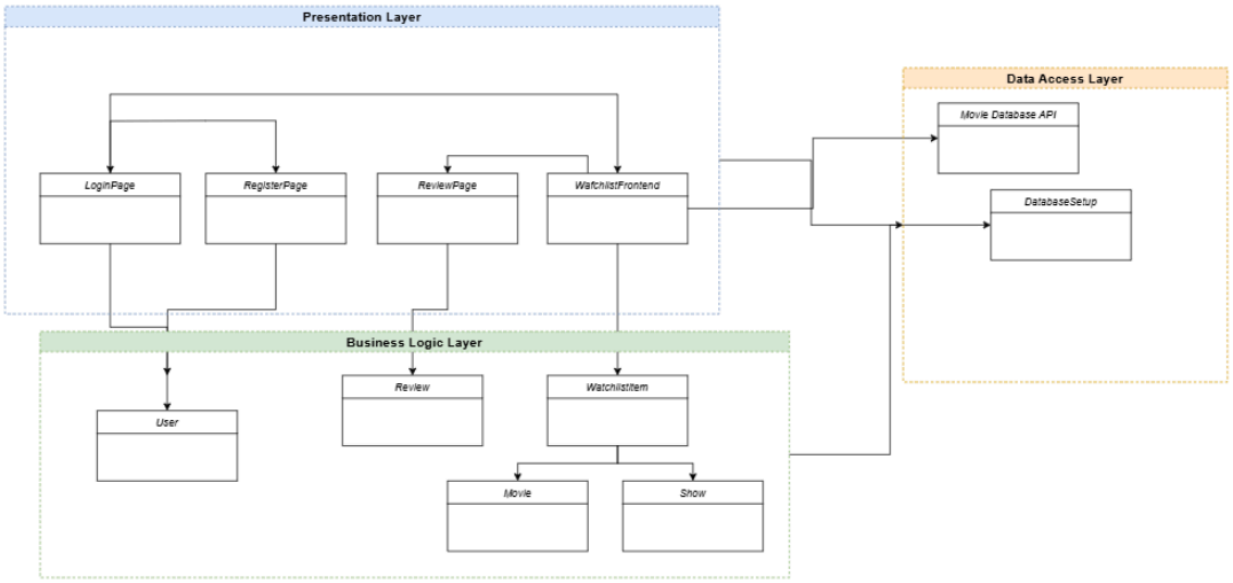
1. Presentation Layer (User Interface):
    - Components:
        - AccountView: Account viewing allows user to view their own account
        - LoginPage: Login page allows users to log in to their account
        - RegisterPage: Register page allows users to create an account
        - ReviewPage: Displays reviews for selected movies.
        - WatchlistFrontend: UI for managing and viewing the user's watchlist.
    - Responsibilities:
        - Manages user interactions.
        - Captures input (e.g., login credentials, movie search).
        - Displays output such as search results, reviews, and watchlists.
        - Communicates with the Business Logic Layer to process actions.
2. Business Logic Layer:
    - Components:
        - Movie: Represents movie objects, including attributes like title, genre, and description.
        - Review: Represents review objects, including content and ratings.
        - Show: Represents TV shows (a possible extension or shared with Movie for uniformity).
        - User: Represents the application's user, with attributes such as username, password, and watchlist.
        - WatchlistItem: Represents individual items in a user's watchlist, linked to movies.
    - Responsibilities:
        - Encapsulates core application logic:
            - Searching for movies or shows.
            - Adding/removing movies from the watchlist.
            - Fetching reviews for a selected movie.
3. Data Access Layer:
    - Components:
        - DatabaseSetup: Manages the SQL database connection and setup (e.g., schema creation).
        - Movie Database API: Collects thumbnails and movie/show posters for users to view and get a visual representation of it
    - Responsibilities:
        - Handles database interactions via SQL queries.
        - Inserts, updates, retrieves, and deletes data for the application.
        - Collects movie/show posters/thumbnails

Interconnections

- Presentation Layer ↔ Business Logic Layer:
    - UI classes (Home, LoginUI, RegisterUI, etc.) invoke methods from the business logic classes (User, Movie, WatchlistItem, etc.) to process user actions.
    - Business logic classes return the processed data to the UI for display.

- Business Logic Layer ↔ Data Access Layer:
    - Business logic classes invoke the DatabaseSetup component or objects to interact with the SQL database.
    - Objects return query results to the business logic layer for further processing.
    - Presentation Layer fetches Movie Database API to display movie/show posters

Diagram:



Link:
https://drive.google.com/file/d/1r2GylrWLs_994pz_ts0hc0qTc92j81Wm/view?usp=sharing