

System Design Document: RateFlix

November 21, 2024

Table of Contents

CRC Cards:	3
System Interaction Description:	7
System Architecture:	8
Interconnections.....	8
Diagram:.....	9

CRC Cards:

Class Name: Home.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Provides a welcoming screen for the user to navigate to other components 	Collaborators: <ul style="list-style-type: none"> - LoginPage.java

Class Name: LoginPage.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Provides a way for users to create an account of log in to access more services 	Collaborators: <ul style="list-style-type: none"> - User.java

Class Name: ReviewPage.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Create a GUI of the review component - Display all reviews made with associated users, time, and rating - Allow users to write and publish reviews with a rating 	Collaborators: <ul style="list-style-type: none"> - Review.java

Class Name: WatchlistFrontend.java	
Parent Class: N/A Subclasses: N/A	

Responsibilities: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Movie.java - Show.java - Watchlistitem.java
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------

Class Name: LoginUI.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Login UI 	Collaborators:

Class Name: RegisterUI.java	
Parent Class: Subclasses:	
Responsibilities: <ul style="list-style-type: none"> - Register UI 	Collaborators:

Class Name: Movie.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Movie object providing description of movie, title, cast, director, data - Setter and getter methods 	Collaborators: <ul style="list-style-type: none"> - WatchlistFrontend.java - Watchlistitem.java

Class Name: Review.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities:	Collaborators:

<ul style="list-style-type: none"> - Review object provides components such as the review, rating, user, and timestamp - Setter and getter methods - SQL Query methods (insert record) 	<ul style="list-style-type: none"> - ReviewPage.java
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------

Class Name: Show.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Provide description of shows, title, director, cast, and data - Setter and getter methods 	Collaborators: <ul style="list-style-type: none"> - WatchlistFrontend.java - Watchlistitem.java

Class Name: User.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - User object - Provide users with a username and password - Setter and getter methods 	Collaborators: <ul style="list-style-type: none"> - LoginPage.java

Class Name: Watchlistitem.java	
Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none"> - Provide movie or shows with a title, rating, and a status - Setter and getter methods 	Collaborators: <ul style="list-style-type: none"> - WatchlistFrontend.java - Watchlistitem.java

Class Name: DatabaseSetup.java

Parent Class: N/A Subclasses: N/A	
Responsibilities: <ul style="list-style-type: none">- Java Database Connector- Easy access to database connection	Collaborators: <ul style="list-style-type: none">- LoginPage.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) 8 or later for development.
- End-users must have JRE installed for application execution.

Environment Configuration:

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

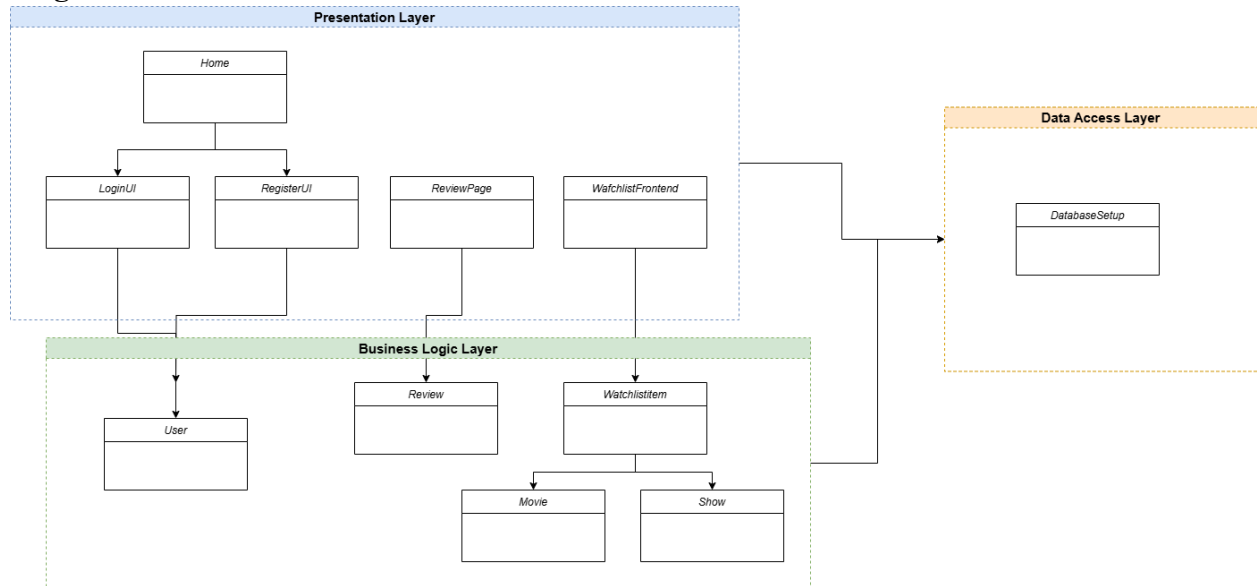
System Architecture:

1. Presentation Layer (User Interface):
 - Components:
 - Home: Main page for navigating the app.
 - LoginUI: User login screen.
 - RegisterUI: Registration screen for new users.
 - 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).
 - Responsibilities:
 - Handles database interactions via SQL queries.
 - Inserts, updates, retrieves, and deletes data for the application.

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.

Diagram:

Link:

<https://drive.google.com/file/d/1mKndiqRXwwp3wTbfJ3YsMWQOW8GL-Vul/view?usp=sharing>