**Software Requirements and Design Document**

**For**

**Group 18**

Version 3.0

**Authors**:

Malcolm Carroll

Aaron Garman

Damian Gonzalez

Joseph White

Sarah Swinnen

1. **Overview (5 points)**

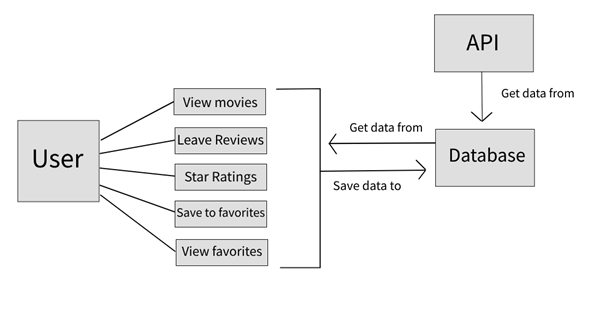
Our project is a web app that displays information about movies. Users can view the movies and can interact with them in ways such as leaving reviews or saving them to their favorites list. The movies are displayed in a scrollable list for users to see all of them. The data for the movies is pulled from the Movie Database API, and is stored in a Google Firebase cloud database. The endpoint that we chose is “Now Playing” to display current movies to users.

The front end of the app displays movie information to users using HTML, CSS, and Javascript. The back end uses Java to connect to the API and a Google Firebase cloud database to store the information about the movies. User interactions are also stored in the database so that they can be displayed in the future for data persistence. Each time the user opens the app the movie data is refreshed, and past interactions are displayed within the movies. The interactions that a user does is tied with their login account that is implemented with Google Firebase Authentication.

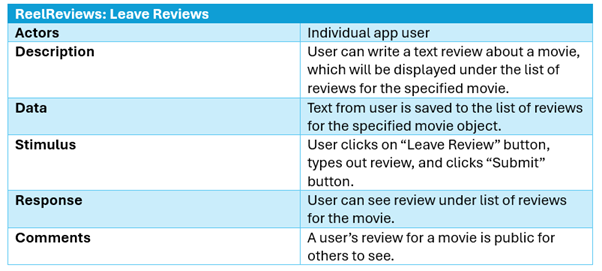
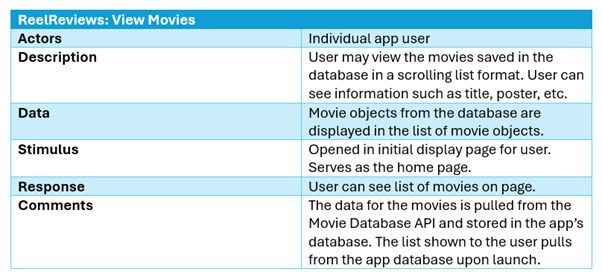
1. **Functional Requirements (10 points)**

1. Display movie title, poster, brief overview, release date, vote average, vote count, and language so that users can easily browse through available movie options. (High priority)
2. Reach API and save movie information into our database. (High priority)
3. Regularly update the database to stay up to date with newly released movies and store information in the database. (High priority)
4. Allow movie reviews and star ratings from users and store them in the cloud database. (High priority)
5. Allow users to add, view, and delete movies from favorites list, with all being recorded in the database. (High priority)
6. Users can sign in or sign out of their account and also sign out. (High priority).
7. **Non-functional Requirements (10 points)**

1. User should have to wait a minimal amount of time for API data to load
2. Codebase follows best code practices and code standards
   1. Rationale: following best practices and well-documented code is important when working with other developers.
3. Page routing takes minimal time to reduce user waiting.
4. Button actions take minimal time to reduce user waiting.
5. **Use Case Diagram (10 points)**



**Textual Descriptions of Use Cases**



1. **Sequence Diagrams and Class Diagram (15 points)**

1. **Operating Environment (5 points)**
2. Project is a web app, so will run on a browser on a desktop or laptop computer.
3. Should run on any operating system used, but the development and testing will be on a Windows 11 operating system.
4. Will be tested using Google Chrome for the browser, but it is expected that other browsers should be able to display it the same.
5. Should be independent and not require other applications for it to run along with it.
6. **Assumptions and Dependencies (5 points)**
7. Assume that the movie API will remain active for the duration of this project.
8. Assume the application will run on all computer types and operating systems since it is a web app displayed in a browser and not reliant on the system itself.
9. Assume that most browsers should be able to display the web app properly.
10. Assume the Firebase database should be active for the duration of the project.