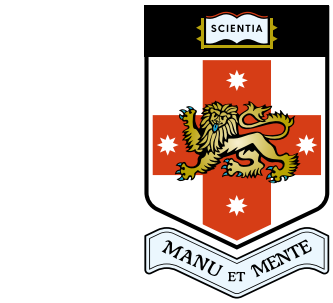


UNSW
SYDNEY

COMP3900 Computer Science Project

Project Proposal – Movie Finder System

By 3900-H18A-lawnchair

Name	**zID**	**Email**	**Role**
Christopher Wu	z5260104	z5260104@unsw.edu.au	Backend
Kane Qian	z5259933	z5259933@unsw.edu.au	Scrum Master Backend
Leighton Fan	z5258226	z5258226@unsw.edu.au	Frontend
Raymond Wang	z5258242	z5258242@unsw.edu.au	Frontend
Wayne Xu	z5255832	z5255832@unsw.edu.au	Backend

Submission Date: 19/06/2022

Contents

1 Background

1.1 The Problem

Cinema is an ever growing, major sector of the entertainment industry. With the fast-paced release of new movies comes the problem of finding suitable movies for the average audience member. Without the existence of movie finder systems, manually finding movies to watch would be a laborious and tedious task. However, despite the existence of many movie finder systems, each has its own significant drawbacks that indisputably adversely affect the movie finding experience. Therefore, movie finders want their movie finding experience to be fast, relevant, comprehensive, and personalised. When looking for movies, users must be able to access a database of all past, present, and upcoming movies released in their country. A movie finder system must allow the user to find movies they want to watch quickly, whilst producing highly relevant results. For example, the functionality to search for movies by keywords that match the movie name, description, genre, etc. would increase the efficiency of the movie finding experience. Furthermore, the system should be able to display extensive details about a requested movie to give users further information, and appropriate recommendations for similar movies.

Movie finders must also be able to leave and read reviews on all listed movies, including text and a rating from 0 to 5. By doing so, other movie finders will be able to read opinions to help them decide on whether to watch a movie. As the purpose of a movie finder system is to assist users in finding movies they want to watch, users should also be able to create a list of movies that they intend to watch and browse the lists of other users.

Overall, a movie finder system's purpose is to assist the user in finding more relevant movies efficiently.

1.2 Existing Systems and their Drawbacks

1.2.1 Rotten Tomatoes

'Rotten Tomatoes' is a popular movie finder system which covers many of the necessary functionalities. However, it's interface severely lacks aesthetic appeal, the website lacks interactivity to increase quality time spent on the site, has a predominantly white background which can cause eye strain, does not have a way to track movies watched, and only displays films released to the country of the user. Finally, the most glaring drawback of Rotten Tomatoes is its extremely limited list of search filters for browsing movies (genre, rating, audience score).

1.2.2 Letterboxd

'Letterboxd' is another movie finder system which unlike Rotten Tomatoes, has a dark background to reduce eye strain, has a more aesthetic interface, and also allows users to track movies they have already watched. However, there is also a lack of interactivity to increase quality time spent on the site, has limited search filtering options, only displays films released in the country of the user, and it does not have an option to blacklist reviewers.

2 User Stories and Sprints

2.1 Product Backlog and User Stories

The product backlog, sorted by highest to lowest priority, can be seen in Figure 1 and Figure 2 below. The descriptions and acceptance criteria for each user story can be found in 6. Appendix, which is sorted by their respective epics. Furthermore, each user story was assigned an estimated time to completion by using story points. Each story point is worth three hours of work for one person.

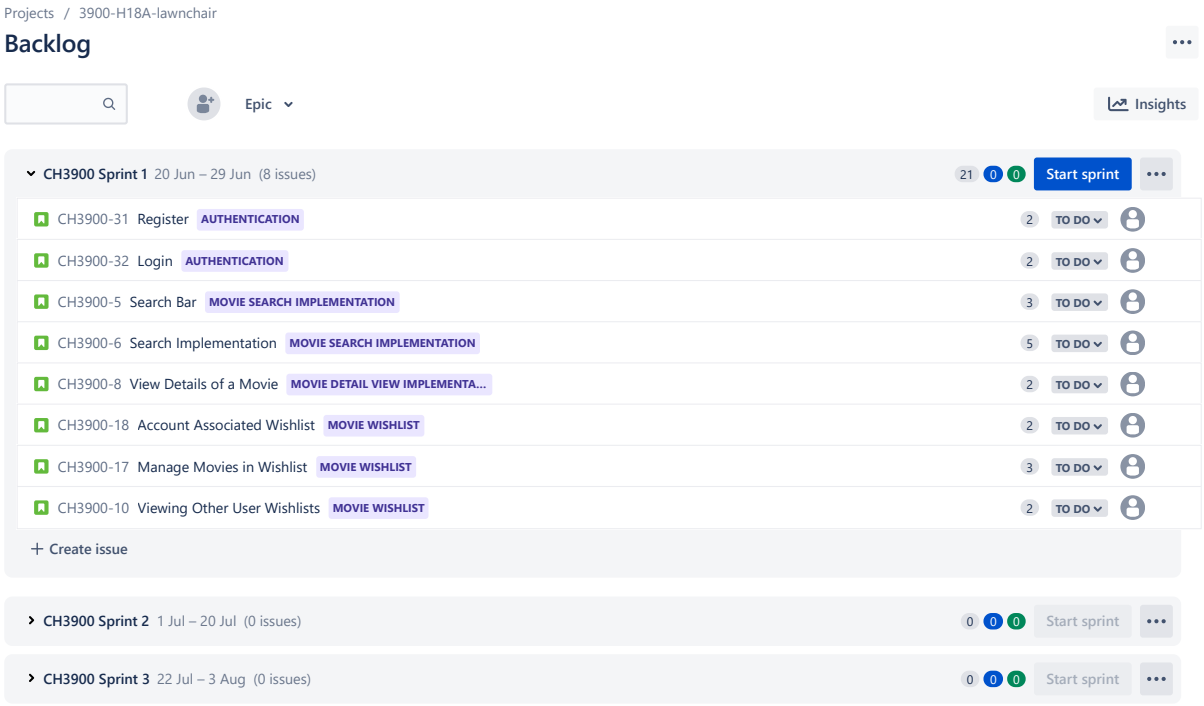


Figure 1: Sprint Structure Product Backlog

2.2 Sprint Structure and Details of First Sprint

As shown in Figure 1, the Sprints are structured as follows:

- Sprint 1: 20/06/2022 – 29/06/2022
- Sprint 2: 01/07/2022 – 20/07/2022
- Sprint 3: 22/07/2022 – 03/08/2022

Sprint 1 begins as soon as possible and ends the day before the first scheduled demonstration of the product. The second and third sprint both begin one day after the latest scheduled demonstration, and end one day before the next demonstration.

The tasks allocated to Sprint 1 shown in Figure 1 are comprised of the most fundamental functionalities of a movie finder system, along with the 'Movie Wishlist' epic due to the simplicity of its implementation along with its relative importance despite not being a fundamental functionality. The details of each user story in the first sprint can be found in 6. Appendix.

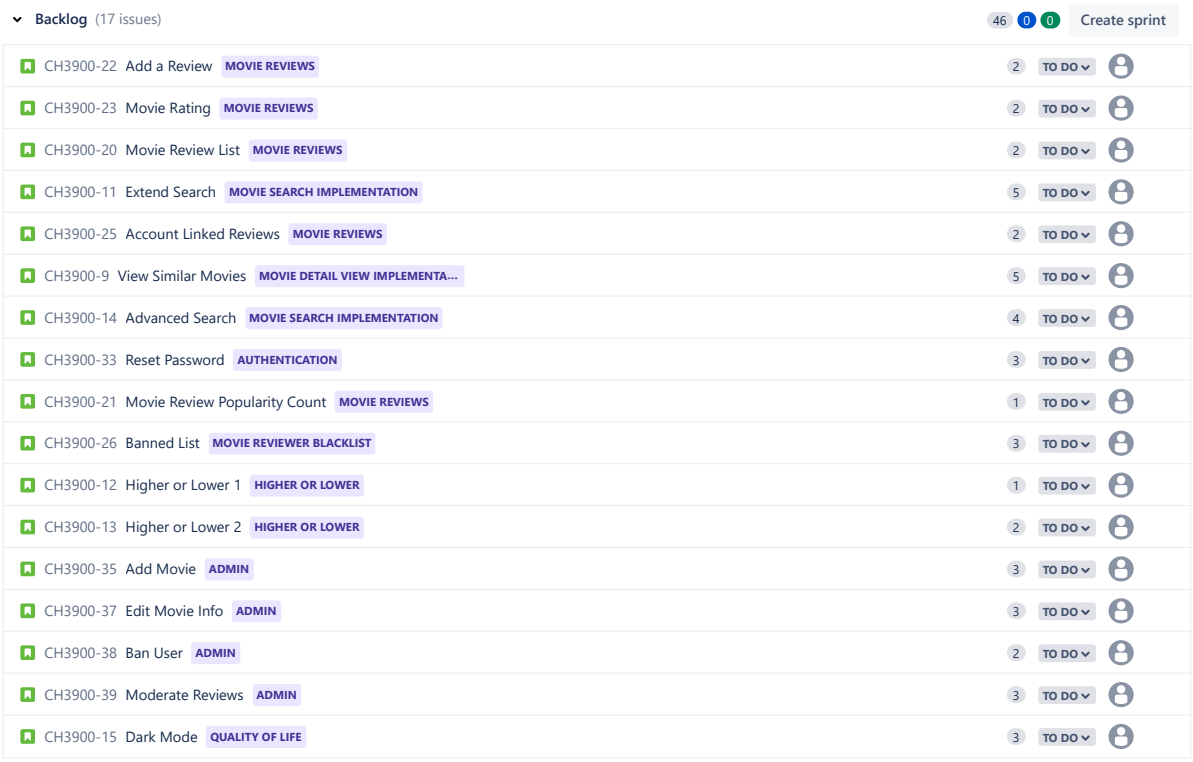


Figure 2: Backlog of Unassigned Issues

2.3 Satisfaction of Project Objectives and Novel Functionalities

Overall, the project objective can be described as a robust solution to the customer needs – 'fast', 'relevant', 'comprehensive', and 'personalised' as stated in 1.1. The Problem, whilst addressing the most significant shortcomings of the existing systems described in 1.2. Existing Systems and their Drawbacks.

Table 1 below shows which customer needs are satisfied by each user story, labelled by their issue number as seen in Figure 1 and Figure 2.

Customer Need	**Fast**	**Relevant**	**Comprehensive**	**Personalised**
User Stories	5, 6, 8, 9, 11, 12, 13, 14, 20, 21, 22, 23, 25	5, 6, 11, 12, 13, 14, 26	8, 9, 20, 21, 22, 23, 25	10, 12, 13, 15, 17, 18, 26, 31, 32, 33, 35, 37, 39

Table 1: Customer Needs Satisfied by User Stories

Qualitatively the following epics satisfy the customer needs as listed below:

- Movie Search is a fundamental feature of movie finders that increases the speed of finding relevant movies
- Movie Details are a fast way to provide comprehensive information on a selected movie
- Movie Wishlists provide a way for users to personalise their movie finding experience

- Movie Reviews are a fast way of providing a comprehensive collection of opinions from regular movie watchers and critics alike
- Quality of life features personalise the movie finding experience
- The 'Higher or Lower' interactive game is a fast and personalised way of providing movie recommendations
- The functionality of blacklisting movie reviewers allows users to personalise the reviews that they are exposed to
- Authentication allows for user accounts to personalise the movie finding experience
- Administration functionalities allows for the manual improvement all features and therefore all customer needs

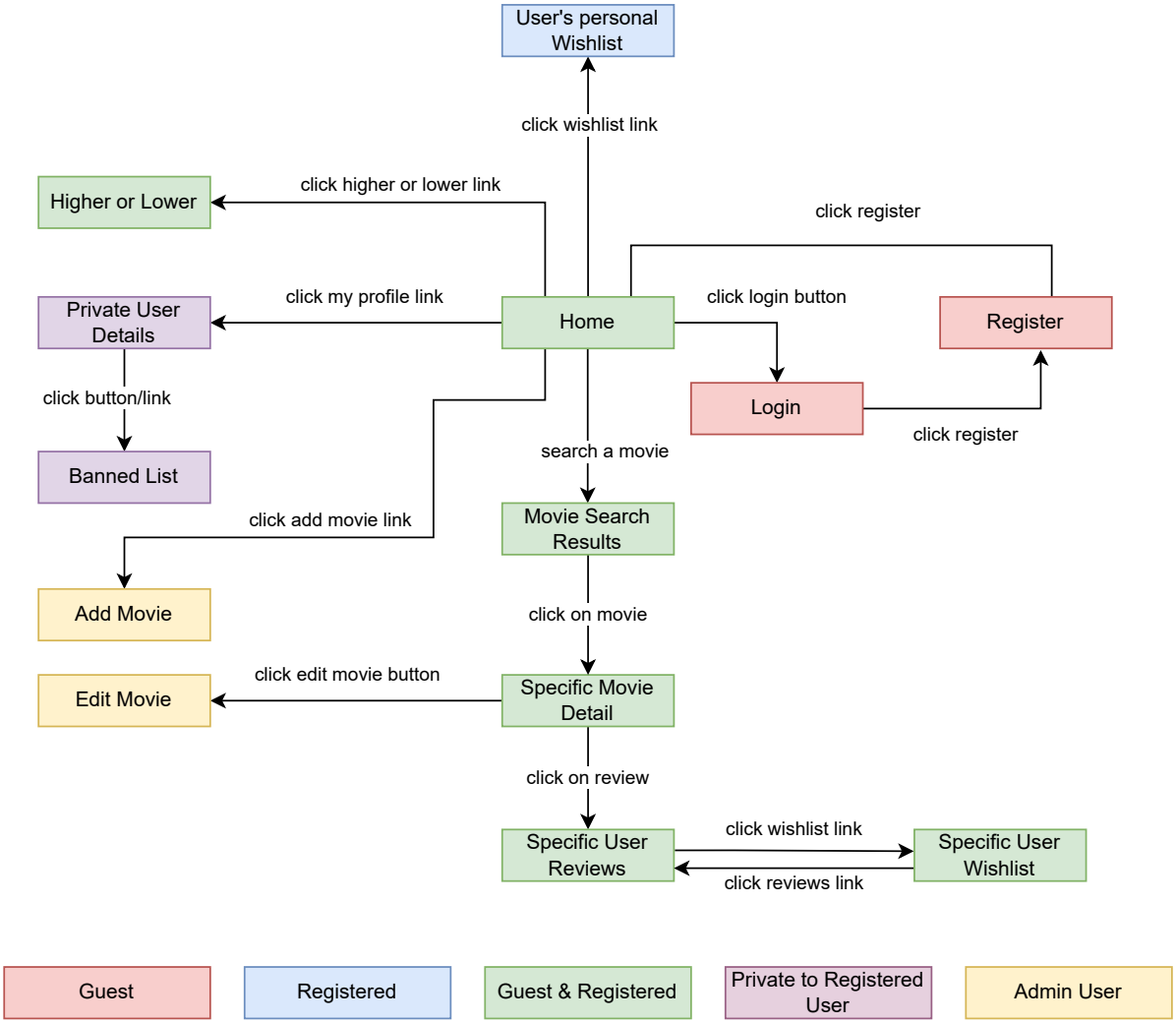
2.4 Novel Functionalities

Novel functionalities were added to address some of the shortcomings of existing movie finder systems discussed in 1.2. Existing Systems and their Drawbacks. These novel functionalities include:

- Advanced search (new filtering categories for more specific and relevant searching
- A 'Higher or Lower' interactive game which introduces new movies to movie finders through the game medium for increased engagement
- A light and dark mode toggle option which allows users to personalise their interface to reduce eye strain

These novel functionalities are presented in the form of user stories and are purposed to differentiate the movie finder system from its competitors.

3 Flow Diagram



Additional Information

All pages contain a navbar that is shown in the next section.

From the Non-Logged-in navbar you can get to Homepage, Movie Search Results, Higher or Lower, Login, Register.

From the Logged-in Member navbar you can get to Homepage, Movie Search Results, Private User Details, Personal Wishlist and Private Banned List.

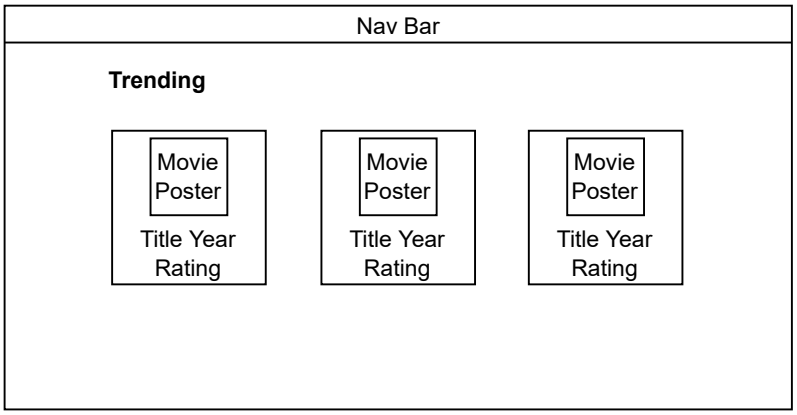
From the Logged-in Admin navbar you can get to Homepage, Movie Search Results, Higher or Lower and Add Movie.

4 Interface Diagrams

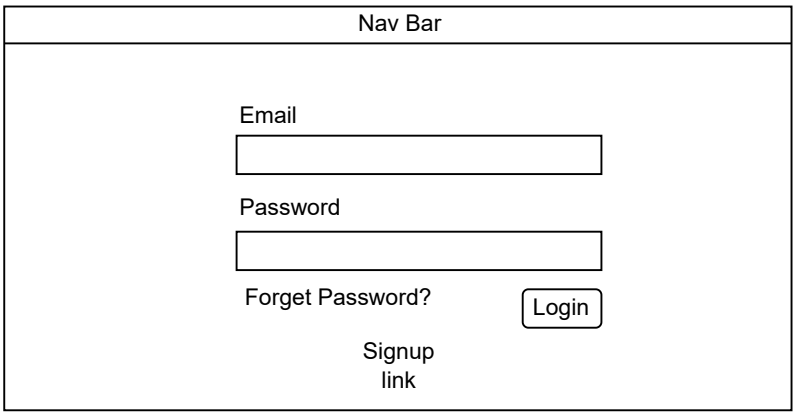
Visibility Indicators

Colours are used to indicate the visibility of components. For example, anything that is in purple is only visible to the logged in user that is viewing a page associated with their own account. Page visibility is shown in the flow diagram.

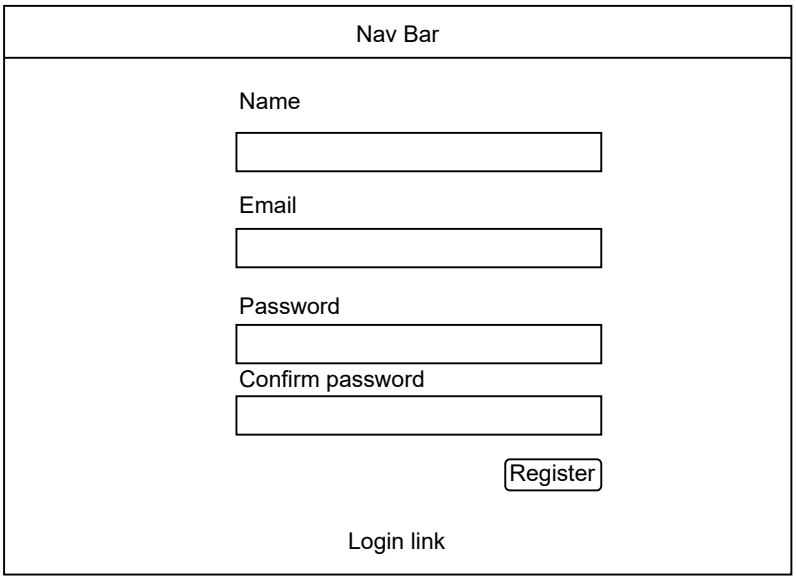
Home Screen



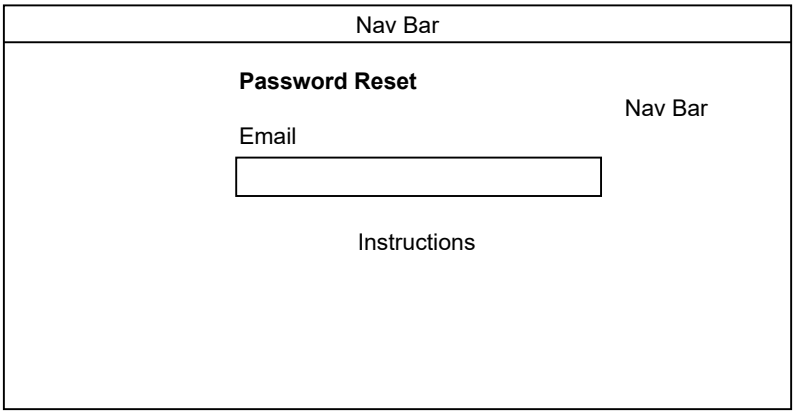
Login Screen



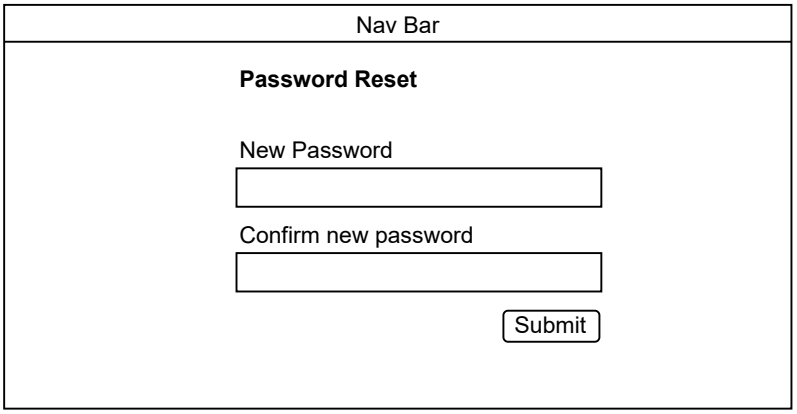
Register Screen



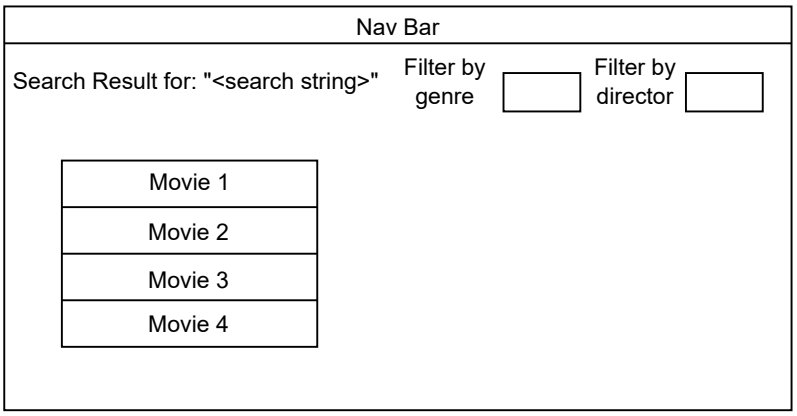
Reset Password Screen



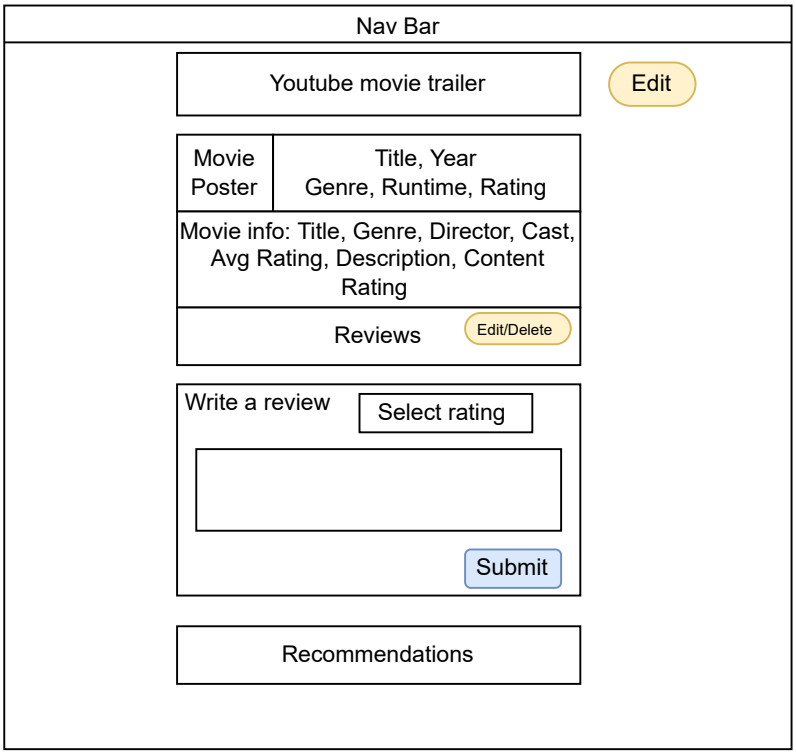
Reset Password
(clicked on email link)



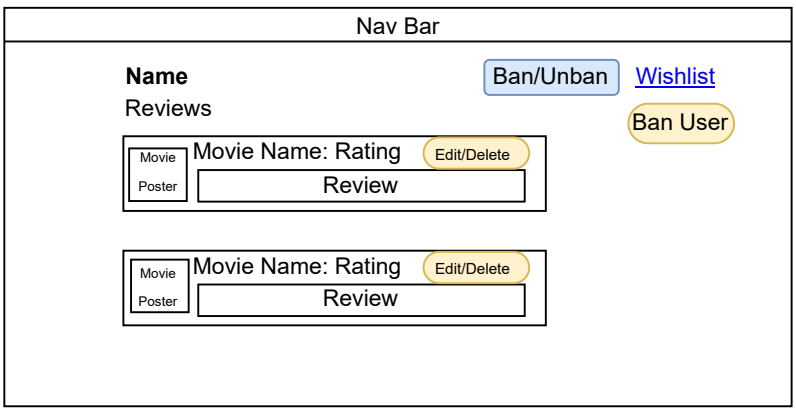
Movie Search Results Screen



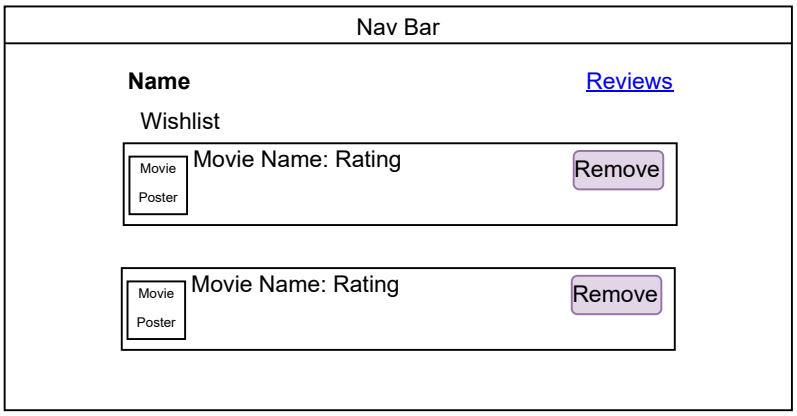
Specific Movie Detail Screen



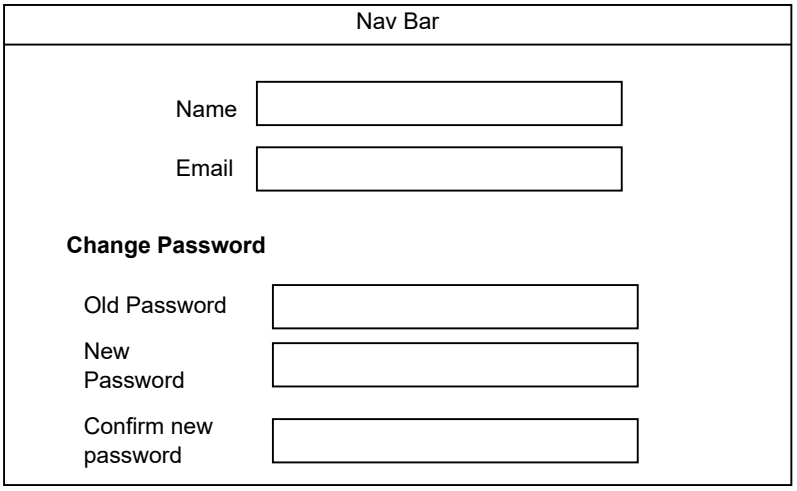
Specific User Profile Screen



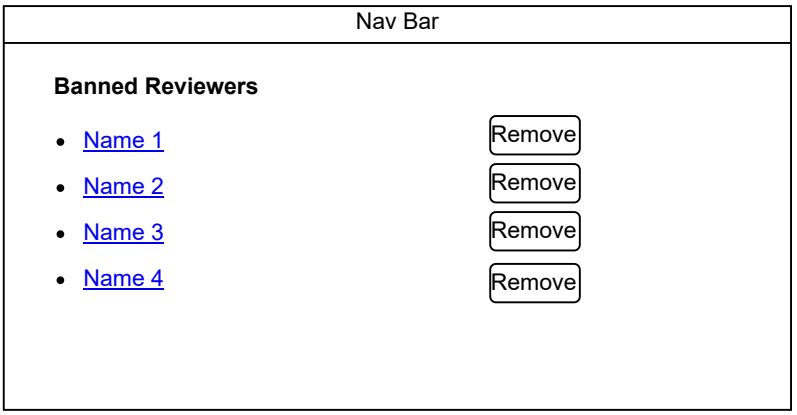
User Wishlist Screen



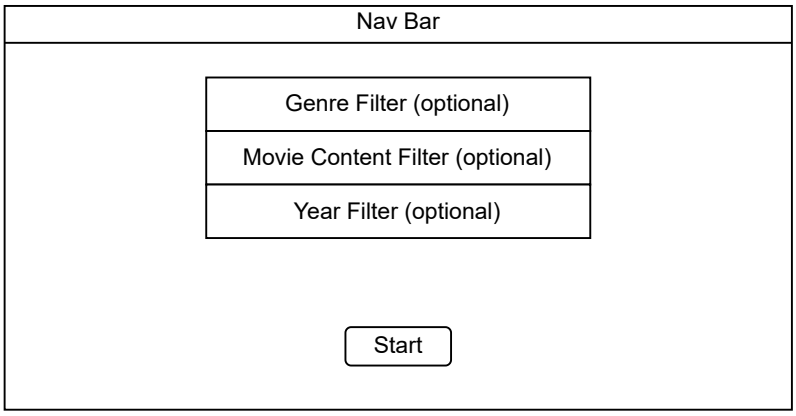
Private User Details Screen



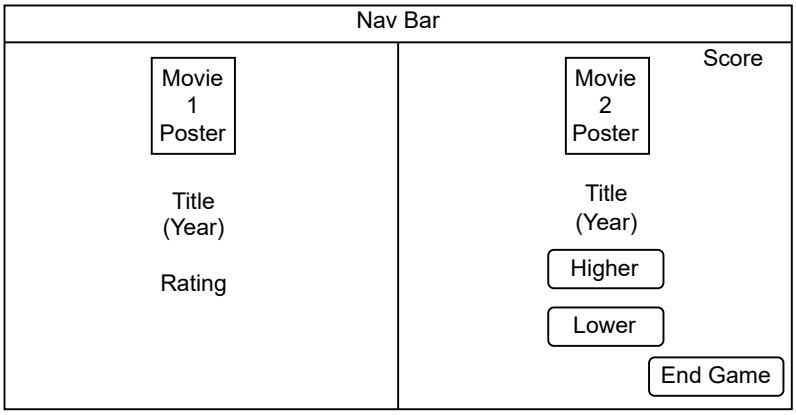
Private Banned List Screen



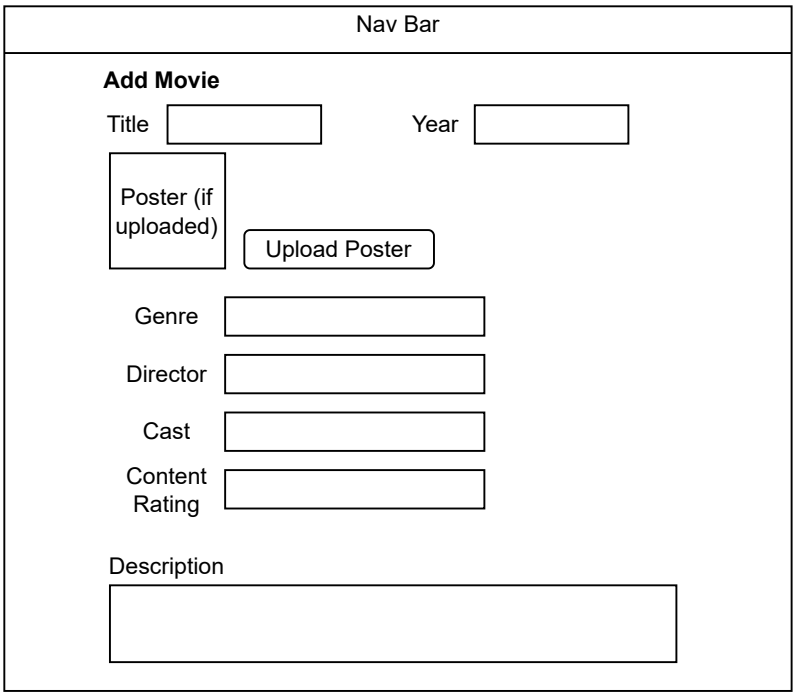
Higher or Lower Start Screen



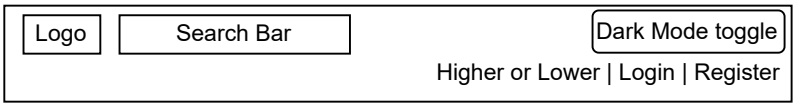
Higher or Lower Screen



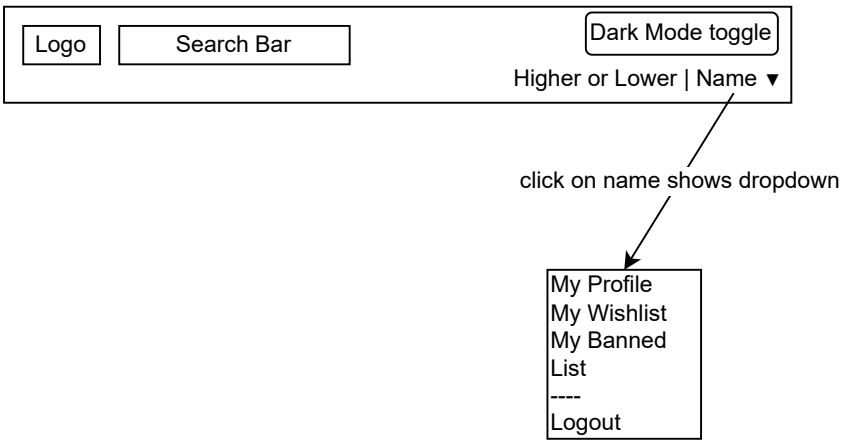
Add/Edit Movie Screen



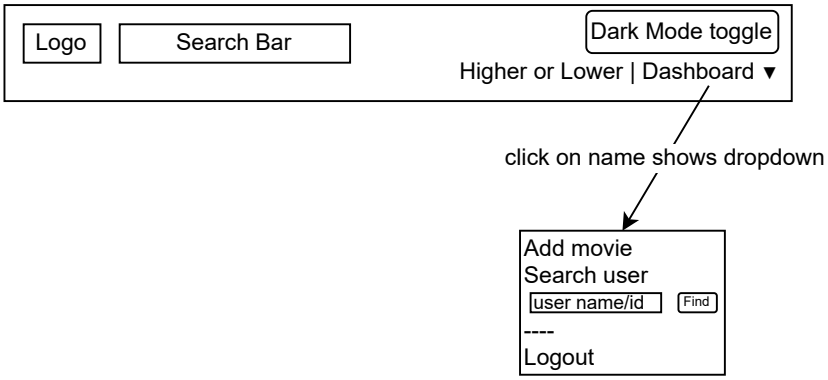
Nav Bar (Non-Logged-in View)



Nav Bar (Logged-in Member View)



Nav Bar (Logged-in Admin View)



5 System Architecture

5.1 System Architecture Layers

The system architecture proposed below consists of four layers that a request object will move through when interacting with our web application. Each layer's purpose and technologies are outlined in further depth below. Our system is designed to support four different classes of users; guests, registered users, administrators and developers. The user classes permissions within the system are defined in 5.2. System User Permissions.

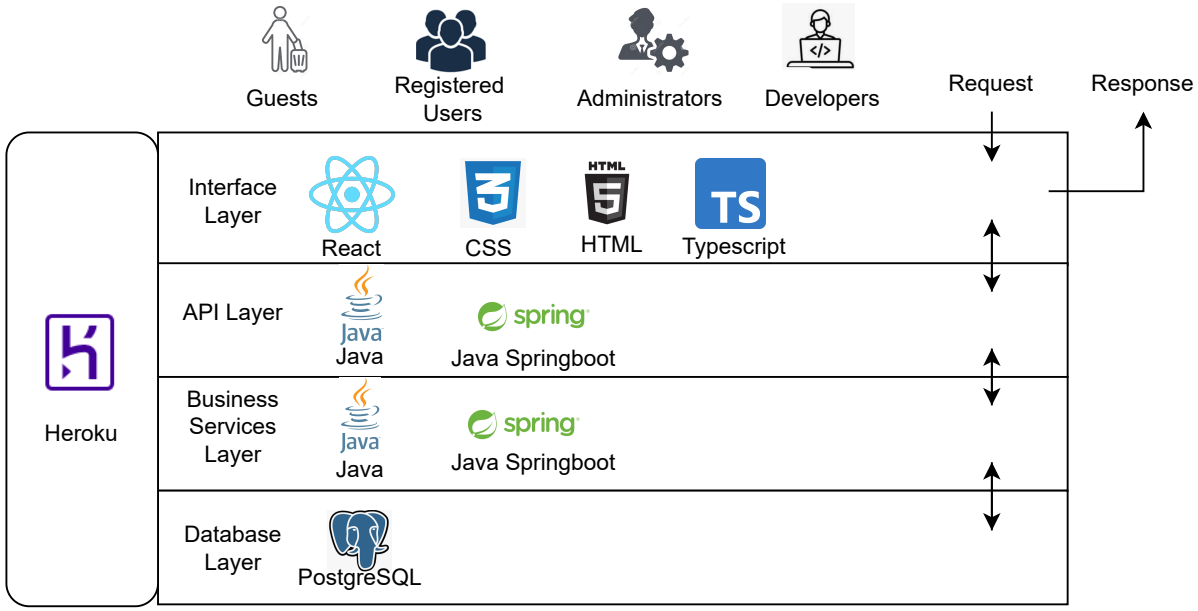


Figure 3: Layered System Architecture & Related Technologies

Interface Layer

The interface layer of the web application is the first point of contact all classes of users will have. Users will be able to interact with the server side of the web application through the react components built within this layer, where a user request from the interface layer will be passed to the API layer. It will also display the response's sent back from the API layer. It will be built using typescript and leverage the react framework.

API Layer

The API layer connects the interface layer and the business logic, where a request from the interface will call a custom-written REST API which then sends the request to the business services layer. This will work vice versa as well for response objects. Our API will be written in Java and utilise the Springboot framework.

Business Services Layer

The Business Service Layer will handle all logic that the movie finder will conduct. It will also be written in Java and utilise the Springboot Java framework. This layer will implement the logic required for all user stories to be functional.

Database Layer

The database will house all the data required for our movie finder system. Using postgreSQL, we will create two ER models based on the different types of data required:

- User Information
- Movie Information

5.2 System User Permissions

Different level of user permissions will be granted based on the user's access level. The different user levels are guests, registered users, admins and developers. Guests consist of users who navigate the website without creating an account and registered users are users who have created an account.

Admins and developers are defined as purely managerial roles. Admins hold the strict purpose of maintaining and updating the website with new data as well as enforcing set rules (e.g. removing inappropriate comments). Developers are admins with extended permissions, they hold the ability to change the websites appearance and functionality as well as managing admin accounts. A full list of each permission granted for different user levels is show in Table 2.

	Guest	User	Admin	Developer
Search Movies (Including Filters)				
Browse Movie Data				
View Reviews/Ratings				
Player Higher or Lower				
Use Light/Dark Mode				
Register an Account				
Login to Account				
Change Password				
Reset Password				
Wishlist Movies				
Blacklist Reviewers				
Create/Write Reviews				
Ban Users				
Add/Remove/Edit Movie Data				
Edit/Delete Reviews				
Change Website Appearance				
Change Website Functionality				
Create/Remove Admin Accounts				

Table 2: User Permissions

5.3 Proposed Technologies/Languages

Our groups proposed technologies and languages were decided by considering the familiarity of each member in specific languages as well as how well different systems would integrate together to create the web application.

5.3.1 Heroku

The website will be deployed with the assistance of Heroku, a cloud platform as a service, which is both simple to use and contains a free tier suitable for completing our application. Deploying the website with Heroku will allow easier development through their pipelines and CLI while also having inbuilt server and network management. By using a cloud platform, we will be able to effectively host our website and utilize a shared database, which will further allow easier development between multiple team members.

5.3.2 Frontend Technologies/Languages

The frontend component of our project will be developed using React Typescript and package managed by NPM. React was chosen due to the inbuilt features of reusable components as well as rendering optimizations over other similar languages, which perfectly suits our groups needs for developing a movie finder due to many reused components as per our interface diagrams. Typescript was chosen as it is considered an object-oriented superset of JavaScript. The addition of "types" as well as other various features improves stability with better error checking in compilation and code development, allowing a more streamlined frontend development process. NPM was chosen as it is the default package manager that is popularly used due to easy installation and management of dependencies as well as being able to keep track of package versions.

5.3.3 Backend Technologies/Languages

The backend framework will be developed in SpringBoot java (java 11). This was chosen due to it being a very stable framework that is optimised for development productivity in creating spring-based applications. The ability to reduce "boilerplate" code and forgo heavy use of configuration files amongst other time-saving features will allow efficient development. SpringBoot's integration with Gradle has also led our group to choose it as our backend build automation tool due to its high flexibility and performance in various build, documentation and distribution tasks.

Our database will be created using PostgreSQL, an open-source object-relational data management system which can handle a wider range of data types than traditional alternatives. This will work well in implementing our entity-relationship models for user and movie information. Its proven robustness in handling large data as well as inbuilt integration with Heroku made it an ideal choice for our movie finder web application.

6 Appendix

6.1 Movie Search Implementation

Search Implementation

Attach | Add a child issue | Link issue

Description

As a user, I want to be able to find a movie based on a certain criteria so that I can watch it.

Acceptance Criteria

1. Users are able to search based on the following:

- Movie Name
- Description
- Genre
- Content Rating

2. The movie finder will return a list of results with:

- Matching movie names
- Average Rating
- Movie Poster

3. The search should be performed in less than 5 seconds.

4. If no movie is in the database that matches the input, then the search should return that there were no results found.

Search Bar

Attach | Add a child issue | Link issue

Description

As an user, I want there to be a search bar, so that I can specify my search terms. (Frontend search bar)

Acceptance Criteria

- Search bar for user to enter a string.
- Page changes to a list of recommendations based on search when enter is pressed.
- Upon searching, will redirect to a new page.

Extend Search

Attach | Add a child issue | Link issue

Description

As a user, I want to browse movies by director/genre so that it is easier to find movies to watch.

Acceptance Criteria

- Movie finders can browse movies by director or genre
- The resulting movies are sorted from most popular to least popular based on the movie average rating
  - Movies with an equal rating are sorted alphabetically
  - Movies with no rating are treated as having an average rating of 0

Advanced Search

Attach | Add a child issue | Link issue

Description

As a user, I want to filter movies by year/country/actor/studio, so that I can more easily find movies that match my preference to watch.

Acceptance Criteria

Functional Requirements

- Add search filter for movies by
  - Year (before, equal, after)
  - Country of production
  - Language (Languages spoken in movie)
  - Actor (If an actor appears in the movie)
  - Studio (If the movie is produced by a studio)

Non-Functional Requirements/Performance concerns and Guidelines

- The search must show results within a reasonable time (5 seconds)
- The filter used in the search should be easily identified (i.e the user should see which filter they have selected)

Impact on other Features

- The advanced search options should be an addition to the previous search implementation, It should not impact previous functionality in any way if a user was to not use advanced search.

Negative scenarios of the functionality/ User experience concerns

- Some movies may not show up by a specific filter if there is missing data about the movie (e.g no data on which actors are in film)
- Some movies may be wrongly categorised due to incorrect data on the movie

6.2 Movie Detail View Implementation

View Similar Movies

Attach | Add a child issue | Link issue

Description

As a user, I want to be recommended similar movies when viewing a movie's details, so that I have more movies to watch.

Acceptance Criteria

1. When viewing a movie's details, the movie finder should also recommend similar movies based on:

- Least Review History
- Genre
- Director
- Specific Actor

2. An algorithm should be implemented to find similar movies based on the above criteria.

3. The algorithm should find these recommendations in less than 5 seconds.

View Details of a Movie

Attach | Add a child issue | Link issue

Description

As a user, I want to be able to see all details of a movie so that I can decide if I should watch it.

Acceptance Criteria

The details that a user should be able to see is shown below:

- Movie name
- Description
- Genre
- Cast
- Director
- Latest average rating
- Associated Reviews
- Content Rating

6.3 Movie Wishlist

Account Associated Wishlist

Attach | Add a child issue | Link issue

Description

As a user who is interested in a lot of movies at once, I would like to keep a personal list of the movies I want to watch in the future.

Acceptance Criteria

- Wishlist of movies directly associated with the logged in account
- Wishlist persists through separate login sessions

Manage Movies in Wishlist

Attach | Add a child issue | Link issue

Description

As a user, I would like to update my wishlist to ensure that it only includes movies that I still want to watch.

Acceptance Criteria

Users must be able to

- Add a movie to their wishlist
- Remove a movie from their wishlist
- Clear their wishlist

Users must not be able to:

- Perform any of the above actions on wishlists not associated with their account

Wishlists should be sorted alphabetically by default.

Viewing Other User Wishlists

Attach | Add a child issue | Link issue

Description

As a user, I want to view the wishlist of other users, so that I can know what movies they want to watch.

Acceptance Criteria

- Movie finders are able to view the wishlist of any other movie finder that has left a review for a given movie
- A message along the lines of "This wishlist is private" should be displayed when attempting to view the wishlist of a user who has set their wishlist to private
- Movies removed from the database should be automatically removed from user wishlists

6.4 Movie Reviews

Add a Review

Attach | Add a child issue | Link issue

Description

As a user, I would like to share my opinion on a movie for others to read.

Acceptance Criteria

Reviews must include:

- A rating out of a 5 star system with 0.5 star increments
- The option to write text up to 'n' characters in a text box
- A ‘post review’ and a ‘cancel’ button
- The status of the reviewer (e.g. Professional Critic)

Movie Rating

Attach | Add a child issue | Link issue

Description

As a user, I want to see a quantitative rating for the film by regular movie watchers and professional critics.

Acceptance Criteria

Movies will have the following ratings out of 5 stars with 0.5 star increments:

- An arithmetic mean of all user ratings
- An arithmetic mean of all critic ratings
- Ratings should be displayed as stars with different colours and a numerical (1 decimal place) value next to them representing user and critic ratings

Movie Review List

Attach | Add a child issue | Link issue

Description

As a user who wants others' opinions on movies, I would like to be able to view a list of reviews and ratings left by other users.

Acceptance Criteria

- Each movie has a tab including a list of all reviews left by users
- Reviews are presented with the comment and star rating visible
- The review list is by default sorted in descending order by ‘likes' - ‘dislikes’
- The review list shows a maximum of 10 reviews per page
- The review list can be navigated using ‘back’, ‘next’, or by numbered pages to show each page of 10 reviews

Account Linked Reviews

Attach | Add a child issue | Link issue

Description

As a user who values the opinions of some people more than others, I would like to be able to see all the reviews of a particular user.

Acceptance Criteria

- Any review a user leaves will be compiled in a list of reviews under their profile

Movie Review Popularity Count

Attach | Add a child issue | Link issue

Description

As a user, I would like to read reviews which give insightful perspectives and opinions.

Acceptance Criteria

- Logged in accounts can 'like' or 'dislike' reviews by clicking 'like' or 'dislike' buttons. Only one vote will count per account
- Each movie review will have a popularity score based on the difference between the number of likes and dislikes
- Sessions without a logged in account must not be able to like or dislike reviews

6.5 Quality of Life

Dark Mode

Attach | Add a child issue | Link issue

Description

As a user, I want a dark mode option, so that the website can match my thematic preference.

Acceptance Criteria

Functional Requirements

- The user must be able to change between dark mode and light mode

Non-Functional Requirements/Performance concerns and Guidelines

- The user must be able to identify which mode they have selected
- The theme of the website must be changed as soon as a mode has been selected (i.e not after reloading website).

Impact on other Features

- The selected mode must not impact the functionality of the website in any way that is not cosmetic.

Negative scenarios of the functionality/ User experience concerns

- The user must not be stuck in one mode without being able to change to the other.
- The user should be comfortable with navigating the page (i.e no black text on black background etc)

6.6 Higher or Lower

Higher or Lower 1

Attach | Add a child issue | Link issue

Description

As a user, I want to play higher or lower, so that I can be introduced to more movies I have never heard of.

Acceptance Criteria

- Movie finders can click a button which takes them to the Higher/Lower Interface

Higher or Lower 2

Attach | Add a child issue | Link issue

Description

As a user I want to filter the movies it shows me in higher or lower so that It gives me movies catered to me

Acceptance Criteria

Movie finders are able to filter movies by

- Year (before, equal, after, within)
- Genre
- Content Rating

6.7 Movie Reviewer Blacklist

Banned List

Attach | Add a child issue | Link issue

Description

As a user, I would like to add movie writers to my banned list so I will not be influenced by their opinions when it comes to movie recommendations/reviews

Acceptance Criteria

Users will see all banned users on a list

- and are able to unban/ban them from the list

When the user has blocked other movie finders:

- They are unable to see their reviews
- Ratings unaffected by their submitted ratings
- Gives an error when clicked on their name

6.8 Authentication

Login

Attach | Add a child issue | Link issue

Description

As a user I want to login so that I have a personalised experience.

Acceptance Criteria

- User is able to write in two fields that take in Email and Password
- Upon entering the correct credentials, User can click the enter button or keydown 'Enter' and login successfully
- User is able to have the privileges of a wishlist, banlist, etc.

Register

Attach | Add a child issue | Link issue

Description

As a user, I would like to register an account so I can login and use all the features available to me.

Acceptance Criteria

- Users will be able to have the text fields of Name, Email, Password and Confirm Password.
- Password must match Confirm Password
- Email must be a new email not currently registered in the database
- Upon filling in all fields correctly, User is able to register after clicking submit or keydown 'Enter' and prompted to the login page to login

Reset Password

Attach | Add a child issue | Link issue

Description

As a user, I want to be able to reset my password in case I forget my login credentials

Acceptance Criteria

- After providing a correct email address, A confirmation email will be sent to that email to change the password of the account.

6.9 Admin

Add Movie

Attach | Add a child issue | Link issue

Description

As an admin, I want to be able to add movies to expand the movie database and keep it up to date.

Acceptance Criteria

- The admin is able to successfully add a movie
- The movie is searchable and can be viewed by all users

Edit Movie Info

Attach | Add a child issue | Link issue

Description

As an admin, I want to be able to edit the details of a movie, to improve/update the movie information.

Acceptance Criteria

- The admin can edit the details of a movie
- The details of the movie are updated and can be seen by all users

Banned List

Attach | Add a child issue | Link issue

Description

As a user, I would like to add movie writers to my banned list so I will not be influenced by their opinions when it comes to movie recommendations/reviews

Acceptance Criteria

Users will see all banned users on a list

- and are able to unban/ban them from the list

When the user has blocked other movie finders:

- They are unable to see their reviews
- Ratings unaffected by their submitted ratings
- Gives an error when clicked on their name

Moderate Reviews

Attach | Add a child issue | Link issue

Description

As an admin, I want to be able to moderate reviews by deleting them so that I can manage spam/inappropriate reviews.

Acceptance Criteria

- The admin can delete the reviews of other users
- All users can no longer see the review if it has been deleted