CS 30700
Bee-Roll
Design document

Team 35:

- Aarna Maheshwari
- Artemii Akhunov
- Denis Kilseev
- Sreekar Tiruchunapally

Index

Purpose

Today, in the world of connectivity and social media abundance, there are not many dedicated places where true movie enthusiasts can share their own thoughts and connect with each other. Existing social media platforms such as Reddit have certain sections devoted to discussing movies, however, these platforms lack the specialized focus and features that are needed to create a dedicated community of movie lovers as they rather aim to provide a more general experience.

Bee-Roll aims to bridge this gap, seeking to create a place where everybody can communicate their experiences with movies. Apart from the “Typical Social Media” functionalities, like commenting or liking a post, Bee-Roll also provides an AI service, where the user gets a personalized watchlist recommendations based on their preferences. This unique idea of combining the “connectivity” aspect of social media and AI based service is guaranteed to place Bee-Roll into its own niche position. With features designed specifically for movie-enthusiasts, we aim to create a community that goes beyond generic discussions, offering a unique and engaging experience for every user.

Functional Requirements:

1. User Account
   As a user,
   a. I want to register and create an account using my email.
   b. I want to log in and out of my account.
   c. I want to change my account username or password.
   d. I want to disable or delete my account.
   e. I want to upload a profile picture and edit my display name
   f. I want to create a profile
   g. I want to update my profile if needed.
   h. I want to follow other users and stay updated about their activity
   i. I want to have a page for the people I’m following and the people that follow me
   j. I want to follow forums I am interested in

k. I want to be recommended movies based on my tastes
l. I want to create a watchlist and have the ability to add/remove movies from the watchlist
m. I want to create a public or private watchlist
n. I want to follow public watchlists that other users have made
o. I want to receive notifications when my profile or posts are interacted with
p. I want to be able to rate a movie and see it's rating
q. I want to be able to see a movie's main cast and description by hovering on the thumbnails
r. I want to see where a movie can be streamed when I click on the thumbnail
s. I want to be able to modify my "forum" alias

2. Discussion Posts and forums

   As a user,

   a. I want to write posts that support text, emojis and images.
   b. I want to view posts from other people
   c. I want to like and reply/comment on other posts
   d. I want to repost other users' posts
   e. I want to be able to tag my posts (like flairs on reddit) (if time allows)
   f. I want to be able to create/participate in a dedicated forum page
   g. I want to be able to be a moderator on a forum page
   h. I want to be able to share my public watchlists through posts
   i. I want to be able to mark if a post I make contains a spoiler

   As an admin,

   a. I want to delete and flag user posts
   b. I want to have a warning system to moderate posts
   c. I want to warn users for violations of ToS

   As a moderator,

   a. I want to ban other people from my forum
   b. I want to make my forum public/private
   c. I want to make posting free/restricted (through moderation) (if time allows)
   d. I want to be able to pin a post

e. I want to delete other posts on my forum
f. I want to be able to make other people moderators on my forum
g. I want to grant different permissions to other users on my forum

3. Search
   As a user,
   a. I want to search for movies using titles and keywords
   b. I want to search for other users using their username
   c. I want to search for other users with similar interests in movies
   d. I want to search for people with similar movies in their watchlists

4. Feed
   As a user,
   a. I want to have a feed of posts from people that I follow
   b. I want to have a feed of posts from forums I follow
   c. I want to be able to like posts, replies, and comments
   d. I want to be able to comment on other users' posts

5. Chat
   As a user,
   a. I want to create a personal chat with other people (if time allows)
   b. I want to be able to send images through chat (if time allows)
   c. I want to be able to send documents through chat (if time allows)
   d. I want to be able to send voice messages through chat (if time allows)
   e. I want to be able to create a group chat (if time allows)

6. Activity
   As a user,
   a. I want to see if a user is online
   b. I want to see when a user was last online

7. Guessing game
   As a user,
   a. I want to play a one-player guessing game where I am given two movies and must guess which one has a higher IMDB rating based on its synopsis (if time allows)

   b. I want to be able to play the guessing game with other people (if time allows)
   c. I want my rant to increase for each correctly guessed movie (if time allows)

Non-functional Requirements:

1. Client Requirements
   As a developer,
   a. I want the application to be able to used on any browser and OS

2. Server Requirements
   As a developer,
   a. I would like the server to be able to store all user, movie and chat data to a database
   b. I would like the server to support real time communication between the client and server

3. Performance Requirements
   As a developer,
   a. I would like the server to be able to handle 5000 simultaneous requests
   b. I would like the server to be able to handle 10,000 users
   c. I would like the response time to be under 500ms
   d. I would like all errors to be handled gracefully

4. Design Requirements
   As a developer,
   a. I would like any changes made to the Git repository to be deployed through build scripts

5. Appearance and Usability Requirements
   As a developer,
   a. I would like the website to be intuitive and aesthetically pleasing

6. Security Requirements
   As a developer,
   a. I would like any user data that we collect to be secure through encryption
   b. I would like to limit the amount of accounts that an individual can have to one a person

Design Issues

Functional Issues:

1. **What information is needed to create an account?**
   - **Option 1:** Username and password
   - **Option 2:** Username, email address, and password

Choice: Option 2

Justification: The username and password are essential to ensure that users of our platform will be able to create an account and log into the service. The email address is not essential but it will enable more functionality for the user in terms of security and the ability to reset their password if needed. Additionally, it allows us to ensure that each email address is only tied to one account, preventing users from making duplicate accounts.

2. **What information is needed to create a post?**
   - **Option 1:** Title and content
   - **Option 2:** Title, content, and rating
   - **Option 3:** Title, content, rating, and tags

Choice: Option 1

Justification: The title and content are necessary parts of making a post. Although option 2 necessitates a user rating and option 3 extends to needing a tag for the post, we chose to go with option one which is simpler and has more flexibility on the user end. By choosing option 1, users don't have the obligation to provide a rating or tags, resulting in a user-friendly and unrestrictive experience when making a post. While we will provide the option of providing a rating and/or tags to a post, the user will still be able to post without giving that information.

3. **How frequently will the AI model update and provide personalized movie recommendations?**
    - **Option 1:** Once a day
    - **Option 2:** Once a week

Choice: Option 1

Justification: Updating the recommendations once a day ensures that the recommendation system is responsive and evolves to the user's tastes and habits quickly and in a timely manner. This allows our platform to provide a more real-time and personalized experience for users.

4. **What privacy settings are available for users to control the visibility of their posts and profiles?**
    - **Option 1:** None
    - **Option 2:** Profiles can be set to private, making all posts visible to only followers
    - **Option 3:** Profiles can be public or private, and each post can be set to public or private

Choice: Option 3

Justification: By allowing users to have the flexibility of setting both their profiles and individual posts as public or private, we are able to ensure that each user has their needs met in terms of autonomy and privacy. This promotes a tailored and personalized user experience which many individuals look for in a social media platform.

Non-functional Issues:

1. **What web service are we going to use to host our backend?**
    - **Option 1:** AWS
    - **Option 2:** Google Cloud
    - **Option 3:** Azure

Choice: Option 1

Justification: AWS offers a robust and scalable infrastructure that aligns with the needs of our platform. Due to the storage and database solutions that AWS offers, along with the extensive documentation online, we will be able to ensure reliable hosting. Along with that, the tools and services that AWS offers will help us monitor, manage and scale our backend efficiently if needed.

2. **What language should we use to create our frontend?**
    - **Option 1:** Python + Flask
    - **Option 2:** ReactJS

Choice: Option 2

Justification: By using ReactJS, we will be able to create a dynamic and interactive user interface. Due to the fact that it is component-based, we will be able to efficiently develop reusable and modular UI elements, making development easier and helping us keep our codebase simple and scalable. Additionally, React has extensive

documentation and resources online, helping us effectively work through any problems that may arise during development.

3. **What language should we use to create our backend?**
    - **Option 1:** Python
    - **Option 2:** Express.js

Choice: Option 2

Justification: Express.js is a lightweight and flexible solution that focuses on simplicity and rapid development, which would help us deliver features in an efficient manner. The service also excels at creating RESTful API's and integrates well with many different databases. By choosing Express.js, we will be able to create a scalable, efficient, and easily maintainable backend for our platform.

4. **What type of database will we use?**
    - **Option 1:** MongoDB
    - **Option 2:** SQL
    - **Option 3:** Firebase

Choice: Option 1

Justification: We chose to use MongoDB as our database solution because it is a document-oriented service. Due to the fact that our movie data consists of text that is interconnected, such as the title and synopsis, the document-oriented structure is the best approach for efficiently storing and retrieving this content. SQL databases, while they are more powerful for structured data, have a more rigid schema that might be less suitable for our use case. Firebase on the other hand leans towards simpler data than what we are working with.

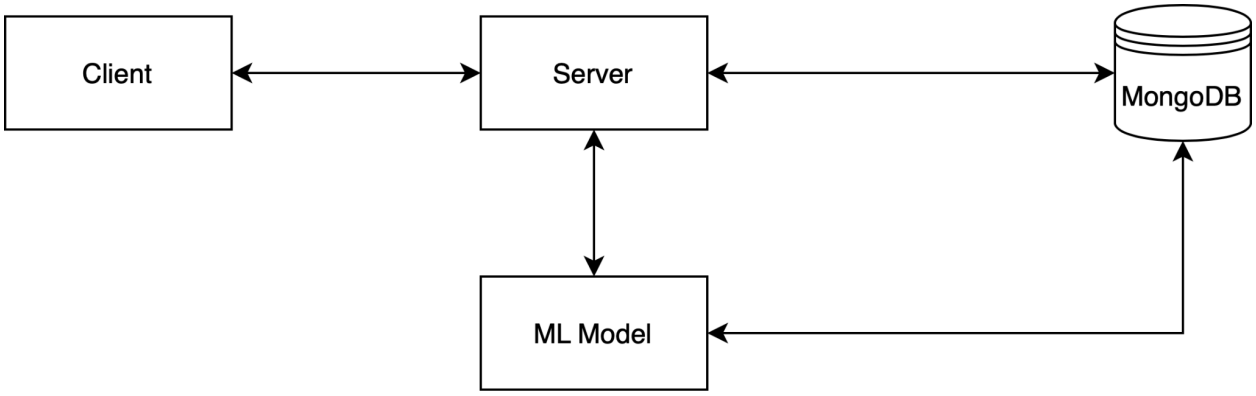
5. **How is the user experience maintained consistently across different operating systems and browsers?**
    - **Option 1:** Manual testing
    - **Option 2:** Using automation tools (Selenium)
    - **Option 3:** Using a real device cloud based tool (BrowserStack)

Choice: Option 1

Justification: Manual testing allows for us to not only test the functional aspects of the website but also subjective parts such as interactivity and aesthetics. Additionally, by not using other tools, we will be able to invest more time and effort in working on and refining the platform itself.

Design Outline

Components

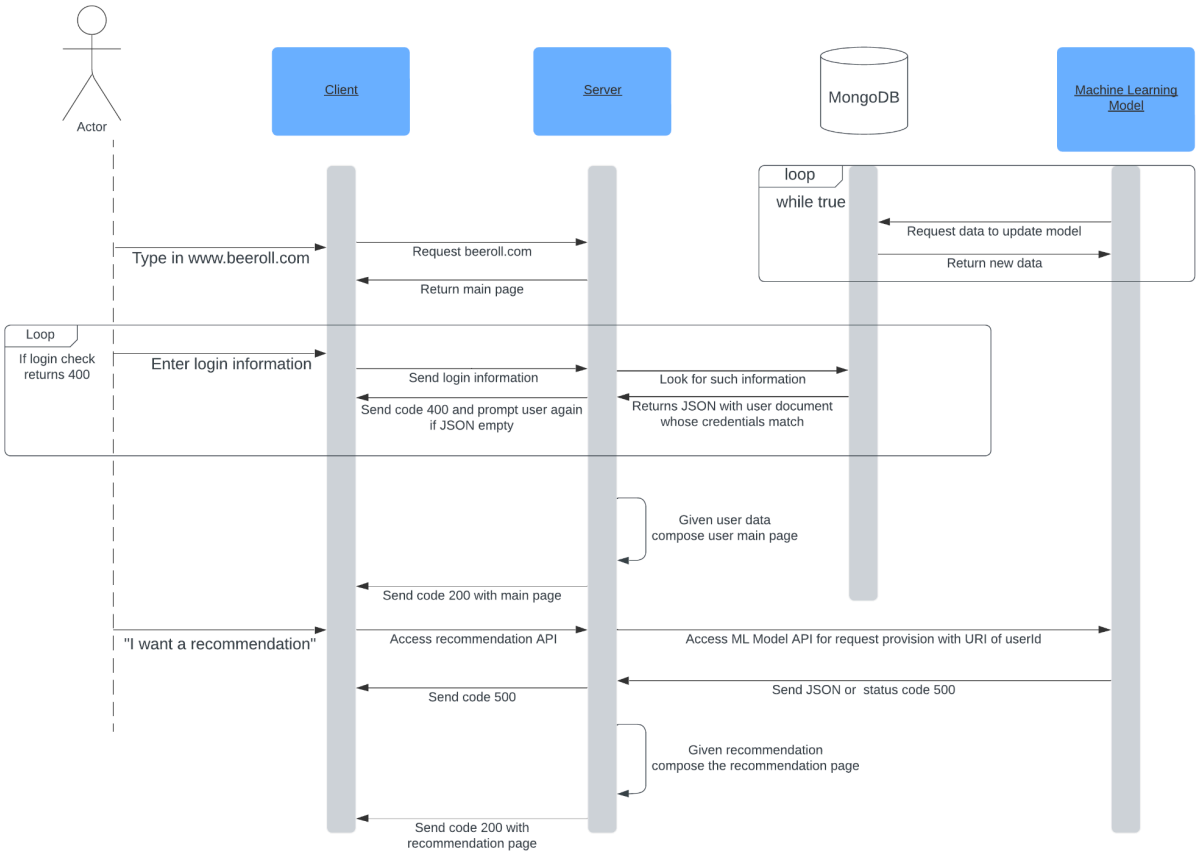


High-Level Overview

The Bee Roll project is a web application where different users can converse about movies and get recommendations based on their preferences. The web application will use a client-server model to handle requests from users and respond with the needed information using a carefully engineered system of backend APIs. The Server will accept the clients requests, access the database for the information, and send requests to the Machine Learning Model to get recommendations for a client if needed and respond accordingly.

1. Client
    a. Provides an interface for the user to interact with the app
    b. Sends API requests to the server
    c. Receives JSON responses from the Server and changes the UI accordingly
2. Sever
    a. Handles multiple API requests from client
    b. Validates requests and sends queries to the database to retrieve and modify the data, and requests to the ML model to get predictions for the client request.
3. Machine Learning Model
    a. Handles multiple requests from the server
    b. Based on the internal algorithm and the data from the database the model sends a prediction to the server
4. Database
    a. Server will implement a series of APIs to implement CRUD operations on MongoDB. Given that we do not want to lose any data, MongoDB is also ACID compliant on a single document. Backend is going to account for that and will implement it by using a two-phase commit.

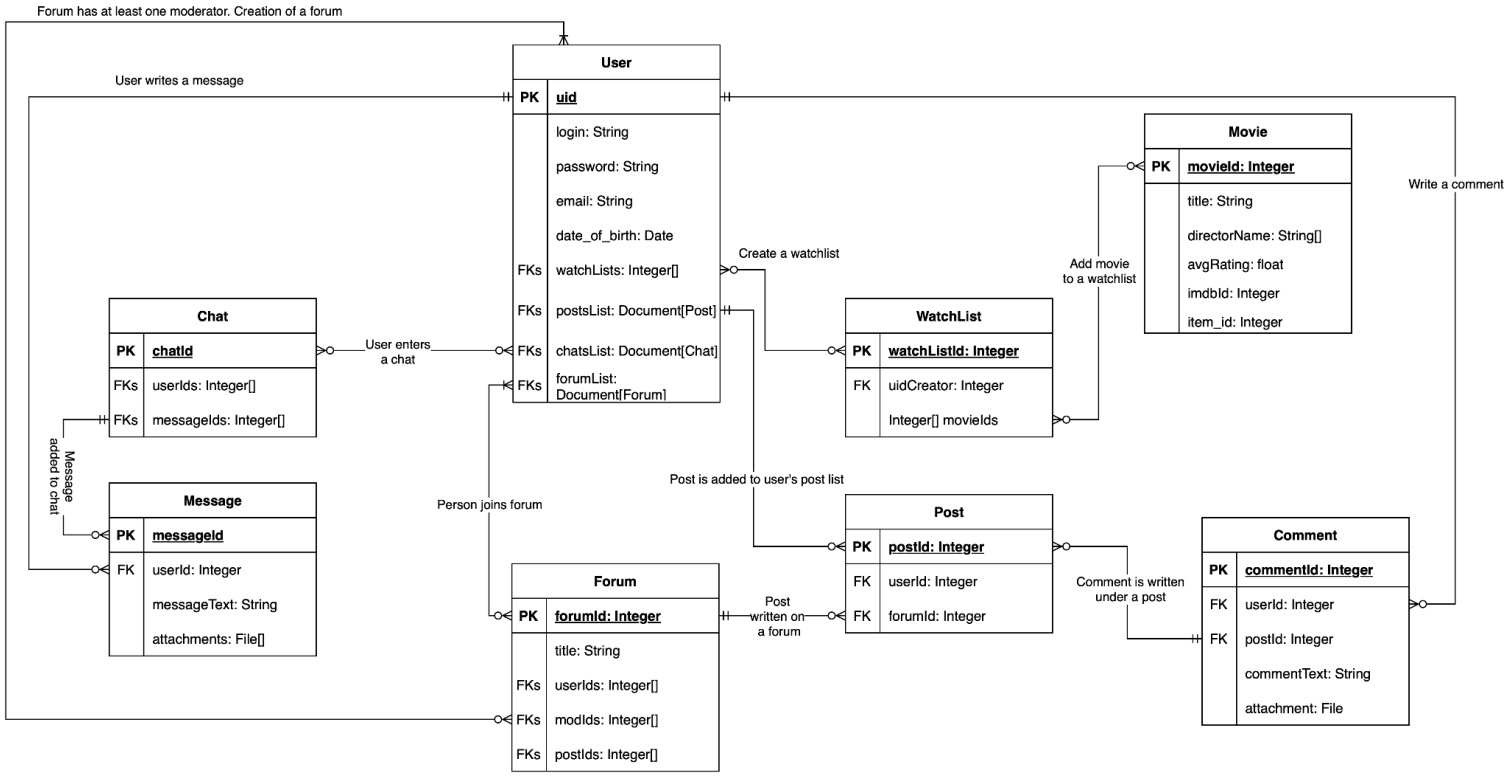
Sequence of Events Overview



Most of our operations are basic sequences of CRUD operations. Basic overview of operations is as follows: client will make request to server and server will handle one or several API calls to a database. After that, it will fill an according ReactJS page and return it. To avoid redundancy, only a few operations are shown.

Design Details

Data Classes Diagram



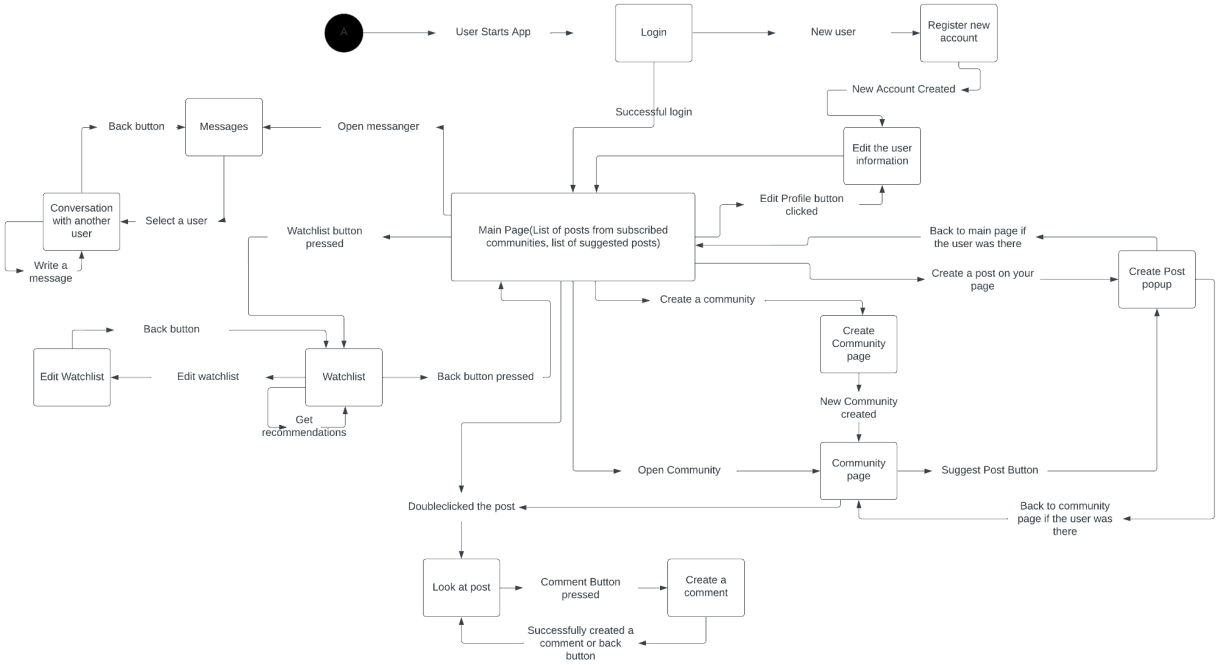
Data Classes Description

Given that the following Data Classes will be implemented through MongoDB, not Primary or Foreign keys have the possibility to be nulled.

- User Class
    - The User Class describes any user of our system. The user class has login and password fields, so that they could login into their account, email, so that they could receive notifications about their updates and date of birth, so that we can approximate their preferences based on age group.
    - The User Class has connections to the following classes:
        i. Watch List Class since every user on our platform has the ability to create a “to watch list”.
        ii. Post Class since every user can write a post either on their own page or on a community page
        iii. Comment Class since every user can comment on other’s posts
        iv. Chat Class since our platform allows the creation of chats and messaging
        v. Message Class since every user on our platform can send messages into chats they are a part of
        vi. Forum Class since every user can create forums and be a part of one forum.
- Watch List Class
    - The Watch List class consists of a user that it belongs to and a list of movies that the watchlist consists of.
    - The Watch List has connections to the following classes:
        i. User Class since the watch list is owned by one person
        ii. Movie Class since the watch list consists of a list of movies which the person wants to watch
- Movie Class
    - The Movie class consists of the movieId, so that ML model can identify it and predict different recommendations and a some information about the movie like the title, the name of the director, the list of film genres, etc
    - The Movie class contains information about title, directorNames storing all directors’ names, avgRating from MovieLens, imdbId that refers to IMDB’s Movie ID, and itemId used for internal purposes.
    - The Movie class has connections to the following classes:
        i. Watch list class since the watchlist consists of movies a user wants to watch

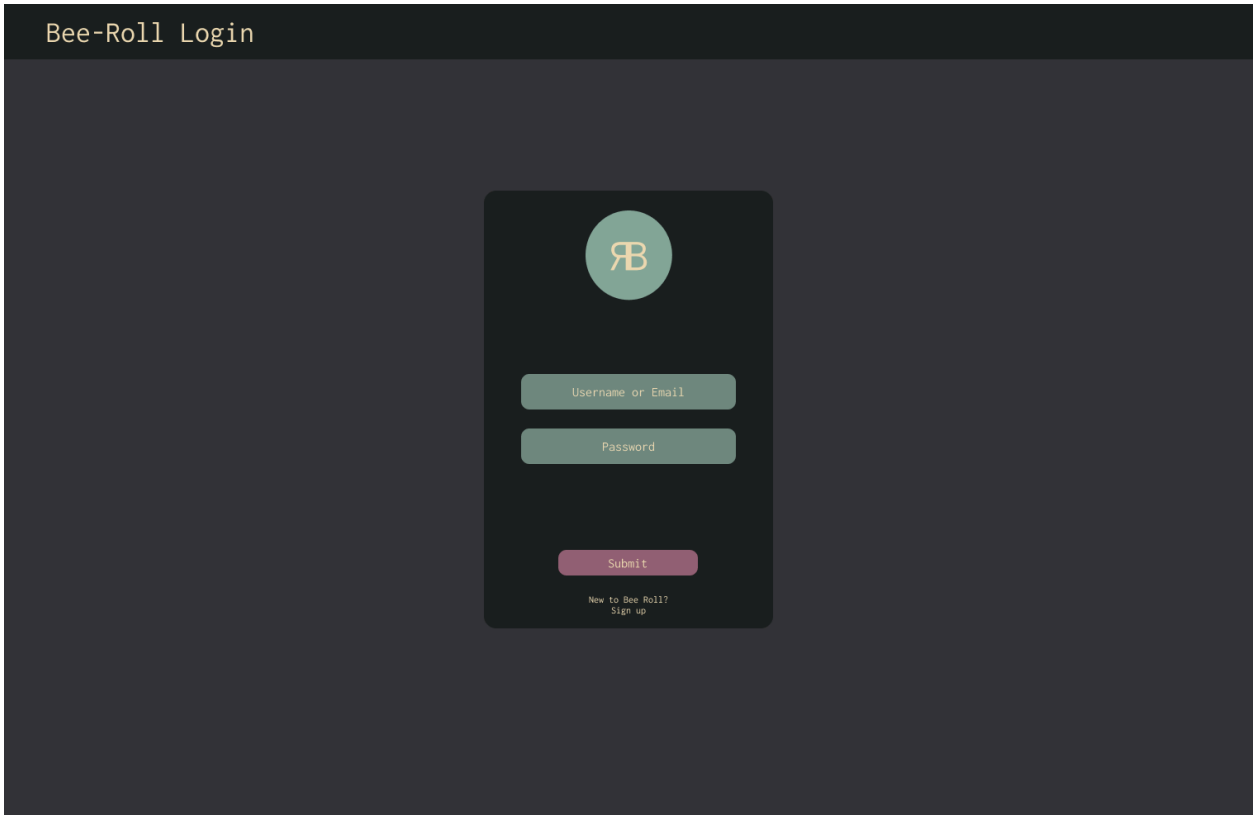
- Post Class
    - The Post class consists of a link to the user, and a link to the forum also it has a text field and a multimedia field, since the user can add pictures and other files to the post
    - The Post class has a connections to the following classes:
        i. User Class since every post has an owner
        ii. Forum Class since every post belongs to some forum (The user's page is also considered a forum)
- Comment Class
    - The Comment class consists of a link to the user and a link to the post. Also it has a text field.
    - The Comment class has a connections to the following classes:
        i. User Class since every post has an owner
        ii. Post Class since every comment is related to one of the posts
- Chat Class
    - The Chat class consists of a name, links to multiple users and links to multiple messages.
    - The Chat class has connections to the following classes:
        i. User Class since every chat has people in them which can text to each other
        ii. Message Class since every chat has messages that are in them
- Message Class
    - The Message class consists of a text field and a link to a chat and a link to a user.
    - The Message class has connections to the following classes:
        i. User Class since every message has the person who sent it
        ii. Forum Class since every message belongs to some chat
- Forum Class
    - The Forum class consists of the name of the forum, a list of moderators and a list of subscribers to the forum, and a list of posts on the forum.
    - The Forum class has connections to the following classes:
        i. User Class since all of the moderators and subscribers are users of the platform
        ii. Post Class since a user can post on the said forum

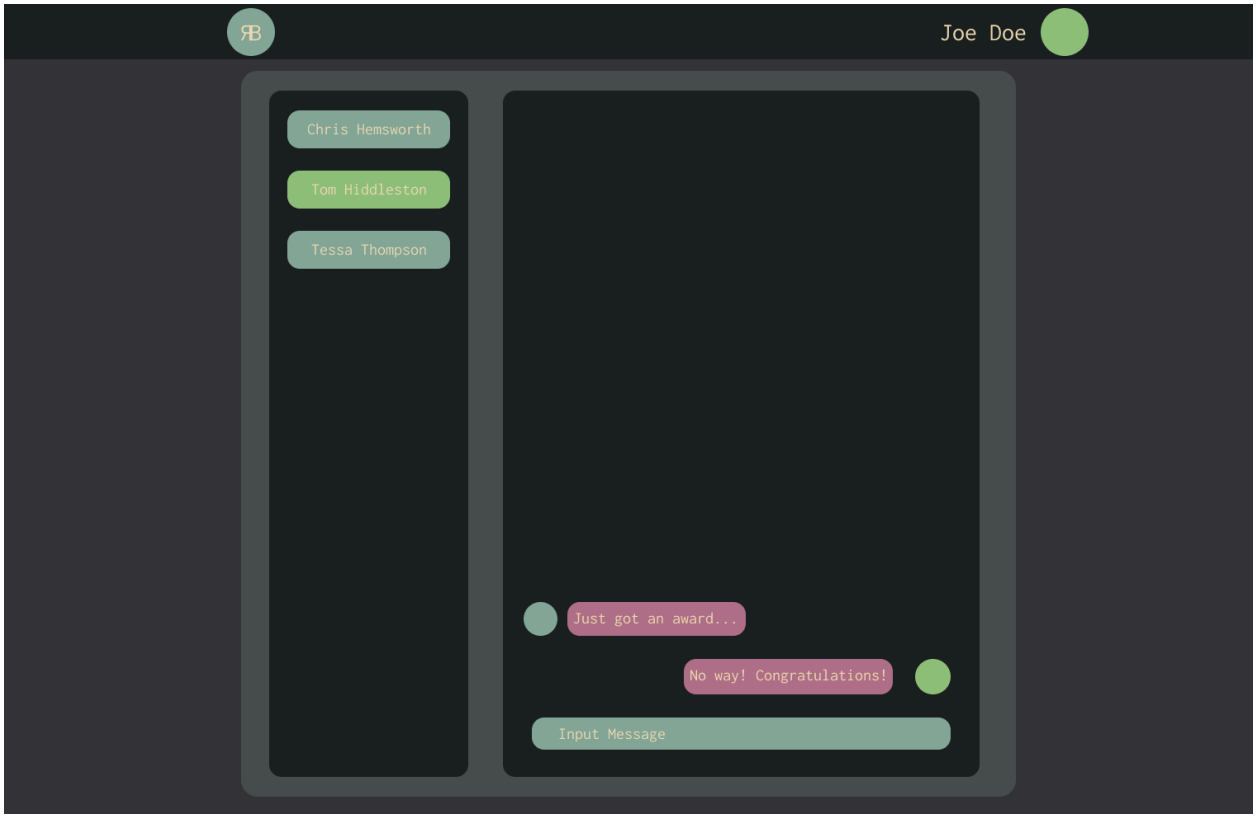
Navigation Flow Map

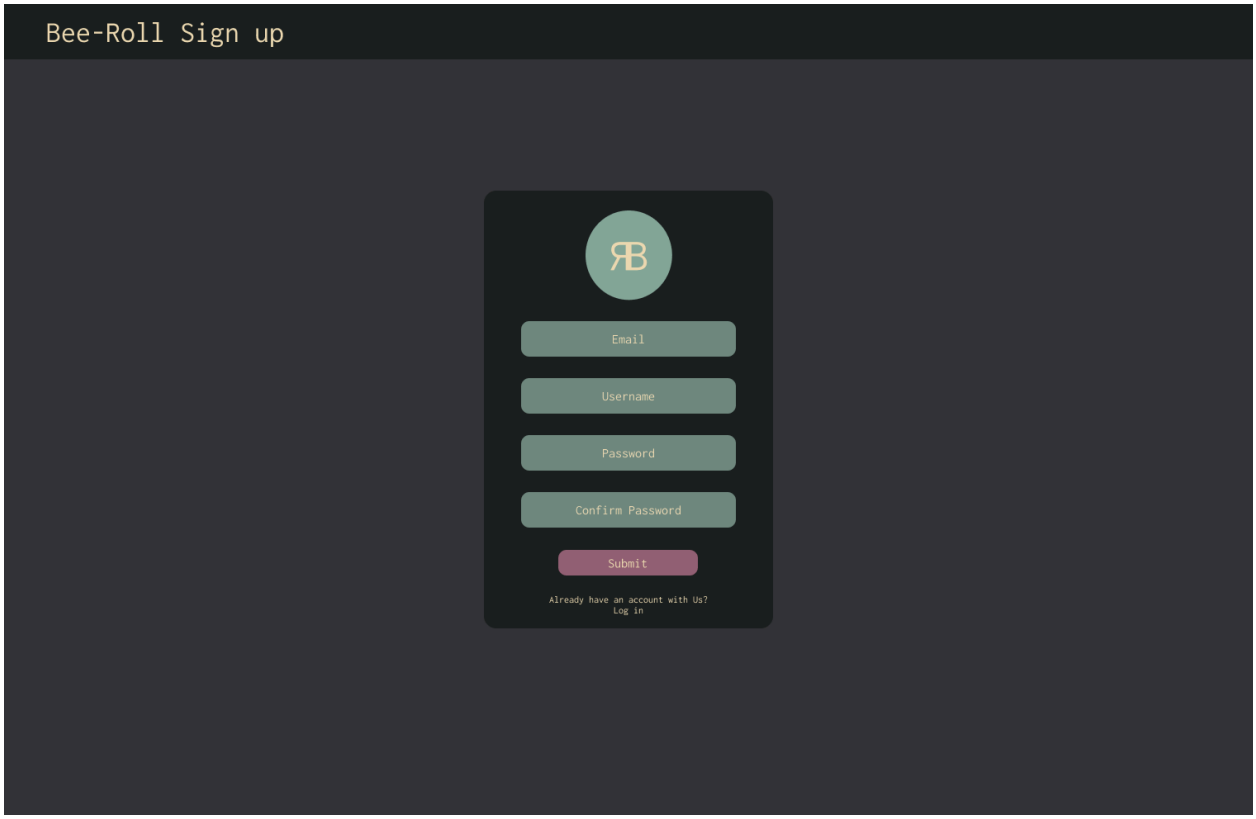


UI Mockup


Joe Doe
Anything to share?
Robert Downey Jr:
Just watched “The Avengers”, the movie was Fire! Can’t wait to see the sequel!
Mark Ruffalo:100%! I give this movie 10/10!
Andy Serkis: Nah, the movie is overrated tbh.
Quentin Tarantino:
The following post may contains spoilers about a movie
“The Hateful Eight”
Do you wish to view the post regardless?
Yes
No


Bee-Roll Login
Username or Email
Password
Submit
New to Bee Roll?
Sign up


Joe Doe
Chris Hemsworth
Tom Hiddleston
Tessa Thompson
Just got an award...
No way! Congratulations!
Input Message


Bee-Roll Sign up
Email
Username
Password
Confirm Password
Submit
Already have an account with Us?
Log in