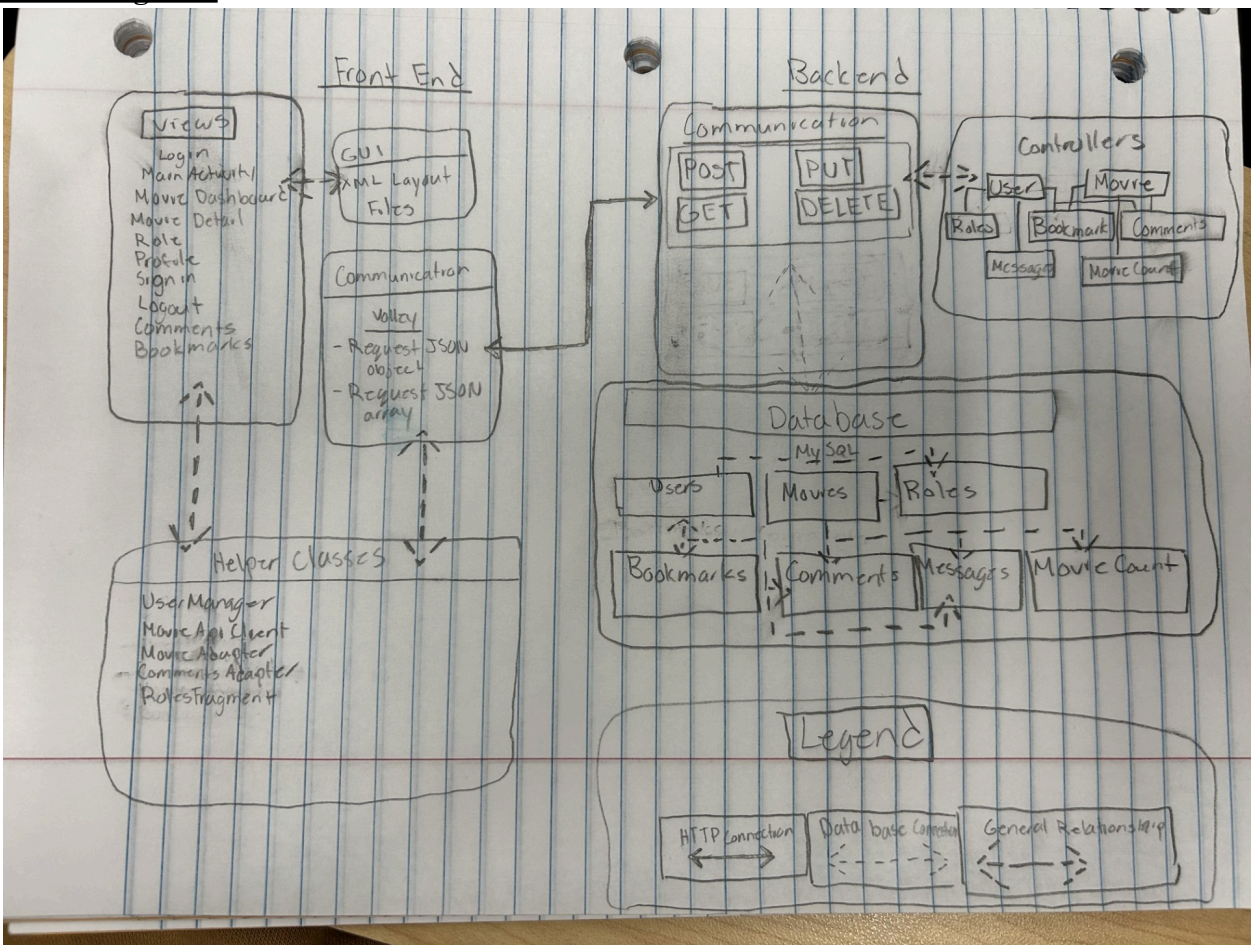
Design Document for Movie Finder

Group 2_Fatem_3
Westin Gjervold: 25% contribution
Nathan Krieger: 25% contribution
Jake Nickel: 25% contribution
Ryan Martin: 25% contribution

Block Diagram:



Frontend (currently implemented)

SignUp/Login (User)

- Sign up generates a page with the following elements
    - EditText: Username
    - EditText: Password
    - EditText: ConfirmPassword
    - Button: SignUp
    - Button: LogIn
- Login has the same elements as Sign up, just not a confirm password button
- If all input validations are passed, username and password are sent in a JSON body to create a new user in the DB. After this, the user is navigated to the home page.
- The login button navigates users to the login page. The login page simply takes the username and password, verifies that it matches the DB row, and navigates users to the home screen

MovieDashboard/MovieDetail (Movie)

- Both of these pages are designed to view movie(s). The dashboard displays a list of trending movies. The detail view displays a single movie in-depth by displaying several TextView components such as Poster, View Count (which is a WebSocket to increment view count in DB), Description, Release Date
- The detail view displays comments as well. A comment can be created and liked from this view.
- A user must be logged in to view this information
- A user can bookmark a movie to add it to their bookmarks list. Bookmarks can be viewed from the account page, where you can also DM other users using WebSockets and view profile information
-

Backend Communication

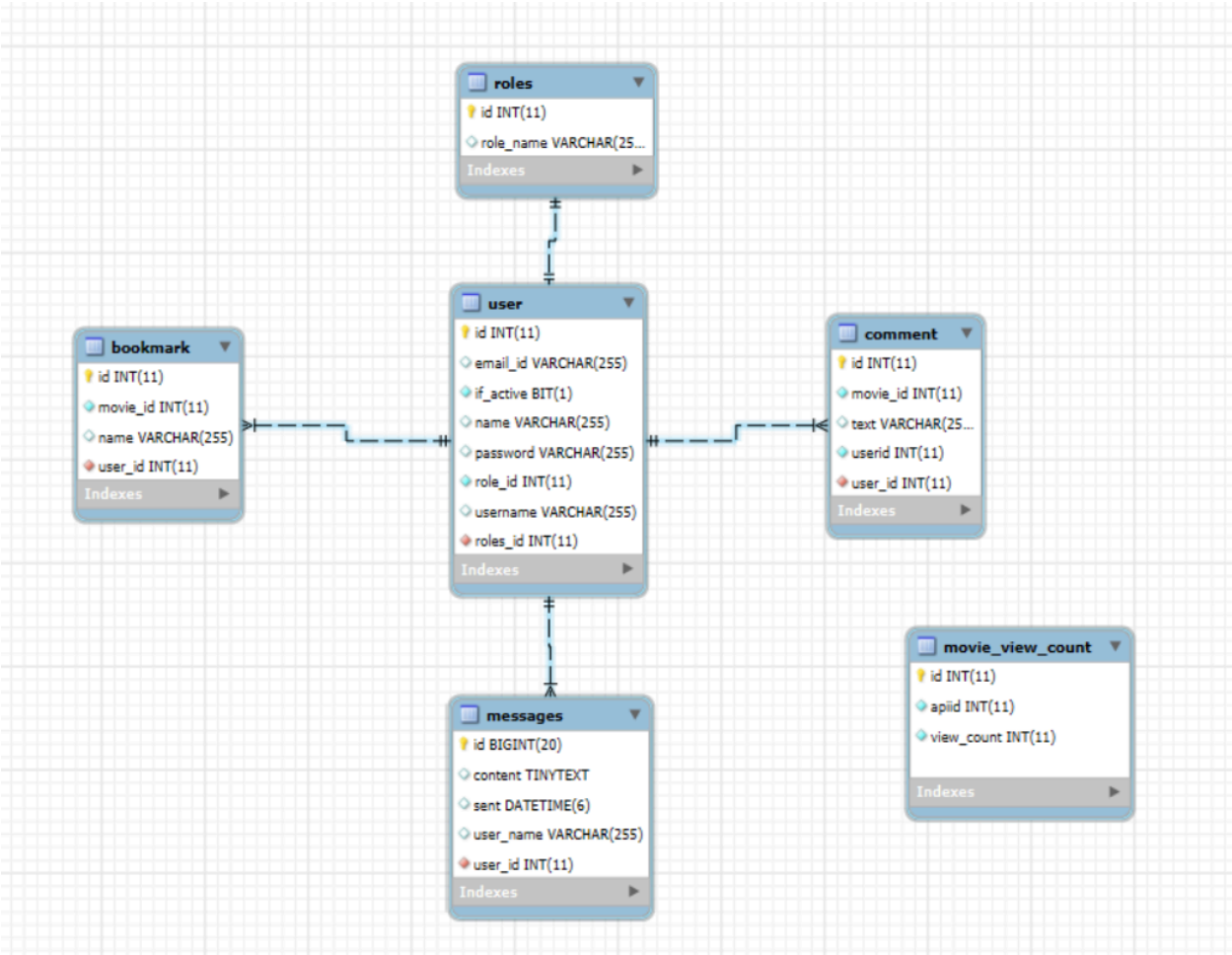
The backend of the movie app uses RESTful mappings to interact with the database based on information sent to the specified endpoints. The primary HTTP methods include:

- **Post**: Sends information to add a new item to the database, such as a new user, comment, or movie view count entry.
- **Get**: Requests information from the database, often using an identifier (like movieId or userId) to fetch specific details such as a list of comments for a movie or the view count of a particular movie.
- **Put**: Sends information to update existing records in the database, such as modifying a user's details or updating a movie's view count.
- **Delete**: Sends an identifier to delete a specific record from the database, such as removing a user's comment.

Controllers

The controllers manage the endpoints that facilitate communication between the frontend and the database. These include:

- **User Controller**: Handles user-related operations such as user creation, login, and profile updates. It maintains one-to-one relationships with the Comment, Bookmark, and Role entities.
- **Comment Controller**: Manages user comments on movies. Each comment is linked to a specific user and movie using userId and movieId. It includes mappings for creating new comments and retrieving comments by user or movie.
- **Movie Views Controller**: Tracks real-time movie view counts using WebSocket connections. It updates the view count in the database when a user views a specific movie and provides live updates to multiple clients.
- **Bookmark Controller**: Allows users to bookmark their favorite movies. It manages a one-to-many relationship where users can have multiple bookmarks, but each bookmark links to one specific movie.
- **Role Controller**: Manages user roles, such as regular users or admins, defining different levels of access and permissions within the app.
- **Messages Controller:** Manages user messages on the Admin DM Websocket feature. Each message is linked to the sender of each message. Stores each message, the sender's userId, and who the message was sent to (broadcasted or particular user).
- **Movies Controller**: Manages the movies and their associated information from the api, allowing for access to lists of popular movies, as well as numerous functions to get certain specific information for each movie, such as all of it’s details, the title, release date, rating, etc.



Movie_view_count shares a one-one relationship with TheMovieDatabase API
https://developer.themoviedb.org/reference/intro/getting-started