CS5200 Project Final Report

Team name: LeeCYunWWuH
Team member: Chih-Tao Lee, Hsin-Yen Wu, Wan-Ting Yun

README

1. Unzip the zip file
2. Dump the data from movie_mania.sql file
3. Use terminal to get to server folder run “npm install” (If you clone the project)
4. Create a .env file at the root level of server folder
5. Inside the .env file type “MYSQLPASSWORD=your_password”. This password should be the password you use to login to the MySQL workbench. Remember to save the file.
6. Check the index.js file in server folder. Make sure host, user, and database are correct in the mysql.creatPool funciton
7. Use terminal to get to client folder run “npm install” (If you clone the project)
8. Go to server folder and run “node index.js” or “nodemon index.js” to start the back-end server.
9. Open another terminal and go to client folder. Run “npm run start” and then you can use the web browser with “http://localhost:3000” to use the application.

*** You might encounter the error like this when our app makes its first attempt to your MySQL database -> (Access denied for user 'root@localhost' (using password:NO)
You can fix the error by running these two line in your MySQL workbench

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your-chosen-password';

FLUSH PRIVILEGES;

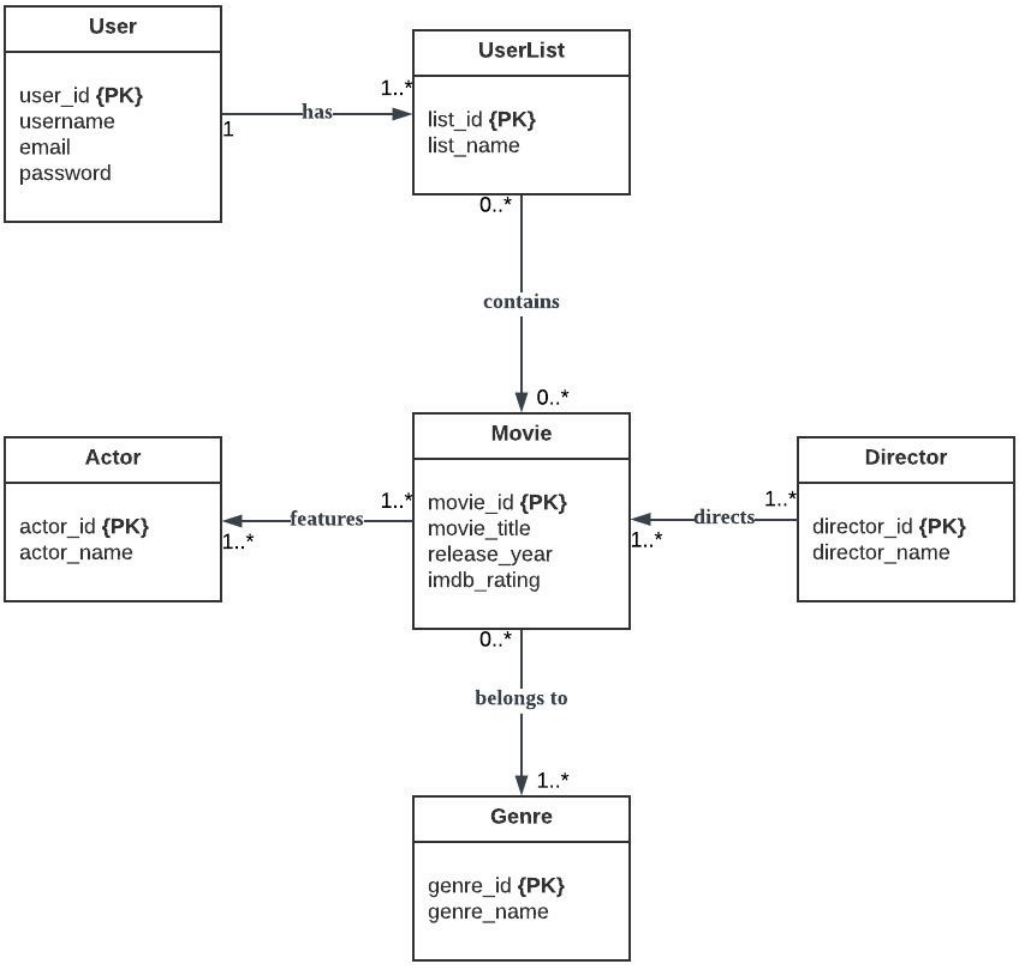
Remember to restart the server if it gets shut down by any errors, so the app can run properly.

https://github.com/Jazzcort/ff_project

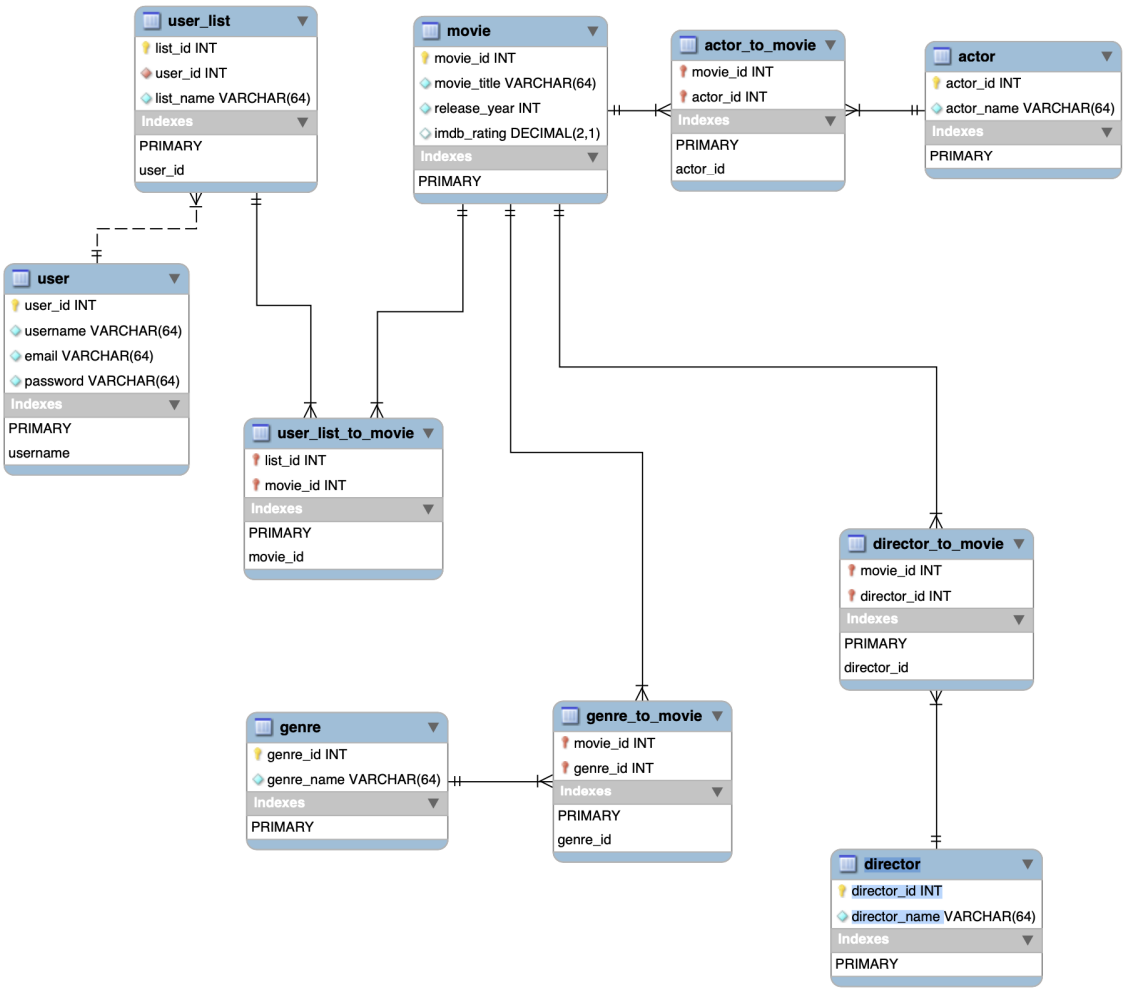
Technical Specifications

We utilized MySQL Workbench for designing the database structure and importing the initial dataset. Our website interface is accessible through web browsers. The back-end architecture is developed using Express.js in JavaScript, which facilitates a RESTful API for executing Create, Read, Update, and Delete (CRUD) operations. For the front-end development, React.js was employed to craft the design and functionality of our website.

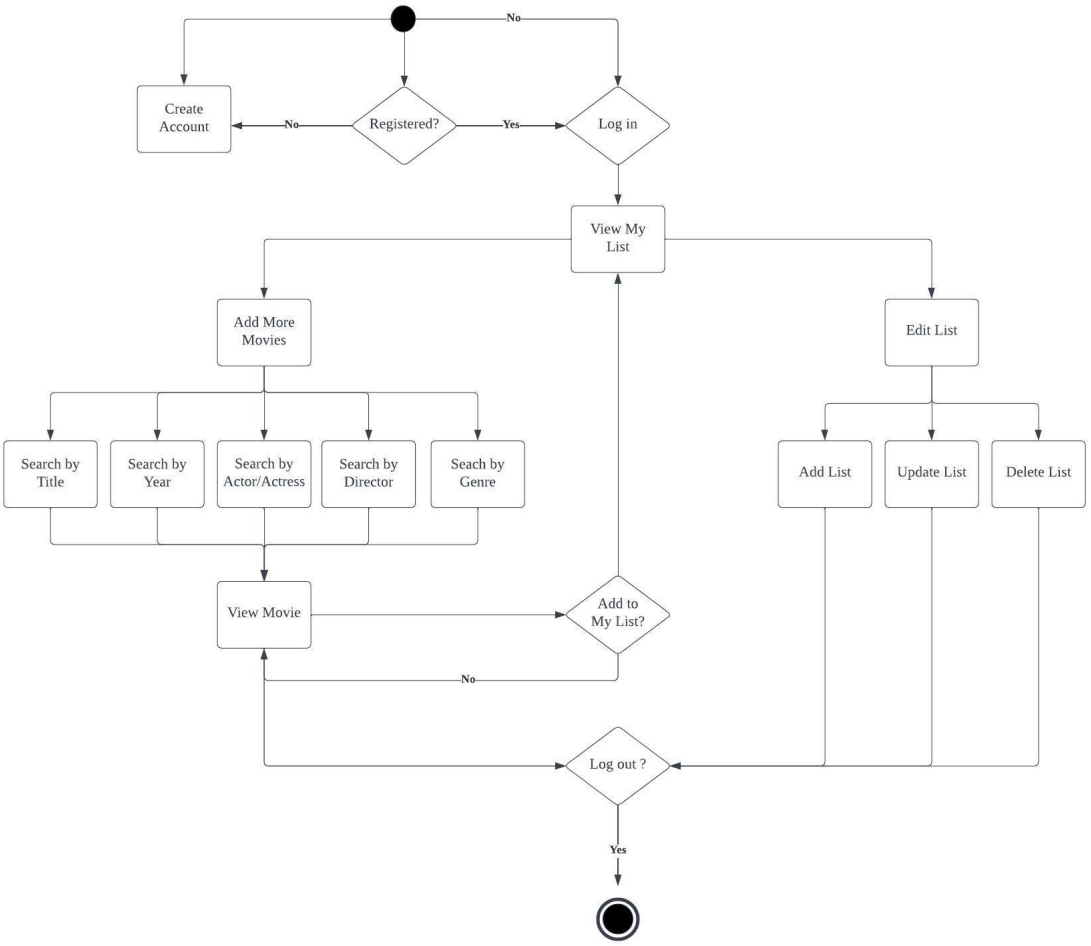
Conceptual Design (UML)



Logical Design



Final User Flow



Lesson Learned

1. **Security: Unhashed Passwords**

Storing passwords without hashing exposed our system to security risks. Future applications will implement robust hashing for enhanced security.

2. **Data Collection: Automation Over Manual Entry**

Manual data entry proved inefficient and error-prone. We should consider developing automated scripts for data parsing to improve accuracy and efficiency.

3. **Code Issue: React.js Anomaly**

A persistent issue with React.js caused discrepancies between the UI and database data. Emphasizing thorough debugging and testing in future projects is essential to prevent such mismatches.

Future Work

1. **Online Server Transition**

We plan to move from a local to an online server to improve accessibility and scalability.

2. **Advanced Security**

Enhanced security measures will be implemented, focusing on stronger encryption and secure authentication.

3. **Social Features**

New features will enable users to connect, share, and interact with each other's movie lists.

4. **List Sharing and Downloading**

Users will gain the ability to download and share their movie lists, enhancing user experience.

5. **Database Expansion**

We aim to expand our movie database, introducing automated scripts for regular data updates.