Just Another Movie: Project Update 1

# Group Members:

**Emmett Storey, Robert Jones, Hayden Richards, Kaitlyn Hardin**

# Purpose

Just Another Movie is an application that uses AI to accurately recommend movies based on the user’s search criteria and user ratings.

# Updates

### UI Design/Frontend Development

The User Interface has changed in many ways. We have come across challenges with designing our application. For example, the preparation and concept took a little over a week, and the actual design took an additional week. Robert is unfamiliar with designing an application, and all the information had to be searched by either Google or GitHub. I have been able to successfully create and implement the UI buttons, such as next, previous and exit.  They all have a unique role in this application. The exit button allows the user to exit from any screen of the application if they wish to close the page. The previous button allows the user to return to the last page that they have made progress on without losing the data that has been recorded. The next button allows the user to progress forward within multiple screens. As you proceed to the search screen the user is allowed to enter their search criteria. This information will then be populated on the Results page when the user clicks the “Get List” button that’s located at the bottom of the page. The information will then be printed to the Results page.

While making progress with the design of the User Interface, there have been some challenges such as Robert having corrupted files. This unexpected issue caused Robert to restart writing his code for the application. As a team we have decided to allow the user to input their criteria and to hold off on the drop-down box unless time permits.  Furthermore, we need to discuss as a team the color scheme and the design of the pages.

Backend            
Most of the classes and their functions have been implemented with the exception of a few functions. Class wise all the classes have been implemented along with most of their functions being completely implemented if not partially. However, the backend functions and code has changed significantly. The query class has been made to send queries to the database replacing the AI class. The AI class has been scrapped. One of the Error Handling classes functions allInputEmpty() has been scrapped. An addMovie() method has been added to the MovieList class and remMovFrmList() has been changed to removeMovie().

Most of what needs to be implemented are unit tests for the Movielist and Query classes. Most of these tests would involve the Query class and making sure it can access the data on the server and search through it appropriately. One of the primary challenges we face is that due to Emmett’s lack of experience with databases and SQL and he is having to teach himself while focusing on implementing the Query class and other interactions within the server. Another problem will be debugging the searchQuery function. Emmett’s lack of experience with SQL will no doubt impede on his ability to understand defects and bugs in the code which will no doubt result in a longer debugging process than normal.

### AI

The design for the AI has changed in a few different ways, and some problems have come up that were unseen. First, the AI was originally supposed to be run from a web server that the application would connect to and send the user input to send to the AI for it to generate recommendations. But the team had some difficulty attempting to implement that, so it was dropped from the design. Now, the AI will be put into the application directly. The next problem that we are facing is that the dataset we are using has two different sizes of user data with the rating. Hayden was planning on using the larger one as the training data set and the smaller data set as the test training set. But the owner(s) that made the dataset for some reason randomized the smaller dataset, so the user ids are different between the two datasets. To solve this problem, we are going to have to take the larger dataset and take roughly 20% from the dataset and use that as the testing data. The machine learning library we are using includes a function that allows us to split the data into testing and training data sets. However, while implementing this function Hayden keeps getting an error because the machine learning algorithm wants all of the data in a IDataView interface. When loading the data and using a function to convert data from csv to a IDataView, it keeps throwing an error saying that the data isn’t being formatted into IDataView.

The next problem we faced is the cold start user problem. The cold start user problem is when a new user is made but you have no previous knowledge of their likes and dislikes. Because of this the AI won’t recommend any movies. There are two different ways to fix this. First, create a short survey the user must fill out when they first open the application and use the information the user provided to base the recommendations off of. Second, take the genre that the user enters and look through the database to find the user that has reviewed the most movies of the genre that the user entered and recommend the user the same movies that would be recommended to the user that reviewed the most movies of that genre. In theory this should work because assuming the person that has reviewed the most movies of a certain genre should have developed the best taste in movies of that genre. This combined with the backend which will filter out more movies based on the user input should provide us with movie recommendations that the user will like.

Even with these challenges I have gotten the AI to produce movie recommendations only using a very small testing set which provided us with a r-square value only .4. For this project we are going to try and achieve a r-squared value at least a .5 but hopefully more. There are only two aspects of the AI that still needs to be developed: saving the AI to a .zip file so that we can load it into the application and using the .zip file to make recommendations. Also, we need to implement a way to change the user that is being chosen to base the recommendations off of. Everything else has been implemented like building and training the model and evaluating the output.           

### Database

The database is hosted on MS SQL Server provided by APSU CIST Department. It was successfully set up and is managed by Kaitlyn Hardin. There are no current issues or any expected issues with hosting the database.

The design of the database has changed significantly since the project proposal and design. The relationship between the tables in the database is different. Instead of two tables in the database, there are six tables. The field data types in each table are updated to adequately reflect the expected value for each column. Moreover, the database will not connect to a web server, but it will directly connect to the application. After many trials with hosting a server, a significant amount of time has passed, and no solution has been found. Therefore, it is unlikely to complete the project following the previous system design. As a result, a connection will be made to the database.

The database tables are being continually added to the database. Two tables out of the six tables have been added to the database. One major issue with adding the rest of the tables is the extraction of data from the source dataset. The way the owner of the source dataset stored the information is unstructured and the format of the data is unique. This makes extraction difficult to automate. Luckily, we have found a technique to extract the data in our necessary format. Now, the technique needs to be implemented and applied to the various csv files in the source dataset to create the rest of the database tables.

The connection to the database has not been implemented but needs to be created promptly. Currently, we have plans to create the database connection. This is a critical objective. We expect unknown challenges pertaining to connecting to the database. For this reason, connecting to the database is a top priority.

### Diagram Description automatically generatedUpdateD Backend Class Diagram

### Updated AI Design

Diagram

Description automatically generated

### updated erd diagram

### Diagram Description automatically generated

### System Design Diagram

System Diagram

Description automatically generated