

COMP2406 Movie Database Project Report

Name: James Yu

Student ID: 101057704

OpenStack Instance Information

Public IP Address: 134.117.132.194

Username: student

Password: 101057704

[Introduction and summary of implemented/non-implemented functionalities.](#)

All information needed to start the database, as well as instructions can be found in the README.txt. I will instead be discussing about my design choices/decisions in this report. To save you some time, I will first be writing about requirements that I had not met for this project. Starting off with the user accounts, I believe the only requirement that I missed was the recommendations feature. Unfortunately, it broke my project, so I had to remove the feature entirely for the final submission. Moving on to the viewing movies, I believe that I managed to do everything that was required except for fact that the ability to review movies is very limited and basic. As for the viewing people portion, I believe that it should be fine in terms of requirements. I had to change a couple of things last minute, but it should not be an issue (hopefully.... finger's crossed....). Viewing other users should be completed, aside from notifications. Full disclosure, the only notifications I have implemented are PUG error pages that will notify you of errors such as invalid search terms (like if you attempt to search for something that does not exist, or the combination of searched terms netted zero results back).

The one of the design decisions that I am the proudest of is my implementation of the movie search engine. Although it was not an extremely difficult or impressive, it took the most amount of time

and I am happy with how it ended up. The search engine takes up to four types of search terms: the title, the genre, the year, and the minimum rating. For the rating system, I used the IMDB ratings as the metric because they were the most consistent in terms of which movies had an IMBD rating, and they were easy to use in calculations thanks to JavaScript's inherent typecasting. The movie searching functionality should be able to handle whatever you throw at it. I created a few base cases to deal with unexpected circumstances. The movie search function is also able to handle searching through the URL and should not crash. due to the error base cases that I had prepared. One of the requirements for the viewing movies portion was to be able to use genre keywords to find similar movies (I am assumed they meant by using genre keywords as search terms) and so I took it one step further. If you successfully search a movie, the genre keywords of that movie will be hyperlinked, and it will allow the user to find similar movies based on genre all at the click of a button. Although, I suppose that is exactly what the internet is.

Some of the design decisions that I made to increase the overall quality and scalability of my project was using subtle use of hyperlinks in the header partial. It is not a revolutionary quality of life design choice, but it certainly made my life a lot easier whenever I was bug testing or limit testing the search system. Another design decision that I made was to lock away 90% of the functionality of the project if the sessions user was not logged in or registering an account. I made this decision, because I believed that it would be important to plan in terms of a hypothetical security issue.

Overall, I am not very satisfied with this project. There were missing requirements and a lot of functionality that I had originally planned to add to the project. Unfortunately, I did not have the time to be able to work on the project the way that I had wanted to, due to conflicted due dates with three other term projects. If I had the time, I would have loved to have made a chat/messaging system between active users or a feature that will\ bring the user to a randomized page. I thought that a randomized search feature would be a nice design decision if the user ever wanted to branch out in terms of movies genres.

For the final submission, I did not use any special modules, but I had the opportunity to test out a couple of new modules while working on the project such as JSDOM and other smaller modules to create

a popup notification window. I had spent the better half of a month attempting to create a popup window in node.js, but I eventually gave up because it was too time consuming for a miniscule feature.

In conclusion, I had a great learning experience because this project had allowed me to search for modules that I had never even heard of before.