Final Report

README:

```
CONTENTS OF THIS FILE
---------------------

  * Introduction
  * Requirements
  * Installation
  * Configuration



INTRODUCTION
------------

The purpose of the project is to offer users a movie searching system. The system enables users to look up the information of movies, to get recommendations, and to visualize the differences among movies.




REQUIREMENTS
------------

This project requires the following software:

  * MySQL Database Server (https://www.mysql.com/)
  * Python 3.8.5 (https: //www.python.org/downloads/)
  * Python library pymysql (https: //pypi.org/project/PyMySQL/)
  * Python library numpy (https: //pypi.org/project/numpy/)
  * Python library matplotlib.pyplot (https: //matplotlib.org/stable/users/installing.html)

This project requires the following dataset:
  * IMDb movies extensive dataset (https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset?select=IMDb+movies.csv)
  * Rotten Tomatoes movies and critic reviews dataset (https://www.kaggle.com/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset?select=rotten_tomatoes_movies.csv)
  * Movies on Netflix, Prime Video, Hulu and Disney+ (https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney)



INSTALLATION
------------

  * Install MySQL service and MySQL Workbench. Visit
```

https: //www.mysql.com/downloads/.

* Install python libraries (pymysql, numpy, matplotlib.pyplot, math). Visit
https: https://pypi.org/.

* You may want to install data science platform such as Anaconda to run. python code. Visit: https: //www.anaconda.com/products/individual.

CONFIGURATION

----

* The users need to log into their account with a username and password. They. will either be accepted or rejected. If rejected, they will be reminded to go back to the login process to try again. If accepted, continue:

- The users can filter movies by multiple attributes to receive the detailed information about the movie.

- The users can update or delete movies from their favorite list.

- The users can input people (actors, directors, producers) to get their biography information.

- The user can input movie titles to see on which platform the movie is available.

- The users can input movie titles to receive critics of the movies.

- The user can input a list of movies to receive a list of recommended. movies that are highly similar to the input movies.

- The user can input some different movies to get visualizations that shows the comparisons of these movies.

Technical specification:

Overview:

Generally speaking, this project is a movie searching system. The system enables users to query for detailed movie information and to filter movies by multiple attributes. The system also has a precise film recommendation function to recommend other movies based on the movies' attributes and users' favorite movie list. In addition, the project enables users to generate visualization that show the properties of movies.

Goal:

1) Users' information storage function

2) Query function

3) Recommendation function

4) Visualization function

Product requirement:

1) Users' information is supposed to be able to update and delete.

2) Query function requires accurate output with understandable data structure and should be able to handle the occurrences of errors.

3) Recommendation function needs to output accurate results based on scientific calculation method.

4) The visualization chart needs to choose the appropriate chart type so that the data and the comparison between them can be displayed to the user intuitively.

Out of scope:

This project does not contain an interface. All functions are implemented in Python and MySQL workbench.

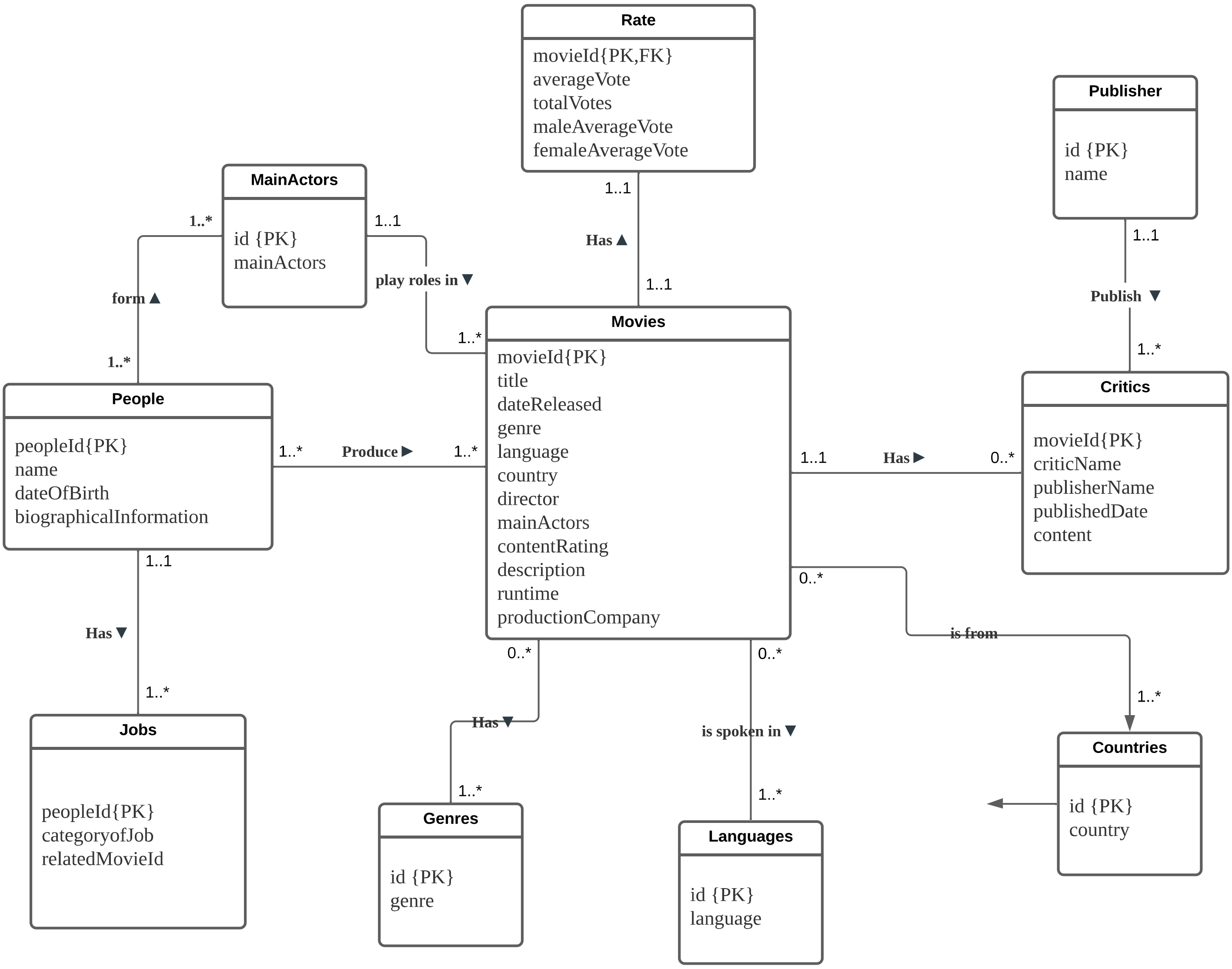
Approach:

I load all databases and create procedures and functions for queries on MySQL workbench. Then I connect my database with Python. By importing pymysql library, I am able to call procedures and functions on Python console. By recording the output of procedures and functions, I build the recommendation function and visualization function.

Measuring Impact:

I tested my function from the perspective of a user. I check whether my users' information storage function would avoid duplicates or not. I check whether my recommendation function works well. I keep testing and modify my codes until they work better.

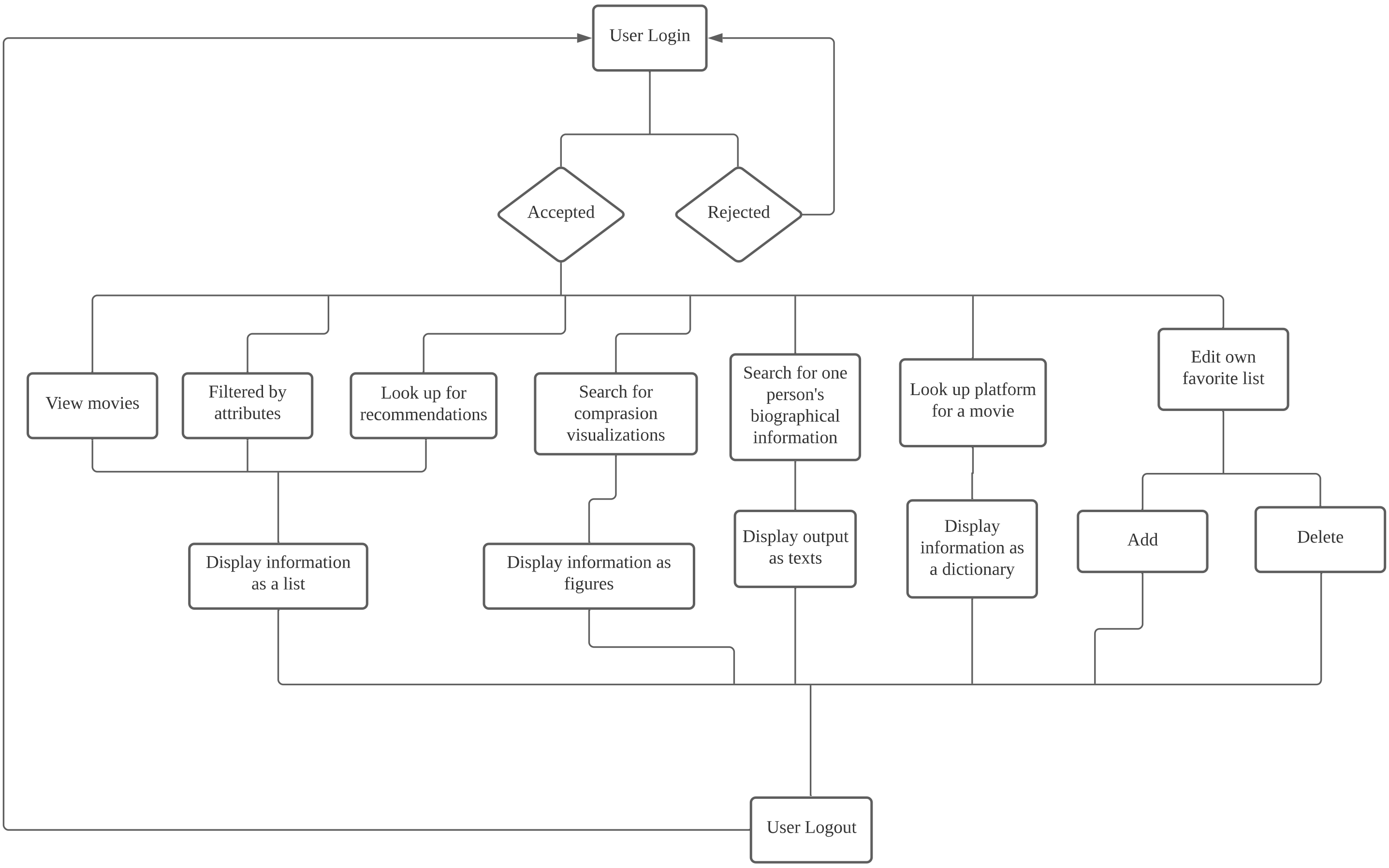
Conceptual Design:



Logical Design:

Logical Design MWB file is included in the zip file.

Flowchart:


User Login
Accepted
Rejected
View movies
Filtered by attributes
Look up for recommendations
Search for comprasion visualizations
Search for one person's biographical information
Look up platform for a movie
Edit own favorite list
Display information as a list
Display information as figures
Display output as texts
Display information as a dictionary
Add
Delete
User Logout

Lesson Learned:

Technical expertise gained: I am getting better at coding with pymysql library. I am familiar with how to call sql procedure and function in Python. It would be better if the arguments and the procedure are recorded separately, which would be easy to modify and would reduce the occurrences of errors. Also, I learned that the argument should be recorded as a list otherwise the python system will recognize each letter as one single character.

Insights: When I was searching for dataset about movies, I realized it is actually not a big data domain, which make me easy to clean the data but brings the limitations to build diverse functions. Therefore, if users expect a multi-functional application, a big data domain would be the better choice. Besides, I gained many insights about time management. This project let me realize what a bad time manager I am. I realize that a big project needs to be planned early and finish every phase on time to save time for testing and modification.

Alternative design / approaches: I can make a web page to implement my project.

Code not working: I used to try to pull out a message with 'handle continue' code when query does not find data. It did not work. I spend almost one hour on modifying it, but it still failed. Finally, I use 'handle exits' instead of 'continue' then my procedure works. However, that 'continue' code still does not work.

Future work:

The project can be used as some movie webs. Except for normal filter function, people are able to get recommendations while they feel like watching some movies. There are still some potential functionalities that I can add into my project. For example. I can build a function that can return a pie plot that indicate the proportion of movies with different genres and other diverse visualizations. Also. I can build a bigger filter system which would enable users to add more limitations. I can build a movie community function to enable users to form their own communities and update their discussion into that community.