Mini Project 1
Design Doc

Small user guide

Our program is similar to a small streaming service app where we provide customers with the option to stream movies and editors with the option of adding new movies.

When you start the program, you are greeted with a login screen, wherein a customer or even an editor can log in. If the customer isn't present in the database, they have the option of creating a new account for the same or exit the program completely.

After this, the customer is greeted with the option of starting a session or logging out entirely if they wish to do so. If they start a session, they have the option of searching for movies with specific keywords(can be multiple, referencing title, cast member, role). They also have the option of following a cast member or starting a movie. After that, they have the option to end watching a movie. Thereafter, they can start watching a new movie or end the session and log out.

The editor, once logged in, can add a movie by providing a unique id and its details. They can also add certain cast members to a role in the movie and also have the option of introducing new cast members. They can also update the recommendations provided based on a monthly, annual, or all-time viewing count of the number of customers. They can also delete a pair of movies from the recommendations. They can also log out at any time.

A detailed description of the program

Login Screen:

Firstly, the user, which can either be the customer or the editor, is asked to input their userID and password. The password has been protected using the getpass library which hides the characters entered while entering the password. Then, using the functionalities of python and SQL, the program runs the credentials through the customer table, and consequently through the editor table if the credentials are not in the customer table. If the same is the case in the editor table as well, the user has been given the option of creating a new customer account which will subsequently be stored in the database so as to allow them to access the program in the future or just exit the program from the signup page. If the sign-in has been successful, the user is directed to the customer or the editor menu respectively. New customers are also directed to the customer menu.

Customer Menu:

In the customer menu, we used python's implementation of classes to break down the problem into smaller problems. With the same, we have implemented the following functionalities in the customer menu:

->Starting a Session: When logged in, the customer is firstly greeted with the choice of starting a session or logging out. If they choose to start a session, a unique session id is assigned to their customer id along with the current date and a time counter which starts and goes on till they end the session. The duration of the session, even after logging in, is NULL until the session has been initialized by the user.

->Searching for Movies: After initializing a session, the user gets to enter certain keywords, which can be anything ranging from a certain substring in the movie name to a character's name to the name of an actor who acted in the movie. After entering the keyword, which can also be multiple rather than only one, the user gets to choose between multiple movies relating to the keyword(s) entered. At most, the user is only allowed to view 5 matches, with the option of viewing the rest in segments of 5. We used a detailed SQL query along with python implementation of conditionals to segment the output in rows of 5. After viewing the output, the user has the option of selecting a movie and viewing the cast members. From that menu, they can either start following a certain cast member, start watching a movie or go back to the menu. If they decide to start watching a movie, using the time functions of python, we note the time and make sure that the viewing time doesn't exceed the duration of the movie.
->End watching a movie: While watching a movie, the customer has the option of stopping the movie at any time. Whenever this happens, using SQL queries, we determine whether the customer has watched at least half the movie, and if they have, we insert the data to the watched table.
->Ending the session: The user has been given the option of ending the session. If they choose this option, any ongoing movies will end and their duration would be recorded similarly to the end watching a movie functionality. Thereafter, the session will end and the duration which would have been calculated using the time counter initialized when the session started would be used to calculate the duration of the session and insert the details into the session table.
Editor Menu:
Here, the editor will have the following two options, implemented using python conditionals and SQL statements.
->Adding a movie: Here, the editor has the option of adding a unique movie id, along with a title, year and its duration, all of which is done through SQL statements. If an invalid movie id has been entered, the editor is asked to enter a new one. They are also able to add cast members to the movie or reject them in case the wrong one has been picked. If it's a new cast member, the editor can add them to the database with a unique id, birth year and name.
->Updating a recommendation: The editor can look at pairs of movies, along with the customer counts, throughout the three given time frames(month, annual and all time). They can also see if the pair is in the recommended list or not, and its score as well. All of this filtering was achieved through SQL statements. After this, the editor has the option of adding or removing a pair from the recommended list, along with updating its score.

Testing Strategy:
For testing, the group used the aforementioned SQL statements with a quality database which was developed by us. The statements were thoroughly tested multiple times on different machines. The python implementation testing was carried out through the general IDLE provided through the official python website, up to date with the latest version.

Contributions:
Agrim Sood - Developed the login screen and implemented the code for the first function in customer functionalities section. Also responsible for the design document.

Juan Fernandez Lezama - Worked on developing the remaining functions for the customer functionalities section and debugged the code, handling any bugs that appeared during the testing with due diligence.
Triaksh Mehta - Worked on the python implementation of the SQL statements, carefully integrating the two subjects,along with creating the design document.