Design document (MINI PROJECT REPORT)

(a) A GENERAL OVERVIEW OF THE SYSTEM (INCLUDES A USER GUIDE)

The basic idea of our system is to store data in a database and provide service to the users by accessing data from tables in the same database, accordingly. We integrated Python and sqlite3 to complete this project. All the tasks that the user can perform have been considered and are prompted when needed by the system. These instructions are straightforward and systematic and will be understood by the user. The user can be either the editor or a customer. Once the user enters the system, they are prompted to enter their login details (i.e, their cid/ eid and password). If they are not an existing user, they can sign up by entering a unique user id, name, and password. These details are then stored in the database.

If the login details of the user matches those of a customer, they are able to perform 6 tasks which are starting a session, searching for a movie, ending a movie, ending the session, logging out, or exiting the system. They can choose any one of these tasks by entering a number from 1 to 6 (These numbers correspond to the order of tasks they can perform).

Similarly, if the login details of the user matches those of an editor, they are able to perform 4 tasks which are adding a movie (including the cast members and their respective roles), updating a recommendation, logging out, or exiting the system.

(b) A DETAILED DESIGN OF THE SYSTEM

After the system has started, the login page will ask for the cid or eid and the password of the user. If they enter the user id which does not start with “c” or “e”, they will get a prompt asking to re-enter the details. A new user is determined by checking in the customers or editors table if the entered user id is present or not present). If not present only then the new account will be created. After a successful login, the system will display the respective tasks a customer or editor can perform. The user has been given 3 attempts to enter the password correctly. If the user doesn’t enter the correct password after 3 attempts, the system will exit and the program will be terminated.
For a customer :
If they select option 1, a new session will be created and a random sid will be generated and displayed . If they select option 2, they will be asked to input no. of keywords and the keywords which will be used to search the movies and upto 5 movies can be displayed. The customer can choose to see more available options

or can select a movie by entering the movie’s serial number. If the customer keeps choosing to see more movies and the movies exhaust, a message saying all the movies were displayed will be shown. If the customer selects a movie, they can see the movie’s cast and the number of other customers who have watched the movie. After this, they can choose to follow the cast member (if they already follow them, a prompt will be displayed saying they already follow the person) and they can start watching the movie. If they haven’t created a session before, they will be asked to create a session before starting the movie they chose. The details of the movie they watch will be stored in the database, accordingly. If they select option 3, the movie they have been watching will end and the duration will be stored in the database. If the customer has been watching more than 1 movie, they will be prompted by the system to end the movie they want from the ones they have been watching. If the duration exceeds the total length of the movie, the movie will automatically be ended by the system. If they choose option 4, the current session will be ended. If the user has been watching more than 1 movie, the movies will also end. If they choose option 5, they will be logged out and be redirected to the login page of the system. If they select option 6, the program terminates and the system will close.

For an editor:
After logging in, the editor can select options from 1 - 4. If they choose option 1, they can add a movie by providing a unique mid (if the mid is already present in the movies table, the system will prompt to enter mid again), title of the movie, the year it released, and its runtime. After the successful insertion of the movie, the system will prompt for the number of cast members to be added. If the data for a cast member already exists, the system displays the details of the cast member (name, birth year) and asks if the editor wants to insert the cast member data or reject it. If the cast member does not exist, the system asks the editor to enter the name, birth year, and the role of the cast member to be inserted in the database.
If they choose option 2, they will be prompted to enter m/y/a (monthly, yearly, or all data records). The data will be displayed with the movie pairs, number of people who watched the movie, the score of the recommended movies (only if the movie pair is present in the recommendations table), and if the pair is present or not present in the recommendations table. The user is then asked to choose any of the 2 options displayed which are added to the recommended list (if not present) or update the score and delete a pair of the recommended list. If they choose option 1, they will be asked which pair they want to enter and then if the movie pairs are not in the recommended list, they will be updated to the recommendations table and the score will be set to NULL. If they are present, then it will show an error. If

the editor chooses option 2, they will be asked which pair they want to delete from the list.
If they choose option 3, they will be logged out and be redirected to the login page of the system. If they select option 4, the program terminates and the system will close.

(c) **TESTING STRATEGY**

We edited the dataset that was provided for Assignment 2 and made our own dataset which was also posted by Sriram Karthik Akella on the course discussion forum: here. We tried all possible test cases and they were running as expected. After testing a few conditions, we used to get errors and we formatted the code in the end in such a way that all the possible conditions are considered and errors are eliminated. To check if the rows (like the new sessions, etc.) are being updated, we used the software available online named DBBrowser for SQLite. We were able to identify the errors if any during the updating or inserting statements and then work on it, accordingly.

Furthermore, we have a query_test() function in main.py which we used to test some of the functionalities of our program.

(d) **GROUP WORK STRATEGY:**

Mini Project Breakdown: After the project was released, during the reading week we discussed how to go about starting the project. We then decided that each of us will do a certain part of the mini project and that all of us will code and upload our files to a **private repo** on Github after every major update rather than storing multiple versions locally on each of our laptops. We used to work on it on every alternate day during the reading week and spent about 3-4 hours each day. During the weekdays, we worked on it by either meeting at the library, the CSC building or having online meetings on Zoom.

The person responsible for each task is stated below:

- **Sriram Karthik Akella** was responsible for the customer’s tasks after a successful login (all the 6 sub parts). Furthermore, he played a part in formatting the code like making functions for each sub-problem as well as debugging the errors in the mini project.

- **Shreya Pekhale** was responsible for most of the SQL part of the code. She was the one who worked on the queries we implemented and debugged SQL errors in the code. In addition to that, she also helped in making the report about the mini project and worked with Almer regarding the editor’s actions.

- **Almer Muneer** was responsible for making the login screen and editor's actions in the mini project. Worked on the debugging process too.

We worked on this group report for the mini project together on the day of submission.

Overall, we feel that we learnt a lot from this project. We implemented everything we learnt by attending the lectures and labs into this mini project. This mini project helped us understand better how the integration between SQL and python works alongside dealing with SQL in Applications.