Design document (MINI PROJECT 2 REPORT)

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

The basic idea of our system is to create a Document Store and store data in a database and provide service to the users by accessing data from collections in the database. We integrated Python and MongoDB 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. Once all the collections are created, the user can interact with the program to perform a variety of tasks. The user is able to choose between options (1-5) or option e to exit. The option numbers correspond to the order of tasks they can perform.

(b) A DETAILED DESIGN OF THE SYSTEM

After the collections are created and the main program is run, the user will be prompted to enter an option.

- If they select option 1, the user will be asked to input the number of keywords he wants to search and accordingly will be able to input the keywords. The system will retrieve all titles that match the keywords and will display all related fields. After this, the user will be asked to enter a movie title from the displayed movies and the system will display the tconst of the movie title entered by the user. Along With this value, the average rating, number of votes, the names of crew members and their characters (if any). After the completion of this option, the user will be directed to the main menu.
- If they select option 2, the user will be asked to enter a genre and a minimum vote count. The system will then return the titles under the provided genre which has the given number of votes or more. The results are also sorted on the basis of the average rating for each title with the title having the highest rating on top. After the completion of this option, the user will be directed to the main menu.
- If they select option 3, they will be asked to enter a crew member name. The professions of that crew member will be returned. The tconsts, jobs, and characters are also returned by the system. After the completion of this option, the user will be directed to the main menu.

- If they select option 4, they will be asked to enter a unique id. If the id is not unique, the user will be prompted to input the correct unique id, again. Then they will be asked to enter the title of the movie, the start year, the runtime, and the genres. These values will be stored in the database in the correct format with title type set to movie and isAdult and endYear both set to /N. After the completion of this option, the user will be directed to the main menu.
- If they select option 5, the user will be able to input the nconst of the cast member (which is the cast member id). If the user enters a value which is not in the collection, an error message will be displayed. After entering the correct value, the system will ask the user to enter a title id (which is the tconst). If the user enters a value which is not in the collection, an error message will be displayed. After entering the correct value, the ordering will be set to the largest ordering plus one. Then, we ask the user to enter the category and the data is stored in the collection in the proper format with other fields set to Null. After the completion of this option, the user will be directed to the main menu.
- If they select any other option by entering a random integer, the user will be asked to enter the correct option, again.
- If they select option e, the program will terminate and will exit from the Document Store.

(c) <u>**TESTING STRATEGY**</u>

We used the dataset which was provided by the professor on eClass which was in the format of 4 .tsv. We converted these files to json format and then used those collections for our main program.
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 data is being updated in the collections (for options 4 and 5), we used the software available online named MongoDB Compass. We were able to identify the errors if any during the updating or inserting statements and then work on it, accordingly.

(d) <u>**GROUP WORK STRATEGY:**</u>

<u>Mini Project Breakdown:</u> After the project was released, 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 which would be chaotic. We made sure that every group member had an equal amount of workload so that it is fair for all. We used to work on it on every alternate day during the week and spent about 2 hours in every meeting. During the weekdays, we worked on it by either meeting at the library, the university buildings or having online meetings on Zoom.

The person responsible for each task is stated below:

- **Sriram Karthik Akella** was responsible for most of the tasks in Phase 2, mainly **"Search for titles"** and **"Search for cast/crew members"**. 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 the **"Search for genres"** and **"Add a cast/crew member"** in Phase 2. She also played a main part in making the code more readable by commenting on the code and adding docstrings to functions in the file.

- **Almer Muneer** was responsible for making Phase 1 and the **"Add a movie"** part which was under Phase 2 in the mini project. Worked on the debugging process too and helped Shreya by commenting on the code to improve code quality/readability.

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 MongoDB and python works alongside dealing with MongoDB in Applications.