CMPUT 291 File and Database Management

Mini-Project 1

Design Document

hammad & jwkorolu

1. Overview

This section will give you a detailed overview on how to use this program

1. Start by specifying the name of the database you wish to work with

```
Please enter the filename of your database: (ex: 'database.db'):
```

2. Once you have connected to a database you will be met with this screen. Enter [1] to login as a customer or [2] to login as an editor.

```
-----------------------------------------------
Please choose an option to Login
[1] Customer
[2] Editor
Please type your selection here:
```

3. If you choose to continue as a customer, you can choose to [1] login or [2] register if you don't have an account yet.

```
Please select [1] to login, [2] to register or type [esc] to go back:
```

   a. If the user chooses to register, you will be prompted to fill in some information. (Your name and a password). You will be generated a cid which you will use to log in.
   b. If you choose to login you will prompted for your cid and password.

4. Once logged in customers can select any of the items from the menu.

```
Welcome Julian Koroluk!

Please select an option below
[1] Begin a new session
[2] Search for movies
[3] End watching movie
[4] End the current session
[0] Logout
Please type your choice:
```

   a. Selecting [1] will begin a new session for the user
   b. Selecting [3] stops the currently watched movie
   c. Selecting [4] will end the current session and any movies being watched
   d. Selecting [0] will log the user out and end any sessions and stop movies being watched

e. Selecting [2] will open another menu where the user can search using a keyword

```
Search has returned 5 result(s).

1. The Shawshank Redemption | Year: 1994 | Runtime: 144 minutes
2. The Terminal | Year: 2004 | Runtime: 128 minutes
3. The Fate of the Furious | Year: 2017 | Runtime: 136 minutes
4. The Avengers | Year: 2012 | Runtime: 143 minutes
5. The Dark Knight | Year: 2008 | Runtime: 152 minutes

Please choose a movie by typing its corresponding number:
```

i. The user can then select the movie of their choice by entering its index

```
Would you like to:
[1] Find out more information about 'The Avengers'
[2] Proceed to the screen
Please type choice here:
```

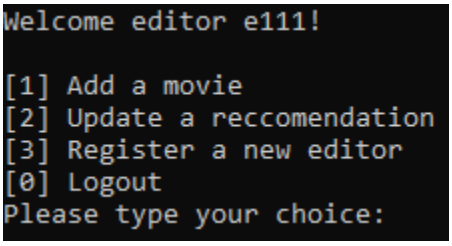
ii. The user can select [1] to have more information displayed about the movie or select [2] to explore more options

```
Movie information - 'The Avengers':

Release year: 2012
       Views: 0
     Runtime: 143 minutes
        Cast:
                Scarlett Johansson as Black Widow
                Steve Rogers as Captain America
                Chris Hemsworth as Thor Odinson

 Press enter to return...
```

```
You are now in "The Avengers's" movie screen, please select one of the following:
[1]Follow a member of the cast
[2]Watch movie

Please type choice here:
```

```
The Avengers's" cast:
                [1] Scarlett Johansson as Black Widow
                [2] Steve Rogers as Captain America
                [3] Chris Hemsworth as Thor Odinson
Please choose one of the following cast members to follow:
```

5. Editors can also log in and perform a variety of functions. The login process is the same for editors as it is customers.



   a. Select [1] allows an editor to add a movie to the database
   b. Selecting [2] allows an editor to update a recommendation
   c. Selecting [3] allows an editor to add a new editor
   d. Selecting [0] logs the editor out.

2. Software Design

We chose to write our program in python as we thought it would be the easiest for us to work with. We broke up most of the requirements into their own functions, and helper functions as needed. We have functions that include:

- *loginScreen :* Allows editors and customers to log into the program
- *newUserScreen :* Allows a person to create a new user profile and adds it to the database
- *idGenerator :* Generates id's for the type specified (pid, cid, sid, mid, and eid)
- *sessionStart :* Starts a session for a customer
- *sessionEnd:* Ends the last opened session
- *main :* main program containing most of the menus and places for functions to run
- *printMovieInfo :* displays information about a selected movie
- *checkIfFollowing:* Returns a Boolean based on whether a cid is following a pid
- *followCastMenu:* Displays cast from a selected movie and allows users to follow one
- *endOneMovie:* Ends watching a specified movie
- *endOneMovieFromGT5:* allows user to select which movie to stop watching (> 5 results)
- *endOneMovieFromLE5:* allows users to select which movie to stop watching (<5 results)
- *endAllMovies:* stops all movies being watched
- *movieScreenMenu:* Displays options for movie titles
- *askMovieMenu:* Displays options for user with specified movie
- *displayMatchesLE5:* Displays matches when query yields <= 5 results
- *displayMatchesGT5:* Displays matches when query yields > 5 results
- *displayMatches:* Displays a list of movies the user can choose from
- *dictToSortedList:* Sorts a dictionary by value and stores the key into a list
- *searchWordsMenu:* Searches db for specified values
- *handleMovies:* manages the movie search menu

We also use several libraries to help our software run. These include:

- ***OS**:* Mostly used to clear the terminal (keeps menus clean)
- ***Random***: Used to generate random id's
- ***Time:*** Used for time.sleep() and datetime functionality
- ***Sqlite3***: Used to manipulate the database
- ***Datetime***: Used to record session duration
- ***Select***: Library of functions hammad wrote

Combining these functions together produces a program that is easy to run and easy to make changes to if needed.

3. Testing Strategy

Our testing strategy involved testing at each point in the development of this program. As we wrote our separate functions, we would try a variety of inputs to ensure proper behaviour. Once we had enough functions, we would also test several functions working together to see if any errors arose. Once we completed all the requirements. We ran the entire program testing the functions we worked on, and then tested the functions the other person worked on. We then found some minor issues and worked those out and completed more testing.

4. Group Work Break-Down

The two of us met on a discord call to break apart the project into parts. We decided to separate the project by the requirements as that would be the clearest and easiest to understand. Any other parts that we discovered as we worked on it, were divided up. We usually met on discord to discuss the project but met in-person if possible. Below is the initial breakdown of the project and how we broke it up.

- Hammad
  - *Search for movies*
  - *End watching a movie*
  - *Add a movie*
  - *Update Recommendation*
  - *Recommendations (Display monthly, annual and all time)*
- Jwkorolu
  - *Start a session*
  - *End the session*
  - *Login Screen + Registration*
  - *Editor/Customer screen*
  - *Recommendations (Create new/Update recommendations)*