291 MINI PROJECT 1

Joshua Ji (jnji)
Kailash Seshadri (kailash)
Vedant Vyas (vvyas1)

## SYSTEM OVERVIEW/USER GUIDE:

Customer User Guide:

This is a movie streaming service that allows customers to sign into their personal accounts to start sessions and watch different movies in the sessions. If they don't already have an account to log into, they can choose to sign up for a new account.

Upon logging in, the user must enter the correct ID (case-insensitive) and password (case-sensitive) combination to log in. To sign up, the user must first enter a cid and password; the cid which will be checked to see if the cid already exists.

Once logged in, they are presented with the main system page, where they can choose from a variety of commands, depending on if they are a customer or an editor. They can also type "H" to see a list of commands. The commands are as follows:

AVAILABLE COMMANDS:

- (SS) Start a session
  Start a session where you can watch a movie. This assumes the user does not start multiple sessions at the same time.
- (SM) Search for a movie
  Search for a movie, and then choose to watch it (if the user has started a session), or follow one of the cast members.
- (EM) End watching a movie
  Stop watching the current movie being watched by the user.
- (ES) End a session
  End the current session if the user is in one, and stops watching the movie in the session.
- (LO) Logout
  Ends any current sessions, and exits the application

Editor User Guide:

Editors can log in with their own ID and password. Once logged in, they can add new movies and the cast members for that movie. They can also generate a report of all the pairs of movies watched by customers in the past month, year, or all time.

Upon logging in, the user must enter the correct ID (case-insensitive) and password (case-sensitive) combination to log in. The eid,password combination is checked to see if it is correct.
Once logged in, they are presented with the main system page, where they can choose from a variety of commands, depending on if they are a customer or an editor. They can also type "h" to see a list of commands. The commands are as follows:

AVAILABLE COMMANDS:

- (AM) Add a movie
  Adds a movie to the database. You'll need a title, year, runtime, and the cast members (which you can also add with this command).

- (UR) Update Recommendation
  See movie statistics (how many people have watched pairs of movies in the past month, year or all time). Add a recommended movie pair, update the score, or delete a movie pair.
- (LO) Logout
  logout and exit the application

Note: All command inputs for all users are case insensitive.

## SOFTWARE DESCRIPTION/SPECIFICATION:

Main program flow is noted in the flowchart below. Please access the flowchart using this link (ualberta email):
https://drive.google.com/file/d/1ZMmihe3PYlYDtKi83pao3K06xylBPcZk/view?usp=sharing

To run the file ensure the following:
- The following python modules installed and are functional:
  doc, sqlite3, time, getpass, datetime, sys
- You have `main.py` in the same folder as the `utils.py`

Once all of these are verified, enter `python main.py <database_path>` into the command line to start the program. For example, `python main.py ./test.db` opens the program with the test database in the same folder.

## TESTING:

1. Automated unit testing
   We used nosetests and the unittest library to test the implementation of our main() and authenticate() using mocks. We also made this run on every commit to main and a PR to main using Github Actions.
2. Manual Testing (with other groups)
   We manually tested our search movie and update_recommendations outputs with other students in the class. Using this, we can test our code in different (usually larger) databases.
3. Manual Testing (with our own dbs)
   We have our own dbs we created which we use in our group, where we know exactly what the outputs of search movie and update_recommendations are. Using this, we can quickly test our code with a small sample size.

## GROUP WORK:

We first devised the program flow during our first meeting, providing a list of functions that we needed to complete. During this session, we completed connect(), login(), signup(), and system() together as a group. At the end of the meeting, we divided up the remaining functions among the members according to difficulty and time commitment. The functions were assigned as follows:

connect(path):	Group
login():	Group
signup():	Group
system(user, cust):	Group
create_session(cid):	Kailash Seshadri
search_movie(user):	Joshua Ji
start_movie(cid, sid, mid):	Kailash Seshadri
follow_cast_member(cid, pid):	Kailash Seshadri
end_session(user, sid):	Kailash Seshadri
end_movie(user, sid):	Kailash Seshadri
add_movie():	Joshua Ji
update_recommendations(user):	Vedant Vyas
close_sessions(user):	Kailash Seshadri
authenticate():	Vedant Vyas

We used VSCode live share to collaborate on the initial design and complete the first few functions. Once this basic structure was complete (up to a scaffolded system() function that had the logic of the main system page), we were able to upload the files needed to GitHub, and work on our own functions on separate personal branches.

Each member was expected to complete their functions by Tuesday (08/03/2022). During this time we would check up on each others' progress on a daily basis on Discord. This allowed us to ensure progress was being made and we could help out anyone that was stuck. We then met on Wednesday (09/03/2022) to merge them to the main branch. Merge commits happened and they were quite annoying, but we were able to fix them together over the meeting.