Milad Mesbahi

3/16/2023

CS 32

Project 4 Report

In this report, I will discuss the testing strategies used for the various classes in the movie recommendation project. These classes include TreeMultimap, User, Movie, UserDatabase, MovieDatabase, and Recommender.

For the TreeMultimap class, I focused on testing the insert and find operations. I began by inserting a few string-integer pairs, such as "car" and "carey". After each insertion, I performed a search for the inserted keys to ensure that the TreeMultimap stored and retrieved the items correctly. In addition to testing with single values, I tested the insertion and retrieval of multiple values with the same key to ensure the multimap functionality worked as expected.

For the User and Movie classes, I tested the various accessor and modifier methods provided by each class. I instantiated a User object with sample data, such as a name, email, and watch history, and then retrieved the data using the respective getter methods. Similarly, I created a Movie object with sample data, such as an ID, title, release year, directors, actors, genres, and rating, and verified the data retrieval using the getter methods.

The UserDatabase and MovieDatabase classes were tested by loading data from the provided input files (users.txt and movies.txt). I checked the correctness of the load function by querying the databases for specific users and movies based on their email addresses and movie IDs, respectively. In addition, I tested the get\_movies\_with\_director, get\_movies\_with\_actor, and get\_movies\_with\_genre methods in the MovieDatabase class to ensure the returned movies matched the search criteria.

Lastly, for the Recommender class, I instantiated the class with the loaded UserDatabase and MovieDatabase objects and tested the recommend\_movies method. To do this, I provided various user email addresses and the desired number of recommendations. I verified the correctness of the recommendations by checking the compatibility scores and ensuring that the recommended movies matched the user's preferences.

Throughout the testing process, I used assert statements and printed output to the console to verify that the results were as expected. Additionally, I tested the program with smaller input files to facilitate debugging and gradually increased the size of the input files to measure the performance and memory usage. By following this comprehensive testing approach, I was able to identify and fix issues in the various classes, resulting in a robust and efficient movie recommendation system.