User Manual

Eugenie, Chi, Madison

OVERVIEW

This is a detailed user manual for our Final Project - COM212 Fall 2021. Please refer to this before running our program for maximum efficiency.

USER GUIDANCE

Run the **CreateSerialization.java** file first before running **Main.java**. Inside **CreateSerialization.java** there are 5 existing customers and 10 existing movies.

Log In Menu: Upon running the program, you'll be able to see a login system with 3 options: Admin, Customer, Quit Program. Admin and User each have their own menu after logging in. **Customers** should put in their credit card number: if the credit card number already exists ➻ this is an existing customer in CreateSerialization; if not ➻ this is a new customer (who is then asked to sign up by putting in their personal info (Name, Email). **Admin** should put in the credentials provided below:

Admin Login Credentials:

- Email: admin@movies.com
- Password: 212admin

There are questions that guide you through our customer/admin menu from start to finish, make sure to follow the instructions closely.

Exception Handling: Java is case-sensitive, make sure to pay attention to Yes/No or names. That said, there is solid exception handling for our entire program. Mistype/ wrong input format will prompt an error message "Wrong input/format. Try again". Our program will not crash.

DESIGN

Our program makes use of the following data structures: (Min) Heap, Hash Table, Queue Array, Binary Search Tree.

Customer.java - Each customer is created similar to a node, every single customer has a name, (last 4 digits of) credit card number, email, a wishlist and a watched list (containing movies they've watched from their wishlist).

Movie.java – Each movie is created similar to a node, every single movie has a title, release date, ID code, its Rotten Tomatoes rating (0-100) and a boolean condition that checks the status of the movie (available: isAvailable = True; unavailable: isAvailable = False).

We created 3 separate classes for our movie database, each is implemented using a different data structure and serves specific purposes:

- ➢ **AllMovies.java** – (Min) Heap to do basic functions including: add, delete, find least rated movie in O(1) runtime, print all movies.
- ➢ **SearchMovie.java** – Hash Table to search any movie by its ID (key) in O(1) runtime.
- ➢ **SortReleaseDate.java** – Binary Search Tree to sort all movies by release date (newest ↣ oldest, vice versa) in O(log n)

Our customers database – **AllCustomers.java** is implemented using Hash data structure. It has functions to insert, remove a customer, search a customer by (the last 4 digits) of their credit number (O(1) constant run time) and print the full list of customers.

WishList.java – A class for each customer's wishlist using Queue Array data structure, allowing users to add/delete movies from their wishlist in O(1) run time.**WatchedMovies.java** – A class for each customer's watched list using Queue Array, allowing users to add/delete movies from the list in O(1) run time and displaying the entire watched list.

Admin Menu (inside Main.java) – 7 options:

1. View All Customers
2. Find Customer – one at a time (by their credit card number (last 4 digits)
3. View All Movies
4. Add New Movie
5. Find and Make Least Rated Movie Unavailable
6. Find and Delete Movie (delete only unavailable ones; admin will be notified if it's available)
7. Log Out (return to customer/admin/quit)

CustomerMenu.java (inside Main.java) – 6 options:

1. View All Movies (sort by release date)
2. Find Movie by ID
3. Watch Next (next movie in user's wishlist)
4. Watch Again (list of movies user has watched from their wishlist)
5. Change Personal Info (user can change their name/email)
6. Log Out (return to customer/admin/quit)

Serialization: Changes made will be saved directly to our customer and movie databases (ser files).