
UTM

UNIVERSITI TEKNOLOGI MALAYSIA

# SECJ1023 TEKNIK PENGATURCARAAN II

# (PROGRAMING TECHNIQUE II)

**Semester II 2023/2024**

## Group Project Deliverable 2

## Problem Analysis and Design

## ***Movie Recommendation System***

**LECTURER : DR LIZAWATI BINTI MI YUSUF**

**SECTION : 4**

**GROUP : 5**

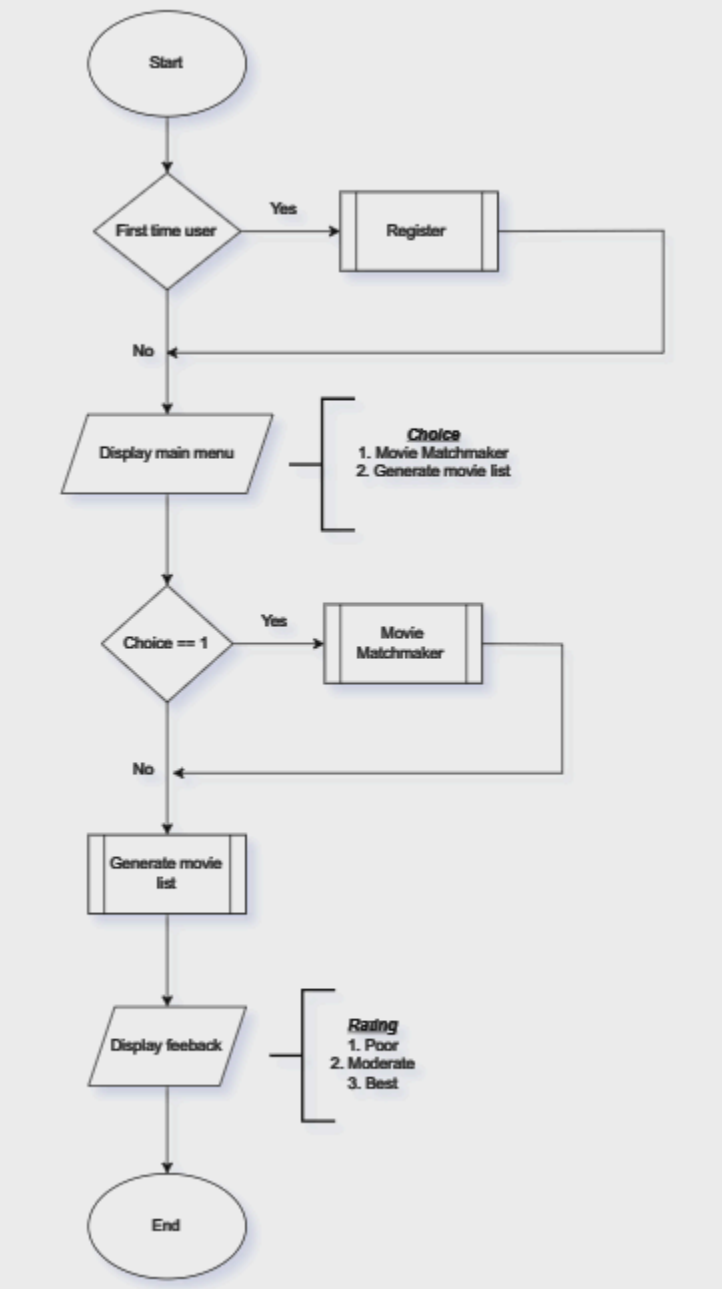
**STUDENTS :**

- **ADRIANA ZULAIKHA BINTI ZULKARMAN (A23CS0035)**
- **MELODY LUI RUO NING (A23CS0244)**
- **LEO MIN XUE (A23CS0237)**

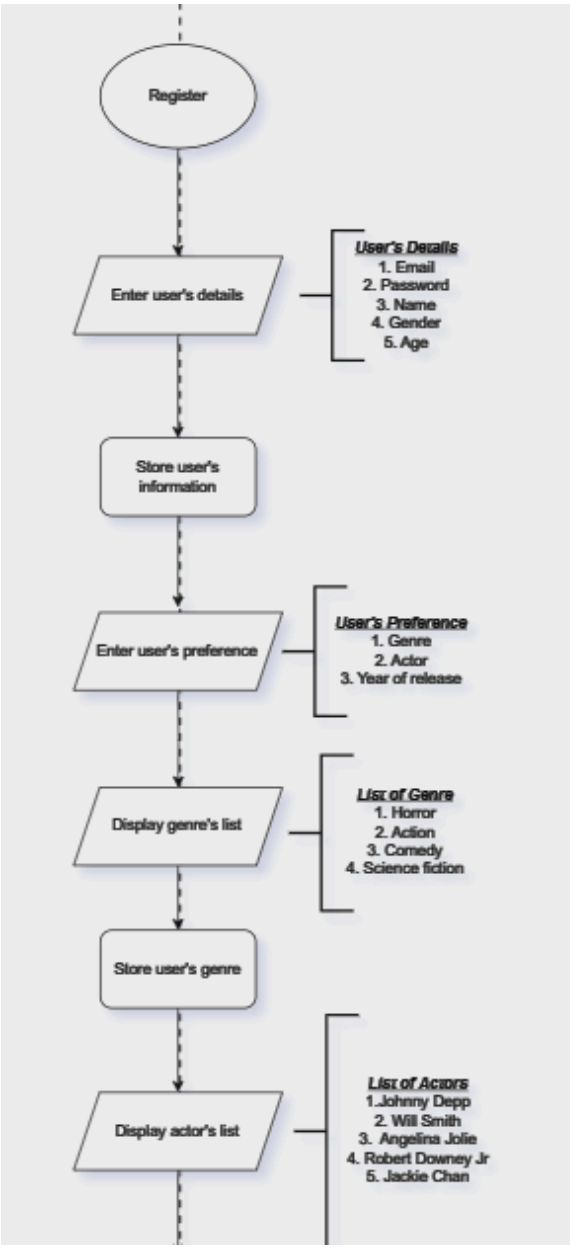
# TABLE OF CONTENTS

# Section A : Flow Chart

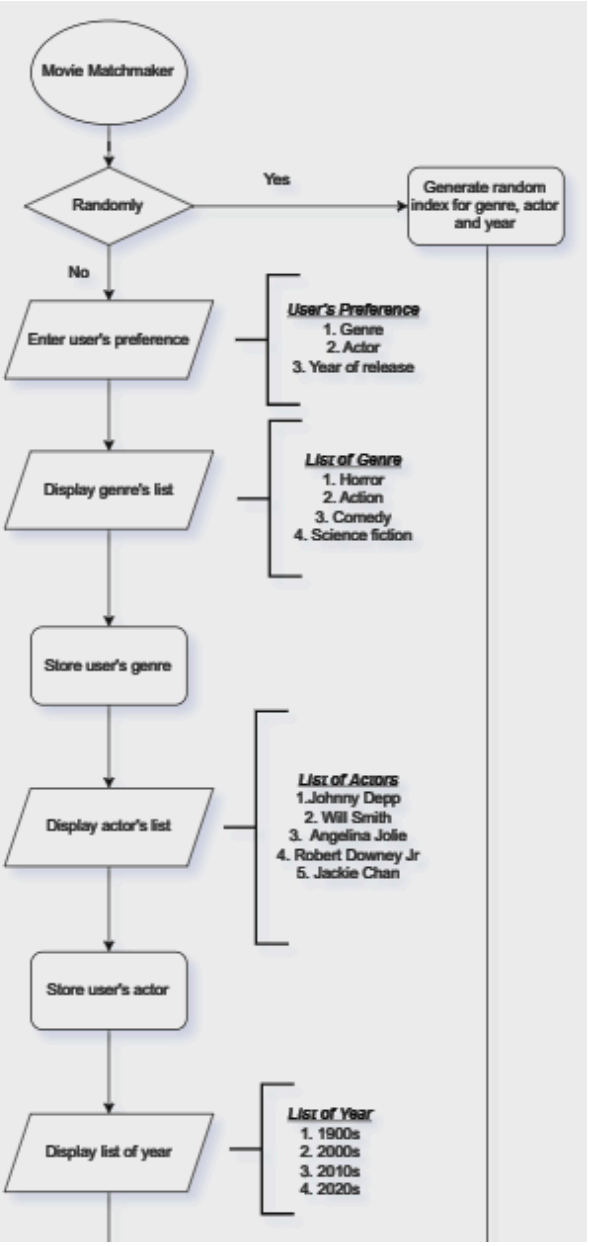
## Main Process

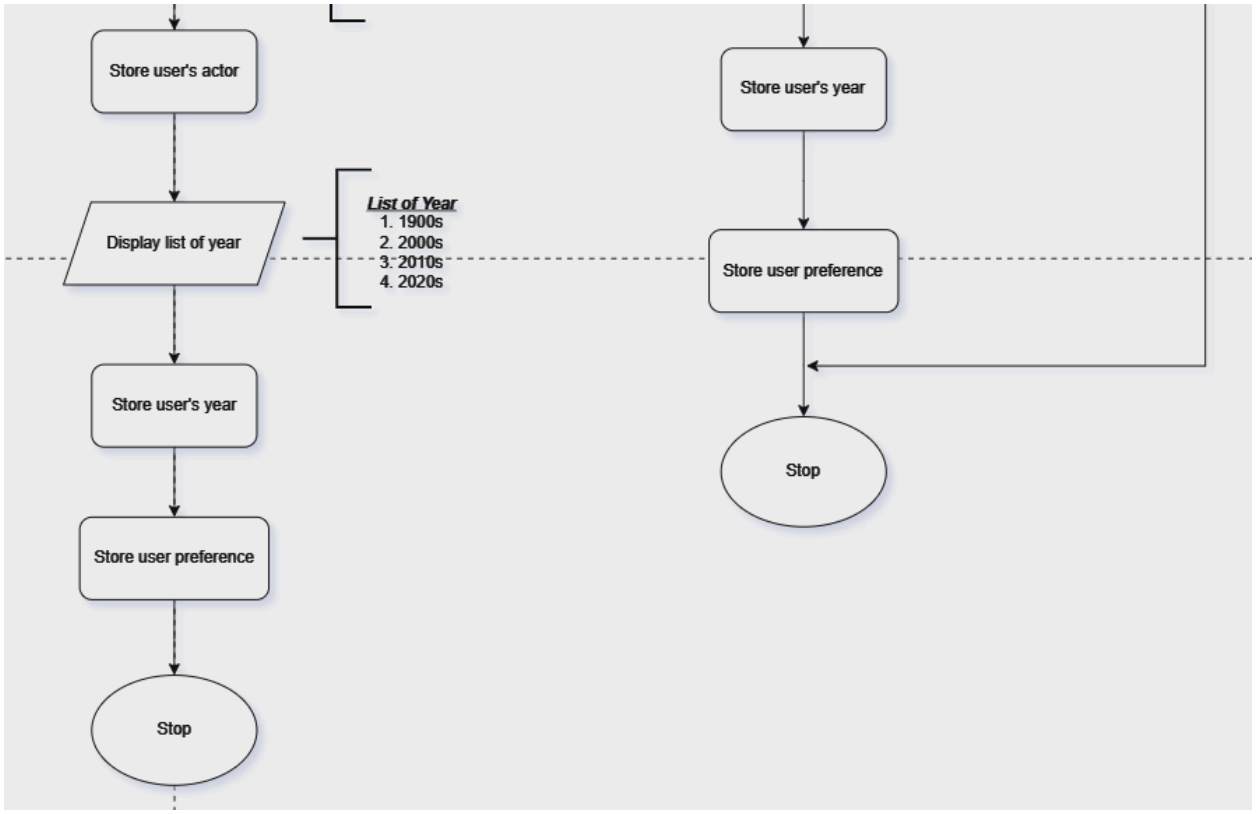


## 1.1 Register

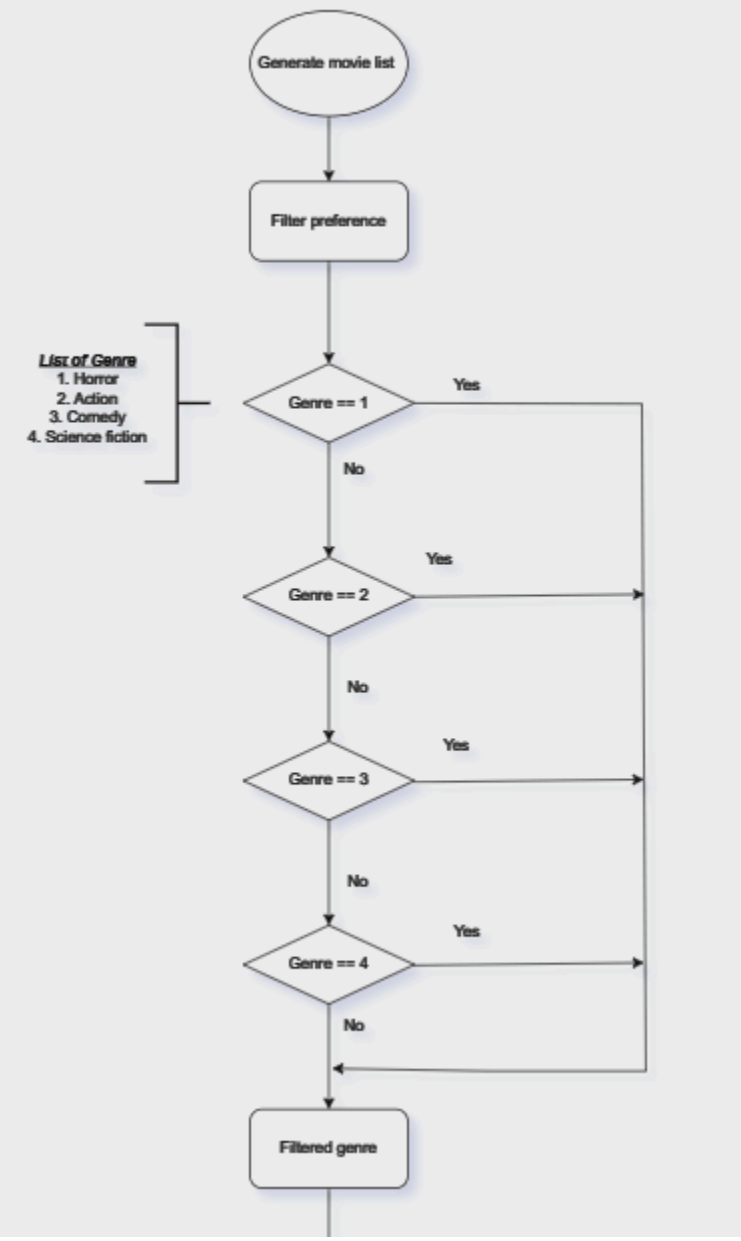


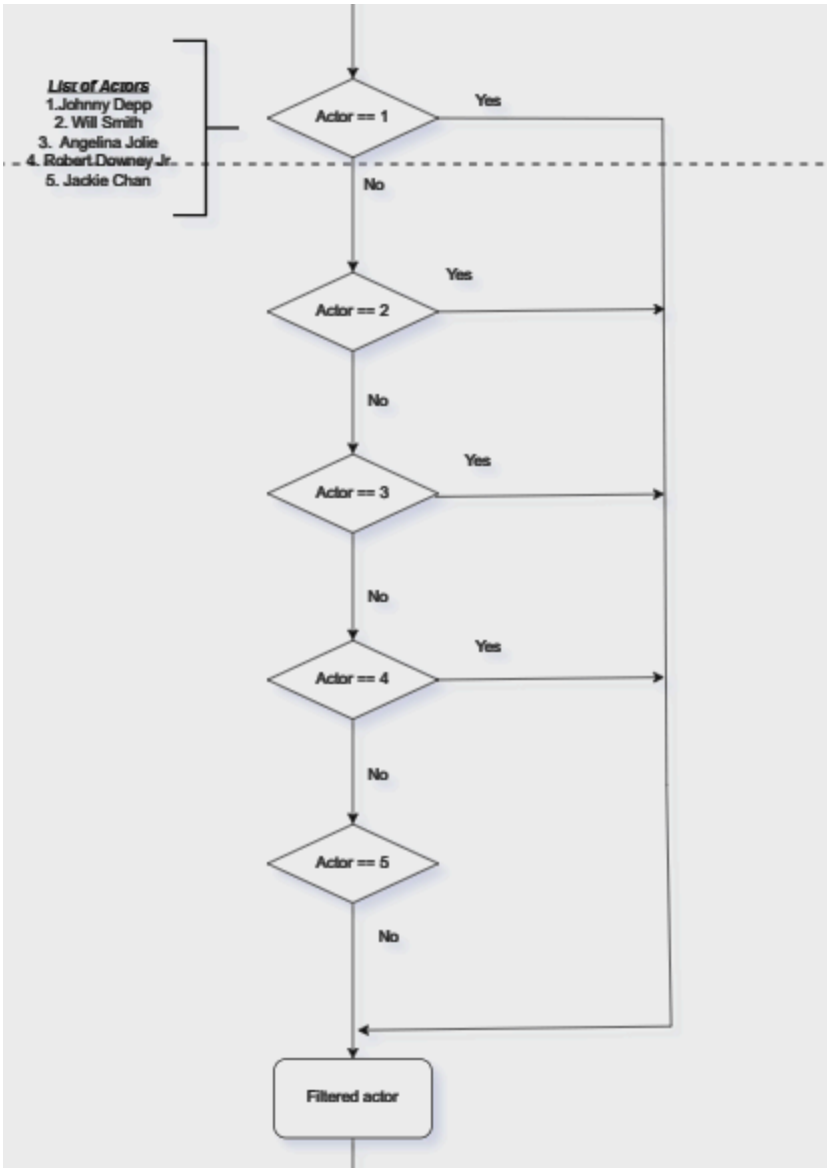

## 1.2 Movie Matchmaker

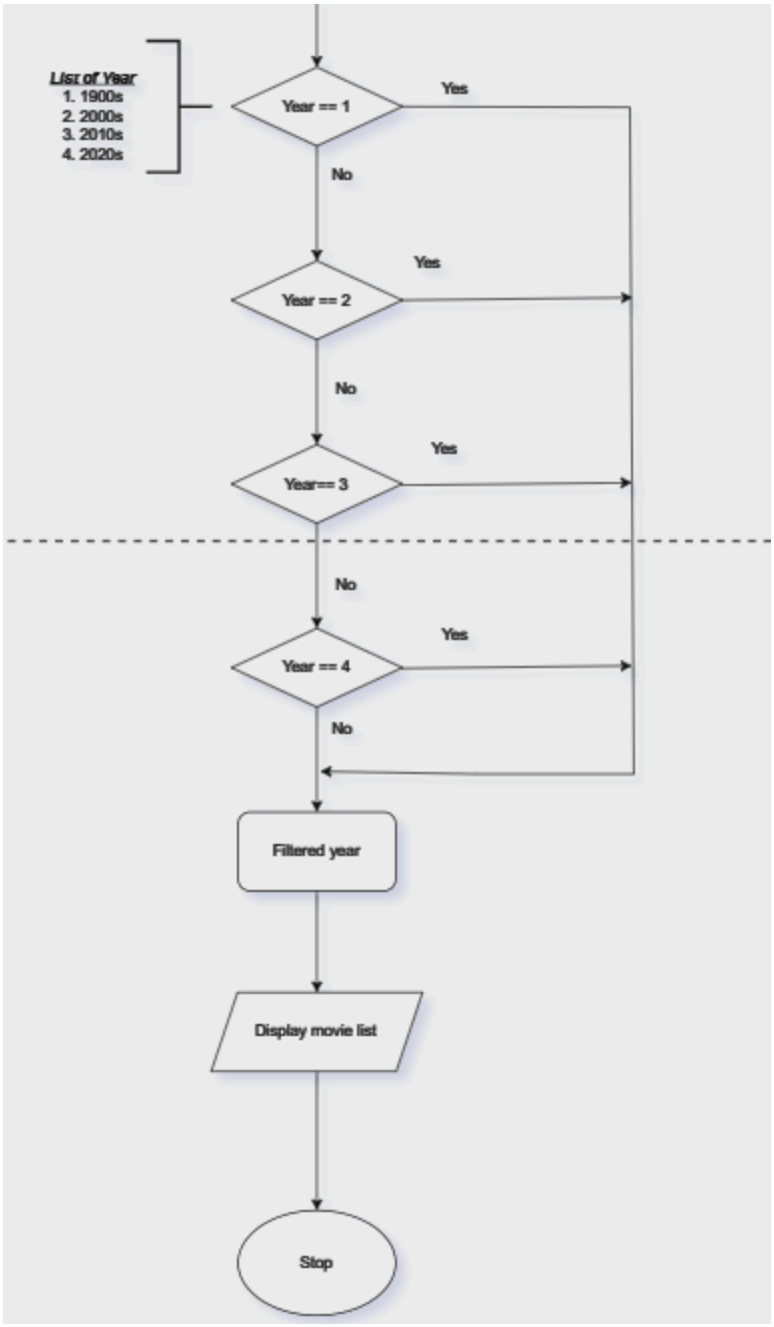



Store user's actor
Display list of year
List of Year
1. 1900s
2. 2000s
3. 2010s
4. 2020s
Store user's year
Store user preference
Stop
Store user's year
Store user preference
Stop

## 1.3 Generate Movie List




List of Actors
1.Johnny Depp
2. Will Smith
3. Angelina Jolie
4. Robert Downey Jr.
5. Jackie Chan
Actor == 1
Yes
No
Actor == 2
Yes
No
Actor == 3
Yes
No
Actor == 4
Yes
No
Actor == 5
No
Filtered actor


List of Year
1. 1900s
2. 2000s
3. 2010s
4. 2020s
Year == 1
Yes
No
Year == 2
Yes
No
Year== 3
Yes
No
Year == 4
Yes
No
Filtered year
Display movie list
Stop

## Main Process of the Flow Chart

If the users are new to the system, they need to register and create a new account. While creating an account, they are required to enter their personal details and also movie preferences such as genres, actors and year of release. The data will be stored inside the system. The system will display the main menu after the users login into the system. Users can choose between two choices which are 1. Movie Matchmaker or 2. Generate Movie List. In Movie Matchmaker, users can generate a movie list based on their current feeling or the system will randomly generate a movie list. In Generate Movie List, the system will filter out the user' preferences in order to generate a movie recommendation list. For example, if the user is a horror movie lover, then the system will filter out all movies that are non-horror. This process will repeat until a list of recommended movies is generated. Next, the recommended movie list will be displayed on the screen, and allow the user to browse and pick a movie from the list. After that, feedback will be displayed in order to collect reviews from the user after they watched the movie.

# Section B : Problem Analysis

**i. <u>Identifying Objects and Classes:</u>**

1. User: Represents a user of the system.

Attributes: username, password, movie preferences, feedback
Methods: update_ preferences() , feedback()

2. Account: Manages user account creation and storage of user information.

Attributes: username, password
Methods: create_account(), store_information()

3. Movie: Represents a movie entity.

Attributes: title, genre, rating
Methods: get_details()

4. Recommendation System: Manages the recommendation process.

Attributes: movie_database, recommendation_list
Methods: filter_preferences(), generate_recommendations()

5. RegisteredUser: A user who has registered an account in the system.

Attributes: username, password, feedback, movie_preferences
Methods: update_preferences(), feedback(),create_account(), store_information()

6. Movie Preferences: To specific tastes and criteria that a user has when selecting or recommending movies.

Attributes: genre, year_of_release,actors,directors
Methods: set_preferemces(), get_preferences()

**ii. <u>Identifying Class Relationships :</u>**

Association:

User and Account: Each user has one account. So, the User object needs to be associated with the Account object for user authentication and account management.

User and Movie: The User object has a relationship with the Movie object through their movie preferences. Each user may have multiple movie preferences.

User and Movie Preferences: Each user may have multiple movie preferences, indicating a one-to-many association between User and Movie Preferences. This association allows users to manage and update their movie preferences as part of their account.

User and Recommendation System: Allowing users to interact with the recommendation system to receive personalized movie recommendations based on their preferences. It can be a one-to-one or one-to-many relationship. At the same time, it can also enable users to benefit from the recommendation system's functionality in order to enhance their movie-watching experience.

Inheritance:

User and RegisteredUser: The flow indicates the concept of creating an account, storing user information, entering movie preferences, and providing feedback, which are typically associated with registered users.

Justification:

The association between User and Account is necessary because each user needs to have an account for authentication and storing their information. At the same time, users can also search back history as those objects store user's details. Furthermore, the association between User and Movie reflects the fact that each user has preferences for multiple movies which indicates a many-to-many relationship between users and movies. Besides, the User class maintains a list of movie preferences associated with each user. This association allows users to express their preferences and interests within the system.

By inheriting from the User class, the RegisteredUser class can access and utilize the common attributes and methods defined in the User class. For instance, it includes additional attributes and methods specific to registered users such as account creation, storing user information, storing movie preferences and updating the system based on feedback. Additionally, it also simplifies code reuse and maintenance by centralizing common user actions in the base class and extending it for registered user-specific tasks. Hence, it can avoid redundancy. Inheritance can also be useful to manage different types of users with varying levels of access and privileges in the system. Lastly, the inheritance relationship between User and RegisteredUser classes aligns with the principle of code reuse and facilitates the implementation of the system's functionality.

# Section C: Class Diagram