



**De La Salle University- Manila
Gokongwei College of Engineering**



LBYCPA2
Data Structures and Algorithms Laboratory

Project Documentation

FlixPick: Movie Recommendation System

A stylized black ink signature of Samuelle P. Barja, featuring a large, sweeping loop at the top.

Samuelle P. Barja

A stylized black ink signature of Aaron Tristan Jethro G. Dizon, featuring a cursive script with a prominent 'A'.

Aaron Tristan Jethro G. Dizon

Novemeber 16, 2023

I. Project Description

Overview

The "Movie Recommendation System" is a program developed to address the challenge of choosing movies from a vast and ever-expanding selection. Based on user choices and information from sources like Rotten Tomatoes, this approach is intended to provide tailored movie suggestions. It has functions including a login system, playlist and bookmark management, and a recommendation engine that generates personalized movie choices. By establishing a system this robust, users can be certain to locate movies that suit their tastes and browse the vast selection of movies more quickly.

This study is significant since it addresses the problem of information overload that often occurs while choosing movies. Users frequently find it difficult to obtain material that suits their interests due to the constant increase in the quantity of films accessible. Not only can personalized movie suggestions improve the user experience, but they also speed up the selection process. It seeks to improve user fulfillment by relieving the strain of having to sort through an overwhelming amount of alternatives and giving users a seamless and entertaining movie-watching experience.

Problem Statement

This project aims to resolve the common issue of users feeling overwhelmed by the vast selection of movies. [1] According to a report, it is revealed that 46% of the streaming respondents who took the survey are overwhelmed, due to the fact that they are exposed to continuous growth in the number of platforms and titles, which results in difficulty in choosing what they really want. Additionally, decision fatigue may develop in such a situation, making it more difficult to make decisions with assurance and effectiveness. [2] Due to the greater likelihood of making a poor choice, this abundance of alternatives can cause tension and anxiety. These problems may be resolved by narrowing the options and concentrating on a manageable number of possibilities. This will simplify the decision-making process, make it more fun, and lessen the weight of self-blame in the case that the decision has a negative effect. People may make better informed selections that are in line with their genuine preferences by being aware of the disadvantages of having too many options. By offering personalized recommendations and a platform for organizing movie selections, it aims to simplify the decision-making process.

II. Objectives

The primary goals and objectives of this project include the development of a recommendation system, user authentication, and efficient management of movie playlists and bookmarks:

1. Recommendation System:

Develop a robust recommendation engine that offers movie suggestions based on user preferences, previous selections, and comprehensive movie data. The system should continuously refine recommendations based on user feedback and viewing history.

2. User Authentication:

Implement a secure login system that allows users to create accounts, manage their profiles, and access personalized recommendations. User accounts should store essential information, such as viewing history and preferences.

3. Bookmark/Playlist Management:

Enable users to create, edit, and delete movie playlists or bookmarks for future viewing. These features should be user-friendly and adaptable to user needs.

The construction of a recommendation system, which seeks to make movie suggestions to users based on their tastes and watching history, is at the heart of the project's scope. The establishment of a user authentication system is also a crucial element since it will make it easier to create and manage user accounts and store user data for customization. The project also places a strong emphasis on the construction and administration of user-specific playlists or bookmarks for the movies users want to view. However, successful constraint management is essential to the project's success. Notably, the correctness and completeness of the movie dataset are necessary for the quality of movie suggestions, underlining the need for accurate data. To guarantee secure user authentication, data protection, and the preservation of user preferences and information, privacy and security are high objectives. To successfully complete the project, resource restrictions such as possible limits on server storage for user data and movie information must also be appropriately handled.

III. Methodology

In this section, we provide an overview of the Movie Recommendation System project, highlighting the need for such a system in the context of the overwhelming abundance of movies available today. The methodology described here outlines the major steps to be taken during the project's development.

1. Materials/Softwares used:

This section outlines the software and resources employed in the development of the program.

Integrated Development Environment (IDE): IntelliJ IDEA is the primary integrated development environment (IDE) used for software development. It provides a robust platform for coding, testing, and debugging.

User Interface Framework: JavaFX is utilized for creating the user interface of the application. It allows for the development of interactive and visually appealing desktop applications.

Rotten Tomatoes API: To ensure the program maintains an up-to-date and relevant movie database, integration with external data sources is recommended. The program is designed to access data from

external sources, such as the Rotten Tomatoes API. This integration enables the retrieval of the latest movie information, reviews, and ratings, enriching the recommendations provided to users.

By incorporating external data sources, the program ensures that users have access to the most current and comprehensive movie database, enhancing the quality of movie recommendations and user experience.

2. System Design:

Here, we delve into the design of the Movie Recommendation System's architecture. We discuss the major components and create visualization tools such as the Information Processing Diagrams (IPO). This diagram will help illustrate the system's flow, user interactions, and structural elements.

Input-Process-Output Table

Feature	Input	Process	Output
User Authentication System	User registration data (Password & Username)	Stores the user's data in a hash table to be read in the system	Confirmation of successful registration
	User Login data (Password & Username)	Reads the data to see if it is in the system	Forwarding to the main menu of the program
Recommendation Algorithm	User preferences	Utilize a decision tree algorithm to generate movie recommendations	List of movie recommendations based on user preferences
Bookmark/Playlist Management	User selection of movies for playlist	Add or remove movies stored in a linked list representing bookmarks and playlists	Customizable playlist/bookmarks to the user's liking
User Interface development	User's interaction in the program (menu, settings)	Use stack data structure to track the user data session with first in-last out concept	Ability to navigate user actions and forward to the choices

4. Code Implementation and program design:

a. User Authentication System:

This section describes the implementation of a user authentication system, allowing users to create accounts and securely log in. We emphasize the importance of data security in managing user account information.

Data Structure:

Hash tables are essential for managing users' accounts, enhancing security. User account information, including usernames and passwords, is stored in a hash table. Hashing Techniques are applied to provide rapid retrieval of user data during login while maintaining privacy.

b. Recommendation Algorithm:

In this part, we detail the development of the recommendation algorithm. The methodology explores how the system processes user preferences and historical data to generate movie recommendations. Different recommendation algorithms, including collaborative filtering, content-based filtering, and hybrid methods, are considered.

Data Structure:

A tree data structure will be utilized. A decision tree will be constructed to make more informed movie recommendations. This tree can consider multiple attributes such as genre, subgenre, and user ratings for an effective recommendation process. By traversing the decision tree, the system can narrow down recommendations based on user preferences.

c. Bookmark/Playlist Management:

In this section, we explore how the program effectively handles the movies users choose to bookmark, ensuring they can easily access and organize their favorite selections. Additionally, users have the flexibility to create personalized playlists for a tailored viewing experience.

Data Structure:

Linked lists are used for managing user interactions within playlists and bookmarks. Each playlist is represented as a linked list of movies. This allows users to efficiently add or remove movies from playlists and change the order of movies within a playlist. Linked lists provide a simple way to manipulate the playlist without the need for resizing.

d. User Interface Development:

This section focuses on the creation of an intuitive and user-friendly desktop application interface. We emphasize the importance of a well-structured menu system for users to access their playlists, bookmarks, and recommendations.

Data Structure:

A stack data structure is used to track user session data. Allowing users to backtrack through their actions within the application. If a user navigates through different movie recommendations or settings, a stack can keep a record of these interactions. Users can then use the stack to return to the previous state in the application.

5. Testing and Quality Assurance:

In this portion, we address the testing and quality assurance phase, emphasizing the importance of thorough testing, including unit testing, integration testing, and user acceptance testing.

JUnit testing: To ensure the reliability and correctness of individual components within the Movie Recommendation System, we employ JUnit, a widely used testing framework in the Java ecosystem. Each unit, including functions, algorithms, and modules, undergoes systematic testing with JUnit.

Mock Data: JUnit allows us to use mock data to simulate real-world scenarios, ensuring that units perform as expected. This is crucial for validating various edge cases and error handling.

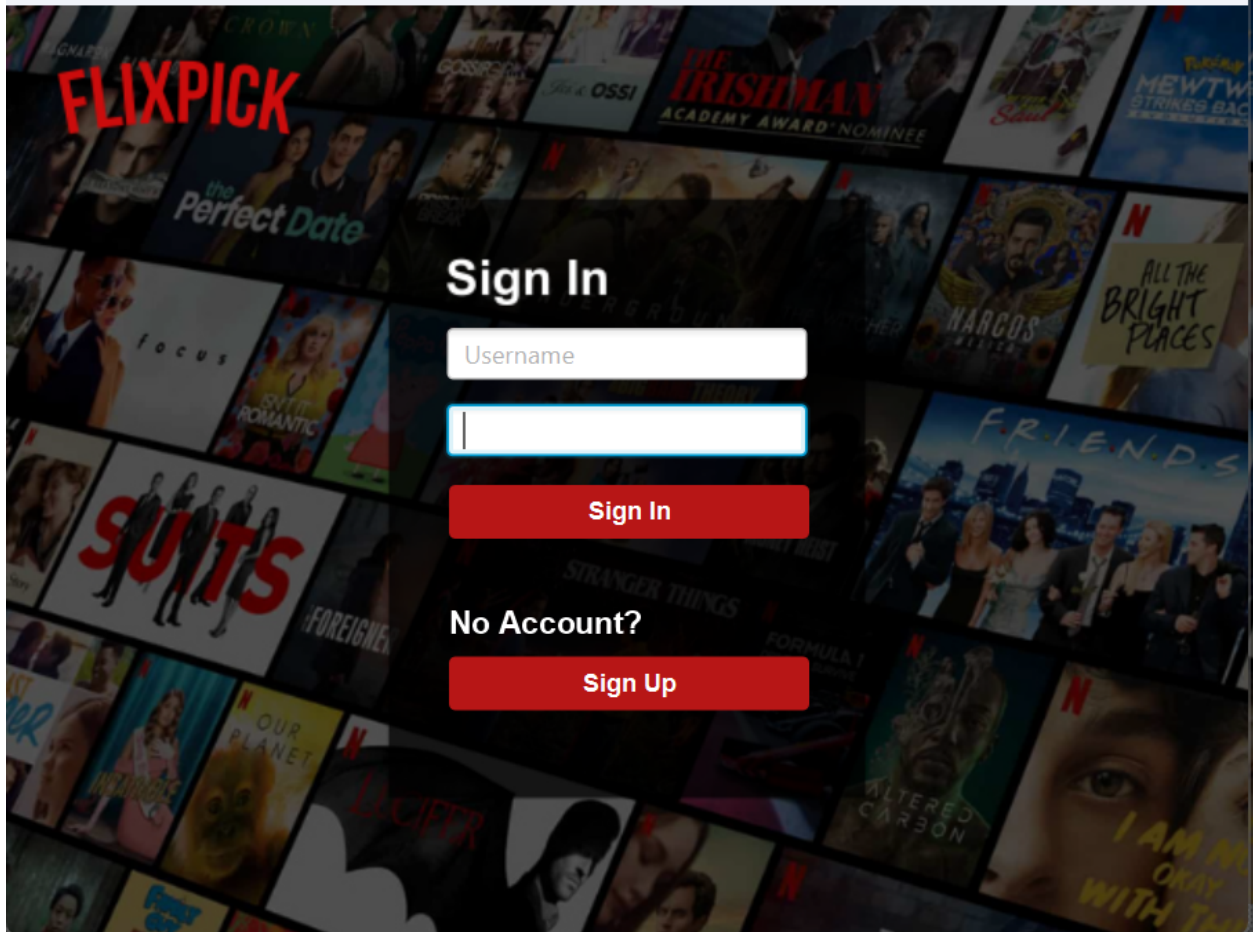
Bug Identification and Resolution: Throughout the testing process, we employ JUnit to identify and document bugs and issues systematically. The testing team works closely with the development team to prioritize and resolve these issues.

JUnit, combined with other testing methodologies, contributes to the overall quality and reliability of the Movie Recommendation System. This framework helps ensure that the software functions correctly and consistently, promoting a seamless user experience.

In conclusion, this research paper's methodology serves as a comprehensive guide for the development of the Movie Recommendation System. The structured approach and detailed descriptions of each phase and component ensure that the system effectively addresses the challenge of assisting users in finding movies to watch.

IV. Results and Discussion

FXML OUTPUT SO FAR





mainMenu.fxml



FLIXPICK

Home

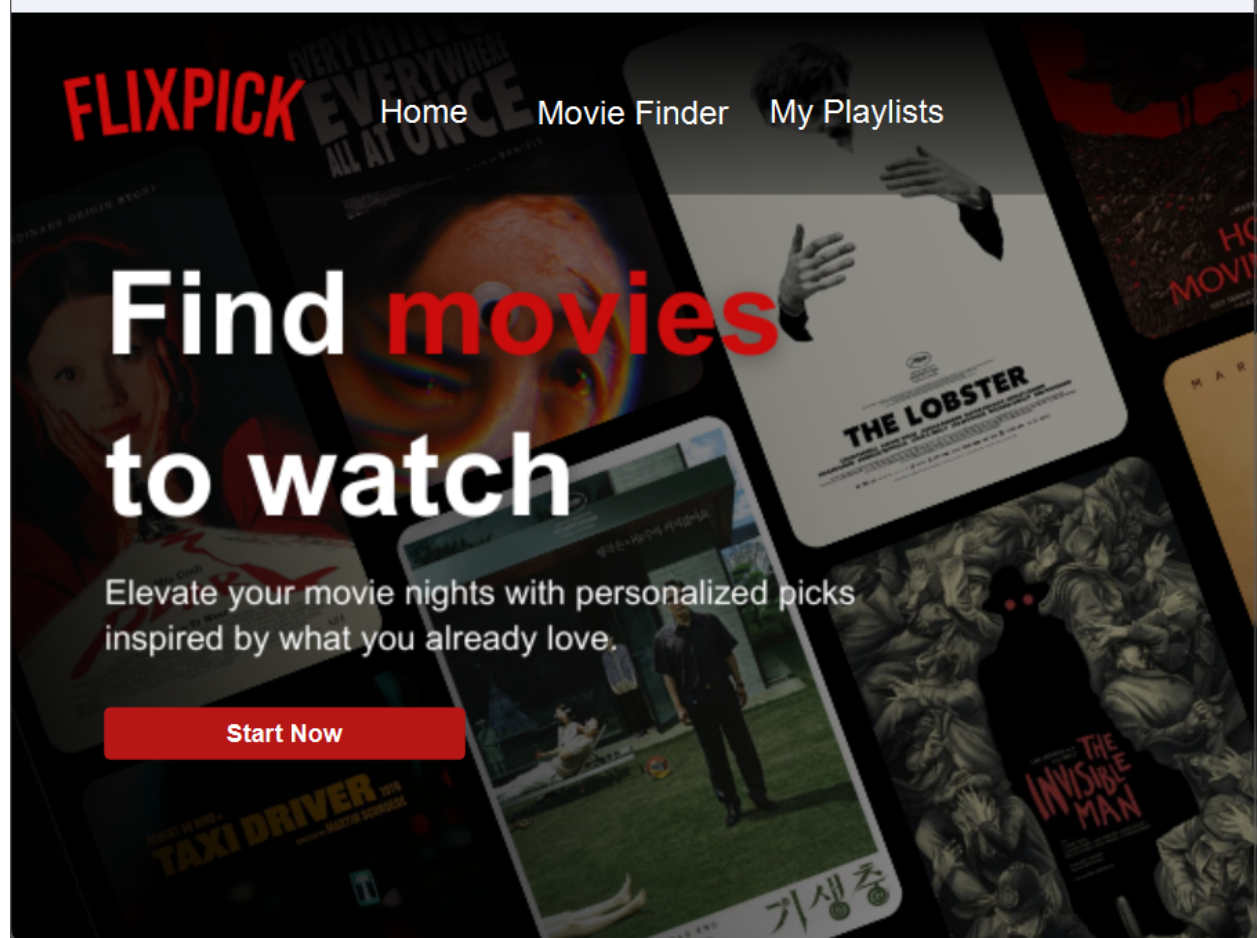
Movie Finder

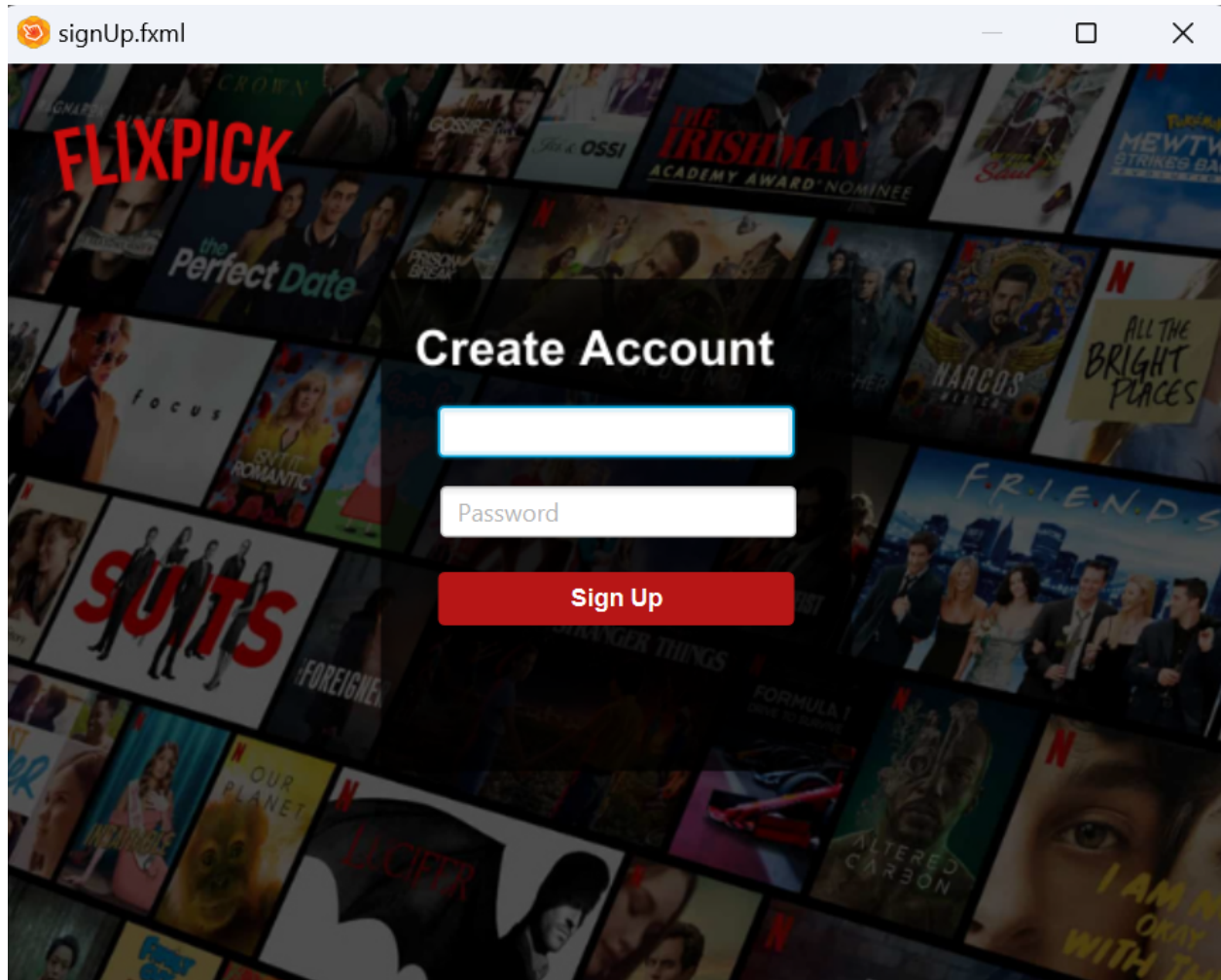
My Playlists

Find **movies** to watch

Elevate your movie nights with personalized picks
inspired by what you already love.

[Start Now](#)





V. Conclusion and Future Work

VI. Contributions

VII. References

- [1] R. Porter, "The Hollywood Reporter," The Hollywood Reporter, Apr. 06, 2022. [Online]. <https://www.hollywoodreporter.com/business/business-news/streaming-users-overload-choices-1235125685/?fbclid=IwAR0wok5kmDWDCDUp-d7Ktz8KfrfOfUCdBZojr3bWmN7EmTeeFCtmfRiI2s>

[2] Jake, “The paradox of too much choice,” *Everyday Psych*, Mar. 24, 2021.
https://everydaypsych.com/paradox-much-choice/?fbclid=IwAR1fvGcu1lK_7K7LQP-b3TC9a1EOiDr18PxwkC2Y9ijgitZG7_U_QgoUwyc