Software Requirements Specification

for

Recommender System

Version 1.0

Prepared by Izzat Ariz Bin Harun

25th January 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Izzat Ariz Bin Harun	25/1/ 2023	First version	1.0

1. Introduction

The recommender system is a web application developed using .Net Core Web API and React. This system has two primary users: the admin and the user. The system will improve the user experience by recommending movies based on their ratings of specific movies. The user can see discover new movies that are relevant to their taste.

This section of the Software Requirements Specification (SRS) document provides the purpose of the SRS, the scope, acronyms, and abbreviations used in the SRS, a list of references and the overview of the SRS.

1.1 Purpose

The purpose of this SRS document is to present a detailed description of the recommender system. This document will describe what the program should do and how it is expected to perform based on the functional and non-functional requirements.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document is the project managers, the testers and the developers of the program. The project manager can refer to the SRS document to monitor progress. The developers can refer to the functional and non-functional requirements in the SRS document to ensure the program is developed as per the specifications. The testers can refer to the SRS to verify that all system functionalities are working and included in the final product.

1.3 Project Scope

The recommender system allows the user to register and log in. The admin can add new users and admin. The admin also can manage movies. The recommender system expects the user to rate certain movies and will utilise the Cosine similarity matrix to provide user movie recommendations.

1.4 Definitions, Acronyms, and Abbreviations

ASCII	American Standard Code for Information Interchange
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
HTTPS	Hypertext Transfer Protocol Secure
SRS	Software Requirement Specification

1.5 References

[1] IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

[2] Collaborative filtering In Wikipedia. https://en.wikipedia.org/wiki/Collaborative_filtering

1.6 Overview

The remainder of this SRS contains a general description of the program and the specific requirements for the software system.

2. Overall Description

This section of the SRS describes the general requirements that drive the program's design.

2.1 Product Perspective

The recommender system is a web application developed using .Net Core Web API and React. The system includes functions for both the admin and the users.

2.2 Product Functions

Register

- This function will allow the user to register an account

Login

- This function will allow the user to login to the system

Logout

- This function will allow the user to logout from the system

View Movie

- This function will allow the user to view movies

Search Movie

- This function will allow the user to search for movies

Add User

- This function will allow the admin to add new user

Add Admin

- This function will allow the admin to add new admin

View User

- This function will allow the admin to view all users

Add Movie

- This function will allow the admin to add a new movie

Update Movie

- This function will allow the admin to update the movie

Delete Movie

- This function will allow the admin to delete the movie

Movie Recommendation

- This function will allow the system to recommend movies to the user

2.3 User Classes and Characteristics

Users of this system are not expected to have a high level of technical expertise. However, the end-user is expected to do basic internet browsing tasks. The characteristic of user classes will be movie lovers and casual movie watchers.

2.4 Operating Environment

The recommender system is optimised for desktops running Windows, Linux, and MacOS but also can be accessed by mobile as well. The system can be accessed by any device with an internet connection.

2.5 Design and Implementation Constraints

- The system will be web-based. Devices need a browser and internet connection to access the system.

2.6 Assumptions and Dependencies

- The program depends on a TMDB API to display movies
- The program depends on movie ratings by the user to provide movie recommendations to the users.

3. Interface Requirements

This section of the SRS describes the interface requirements for the system.

3.1 User Interfaces

Home page:

- The home page will show the list of movies in a card layout.

Search results:

- The search results display will list movies based on the keyword inputted by the user

Admin dashboard:

- Admin dashboard will show all functions for the admin

Log in / Register page:

- The login / Register page will display a form for the user to log in/register

Add user/admin page:

- The add user/admin page will display a form for the admin to add a user/admin

View user page:

- The view user page will display a form for the admin to view all users

Manage movie page:

- The view user page will display all the managing movie functions

3.2 Hardware Interfaces

- The recommender system is optimised for desktops but can be accessed by other devices mobile phones. As long as the device has a browser and internet connection, it can access the system.

3.3 Software Interfaces

Operating System Used During Development

- Windows 11

Program Used During Development

- Visual Studio 2022
- Visual Studio Code
- Microsoft SQL Server Management Studio
- Postman – Test API

Front-End

- React

Back-End

- ASP.Net Web API

4. System Features

This section of the SRS describes the requirements for the program's features.

4.1 Functional Requirements

REQ-1: The system must allow the user to register an account

REQ-2: The system must allow the user to login to the system

REQ-3: The system must allow the user to logout from the system

REQ-4: The system must allow the user to view movies

REQ-5: The system must allow the user to search for movies

REQ-6: The system must allow the admin to add new user

REQ-7: The system must allow the admin to add new admin

REQ-8: The system must allow the admin to view all users

REQ-9: The system must allow the admin to add a new movie

REQ-10: The system must allow the admin to update a movie

REQ-11: The system must allow the admin to delete the movie

REQ-12: The system must recommend movie to the user

4.2 Use Case Diagram

Below is a use case diagram of the program.

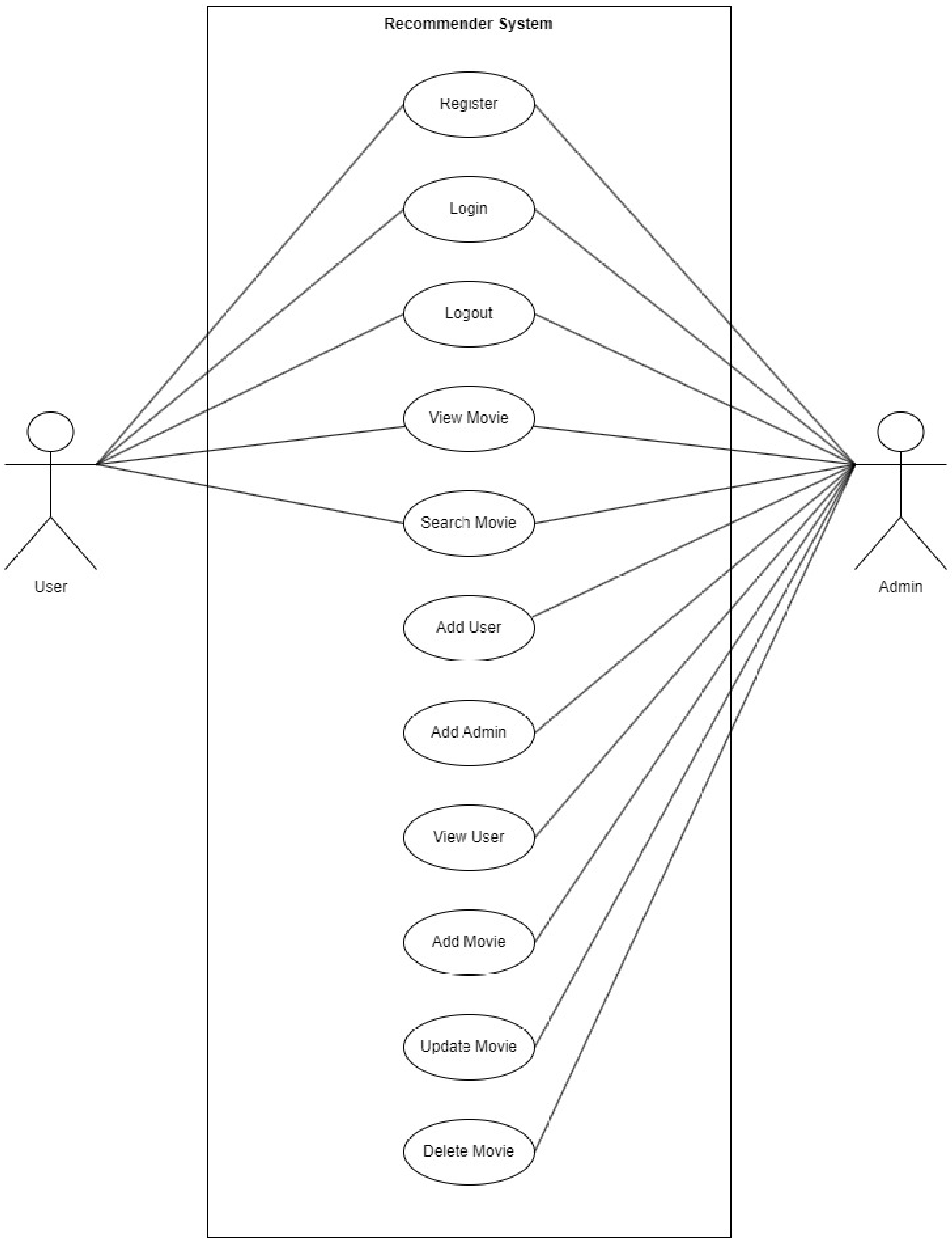

Figure 1 Recommend System Use Case Diagram

5. Non-functional Requirements

This section of the SRS describes the non-functional requirements for the program.

5.1 Performance Requirements

- The system shall load the list of movies in under 3 seconds.
- The system shall load the results of a search in under 3 seconds.

5.2 Safety Requirements

- The program shall not contain flickers or flashes that can trigger photosensitive epilepsy.

5.3 Security Requirements

- The system shall use HTTPS protocol for a secured connection.

5.4 Software Quality Attributes

5.4.1 Reliability

- The system shall provide reliable search results

5.4.2 Maintainability

- The system shall be built with modularity in mind to ease maintenance work.
- The system's architecture, design, implementation, and documentation must minimise the program's maintenance costs.

5.4.3 Availability

- The system shall always be available, 24 hours a day, 7 days a week.

Appendix A: Glossary

Flowchart A Visual representation of the sequence of steps and decisions needed to perform a process.

Use Case Diagram A Graphical depiction of a user's possible interactions with a system.