Software Requirements Specification

for

Movie Recommendation System

Prepared by

Student Name: Pearl Patel
ID: 21ITUOS134
Roll No.: IT112

B.Tech(IT)
SEM VI

Faculty of Technology
Dharmsinh Desai University
Nadiad

1. Introduction

1.1 Purpose of Document

The primary purpose of this document is to provide a comprehensive and detailed set of requirements for the development of a Movie Recommendation System. It serves as a reference guide for stakeholders, including developers, project managers, and quality assurance teams, ensuring a clear understanding of the system's functionalities, features, and performance expectations. The document aims to establish a shared vision and blueprint for the development process, aligning the efforts of all involved parties toward the successful implementation of the Movie Recommendation System.

1.2 Scope

The Movie Recommendation System is designed to revolutionize user engagement by delivering personalized movie suggestions based on individual preferences and viewing history. The scope of this system encompasses both frontend and backend components, combining React for an interactive user interface and Python for advanced machine learning algorithms. The system's reach extends from user authentication and profile management to seamless integration with a comprehensive movie database, ensuring a holistic and user-centric movie-watching experience.

1.3 Stakeholders

1) **Users:** Individuals seeking personalized and relevant movie recommendations based on their unique preferences and viewing history.
2) **Developers:** The team entrusted with designing, implementing, and maintaining the Movie Recommendation System.
3) **Project Managers:** Overseeing project timelines, resource allocation, and ensuring the successful delivery of the project.
4) **Quality Assurance Teams:** Responsible for rigorous testing to ensure the system meets specified requirements and functions as intended.
5) **Administrators:** Tasked with user account management, system analytics, and maintaining the overall health of the system.

1.4 Importance of Movie Recommendation System

In a digital landscape flooded with entertainment options, users often face the challenge of content discovery. The Movie Recommendation System stands as a solution to this challenge by employing sophisticated machine learning algorithms to analyze user preferences and recommend movies tailored to individual tastes. Beyond enhancing user satisfaction, the system aims to foster user retention and engagement with the platform. Its adaptability and continuous improvement through user feedback underscore its significance in delivering a dynamic and user-centric movie-watching experience.

1.5 Project Goals

- **Personalization:** Deliver a highly personalized movie recommendation experience that adapts to individual user preferences.
- **Engagement**: Increase user engagement by providing relevant movie suggestions, thereby prolonging user interaction with the platform.
- **Accuracy:** Develop a recommendation algorithm that continuously improves its accuracy over time through user feedback and collaborative filtering.
- **Scalability:** Design a system architecture that is scalable to accommodate an increasing user base and evolving database requirements.

1.6 Document Audience

This document caters to a diverse audience, including developers, project managers, quality assurance teams, and administrators. Developers will use it as a blueprint for system design and implementation, project managers will rely on it for resource allocation and timeline planning, quality assurance teams will refer to it for testing criteria, and administrators will utilize it for insights into user management and system monitoring.

1.7 Technologies to be Used

React will be employed for frontend development, focusing on creating an intuitive and responsive user interface, while Python will serve as the backend technology, handling user authentication, integrating with databases, and implementing the machine learning algorithms driving movie recommendations. The collaboration between React and Python will contribute to a seamless and personalized movie-watching experience for users.

1.8 Overview

The following sections of this SRS document are organized systematically to holistically define the "Movie Recommendation System" project:

- **Section 1: Software Description**

  This section aims to provide a comprehensive portrayal of the software, encompassing various aspects including user proficiency levels, general constraints, and assumed dependencies pertinent to the "Movie Recommendation System" project.

- **Section 2: Functional Requirements**

  Within this section, specific requirements and functionalities expected from the "Movie Recommendation System" software will be articulated. Functional requirements will be elaborated using use cases to ensure a profound and thorough understanding of thesystem's expected behaviours and capabilities.

- **Section 3: Non-Functional Requirements**

- This section is dedicated to describing the Non-functional requirements. Non-functional requirements define the aspects of a system that are not directly related to its functionalities but are crucial for its overall success, including aspects such as performance, security, and usability.

2. Overall Description for Movie Recommendation System

2.1 Product Perspective

The Movie Recommendation System is a dynamic and user-centric platform designed to revolutionize the way users discover and engage with movies. From a product perspective, it functions as an intelligent intermediary between users and a vast array of movies, utilizing advanced machine learning algorithms to tailor recommendations to individual preferences. The system operates within a larger digital entertainment ecosystem, seamlessly integrating with movie databases, user profiles, and external platforms to provide a comprehensive and personalized movie-watching experience.

1) **User Authentication and Profile Management:**
    - Enables secure user authentication.
    - Facilitates the creation and management of user profiles, including preferences and privacy settings.
2) **Movie Database Integration:**
    - Integrates with a comprehensive movie database, fetching and updating movie information regularly.
3) **Recommendation Algorithm:**
    - Utilizes advanced machine learning algorithms to analyze user preferences and generate accurate movie recommendations.
    - Incorporates collaborative filtering and continuous learning from user feedback for improved accuracy over time.
4) **Responsive User Interface (React Frontend):**
    - Develops an intuitive and responsive React-based frontend for a seamless user experience.
    - Features a personalized homepage, interactive search, and filtering options.
5) **Movie Details Page:**
    - Provides detailed information about each movie.
    - Suggests additional movies related to the user's current selection.

6) **User Feedback System:**

- Allows users to rate movies and provide comments.
- Integrates user feedback into the recommendation algorithm for continuous improvement.

7) **Admin Panel:**

- Empowers administrators with tools for user account management, system analytics, and monitoring.

8) **User Notifications:**

- Sends relevant notifications to users, keeping them engaged with the platform.

9) **Trending and Popular Lists:**

- Generates and displays trending and popular movie lists based on user activity and overall ratings.

2.2 Constraints for Movie Recommendation System

Despite its robust functionality, the Movie Recommendation System operates within certain constraints:

- **Data Privacy Regulations:**
  - Adheres to data privacy regulations, limiting the collection and usage of user data.
- **Dependency on External Movie Databases:**
  - Relies on external movie databases for up-to-date and accurate movie information.
- **Machine Learning Accuracy:**
  - The accuracy of movie recommendations is contingent on the quality and diversity of user data and the effectiveness of the machine learning algorithms.
- **Platform Compatibility:**
  - User experience may vary based on device capabilities and network conditions.

2.3 Operating Environment for Movie Recommendation System

The Movie Recommendation System operates in a digital entertainment environment, interacting with various components:

- **Frontend (React):**
  - Compatible with modern web browsers on desktops, tablets, and mobile devices.
- **Backend (Python):**
  - Hosted on servers capable of handling concurrent user requests.
  - Utilizes machine learning libraries such as Scikit-Learn and TensorFlow.
- **Database:**
  - Integrates with a movie database, ensuring data consistency and integrity.
- **External Platforms:**
  - Interfaces with social media platforms for user sharing.
  - Depends on external platforms for user authentication (if applicable).

3. Functional Requirements:

R.1 User Authentication

Description: Users must have a secure and seamless authentication process to access the Movie Recommendation System.

R.1.1 : Implement user registration with email verification.

R.1.2 : Provide a secure login mechanism with password encryption.

R.1.3 : Allow users to reset their passwords through a secure process.

R.2 User Profile Management

Description: Users should have the ability to create and manage their profiles to enhance the personalization of movie recommendations.

R.2.1 : Enable users to create and update their profiles with information such as preferred genres, languages, and release year preferences.

R.2.2 : Allow users to set privacy preferences for their profiles.

R.3 Movie Database Integration

Description: The system needs to integrate with a comprehensive movie database to fetch and display accurate movie information.

R.3.1 : Integrate with a third-party movie database API i.e. TMDb API

R.3.2 : Retrieve movie details including title, genre, language, release year, and synopsis.

R.4 Recommendation Algorithm

Description: The system needs to integrate with a comprehensive movie database to fetch and display accurate movie information.

R.4.1 : Users can view posts from others in their news feed.

R.4.2: Users can like and comment within the platform.

R.5 Responsive User Interface (React Frontend)

Description: Develop a user-friendly and responsive React-based frontend to facilitate an intuitive user experience.

R.5.1 : Create an interactive homepage with personalized movie recommendations.

R.5.2 : Implement a search functionality for users to explore and find specific movies.

R.5.3 : Enable filtering options based on genres, languages, and release years

R.6 Movie Details Page

Description: Each movie should have a dedicated details page providing comprehensive information and related recommendations.

R.6.1 : Display detailed information about a selected movie, including cast, crew, and user ratings.

R.6.2 : Suggest additional movies related to the user's current selection.

R.7 Admin Tools for Content Moderation

R.7.1 : Admins can access tools to moderate user-generated content, flag inappropriate posts, and manage user accounts.

R.7.2 Display analytics on user engagement, popular movies, and system performance.

R.8 User Notifications

Description: Users should receive relevant notifications to enhance their engagement with the platform.

R.8.1 : Send notifications for recommended movies, user feedback, and system updates.

R.8.2 : Allow users to customize notification preferences.

R.9 Trending and Popular Lists

Description: The system should generate and display trending and popular movie lists.

R.9.1 : Utilize algorithms to identify trending movies based on user activity.

R.9.2 : Display popular movies based on overall user ratings and views.

4. Non-Functional Requirements:

4.1 Performance:

- Ensure the system can handle concurrent user requests efficiently.
- Movie recommendation response time should meet acceptable limits.

4.2 Security:

- Encrypt user passwords for secure storage.
- Implement HTTPS for secure communication between frontend and backend.

4.3 Scalability:

- Design the system architecture to scale horizontally.
- Accommodate increased user load without compromising performance.

4.4 Usability:

- Develop an intuitive and accessible user interface.
- Ensure compatibility with various devices (desktop, tablet, mobile).

4.5 Technology Stack:

- Frontend: React for the user interface.
- Backend: Python for machine learning algorithms and API development.
- Database: Select an appropriate database for storing user data and movie information.

These non-functional requirements collectively ensure that the Movie Recommendation System not only delivers accurate and personalized movie recommendations but also meets high standards of performance, security, usability, scalability, and regulatory compliance. They contribute to the overall reliability and effectiveness of the system in providing an optimal user experience.