Software Requirements Specification

for

Recommender System

Version 1.0 approved

Prepared by
Muhammad Yasin Aiman Bin Rohalim

.NET TRAINING COLLABERA

30th January 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

A recommender system's primary objective is to enhance the user experience by making personalized recommendations based on the user's past behavior and preferences. The system analyzes user data, such as purchase history, ratings, and clicks, to identify patterns and predict future interests. By providing relevant and customized recommendations, the system aims to increase user engagement, reduce search time and frustration, and drive sales or content consumption. As a result, businesses benefit from increased customer satisfaction, loyalty, and revenue.

1.2 Intended Audience

Recommender systems aim to provide users with a personalized experience by considering their unique preferences and behaviors. The intended audience for these systems will vary based on the context in which they are being used. For example, in a commercial setting, such as an e-commerce website, the target audience is typically customers looking for product recommendations. In a content-based environment, like a news website, the intended audience is readers seeking personalized news suggestions. Additionally, there is a more general audience, such as movie-goers, who can benefit from movie recommendations.

1.3 Project Scope

The scope of recommendation systems for a movie website focuses on providing personalized and relevant suggestions based on user preferences, viewing history, and overall trends. This includes: personalized recommendations, similar movie suggestions, popular and trending movie recommendations, new releases and upcoming movie suggestions, and collection-based recommendations.

The main objective of a recommendation system(login and registration,creating api) for movies is to provide personalized movie recommendations to users based on their preferences and viewing history, in order to enhance their experience and satisfaction with the movie platform. The goal is to help users discover new and interesting movies they may not have otherwise found as well as, provide relevant and personalized movie suggestions to increase user engagement and satisfaction.

Recommendation systems for a movie website bring several benefits to both the website and its users. Firstly, by providing personalized and relevant movie suggestions, the recommendation system can increase user engagement and satisfaction. This can result in users spending more time on the site and having a better overall experience, leading to increased loyalty and repeat visits. Another benefit of recommendation systems is improved content discovery. With the vast amount of content available on movie websites, it can be challenging for users to find new and interesting movies. Recommendation systems can help users discover new titles they may not have otherwise found, increasing the chances of them finding something they like and improving their overall experience.

1.4 References

- Collaborative filtering:
  https://en.wikipedia.org/wiki/Collaborative_filtering
- Pearson correlation coefficient:
  https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
- Jaccard index:
  https://en.wikipedia.org/wiki/Jaccard_index
- Cosine similarity:
  https://analyticsindiamag.com/cosine-similarity-in-machine-learning/

2. Overall Description

2.1 Product Perspective

The product outlined in this SRS is a recommender system that leverages collaborative filtering to provide tailored recommendations to organizations. It is a novel, standalone application developed to address the increasing need for data analysis and optimization in various businesses. The system collects user behavior and preferences data to generate relevant recommendations aimed at diversifying markets, streamlining operations, and optimizing supply chains. The recommender system is integrated with other elements of the larger ecosystem through clearly defined interfaces to enable real-time data flow and recommendations. This product represents a substantial improvement over existing solutions and is expected to bring significant value to organizations.

2.2 Product Features

The product features include:

Personalized recommendations: Utilizing collaborative filtering, the system provides tailored recommendations based on user behavior and preferences data.
Data collection and analysis: The system gathers data from user behavior and preferences to generate relevant recommendations.
Integration with other systems: The recommender system integrates with other components of the larger ecosystem through defined interfaces to enable real-time data flow and recommendations.
Optimization of supply chains, diversification of markets, and streamlining of operational methodologies: The system's recommendations aim to help organizations in these areas.
Overall, the product is designed to provide a simple yet effective recommendation solution to organizations, leveraging the power of collaborative filtering to provide relevant and personalized recommendations.

2.3 User Classes and Characteristics

The anticipated user classes for the recommender system include:

- Business Analysts: These users are responsible for data analysis and optimization in their organizations. They will use the system to gather relevant data and generate actionable recommendations.
- Marketing and Sales Teams: These users will use the recommendations generated by the system to diversify their markets and optimize their supply chains.
- Operations Teams: These users will use the recommendations generated by the system to streamline their operational methodologies.

The user classes are differentiated based on their frequency of use and the subset of product functions used. Business Analysts and Operations Teams will use the system on a daily basis, whereas Marketing and Sales Teams will use it as required. The technical expertise, security, or privilege levels, educational level, and experience may vary for each user class, but the system is

designed to be user-friendly and accessible to users with varying levels of technical expertise. All user classes are important to satisfy, but Business Analysts are favored due to their critical role in data analysis and optimization in organizations.

2.4 Operating Environment

The operating environment for the software consists of the following components:

- Hardware Platform: The software can run on a variety of hardware platforms, including laptops, desktops, servers, and cloud-based systems. The specific hardware requirements will depend on the scale of deployment and the amount of data being processed.
- Operating System: The software is designed to run on Windows and Linux operating systems, with support for both desktop and server editions.

- .NET Core: The software uses .NET Core 7 as its primary framework. .NET Core provides a stable, secure, and scalable platform for building modern applications.
- React JS: The front-end of the software is built using React JS, a JavaScript library for building user interfaces. React JS provides a dynamic and interactive user experience, allowing users to interact with the system in real-time.

The software will peacefully coexist with other software components or applications, as long as they are compatible with the specified hardware platform, operating system, and software components. It is important to ensure that the software is deployed on a stable and secure infrastructure to ensure maximum performance and security.

2.5 Design and Implementation Constraints

The following constraints must be considered while building the recommender system:

- Corporate Regulations: The development must align with any relevant corporate policies and regulations, especially related to data security and privacy.
- Hardware Restrictions: The system must be capable of handling large amounts of data and perform real-time operations while considering memory and processing limitations.
- Integration with Other Systems: The recommendation engine must be able to seamlessly interface with other applications and data sources to gather necessary information. Interfaces must be secure, robust, and adaptable.
- Required Technologies and Tools: The recommender system will be built using .NET Core 7 and React JS, thus the development team should be well-versed in these technologies. Integration with various databases including SQL and NoSQL databases is also necessary.
- Concurrent Operations: The system must be capable of handling multiple parallel operations to accommodate multiple users accessing the system simultaneously.
- Language Support: The system must be capable of supporting multiple languages for user convenience.
- Security: The system must be designed to handle sensitive data securely, and also be protected against hacking, tampering, and other malicious activities.

- Design and Coding Standards: The development team must follow best software development practices, including coding standards, design patterns, and best practices for software development. The code must be maintainable for future improvements.

2.6 User Documentation

User Documentation Components:

- User manual - A comprehensive guide on how to use the movie recommendation system, including step-by-step instructions on how to navigate the application and its features.
- On-line help - An easily accessible source of information that provides answers to common questions about the application and its features.
- Tutorials - A series of interactive tutorials that teach users how to use the application effectively, including step-by-step guidance and examples.

Delivery Format: PDF, HTML

Standards: The user documentation will comply with the industry standards for software documentation, including clear and concise language, use of screenshots, and well-structured headings and content.

2.7 Assumptions and Dependencies

Assumptions and Dependencies for the Recommender System:

- Availability of data: The project assumes that the required data for implementing the Collaborative Filtering algorithms is available and accessible.
- Algorithm Accuracy: The project assumes that the User-Based and Item-Based Collaborative Filtering algorithms chosen for implementation will provide accurate recommendations.
- Third-Party Components: The project assumes that any third-party components used in the implementation are compatible with the project requirements and do not pose any limitations.
- Integration with other systems: The project assumes that the recommender system will be able to integrate with other systems seamlessly and that no compatibility issues arise.
- Dependencies:

- Data quality: The project's success is dependent on the quality of the data used for implementing the Collaborative Filtering algorithms.
- Integration with other systems: The project is dependent on the ability to integrate with other systems and access the required data.

3. System Features

3.1 Login And Register

3.1.1 Description and Priority:

- Login and Register feature for users to access the system.
- High priority as it is essential for the system to function properly.
- Benefit: 9 (allows users to access the system and use its services)
- Penalty: 9 (users cannot use the system without being able to log in)
- Cost: 7 (implementation may require time and resources)
- Risk: 5 (potential security risk if login process is not implemented properly)

3.1.2 Stimulus/Response Sequences:

- User enters username and password and clicks on "Login" button.
- System verifies the credentials.
- If the credentials are valid, the user is logged into the system.
- If the credentials are invalid, the system displays an error message.
- User clicks on "Register" button.
- User enters required information such as name, email, username, and password.
- System verifies the information.
- If the information is valid, the user is registered and logged into the system.
- If the information is invalid, the system displays an error message.

3.1.3 Functional Requirements:

REQ-1: User must be able to enter a valid username and password to log in.
REQ-2: System must verify the entered credentials and log in the user if they are valid.
REQ-3: System must display an error message if the entered credentials are invalid.
REQ-4: User must be able to register by providing the required information such as name, email, username, and password.
REQ-5: System must verify the information provided during registration and register the user if the information is valid.
REQ-6: System must display an error message if the information provided during registration is invalid.
REQ-7: Password must be encrypted and stored securely in the database.
REQ-8: System must provide password reset functionality for users who have forgotten their password.

3.2 Movie Details

3.2.1 Description and Priority:

- Movie Details feature for users to view information about a specific movie.
- Medium priority as it is important for the user experience but not essential for the system to function properly.
- Benefit: 7 (provides users with important information about the movie)
- Penalty: 4 (limited impact on the system if the feature is not available)
- Cost: 5 (implementation may require some time and resources)
- Risk: 2 (minimal risk involved with implementing the feature)

3.2.2 Stimulus/Response Sequences:

- User selects a movie from the list of available movies.
- System displays the movie details page.
- User views the movie details such as title, release date, director, actors, description, and ratings.
- User has the option to add the movie to their watchlist or mark it as seen.

3.2.3 Functional Requirements:

REQ-1: System must display detailed information about a specific movie when selected by the user.
REQ-2: Movie details must include title, release date, director, actors, description, and ratings.
REQ-3: System must provide the option for users to add a movie to their watchlist.
REQ-4: System must provide the option for users to mark a movie as seen.
REQ-5: System must allow users to sort and filter the list of available movies based on various criteria such as genre, release date, and ratings.
REQ-6: System must display error messages in case of any invalid inputs or missing information about the movie.

3.3 Movie List

3.3.1 Description and Priority:

- High priority
- This feature allows users to view a list of movies in a centralized location.

3.3.2 Stimulus/Response Sequences:

- User selects "Movie List" option from main menu
- System displays a list of movies with movie titles, genres, and release dates
- User can select a movie to view its details

- System displays the selected movie's details including synopsis, cast, and ratings

3.3.3 Functional Requirements:

REQ-1: The system must be able to retrieve movie information from a database and display it in a list format.
REQ-2: The system must allow users to filter the movie list by genre.
REQ-3: The system must allow users to search for a specific movie by title.
REQ-4: The system must display the movie's title, genre, and release date for each movie in the list.
REQ-5: The system must allow users to view a selected movie's details by clicking on it.

3.4 System Recommendation

3.4.1 Description and Priority

- High priority

3.4.2 Stimulus/Response Sequences

- User selects the "Get Recommendations" option
- System displays a list of recommended movies based on the user's viewing history, ratings, and preferences
- User can choose to view more details about a movie by selecting it
- System displays the movie details
- User can rate a movie or add it to their watchlist

3.4.3 Functional Requirements

REQ-1: User must be logged in to access recommendations
REQ-2: System must generate recommendations based on user's viewing history, ratings, and preferences
REQ-3: System must display a list of recommended movies
REQ-4: System must allow users to view movie details
REQ-5: System must allow users to rate movies and add them to their watchlist
REQ-6: System must update recommendations based on user's updated ratings and watchlist.

3.5 User Management and Admin Management

3.5.1 Description and Priority

- High priority.

3.5.2 Stimulus/Response Sequences

- User signs up for an account: System displays a sign-up form and prompts for required information (username, email, password, etc.).
- User logs in: System prompts for username and password, and if correct, displays the user dashboard.
- User updates profile information: System displays a form to update profile information (name, address, email, etc.) and saves the changes.
- User requests password reset: System prompts for email address associated with the account, and sends a password reset link to the email.
- Admin adds a new user: System displays a form to enter new user information and saves the changes.
- Admin updates user information: System displays a form to update user information and saves the changes.
- Admin deletes a user: System prompts for confirmation and removes the user from the system.

3.5.3 Functional Requirements

REQ-1: User sign-up must require unique username and email address.
REQ-2: User login must use secure password authentication.
REQ-3: User profile information must be editable by the user.
REQ-4: Password reset must be initiated by a request from the user's email address.
REQ-5: Admin must have the ability to add, update, and delete users.
REQ-6: Admin changes to user information must be audited and logged.
REQ-7: The system must enforce password policies, such as minimum length and complexity.
REQ-8: The system must provide a secure way to store and retrieve user information, such as using encryption or hashing.

3.6 Movie Trailer

3.6.1 Description and Priority:

- High priority. Benefit: 8, Penalty: 2, Cost: 5, Risk: 4

3.6.2 Stimulus/Response Sequences:

- User clicks on "watch trailer" button for a movie.
- System displays the movie trailer.
- User closes the trailer.
- System returns to movie details page.

3.6.3 Functional Requirements:

REQ-1: System should have the ability to play trailers for multiple movies.
REQ-2: System should provide high-quality and accurate trailers for the movies.
REQ-3: System should allow users to watch trailers in full screen mode.
REQ-4: System should provide controls for pausing, rewinding, and fast-forwarding the trailer.
REQ-5: System should allow users to share the trailer link through social media or email.
REQ-6: System should handle errors and display appropriate messages when the trailer is not available or cannot be played.

4. External Interface Requirements

4.1 User Interfaces

The user interface (UI) is the way in which individuals interact with a digital device or software. This interaction is made possible through visual and interactive components that enable the input and retrieval of information. Effective UI design can boost user productivity and efficiency, as well as improve the overall image and perception of a brand. In essence, the UI acts as the bridge between users and digital products, and well-designed interfaces can positively impact user experience and brand reputation.

The user interfaces used for this project :

- React JS

  React JS is a JavaScript library utilized for constructing user interfaces. With React, developers can create UI components that can be reused, thereby increasing consistency and streamlining the development process. React employs a virtual DOM which facilitates the efficient updating of the user interface without the need to refresh the whole page. This results in improved performance for web applications created using React.

- Tailwind CSS

  Tailwind CSS is a utility-first CSS framework. It provides a set of pre-designed CSS classes that can be used to quickly build custom user interfaces. Developers can use

the pre-designed classes to quickly build custom interfaces, rather than having to write custom CSS code from scratch. It offers speed, customization, and a wide range of design options, making it a popular choice for web development projects.

4.2 Hardware Interfaces

The hardware interface in a recommendation system can impact its overall efficiency and performance. It's crucial to consider the supported devices, data interactions, and control interactions between hardware and software when designing the interface. Communication protocols must also be considered. For instance, if the system will be integrated with mobile devices, compatibility with the device's operating system and available communication protocols must be considered. If the system is meant for a large organization, it should be designed to work with the current IT infrastructure, including servers, databases, and networks. The hardware's physical characteristics, like memory capacity, processing power, and storage, should also be taken into account to guarantee the system operates smoothly and efficiently.

The hardware interfaces used for this project:

- Windows

  Windows is a Microsoft-made operating system (OS) commonly used on personal computers and other computing devices like laptops and tablets. It offers a graphical user interface (GUI) for users to interact with their computer through the use of input devices such as mouse and keyboard. The OS functions as a platform to run applications and manage data like files and folders.

4.3 Software Interfaces

A software interface acts as a communication point between two software systems. It outlines the way in which one software component interacts and exchanges information with another. The interface outlines the input/output specifications and accessible methods/functions between components. This separation of implementation details allows

components to work together seamlessly, making it possible to replace or update one component without affecting the others, promoting a modular and adaptable software design. Examples of software interfaces include APIs, GUIs, and web service interfaces.

The software interface used for this project :

Software	Justification
Visual Studio 2022	Visual Studio is a popular integrated development environment (IDE) that provides a comprehensive set of tools for developing software, including a code editor, debugger, and integrated tools for testing, deployment, and collaboration, making it a versatile platform for efficiently creating and managing projects.
Microsoft SQL Server 2019(MSSQL)	Microsoft SQL Server 2019 is a relational database management system (RDBMS) developed by Microsoft. It provides a variety of features such as data storage, data retrieval, and data manipulation. MSSQL supports multiple programming languages including T-SQL, .NET, and Python. It also provides built-in security features such as encryption, auditing, and backup/recovery. MSSQL can be deployed on-premises, in the cloud, or as a hybrid solution.

4.4 Communications Interfaces

The Communications Interfaces for the system recommendation movie application are as follows:

- API-based communication: The system will communicate with other movie databases through API to retrieve the latest movie information.
- Data Transfer: The system will transfer user data, movie data and other relevant data between the client-side and the server-side.
- Network Server Communication: The system will communicate with the network server using standard communication protocols such as HTTP or HTTPS to ensure secure data transfer.

- Data Security: The system will use encryption methods such as SSL or TLS to secure the data transfer between the client and server and protect sensitive user data.

- Synchronization: The system will use synchronization mechanisms to ensure that the user's movie recommendations are updated in real-time.

- Web Browsing: The system will support web browsing for users to access movie trailers, reviews and other relevant information.

- E-mail notifications: The system will send e-mail notifications to users regarding new movie releases, recommendations, and other updates.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The recommendation system should provide movie suggestions within 2 seconds of user input.
- The recommendation system should be able to handle a maximum of 100 concurrent users without any significant drop in performance.
- The recommendation system should have a 99% uptime.

5.2 Safety Requirements

- The recommendation system should not show inappropriate or explicit content to minors.
- The recommendation system should implement data protection measures to ensure the privacy of user data.

5.3 Security Requirements

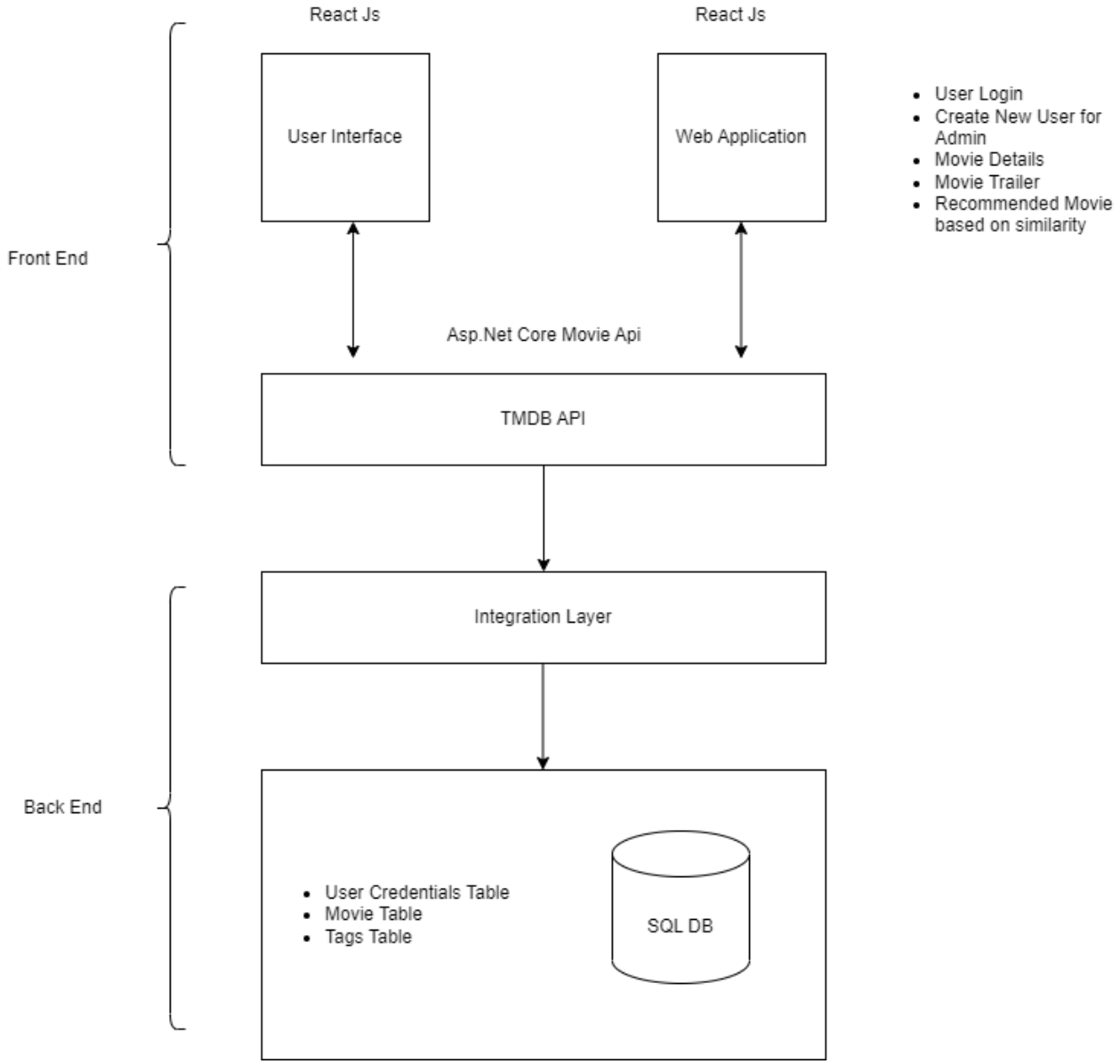
- The recommendation system should implement a secure login process with password encryption.
- The recommendation system should comply with relevant data protection laws and regulations.
- The recommendation system should have regular security updates and vulnerability scans.

5.4 Software Quality Attributes

- The recommendation system should have a user-friendly interface for ease of use.
- The recommendation system should have a flexible recommendation algorithm to adapt to changing user preferences.
- The recommendation system should be maintainable, with clear and concise code for future updates and bug fixes.

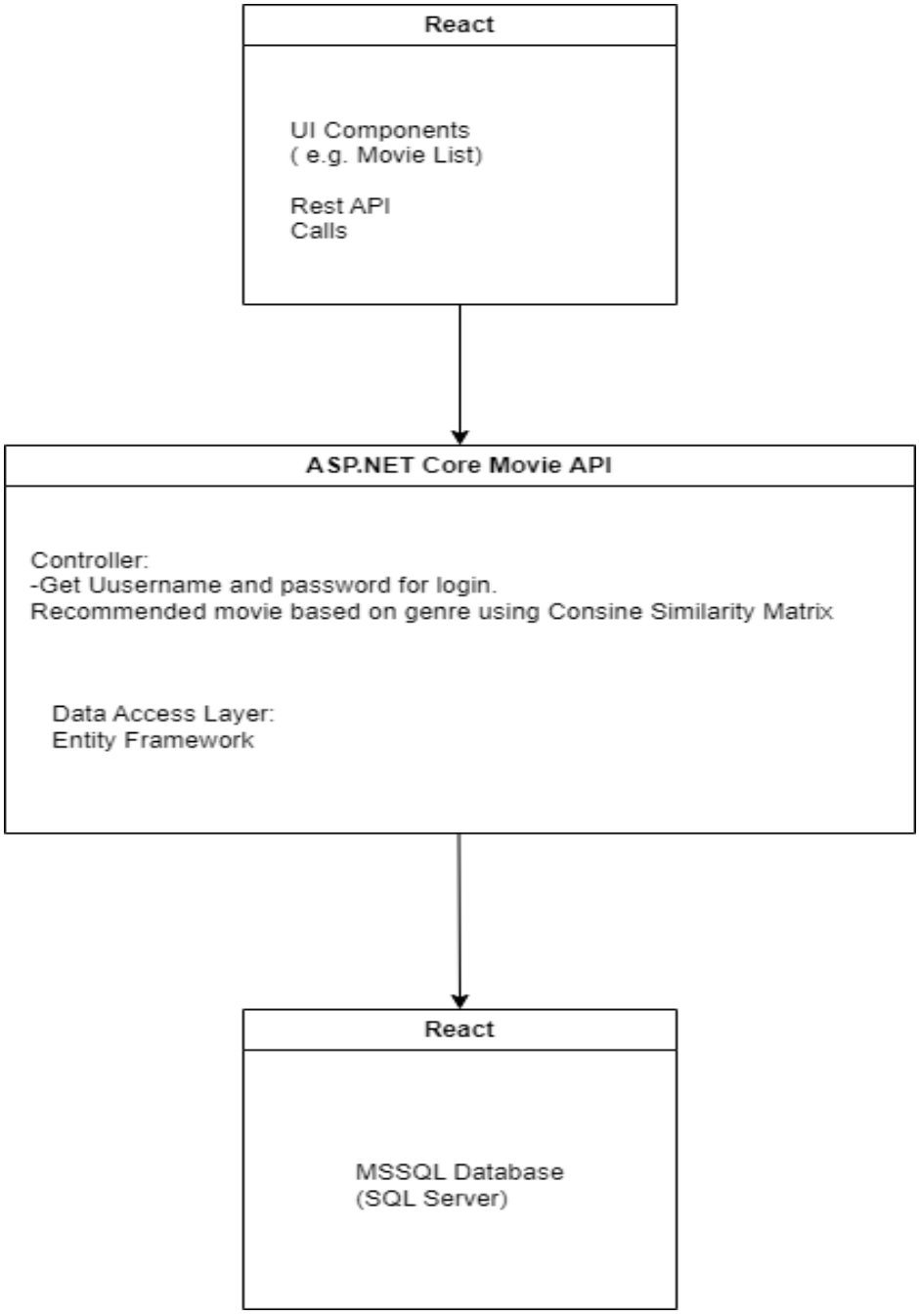
Appendix A: High Level Design

High level design is a conceptual design of a system that provides an overview of the system's architecture and major components, and defines the relationships between them. It is a high-level view of the system that provides a blueprint for the detailed design and implementation. It identifies major modules, data structures, interactions between the modules, and key algorithms that will be used to implement the system. The high-level design serves as a foundation for the more detailed design, testing and implementation phases of the project.



Appendix B: Low Level Design

Low-level design is a stage in the software development process that details the implementation of a system's components and data structures. It includes the design of algorithms, data structures, and interfaces for individual software modules, as well as the design of the interactions between modules and how they will be connected and integrated to form the complete system. The goal of low-level design is to produce a detailed plan that defines how the software will be implemented, tested, and integrated, and to ensure that the final product meets the requirements and quality attributes specified in the high-level design.


React
UI Components
(e.g. Movie List)
Rest API
Calls
ASP.NET Core Movie API
Controller:
-Get Uusername and password for login.
Recommended movie based on genre using Consine Similarity Matrix
Data Access Layer:
Entity Framework
React
MSSQL Database
(SQL Server)