Software Requirements Specification

for

Hello Movie

Version 1.0

Prepared by Muhammad Harith Adam Bin Zolkafli

28/1/2023

Table of Contents

1. Introduction

1.1 Purpose

This document details the project plan for the development of “Hello Movie”. It is intended for developers, designers, and testers working on “Hello Movie” as well as project investors. This plan will include a summary of:

- How the system will function
- The scope of the project from the development viewpoint
- The technology used to develop the project, and
- The metrics used to determine the project’s progress
- Overall description

This document outlines the requirements for a program that utilizes API data from two sources: the TMDB API and a custom API built using ASP.NET Web API. The program will utilize data from these two APIs to provide users with a comprehensive and seamless experience. The purpose of this document is to provide a clear understanding of the program's functionality, constraints, and overall requirements

1.2 Intended Audience and Reading Suggestions

The intended audience for this program is the general public, specifically individuals who are interested in movies. This includes movie fans, casual movie-goers, and anyone else who is interested in discovering new movies and learning more about existing movies.

The intended use of this program is to provide users with a comprehensive and seamless experience when searching for movies. This includes the ability to search for movies using various criteria, view movie details and information, and access data from multiple sources. The program will utilize data from the TMDB API and a custom API built using ASP.NET Web API to provide users with a wide range of information and functionality.

1.3 Project Scope

This project will be a very simple website about movie details and rating for general use. This website will be designed to use TMDB-API. Moreover, this system is designed to allow user to view any movie details according to their preferences and they also can view the trailer for the movie.

In addition to utilizing data from the TMDB API, this project will also incorporate data from a custom ASP.NET Web API. This will provide users with additional information and functionality beyond what is available from the TMDB API alone. The ASP.NET Web API will be used to supplement the information available from the TMDB API and provide a more comprehensive experience for the users.

1.4 References

The program will use several external data sources to provide users with a comprehensive movie-watching experience. These data sources include:

1. Movie Lens Dataset: https://grouplens.org/datasets/movielens/
2. TMDBAPI: https://www.themoviedb.org/

To provide personalized movie recommendations to users, the program will use several recommendation algorithms, including:

3. Collaborative Filtering: https://en.wikipedia.org/wiki/Collaborative_filtering
4. Pierson Correlation coefficient: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
5. Jaccard Index: https://en.wikipedia.org/wiki/Jaccard_index
6. Cosine Similarity: https://analyticsindiamag.com/cosine-similarity-in-machine-learning/

These algorithms will be used to recommend movies to users based on their ratings and preferences. The program will also provide options to users to customize the recommendation according to their preferences.

2. Overall Description

2.1 Product Perspective

The product perspective of the "Hello Movie" project is focused on providing a comprehensive and seamless platform for movie information and trailers. The product will be designed to meet the needs of users who are looking for an easy-to-use and informative platform for movie information. The product will be developed using the latest technologies, including the TMDB API, ASP.NET Web API, and React.js, to ensure a smooth and efficient user experience.

The product will also provide robust features, including user login, movie information, trailer videos, and a search functionality, to meet the needs of users and provide them with a comprehensive and enjoyable experience. The product will be developed with the goal of being accessible and user-friendly, and it will be tested to ensure that it meets the needs and expectations of its users. Overall, the product perspective of the "Hello Movie" project is focused on delivering a high-quality and reliable platform for movie information and trailers.

2.2 Product Features

The "Hello Movie" product features are designed to meet the needs and expectations of users who are looking for a comprehensive and user-friendly platform for movie information and trailers. These features include:

1. User Login: The system will provide users with the ability to log in and access their personal profile, which will store their movie history and preferences.
2. Movie Information: Users will be able to view information about movies, including details such as title, release date, cast, and plot synopsis.
3. Trailer Videos: Users will be able to view trailers for movies that are currently showing or have been recently released.
4. Search Functionality: The system will provide a search functionality that allows users to search for movies based on different criteria, such as title, genre, release date, or cast.

5. Admin Control: The system will provide admin control, allowing admins to add, update, or delete movies in the system, as well as manage user accounts.
6. User-Friendly Interface: The system will be designed with a user-friendly interface to provide a seamless and enjoyable experience for users.
7. Technical Efficiency: The system will be developed using the latest technologies, including the TMDB API, ASP.NET Web API, and React.js, to ensure a smooth and efficient user experience.

These product features are crucial to the success of the "Hello Movie" project and will provide users with a comprehensive and enjoyable platform for movie information and trailers.

2.3 User Classes and Characteristics

The "Hello Movie" system has one main user classes:

1. Users: There is 2 user type which is Administrator and Normal user. Normal user are individuals who use the system to search for movie information and trailers. They have basic privileges, such as the ability to view movie details and trailers, save their favorite movies, and personalize their profile. Administrators are individuals who have additional privileges, such as the ability to add, update, or delete movie information, manage user accounts, and access administrative reports.

User class have the following characteristics:

1. Accessibility: Both normal users and administrators can access the system from any location with an internet connection.
2. Technical Proficiency: Users should have a basic understanding of how to use a computer and navigate websites.
3. Movie Interest: Both normal users and administrators are expected to have an interest in movies and be actively seeking movie information and trailers.
4. User Experience: Both normal users and administrators expect a seamless and user-friendly experience when using the "Hello Movie" system.

By targeting this user class and its characteristics, the "Hello Movie" system aims to provide a comprehensive and user-friendly platform for movie information and trailers.

2.4 Design and Implementation Constraints

The design and implementation of the "Hello Movie" system are subject to the following constraints:

1. Technology: The system must be developed using technologies such as ASP.NET Web API, Microsoft SQL Server, React.js, and CSS.
2. Data Accuracy: The system must ensure the accuracy of the data obtained from the TMDB API and the custom API.
3. Data Security: The system must protect sensitive information, such as user login credentials.
4. User Experience: The system must be designed to provide a user-friendly and seamless experience for both normal users and administrators.
5. Performance: The system must be designed to perform efficiently, with fast load times and minimal downtime.
6. Scalability: The system must be scalable to accommodate future growth and increased user traffic.

By taking these constraints into consideration during the design and implementation phase, the "Hello Movie" system can be developed to meet the needs of its users and stakeholders while maintaining a high level of functionality and security.

2.5 User Needs

The user needs for this project are centered around providing a comprehensive and user-friendly platform for movie information and trailers. Users should have the ability to view information about movies, including details such as movie title, release date and synopsis. They should also be able to view trailers for movies that are currently showing or have been recently released.

To enhance their experience, users should be able to log in to the system and access their personal profile. Additionally, the system should provide users with a search functionality that

allows them to search for movies-based title. Finally, the system should provide admin control, allowing admins to add, update, or delete movies in the system, as well as manage user accounts. These user needs are crucial to the development of a system that provides a positive experience for users and meets their expectations.

2.6 User Documentation

2.6.1 User Manual

1. Launch the program either by visiting the URL where the front-end program is hosted, or by running the front-end program locally (if you have followed the installation instructions).
2. The program's main screen displays a list of popular movies, along with their posters and a brief summary.
3. You can search for a specific movie by using the search bar at the top of the screen.
4. To view more details about a movie, simply click on its poster. This will show you the details of the movie at the top of the page with its big poster.
5. That's it! Enjoy exploring the world of movies with "Hello Movie".

2.6.2 Installation Manual

This is your installation manual to run the program:

1. Download the code from the Github repository.
2. Install Microsoft SQL Server and Microsoft Visual Studio or Microsoft Visual Code.
3. Change the connection string in the program.cs file and the MovieDbContext class to match your local SQL Server.
4. Open the Nuget Package Manager Terminal and run the following commands:
    a. “Add-migration ‘Migration Name’” to add the necessary migrations to the database.
    b. “Update-database” to update the database with the latest changes.
5. Run the ASP.NET program to start the API.
6. To run the front-end program, navigate to the front-end directory in the terminal and run “npm start”.
7. After the front-end program is up and running, you should be able to access the "Hello Movie" application.

Make sure to follow these steps carefully to ensure the program is installed and run correctly.

2.7 Assumptions and Dependencies

Assumptions:

- The program assumes that the user has internet connectivity to access the external data sources.
- The program assumes that the user has a web browser that supports HTML, CSS, and JavaScript to use the program's user interface.
- The program assumes that the API keys for the TMDBAPI and the API that you've created using asp.net web API are correctly configured and accessible.
- The program assumes that the data from the external data sources is accurate and up-to-date.

Dependencies:

- The program depends on the availability and reliability of the external data sources (Movie Lens Dataset and TMDBAPI).
- The program depends on the availability and reliability of the API keys for the TMDBAPI and the API that you've created using asp.net web API.
- The program depends on the React.js, CSS, and Bootstrap for the front-end development.
- The program depends on the asp.net web API for the back-end development.

3. System Features

This section outlines the use cases for each of the users.

3.1 System Feature 1

User Login

3.1.1 Description and Priority

The User Login feature allows users to create an account and log in to the system. When a user visits the program's main screen, they are prompted to log in if they

have not already. After entering their username and password, the user is logged into the system and can access the program's features.

3.1.2 Stimulus/Response Sequences

1. User visits the program's main screen.
2. User clicks the "Sign In" button.
3. User enters their username and password.
4. User clicks the "Log In" button.
5. Program verifies the user's credentials and logs them into the system.

3.1.3 Functional Requirements

Feature	User Login
Trigger	User clicks the "Login" button on the user interface.
Precondition	User must have a registered account in the system.
Basic Path	a. User inputs their registered username and password into the login form. b. System processes the user's inputs and checks for a match with the registered account information in the database. c. If the information matches, the system grants the user access to the system and displays the user dashboard. d. If the information does not match, the

	system displays an error message indicating the login information is incorrect.
Alternative Paths	If the user forgets their password, they can click the "Forgot Password" link to initiate the password reset process.
Postcondition	User is able to access the system and use its features.
Exception Paths	If the database connection is lost or an error occurs during the login process, the system displays an error message and provides options for the user to resolve the issue.

3.2 System Feature 2

Admin Login

3.1.1 Description and Priority

The Admin Login feature allows administrators to log in to the system and access administrative functions. Administrators can log in by visiting the program's main screen and clicking the "Admin Login" button. After entering their username and password, the administrator is logged into the system and can access the administrative functions.

3.1.2 Stimulus/Response Sequences

1. Administrator visits the program's main screen.
2. Administrator clicks the "Admin Login" button.
3. Administrator enters their username and password.

4. Administrator clicks the "Log In" button.

5. Program verifies the administrator's credentials and logs them into the system.

3.1.3 Functional Requirements

Feature	Admin Login
Trigger	Admin clicks on the "Login" button on the user interface.
Precondition	User must have an existing admin account.
Basic Path	a. User enters their admin username and password into the provided fields. b. System verifies the entered information against the database. c. If the information is correct, the system logs the user in as an admin and redirects them to the admin dashboard. d. Admin can perform various administrative tasks such as creating, updating, and deleting users.
Alternative Paths	If the entered information is incorrect, the system displays an error message and prompts the user to try again.
Postcondition	User is successfully logged in as an admin and has access to administrative tasks.

Exception Paths	If the database connection is lost or an error occurs during the login process, the system displays an error message and provides options for the user to resolve the issue.

3.3 System Feature 3

User Management

3.1.1 Description and Priority

The User Management feature allows administrators to create, update, and delete user accounts. Administrators can access this feature by logging into the system and will direct to Admin page. From this screen, administrators can add new users, edit existing users, and delete users. This feature is of high priority for the program, as it is essential for managing user accounts and ensuring the security of user data.

3.1.2 Stimulus/Response Sequences

1. Administrator logs into the system.
2. Administrator clicks the "User Management" button.
3. Administrator selects an action (create, update, or delete).
4. Administrator enters the required information for the selected action.
5. Administrator clicks the "Update/Register" button.
6. Program performs the selected action and updates the user account information.

3.1.3 Functional Requirements

Feature	User Management
Trigger	Admin accesses the User Management section of the system.

Precondition	Admin must be logged in to the system.
Basic Path	a. Admin views the list of current users in the system. b. Admin selects an option to create a new user, update an existing user, or delete a user. c. If creating a new user, the admin inputs the new user's information, including username and password. d. If updating an existing user, the admin selects the user to update and makes the necessary changes. e. If deleting a user, the admin selects the user to delete and confirms the action. f. System updates the user database with the changes made by the admin.
Alternative Paths	If the admin attempts to delete a user but there is an error in the process, the system displays an error message and provides options for the user to resolve the issue.
Postcondition	User database is updated with the changes made by the admin.
Exception Paths	If the database connection is lost or an error occurs during the user management process,

	the system displays an error message and provides options for the admin to resolve the issue.

3.4 System Feature 4

User List

3.1.1 Description and Priority

The User List feature allows administrators to view a list of all users registered in the system. Administrators can access this feature by logging into the system and will redirect to Admin page. From this screen, users can view a list of all registered users, along with their usernames and password. This feature is of medium priority for the program, as it provides valuable information to both administrators and users.

3.1.2 Stimulus/Response Sequences

1. Administrator logs into the system.
2. Administrator clicks the "User List" button.
3. Program displays a list of all registered users.
4. Administrator selects a user from the list to view their details.
5. Program displays the selected user's information.

3.1.3 Functional Requirements

Feature	User Lists
Trigger	Admin logged into the system.
Precondition	Admin must be logged in to the system.
Basic Path	a. System retrieves the user's movie lists

	from the TMDAPI. b. System displays the user's movie lists in a list format on the user interface. c. User selects a movie list from the displayed lists. d. System displays the movies in the selected list.
Alternative Paths	If the user has not created any movie lists, the system displays a message indicating there are no lists available.
Postcondition	User is able to view the movies in their selected list.
Exception Paths	If the database connection is lost or an error occurs during the retrieval of the user's movie lists, the system displays an error message and provides options for the user to resolve the issue.

3.5 System Feature 5

Movie List

3.1.1 Description and Priority

The Movie List feature allows users to view a list of popular movies, along with their posters and a brief summary. Users can access this feature by logging into the system and will redirect to movie page. From this screen, users can view a list of popular movies and select a movie to view additional information. This feature is of high priority for the program, as it provides users with a convenient way to discover new movies.

3.1.2 Stimulus/Response Sequences

1. User logs into the system.
2. Program will redirect User to Movie page.
3. Program displays a list of popular movies.
4. User selects a movie from the list to view additional information.
5. Program displays the selected movie's details.

3.1.3 Functional Requirements

Feature	Movie List
Trigger	User logged into the system.
Precondition	User must be logged in to the system.
Basic Path	a. System retrieves the list of movies from the TMDBAPI. b. System displays the movie list, including movie title, release date, and rating. c. User selects a movie from the movie list. d. System displays the details of the selected movie, including its description, trailer video, and rating.
Alternative Paths	If there are no movies in the database, the system displays a message indicating no

	movies are available.
Postcondition	User is able to view the details of the selected movie.
Exception Paths	If the database connection is lost or an error occurs during the retrieval of the movie list, the system displays an error message and provides options for the user to resolve the issue.

3.6 System Feature 6

Movie Details

3.1.1 Description and Priority

The Movie Details feature allows users to view additional information about a movie, including its trailer, and release date. Users can access this feature by selecting a movie from the movie list. From this screen, users can view a detailed summary of the movie, along with information about the trailer and release date. This feature is of high priority for the program, as it provides users with a comprehensive view of a movie and its reception.

3.1.2 Stimulus/Response Sequences

1. User selects a movie from the movie list.
2. Program displays the selected movie's details.
3. User views the movie's trailer, and release date.

3.1.3 Functional Requirements

Feature	Movie Details

Trigger	User selects a movie from the movie list or from the search results.
Precondition	User must be logged in to the system.
Basic Path	a. System retrieves the details of the selected movie from the TMDBAPI. b. System displays the details of the selected movie, including its title, description, trailer video, and rating.
Alternative Paths	If the selected movie is not found in the database, the system displays a message indicating the movie was not found.
Postcondition	User is able to view the details of the selected movie.
Exception Paths	If the database connection is lost or an error occurs during the retrieval of movie details, the system displays an error message and provides options for the user to resolve the issue.

3.7 System Feature 7

Trailer Video

3.1.1 Description and Priority

The Trailer Video feature allows users to view the trailer of a movie. Users can access this feature by selecting a movie from the movie list. From the movie details screen, users can click the "Watch Trailer" button to view the movie's trailer. This

feature is of high priority for the program, as it provides users with a convenient way to preview a movie.

3.1.2 Stimulus/Response Sequences

1. User logs into the system
2. Program will redirect User to movie page.
3. System displays a list of all available movies
4. User selects a movie
5. System displays detailed information about the selected movie, including the trailer video.
6. User click play trailer video button.
7. System will play the trailer.

3.1.3 Functional Requirements

Feature	Trailer Video
Trigger	User selects to view the trailer video for a movie from the movie details page.
Precondition	User must be logged in to the system and have selected a movie from the search results or movie list.
Basic Path	a. System processes the request and retrieves the trailer video for the selected movie. b. System displays the trailer video on the movie details page.

	c. User can play/pause the video and adjust the volume as needed.
Alternative Paths	If the trailer video is unavailable or an error occurs during retrieval, the system displays a message indicating the video is not available.
Postcondition	User is able to view the trailer video for the selected movie.
Exception Paths	If the user's internet connection is lost or an error occurs during video playback, the system displays an error message and provides options for the user to resolve the issue.

3.8 System Feature 8

Search Functionality

3.1.1 Description and Priority

The Search Functionality is a feature that allows users to search for specific movies within the TMDBAPI. This feature enables users to find the desired movie quickly and easily by entering keywords related to the movie title. The search results are displayed in a list format and include relevant movie details such as title. The user can then select a movie from the search results to view its detailed information, including its description, trailer video, and rating. This feature is important for users to easily access information about movies and helps improve their overall experience with the system.

3.1.2 Stimulus/Response Sequences

1. User inputs search keyword into the search bar on the user interface.

2. The system processes the request and send API request to find for any matches with the keyword.

3. The system displays the results of the search, including any relevant movie title, description and release date.

4. User selects a movie from the search results to view its details.

5. The system displays the details of the selected movie, including its description, trailer video, and rating.

3.1.3 Functional Requirements

Feature	Search Functionality
Trigger	User inputs search keyword into the search bar on the user interface.
Precondition	User must be logged in to the system.
Basic Path	a. System processes the request and searches the TMDBAPI for any matches with the keyword. b. System displays the results of the search, including any relevant movie titles, release date and description. c. User selects a movie from the search results. d. System displays the details of the selected movie, including its description, trailer video, and rating.

Alternative Paths	If there are no results for the search keyword, the system displays a message indicating no results were found.
Postcondition	User is able to view the details of the selected movie.
Exception Paths	If the database connection is lost or an error occurs during the search process, the system displays an error message and provides options for the user to resolve the issue.

4. External Interface Requirements

4.1 User Interfaces

- Front-end software: React.js, CSS and Bootstrap
- Back-end software: ASP.NET Core Web API

4.2 Hardware Interfaces

- The application will communicate with various hardware devices over a network using the TCP/IP protocol.
- The application will be able to receive data from and send commands to the hardware devices via a serial connection.

4.3 Software Interfaces

- The web application will interact with a MSSQL database to store and retrieve data.
- The application will use the REST API to communicate with external systems.

- The application will be able to receive and process JSON data format.
- The application will be capable of handling and responding to errors and exceptions that may occur during communication with external systems.
- The application will provide detailed logs of all communication with external systems for debugging and troubleshooting purposes.
- The application will be designed to handle high-volume traffic and to scale horizontally as needed.
- The application will have a retry mechanism to handle failed requests.

4.4 Communications Interfaces

- The web application will be accessible via a web browser on desktop.
- The application will support the latest versions of Google Chrome, Internet Explorer, and Mozilla Firefox.
- The application will be designed to adapt to different screen sizes and resolutions, and to be usable on both desktops.
- The minimum browser version supported is Chrome v90, Firefox v85, Internet Explorer v11.
- The application will make use of responsive design techniques to ensure that the user interface is optimized for different screen sizes and resolutions.
- The application will use modern web technologies such as HTML5, CSS3, and JavaScript.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system must be able to handle at least 1 concurrent user without any significant degradation in performance.
- The system must have a response time of less than 3 seconds for all critical functions.
- The system must have an availability rate of at least 99.9%

- The system must be able to handle large amount of data and process it in a reasonable amount of time.
- The system must be able to handle data input/output without losing any data.

5.2 Security Requirements

- The user needs to access the project through HTTPS
- The project directory is not revealed

6. Flow Diagrams

6.1 High Level diagram