Software Requirement Specification (SRS) Document

Recommendation Engine for Media and Entertainment Platform

By: -Team 2

Under Guidance of Abinash Sahoo

TEAM MEMBERS (Team 2)

- Ashutosh
- Lahari
- Varshini
- Mayank
- Mohana
- Pratish
- Sai
- Sanika
- Veera
- Vivek
- Vaishnavi
- Pratibha

INDEX

1. Purpose

The aim of this project is to suggest related contents based on different users hobbies, habits and likings while surfing over media and entertainment platforms. This recommendation system on media and entertainment platform aims to hook audience by suggesting them content based on their liking.

2. Scope

The scope of this project is to allow the client of media/entertainment platform like Amazon Prime / Netflix / YouTube etc… to suggest the media/entertainment content of the platform to the individual person based on their liking of the content and similarity of the content.

3. Project Description

Personalized Content Recommendation is the main aim of the project for the clients of media and entertainment platform to use and improve the content showing mechanism for audience accessing the platform. This project will be able to demonstrate the same replica for the way to personalize the content for the user based on certain attributes.

4. Project Strategy

The project aims to develop a sophisticated recommendation system tailored to individual user's preferences, habits, and interests on media and entertainment platforms. By leveraging user data and content similarity algorithms, the system will enhance user engagement by suggesting relevant content aligned with their tastes. This personalized approach seeks to captivate users and increase their satisfaction with the platform's offerings. Implement a recommendation system capable of analysing user behaviour, preferences and past interactions with content.

5. Tools and Technology used

Technologies that will be used are Python, HTML, CSS, JavaScript, React framework. These technologies will collaborate together to build a platform for the making of recommendation system. A visualization tool such as Power BI will also be needed to visualize the insights obtained during the project creation phase.

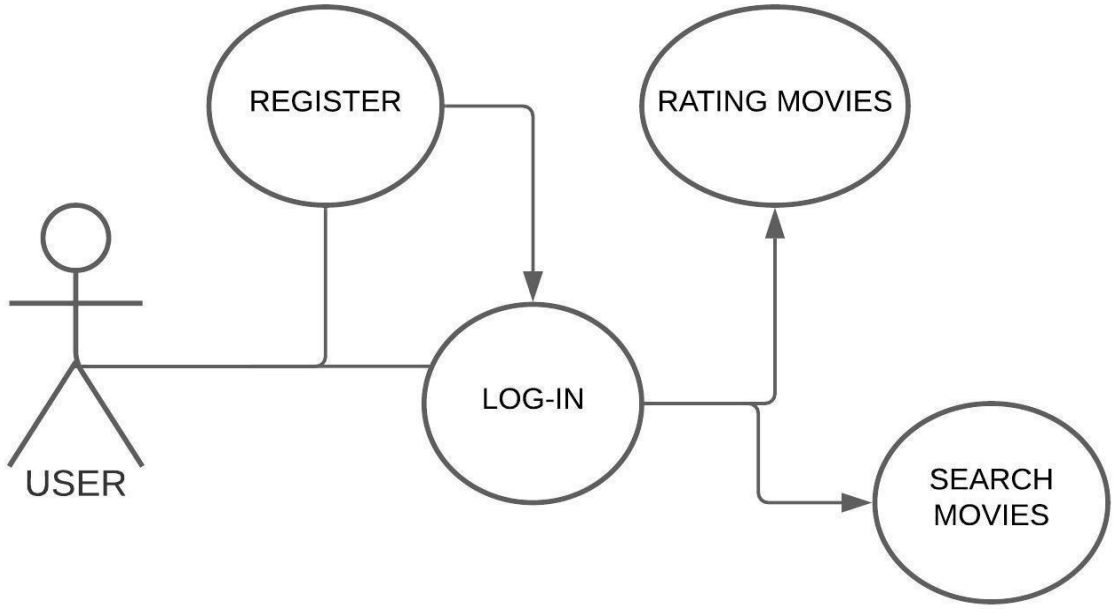
6. Flowchart application:

User Characteristics

The user is expected to be Internet literate and be able to use a search engine. The main screen of the Recommendation Website will have the search function and a link to add new content for the content providers.

The Content provider is expected to be Internet literate, have login credentials to navigate through the website, and be able to log in and have full details about the content to be added so that the recommendation system works optimally.

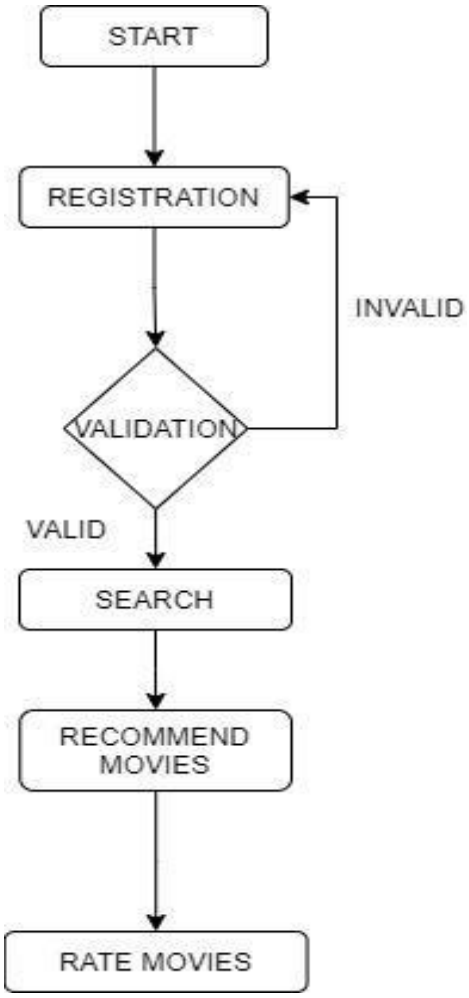
User Use Case Diagram



Functional Requirements Specification:

This section outlines the functionality of each actor in this system.

Registration For New User
Activity Diagram For New User



REQUIREMENT-1

If the User is new to the site, he first registers himself then searches for the movies he wants and then gives likes or dislikes, which helps create a profile for the user based on his likes/dislikes and the content of movies he likes; the system will recommend movies to the user.

REQUIREMENT-1.1

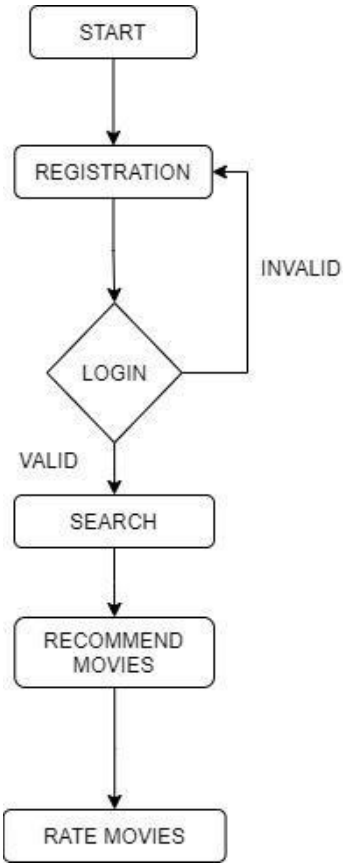
INPUT: “REGISTER” option selected by the user
OUTPUT: User prompted to enter personal data for registration

REQUIREMENT-1.2

INPUT: The user enters registration data
OUTPUT: User redirected to the home page with the search engine
PROCESSING: Check if the input values are in the correct format as needed

Recommendations For Existing User:

Activity Diagram for Existing User



REQUIREMENT-2

If the user is an existing one, he logs into the website. Based on what he searches in the search box and based on his likes/dislikes and the content of movies he likes we will recommend movies to the user.

REQUIREMENT-2.1

INPUT: "LOG-IN" option selected by the user
OUTPUT: User prompted to enter USER NAME and PASSWORD.

REQUIREMENT-2.2

INPUT: Enter USER NAME and PASSWORD OUTPUT
If the credentials are correct, then redirect to the home page with the search engine. If the credentials are wrong, it shows a prompt of wrong credentials and asks to re-enter the credentials.

REQUIREMENT-2.3

INPUT: "search" option,
OUTPUT: User prompted to enter "MOVIE" name.

REQUIREMENT-2.4

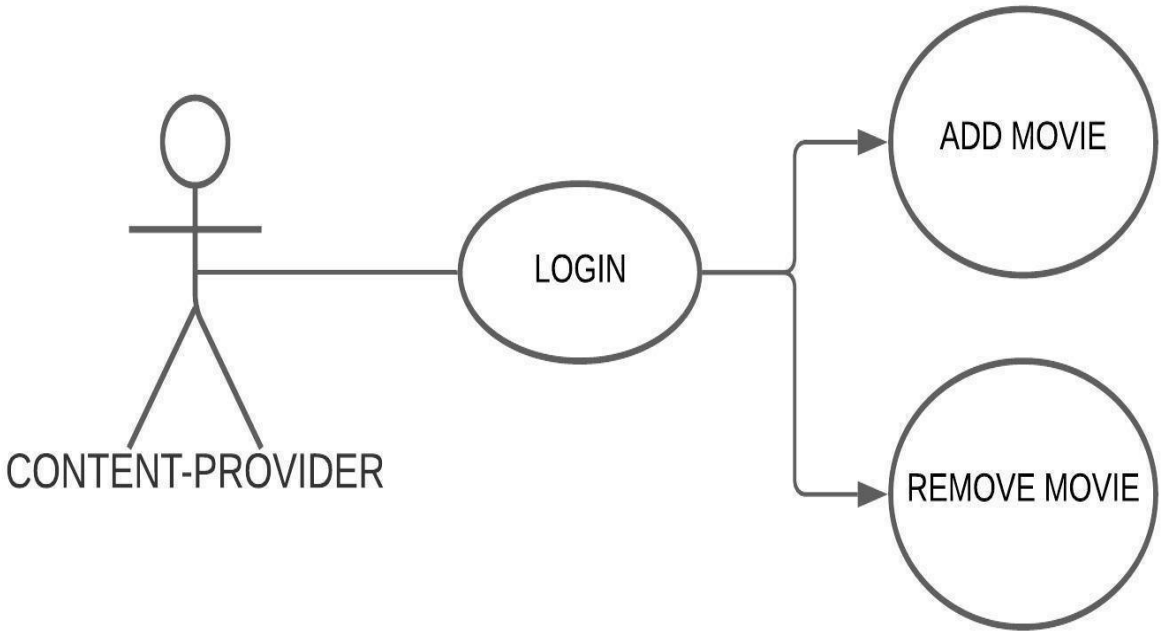
INPUT: "MOVIE NAME"
OUTPUT: Details about the movie if present in dataset and related recommendations of other movies.
PROCESSING: If the user is logged in, then based on collaborative and content filtering, new movies are recommended if not based only on content filtering, movies are recommended.

REQUIREMENT-2.5

INPUT: LIKE / DISLIKE MOVIE
OUTPUT: The movie rating is added to the user profile.

Content Provider:

Content Provider Diagram



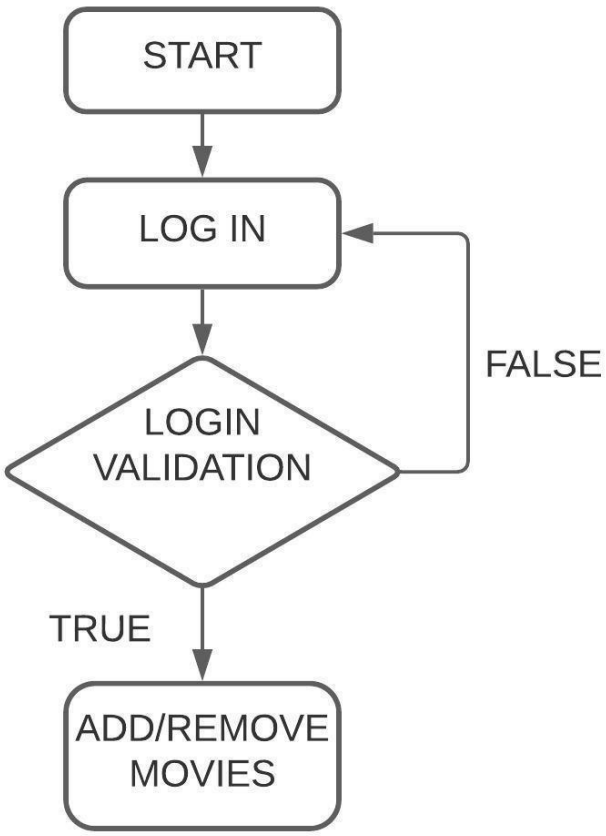
Brief-Description:

The content provider can add movies and their related information to the platform.

Initial Step-By-Step Description:

- The content provider navigates to the main recommendation website and then navigates to the add content subpage
- Upon logging in with the correct credentials
- HTML form appears when the data regarding the new movie to be added is entered, which is then added to the database.

Activity Diagram For Content-Provider



REQUIREMENT-3

The content provider has already had the required credential using which he logs in to the site and then adds new movies data.

REQUIREMENT-3.1

INPUT: "ADD/REMOVE MOVIES" option selected by the user. OUTPUT: user prompted to enter USER NAME and PASSWORD

REQUIREMENT-3.2

INPUT: Enter USER NAME and PASSWORD
OUTPUT: If the credentials are correct, then redirect to ADD/REMOVE page with search engine

-If not, it sends a warning of wrong credentials and asks for login again.

REQUIREMENT-3.3

INPUT: If Add movie is selected then Enter MOVIE DATA
OUTPUT: If the DATA is in the correct format, it is added to the database

REQUIREMENT-3.4

INPUT: If Remove data is selected then search movie option selected.
OUTPUT: Search box is prompted.

REQUIREMENT-3.5

INPUT: Proper keyword is entered to select the movie.
OUTPUT: If the movie is present in the data set then delete the movie from the dataset. If movie is not present in the dataset show it to the content provider.
PROCESSING: Based on the keyword of movie entered the movie is removed from the database.

7. Functional Requirements

7.1. Data collection and analysis

Data Collection:

- User Interaction Data: Collect data on user interactions with the platform, including but not limited to views, clicks, watch history, search queries, and interaction times. This data is essential for understanding user preferences and behaviours.
- User Feedback: Implement features to collect explicit user feedback, such as ratings, reviews, and likes/dislikes, to further refine recommendation accuracy.
- User Demographics and Profiles: Collect user demographic information (age, gender, location) and profile details (interests, subscriptions) at registration and through user profile updates.
- Content Metadata: Gather metadata on all media content, including titles, genres, release dates, descriptions, cast, and crew information. This metadata is crucial for content categorization and recoendation.

Data Processing:

- Data Cleaning and Pre-processing: Implement procedures for cleaning and pre-processing the collected data, removing outliers, and handling missing values to ensure data quality for analysis.
- User Segmentation and Profiling: Use advanced algorithms to segment users based on their behaviours, preferences, and demographics, creating detailed user profiles for personalized recommendations.
- Content Analysis and Categorization: Analyse content metadata to categorize media into genres, themes, and other relevant groups, aiding in the identification of content similarities.
- Recommendation Algorithms: Develop and integrate various recommendation algorithms (e.g., collaborative filtering, content-based filtering, and hybrid methods) to analyse user data and content metadata, generating personalized content recommendations.

Data Visualization

- Data visualization plays a crucial role in presenting insights and patterns derived from user data. The Recommendation Engine will utilize various types of graphs and charts to represent different aspects of user behaviour and content preferences.

1. Graph Types

- Bar Charts: These will be used to compare the popularity of different genres, artists, or content types among users. For example, a bar chart can show the number of times a particular genre is streamed or liked by users.
- Line Charts: These will depict trends over time, such as the increase in user engagement with specific content categories during weekends or holidays.
- Pie Charts: Pie charts can be used to illustrate the distribution of user preferences among different age groups or demographics. This visualization can also provide a visual representation of the distribution of male and female users relative to the total user base.
- Heat Maps: Heat maps can show the intensity of user interactions with content, highlighting areas of high and low engagement on the platform.
- Scatter Plot: Use a scatter plot to visualize the relationship between content popularity (e.g., number of views, listens) and user ratings. This visualization helps identify highly-rated but less popular content and vice versa, aiding in refining recommendation strategies.

- Stacked Bar Chart: A stacked bar chart is ideal for displaying the proportion of male and female viewers for various types of content on the platform. This visualization can provide insights into gender-specific preferences, aiding in tailoring recommendations.

2. Areas of Visualization

- User Profiles: Visualizing user profiles can help in understanding individual preferences and behaviour patterns. Graphs can show the most listened-to genres, favourite artists, or frequently watched content genres.
- Content Popularity: Graphs depicting the popularity of content categories, trending topics, or top-rated shows/movies can assist in content curation and recommendation strategies.
- Recommendation Effectiveness: Visualization can be used to evaluate the effectiveness of recommendations by comparing user interactions before and after receiving recommendations.

3. Graph Plotting Procedure

- Data Collection: Gather user interaction data, including streaming history, likes, ratings, and user profile information.
- Data Pre-processing: Clean and pre-process the data to remove outliers and ensure data quality. (If required)
- Graph Creation: Utilize data visualization libraries such as Matplotlib, Plotly, Power BI or Tableau to create interactive and informative graphs.
- Interpretation: Provide clear interpretations of the graphs, highlighting key insights and actionable recommendations based on the visualized data.

7.2. Content Recommendation

- Content recommendation systems rely on the analysis of user behaviour. These systems collect data on what users watch, listen to, click on, and how long they engage with content. Utilizing machine learning algorithms to categorize content accurately. This can include genre, mood, theme, actors, directors, etc. This categorization will form the basis for personalized recommendations.
- Recommendations are made by analysing the similarity between items the user has interacted with and other items in the system. Feature extraction involves transforming raw data (such as movie genres, directors, etc.) into

a format suitable for analysis. Techniques like binary feature matrix, bags of words, and TF-IDF are commonly used.

- Several similarity measures can be employed, including Pearson Correlation Coefficient, Cosine Similarity, Jaccard Similarity, Euclidean Distance, and Manhattan Distance.

- **Formula of some of these are:**
- Cosine Similarity:
  cosine similarity (A, B) = (A · B) / (||A|| ||B||)

- Euclidean Distance:
  d (A, B) = √ (∑ (A_i – B_i) ^2)

- Manhattan Distance:
  d (A, B) = ∑|A_i - B_i|

7.3. User Profile Creation:

- This phase is divided into three parts:
- Firstly, we need specifications for user registration process, including required and optional fields (e.g., username, email, password, profile picture).
- Second phase involves requirements for validating user input (e.g., email format, password strength).
- Third phase deals with guidelines for handling errors during registration (e.g., duplicate usernames, invalid email addresses).

7.4. Content Categorization:

- Content categorization is a crucial aspect of the recommendation engine for media and entertainment platforms. It involves classifying media content into distinct categories or groups based on various attributes such as genre, theme, mood, actors, directors, and more. The goal is to create a structured taxonomy that can facilitate accurate content recommendations according to individual user preferences. The following could be the key components of content categorization:

- Metadata Analysis: Gather metadata for all media content available on the platform, including titles, descriptions, genres, release dates, cast, crew, and any other relevant information. This metadata serves as the foundation for content categorization.
- Genre Classification: Classify media content into different genres such as action, comedy, drama, romance, horror, science fiction, etc. This classification helps users discover content that aligns with their genre preferences.
- Theme Identification: Identify common themes or motifs present in media content, such as friendship, love, betrayal, adventure, mystery, etc. Understanding the underlying themes can assist in recommending content with similar thematic elements.

7.5. Real Time Updating:

- Real-time updating in Recommendation Engine for the Media and Entertainment Platform, implementing a system that continuously analyses user behaviour and content availability to provide immediate and relevant recommendations as soon as new data becomes available. This could involve real-time data streaming, efficient algorithms for processing and updating recommendations, and a responsive user interface to display the updated suggestions promptly.

7.6. Integration With Platform:

- Specifications for how user profiles are integrated into the platform's existing infrastructure should follow
- Storage and Retrieval: User profiles should be seamlessly integrated into the platform's existing database infrastructure, allowing for efficient storage and retrieval of user data.
- Authentication: Ensure that user profile integration includes authentication mechanisms to verify user identity and permissions.
- Customization: Enable customization of user profiles to capture specific preferences, habits, and interactions relevant to the recommendation system.
- The module should communicate between the user profile module and other platform components (e.g., recommendation engine, content

management system) in a proper way by the use of API's. Also, there should be data security while doing this.

8. Non-Functional requirements

8.1. Performance:

- Performance is critical for ensuring that the recommendation engine delivers timely and responsive content suggestions to users. Here are some considerations for optimizing performance.
- Response Time: Minimize the response time of the recommendation engine to provide users with quick and seamless access to personalized content suggestions. This includes optimizing algorithms, database queries, and network latency.
- Concurrency: Design the recommendation engine to handle multiple concurrent requests efficiently, especially during peak usage periods.
- Optimized Algorithms: Implement efficient recommendation algorithms that can scale with the size of the user base and content library.

8.2. Scalability:

- The system should be able to handle increasing loads as the user base grows without sacrificing performance or reliability. It should scale horizontally by adding more resources or nodes to the system as needed.

8.3. Security:

- Ensure that user data is handled securely to maintain user trust and compliance with privacy regulations. Implement encryption, access controls, and data anonymization techniques to protect sensitive user information.

8.4. Reliability:

- Availability: The recommendation engine should be available 24/7 with minimal downtime. Any scheduled maintenance should be communicated in advance, and downtime should be kept to a minimum.
- Fault Tolerance: The system should be resilient to failures, including hardware failures, network outages, and software errors.
- Data Integrity: The recommendation engine should ensure the integrity of the data it processes and stores. This includes implementing measures to prevent data corruption, unauthorized access, and ensuring the accuracy of recommendations generated.
- Scalability: The system should be able to handle increasing loads as the user base grows without sacrificing performance or reliability.