

---

**Software  
Specification**

**Requirements**

**for**

**SPECTRUM                      MOVIE  
RECOMMENDATION SYSTEM**

**Version 1.0 approved**

**Group 11**

**25.12.23**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	1
1.5 References	2
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Features	3
2.3 User Classes and Characteristics	3
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	4
2.7 Assumptions and Dependencies	5
<b>3. System Features</b>	<b>6</b>
3.1 System Feature 1	6
3.2 System Feature 2	6
3.3 System Feature 2	7
<b>4. External Interface Requirements</b>	<b>8</b>
4.1 User Interfaces	8
4.2 Hardware Interfaces	8
4.3 Software Interfaces	8
4.4 Communications Interfaces	8
<b>5. Other Nonfunctional Requirements</b>	<b>9</b>
5.1 Performance Requirements	9
5.2 Security Requirements	9
5.3 Software Quality Attributes	9
<b>6. Other Requirements</b>	<b>10</b>
<b>Appendix A: Glossary</b>	<b>10</b>
<b>Appendix B: Analysis Models</b>	<b>11</b>

## Revision History

Name	Date	Reason For Changes	Version

# **1. Introduction**

## **1.1 Purpose**

This Software Requirements Specification outlines the Spectrum Movie Recommendation system's goals, features, and technical needs, acting as a development blueprint for all stakeholders. Analyzing user needs and the problem statement, it defines high-level functionalities, a simple user interface, and necessary hardware and software, ensuring everyone's on the same page for flawless creation and implementation.

## **1.2 Document Conventions**

Fonts:

Body text: Times New Roman 11pt

Headings: Times New Roman 18pt bold

Subheadings: Times New Roman 14pt bold

Terms and definitions: Times New Roman 11pt

Numbering:

Requirements: Arabic numerals (e.g., 1.1, 2.2.1)

Figures and tables: Consecutively numbered within each section

## **1.3 Intended Audience and Reading Suggestions**

The Software Requirements Specification (SRS) for the Spectrum movie recommendation system caters to a diverse audience, encompassing developers, project managers, marketing staff, users, testers, and documentation writers. Developers will find technical details such as system architecture and algorithms, while project managers can focus on project timelines and resource allocation. Marketing staff will benefit from information on promotional strategies and unique selling points, and end users can explore user interfaces and features. Testers will find details on test scenarios and performance metrics, and documentation writers can gain insights for creating user manuals. The document's reading sequence recommends starting with the overview sections for a high-level understanding before delving into role-specific sections, ensuring each reader gains a comprehensive perspective on the Spectrum movie recommendation system.

## **1.4 Project Scope**

This software helps users of the customer platform explore content quickly with our recommendation system's help. The software we are developing is a Spectrum Recommendation System for Movies, which uses the combination of collaborative and content-based filtering in the context of web-based recommender systems. In particular, we will link the well-known IMDB data set. The content filtering part of the system is based on trained neural networks representing individual user preferences. Using various experiments, we will demonstrate the influence of supplementary user and item features on our proposed hybrid recommender's prediction accuracy. To decrease system runtime and reveal latent user and item relations, we will factorize our hybrid model via singular value decomposition (SVD). Due to the enormous amount of information available online, the need for highly developed personalization and filtering systems is growing

permanently. Recommendation systems constitute a specific type of information filtering that attempts to present items according to the interests expressed by a user.

## **1.5 References**

- 1)Movie Recommendation System Vishwa Gosalia<sup>1</sup>, Bhavesh Chatnani<sup>2</sup>, Aman Gupta<sup>3</sup>, Pranal Soni<sup>4</sup> 1234  
Dept. of Computer Engineering, Thakur Polytechnic, Maharashtra, India  
<https://www.irjet.net/archives/V5/i2/IRJET-V5I270.pdf>
- 2)A Movie Recommender System: MOVREC  
[https://www.researchgate.net/publication/283042228\\_A\\_Movie\\_Recommender\\_System\\_MOVREC](https://www.researchgate.net/publication/283042228_A_Movie_Recommender_System_MOVREC)

## **2. Overall Description**

### **2.1 Product Perspective**

So our hybrid recommender system Spectrum that uses a combination of content-based and collaborative filtering techniques to generate movie recommendations. The system takes into account the user profile, movie data, and ratings data to generate personalized recommendations.

From a product perspective, the hybrid recommender system Spectrum offers several benefits: Enhanced User Experience:

- More personalized recommendations: By combining content-based and collaborative filtering, the system provides recommendations that are tailored to the individual user's preferences and past behavior, leading to higher engagement and satisfaction.
- Improved discovery of new content: The system recommends movies that are not only similar to what the user has liked before but also aligns with their broader interests, helping them discover hidden gems or new genres they might enjoy.
- Reduced choice overload: By filtering out irrelevant options, the system simplifies the decision-making process for users, making it easier for them to find movies they'll enjoy.

Increased Business Value:

- Higher user retention: Satisfied users are more likely to keep using the platform, leading to increased engagement and longer subscription periods.
- Boosted conversion rates: Recommendations that resonate with users are more likely to convert into actual watch time or purchases, driving revenue growth.
- More informed content acquisition: Understanding user preferences through the recommender system helps inform content acquisition strategies, ensuring that the platform offers movies that users actually want to watch.

Competitive Advantage:

- Differentiation from single-approach systems: Offering a hybrid system sets the product apart from platforms that rely solely on content-based or collaborative filtering, leading to a more competitive edge.
- Adaptability to diverse user bases: The ability to cater to both broad and niche interests through personalization helps attract a wider user base and cater to their evolving preferences.
- Improved brand image: Providing a consistently satisfying experience with relevant recommendations enhances user trust and loyalty, strengthening the product's brand image.

Overall, the hybrid recommender system presents a valuable product feature that enhances user experience, drives business value, and creates a competitive advantage. By offering personalized and relevant recommendations, it can increase user engagement, satisfaction, and ultimately, the success of the product.

## **2.2 Product Features**

Major Features and Functions:

**Movie Rating Predictor** The Movie Rating Predictor analyzes various factors that influence a movie's reception, including genre, director, cast, plot synopsis, and historical box office performance. It employs sentiment analysis to gauge public sentiment and reviews, and takes into account user preferences and past viewing history. By combining these factors, the system generates a predicted rating for a given movie.

**Box Office Success Predictor** The Box Office Success Predictor is a cutting-edge software application that harnesses the power of data analytics and machine learning to forecast the potential box office performance of upcoming movies. Users can input information about an upcoming movie, including its details, promotional strategy, and anticipated release date. The application not only provides a predicted box office performance but also offers actionable recommendations to maximize a movie's chances of success.

**Chatbot for Movie Discussions** The Chatbot for Movie Discussions is a versatile and user-friendly conversational AI system designed to enhance the movie-watching experience for film enthusiasts and cinephiles. Leveraging natural language processing and a vast database of movie-related information, this chatbot provides a platform for engaging, insightful, and interactive conversations about all aspects of the cinematic world.

**Character Tree** The Character Tree is a novel concept that explores the representation and visualization of complex character relationships in various narrative contexts, such as literature, film, and interactive storytelling. This abstract presents an overview of the Character Tree as a tool designed to enhance our understanding of character dynamics within stories and their impact on the overall narrative.

## **2.3 User Classes and Characteristics**

1. **Movie Enthusiasts:** Characteristics: Avid movie watchers, engaged with various genres and subgenres. Actively rate and review movies, providing rich data for the system. Open to exploring new movies beyond their usual preferences. Recommendation needs: Personalized suggestions based on past watch history, genre preferences, and critical acclaim. Serendipitous discoveries of hidden gems or niche movies they might enjoy.
2. **Casual Viewers:** Characteristics: Watch movies occasionally, primarily for entertainment or relaxation. Less likely to rate or review, making collaborative filtering less effective. Prefer familiar genres or popular recommendations. Recommendation needs: Easy-to-digest suggestions based on trending movies, popular choices, or light-hearted entertainment options. Recommendations based on their limited data points should be weighted towards content-based approaches.
3. **Genre Specialists:** Characteristics: Deeply invested in specific genres, with a comprehensive knowledge and strong preferences. Actively seek out similar movies by the same directors, actors, or within specific subgenres. Recommendation needs: Highly targeted suggestions within their preferred genres, including new releases, hidden gems, or critically acclaimed works. The system should prioritize content-based features and leverage collaborative filtering within their specific genre community.
4. **New Users:** Characteristics: Lack of data on watching history or explicit preferences presents a cold-start challenge. May explore various options to discover their interests. Recommendation needs: Diverse suggestions based on popular picks, current trends, or thematic groupings to help them identify preferred

genres and styles. Content-based features like movie descriptions, cast, and crew information can be helpful initial guidance.

5. Socially Influenced Viewers: Characteristics: Heavily influenced by friends, family, or online communities in their movie choices. Actively seek recommendations and trust suggestions from trusted sources. Recommendation needs: Personalized suggestions based on the preferences of their social circles or trusted opinion leaders. Collaborative filtering techniques leveraging their social connections can be highly effective. These are just a few examples, and the actual user classes and characteristics will depend on the specific platform and context. Identifying and understanding these user segments is crucial for tailoring the recommender system to address their individual needs and preferences, ultimately leading to a more personalized and satisfying user experience.

## **2.4 Operating Environment**

Hardware Platform: Software runs smoothly on desktops and laptops, requiring a capable graphics processing unit (GPU) for parallel processing. Operating System: Compatible with most major os such as windows ,macOs and linux Software Components and Applications:

Various libraries and frameworks such as openCv,Mediapipie and Tensorflow must be installed.

Server specifications: High-performance servers with sufficient RAM, CPU cores, and storage capacity to handle data processing, model training, and real-time recommendation generation. Consider GPUs or specialized AI accelerators for faster model training and inference.

Web development frameworks  
Server-side scripting language  
API integration for data retrieval  
Database management systems  
Machine learning libraries and frameworks  
Data collection preprocessing and visualization tools  
Hosting infrastructure

## **2.5 Design and Implementation Constraints**

Technical Constraints:

Hardware Needed: CPU, RAM, GPU Memory requirements for model and libraries:4/8 gb Real-time performance requirements :GPU required

Compatibility with communication protocols:TCP protocols and Web protocols like https

Non-Technical Constraints:

Language support for diverse users:English

Security and privacy measures:User data are stored in their device only

Coding conventions and standards:oops concepts are used.

## **2.6 User Documentation**

The user documentation for the Movie Recommendation System will include the following components:

- User Manuals: Detailed guides on how to use the system, including account setup, profile management, and understanding movie recommendations.

- Online Help: Interactive online help resources accessible within the system to guide users through specific tasks and features.
- Tutorials: Step-by-step tutorials demonstrating common tasks, such as rating movies, exploring genres, and managing preferences.

## **2.7 Assumptions and Dependencies**

The are the assumptions and dependencies involved:

Assumptions:

- Availability of Movie Metadata: Assumption: Accurate and comprehensive movie metadata, including genres, ratings, and cast information, is available from a reliable external source. Rationale: The system relies on up-to-date and accurate information for effective movie recommendations.
- User Ratings Reflect Preferences: Assumption: Users' movie ratings and preferences accurately reflect their taste. Rationale: The recommendation algorithm relies on user input to provide relevant and personalized movie suggestions.
- Stable Internet Connection: Assumption: Users have a stable internet connection. Rationale: The system requires real-time access to external databases for fetching movie data and providing timely recommendations.
- Consistent User Behavior: Assumption: User behavior, preferences, and movie-watching patterns remain relatively consistent over time. Rationale: The recommendation system assumes a certain stability in user preferences to enhance the accuracy of suggestions.
- Third-Party Authentication System: Assumption: The system integrates with a third-party authentication system for user account management. Rationale: User authentication is a critical component for personalization and security.

Dependencies:

- External Movie Databases: Dependency: The system relies on external movie databases for fetching and updating movie information. Impact: Changes or issues with external databases can affect the accuracy and timeliness of recommendations.
- Authentication Service: Dependency: Integration with a third-party authentication service for user login and account management. Impact: Issues with the authentication service may disrupt user access and account-related functionalities.
- Data Security Standards: Dependency: Compliance with industry standards for data security and privacy. Impact: Failure to meet security standards may result in unauthorized access or compromise of user data.
- User Engagement: Dependency: The effectiveness of the recommendation system depends on user engagement, including consistent rating and interaction with the platform. Impact: Low user engagement may lead to less accurate recommendations.
- Technology Stack Compatibility: Dependency: Compatibility with specific technologies, frameworks, and programming languages used in the development. Impact: Changes in technology may require updates to maintain system functionality.
- Legal and Regulatory Compliance: Dependency: Adherence to legal and regulatory requirements related to data privacy and content distribution. Impact: Non-compliance may result in legal issues and restrictions on content access.

### 3. System Features

Here's a breakdown of the key features

#### 3.1 System Feature 1

Movie Rating Predictor

##### 3.1.1 Description and Priority

Predicts the likely user rating for a given movie based on historical user preferences.

Priority: High

Benefit: Enhances user experience by providing personalized movie recommendations. Penalty: Inaccurate predictions may lead to dissatisfaction. Cost: Moderate (Development and training data acquisition) Risk: Moderate (Dependent on the accuracy of the prediction algorithm)

##### 3.1.2 Stimulus/Response Sequences

###### 1. User Rates a Movie:

Stimulus: User submits a rating for a movie.

Response: The system updates the user's rating history and re-calibrates the prediction model.

###### 2. User Requests Rating Prediction:

Stimulus: User requests a predicted rating for a specific movie.

Response: The system provides the predicted rating based on the user's historical preferences.

##### 3.1.3 Functional Requirements

REQ-1: Capture and store user movie ratings.

REQ-2: Implement a machine learning model for rating prediction.

REQ-3: Update the prediction model regularly based on new user ratings.

REQ-4: Provide an API endpoint for rating prediction requests

#### 3.2 System Feature 2

Box Office Success Predictor

##### 3.2.1 Description and Priority

Predicts the potential box office success of a movie based on various factors like genre, cast, and historical performance.

Priority: Medium

Benefit: Assists filmmakers and studios in making informed decisions.

Penalty: Inaccurate predictions may lead to financial losses.

Cost: Moderate to High (Data acquisition and model development)

Risk: Moderate (Dependent on the accuracy of prediction factors)

##### 3.2.2 Stimulus/Response Sequences

###### 1. User Inputs Movie Details:



Stimulus: User provides details of an upcoming movie.

Response: The system predicts the potential box office success based on the input.

2. Studio Requests Prediction:

Stimulus: Film studio requests a box office success prediction for a planned movie. Response: The system provides the predicted outcome based on historical data.

### 3.2.3 Functional Requirements

REQ-1: Collect and maintain a comprehensive database of movie details and historical box office data.

REQ-2: Implement a predictive model for box office success.

REQ-3: Integrate external data sources for real-time updates.

REQ-4: Provide a user interface for inputting movie details and viewing predictions.

## 3.3 System Feature 3

### 3.3.1 Description and Priority

Facilitates interactive and engaging conversations with users about movies, genres, and recommendations.

Priority: High

Benefit: Enhances user engagement and provides a personalized movie discussion experience. Penalty: Inadequate responses may lead to user dissatisfaction.

Cost: Moderate (Development and training of the chatbot) Risk:

Low to Moderate (Dependent on the quality of natural language processing)

### 3.3.2 Stimulus/Response Sequences

1. User Initiates Chat:

Stimulus: User starts a chat session.

Response: The chatbot greets the user and awaits further input.

2. User Asks for Movie Recommendations:

Stimulus: User requests movie suggestions.

Response: The chatbot provides personalized movie recommendations based on user preferences.

### 3.3.3 Functional Requirements

REQ-1: Develop a natural language processing (NLP) engine for understanding user inputs.

REQ-2: Implement a knowledge base for movie-related information.

REQ-3: Train the chatbot to respond contextually to user queries.

REQ-4: Provide options for users to explore genres, ask for reviews, or get trivia.

## 4. External Interface Requirements

### 4.1 User Interfaces

The Movie Recommendation System will feature a user-friendly graphical interface, accessible through web and mobile platforms.

The interface will include:

1. Homepage: Display recommended movies based on user preferences. Provide navigation options for exploring genres, top-rated movies, and user profile settings. Include visually appealing movie thumbnails with brief details.
2. Movie Detail Page: Show comprehensive information about a selected movie, including synopsis, cast, and user ratings. Allow users to rate, review, and add the movie to their watchlist. Include a "Back" button for easy navigation.
3. User Profile: Display user preferences, watchlist, and rated movies. Provide options for profile customization and settings. Include a logout button for account management.
4. Search Functionality: Allow users to search for specific movies, genres, or actors. Display search results in a clear and organized manner. Include filters for refining search results.
5. Notification Center: Notify users of new recommendations, movie releases, or updates. Include a clear indicator for unread notifications. Allow users to customize notification preferences.

### 4.2 Hardware Interfaces

The Movie Recommendation System is designed to be compatible with various hardware configurations commonly found in user devices.

The key specifications are: Supported Devices: Desktops, laptops, tablets, and smartphones. Hardware Requirements: CPU: Core i5 10th gen or higher.

GPU: GeForce GTX 980 or higher.

RAM: 8GB or higher.

Operating System Compatibility: No specific OS constraints.

### 4.3 Software Interfaces

1. Database: Name and Version: MongoDB v4.2 Data Items: Movie details, user profiles, ratings, and preferences. Purpose: Store and retrieve movie-related data and user information.
2. Machine Learning Services: Name and Version: TensorFlow v2.6 Data Items: Trained models for movie rating prediction and box office success prediction. Purpose: Provide predictive analytics for personalized recommendations.

### 4.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

To assess the performance of a system, the following parameters are considered:

1. Response Time
2. Workload-Workload is relatively heavy, as it requires cv models which are quite resource extensive.
3. Scalability - Highly scalable using ML-based cloud services such as TensorFlow, AWS-ML, and Google Cloud-ML.
4. Platform
  - No OS constraints.
  - CPU: Core i5 10th gen or higher.
  - GPU: GeForce GTX 980 or higher
  - RAM: 8GB or higher.

To ensure optimal performance, the system must be compatible with modern hardware specifications commonly found in user devices.

### **5.2 Security Requirements**

- 1) Confidentiality: Requirement: The system must enforce access control and disclosure restrictions to maintain user privacy and protect proprietary information. Rationale: Preserving confidentiality prevents unauthorized access to sensitive user data.
- 2) Integrity: Requirement: The system should prevent improper (unauthorized) information modification or destruction. Rationale: Ensuring the integrity of the data maintains the accuracy and reliability of the movie-related information.
- 3) Availability: Requirement: All processing and temporary data storage should occur on the local device to avoid foreign interventions. Rationale: Keeping private data local enhances data security and prevents unauthorized access during data processing.

### **5.3 Software Quality Attributes**

- 1) Usability: Requirement: The application should be designed to be simple and user-friendly, providing an intuitive user interface for easy navigation. Rationale: A user-friendly interface enhances the overall usability of the movie recommendation system.
- 2) Availability: Requirement: The system must ensure availability, maintaining integrity, dependability, and confidentiality even during high user demand. Rationale: Availability is crucial for a seamless user experience and to meet user expectations.
- 3) Functionality: Requirement: The system, although still under development, should actively work toward achieving the goal of providing accurate movie recommendations and features like translating word-level signs. Rationale: Continuous development and improvement are essential to achieving the system's functionality goals.

## **6. Other Requirements**

### **Other Requirements**

#### Database Requirements

##### Database Backup and Recovery:

The system should implement regular database backup procedures to ensure data integrity and availability. A recovery plan must be in place to restore the database in the event of data loss or system failure.

##### Data Retention Policy:

Define a data retention policy specifying how long user data, movie ratings, and other records will be stored.

Ensure compliance with privacy regulations regarding data retention.

#### Internationalization Requirements

##### Language Support:

The system should support multiple languages for user interfaces.

Language preferences should be configurable on a per-user basis.

##### Region-Specific Content:

Implement features to adapt content recommendations based on regional preferences and availability.

#### Legal Requirements

##### Copyright Compliance:

Ensure compliance with copyright laws when displaying movie details, images, and other copyrighted content.

Provide proper attribution for third-party content.

##### Data Privacy Regulations:

Adhere to data privacy regulations such as GDPR, ensuring user consent and data protection.

Clearly communicate the system's privacy policy to users.

#### Reuse Objectives

##### Code Reusability:

Design code modules with reusability in mind, allowing for potential reuse in future projects or system enhancements.

Document reusable components and their intended applications.

## **Appendix A: Glossary**

### Terms and Definitions

SRS: Software Requirements Specification

API: Application Programming Interface

NLP: Natural Language Processing

ML: Machine Learning

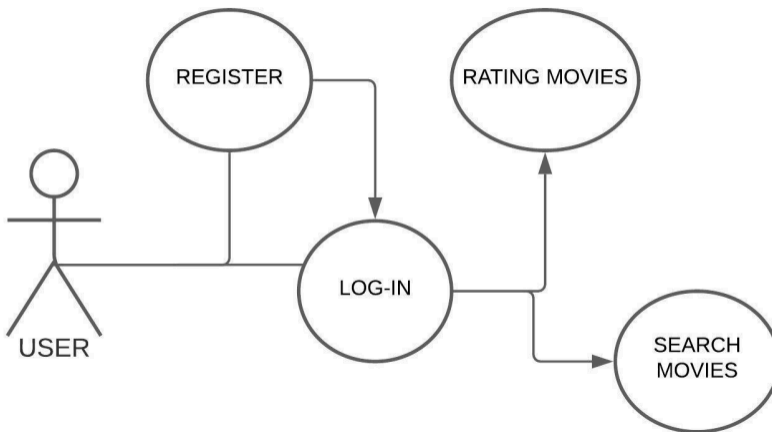
GPU: Graphics Processing Unit

FCM: Firebase Cloud Messaging

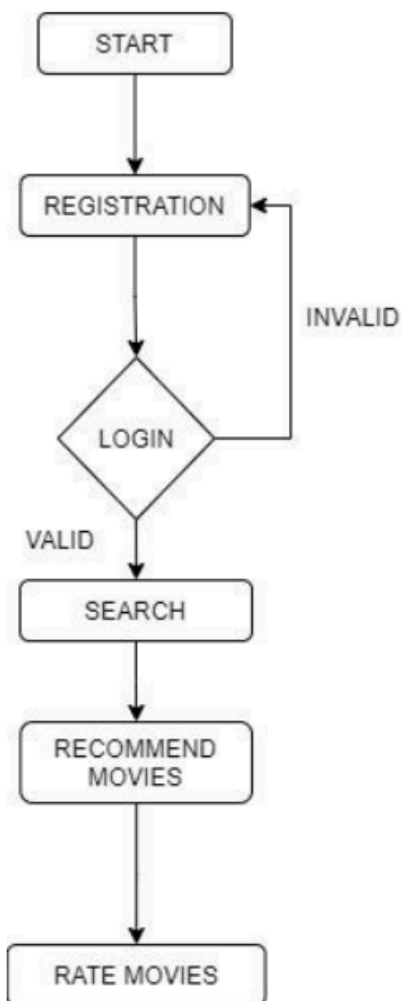
OAuth: Open Authorization

## Appendix B: Analysis Models

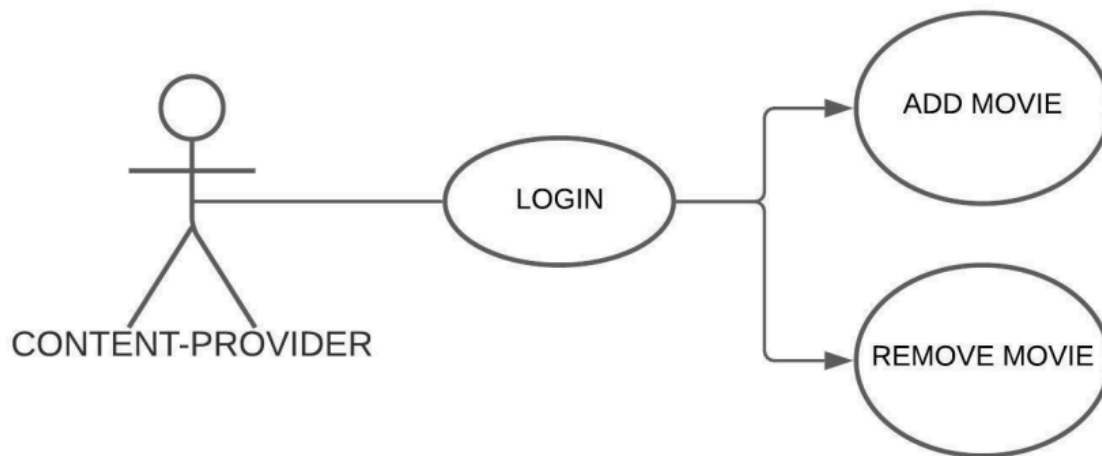
Use Case Diagram-User



Activity Diagram For Existing User



Content Provider Use Case Diagram



Activity Diagram For Content-Provider

