

1. Experiment 1

1.1. Aim

Prepare a SRS document in line with the IEEE recommended standards

1.2. Table of Contents

- 1. Experiment 1
 - 1.1. Aim
 - 1.2. Table of Contents
 - 1.3. Software Requirements Specification for Movie Recommender System
 - 1.3.1. Introduction
 - 1.3.1.1. Purpose
 - 1.3.1.2. Product Scope
 - 1.3.2. Overall Description
 - 1.3.2.1. Product Perspective
 - 1.3.2.2. Product Features
 - 1.3.2.3. User Classes and Characteristics
 - 1.3.2.4. Operating Environment
 - 1.3.2.5. Design and Implementation Constraints
 - 1.3.3. System Features
 - 1.3.3.1. Feature: Movie Recommendation based on User Preferences
 - 1.3.3.1.1. Description and Priority
 - 1.3.3.1.2. User Classes and Characteristics
 - 1.3.3.1.3. Functional Requirements
 - 1.3.4. External Interface Requirements
 - 1.3.4.1. User Interfaces
 - 1.3.4.2. Hardware Interfaces
 - 1.3.4.3. Software Interfaces
 - 1.3.4.4. Communications Interfaces
 - 1.3.5. Nonfunctional Requirements
 - 1.3.5.1. Performance Requirements
 - 1.3.5.2. Safety Requirements
 - 1.3.5.3. Security Requirements
 - 1.3.5.4. Software Quality Attributes
 - 1.3.5.5. Other Requirements

1.3. Software Requirements Specification for Movie Recommender System

1.3.1. Introduction

Movie recommendation systems have become increasingly popular in recent years, providing users with a personalized selection of movies that they are likely to enjoy. The purpose of this software is to create a movie recommender system that recommends movies similar to the movie the user likes, while also analyzing the

sentiments of the reviews given by the user. The aim is to provide users with a more accurate and tailored movie recommendation system.

1.3.1.1. Purpose

The purpose of this software is to provide a movie recommendation system that takes into account the user's movie preferences and sentiment analysis of the reviews provided by the user. The system will provide personalized movie recommendations that the user is likely to enjoy.

1.3.1.2. Product Scope

The movie recommender system will be a standalone application that runs on various operating systems such as Windows, Linux, and macOS. The software will use machine learning algorithms and natural language processing techniques to analyze user reviews and provide movie recommendations that match their interests. The system will also provide a user-friendly interface for users to input their preferences and interact with the system. The software will also have an option for users to provide feedback on the recommendations they receive, which will help to improve the system's accuracy.

1.3.2. Overall Description

1.3.2.1. Product Perspective

The movie recommender system is a standalone application that operates independently of other software systems. However, it will use machine learning algorithms and natural language processing techniques to analyze user reviews and provide movie recommendations that match their interests.

1.3.2.2. Product Features

The following are the key features of the movie recommender system:

- Movie recommendation based on user preferences
- Sentiment analysis of user reviews
- Personalized movie recommendations
- User-friendly interface
- Feedback mechanism to improve the accuracy of the recommendations

1.3.2.3. User Classes and Characteristics

The movie recommender system is designed for movie enthusiasts who are looking for personalized movie recommendations. Users can be of any age, gender or cultural background.

1.3.2.4. Operating Environment

The movie recommender system can be installed and run on any operating system that supports Python programming language. It is recommended to have a minimum of 4GB of RAM and 1GHz processor speed to ensure smooth functioning.

1.3.2.5. Design and Implementation Constraints

The movie recommender system will use Python-based libraries such as Scikit-learn, TensorFlow and NLTK to analyze user reviews and provide movie recommendations. The software must be designed to minimize processing time and ensure maximum accuracy in movie recommendations. The system must also be scalable to accommodate large volumes of user data.

1.3.3. System Features

1.3.3.1. Feature: Movie Recommendation based on User Preferences

1.3.3.1.1. Description and Priority

The movie recommendation feature will provide personalized movie recommendations to users based on their movie preferences. The system will analyze user input such as movie genre, actors, director, and movie title to provide recommendations that match their interests. This feature has a high priority as it is the main function of the movie recommender system.

1.3.3.1.2. User Classes and Characteristics

The movie recommendation feature is designed for users who are looking for personalized movie recommendations based on their movie preferences. Users can be of any age, gender, or cultural background.

1.3.3.1.3. Functional Requirements

- The user must be able to input their movie preferences through a user-friendly interface.
- The system must analyze the user input to provide personalized movie recommendations.
- The system must provide accurate and relevant movie recommendations based on the user's movie preferences.
- The system must allow the user to filter their recommendations by movie genre, actors, director, and other criteria.
- The system must continuously update its recommendation algorithm based on user feedback to improve the accuracy of future recommendations.

1.3.4. External Interface Requirements

1.3.4.1. User Interfaces

The movie recommender system will have a graphical user interface that will be easy to use and navigate. The user interface will allow users to input their movie preferences and interact with the system. The interface will be intuitive and easy to understand for all types of users.

1.3.4.2. Hardware Interfaces

The movie recommender system will be compatible with standard hardware components, such as a keyboard and mouse, and will run on any computer with the minimum required specifications.

1.3.4.3. Software Interfaces

The movie recommender system will use Python libraries such as flask, Scikit-learn, TensorFlow, and NLTK to analyze user reviews and provide movie recommendations. The system will also use web scraping tools to

gather information on movies and user reviews from various sources.

1.3.4.4. Communications Interfaces

The movie recommender system will communicate with external data sources, such as movie databases, to gather information on movies and user reviews. The system will also have a feedback mechanism to allow users to provide feedback on the recommendations they receive. The feedback mechanism will help improve the accuracy of future recommendations.

1.3.5. Nonfunctional Requirements

1.3.5.1. Performance Requirements

The movie recommender system should provide accurate movie recommendations within a reasonable time frame. The system should be able to handle large volumes of user data and provide recommendations that are personalized and relevant to the user. The system should have a response time of less than 5 seconds when providing movie recommendations.

1.3.5.2. Safety Requirements

The movie recommender system will not collect any personal information from users. The system will only gather information on movie preferences and reviews to provide personalized movie recommendations. The system will ensure that the user data is kept confidential and is not accessible to unauthorized users.

1.3.5.3. Security Requirements

The movie recommender system will be designed to prevent unauthorized access to user data. The system will use encryption to protect user data and will have a secure login system to prevent unauthorized access to the system. The system will also be designed to prevent attacks such as SQL injection and cross-site scripting.

1.3.5.4. Software Quality Attributes

The movie recommender system will be designed to be user-friendly, efficient, and accurate. The system will have a simple and intuitive user interface that is easy to navigate. The system will also be designed to minimize processing time and maximize accuracy in movie recommendations. The system will be tested to ensure that it is free of bugs and errors.

1.3.5.5. Other Requirements

The movie recommender system should be compatible with all major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. The system should also be able to handle concurrent user requests without any performance issues. The system should have a backup mechanism in case of a system failure, to ensure that user data is not lost.