Software Requirements Specification

for

<*MOVIE RECOMMENDER*>

Version 1.0 approved

Prepared by <*MISHA BHARTI(1929023),ARNAV KUMAR MISHRA(1929079),HARSHVARDHAN CHANDRA (1929088),RAJA KASHYAP(1929105),SHASWAT PASARI(1929117)***>**

<KIIT>

<20/8/2021>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

A recommendation system has become an indispensable component in various e-commerce applications. Recommender systems collect information about the user's preferences of different items (e.g. movies, shopping, tourism, TV, taxi) by two ways, either implicitly or explicitly.

Simply put a Recommendation System is a filtration program whose prime goal is to predict the "rating" or "preference" of a user towards a domain-specific item or item. In our case, this domain-specific item is a movie, therefore the main focus of our recommendation system is to filter and predict only those movies which a user would prefer given some data about the user him or herself.

The Simple Recommender offers generalized recommendations to every user based on movie popularity and (sometimes) genre. The basic idea behind this recommender is that movies that are more popular and more critically acclaimed will have a higher probability of being liked by the average audience.

1.1 Purpose

The goal of this document is to provide support information on the *project* (current version v1.0).
It will attempt to explain the functionality of the program and the features it provides.

It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the customer company for their approval and a reference for development of the system.

1.2 Document Conventions

- **Entire content should be justified**
- **The table gives the idea that how the document/srs layout is going to be**

FONT	**STYLE**	**SIZE**
Heading Times New Roman	**Heading** Bold	**Heading** 18
Sub-Heading Times New Roman	**Sub-Heading** Bold	**Sub-Heading** 14
Other's Arial	**Other's** Regular	**Other's** 12

1.3 Intended Audience and Reading Suggestions

This Software Requirements document is intended for:
− Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (design and code the application – it sets the guidelines for future development).
− Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.

− End users of this application who wish to read about what this project can do.

1.4 Product Scope

Movie Recommender is a music recommendation system, which provides users movies which they may like, based on the movies that they previously saw. Every logged in user should have access to the recommender system. The system will go through the movies that user previously saw and rated, then according to those information it should provide movies to the user. The project's main aim is to provide accurate movie recommendations to the user. This project is beneficial for the users and the companies. For users, they may find movies that they may like without consuming time and even they can encounter new movie s which they like from the recommendations. For the company, they make the website more attractive, so they draw more users to the website and the system makes the users of the website spend more time online.

1.5 References

https://www.crio.do
https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

2. Overall Description

2.1 Product Perspective

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can listen to our preferred music or can watch movies of our choice. For watching favorable movies online we can utilize movie recommendation systems, which are more reliable, since searching of preferred movies will require more and more time which one cannot afford to waste. This product is a follow-on member of already existing Movie recommendation system. This system will have some UI where people can go and check out the recommended movies for them, after following some step.

2.2 Product Functions

There will be two major functions which includes-

- Recommend the movies, to the users, which are trending in their region in a particular week.
- It will recommend few movies based in the genre provided by the user

2.3 User Classes and Characteristics (left for whoever is doing the code part)

The main aim of this product is to eliminate the tedious task of searching through a plethora of movies in a collection and hence save the user time by curating a list of movies the user might like based on the user’s preferences, hence the product is not class oriented. People of every age groups can use this product.

2.4 Operating Environment

Operating environment for the movie recommendation system is as listed below.

- distributed database
- client/server system
- Operating system: Windows.
- database: sql+ database
- Platform : Chrome/Edge/Firefox

2.5 Design and Implementation Constraints

- SQL commands for above queries/applications
- Python language for the implementation of movie recommendation system
- HTML, CSS, JavaScript, ReactJS for the development of UI

2.6 User Documentation

How to use?
Set your minimum five genres in the section which asks for the same, and click on the button which says *Recommend* and the algorithm will provide you with a list of movies.

Before you start
Login to our website adhering all the terms and conditions and you are good to go.

2.7 Assumptions and Dependencies

One assumption is that the application is used on a computer or a mobile phone with enough performance ability, and the use of an up-to-date internet browser.

3. External Interface Requirements

3.1 User Interfaces

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Google Chrome by which user can access to the system.

The user interface shall be implemented using any tool or software package like Java Applet.

3.2 Hardware Interfaces

Since the application must run over the internet, this brings out the requirement of a network interface on the device. User should have a device with valid internet connection, Wi-Fi or 3G.

3.3 Software Interfaces

- The data of the project is organized in a relational database as it makes it easier to curate data with a large number of attributes.
- Python 3.9.6 will be used as the predominant programming language for this project. Since the project is based on Machine Learning Python is the clear as it has build in modules for handling and cleansing data eg. Pandas and numpy and also for training the recommendation model eg. SkikitLearn .

3.4 Communications Interfaces

The movie recommender system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

4. System Features

This project includes sorting of different movies according to the mood of the user which is based on the past experiences of the user. It is based on different past preferences of the user and what kind of Films user would like to watch.We only implement a simple user interface for showing system recommendations.In main system,user logins first and starts giving options of movies which are created on our recommendation service data. The output is shown in main applications interface.

4.1 SORTING OF MOVIES

4.1.1Description and Priority

Recommending movies on past preferences of the user.High priority is given to preferences of the user.

4.1.2Stimulus/Response Sequences

- *GENERATE DATA*:User can get movie recommendation on the device. Movie information will be collected according time of action,mood,actor information,genre and rating. The database will be filled with info.

RECOMMENDATION:It can suggest movies as recommendation based on data set by user's approach.The main function will show movies based on recommendation algorithm.When a user will choose movie recommendation the system will give it recommended movies through their past experiences.

4.1.3Functional Requirements

>Applicant's dasboard page

>A page with list of movies

>A feature to filter and organize movies based on rating, genre and other desciptions.

>A feature to enable applicants to explore different movies based on their preferences.

*Database:*The database based on past preferences of the user will be created.

Algorithm:It will capture preferences and suggest movies based on it.

*Historical Data:*It will be collected by the System.

Searching reporting requirement:Will tell users how to search data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 Safety Requirements

No safety requirements has been required.

5.3 Security Requirements

Database should be reached securely and the data should not be lost. It also should not change except inter-agent updates. Moreover, since our data-set contain some information of user such as tracks he/she listened, security design is important in the web service , so we must choose their database partner carefully.

5.4 Software Quality Attributes

1.AVAILABILITY: The site should be available on the specified date and specified time as many users are doing their search.
2.CORRECTNESS: The site will give proper recommendation based on one's view.
3.MAINTAINABILITY: The administrators and back-end developers should maintain correct output of recommendations.
4.USABILITY: The site should satisfy a maximum number of users needs.