Software Engineering
Software Requirements Specification (SRS) Document

Vinod Kuril
Harshita Mishra
Dhara Modi
SoftEng06
10/20/2009

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 2.10	Team	All sections being Filled	3/11/21
Version 2.12	Team	All sections being Filled	14/03/22
Final draft	Team	All sections being Filled	17/02/23

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date
Barry Chapman			
Dr. T. L. Lewis			

Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date
Drashti Shrimal	2.130		24-02-22
Aniket Mishra	2.130		24-02-22

Contents

1. Introduction

Introduction

The purpose of this document is to define and describe the requirements of the project and to spell out the system's functionality and its constraints.

Scope of this Document

The customer and the user for the system can be anyone from any age-group. The objective of our music recommendation system is to provide suggestions to the users that fits the user's preferences.

Overview

The system is used to recommend songs based on the factors that have lyrics similarity between songs , audio features of songs, metadata of songs using ANN and KNN regression algorithms. Recommendations are also made based on the same artist.

Business Context

The system uses the freemium business model and generates revenue through paid subscriptions and advertisements. Users can choose from paid premium plans that suit the collective number of users and offer discounts.

2. General Description

2.1 Product Functions

By using a music recommender system, the music provider can predict and then offer the appropriate songs to their users based on the characteristics of the music that has been heard previously.

2.2 Similar System Information

In the music industry, recommendation systems are part of a big engine of streaming apps like Spotify, YouTube Music, Deezer, Tidal, and the like. The possible strength that our system has over the majority is the music recognition, melodic analysis feature of the program.

2.3 User Characteristics

To use this music web app, a user should have a basic level of technical knowledge and be comfortable using mobile devices, navigating mobile interfaces, and connecting to the internet. This may include understanding how to download and install apps, create and manage user accounts, search for music, create playlists, and access premium features. Users may also need to have a basic understanding of the app's terms of service, including privacy policies and copyright regulations.

2.4 User Problem Statement

Users may face problems with music recommendation systems, including limited scope, lack of personalization, inaccurate recommendations, limited access to music, lack of transparency, and privacy concerns.

2.5 User Objectives

It allows end-users to find out more tracks and albums according to their choice. This helps users discover more songs present on your platform. Without a recommendation, many of the users will not be able to discover their favorite tracks.

2.6 General Constraints

Most important concern of the system is producing accurate recommendations. To provide expected accuracy and to handle sparse and huge data at the same time is critical.

3. Functional Requirements

- ❖ Client:

The client-side of the system will be an application with a user interface that is integrated into a music listening website or application. This application gathers the information from users, investigates some actions of the users, and provides the connection with the server. This application is the client-side interface of the Music Recommender, so it does not include the functionalities of the host music environment such as playing music etc.

- Requesting recommendations: The client-side application must allow users to request recommendations manually, and interact with the server to receive recommendations.
- Evaluating songs: It must be able to evaluate songs and send appropriate information to the server.
- Investigating user: The system will follow the interaction of users with music items., and send these obtained back to the server.
- Display recommendations: The application must display the recommendations that are obtained from the server to the user in a proper way by providing a GUI.

- ❖ Server

The server-side system will hold the entire data in a graph database, and must include all functionality to perform operations on this database, receive requests from the clients, evaluate, create and send recommendations etc.

● Handle recommendation requests: The server application shall obtain and handle requests for recommendations.

● Store evaluations: The server application shall receive and store music evaluations

● Data storing: The server application shall be able to store the newly retrieved data to the database.

● Recommend using content based filtering: The server application shall be capable of producing recommendations by interpreting the content and evaluations by actual user.

● Recommend using collaborative filtering: The server application shall be capable of producing recommendations by interpreting the evaluations given by actual user and other similar users.

● Recommend using contextual collaborative filtering: The server application shall be capable of producing recommendations by interpreting contextual information given by the users and evaluations given by the actual user and other similar users.

4. Interface Requirements

4.1 User Interfaces

4.1.1 GUI

Our music web app GUI contains user-friendly, visually appealing, and intuitive. It should include a home screen, music library, player, personalized recommendations, social features, settings, and help and support. The aim is to provide a seamless experience for users to find, organize, and play music, as well as connect with other users and discover new music.

4.1.2 CLI

There is no command line interface

4.1.3 API

We are using RapidAPI for the system

4.2 Hardware Interfaces

The recommendation system can work on any internet connected device. These devices should have some limit requirements to make the application run effectively. We expect that the processor speed and internet speed are high

4.3 Software Interfaces

The system will work on any platform (eg : Chrome , Firefox , Bing etc.). Internet connection is a must to reach the system. Moreover, most of the applications will be coded by JavaScript. Moreover, some api will be used to fetch required data for the web application.

5. Performance Requirements

You can only play on one device at a time, per account.

Supported

iOS	iOS 14 or above
Android	Android OS 5.0 or above
Mac	OS X 10.13 or above

Windows

Windows 7 or above

You can play the music web app from your web browser.

Go to the web player

Supported web browsers:

Chrome
Firefox
Edge
Opera
Safari

6. Other non-functional attributes

.

6.1 Security

The security requirement required in a music recommendation system is to ensure the confidentiality, integrity, and availability of user data and system resources. It includes measures such as authentication, access control, data encryption, and secure communication protocols to prevent unauthorized access, data breaches, and other security threats

6.2 Binary Compatibility

Binary compatibility is not a critical requirement for a music recommendation system as it primarily deals with software and hardware interactions. The compatibility of different software and hardware components may affect the performance and functionality of the system. However, music recommendation systems rely more on data analytics, machine learning algorithms, and user preferences. As long as these components are compatible and work together seamlessly, the system can provide accurate and personalized music recommendations.

6.3 Reliability

A music recommendation system requires high reliability to provide accurate and consistent recommendations based on user preferences. The system should be robust and handle errors efficiently while providing uninterrupted service. Regular maintenance, testing, and user feedback can help ensure reliability.

6.4 Maintainability

The system shall be maintained by an employee in maintenance team

6.5 Portability

Portability is an essential requirement for a music recommendation system as it should be able to run on various platforms and devices without any issues. It should be designed to support different operating systems, hardware configurations, and network environments. This is necessary to ensure that users can access the system from different devices and locations, such as smartphones, tablets, laptops, and desktop computers.

6.6 Extensibility

The extensibility required in a music recommendation system is to adapt to new music genres, data sources, and user preferences. This is achieved through modular architecture, open APIs, and flexible data models to integrate with other music-related platforms and services, staying up-to-date and providing relevant recommendations to users.

6.7 Reusability

Reusability is a desirable requirement in a music recommendation system as it enables developers to use existing code, models, and data to build new or improved versions of the system. For instance, a music recommendation system can reuse data sets, machine learning models, or analytics tools from other systems or research projects. Reusability can save development time and resources and promote standardization and interoperability between different music recommendation systems.

6.8 Application Affinity/Compatibility

The compatibility required in a music recommendation system is mainly related to data and application compatibility. The system should be able to process and analyze different types of music data, including user preferences, listening history, and metadata, from various sources such as streaming platforms, music libraries, and social media. Additionally, the system should be compatible with various devices and platforms, including smartphones, tablets, and desktops.