



# **PROJECT REQUIREMENTS SPECIFICATION**

## **Music Recommendation System**

**UE19CS390A – Capstone Project Phase – 1**

*Submitted by:*

**Jason Joachim Carvalho  
Shriram Duraikkannan  
Swarna Kumaran**

**PES1UG19CS195  
PES1UG19CS471  
PES1UG19CS531**

Under the guidance of

**Dr. Uma D**  
Professor  
PES University

**January – May 2022**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
FACULTY OF ENGINEERING  
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

# PROJECT REQUIREMENTS SPECIFICATION

## TABLE OF CONTENTS

1. Introduction	3
1.1 Project Scope	3
2. Product Perspective	3
2.1 Product Features	3
2.2 Operating Environment	3
2.3 General Constraints, Assumptions and Dependencies	4
2.4 Risks	4
3. Functional Requirements	5
4. External Interface Requirements	5
4.1 User Interfaces	5
4.2 Hardware Requirements	6
4.3 Software Requirements	6
4.4 Communication Interfaces	5
5. Non-Functional Requirements	7
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	7
6. Other Requirements	7
Appendix A: Definitions, Acronyms and Abbreviations	7
Appendix B: References	8

# PROJECT REQUIREMENTS SPECIFICATION

## 1. Introduction

This document provides a debrief on the requirements and specifications required for the project “Music Recommendation System”.

### 1.1. Project Scope

The purpose of this project is to tackle two major problems with music recommender systems listed below and therefore give a better user experience.

- **Cold start problem:**
  - One of the major problems of recommender systems in general, and music recommender systems in particular is the cold start problem, i.e., when a new user registers to the system or a new item is added to the catalog and the system does not have sufficient data associated with these items/users.
- **Automatic playlist continuation:**
  - In its most generic definition, a playlist is simply a sequence of tracks intended to be listened to together. The task of automatic playlist generation (APG) then refers to the automated creation of these sequences of Tracks to improve user experience

## 2. Product Perspective

### 2.1. Product Features

- The system is able to perform recommendations of songs based on the user’s preferences.
- The system will provide various forms of recommendations that a user can pick from on the navigation pane (Dig1), for example “Content-based” recommendations will be placed under a label saying “Customized for you”
- Generate playlists of songs based on the user’s preferences.
- Provides links to play the recommended songs on the spotify website

### 2.2. Operating Environment

#### System Requirements

- The Operating System should be capable of playing music and have any of the mentioned OS installed.
- Constant Internet Connection is required to get the recommended songs.

## **PROJECT REQUIREMENTS SPECIFICATION**

### **Platforms Include**

#### **GNU/Linux**

- CPU Type : Pentium 4 or higher; 2 GHz or higher
- Memory/RAM : 1 GB minimum, up to the system limit
- Hard Disk : 4 GB or higher
- Graphics : X Window server or similar graphics server

#### **Windows**

- Processor: 1 gigahertz (GHz) or faster.
- RAM: 1 gigabyte (GB) (32-bit) or 2 GB (64-bit)
- Free hard disk space: 16 GB.
- Graphics card: Microsoft DirectX 9 graphics device with WDDM driver.

### **2.3. General Constraints, Assumptions and Dependencies**

#### **Constraints:**

- Dataset may not have enough songs to tackle niche recommendations
- Dataset only covers songs till 2020.
- Users must rate the songs initially so that our system can start using the collaborative model to make recommendations.
- Must have constant internet connection.

#### **Assumptions:**

- We are considering only english songs for this prototype.

#### **Dependencies:**

- Spotify service - spotipy library to retrieve the data for recommendation.

### **2.4. Risks**

- Unable to evaluate against other models as there are no metrics available to measure accuracies between models.
- In case of bad internet connection, the entire system would not be able to function.

# PROJECT REQUIREMENTS SPECIFICATION

## 3. Functional Requirements

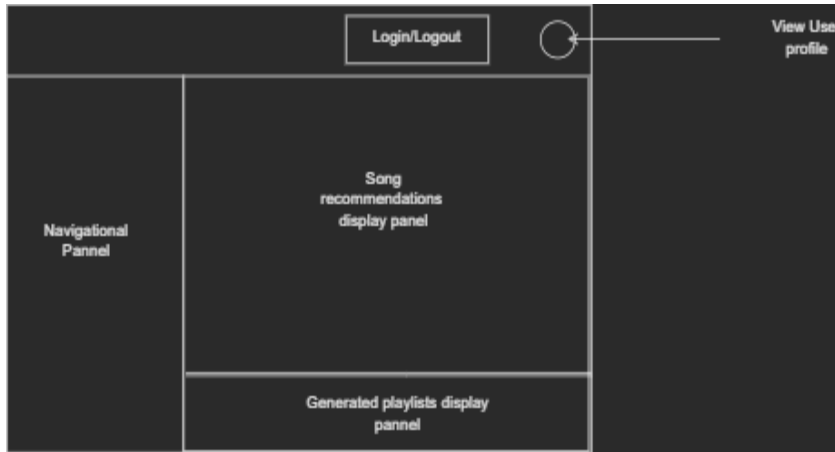
- The system should display music recommendations
- The system should allow for a user to login to spotify through the application in order to authorize our application to use their data
- The system should display a list of genres prior to making recommendation
- The system should allow for collection and storage of user ratings in the form of numbers or stars
- The system should create user taste profiles
- The system should make music recommendations
- The system should store information about new songs that were previously not in the dataset but were found through the obtained user data
- The system should provide links to play the recommended songs on spotify

## 4. External Interface Requirements

### 4.1. User Interfaces

- User interface will be implemented using the PyQt python library.  
The application consists of one home page divided into multiple frames
- The user will be made to authorize the application to use their spotify data by logging into spotify using the login button (Dig 1) and further prompted to allow the application access to the data fields listed on screen.
- The user will also be made to enter their age, region mandatorily and some favorite genres optionally
- Song recommendations will be displayed in the recommendation pane (Dig1)
- Auto-generated playlists of the recommended songs categorized by genre are displayed in the playlists display pane (Dig1)
- Navigation pane (Dig1) would be utilized by the user to change the types of recommendation
- The view profile button (Dig1) would be utilized by the user to change their preferences in genre or region manually if they so chose to

## PROJECT REQUIREMENTS SPECIFICATION



**Dig 1**

### 4.2. Hardware Requirements

- Minimum Requirements
  - Processor : Pentium 4 or higher, 2 GHZ
  - Ram : 1GB
  - Hard Disk : 4GB
- Internet connection (NIC)
- Input device to interact with system
- Display device to interact with the user (Monitor)
- Playback device for sound output (Speaker)

### 4.2. Software Requirements

- Windows/Linux Operating System running python 3+
- Spotipy python library for spotify user data retrieval
- Flask python library for creating api endpoint
- Scikit-learn python library for clustering and other machine learning tools
- Mongo Atlas database used to store data
- PyMongo python library to connect to mongodb

## PROJECT REQUIREMENTS SPECIFICATION

### 4.3. Communication Interfaces

- Utilizes HTTPS,TLS for communication with the spotify web service for user data retrieval
- Utilizes TCP socket,TLS for retrieving and storing music data to and from mongodb atlas

## 5. Non-Functional Requirements

- System should be able to respond to user actions and generate recommendation with minimal buffering
- System should not cause or trigger any events that will leave Operating System in unrecoverable state
- System should be able to deal with common scenarios such as loss of internet connection
- System should utilize HTTPS,TLS for secure retrieval of data

### Appendix A: Definitions, Acronyms and Abbreviations

- **NIC : (Network Interface Card)**, a hardware component, typically a circuit board or chip, which is installed on a computer so it can connect to a network
- **HTTPS : (Hypertext Transfer Protocol Secure)** is an extension of the Hypertext Transfer Protocol. It is used for secure communication over a computer network
- **TCP: (Transmission Control Protocol)** is an important network protocol that lets two hosts connect and exchange data streams.
- **TLS : (Transport Layer Security)**, the successor of the now-deprecated Secure Sockets Layer, is a cryptographic protocol designed to provide communications security over a computer network
- **RAM : (Random Access Memory)** is a system's short-term data storage

# PROJECT REQUIREMENTS SPECIFICATION

## Appendix B: References

[https://github.com/ProjectRecommend/docs/blob/master/design-docs/SRS\\_final.md#eir](https://github.com/ProjectRecommend/docs/blob/master/design-docs/SRS_final.md#eir)