Software Requirements Specification

for

Tune Tech Music Player

Version 1.0 approved

Prepared by

Afzal Mukhtar	Hritika Rahul Mehta	Meghana I.
PES2201800675	PES2201800024	PES2201800028
Section – A	Section – A	Section – A
Chapters: 2, 3, 4	Chapters: 1, 4, 7	Chapters: 5, 6

PES University EC Campus

January 12, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Music Player system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Intended Audience

There are four sections of the document following this. The next chapter, the Overall Description section, gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter. The users must pay attention to Chapter 5 to obtain knowledge about the functioning of the application.
The third section, Requirements Specification , of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.
This document can also help testers understand how the project is supposed to work hence they can identify the errors that may occur.

1.3 Product Scope

The music player software is designed for avid music listeners as well as people who use music as a background noise. This system can suggest new music based on the listener's previous preferences. There are options to create playlists based on the user's choice and some are created automatically based on the listening history.
Additionally, the playlists created for a given listener can be shared via a link on other platforms. The user can indicate their like and dislike for a particular song by using a button and based on the response the suggestions can be fine-tuned. This system contains a relational database containing the song name, artists, album and genre.

1.4 References

- Example SRS document-*https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database*
- Music player SRS example- http://musicboxdotnet.sourceforge.net/srs/download/mbdn_srs.pdf
- Guide to writing user requirements- https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

2. Overall Description

2.1 Product Perspective

In today's busy world most of the population spends maximum amount of time commuting to and from their workplace or educational institutes so they prefer to listen to some music or podcasts to pass the time. Our software can provide user friendly platform where people can listen to from a wide range of genres and artists and create unlimited playlists of their choice. A few podcasts will also be available after a small user fee is paid. The major components of the system relate to the user and artist information.

2.2 Product Functions

The Music Player allows users the functionalities of a simple music player, with some added functionalities for a better user experience. The product has many functionalities.

- Register User/Artist
- Search for a song/artist
- Users can play music what they like
- Song recommendations
- Adding new songs to the database

2.3 User Classes and Characteristics

2.3.1 User

Users or Listeners are the primary consumers of the music player system. They listen to the music posted by the artists and like or dislike a song based on their choice.

2.3.2 Artist

Artists are primary content administrators of the music player system. They post their music on the platform for others to listen, like and share the songs.

2.4 Operating Environment

The operating environment required for the system is as provided below.

- Operating System: Mac OS
- Database: SQL
- Platform: Python

2.5 Design and Implementation Constraints

2.5.1 User Interface Constrains

Using this system is fairly simple and intuitive. A user with basic computer navigation skills can easily work with all the functionalities provided by the system.

2.5.2 Hardware Constrains

The system will work on almost every system which supports python programming language.

2.5.3 Data Management Constraints

There should be a large memory storage service for storing the songs in the server.

2.6 Assumptions and Dependencies

Most of the music streaming platforms are rich in features which might or might not be used by many users. Our system focuses on the essential features of a music player only.

3. External Interface Requirements

3.1 User Interfaces

- **Front End:** Python
- **Back End:** SQL

3.2 Software Interfaces

Software Used	Description
Mac OS and Windows	These operating systems have been used for their user friendliness and responsiveness, and also to ensure it works uniformly in different platforms.
SQL Database	To store the music, and the user's data, we chose SQL database.
Python	To implement the project, we have chosen python as our language for its interactive and wide support on any system.

3.3 Communications Interfaces

This supports database update which allows user to get access to more and more songs.

4. Analysis Models

4.1 Context Diagram

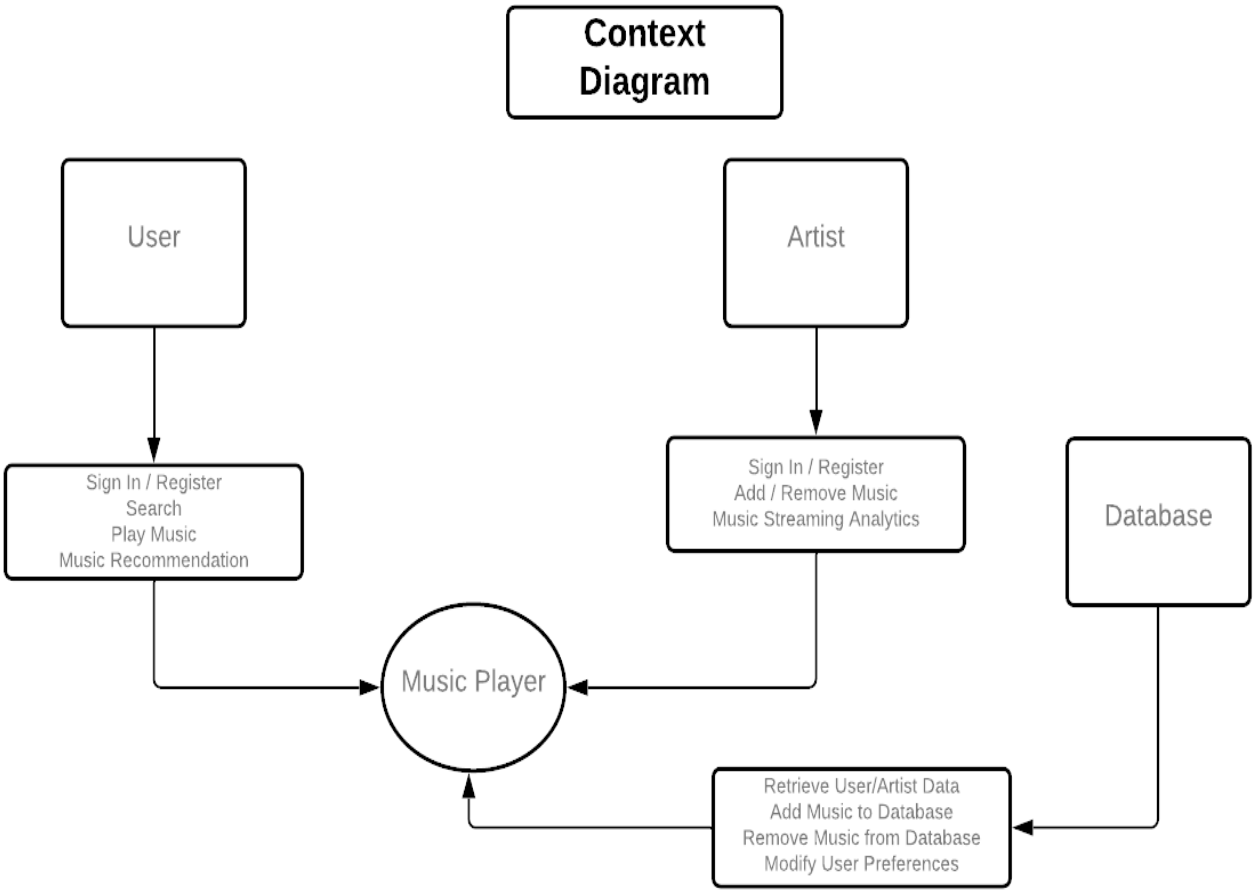


Figure 1: Context Diagram

4.2 Use Case models

4.2.1 User Login/Register

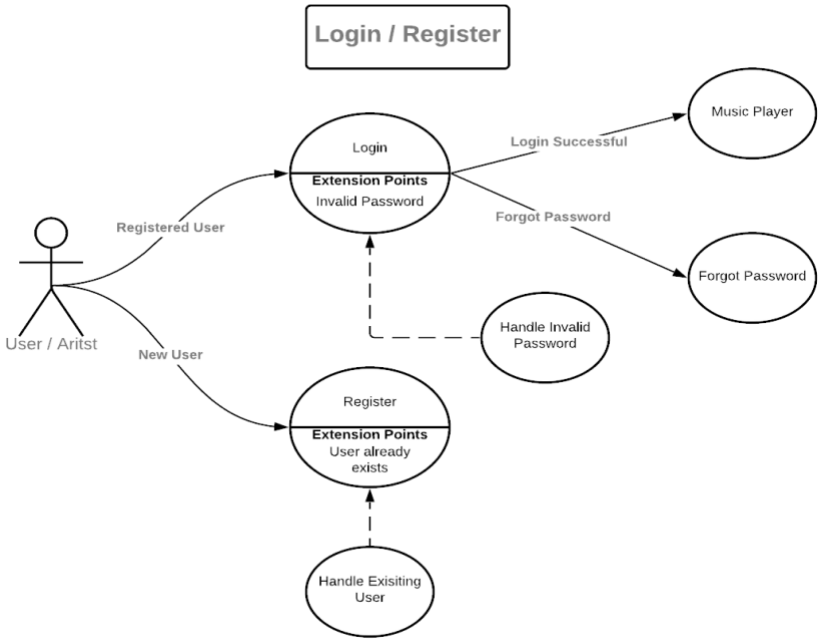


Figure 1: User Login/Register

4.2.2 Search

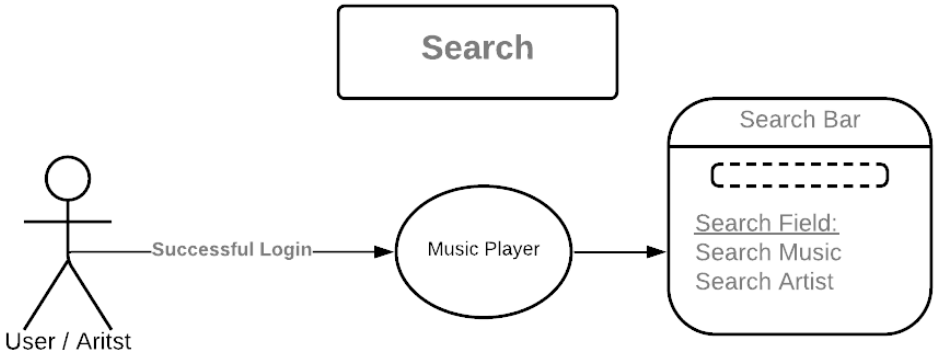


Figure 2: Search

4.2.3 Play Music

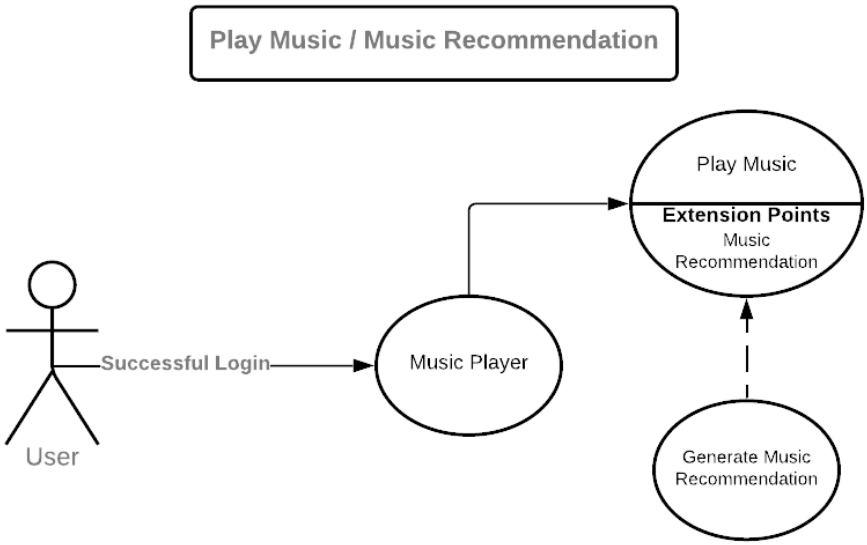


Figure 3: Play Music

4.2.4 Add/Remove Music

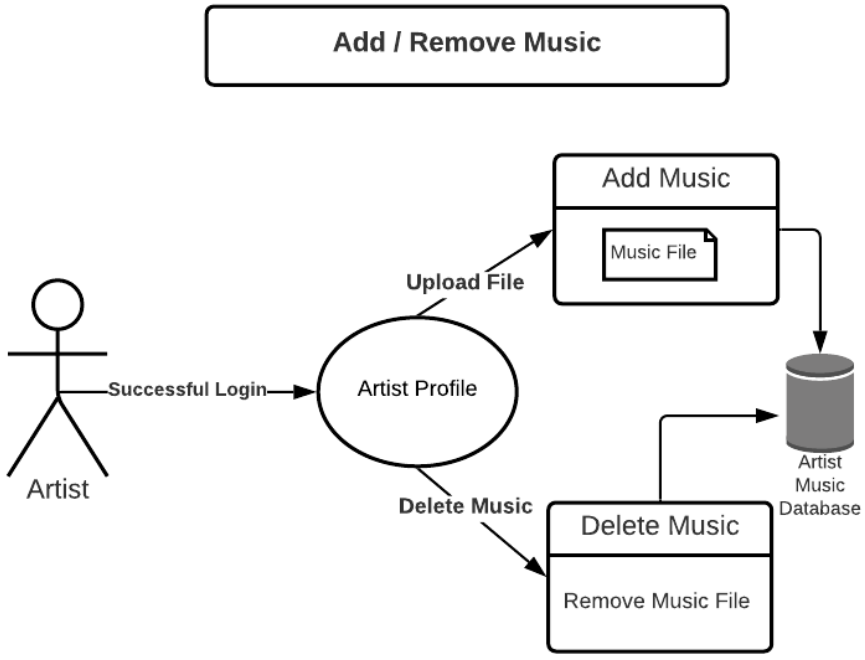


Figure 4: Add or Remove Music

4.2.5 Music Analysis

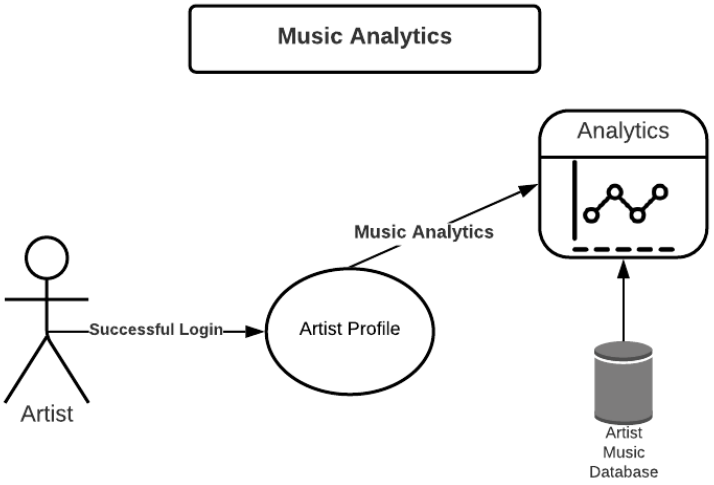
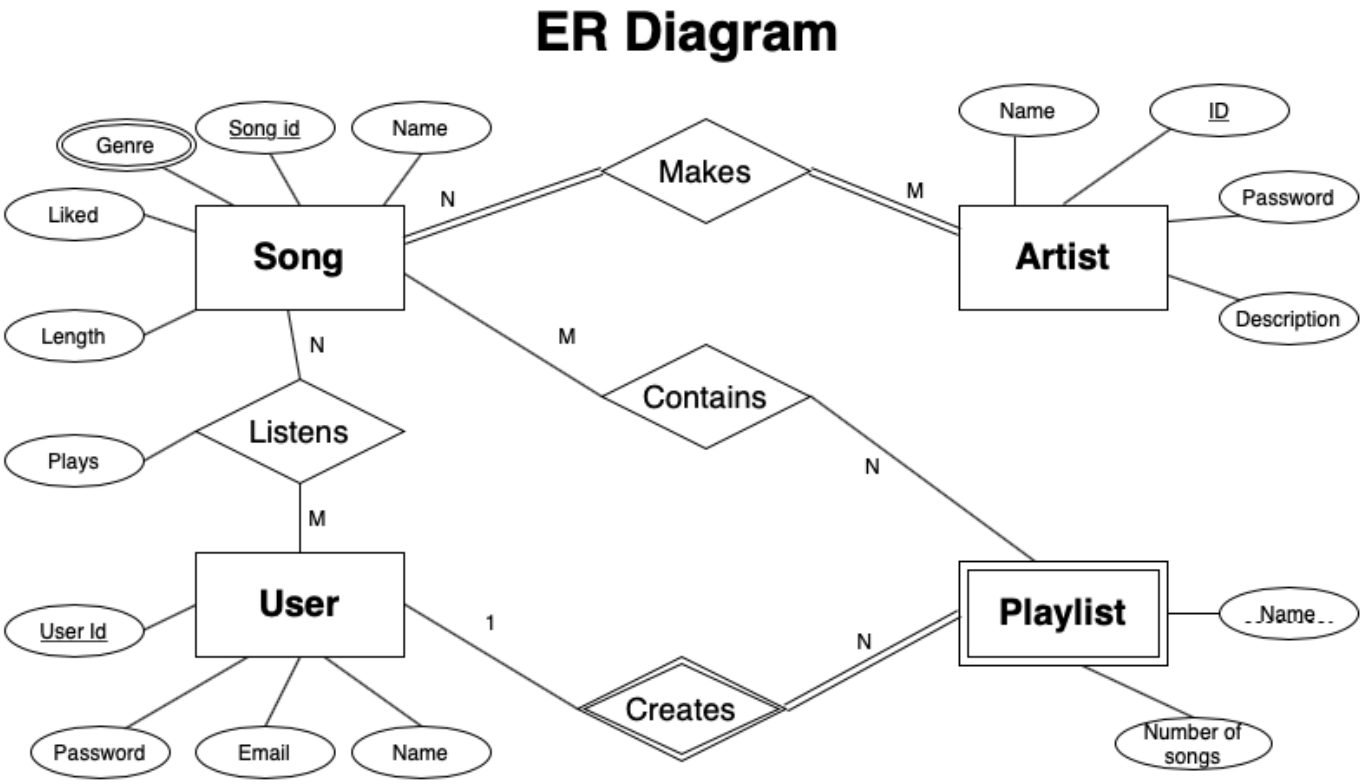


Figure 5: Music Analysis

4.3 ER Diagram



5. System Features

5.1 Search for Tracks and Play Music

5.1.1 Description and Priority

- The music player stores tracks, and enables users (listeners) to search for and play a selected track. Users may also—in addition to searching for tracks and playing the selected song—pause or stop track, play next or previous track, control volume, and seek track.
- Priority: HIGH
    - This is one of the cardinal requirements on which the Music Player System is built.

5.1.2 Stimulus/Response Sequences

- User (listener) launches app and
    - Searches for (using the search bar) and plays selected track, or
    - Navigates between tracks to find and play selected track.

5.1.3 Functional Requirements

Identifier	Name	Description
REQ-1	Play Track	Enables user to play selected track by clicking on “play”.
REQ-2	Pause Track	Enables user to pause track to play it again.
REQ-3	Stop Track	Enables user to stop track and play another song, or exit app.
REQ-4	Seek Track	Enables user to move along the track’s timeline to play track from selected timestamp.
REQ-5	Play Next	Enables user to play the next track by clicking on “next”.
REQ-6	Play Prev	Enables user to play the previous track by clicking on “prev”.
REQ-7	Repeat	Replay current track from the beginning.
REQ-8	Volume Control	Enables user to turn the volume up or down.

5.2 Create Playlists and Recommend Tracks

5.2.1 Description and Priority

- Automated creation of curated playlists based on user’s history (search and track history) and preferences (likes and dislikes).
- Recommend tracks based on user’s history and preferences.
- Priority: MEDIUM

5.2.2 Stimulus/Response Sequences

- Search for and play tracks.
- Like or dislike tracks.

5.2.3 Functional Requirements

Identifier	Name	Description
REQ-9	Recommendation System	• Recommends new songs to user. • Helps create a curated playlist for the user.

5.3 Add/Remove Songs

5.3.1 Description and Priority

- Allows artists (actors) to add/remove songs from the database.
- Priority: HIGH

5.3.2 Stimulus/Response Sequence

- Artist (an actor) uploads or deletes files.

5.3.3 Functional Requirements

Identifier	Name	Description
REQ-11	Add Song	Enables artists to add a song to the database.
REQ-12	Remove Song	Enables artists to remove a song from the database.

5.4 Music Analytics

5.4.1 Description and Priority

- Enables artists (actors) to view their songs, the number of streams, trends, and reactions received.
- Priority-MEDIUM

5.4.2 Stimulus/Response Sequence

- Artist posts song and views trends or analytics.

5.5 Database

5.5.1 Description and Priority

- A SQL database is used to store music and users' data.
- Priority-HIGH

5.5.2 Stimulus/Response Sequence

- Stores added tracks, created playlists, users' track history, and users' preferences (likes and dislikes).

REQ-1: Local-store (Database)

6. Other Nonfunctional Requirements

6.1 Performance Requirements

- Quickness: The Music Player System should be quick enough to handle users' requests and play tracks, without buffering or crashing unexpectedly.
- Robustness: Should be equipped to handle common errors—no internet connection, unsupported file type, or insufficient metadata.
- Failure Handling: Should recover swiftly from failures and crashes.

6.2 Safety Requirements

- Exception Handling: The software should warn users against performing actions things that might be hazardous to an individual's well-being, like increasing volume beyond the recommended threshold. Also remind users/artists to save any edited data before exiting the application.

6.3 Security Requirements

- Data protection: Safeguard users' data and login credentials.
- Prevent unauthorized access.
- Provide an encrypted connection.

6.4 Software Quality Attributes

- Efficient memory management: Ensure that system doesn't leak any memory.
- Reliability
- Availability
- Efficient Error Handling

6.5 Business Rules

- This is an Open-Source Software.
- Only registered users and artists may play music/add or remove tracks.

7. Other Requirements

Appendix A: Glossary

Artist- A music artist is someone who creates, performs and releases music either independently or through a record label.

Database- A database is an organized collection of data, generally stored and accessed electronically from a computer system.

Encryption- Encryption is the process of converting data to an unrecognizable form. It is commonly used to protect sensitive information so that only authorized parties can view it

Genre- a category of artistic, musical, or literary composition characterized by a particular style, form, or content.

Interface-the place at which independent and often unrelated systems meet and act on or communicate with each other.

Open source- denoting software for which the original source code is made freely available and may be redistributed and modified.

Playlist- a list of recordings to be played on the air by a radio station. also : a similar list used for organizing a personal digital music collection.

Recommendations- a suggestion that something is good or suitable for a particular purpose or job.

Robustness- the ability of a system to resist change without adapting its initial stable configuration.

Stream- Streaming media is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. The verb to stream refers to the process of delivering or obtaining media in this manner.

Track- A single song, or continuous musical selection on a vinyl record, or a single continuous musical selection on a CD or other recording medium.

Appendix B: Field Layouts

Information required to register the customer

Field	Length	Data Type	Description	Is Mandatory
Name	25	String	Full name of customer	Y
Email address	25	Alphanumeric	Valid email address	Y
User ID	15	Alphanumeric	ID for the application	Y
Password	20	Alphanumeric	Secret string of characters to authenticate access	Y

Information required to register the Artist

Field	Length	Data Type	Description	Is Mandatory
Name	25	String	Full name of artist	Y
Email address	25	Alphanumeric	Valid email address	Y
User ID	15	Alphanumeric	ID for the application	Y
Password	20	Alphanumeric	Secret string of characters to authenticate access	Y
Description	100	Alphanumeric	Short description of previous work	Y

Report Requirements

Registration Report
User ID
User name
Email ID
Date joined
Liked songs
Playlists made

Streaming Report
Artist ID
Artist name
Email ID
Description
Number of hours streamed
Number of users who listened to their songs
Genres of songs made

Appendix C: Requirement Traceability Matrix

Not applicable