



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

MUSIC PLAYER

SOFTWARE REQUIREMENT SPECIFICATIONS



Rhythmy

19MIS0411 – Dev Swaika

19MIS0263 – Niharika Gupta

19MIS0310 – Monish V

19MIS0212 – Shambhavi Swarup

A sample User Interface of Rhythmy



Software Requirements Specification

Table of Contents

1. The Team
2. Introduction
 - i) Purpose
 - ii) Intendent Audience
 - iii) Project Scope
3. Overall Description
 - i) Product Perspective
 - ii) User Class and Characteristics
 - iii) Process Model – Iterative
 - iv) Use Case Diagram
 - v) Operating Environment
4. System Features
 - i) Product Description
 - ii) Versions and Releases
5. External Interface Requirements
 - i) User Interfaces
 - ii) Hardware Interfaces
 - iii) Software Interfaces
6. Functional Requirements
7. Non-functional Requirements
8. Software Testing
9. Conclusion

1. The Team

This project was a cumulative work of 4 students from the MIS branch. They were assigned the following roles

Name	Reg No	Role
Dev Swaika	19MIS0411	Project Lead, Systems and Requirements Analyst
Niharika Gupta	19MIS0263	Developer, Project Manager
Monish V	19MIS0310	Designer
Shambhavi Swarup	19MIS0212	Software tester

2. Introduction

1.1 Purpose

The purpose of this document is to build an application (web and mobile) that gives you access to any mp3 track by any artist available on the internet.

1.2 Intendent Audience

This project is a prototype of a music player allowing free streaming online unrestricted to any age group and category of audience.

1.3 Project Scope

The primary purpose of music streaming application is to play music available in the databases of the service, compose custom playlists and suggest similar songs. We will have a database server supporting content of major and minor artists around the world. Above all, we hope to provide a comfortable user experience to all the music lovers.

1. OVERALL DESCRIPTION

1.1 Product Perspective

The music player database system stores the following information

- 1) User Details: Name, Phone number, email address, language and genre preferences, artist choices.
- 2) Artist details: Name, Genre, Songs, Albums, Movies, Soundtracks, Latest Releases
- 3) Playlist Description: Genres, Moods, Languages, Timelines, Search History, Trending, New Releases, Top playlists, Shows and podcasts, Downloads

1.2 User Class and Characteristics

Users of the system should be able to access, play and download audio tracks, displayed with details about the song, genre, album, The system will support two types of users privileges Client and Artists. Clients will have access to client functions, and the artists will have access to both client and artist functions.

The client should be able to perform the following functions:

- Login
 - i) Register
 - ii) Manage profile
 - iii) Change preferences
- Create a new playlist
 - i) Search for tracks
 - ii) Add tracks to a playlist
 - iii) Remove tracks
 - iv) Sort tracks – line up according to his/her wish
- Jump between playlists
- Offline player
 - i) Download tracks
 - ii) Delete tracks
 - iii) Sort tracks
 - iv) Search for songs
- Play, Pause, Loop, Forward, Back

- I) Forward in tracks
- II) Rewind
- Track Description
 - i) Artist
 - ii) Lyrics of the track
- Share tracks, playlists – via social media and
- Follow other playlists, artist

The artist should be able to perform the following functions:

A) Client privileges - since the artist is also a client, he also has the client privileges as stated above.

B) Artist privileges

- Profile
 - i) Registration
 - ii) Verification
 - iii) Manage
- Albums
 - i) Create album
 - ii) Upload tracks
 - iii) Delete tracks

1.3 Operating Environment

Operating environment for the airline management system is as listed below.

- client/server system
- Operating system: Android / iOS
- database: PostgreSQL
- platform: Python and JavaScript

2. SYSTEM FEATURES

2.1 Product Description

The music player application maintains information on artists and the user clients, music tracks and podcasts.

- i) Registration / Signing up – An essential option that identifies the user to synchronize his playlist and music in service.
- ii) Track Search – Give users the capability to find music they want to hear and using machine learning algorithms create playlists.
- iii) Catalogue – Provide users with a music catalogue to help them find out how many music contents the service is offering.
- iv) Push notifications – Provides push notifications to notify the users of new content available at length.

2.2 Versions and Releases

- i) Version 1.0
 - System Requirements
Android – 5 and above
iOS – 11 and above
 - Bit rate set to 96 kbps
- ii) Version 1.1
 - Improved Bit rates
 - Fixed Bugs
- iii) Version 1.1.2
 - Improved Bit rate
 - Fixed bugs
 - App made more responsive
- iv) Version 1.2
 - Improved Bit rates
 - Added themes for holidays and festivals

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 USER INTERFACES

- Front-end software: React Js
- Back-end software: Cassandra, PostgreSQL, PHP

3.2 HARDWARE INTERFACES

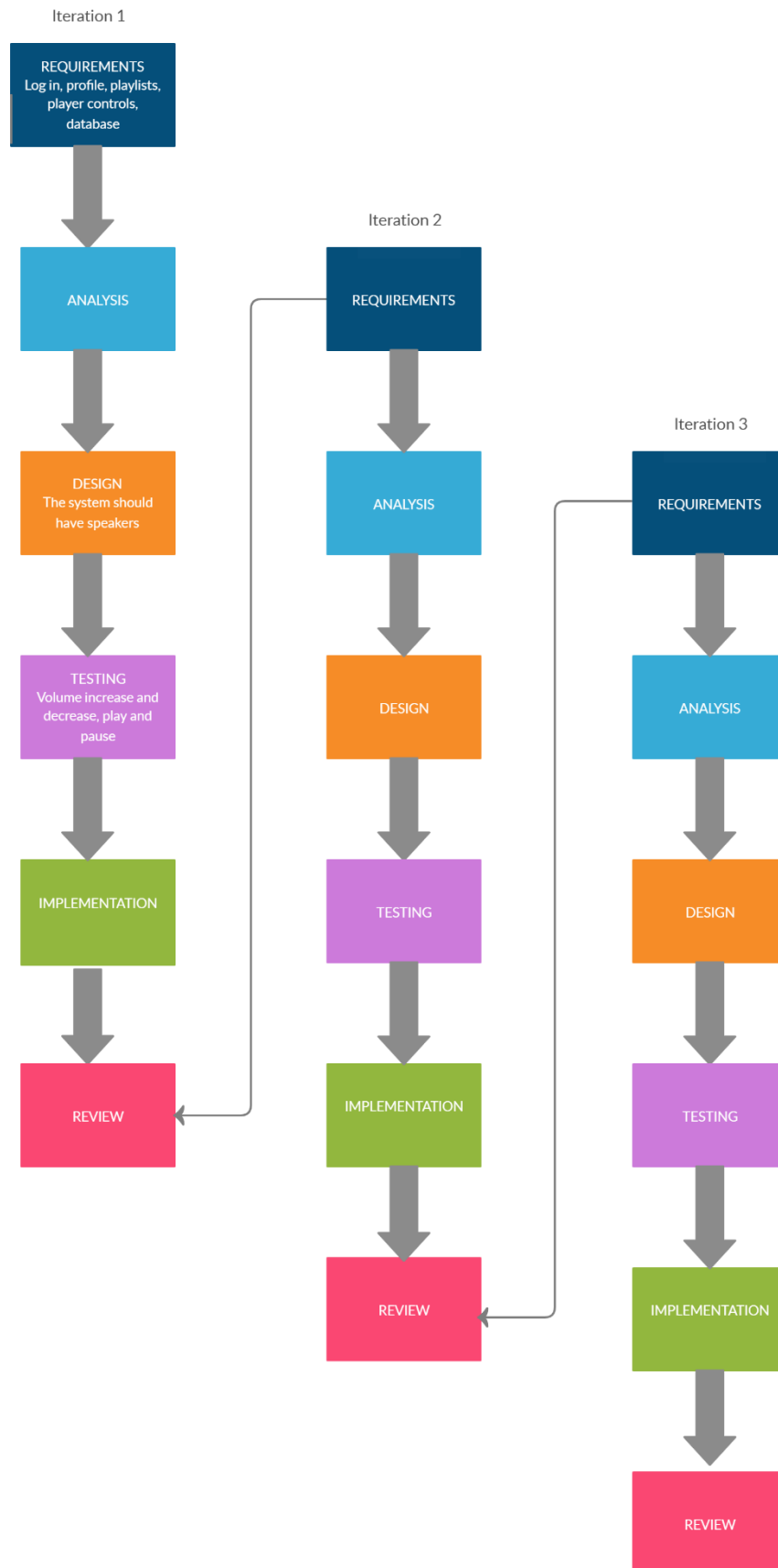
- Android , iOS
- A browser which supports CGI, HTML & Javascript.

3.3 SOFTWARE INTERFACES

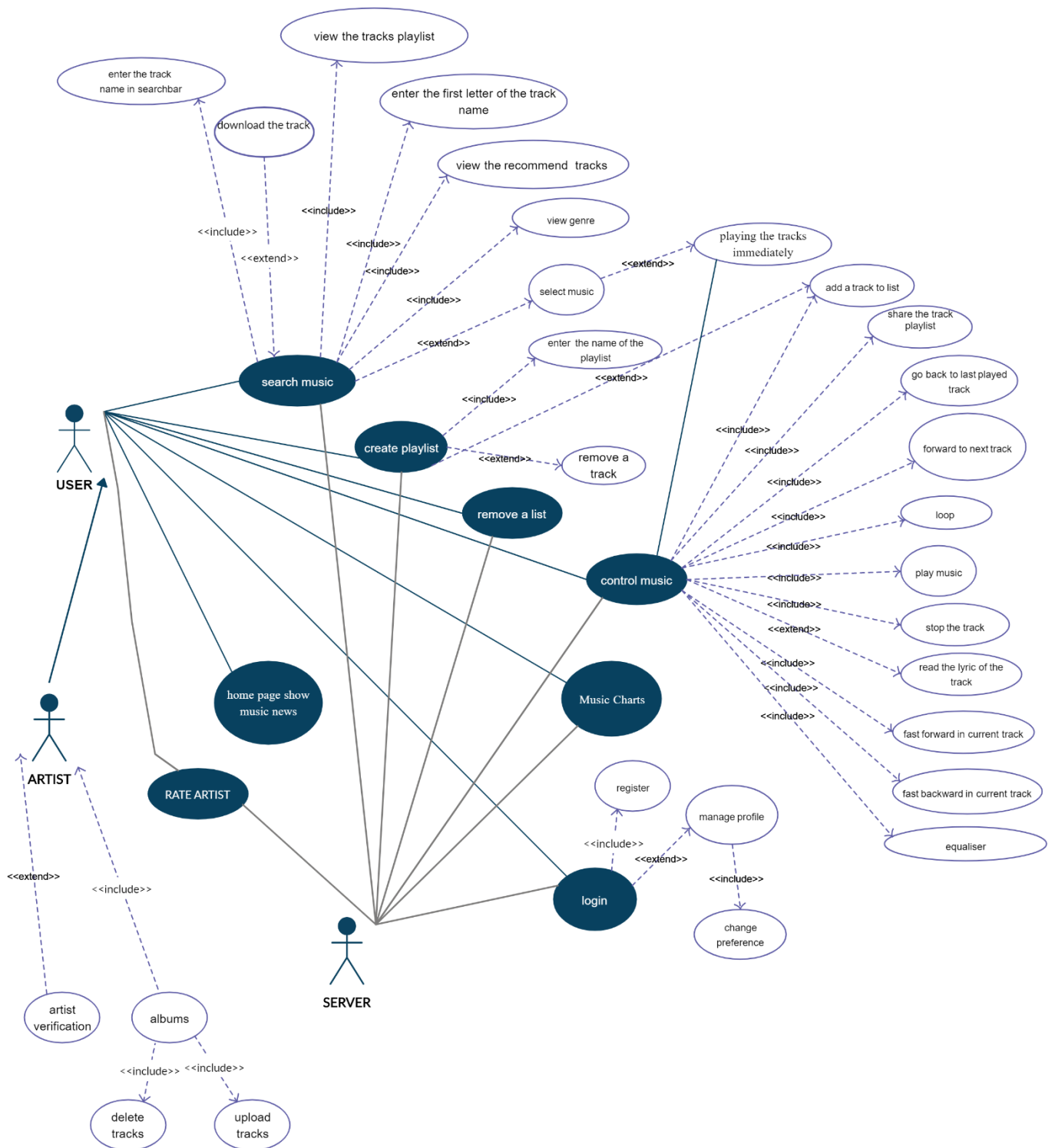
Following are the software used for the music player application

Software used	Description
Operating system	We have chosen both Android and iOS operating system for its best support and user-friendliness in most mobile phones.
Database	PostgreSQL as it is very robust and powerful.
Python	To implement this project we have chosen Python language for its automation abilities and the usage as a scripting language.

3.4 Process Model – Iterative Model



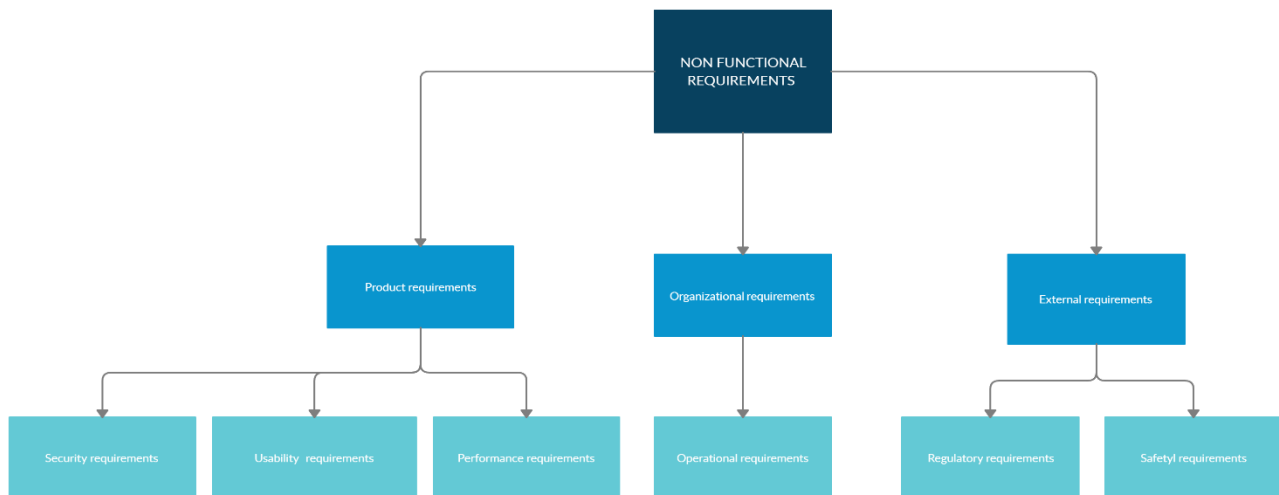
3.4 Use Case Diagram



4. Functional Requirements

- Social Login
- Edit user profile
- Add streaming services profiles
- Manage audio uploads and downloads
- Search in connected music service
- Create / edit playlist
- Music player with controls (Play, pause, forward, rewind)

5. Non-functional Requirements



6. SOFTWARE TESTING

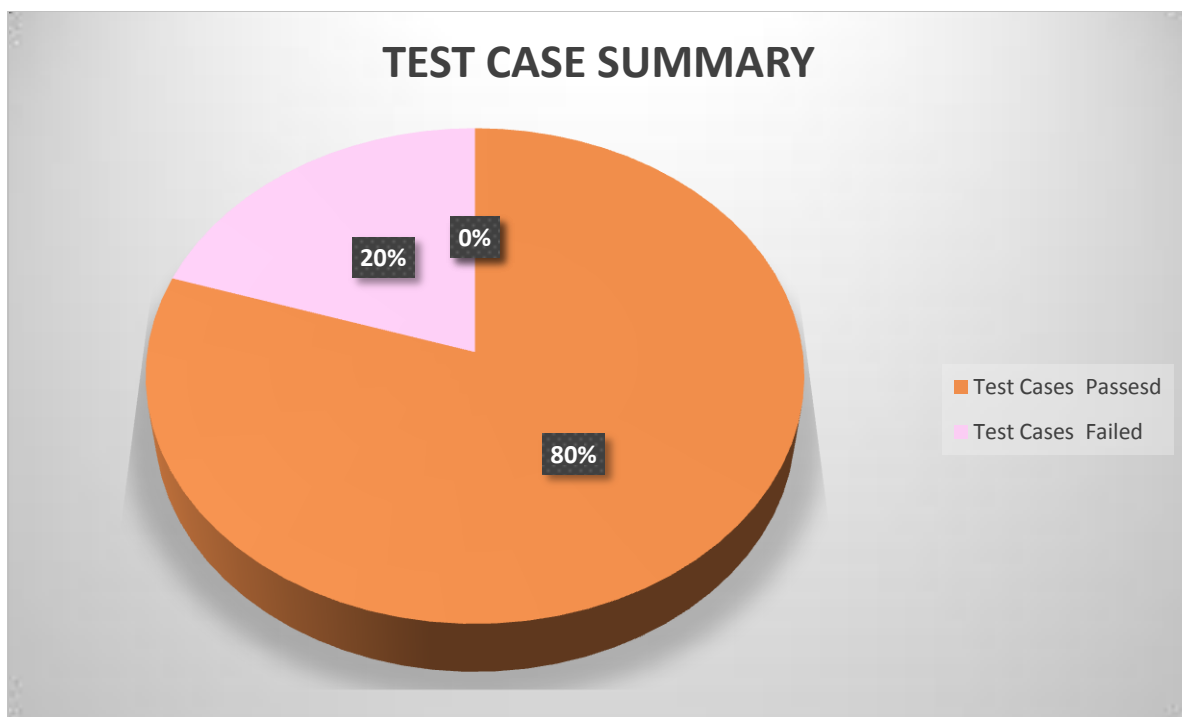
Report

Test Case ID: 05156484	Test Designed by: Shambhavi Swarup
Test Priority: High	Test Designed date: 23/02/2020
Module Name: Rhythmy	Test Executed by: Shambhavi Swarup
Test Title: Rhythmy Application Test Run	Test Execution date: 23/02/2020
Description: Music Player Functionality Test	

Pre-Condition: User and artist has valid username and password
Dependencies: none

Step	Test Steps	Expected Results	Actual Result	Status (Pass/Fail)	Notes
1	Navigate login page			PASS	
2	Provide valid username	Username verified	Username verified	PASS	
3	Provide valid password	Password verified	Password verified	PASS	
4	Click on Login button	Login success	Login success	PASS	Works when user double clicks it.
5	Asking user's preference	List of artist with images. List of various genres.	List of artist with images. List of various genres.	PASS	
6	Home Page display	Songs as per user preference	Songs as per user preference	PASS	
7	Push Notification	Keeps user updated about the app	Error	FAIL	This feature isn't present in application.
8	Search Button	Opens keyboard to allow user input	Opens keyboard to allow user input	PASS	
9	Search Operation: Displaying recently searched	Recent songs searched	Recent songs searched	PASS	
10	Search Operation: Opening desired song	Plays the that user inputs	Error	FAIL	All songs aren't available in application
11	Create playlist button	Creates empty playlist and ask for its name	Creates empty playlist and ask for its name	PASS	
12	Adding and Removing songs in playlist	Adds and removes the selected songs in playlist	Adds and removes the selected songs in playlist	PASS	
13	Ordering songs in playlist	User can order songs as per their wish	Songs reset their order to default	FAIL	
14	Audio Control	User can pause, play, move forward and backward the song	User can pause, play, move forward and backward the song	PASS	
15	Song Sharing	Sharing songs among people	Sharing songs among people	PASS	
16	Song Download	Saves song for offline use	Saves song for offline use	PASS	

17	Sending invitations to friends	A readymade message to invite friends.	Error	FAIL	Users should be able to invite friends for easy sharing of playlist.
18	Artist Profile	Profile Verification	Profile Verification	PASS	
19	Uploading Songs	Uploading Songs by artist	Uploading Songs by artist	PASS	
20	Ask before quitting	App should confirm user before quitting	App should confirm user before quitting	PASS	



7. CONCLUSION

SRS document is an important part of the process of project development. Moreover, it is a prerequisite for creating the following design documentation. In this document, we have provided the information about general product description, data elements that the product deal with, specific requirements like product interfaces and the functions. In addition, general behaviour of the product has been explained in order to make it easy for the customer to understand the product's usage clearly. This document has been created through the help of various researches and depending on the demands of the customer, keeping in view of his requirements. However, some specifications are prone to be changed in the future, which have been enlisted as limitations in the testing of this app, for a better user experience.
