Software Requirements Specification

for

Online Music Management System (OMMs)

Version 1.0

Prepared by: Group S6_T9

ID	Name
201901076	Utsav Ladani
201901090	Pandar Mayur
201901131	Bhavya Solanki
201901304	Dev Joshi

Dhirubhai Ambani Institute of Information and Communication Technology

21 Sept, 2021

Table of content

Table of content **1**

Introduction **3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product scope 4

1.5 Description 5

2. Fact finding Phase **8**

2.1 Background Readings 8

2.1.1 Description of each reading done 8

2.1.1.1 YT Music 8

2.1.1.2 Spotify 11

2.1.2 References: 13

2.1.3 Combined Requirements 14

2.2 Interviews 14

2.2.1 Interview plans and summaries 14

2.2.2 Combined Requirements 23

2.3 Questionnaires 23

2.3.1 Google Form 23

2.3.2 Summary 25

2.3.3 Combined Requirements 28

2.4 Observations 28

2.4.1 Observation Summary 28

2.4.2 Combined Requirements 30

3 Fact FInding Chart **31**

4 List Requirements **31**

5 User Classes and Characteristics **33**

6 Operating Environment **33**

External interface requirements: 34

7 Product Function **34**

Client Application 34

Server Application 35

Database System 36

8 Privileges **36**

Listener 36

Artist 36
Production Company 37
Admin 37

9 Assumption 37

10 Business Constraints 38

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the **Online Music Management System (OMMS)**. The OMMS (Online Music Management System) is a software that facilitates Good user experience with synchronization of both user experience and copyrights while providing services like online music management, legal downloads, artists' management. There are several other applications available in the market(Jio Savan, YT music Spotify etc.) that either provide some specific services or large scale integrated solutions. This Software differs from the rest in a way that we give more power to the users remaining within the copyrights circle and also some other functionality. This document is for both developers and users of this system. This software provides users to give review and rating to a song. It also provides copyright management for different songs and albums.

This document contains the basic functionality of the frontend and backend of OMM system, system flow or working principles, error handling, efficient database design, and common problems.

1.2 Document Conventions

This document is written in Latex. Whole document is written in Arial font with simple text having fontsize of 12.
Important terms In this document are written in bold style.

1.3 Intended Audience and Reading Suggestions

This SRS doc is helpful for developers who want to develop a music application with an online database, Database architecture who want to make an efficient database system, Product managers, startups, etc.

This document is written in a few sections. Product sections are for product managers, startups, etc. These sections describe the use case of the product, user charactrics, etc. Design and architecture sections are helpful for developers to develop efficient online databases. This section gives the information about schema, data type and data. Other sections are written with simple language so easily understood by any readers who have basic knowledge of databases.

Any reader and developers are welcomed to use this document for their work. If a reader has any suggestions about this document or this OMMS system they can contact us on ommsproduct@gmail.com.
If you want to send lyrics in your language use the above email address to send lyrics to us. We will verify lyrics and add it to our system if it is correct. To appreciate this effort of the user we will provide a 2 month premium subscription to the user.

SRS Docs includes:

1. Description of OMMS
2. Details of user interface
3. System features
4. Efficient Database design
5. Privileges of Database
6. Business rules

1.4 Product scope

According to the survey 60-70% of youth listen to music, so there is always a need for a music system which can fulfill the requirements of people. They want to listen to music anywhere, anytime. To satisfy this requirement we develop OMMS, so that our collection of music is accessed by anyone, anywhere, anytime with internet connection and no need to fill device space with many songs, just select a song from the list and listen to it. We use steam to transfer music over the internet and it takes only a few KBs of space. Also the online database is very easy to maintain and update with the latest songs. This OMMS system touches various different Operating systems (Android to Microsoft). We are also providing the advanced facility of search in our large music database.

We are concerned that our system does not violate any copyright rules, so users can't download the song on his/her device without permission of the creator to preserve the copyright of music. So there is no issue related to piracy of songs. If a user wants to download the song, then a premium account is required. Only users with a premium account can download the song, but we store it in encrypted format so they can't distribute their own. And this encryption is different for each device. So there is total security.

To make money, the application contains ads and if a user wants to remove these ads, then he/she must buy a premium account, which provides an add-free user experience.

Till now our OMMS is a standalone system, We can make more money by integrating our OMMS system with telecom. Companies, and giving their users caller tunes from our Database. More details about this are given in the **Business constraints** section.

1.5 Description

The Online Music Management System will provide a large collection of songs to every user who is connected with our service over the internet. Users can search music and/or select from a given list of songs and that song will play in the appropriate app or system (web / android / desktop). As far as the IT214 subject is concerned, this project will be database oriented. Our goal is to make efficient database design, so that maximum users can connect with our service using minimum resources.

The system consist of four parts;

1. **User interface:** installed on device or accessed via browser
2. **Server:** connect application and database, handle user request, stream song, register user,etc.
3. **Database:** Store the songs and user information, provide some basic functions to modify and search into the database. It uses a PostgreSQL database.

Basic Workflows of Client Application

- **Register/SignUp:** Every user needs to register in application when he/she operates for the first time.Without registering users can listen to only 20 songs.

- **SignIn:** Every time when a user opens an application, he/she must have to sign in using their credentials.

- **Suggestion/Song List:** After signing in, users get the list of some song according to their preference and history.

- **Search Bar:** Search the song in our database using various filters like artist name, rating, views, duration, date, movies, etc.

- **Music Player:** Play, pause, prev, next, queued, etc. functionalities are handled by the music player.

- **User Profile:** show the information about user, account type(premium/comman), update profile, etc.

- **Upload Song:** Users can upload songs on the server.

- **Money:** Give the money to artists according to views and rating.

Basic Workflows of Server

- **Authentication Handling:** all authentication related functionality like register user, check whether the user's credential is right or not, etc. are done by this part.

- **Search request:** Send the result of the search query to the application in JSON format.

- **Song request:** Start the streaming of songs on a particular user's device, download, upload can be done via this service.

- **Update:** Update song rating, views,etc. and user data also.

- **Money:** All money transactions controlled by this part.

Basic Workflow of Database

- **User:** Add, delete, update user and also update account to premium account.

- **Song:** Add, delete, update song.

- **Money:** Money transaction.

Online Music Management System(OMMS) will help to categorize using song name, artist, albums, rating, type, time, duration, number of listener, etc. Admin will be able to add songs. Users will be able to create their favourite playlists, Download and Share songs. This system also has features for singers and authors of albums. If some other singer wants to use a song or it's lyrics in his song, He will have to take permission from the owner of that song. This system will alert the owner of a song or album when someone publishes a song or album containing the song of the owner. Based on that if he has not taken permission from the owner he will be charged a penalty. We are providing **Repeat Current song, Repeat playlist, Shuffle playlist** like features for users. Users can make their own playlist by adding their favourite songs. The system also has a **play in background** feature so a user doesn't have to keep the application open while listening to music. To use this play in background feature the user has to buy premium.To set any song for caller tune(song for incoming call) the user has to buy premium.

There are several other applications in the market which are similar to our OMMS software.these applications/softwares provide several functionalities like search music,add different songs in the favourite song list,set caller tune,lyrics,etc.

Search Music: By this function you can search the music song whichever you want.

Favourite song list:some applications provide a Favourite song library/folder and in this library you can add the song so it notes the song to your favourite song.

Caller tune:this function provides the ability to set the song for incoming calls.

Lyrics:some other applications provide lyrics of songs.

2. Fact finding Phase

2.1 Background Readings

We have analyzed two products from the market.

2.1.1 Description of each reading done

2.1.1.1 YT Music

YT Music is a music streaming service developed by YouTube. YT Music provides a tailored interface for the service, oriented towards music streaming. It allows users to browse through songs and music videos on YouTube based genres, playlists and recommendations. YT music is available for android (application), Desktops (Website).YT Music service consists of three parts.

1. **User Interface:** Device application or website
2. **Database:** Stores songs and user information. This information is stored in Youtube’s cloud storage.
3. **Server:** It connects application to Database. It provides many features like search, share, download, sign in etc.

Basic workflows of YT Music application

- **Sign Up or Register :** Users don’t need to register to listen to songs on YT music, But If a user wants to use his premium features he should have signed up in YT music.

- **Login or Sign in :** Again if users just want to listen to songs and don’t want to use premium features they can do it so without log in. to use premium features one should have to login to his account.

- **Search bar :** Searches songs in Youtube database by using various filters like song name, artist name, movie or album name etc. It uses Google’s Powerful algorithms for searching.

- **Suggestions/Recommendations :** It provides suggestions/recommendations based on various things like Trending songs, Popular, albums, Your past listened songs, most listened songs etc.

- **Library :** Stores recent activity of users. Users can also view history.
- **Account** : This has many features like switch account, create channel, settings, Your channel etc

- **Create Channel :** Users can create their own channel on YT music.

- **Upload Music :** Users can upload their music in YT music.

- **Upgrade to premium :** Users can buy premium from here

- **Help :** Gives Users help about application or website interface.

- **Terms and privacy policy :** Provides users terms of usage (for example a user must be 13 years old) and gives information about privacy that YT Music provides to users.

- **Feedback :** Users can send feedback about anything in YT Music.

- **Money :** Gives money to the channel owner according to certain rules.

- **Restrictions :** It has some restrictions for some premium songs. Users need a premium account to access those songs.

- **Connect to a device :** Users are able to connect YT Music with their smart TV.

- **Ratings :** Users can give likes or dislikes based on their choice.

- **Music Player :** Basic Music player that can play, pause, skip songs. It also gives features like repeat this song, shuffle etc.

Basic workflows of Database

- **Users :** Stores information about user account, also stores information about user's premium account, It adds and deletes user according to server requests.

- **Storage :** YT Music stores its data in YouTube's Cloud Storage.
  .
- **Song :** Stores music, Add music, Delete Music.

- **Money :** Stores transaction information of its users.

Basic workflows of Server

- **Search request :** It finds Songs in Database using advanced algorithms of google or youtube and sends results to YT Music application or website.

- **Connect request :** It connects YT Music with other devices like Android TV, Desktop, Laptop etc.

- **Update request :** It updates songs, song views etc. based on request from application or website.

- **Song request :** It provides facilities like Streaming song in a particular device, share song, download song etc.

- **Money :** It does all transactions successfully without any errors.

- **Authentication Handling :** All tasks like signing in user, logging in user, checking premium of user etc. are done by this part.

This application also provides premium features which enables ad-free playback, background audio only playback and downloading songs for offline playback. This service is designed for those users who primarily listen to music through YouTube. This YT Music works on several devices having different operating systems. It recommends Popular songs to users. Searching on the base of Lyrics, Song name, Singer, Album name etc. is very fast in YT Music. It also provides a voice search feature in which we can say anything about the song and it will give us the best matching result of that song. It provides a selection of bit rate at which song is streaming or downloading to premium users. You can download YT Music in Smartphones from their appropriate app store. For other devices, below is the website of YT Music service.

https://music.youtube.com/

2.1.1.2 Spotify

Spotify is an application that provides the service to listen to music online. Spotify streams the songs over the internet so that people can listen to it everywhere. It allows users to stream music, share music with everyone who is using Spotify. This application provides premium features which allow users to play ad-free playback. It also provides digital copyright restrictions for recorded songs/music.it also supports searching the music based on artists, albums and genres. When you search for some music, it provides suggestions based on artists and songs and albums. On Spotify, you can create your own playlist and also stream it. Based on your music or playlist users can give you ratings and likes.on Spotify, some basic features are free with advertising but for the additional features like offline music listening and commercial-free listening, you need to buy a subscription.Spotify can be represented in three parts.

1. **User Interface:** two types of interfaces are possible 1. Application 2. website

2. **Database:** stores songs, user information, favourite song list, etc.

3. **Server:** server connects application or website to the database and handles some user requests like pause, play, change song, search song etc.

Basic Workflow of Spotify application.

- **Register or sign up:** Every user needs to register on Spotify when the user uses it for the first time. The listened songs, favourite songs and others are all linked with the registration.

- **Sign In:** when a user uses the website interface for Spotify then the user needs to sign in every time to fetch the user details but in the application sometimes you need to sign in.

- **Search song or search bar:** On Spotify, you can search songs based on the artist, song name, Lyrics, album name etc. this searching happens in the databases with various commands and algorithms.

- **Update account:** users can update their accounts on spotify. If a user uses a free account then He/she can update his/her account to a premium account.

- **Reset password:** in some cases, if the user doesn’t know their account password then they can reset the by forgot password option.

- **Suggestion:** when the user searches for some song, album or anything then in the below search bar Spotify gives suggestions Based on trending, popular artists, albums, songs, podcasts, recommendations and also your past history.

- **Favourite songs list:** user can add the song in his favourite songs list. So, when a user needs to listen to his/her favourite song , he/she does not need to search for the song everytime. He/she can listen to it from his/her favourite songs list.

- **Create playlist:** users can create their own playlist and also can stream the song which is created by them.

- **Copyright restrictions:** if a user wants to use some other user's lyrics or song then he/she(user) needs to take permission from the lyrics's owner.if user use the song's lyrics without lyrics's owner permission then this will be called infringement and user break the rules of terms and conditions rule. Spotify can take legal action against him/her.

- **Connect to a device:**You can connect some other device with Spotify like TV, speaker etc.
  .

- **Money :** gives the money to the playlist owner according to some specific rules.

- **Help:** users can take help about the application interfaces.

- **Feedback:** users can give their feedback related to the application.

Basic workflows of database

- **User:** it stores account information and user related details. It adds and deletes the user according to registration on spotify.

- **Money:** all the money transition of user's is stored in this area.

- **Store:** spotify uses all user's cache memory and also hard drive to store the information about listened songs history, liked song etc.

- **Song:** new songs will be added in this database and deleting will also happen in this database.

Basic workflows of server

- **Connect request:** when we connect Spotify with other devices like TV, speaker. Then it will connect via connection request from the server.also when the user enters in spotify then a connection request will arise.

- **update request :** it updates song's information like views, likes, lyrics and also updates the songs.for particular user, if user updates his/her profile then this is also handled by this update request.

- **Money related or transaction related request:** it handles all the errors related to the money transaction.

- **Search request:** all the searching related work will be handled by this request like searching suggestion. This searching will happen by some algorithms.

- **Authentication handling:** all user's premium plans remaining days, sign in or loging related information will be handled by this

- **Song request:** this handles the sharing song, download songs and streaming songs requests.

On the Smartphone, you can download Spotify from the appropriate app store and for desktop, you use their website which is given below.

https://www.spotify.com/us/

2.1.2 References:

YT music :

- By using YT music website, Application.
- https://en.wikipedia.org/wiki/YouTube_Music
- https://www.pocket-lint.com/apps/news/youtube/144541-what-is-youtube-music-youtube-s-new-music-streaming-service-explained
- https://www.androidpolice.com/2020/06/09/hands-on-youtube-music-upload/

Spotify :

- https://en.wikipedia.org/wiki/Spotify
- https://pageflows.com/product/spotify/
- Using the spotify application we wrote some functions or features.

2.1.3 Combined Requirements

- To check information about users' premium accounts we should have a good authentication handler in our OMMs system.

- We should make our system platform Independent. (For example, our system should work on various Operating systems and various devices.)

- We should have better storage options to store songs in our Database of OMMs.

- We will need a strong server for our OMMs which can communicate with Database nicely.

- For Online transactions between system and users, we will need a better transaction management system.

2.2 Interviews

2.2.1 Interview plans and summaries

From the next page Interviews of various roles are given. For each Interview, First Interview plan is written and then Interview summary is written.

OMMs: (Actual) Interview Plan 1

System: OMMs (Online Music Management System)

Project Reference:

Interviewee: 1) Dharmik Solanki(**Actual**)

Designation:User of OMMs

Contact Details: solbha9@gmail.com.

Organization Details: Student at DAIICT

Interviewer: 1) Mayur Pandar
Designation: OMMs admin

Interviewer: 2) Dev Joshi
Designation: OMMs admin

Date: 05/10/2021 **Time:** 21:00

Duration: 60 minutes **Place:** Online (Google meet)

Purpose of Interview:

Simple meeting to know user requirements for OMMs.

Agenda:

Users' expectations from an Online Music Management System.

Users’ suggestions about improving these kinds of systems.

Follow-up actions

Documents to be brought to the interview:

Nothing

OMMs: (Actual) Interview Summary 1

System: OMMs (Online Music Management System)

Project Reference:

Interviewee: 1) Dharmik Solank(Actual)

Designation: User of OMMs

Contact Details: solbha9@gmail.com

Organization Details: Student at DAIICT

Interviewer: 1) Mayur Pandar

Designation:OMMs admin

Interviewer: 2) Dev Joshi

Designation: OMMs admin

Date: 05/10/2021 **Time:** 21:00

Duration: 60 minutes **Place:** Online(Google meet)

Purpose of Interview:

Simple meeting to know user requirements for OMMs.

1. System should have a beautiful UI and eye protecting theme.
2. All types of songs should be available in the system.
3. System should not face network related issues if the user has a good internet connection.

OMMs: (**Actual**) Interview Plan 2

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Vrinda Mayatra(**Actual**)

Designation: User of OMMs

Contact Details: vrinda22mayatra@gmail.com

Organization Details: Student at DAIICT

Interviewer: 1) Bhavya solanki

Designation: OMMs admin

Interviewer: 2) Utsav Ladani

Designation: OMMs admin

Date: 06/10/2021 **Time:** 17:00

Duration: 60 minutes **Place:**Online(Google meet)

Purpose of Interview:

Simple meeting to know user requirements for OMMs.

Agenda:

Users' expectations from an Online Music Management System.

Users’ suggestions about improving these kinds of systems.

Follow-up actions

Documents to be brought to the interview:

Nothing

OMMs: Mock/Actual Interview Summary 2

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Vrinda Mayatra(**Actual**)

Designation:User of OMMs

Contact Details: vrinda22mayatra@gmail.com

Organization Details: Student at DAIICT

Interviewer: 1) Bhavya Solanki

Designation: OMMs admin

Interviewer: 2) Utsav Ladani

Designation:OMMs admin

Date: 06/10/2021 **Time:** 17:00

Duration: 60 minutes **Place:** Online(Google meet)

Purpose of Interview:

Simple meeting to know user requirements for OMMs.

1. System should have less ads.
2. System should have less price for premium subscription.
3. System should give discounts on the price of premium subscriptions.

OMMs: (Role Play) Interview Plan 3

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Camila Cabello(**Role Play**)

Designation:Singer

Interviewer: 1) Mayur Pandar
Designation: OMMs admin

Interviewer: 2) Bhavya Solanki
Designation: OMMs admin

Date: 06/10/2021 **Time:** 21:00

Duration: 60 minutes **Place:** Online(Google meet)

Purpose of Interview:

Artists' Expectations from the System

Agenda:

Artists' Expectations from the System

Follow-up actions

Documents to be brought to the interview:

Nothing

OMMs : Mock Interview Summary 3

System:OMMs(Online Music Management Systems)

Project Reference:

Interviewee: 1) Camila Cabello(**Role Play**)

Designation: Singer

Interviewer: 1) Mayur Pandar
Designation: OMMs admin

Interviewer: 2) Bhavya Solanki
Designation: OMMs admin

Date: 06/10/2021 **Time:** 21:00

Duration: 60 minutes **Place:** Online(Google meet)

Purpose of Interview:

Artists' Expectations from the System

- According to Artists', The system should have a high user base who listens to songs.
- System should have high Production companies who make contracts with artists.

OMMs: (Role Play) Interview Plan 4

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Daniel Ek(**Role Play**)

Designation: CEO at Spotify

Interviewer: 1) Utsav Ladani **Designation**: OMMs admin

Interviewer: 2) Dev joshi **Designation**: OMMs admin

Date: 07/10/2021 **Time:** 21:00

Duration: 60 minutes **Place:** Online(Google meet)

Purpose of Interview:

Reputation of business and robustness of the System

Agenda:

To develop Business between company and Owner of the system

Initial ideas

Follow-up actions

Documents to be brought to the interview:

Nothing

OMMs: Mock Interview Summary 4

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Danieal Ek(**Role Play**)

Designation: CEO at Spotify

Interviewer: 1) Utsav Ladani
Designation: OMMs admin

Interviewer: 2) Dev joshi **Designation**: OMMs admin

Date: 07/10/2021 **Time:** 21:00

Duration: 60 minutes **Place:**Online(Google meet)

Purpose of Interview:

Reputation of business and robustness of the System

1. System should provide better privacy to albums or songs of the product company.
2. System should be able to generate high profit for the Production company.
3. System should be able to do all transactions smoothly.

OMMs: (Actual) Interview Plan 5

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Yash Vasani(**Actual**)

Designation:Enthusiastic Linux system administrator

Contact Details: 201901081@daiict.ac.in

Organization Details: Student at DAIICT

Interviewer: 1) Utsav Ladani **Designation**: OMMs admin

Interviewer: 2) Mayur Pandar
Designation: OMMs admin

Date: 08/10/2021 **Time:** 17:00

Duration: 75 minutes **Place:** Online(Google meet)

Purpose of Interview:

To identify problems and requirements regarding security in the OMMS system.

Agenda:

Security problems in system and any other concerns

Current security procedures

Follow-up actions

Documents to be brought to the interview:
Nothing

OMMs:Actual Interview Summary 5

System: OMMs(Online Music Management System)

Project Reference:

Interviewee: 1) Yash Vasani(**Actual**)

Designation: Enthusiastic Linux System Administrator

Contact Details: 201901081@daiict.ac.in

Organization Details: Student at DAIICT

Interviewer: 1) Ustav Ladani
Designation: OMMs admin

Interviewer: 2) Mayur Pandar
Designation: OMMs admin

Date: 08/10/2021 **Time:** 17:00

Duration: 75 minutes **Place:**Online(Google meet)

Purpose of Interview:

To identify problems and requirements regarding security in the OMMS system.

1. System should be robust and privacy oriented.
2. It should stay stable despite any failures.
3. System should be able to generate big profit for production company and Administrator
4. System should provide concurrent access to multiple users.
5. System should be secure and easy to use.

2.2.2 Combined Requirements

- For a good user interface, we should have better UI/UX designers or we should have good knowledge about UI/UX designing.
- We should have a better chief information security officer (CISO) to secure our system and serve good privacy to users.
- We will require a good testing team who can test our system before production.
- We should have good algorithm developers to provide optimized suggestions.

2.3 Questionnaires

2.3.1 Google Form

Questionnaire form is given in the next page. We have created a similar google form to analyze this questionnaire. Google form link is given below
https://forms.gle/qUaxtigovBRwgkQg8

OMMs– Online Music management system survey

Please circle your answers to the following questions:

1. Do you like to listen Music?

   Yes / No

2. How frequent do you listen to music?

   Always / Often / Sometimes / Not frequent / never

3. Which mode do you prefer to listen music?

   Online / Offilne

4. Which online platforms do you prefer to listen music? (Multiple choice available)

   YT Music / Spotify / Jio Saavan / Gaana / Other : _____________

5. Have you ever bought premium in any music platform?

   Yes / No

6. Which kind of music do you prefer most? (Multiple choice available)

   Classical / Instrumental / Folk / Rock / Pop / EDM / Bhajan

7. In which languages do you listen the music most?(Multiple choice available)

   English / Hindi / Other : _____________

8. Who is your favourite music artist?

----

9. What is your favourite song?

----

10. How much rating would you like to give to your favourite online music platform?

(Strongly Dissatisfied) **1** **2** **3** **4** **5** (Strongly Satisfied)

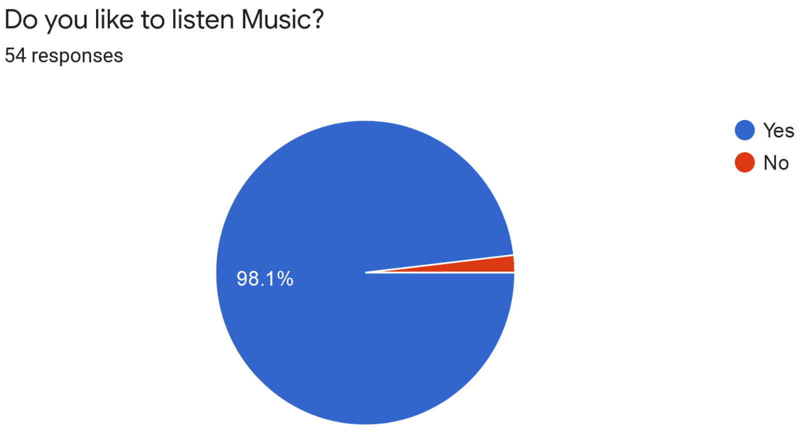
11. Any suggestion about Online Music Management System

Your Name and Email address

Thank you for completing this questionnaire

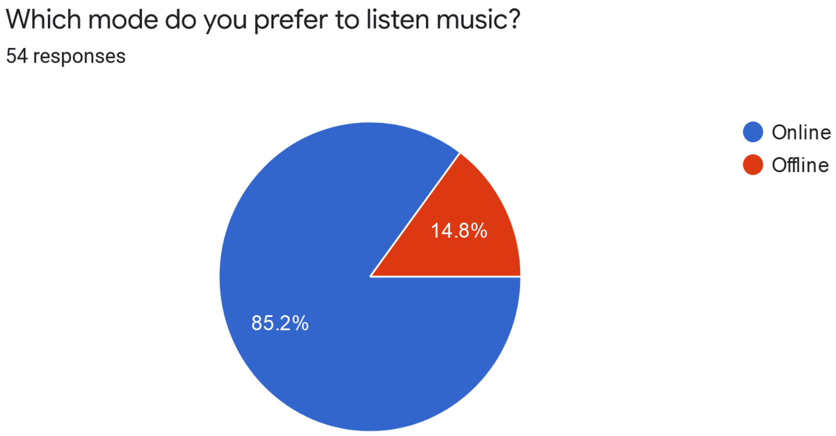
2.3.2 Summary

Summary of responses:



Do you like to listen Music?

54 responses

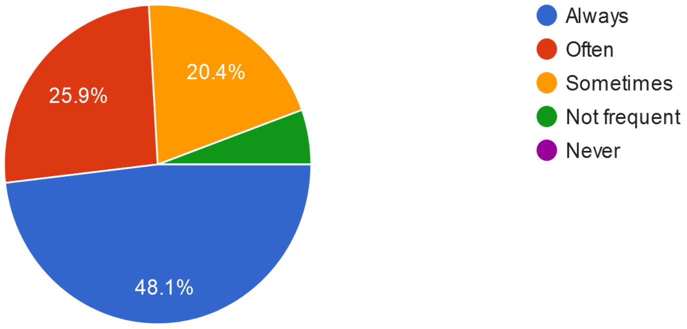


Which mode do you prefer to listen music?

54 responses

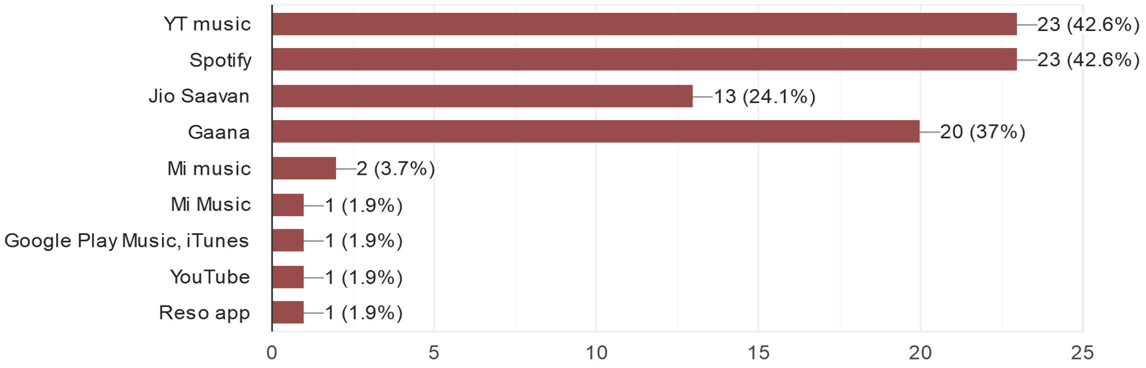
How frequent do you listen to music?

54 responses



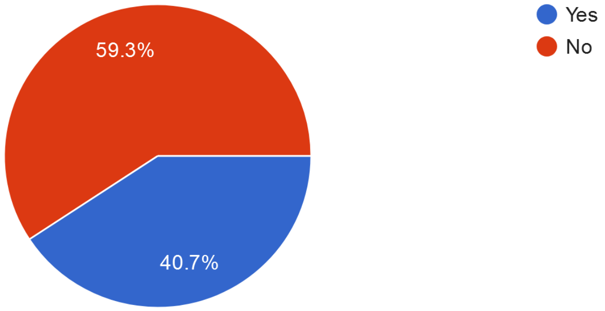
Which online platforms do you prefer to listen music?

54 responses



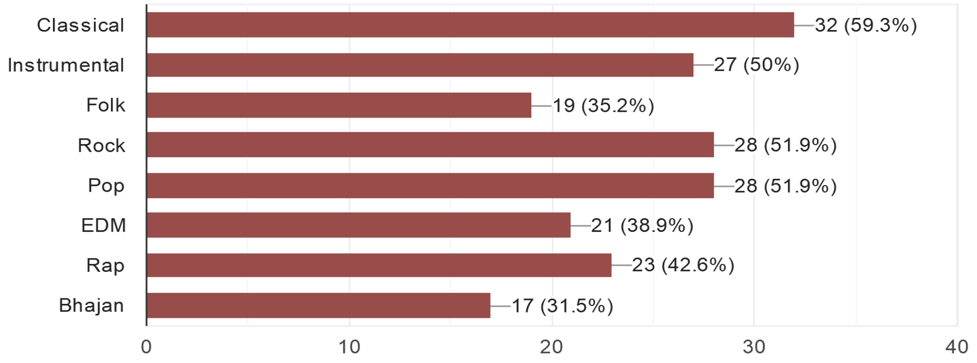
Have you ever bought premium in any music platform?

54 responses



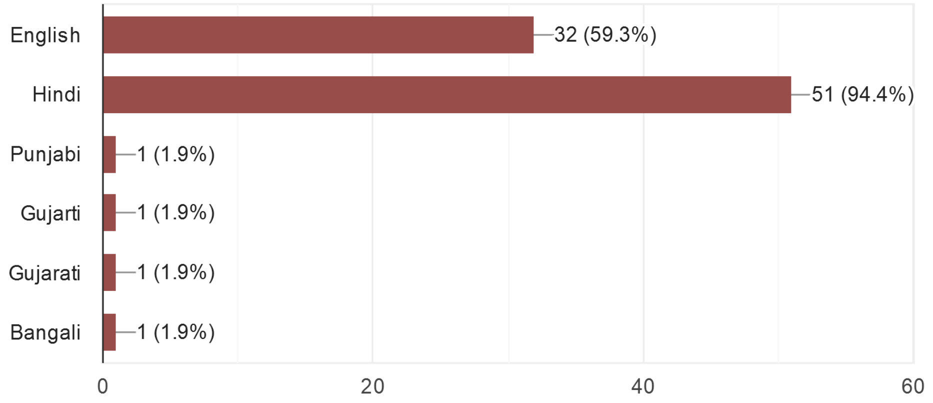
Which kind of music do you prefer most?

54 responses



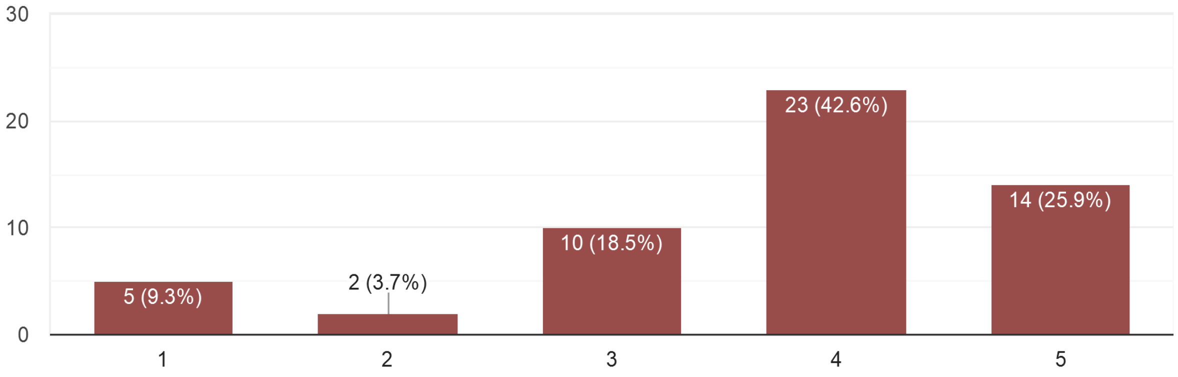
In which languages do you listen the music most?

54 responses



How much rating would you like to give to your favourite online music platform?

54 responses



That’s it, These are the summaries for objective questions. For the short answer type question (8 and 9 eg. who is your favorite artist) different responses had different answers for that. For the Description type question (suggestion about Online Music Management System) common answers are written below.

1. There should be an option to make (and recommend) a playlist containing whole albums, instead of different singers.
2. It Should work more on collecting the music from smaller artists.
3. More than you think people love music You should try to reach them all.
4. There should be less ads in the Online Music Management System.
5. Online Music Management Systems should not have any ads.

2.3.3 Combined Requirements

- We will need better ad providing algorithms which provide related ads to users such that we can make profit and more important users will stay happy with the system by seeing related ads.

- We will need a good song suggesting algorithms such that users will use our service for more time.

- We will require some Networking features to stream music on different devices like android TV etc.

2.4 Observations

2.4.1 Observation Summary

Summary from all the Observations is given in the next page.

OMMs: Observations

System :OMMs(Online Music Management System)

Project Reference :

Observations by: Mayur Pandar,Bhavya Soanki, Utsav Ladani, Dev Joshi (All are OMMs admin).

Date: 09/10/2021 **Time:** 11:00

Duration: 60 minutes **Place:**Online(Google Meet)

Observations:

When we were designing our OMMs(Online Music Management System) and when we got a summary from all interviews and when we had seen all responses from google form we came to the following observations.

1. Many online music systems contain authentication handling.
2. Most systems are mainly divided into three parts. 1. User interface 2.Sever 3. Database.
3. To make money many Online Music Management systems are giving ads to users.
4. Some systems provide a feature to connect systems to other devices.

5. Many systems provide privacy to users and have copyright restrictions to premium songs.

2.4.2 Combined Requirements

Combined requirements are mostly written before in Background reading, Interviews, Questionnaires, Rest of the requirements are given below.

- We will need a Great UI design for the User Interface of our system.
- We need to develop a Good Music player which can work in various type of operating systems.
- We will need good hardwares (Good Storage devices to store data and Good server which optimizes queries) for smoothness of our system.
- A very good and robust copyright management system is needed to stop pirating of premium songs.
- Strong Firewalls or any other security providing devices will be needed to Keep our database secure from attackers.
- We have to find a reliable operating system (RHEL (Red Hat Enterprise Linux), CentOS etc.) which can act as a server for queries in our system.
- We will need a reliable and robust Transaction management system for handling transactions.
- At the end We will need a good audience who use our system and will need good investors who help us to improve our system by investing money in our system.

3 Fact FInding Chart

Objective	**Technique**	**subject(s)**	**Time commitment**
To understand various existing systems.	Background reading	Systems’ workflows, manual pages, Documentations etc.	2 days
Simple meetings to know users’ expectations from the OMMs	Interview	2 User	2 x 1 hours each
Simple meetings to know artists’ expectations from the OMMs	Interview	1 artist	1 hours
To develop business between production company and system administrator	Interview	1 CEO	1 hours
To identify security problems in system and their solutions	Interview	1 Enthusiastic security officer / Linux system administrator	1.5 hours
To collect information about system related questions from the users for the system.	Questionnaires	4 information(data) analyzers	2 days to get data + 2 hours for the summary (total)
For continuation of the development of the system	Observation	2 creative admins	0.5 days
To make the SRS document look clear.	Document formatting	1 member	0.5 hours

4 List Requirements

- To check information about users’ premium accounts we should have a good authentication handler in our OMMs system.

- For Online transactions between system and users, we will need a better transaction management system.

- We should make our system platform Independent. (For example, our system should work on various Operating systems and various devices.)

- We should have better storage options to store songs in our Database of OMMs.

- We will need a strong server for our OMMs which can communicate with Database nicely.

- We should have a better chief information security officer (CISO) to secure our system and serve good privacy to users.

- We should have good algorithm developers to provide optimized suggestions.

- We will need better ad providing algorithms which provide related ads to users such that we can make profit and more important users will stay happy with the system by seeing related ads.

- A very good and robust copyright management system is needed to stop pirating of premium songs.

- Strong Firewalls or any other security providing devices will be needed to Keep our database secure from attackers.

- For a good user interface, we should have better UI/UX designers or we should have good knowledge about UI/UX designing.

- We will require a good testing team who can test our system before production.

- We will require some Networking features to stream music on different devices like android TV etc.

- We have to find a reliable operating system (RHEL (Red Hat Enterprise Linux), CentOS etc.) which can act as a server for queries in our system.

- At the end We will need a good audience who use our system and will need good investors who help us to improve our system by investing money in our system.

5 User Classes and Characteristics

System has 4 user classes:

1. **Listener:** All users are listeners. They can play songs whatever they want.

2. **Artist:** Singer, lyrics writer, music composer, etc. are artists. They are similar to listeners, but the difference is that they have more information fields.

3. **Production Company:** Include all production companies. They can produce and publish the songs.

4. **Admin:** Owner of the OMMs. Admin approves the song, removes users, etc. But not all privileges. More details in the privileges section.

6 Operating Environment

Hardware: Any smartphone, tablet, pc, laptop which support internet connection

System requirements:

- All systems are capable of using the internet using TCP/IP protocol..
- System have to allot at least 10 KB bandwidth to this application

Platforms :

Windows

- Processor: 1 GHz or faster
- RAM: 1 GB or more
- Hard disk: 4GB or more
- Graphics: DirectX 9 or later
- Version: 7, 8, 8.1, 10, 11

Linux

- Processor: 1 GHz or faster
- RAM: 1 GB or more
- Hard disk: 4GB or more
- Graphics: X window server or similar
- Version: Ubuntu 8 or later, Android 6.2 or later

External interface requirements:

Hardware:

- Good quality of speaker

Software:

- Web browser or Web view with support of ECMAscript 5 or later
- Preinstalled .mp3 player in the operating system, if the application is native application, otherwise web browser if enough.

User Interface:

- Web application use JavaScript with react + material UI
- Android Application use Java or Kotlin to make GUI
- Desktop application use Electron framework with react + material UI
- Design : *TBD*

Communication Protocol:

- HTTP to transfer web pages over internet or handel the API request
- FTP to transfer music file over internet

APIs:

- Google,Facebook,Github API for *SignIn/Signup* purpose.
- Google cloud to host server.
- Google AdSense for ads.
- Google Pay, Paytm, etc. APIs for payment.

7 Product Function

OMMs has three major subsystems:

Client Application

Client application is installed on user’s devices. I handle the HTTP request/response, music operations, etc.

Workflow:

- **Register/SignUp/SignIn:** Clients can create a new account using their email id or signup with google, facebook or github account. To create a user, the user sends a request to the server, server verifies information and creates a new user. SignIn can be done using social media accounts or email.

- **Music Player:** Music player uses .mp3 player to play music into the client application. The modern browser provides inbuilt functionality to play music, otherwise we have to add a library according to the platform.

  Music Player functions:
  - Play
  - Pause
  - Next
  - Previous
  - Seek slider

- **Search:** Search the music in the database and send the list of metadata of music to the user. Users select the music and that song plays in the music player. Search user different filters to get a list of specific types of songs. Rating, views, popularity, duration, etc can be used as filters.

- **User Profile:** User profile provides the function to change their details, password, premium, etc.

- **Upload Song:** Song can upload only by production Company

- **Money Operations:** Payment transactions are done using this function with security. Check money information, give money to premium accounts,etc. Handled with Money function

Server Application

Server application is hosted on google cloud or any similar hosting platform. It handles the user request and sends responses, manipulating the data into the database, etc. It works as a controller between client and database.

Workflow:

- **Authentication:** Handle all *Register* and *SignIn* requests. Register new user, login user, etc. authentication related requests are handled by authentication function

- **Search Query:** Handle search request. According to the user's query and filter, the function returns the list of metadata of songs.

- **Music Operations:** All music related operations handled by Music Ops. Insert, delete, update music and it's metadata, etc.

- **User Operation:** User profile update and delete using this function.

- **Payment Transaction:** Payment transactions are done using this function with security.

Database System

Database system store and manipulate the music and client information, make the system secure, manage the user access rules, etc.

Workflow:

- **User:** Insert, update and delete user are done using this function

- **Music:** Insert, update and delete song are done using this function

- **Money:** All transection update done using this function

8 Privileges

1. Listener

   Privileges:
   - Create, update and delete their account
   - Search song with filters
   - Give rating to particular song
   - Switch between free and premium account

   ***Who*:** All user

2. Artist

   ***Privileges*:**
   - Create, update and delete their account
   - Search song with filters
   - Give rating to particular song

- Switch between free and premium account

***Who*:** All singer, lyrics writer, music composer, etc.

3. Production Company

***Privileges*:**
- Update, delete and publish music
- Withdraw money

***Who*:** Production companies who are producing music like T-series, Sony, etc.

4. Admin

***Privileges*:** Usually admin have all privileges, but in our system, admin do not have all privileges.

- Insert, update, delete any user, any song and any detail except money, because money transactions are automated.
- Approve song publish request from production company

***Who*:** Group of few people like system owner, manager and staff of company

All money transactions are automatic, so there is no need for any money related role. Some functions are not included because that operations are automated so no privileges required.

9 Assumption

- All money related transactions are automated and done via external payment gateway (APIs)
- Artist is not publishing music, they make music for production company, only production company publishing music
- Admin has to approve all song before publishing
- User's device must fulfill all hardware and software requirements.

10 Business Constraints

- Production company is paid according to views of music
- System owner earn money by two popular way
  (1) Visual Ads between 2-3 music
  (2) Ads-free premium account
- Almost all functions are automated, so avoid hiring more staff.
  So Staff salary + money pay to Production company + system hosting service change + other < Money earned by Ads

- Song approval is approved as soon as possible
- Any system failure must be resolve very quickly so user experience can not affect
- Use social media marketing technique to reachout to more user