

KORDZ (music streaming)

**Your name: Kenji Dasal**

**Your student number: N00202534**

Software Project

Develop a PHP Music Streaming Website

Year 2 2021-22

DL836 BSc (Hons) in Creative Computing

Table of Contents

[1 Introduction 1](#_Toc96009471)

[2 Business Concept 2](#_Toc96009472)

[2.1 Business Idea 2](#_Toc96009473)

[2.2 Business model 2](#_Toc96009474)

[2.3 Market Research 2](#_Toc96009475)

[2.4 Marketing/Advertising 2](#_Toc96009476)

[2.5 Suppliers 2](#_Toc96009477)

[2.6 Competitors 2](#_Toc96009478)

[2.7 Employees 2](#_Toc96009479)

[2.8 Environmental Impact 2](#_Toc96009480)

[3 Requirements 3](#_Toc96009481)

[3.1 Introduction 3](#_Toc96009482)

[3.2 Requirements gathering 3](#_Toc96009483)

[3.2.1 Similar applications 3](#_Toc96009484)

[3.2.2 Interviews 3](#_Toc96009485)

[3.3 Requirements modelling 3](#_Toc96009486)

[3.3.1 Functional requirements 3](#_Toc96009487)

[3.3.2 Non-functional requirements 3](#_Toc96009488)

[3.3.3 Use Case Diagrams 4](#_Toc96009489)

[3.4 Feasibility 4](#_Toc96009490)

[4 Web application Design 6](#_Toc96009491)

[4.1 Layout 6](#_Toc96009492)

[4.2 Interaction 6](#_Toc96009493)

[4.3 Colour schemes 6](#_Toc96009494)

[4.4 Font choices 6](#_Toc96009495)

[4.5 Wireframes 6](#_Toc96009496)

[5 Database Design 7](#_Toc96009497)

[5.1 Description 7](#_Toc96009498)

[5.2 Business Reporting Requirements 7](#_Toc96009499)

[5.3 Textual Representation of Data-Set 7](#_Toc96009500)

[5.4 Business Rules 8](#_Toc96009501)

[5.5 Entity Relationship Diagram 8](#_Toc96009502)

[5.6 Tables 9](#_Toc96009503)

[5.7 Database Dictionary 10](#_Toc96009504)

[6 System Design/ Architecture Overview 11](#_Toc96009505)

[6.1 Introduction 11](#_Toc96009506)

[6.2 Model View Controller 11](#_Toc96009507)

[6.3 User Authenticaion 11](#_Toc96009508)

[6.4 Routing 11](#_Toc96009509)

[6.5 Templating 11](#_Toc96009510)

[7 Testing 12](#_Toc96009511)

[7.1 Introduction 12](#_Toc96009512)

[7.2 Functional Testing 12](#_Toc96009513)

[7.2.1 Login/Registration 13](#_Toc96009514)

[7.2.2 Navigation 13](#_Toc96009515)

[7.2.3 Calculation 13](#_Toc96009516)

[7.2.4 CRUD 13](#_Toc96009517)

[7.2.5 Discussion of Functional Testing Results 14](#_Toc96009518)

[7.3 User Testing 14](#_Toc96009519)

[7.4 Conclusion 14](#_Toc96009520)

[8 Project Management 15](#_Toc96009521)

[8.1 Introduction 15](#_Toc96009522)

[8.2 Project Phases 15](#_Toc96009523)

[8.2.1 Requirements 15](#_Toc96009524)

[8.2.2 Design 15](#_Toc96009525)

[8.2.3 Implementation 15](#_Toc96009526)

[8.2.4 Testing 15](#_Toc96009527)

[8.3 SCRUM Methodology 15](#_Toc96009528)

[8.4 Project Management Tools 16](#_Toc96009529)

[8.4.1 Github Project 16](#_Toc96009530)

[8.4.2 GitHub 16](#_Toc96009531)

[9 Reflection 17](#_Toc96009532)

[9.1 Your views on the project 17](#_Toc96009533)

[9.2 How could the project could be developed further? 17](#_Toc96009534)

[9.3 Assessment of your learning. 17](#_Toc96009535)

[9.4 Completing a large software development project 17](#_Toc96009536)

[9.5 Technical skills 17](#_Toc96009537)

[9.6 Further competencies and skills 17](#_Toc96009538)

[10 References 18](#_Toc96009539)

# Introduction

The Overall aim of this project to produce a music streaming website where people could play music together, upload and listen to their favorite music with just one click. It is also aim to allow artists to express their creativity and arts along with users who have the opportunity to enjoy their favorite artists.

The application area is going to be a Website and desktop application. However, the project can continue to be a mobile application. The project will be using PHP, My SQL, Bootstrap, CSS, and Javascript. Tools that are going to be used in IDE, phpMyAdmin, Miro and Visual Studio Code. To save and watch over the project we use GitHub for to keep a copy of the file and save the project by pushing and committing and Use GitHub Projects to track the progress of the report and the stage of the project.

The Business Concept of the project is that the streaming website is free to use however theres features that are needed to be purchased a monthly or yearly. For the requirements for this project is the use of CRUD system, design, different pages for each user or admin and a way to import music.

# Business Concept

## Business Idea

Music Streaming App.

This is a music streaming app for remixes, originals and covers. The app is to provide copyright free music to users for streaming or making videos and to provide a platform for new artists. The app will be free for all user however for new artist the fee for advertising will low for the first weekly and monthly prices. For listeners advertisement will only be on the sides on web while it’s a pop-up ad for mobile. New listeners can arrange a playlist for their stream, videos or for their enjoyment. For artist they can upload and share their music while making a profile for their cv.

## Business model

1. **Attracting a large base of users with the free service:**
2. **Convert free users to premium by lower rates of subscription.**
3. **Manage listeners’ rates of usage and attrition**
4. **Balance Cost of free, premium and advertisement**
5. **Finance it all with the revenue from the premium and advertising**

## Market Research

The Market for this product is high having 523.9 million subscribers across the streaming products and with the largest, Spotify having 31% and Apple Music has 15% while Tencent and Amazon having 13% apiece.

The demographic for the streaming service is that 70% of the subscribers is aged to 18 – 34 years old and 49% are of 65 to above in age. The older generation more likely to say that they never had subscribe to streaming service whereas the majority of the 18 – 44 years old having to subscribe to subscribe video-on-demand (SVoD) services.

## Marketing/Advertising

For marketing this can easily be done thanks to social media to share or advertise the service. Another is to listen to feedbacks from users and work with the artists in terms of where they see apps like the product. And lastly spread out to other alternative app stores which increases the chance of the app being downloaded.

## Suppliers

For music streaming one main supplier is needed is a server to save music, accounts, advertisements and more. This is the vital part of the music streaming service to save artist’s music.

## Competitors

The main competitors for this service are Spotify, Apple Music, Tencent and Amazon.

The most competitors’ charges 9.99 for 1 month while amazon charges for 7.99 and apple for 4.99. The reason why Spotify is popular more than the others is that it provides a free version of the product while the others provide a free trial and offer the subscriptions.

## Employees

For employees the work can be divided to many parts of the websites. These teams can be divided by:

* Body of website
* Music Player
* Advertisement
* Subscriptions
* Merchandise shop

## Environmental Impact

This product is a good way of reducing the need for plastics like plastic CDs, cassettes and vinyl. This lowers the amount of carbon footprint produced physically, however this causes the need of storing the songs on massive servers that is needed to be kept cool and be constantly be provide power. Streaming music will lead up to 200-350 million kilograms of emissions in the U.S. alone.

# Requirements

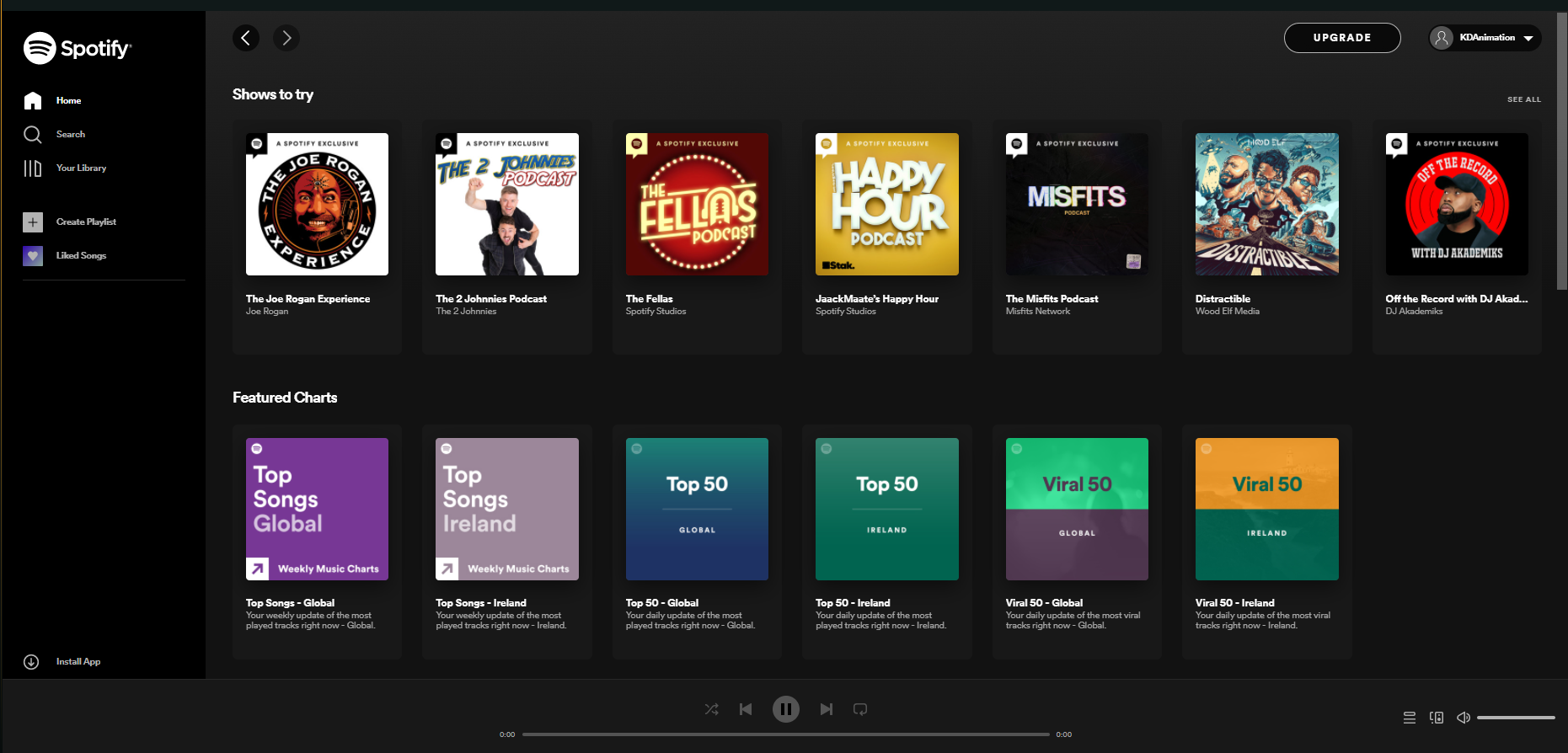
## Introduction

My project area is on the music streaming. I’ll be taking a look at two sites that are currently are being regarded to be two best music streaming sites which are Spotify and Tidal. There are many differences between the two which is going to be mention bellow however the main difference between the two is that the one is free and the other is a monthly subscription.

## Requirements gathering

### Similar applications

Spotify



Spotify is the biggest music podcast and video streaming website. This streaming website provides millions of music and other content from creators all over the world. This is the best free music streaming services.

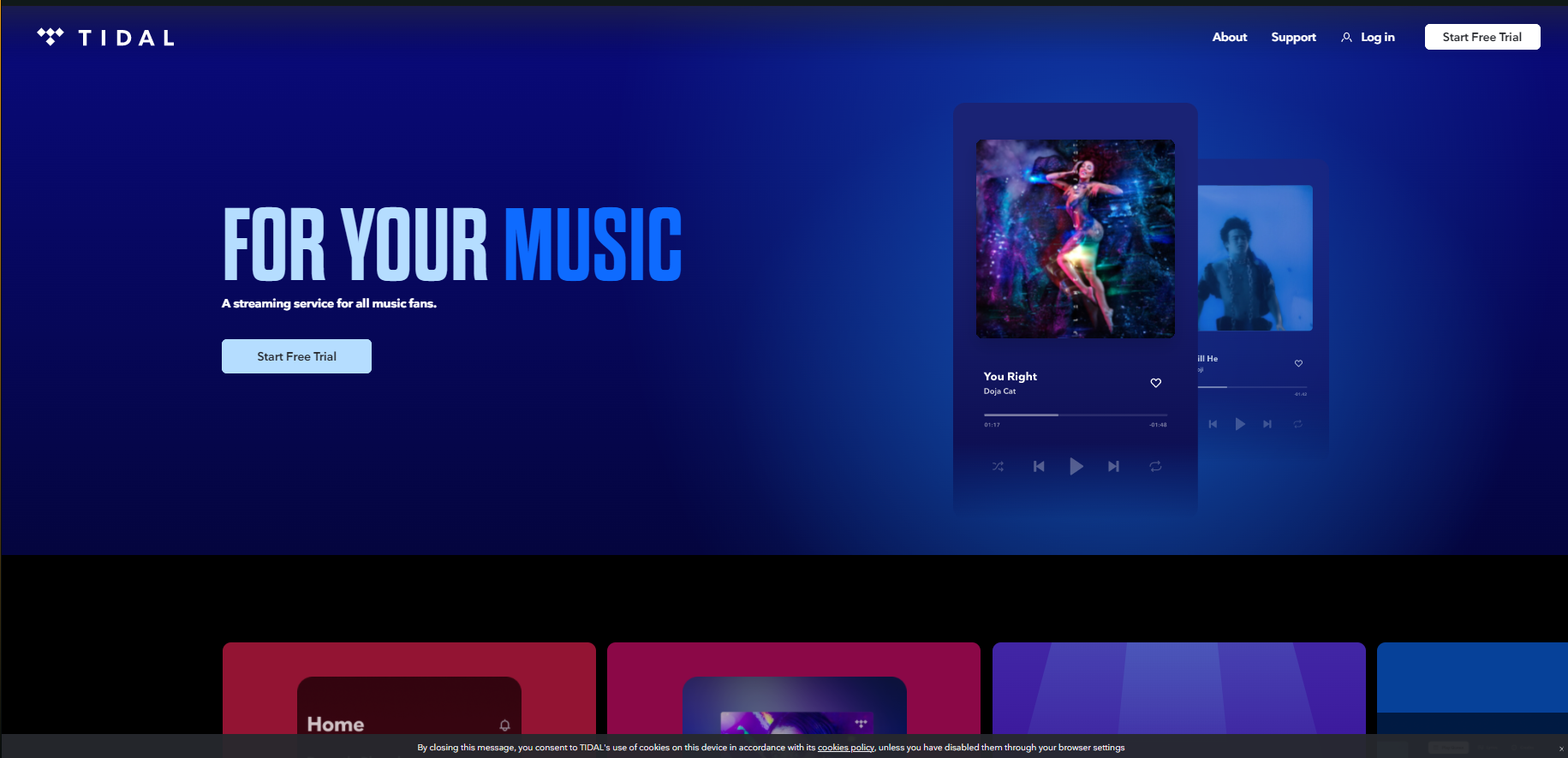
Advantages:

* Free for everyone and can switch to premium for no ads.
* Can be used on web or phone.
* Simple to use.

Disadvantages:

* No longer have lyrics features
* Available to limited countries

Tidal



Tidal is another music streaming service that has access to 80 million songs and 350, 000 videos from music artist all over the world. The service is the top paid music streaming service.

Advantages:

* Good sound quality
* Has a good interface

Disadvantages:

* Tidal is very expensive with 19.99 per month

### Interviews

Questions:

1. What features do you want to see on a music streaming service?
2. What is your view on copyright claiming music?
3. If you’re to make music what do you want from a music streaming service?
4. How often do you use music streaming services?
5. How likely are you to use the music streaming websites?

User1:

1. “I want to allow people to listen with me without sharing my screen”
2. “I make videos and stream and I’m afraid of copyright music and taking down the videos”
3. “for music streaming service I want them to provide a helpful way to share musics”
4. “I use music a lot especially from what I said earlier”
5. “I’m might use it but not as much as spotify”

User2:

1. “If I want to make a music career I want a profile for my music”
2. “I can agree that you should earn from your work however it should only warn the people not to stop how they earn money”
3. “I want them to provide help to new artists in terms of making and signing to a studio or on their own”
4. “I like to listen to music every time I work on projects and interested on making some”
5. “yeah sure it seems easy enough from the description”

## Requirements modelling

### Functional requirements

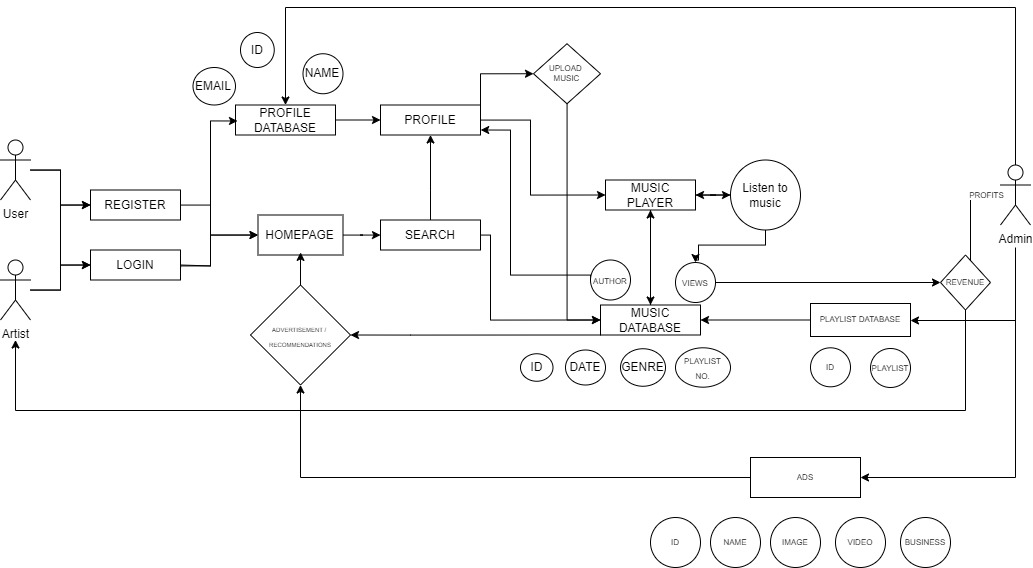
Create a numbered list of what the application should be able to do. Start with the most important feature.

1. Music player
2. Music Playlists
3. Profiles

### Non-functional requirements

* Usability
* It should have a music player on the bottom indicating the music is playing and is able to be interacted or opening up the full player
* Personal Playlist on the right and usable tabs on the left
* All languages are available to be used.
* Performance
* The response time will be quick on making pages as it loads music and the page you are on.
* For Artist they can import music by uploading them in their profiles
* Security
* The access levels will be divided between free, premium, artist and admin
* It should have a minimum of 8 characters for a password
* The service will have no duration however the users can set a duration timer to stop the music from playing.
* The service will pause if another video or music service is playing.

### Use Case Diagrams



## Feasibility

* Servers is the most essential part of the product / service as it is needed to store music and details of users and artist.
* XAMPP can help with the setting up of the website and the database needed for the product.
* A music API can help play music.

# Web application Design

## Layout

The Layout will have a music player on the bottom and navbar on the side sticking on the screen, which will depend on a framework like bootstrap and with this It can be responsive.

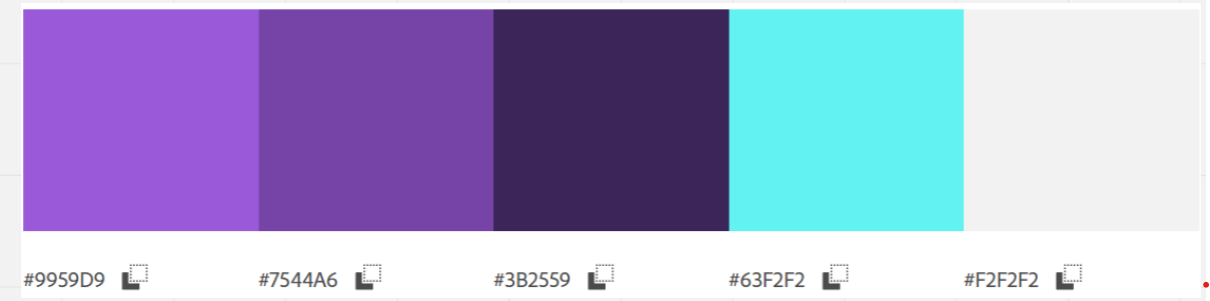
I can use Inertia Framework to produce the music player for the project which will help me insert chose and play music and control the volume and duration. In the project a music visualizer was added however ill use my own p5js music visualizer for the project.

## Interaction

For Navigation Home Button is stuck on the side along with playlist, history, Purchase (Shop). Under this is the list of friends and at the bottom is the login and user’s profile.

## Color schemes

For background users can toggle between light and dark mode changing the background and text only for the side navbar and player it’s a dark shade of purple for the buttons and underline as a gradient of purple to light blue.



## Font choices

For the font I’ll go with Smooch Sans for my font ‘This is the testing for paragraph text’ **‘This is the Bolded titles’,** ‘This is the Headings’ and *‘this is the italicized’.*

## Wireframes

The navbar is on the side when pressing on the user which opens a dropdown to the side it goes and a profile page which shows the carousel playlists, your list of songs and your purchased albums. Clicking on your playlists will open up a full-page of playlists clicking a playlist will open up the playlist and the songs it has. Clicking a song will open the music player and playing the song.



# Database Design

## Description

A Music Streaming website that streams and sell music. It would need a database for all the songs and their artists. For each artist the data should have the songs, description and date released. The database should keep a track on all the songs played by the user. The user will have to input email and password when registering an account.

## Business Reporting Requirements

1. Admins need to be able to create, read, update, and delete: songs, artists, playlists and genres.
2. Users will need to be able to find all songs by their released date.
3. Users may want to find a song by a specific start date.
4. Users need to find all songs using a list of genres.
5. Users need to find the songs using an artist name.
6. Performers may need to find the list of admin contacts.
7. Users need to find stages within a festival by the stage’s location
8. Organizers need to display a list of employees that are assigned to a specific festival

## Textual Representation of Data-Set

## Business Rules

 A **Song** has many **Genre**.

 A **Song** belongs to one **Artist**.

 An **Artist** can have multiple **Songs**.

 An **Image** can be associated with a **User** or **Song**

 A **Song** can has a single **Image**.

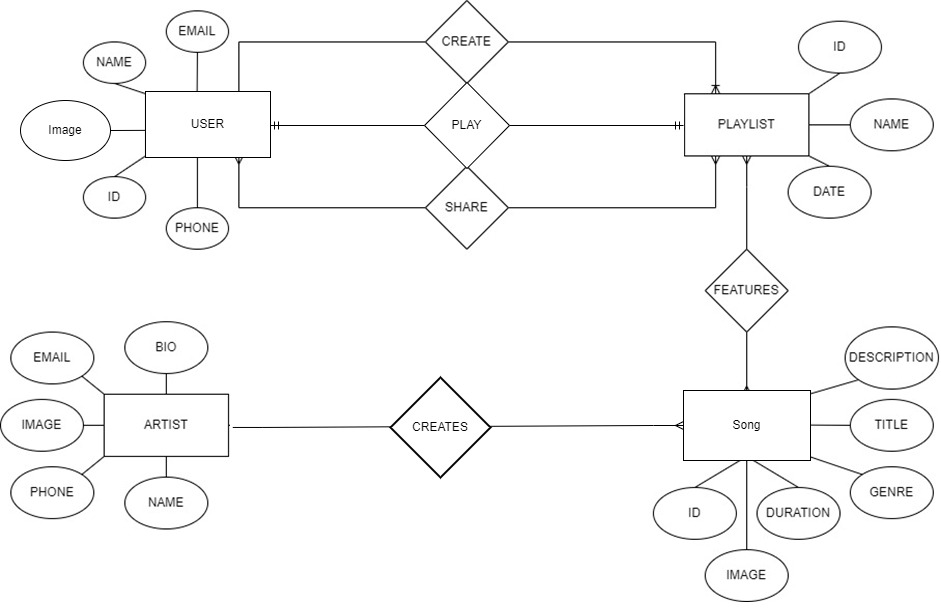
 An **Artist** can have a single **Image**.

 A **Playlist** can have multiple **Songs**.

 A **User** can have many **Playlists**

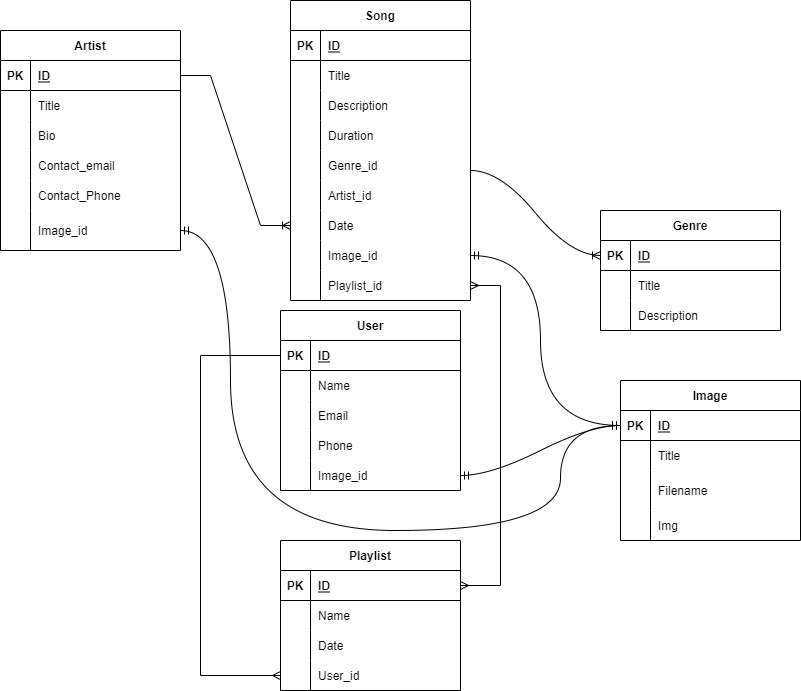
## Entity Relationship Diagram

Substitute in here your ERD from draw.io



## Tables

Substitute in here your tables and the relationships between tables from draw.io in the format you used in DBMS with Mohammed.



## Database Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table | Attribute | Datatype | Range | Required | PK/FK | FK Ref Table |
| Genre | Title,  Description | Varchar | 255 | Yes | ID |  |
| Artist | Title, Bio, Contact\_email, Contact\_phone | VarChar,  VarChar,  VarChar, | 255 | yes | ID, image\_ID | Image |
| Song | Title, Description, Duration,  Date | VarChar,  VarChar,  Time, Date | 255 | yes | ID,  genre\_ID,  artist\_ID,  image\_ID | Genre,  Artist,  Image |
| Playlist | Name,  Date, | VarChar,  Date | 255 | yes | ID,  user\_ID,  song\_ID | Image,  Song |
| User | Name,  Email,  Phone | VarChar | 255 | yes | ID,  Image\_ID | Image |
| Image | Title,  FileName,  Img | VarChar | 255 | yes | ID |  |

# System Design/ Architecture Overview

* 1. Introduction

For Bootstrap it’s a helpful for formulating the website and allow it to be responsive. This can help with margins and spacing of items. This can also help with automatically implementing premade Css to allow me to quicken up the process.

For Inertia it is a Laravel web framework to allow me to input a music player for the project. This includes a music player, playlist, buttons and progress bar. In this framework it also has a music visualizer ill implement this for the full music player page.

* 1. Model View Controller

The Model contains all the data used by the user, like schemas, the databases and their fields and the projects interfaces. And the View contains the UI that the users can interact with this can range from dropdowns and more. This is then accessed by the Controller which contains the business-related logics and handles the incoming requests. The controller will handle all the interactions and inputs of the costumer and the controller will view the data that is requested.

* 1. User Authentication

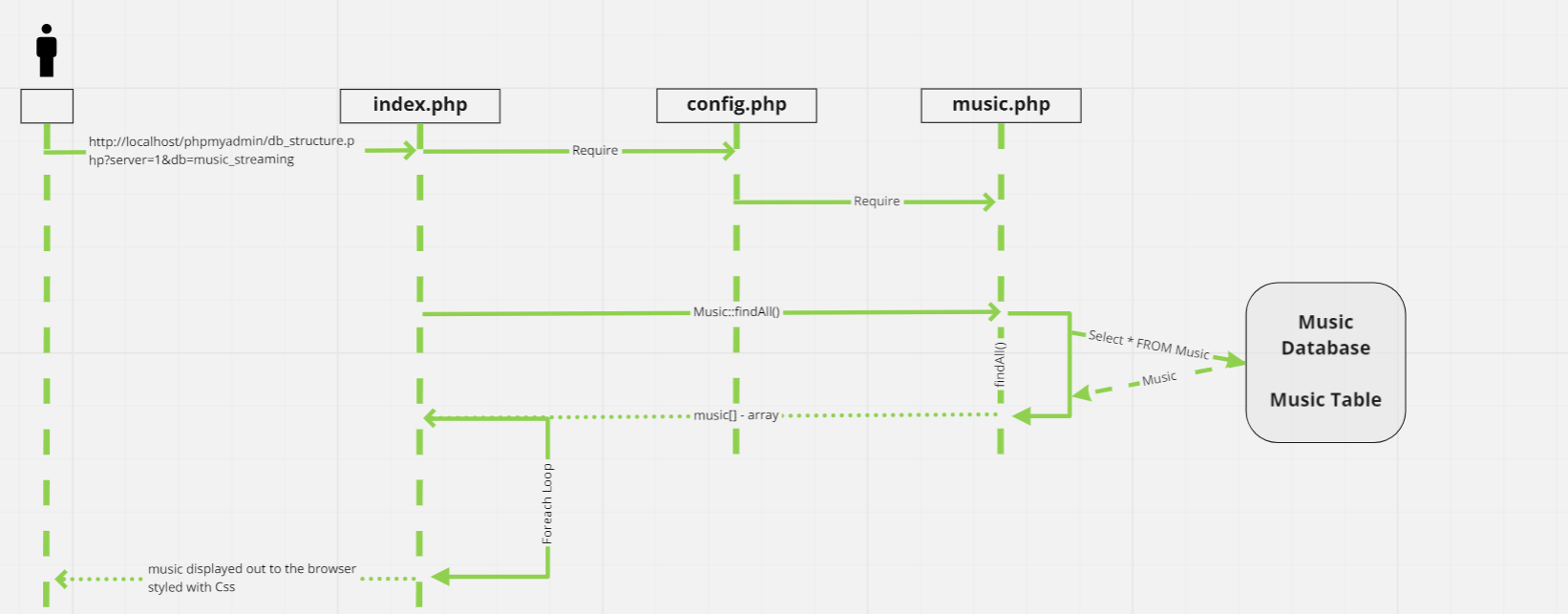
The User Authentication is implemented in the web application framework using the terminal I make an Auth controller to handle the login, register, reset of passwords, verification and more. I also make views for the authentications

* 1. Routing

In order to access the data from the controls we create a routing using web.php file in the routes folder and add in the controls by using use {directory} and calls this directory using Route and the views directory.

* 1. Templating

For Laravel in order to use the website its needed to be made in the name.blade.php file within the resources/view file. In the blade you can access a layout by using @extends however to make some changes in the body @sections is needed.



# Testing

* 1. Introduction

This chapter describes the testing that has been undertaken for the application. This chapter is presented in two sections:

1. Functional Testing
2. User Testing

Functional testing is a type of software testing whereby the system is tested against the functional requirements. The app is tested by looking to see if the actual output for a given input corresponds with the expected output. The tests should be based on the requirements for the app. The results of functional testing can indicate if a piece of software is functional and working, but not if the software is easy to use.

User testing looks to see if a piece of software is easy and intuitive for the user.

* 1. Functional Testing

This section describes the functional tests which were carried out on the app. These functional tests can be categorised as: (whatever is relevant to your app)

Login/Registration

Navigation

Calculation

CRUD

Functional testing generally uses a Black Box Testing technique which means that the internal logic of the system being tested is not of interest to the tester. The tester is only interested in whether the actual output agrees with the expected output.

* + 1. Login/Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Login is needed to allow either admin or user to enter | Login Button | Enters and give roles | Enters and get role based on the database |  |
| 2 | Registration allows user to register an account to use to enter the site | Registration Button | Enters detail and gets given a role | Automatically gives User role to new register |  |
| 3 | Logout allows users and admins to leave their pages. | Logout Button | This can be a drop down | Immediately leaves the admin/user pages |  |

* + 1. Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Side Navigation | Allows to go to other parts of the site and access the login/registration/logout | Allowed register and allow to select admin or user | The Side navbar holds Home, Playlist, History and Shop along with Login and register |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  | Durations and current Time | slider | Slider that can be changed and move based on the current Time | Using the input tag I can have a value dictating the current time and can be changed to a slider through css and js | In Laravel the changing of the value wasn’t working due to a glitch between blade.php and html when using the script. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
|  | Delete | Delete Button | Allow admins and users to delete posts | Deletes Music |  |
|  | Create | Create Button | Allows The user to add | Create and store music to the database |  |
|  | Read | Play Button | Allows the user to play the music | Plays music |  |
|  | Update | Edit Button | Edit allows the user to change the name. |  |  |

### Discussion of Functional Testing Results

Describe the results from the tests. Address any functionality where unexpected behavior could not be debugged.

The first test resulted with a bug with the pathing for the images and music however this was quickly fixed. However, the slider for the music player had a bug which prevent it to change the current time of the music and restart the music back to 0.

* 1. User Testing

User 1:

“For the design I like the idea of gradients but change the colour from purple to pink to black and purple”.

“The website, overall is strait forward. There’s no other things to do other than listen to music add some pages like showing the songs details.”

User 2:

“I like the design of the website but the colours should be lighter and add some shadows to differentiate the nav to the body.”

“You should change the playlist look in the homepage to cards instead.”

* 1. Conclusion

Overall there’s a need to fix the design and try adding more to do in the website. Cards should be implemented for the homepages of the users/admins.

# Project Management

## Introduction

This chapter is being used to manage the project and helps to show and indicate the phases of the project from the Idea to the requirements that are going to be mentioned bellow. I will list out the requirements, the design and the implementations. I will also show the different testing done, implementations of the requirements and how GitHub is used in the project

## Project Phases

### Requirements

### **User Registration & User Authentication** is mandatory to give the user access to the website. However, there will be two types of registration a normal registration and premium registration.

* For **Design** I want to give the website a gradient background. This gives me a use of color theory and gives the website a vibrant feeling. For the interface I went for a constant side navbar and a music player.
* Adding to a **Playlist and Song pages** can be done by using the database. The Users or Admin can add music to a specific playlist and the admin can upload music that can be viewed by the user.
* **Uploading Music** this can be done by using the controllers of the playlist for both user and admin and in the song by the admin.

### Design

### For the Design I choose to give the background a gradient of purple and pink. And for the Interfaces Side Navbar will be implemented and has a homepage, playlist, history and purchase pages. Another is that the It shows the list of friends

### Implementation

### For Implementations a music player, playlist, import, premium page and a store. These are the core implementation for the website.

For the Music Player JavaScript can be used to create the buttons, duration bar and the images. The import can be used by using the CRUD system in the Laravel. For the premium and store can be made by only using the View pages.

### Testing

The first thing to test is the authentication for the website, so far, the authentications are working as intended however there’s no way of making a user into an admin without changing it in either the seeding or in the Database.

Next on to the Music Player when importing the scripts and html to blade and in Laravel is easy due to blade can display html tags however this did take a toll on the scripts. The scripts used for the music player differed from the database dictionary and tables given in the report above, this is due to time constraints and bugs taking time to try and fix.

The main bug I encounter is the slider for the music player both on the small and big player.

During separate testing the sliders were working as intended however, when placed in Laravel and with minor changes mainly on the pathing of images and music, the calculations for the sliders don’t change when tried. Another is during the duration of making the project the scripts have entirely broke and isn’t working as intended and have to be removed.

On testing the CRUD system, it is working however it hasn’t been tested on the new music player. Few pages like shop, history and profile will be removed this is due in part of the time constraint. However, they will be kept on the project to show the idea for the project.

## SCRUM Methodology (OPTIONAL)

Sprints

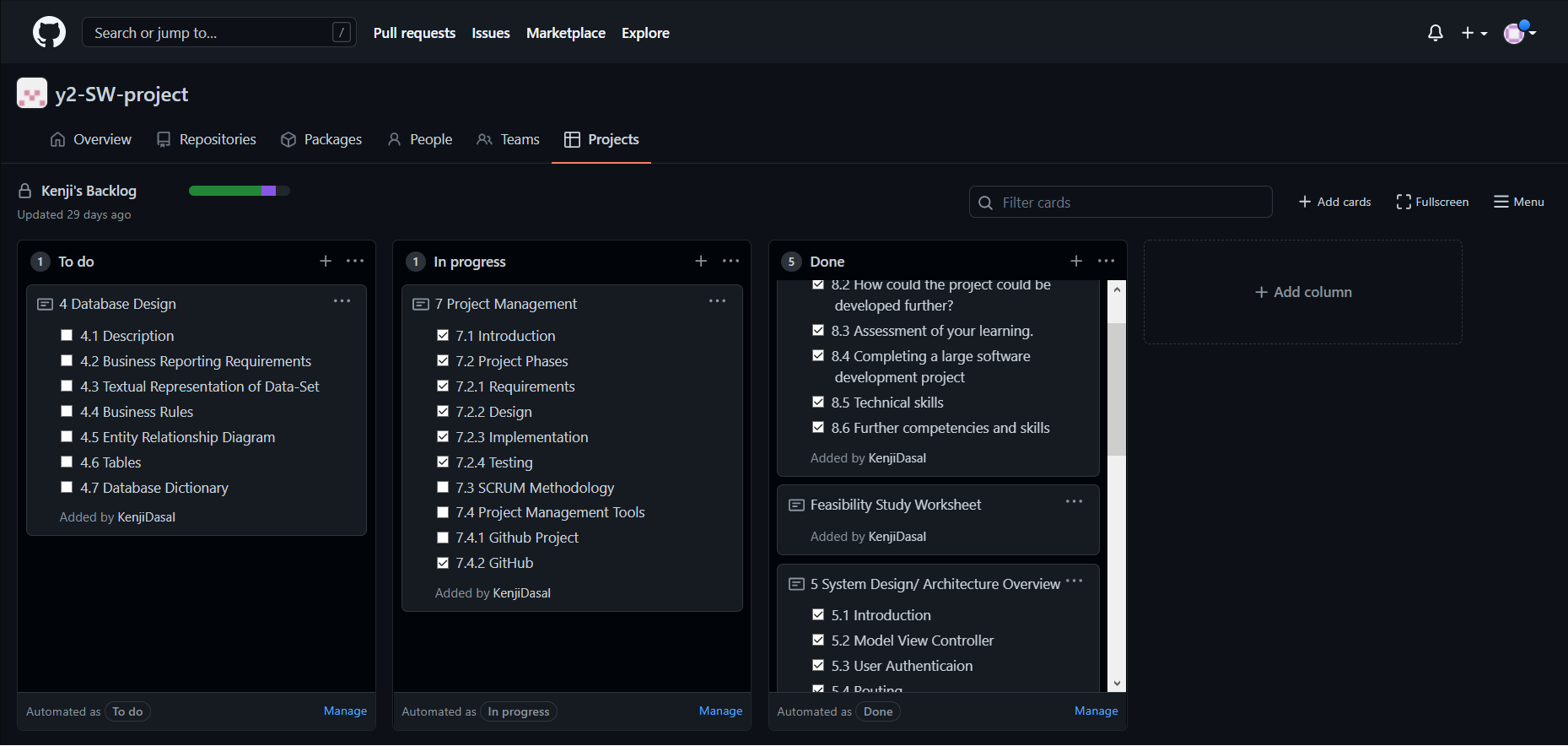
## Project Management Tools

### Github Project

**Description:**

For the Project Design will be added first and the implementations will be next. With the help of GitHub projects this can help monitor the project and see any problems and needing changes.

**Screen shots:**

****

**How it worked in practice:**

By having a guide like the projects, helps give me a blueprint on what to start on and when to add the Implementations or what to do next. We then repeat this process until the project is finished we run into a problem.

### GitHub

**Description:**

For the Project the GitHub is used to upload and can be shared to the other teams which can help with the time and location constraint.

**How it is used**

When downloading the clone of the repository and by pushing any changes to the repository will be saved on the GitHub servers and other teams can download this and change based on their part of the projects

**How it worked in practice**

This can increase the speed of making the project and can be used by others to change and add their work from anywhere and can help when people are in a different location.

# Reflection

## Your views on the project

In my honest the project is needing more time to polish and fix. Many problems popped up during the testing and is still needing to be fixed however this is mainly due to my ambitious characteristic. In spite of these the project was enjoyable to make from start to the current stage of the project.

## How could the project could be developed further?

The project could develop more by adding more time to the pages and the music players.

Another way to develop the project further is to study veu. This is mainly because of the glitch or bug experience when using is files in Laravel.

## Assessment of your learning.

I’m slow to commit and keep up with GitHub projects this is due to having tons of other projects and problems with the JavaScript and the controllers in the Laravel. However, I learnt how to make a Laravel quickly and found a way to design and a blade.

## Completing a large software development project

I learnt that projects like this takes more people in order to make a proper project and I mainly learned more about using controllers and views.

## Technical skills

I’ve learnt to use more of the GitHub and the projects however but maintaining to Push and Commits for the GitHub and checking the boxes in the project this is due to the time spent on fixing problems and bugs.

## Further competencies and skills

I need to space out the time needed for the project. This is due to other CAs given by other classes and the project either taking or losing too much time to make or fix.

# References

[Comapritech](https://www.comparitech.com/tv-streaming/streaming-statistics/#:~:text=IFPI%20found%20that%2038%20percent,taking%20advantage%20of%20audio%20streaming.)

[GSMArena](https://www.gsmarena.com/counterpoint_music_streaming_market_grows_32_in_2019_spotify_still_on_top-news-42452.php)

[Miro](https://miro.com/app/board/uXjVOOUSF20=/?share_link_id=268060981975)

[GitHub](https://github.com/y2-SW-project/sw-project-ca1-KenjiDasal)

[Spotify](https://www.technobezz.com/advantages-and-disadvantages-of-spotify/)

[Tidal](https://custom.ultimateears.com/blogs/custom-made/pros-and-cons-of-the-top-3-music-streaming-platforms)