

Table of Contents

[Introduction 2](#_Toc2090288)

[1.1 Purpose 2](#_Toc2090289)

[1.2 Scope 2](#_Toc2090290)

[1.3 Definitions, Acronyms and abbreviations. 2](#_Toc2090291)

[System Overview 3](#_Toc2090292)

[2.1 System Overview 3](#_Toc2090293)

[Design Considerations 3](#_Toc2090294)

[3.1 Assumptions 3](#_Toc2090295)

[3.2 Development Methods 5](#_Toc2090296)

[3.3 Architectural Strategy 5](#_Toc2090297)

[4. Use Case 6](#_Toc2090298)

[5. Music Player Flow 10](#_Toc2090299)

[6. Data Design 11](#_Toc2090300)

[7. Gui Design (Mock up) 12](#_Toc2090301)

# Introduction

## 1.1 Purpose

The purpose of this design specification is to cover all technical aspects for the creation of a music player concept for Rebmem Engineering.

## 1.2 Scope

This Design Specification is to be used by the software engineering/Quality Assurance team as the definition of the design and requirements to implemented in the creation of the “Music player project”.

**Project Team**

|  |  |  |
| --- | --- | --- |
| Name | Email | Phone |
| Lewis Holland Clarke | [Lewis.clarke@Looydsbanking.com](mailto:Lewis.clarke@Looydsbanking.com) | 07741 933 123 |

## 1.3 Definitions, Acronyms and abbreviations.

|  |  |
| --- | --- |
| Acronyms & Abbreviations | Definition |
| SDS | Software |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheet |
| JS | JavaScript |
| GUI | Graphical User Interface |
| The Client | Rebmem Engineering |
| The User | The Person(s) using the system |

Introduction

System Overview

# System Overview

## 2.1 System Overview

This project is to provide the client with a new music player that they can implement in their new line of interactive devices.

Currently the client is a large supplier of electronic goods to 3rd parties and are looking to expand the current catalogue of devices.

The device the music player will be created for is a portable storage system.

This music player must have the ability to control music from the device’s storage as well as a GUI for the user to navigate the controls of the music player.

Design Considerations

# Design Considerations

The section describes many of the issues which need to be addressed or resolved before attempting a complete design solution.

## 3.1 Assumptions

**Related software or Hardware**

We are looking at creating the GUI using HTML/CSS and JavaScript to allow us to create a GUI that is both attractive and functional for the user. These technologies will allow use and the client to further develop this project in the future both for online and offline solutions.

We will look to use the local storage on the device that it is to be installed on (Local DB,SQLITE)

**General Constraints**

1) **Hardware or software Environment.**

As yet the client has not provided the software environment or platform that is going to be integrated with the hardware.

For this reason, we have decided to use the above software languages and implement the PhoneGap platform to push or solution out to multiple platforms at once.

This can then be built upon during phase two of this project.

The application is to be hosted internally on each portable device.

2) **End user environment.**

The user should have access to all music controls.

The user should be able to control the music playback.

The user should have the ability to shuffle the current song list/ play list

The user should have the ability to create a new playlist.

The user should have the ability to search for audio files with a database.

The user should be able to see a list that can display either the song title or album.

If the user is idle for more than 30 seconds the device should switch to idle mode to save power.

3) **Data repository and distribution requirements**

Data will be store in a database on the local device.

4) **Interface/ Protocol Requirements**

Network connectivity and TCP/IP support required.

5) **Performance requirements**

No internal failures are acceptable. The only known and accepted failures are from the network or Internet connection that is providing the communication between the two machines.

Speed is key for the client the front end needs to be lightweight and load quickly on multiple devices.

6) **Verification and validation requirements (testing)**

For testing purposes one machine (Dev environment) will be required to perform unit tests.

One machine will be required for UAT as well as one mobile device to test code implementation.

## 3.2 Development Methods

For this project we will be implementing an agile methodology and looking to break this project down into 2 separate week-long sprints.

The project has a dead line of 22nd February.

Sprint 1 – Planning, Analysis, Design

Sprint 2 – implementation, Testing, Maintenance and CI

## 3.3 Architectural Strategy

The front-end system (GUI) will be compiled using HTML/CSS and JavaScript.

The reason for this choice is due to the time frame in which to deploy the MVP to the client is very short.

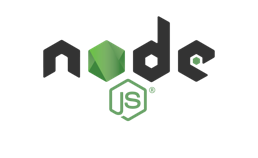
Once sign off is complete and the client has completed testing on the hardware being used the code could be refactor to react native for use on android OS devices, this is an assumption that the client will opt for this OS to use on the device as it has been the most popular open source platform used on similar devices in this market sector.

The back end for this project will run and express server using node and include a SQLite database.

The reason to use SQLite is that the data base can be store within the device itself taking up very minimal storage space.







For the deployment of this project we are looking at using the cloud bases system called Heroku to set up a full route to live.

The application we have a production location at URL: <https://music-player-epa.herokuapp.com/>

## 4. Use Case

|  | System use Case Description |
| --- | --- |
| Use Case Reference & Name | Music\_player\_001 Access System |
| Primary Actor | user |
| Goal | To access the Music Player |
| Brief Description | The purpose of this use case is to access the Music Player |
| Assumptions | The User has internet connection |
| Secondary Actors | None |
| Trigger | The use case starts when the actor selects the uniform resource locator (URL) link to the Music Player. |
| Pre-conditions | The Music Player system needs to be available.  The URL needs to be available. |
| Post Conditions (Main Flow) | The user has accessed the Music Player |
| Alternate Flow | None |
| Includes | None |
| Includes By | None |
| Extends | None |

**Main Flow**

* 1. The use case starts when the actor selects the uniform resource locator (URL) link to the Music Player System.
  2. The system displays the home page for the Music player.

* 1. USE CASE ENDS.

|  | System use Case Description |
| --- | --- |
| Use Case Reference & Name | Music\_player\_002 Player Controls |
| Primary Actor | user |
| Goal | To be able to control music play back |
| Brief Description | The purpose of this use case is to control music playback |
| Assumptions | Music is loaded into controls |
| Secondary Actors | None |
| Trigger | The use case starts when the actor selects the play button. |
| Pre-conditions | The Music Player system needs to be available.  Data should be loaded into the system |
| Post Conditions (Main Flow) | The user can perform all controls from available buttons |
| Alternate Flow | None |
| Includes | None |
| Includes By | None |
| Extends | None |

**Main Flow**

1. The use case starts when the actor selects the play button.
2. The actor is able to control the playing audio with the controls on screen

1. USE CASE ENDS.

|  | System use Case Description |
| --- | --- |
| Use Case Reference & Name | Music\_player\_003 Search |
| Primary Actor | user |
| Goal | To search for artist or track |
| Brief Description | The purpose of this use case is to perform a search against the music database. |
| Assumptions | The User has internet connection |
| Secondary Actors | None |
| Trigger | The use case starts when the actor enters text into the search bar and hits the search button |
| Pre-conditions | The Music Player system needs to be available.  The user needs to be located at the search bar |
| Post Conditions (Main Flow) | The user has accessed the search bar |
| Alternate Flow | None |
| Includes | None |
| Includes By | None |
| Extends | None |

**Main Flow**

1. The use case starts when the actor enters text into the search bar and hits the search button
2. The actor is presented with the results of the search

1. USE CASE ENDS.

**Alternative Flows**

1. The search the user had entered is invalid
2. There is no data against the searched input from the user

|  | System use Case Description |
| --- | --- |
| Use Case Reference & Name | Music\_player\_004 Shuffle |
| Primary Actor | user |
| Goal | To be able to shuffle the order of a played playlist |
| Brief Description | The purpose of this use case is to trigger the shuffle function in the music controls |
| Assumptions | The User has selected a playlist to play |
| Secondary Actors | None |
| Trigger | The use case starts when the actor clicks and activates the shuffle function |
| Pre-conditions | The Music Player system needs to be available.  The shuffle button is available from the music controls |
| Post Conditions (Main Flow) | The user has access to the shuffle button |
| Alternate Flow | None |
| Includes | None |
| Includes By | None |
| Extends | None |

**Main Flow**

1. The use case starts when the actor clicks and activates the shuffle function
2. The playlist playing should play in a random order

1. USE CASE ENDS.

|  | System use Case Description |
| --- | --- |
| Use Case Reference & Name | Music\_player\_005 Create Playlist |
| Primary Actor | user |
| Goal | To be able to create a new playlist |
| Brief Description | The purpose of this use case is to create a new playlist |
| Assumptions | The User has selected the create playlist button |
| Secondary Actors | None |
| Trigger | The use case starts when the actor clicks the create playlist button |
| Pre-conditions | The Music Player system needs to be available.  The create playlist button is available to the actor |
| Post Conditions (Main Flow) | The user has access to the create playlist button |
| Alternate Flow | None |
| Includes | None |
| Includes By | None |
| Extends | None |

**Main Flow**

1. The use case starts when the actor clicks the create playlist button
2. The actor is asked to input a name for the new play list
3. The new playlist is Pushed to the database

1. USE CASE ENDS.

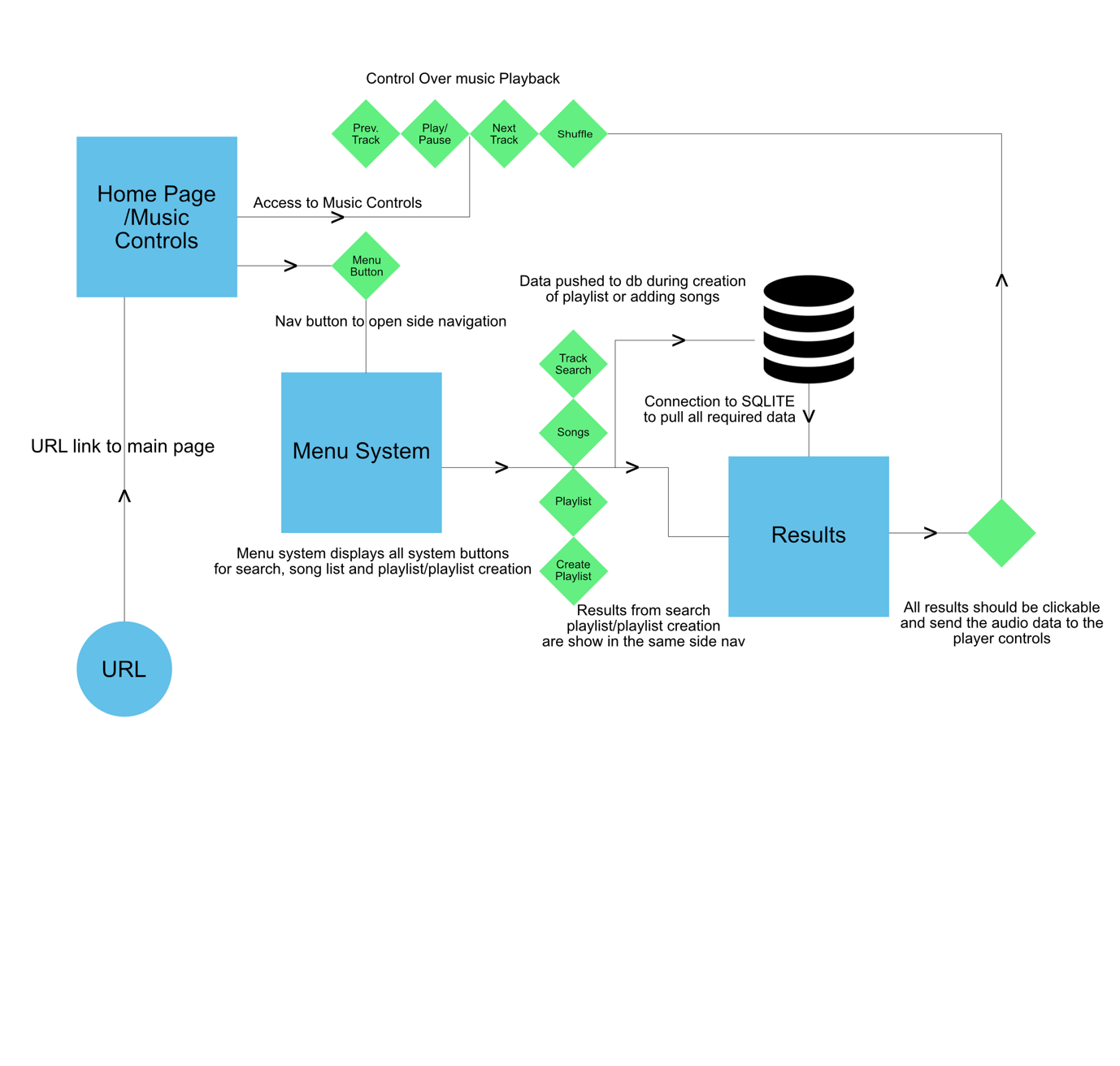
|  | System use Case Description |
| --- | --- |
| Use Case Reference & Name | Music\_player\_006 Add songs To Playlist |
| Primary Actor | user |
| Goal | To be able to add songs to playlist |
| Brief Description | The purpose of this use case is to add songs to an existing playlist |
| Assumptions | The User has selected a song to add |
| Secondary Actors | None |
| Trigger | The use case starts when the actor clicks on a song to add to the playlist |
| Pre-conditions | The Music Player system needs to be available.  The create playlist button is available to the actor |
| Post Conditions (Main Flow) | The user has access to the create playlist button |
| Alternate Flow | None |
| Includes | None |
| Includes By | None |
| Extends | None |

**Main Flow**

1. The use case starts when the actor clicks on a song to add to the playlist
2. The song is added to the current play list.
3. The new song is Pushed to the database

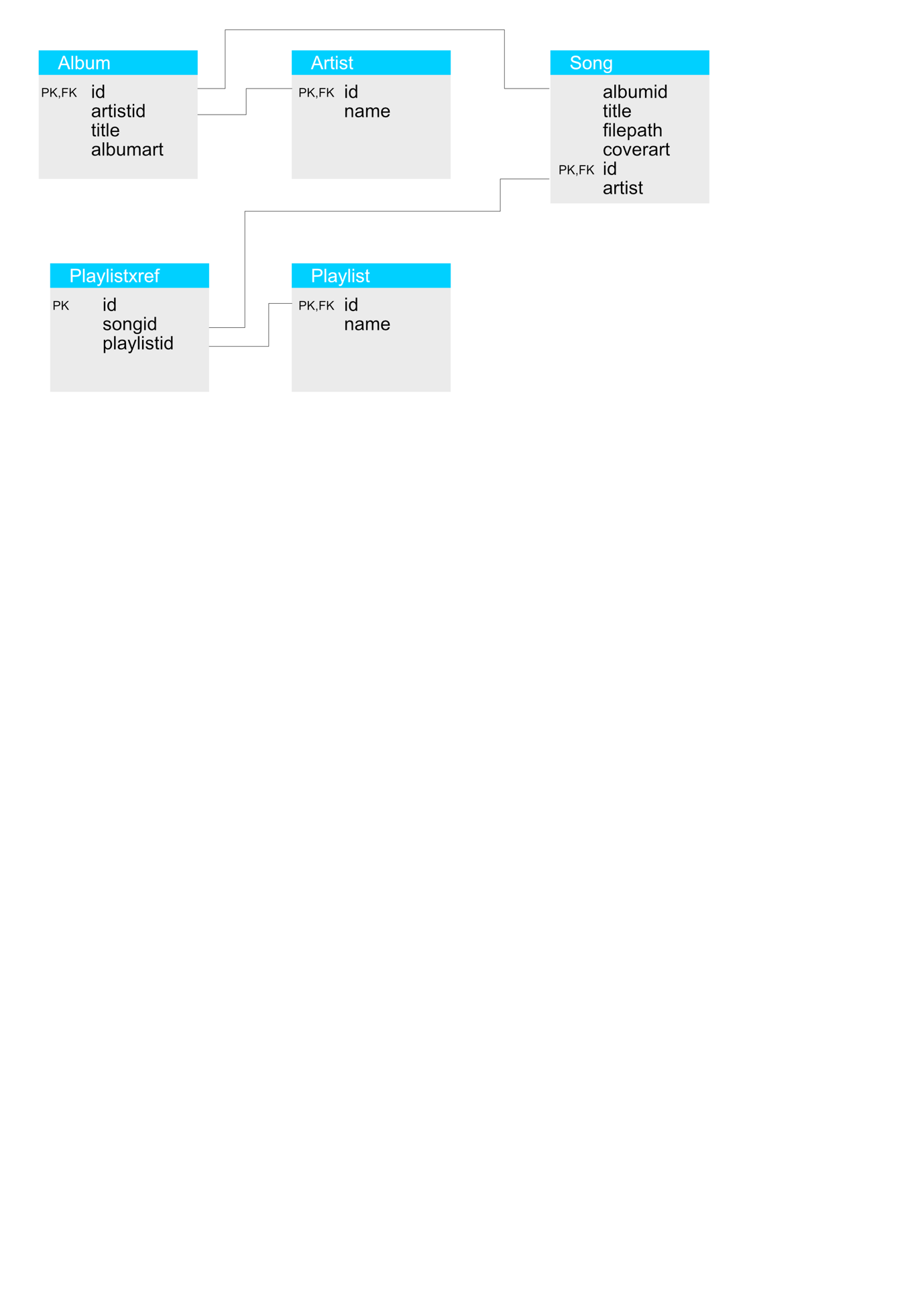
1. USE CASE ENDS.

## Music Player Flow



## Data Design

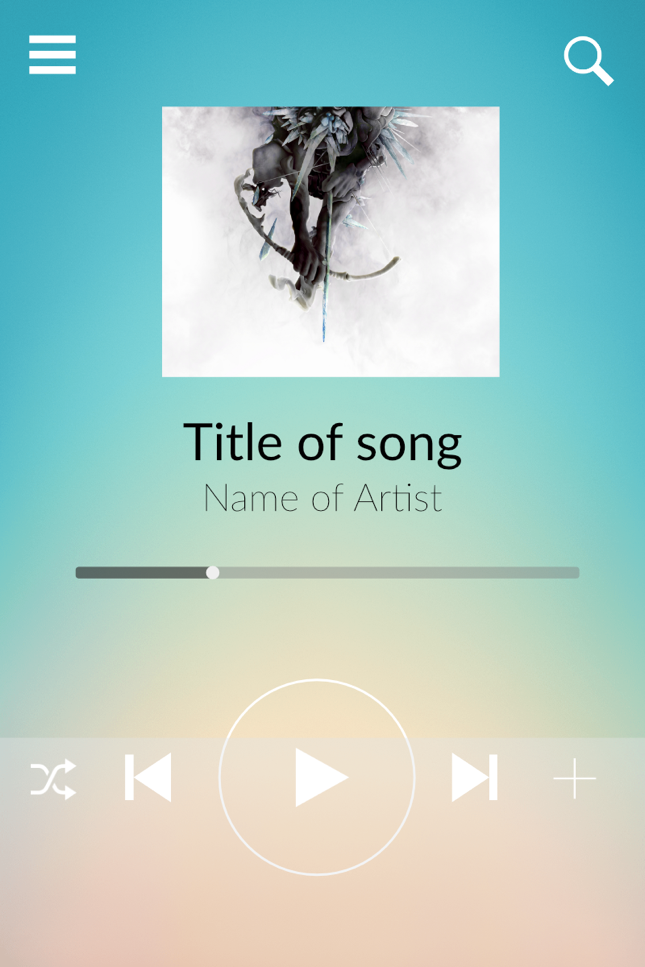
The SQLITE schema is going to be made up of 5 tables as you can see below, the idea would be to create two views from one that will cross reference all the songs, artist and albums together to allow the user to perform searches against those metrics and also a view of the playlist, to allow the user to view all the playlist and associated songs within.



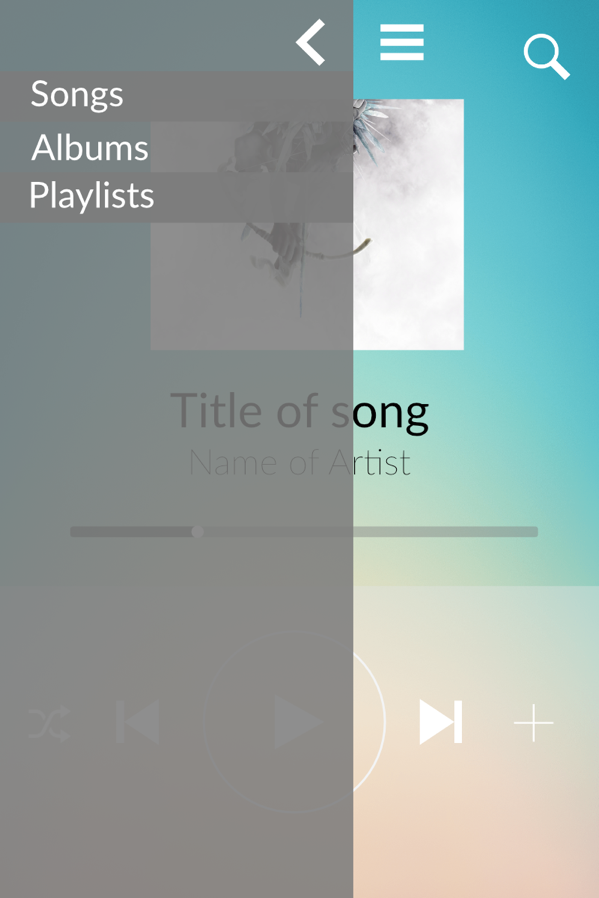
## Gui Design (Mock up)

Please not these are early designs and can changes thought-out the production process.

Music Player Front screen.



Music Player Menu Screen



Music Player Search Screen

