APRIL 22, 2021

# DATABASE MANAGEMENT SYSTEM

PROJECT TOPIC: MUSIC LIBRARY MANAGEMENT SYSTEM

**Branch:** Computer Science

SECTION: E1

**Faculty:** Mr.T Senthilkumar

**Submitted by:**

**SHUBHANKAR GARG(RA1811003010262)**

# ABSTRACT

This document presents the procedure and issues of the database project entitled 'Music Library Management System'.

Its aim is to formally describe the phases of the design and development. These phases are categorized into 3 main steps: Database, Application and Graphic User Interface. The Music Library Management System is a music library organization and storage registry.

The platform is a SQL database with python support. Application and the GUI are developed in Python using PyQt5 and Qt Framework. The result is a database which enables the storage, organization and metadata retrieval of Music Files to manage the information of artists, albums and songs and playlists.

# INTRODUCTION

The aim of this project is to develop a sample centralized relational Music application. This application has to store information of various songs along with its artist, genre, album etc. We will be looking closely at well established music databases such as Spotify, MusicBrainz and Apple Music and take note of how they store and retrieve files. The goal is to create a database that stores all the metadata of songs to help in identifying, storing, tagging, playlist management and even as back-end api of a media player.

In this context the functionality is to update, remove and insert records for the different entities. Additionally, we would like to try to include some form of creating a playlist if possible.

Scope of this project include:

- Music Tagging
- Song metadata retrieval
- Searching a song as per its album, genre, artist or release year

- Creating or viewing playlists
- Music classification
- Music library organization

## FUNCTIONALITIES SUPPORTED

- Finding songs by title, artist, duration, format, and genre
- Finding albums by title, artist, and year of release
- Finding artists by name and displaying all of their songs or albums
- Insert new Albums, Artists and Songs to the database
- View and Create Playlists

The team’s minimum target is to show these core functionalities in a user friendly Graphic User Interface (GUI). This consideration is taken into the architecture of the database, which tries to ensure a built on architecture. Additional functionality has to be integrated in an easy way.

## SOFTWARE USED

### TOOLS & IDE USED:

- Python 3.9
- PyCharm IDE
- PyQt5 Framework
- Qt Designer

### DATABASE USED:

- MySQL

## ASSUMPTIONS

- Each Song File has an songID, artist, album, genre, duration, and file format
- Each Artist has many songs, many albums, and many genre
- Each Album has many songs, an artist, and one genre
- A Playlist has a name and is composed of many songs

## ENTITIES USED

In this domain, we will be modeling the following entities:

- Song/File

  A song file is an entry for a unique digital audio data released in public domain.
  It is represented by an ID

- Album

  An album is a compilation of songs, released by an artist/ band. It has many songs in it, and those songs are referred to as an album.

- Artist

  An artist is generally a musician, a group of musicians, or another music professional (composer, engineer, illustrator, producer, etc.)

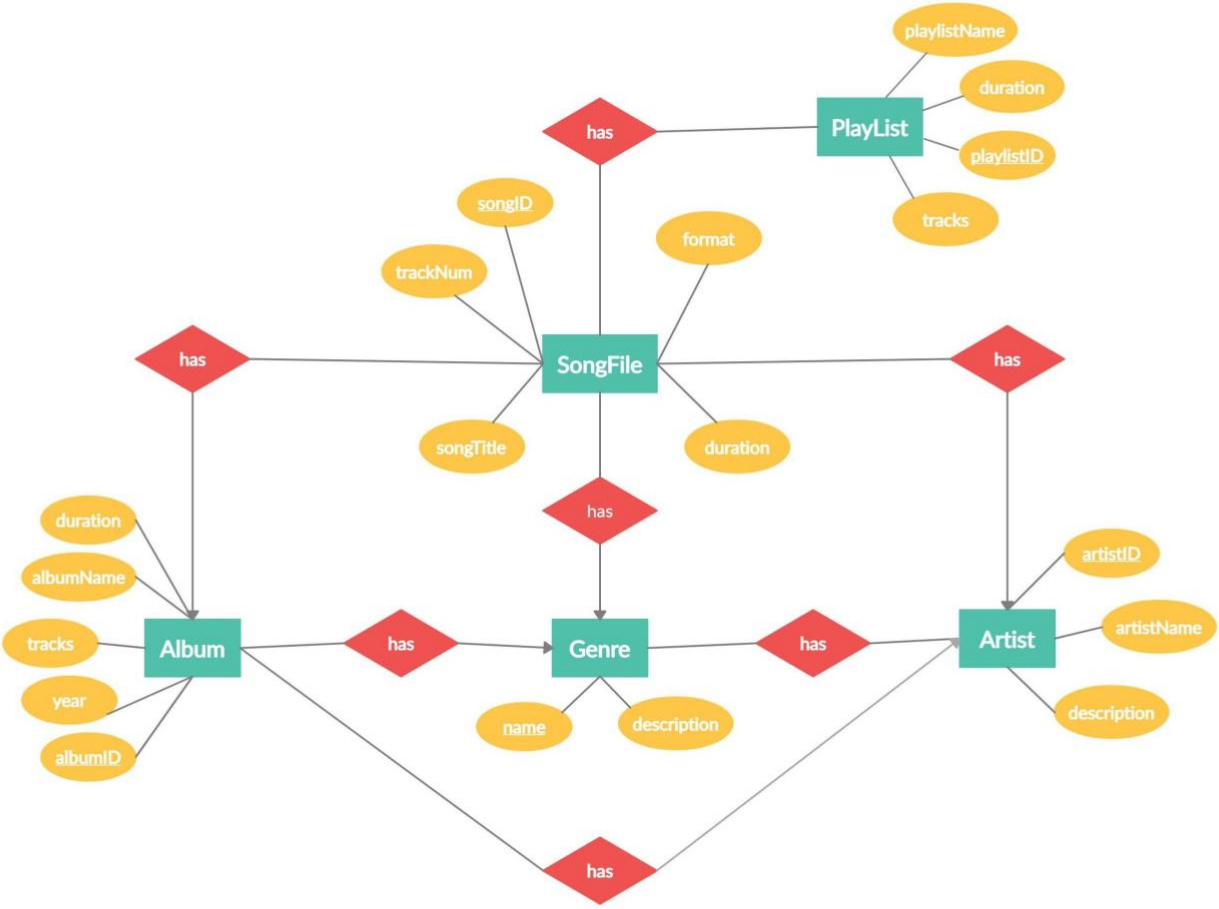
- Genre

A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions.

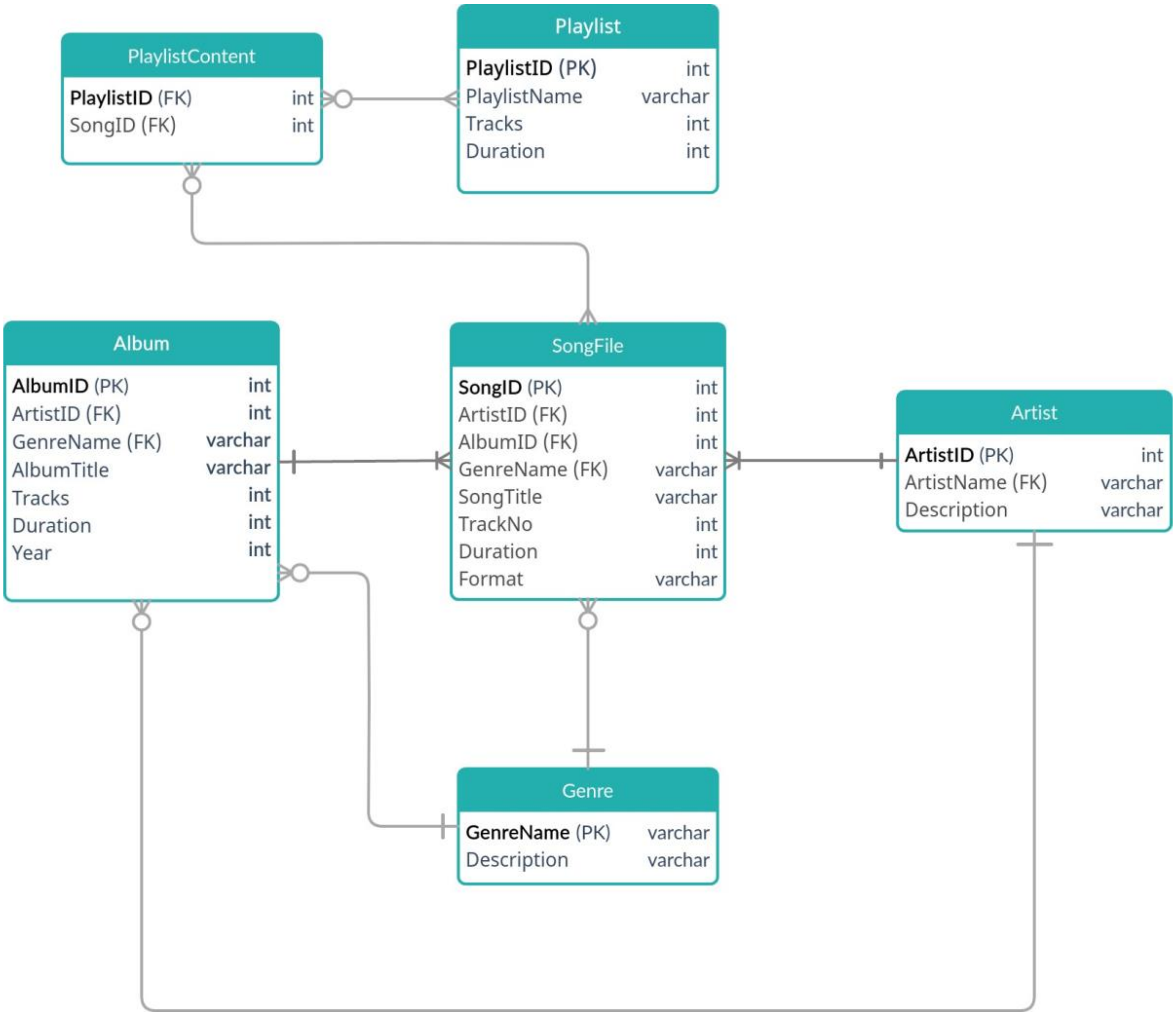
- Playlist

A music playlist is a list of audio files that can be played back on a media player either sequentially or in a shuffled order.

## ER DIAGRAM



# SCHEMA



SongFile(<u>songID</u>, artistID, albumID, genreName. songTitle , trackNo, duration, format)

Artist(<u>artistID</u>, artistName, description)

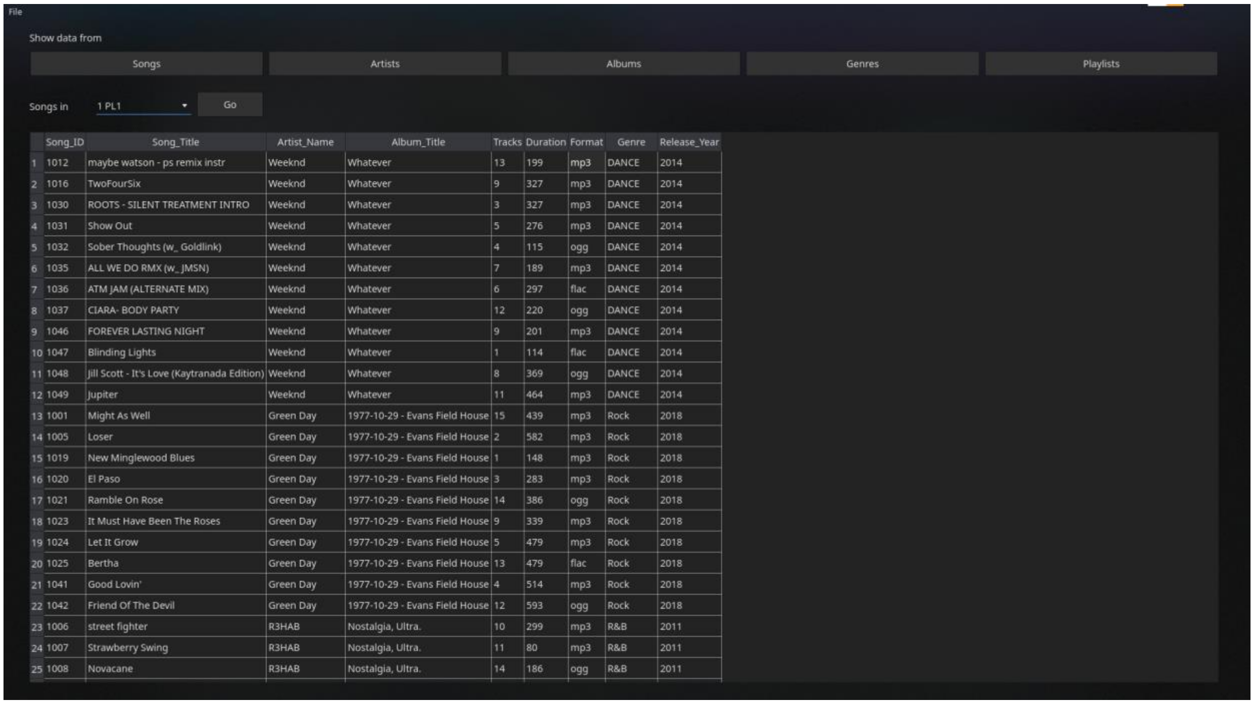
Album(albumID, albumTitle, artistID, tracks, duration, year, genreName)
Genre(genreName, description)

Playlist(playlistID, playlistName, tracks, duration)

PlaylistContent(playlistID, songID)

## GUI WITH SCREENSHOTS

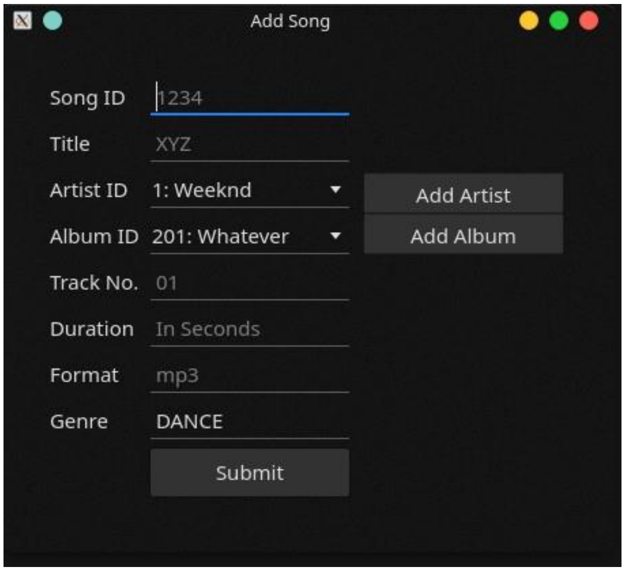
The GUI is developed using Qt Framework so as to enable cross-platform usage and compatibility. Also, this helps in minimizing errors and enabling future improvements to the system while having backwards compatibility with previous versions.



| | Song_ID | Song_Title | Artist_Name | Album_Title | Tracks | Duration | Format | Genre | Release_Year |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1012 | maybe watson - ps remix instr | Weeknd | Whatever | 13 | 199 | mp3 | DANCE | 2014 |
| 2 | 1016 | TwoFourSix | Weeknd | Whatever | 9 | 327 | mp3 | DANCE | 2014 |
| 3 | 1030 | ROOTS - SILENT TREATMENT INTRO | Weeknd | Whatever | 3 | 327 | mp3 | DANCE | 2014 |
| 4 | 1031 | Show Out | Weeknd | Whatever | 5 | 276 | mp3 | DANCE | 2014 |
| 5 | 1032 | Sober Thoughts (w_ Goldlink) | Weeknd | Whatever | 4 | 115 | ogg | DANCE | 2014 |
| 6 | 1035 | ALL WE DO RMX (w_ JMSN) | Weeknd | Whatever | 7 | 189 | mp3 | DANCE | 2014 |
| 7 | 1036 | ATM JAM (ALTERNATE MIX) | Weeknd | Whatever | 6 | 297 | flac | DANCE | 2014 |
| 8 | 1037 | CIARA- BODY PARTY | Weeknd | Whatever | 12 | 220 | ogg | DANCE | 2014 |
| 9 | 1046 | FOREVER LASTING NIGHT | Weeknd | Whatever | 9 | 201 | mp3 | DANCE | 2014 |
| 10 | 1047 | Blinding Lights | Weeknd | Whatever | 1 | 114 | flac | DANCE | 2014 |
| 11 | 1048 | Jill Scott - It's Love (Kaytranada Edition) | Weeknd | Whatever | 8 | 369 | ogg | DANCE | 2014 |
| 12 | 1049 | Jupiter | Weeknd | Whatever | 11 | 464 | mp3 | DANCE | 2014 |
| 13 | 1001 | Might As Well | Green Day | 1977-10-29 - Evans Field House | 15 | 439 | mp3 | Rock | 2018 |
| 14 | 1005 | Loser | Green Day | 1977-10-29 - Evans Field House | 2 | 582 | mp3 | Rock | 2018 |
| 15 | 1019 | New Minglewood Blues | Green Day | 1977-10-29 - Evans Field House | 1 | 148 | mp3 | Rock | 2018 |
| 16 | 1020 | El Paso | Green Day | 1977-10-29 - Evans Field House | 3 | 283 | mp3 | Rock | 2018 |
| 17 | 1021 | Ramble On Rose | Green Day | 1977-10-29 - Evans Field House | 14 | 386 | ogg | Rock | 2018 |
| 18 | 1023 | It Must Have Been The Roses | Green Day | 1977-10-29 - Evans Field House | 9 | 339 | mp3 | Rock | 2018 |
| 19 | 1024 | Let It Grow | Green Day | 1977-10-29 - Evans Field House | 5 | 479 | mp3 | Rock | 2018 |
| 20 | 1025 | Bertha | Green Day | 1977-10-29 - Evans Field House | 13 | 479 | flac | Rock | 2018 |
| 21 | 1041 | Good Lovin' | Green Day | 1977-10-29 - Evans Field House | 4 | 514 | mp3 | Rock | 2018 |
| 22 | 1042 | Friend Of The Devil | Green Day | 1977-10-29 - Evans Field House | 12 | 593 | ogg | Rock | 2018 |
| 23 | 1006 | street fighter | R3HAB | Nostalgia, Ultra. | 10 | 299 | mp3 | R&B | 2011 |
| 24 | 1007 | Strawberry Swing | R3HAB | Nostalgia, Ultra. | 11 | 80 | mp3 | R&B | 2011 |
| 25 | 1008 | Novacane | R3HAB | Nostalgia, Ultra. | 14 | 186 | ogg | R&B | 2011 |

Main Window

The main window has a table view and buttons for user friendly viewing of data stored in various tables in the database. All buttons have their keyboard shortcuts and the table has scrolling & auto adjust features to view big amounts of data if needed.



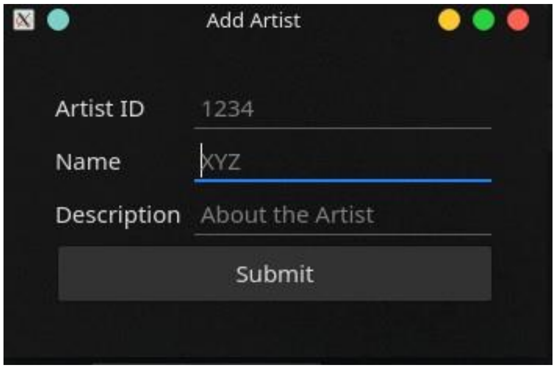
Song ID: 1234
Title: XYZ
Artist ID: 1: Weeknd
Album ID: 201: Whatever
Track No.: 01
Duration: In Seconds
Format: mp3
Genre: DANCE

Add song dialog

The Add dialogs can be accessed through File ⟶ Add ⟶ Song / Artist / Album.

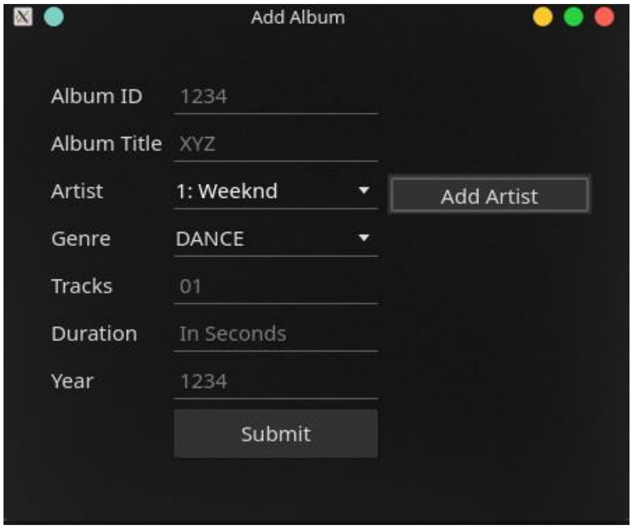
All fields have regex validation to validate input, decreasing user errors and enabling input sanitization. The comboboxes auto update according to the values in other comboboxes. Eg. Only Albums by the selected Artist will be available.

Option to Add Artist/Album are also available on Song form for easy access.



Artist ID: 1234
Name: XYZ
Description: About the Artist

Add Artist dialog



Album ID: 1234
Album Title: XYZ
Artist: 1: Weeknd
Genre: DANCE
Tracks: 01
Duration: In Seconds
Year: 1234

Add Album dialog

# CONCLUSION

As stated in the introduction the minimum target at the beginning of this project was to show the core functionalities in a user friendly GUI. This section will describe where the planning was realistic and will also give recommendations for similar or further projects. The core functionalities were almost all reached. Although a robust GUI part for playlist management could not be implemented without risking the integrity of the database or making a more complex method of table selection to accomplish the same.

The additional features that were reached are creating the overviews of the entities, and error checking. This helps in keeping the integrity of data and also sanitizing the user input to increase the security of the system. Also, some additional features like auto-update were added to bring a better user experience.

The GUI passed well 2 test scenarios. The team created an easy to use application with self-guiding windows. During the process the team was faced with challenges in the problem of data update. If data is changed the view window must be closed and opened again to see this change. Through research and discussion this problem was solved in implementing a

auto-refresh possibility in the view windows.

It is advised to similar projects to consider the problems stated in this conclusion. Such projects should plan from the start how to surpass the limitations of their programming language. This implies a better knowledge of programming language itself as well as all the APIs used. A well planned teamwork is crucial for facing problems during the process.

# REFERENCES/BIBLIOGRAPHY

https://musicbrainz.org/doc/MusicBrainz_Database/Schema

https://dev.mysql.com/doc/refman/8.0/en/create-table-foreign-keys.html

Beginning PyQt: A Hands-on Approach to GUI Programming (2020) Book by : Joshua Willman

https://javascriptgorilla.wordpress.com/2016/08/23/spotify-database-schema/

https://doc.qt.io/qt-5/qtsql-index.html

https://doc.qt.io/archives/qtforpython-5.12/PySide2/QtSql/QSqlDatabase.html

https://doc.bccnsoft.com/docs/PyQt5/designer.html

https://realpython.com/python-pyqt-database/#using-sql-databases-in-pyqt-best-practices

https://doc.qt.io/archives/3.3/designer-manual-14.html

https://www.learnpyqt.com/tutorials/qtableview-modelviews-numpy-pandas/

https://dev.mysql.com/doc/refman/8.0/en/join.html