OOPs Project Documentation:

Music Recording Company

A top-down report of the project completed as part of the S3 CSE 204 (OOPS) Course

# Team 6: Adithya Nair AM.EN.U4CSE19103

# Akshay Harikumar AM.EN.U4CSE19104

# Alekh Avinash AM.EN.U4CSE19105

# Sreejith Kumara Pai AM.EN.U4CSE19153

# 

# 

# Submission: 11/ 12/ 2020

# **Introduction**

The project was a demonstration of the link between the Object-Oriented Paradigm, Databases, and GUI basics. Through the process, we were able to learn the importance of time scheduling, goal setting, the importance of good leadership, and most of all planning ahead.

## Question:

A **Music Recording Company** wants to design a database in order to maintain and organize the information appropriately that can facilitate the storage and retrieval of data efficiently. There are many musicians whose details like Id, name, and address are to be maintained. Each album recorded has an id, title, copyright date, and format. Each song has a title and an author. Each album has a number of songs and no song can appear in more than one album. Each song is performed by one or more musicians and a musician can perform a number of songs. Design a database system for the music company to maintain all the data conveniently and efficiently.

## Objectives

The main objective of the project is to set up a system for a Music Recording Company to maintain a record of the artists and albums in circulation. This facilitates the Artists and the Administrators of the company a reliable system to add/remove, create reports, and monitor the changes in trends and profits.

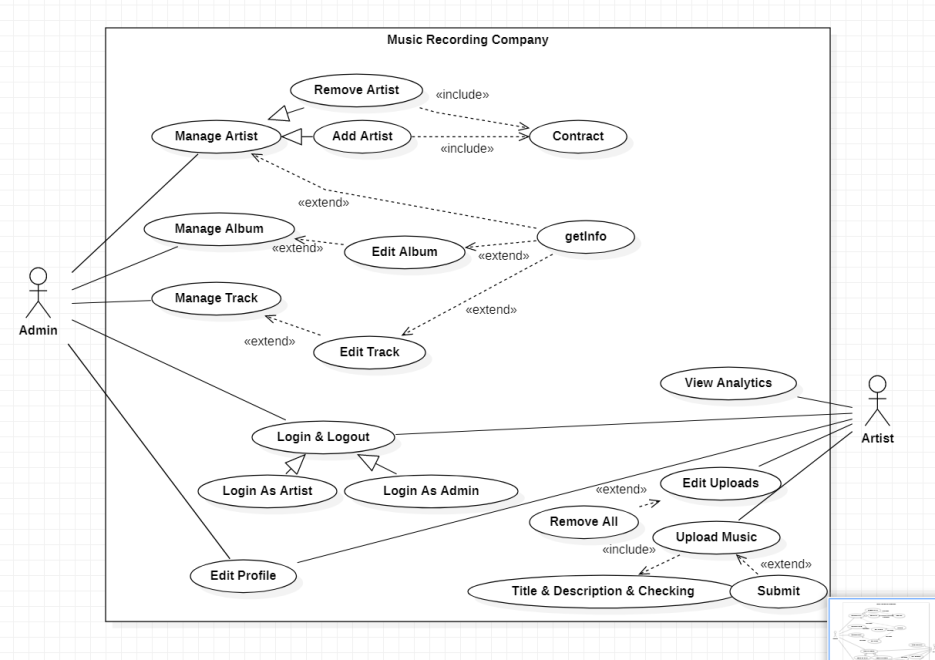
This requires building an ideal database for storage, a reliable GUI, and a system of functions able to fulfill most requirements. The following are the steps that we followed as we worked towards the project:-

# Steps:

* Requirement Analysis
* Class Diagrams
* Implementation and Designing GUI

## Requirement Analysis

Most of the planning and basic properties of the app were set at this stage. Most possible interactions of the Admin or the Artist entities with the databases were considered and scrutinized. A final basic set of requirements and functions that would meet such requirements were set by the end of this stage. Also, a simple Use-Case Diagram was set as follows:

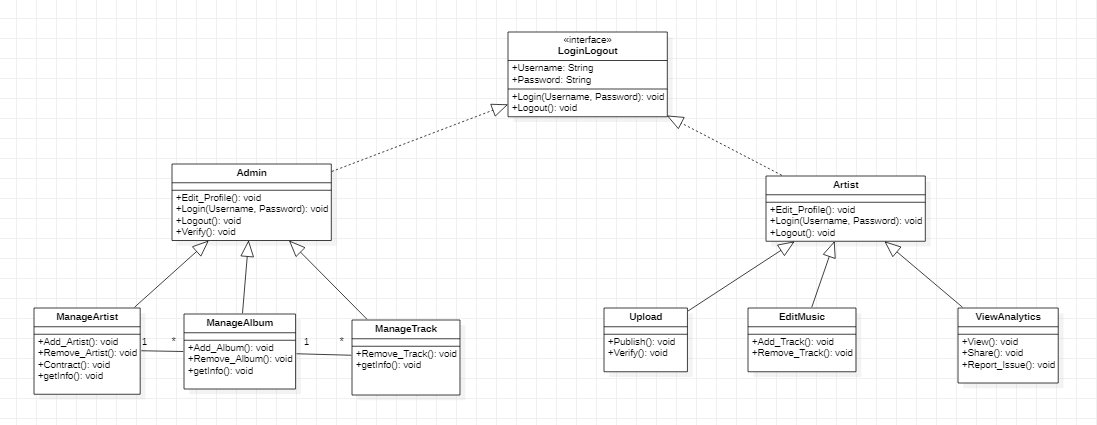


## 

## Class Diagram

At this stage, we worked on the different classes and their relationships. Our main points of discussion were to maintain the flexibility and reliability of the system without compromising any of the requirements put forth in the initial stage. Here we learned about the importance of planning. If there weren’t a pre-determined requirement analysis, it would have been harder for us to design a system for the users.

The system required a basic login logout system. Also, the system calls for a separate entity relationship with the database for the Admin and the Artist. Therefore we ended this stage with a class diagram like so:-



## Implementation and Designing GUI

This stage is where we began the programming and function designing aspect of our project.

Initially set up a database for the application as instructed for our DBMS project. To connect the database to the java project we used the **JDBC** module. The system was mostly connected to the database by inheriting the connect object to the required classes.

Next, we implemented the basic functions of the program, like:-

* Login/Logout Process
* Adding/Removing artists, albums, and songs from the database,
* Viewing required details from the database.
* Editing information of respective entity identities etc.

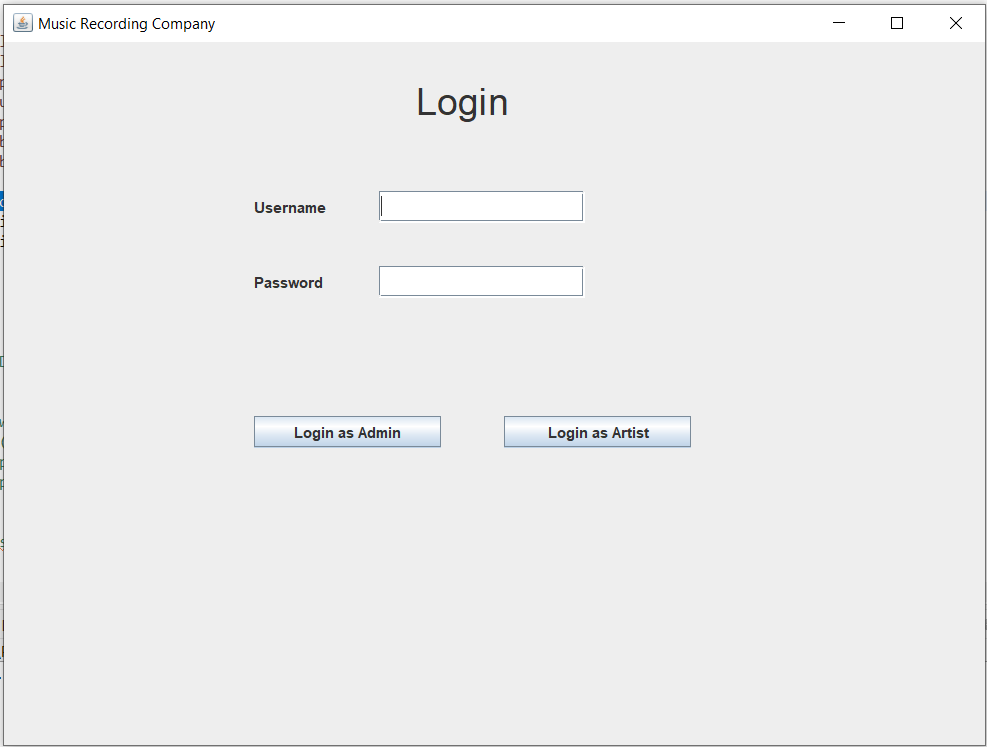
Here we used many techniques for OOPs like Inheritance, encapsulation of certain processes, and abstractions.

Next, we started with the implementation of the GUI. Here we made the above-mentioned processes easy to understand and reliable to work on. This was done by implementing the **Swing** module.

Finally, the project completion was completed by testing the processes by their respective authors and scrutinizing the functions and processes of the application. The use of a class-based abstraction allowed easy editing and updating of the program.

# Final App Processes

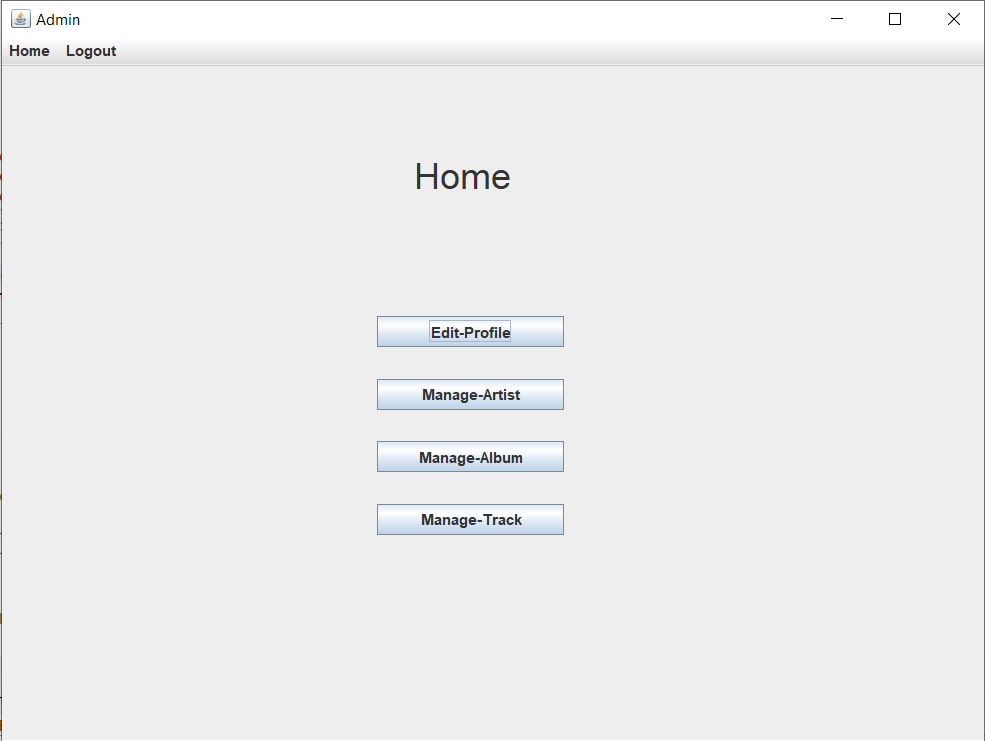
## Login Page



The Login/Logout process is split into two parts, However, the user can enter their credentials and log in as an **Administrator** or an **Artist** without knowing the internal mechanisms. The logged-in user will be able to exit the application by logging out using the option in the Menu Bar and closing the app. Also, the user can use the Menu Bar for navigation.

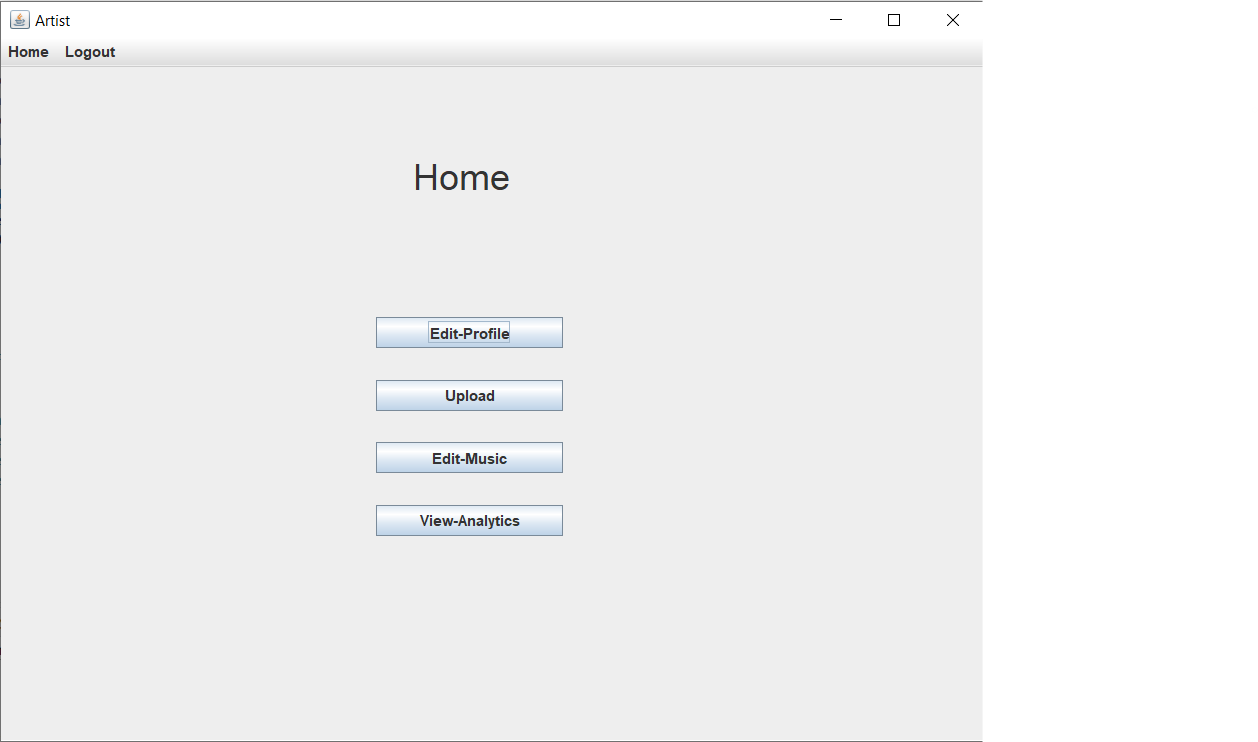
.

## Admin Page



The Admin page contains the main operations for the use of the Admin. **Edit Profile** allows the Artist to edit their own credentials in the Database. The **Manage Artist**, **Manage Album**, and **Manage Track** options allow the Admin to view details of the respective options and update the information of the same.

## Artist Page



The Artist Page allows the artist to control and keep track of their music. The **Edit Profile** option allows the Artist to edit their own credentials. The **Upload** option allows the Artist to publish their tracks. **Edit Music** allows the Artist to update/remove their albums and tracks. **View Analytics** helps the artist to keep track of his/her songs.

# Conclusion

The project involved a lot of cooperation and discussions. This in turn required good participation of all members involved. There were a lot of problems we encountered during the development process. This allowed us to get a taste of the real world app designing process.

**Thank You.**