# Music Box Service System

## "Documentation"

# Abstract

This system supports two main roles**: Service Administrators** and **Clients**. **Service Administrators** have full control over room management, including booking and validity checks, and comprehensive Update, Delete operations for songs, albums, artists library and also staff & language management. They can also access automated sales reports for efficient service oversight. **Clients**, on the other hand, are provided with a delightful and intuitive experience. They can effortlessly select their preferred songs from collection of melodies and enjoy seamless playback through a user-friendly interface. This system offers a modern, efficient, and highly accessible platform for musical engagement, designed for enhanced usability and convenience.

# Acknowledgement

# Table of Contents

# Introduction

## Objective

The primary objective of the System is to deliver a modern, efficient, and highly accessible system for musical engagement, designed for enhanced usability and convenience. This is achieved by providing Service Administrators with comprehensive control over content and operational management, while simultaneously offering Clients a delightful, intuitive, and seamless experience for selecting and enjoying music box melodies.

## Product Scope

- This system will be used by Music Box's Service Administrators and Clients. It will focus on enhancing musical engagement and service delivery, providing comprehensive control for administrators and a seamless experience for clients. The system's scope includes:

- **Booking Management :** Management of room bookings and tracking of availability and impending reservations.
- **Content Management :** Administration of the library of songs, albums, and artists.
- **Staff and Language Management :** Tools for managing staff accounts and system language settings.
- **Sales Reporting  :** Access to automated sales reports for service oversight.
  **Melody Selection :** Facilitating client selection of songs from a collection of melodies.
- **Seamless Playback :** Ensuring uninterrupted music playback for clients.

## Intended Audience

This system is designed specifically for **Service Administrators** who oversee the daily operations of the music box service and **Clients** who wish to enjoy music box melodies.

# System Requirements

## Functional Requirements

**For Service Administrators**

- **Room Management :** Service Administrators will have full control over room management, including the ability to see, add and amend room bookings. The system will also track room availability and impending reservations.

- **Content Management :** Service Administrators can update, delete, and manage the library of songs, albums and artists.

- **Staff and Language Management :** The system will provide Service Administrators with tools to manage staff accounts and system language settings.

- **Sales Reporting :** Service Administrators will have access to automated sales report for efficient service oversight.

**For Clients**

- **Melody Selection :** Clients can effortlessly select their preferred songs from a collection of melodies.
- **Seamless Playback :** The system will ensure clients enjoy uninterrupted playback through a user-friendly interface.

## Non-Functional Requirements

The system aims to be modern, efficient, and highly accessible.

- **Usability :** The system is designed for enhanced usability and convenience, offering a delightful and intuitive experience for clients.
- **Performance :** The system will provide seamless playback for clients.
- **Accessibility :** The system is intended to be highly accessible.

# Methodology

The development of the Music Box System will employ an **Agile development methodology**. Agile is chosen for its iterative, flexible, and collaborative nature, which is highly beneficial for dynamic projects with evolving requirements.

## Why Agile for This Project?

Agile methodology is particularly well-suited for the Music Box Karaoke System due to its ability to:

- **Respond to Change Quickly:** Agile enables the team to rapidly integrate customer feedback and adapt to new requirements without significantly disrupting project timelines or plans.

- **Mitigate Risk:** Our system follows the Agile methodology, allowing continuous improvement and quick adaptation to changes. Our system, designed for

everyone who wants to enjoy music, emphasizes portability and ease of use. By enabling users to sing effortlessly anytime, anywhere, we reduce user frustration and technical issues that could hinder the singing experience.

- **Enhance Collaboration:** Agile fosters close communication and teamwork, ensuring all team members are aligned and contributing effectively towards a shared vision.

- **Deliver Value Incrementally:** Instead of a single, large release, Agile allows for the frequent delivery of working system components. This provides tangible progress, facilitates early user testing, and ensures the final product truly meets user needs.

# Design

## System Design and Diagram

This section outlines the system's architecture through use case diagrams, flow charts, and the database schema.

## Use Case Diagrams

Use case diagrams illustrate the interactions between users (actors) and the system

# Use Case for User Dashboard

**Usecase for UserDashboard**



# Use Case for Admin Dashboard

**Usecase for AdminDashboard**



8

# Flow Charts

Flow charts map the step-by-step processes within the system.

## Flow Chart for AdminDashboard

## Flow Chart for UserDashboard

# Database Schema

# Class Diagram

**Songs**
- -songId(int)
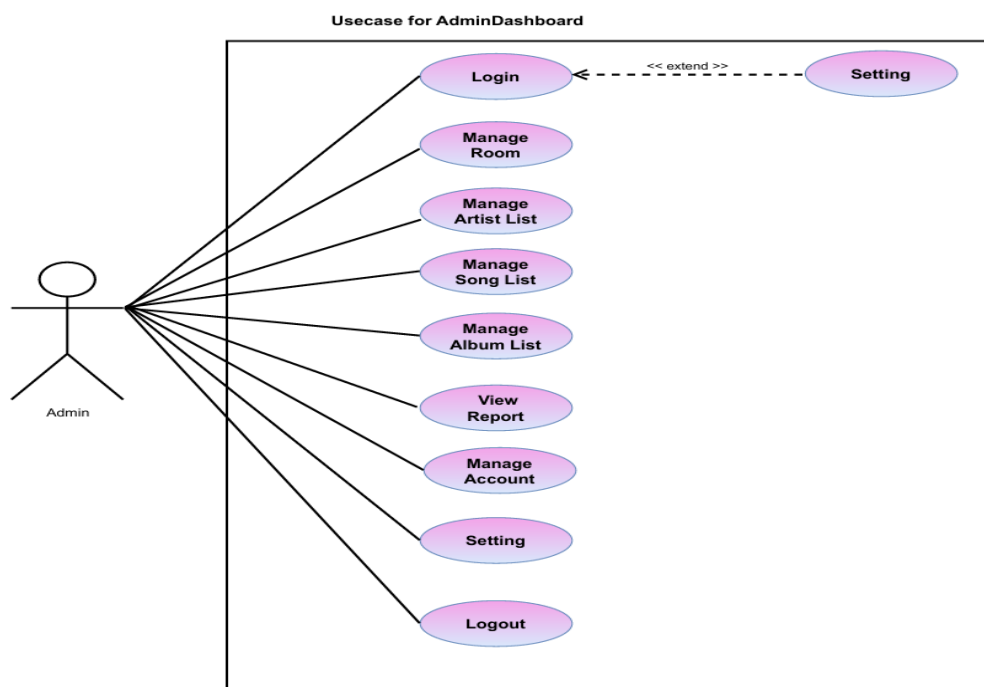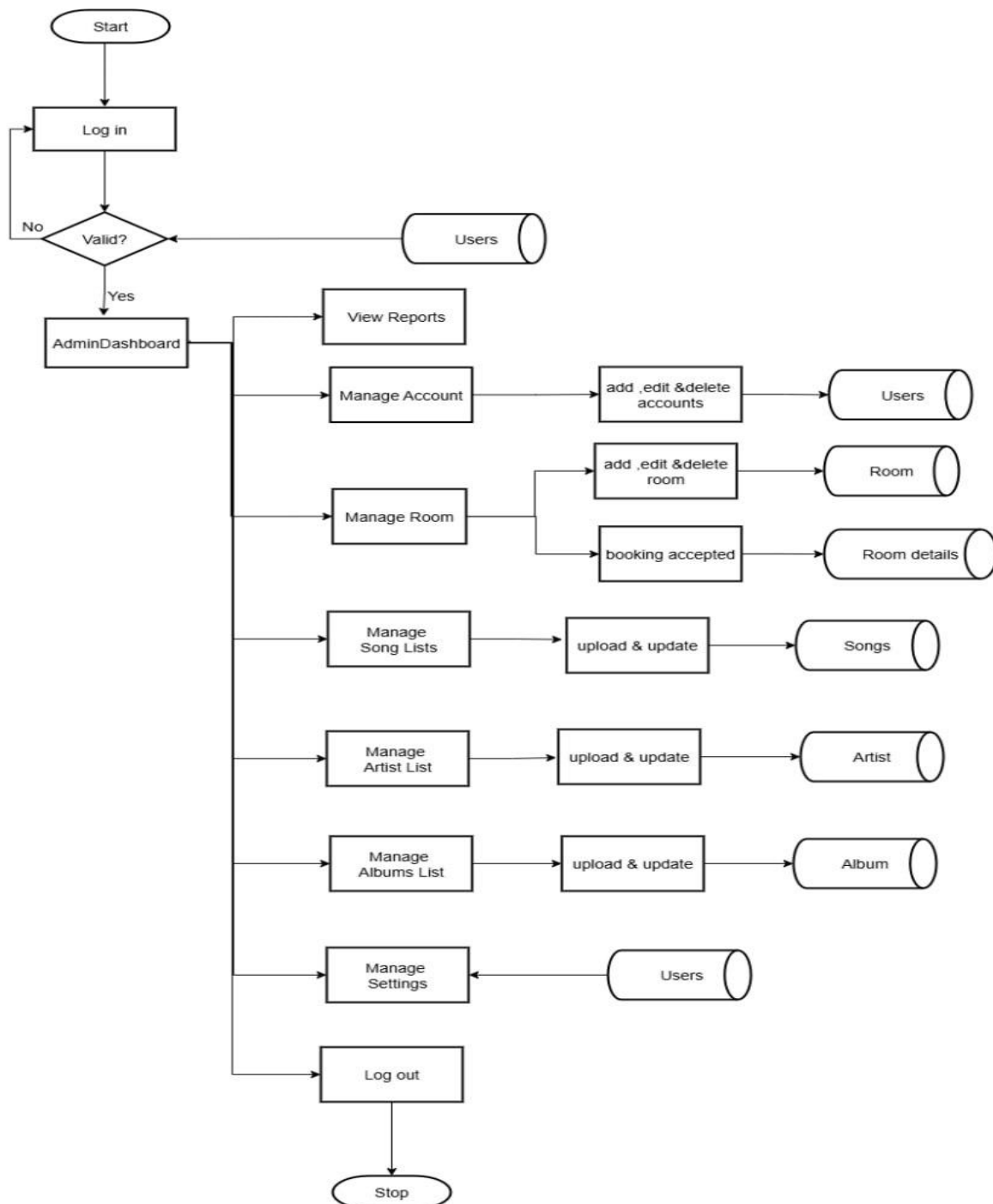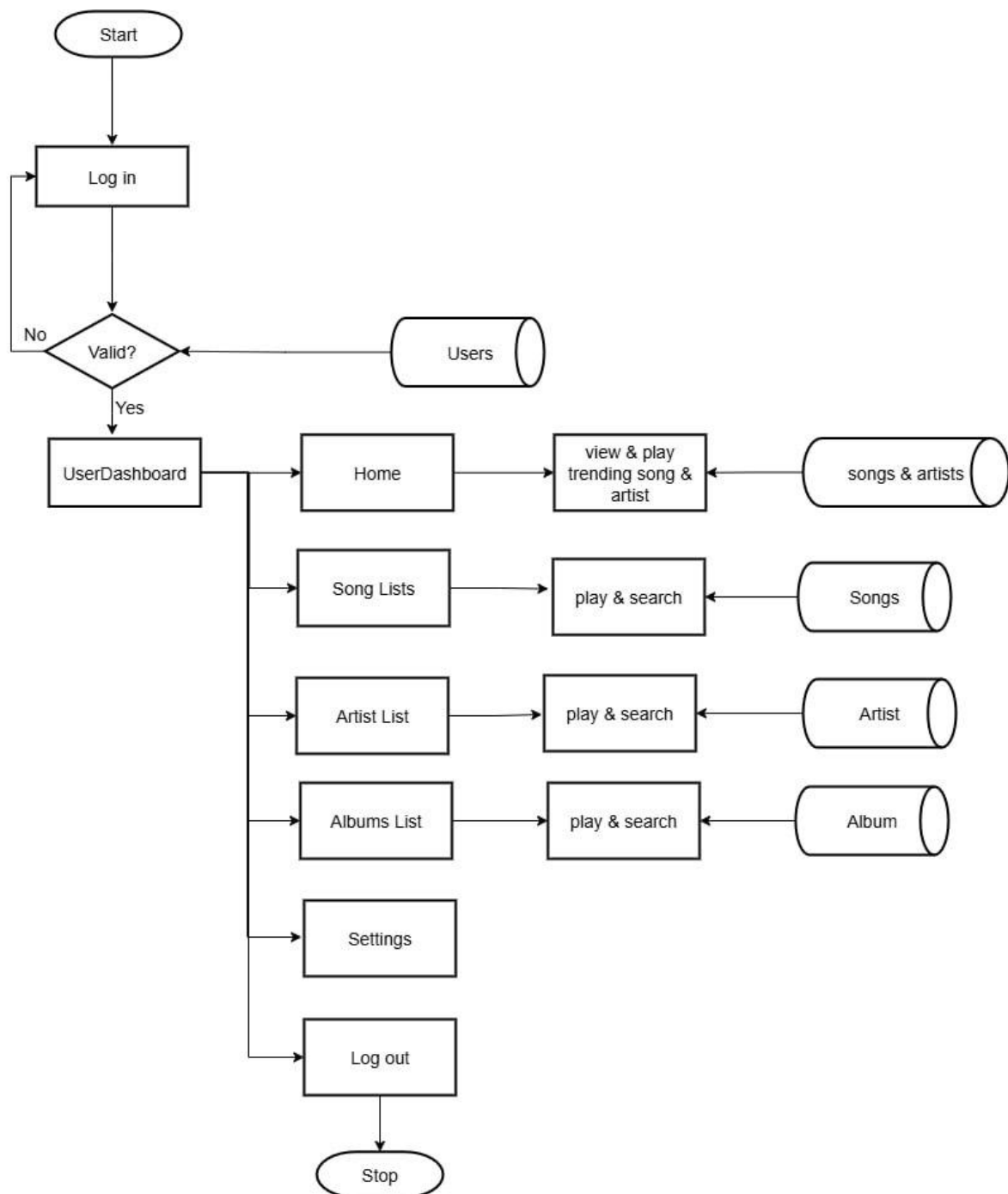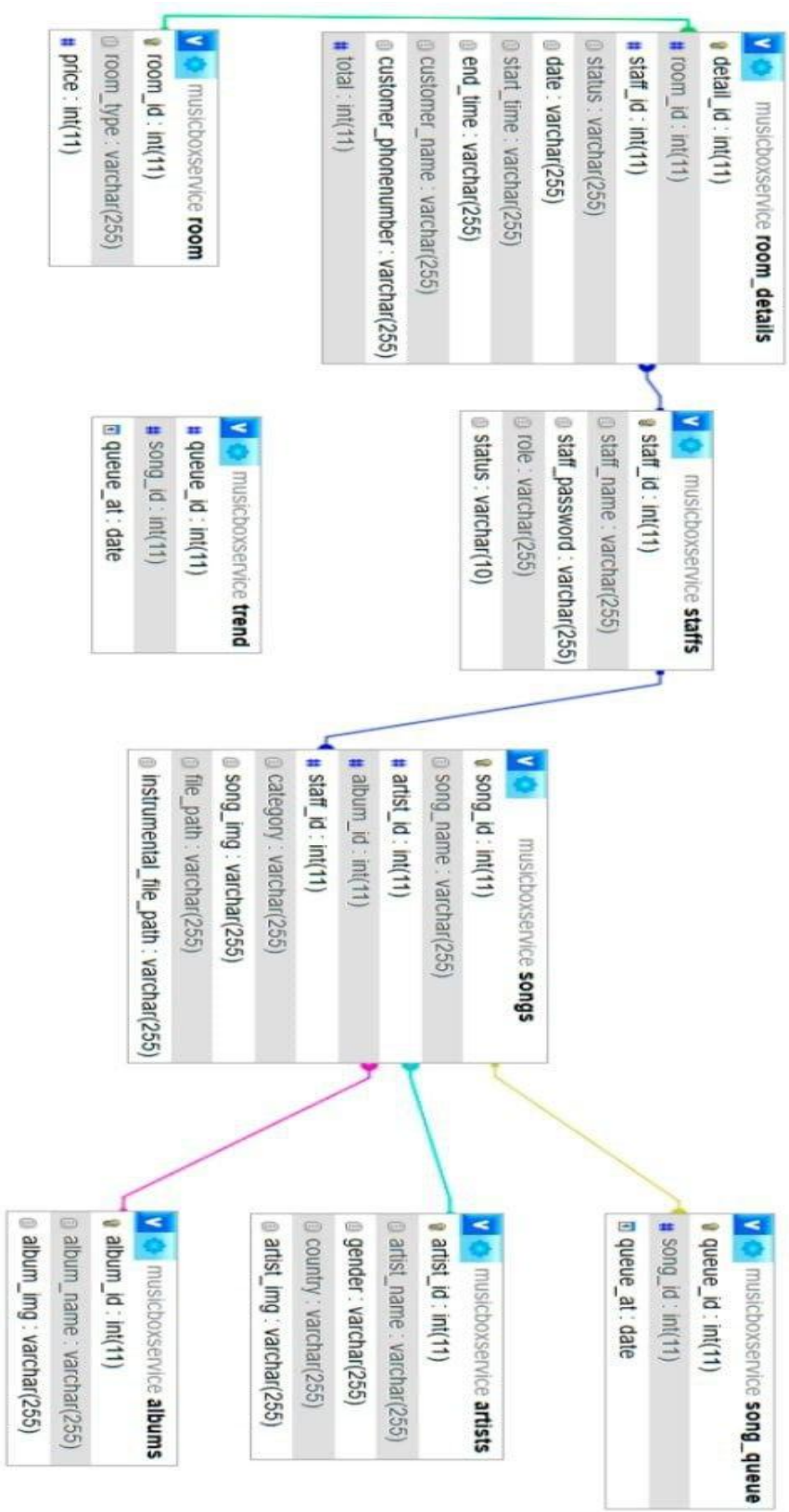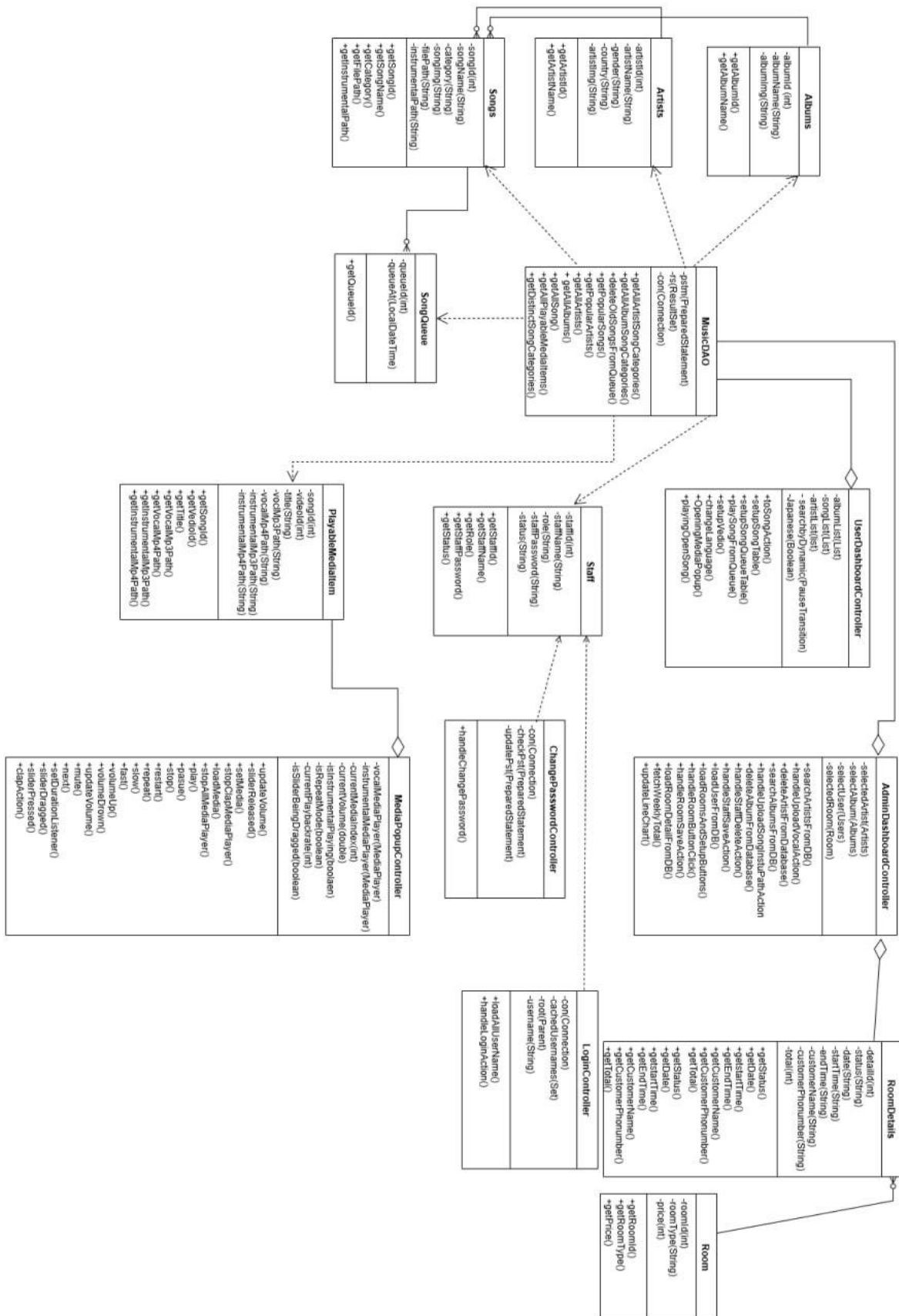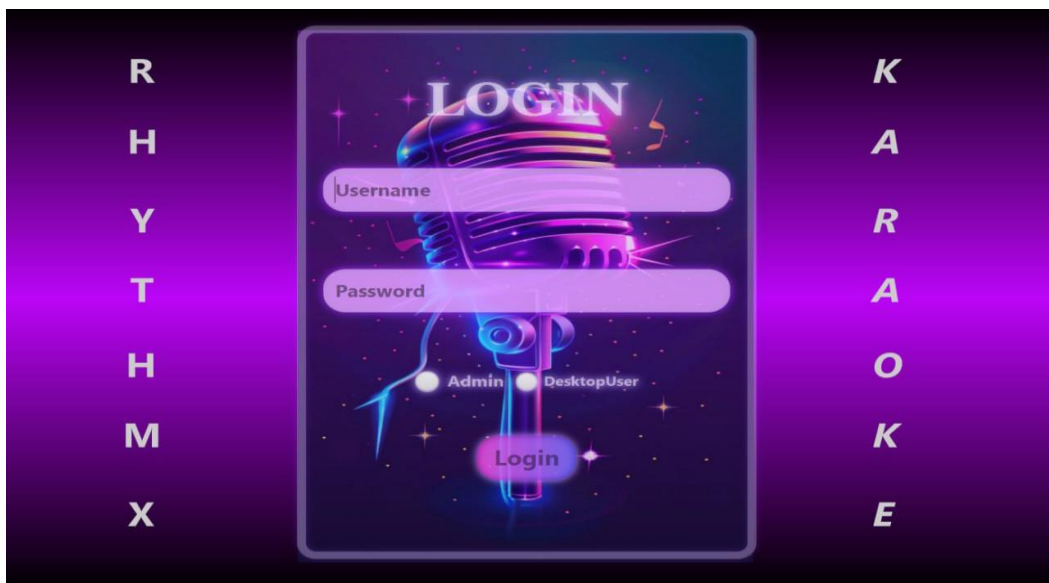- -songName(String)
- -category(String)
- -songImg(String)
- -filePath(String)
- -instrumentalPath(String)
- +getSongId()
- +getSongName()
- +getCategory()
- +getFilePath()
- +getInstrumentalPath()

**Artists**
- -artistId(int)
- -artistName(String)
- -gender(String)
- -country(String)
- -artistImg(String)
- +getArtistId()
- +getArtistName()

**Albums**
- -albumId (int)
- -albumName(String)
- -albumImg(String)
- +getAlbumId()
- +getAlbumName()

**MusicDAO**
- -pstm(PreparedStatement)
- -rs(ResultSet)
- -con(Connection)
- +getAllArtistSongCategories()
- +getAllAlbumSongCategories()
- +deleteOldSongsFromQueue()
- +getPopularSongs()
- +getPopularArtists()
- +getAllArtists()
- +getAllAlbums()
- +getAllSongs()
- +getAllPlayableMediaItems()
- +getDistinctSongCategories()

**SongQueue**
- -queuedId(int)
- -queueAt(LocalDateTime)
- +getQueueId()

**PlayableMediaItem**
- -songId(int)
- -videoId(int)
- -title(String)
- -vocalMp3Path(String)
- -vocalMp4Path(String)
- -instrumentalMp3Path(String)
- -instrumentalMp4Path(String)
- +getSongId()
- +getVedioId()
- +getTitle()
- +getVocalMp3Path()
- +getVocalMp4Path()
- +getInstrumentalMp3Path()
- +getInstrumentalMp4Path()

**Staff**
- -staffId(int)
- -staffName(String)
- -role(String)
- -staffPassword(String)
- -status(String)
- +getStaffId()
- +getStaffName()
- +getRole()
- +getStaffPassword()
- +getStatus()

**UserDashboardController**
- -albumList(List)
- -songList(List)
- -artistList(list)
- +toSongAction()
- +setupSongTable()
- +setupSongQueueTable()
- +playsSongFromQueue()
- +setupVedio()
- +changeLanguage()
- +OpeningMediaPopup()
- +playingOpenSong()
- -search(byDynamic(Pause Transition)
- -Japanese(Boolean)

**MediaPopupController**
- -vocalMediaPlayer(MediaPlayer)
- -instrumentalMediaPlayer(MediaPlayer)
- -currentMediaIndex(int)
- -currentVolume(double)
- -isInstrumentaPlaying(boolean)
- -isRepeatMode(boolean)
- -currentPlaybackrate(int)
- -isSliderBeingDragged(boolean)
- +updateVolume()
- +sliderReleased()
- +setMedia()
- +stopClapMediaPlayer()
- +loadMedia()
- +stopAllMediaPlayer()
- +play()
- +pause()
- +restart()
- +stop()
- +repeat()
- +slow()
- +last()
- +volumeUp()
- +volumeDrown()
- +updateVolume()
- +mute()
- +next()
- +setDurationListener()
- +sliderDragged()
- +sliderPressed()
- +clapAction()

**ChangePasswordController**
- -con(Connection)
- -checkPst(PreparedStatement)
- -updatePst(PreparedStatement)
- +handleChangePassword()

**AdminDashboardController**
- -selectedArtist(Artists)
- -selectedAlbum(Albums)
- -selectUser(Users)
- -selectedRoom(Room)
- -searchArtistsFromDB()
- -handleUploadVocalAction()
- -deleteArtistFromDatabase()
- -handleUploadSong/instuPathAction
- -deleteAlbumFromDatabase()
- -handleStaffDeleteAction()
- -handleStaffSaveAction()
- -loadUserFromDB()
- +getCustomerFromDB()
- +handleRoomsAndSetupButtons()
- +handleRoomButtonClick()
- +handleRoomSaveAction()
- +loadRoomDetailFromDB()
- +fetchWeeklyTotal()
- +updateLineChart()

**LoginController**
- -con(Connection)
- -cachedUsernames(Set)
- -root(Parent)
- -username(String)
- +loadAllUserName()
- +handleLoginAction()

**RoomDetails**
- -detailId(int)
- -status(String)
- -date(String)
- -startTime(String)
- -endTime(String)
- -customerName(String)
- -customerPhonumber(String)
- -total(int)
- +getStatus()
- +getDate()
- +getStartTime()
- +getEndTime()
- +getCustomerName()
- +getCustomerPhonumber()
- +getTotal()

**Room**
- -roomId(int)
- -roomType(String)
- -price(int)
- +getRoomId()
- +getRoomType()
- +getPrice()

12

# User Interface

## Login

The system is accessed via a login screen where users enter their Username and Password. Users must select their role, either "Admin" or "Desktop User," before clicking the "Login" button.



## Home

The home screen serves as the main dashboard after login. It features a section for "Trending Songs" and "Popular Singers" to give users an overview of what is currently popular. The sidebar allows navigation to Home, Artists, Songs, Album, Setting, and includes a Log out option.
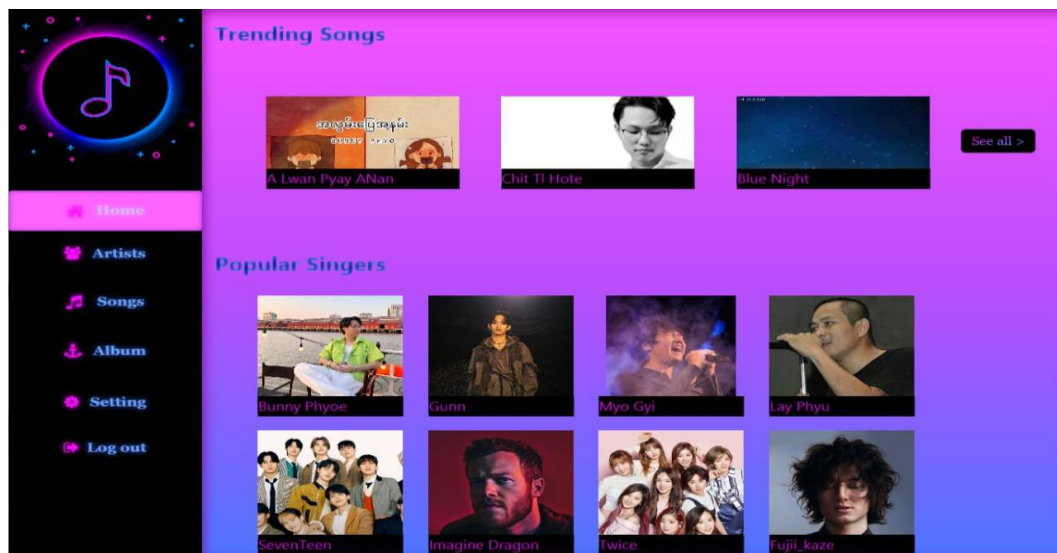
## Artist

On this page, users can view all available artists displayed in a grid format. A search bar and a category dropdown menu are provided to help users find specific artists easily.
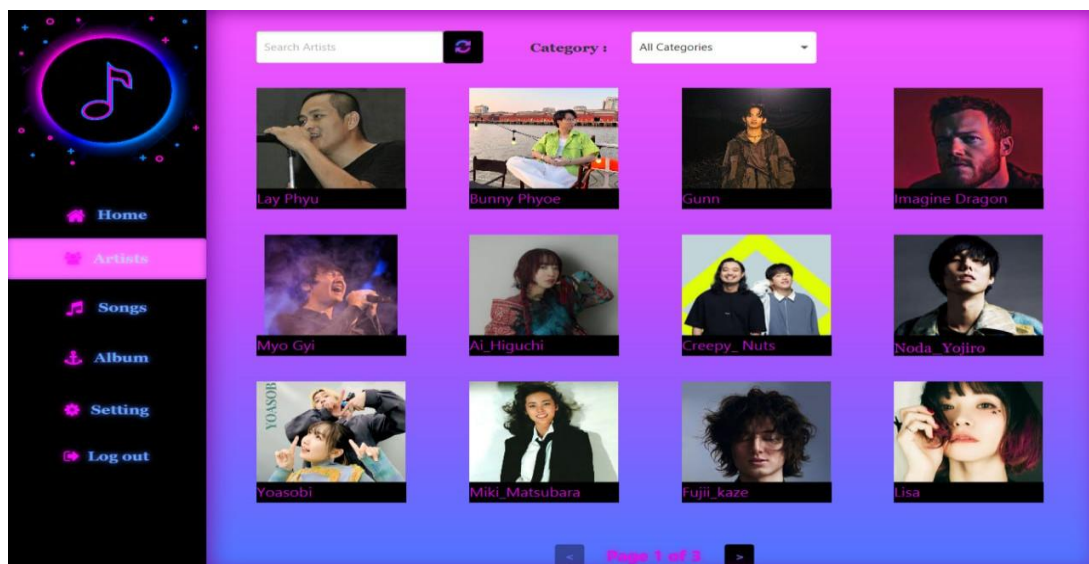
## Song

This screen allows users to browse all available songs. They can select songs to add to a personalized playlist shown on the right side of the screen, labeled "Your Song List". Users have the option to remove songs one by one or clear the entire list with a "Remove all" button.
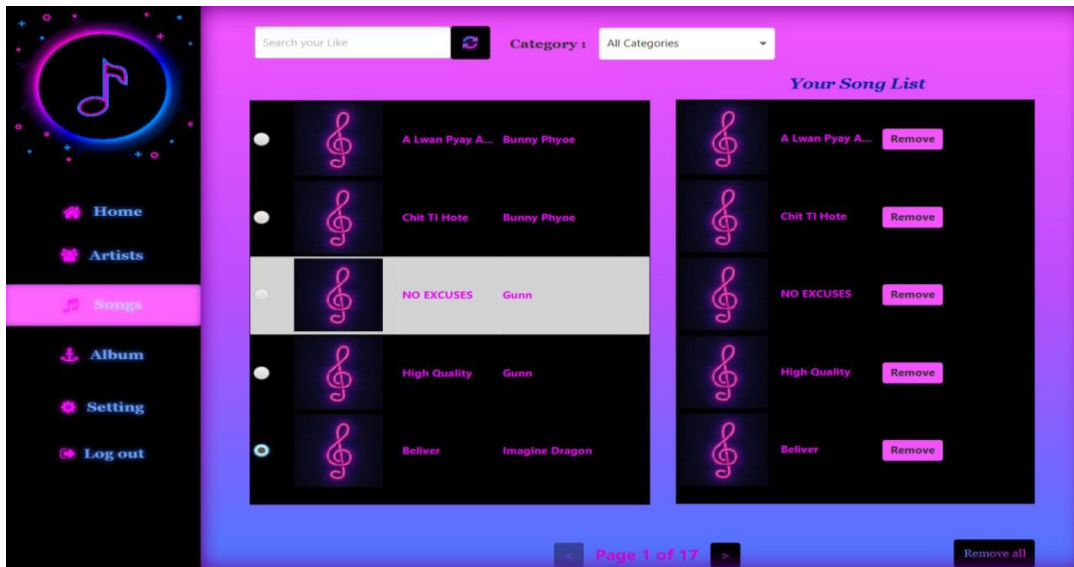


## Album

The album page displays a collection of album covers in a grid layout. Similar to the Artists and Songs pages, it includes a search bar and a category filter to help users browse the available albums.
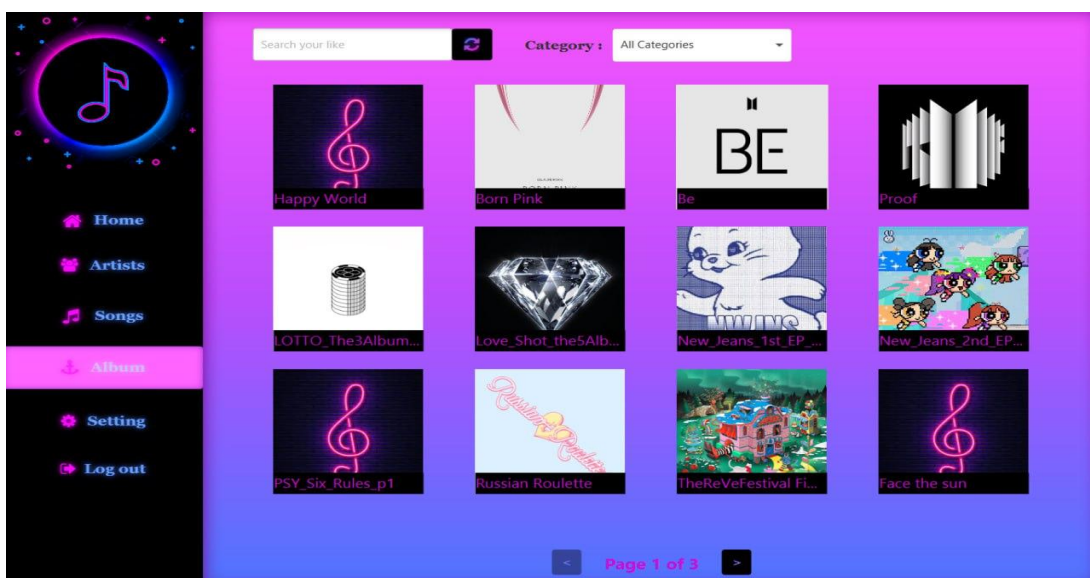
## Setting

The settings menu provides users with options for account and language preferences. Users can "Change to Japanese" via a toggle switch and navigate to a separate screen to "Change Password".
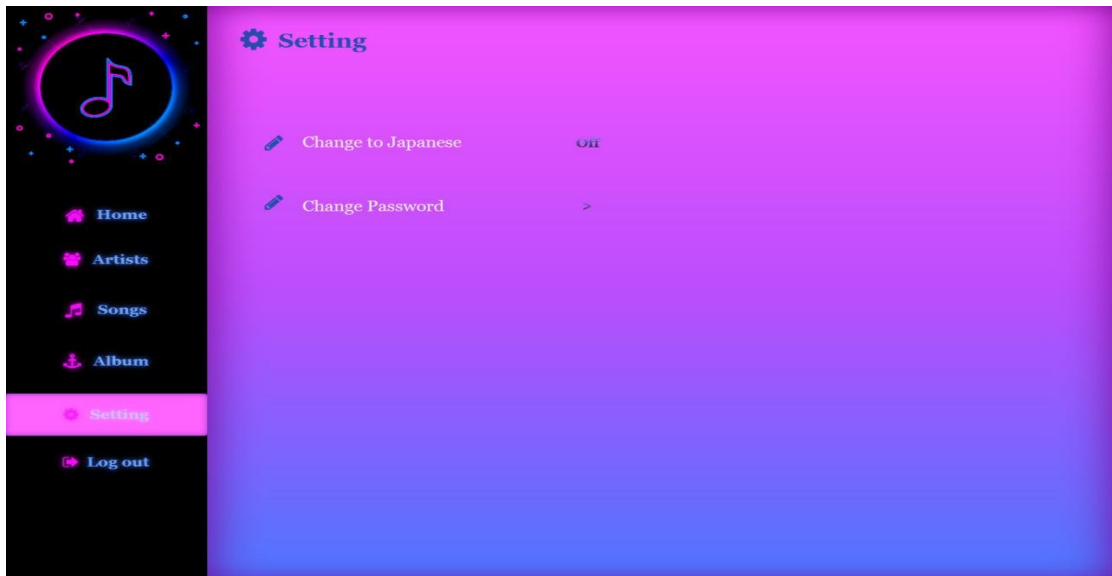


## Admin Dashboard Interface

The system provides a comprehensive administrative backend to manage the karaoke service's operations. Access is divided into at least two roles, an Admin (for regular staff) and a Main Admin (for management), each with a specific set of permissions and capabilities.

## Admin (Staff) Role

The standard Admin role is focused on day-to-day operational tasks, primarily managing customer room bookings.
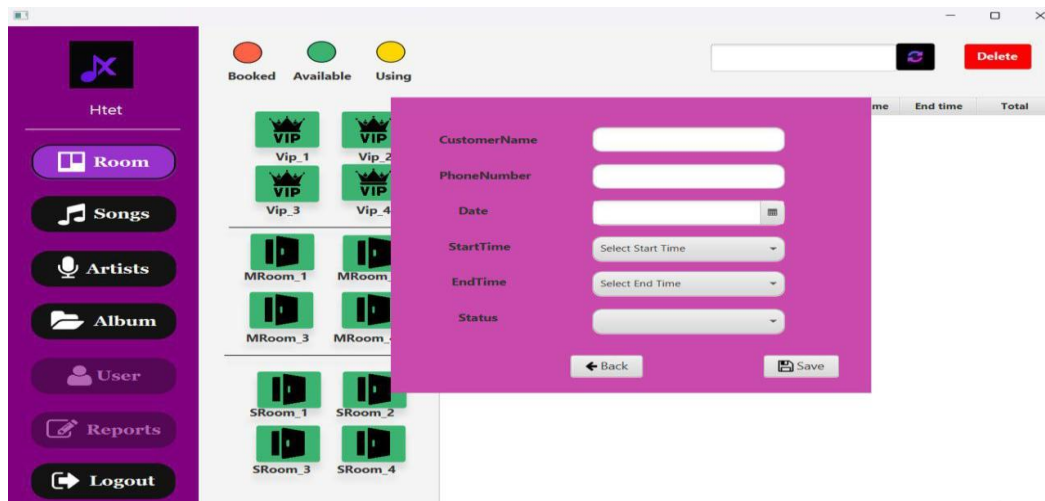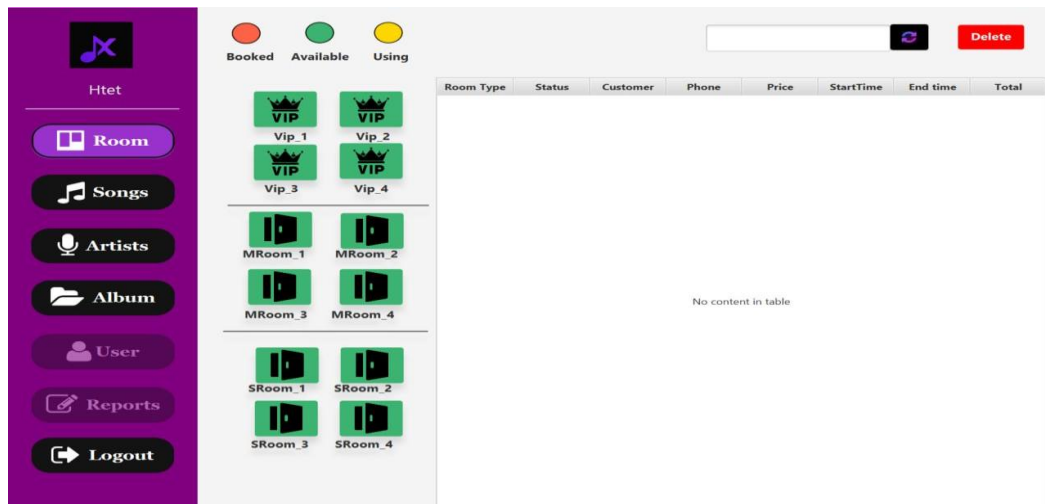
## Main Admin Role

The Main Admin has elevated privileges, including full control over the system's library management(CRUD) and access to financial reports for business analysis. The Main Admin sidebar includes navigation for Room, Songs, Artists, Album, User, Reports, and Logout.

**Artists Management**: The "Artists" tab displays all artists in a grid format. To add or edit an artist, a modal window appears.
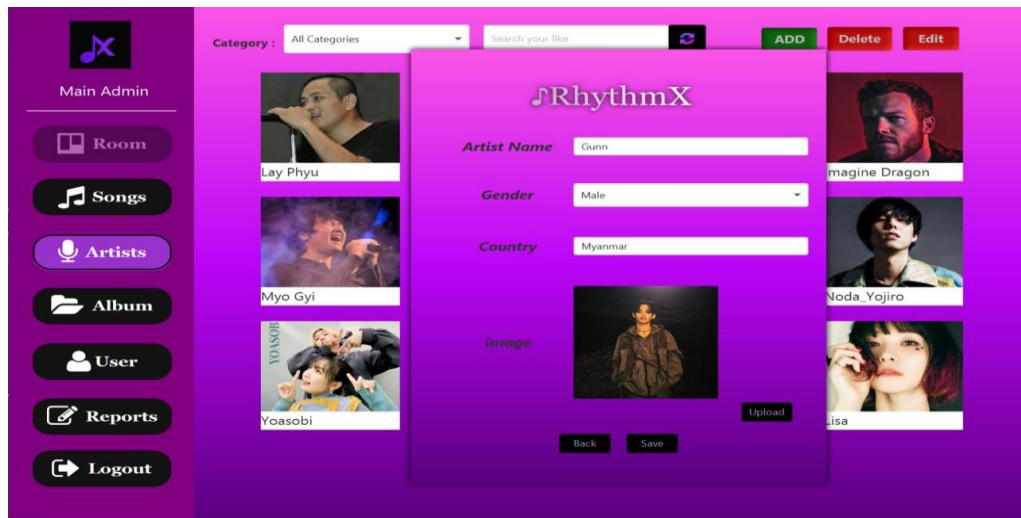
**Songs Management**: The "Songs" page shows the library of available songs. The modal for adding or editing a song.
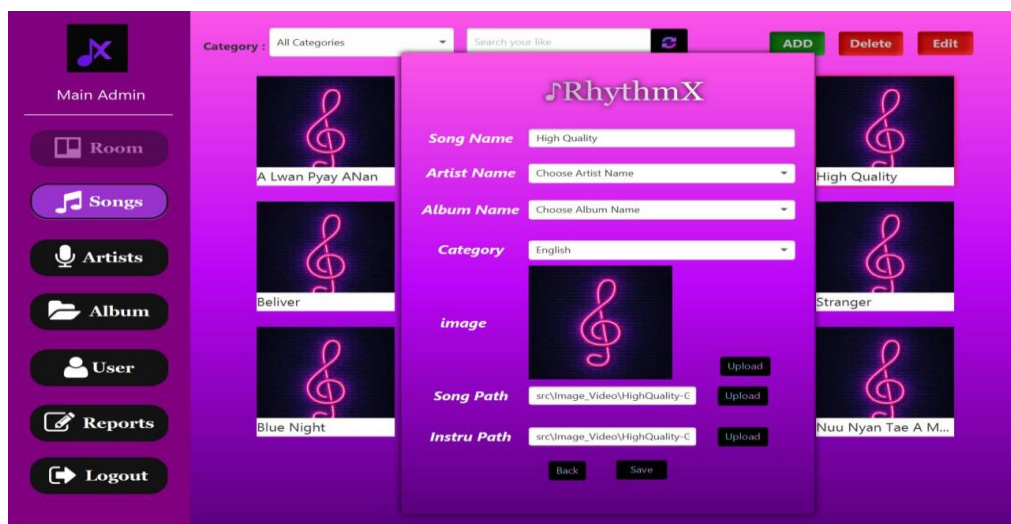


**Album Management**: The "Album" tab allows for the management of music albums. The pop-up form for adding or editing requires the Album Name and an upload of the album's cover image.
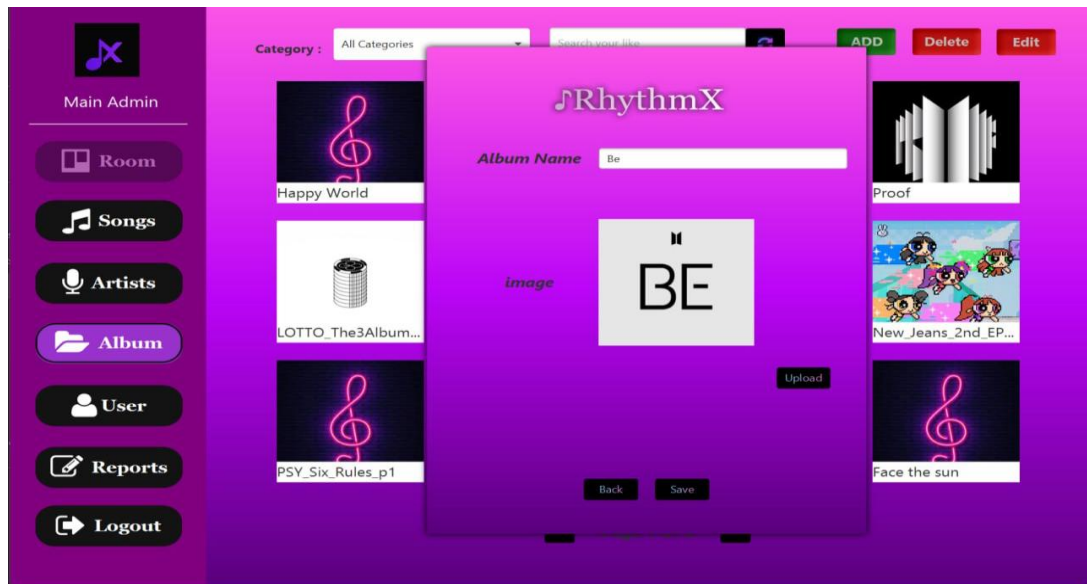
**Financial Reporting**

- A bar chart showing "Weekly Sales" amounts for each week of the month.
- A line chart comparing the sales performance of the "Current Month" against the "Previous Month".
- A "Details Income" section provides a breakdown of revenue generated from different room types: VIP Room, Medium Room, and Small Room.

- **User Management**

  From here, the Main Admin can likely perform actions such as:
  - Creating new user profiles for staff members (e.g., "Htet").
  - Assigning roles (e.g., "Admin") to limit access to certain features.
  - Editing user details and resetting passwords.
  - Deactivating or deleting accounts for former employees.

# REFERENCES

PopPop.ai. (n.d.). AI Vocal Remover: Song split instrumental and MP4

[Truly Free Vocal Remover: Online AI Vocal Isolator | PopPop AI](#)


YT1D. (n.d.). Song Downloader. [https://yt1d.com/en307/](https://yt1d.com/en307/)


Google. (n.d.). Gemini: AI for research.   [https://gemini.google.com/app](https://gemini.google.com/app)


Music Box Karaoke Food and Bar (Junction Square).

Location: R48J+XJ8 Junction Square, Pyay Rd, Yangon