TK MUSIC LIBRARY

Tasnia Kader

November 16, 2023

Table of Contents

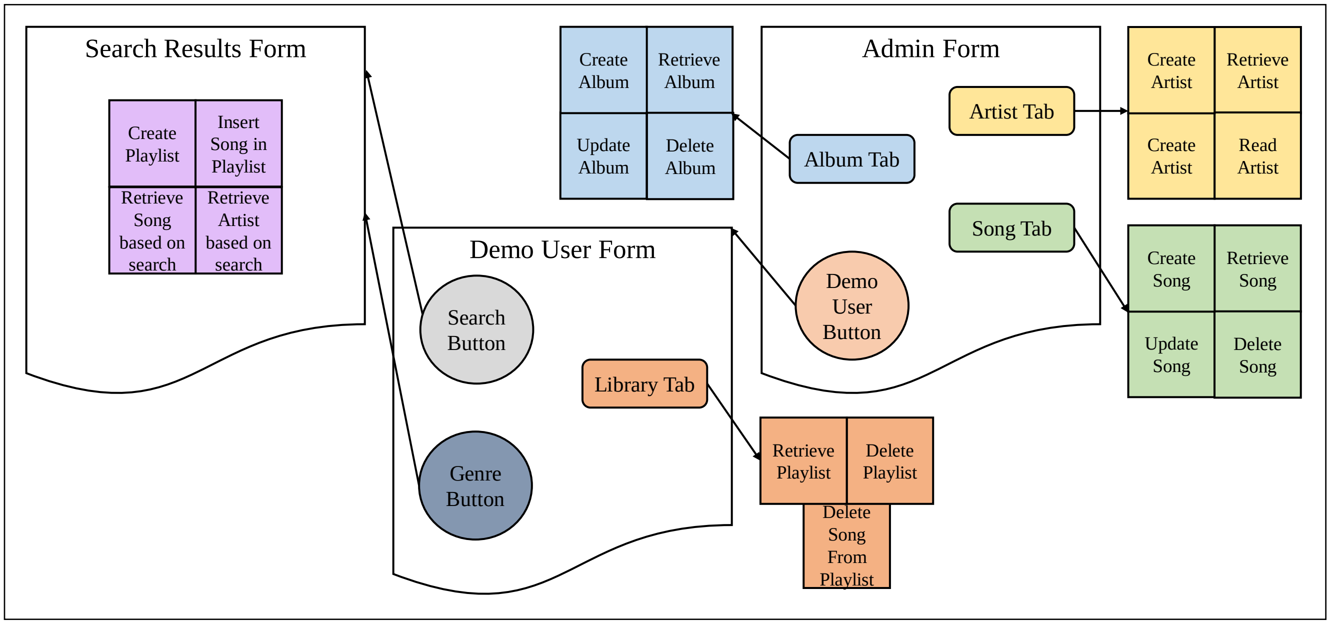
Introduction

From my early days in the elementary school choir, through my middle school band experience, to my high school guitar lessons, music has remained a constant and essential part of my life. Music is not just passive enjoyment for me; it's an active source of delight that involves listening, singing along, and dancing to my favorite songs. Moreover, I love curating playlists on popular platforms like YouTube, Spotify, and Apple Music. This activity allows me to craft personalized collections of my treasured tunes. These platforms, as I've come to appreciate, heavily depend on sophisticated databases for the storage and retrieval of music, user preferences, playlists, artist profiles, albums, and other data. Consequently, this project is born out of my desire to create a similar database system, providing me with insights into the intricacies of their construction and functionality.

Project Specifications

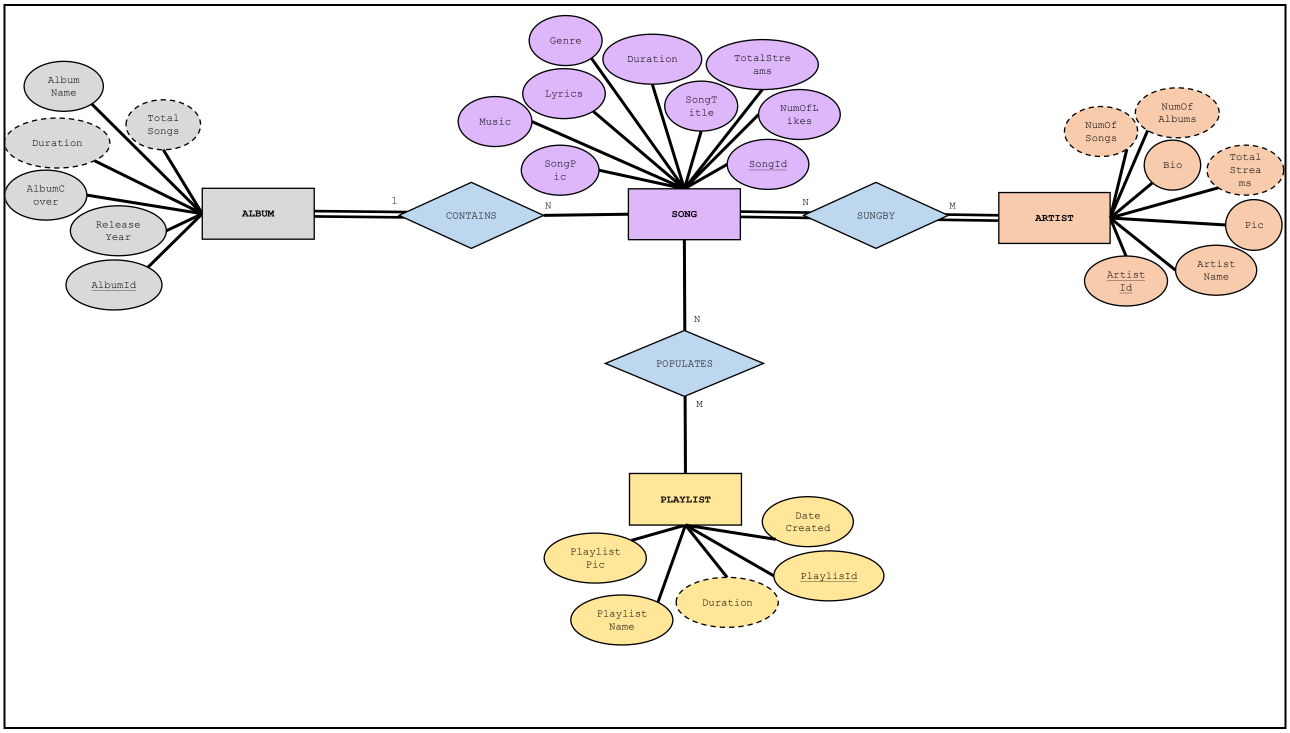
The TK music library is designed for the administrators and allows them to create, retrieve, update, and delete songs, artists, and albums. Additionally, administrators can access a demo user form designed to simulate the user experience. This demo user form enables users to search for songs or artists, explore content based on genre, and curate personalized playlists. It is important to note that this demo user form is not tied to a specific user; rather, it serves as a tool for administrators to preview and understand the interface from the perspective of regular users.

In this project, I have assumed that artists, songs, albums, and playlists have unique names.

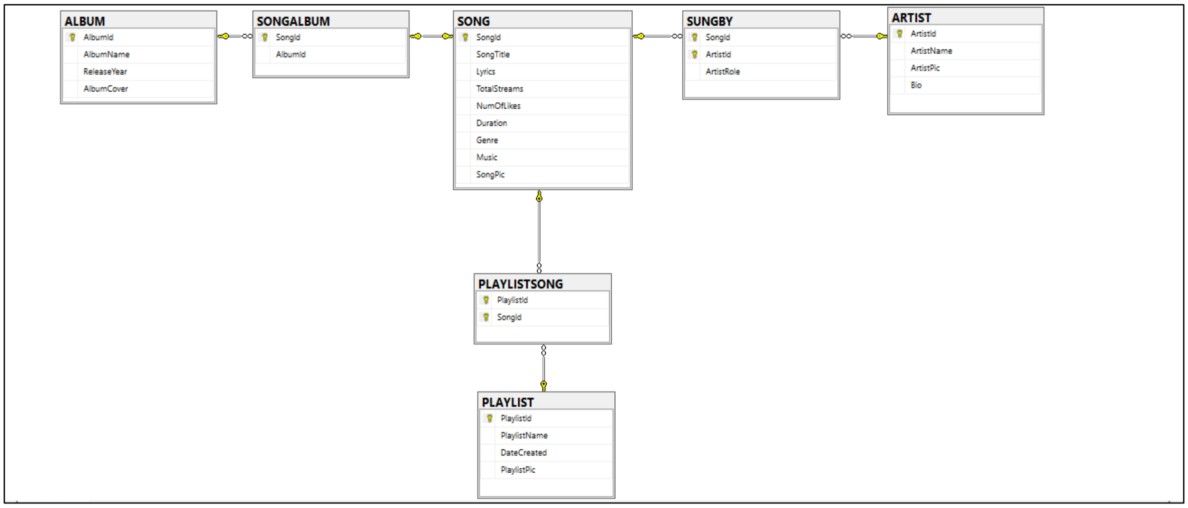


Design

Entity Relationship Diagram



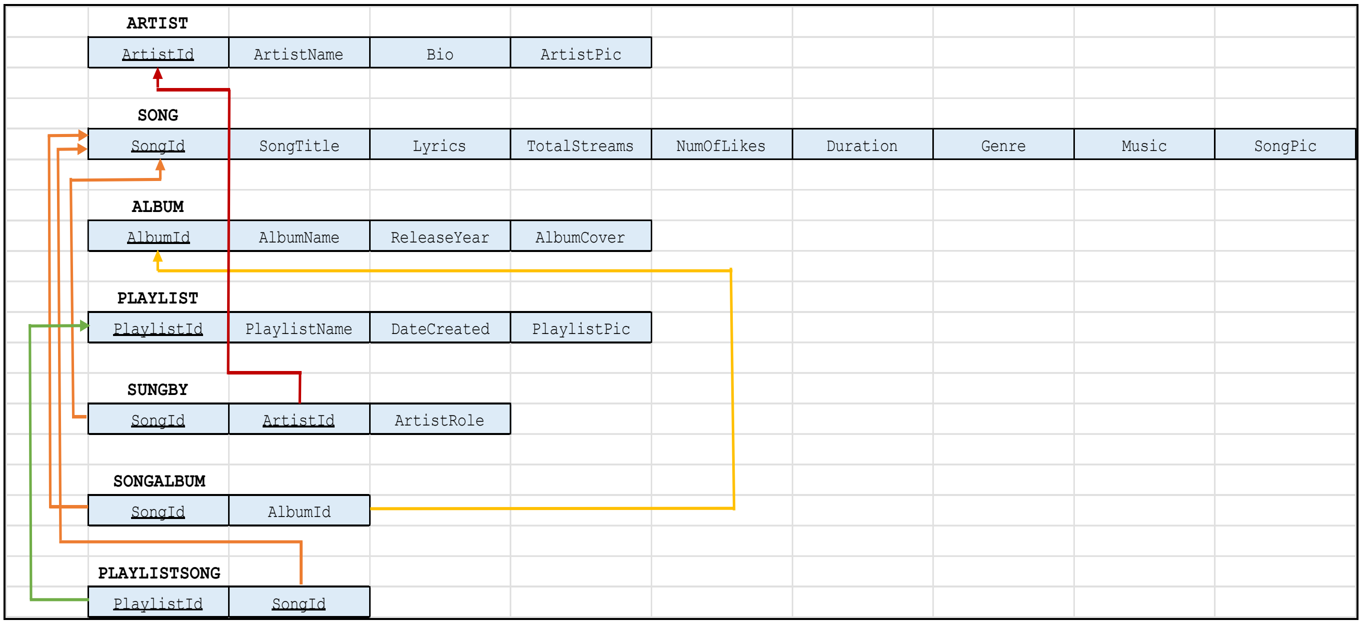
SQL Database Diagram



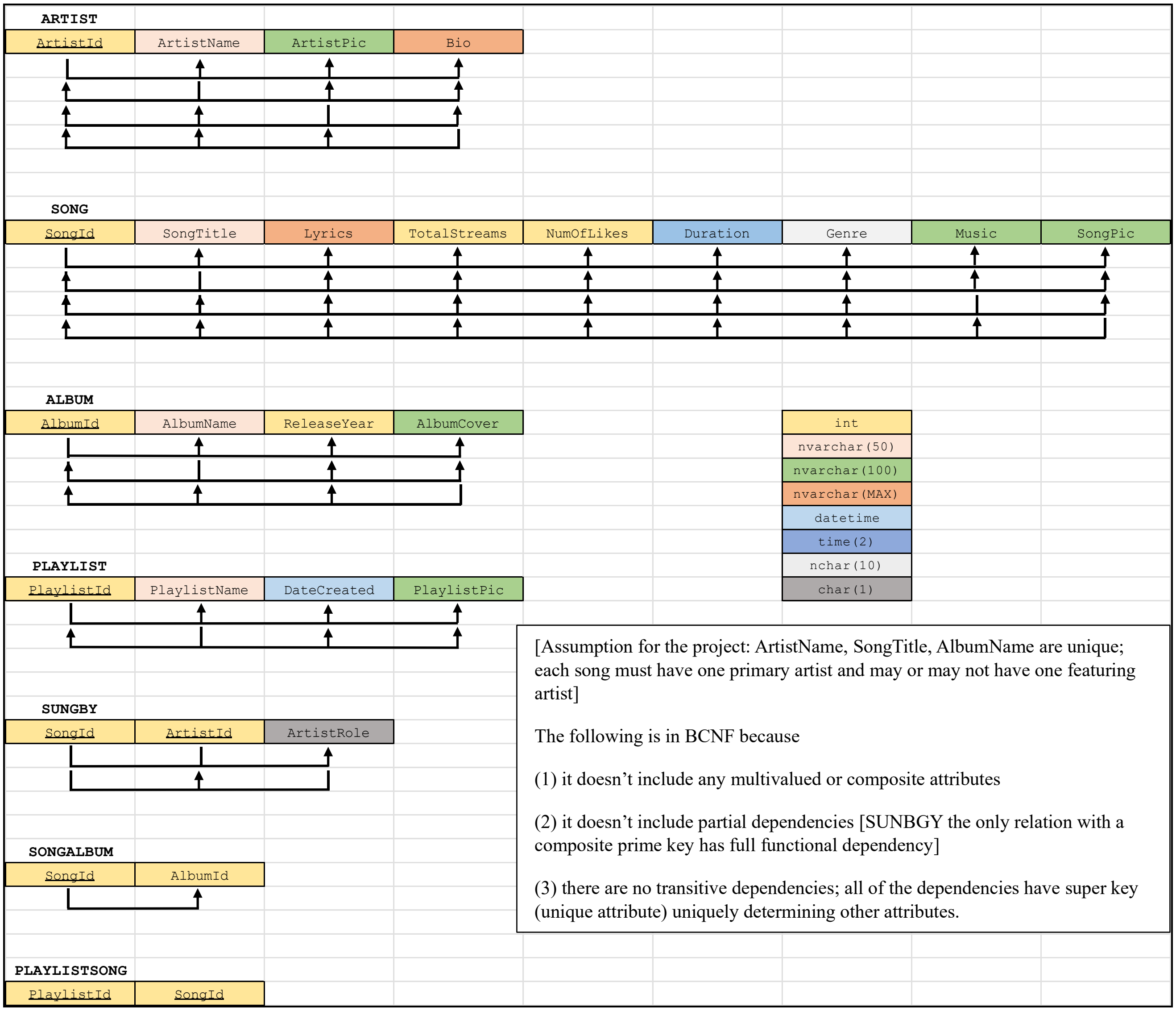
Schema Diagram

ARTIST								
ArtistId	ArtistName	Bio	ArtistPic					
SONG								
SongId	SongTitle	Lyrics	TotalStreams	NumOfLikes	Duration	Genre	Music	SongPic
ALBUM								
AlbumId	AlbumName	ReleaseYear	AlbumCover					
PLAYLIST								
PlaylistId	PlaylistName	DateCreated	PlaylistPic					
SUNGBY								
SongId	ArtistId	ArtistRole						
SONGALBUM								
SongId	AlbumId							
PLAYLISTSONG								
PlaylistId	SongId							

Relationships



Functional Dependencies



ARTIST

ArtistId	ArtistName	ArtistPic	Bio

SONG

SongId	SongTitle	Lyrics	TotalStreams	NumOfLikes	Duration	Genre	Music	SongPic

ALBUM

AlbumId	AlbumName	ReleaseYear	AlbumCover

PLAYLIST

PlaylistId	PlaylistName	DateCreated	PlaylistPic

SUNGBY

SongId	ArtistId	ArtistRole

SONGALBUM

SongId	AlbumId

PLAYLISTSONG

PlaylistId	SongId

int
nvarchar(50)
nvarchar(100)
nvarchar(MAX)
datetime
time(2)
nchar(10)
char(1)

[Assumption for the project: ArtistName, SongTitle, AlbumName are unique; each song must have one primary artist and may or may not have one featuring artist]

The following is in BCNF because

(1) it doesn't include any multivalued or composite attributes

(2) it doesn't include partial dependencies [SUNBGY the only relation with a composite prime key has full functional dependency]

(3) there are no transitive dependencies; all of the dependencies have super key (unique attribute) uniquely determining other attributes.

User Manual

Artist

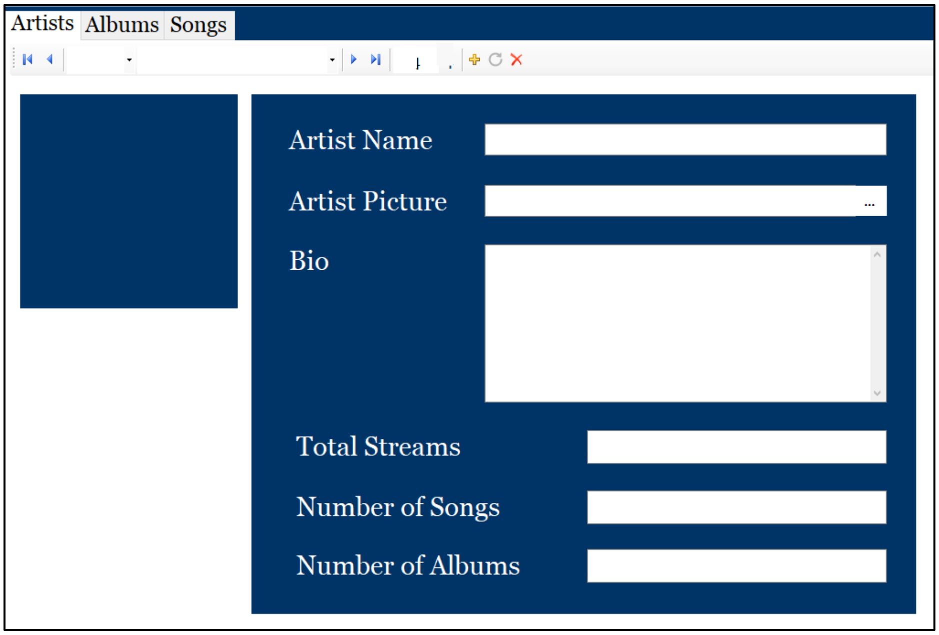


Figure 1 : Artist Tab

Create New Artist

1. Click on ✚ to clear the form.
2. Type the artist's name (must be unique)
3. Click on **…** to upload artist picture.
4. Type in or paste artist bio.
5. Click on ✚ to finish adding artist.

Notes:

- The artist id will automatically be generated and displayed in the navigation panel after artist has been inserted.
- The total streams, number of songs, number of albums will automatically be set to 0.
- The following formats are supported for the artist picture: .bmp, .gif, .jpeg, .png, .tiff, .jfif, .jpg.
- To get out of the insert state, click on a different tab.

Update Artist Information

1. Navigate to the artist whose information needs to be updated.
2. Type the updated artist name and/or bio. Click on **…** to insert a new artist picture.
3. Click on ↻ to update artist information.

Notes:

- The total streams, number of songs, number of albums cannot be updated manually and will remain the same after artist info has been updated.

Delete Artist

1. Navigate to the artist who needs to be deleted.
2. Click on ✕ to delete artist.

Notes:

- Deleting an artist will delete all the artist's songs and albums.

Album

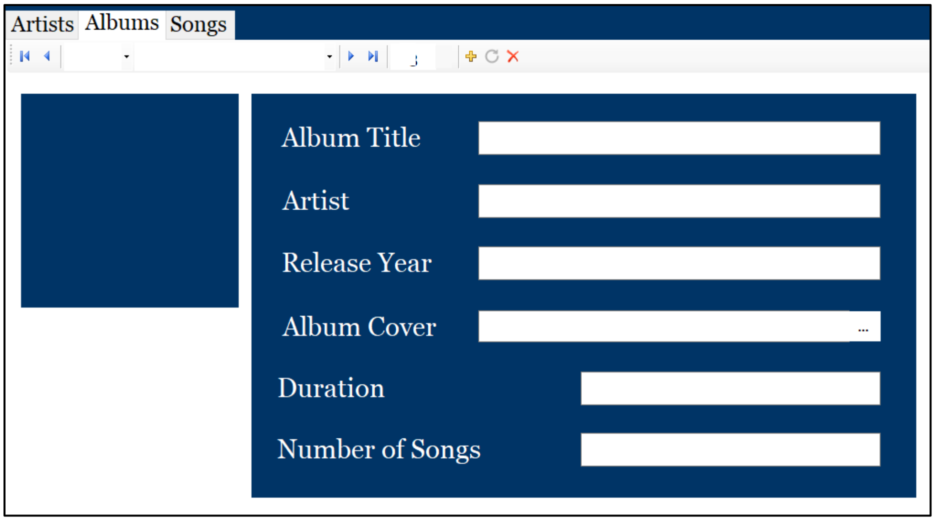


Figure 2 : Album Tab

Create New Album

1. Click on ✚ to clear the form.
2. Type the album title (must be unique)
3. Type the release year.
4. Click on **…** to upload album cover.
5. Click on ✚ to finish adding album.

Notes:

- The artist, duration, and number of songs will automatically be set to default values and updated once songs are inserted into the album.
- The album id will automatically be generated and displayed in the navigation panel after artist has been inserted.
- The following formats are supported for the album picture: .bmp, .gif, .jpeg, .png, .tiff, .jfif, .jpg.
- To get out of the insert state, click on a different tab.

Update Album Information

1. Navigate to the album whose information needs to be updated.
2. Type the updated album title and/or release year. Click on **…** to insert a new album cover.
3. Click on ↻ to update album information.

Notes:

- The artist, duration, and number of songs cannot be updated manually and will remain the same after album info has been updated.

Delete Album

1. Navigate to the album that needs to be deleted.
2. Click on × to delete the album.

Notes:

- Deleting an album will delete all the songs in the album.

Song

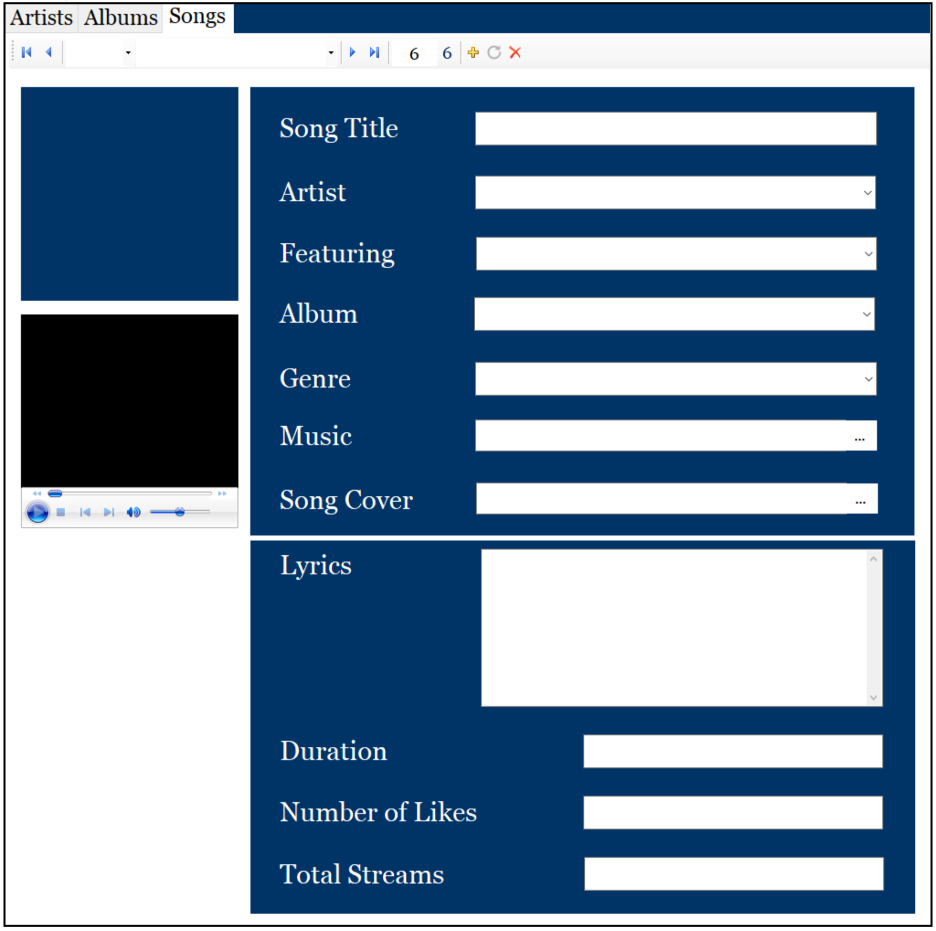


Figure 3 : Song Tab

Create New Song

1. Click on ✚ to clear the form.
2. Select the primary artist’s name and the featuring artist’s name if it exists from the drop-down options.
3. Select the album name if it exists from the drop-down options.
4. Select the genre from the drop-down options.
5. Click on **…** located next to **Music** to upload the song.
6. Click on **…** located next to **Song Cover** to upload the song cover.
6. Type in or paste the song lyrics.
7. Click on ✚ to finish adding song.

Notes:

- The **Song Title** is automatically set to the name of the media file.
- The duration, total streams, and number of likes will automatically be set to default values and updated.
- The song id will automatically be generated and displayed in the navigation panel after the artist has been inserted.
- The following formats are supported for the song picture: .bmp, .gif, .jpeg, .png, .tiff, .jfif, .jpg.
- The following formats are supported for the music: .wav, .mp3, .ogg, .flac, .mp4.
- To get out of the insert state, click on a different tab.

Update Song Information

4. Navigate to the song whose information needs to be updated.
5. Type the updated song title and/or lyrics. Select the updates primary artist, featuring artist, album, and/or genre. Click on **…** to insert a new song cover and/or music.
6. Click on ↻ to update album information.

Notes:

- The duration, number of likes, and total streams cannot be updated manually and will remain the same after song info has been updated.

Delete Song

7. Navigate to the song that needs to be deleted.
8. Click on × to delete the song.

Notes:

- Deleting a song will delete that song from all the playlists.

Demo User

Search for Song/Artist

1. Click on 👤 located on top right corner of the admin page.
2. Click on the home tab.
3. Type the name of the artist or song.
4. Click on 🔍.

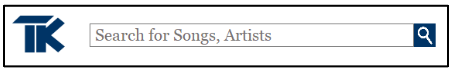
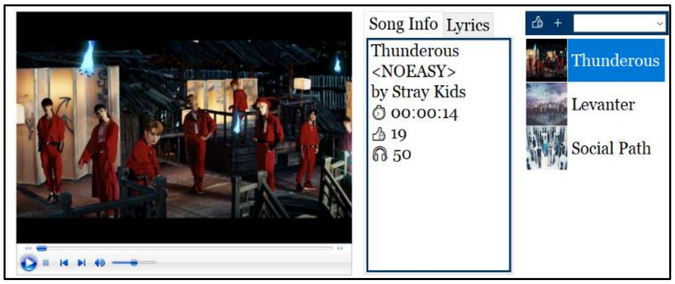


Figure 4 : Searching for songs/artists

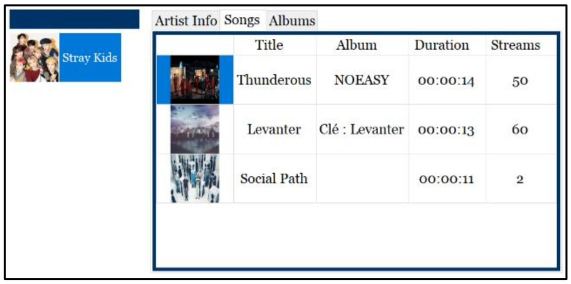
View Songs or Artists

1. Click on 👤 located on top right corner of the admin page.
2. Click on the home tab.
3. Search for a song or click on one of the genres to display songs.
4. From the songs displayed on the right-hand side of the screen, click on a song. The song information will be displayed in the Song Info Tab and the lyrics will be shown in the Lyrics Tab.
5. Scroll down to view the artists.
6. Click on an artist to view artist bio in the Artist Info Tab, their songs in the Songs Tab and their albums in the Albums Tab.



Thunderous
<NOEASY>
by Stray Kids
⏱ 00:00:14
👍 19
🎧 50

Figure 5 : Viewing songs



Title	Album	Duration	Streams
Thunderous	NOEASY	00:00:14	50
Levanter	Clé : Levanter	00:00:13	60
Social Path		00:00:11	2

Figure 6 : Viewing artists

Like a Song

1. Search for a song in the home tab of the demo user page or click on one of the genres.
2. Click on a song that is displayed on the right-hand side of the screen.
3. Click on 👍.

Notes:

- One song can be liked as many times as desired.

Insert Song in Playlist

1. Click on 👤 located on top right corner of the admin page.
2. Click on the home tab.

3. Search for a song or click on one of the genres to display songs.
4. From the songs displayed on the right-hand side of the screen, click on a song.
5. Type in the name of the playlist in the drop-down box to create a new playlist or select from one of the existing playlists.
6. Click on + to add the selected song to the playlist.

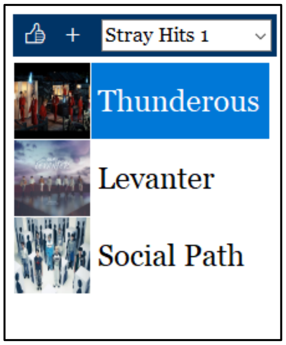


Figure 7 : Adding song to playlist

Delete Song from Playlist

1. Click on 👤 located on top right corner of the admin page.
2. Click on the library tab.
3. Click on one of the playlists displayed above the media player.
4. Click on one of the songs displayed on the right-hand side of the screen.
5. Click on × to delete the song from the playlist.

Notes:

- Once all the songs from a playlist is deleted, the playlist will automatically be deleted.
- To see the updated playlist, click on a different tab, and then click on the library tab.

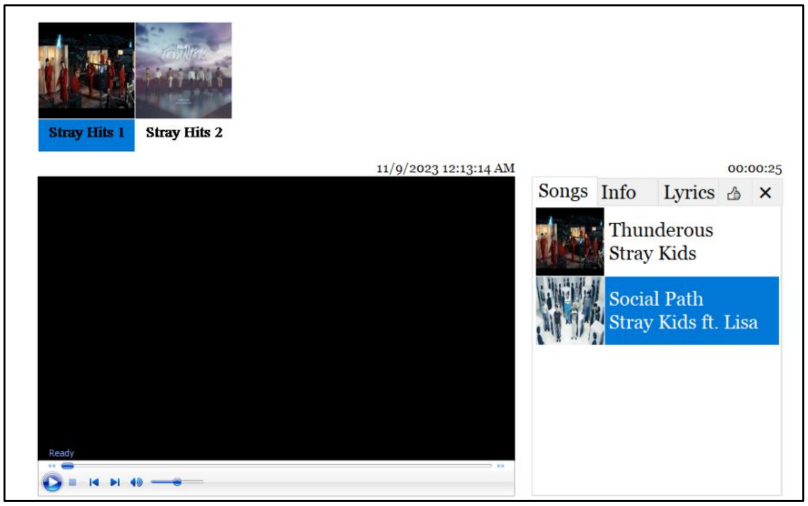


Figure 8 : Deleting a song from a playlist

Summary and Conclusion

In this project, I crafted a comprehensive music library. Initially, a SQL database was established to store data pertaining to albums, artists, songs, and playlists. The database underwent normalization to the BCNF to enhance efficiency and eliminate redundancy, with a particular emphasis on preventing the introduction of spurious tuples.

Following this, I designed an interface using C# in Visual Studio 2022 tailored for administrative use. The interface incorporates a binding navigator for seamless navigation, presenting the name and ID of each object. Additionally, the binding navigator features essential functionalities such as add, update, and delete buttons, allowing the admin to efficiently manage database entries. A demo user interface was also integrated, providing the admin with a preview of the user experience. This demo interface enables the creation, updating, viewing, and deletion of playlists, alongside a search function for exploring songs and artists.

Throughout this project, I've gained valuable insights into various aspects of database design and software development. The experience of designing a well-structured database, implementing normalization techniques has been particularly insightful. Crafting a user interface in C# using Visual Studio by utilizing Toolbox objects and creating event handlers has deepened my understanding of the C# programming language. I've also honed skills in loading data into a binding navigator and connecting to a SQL database for effective data manipulation through SQL statements. I have learned immensely from this project as it provided hands-on experience in creating a functional and user-friendly music library system.

Future Work

To simplify the program, it was originally designed with a critical assumption that every artist, album, song have unique names. However, in reality, this assumption may not hold true. Uniqueness in these attributes is not a strict necessity. Therefore, for future development, I intend to enhance the code to accommodate scenarios where artists, albums, and songs can share names without conflicts.

Moreover, a more logical approach would involve assigning individual accounts to both artists and users. This allocation empowers artists to independently upload their songs and albums, manage playlists, and enables users to create playlists, search for songs/artists/albums, and oversee their music library. As a result, my plan is to enhance the program, shifting the administrator's role to primarily manage both artists' and users' accounts. This modification ensures that artists retain responsibility for creating, updating, and deleting content within their accounts, while users manage their playlists with the capability to create, update, and delete playlists within their accounts.

References

- Retrieving the duration of a wmv in C#: https://j.hn/retreiving-the-duration-of-a-wmv-in-c/
- C# Tutorial 37: How to add a (Windows Media Player) Video clip to the form: https://www.youtube.com/watch?v=K7l1OGHNfeE
- Insert Update Delete Using DataGridView and Binding Navigator: https://www.youtube.com/watch?v=CfHjuThEdP0
- C# Tutorial - How to use BindingSource and BindingNavigator | FoxLearn: https://www.youtube.com/watch?v=C-pEXZFk6FU

- How to use BindingSource and BindingNavigator: https://www.youtube.com/watch?v=AVZPTzkQOcY&t=491s
- C# Tutorial - Display Images in DataGridView | FoxLearn: https://www.youtube.com/watch?v=Bb1VIMCq8SA
- Creating a File Upload Control in Visual Studio : https://www.youtube.com/watch?v=roHLK-46FC8
- Crop Image or Insert text into an image in C# : https://www.youtube.com/watch?v=pDmlHJU0hXw
- Database transactions with C# and SQL Server - Part 1 : https://www.youtube.com/watch?v=L_2xu8TgEiM

Code Listing

[Please find attached:

DigitalMusicLibraryTK.mdf

DigitalMusicLibraryTK_log.ldf

TK Music Library.sln]