

MUSICLY

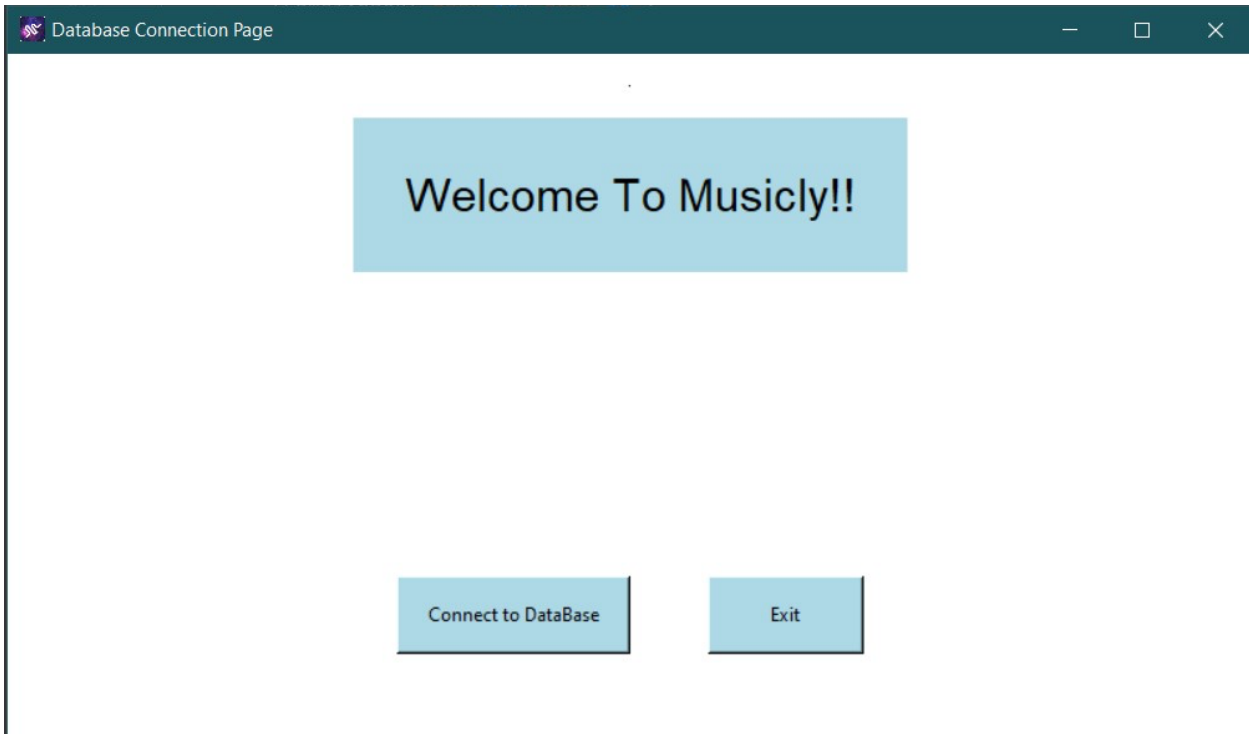
“Musicly” is one stop for the user to get all the information about a song like Song’s Release date, Lyrics, Artists involved, Album/Movie, Genre, Language, Reviews, Ratings. In this application, a user can perform **create, read, update, and delete** operations based on the options selected. Musicly can be inferred as IMDB for the Music Industry in which user can makes changes after creating an account. User will also be able to search their top song preferences based on their mood, genre, language favorite artist and favorite song.

1. README:

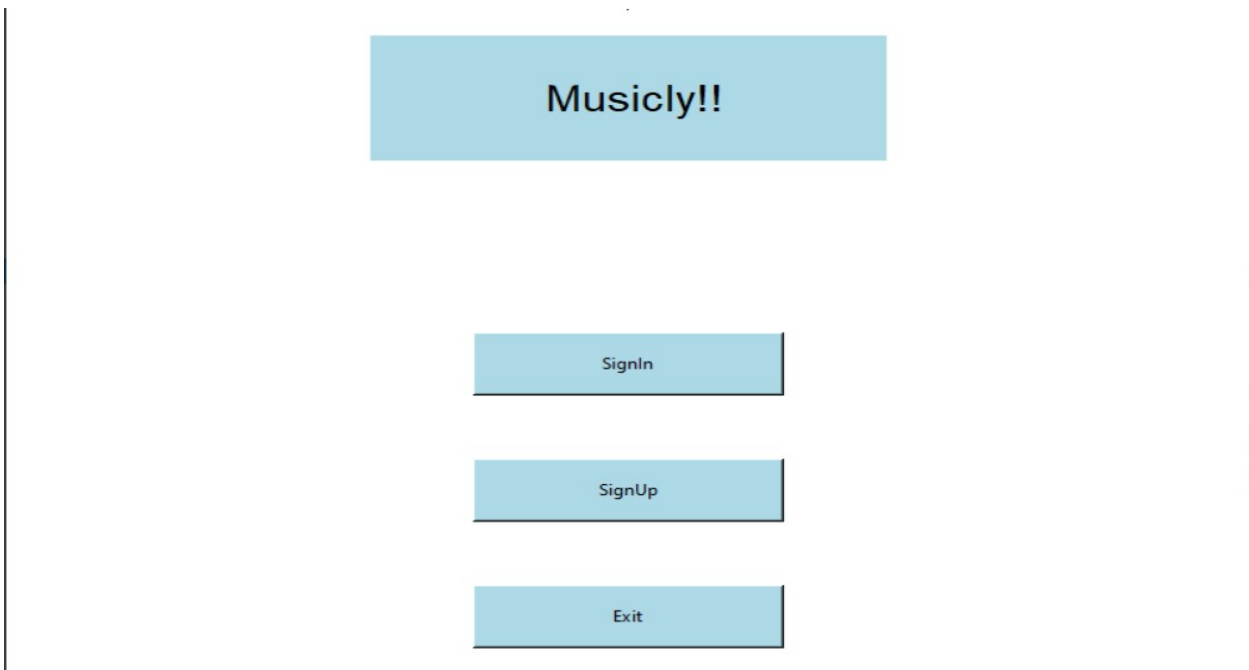
Below are the steps to run our program:

- A. To run our project (“Musicly”) you first need to install PyCharm on your machine. Below is the link for the same.

link: <https://www.jetbrains.com/pycharm/download/#section=windows>
- B. To run the project, you need to have MYSQL Workbench in your system to create the database and perform all the commands, open Terminal and run the following command to import libraries pymysql & tkinter in your PyCharm.
- C. Once you are done with the installation part, please run the data export SQL File “” in your MySQL workbench to create the Database, Tables, Functions, Procedures, Triggers and to dump the data into the tables to run the project.
- D. Once the above script is executed in MySQL workbench and database is created. Please run the Python script in your PyCharm. In order to connect to the created “Musicly” database in MySQL workbench you need to enter your local username and password.
- E. Once, you press the enter a GUI will appear with **“Connect to Database”** and **“Exit”** options. To proceed further please click on **“Connect to Database”**



- F. Once you click on “**Connect to Database**” a new window will appear with SignIn, SignUp and Exit option. As you are new user, please click on “**SignUp**” button to create your account or if have already created your account please click on “**SignIn**” button or if you want to exit, please click on “**Exit**”



- G. If you clicked on “**SignUp**” button, then below window will appear and please enter the mentioned fields and click on “**Next**”. This will **create** a new record for the user in the users table. And this will implement the **CREATE** part of the **CRUD operation**.

User Info

Musicly!!

Name -

Member - Select Status Option Selected

Password -

DOB -

Phone Num -

Email Id -

Address -

City -

State -

Postal Code -

- H. If you clicked on **“SignIn”** button, then you need to enter the existing username and password else it will throw an exception as follows:

Musicly!!

Invalid Input

Invalid Username or Password!!

UserName -

Password -

- I. Once you click on the **“Next”** button a below window will appear with all the options that can perform CRUD operation based on selection.



- J. If you click on **"Song Info"** button, below window will appear and you can select the song from the drop to for which you want to fetch or **READ** the information from the database. After selecting the song from the drop down please click on **"Click Me"** button and then click on **"Next"** to display the retrieved information or **"EXIT"** or can go to **"HOME"**. Similarly, if you want to fetch the details for Artist and User, you can click on **"Artist Info"** and **"User Info"** button respectively and proceed in the similar manner. This will implement the **READ** part of the **CRUD** operation.

For Song

Info:

Musicly!!

Song Name -

Select Song

click Me

Next

Exit

Home

For Artist Info:

Musicly!!

Artist Name -

Alan Walker

click Me

Alan Walker

Display

Exit

Home

For User Info:

Musicly!!

User Name: Umang

Membership Status: Yes

User DOB: 2022-12-01

Phone Num: 6176176171

Fav Song: {Ratta Maar}

Fav Artist: {Alka Yagnik}

Exit Home

- K. Once you click “Next” button, below window will appear that will show the data retrieved from the database for the selected song. And similarly for the User Info and Artist Info.

Musicly!!

Song Name: {All of Me}

Song Release Date: 2013-11-14

Song Lyrics: {} Exit

Song Movie/Album: {Love in the Future}

Song Artist Name: {John Legend} Home

Song Language: English


Song Genre: Hip-Hop

Song Mood: Happy

Song Review: Amazing

Song Rating: 10

- L. After reviewing the information, you can click on **“Home”** to go to the main options window, or you can click on **“EXIT”**.
- M. If you select the **“Modify Song Review”** button from the main options, then the below window will appear and then please select the Song from the drop down on which you want to **Update** Review or want to **Add** a new review. Similarly, you can **Add** or **Update** the review for an Artist by clicking on **“Modify Artist Review”** button. This will perform the **Update** part of the **CRUD operation**.

 Modify Song Review

Musicly!!

Song Name -

All of Me

click Me

All of Me

Review -

Awesome

Add Review

Delete Review

Exit

Home

- N. If you select the **“Modify Song Rating”** button from the main options and you want to delete the rating for a song, a below window will appear and you can select the song from the drop down and click on **“Click Me”** button and then click on **“Delete Rating”** button. This will delete the rating for the selected song. Similarly, you can delete the rating for an Artist by clicking on **“Modify Artist Rating”**. This will perform the **DELETE** part of the **CRUD operation**.

Musicly!!

Song Name -

All of Me

click Me

All of Me

Review -

Awesome

Add Review

Delete Review

Exit

Home

O. If you click on “Search Songs” button, you will get the below window and you can choose to click on any button to fetch the list of songs for the selected option.

Musicly!!

Artist

Genre

Language

Mood

Exit

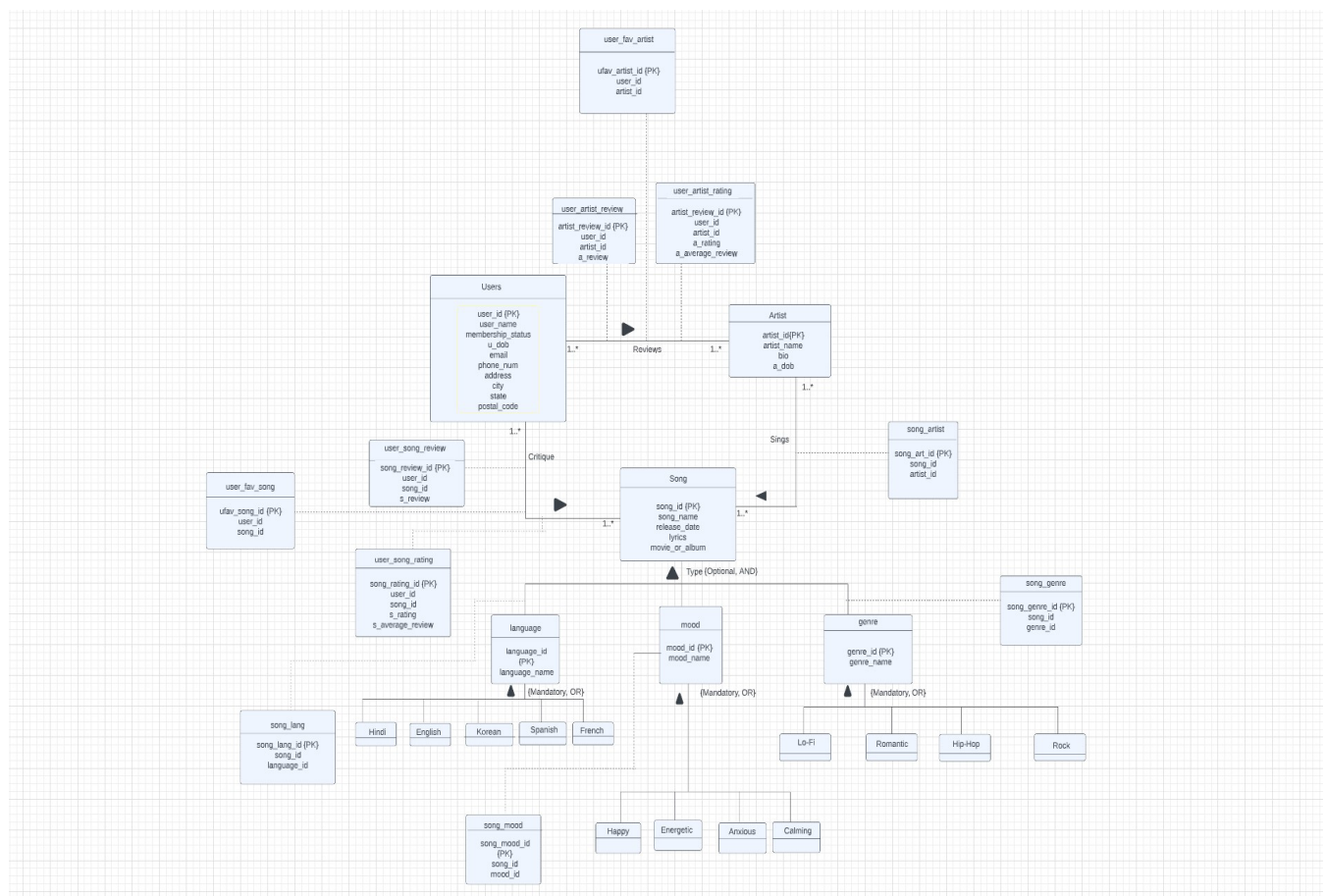
Home

Once you're done performing the actions on the application you can click on **"EXIT"** button to close the application. And changes made in the database will persist and will reflect when you reopen the application again.

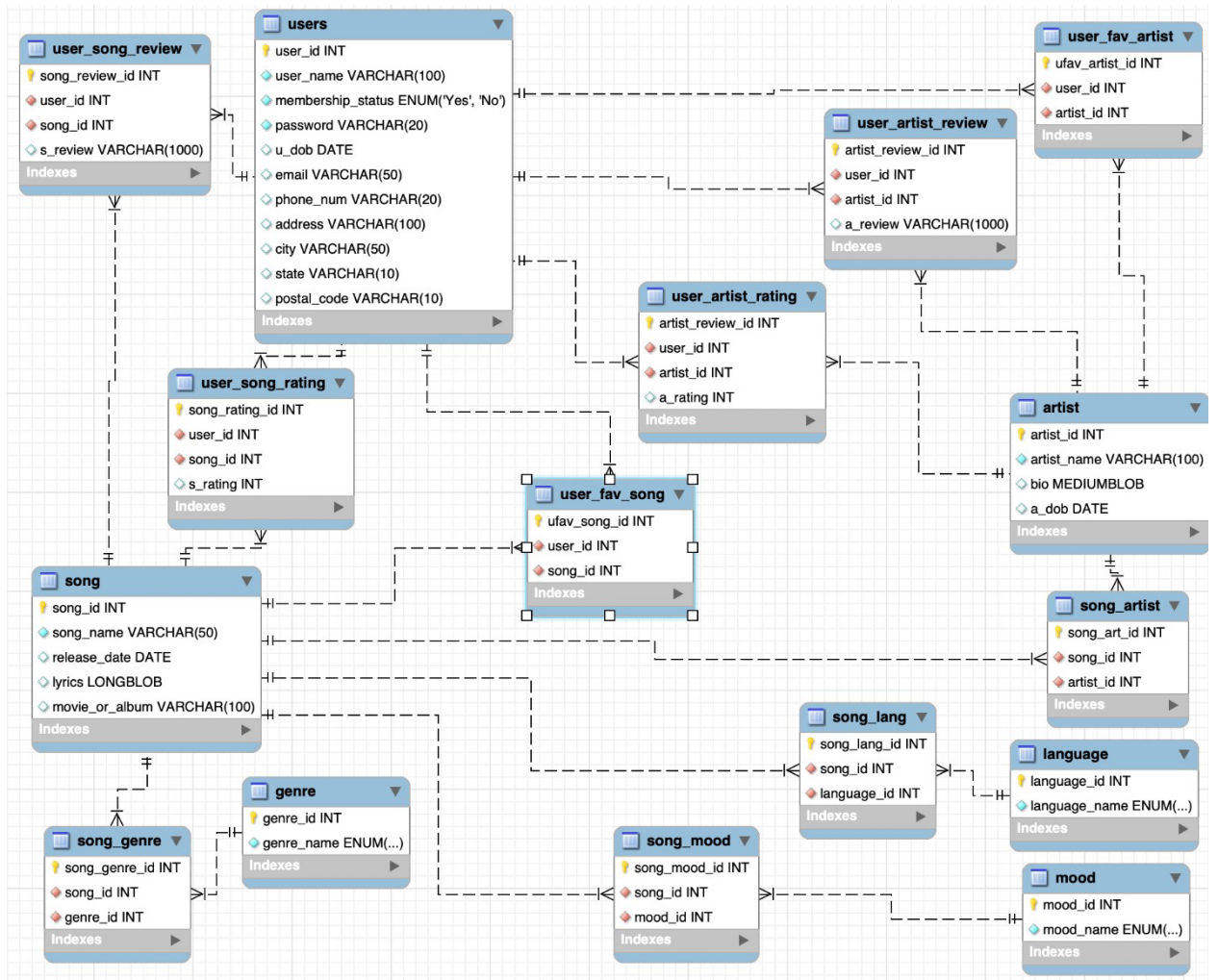
2. Technical Specifications for the project:

We have created database using MySQL workbench which will be able to **create, read, update, and delete** operations on specific data based on their access. We have created **GUI in Python** and used **Tkinter framework** for creating a user interactive interface which allows them to perform all the **CRUD operations**.

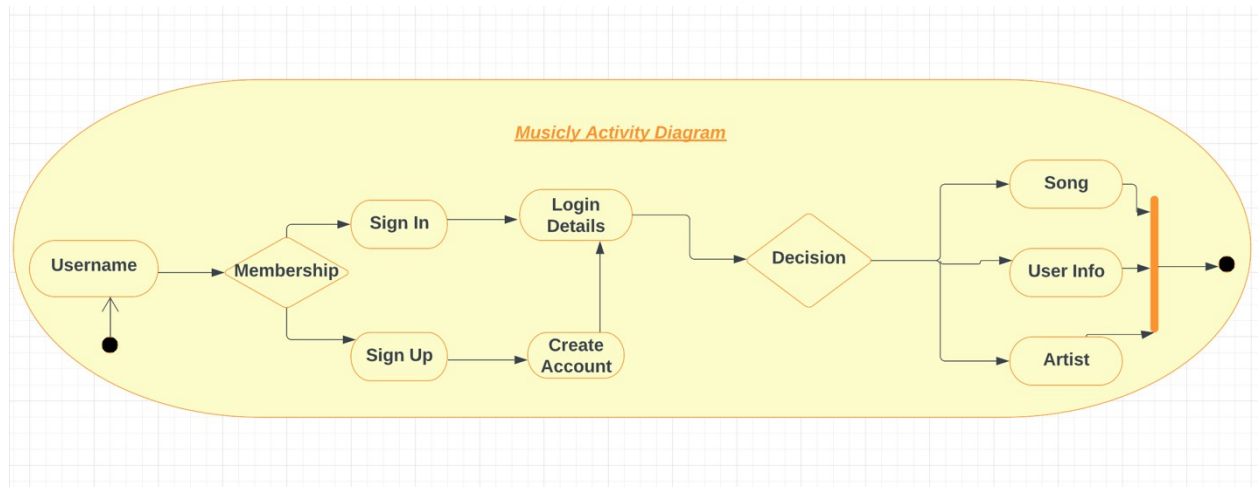
3. UML



4. Reverse Engineer your final schema



5. User flow of the system:



For the flow of the project, please refer point 1(README).

6. Lessons Learned

1. Technical expertise gained:

Whilst doing this project we learned more about Python and its framework for creating the Graphical User Interface and learned more about MySQL procedures, exception handling, triggers, and various other aspects of Database in MySQL to keep the data synchronized and normalized.

2. Insights, time management insights, data domain insights etc.:

- Week1: We decoded the flow of the code using class diagram and decided various new functionalities that could be added.
- Week 2: We worked on creation of database schema and other related aspects. Along with this we focused on creating interesting Sql create queries and complicated schema.
- Week 3: We started working on the front-end part of the code using Web Development concepts in Python using Flask framework.
- Week 4: We switched our approach to Python for Graphical Using Interface using Tkinter Framework.
- Week 6: Once the functionalities of our GUI were finalized, in week 6 we created procedure, functions along with that dealt all the exceptions.
- Week 7: We loaded data into the tables and tested our Project.

3. Realized or contemplated alternative design / approaches to the project

Initially, we started working on the front-end part of the code using Web Development concepts in Python using Flask framework, but we were not able to implement it successfully due to lack of knowledge about the Flask framework. Then switched our approach to Python for front end Graphical User Interface using Tkinter Framework.

4. Document any code not working in this section

As of now we don't have any component that is not functional.

7. Future work

1. Planned uses of the database:

This database is very comprehensive to track the user activity on "Musicly". Along with this we can extend the database and provide various payment options to the users and restrict their flow based on their subscription.

2. Potential areas for added functionality: We can add payment functionality to the "Musicly" system, along with this we can add a functionality to play the music as well. As of now our aim was to create a one stop place for the user to know all the information about their favorite artist and songs and implement all the CRUD operations but we can further extend this project to play songs as per their choice of artist, genre, mood, language.