

Programming Assignment 2 -- Tom Gallagher

Music Listening Database

For this assignment I will be creating a music database of my music listening history. The two ways which I will source this data will be through API's and through manual uploads. Data from my main sources of listening (Spotify and iTunes), will be gathered using API's. Manual uploads will be used to input data from lesser used sources (YouTube, Sound Cloud, Radio, Live Concerts, Physical Music (CD's/Records)). The program will be interacted with through a GUI. The python data will then be written into a mysql* database.

**I can use a different database if there is a preferred one mentioned in the next lecture.*

The user must first make an account with a unique User Name, their Spotify user ID and their iTunes user ID which will be used to access the Spotify and iTunes user ID to access their respective accounts. A user class will be written into python and then added to the mysql database.

When logging-in the program will access the database through python and check if there is a matching User with a matching password, if so the users' homepage will be opened.

The program will then access those accounts (if available) on every log-in and retrieve the data. Only data from after the user has created an account will be available to the program.

Information on the songs, artists and albums will be taken from the Spotify API. However the user will have to option to edit details or add artists through the GUI. This will help deal with the problem of individual songs/artists appearing more than once because of spelling mistakes.

When updating smaller information the updates will occur immediately when the user presses the update button in the update page. When adding songs/playlists the user will have the option to undo all changes, updates will only occur when they have pressed the add button and then closed the page.

When adding data the users' actions and inputs will directly result in the creation of Python classes. The data from the Python Classes will then be automatically written into the database.

Upon closing the homepage the program will end, the API's will stop retrieving data from the users' accounts

More information on the Python Classes and basic GUI Design Templates are on the second page.

The following are the key Python Classes of the program.

Class	User Interactions
User: {User_Name, Firstname, Surname, Spotify, iTunes, Playlist}	Created when the User first creates an account Add Account: Adds a Spotify or iTunes account if they create one later
Plays: {Play_ID (count from 1 up), Song Title, Artist, Album/EP, Source, Date, Time, Session, User_ID} (If no date or time available, the date and time added will be used)	(Playlist in GUI) The user can add 1 or more songs, to the overall plays class by filling in all parameters (except Play ID) Time for all songs is the start time of the session.
Song: {Song_ID, Song Title, Length, Artist, Album/EP, Plays, Genre}	Stats: Show most played song(s) over a specified time period Update: Add songs that are not on Spotify/Append false details
Artist: {Artist_ID, Artist Name, Country, Albums, Eps, Plays}	Stats: Show most played artist(s) over a specified time period Update: Add artists that are not on Spotify/Append false details
Album: {Album_ID, Album Title, Artist, Tracks, Plays}	Stats: Show most played artist(s) over a specified time period Update: Add albums that are not on Spotify/Append false details
Source: {Name, Play ID}	Stats: Show plays, total time, most played song of a selected source. Update:
Session: {Session Type, Date, Start, End, Play ID}	Stats: Show plays, total time, most played song of a selected session. Update: All the songs played within the Start/End times are added to the specified session type

Login Page

Name	Full Name
User	Username
P_Word	Password

Login

Create New User

New User Page

Name	Full Name
User	Username
P_Word	Password
Spotify	API Details
Itunes	API Details

Create New User

Song Stats

Your most listened to song between 12/12/2021 and 1/1/2022 was

A song by an Artist

Total Listens: 56

Update Artist

Update existing artist /add new artist

Name	Artist
Country	Australia
Albums	Album List
EPs	EP List

Update

Music Listening Database

Tom Gallagher
Total Songs: 789

Total Time: 1:23:34(H:M:S)

Songs	stats	update
Artists	stats	update
Album	stats	update
Source	stats	update
Session	stats	update

Add Playlist

Add Playlist

Number of Songs 12

Song	Song1
Artist	Band
Album/EP	Album
Source	Spotify
Date	26/03/1998
Time	4:55pm
Session	Study

Add Undo