CRITERION A

---

1. Defining the Problem

My client, Angella Hao, is an avid music lover who primarily enjoys music through the software, Spotify, in its desktop version. During my initial conversation with her, I learned that she was planning on reorganizing her library of songs. In describing this, she said that was unsure where to start reorganizing since she had a large number of songs in her library, and she wasn't sure what criteria she was going to use to remake her playlists by yet. She also mentioned that tracking duplicate songs that her listened-to artists released in multiple collections was "annoying," which contributed to the reorganization process becoming "overwhelming."

I proposed creating a database program to help the reorganization process. Such a program might not only remedy her issues with identifying duplicates, but could also help her generate some playlists that could inspire her when creating playlists later in her reorganization process.

I followed up with an interview, which is documented in the **Appendix**, to better understand how Angella might envision such a program. She reiterated that she primarily wants to be able to get rid of duplicate songs, and she approved of including a playlist generation capability. In addition, she mentioned that it would be nice to have more interactive functions in the program, something inspired by the minigames seen daily on the Google.com homepage or Spotify's year-end interactive analytics. Angella also later said that she could provide a short list of songs she already knows will be present in her final organized library in a spreadsheet.

Following these consultations, I drafted a proposal describing this program's purpose, capabilities, and the tools to be used in its development. I then discussed this project with Mr. B, who agreed to be my advisor for this project, and also proposed it to my Computing Science teacher, who approved it.

2. Rationale Behind the Proposed Solution

The purpose of the program is to help kickstart and streamline the client's reorganization process. The program will enable my client to store any number of songs, identify duplicate songs, and generate song playlists. The program will also be able to interact with my client by guessing her favourite artist or genre.

Python and SQL will be accessed using the "sqlite3" library inside Python and used to develop this program.

These two tools were chosen because:

- Python and SQL are being taught in my computing science course
- I have previous experience in Python programming

- A well-known Python IDE that my client is also familiar with can be used to create a user-friendly interface that is easy to navigate, especially when the program also contains streams of questions for the user to use to navigate the program
- My advisor is also experienced in Python programming
- Python can extract and manipulate data from a CSV file and can run SQL queries
- SQL can be used to effectively manage the kind of data my client will be providing after some format conversions

3. Success Criteria

Criteria #	**Success Criteria Description**
1	Program displays a greeting and operational menu upon running
2	Program displays all songs in an easy-to-read list if user selects ‘view all songs’ option
3	Program asks for song info and adds songs if the ‘add song’ option is selected
4	Program can display the current information stored about a song and can ask the user to update any/all fields of song info while maintaining the information of any fields the user doesn’t want updated, and successfully updates the song in the database
5	Program displays all songs in an easy-to-read list, and deletes the correct song indicated by the user.
6	Program displays names (and artists) of songs and the correct number of potential duplicates in a readable list and correctly removes duplicates of the song selected by the user.
7	Program generates a playlist of proper length and displays the playlist in an easy-to-read list.
8	Program guesses user’s favourite artist/genre the correct number of times and responds to user’s input with the correct message depending on whether the guess was correct or not
9	Program outputs error message and redirects the user as needed if an invalid piece of data is inputted in a field
10	Program exits when the last option is selected
11	Program restarts automatically without user action

Word Count: 392