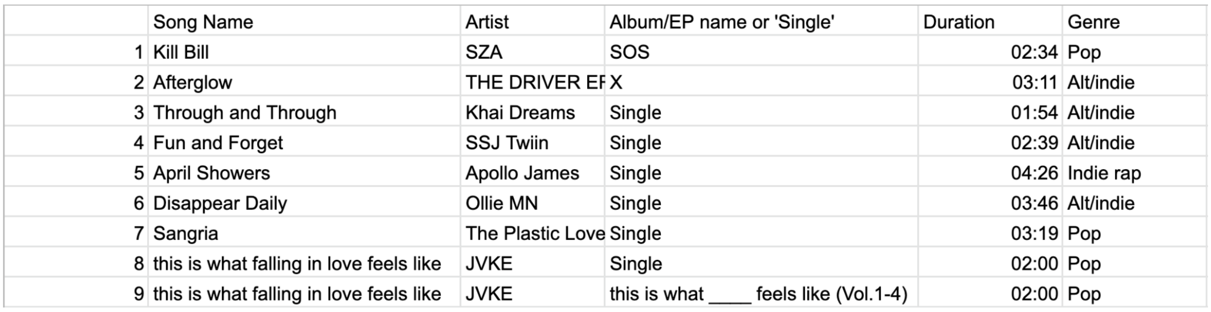
CRITERION B

Files Supplied by Client

The client provided a Google spreadsheet with information about 9 of her favourite songs. The spreadsheet was converted to a .CSV file to be imported into the database in the program.



	Song Name	Artist	Album/EP name or 'Single'	Duration	Genre
1	Kill Bill	SZA	SOS	02:34	Pop
2	Afterglow	THE DRIVER EF	X	03:11	Alt/indie
3	Through and Through	Khai Dreams	Single	01:54	Alt/indie
4	Fun and Forget	SSJ Twiin	Single	02:39	Alt/indie
5	April Showers	Apollo James	Single	04:26	Indie rap
6	Disappear Daily	Ollie MN	Single	03:46	Alt/indie
7	Sangria	The Plastic Love	Single	03:19	Pop
8	this is what falling in love feels like	JVKE	Single	02:00	Pop
9	this is what falling in love feels like	JVKE	this is what ____ feels like (Vol.1-4)	02:00	Pop

Figure 1: Spreadsheet as supplied by the client

Flowchart

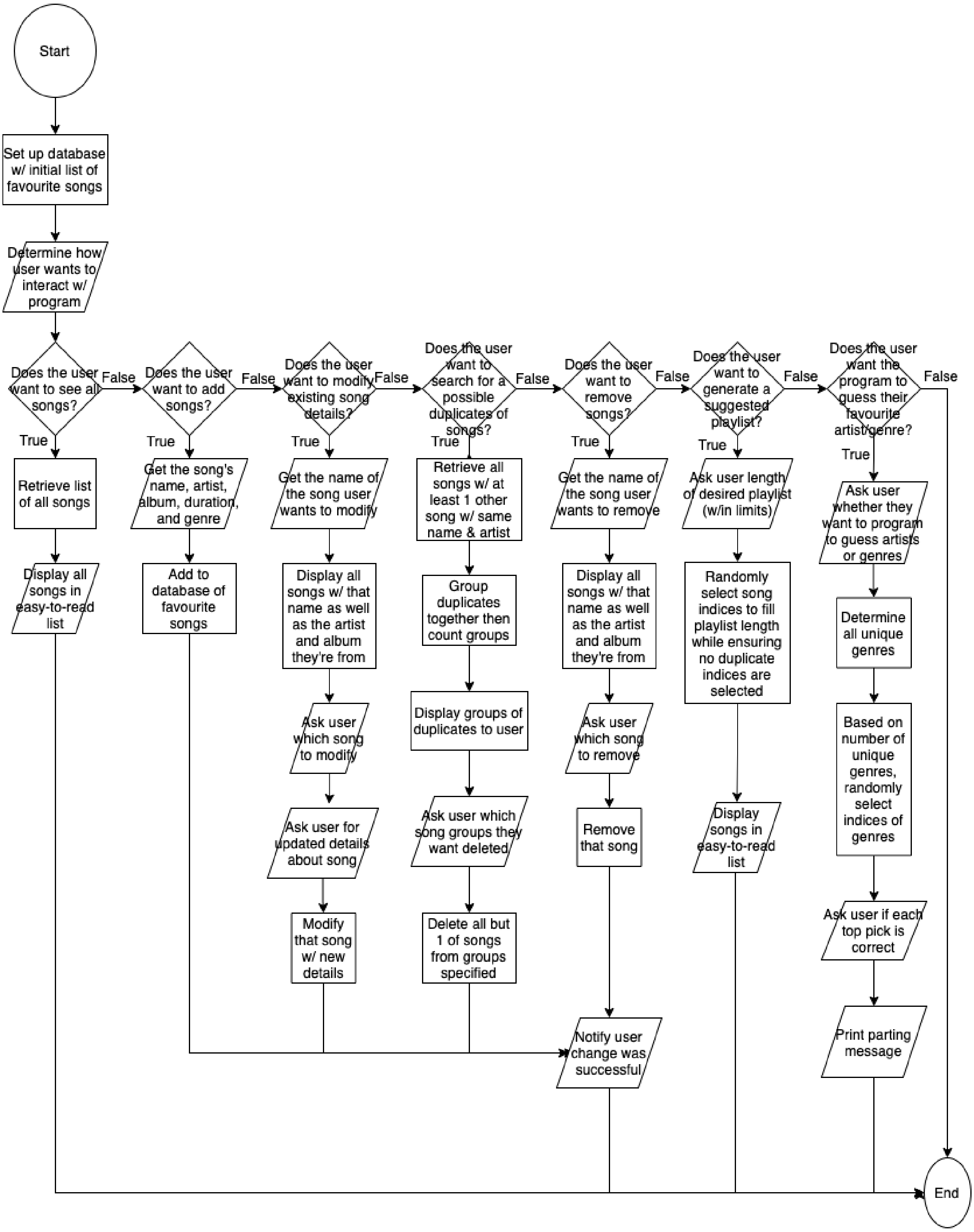
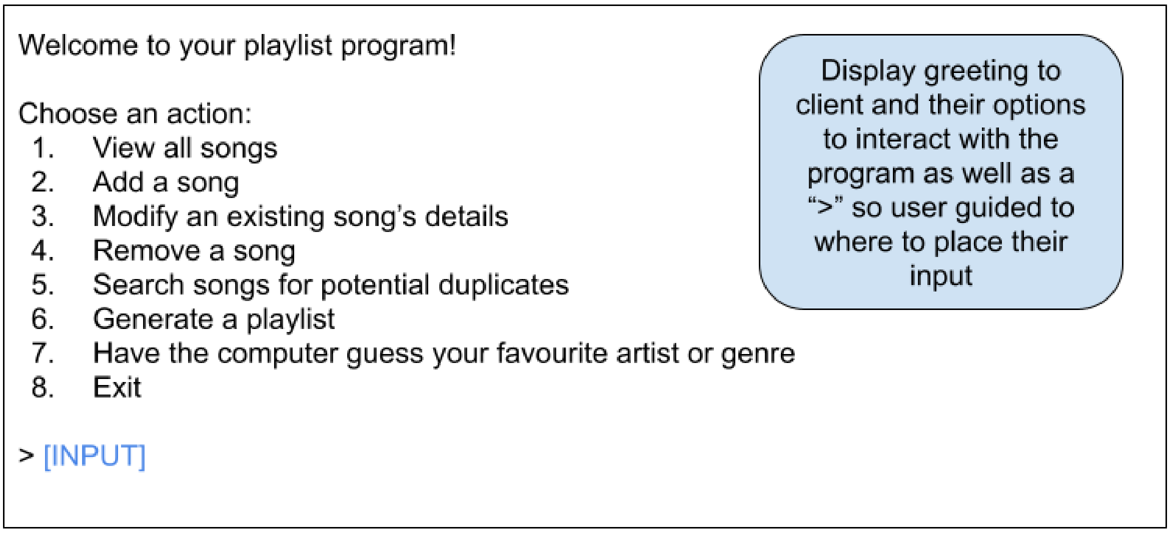

Start
Set up database w/ initial list of favourite songs
Determine how user wants to interact w/ program
Does the user want to see all songs?
False
True
Retrieve list of all songs
Display all songs in easy-to-read list
Does the user want to add songs?
False
True
Get the song's name, artist, album, duration, and genre
Add to database of favourite songs
Does the user want to modify existing song details?
False
True
Get the name of the song user wants to modify
Display all songs w/ that name as well as the artist and album they're from
Ask user which song to modify
Ask user for updated details about song
Modify that song w/ new details
Does the user want to search for a possible duplicates of songs?
False
True
Retrieve all songs w/ at least 1 other song w/ same name & artist
Group duplicates together then count groups
Display groups of duplicates to user
Ask user which song groups they want deleted
Delete all but 1 of songs from groups specified
Does the user want to remove songs?
False
True
Get the name of the song user wants to remove
Display all songs w/ that name as well as the artist and album they're from
Ask user which song to remove
Remove that song
Notify user change was successful
Does the user want to generate a suggested playlist?
False
True
Ask user length of desired playlist (w/in limits)
Randomly select song indices to fill playlist length while ensuring no duplicate indices are selected
Display songs in easy-to-read list
Does the user want the program to guess their favourite artist/genre?
False
True
Ask user whether they want to program to guess artists or genres
Determine all unique genres
Based on number of unique genres, randomly select indices of genres
Ask user if each top pick is correct
Print parting message
End

Figure 2: Flowchart outlining the main algorithms of the program

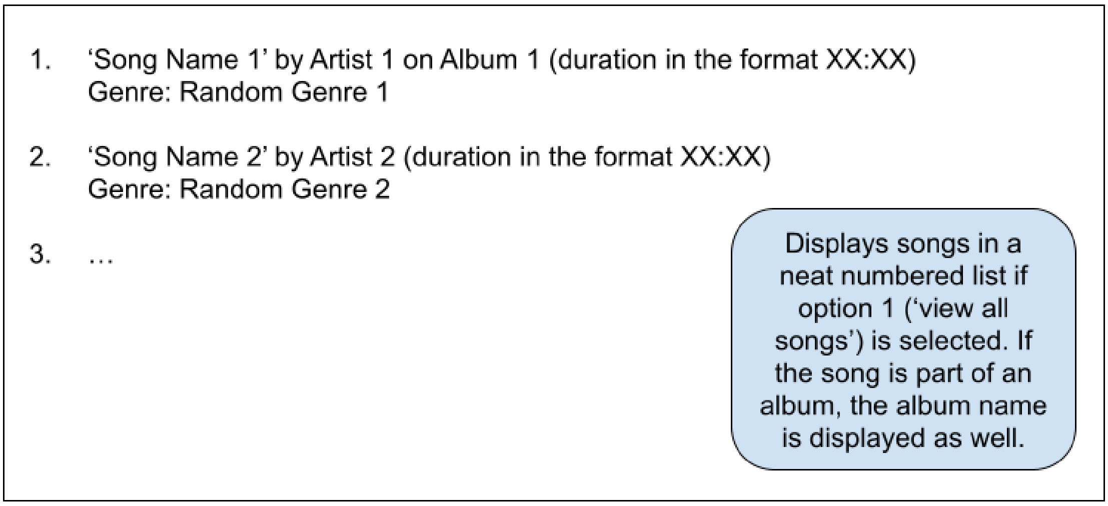
User-Interface Overview



```
Welcome to your playlist program!

Choose an action:
 1.    View all songs
 2.    Add a song
 3.    Modify an existing song’s details
 4.    Remove a song
 5.    Search songs for potential duplicates
 6.    Generate a playlist
 7.    Have the computer guess your favourite artist or genre
 8.    Exit

> [INPUT]
```

Display greeting to client and their options to interact with the program as well as a “>” so user guided to where to place their input

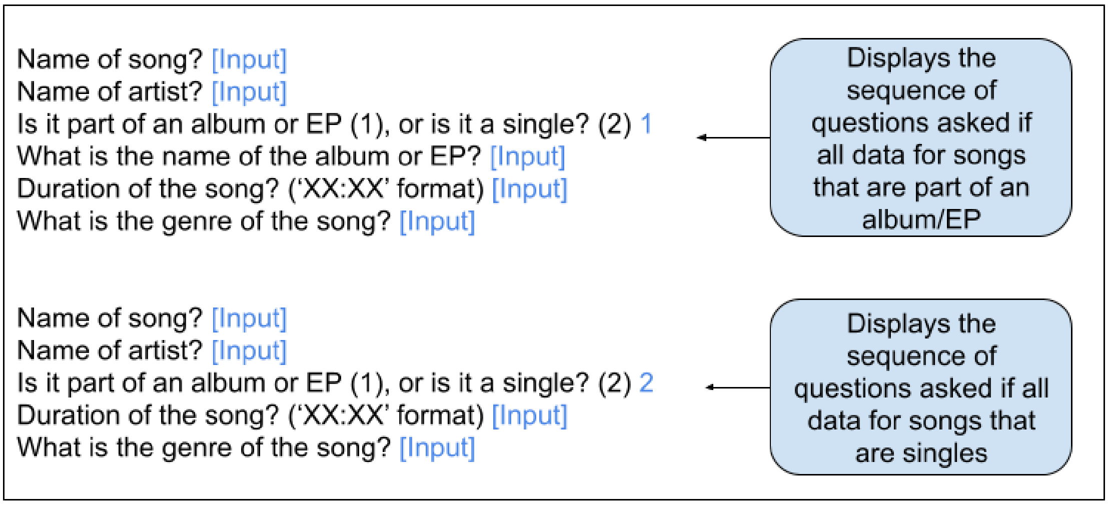
Figure 3: Main menu displayed to user when first running and entering the program



```
1.    ‘Song Name 1’ by Artist 1 on Album 1 (duration in the format XX:XX)
      Genre: Random Genre 1

2.    ‘Song Name 2’ by Artist 2 (duration in the format XX:XX)
      Genre: Random Genre 2

3.    …
```

Displays songs in a neat numbered list if option 1 (‘view all songs’) is selected. If the song is part of an album, the album name is displayed as well.

Figure 4: Song list displayed if option 1 is selected in main menu



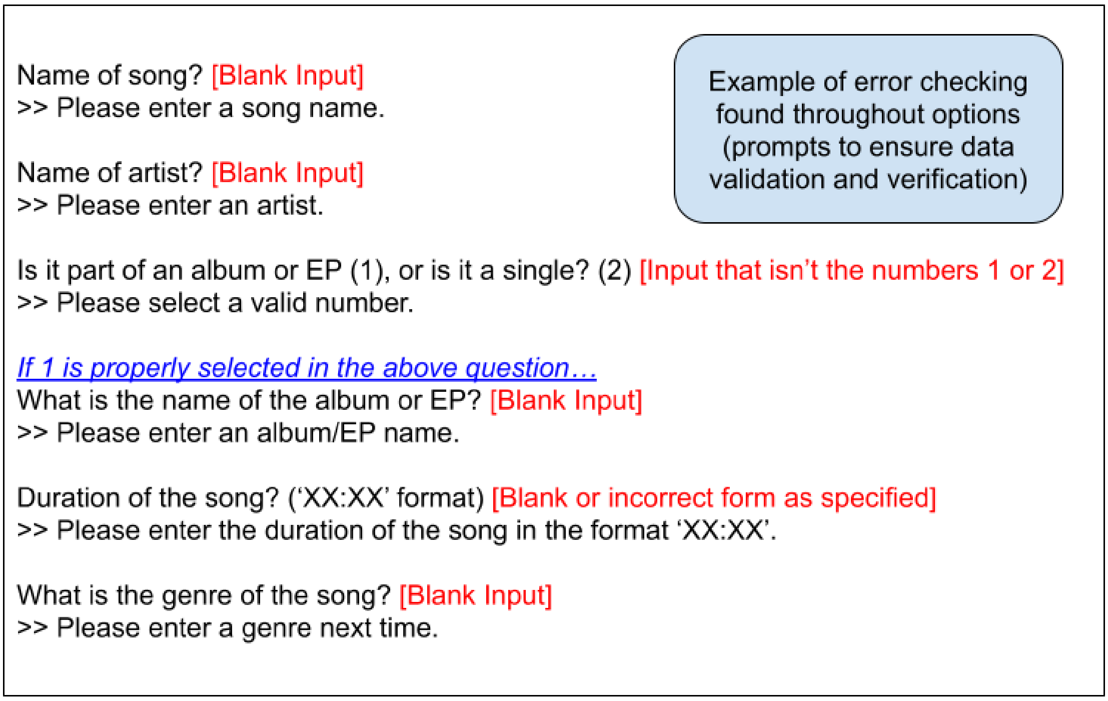
```
Name of song? [Input]
Name of artist? [Input]
Is it part of an album or EP (1), or is it a single? (2) 1
What is the name of the album or EP? [Input]
Duration of the song? ('XX:XX' format) [Input]
What is the genre of the song? [Input]
```

Displays the sequence of questions asked if all data for songs that are part of an album/EP

```
Name of song? [Input]
Name of artist? [Input]
Is it part of an album or EP (1), or is it a single? (2) 2
Duration of the song? ('XX:XX' format) [Input]
What is the genre of the song? [Input]
```

Displays the sequence of questions asked if all data for songs that are singles

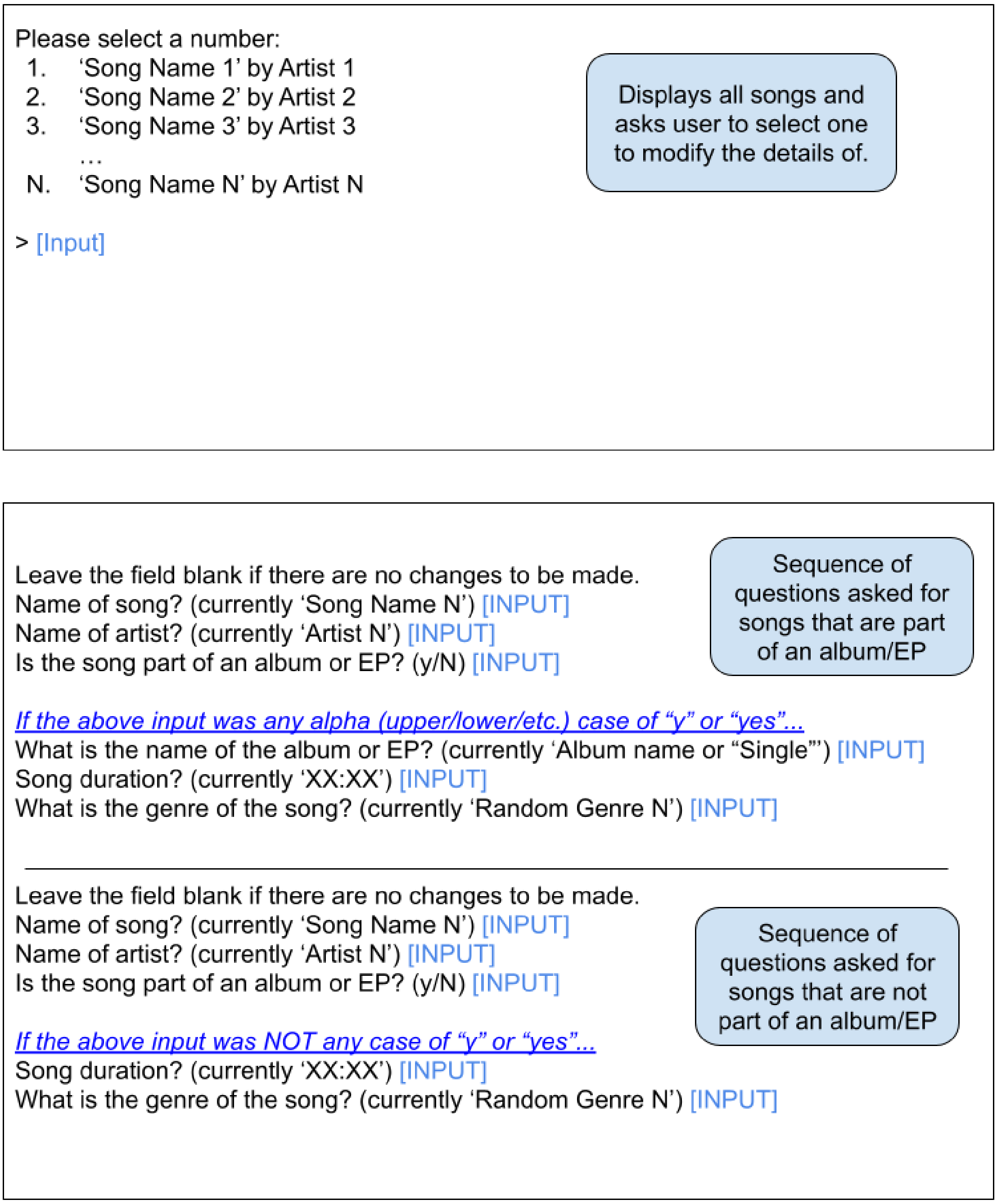
Figure 5: Sequence of questions asked to user as they enter accepted forms of data.



```
Name of song? [Blank Input]
>> Please enter a song name.

Name of artist? [Blank Input]
>> Please enter an artist.

Is it part of an album or EP (1), or is it a single? (2) [Input that isn't the numbers 1 or 2]
>> Please select a valid number.

If 1 is properly selected in the above question…
What is the name of the album or EP? [Blank Input]
>> Please enter an album/EP name.

Duration of the song? ('XX:XX' format) [Blank or incorrect form as specified]
>> Please enter the duration of the song in the format 'XX:XX'.

What is the genre of the song? [Blank Input]
>> Please enter a genre next time.
```

Example of error checking found throughout options (prompts to ensure data validation and verification)

Figure 6: Demonstration of style of error checking found throughout program but prompts specific to option 1



```
Please select a number:
 1.    ‘Song Name 1’ by Artist 1
 2.    ‘Song Name 2’ by Artist 2
 3.    ‘Song Name 3’ by Artist 3
       …
 N.    ‘Song Name N’ by Artist N

> [Input]
```

Displays all songs and asks user to select one to modify the details of.

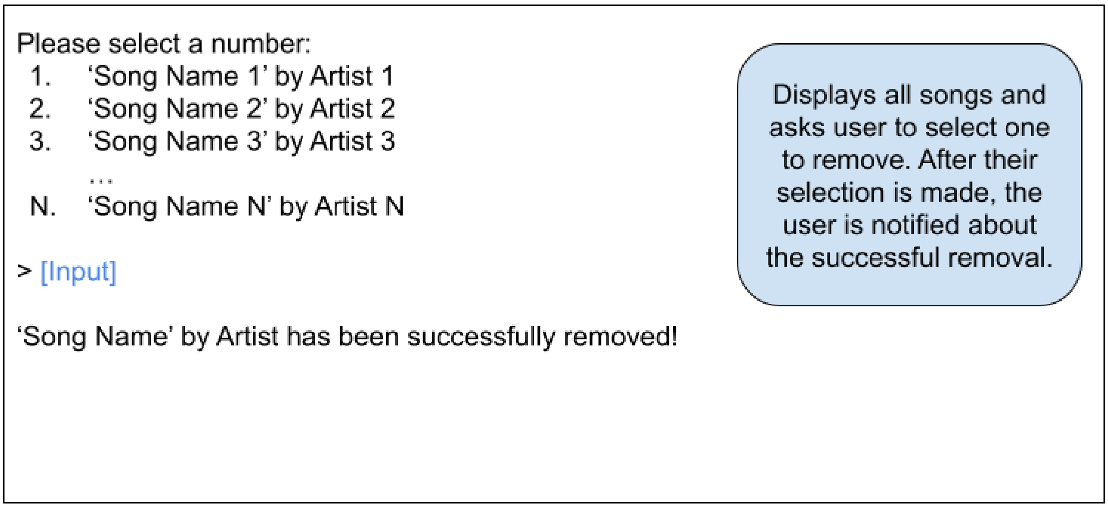
```
Leave the field blank if there are no changes to be made.
Name of song? (currently ‘Song Name N’) [INPUT]
Name of artist? (currently ‘Artist N’) [INPUT]
Is the song part of an album or EP? (y/N) [INPUT]

If the above input was any alpha (upper/lower/etc.) case of “y” or “yes”...
What is the name of the album or EP? (currently ‘Album name or “Single”’) [INPUT]
Song duration? (currently ‘XX:XX’) [INPUT]
What is the genre of the song? (currently ‘Random Genre N’) [INPUT]
```

Sequence of questions asked for songs that are part of an album/EP

```
Leave the field blank if there are no changes to be made.
Name of song? (currently ‘Song Name N’) [INPUT]
Name of artist? (currently ‘Artist N’) [INPUT]
Is the song part of an album or EP? (y/N) [INPUT]

If the above input was NOT any case of “y” or “yes”...
Song duration? (currently ‘XX:XX’) [INPUT]
What is the genre of the song? (currently ‘Random Genre N’) [INPUT]
```

Sequence of questions asked for songs that are not part of an album/EP

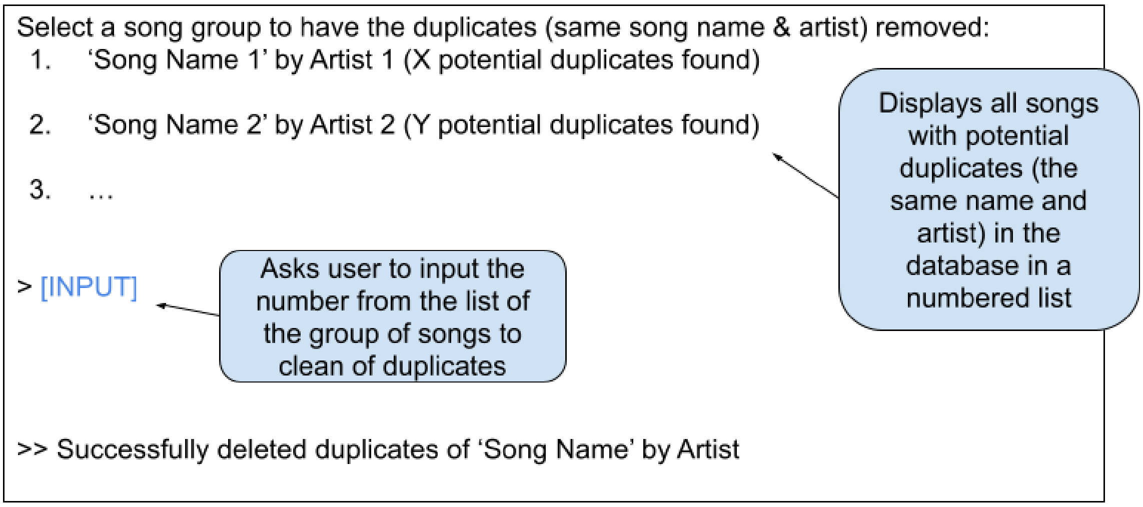
Figure 7: List of songs and sequence of questions user could be asked if option 3 selected from menu



```
Please select a number:
  1.   'Song Name 1' by Artist 1
  2.   'Song Name 2' by Artist 2
  3.   'Song Name 3' by Artist 3
       …
  N.   'Song Name N' by Artist N

> [Input]

'Song Name' by Artist has been successfully removed!
```

Displays all songs and asks user to select one to remove. After their selection is made, the user is notified about the successful removal.

Figure 8: List of songs, prompt, and alert message user receives for menu option 4

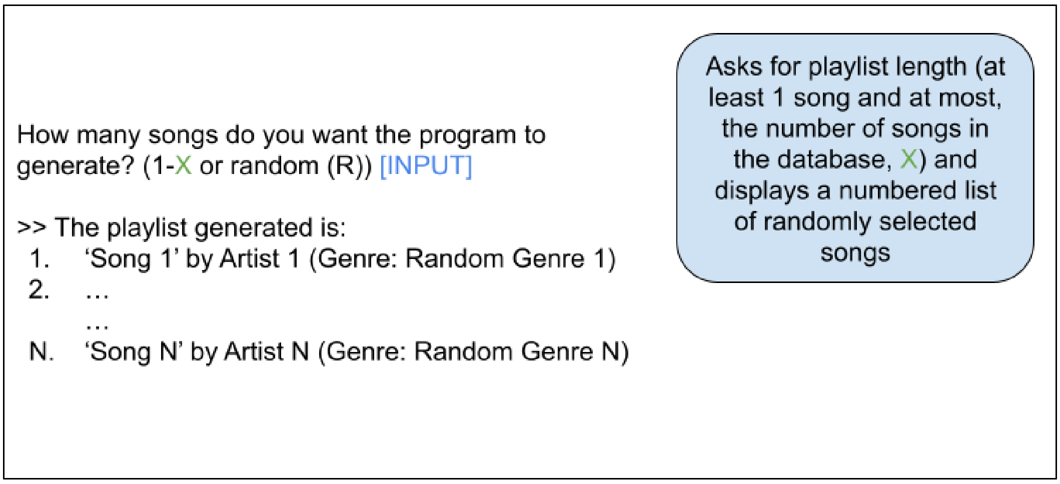


```
Select a song group to have the duplicates (same song name & artist) removed:
  1.   'Song Name 1' by Artist 1 (X potential duplicates found)

  2.   'Song Name 2' by Artist 2 (Y potential duplicates found)

  3.   …

> [INPUT]


>> Successfully deleted duplicates of 'Song Name' by Artist
```

Displays all songs with potential duplicates (the same name and artist) in the database in a numbered list

Asks user to input the number from the list of the group of songs to clean of duplicates

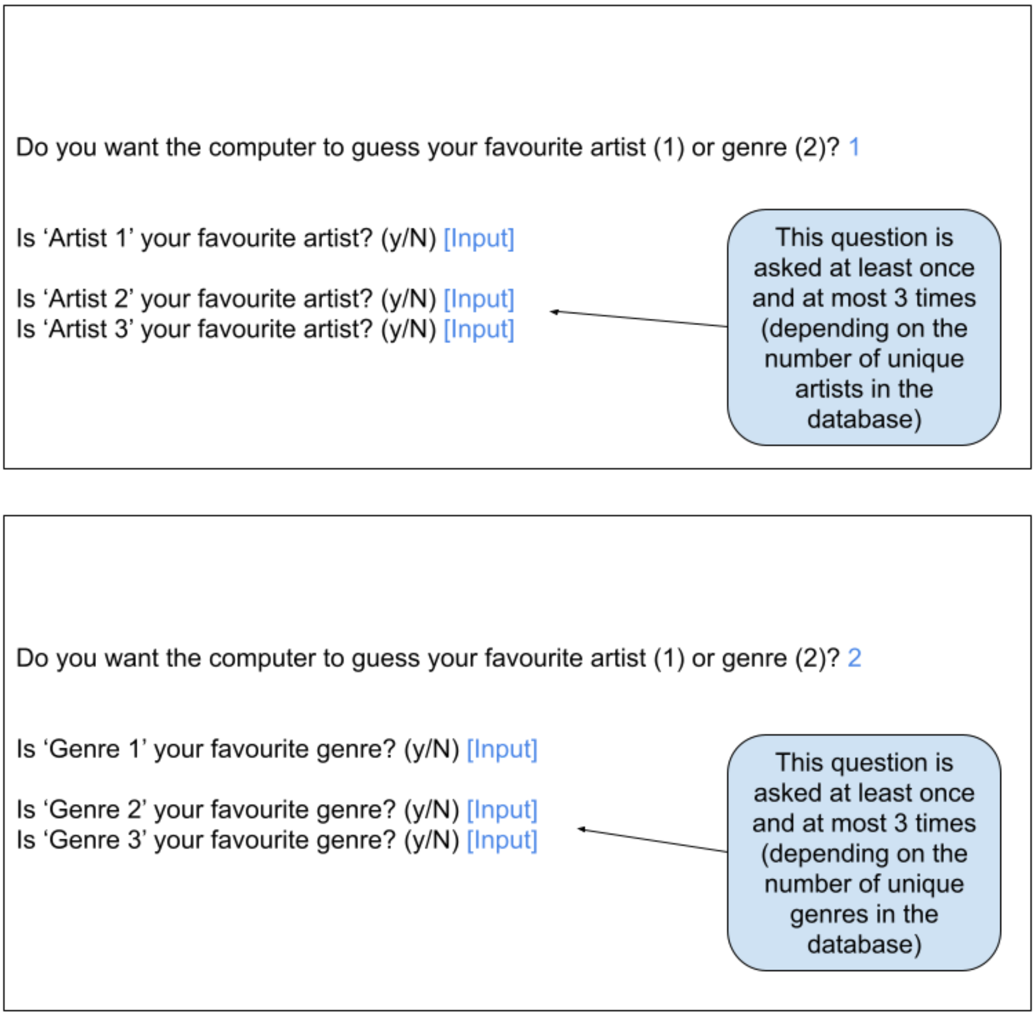
Figure 9: List of songs program identified as having duplicates in the database



```
How many songs do you want the program to
generate? (1-X or random (R)) [INPUT]

>> The playlist generated is:
 1.   'Song 1' by Artist 1 (Genre: Random Genre 1)
 2.   …
      …
 N.   'Song N' by Artist N (Genre: Random Genre N)
```

Asks for playlist length (at least 1 song and at most, the number of songs in the database, X) and displays a numbered list of randomly selected songs

Figure 10: Generating a playlist of randomly selected songs of a length indicated by user



```
Do you want the computer to guess your favourite artist (1) or genre (2)? 1

Is 'Artist 1' your favourite artist? (y/N) [Input]

Is 'Artist 2' your favourite artist? (y/N) [Input]
Is 'Artist 3' your favourite artist? (y/N) [Input]
```

This question is asked at least once and at most 3 times (depending on the number of unique artists in the database)

```
Do you want the computer to guess your favourite artist (1) or genre (2)? 2

Is 'Genre 1' your favourite genre? (y/N) [Input]

Is 'Genre 2' your favourite genre? (y/N) [Input]
Is 'Genre 3' your favourite genre? (y/N) [Input]
```

This question is asked at least once and at most 3 times (depending on the number of unique genres in the database)

Figure 11: Sequence of questions computer asks user as it tries to guess the user's favourite artist or genre

TEST PLAN

Menu Option	Action to be Tested	Test Method
	Program displays a greeting and working menu upon start	- Run program - Put an option not in the menu to see if it redirects back to the menu - Input the menu option and check if it is redirected to the correct option
1	Program displays songs in a numbered list	- Select ‘view all songs’ option in menu - See if all songs and all relevant data is properly displayed
2	Program can add a new song to database when given expected info	- Select ‘add song’ option in menu - Input three strings, then a string in the format ‘XX:XX’, and finally another string (representing song name, artist, collection name, duration, genre) - See if program can successfully insert the correct new id (next largest unique in id column) and successfully add new song data - See if program notifies user that the action was successful with a printed message
2	Program can catch errors in data inputs by user and redirects user to correct them	- Select ‘add song’ option in menu - Purposefully enter blank strings into song name, artist, collection type (album/EP or single), and collection name fields to see if program redirects user to fix input data errors
3	Program can modify an existing song’s details in the database when given expected info	- Select ‘modify song’ option in menu - Select a song to test - Input three strings, then a string in the format ‘XX:XX’, and finally another string (representing song name, artist, collection name, duration, genre) - See if program can correctly update the correct song with new data - See if program notifies user that the action was successful with a printed message
3	Program can correctly keep details the user doesn’t want to modify the same in updating a song’s info	- Select ‘modify song’ option in menu - Purposefully enter blank strings into song name, artist, collection type (album/EP or single), and collection name fields - See if any changes are made to song details
3	Program can catch data input errors and redirects	- Select ‘add song’ option in menu - Purposefully enter a blank string into the

	user to correct them	collection field of input (“Is it part of an album/EP (1) or a single (2)?”) to see if program redirects user to fix input data errors - On another run of this option, input ‘1’ to indicate a song is part of an album/EP then enter a blank string to see if program catches error
4	Program correctly removes a song from the database	- Select ‘remove a song’ from the menu - See if all songs displayed to the user - See if correct song is removed - Rerun program and select same option to see if numbered list numbers update correctly after removal
5	Program retrieves and correctly displays groups of songs that have the same name and artist and correctly counts number of duplicates found	- Add 2 songs with the name “test” and artist “art” and one song with the name “test” and artist “a” to the database - Select ‘Search songs for potential duplicates’ option from the menu - Check if the test-duplicate songs (same name and same artist) are displayed in a numbered list and the correct number of duplicates are counted (2 not 3) - Check if ‘this is what falling in love feels like’ (a duplicate song given by client) is also in the displayed numbered list (2 duplicates should be counted) - Input the list number to remove the test-duplicate songs and see if program can process that request - Check if duplicates (but not all songs) are removed from database (only 1 song with that name and artist should remain)
5	Program correctly catches input errors in checking for duplicate songs	- Add 2 songs with the name “test” and artist “art” and one song with the name “test” and artist “a” to the database - Select ‘Search songs for potential duplicates’ option from the menu - Enter ‘0’ to see if program catches the input error from the user and prompts the user to fix it - Enter a number larger than the largest number in the list to see if program catches the input error from the user and prompts the user to fix it
6	Program randomly selects a number of songs to fill a playlist of user’s desired length without repeating songs and outputs in a	- Select ‘generate a playlist’ option from the menu - Enter a number within the given range of numbers stated by program for the playlist length

	numbered list	- See if a numbered list of proper length of all unique songs is outputted - Select ‘generate a playlist’ option from the menu again - Select ‘random (R)' as the playlist length - See if a numbered list of the random length is within the allowed range (stated by program when asking for playlist length) of all unique songs is outputted
6	Program catches input errors for playlist length	- Select ‘generate a playlist’ option from the menu - Enter ‘0’ to see if program catches the input error from the user (can’t generate a playlist of 0 songs) and prompts the user to fix it - Enter a number larger than the largest number in the list (the total number of songs in the database) to see if program catches the input error from the user (not enough unique songs to put in the playlist) and prompts the user to fix it
7	Program correctly retrieves 1-3 randomly selected artists from database depending on number of unique artists	- Select ‘have the computer guess your favourite artist or genre’ option from the menu - Select ‘1’ to indicate that computer should guess the user’s favourite artist - Fill in a blank string for the first guess of the user’s favourite artist, a “n” for the second guess, and an arbitrary digit (number) for the third guess to see if the program takes all as a “No,” which is indicated as its default - See if program outputs a message indicating it was unable to guess the user’s favourite artist afterwards -- - Reselect ‘have the computer guess your favourite artist or genre’ option from the menu - Select ‘1’ to indicate that computer should guess the user’s favourite artist - Enter a “y” for the program’s first guess of the user’s favourite artist to see if program does not guess any more and outputs a “successfully guessed” message to the user - Repeat the first 2 dashed actions but enter a “yes ” to check if the program can correctly remove whitespace and check the full word, ‘yes’, by not guessing any more and outputting a success message

7	Program correctly retrieves 1-3 randomly selected genres from database depending on number of unique genres	- Select ‘have the computer guess your favourite artist or genre’ option from the menu - Select ‘1’ to indicate that computer should guess the user’s favourite genre - Fill in a blank string for the first guess of the user’s favourite genre - See if program can keep guessing, then input a “n” for the second guess, and ten an arbitrary digit (number) for the third guess to see if the program takes all as a “No,” which is indicated as its default - See if program outputs a message indicating it was unable to guess the user’s favourite genre afterwards --- - Reselect ‘have the computer guess your favourite artist or genre’ option from the menu - Select ‘1’ to indicate that computer should guess the user’s favourite genre - Enter a “y” for the program’s first guess of the user’s favourite genre to see if program does not guess any more and outputs a “successfully guessed” message to the user - Repeat the first 2 dashed actions but enter a “yes ” to check if the program can correctly remove whitespace and check the full word, ‘yes’, by not guessing any more and outputting a success message
8	Program stops infinite loop (exits itself) after outputting a parting message to user	- Run program - Select ‘8. Exit’ from the menu - See if program outputs a farewell message and then fully exist the program’s infinite loop (automatically restarting)
	Program restarts without user manually having to rerun it	- After checking ever option in the menu, see if user is redirected to the main menu

Word Count: 39