1. Basic GUI

Simple GUI framework to allow me to easily add the later features.

At this stage, I will not worry about any colours, just layout.

2. Library management

Displaying the music stored in the collection is essential to allow the user to select music to listen to.

Other features, such as being able to sort the library based on different criteria are also important but not essential so could possibly be added later, although the main library system must be designed in a way that makes sorting possible.

Dependencies:

- Creating the basic GUI frame will be a part of this step

3. Play music

The most important part of my program is to load music files stored on the computer and play them.

This is essential and all other aspects of the program will be based off this.

Will need to set up basic GUI frame as part of this step – this will not need to be too complicated yet just basic two panes and show title of what is playing

Dependencies:

- Library management to allow a user to select a song

Rough test table:

Input	Explanation	Expected result
File Types		
Testfile.mp3	Standard lossy mp3 file	Plays track

Testfile.flac	Standard lossless flac file – larger file than mp3 so could take longer to load	Plays track
Testfile.wav	Standard lossless wav file	Plays track
AAC/ALAC/AIFF	Apple’s audio formats	Unlikely to be able to get these to work as they are controlled by apple
Testfile.mp4	Video file – could implement error message i.e. “That file is not supported” If possible, just play audio from the video?	Tells user that they cannot play that file
File names		
Test_file.flac	Underscores are common in titles in place of spaces or slashes – should just be counted as part of a string	Plays track
1testfile1.flac	Some tracks use numbers – should just be counted as part of a string	Plays track
Other		
Testfile.flac (longer song e.g. > 20 minutes long)	Some tracks in my library are very long (longer than 20 minutes). Should be run same as any other track but the length could cause unexpected issues	Plays track
Testfile.flac (shorter song e.g. < 30 seconds long)	Opposite of previos; should run perfectly	Plays track

Testing:

- Different scale of library e.g. with thousands of files
- Different types of files – show all types of music files but ignore any files that are not music
- Respond to subfolders

4. Currently playing information

Display information about the currently playing track, such as lyrics and credits.

This is the main goal of the program so is an essential feature to add.

Dependencies:

- Play music

Testing:

- Test a number of different tracks
    - Some from very small artists
- Ensure placeholders for when there is no information available

5. Similar artists/albums

Using information about the currently playing media to find recommendations of new music.

This will likely be much more complicated than the previous features, but it is a valued feature and is important for the program.

Dependencies:

- Play music

Testing:

- Test a number of different tracks
    - Some from very small artists
- Ensure placeholders for when there is no information available

6. Search for albums

Allow the user to search for albums based on name, artist, genre etc. to receive information about it. This is also more difficult but still important to people.

Dependencies:

- None other than main GUI

Testing:

- Test different artists, small in popularity to large
- Test different genres – some niche

7. If I have time

Some other features that I am unlikely to add, but could look at if I have time at the end include:

- Smart playlists (playlists that dynamically respond to listening activity) – would need me to build a system to track listening activity (e.g. a database)
- Radio – This is a low priority feature and would only be considered if everything else is completed fully