

C.R.I.N.G.E Music player v5

Caution **R**ick Is **N**ot **G**oing to **E**ngage

0 Title Page

0.1 - Title Page

Name of project, authors

Season/year

Illustration optional

C.R.I.N.G.E Music player v5

Nicholas Ayson

Ashley Thorlin

Jacob Re

Sean Labitigan

Wyatt Allen

Spring 2022 CPSC 362-09



1 Introduction

1.1 - Executive Summary

Creating a user friendly software that allows users to play music easily in any environment. The main purpose of this software is to play music from local files with many features such as creating playlists, favoriting a song, creating a queue, and many more. The goals of this software are:

- Search for songs the user wants
- Increase/Decrease volume
- Play songs from a playlist
- Create a queue of songs
- Store songs within a library on the user's device

1.2 - Terms and Acronyms - MP

- MP: Music Player
- C.R.I.N.G.E: Creative Reading Interface, Not Game Engine

1.3 - Requirements (Use cases and UML Class diagram should have all items tied back to a requirement)

- A way to access and upload music from the user's device
- A way to play and control music from the user's device
- A way to store song files within a library on the user's device
- A way for the user to view and search for songs within the library
- A way for the user to create playlists to organize songs within the library
- A way for users to view and edit the queue of songs

2 Design

2.1 - Use Cases (step-by-step instructions, start including exceptions, not all required yet)

Favorite Song - The user is able to like songs that are added to a library through a button, as well as unliking a song if the user changes their mind.

Control Music - Once the user selects a song the user has the option to play the song via a play button or pause the song via a pause button. The user will also have the option to skip the song via a button and a bar to skip forward or backward to a specific time in the song.

Control Volume - User is able to increase and decrease the volume of the application by using a slider on the control bar. The user can also mute the application by clicking a button that changes the volume to zero.

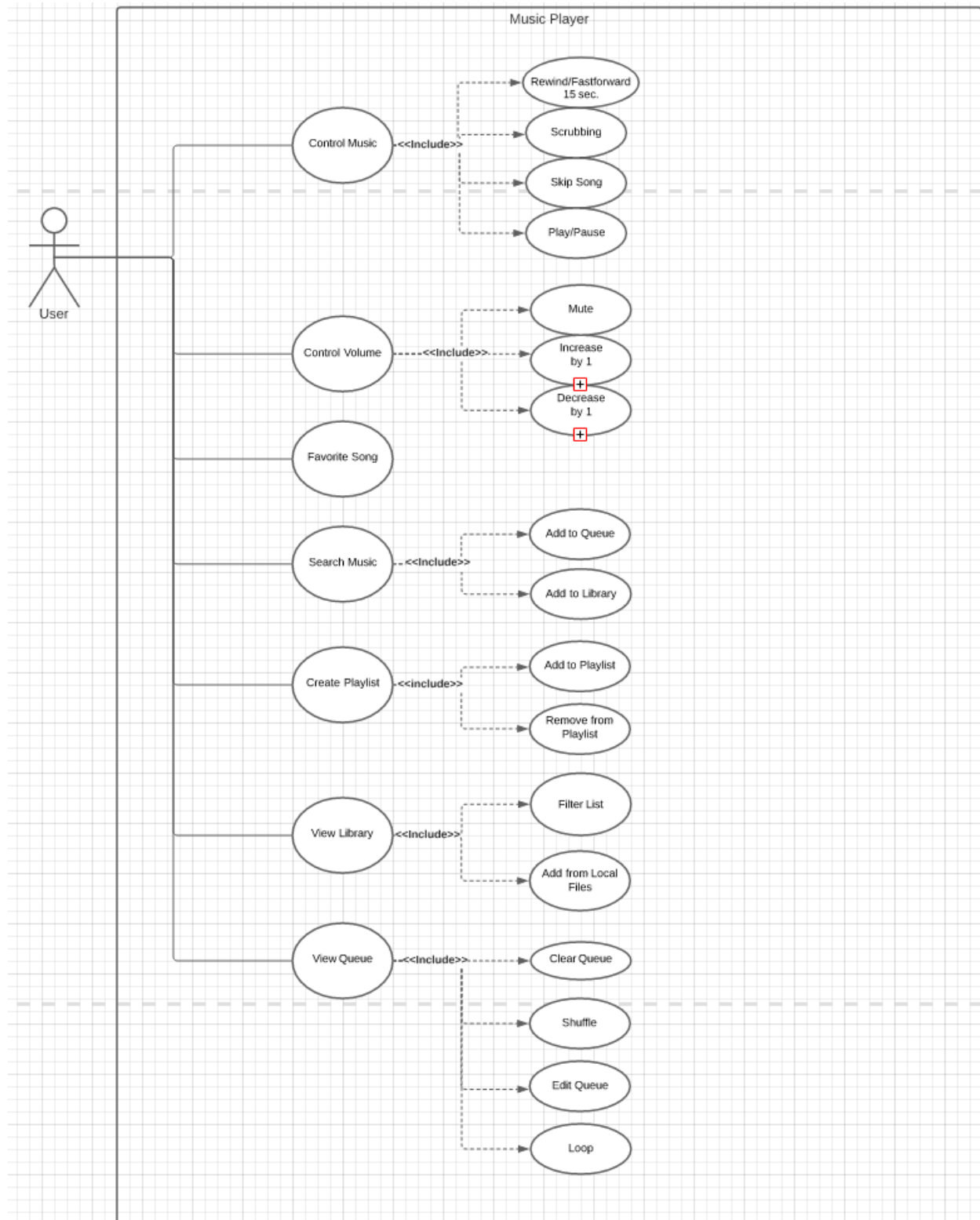
Create Playlist - User can create a list of songs and add to/remove songs from the list

Search Music - Users can use the keyboard to input the name of a song to search for it. From here, the user can choose to either add that song to the current queue or add that song to a playlist.

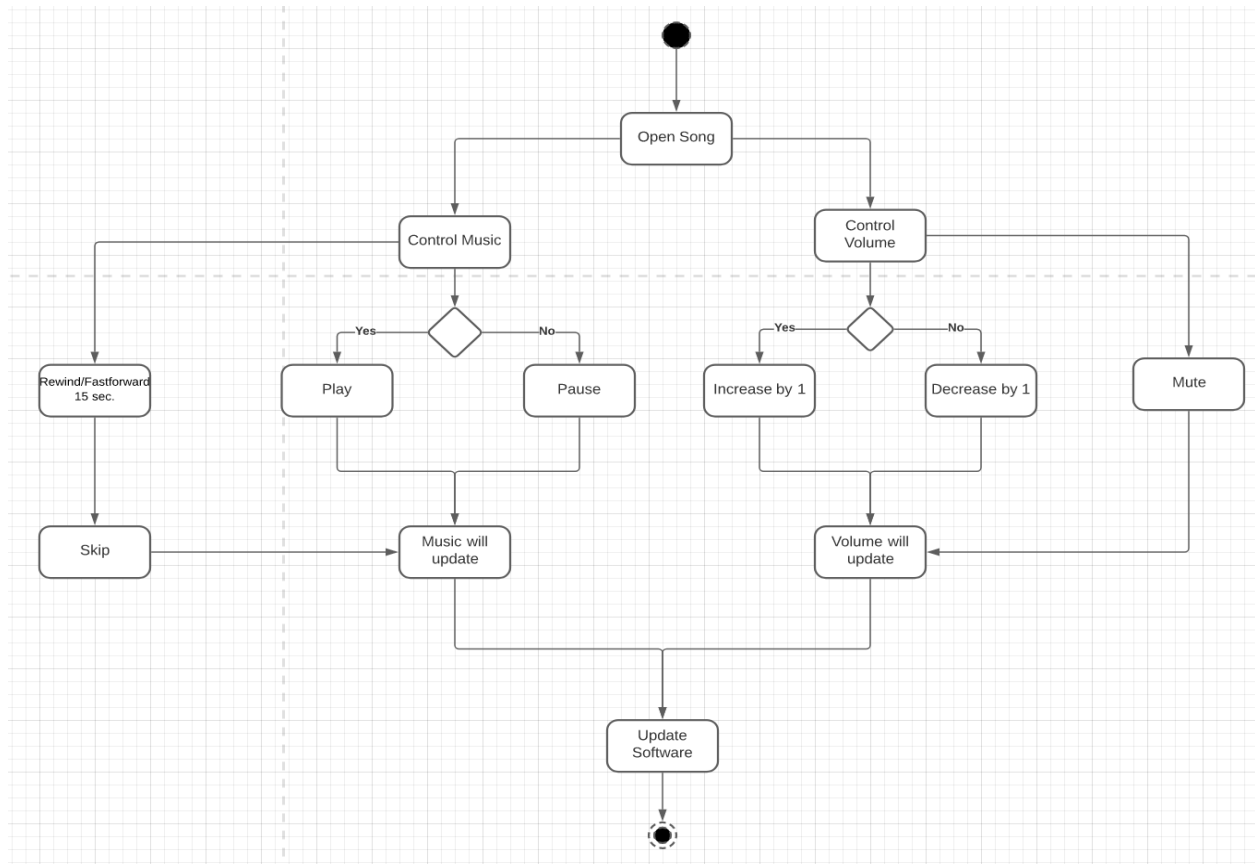
View Library - Users can view their entire collection of songs (their library), and can change the order in which the songs are listed by using filters to sort. The user can also choose to add a song to their library by uploading a local file on their device.

View Queue - Users can open the queue to view what song is currently playing as well as see what songs will be played next. Users also have options to choose to delete the queue, shuffle/randomize the order of the queue, add or remove songs from the queue, and set the current song to loop.

2.2 - Use Case Diagram

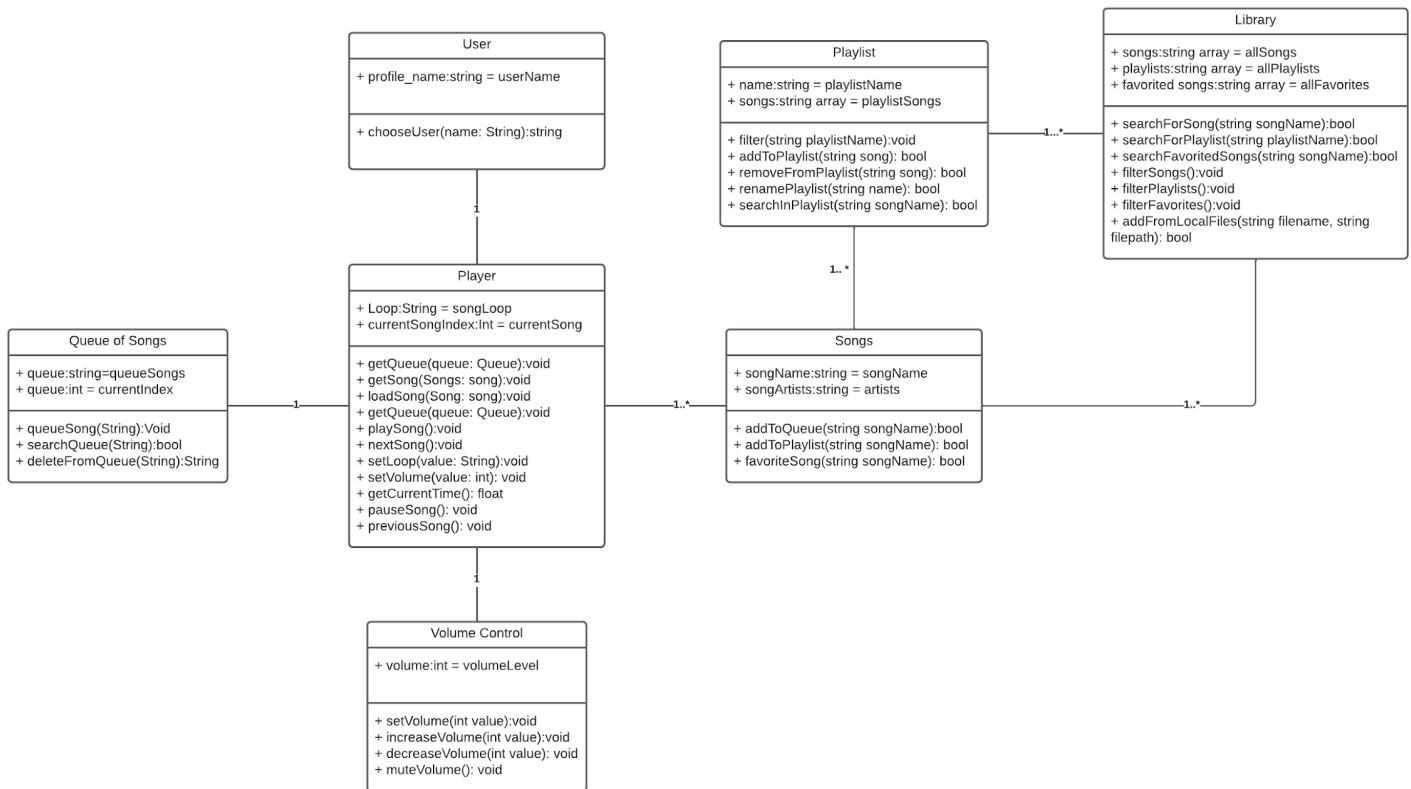


2.3 - [Activity Diagram](#) (at least 2 use cases, aim for biggest/most frequent)



2.4 - UML Class Diagram with connections , relationship multiplicity, and attributes

https://lucid.app/lucidchart/5f2d7cdb-4b8f-4622-94d6-e08a8b42976e/edit?invitationId=inv_b225ebd4-e756-45cb-8a45-4b795e2c4e89&page=0_0#

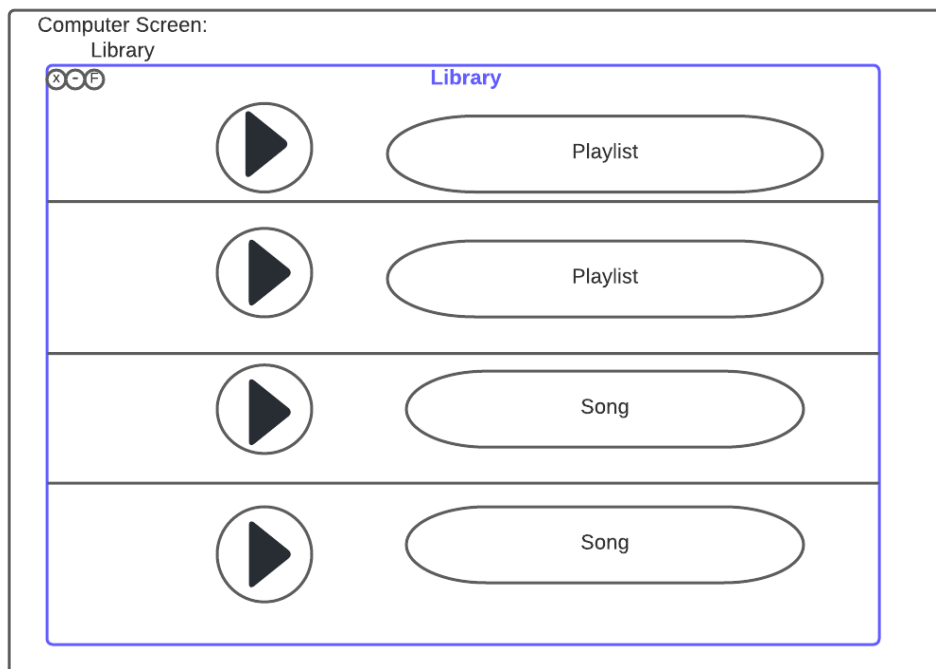
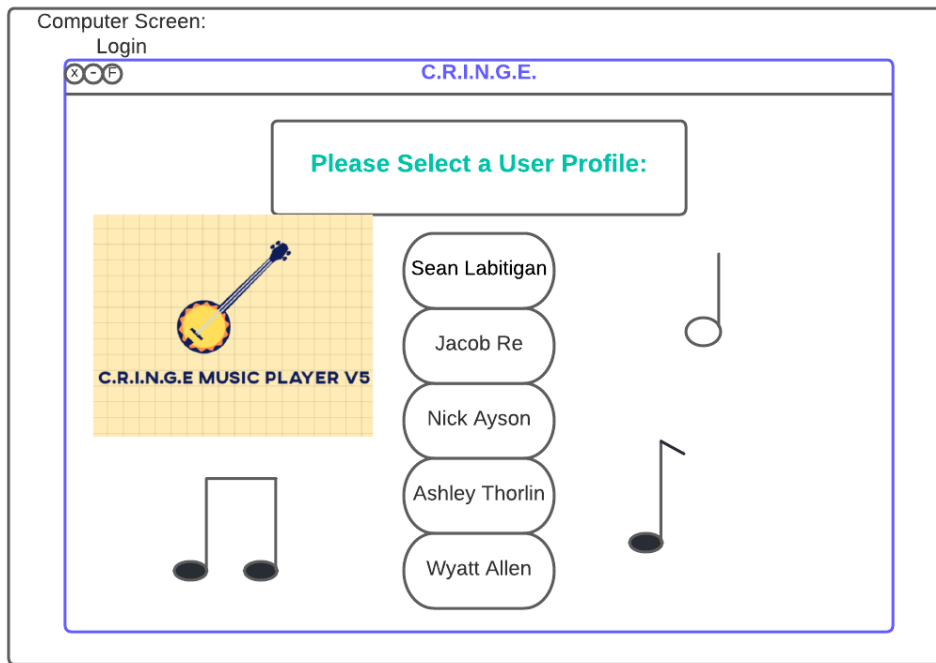


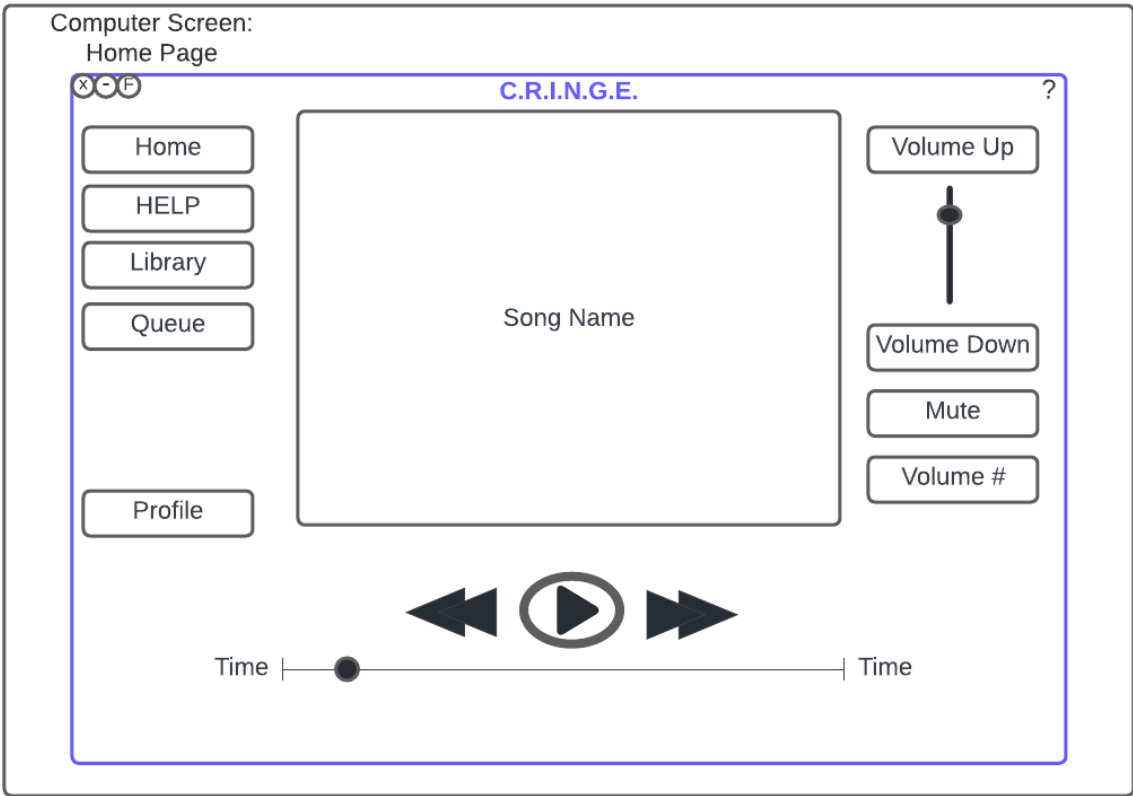
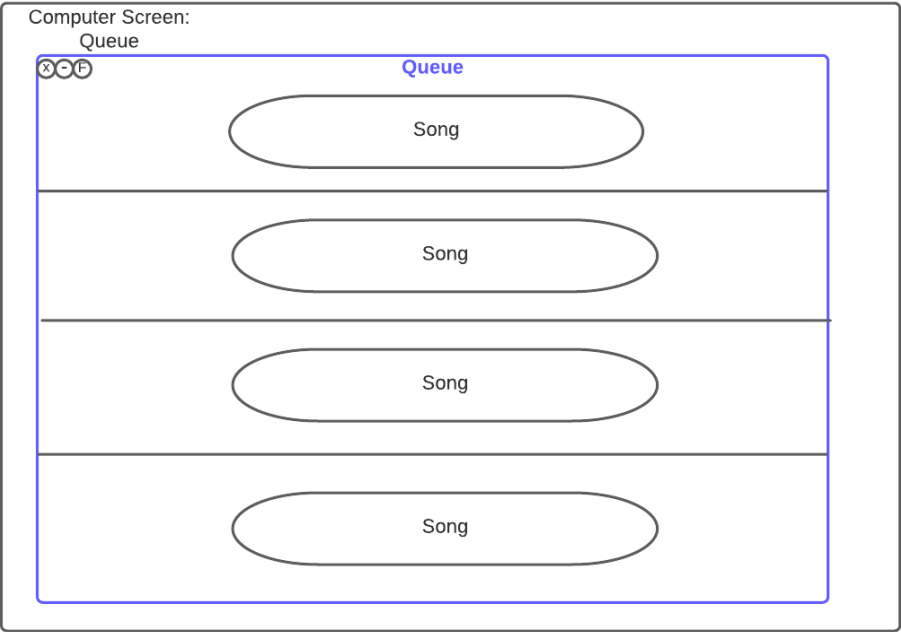
2.5 - Operating environment and dependencies (what software/components are you using?)

Operating Environment - Visual Studio Code with Live Share extension. Allows developers to view source code at the same time and discuss changes made. The project source code will be stored in a public repository on github which can run on all operating systems.

To run the software we are going to pull the code from github and simply run the executable with the command given in the readme. Creating each chart was made from a free chart software building website called LucidChart. We are also using the TKinter library via Python to build a user-friendly interface.

2.6 - Mockup UI screens (at least 2)





3 Notes

3.1 - Developer documentation (opt.)

3.2 - User documentation (opt.)

3.3 - Additional notes (opt.)

3.4 - Version Control/github link (opt.)

<https://github.com/nickayson/Music-Player-CPSC-362.git>