By: Maya Bishop

1. Overview of Classes

- **What class(es) did you design? - Describe each class in your design; explain the purpose of the class and of their member variables and member function**

Class:

Songlist

Description:

Represents a songlist of songs in an array and allows songs to be added, played and erased.

Member Variables:

- Array that stores songs
- Total size of the array
- Amount of the Array that has been filled

Member Functions (operations)

- addSong: Adds a song to the songlist if there is room to the songlist and it is not a duplicate of another song
- playSong: Plays song if there is a song at the position
- removeSong: Removes song from songlist and moves up the following songs in the order if it is a valid position in the songlist
- createList: Creates a songlist of a set size based off the input
- inSonglist: Checks if a Song is in a songlist by both the artist and song name

Class:

Song

Description:

Stores a Song by separating it by its song name and the artist it is by.

Member Variables:

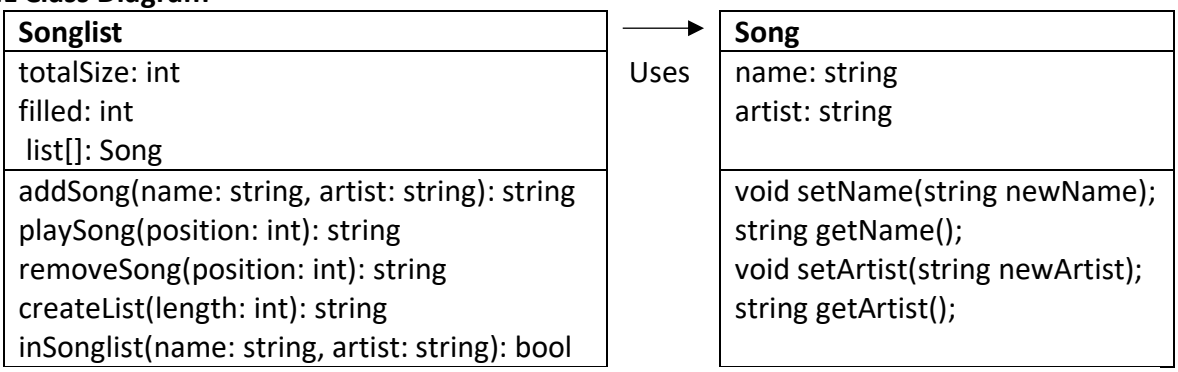
- Name of Song
- Name of Artist

Member Functions (operations)

- Functions to set and get the above variables along the Song class to be edit

By: Maya Bishop

UML Class Diagram



2. Constructors, Destructors, Operator Overloading

- **For each class, what are your design decisions regarding constructors?**
- **For each class, what are your design decisions regarding destructors?**
- **Provide your rational for any operators that you needed or decided to override.**

Songlist:

The constructor initializes a songlist of size 1 that is not filled since the size gets set later using createList(). The destructor deallocates the Song array and sets the pointer to the array to a null pointer. I did not find I needed any operators for this class.

Song:

The constructor for the Song class initializes the name and artist attributes to empty strings as they are set later on once they are used. There was no destructor needed for this class as there are no pointers or dynamic allocations and there was no need to override any of the operators.

3. Test Cases

- Describe your testing strategy for this class. What are some of the test cases you need to consider?

  I tested the basics such as setting the size of the list, adding an item to the list playing the item and then erasing it. Some edge cases I have tested for are erasing from the beginning, middle and end of the list and as well as adding more items into the list after erasing. Adding items until I have reached the max length of the list and making sure it stops adding. Trying to play an item past the length of the list and erase an item that is past the length of the list. I also have tried adding songs that have the same artist, the same song title and the exact same song that is in the songlist and making sure it works.