

[***Walker School of Business & Technology/Math & Computer Science Department***](http://www.webster.edu/catalog/current/undergraduate-catalog/colleges-schools/walker-school-of-business-and-technology.html#math)

**musicDB**

**COSC 3100 Data Structures II**

**Prof. Dr. Claude Chaudet**

**Student: Jose Pablo Murilo**

**Geneva**

**Fall 1, 2019**

[**Description of the problem**](#_fyhnqsx8bm5o) **2**

[**Proposed design and architecture**](#_9riz0fjtcjqn) **2**

[C++ program](#_an8lsrp94hse) 2

[File input/output structure](#_boz5yuumqnop) 4

[**Lessons learned**](#_v7ikiyge40cx) **4**

# **Description of the problem**

The program was developed in order to store information about Artists, their albums, and songs. The program is able to store data in memory, using the use of arrays, and can read and store data into files so they can be stored persistently. Information can be added and stored but no deletion of modification is permitted as of the writing of this document.

Artists have a unique identifier as well as a name. Albums have a unique identifier, an artist identifier, a name and the year of release. Songs have an album identifier, a track number, a name, and a duration in minutes.

The application must include an interface in the form of a command line interface where a series of menus and available options will be shown to the user. The user selects an option and each one will provide the input from the user or show the corresponding data.

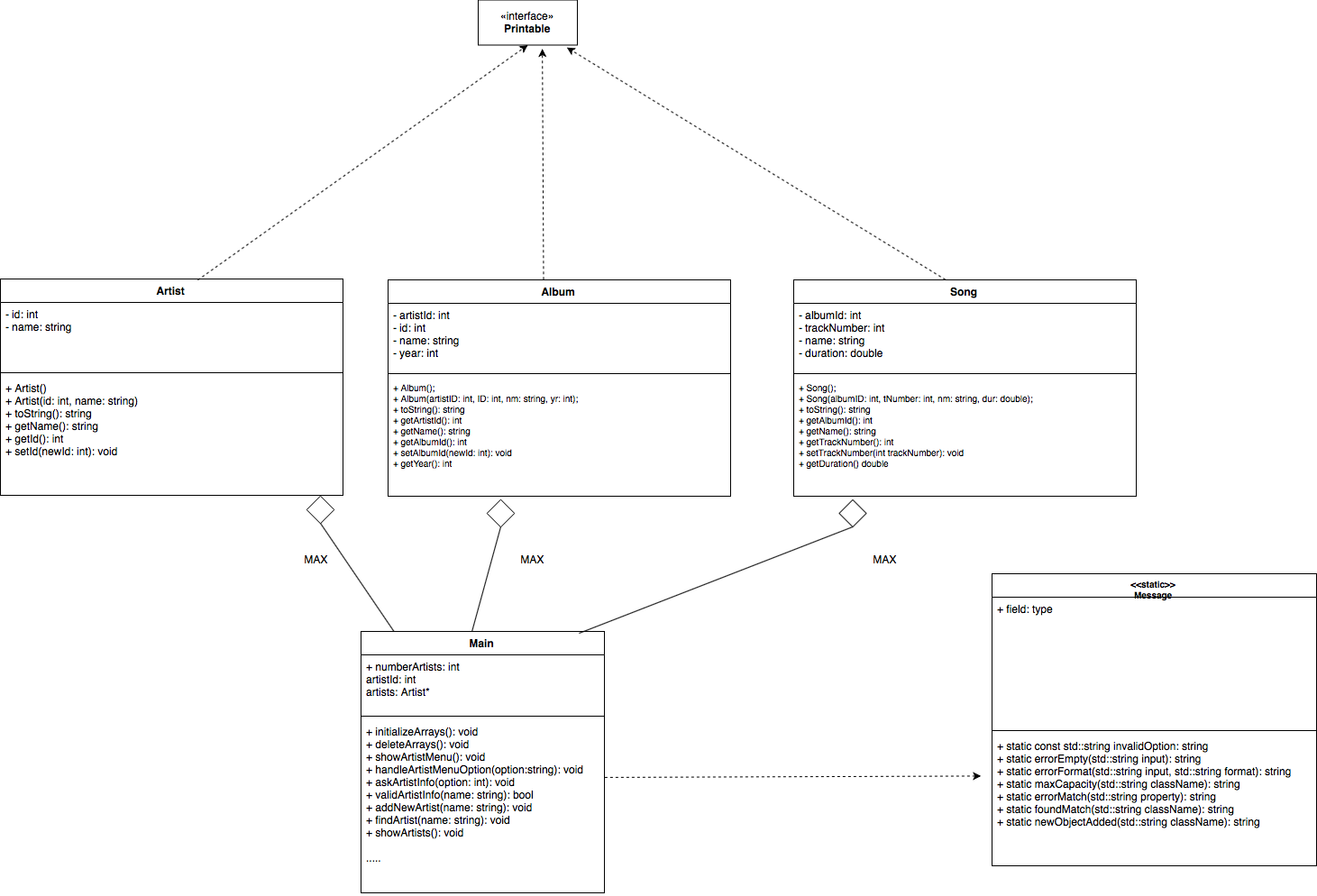
# **Proposed design and architecture**

## **C++ program**

The c++ program was developed using the text editor Visual Studio Code. The following libraries were used:

* **string:** library that uses char arrays but whose class takes care of the memory management, null termination, and allocation. Another advantage is its ease of use.
* **iostream:** library used for standard input/output streams. Permits the user to input data and data to be shown to console.
* **sstream:** iostream library that permits operation with strings
* **fstream:** iostream library that permite file manipulation

The program includes the following class diagram:



The Album, Song, and Artist classes are used to represent the objects that will be managed in the application. They all implement the Printable interface and implement the ToString() method. The main program also makes use of the static class Message in order to make output of error message easier and more understandable to read in code.

The main program also includes 3 collections: one for artists, one for albums and one for songs. A max size has been defined for each of the arrays. A predefined maximum of 500 was defined. The main program also includes counters in order to assign unique identifier to objects when adding a new artist or album.

## **File input/output structure**

The files that can be read by the application must follow the following format. Each line can only contain 1 artist/file/song. For the values for each object, they must be separated by the ‘,’ character. If a name contains a single or double quote, it will be considered as part of the name.

**Artist:** numericId,name

**Album:**numericIdArtist,numericIdAlbum,name,numericYear

**Song:**numericIdAlbum,numericTracknumber,name,duration

The files where the application saves the database is a default **dataDB.txt**. The name of the file can be changed in line 964 if needed.

# **Lessons learned**

The development of this program was helpful to put into practice the use of Object Oriented Programming, writing standardized code, error handling, and practice of concepts in C++. Object Oriented Programming helps abstract the technical specification and helps organize the structure of code into objects for better understanding and maintenance. Writing standardized code helps understand what is the role of the function or variable name and makes debugging faster because you can identify it in code. Error handling improves the robustness of the program and also helps user accessibility to avoid seeing unwanted errors that may discourage them from using the application.

There is a great cost in the use of CPU power in the functionalities of searching in the internal database. In the worst case scenario, the program has to search all 3 lists in order to find a match. A recommendation for this would be to implement the data storage in a different data structure or to replace the current method with binary search. The search was not implemented due to a problem with time. The arrays were later tried to be replaced for arrays with pointers in order to avoid the creation of empty objects and reduce the usage of more memory than needed, but it couldn’t be completed for problems with time as well.