

National University of Sciences & Technology

School of Electrical Engineering and Computer Science

Department of Computing

**CS-250: Data Structures and Algorithms (3+1): BSCS-7AB Fall 2018**

**End semester project**

**Music Player**

**SUBMITTED TO: Dr. Qaiser Riaz**

**SUBMITTED BY:**

1. **Name (CMS ID) Hasnat Amir 210752**
2. **Name (CMS ID) Faizan Abid Awan 211750**
3. **Name (CMS ID) Nouman Imran 215192**

**Class (Section): CS-7-A**

**Date of Submission: 23 December, 2018**

**Music Application**

**Motivation:**

We wanted to use the concepts of data structure and algorithms on practical level. And to do so a music application was the best and closest option for us. As it covers all the concepts of data structures and also best for practical use.

**Problem statement:**

No such practice was made in our course time to use our concepts in user defined data types. And a music player is a best example to practice our concepts in data structure and algorithms on practical scale.

**About the project:**

At start we had problems related to use of data handling for songs. Also most of the standard C++ libraries are not made for media handling. So we had to use a new framework that is Qt Creator, which is a professionally used C++ framework around the globe. All the media handling is made possible by help of Qt build-in libraries and combining it with linked list and sorting algorithms we created our music application.

We used a combination of single and double liked list for storing songs, and making a playlist out of it. Every song is stored in a **node of doubly linked** list and which is stored in a **singly linked** list which put all songs in form of a playlist (song list).

For sorting of songs we use **Quick sort** algorithm as it is the most efficient and fastest algorithm among all. It is used in **songlist.cpp** and is used by **mainWindow.cpp.**

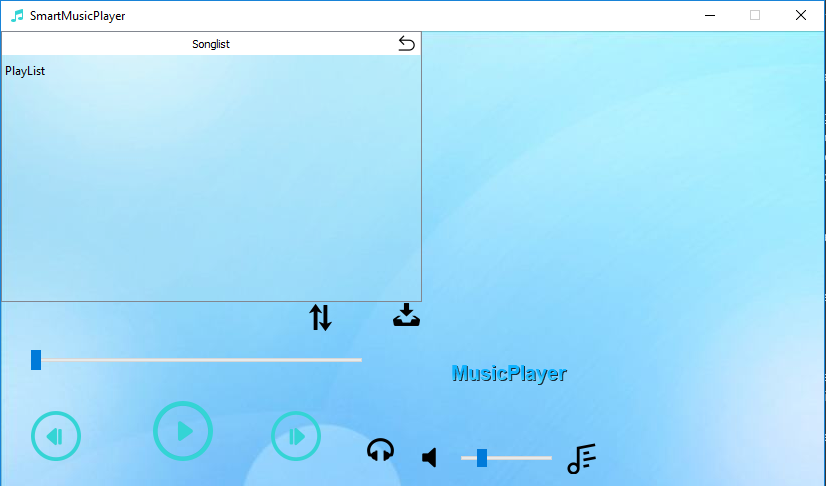
As we used single linked list, and quick sort so the time complexity is **nlogN**.

**Demonstration:**

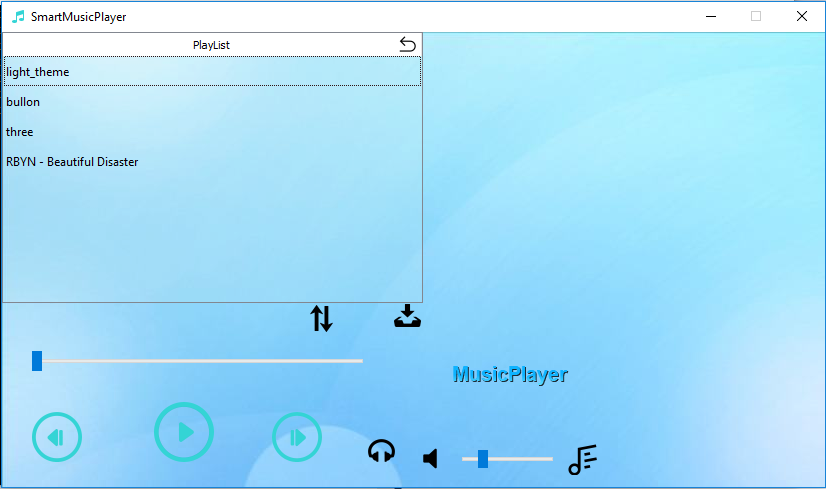
The program consists of 7 .cpp files each for a specific task:

* Main.cpp: for the start-up execution
* Mainwindow.cpp: it consists of all the music player features, including play, pause, next, previous, import, system tray, menu window, spectrum, lyrics, and background. This file also connects all the rest of the player file.
* Spectrum.cpp: contains the calculations and all the libraries related to sound spectrum.
* Lrc.cpp: contains libraries related to lyrics includes reading and showing it on music player window.
* FFT.cpp: all the calculations related to the conversion of sound to visible spectrum and its size is calculated by this file
* Songlist.cpp: all the songs are stored in this file, the head and tails of linked lists, and location of every songs are stored in this file. Import button, createList, remove are connected to this file.
* myPlaylist.cpp: the playlist which is being played is handled by this file, the playOrstop, play-chosen, playnNext, playPrev buttons are connected to this file.

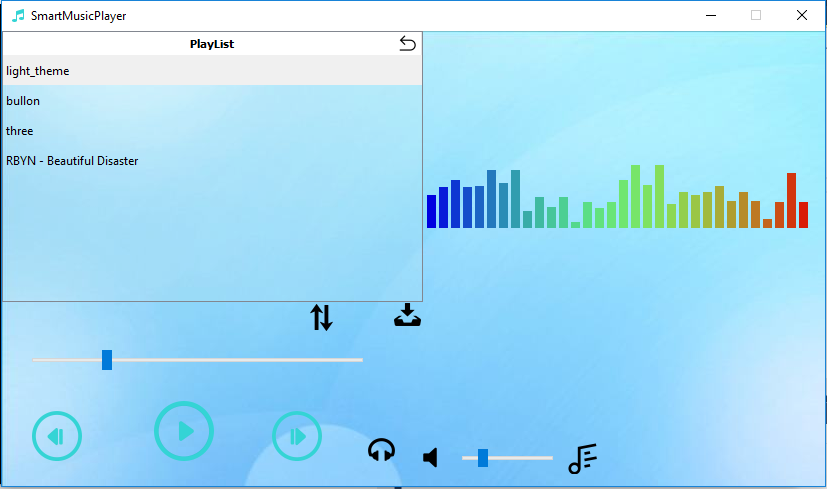
**The following are some screenshots of the music player at different stages:**



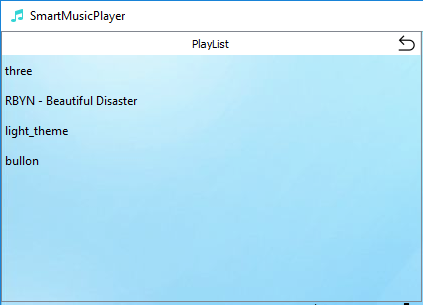
* The main window



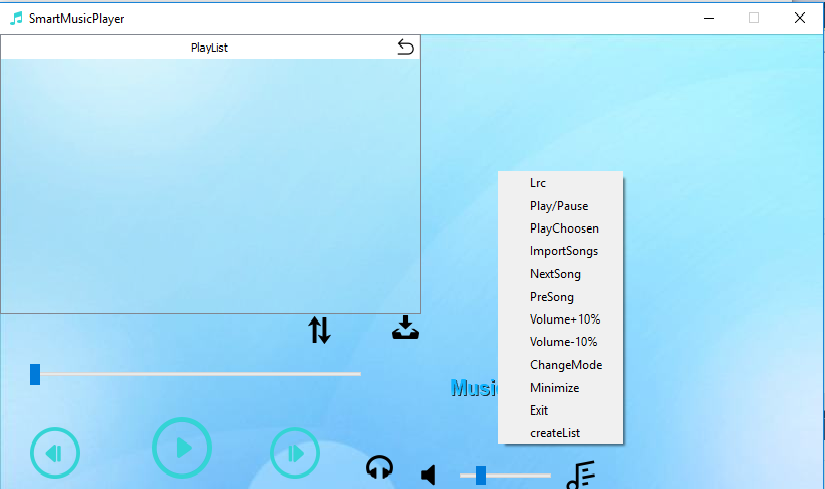
* Inside the playlist after importing the songs



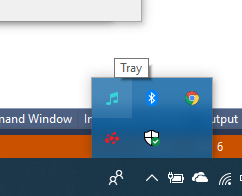
* The first song being played, along with its spectrum.



* The sorted play list according to their name in descending order.



* A view at in-window menu



* System tray