**CSE 310 – Applied Programming**

**Module Plan**

Name: Ferrin Mutuku

Date: 10/08/2025

Module # (1-3): Simple C++ Music Player

1. Identify which module you have selected to work on. Place an “X” in front of your selected module.

Cloud Databases

Data Analysis

Game Framework

GIS Mapping

Mobile App

Networking

SQL Relational Databases

Web Apps

X Language – C++

Language – Java

Language – Kotlin

Language – R

Language – Erlang

Language – JavaScript

Language – C#

Language - TypeScript

Language – Rust

1. At a high level, describe the software you plan to create that will fulfill the requirements of this module. Describe how each requirement will be met. This may change as you learn more about the technology or language you are learning.

This project is a basic command-line music player written in C++17 using the SFML Audio library.  
It allows users to add, play, pause, stop, and manage songs through a playlist stored in a text file.  
The application meets the module requirements as follows:

* **Variables and Expressions:** Used throughout for storing file paths, playback state, and volume levels.
* **Conditionals:** Used for handling user menu choices and playback logic.
* **Loops:** Implemented for the command menu and playlist iteration.
* **Functions:** Used to organize playback, playlist loading/saving, and input handling.
* **Classes:** A MusicPlayer class manages all song-related operations.
* **STL Data Structure:** A std::vector<std::string> stores the playlist.
* **File I/O:** Reads from and writes to playlist.txt for song management.
* **Dynamic Memory:** Uses new and delete for dynamically managing SFML sf::Music objects.

1. Create a detailed schedule using the table below to complete your selected module during this Sprint. Include the task and duration for each day. You are expected to spend 24 hours every Sprint working on this individual module and other activities in the course. Time spent on this individual module should be **at least** 12 hours.

**Time Spent – 20 hours**

|  |  |  |
| --- | --- | --- |
|  | **First Week of Sprint** | **Second Week of Sprint** |
| **Monday** | Install SFML and set up project folder structure | Test file loading and saving functions |
| **Tuesday** | Implement base class with variables and functions | Integrate playback (play, pause, stop) |
| **Wednesday** | Add menu system with loops and conditionals | Test playlist navigation (next, previous) |
| **Thursday** | Implement STL vector playlist managemen | Add volume and loop controls |
| **Friday** | Add file I/O and dynamic object creation | Code cleanup and commenting |
| **Saturday** | Compile, debug, and test on Fedora Linux | Create README, finalize submission |

1. Identify at least two risks that you feel will make it difficult to succeed in this module. Identify an action plan to overcome each of these risks.

Risk

Difficulty linking SFML audio libraries on Linux

Action Plan

Follow installation documentation, verify with test compilation, and use -lsfml-audio -lsfml-system flags correctly.

Risk

Limited support for MP3 files in SFML

Action Plan

Convert all audio files to .ogg using ffmpeg and document the process in the README file.