Music Player Application - Developer's Guide

Introduction

This guide provides an overview of the architecture, functionality, and code organization of the Music Player application.

Key Information Regarding Environment Configuration: Apache NetBeans is highly recommended as the running environment. In addition, since this project is developed using JavaFX, which might not be included in your JDK depending on the versions, please add JavaFX as well before running the application. Besides, we have included 4 songs in our final deliverables, please refer to the Troubleshoot Section of the User Manual for importing these 4 songs.

Project Structure

1. Main Components

MusicPlayerApp.java: Entry point of the application.
MusicPlayerController.java: Handles all user interactions and controls application logic.
FXML and CSS Files: Define the user interface layout and styles.

2. Key Libraries

JavaFX: Provides UI components.
MediaPlayer: Manages audio playback.

3. Folder Structure

```
/src
   /musicplayerapp
      MusicPlayerApp.java
      MusicPlayerController.java
      MusicPlayerFXML.fxml
      MusicPlayerCSS.css
```

How It Works

1. Core Functionalities
   - Play Songs:
     Uses JavaFX Media and MediaPlayer for audio playback. playSong() method initiates playback and displays song details.
   - Playback Modes:
     Order: Sequential playback using indices.
     Shuffle: Random playback via shuffled indices.
     Loop: Repeat the current song.

2. Playlist Management

- Library Playlist:
  Populated during initialization (initialize method).
- User Playlist:
  Managed dynamically via addToPlaylist() and removeFromPlaylist().

Adding Features

1. Add More Playback Modes
Extend ComboBox options and update skipClicked() and previousClicked() methods.
Modify handleEndOfSong() to handle new modes.

2. Support Additional Audio Formats
Modify Media initialization to support formats beyond MP3. Add error handling for unsupported formats.

3. Save Playlists to File
Write the User Playlist to a .txt or .json file.
Load the saved playlist on application startup.
Error Handling
Invalid File Path: Display an error in lblSongInfo.
Corrupt Media: Handle via setOnError in MediaPlayer.
Testing
Test for:
File path errors.
Playback of corrupted or unsupported files.
UI responsiveness under various actions.