Console Music Player Assignment

Assignment Overview

Create a console-based music player application that allows users to manage songs and playlists through a command-line interface.

Learning Objectives

- Practice object-oriented programming concepts
- Implement data structures (arrays/lists, dictionaries/maps)
- Handle user input and validation
- Create a menu-driven console application
- Work with file path management
- Implement CRUD (Create, Read, Update, Delete) operations

Requirements

Core Features

Your application must implement the following features:

1. **Song Management**

- Add songs by providing file path
- View all songs in the library
- Play songs (simulate playback)

2. **Playlist Management**

- Create new playlists
- Add songs to playlists
- View all playlists
- View songs in a specific playlist

3. **Navigation**

- Main menu system
- Sub-menus for different operations
- Option to return to previous menu or exit

Technical Requirements

Data Structure Requirements

- Use appropriate data structures to store songs and playlists
- Songs should have: title, artist, file path, duration (optional)
- Playlists should have: name, list of songs

Input Validation

- Validate file paths (check if they exist or follow expected format)
- Handle invalid menu choices gracefully
- Prevent duplicate song additions
- Validate playlist names (no empty names)

Error Handling

- Handle cases where songs don't exist
- Handle empty playlists
- Provide meaningful error messages

Marking Rubric (Total: 100 points)

Functionality (60 points)

- **Song Management (20 points)**

Add songs: 8 pointsView songs: 6 pointsPlay songs: 6 points

- **Playlist Management (25 points)**

- Create playlists: 8 points

- Add songs to playlists: 10 points

- View playlists: 7 points

- **Menu System (15 points)**

Main menu: 5 pointsSub-menus: 5 pointsNavigation flow: 5 points

Code Quality (25 points)

- **Structure and Organization (10 points)**

- Proper use of classes/functions
- Logical code organization
- Appropriate data structures

- **Input Validation (8 points)**

- Validates user input
- Handles edge cases
- Prevents crashes

- **Error Handling (7 points)**

- Meaningful error messages
- Graceful failure handling

User Experience (15 points)

- **Interface Design (8 points)**

- Clear menu options
- Intuitive navigation
- Consistent formatting

- **User Feedback (7 points)**

- Confirmation messages
- Status updates
- Clear instructions

DO's 🔽

- 1. **Use meaningful variable and function names**
- 2. **Implement proper input validation**
- 3. **Provide clear user instructions and prompts**
- 4. **Use appropriate data structures (lists, dictionaries)**
- 5. **Include comments explaining complex logic**
- 6. **Test your application thoroughly**
- 7. **Handle edge cases (empty inputs, invalid choices)**
- 8. **Use consistent formatting and indentation**
- 9. **Implement a clean menu system**
- 10. **Show appropriate success/error messages**

DON'Ts X

- 1. **Don't ignore input validation**
- 2. **Don't use global variables excessively**
- 3. **Don't hardcode values that should be dynamic**
- 4. **Don't leave the application vulnerable to crashes**
- 5. **Don't create overly complex menu structures**
- 6. **Don't forget to handle empty collections gracefully**
- 7. **Don't use misleading variable names**
- 8. **Don't skip error handling**
- 9. **Don't create infinite loops without exit conditions**
- 10. **Don't submit code without testing all features**

Submission Requirements

Documentation

- Include brief comments explaining your approach
- Document any assumptions made
- Explain any additional features implemented

Testing

- Test all menu options
- Test with both valid and invalid inputs

- Ensure the application doesn't crash unexpectedly ## Bonus Features (Extra Credit - 10 points)

- **Search functionality** (5 points): Search songs by title or artist
- **Remove songs from playlist** (3 points): Allow users to remove songs
- **Shuffle playlist** (2 points): Randomly order playlist songs

Deadline and Submission

- **Due Date**: [Insert your deadline]
- **Submission Method**: [Insert your submission method]
- **Late Penalty**: [Insert your late policy]

Getting Started Tips

- 1. Start with the basic menu structure
- 2. Implement one feature at a time
- 3. Test each feature before moving to the next
- 4. Keep your code simple and readable
- 5. Don't try to implement everything at once

Sample File Paths for Testing

You can use these example paths for testing:

- \music/songs/song1.mp3\
- `/music/rock/bohemian_rhapsody.mp3`
- `/downloads/music/favorite_song.wav`
- `C:\Music\pop\dancing_queen.mp3` (Windows format)

Good luck with your assignment! Remember to start early and test frequently.

Complete Console Flow Demonstration

Below is a comprehensive example showing exactly how your application should behave when running. Pay attention to the menu structure, user prompts, and system responses.

Application Startup and Main Menu

™ WELCOME TO MUSIC PLAYER
====== MAIN MENU ======
1. Song Management

2. Playlist Management3. View Library
4. Exit Application
Please enter your choice (1-4): 1
Song Management Flow
····
====== SONG MANAGEMENT ======
1. Add New Song
2. View All Songs
3. Play Song
4. Back to Main Menu
Please enter your choice (1-4): 1
ADD NEW SONG
Enter song title: Bohemian Rhapsody
Enter artist name: Queen
Enter file path: /music/rock/bohemian_rhapsody.mp3
Enter duration (optional, press Enter to skip): 6:07
Song "Bohemian Rhapsody" by Queen added successfully!
Press Enter to continue
====== SONG MANAGEMENT ======
1. Add New Song
2. View All Songs
3. Play Song
4. Back to Main Menu
Please enter your choice (1-4): 2
ALL SONGS IN LIBRARY
1. Bohemian Rhapsody - Queen (6:07)
Path: /music/rock/bohemian_rhapsody.mp3
Total songs: 1

Press Enter to continue...

====== SONG MANAGEMENT ======
1. Add New Song
2. View All Songs
3. Play Song
4. Back to Main Menu
=======================================
Please enter your choice (1-4): 1
ADD NEW SONG
Enter song title: Imagine
Enter artist name: John Lennon
Enter file path: /music/classic/imagine.mp3
Enter duration (optional, press Enter to skip): 3:07
Song "Imagine" by John Lennon added successfully!
Press Enter to continue
====== SONG MANAGEMENT ======
1. Add New Song
2. View All Songs
3. Play Song
4. Back to Main Menu
Please enter your choice (1-4): 3
PLAY SONG
Available songs:
1. Bohemian Rhapsody - Queen
2. Imagine - John Lennon
Enter song number to play (1-2): 1
Now playing: "Bohemian Rhapsody" by Queen File: /music/rock/bohemian_rhapsody.mp3 Duration: 6:07
[Simulating playback] Playing 100%
✓ Playback completed!

Press Enter to continue...

======= SONG MANAGEMENT ====================================
Please enter your choice (1-4): 4
Playlist Management Flow
···
1. Song Management 2. Playlist Management 3. View Library 4. Exit Application
Please enter your choice (1-4): 2
 ========= PLAYLIST MANAGEMENT ====================================
Please enter your choice (1-5): 1
CREATE NEW PLAYLIST Enter playlist name: My Favorites
✓ Playlist "My Favorites" created successfully!
Press Enter to continue
======== PLAYLIST MANAGEMENT ====================================

5. Back to Main Menu

=======================================
Please enter your choice (1-5): 1
CREATE NEW PLAYLIST Enter playlist name: Rock Classics
✓ Playlist "Rock Classics" created successfully!
Press Enter to continue
 ====================================
Please enter your choice (1-5): 2
ADD SONG TO PLAYLIST Available playlists: 1. My Favorites 2. Rock Classics
Select playlist (1-2): 2
Available songs: 1. Bohemian Rhapsody - Queen 2. Imagine - John Lennon
Select song to add (1-2): 1
☑ "Bohemian Rhapsody" added to "Rock Classics" playlist!
Press Enter to continue
1. Create New Playlist 2. Add Song to Playlist 3. View All Playlists 4. View Playlist Contents 5. Back to Main Menu

Please enter your choice (1-5): 4
VIEW PLAYLIST CONTENTS Available playlists: 1. My Favorites (0 songs) 2. Rock Classics (1 song)
Select playlist to view (1-2): 2
PLAYLIST: Rock Classics 1. Bohemian Rhapsody - Queen (6:07)
Total songs in playlist: 1
Press Enter to continue
1. Create New Playlist 2. Add Song to Playlist 3. View All Playlists 4. View Playlist Contents 5. Back to Main Menu
Please enter your choice (1-5): 5
View Library Flow
1. Song Management 2. Playlist Management 3. View Library 4. Exit Application ====================================
Please enter your choice (1-4): 3
====== LIBRARY OVERVIEW ======
SONGS LIBRARY:

📊 Total songs: 2

- PLAYLISTS:
- 1. My Favorites (0 songs)
- 2. Rock Classics (1 song)
- Total playlists: 2

Press Enter to continue...

Error Handling Examples

====== SONG MANAGEMENT =======

- 1. Add New Song
- 2. View All Songs
- 3. Play Song
- 4. Back to Main Menu

Please enter your choice (1-4): 5

X Invalid choice! Please enter a number between 1-4.

Please enter your choice (1-4): 1

--- ADD NEW SONG ---

Enter song title:

X Song title cannot be empty! Please try again.

Enter song title: Hotel California

Enter artist name:

X Artist name cannot be empty! Please try again.

Enter artist name: Eagles Enter file path: invalid_path

Marning: File path may not exist, but song will be added anyway.

Enter duration (optional, press Enter to skip):

Song "Hotel California" by Eagles added successfully!

--- PLAY SONG ---

Available songs:

- 1. Bohemian Rhapsody Queen
- 2. Imagine John Lennon
- 3. Hotel California Eagles

Enter song number to play (1-3): 5

Enter song number to play (1-3): 2
Now playing: "Imagine" by John Lennon
Application Exit

====== MAIN MENU ======
1. Song Management
2. Playlist Management
3. View Library
4. Exit Application
Please enter your choice (1-4): 4
□ Thank you for using Music Player! □
Goodbye!
[Application terminates]
[Application terminates]
Key Points Demonstrated
1. **Clear Menu Structure**: Each menu is clearly labeled and numbered
2. **User Feedback** : Success messages (☑), errors (ズ), warnings (⚠)
3. **Input Validation**: Handles empty inputs and invalid choices
4. **Navigation**: Users can always return to previous menus

5. ****Visual Elements****: Uses emojis and formatting for better UX

7. **Status Messages**: Keeps users informed of what's happening

6. **Data Display**: Shows counts, lists items clearly

8. **Error Recovery**: Allows users to retry after errors

X Invalid song number! Please enter a number between 1-3.