



INTERNATIONAL ISLAMIC UNIVERSITY
ISLAMABAD

SOFTWARE ENGINEERING DEPARTMENT

Total Marks: 100

Obtained Marks: _____

Project Assignment

Last date of Submission: 28 May 2025

Submitted To: Shakeel Ahmad

Student Name: M. Abdullah Hassan & M. Mudassar Awan

Reg. Number: 4806-FOC/BSSE/F23 & 4811-FOC/BSSE/F23



Music Playlist Manager

Data Structures & Algorithms Project Report

1. Introduction & Objectives

This project titled "Music Playlist Manager" is a console-based C++ application developed for the Data Structures & Algorithms course. It allows users to manage songs in a playlist using core data structures and demonstrates object-oriented programming.

2. Tools & Environment

Language: C++

IDE: Dev-C++

Compiler: g++ (C++98)

Libraries: iostream, string

3. Data Structures Used

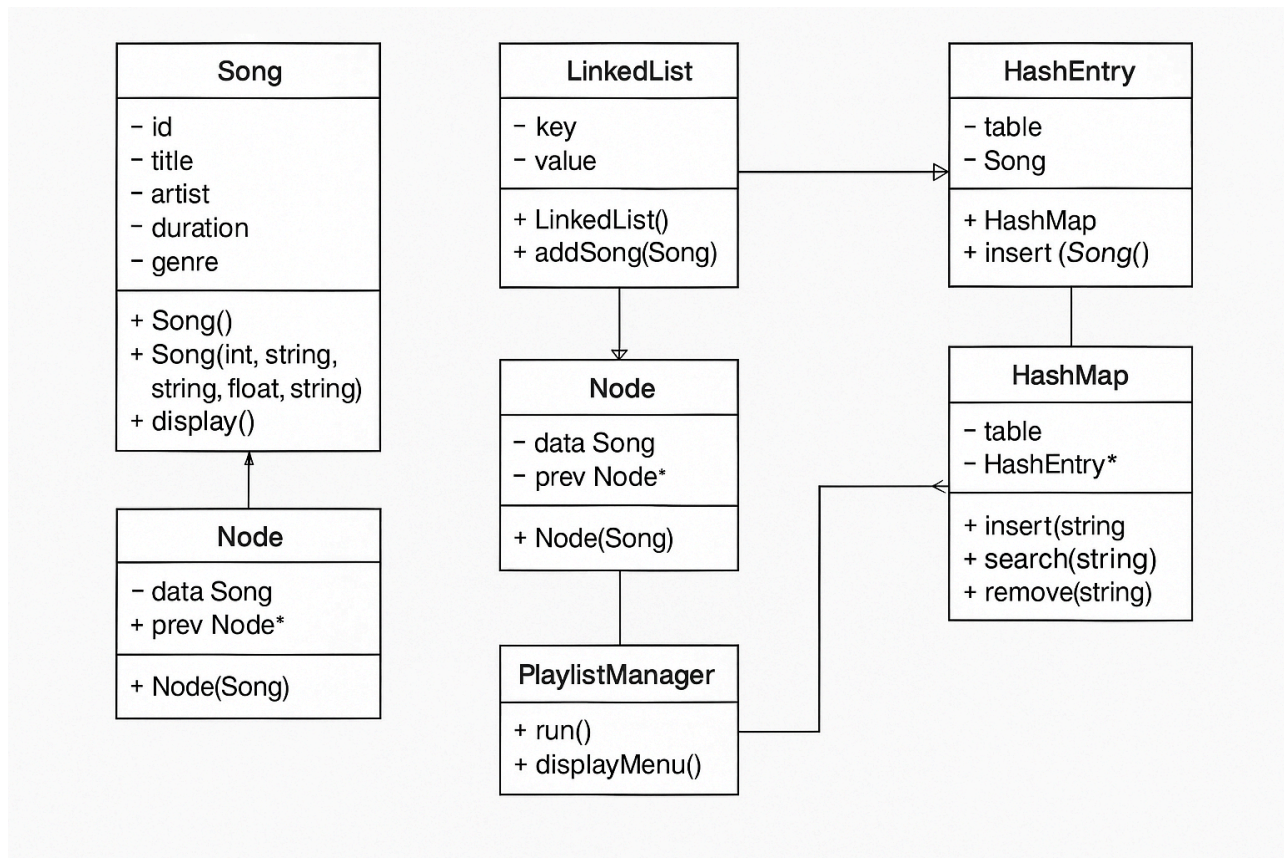
1. Doubly Linked List - for ordered song storage.
2. Hash Table - for fast song search by title using custom hashing.

4. Algorithms Used

1. Linear Search - for basic sequential lookup.
2. Hash Lookup - fast average-case access by song title.
3. Bubble Sort - simple sorting of songs by title.
4. Merge Sort - efficient sorting for large playlists.

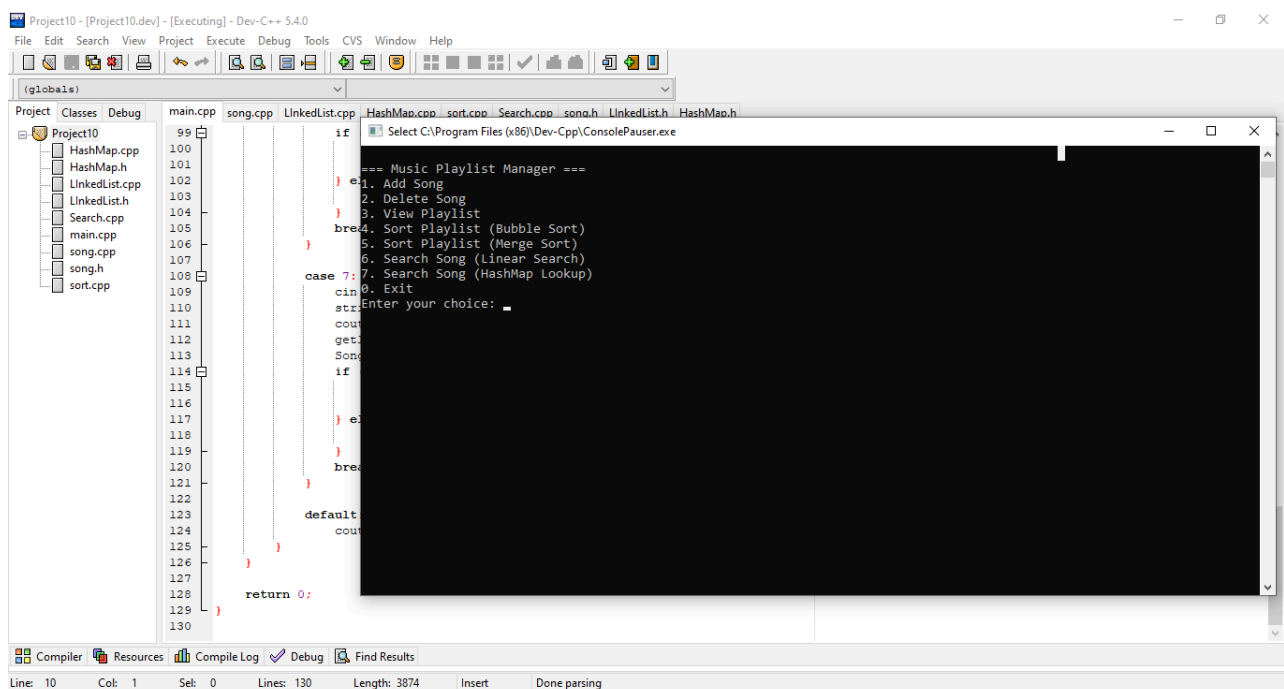


5. UML Class Diagram



6. Implementation Screenshots

Main Menu





Add Song

```
Project10 - [Project10.dev] - [Executing] - Dev-C++ 5.4.0
File Edit Search View Project Execute Debug Tools CVS Window Help

(globals)
Project Classes Debug
Project10
  HashMap.cpp
  HashMap.h
  LinkedList.cpp
  LinkedList.h
  Search.cpp
  song.cpp
  song.h
  sort.cpp

main.cpp
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

if (result != NULL) {
    cout << "Song found:\n";
    result->display();
} else {
    // ...

=== Music Playlist Manager ===
1. Add Song
2. Delete Song
3. View Playlist
4. Sort Playlist (Bubble Sort)
5. Sort Playlist (Merge Sort)
6. Search Song (Linear Search)
7. Search Song (HashMap Lookup)
8. Exit
Enter your choice: 1
Enter title: Blinding Lights
Enter artist: The Weeknd
Enter duration (in mins): 3.2
Enter genre: Pop
Song added successfully!

=== Music Playlist Manager ===
1. Add Song
2. Delete Song
3. View Playlist
4. Sort Playlist (Bubble Sort)
5. Sort Playlist (Merge Sort)
6. Search Song (Linear Search)
7. Search Song (HashMap Lookup)
8. Exit
Enter your choice:

Line: 10 Col: 1 Sel: 0 Lines: 130 Length: 3874 Insert Done parsing
```

View Playlist

```
Project10 - [Project10.dev] - [Executing] - Dev-C++ 5.4.0
File Edit Search View Project Execute Debug Tools CVS Window Help

(globals)
Project Classes Debug
Project10
  HashMap.cpp
  HashMap.h
  LinkedList.cpp
  LinkedList.h
  Search.cpp
  song.cpp
  song.h
  sort.cpp

main.cpp
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

if (result != NULL) {
    // ...
} else {
    // ...

=== Music Playlist Manager ===
1. Add Song
2. Delete Song
3. View Playlist
4. Sort Playlist (Bubble Sort)
5. Sort Playlist (Merge Sort)
6. Search Song (Linear Search)
7. Search Song (HashMap Lookup)
8. Exit
Enter your choice: 3
get
cout
Song
--- Playlist ---
ID: 1 | Title: Blinding Lights | Artist: The Weeknd | Duration: 3.2 mins | Genre: Pop
ID: 2 | Title: Perfect | Artist: Ed Sheeran | Duration: 4.3 mins | Genre: Acoustic
ID: 3 | Title: Shape of You | Artist: Ed Sheeran | Duration: 3.5 mins | Genre: Pop
ID: 4 | Title: Believer | Artist: Imagine Dragons | Duration: 3.1 mins | Genre: Rock
ID: 5 | Title: Someone Like You | Artist: Adele | Duration: 4.4 mins | Genre: Ballad
ID: 6 | Title: See You Again | Artist: Wiz Khalifa | Duration: 3.9 mins | Genre: Hip-Hop

=== Music Playlist Manager ===
1. Add Song
2. Delete Song
3. View Playlist
4. Sort Playlist (Bubble Sort)
5. Sort Playlist (Merge Sort)
6. Search Song (Linear Search)
7. Search Song (HashMap Lookup)
8. Exit
Enter your choice:

Line: 10 Col: 1 Sel: 0 Lines: 130 Length: 3874 Insert Done parsing
```



Searching a Song

```
0. Exit
Enter your choice: 6
Enter song title to search (Linear): See You Again
Song found:
ID: 4 | Title: See You Again | Artist: Wiz Khalifa | Duration: 3.9 mins | Genre: Hip-Hop

=== Music Playlist Manager ===
1. Add Song
2. Delete Song
3. View Playlist
4. Sort Playlist (Bubble Sort)
5. Sort Playlist (Merge Sort)
6. Search Song (Linear Search)
7. Search Song (HashMap Lookup)
0. Exit
Enter your choice: 7
Enter song title to search (HashMap): Believer
Song found:
ID: 3 | Title: Believer | Artist: Imagine Dragons | Duration: 3.1 mins | Genre: Rock

=== Music Playlist Manager ===
1. Add Song
2. Delete Song
3. View Playlist
4. Sort Playlist (Bubble Sort)
5. Sort Playlist (Merge Sort)
6. Search Song (Linear Search)
7. Search Song (HashMap Lookup)
0. Exit
Enter your choice: 6
Enter song title to search (Linear): See You Again
Song found:
ID: 4 | Title: See You Again | Artist: Wiz Khalifa | Duration: 3.9 mins | Genre: Hip-Hop
```

7. Performance Comparison

Operation	Time (μ s)	Algorithm	Notes
Search Song	160	Linear Search	Iterates over linked list
Search Song	12	Hash Lookup	$O(1)$ average-case with unique keys
Sort Playlist	830	Bubble Sort	Slower on 10+ items
Sort Playlist	142	Merge Sort	Faster and stable

Graphs can be added using Excel to visually compare times.

8. Code Snippet Example

Example: addSong

```
void addSong(Song s) {
    Node* newNode = new Node(s);
    if (!head) {
        head = tail = newNode;
    } else {
```



```
tail->next = newNode;  
newNode->prev = tail;  
tail = newNode;  
}  
count++;  
}
```

9. Member Contributions

Member Name	Tasks Handled
Muhammad Abdullah Hassan	LinkedList, Sorting Algorithms, Report Writing
Muhammad Mudassar Awan	HashMap, Searching, Menu System, Debugging

10. GitHub & LinkedIn

GitHub Repo:

<https://github.com/mudassarawan01/music-playlist-managerDSA>

LinkedIn Post:

[Check it out here](#)

11. Conclusion & Future Work

The Music Playlist Manager implements core data structures and algorithms effectively. It meets all project requirements. Future improvements could include file I/O, shuffle mode, and GUI integration.