

**Semester Project**

**Data Structures and Algorithms**

**(ES-221)**

**Muhammad Sohaib Abdullah (2023520)**

**Saad Afridi (2023617)**

**Deliverable 2: Initial Implementation & Data Structure Integration**

**1. Introduction**

A C++-based Train Seat Booking System is described in this report. Using a doubly linked list, the system lets users book and cancel train seats while also keeping track of available seats. The program provides a simple, menu-driven interface for managing bookings efficiently.

**2. System Overview**

**2.1 Important Parts**

* Node Structure: Links to adjacent nodes (next, prev) and stores passenger names (data).
* Train Class: Uses functions to manage reservations:

append() = Adds a booking.

del()= cancels a reservation.

print() =shows both booked seats and empty seats.

* Users are able to book, cancel, or exit from the main menu.

**2.2 Features**

* Dynamic Seat Management (a double-linked list for insertions of O(1)).
* Real-Time Updates (Shows remaining seats after each operation).
* A User-Friendly Menu and a Simple Interface

**3. Functionality**

**3.1 Purchasing a Seat**

* (total\_seats\_avail) Checks seat availability.
* Appends passenger name to the list.
* Makes available seats available.

**3.2 Cancelling a Reservation**

* Searches for the passenger name.
* Removes the node if found.
* Makes more seats available.

**3.3 Displaying Reservations**

* Arranged the booked passengers on a list.
* Identifies empty seats with "Empty."

**4. Limitations and Work to Come**

* There is no persistence of data (bookings are lost upon exit).
* Fixed Capacity (Hardcoded to 10 seats).
* Enhanced Features: File the information.
* Include classes and seat numbers.

**5. Conclusion**

This system demonstrates an efficient linked list-based booking system. It can be adapted for use in the real world with a few minor modifications.

1. **Core code in C++**
2. #include<iostream>
3. #include<string>
4. using namespace std;
5. struct Node {
6. string data;
7. Node\* next;
8. Node\* prev;
9. Node(string val) {
10. data = val;
11. next = NULL;
12. prev = NULL;
13. }
14. };
15. class Train {
16. public:
17. int total\_seats\_avail = 10;
18. Node\* head;
19. Train() {
20. head = NULL;
21. }
22. Train(string v) {
23. head = new Node(v);
24. }
26. void show\_total\_seats(){
27. cout<<"\nTotal seats available = "<<total\_seats\_avail;
28. }
30. void append(string val) {
31. if(total\_seats\_avail == 0){
32. cout<<"\nAll seats are booked\n";
33. return;
34. }
35. Node\* newN = new Node(val);
36. if (head == NULL) {
37. head = newN;
38. total\_seats\_avail--;
39. return;
40. }
41. Node\* temp = head;
42. while (temp->next != NULL)
43. temp = temp->next;
44. temp->next = newN;
45. newN->prev = temp;
46. total\_seats\_avail--;
47. }
48. void prepend(string val) {
49. Node\* newN = new Node(val);
50. if (head != NULL) {
51. head->prev = newN;
52. newN->next = head;
53. }
54. head = newN;
55. }
56. void del(string val) {
57. if (head == NULL) {
58. cout << "TRAIN SEAT LIST IS EMPTY !!\n";
59. return;
60. }
61. if (head->data == val) {
62. head = head->next;
63. if (head != NULL)
64. head->prev = NULL;
65. cout<<"\n\nBOOKING CANCELLED\n\n";
66. return;
67. }
68. Node\* temp = head;
69. while (temp != NULL && temp->data != val)
70. temp = temp->next;
71. if (temp != NULL) {
72. if (temp->prev != NULL)
73. temp->prev->next = temp->next;
74. if (temp->next != NULL)
75. temp->next->prev = temp->prev;
76. cout<<"\n\nBOOKING CANCELLED\n\n";
77. } else {
78. cout << "\n\nNAME DOES NOT EXIST IN THE SEATING LIST !!\n";
79. }
80. }
81. void print() {
82. Node\* temp = head;
83. while (temp != NULL) {
84. cout << temp->data;
85. if (temp->next != NULL) {
86. cout << "\t->\t";
87. }
88. temp = temp->next;
89. }
90. for(int i = total\_seats\_avail; i>0; i--){
91. if(i == 1){
92. cout<<"->\tEmpty\n";
93. }
94. else{
95. cout<<"->\tEmpty\t";
96. }
97. }
98. }
99. };
100. int main() {
101. Train TT;
102. int choice;
103. char choice2;
104. string name;
105. do {
106. cout << "\n1. Add Booking\n"
107. << "2. Cancel Booking\n"
108. << "3. Exit\n";
109. cin >> choice;
110. switch (choice) {
111. case 1: {
113. cout<<"\nPeshawar\n"
114. <<"Time =>  14 : 02";
115. TT.show\_total\_seats();
117. cout<<"\nWould you like to book your seat in this train ?\n"; cin>>choice2;
118. if(choice2 == 'y' || choice2 == 'Y'){
119. cout<<"\nEnter your name: "; cin>>name;
120. TT.append(name);
121. cout<<"\nSeat Added !!!\n\n";
122. TT.print();
123. }
124. break;
125. }
127. case 2:{
128. cout<<"\nPeshawar\n"
129. <<"Time =>  14 : 02";
130. TT.show\_total\_seats();
131. cout<<"\nWould you like to cancel your seat in this train ?\n"; cin>>choice2;
132. if(choice2 == 'y' || choice2 == 'Y'){
133. cout<<"\nEnter your name: "; cin.ignore(); cin>>name;
134. TT.del(name);
135. TT.print();
136. }
137. break;
138. }
140. } }while (choice != 3);
141. }