

```

1 // Implementation can be found at 'LinkedList.cpp'.
2 #ifndef __LinkedList__
3 #define __LinkedList__
4
5
6 #pragma region Inclusions
7 // This is a temporary solution for now, this inclusion must be resolved before ↗
8 // final submission!
9 //TODO Fix inclusion at header; std::string. LinkList.h >> LinkList.cpp
10 #include <string>
11 #pragma endregion
12
13 // Node Methodology:
14 // {Reservation}
15 // =====
16 // | DATA FIELDS |
17 // | {Node} |
18 // | ----- |
19 // | Next Node --> |
20 // =====
21
22
23 struct Node
24 {
25     // Data within the node
26     std::string nameFirst; // Holds the passenger's first name
27     std::string nameLast; // Holds the passenger's last name
28     int passengerID; // Retains the passenger's ID
29     int reservationNum; // Retains the passenger's reservation number
30     int telephoneNum; // Holds the passenger's preferred phone number.
31     int seatNum; // Holds the passenger's seat number
32     std::string mealType; // What meal does the passenger want during the ↗
33     // flight?
34     bool checkedIn; // Whether or not the passenger has checked in for ↗
35     // the flight.
36
37     //----
38
39     // Next node
40     Node* next; // Point to the next node
41 };
42
43 // =====
44 // -----
45 // =====
46
47 struct Reservation
48 {
49     Node* head;

```

```
50
51     Reservation();
52
53     void MainMenu();                // Displays a list of functionalities available to the end-user.
54     void Instructions();            // Provides instructions to the end-user.
55     char PromptUser_MainMenu();    // Fetch the STDIN from the end-user [keyboard or emulated keystrokes only]
56     void EvaluateAndRun(char);      // Inspect the user's input and try to run the request
57     void ClearBuffer();             // Thrash the terminal buffer.
58
59
60     void Print_Passenger_List(Node*, bool);
61     void InsertNode(Node*);
62     void CreateNewNode(std::string, std::string, int, int, int, int, std::string);
63     int Autofill_List_Numbers(int);
64     std::string Autofill_List_MealChoice();
65     int GetSeatAvailable(int);
66     void Autofill_List();
67     std::string UserInput_String(bool);
68     int UserInput_Number(bool);
69     bool UserInput_Bool(bool);
70     std::string ManualCustomerAdd_MealChoice();
71     void ManualCustomerAdd();
72     bool Search(Node**, Node**, int, std::string, int);
73     void FindPrintPassenger();
74     bool delete_node();
75     void UpdatePassengerInformation();
76     void CheckInPassenger();
77     void Print_CheckIn_List();
78     void Print_Meal_List();
79     void Sort();
80     bool Alphabetize(std::string, std::string);
81     int ListSize();                // Evaluates the entire size of the list.
82 };
83 #endif // !__LinkedList__
84
```