**East West University**

**Department of CSE**

|  |  |
| --- | --- |
| **Project Report** | |
| **Project Title: Flight Management System Report** | |
| **GROUP MEMBERS:**  **Name: Sami Al Zabid**  **ID: 2022-1-60-103**  **Name: Rakib Hasan**  **ID: 2022-1-60-342**  **Name: Rahin Arefin Ahmed**  **ID: 2022-1-60-263**  **Name: Md. Anik Chowdhury**  **ID: 2022-1-60-248** | **Course Code: CSE246**  **Section: 06**  **Submitted To: Amit Mandal**  **Submission Date:08/06/24** |

**Flight Management System Report**

**Problem Statement**

The problem addressed in this project is the efficient management of a flight network between multiple cities. The system needs to handle various operations, including adding and removing flights, booking flights, finding the shortest path between cities, displaying all available flights, scheduling flights to maximize non-overlapping trips, and searching for the nearest available flight based on a specified departure time.

**Objective**

The objective of this project is to develop a comprehensive flight management system that can:

1. Add and remove flights in the network.
2. Book flights for users.
3. Find the shortest path between two cities based on the flight cost.
4. Display all available flights from a specified city.
5. Schedule flights to maximize the number of non-overlapping trips from a city.
6. Search for the nearest available flight from a specified departure time.

**Methodology**

**Flow Chart**

1. **Initialization**
   * Input: Number of cities
   * Create a flight network with the specified number of cities
2. **Display Menu**
   * Options: Add Flight, Remove Flight, Book Flight, Find Shortest Path, Display All Flights, Schedule Flights, Search for Nearest Flight, Exit
3. **Operations**
   * **Add Flight**
     + Input: Source, Destination, Cost, Duration, Departure Time
     + Action: Add the flight to the network
   * **Remove Flight**
     + Input: Source, Destination, Departure Time
     + Action: Remove the specified flight from the network
   * **Book Flight**
     + Input: Source, Destination, Departure Time
     + Action: Book the specified flight
   * **Find Shortest Path**
     + Input: Start City, End City
     + Action: Use Dijkstra’s algorithm to find the shortest path based on cost
   * **Display All Flights**
     + Input: City
     + Action: Display all available flights from the specified city
   * **Schedule Flights**
     + Input: City
     + Action: Schedule flights to maximize the number of non-overlapping trips
   * **Search for Nearest Flight**
     + Input: City, Desired Departure Time
     + Action: Search for the nearest available flight after the specified time
   * **Exit**
     + Action: Exit the program

**Limitations**

1. The system currently handles only flight management operations and does not include other functionalities such as passenger management or flight status tracking.
2. The shortest path algorithm used (Dijkstra’s algorithm) is based on the cost of flights only and does not consider other factors such as duration or number of stops.
3. The system assumes that the input data (e.g., city IDs, flight details) are provided correctly and does not include extensive validation checks.

**Results**

The flight management system successfully handles various operations related to flight management between cities. It allows users to add and remove flights, book flights, find the shortest path based on cost, display available flights, schedule non-overlapping flights, and search for the nearest available flight after a specified departure time.

**Example Outputs**

1. **Add Flight**
   * Input: source = 1, dest = 2, cost = 300, duration = 1.5, deparTime = 10.00
   * Output: Flight added successfully.
2. **Remove Flight**
   * Input: source = 1, dest = 2, deparTime = 10.00
   * Output: Flight removed successfully.
3. **Book Flight**
   * Input: source = 1, dest = 2, deparTime = 10.00
   * Output: Flight booked successfully.
4. **Find Shortest Path**
   * Input: start = 1, end = 3
   * Output: Shortest path from city 1 to city 3: 1 -> 2 -> 3
5. **Display All Flights**
   * Input: city = 1
   * Output:  
     All flights from city 1:  
     1. To city 2, Cost: 300, Duration: 1.5, Departure time: 10.00  
     2. To city 3, Cost: 400, Duration: 2.0, Departure time: 12.00
6. **Schedule Flights**

* Input: city = 1
* Output:  
                Scheduled flights from city 1:  
                To city 2, Cost: 300, Duration: 1.5, Departure time: 10.00  
                To city 3, Cost: 400, Duration: 2.0, Departure time: 12.00
* **10. Search for Nearest Flight**
  + Input: city = 1, desiredTime = 11.00
  + Output:
  + Nearest available flight departs at 12.00 to city 3 with cost 400 and duration 2.0.

**Conclusion**

The developed flight management system effectively manages flight operations between cities, providing users with the ability to add, remove, and book flights, find the shortest path, display available flights, schedule non-overlapping flights, and search for the nearest available flight based on departure time. The system is efficient and user-friendly, making it a valuable tool for managing flight networks. Future improvements could include adding more functionalities such as passenger management, real-time flight status updates, and more advanced pathfinding algorithms that consider multiple factors.