*DEEPAK ATHIPAN A M B*

*SANJAY G*

*JEFFREY SAMUEL S*

*SHRISH R*

*Project Report*

*Airline Management and Tracking System*

ABSTRACT

Every day millions of people use flights as a mode of travel from one place to another. Millions of tickets are booked and people use connecting flights to travel from one place to another. Thus, to satisfy airline and user requirements a sufficient airline management system is required to manage the user requests and track the flight paths accordingly. Airline Management and Tracking System is a computerized system used to store and retrieve information and conduct transactions related to air travel. The project is aimed at exposing the relevance and importance of airline reservation Systems. It is projected towards enhancing the relationship between customers and airline agencies through the use of ARSs, and thereby making it convenient for the customers to book the flights as when they require such that they can utilize this software to make reservations. Data abstraction and other object oriented programming concepts can be used to securely manage transactions and graph algorithms help to track the flight paths from one city to another.

INTRODUCTION

Overview:

Airline reservation systems incorporate airline schedules, passenger reservations and ticket records. The project is aimed at exposing the relevance and importance of airline reservation Systems. It is projected towards enhancing the relationship between customers and airline agencies.

This software has two parts. First is user part and the administrator part. The user part allows booking, cancelling, viewing and modifying of tickets while at the same time viewing all the available flight paths and track the connecting flights. The admin part allows the admin to add new flight paths from one city to another. The main purpose of this software is to reduce the manual errors involved in the airline reservation process and make it convenient for the customers to book the flights as when they require it.

Objectives:

The following below are the objectives of the developers that may help the proposed project to take advantage to the existing system:

* To Develop Flight Reservation System using C++ Concepts like Class-Objects, Inheritance, Polymorphism.
* To Implement Graph Algorithm like Dijkstra’s Algorithm to find shortest path between two destinations.
* Provide a ticketless flight travel using the proposed project ticketing system feature
* Provide the proposed project the capability to book different kind of passengers according to the required class.

Purpose:

The purpose of this project is to implement or to design a database for an airline reservation system to check the flight details, book and cancel flight tickets. It makes the process of booking and cancelling flight tickets simple and easy for the passengers. The tracking system helps the user to identify the flight paths available and allows the admin to add more flight paths when necessary.

Problem Definition:

Normally a person wants to reserve his ticket and he has to contact at nearest Overseas Travels branch. The Airline Reservation System provides an interface to schedule flights and reservations for an airline through internet. Its responsibility is to keep track of system users, customers, flight information and cancellation.

Project Plan and Scope:

Airline Reservation System is one the modifications that were carried out in the Passenger Service System so that the working and availability of Service area can be broadened. On one hand, it helps the customers and on the other, it also makes the life of the airline service companies easier by keeping all the records of the passengers and if there is any change in the fight due to some reason, the passengers are promptly informed. This system is also used by companies to keep track of user preferences of regular travellers so that they can provide better service and give offers to customers.

This can be implemented in real-time flight reservation system to

* Remove and Add cities
* Track the flights
* Connect the flight paths
* Maintain User Database

Implementation

Object Oriented Concepts:

Data Abstraction:

Database systems comprise complex data-structures. In order to make the system efficient in terms of retrieval of data, and reduce complexity in terms of usability of users, developers use abstraction i.e. hide irrelevant details from the users. This approach simplifies database design.

Data Hiding:

Data hiding is a technique of hiding internal object details, i.e., data members. It is an object-oriented programming technique. Data hiding ensures, or we can say guarantees to restrict the data access to class members. It maintains data integrity. Data hiding means hiding the internal data within the class to prevent its direct access from outside the class.

Inheritance:

The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming.

Inheritance is a feature or a process in which, new classes are created from the existing classes. The new class created is called “derived class” or “child class” and the existing class is known as the “base class” or “parent class”. The derived class now is said to be inherited from the base class.

When we say derived class inherits the base class, it means, the derived class inherits all the properties of the base class, without changing the properties of base class and may add new features to its own. These new features in the derived class will not affect the base class. The derived class is the specialized class for the base class.

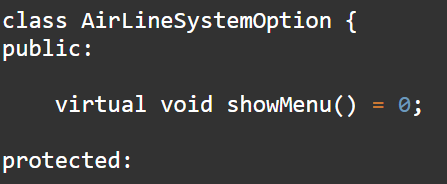
Polymorphism:

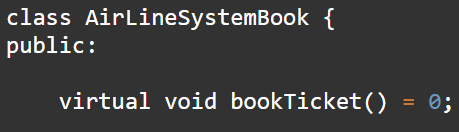
Run Time Polymorphism:

Runtime polymorphism is also known as dynamic polymorphism or late binding. In runtime polymorphism, the function call is resolved at run time.

Virtual function:

A virtual function is a special type of function that, when called, resolves to the most-derived version of the function that exists between the base and derived class. This capability is known as polymorphism.





Object Concepts:

Standard Template Library:

The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators. It is a generalized library and so, its components are parameterized. Working knowledge of template classes is a prerequisite for working with STL.

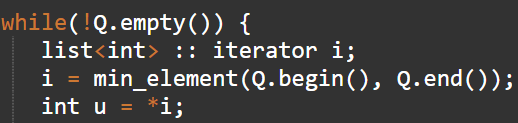
STL has 4 components:

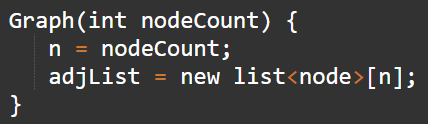
. Algorithms

. Containers

. Functions

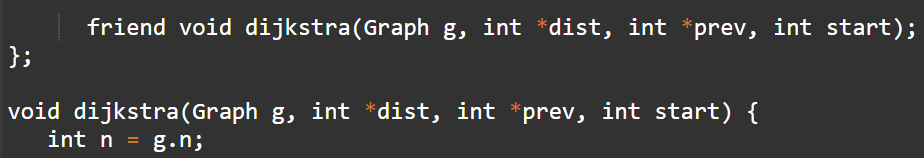
. Iterators





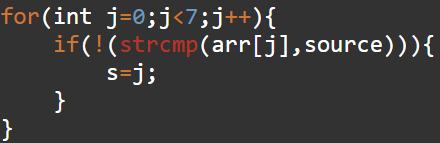
Friend Function:

In object-oriented programming, a friend function, that is a "friend" of a given class, is a function that is given the same access as methods to private and protected data. A friend function is declared by the class that is granting access, so friend functions are part of the class interface, like methods.



String Handling:

C++ programming language provides a set of pre-defined functions called string handling functions to work with string values. The string handling functions are defined in a header file called “string”. Whenever we want to use any string handling function we must include the header file called string.



Dijkstra’s Algorithm:

With Dijkstra's Algorithm, you can find the shortest path between nodes in a graph. Particularly, you can find the shortest path from a node (called the "source node") to all other nodes in the graph, producing a shortest-path tree.

This algorithm is used in GPS devices to find the shortest path between the current location and the destination. It has broad applications in industry, specially in domains that require modelling networks.

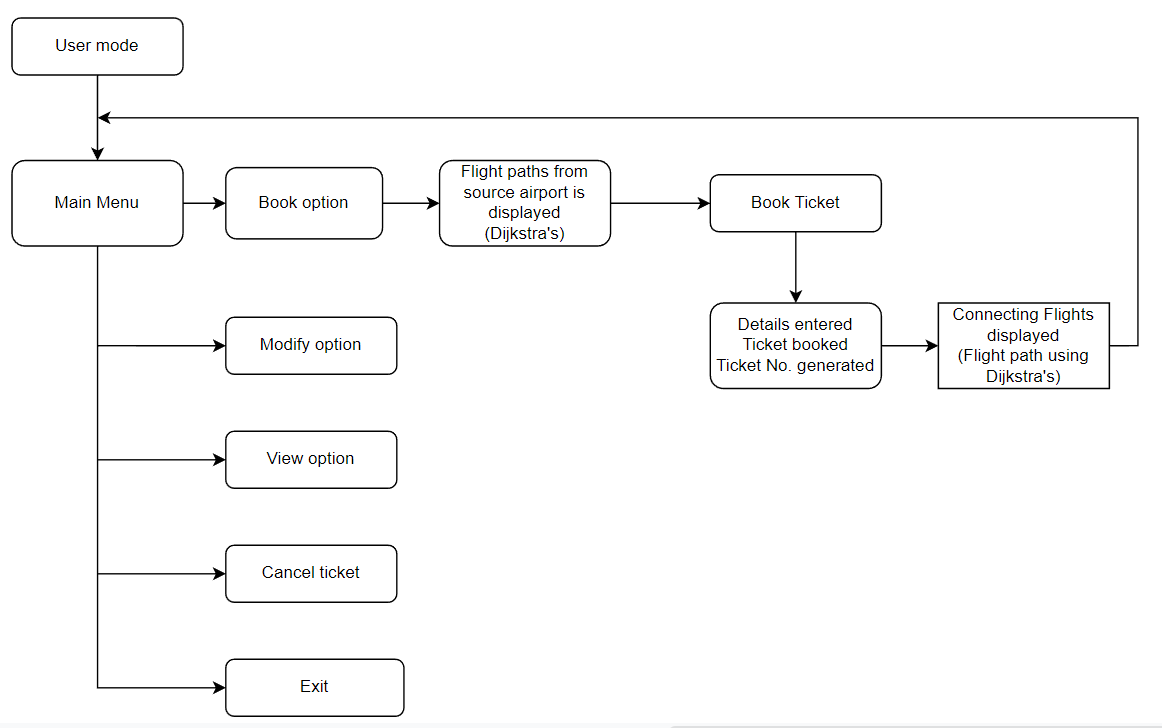
Adjacency List:

An adjacency list represents a graph as an array of linked lists. The index of the array represents a vertex and each element in its linked list represents the other vertices that form an edge with the vertex.

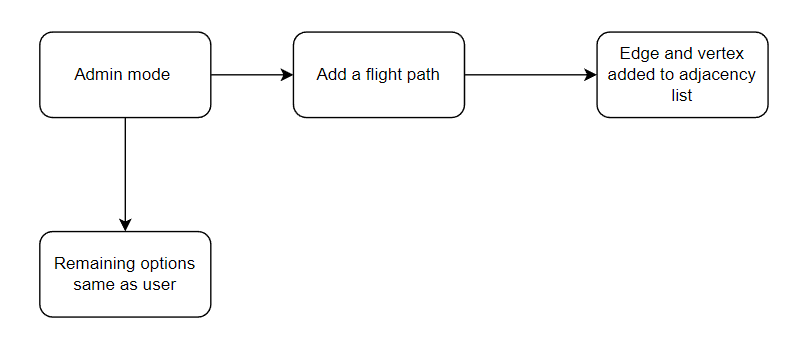


Flow Diagram

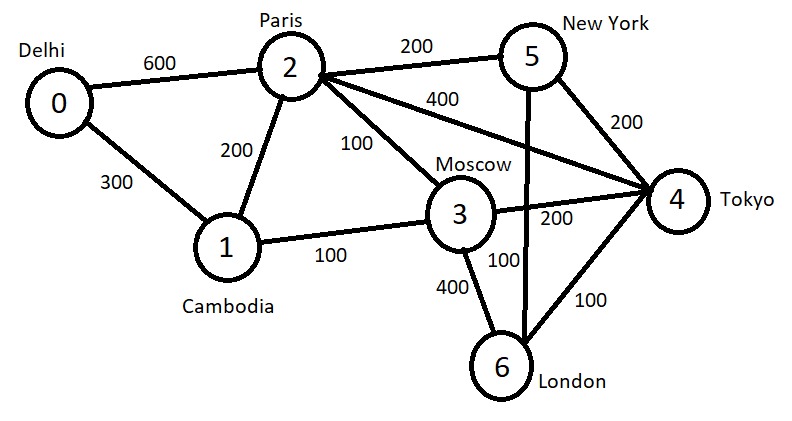
User:



Admin:

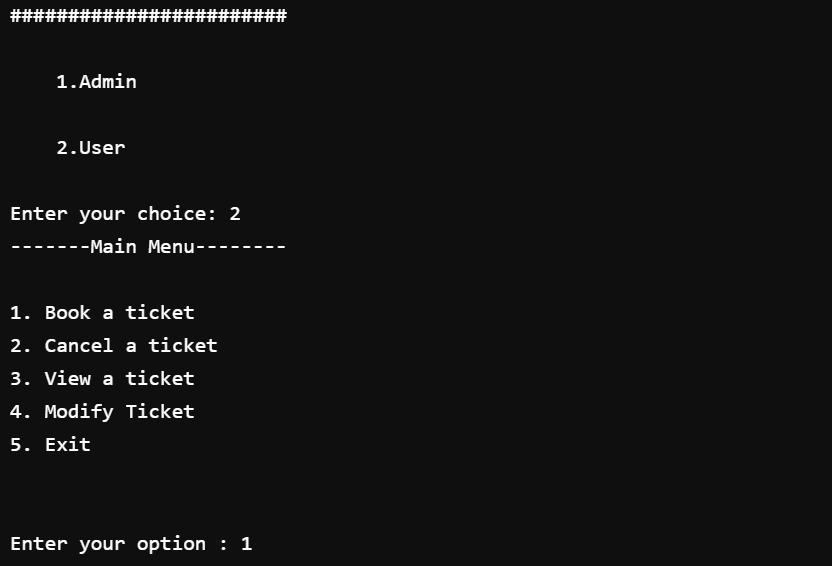


Route map:

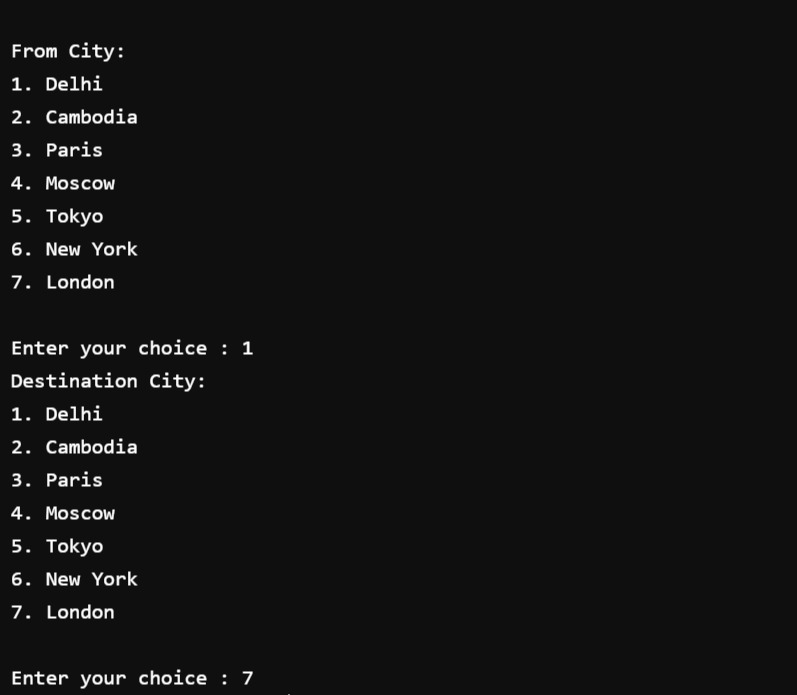


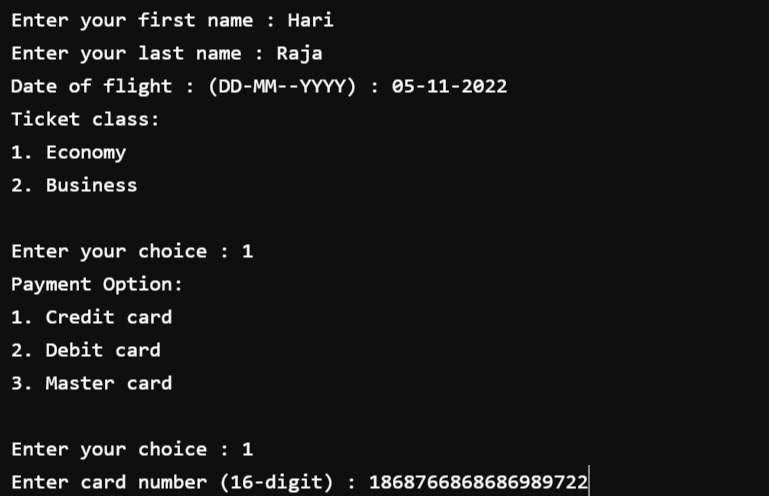
RESULTS:

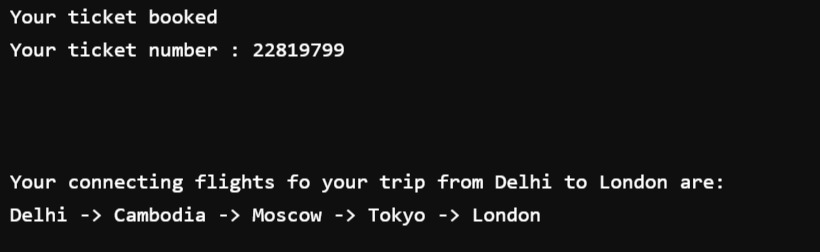
USER AND BOOKING OF TICKETS:



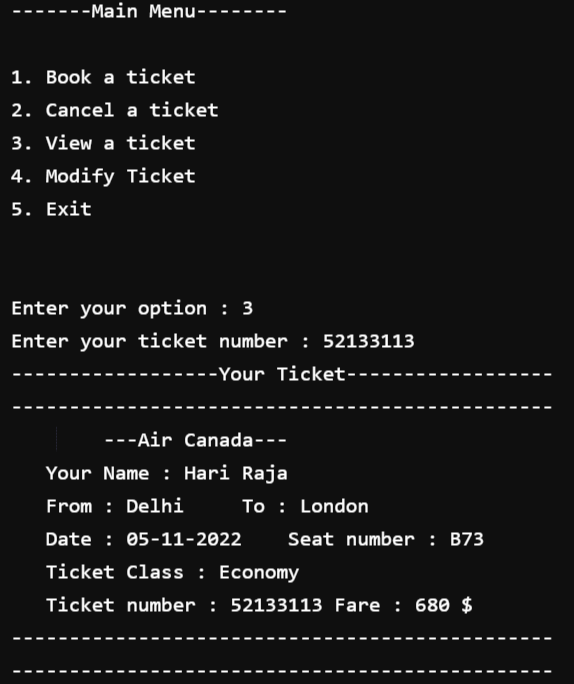




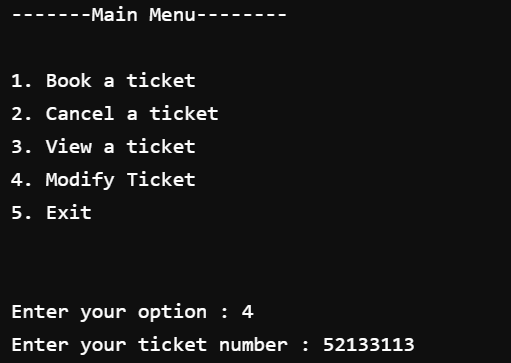


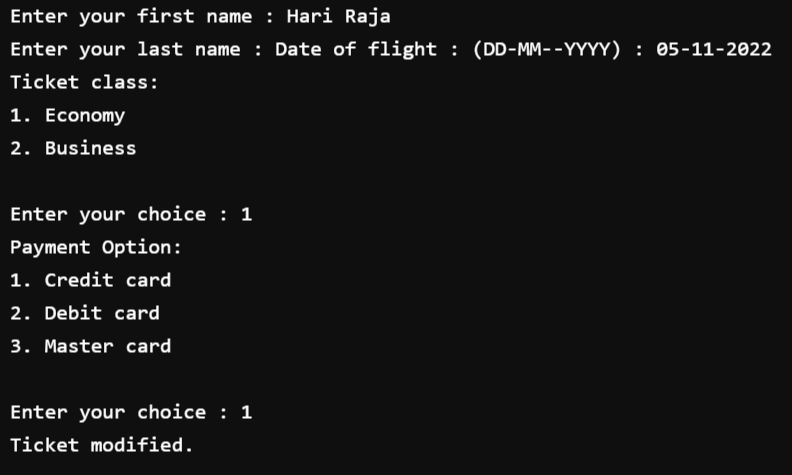


VIEW TICKET:

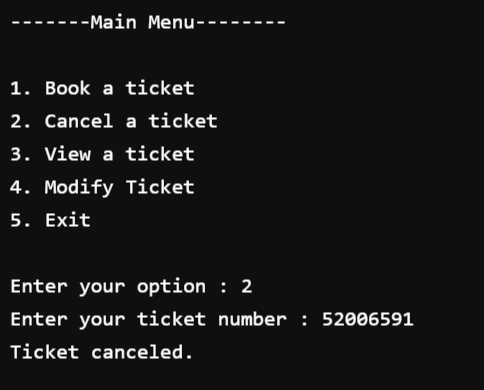


MODIFY TICKET:

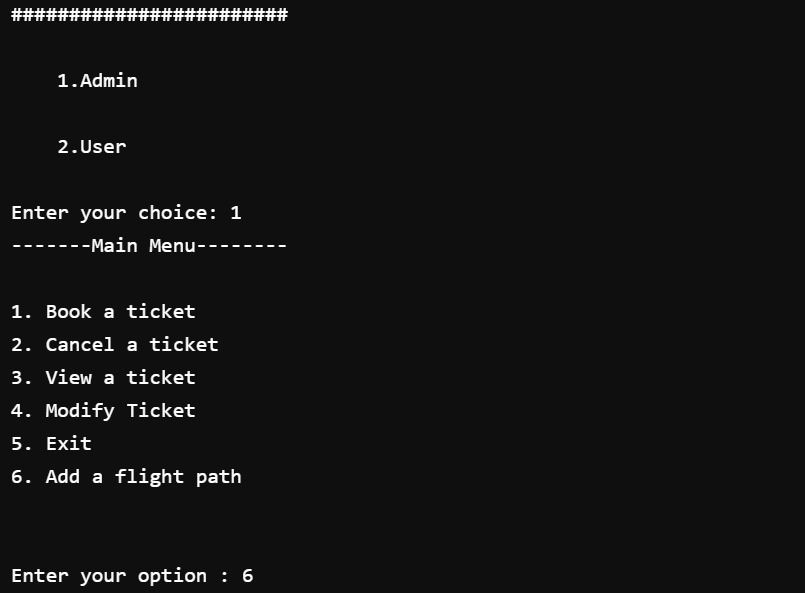


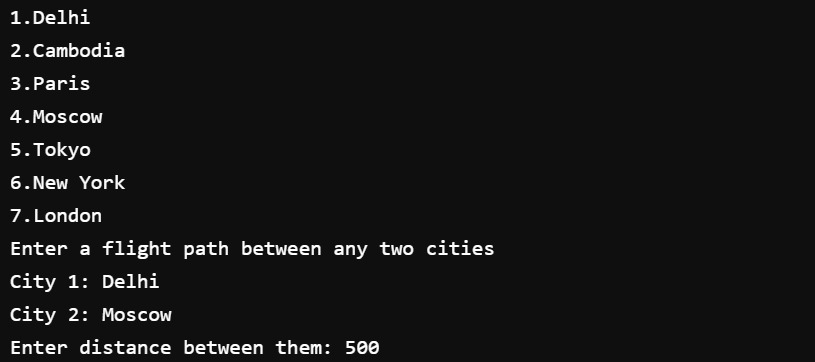


CANCEL TICKET:

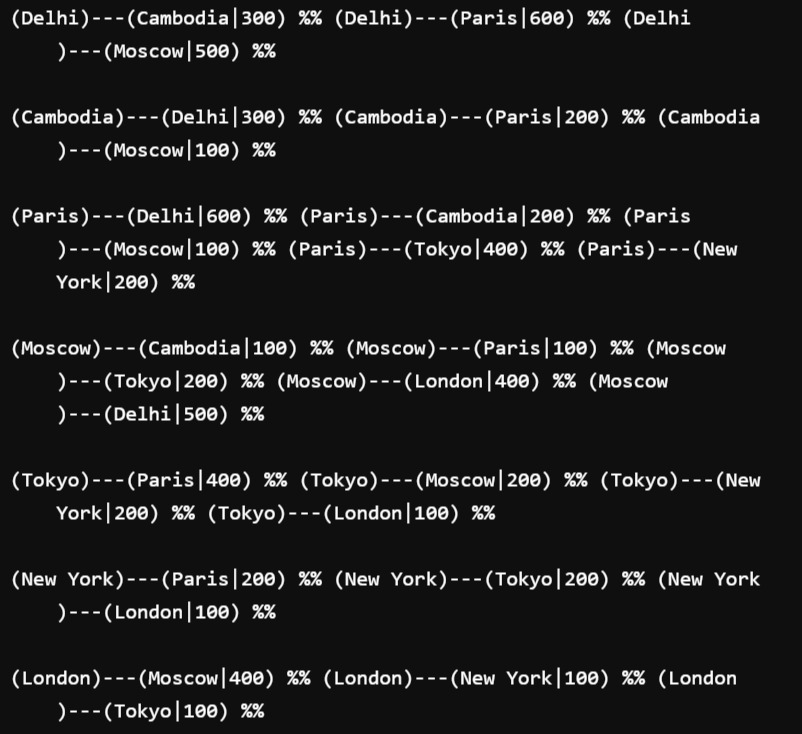


ADMIN:





NEW FLIGHT PATH IS ADDED:



Conclusion

Thus, a flight management and tracking system has been implemented successfully. Concepts like data abstraction, data hiding, inheritance polymorphism has been used. User can easily book tickets and know their flight paths accurately. The flight management system has enabled users to modify and cancel their reservation as well. The system can be implemented in real time problems including land-based transportation. The flight management system can be expanded by adding more cities and flight paths can be added.