Airport Ticket Booking

1. Description:

- There will be a home city and list of 10 other cities which have flights to and from the home city
- There will be 5 airlines and multiple flights under each airline operating between different cities. Each airline and flight should have their unique codes (eg. 6E 144 may be an Indigo flight between Ahmedabad and Kolkata, SG 890 might be a Spicejet flight between Delhi and Mumbai, etc.)
- A customer should be able to see a list of flights by giving a source and destination city , date of departure, number of seats required, etc. The list of available flights should be displayed to the passenger and the passenger should be able to select one flight, see its details (eg. flight duration, departure time, fare, etc) and then finalize the ticket on a flight. There should be an option to enter passenger details such as name, Aadhar number, age, etc.
- There should be another related application which deals with the check-in of passengers at the airport. It should have features to verify customer identity (eg. by giving an Aadhar Card number and comparing it with that given at the time of booking). There should be a system of handing out seats to the passengers based on the remaining seats in the flight (eg. with row number and seat position like 16D, 16E, 16F, etc). Once a seat is given, it should not be given to another passenger.
- The check-in system should also charge passengers based on the weight of their check-in baggage and finally print out a boarding pass with details as found on a regular pass on a commercial airline.

2. Class Design:

Here is the list of the basic classes, interfaces and methods one should implement -

- *City*: an imaginary city which has an airport and also maintains the distance between other cities
- *Airline*: a commercial airline which has flights running between cities. Should maintain a list of flights that it operates and between which cities
- *Flight:* a flight belonging to a particular airline and operating between a pair of cities. Should store information about departure, arrival, running days, fare, etc.
- *Staff:* A staff of the airline who should have access to the check-in system
- *Customer:* the person who books a ticket on a flight. Should have attributes such as name, Aadhar number, age, gender, etc.
- *Booking*: a record of a booking confirmed on a flight for a particular day for a customer. Invent necessary information.
- *Boarding pass*: A final copy of the booking information with additional details such as baggage charges, gate number, etc.

3. Sample input and outputs:

- An example run of the application can be as follows:

  Assumption: the cities and distances from other cities, flights, etc are already initialized and stored in some way (eg. a text file).

  A customer opens the application, creates an account (or logs into it). Based on the specifications in the description above, the customer books a flight. This booking data should also be appropriately stored. In a later run of the application, the user logs in using the admin/staff account. This session checks-in the customer (in the previous run) to their flight and carries out the operations as per the description. You can invent necessary details.