FLIGHT BOOKING SYSTEM

Paritosh Tiwari

MIS 6382 OOPS with Python

Within the code for flight management system, I have performed the functionality of booking flights, cancelling and rescheduling the flights. I have also maintained the data of the passengers travelling in the flight.

To achieve this, I have created classes such as

Flight: Flight class which consists of only a constructor to initialize the values and display flight method which displays the flight information

```
class Flight:
    def __init__(self, id, airline, fromDestination, toDestination, departTime, arrivalTime, totalSeats, cost):
        self.id = id
        self.airline = airline
        self.fromDestination = fromDestination
        self.toDestination = toDestination
        self.departTime = departTime
        self.arrivalTime = arrivalTime
        self.totalSeats = totalSeats
        self.cost = cost
        self.availableSeats = totalSeats
        self.passen = []

    def displayFlight(self):
        return f"==================Flight Info==================\nId: {self.id}\nAirline: {self.airline}\nFrom: {self.from
```

The result of the display flight is displayed below.

```
==================Flight Info==================
Id: 1
Airline: Southwest
From: Dallas
To: Chicago
Departure Time: 6:00 AM
Arrival Time: 8:00 AM
Available Seats: 249
Cost: 100
```

To maintain the data for the passenger I have created a passenger class, which initializes as name, flights booked and amount due. It contains the information as per passenger because if we need to cancel or reschedule the flights, the passenger object can be used.

```
class Passenger:
    def __init__(self, name, flight = None, amtDue = None):
        self.name = name
        self.flight = flight
        self.amtDue = amtDue
```

The main business logic is in reservation class as it contains the information about the flights and passengers and contains methods like book flight, cancel flight, and print ticket.

```
class Reservations:
    def __init__(self, passenger, flight):
        self.passenger = passenger
        self.flight = flight

    def bookFlight(self):
        self.flight.availableSeats -= 1
        if self.passenger.amtDue is None:
            self.passenger.amtDue = 0

        self.passenger.amtDue += self.flight.cost
        self.flight.passen.append(self.passenger)
        print(f"Flight booked for {self.passenger.name}")

    def cancelFlight(self):
        self.flight.availableSeats += 1
        self.passenger.amtDue -= self.flight.cost
        self.flight.passen = [item for item in self.flight.passen if item.name != self.passenger.name]
        print("Flight cancelled")

    def printTicket(self):
        return f"================Ticket=================\nPassenger Name: {self.passenger.name}\nAirline: {self.flight.ai
```

After handling the edge case scenarios for the flight reservation systems, I have provided user with the option to book, cancel and reschedule.

The flow of the reservation system is, once the user arrives to the booking system, they are asked if they want to book, cancel or reschedule the tickets, based on their choices, a list of flights is displayed.

```
What do you want to do? (Book/Cancel/Reschedule): book
What is your name?: paritosh
From where do you want to book flight?: dallas
To where do you want to book flight?: chicago
```

Once the list of flights have been displayed to the user, user is allowed to select the id of the flight they want to book, based on the selection, the flight and passenger object is created and sent to reservations object to perform the operations.

Once this process of selecting flight is completed, the flight is booked, the amount due is updated on passenger and the available seats are decreased from the selected flight.

```
==================Flight Info==================
Id: 1
Airline: Southwest
From: Dallas
To: Chicago
Departure Time: 6:00 AM
Arrival Time: 8:00 AM
Available Seats: 250
Cost: 100
==================Flight Info==================
Id: 3
Airline: Spirit
From: Dallas
To: Chicago
Departure Time: 11:00 AM
Arrival Time: 1:00 PM
Available Seats: 90
Cost: 125
Please select the id of the flight: 1
```

And then the user is asked to check if they want to perform further operations.

For the cancel workflow, the name of the user is taken as a input, then through the name, all the flights are checked and displayed to user for cancellation, and user has been given an option to choose the flight they want to cancel. Once the user selects the id of the flight they want to cancel, the cancel operation is performed.

```
What do you want to do? (Book/Cancel/Reschedule): cancel
What is your name?: paritosh
The flight you booked is:
==================Flight Info==================
Id: 1
Airline: Southwest
From: Dallas
To: Chicago
Departure Time: 6:00 AM
Arrival Time: 8:00 AM
Available Seats: 249
Cost: 100
To confirm, enter the id?: 1
Flight cancelled
```

Reschedule workflow follows reusability of code as I already have implemented the cancellation and booking, so what I have done is while providing the options for the user to cancel the flights, On choosing of flight to cancel, I am saving the information of cancelled flight and adding in the

logic to book the flight, making changes to flights availability and updating the amount due for the passenger

```
What do you want to do? (Book/Cancel/Reschedule): reschedule
What is your name?: paritosh
The flight you booked is:
=================Flight Info==================
Id: 1
Airline: Southwest
From: Dallas
To: Chicago
Departure Time: 6:00 AM
Arrival Time: 8:00 AM
Available Seats: 249
Cost: 100
To confirm, enter the id?: 1
Flight cancelled
=================Flight Info==================
Id: 1
Airline: Southwest
From: Dallas
```

As soon as a flight is cancelled for the user, the user is provided with the option of same route to book the flight. Once the user selects the flight, the flow is same as that of flight booking.