

Report: Design and Architecture of the Flight Booking System

1. System Overview:

The flight booking system is designed to allow users to book flights between different airports with various airlines. The system includes features for selecting flights, choosing meal options, and adding additional services. A graphical user interface (GUI) was implemented using Tkinter to provide a user-friendly experience.

2. Design and Architecture:

The system follows an object-oriented design, organized into several classes, each responsible for a specific aspect of the booking process. The key classes include:

- **Airport:** Represents an airport with a name.
- **Airline:** Represents an airline with a name.
- **Flight:** Represents a flight with attributes like flight ID, airline, departure and arrival airports, price, and time.
- **Customer:** Represents a customer with a name and email.
- **Booking:** Represents a booking, associating a customer with a flight, along with optional meal and additional services.
- **FlightBookingSystem:** Manages flights, meal options, and additional services, and provides methods for searching flights and booking them.

3. Application of OOP Concepts:

- **Encapsulation:**
 - Each class encapsulates its data and behavior. For instance, the Flight class encapsulates the flight details and provides a `__str__` method to present the flight information in a readable format.
- **Abstraction:**
 - The system hides the complexity of flight searching and booking behind simple interfaces. The FlightBookingSystem class provides methods like `add_flight`, `search_flights`, and `book_flight`, abstracting the internal workings.

- **Inheritance:**

- While inheritance isn't explicitly used in this system, the modular design allows for easy extension. For example, if different types of flights (e.g., domestic vs. international) were needed, inheritance could be applied by creating subclasses of Flight.

- **Polymorphism:**

- The system leverages polymorphism in the way it handles different airlines and services. The Flight class can work with any Airline object, regardless of the specific airline, allowing the system to handle multiple airlines seamlessly.

4. Relationships Between Classes:

- **Flight and Airline:**

- Each Flight object is associated with an Airline, representing the carrier of the flight. This association is established through composition, where a Flight has an Airline object.

- **Flight and Airport:**

- A Flight is also associated with two Airport objects, representing the departure and arrival locations.

- **Booking and Customer:**

- Each Booking object is linked to a Customer, representing the person making the booking.

- **Booking and Flight:**

- A Booking is also associated with a specific Flight, linking a customer to a particular flight.

5. GUI Implementation (Bonus):

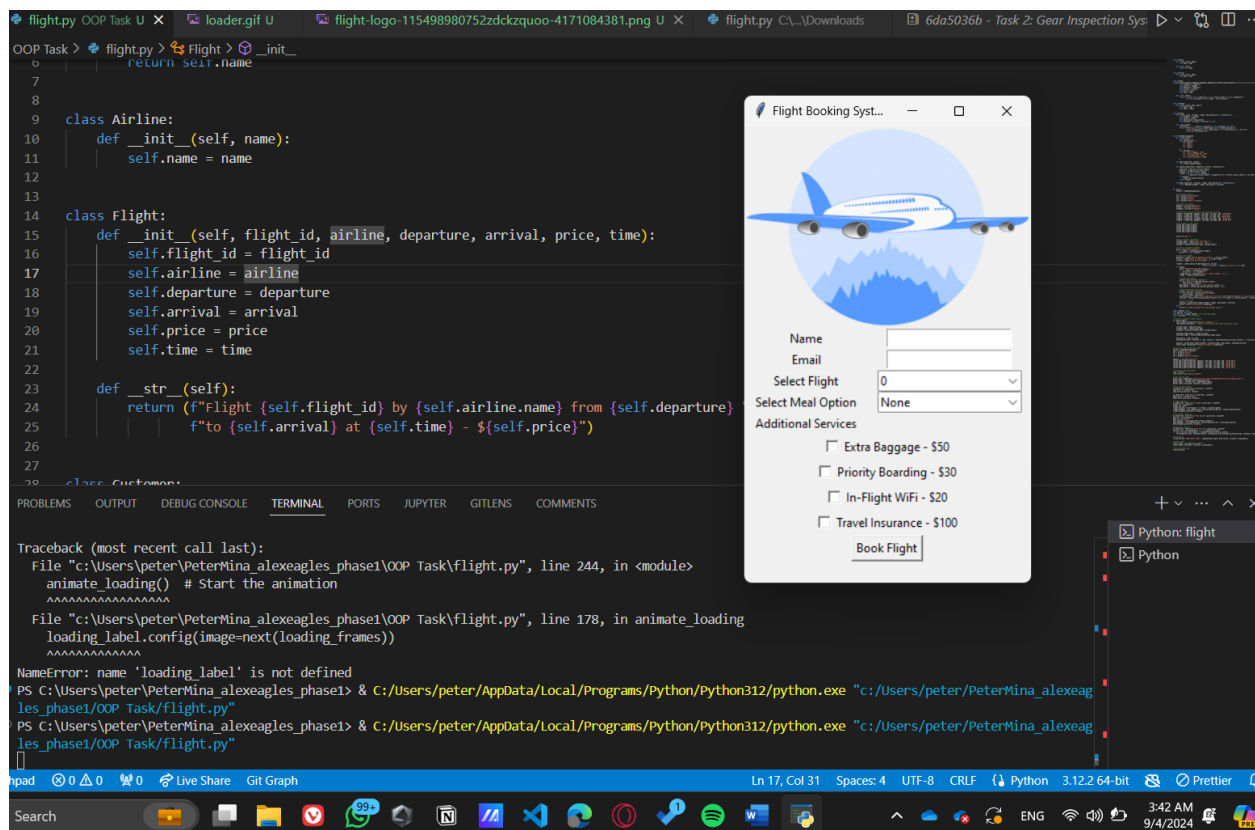
The GUI was implemented using the Tkinter library. It allows users to:

- Enter customer information (name and email).
- Select a flight from a dropdown menu.
- Choose a meal option from a dropdown.

- Select additional services using checkboxes.
- Book the flight and display the booking details.

GUI Implementation Screenshots

The flight booking system: includes a simple Tkinter interface. Users can input their name, email, select a flight, choose meal options, and add additional services. The interface is designed for ease of use with drop-downs and checkboxes.



Flight Booking Example: The system initializes multiple airports and airlines, creating various flight options. The booking interface allows a user to select Flight 104 from Borg Al Arab to Sharm El Sheikh, choosing Meal2 and additional services like Priority Boarding and In-Flight WiFi.

flight.py OOP Task U loader.gif U flight-logo-115498980752zdczquoo-4171084381.png U flight.py C:\...\Downloads 6da5036b - Task 2: Gear Inspection Sys

OOP Task > flight.py > FlightBookingSystem > _init_


```
81 def main():
82     system = FlightBookingSystem()
83
84     # Initialize airports
85     hba = Airport("Borg AlArab")
86     lxr = Airport("Luxor")
87     cai = Airport("Cairo")
88     ssh = Airport("Sharm El Sheikh")
89
90     # Initialize airlines
91     egyptair = Airline("EgyptAir")
92     Nileair = Airline("Nile Air")
93     flyegypt = Airline("FlyEgypt")
94
95     # Initialize available flights with different airlines
96     flight1 = Flight(101, egyptair, hba.name, cai.name, 500, "10:00 AM")
97     flight2 = Flight(102, Nileair, cai.name, lxr.name, 600, "12:00 PM")
98     flight3 = Flight(103, Nileair, cai.name, lxr.name, 450, "02:00 PM")
99     flight4 = Flight(104, flyegypt, hba.name, ssh.name, 400, "03:00 PM")
100    flight5 = Flight(105, egyptair, ssh.name, cai.name, 550, "05:00 PM")
101
102    # Add flights to the system
103    system.add_flight(flight1)
104    system.add_flight(flight2)
105    system.add_flight(flight3)
106    system.add_flight(flight4)
107    system.add_flight(flight5)
108
109
110    print("Welcome!!")
111
112    # Enter customer information
113    customer_name = input("Enter your name: ")
114    customer_email = input("Enter your email: ")
115    select_flight = input("Select flight number: ")
116    select_meal = input("Select meal option: ")
117    additional_services = []
118    while True:
119        print("Additional Services")
120        extra_baggage = input("Extra Baggage - $50 (Y/N): ")
121        priority_boarding = input("Priority Boarding - $30 (Y/N): ")
122        in_flight_wifi = input("In-Flight WiFi - $20 (Y/N): ")
123        travel_insurance = input("Travel Insurance - $100 (Y/N): ")
124        if extra_baggage == 'Y' or priority_boarding == 'Y' or in_flight_wifi == 'Y' or travel_insurance == 'Y':
125            additional_services.append(extra_baggage)
126            additional_services.append(priority_boarding)
127            additional_services.append(in_flight_wifi)
128            additional_services.append(travel_insurance)
129        book_flight = input("Book Flight (Y/N): ")
130        if book_flight == 'Y':
131            break
132
133    # Booking successful
134    booking_successful = True
135    booking_successful_message = "Booking successful: "
136    booking_successful_message += "Flight 104 by EgyptAir from Borg AlArab to Sharm El Sheikh at 03:00 PM - $400"
137    booking_successful_message += "Meal: Meal2"
138    booking_successful_message += "Services: Priority Boarding - $30, In-Flight WiFi - $20, Travel Insurance - $100"
139    print(booking_successful_message)
```

flights

Aa .ab,* 1 of 23

↑ ↓ ≡ ×

Flight Booking System



Name

Peter

Email

petermina@gmail.com

Select Flight

3

Select Meal Option

Meal2

Additional Services

☐ Extra Baggage - \$50

☒ Priority Boarding - \$30

☒ In-Flight WiFi - \$20

☒ Travel Insurance - \$100

Book Flight

Booking successful:

Booking for Peter (petermina@gmail.com):

Flight 104 by EgyptAir from Borg AlArab to Sharm El Sheikh at 03:00 PM - \$400

Meal: Meal2

Services: Priority Boarding - \$30, In-Flight WiFi - \$20, Travel Insurance - \$100