THE UNIVERSITY
OF TEXAS AT DALLAS

MIS 6382.503 – Object-Oriented Programming in Python

Airline Reservation System

Submitted By:
Ishan Chandrahas Haldankar
Ned Id: ich220000

Acknowledgment

To begin with, I would like to express our gratitude to Prof. Vatsal Maru and Teaching Assistant Harshita Katakam for allowing me to complete a project and helping me understand the subject and concepts throughout the semester and coursework. Their enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas have helped me tremendously in our analysis and writing of this project. Their immense knowledge, profound experience, and professional expertise in object-oriented programming have enabled me to complete this project successfully. Without their support and guidance, this project would not have been possible.

Thank You

Introduction

The airline business depends on advanced reservation systems to manage the intricate logistics of flight bookings, schedule coordination, and other related tasks. The objective of this project was to create a rudimentary model for a system capable of managing customer reservations, cancellations, and modifications while also considering the limitations on flight seat availability.

Python software was developed to provide consumers with up to three flight possibilities. It enables users to modify or cancel an existing reservation as required and generates a booking confirmation that includes flight details and fare information. In addition, the system monitors the seating capacity of flights to ensure that bookings do not surpass the available seats.

The objective was to showcase the fundamental features of an actual airline booking system in a limited scope. The main criteria consisted of versatile consumer booking choices, seamless reservation adjustments, and adherence to availability and capacity limitations. The system underwent testing using sample scenarios and usage cases to confirm its capability to replicate fundamental activities performed by airlines and travel websites.

This report comprehensively summarizes the methodologies, data structures, and programming principles in constructing this prototype. The text delves into the methods used to fulfill project specifications, scrutinizes the implementation of Python code, and deliberates on the results obtained from testing. This study aims to provide and elucidate the development process for a straightforward yet authentic model of an airline reservation platform.

Method

1. **First Class:**

   - The Flight class is defined as an object representing a specific flight, including its capacity, number of available seats, and ticket (the price per seat).
   - The __init__ method initializes the flight by assigning its stated capacity, setting the starting number of available seats to the total capacity, and determining the pricing.
   - The function check_availability provides the count of unoccupied seats.
   - The function book_seat enables reserving a specific quantity of seats if there are sufficient seats available. It refreshes the number of available seats and provides a status indicating success.
   - The cancel_booking function enables the cancellation of a specified number of tickets, provided the number is legitimate. The function refreshes the number of available seats and provides a status indicating success.
   - The function "amount_due" computes the total amount that must be paid for a given number of seats, considering the fare.

2. **Display Options Function:**

   - The display_options function outputs the flight choices that are currently available, along with their corresponding capacities and rates.

3. **Main Block:**

   Within the block labeled "if __name__ == "__main__":

   - Three instances of the Flight class are instantiated to represent three distinct flights (Flight A, B, and C) with varying capacities and pricing.

- The application operates continuously, presenting several flight choices to the user and requesting their selection or termination.
- The user's input is analyzed to ascertain the chosen flight or the decision to exit.
- When a legitimate flight option is chosen, details regarding the selected flight and the number of seats available are shown.
- The user is presented with the option to select among booking, canceling, or returning.
- Upon selecting the option to book, the user is prompted to specify the quantity of seats they wish to reserve. The total amount owed will be presented if the booking process is completed successfully.
- Should the user opt for cancellation, they will be prompted to provide the number of seats to be canceled. Upon successful completion, the cancellation will be verified.

Result

1. **Flight options are initially shown.**

   - The program initiates by presenting the flight choices, including information such as flight ID, cost, and vacant seats.

2. **User Chooses Flight A (Option 1):**

   - The user chooses Flight A (Option 1) and is notified that are 50 tickets available.
   - The user selects a reservation for 2 seats, and the system verifies the booking with a total payment of $200.

3. **Updated Display After Booking:**

   The system refreshes the presentation of the available flight choices, displaying the up-to-date status following the reservation. Flight A currently has 48 seats that are now unoccupied and available for booking.

Summary of the Project:

The Airline Reservation System project is a Python implementation that offers users a straightforward and interactive interface to book and oversee flights. The system functions via a command-line interface, providing a variety of capabilities such as accessing flight options, reserving seats, canceling reservations, and displaying the total cost of bookings.

Key Features:

1. Class of the flight:

   The project includes a Flight class that represents specific flights, with properties such as capacity, available seats, and pricing.

   The class contains methods that allow checking the availability of seats, booking seats, canceling bookings, and computing the total amount payable for reservations.

2. User Engagement:

   Users are provided with a comprehensive list of flight possibilities, where each option includes essential information such as flight ID, price, and the number of seats available.

   The program assists users in navigating the procedure of choosing a flight, indicating the number of seats to reserve or cancel, and presenting pertinent details.

3. Real-time updates:

   The system continuously changes the number of available seats upon each booking or cancellation, guaranteeing immediate feedback for consumers.

4. Intuitive User Interface:

The command-line interface is intentionally designed to be user-friendly, incorporating explicit prompts, input validation, and valuable messages to navigate users through the reservation process effectively.

5. Adaptable Reservation Choices:

   Users can select from various actions during the interaction, including reserving seats, canceling reservations, or leaving the system.