06/11/2023

Python Lab (CSC 313)

Final Lab Test

10 marks

Q:

Airplane Ticket and Food Booking System

Class Structures:

class **Passenger**:

Attributes:

passenger_id (unique identifier for a passenger)

name

email

Methods:

get_info() : returns passenger information.

class **Flight**:

Attributes:

flight_id (unique identifier for a flight)

departure

destination

available_seats

ticket_price

Methods:

check_availability() : returns the number of available seats.

book_ticket(passenger, seats) : books the specified number of seats for the passenger.

get_ticket_price() : returns the ticket price.

class **FoodMenu**:

Attributes:

menu_id (unique identifier for a menu)

items (list of available food items)

prices (corresponding prices for food items)

Methods:

display_menu() : prints the available food items and their prices.

Database Interaction:

Utilize a database to store and retrieve information related to flights, passengers, and food orders. You can use any database system of your choice (e.g., SQLite, MySQL, or others). Ensure the following functionalities are implemented:

1. Storing and retrieving flight information (flight_id, available_seats, ticket_price, etc.).
2. Storing and retrieving passenger details.
3. Storing and retrieving food menu items and their prices.

Problem Requirements:

1. Create classes for Passenger, Flight, and FoodMenu with appropriate attributes and methods.
2. Implement the database functionalities to interact with the system to store and retrieve data.
3. Develop a method to allow a passenger to book a ticket for a chosen flight.
4. Implement a method for a passenger to select and order food items.
5. Ensure that the system manages the available seats on flights and updates them accordingly.
6. Ensure the food orders are associated with the passenger's ticket.

Evaluation:

1. Create instances of flights, passengers, and the food menu.
2. Simulate interactions such as booking tickets for flights and placing food orders for passengers.
3. Retrieve and display the booked tickets, passenger information, and food orders.

Constraints:

1. Implement the solution using object-oriented programming paradigms.
2. Utilize database operations for data storage and retrieval.
3. Ensure proper error handling for cases like invalid inputs or unavailable seats.

Your task is to implement the system as described, focusing on modularity, clarity, and effective database interactions. Ensure the system covers booking flights, handling passenger details, and managing food orders.