Advanced Object Oriented Programming Final Assignment

(Group size : 1 to 4 people)

Due: **Friday** 3rd December 2021 at 11:30 pm on Blackboard

Instructions:

Your group must develop a Flight Reservation system with a complete graphical user interface. You must also implement appropriate error checking and messages for all fields in your system. Please note the marking scheme below.

Main points:

* It must adhere to the “Eight Golden Rules” of Interface design. Covered in class and link included in the assignment description on blackboard.
* It must be a professional application that a travel agency or airline will be willing to buy.
* It must have a consistent interface following the guidelines that will be covered in lectures and labs.
* It must have error checks for field input and must be basically crash proof.

Submission Instructions:

1. At the top of you main code you must comment the name and id numbers of all group members.
2. Upload to blackboard your **complete solution compressed as a zip file** (you may submit as many times as you wish). Only the last submission will be marked.
3. You must also submit a Microsoft word document with your classes copied and pasted neatly into it.
4. **Only ONE member from each group must upload the solution zip file.**

|  |  |
| --- | --- |
| Core UI Design Principles(menu and layout consistency, readability, ease of use, etc) -15% |  |
| Design and Functionality –. **The design is a correct Object Oriented one taking all principles into account** and all operations work as expected -40% | NOTE THE SUPPLIED STARTING CODE CARRIES NO MARKS AS IT IS GIVEN TO YOU. |
| Robustness of application (error handling without crashing, checks for duplicate data etc) -15% |  |
| standards (is the code indented, commented, appropriate variable names, classes in capitals,etc) -10% |  |
| Completeness and professionalism (interface and overall presentation of project) -10% |  |
| Persistent Storage of entered data. This can be done using a Database or Text files -10% |  |

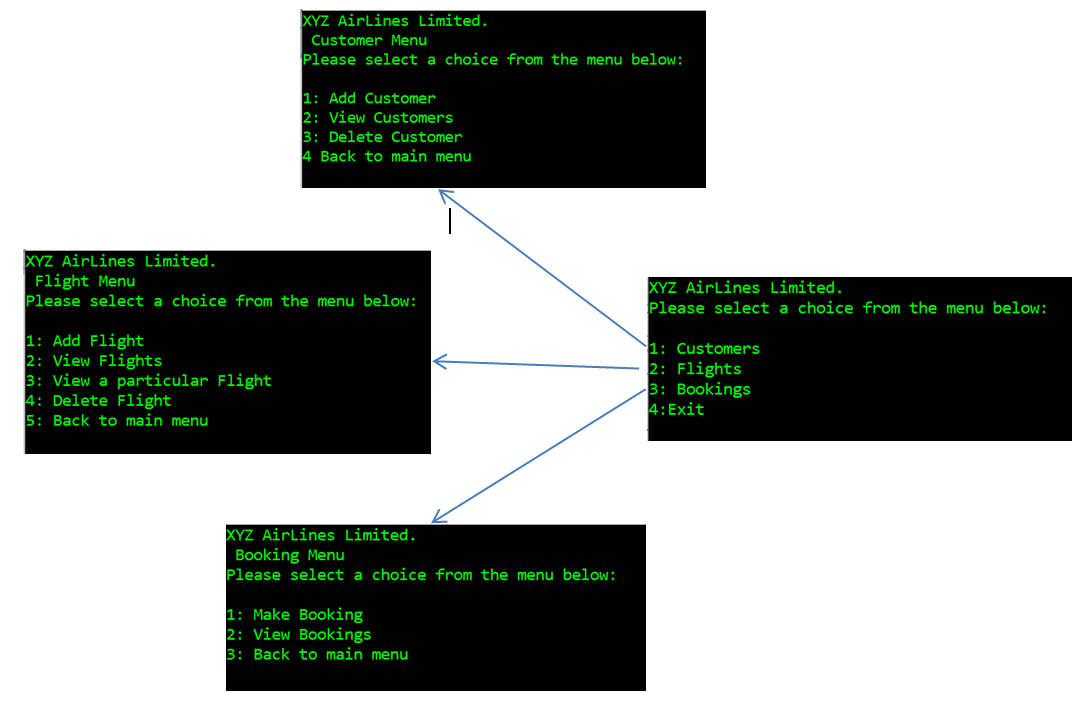
It is highly recommended that you initially design and create a text based system. If it is implemented perfectly this will give you 85% of the marks. The last 15% of the marks is related to **the graphical user interface**.

**PLEASE NOTE THIS IS ALSO A RESEARCH PROJECT ON GUI COMPONENTS.**

**Suggested Approach:**

1. **Develop the complete system with a CONSOLE interface by extending the starting code given.(This should be completed by the 20th November) This does NOT need to be submitted.**
2. **Create a new Graphical User Interface (GUI) project that reuses the fully working back end classes tested in part 1(Completed By December 3rd).**

**An example of the console application menues (this menu layout is different from the one in the starting code)**



Additional Notes:

* You cannot add two flights with the same flight number
* When you select “View Flights”, a list containing the Flight number, origin and destination for each flight must be shown.
* When you select “View a particular flight”, a list of all flights should be displayed and the user must be allowed to enter a particular flight number. All the information on that selected flight must be displayed. All the customers who are booked on that flight must

SPECIAL NOTE:

**A customer can only be deleted if there are no bookings for that customer.**

**A flight can only be deleted if there are no customers booked on the flight.**

You must extend the core design and add the functionality to add customers and booking features to the system.

**The information that must be recorded on a customer is as follows:**

* A customer ID must be assigned to each customer by the system. (NOT entered by the user)
* The customer’s first name
* The customer’s last name.
* The customer’s phone (a string type is fine for this)
* The number of bookings the customer has made

Your main menu should have the new options:

* Make booking
* View bookings

ADD CUSTOMER

When the “Add Customer” option is selected, all information necessary to make the customer object must be asked for.

A customer can only be added if there is no other customer with the same first name, last name and phone number (all three cannot be the same). An appropriate message as to whether the customer addition was successful or not should be displayed.

VIEW CUSTOMERS

When “View Customers” is selected, the first name, last name and phone number must be displayed for each customer.

**The information that must be recorded on a booking is as follows:**

* The date of the booking ( a string)
* A booking number must be assigned by the system. (NOT entered by the user)
* The flight object the booking is being made for.
* The customer object the booking is being made for

Note : you can use the following code to get the date and time from your system as a string…

string date = DateTime.Now.ToString(@"MM\/dd\/yyyy h\:mm tt");

Your main menu should have the new options:

* Make booking
* View bookings

MAKE BOOKINGS

When the “Add booking” option is selected, a list of all customers and all flights should be displayed first. The user must then be asked for the Customer id and flight id to make the booking for.

A booking can only be made if there is free space on the plane, the customer id exists and the flight id exists. All associated objects should be updated. An appropriate message as to whether the booking was successful or not should be displayed.

VIEW BOOKINGS

When “View bookings” is selected, the date, booking number, customer name and flight number must be displayed for each booking.

Your main class/program (the default one created when you make a new project) will be responsible for creating an AirlineCoordinator object as well as the complete menu system

Ensure that the interface/s generated by your program is user-friendly and gives a professional look.