**Assignment 6 (Second Part)**

The second part of this assignment is to take the new code and finish it. You must start from the code sent out, however you may use any code you have already written by adding it to the new code’s project or copy and pasting it in. Here is what needs to be done for the rest of the assignment. The first part of assignment 6 requirements still apply.

Once a flight is selected, all seats that are taken by passengers need to be turned red to indicate a taken seat. Next the user may choose to enter a new passenger and assign them a seat, select an existing passenger and change their seat, or delete a passenger from the flight.

To add a passenger the user will select the flight, and then click the “Add Passenger” button. This will bring up another form where the passenger’s information (First and Last Name) may be entered. If the user clicks the save button, the new passenger is inserted into the database. Once the information is saved, the passenger’s name will show up in the list of passengers combo box. From there the user must select a seat for the passenger (rest of the form is disabled) by clicking an empty seat on the diagram, which will update the seat with the “Selected Passengers” seat color (green).

To delete a passenger the user will select the flight, and then select the passenger. The passenger may be selected from the list of passengers or by clicking a taken seat. The selected passenger’s seat will show up in the Selected Passenger’s seat color (green). From there the “Delete Passenger” button may be pressed to delete the user from the list of passengers and turn the seat color back to an empty seat (blue).

To change a passenger’s seat the user will select the flight, and then select the passenger. The passenger may be selected from the list of passengers or by clicking a taken seat. The selected passenger’s seat will show up in the Selected Passenger’s seat color (green). From there the “Change Seat” button may be pressed (rest of the form is disabled), which will allow the user to click on an empty seat (blue) to assign it to the passenger. The passenger’s new seat will then show up in the Selected Passengers seat color (green).

On the plane seating diagram the color blue will mean that a seat is empty, the color red will mean that a seat is taken, and the color green will be the current selected passenger’s seat.

Please use help files provided for Assignment 6 that are contained in the “Assignment 6 Help” folder. There is an example of a completed Assignment 6 that may be ran to see how a full operational program should behave.

All business logic must be in separate classes and not behind the UI. All top level methods need to handle exceptions, and all other methods need to throw the exceptions. All methods and each attribute needs an XML comment.

For the flight and passenger combo box, they must be filled up with flights and passengers by binding a collection to the source of the combo box. This means that classes of type “FlightData” and “PassengerData” will have to be used. These classes will just hold the data for each, then added to a collection, like a List, then returned and bound to the combo box, to display the data.

**NOTE**: Microsoft Access should be used to store the data.

**Commonly Missed Items**

- Didn't meet the requirement "In the combo box, both the Flight Number and Aircraft Type should be displayed. "

- If a flight is selected, then a passenger, then the flight is again changed, the program throws an exception.

- Should be able to select a passenger by clicking on a taken seat.

- After a passenger is added, the program allows the user to click on a taken seat.

- If I change the passenger’s seat I shouldn’t be able to select a taken seat.

- If a passenger is deleted then the change seat and delete seat buttons stay enabled, so if they are clicked again an exception is thrown.

- When selecting a passenger from the combo box, their seat should turn green.

- If no passenger is selected, and the change or delete passenger button is clicked, an exception is thrown.

- Switching between flights does not load the passengers assigned to that flight in the combo box.

- After a passenger is added you should force the user to select the passenger’s seat.

- Passengers on a flight not being loaded into taken seats.

- Change seat requirements not implemented.

- Delete passenger requirements not implemented.

- Add passenger requirements not implemented.

- Didn't meet the requirement "For the flight and passenger combo box, they must be filled up with flights and passengers by binding a collection to the source of the combo box. This means that classes of type “FlightData” and “PassengerData” will have to be used. These classes will just hold the data for each, then added to a collection, like a List, then returned and bound to the combo box, to display the data."

- All business logic code is behind UI

- Try catch blocks not implemented correctly. Top level methods should handle exceptions and lower level methods should raise exceptions.

- All methods need a try catch block.

- Exception handling not properly implemented.

- All attributes and methods need XML comments.

- Did not meet the requirement to take the new WPF code and implement the second part of the requirements.