Project Description

Airline Registration

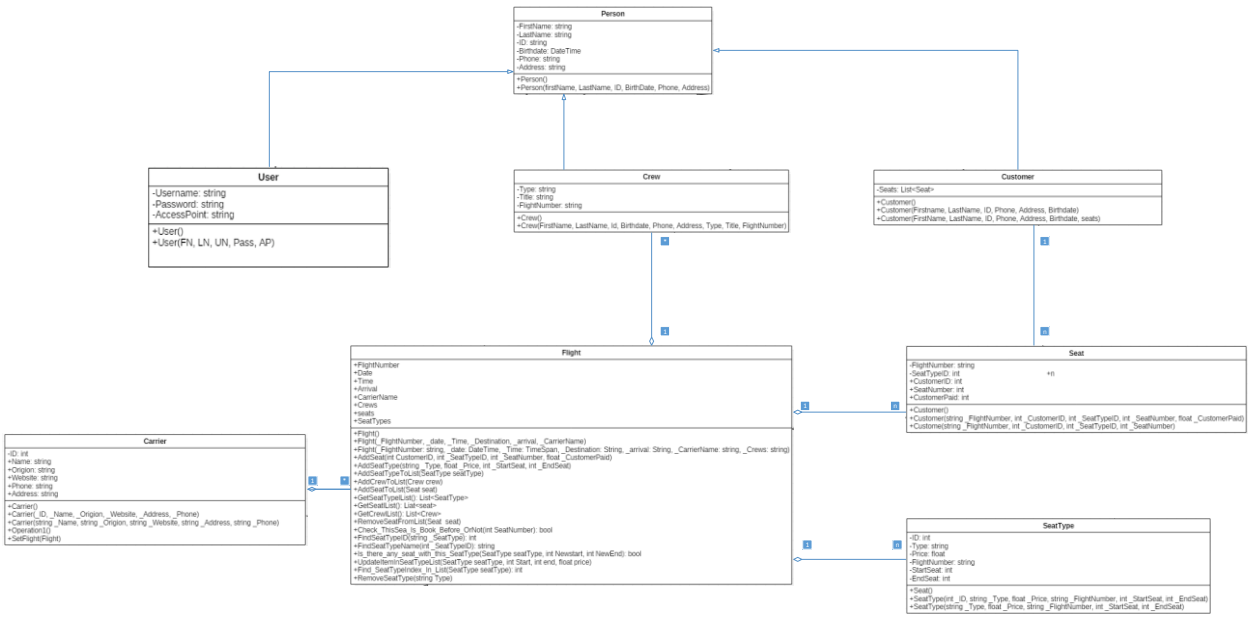
In this project I tried to implements everything that you specified in the project in the desire deadline. For me it was important to implement according to object oriented architecture, implement every things that we learned in the class, and design forms simple and user friendly because I believe that the program is for users if it be complicate and sophisticate that users cannot work with it makes the program useless although there are a lot of programming behind it.

I describe my class diagram, program, and add balsamiq design.

Class Diagram Description

Airline registration as it is defined in our project Flight, Carrier, Crew, Seat, Customer, and user are the base classes but it needs some classes for more efficiency. As Customer, Crew, and User have common attributes I make the person class that they inherit from this class. Moreover, the project mention that each flight has three type of seat (Economy, Economy Plus, Business) and we know that each of them has its own price and their seat start and end from specific place. We can consider an inheritance relationship but as child class just have one attribute of type I prefer to consider as association relationship and make just one class of SeatType. SeatType like Crew strongly depends on flight so that its relation is aggregation.

One thing remains, the relation between Customer and flight. I consider customer as a person who can ask a ticket from the agency a lot. If I connect it to the flight their relation will be many to many, that it shows that I do not consider one class between them and this class is a Seat or Ticket.

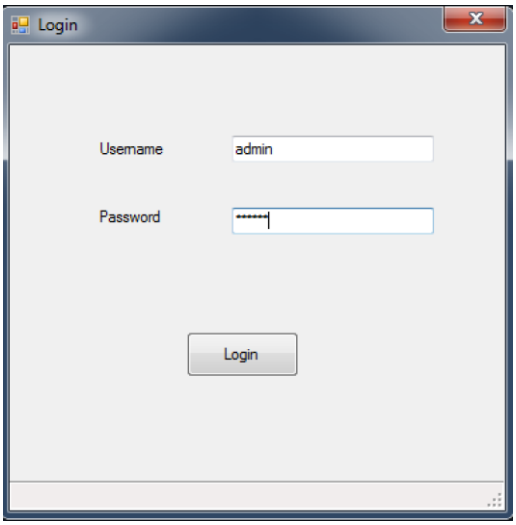


Program Description

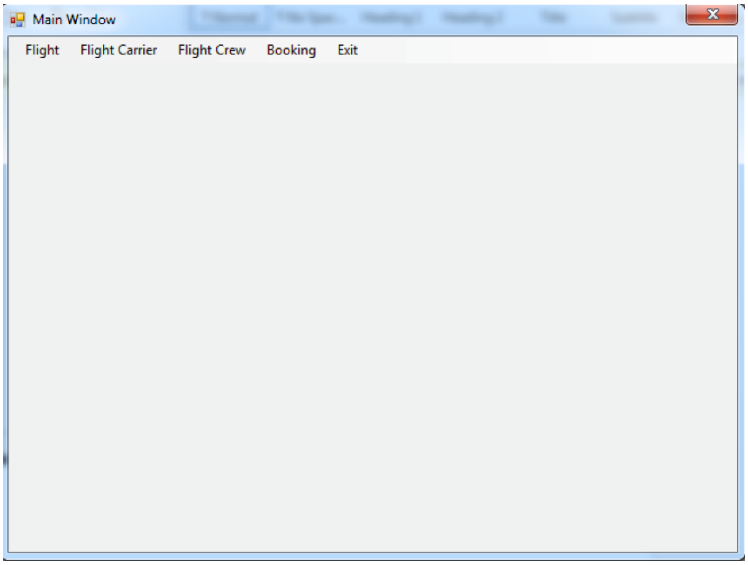
For developing this project I tried to implement a program in both 3-Tier and Object-oriented Architectural style. It means that I do everything with objects that they instantiate from my class diagram classes and use 3-tier architecture for store and retrieve data from database. Therefore, there are four group of classes. First group are UI classes for contracting with user for this demand I chose WinForms. Second group is my objects. They communicate through interfaces, by calling methods or accessing properties in other objects, and by sending and receiving messages. The third is classes that get object and send its information to DB or take data from DB and fill the relevant object with that information I called them Business layer classes. The last one are data access layer classes that they use for connecting program to DB.

Specific features of each form:

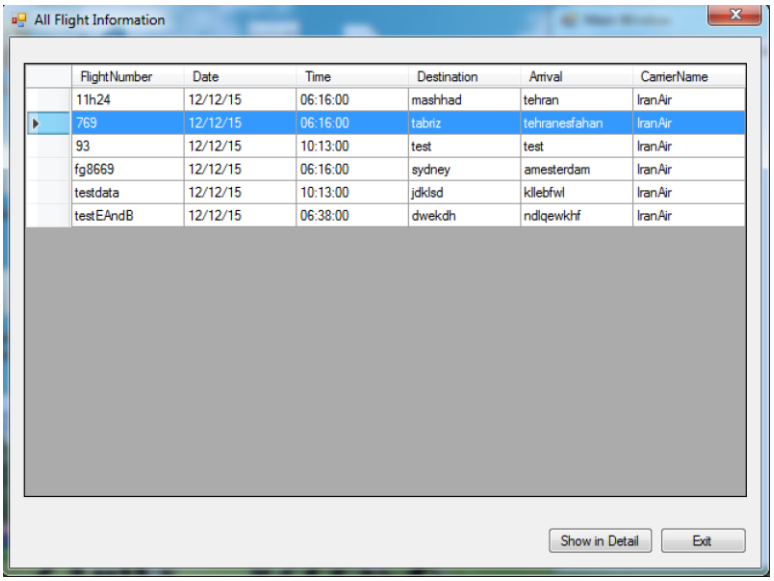
Login Form: there are two type of user can access to the program one of them is admin and one of the is regular user, admin user can access to all of forms but regular user can just book and unbook flight, see flights information.

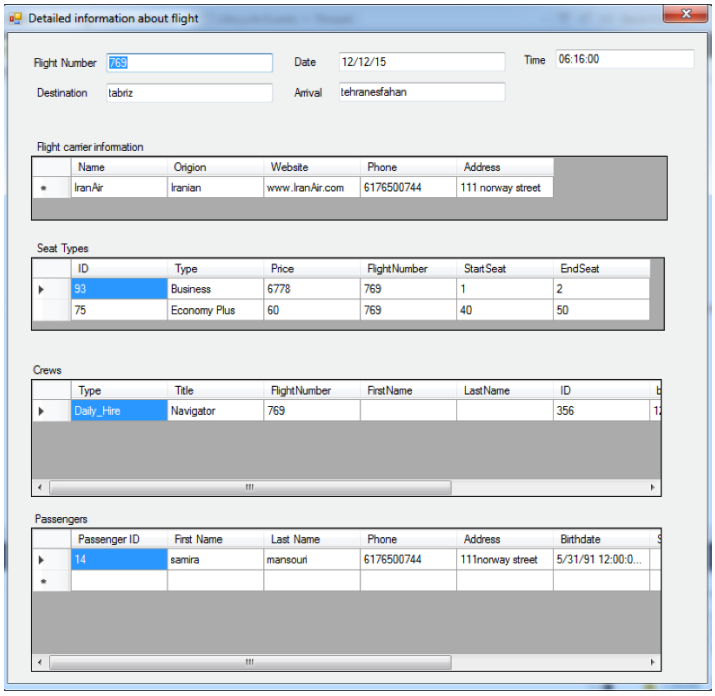


Main Window: This form is parent of all forms all the things are accessible through manuscript of this form.

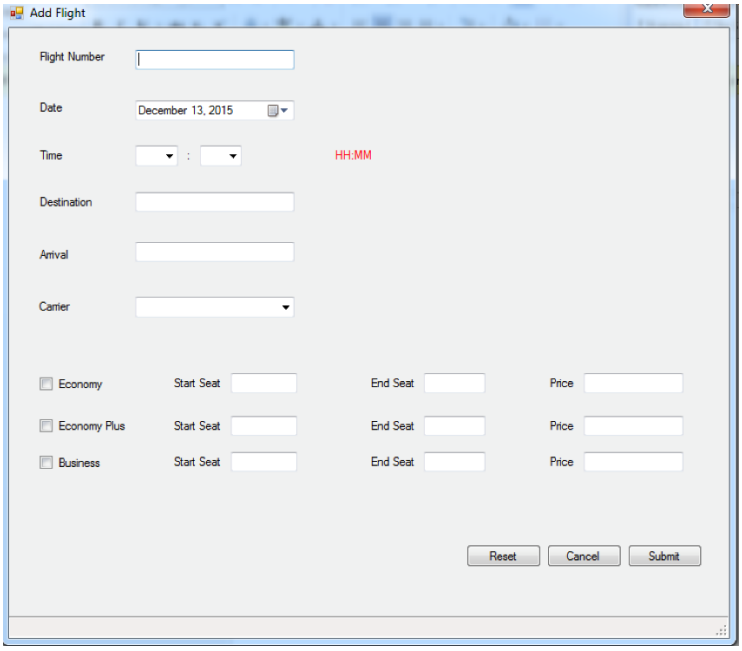


Show all Flight: this form show information of all flight and by clicking the one row and click “Show In Detail” can see all the information of the flight





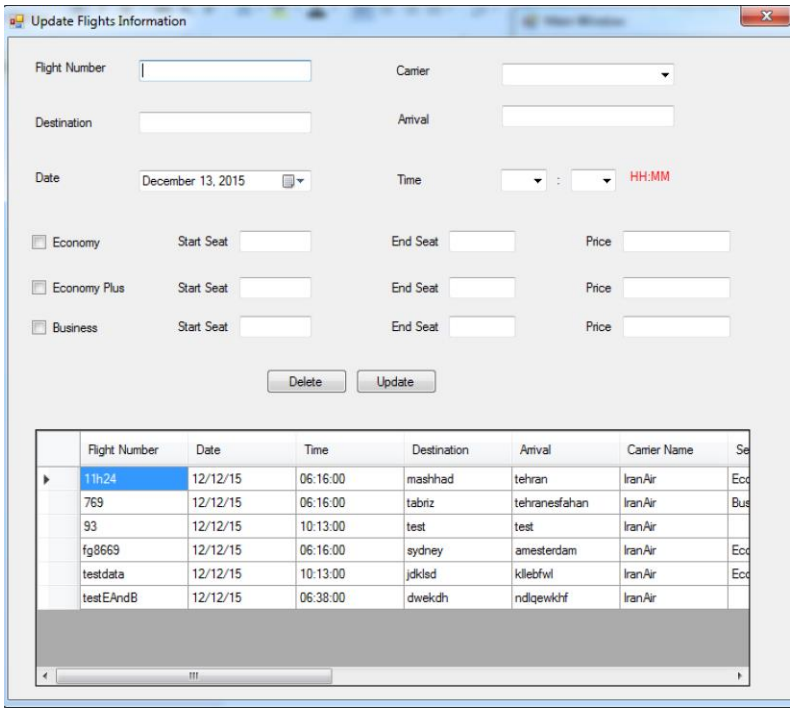
AddFlight: gives the opportunity for adding a flight by considering all possible exception and data validation such as entering correct integer or string data, check overlapping of the seats, seats price that business class is expensive than Economy plus and Economy plus expensive than Economy, and etc.



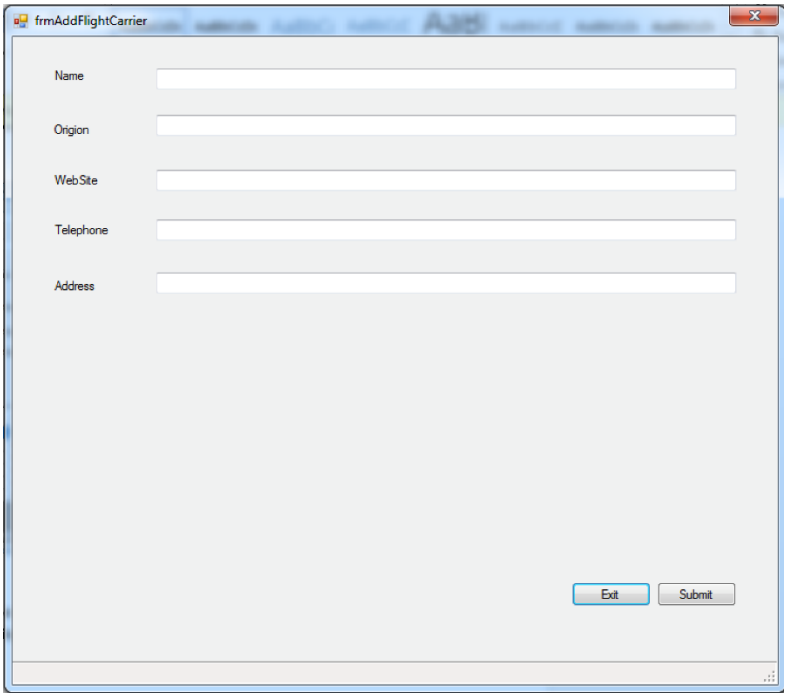
Update Flight: this form for updating data. As use may not consider the flight number I provide a table of the flight information and to help her/him to choose its flight, fill the information by clicking on one row, and sorting data by clicking on specific column.

For deleting it checks whether is there any data that affect by deleting this flight if yes it does not allow user to delete flight. It is for data protection and avoid the unwanted deleting. User can unbook tickets and then delete flight.

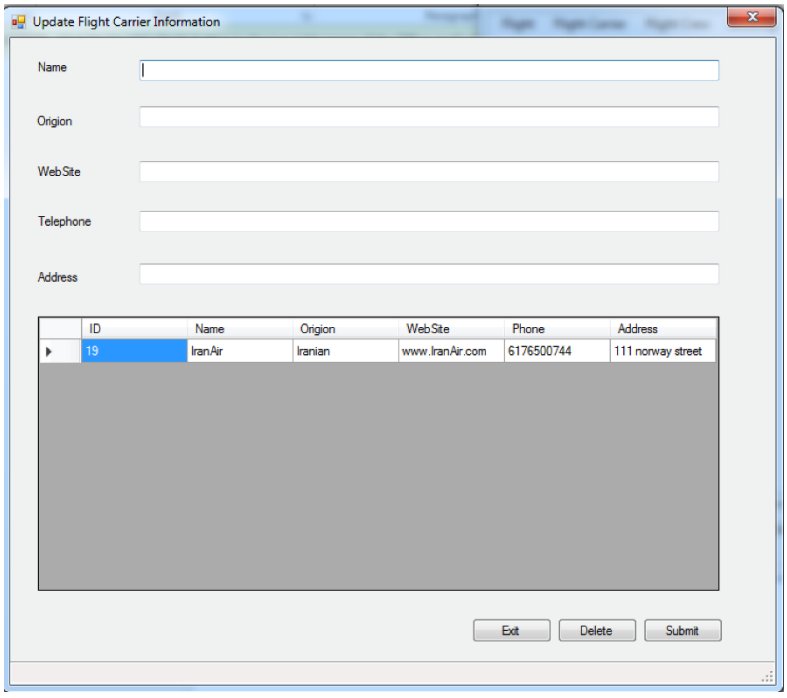
For Updating like deleting it consider data protection and does not allow user to change range of the seat if there is any tickets that was booked in the previous type. Moreover, it considers all the data validations of the add flight. In addition, it does not allow user to enter the new flight in this form, in other words, if the user enter the FlightNumber that does not exist before it gives the error because it is adding a flight and adding flight should be done in Addflight form.



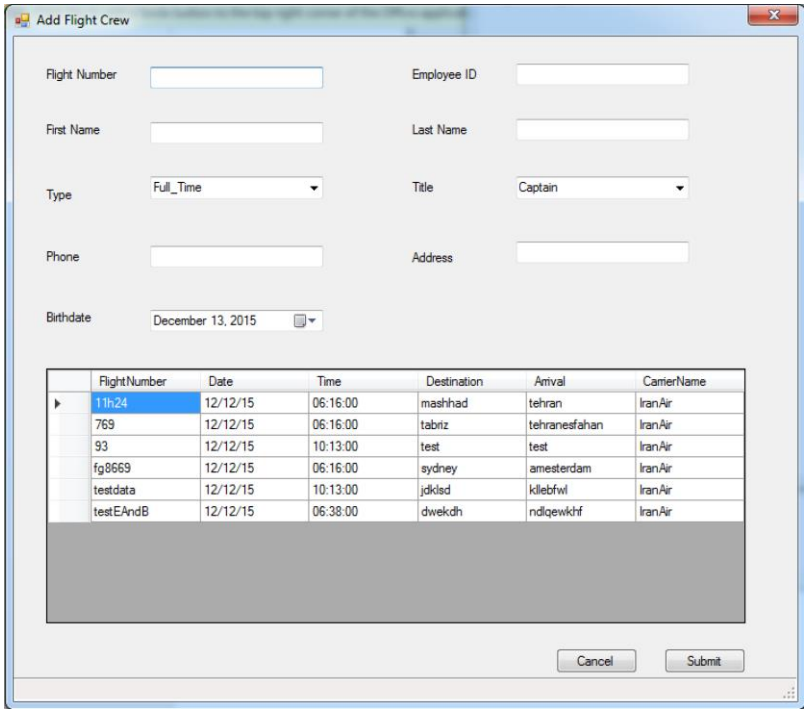
AddCarrier: We know that name of each flight carrier is different from others therefore user should fill this field and for adding Flight carrier we should know at least its address and its phone.



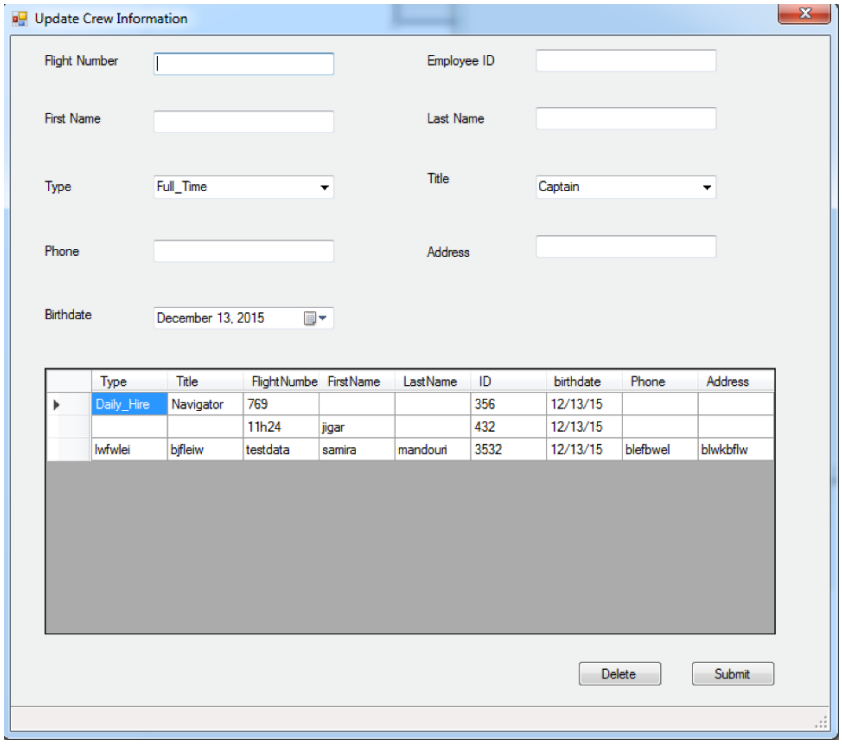
UpdateCarrier: user can enter the flight name by its self, searching in table, or add automatically by selecting one row. Flight name phone and address should be filled like add carrier and name should not be a new because it is considering as a new carrier.



Add Flight Crew: for adding flight crew user should at least fill FlightNumber and EmployeeID. FlightNumber should be in the flightNumbers that submitted in the system, this table of the flights exists for this reason and help user to search its desired flight and fill the the flightNumber by selecting one row.

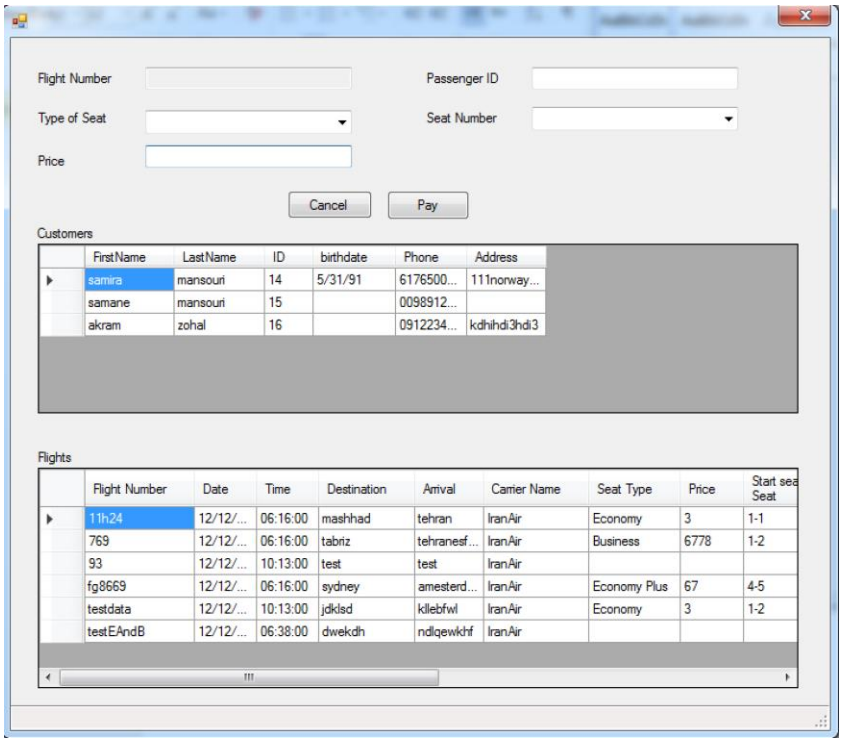


Update Crew information: like other update forms this form checks the data validation and check the specific crew that the user want to update or delete was added before to that flight.

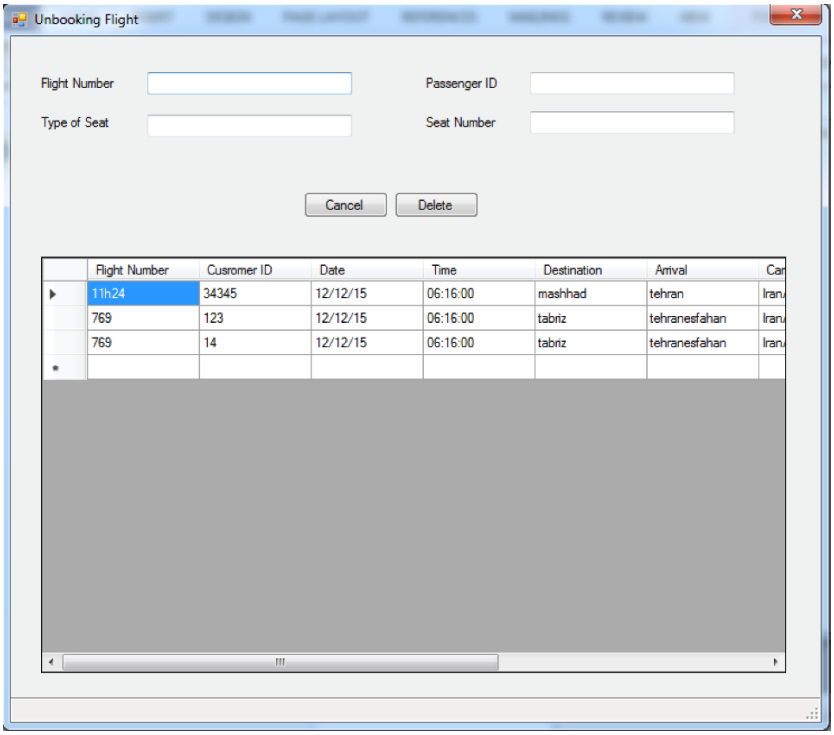


Book Flight:for booking flight user should selects its desire flight from flight table and then fill the CustomerID of the person who buy the ticket. When user select the flight the type of seat of the specific flight will be uploaded to the combo box and by selecting the type of seat the seat numbers of this seattype that is not booked before for other users and the price of that seat will be shown. As in the project specific that each user can pay each amount of money that it wants I just fill the text and I allow user to change it.

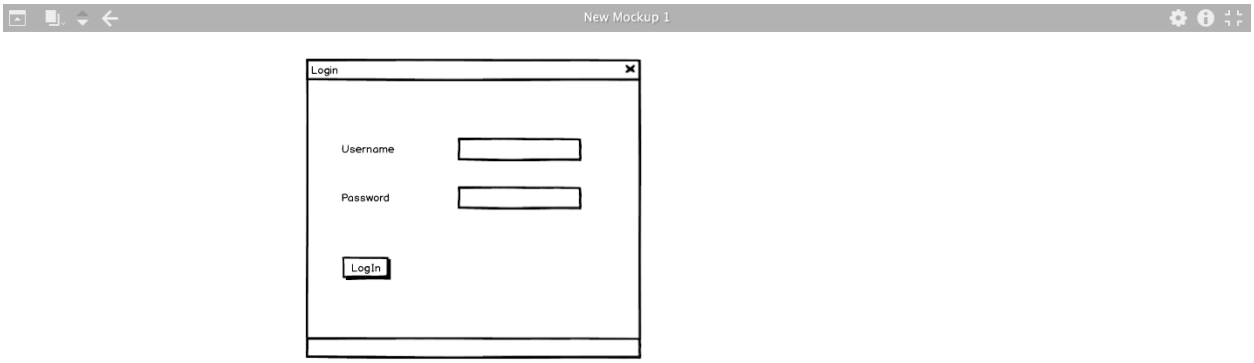
I should mention that if the user wants to fill the PassengerID by her/himself system checks whether this ID exist in system or not.

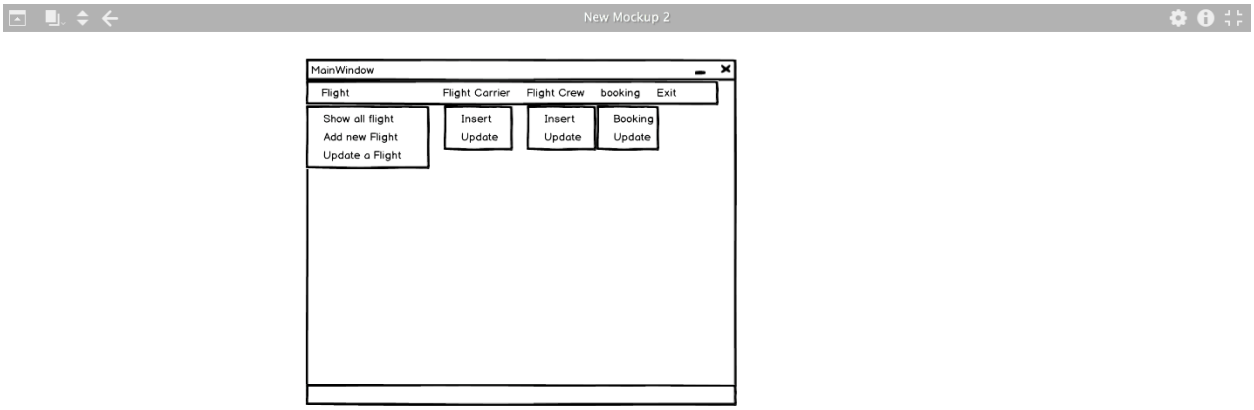


Unbook Flight: for this user fill passengerID and FlightNumber , things that unique tickets from each other. For user convenience I provide the Tickets table by possibility of sorting to choose its tickets easily. Each user wants to fill by her/himself that this ticket exist in system.

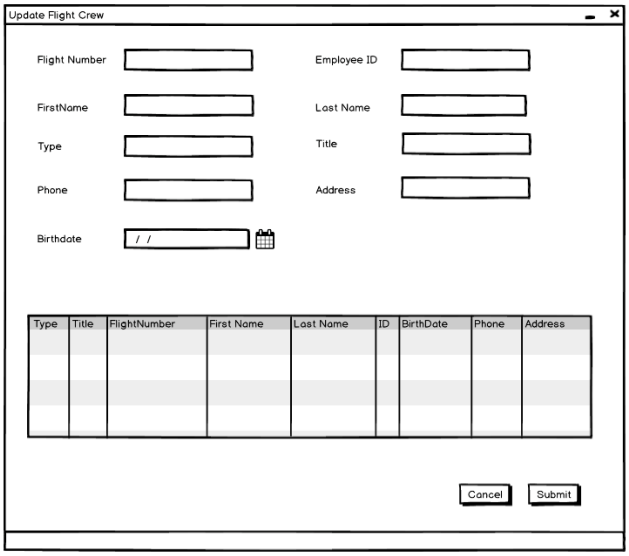


Balsamiq:

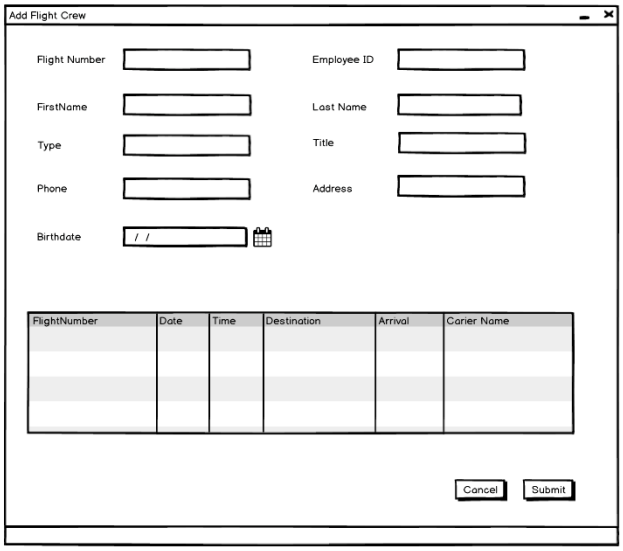

New Mockup 1
Login
Username
Password
LogIn


New Mockup 2
MainWindow
Flight
Flight Carrier
Flight Crew
booking
Exit
Show all flight
Add new Flight
Update a Flight
Insert
Update
Insert
Update
Booking
Update

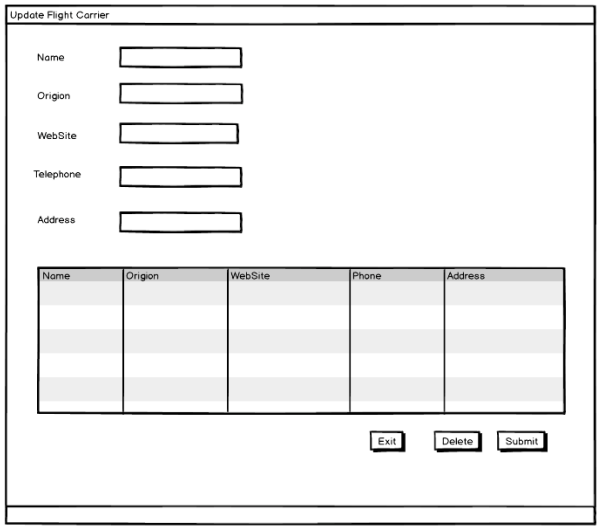
New Mockup 15



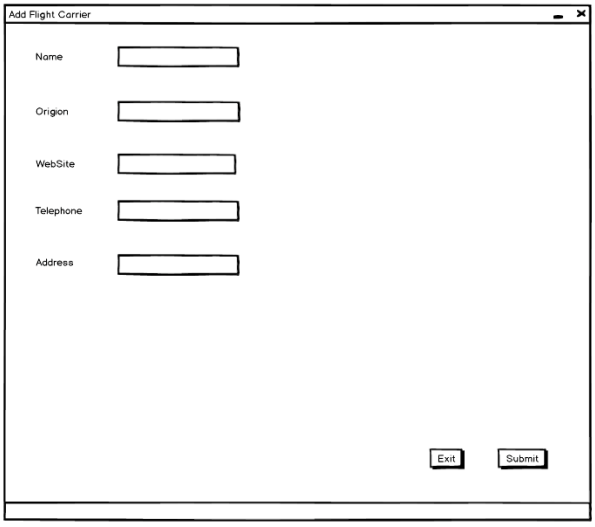
New Mockup 14



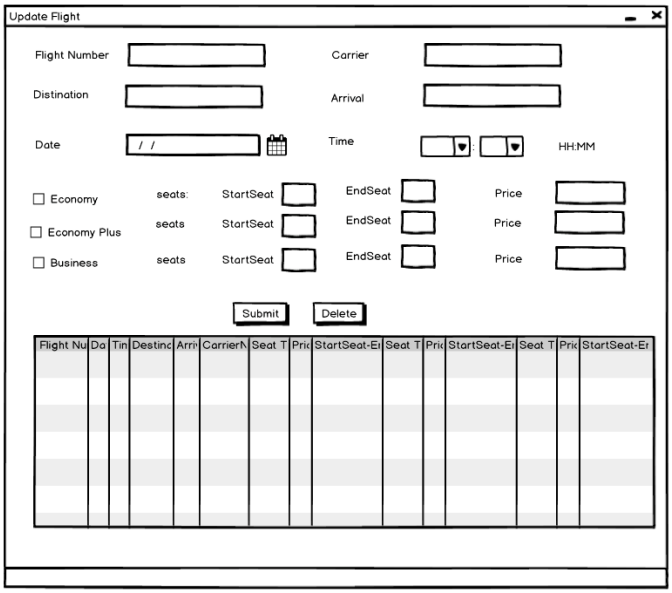
New Mockup 11



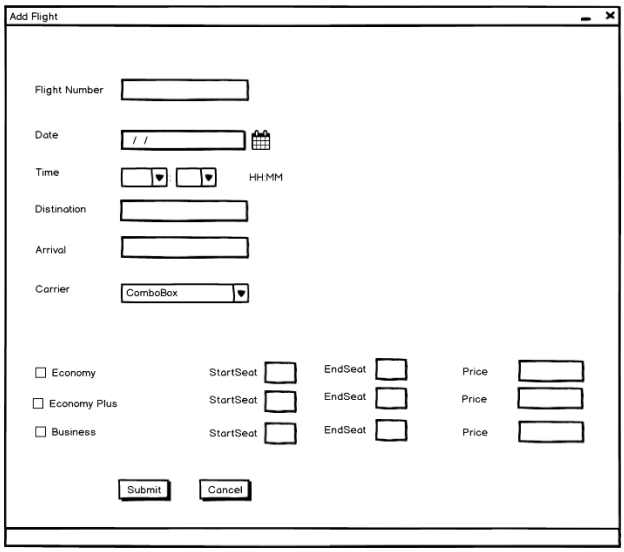
New Mockup 10



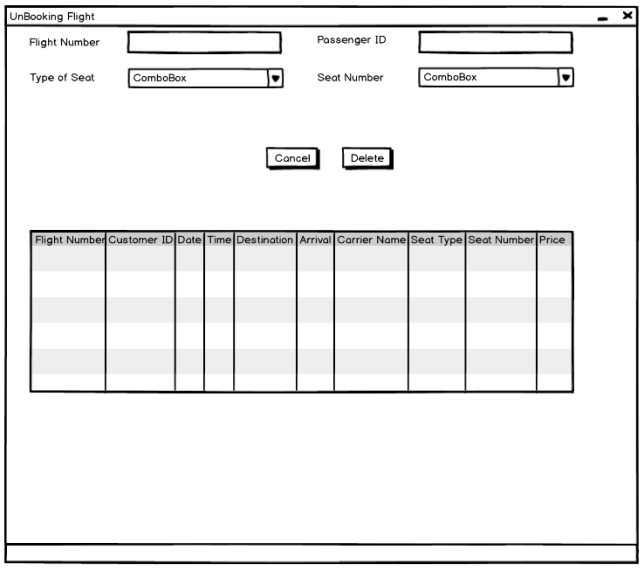
New Mockup 8



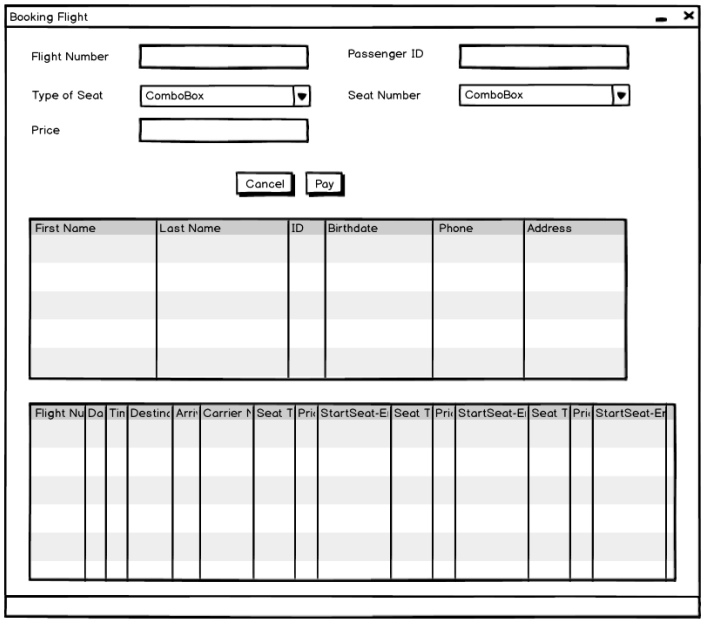
New Mockup 7



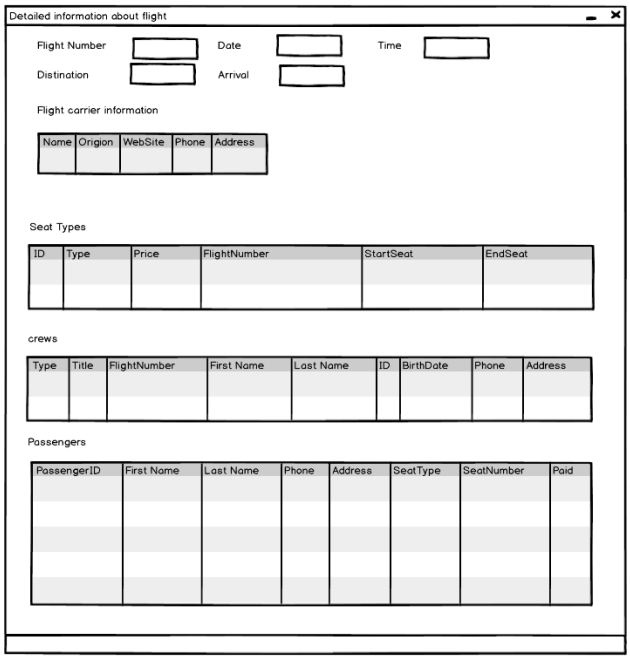
New Mockup 6



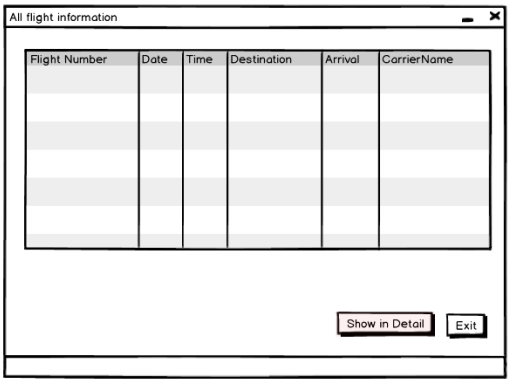
New Mockup 5

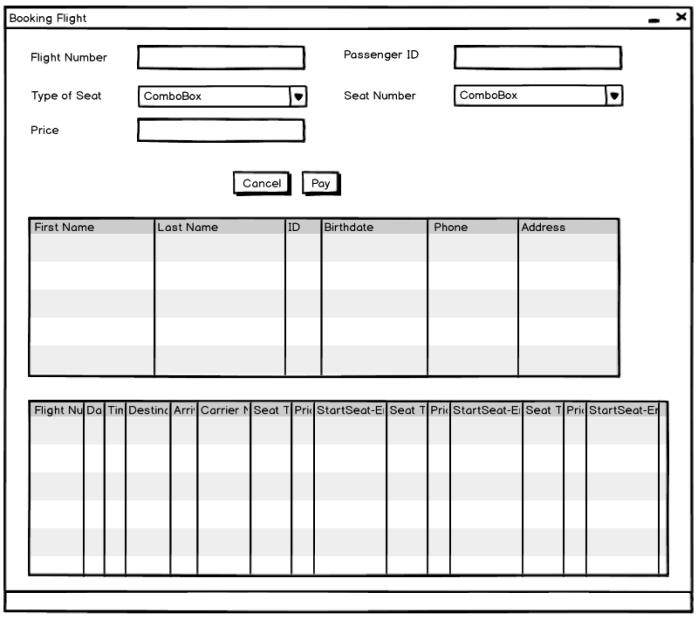


New Mockup 4



New Mockup 3




Booking Flight
Flight Number
Passenger ID
Type of Seat
ComboBox
Seat Number
ComboBox
Price
Cancel
Pay
First Name
Last Name
ID
Birthdate
Phone
Address