Index

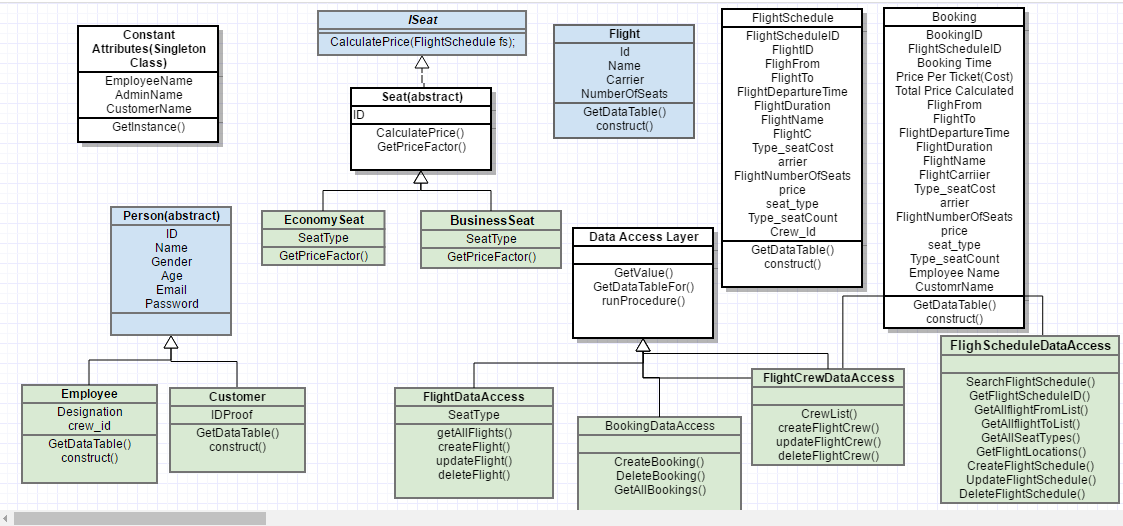
1. Objective
2. Tools Used in the Project
3. Class Diagram
4. ER Diagram
5. Functionalities Applied with Screenshots
6. Code Example through Implementation
7. References
8. **Objective:**

The objective of the project is to create an airline reservation system using C# language with OOPs concepts and providing Unit test using NUnit.

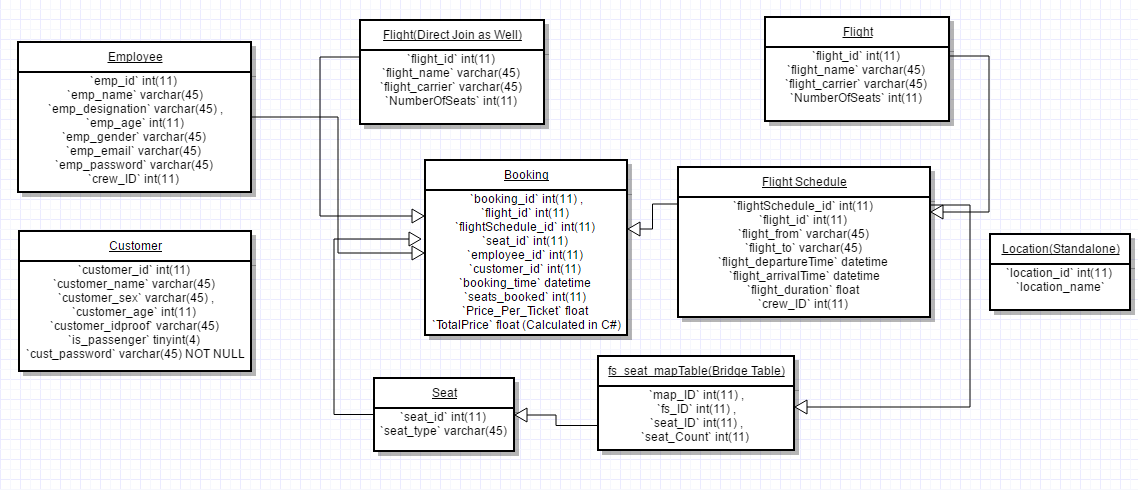
1. **Tools Used:-**

* Visual Studio
* MySQL Database
* NUnit
* Log4Net for Logging Events

1. **Class Diagram**



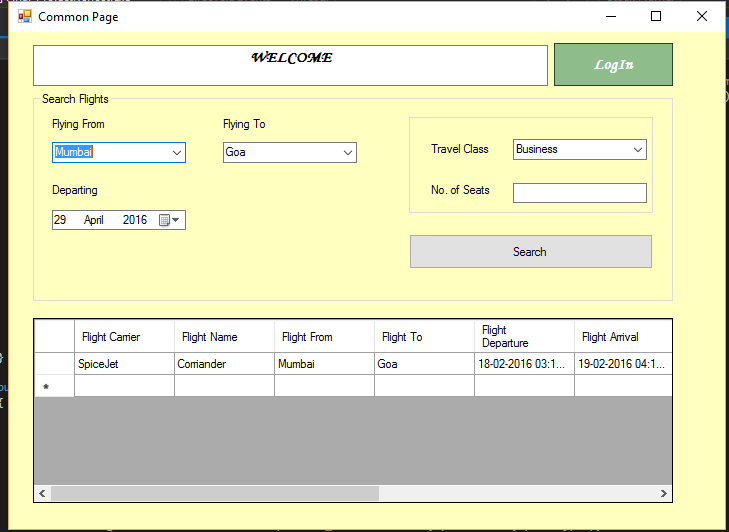
1. **ER Diagram**



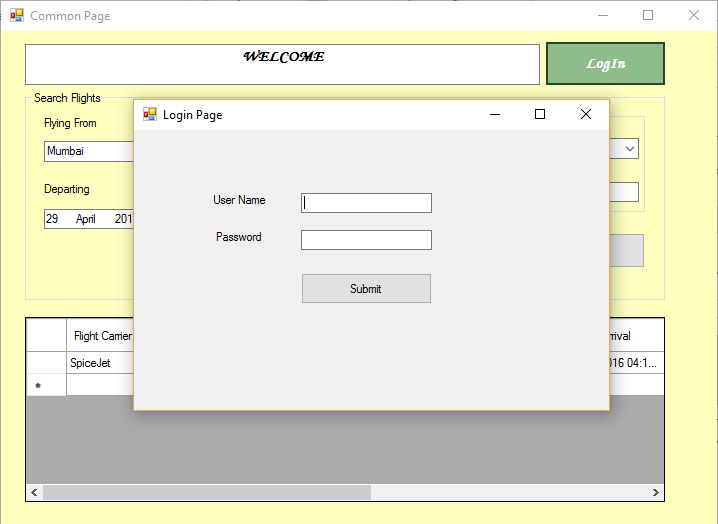
1. **Functionalities :**

* Users should access to the application through a login screen

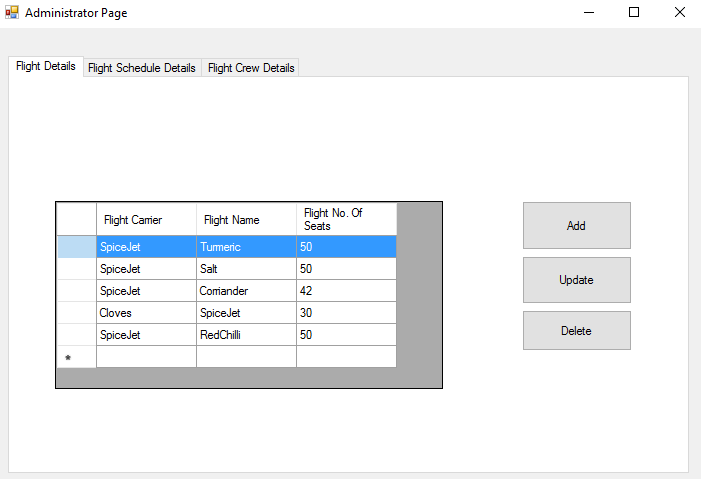
A common screen is created to search flights and then you can login if you find your required flight schedules

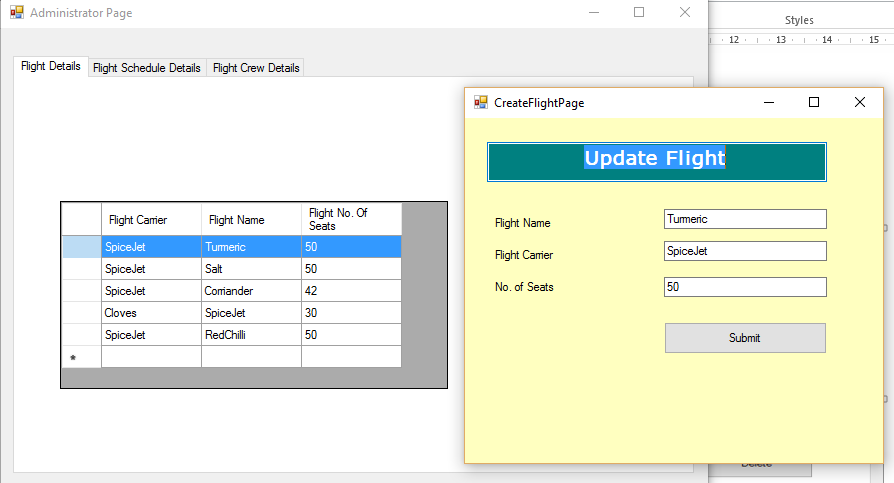


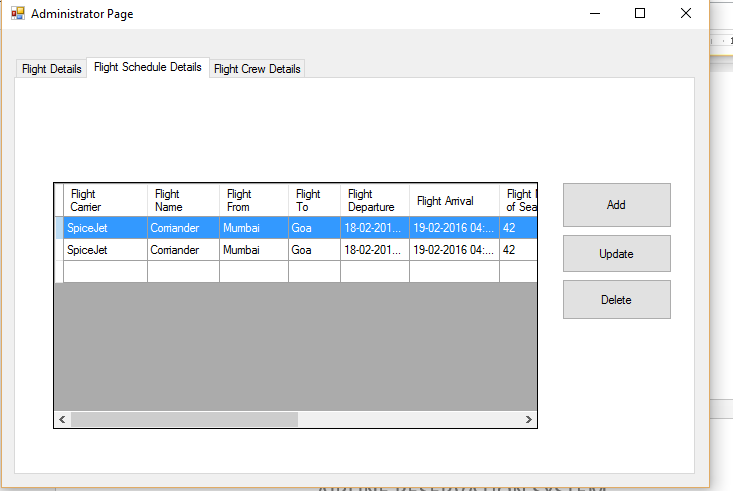
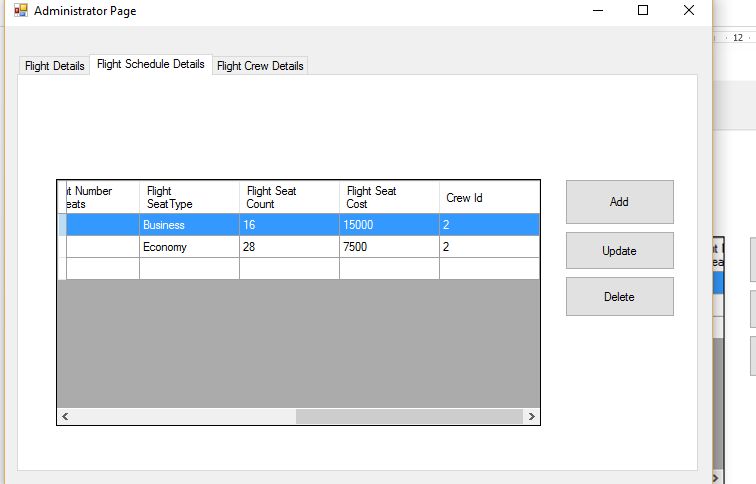
* Users have different access levels:
* a. Administrator – Can also execute the workflows mentioned below
* b. Regular – Restricted to the workflows mentioned below

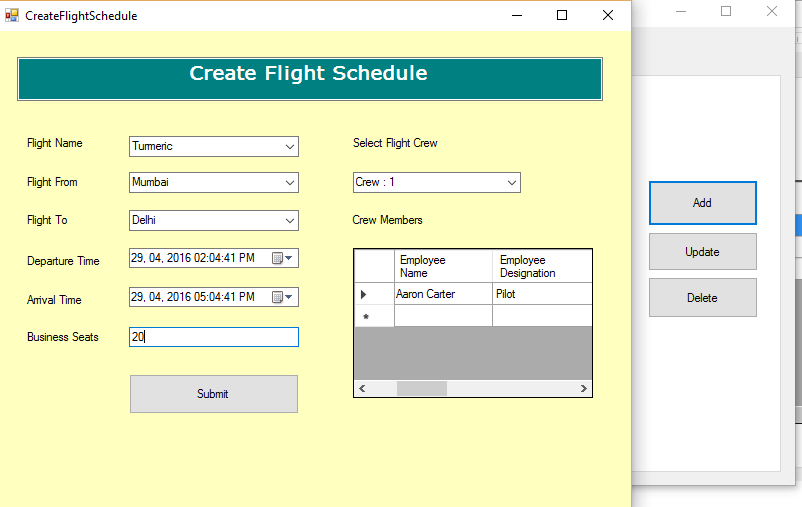


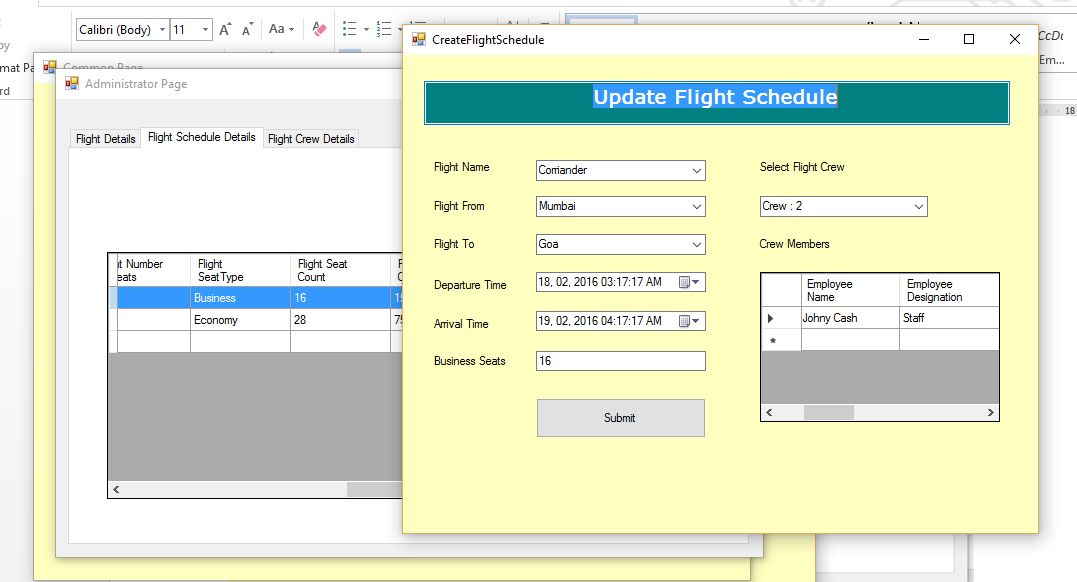
* For Admin Login Functionalities
* Administrator can introduce/update/delete a flight
* Administrator can introduce/update/delete a flight carrier
* Administrator can introduce/update/delete flight crew

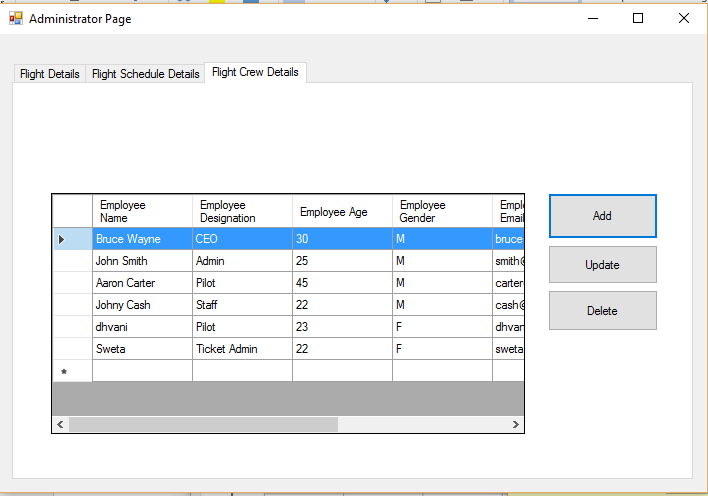


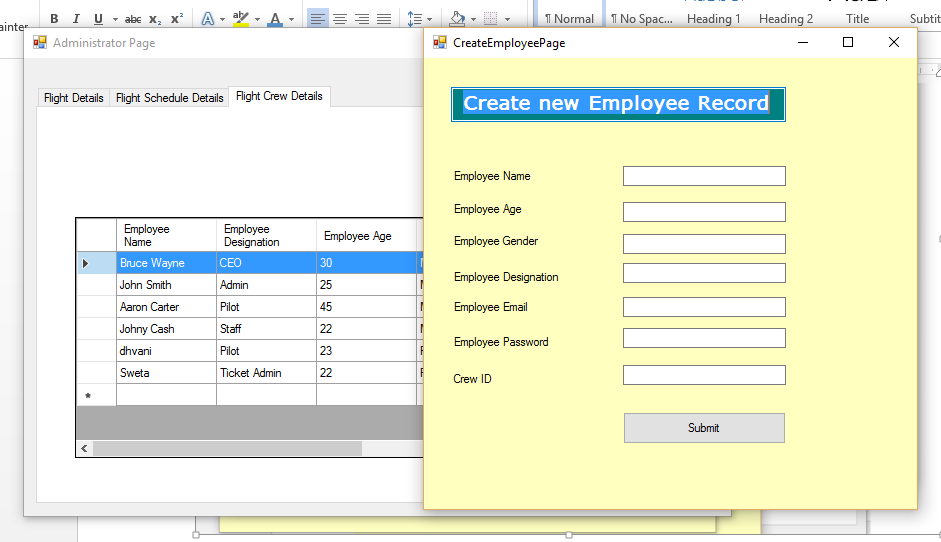




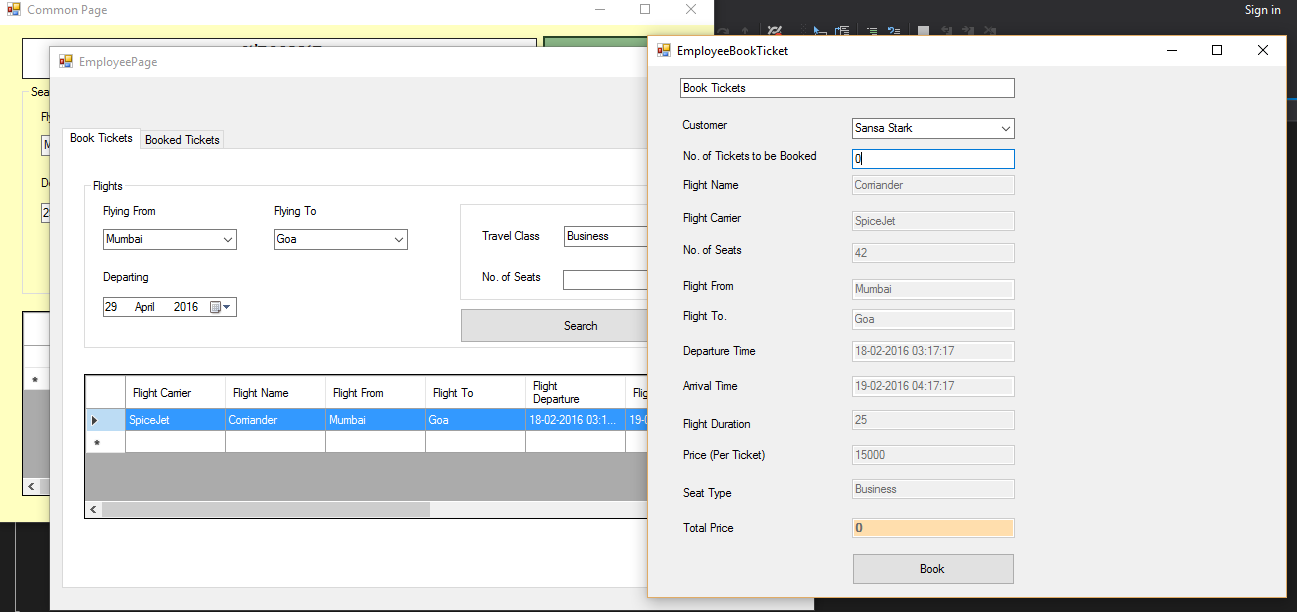


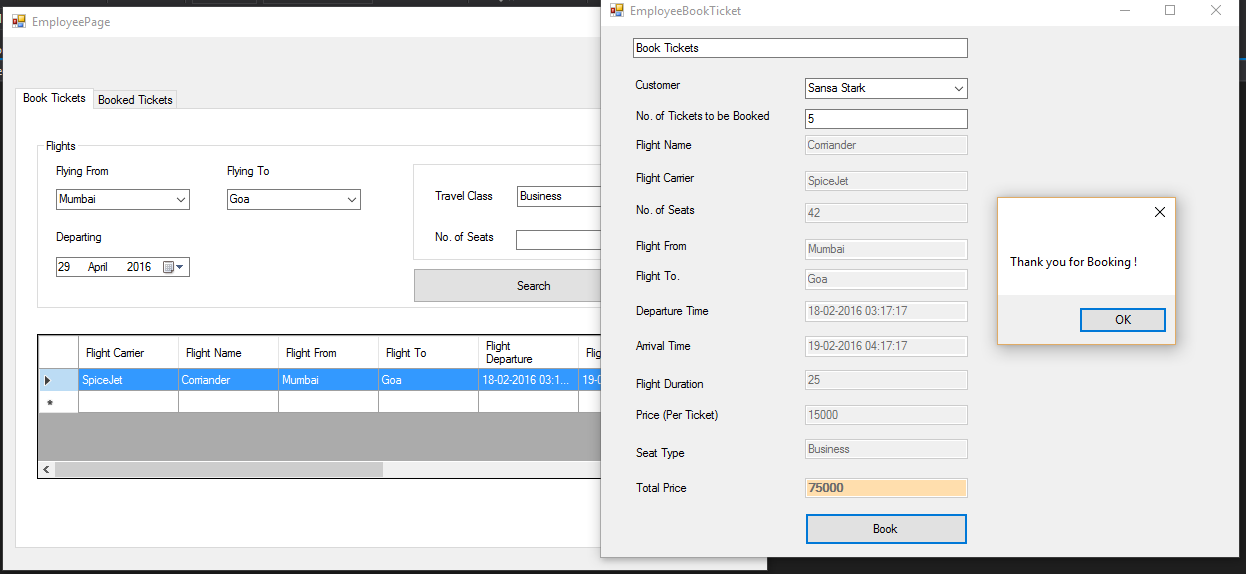


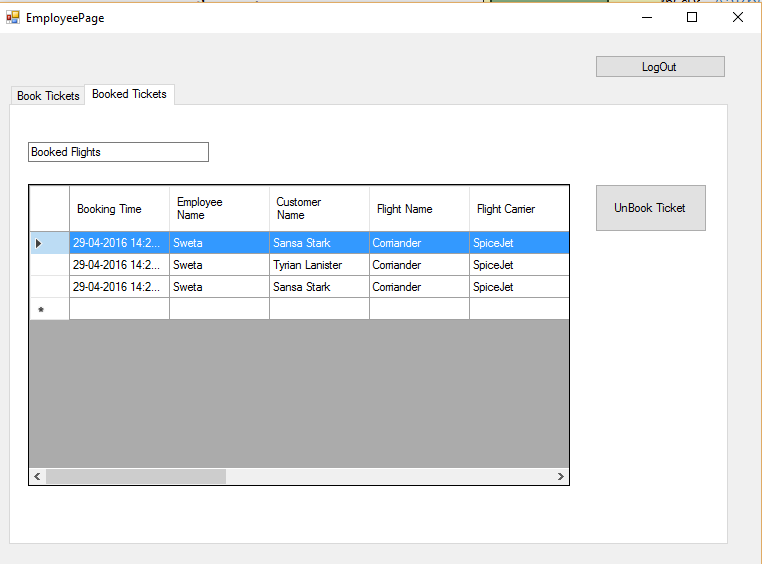


* End user should be able to see all the flights in one location
* End user should be able to see all detailed info about the flight – This includes a. Carrier info b. Flight crew info c. All passengers booked for that flight so far d. How much each user paid for that flight (Dependent on the seat level mentioned above)
* End user should be able to book a flight for at least one customer at any given point
* End user should be able to unbook a flight for at least one customer at any given point

End User has now Logged In and can perform following activities.

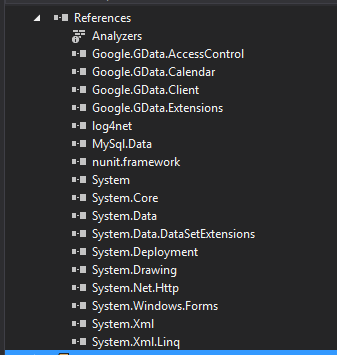


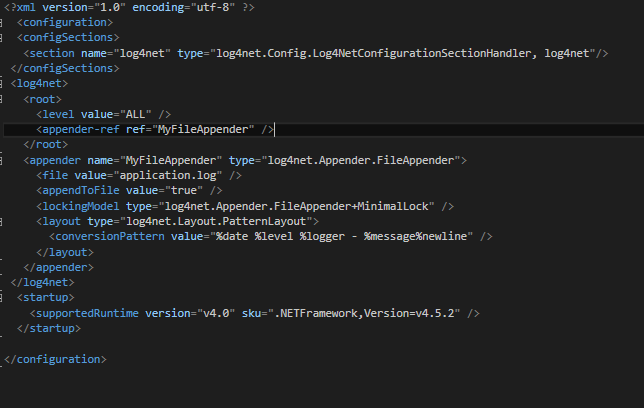


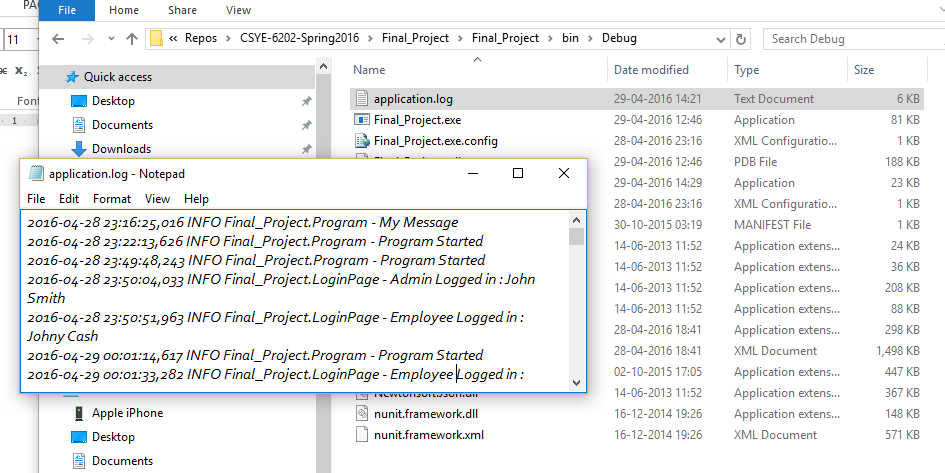


* The app must provide proper logging (can be plain file, a table in a DB, or even a XML file) to trace what the app is doing at any given point (Log file can be saved in any location, but needs to be mentioned in design documents)

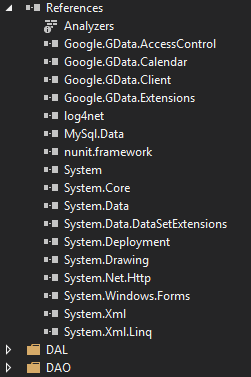
**Log4Net is used**

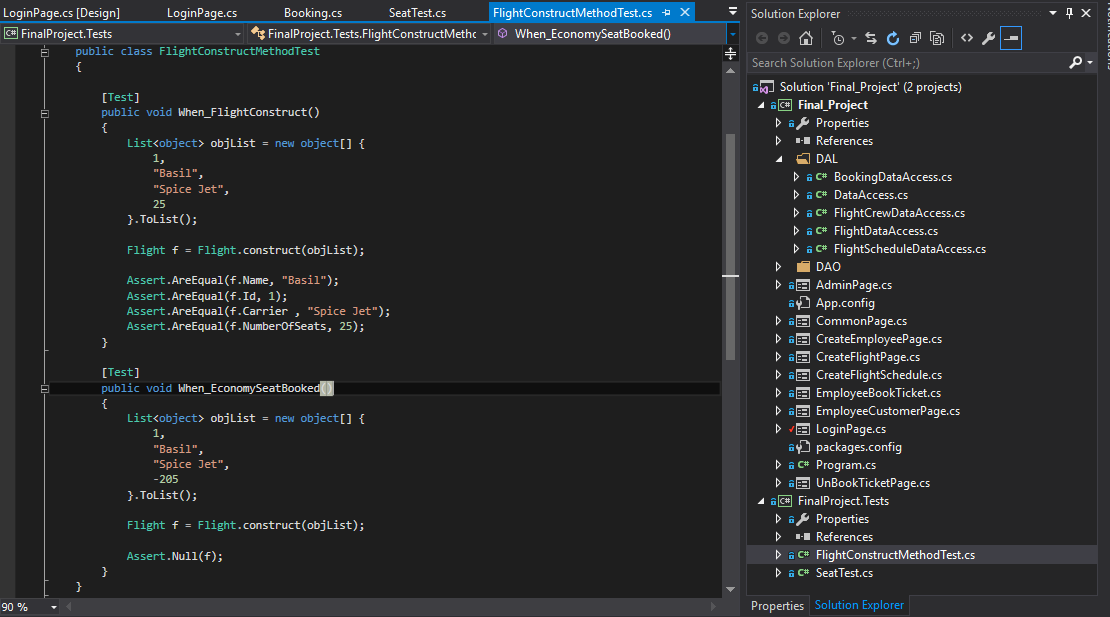


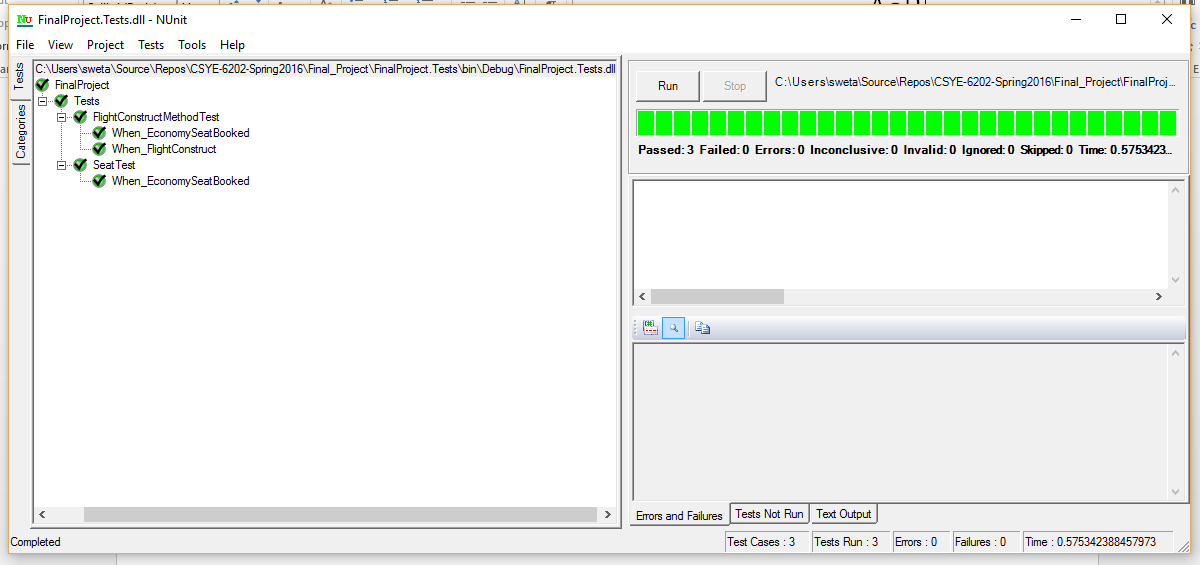




* **Using NUnit**



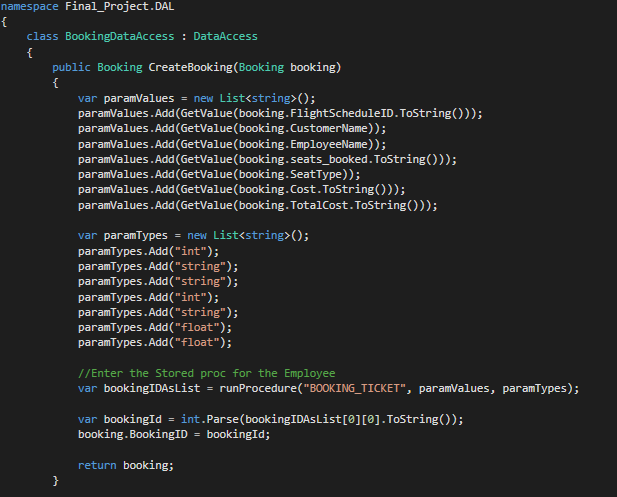




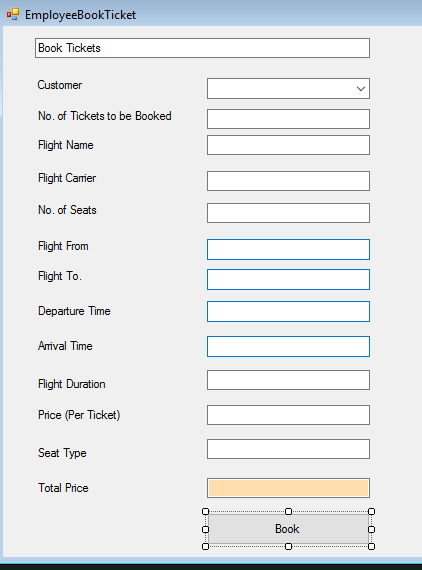
1. **Code Example through Implementation**

**Taking an example of Booking a Ticket**

* Booking Access Layer has all the functions to call the stored Procedures

This function is taking a hydrated booking object to insert the booking details into the booking table.

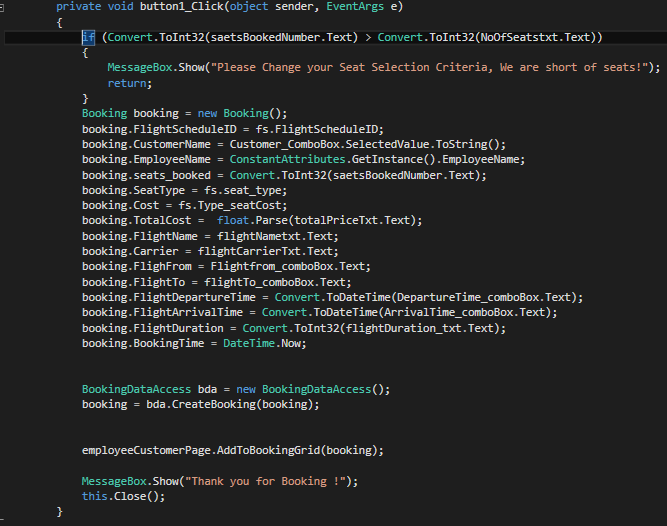
Once successfully inserted it takes the new booking id and stores it into the booking object and returns it back to the caller.

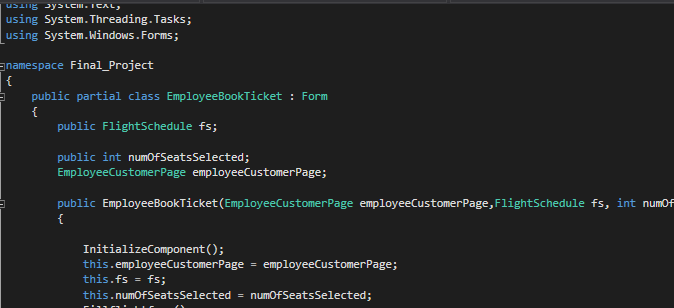


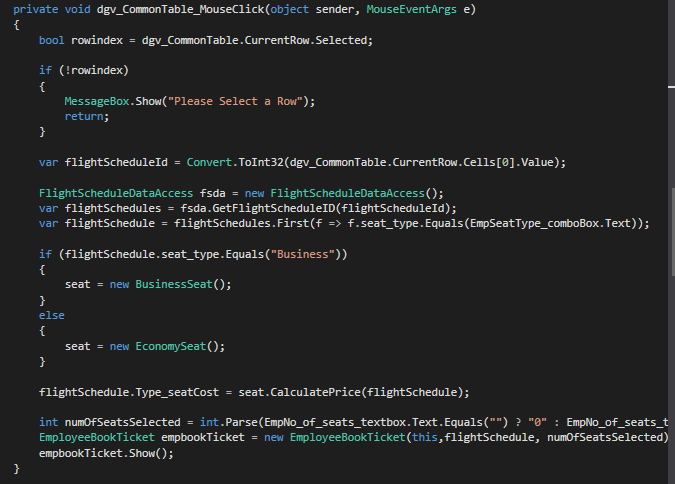
To initiate the booking the user fills the booking form

The Total price is automatically calculated based on the Seat Type, duration and Number of Tickets Booked.

Once the user clicks on ‘Book’, the application hydrates the booking object and passes it to the DAL class mentioned above.



To open the employee book ticket screen we passed the object reference of its parent page – ‘Employee Customer Page’.

When a User selects a particular flight schedule from the Data Grid View, we fetch the flight schedule object from the list.

Using the ISeat Interface, flight duration and Seat Type we calculate the Ticket Cost.

And further depending on the tickets booked we multiply the price to get the Total Amount.

These objects are then passed to the Booking Page to book the Ticket.

1. Reference :

Class Discussions, Project Presentation

TA Sessions,

PPTs on Blackboard

StackOverflow.com

Google.com