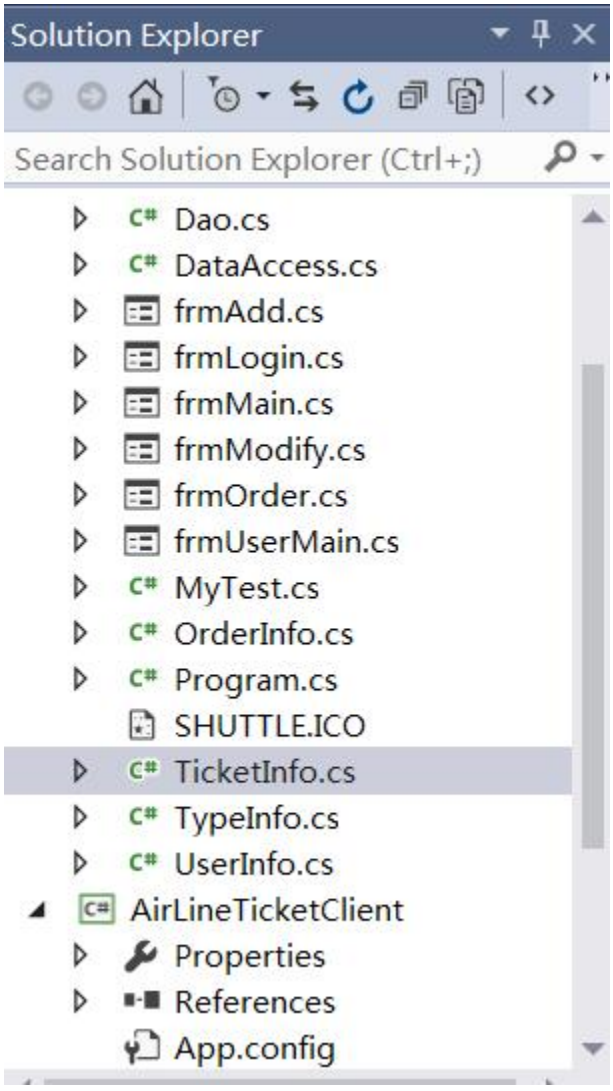
Project Polishing Document

1. Add five classes to encapsulate user, ticket/flight info and order info. What's more I created DataAccess class, to encapsulate all methods that connect to database.

The following are the classes I created:



User_Login Class

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AirlineTicket
{
    public class UserInfo
    {
        private string username;
        private string password;
        private int author;


        public string Username
        {
            get
            {
                return username;
            }

            set
            {
                username = value;
            }
```

Ticket/Flight Info

```
namespace AirlineTicket
{
    public class TypeInfo
    {
        private int id;

        public int Id
        {
            get { return id; }
            set { id = value; }
        }
        private String content;

        public String Content
        {
            get { return content; }
            set { content = value; }
        }

        public TypeInfo(String content,int id)
        {
            this.id = id;
            this.content = content;
```

Order Info

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AirlineTicket
{
    public class OrderInfo
    {
        private String flightNo;
        private String leaveDate;
        private String seatType;
        private int number;
        private int id;
        private String iDCard;
        private String  state;
        private int userID;
        private String username;

        public string FlightNo
        {
            get
            {
                return flightNo;
            }
```

Add DataAccess class, to encapsulate all methods that connect to database.(DataAccess.cs)

Such as: Insert, update, inquiry and delete.

```
public bool UpdateTicket(TicketInfo ticket)
{


    string Sql = string.Format("Update TicketInfo Set FlightNO='{0}', LeaveCity='{1}',Destination='{2}',

    return ExcuteSql(Sql);
}
```

```
public DataTable AllTicket()
{
    string Sql = "SELECT Id, FlightNO, LeaveCity," +
     " Destination, LeaveTime,arriveTime, SecondClass" +
     ", FirstClass ,SeatCount FROM TicketInfo";
    return GetDataSource(Sql, "Ticket");

}

public bool InsertOrder(OrderInfo orderinfo)
{

    string Sql = "Insert into OrderInfo (FlightNo,LeaveDate,SeatType,Number,IDCard,username,State) value


    return ExcuteSql(Sql);
}
```

```
public bool DeleteTicket(TicketInfo ticket)
{


    string Sql = "Delete From TicketInfo Where Id=" + ticket.Id;

    return ExcuteSql(Sql);
}

public DataTable InquireTicket(TicketInfo ticket)
{


    string Sql = "SELECT Id, FlightNO, SeatCount,LeaveCity," +
  " Destination, LeaveTime,arriveTime, SecondClass" +
  ", FirstClass FROM TicketInfo Where LeaveCity like '%" + ticket.LeaveCity + "%' and " +
  "Destination like'%" + ticket.Destination + "%'";

    return GetDataSource(Sql, "Query");
}
```

```
    public bool InsertTicket(TicketInfo ticket)
    {


        string Sql = string.Format("Insert into TicketInfo values('{0}','{1}','{2}','{3}','{4}','{5}'

        return ExcuteSql(Sql);
    }
```

2. Add Test class named my test .

Initialize the connection.

```
namespace AirlineTicket
{
    [TestFixture]
    public  class MyTest
    {
        public static SqlConnection conn;
        static string strConnection = ConfigurationManager.ConnectionStrings["connection"].ToString();//

        [SetUp]
        public static void init()
        {
            conn = new SqlConnection(strConnection);
        }

        [Test]
        public static void LoginTest()
        {
            conn.Open();
            try
            {
                SqlCommand cmd = new SqlCommand("select * from User_Login where UserID='1' and Password='
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                da.Fill(dt);
```

Test QueryFlightInfo

```
        [Test]
        public static void QueryFlightsTest()
        {
            conn.Open();
            try
            {
                SqlCommand cmd = new SqlCommand("select * from TicketInfo", conn);
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                da.Fill(dt);
                Assert.AreEqual(true, dt.Rows.Count > 0);
            }
            catch
            {
                Assert.AreEqual(1, 0);
            }
            finally
            {
               conn.Close();
            }
        }
```

Login Test:

```
        [Test]
        public static void AddLogTest()
        {
            conn.Open();
            try
            {
                SqlCommand cmd = new SqlCommand("insert into User_Login values("
                            + "'UserID'" + ","
                            + "'UserName.'" +"'Pwd'" +")", conn);
                Assert.AreEqual(1, cmd.ExecuteNonQuery());
            }
            catch
            {
                Assert.AreEqual(1, 0);
            }
            finally
            {
                conn.Close();
            }
        }
    }
```

Update Flight Test:

```
        [Test]
        public static void UpdateFlightsTest()
        {
            conn.Open();
            try
            {
                SqlCommand cmd = new SqlCommand("Update TicketInfo set SeatCount=SeatCount+1 where Flight
                Assert.AreEqual(1, cmd.ExecuteNonQuery());
            }
            catch
            {
                Assert.AreEqual(1, 0);
            }
            finally
            {
                conn.Close();
            }
        }
        [Test]
```

Load in the Nunit, all pass