ASP.Net </>

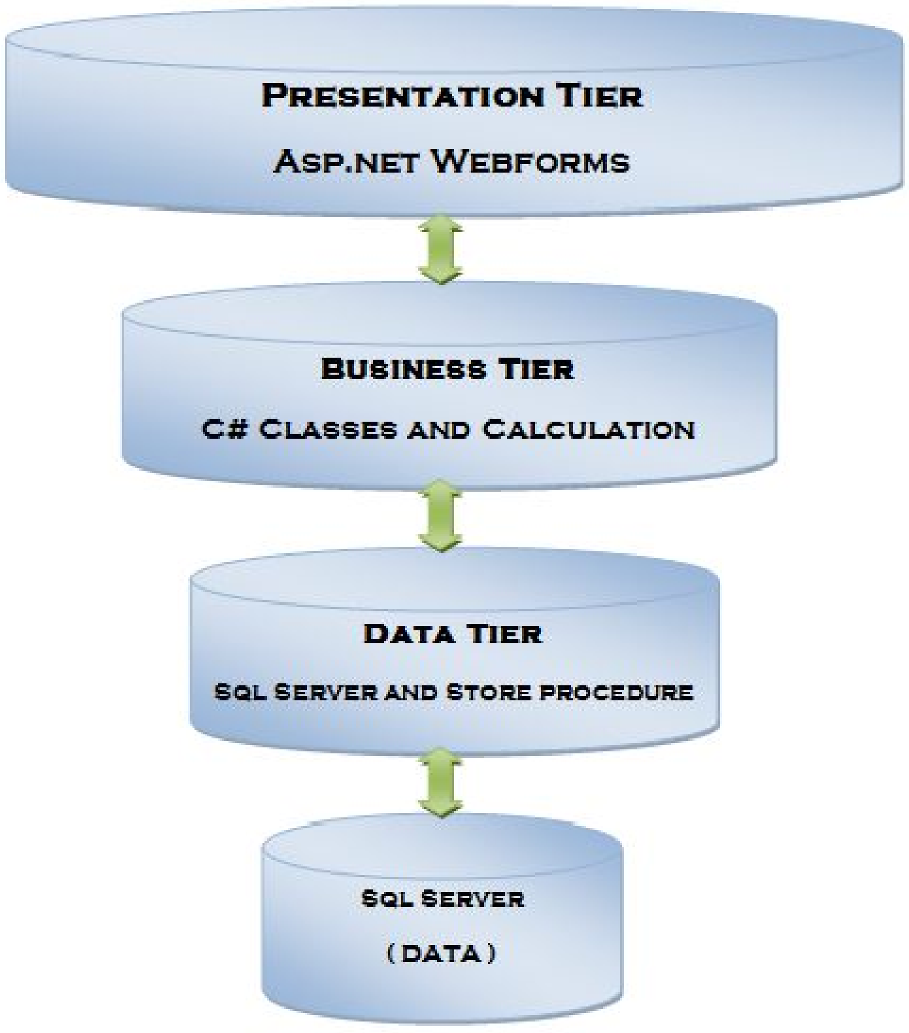
Create and Implement 3-Tier Architecture in ASP.Net

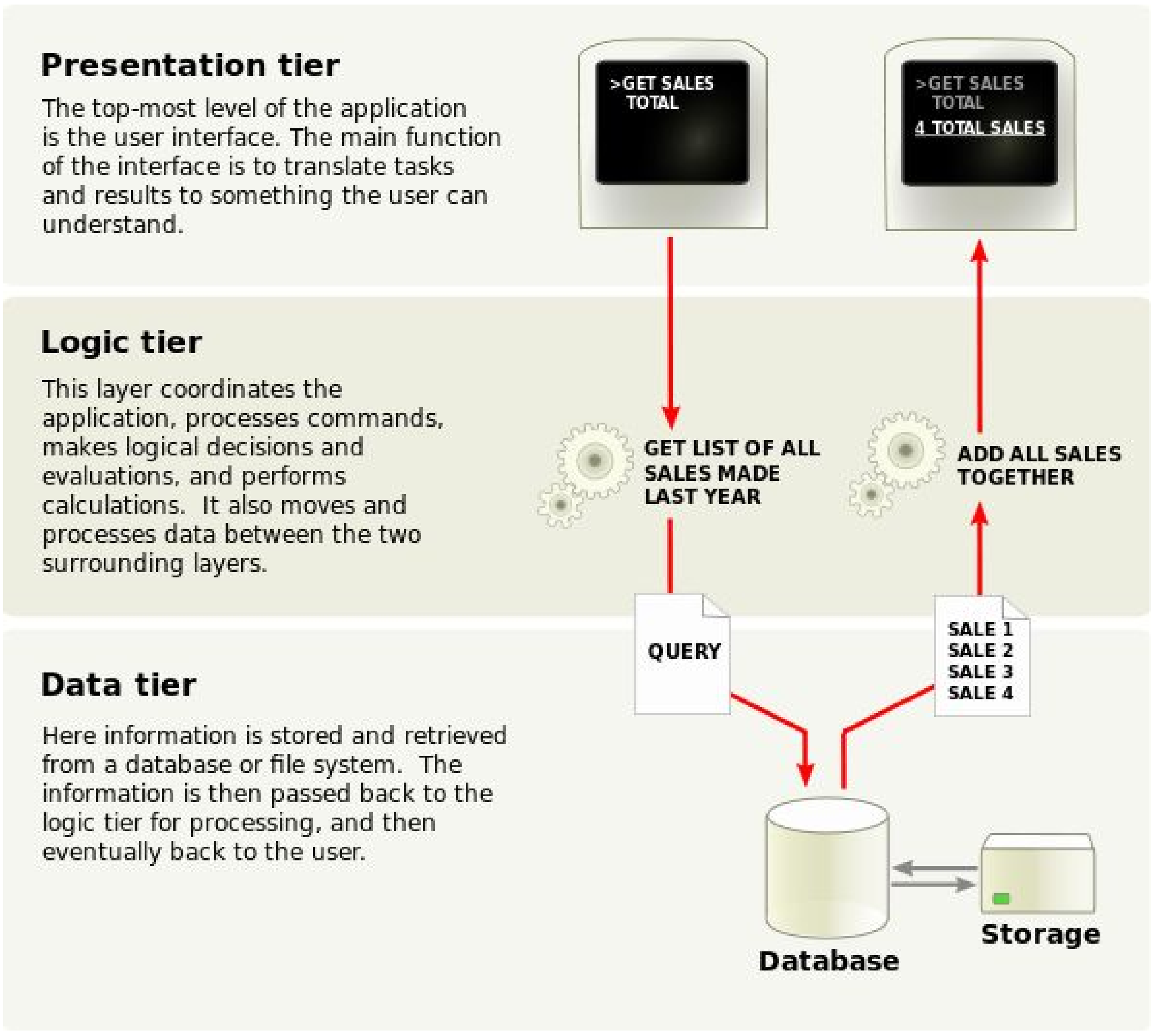
By Saineshwar Bageri on Dec 29, 2013

This article explains how to create and implement a 3-tier architecture for our project in ASP.Net.

Download Files: ThreeTierApp.rar | SQLScripts.zip

This article explains how to create and implement a 3-tier architecture for our project in ASP.Net.




Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
Database
Storage

What is a Layer?

A layer is a reusable portion of code that performs a specific function.

In the .NET environment, a layer is usually set up as a project that represents this specific function. This specific layer is in charge of working with other layers to perform some specific goal.

Let’s briefly look at the latter situation first.

Data Layer

A DAL contains methods that helps the Business Layer to connect the data and perform required actions, whether to return data or to manipulate data (insert, update, delete and so on).

Business Layer

A BAL contains business logic, validations or calculations related to the data.

Though a web site could talk to the data access layer directly, it usually goes through another layer called the Business Layer. The Business Layer is vital in that it validates the input conditions before calling a method from the data layer. This ensures the data input is correct before proceeding, and can often ensure that the outputs are correct as well. This validation of input is called business rules, meaning the rules that the Business Layer uses to make “judgments” about the data.

Presentation Layer

The Presentation Layer contains pages like .aspx or Windows Forms forms where data is presented to the user or input is taken from the user. The ASP.NET web site or Windows Forms application (the UI for the project) is called the Presentation Layer. The Presentation Layer is the most important layer simply because it’s the one that everyone sees and uses. Even with a well structured business and data layer, if the Presentation Layer is designed poorly, this gives the users a poor view of the system.

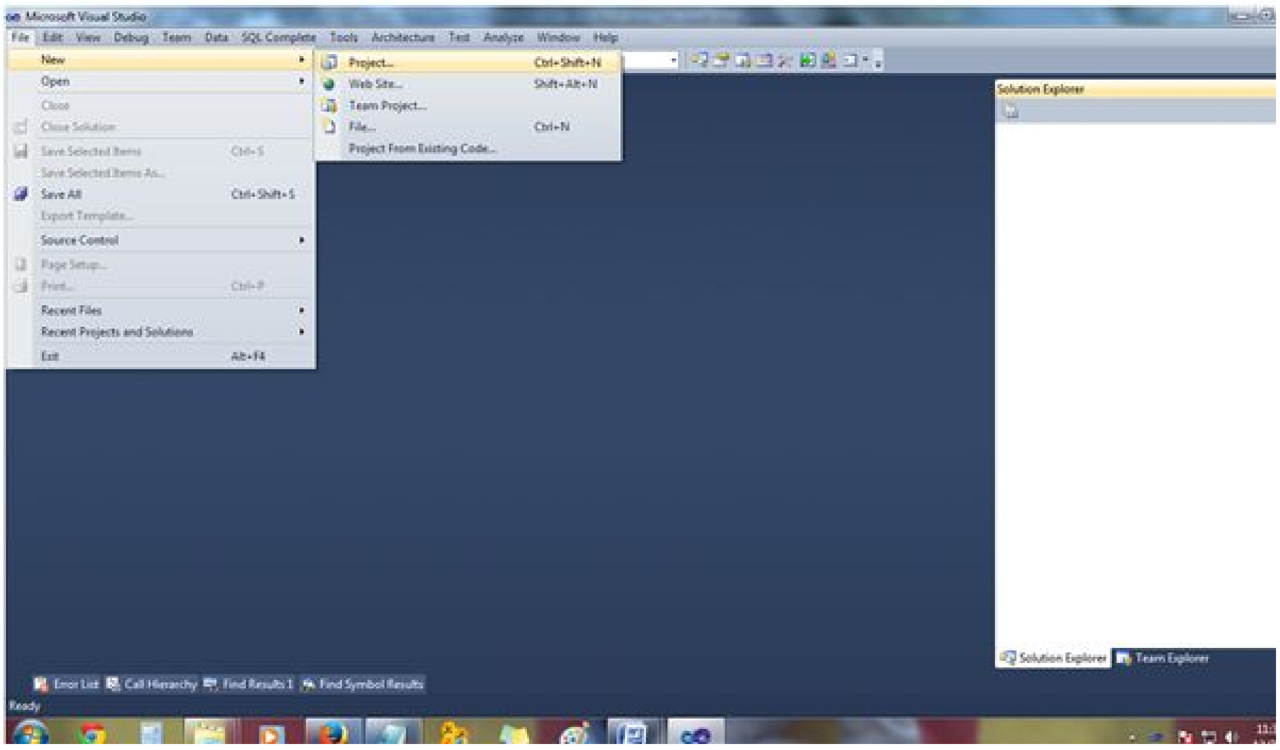
Advantages of a Three-Tier Architecture

The main characteristic of a Host Architecture is that the application and databases reside on the same host computer and the user interacts with the host using an unfriendly dumb terminal. This architecture does not support distributed computing (the host applications are unable to connect to a database of a strategically allied partner). Some managers find that developing a host application takes too long and it is expensive. These disadvantages consequently led to a Client-Server architecture.

A Client-Server architecture is a 2-Tier architecture because the client does not distinguish between Presentation Layer and Business Layer. The increasing demands on GUI controls caused difficulty in managing the mixture of source code from a GUI and the Business Logic (Spaghetti Code). Further, the Client Server Architecture does not support enough the Change Management. Let us suppose that the government increases the Entertainment tax rate from 4% to 8 %, then in the Client-Server case, we need to send an update to each client and they must update synchronously on a specific time otherwise we may store invalid or incorrect information. The Client-Server Architecture is also a burden to network traffic and resources. Let us assume that about five hundred clients are working on a data server. Then we will have five hundred ODBC connections and several ruffian record sets, that must be transported from the server to the clients (because the Business Layer remains in the client side). The fact that Client-Server does not have any caching facilities like in ASP.NET causes additional traffic in the network. Normally, a server has better hardware than the client, therefore it is able to compute algorithms faster than a client, so this fact is also an additional argument in favor of the 3-Tier Architecture. This categorization of the application makes the function more reusable easily and it becomes too easy to find the functions that have been written previously. If a programmer wants to make further updates in the application then he can easily understand the previous written code and can update it easily.

Let's start creating a 3-Tier Architecture Application.

1. Create a new project using "File" -> "New" -> "Project...".

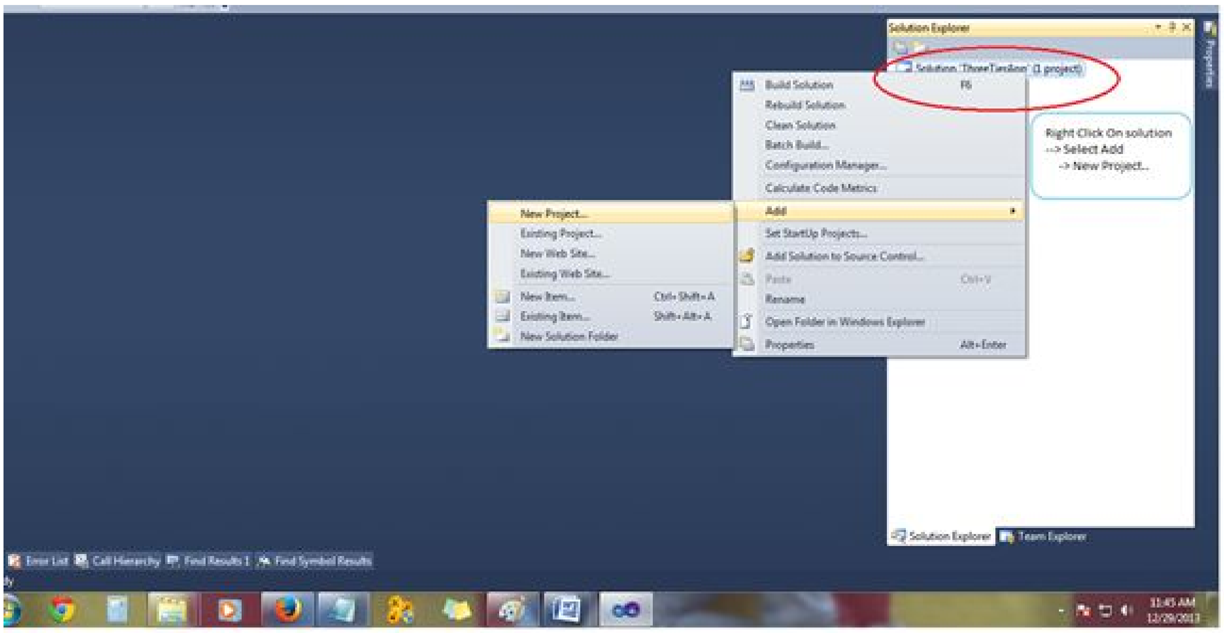

Microsoft Visual Studio
File Edit View Debug Team Data SQL Complete Tools Architecture Test Analyze Window Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Source Control
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
File... Ctrl+N
Project From Existing Code...
Solution Explorer
Error List Call Hierarchy Find Results 1 Find Symbol Results
Ready

2. In Visual C# select "Web".

(Select "ASP.NET Web application" and name it’s as: ThreeTierApp)

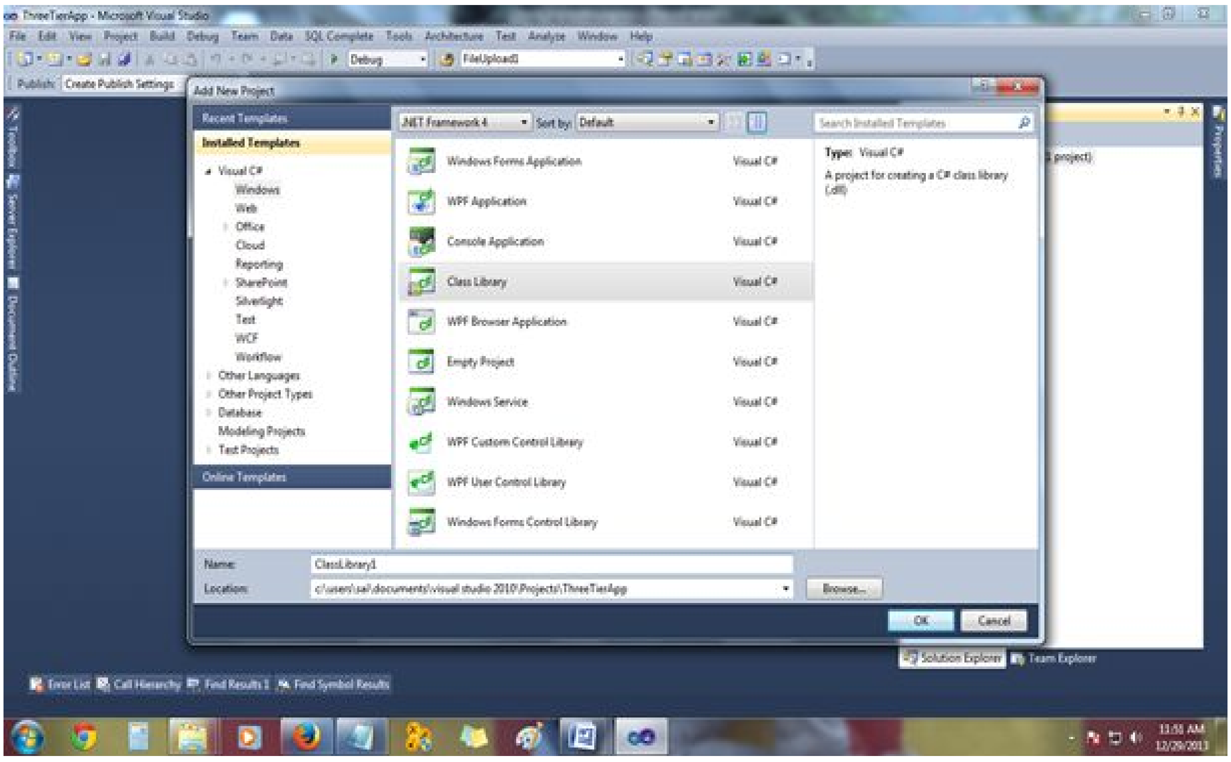
ASPNET Web application

3. How to add class library to solution:



After clicking on a new project you would see the following screen.

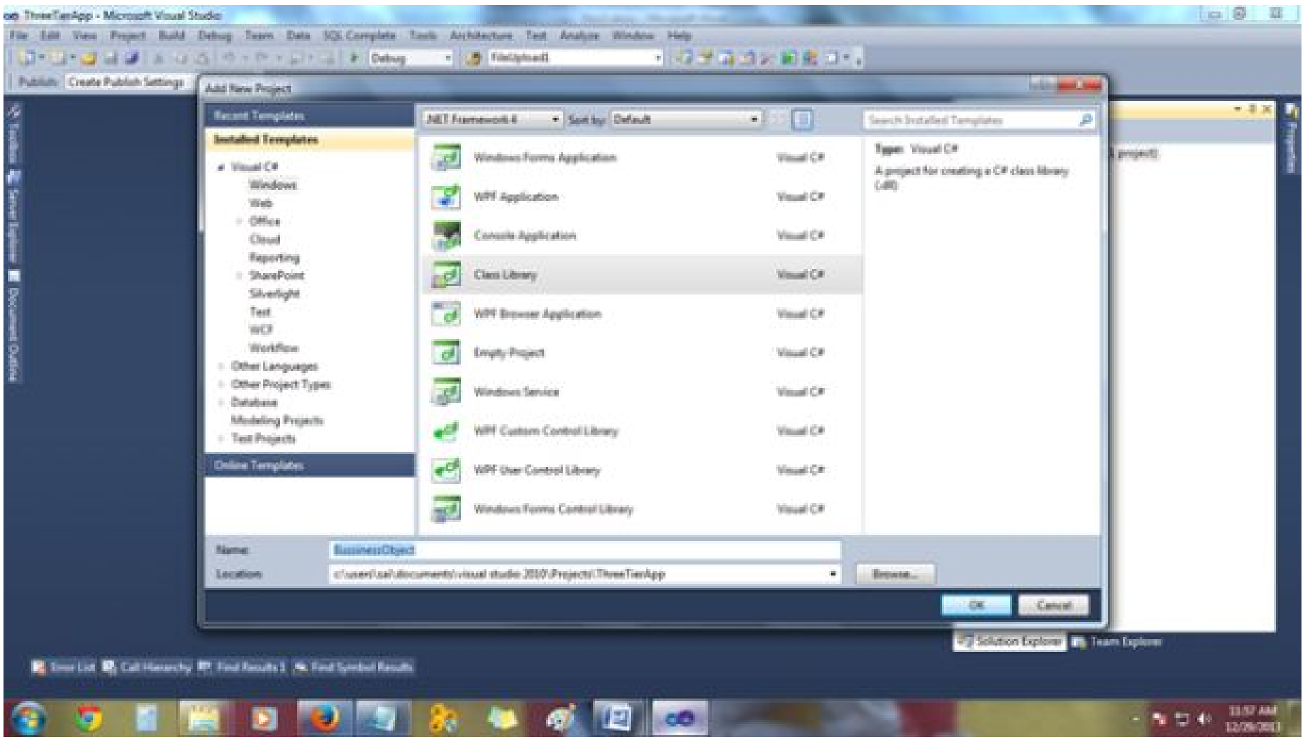
- Select "Class Library" from this and name it "BussinessObject".



Type: Visual C#
A project for creating a C# class library (.dll)

Name: ClassLibrary1
Location: c:\users\sai\documents\visual studio 2010\Projects\ThreeTierApp

- Provide the name of the Class library as "BussinessObject".



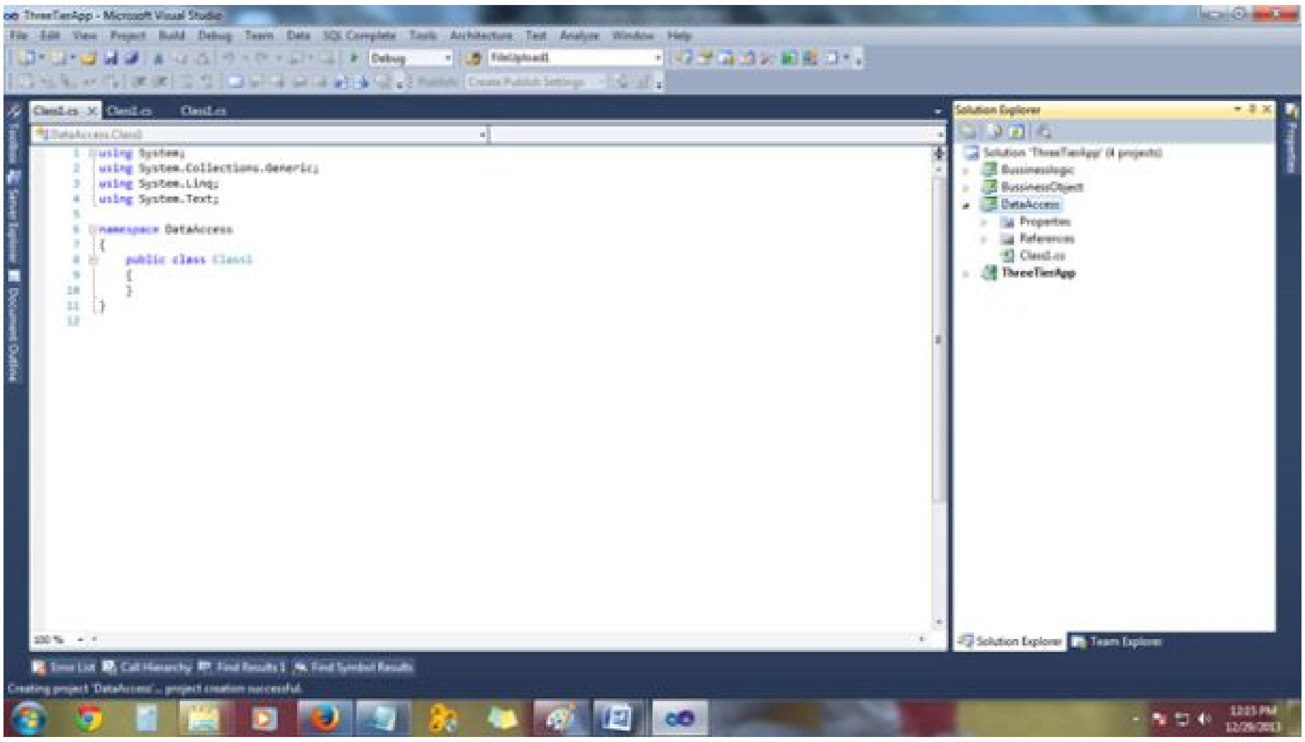
Type: Visual C#
A project for creating a C# class library (.dll)

Name: BussinessObject
Location: c:\users\sai\documents\visual studio 2010\Projects\ThreeTierApp

- Add another Class Library to our project.

  (In the same way as you added a BussinessObject.)

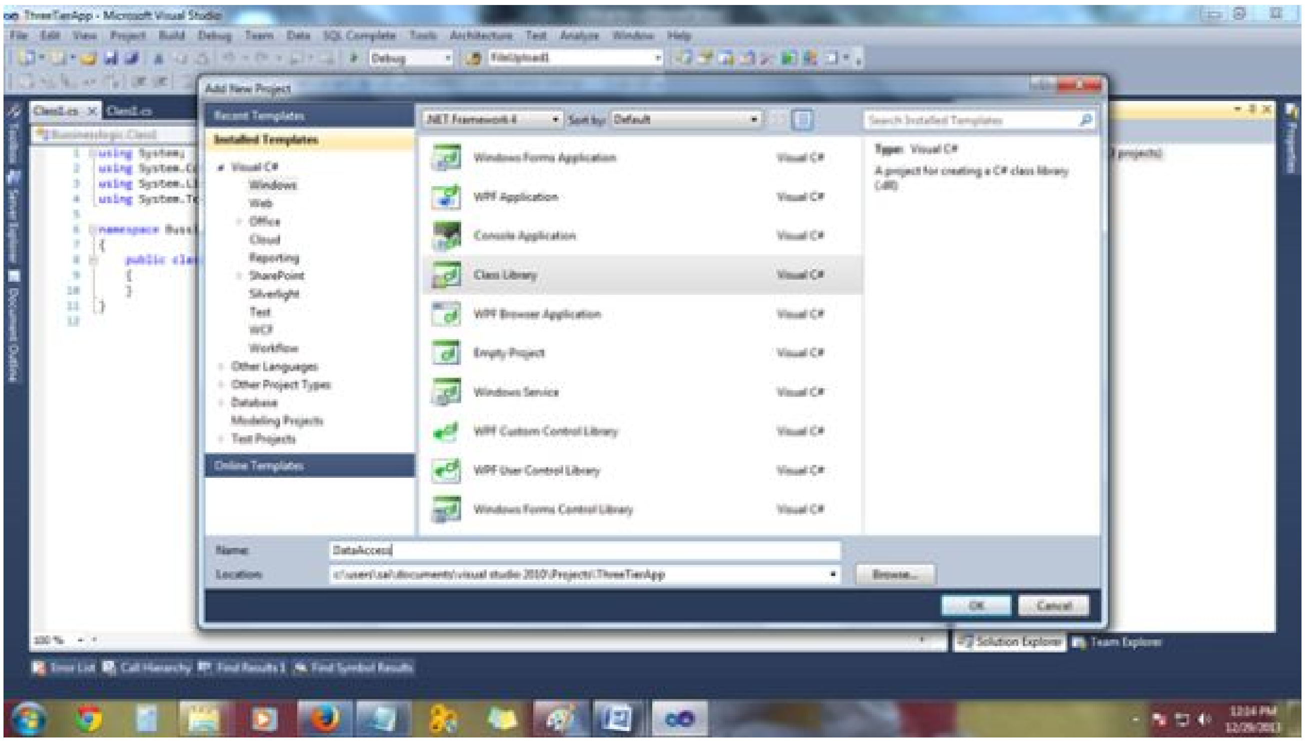
- Name the Class library "Bussinesslogic".
- After adding you will see as in this view.



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

namespace DataAccess
{
    public class Class1
    {
    }
}
```

Creating project 'DataAccess'... project creation successful.

- Add the last Class Library to our project called "Data Access Layer".
- In the same way as you added BussinessObject.
- Name the Class Library "DataAccess".



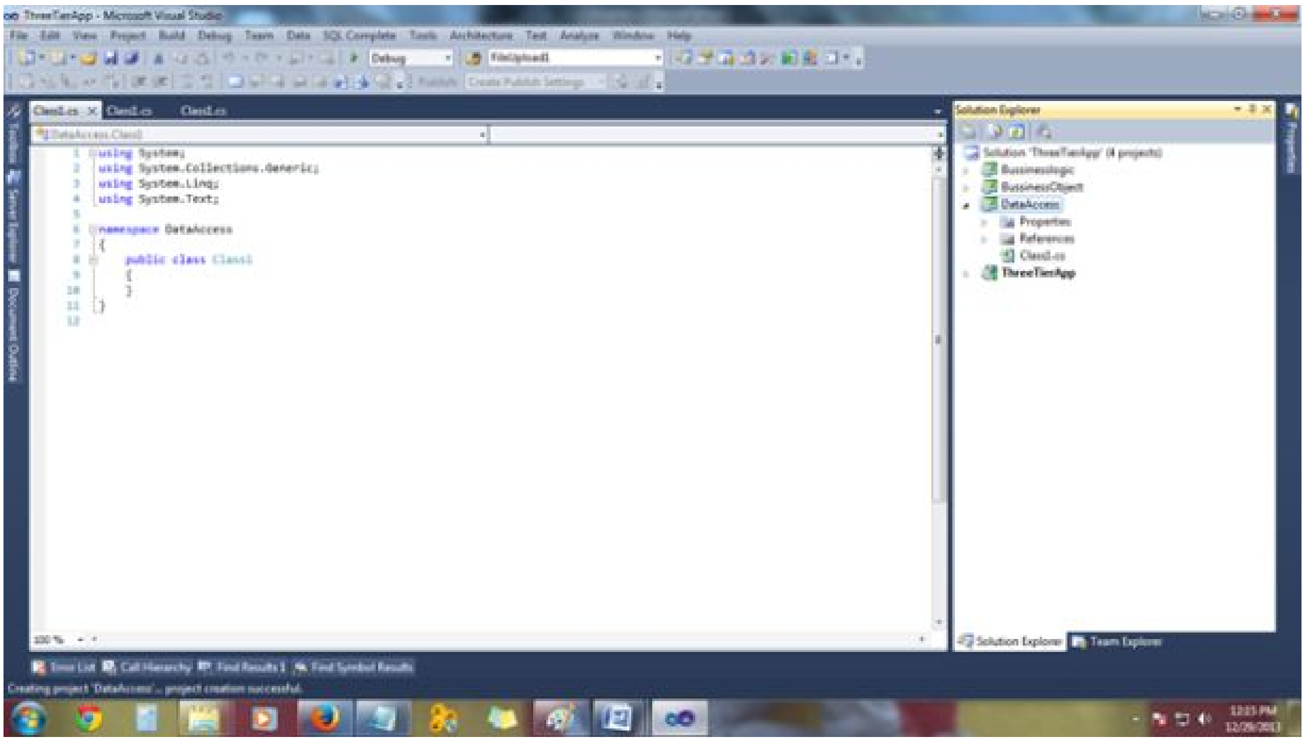
Type: Visual C#

A project for creating a C# class library (.dll)

Name: DataAccess

Location: c:\users\sai\documents\visual studio 2010\Projects\ThreeTierApp

After adding, your solution would look like this:

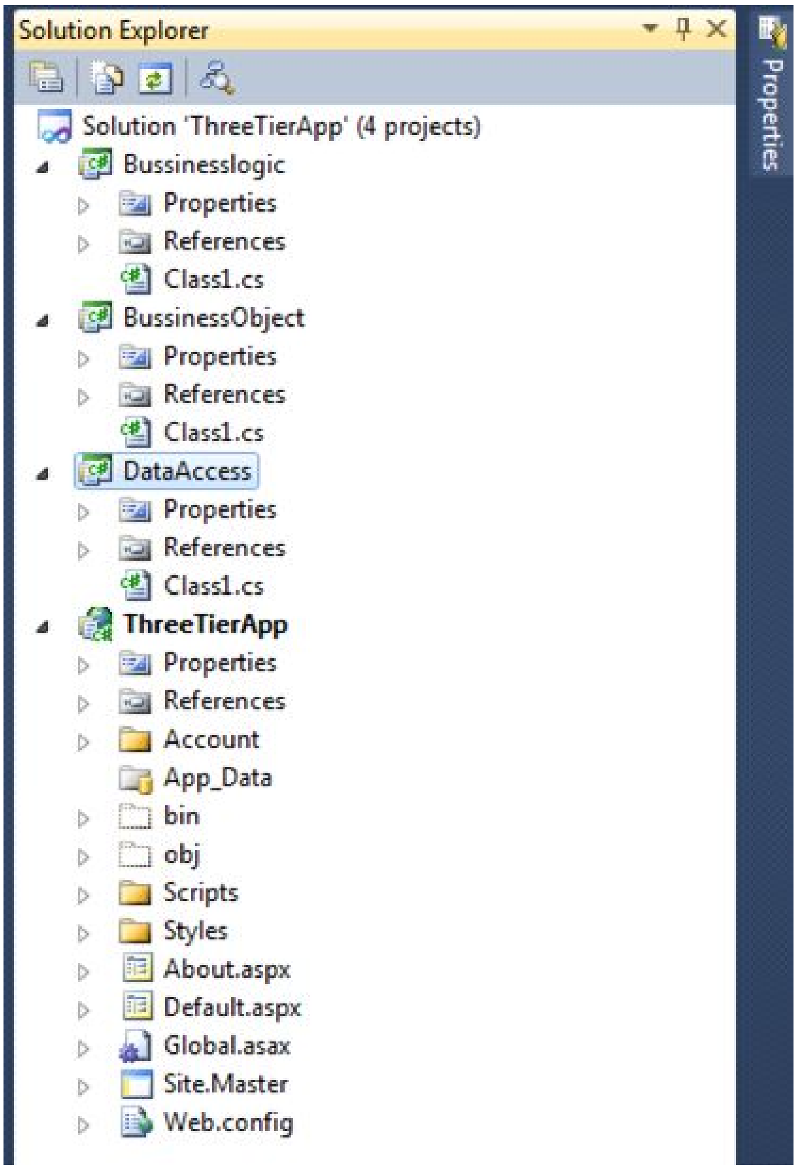


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DataAccess
{
    public class Class1
    {
    }
}
```

ThreeTierApp - Microsoft Visual Studio
Solution Explorer
Solution 'ThreeTierApp' (4 projects)
Bussinesslogic
BussinessObject
DataAccess
Properties
References
Class1.cs
ThreeTierApp
Creating project 'DataAccess'... project creation successful.

Now we are done. Add all layers to our project.

You can see the three tiers in the image given below.



```
Solution 'ThreeTierApp' (4 projects)
  Bussinesslogic
    Properties
    References
    Class1.cs
  BussinessObject
    Properties
    References
    Class1.cs
  DataAccess
    Properties
    References
    Class1.cs
  ThreeTierApp
    Properties
    References
    Account
    App_Data
    bin
    obj
    Scripts
    Styles
    About.aspx
    Default.aspx
    Global.asax
    Site.Master
    Web.config
```

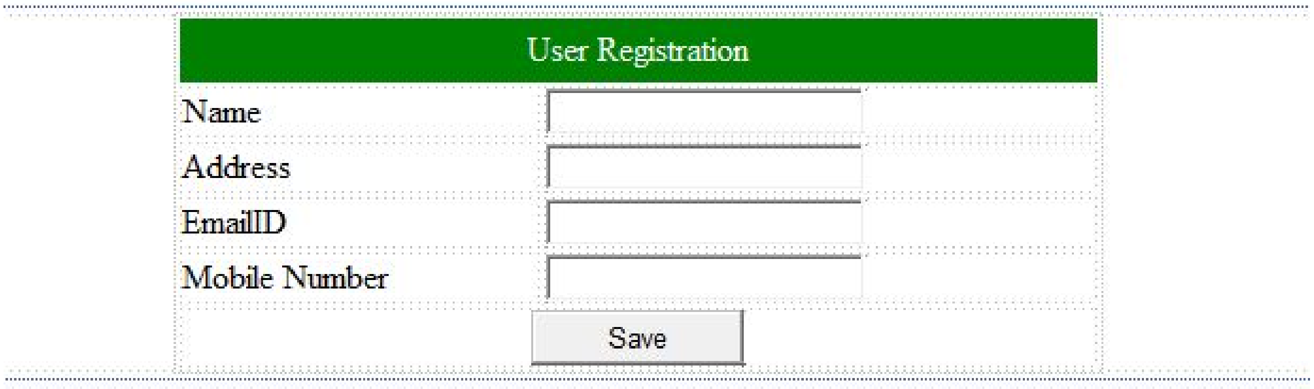
Basically a 3-Tier architecture contains the following 3 layers:

1. Application Layer or Presentation Layer (our web form and UI Part)
2. Business Logic Layer (Bussinesslogic)
3. Data Access Layer (DataAccess)

Here I will explain each layer with a simple example that is a User Registration Form.

Presentation Layer

Here, I have designed a user interface for the Presentation Layer.



Here is the design of "Userregistration.aspx":

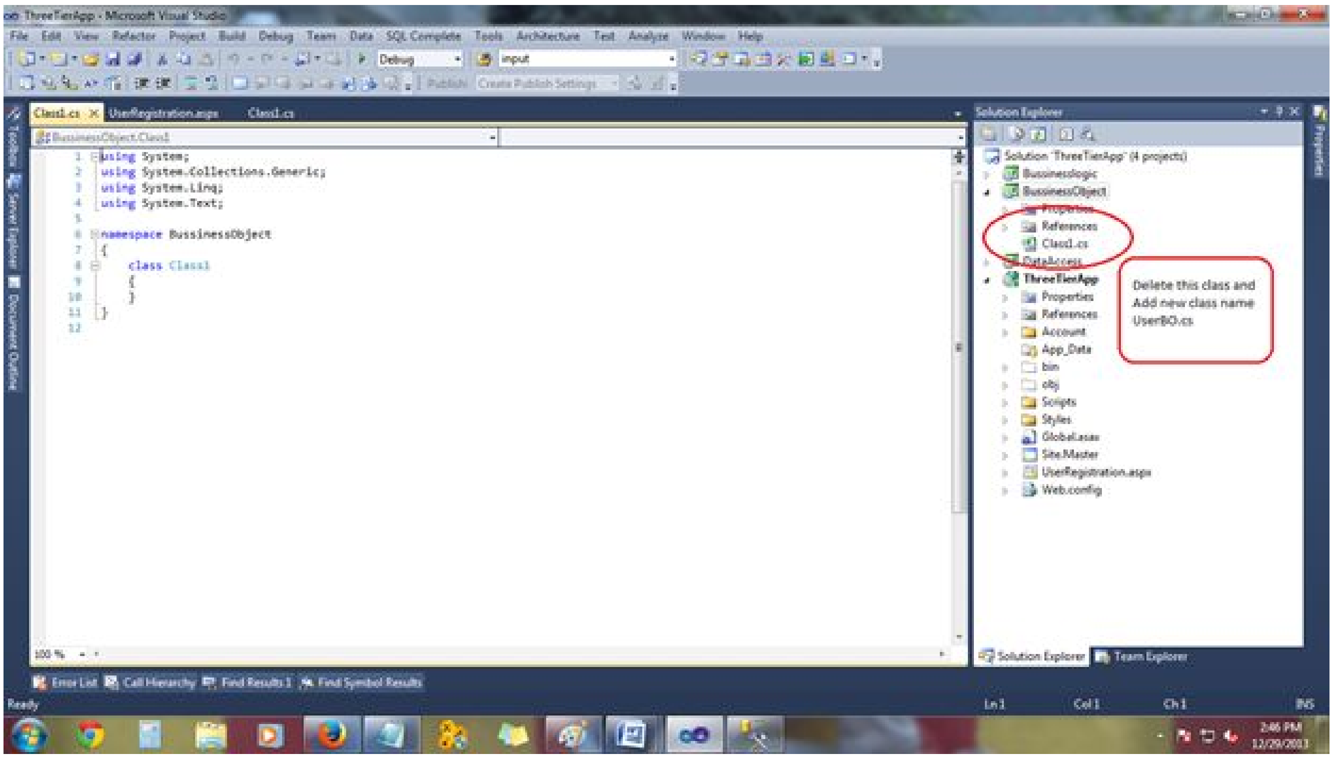
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div style="margin: 0px auto; padding-left: 370px; padding-right: 30px;
                                                  overflow: auto;">
        <div>
            <table width="50%">
                <tr>
                    <td colspan="2" style="background-color: Green; height: 30px;
                                                  color: White;" align="center">
                        User Registration
                    </td>
                </tr>
                <tr>
                    <td> Name </td>
                    <td>
          <asp:TextBox ID="txtname" Width="150px" runat="server"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td>
                        Address </td>
                    <td>
            <asp:TextBox ID="txAddress" Width="150px" runat="server"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td> EmailID </td>
             <td>
            <asp:TextBox ID="txtEmailid" Width="150px" runat="server"></asp:TextBox>
             </td>
                </tr>
                <tr>
                    <td> Mobile Number   </td>
                    <td>
            <asp:TextBox ID="txtmobile" Width="150px" runat="server"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td align="center" colspan="2">
         <asp:Button ID="BtnSave" runat="server" Width="100px" Text="Save" OnClick="BtnSave_Click" />
                    </td>
                </tr>
            </table>
        </div>
    </div>
    </form>
</body>
</html>
```

Now let’s start to create a table for saving this data using our 3-Tier Architecture.

Let's start with a bussiness object first, as in the following:

Delete Class Class1

Create a new class name, "UserBO".



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

namespace BussinessObject
{
    class Class1
    {
    }
}
```

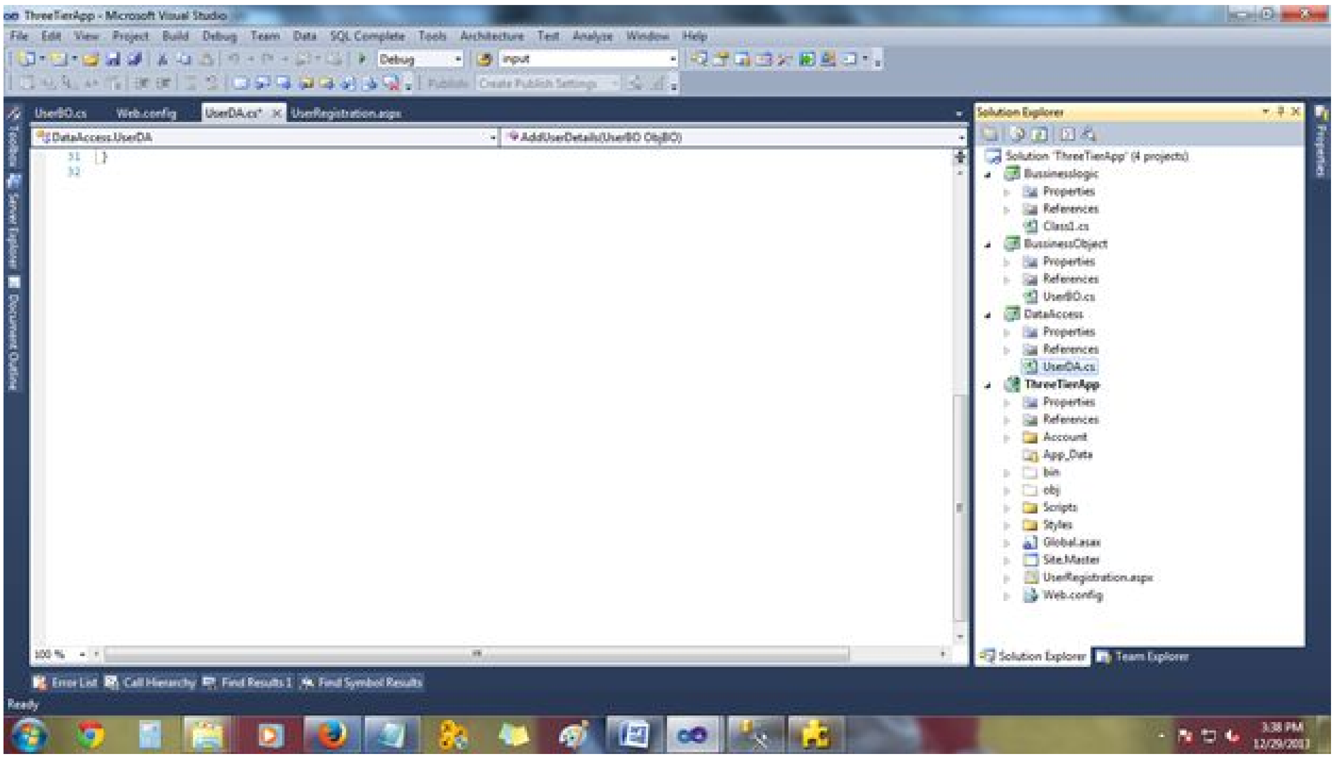
Creating UserBO.cs

Then declare variables in UserBO as in the following:

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

namespace BussinessObject
{
  Public class UserBO // Declare Class Public to Access any where
   {
       //Declaring User Registration Variables

       private string _Name;
```

```
        private string _Address;

        private string _EmailID;

        private string _Mobilenumber;

        // Get and set values

        public string Name
        {
            get
            {
                return _Name;
            }
            set
            {
                _Name = value;
            }
        }
        public string address
        {
            get
            {
                return _Address;
            }

            set
            {
                _Address = value;
            }
        }

        public string EmailID
        {
            get
            {
                return _EmailID;
            }

            set
            {
                _EmailID = value;
            }
        }

        public string Mobilenumber
        {

            get
            {
               return _Mobilenumber;
            }
            set
            {
                _Mobilenumber = value;
            }
        }
    }
}
```

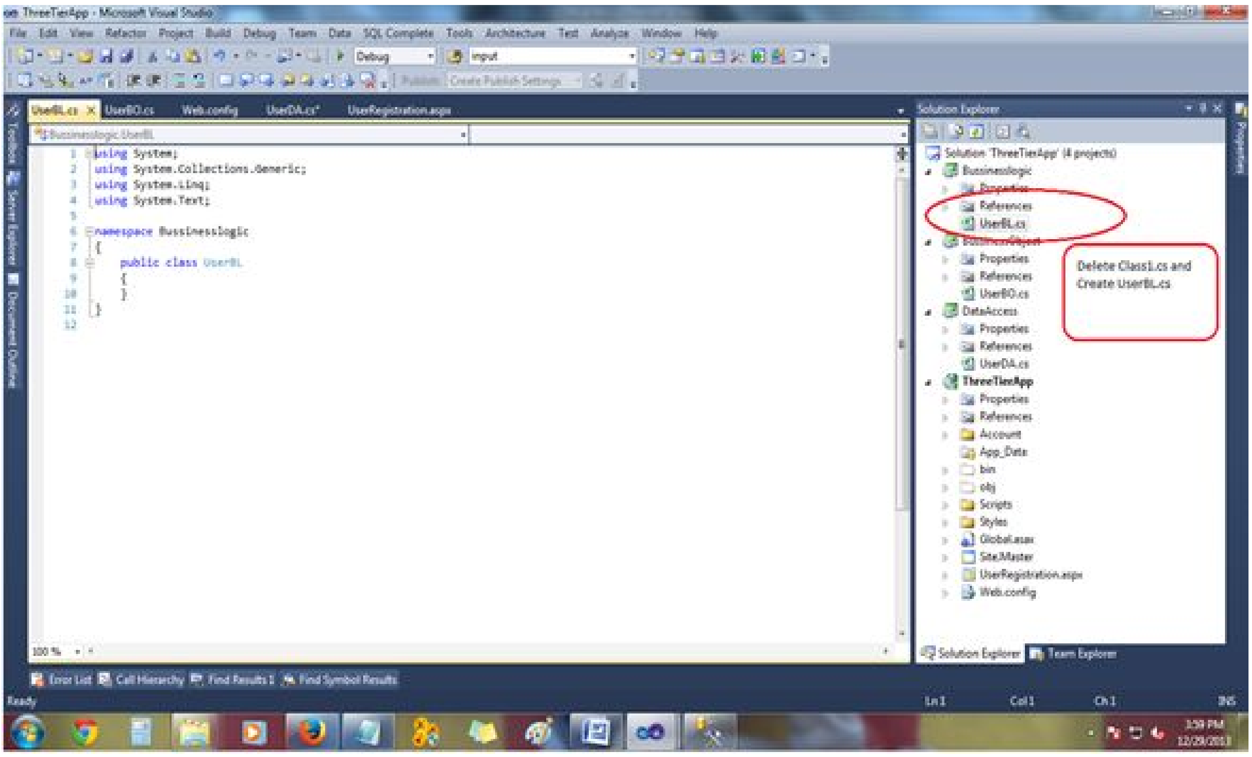
Now in the same way as we created UserBO, create a new class, UserDA, in DataAccess.


```
}
```

- In Dataaccess
- We will use this layer for communicating with the database
- All operations (insert , update, delete and selecting records) for the database is done in this layer.

Now in the same way as we created UserDA:

Create New Class UserBL.cs in (Bussinesslogic)


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

namespace Bussinesslogic
{
    public class UserBL
    {
    }
}
```

Delete Class1.cs and Create UserBL.cs

The main thing TO DO

The main thing to do nest is to add the three layers:

- UserBO.cs
- UserBL.cs
- UserDA.cs

But they are not inter connected to each other.

Let's connect them.

DataAccess connections

First right-click on DataAccess then:

- Click the "Add Reference" tab
- Select "Project"
- Inside that you will see all Layer Name
- Select BussinessObject from that and click "Ok"

Bussinesslogic connections

Then right-click on Bussinesslogic then:

- Click the "Add Reference" tab
- Select "Project"
- Inside that you will see all Layer Name.
- Select DataAccess from that and click "Ok"

New rebuild all layers.

Last step

- Right-click on the project and select "Add references"
- Click the "Add Reference" tab
- Select "Project"
- Inside that you will see all Layer Name.
- Select all add thn click "Ok"

Now build the project.

UserDA.cs (adding Records)

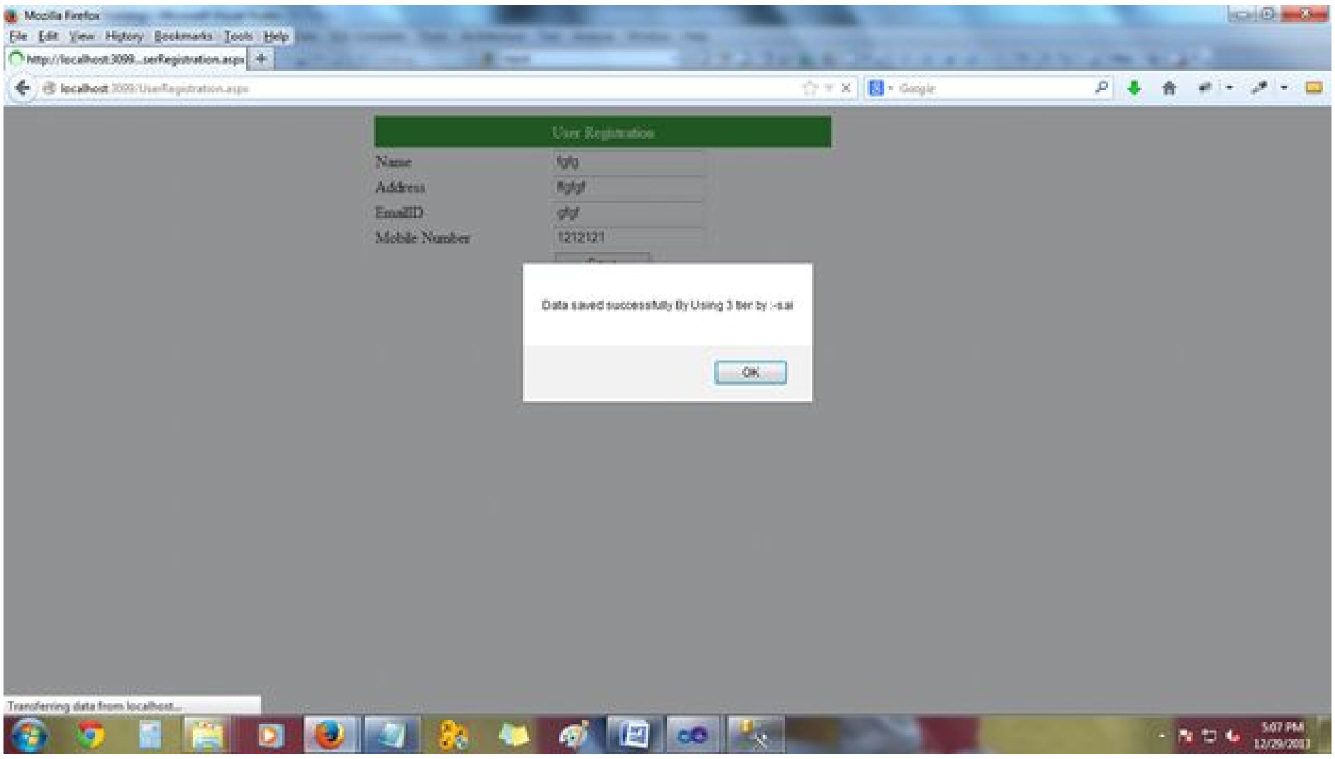
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data; // Required for using Dataset , Datatable and Sql
using System.Data.SqlClient; // Required for Using Sql
using System.Configuration; // for Using Connection From Web.config
using BussinessObject;  // for acessing bussiness object class

namespace DataAccess
{
    public class UserDA
    {
        SqlConnection con = new
```

```
          SqlConnection(ConfigurationManager.ConnectionStrings["Myconstr"].ToString());
        public int AddUserDetails(UserBO ObjBO) // passing Bussiness object Here
        {
            try
            {
                /* Because We will put all out values from our (UserRegistration.aspx)
                 To in Bussiness object and then Pass it to Bussiness logic and then to
                 DataAcess
                 this way the flow carry on*/
             SqlCommand cmd = new SqlCommand("sprocUserinfoInsertUpdateSingleItem", con);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@Name", ObjBO.Name);
                cmd.Parameters.AddWithValue("@Address", ObjBO.address);
                cmd.Parameters.AddWithValue("@EmailID", ObjBO.EmailID);
                cmd.Parameters.AddWithValue("@Mobilenumber", ObjBO.Mobilenumber);
             con.Open();
             int Result = cmd.ExecuteNonQuery();
             cmd.Dispose();
            return Result;
            }
            catch
            {
                throw;
            }
        }
    }
}
```

UserBL.cs

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

using DataAccess; // for acessing DataAccess class
using BussinessObject; // for acessing bussiness object class

namespace Bussinesslogic
{
    public class UserBL
    {
      public int SaveUserregisrationBL(UserBO objUserBL) // passing Bussiness object Here
        {
            try
            {
                UserDA objUserda = new UserDA(); // Creating object of Dataccess

              return objUserda.AddUserDetails(objUserBL); // calling Method of DataAccess

            }
            catch
            {
                throw;
            }
        }

    }
}
```

Final Output


Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost:3099/UserRegistration.aspx

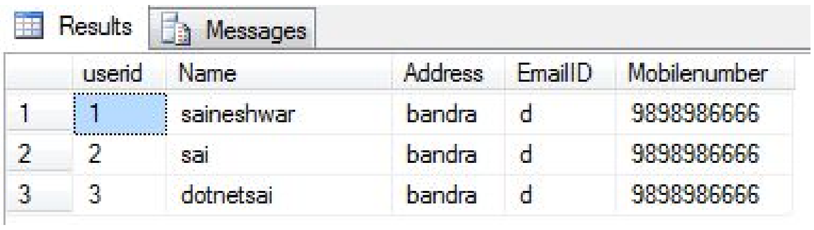
User Registration

Name	fgfg
Address	ffgfgf
EmailID	gfgf
Mobile Number	1212121

Data saved successfully By Using 3 tier by :-sai

OK
Transferring data from localhost...

```
SELECT * FROM Userinfo
```


Results
Messages

	userid	Name	Address	EmailID	Mobilenumber
1	1	saineshwar	bandra	d	9898986666
2	2	sai	bandra	d	9898986666
3	3	dotnetsai	bandra	d	9898986666

Download 100% FREE Spire Office APIs

ASP.Net ASP.Net 3-Tier architecture Implement 3 Tier Architecture Layer

Tier architecture

SAINESHWAR BAGERI TOP 500

Indian Software Developer and MVP from c-sharpcorner working on .Net Web Technology (Asp.net , C# , Sqlserver , MVC , Windows ,Console Application, javascript , jquery , json , ORM Dapper) and also freelance... Read more

http://dotnet-sai.blogspot.in

116 Rank

5.3m Read

Platinum Member

3 Times

COMMENTS (115)

TRENDING UP

01	Using Generics With C#
02	When To Use Abstract Class and Interface In Real Projects
03	Web Crawling With C#
04	What Is The Future Of XAML
05	Best Practices For ASP.NET MVC Application
06	C# Version 7.0 Announcements
07	What Is New In Visual Studio "15" Preview 4
08	Introduction To .NET Core 1.0
09	Factory Design Pattern Real World Example
10	Highest Paying Tech Jobs Of 2016

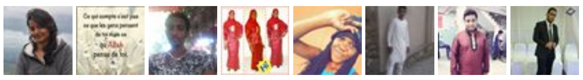
View All

Follow @csharpcorner 75.1K followers

C# Corner
402,766 likes

Like Page Share

2 friends like this



MVPs MOST VIEWED LEGENDS NOW PRIZES REVIEWS SURVEY DOWNLOADS

Hosted By CBeyond Cloud Services

ABOUT US FAQ MEDIA KIT MEMBERS STUDENTS LINKS

CONTACT US PRIVACY POLICY TERMS & CONDITIONS SITEMAP REPORT A BUG