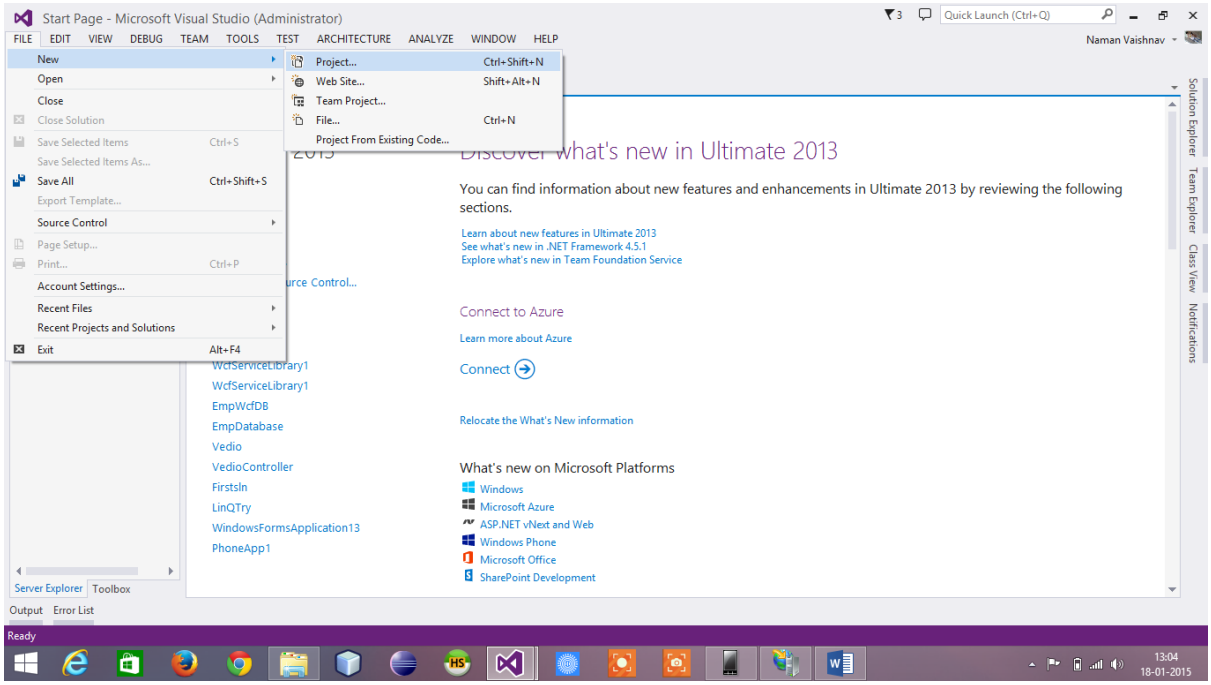
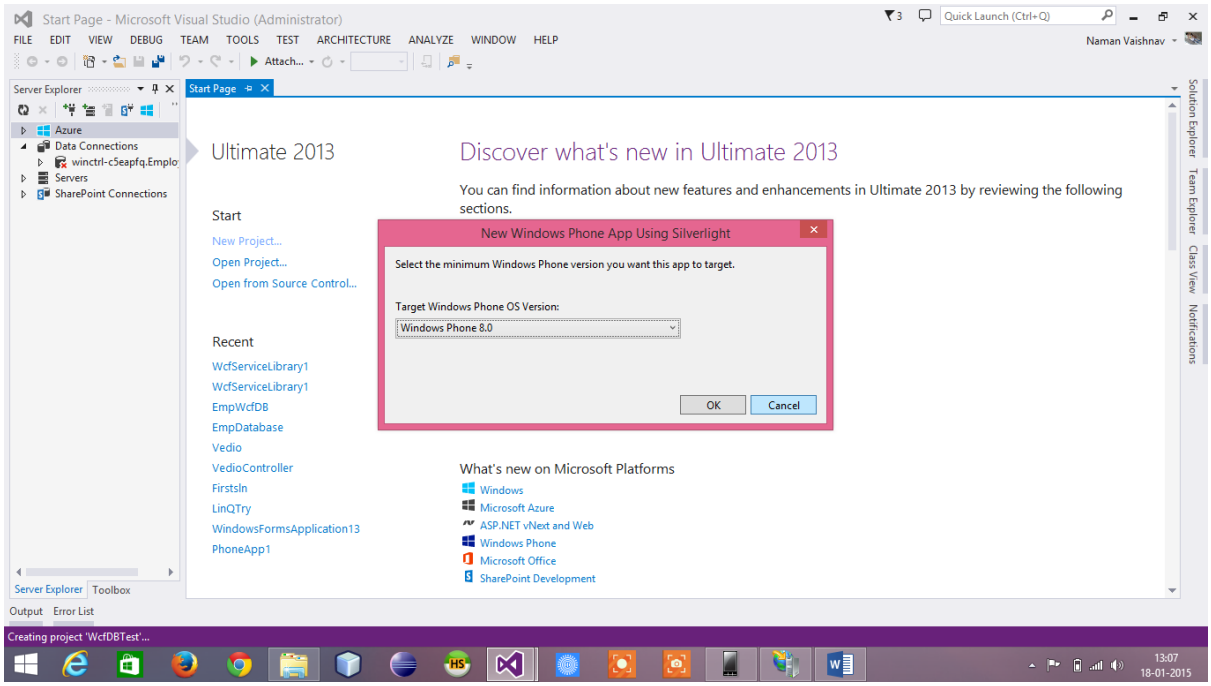
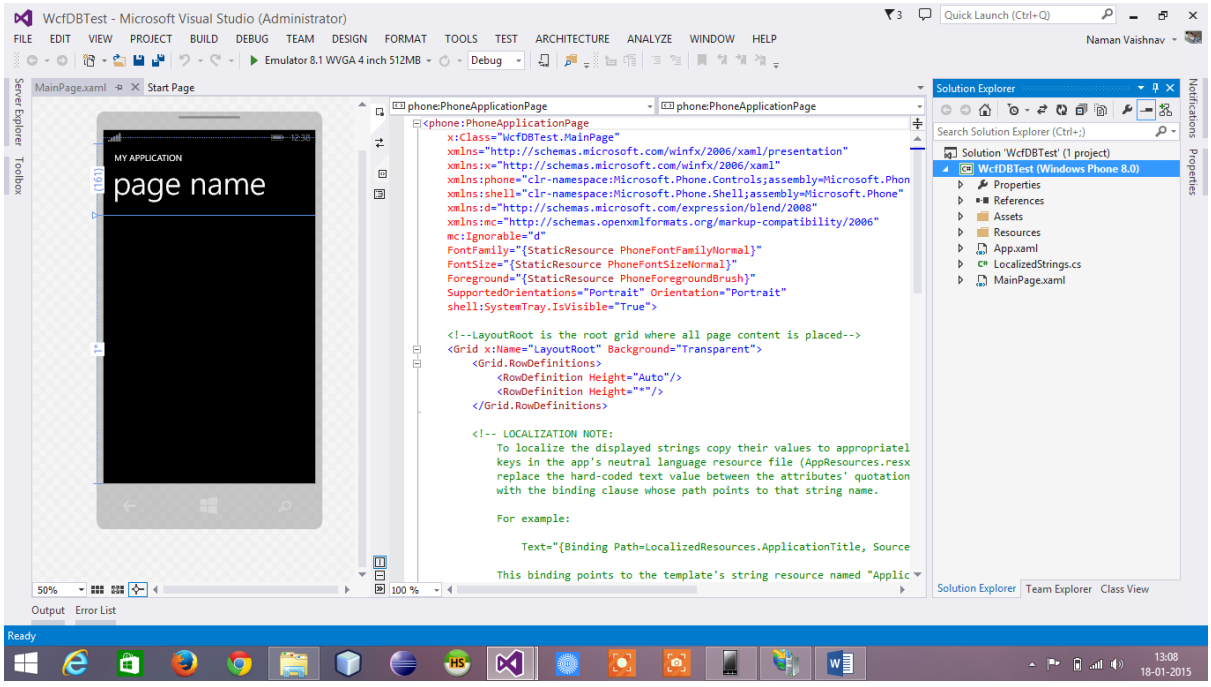
Start Visual Studio In administrative mode..

Create new Project for Windows mobile app > select 8.0 Without Fail

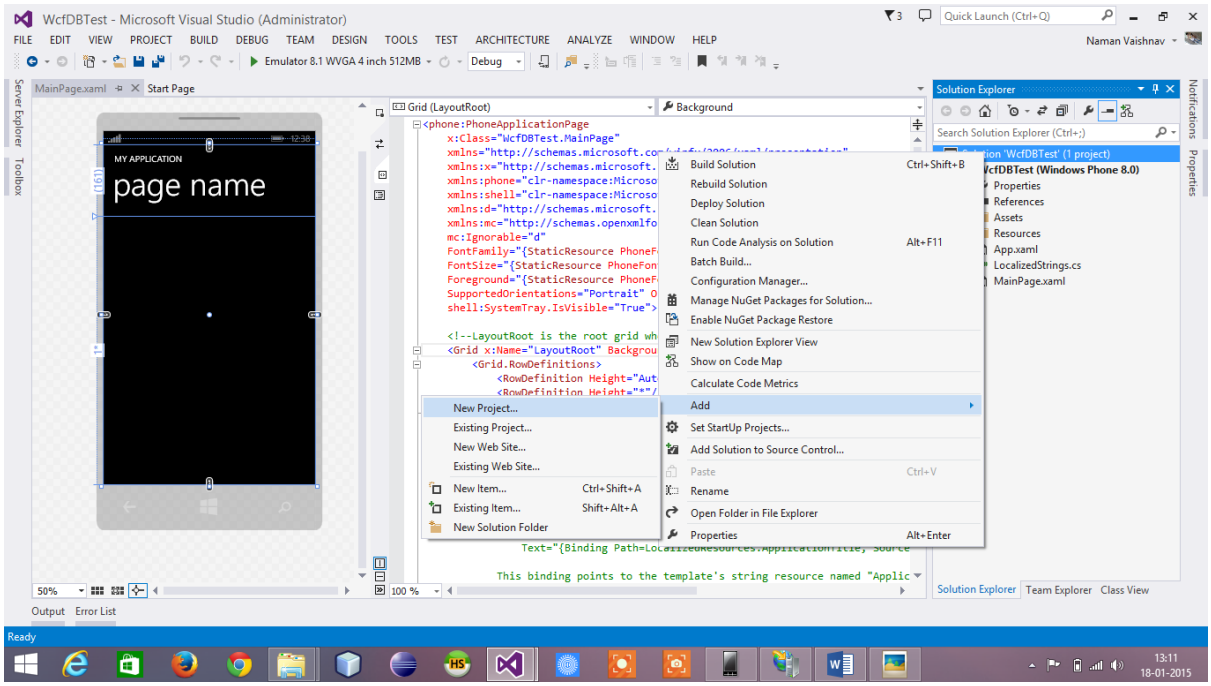


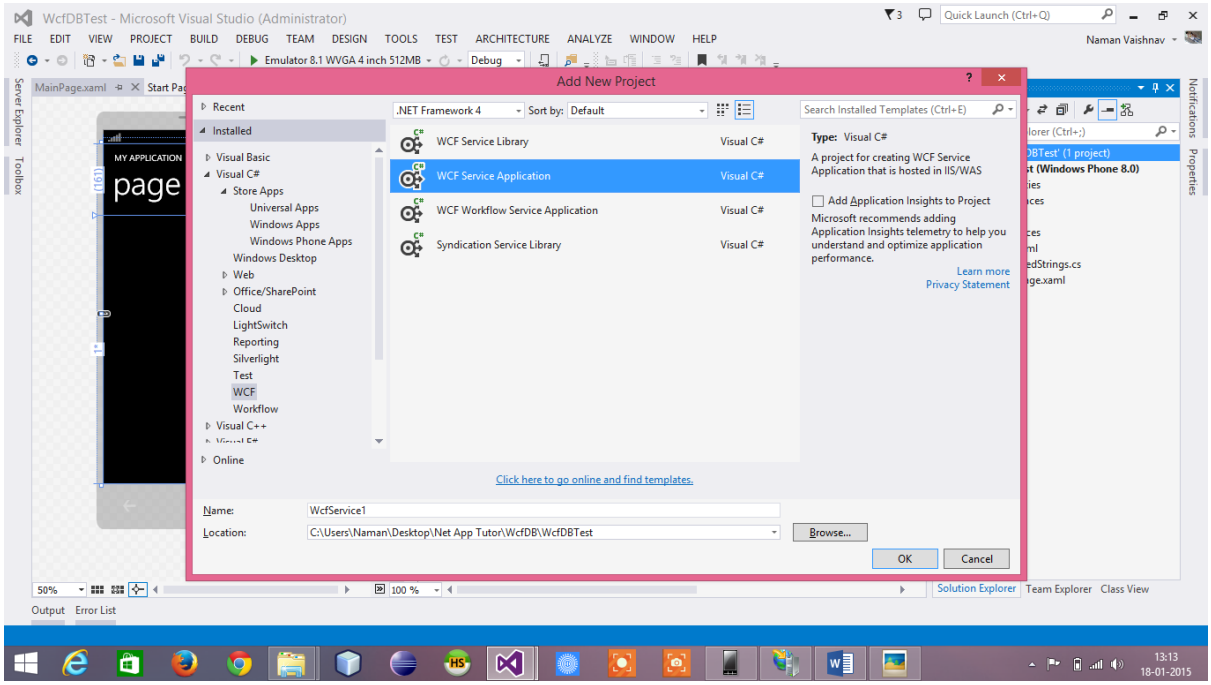


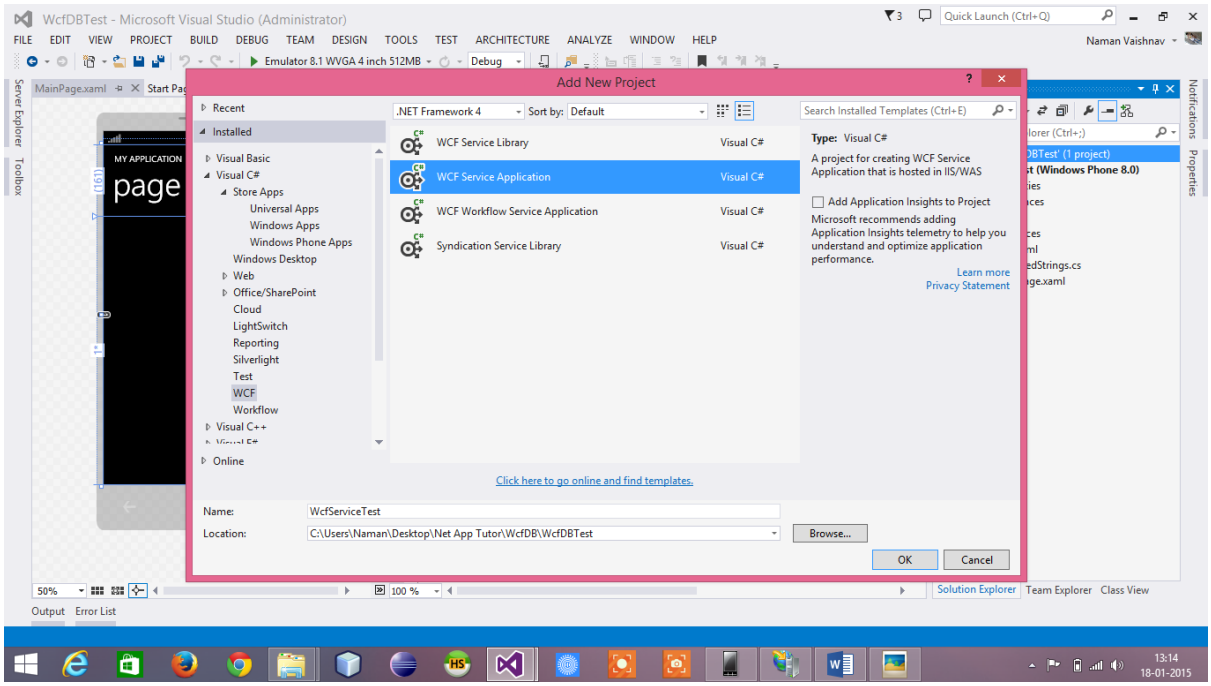


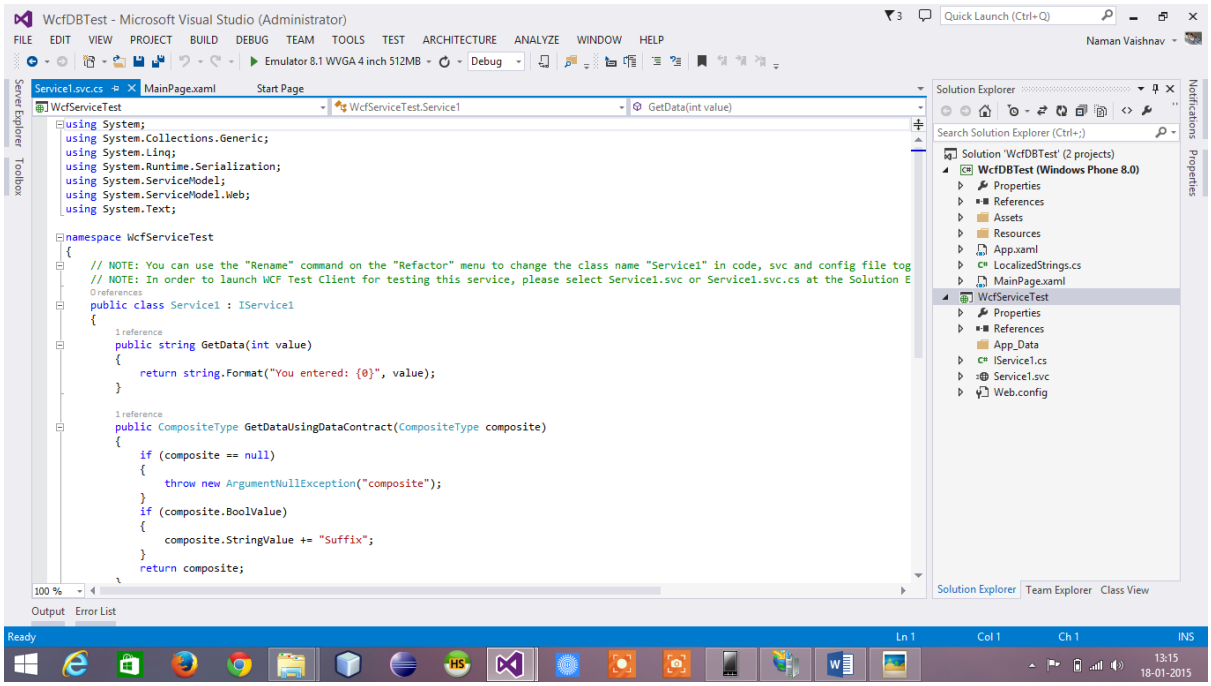
Now create new Project in the solution for wcf service

Ri8 click o Solution > add> new project > WCF > WCF Service Application

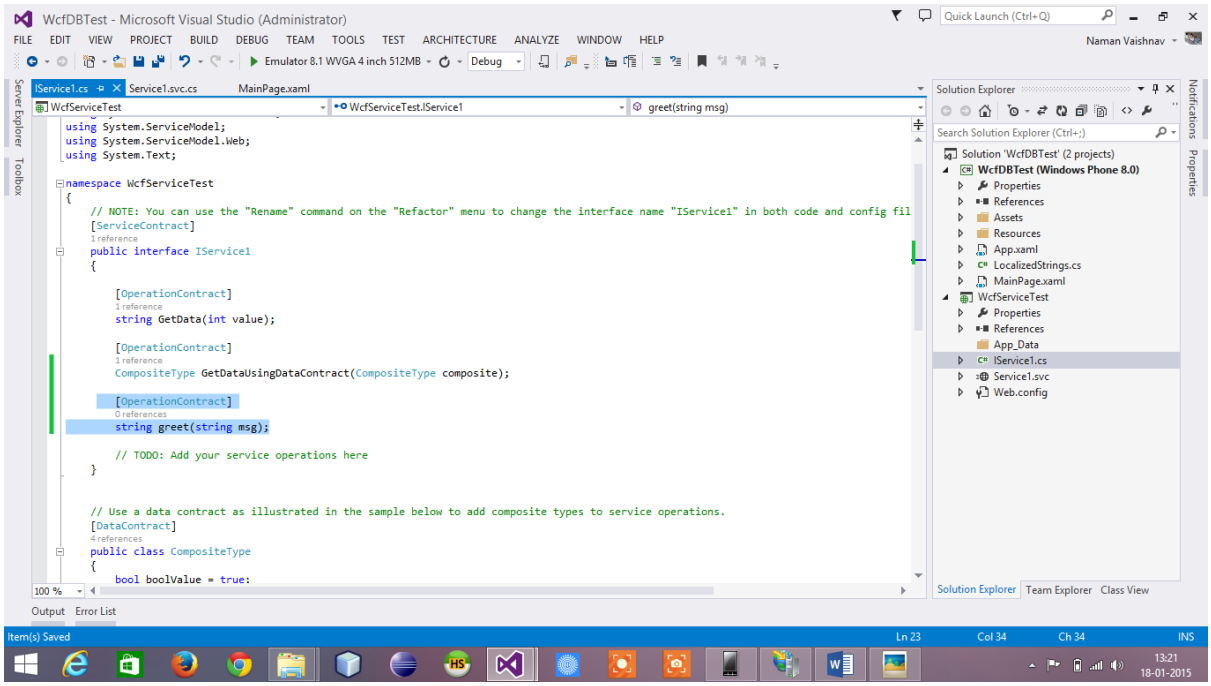

WcfDBTest - Microsoft Visual Studio (Administrator)
Build Solution Ctrl+Shift+B
Rebuild Solution
Deploy Solution
Clean Solution
Run Code Analysis on Solution Alt+F11
Batch Build...
Configuration Manager...
Manage NuGet Packages for Solution...
Enable NuGet Package Restore
New Solution Explorer View
Show on Code Map
Calculate Code Metrics
Add
Set StartUp Projects...
Add Solution to Source Control...
Paste Ctrl+V
Rename
Open Folder in File Explorer
Properties Alt+Enter
New Project...
Existing Project...
New Web Site...
Existing Web Site...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Solution Folder
MY APPLICATION
page name
Ready


WcfDBTest - Microsoft Visual Studio (Administrator)
Add New Project
Recent
Installed
Visual Basic
Visual C#
Store Apps
Universal Apps
Windows Apps
Windows Phone Apps
Windows Desktop
Web
Office/SharePoint
Cloud
LightSwitch
Reporting
Silverlight
Test
WCF
Workflow
Visual C++
Online
.NET Framework 4
Sort by: Default
WCF Service Library Visual C#
WCF Service Application Visual C#
WCF Workflow Service Application Visual C#
Syndication Service Library Visual C#
Type: Visual C#
A project for creating WCF Service Application that is hosted in IIS/WAS
Add Application Insights to Project
Microsoft recommends adding Application Insights telemetry to help you understand and optimize application performance.
Learn more
Privacy Statement
Click here to go online and find templates.
Name: WcfService1
Location: C:\Users\Naman\Desktop\Net App Tutor\WcfDB\WcfDBTest
Browse...
OK
Cancel

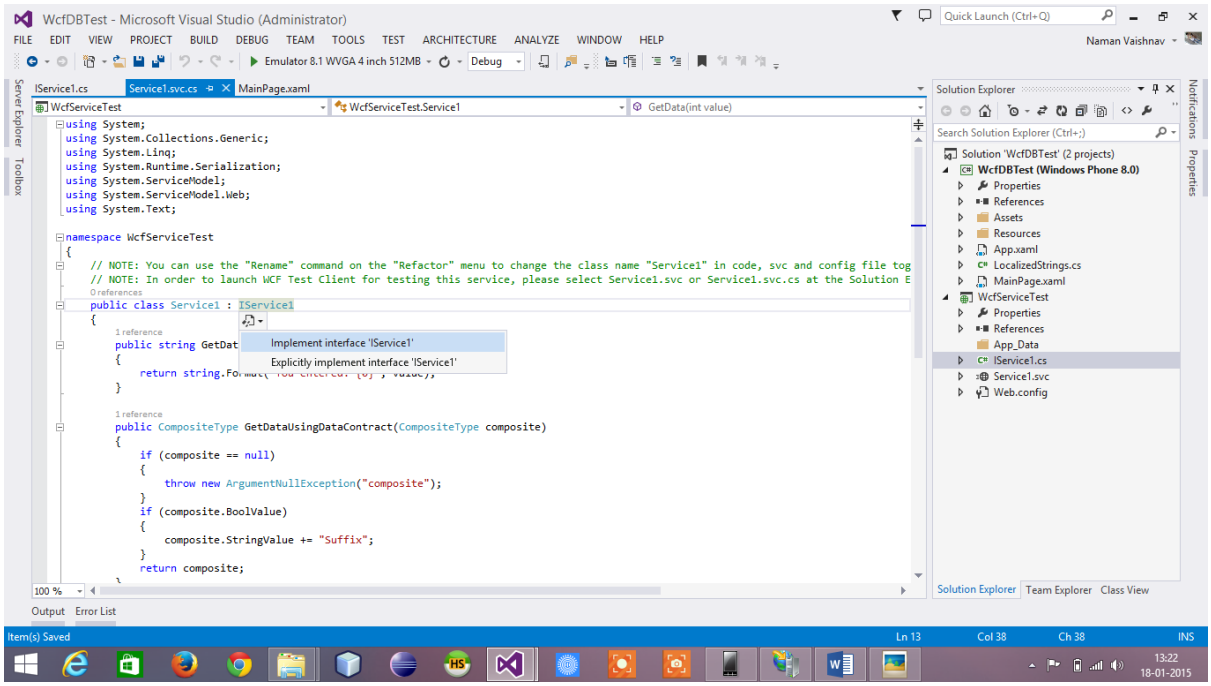




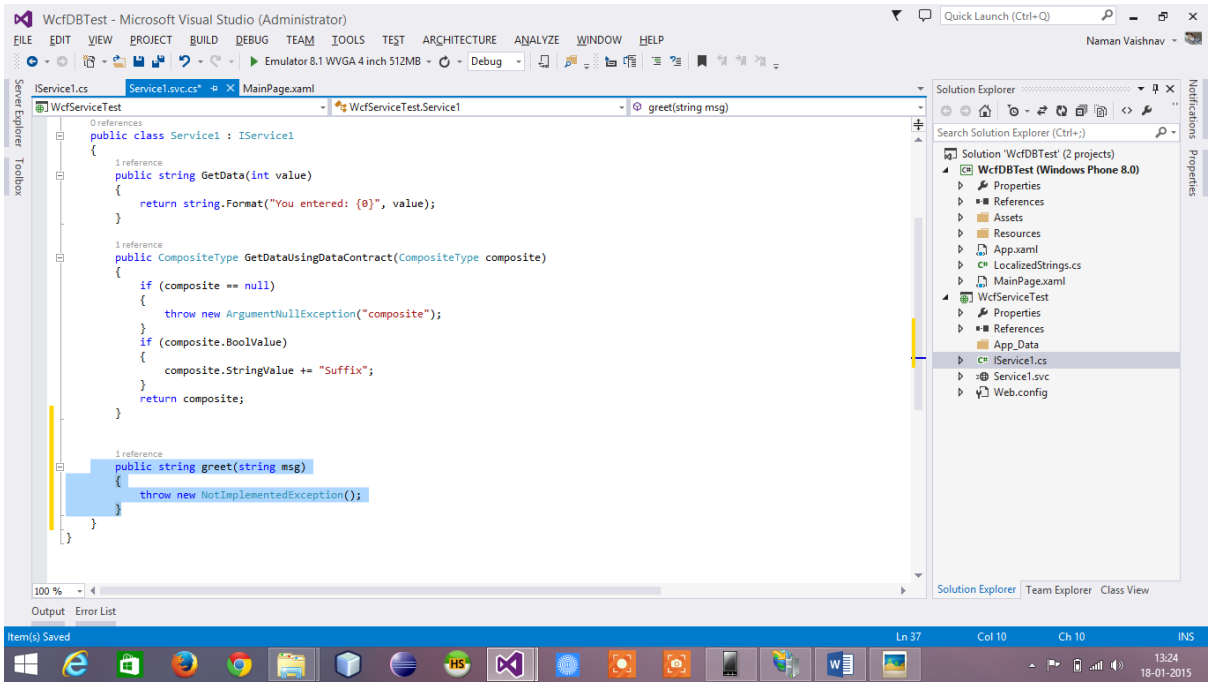
Write some code into IService1.cs File

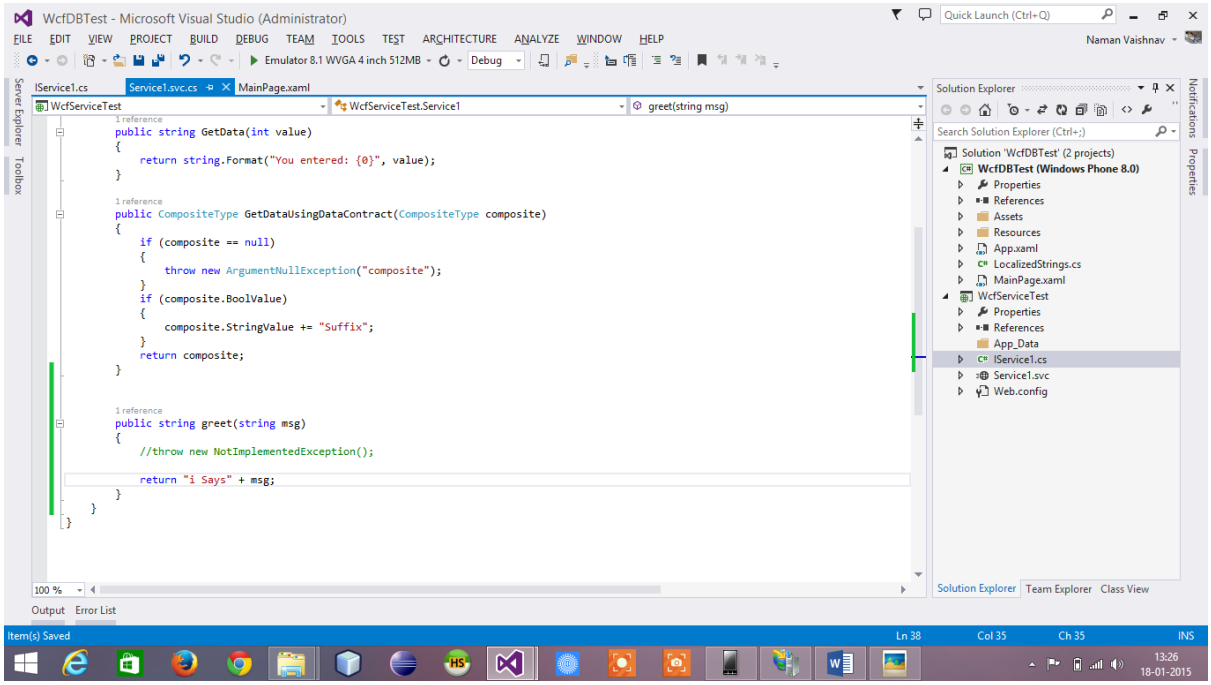


Nw go to the Service.svc.cs and click on the IService1

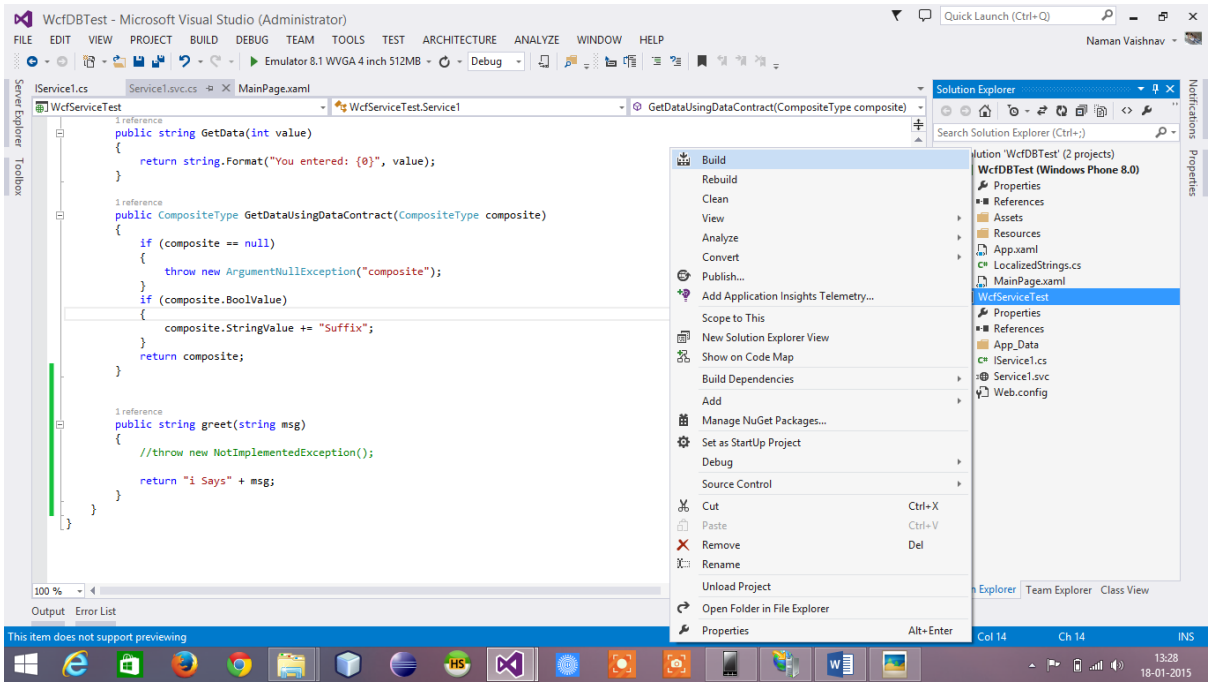


It Will add the method greet which created by us in the IService1.cs File

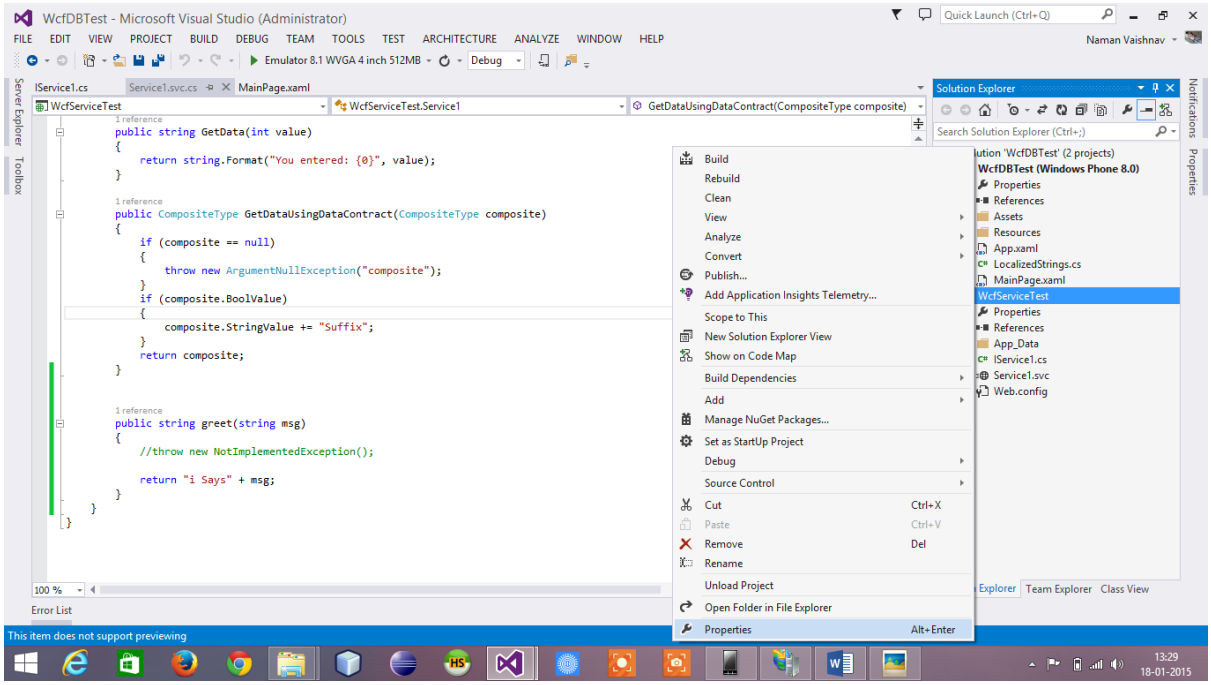




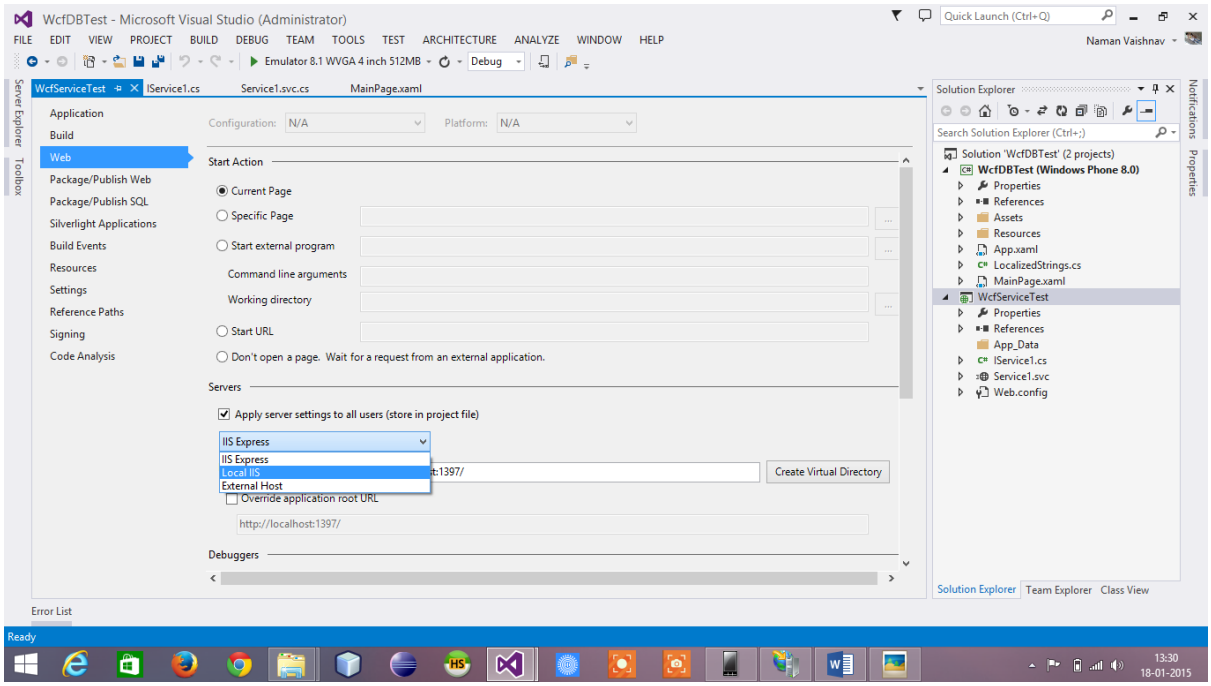
Ri8 click on the Service And Build it once its not Mandatory Bt its good to Having some precotions



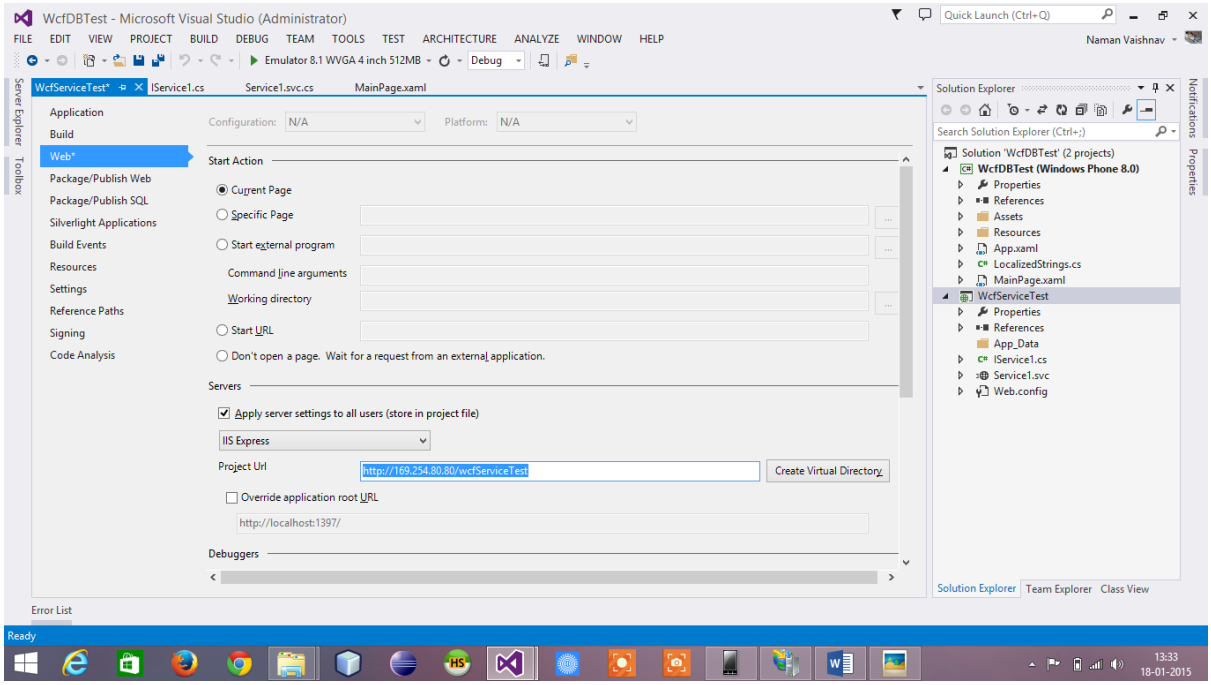
Now Go to the wcfServicetest's properties



Go to the Web >choose Local IIS Express



In the project url write ur machines ip and Web servicename see below fig.

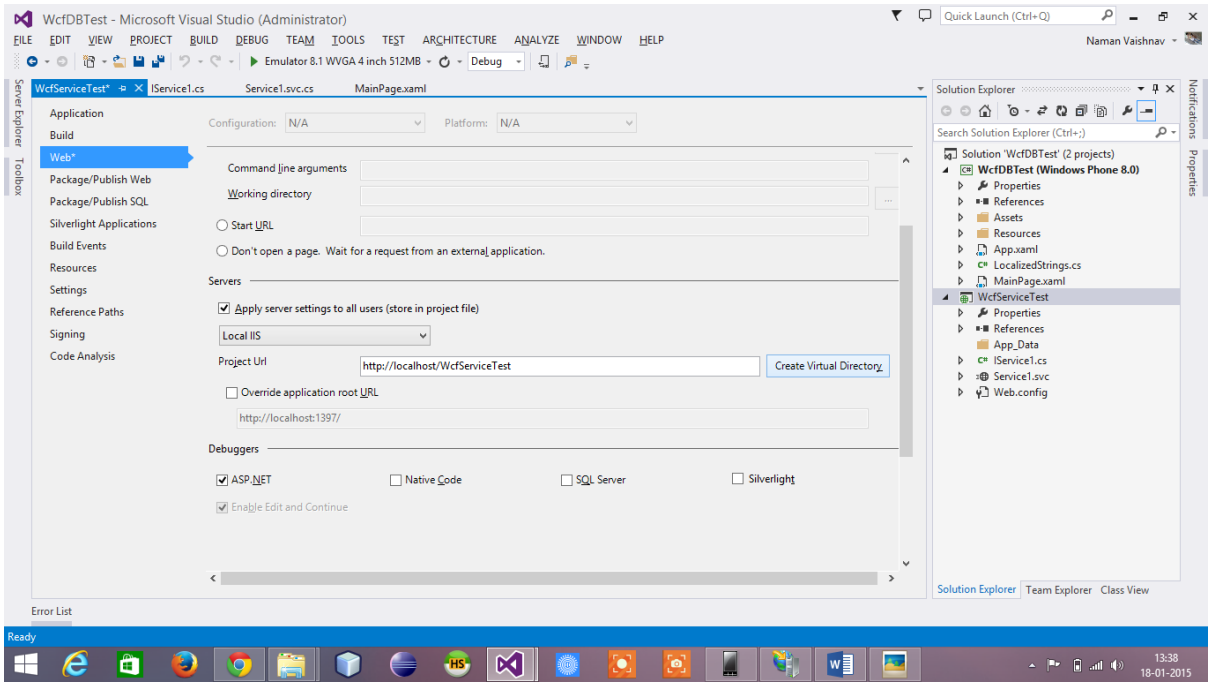


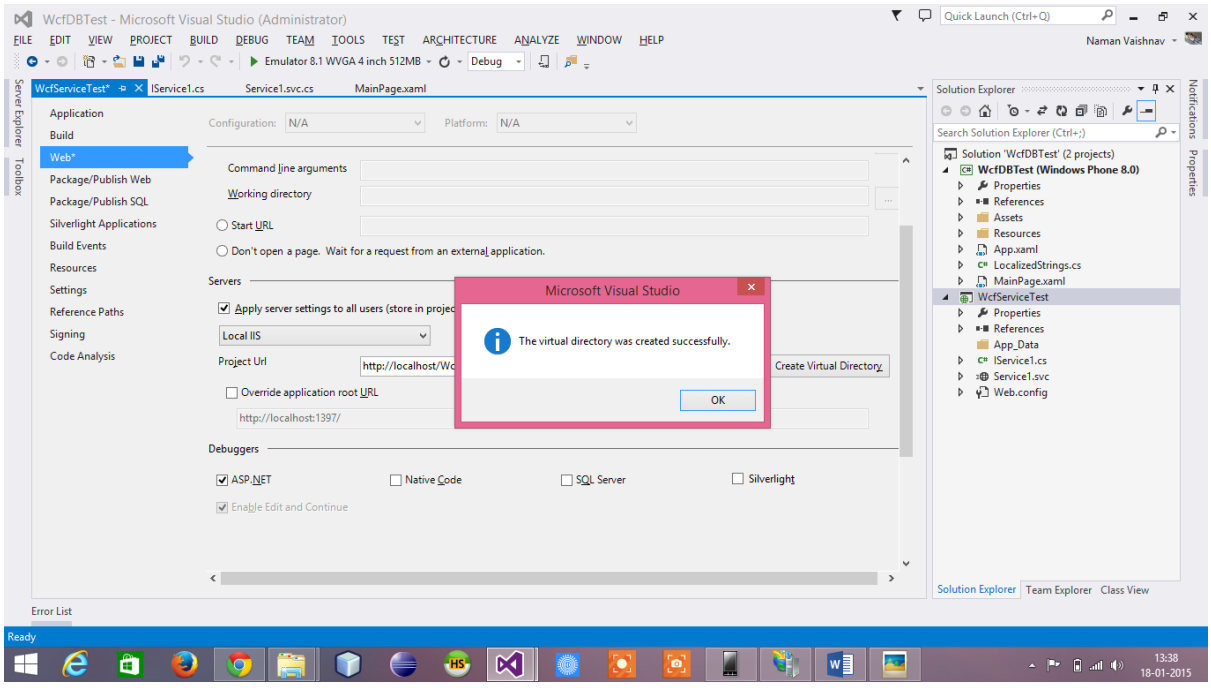
This thing gives u an error...

Then do this operation return

Web> select Local IIS >project URL: http://localhost/WcfServiceTest ←← ← AA Automatically j avi jashe URL

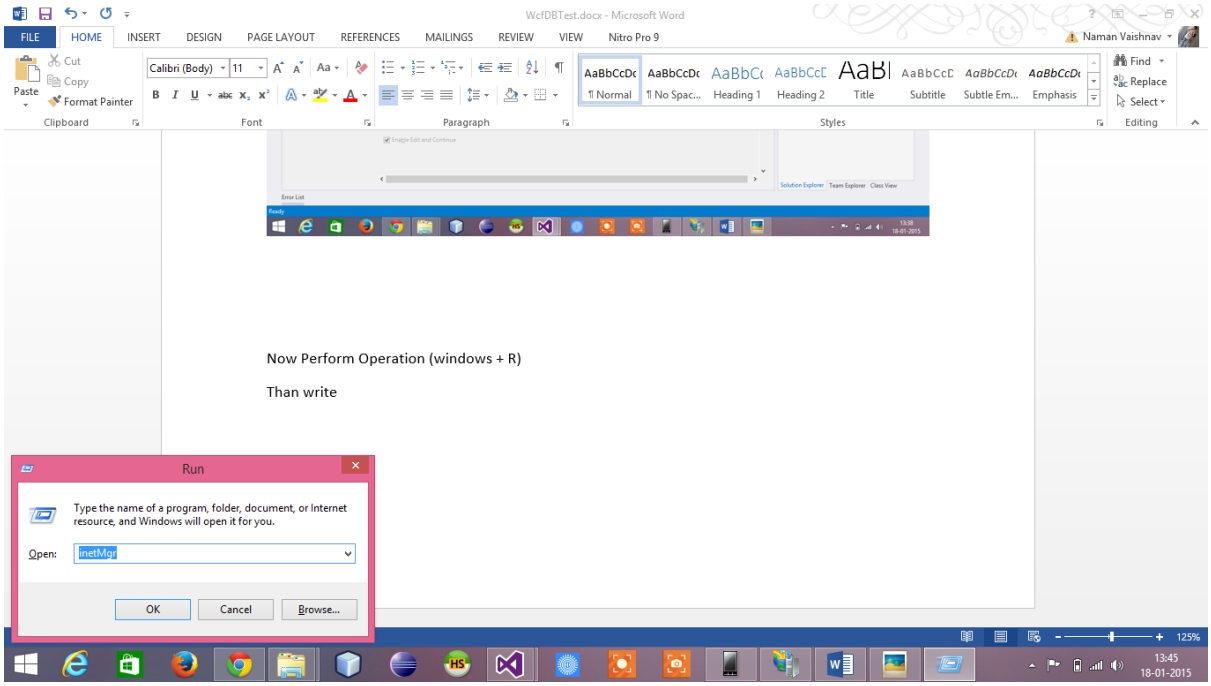
Than create virtual Directory



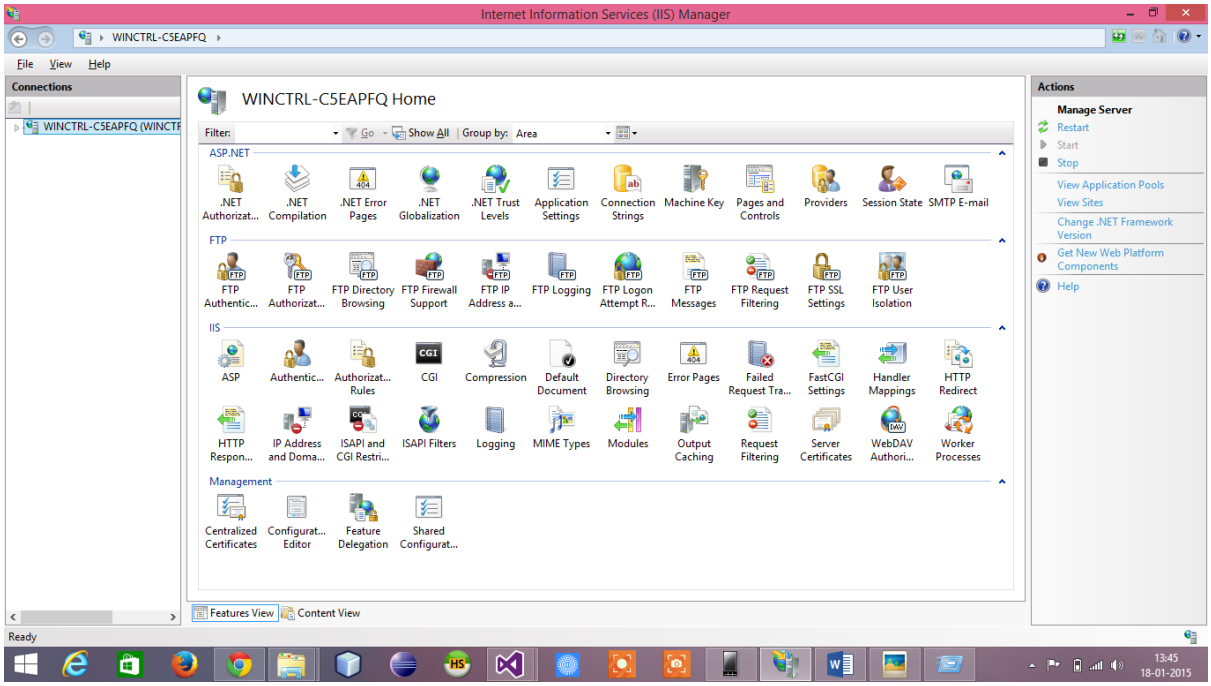


Now Perform Operation (windows + R)

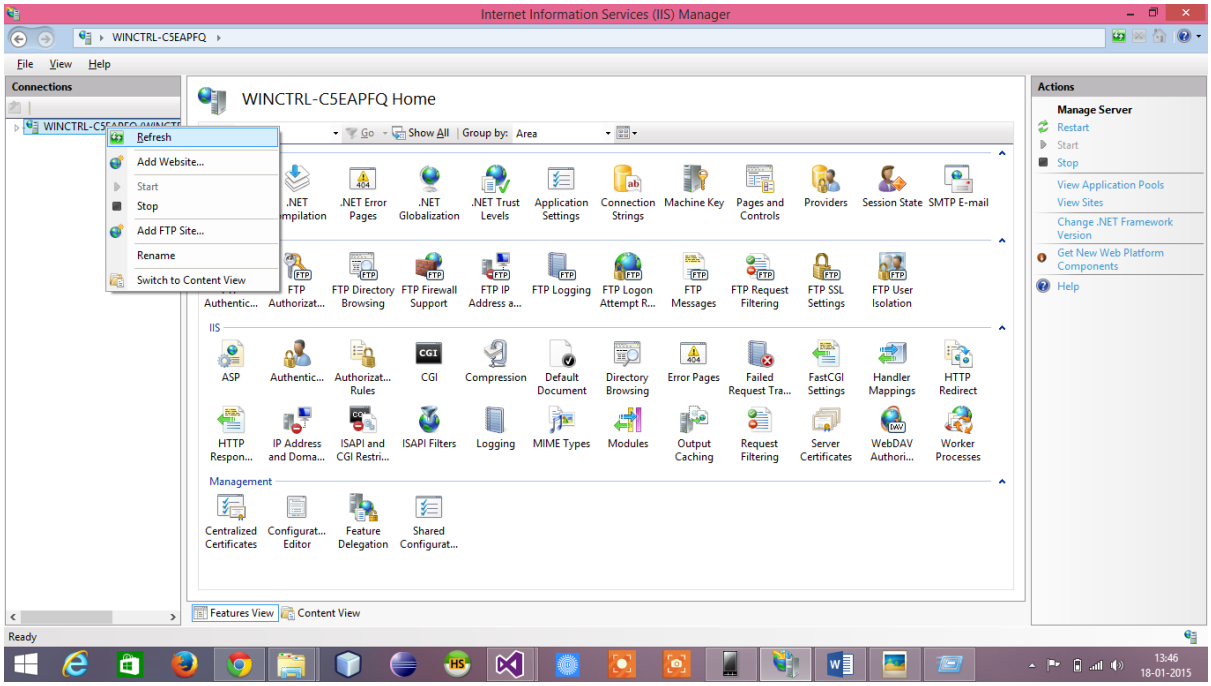
Than write **inetMgr**



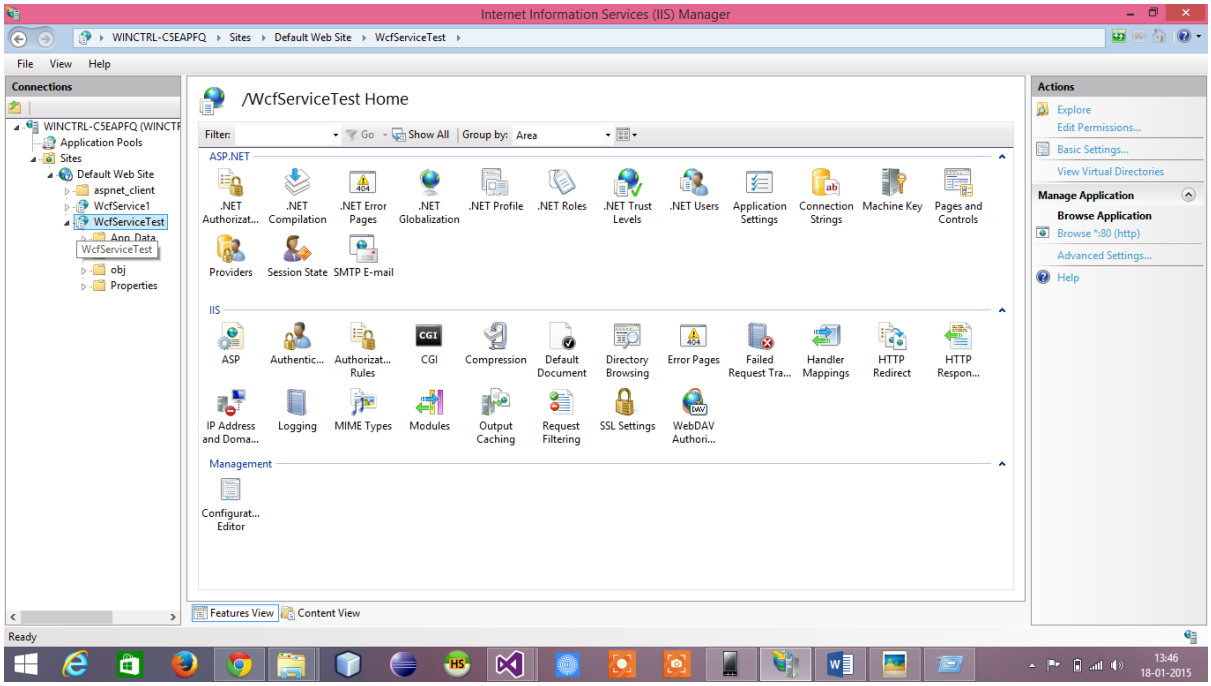
U will see this window



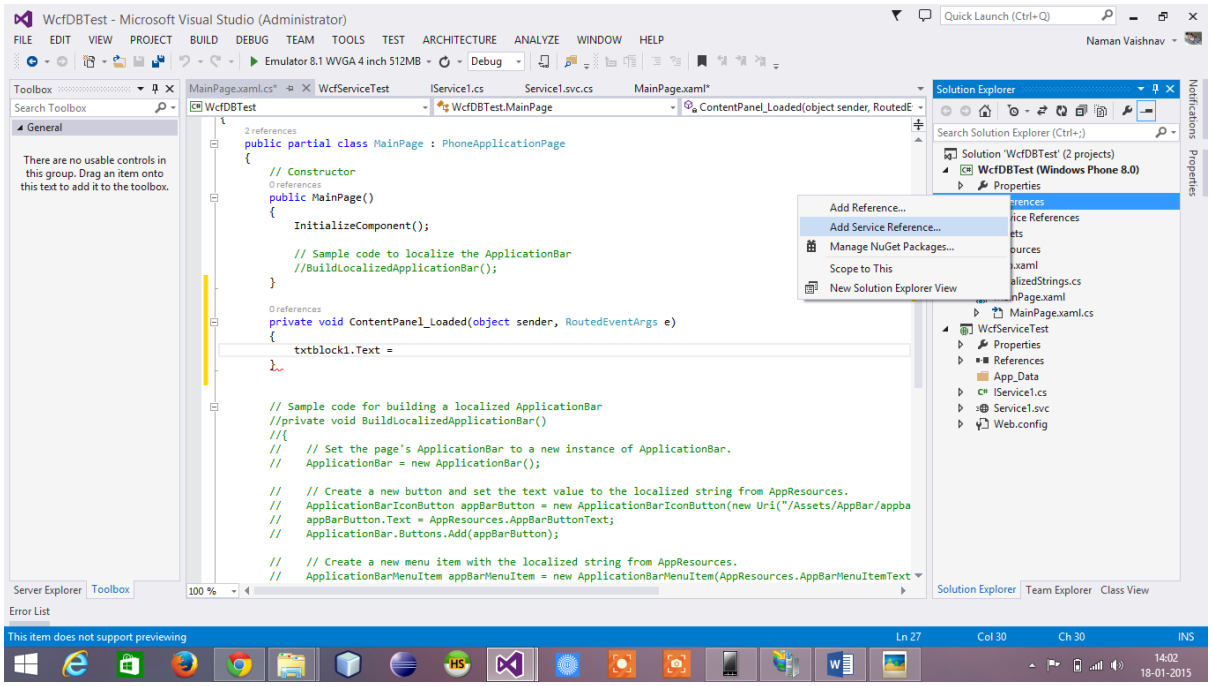
Refresh once



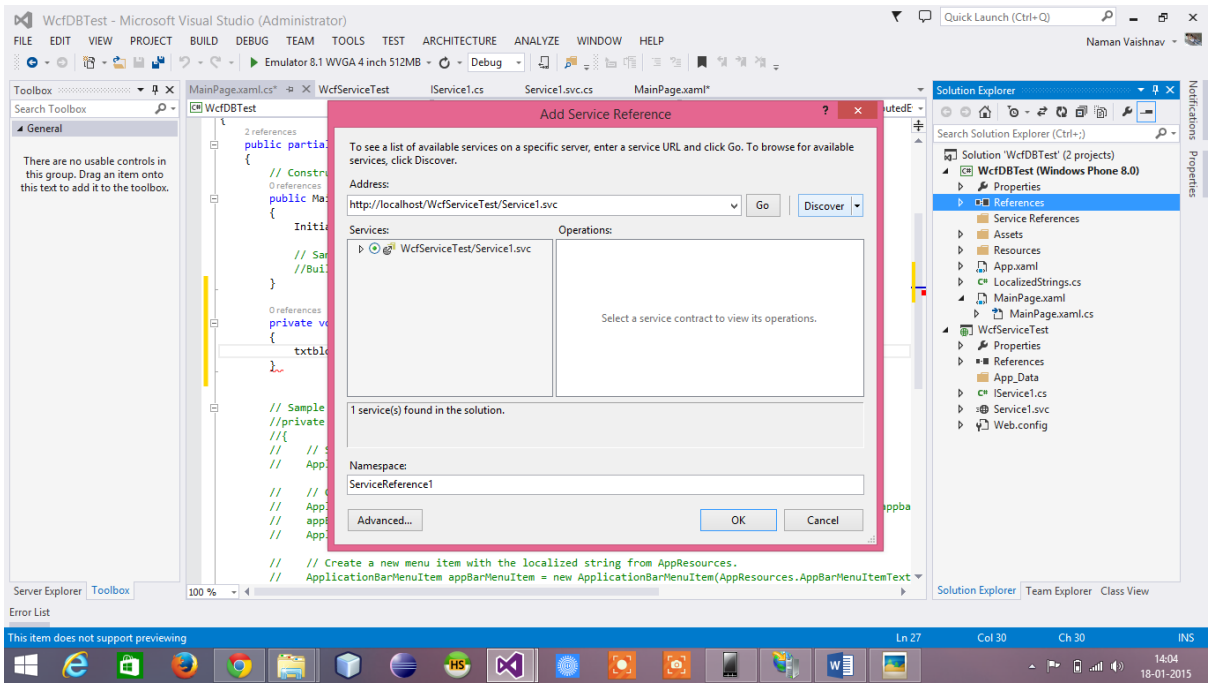
Now you will see your service →here it is **WcfServiceTest**



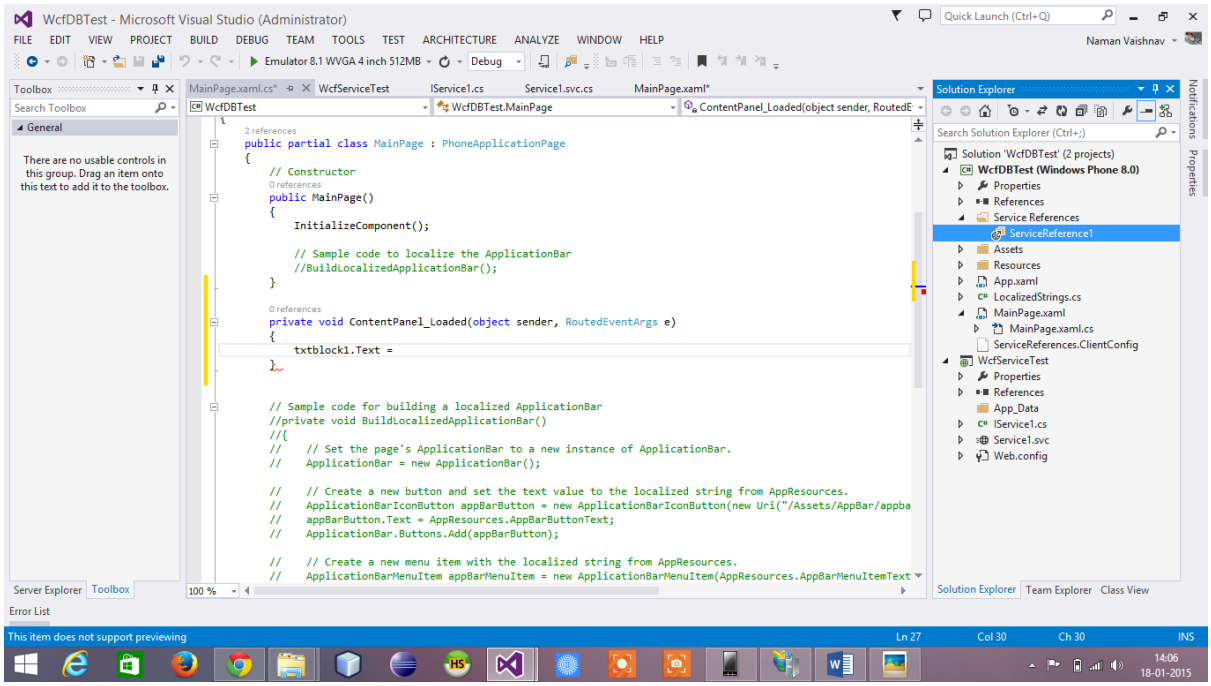
Now ri8 click on **Refrence** in **WcfDBTest Wndows Project** and add Service Refrence



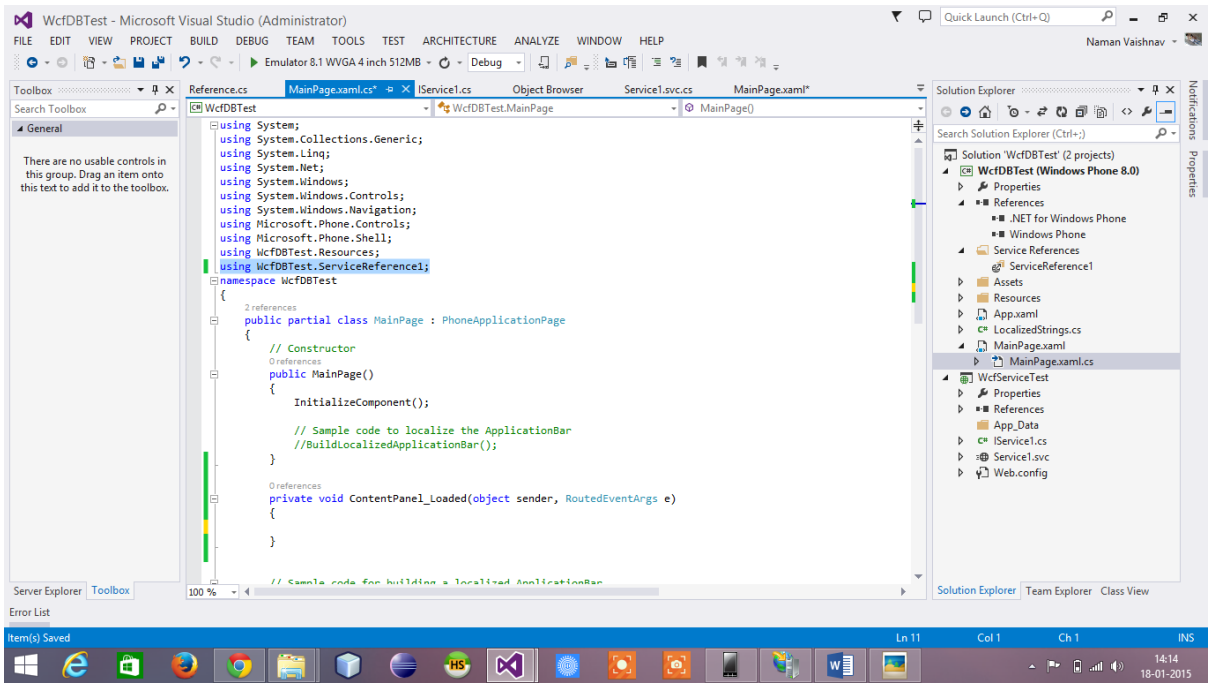
Click on the Discover and u will see the Service That We had Created see the Namespace its ServiceReference1



Click on okk u'll see the Service Reference in the project solution

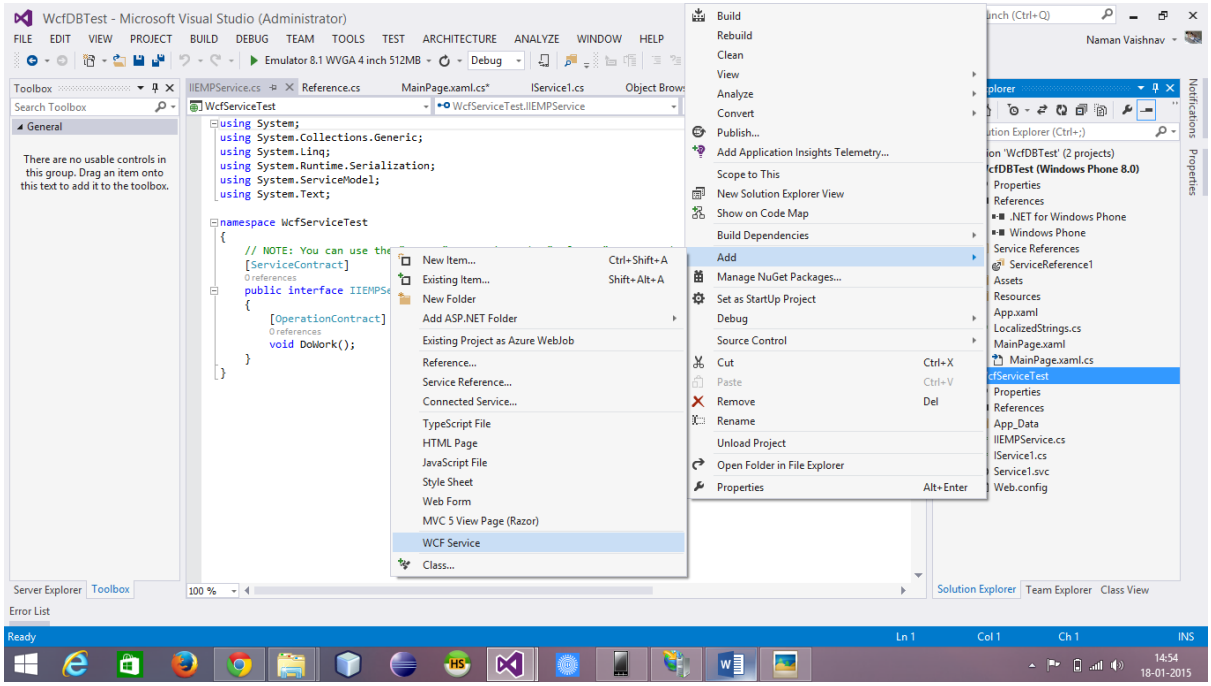


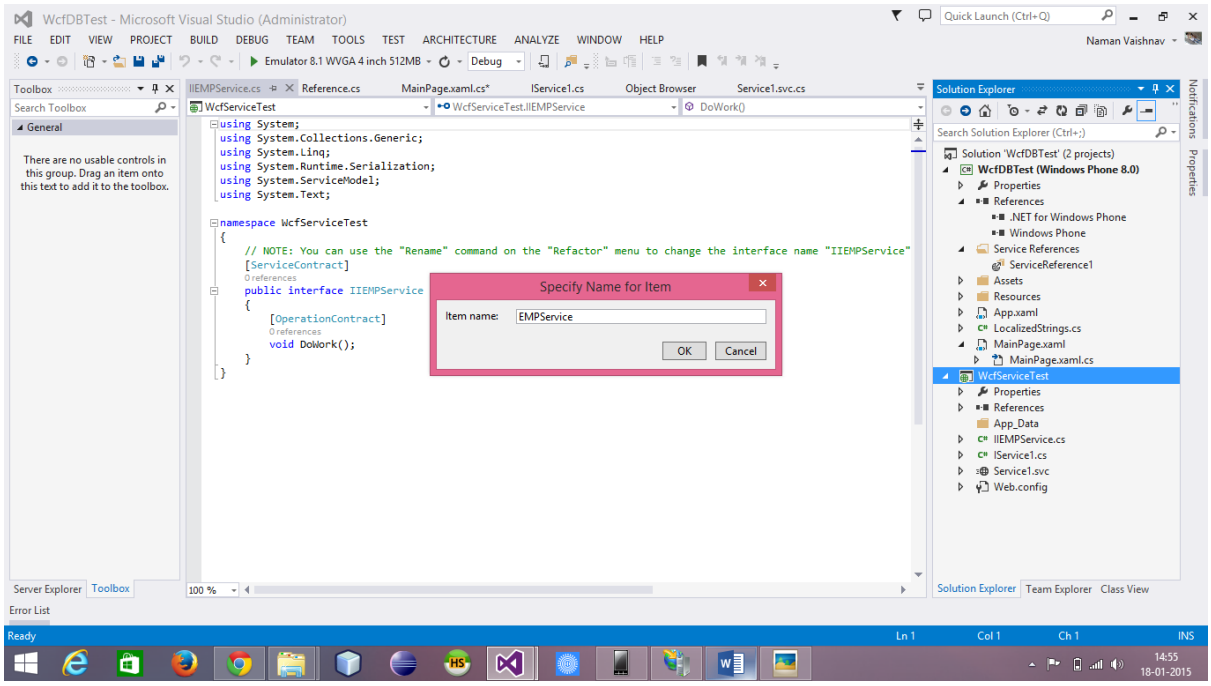
Add the Namespace in the page that u want to use Service



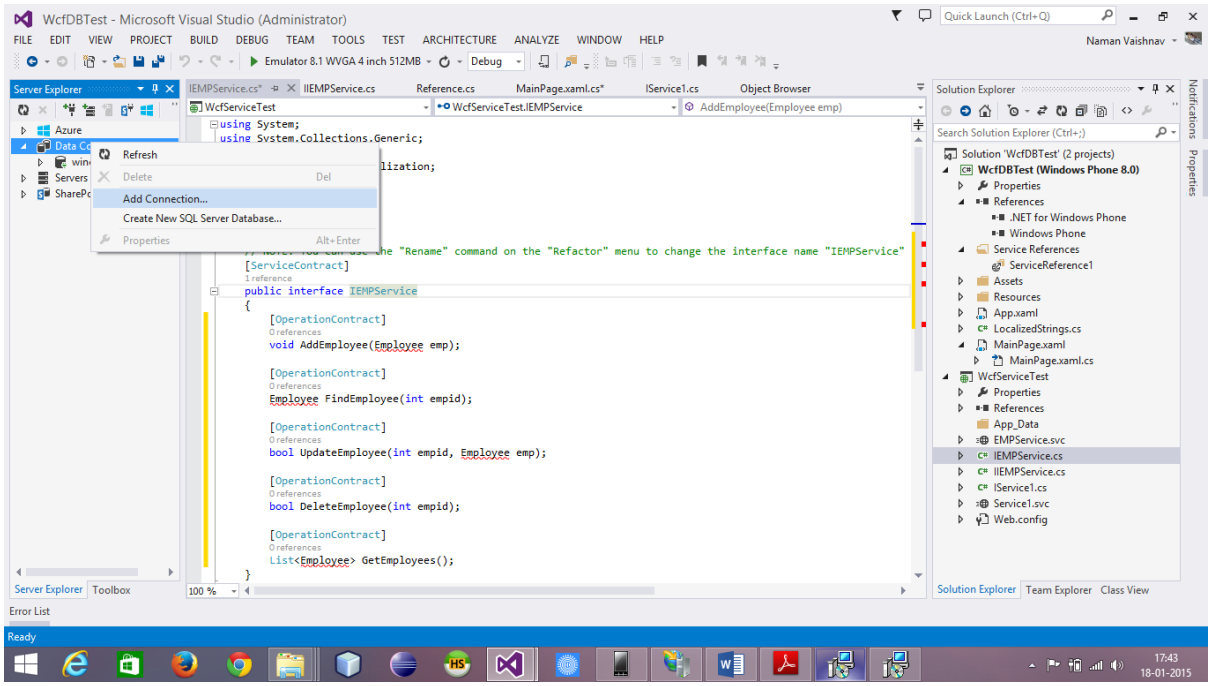
This is how you ll use Wcf Service lets take one Example of it

Create One Service

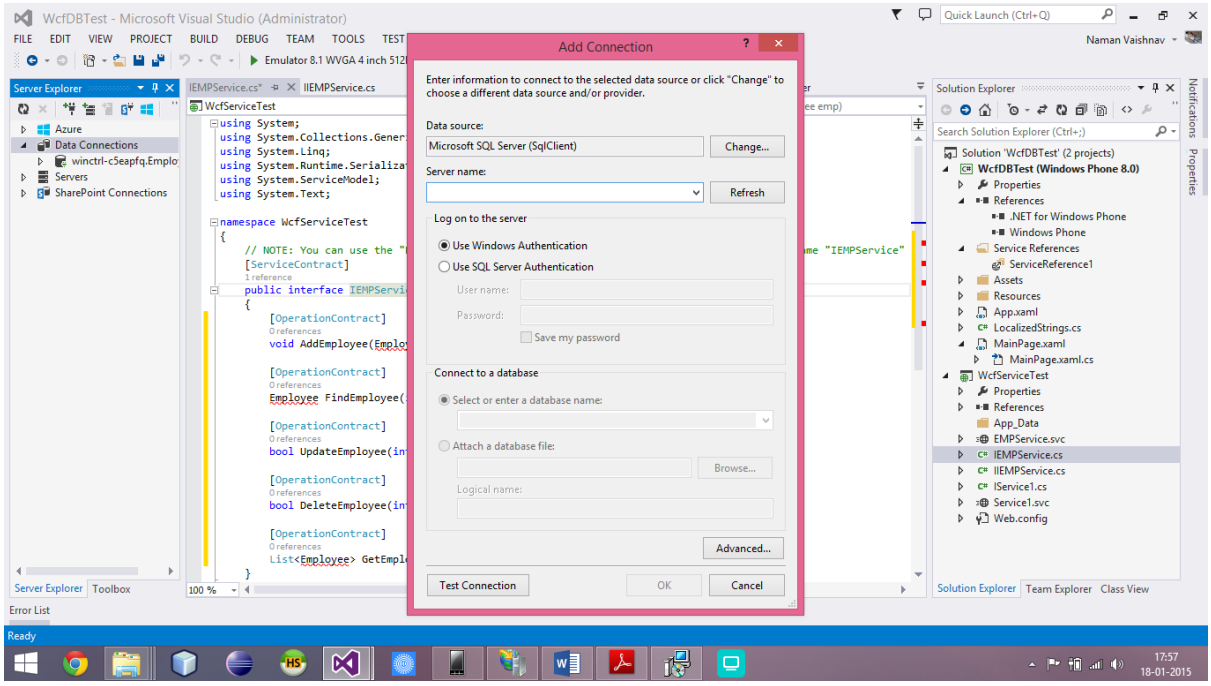




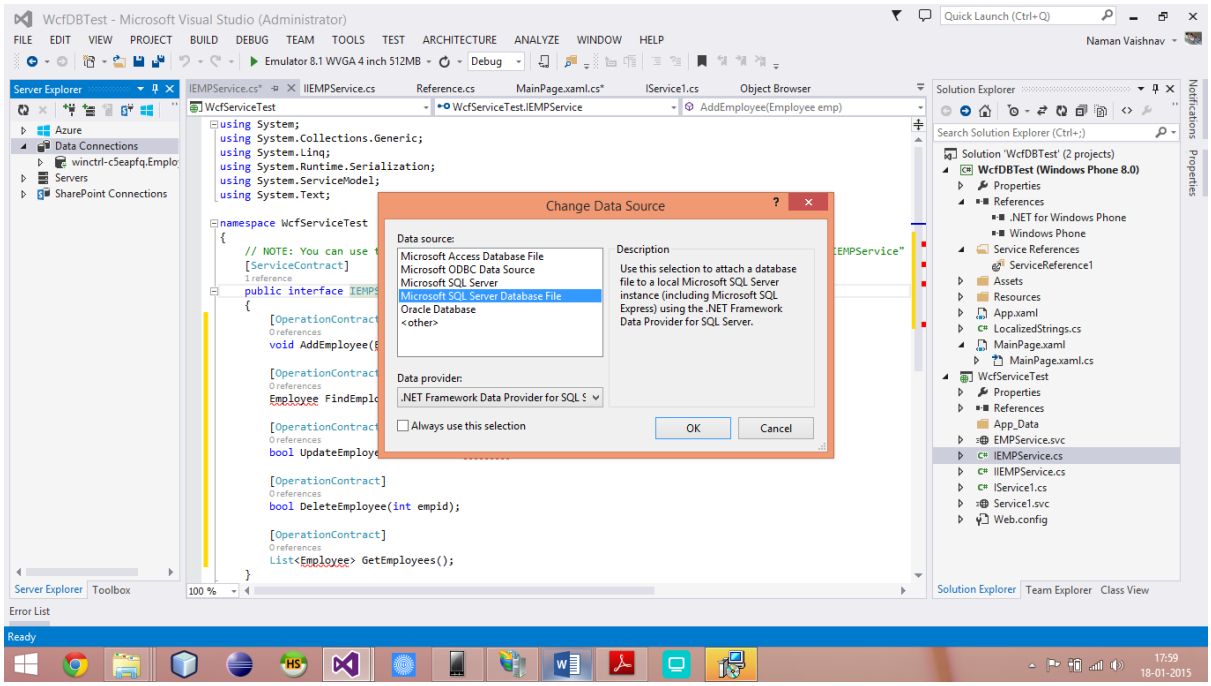
We need to create Employee Database to add delete and view the info about emps..

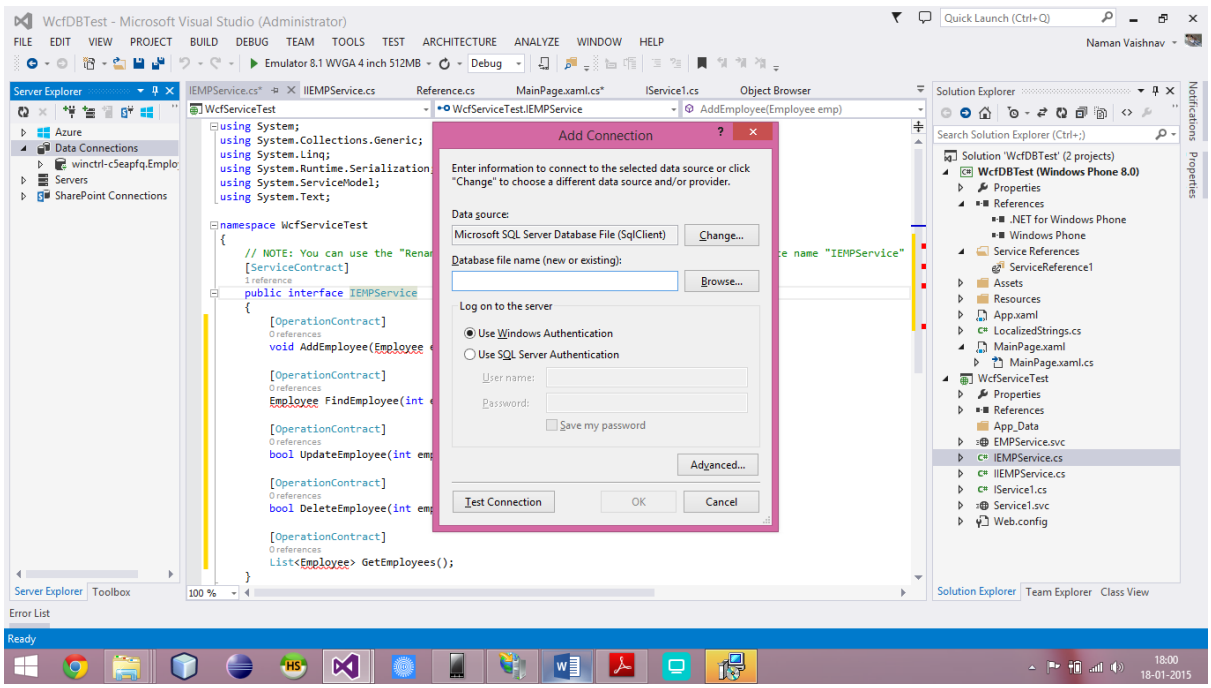


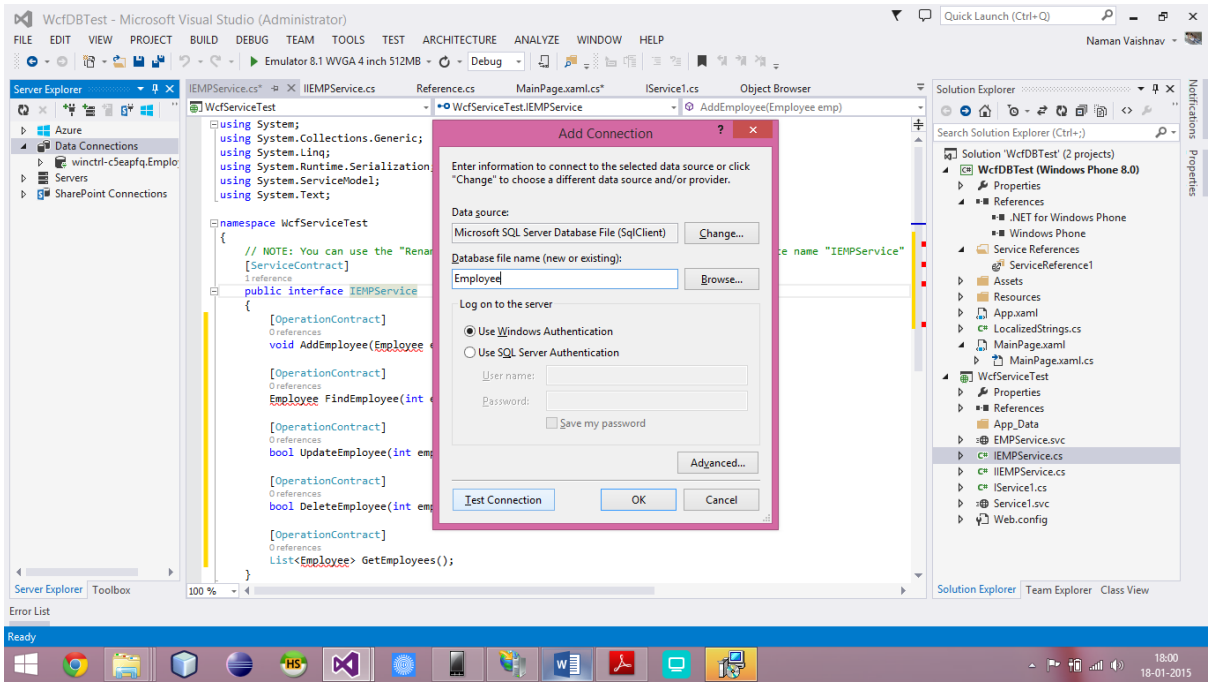
now click on the datasource > change

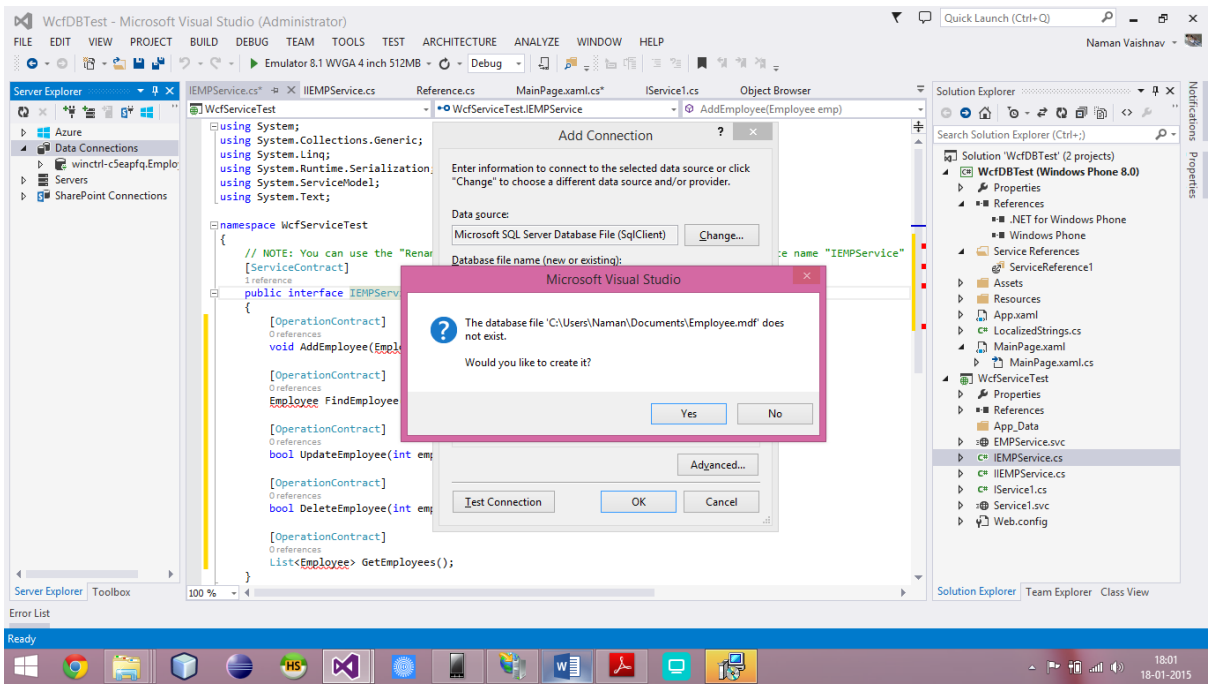


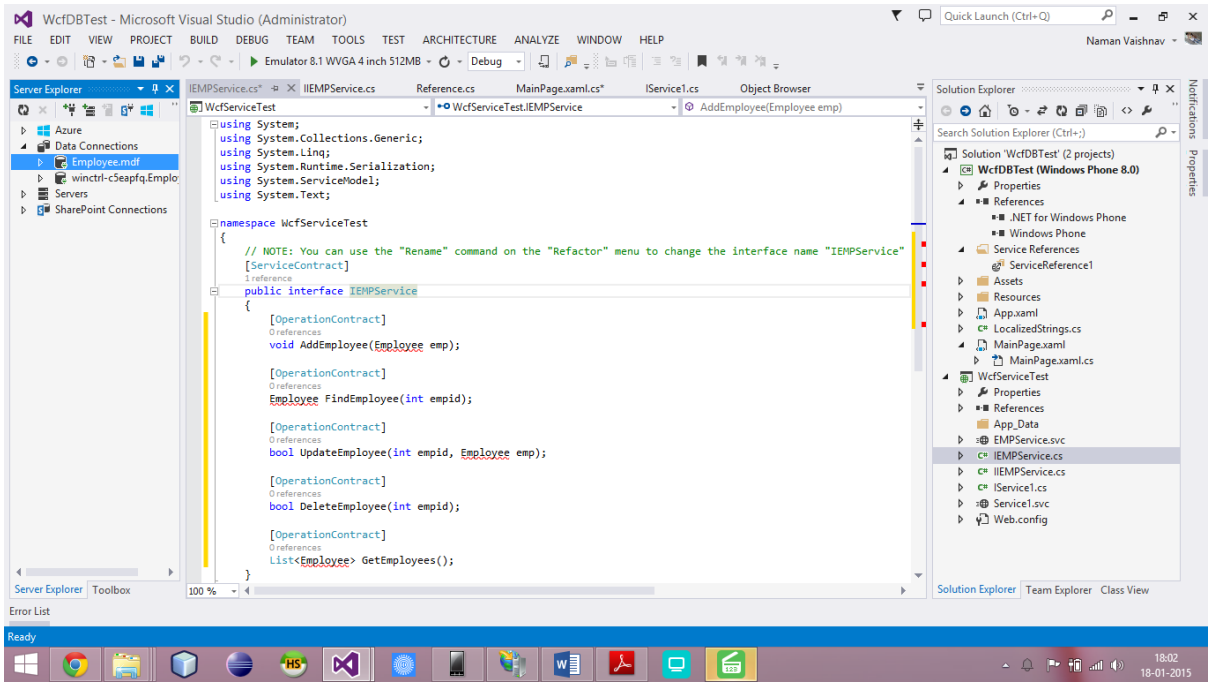
Select Microsoft Sql Server Database File → OK


WcfDBTest - Microsoft Visual Studio (Administrator)
Change Data Source
Data source:
Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Oracle Database
<other>
Description
Use this selection to attach a database file to a local Microsoft SQL Server instance (including Microsoft SQL Express) using the .NET Framework Data Provider for SQL Server.
Data provider:
.NET Framework Data Provider for SQL S
Always use this selection
OK
Cancel

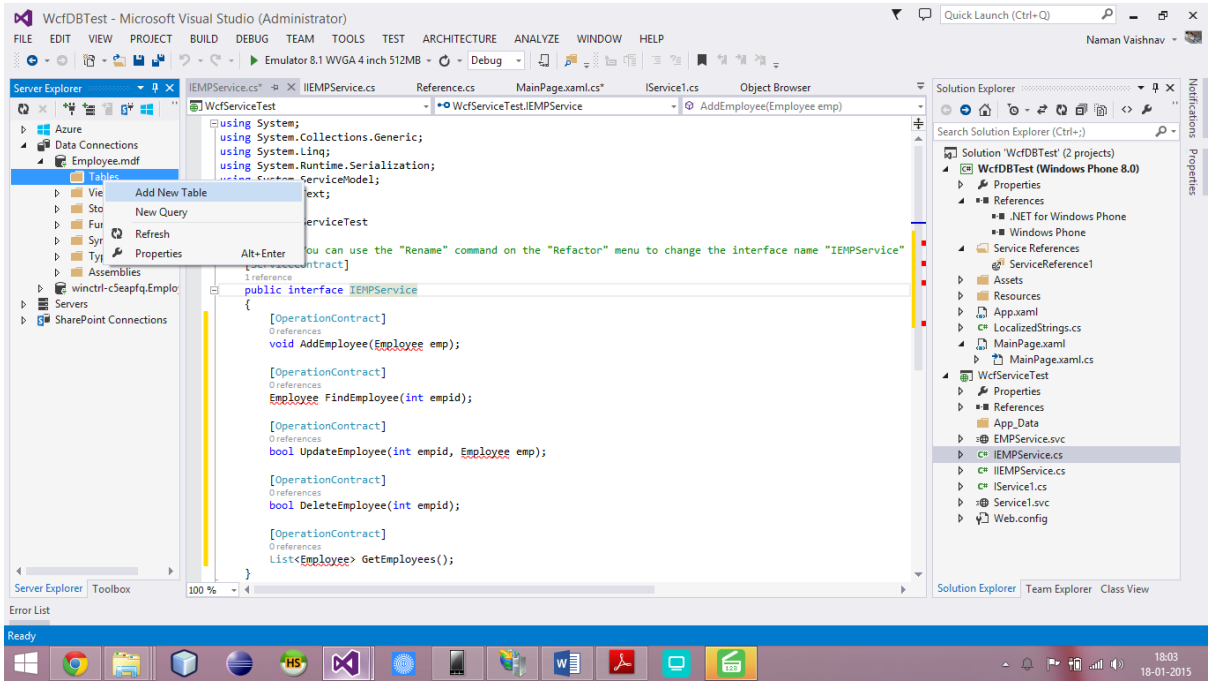

WcfDBTest - Microsoft Visual Studio (Administrator)
Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server Database File (SqlClient)
Change...
Database file name (new or existing):
Browse...
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Save my password
Advanced...
Test Connection
OK
Cancel

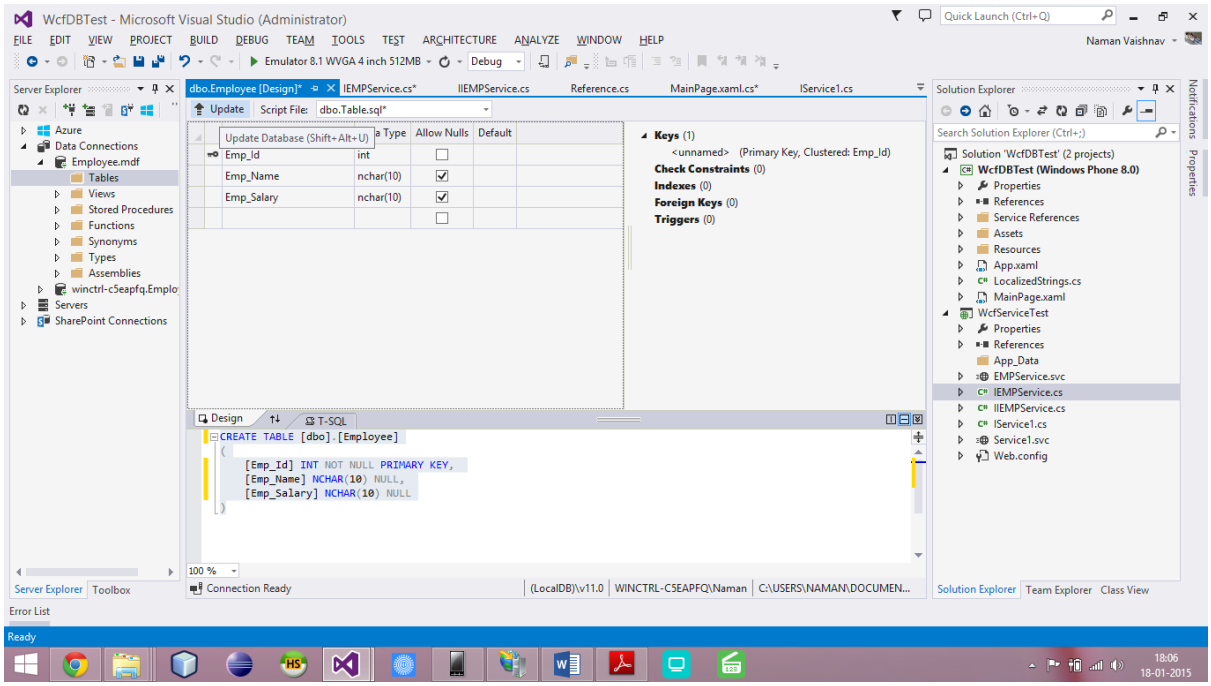

WcfDBTest - Microsoft Visual Studio (Administrator)
Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server Database File (SqlClient)
Change...
Database file name (new or existing):
Employee
Browse...
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Save my password
Advanced...
Test Connection
OK
Cancel

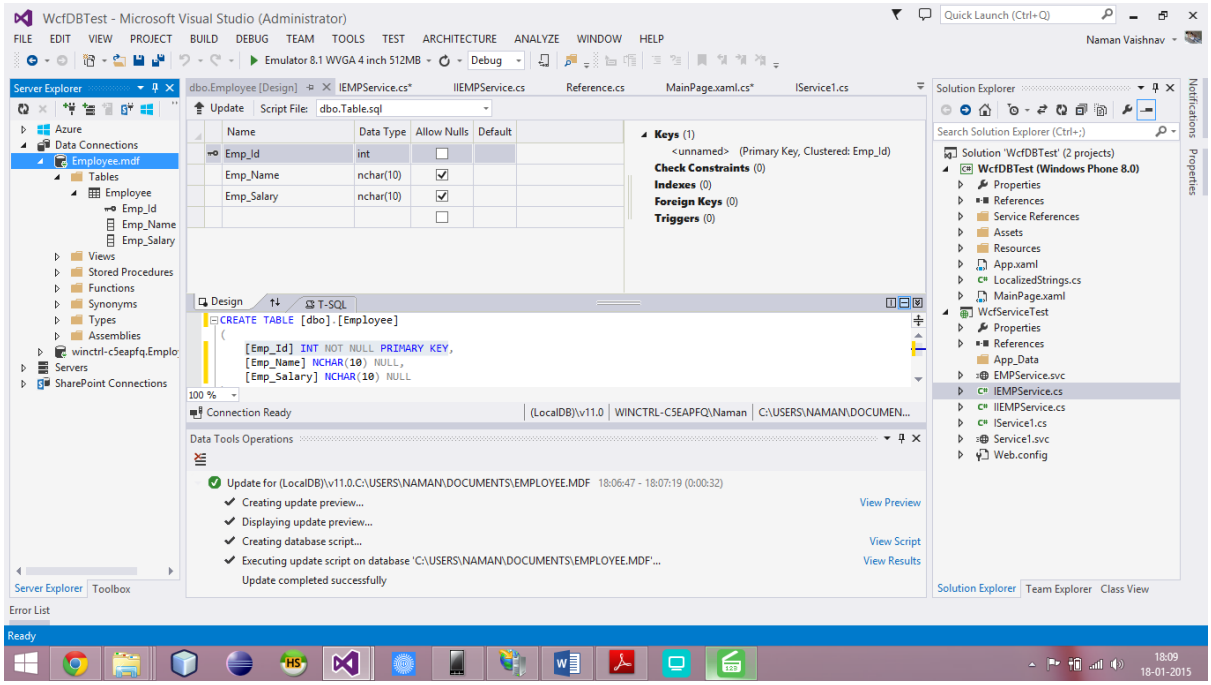

WcfDBTest - Microsoft Visual Studio (Administrator)
Add Connection
Microsoft Visual Studio
The database file 'C:\Users\Naman\Documents\Employee.mdf' does not exist.
Would you like to create it?
Yes
No
Test Connection
OK
Cancel



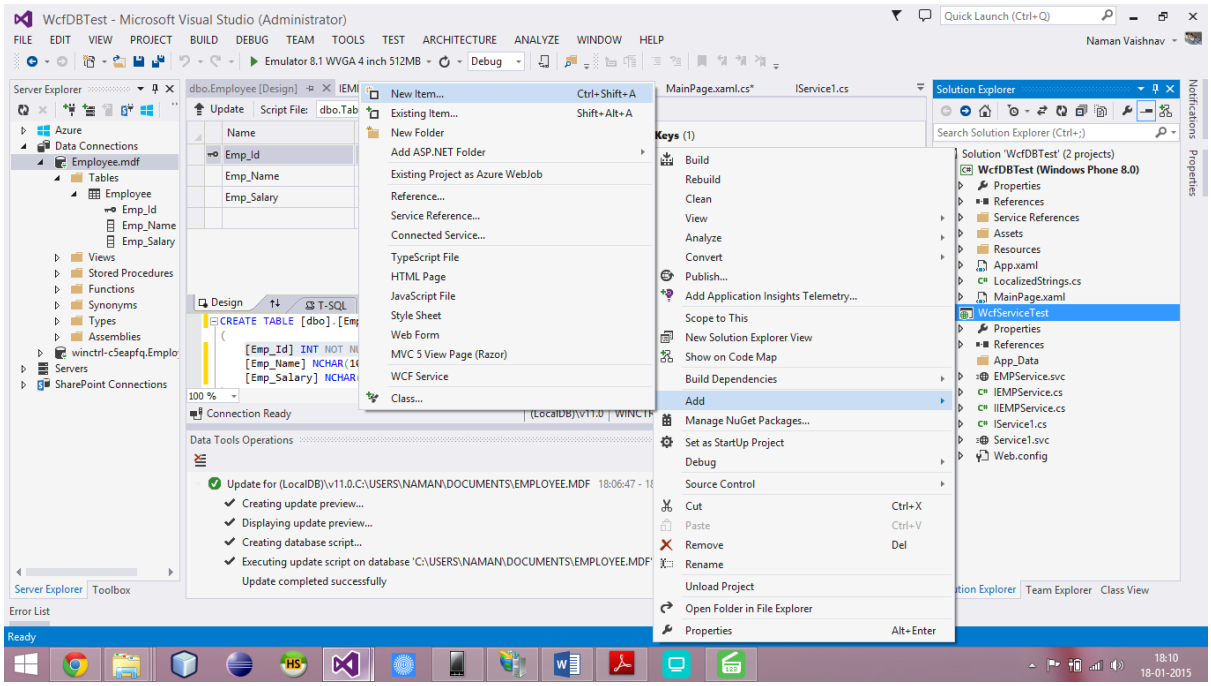
Now Create New Table named Employee

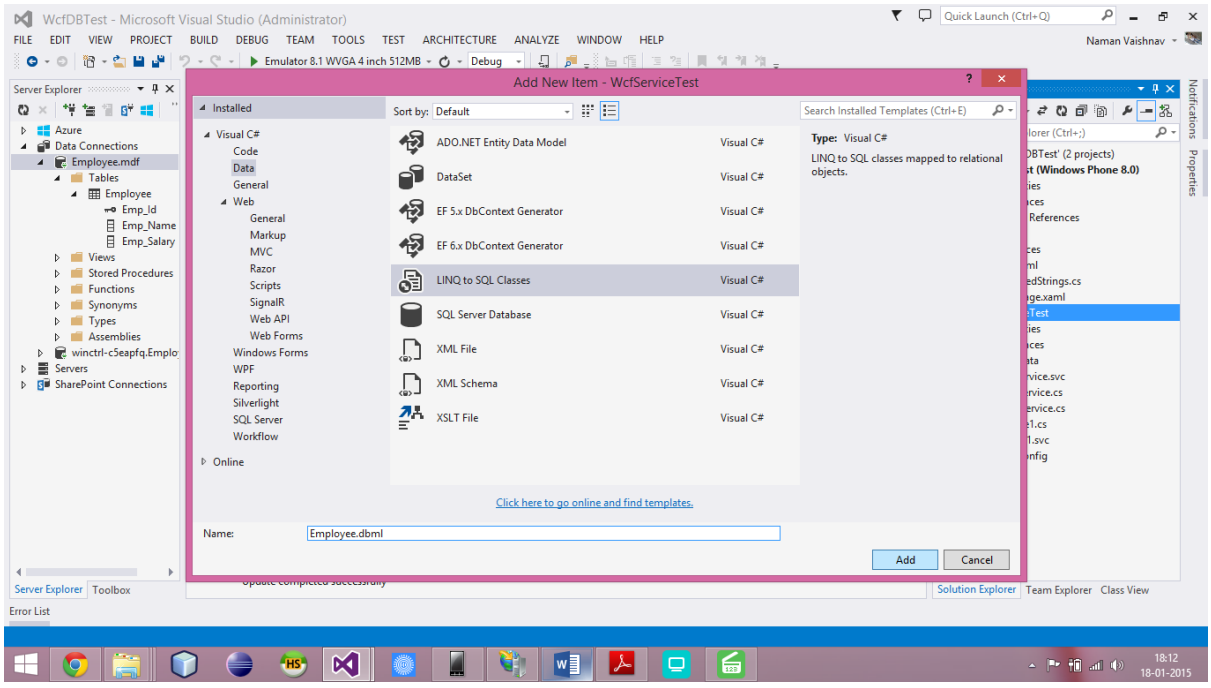




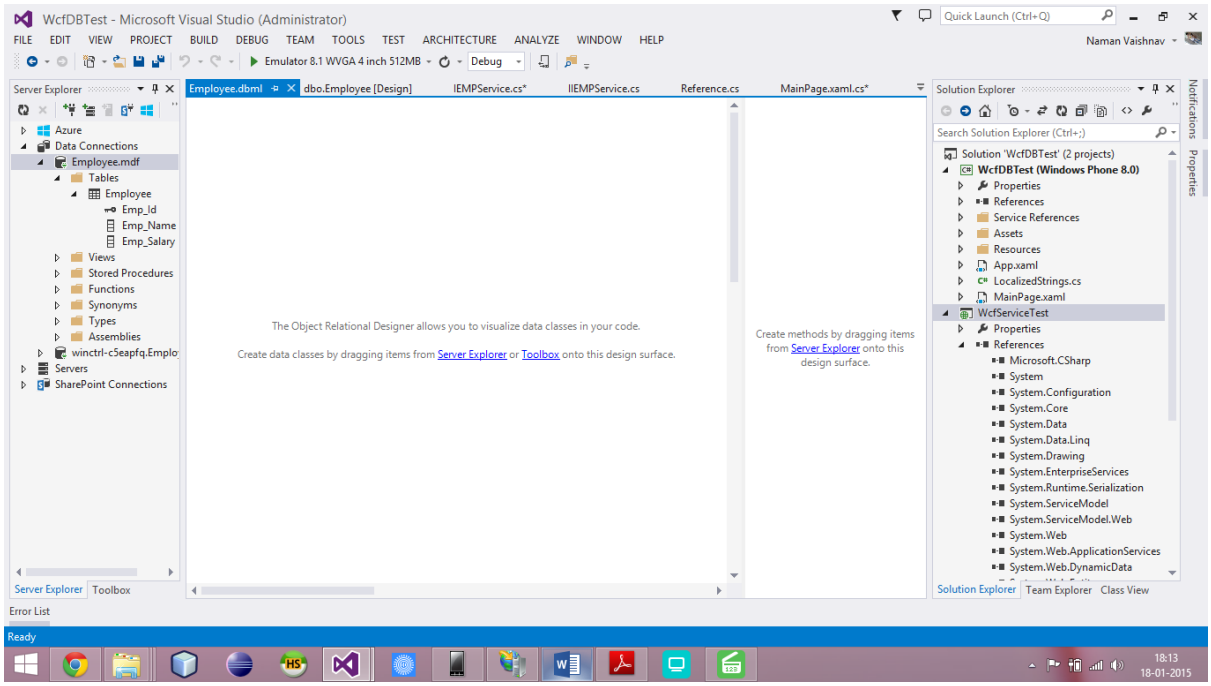


Now Database is ready now In the Service Project Add New Iteme > Data >LinQ to Sql Class

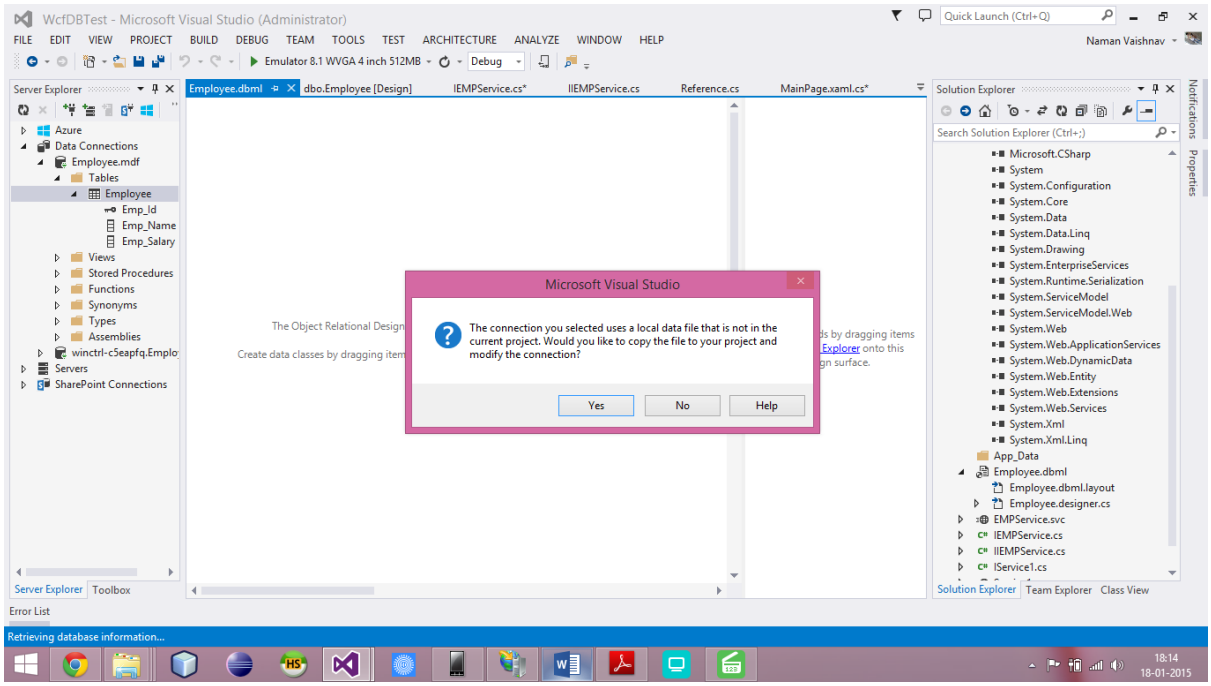




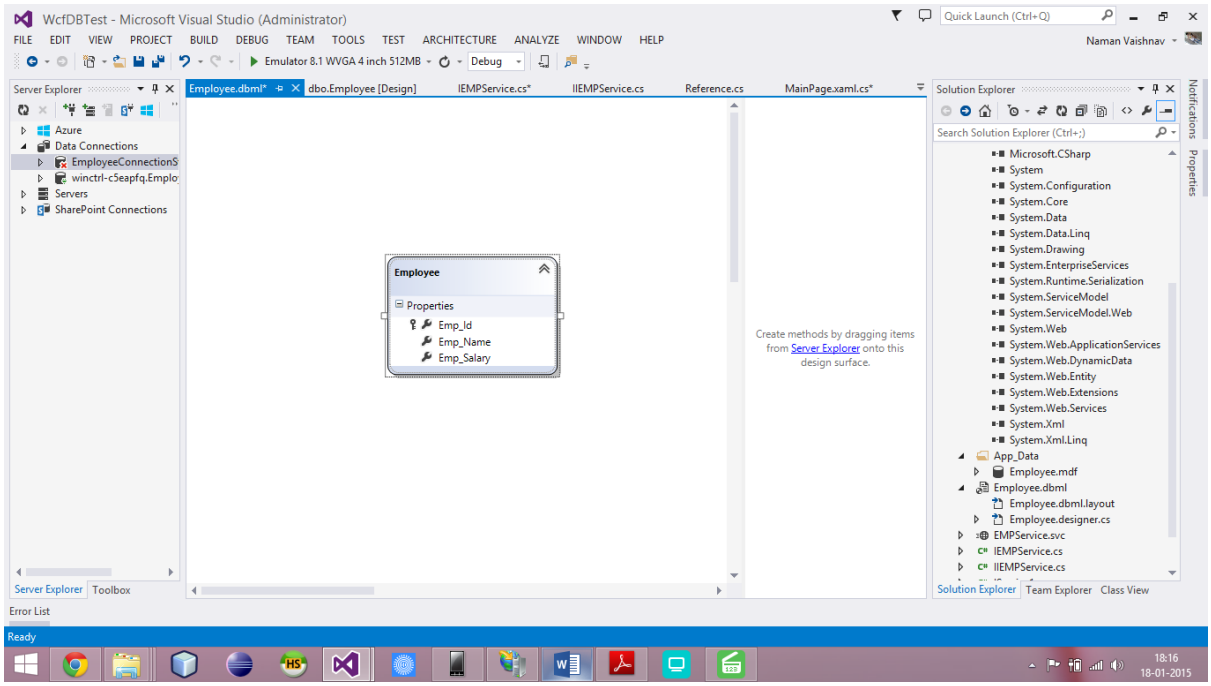
Now u will see the Employee.dbml blank page it tells you to create database..

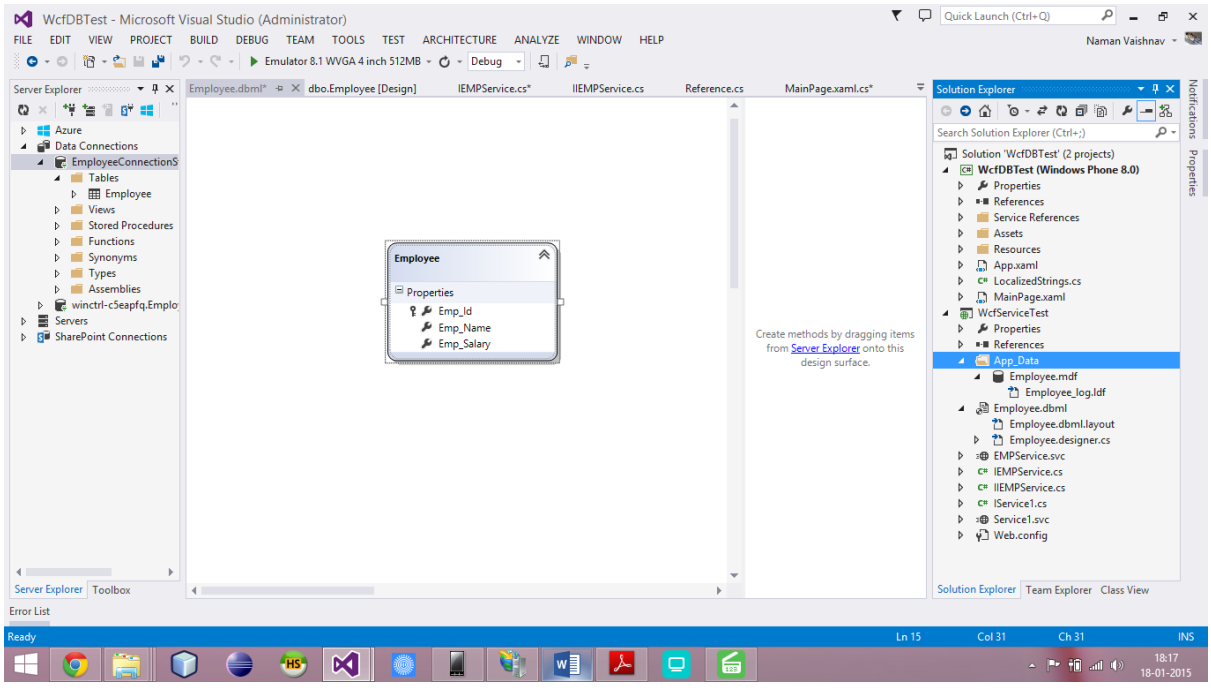


We already created database now dage the Employee table to Employee.dbml page you will get a notification



Say Yes you can see the database in our service into App_Data folder (Solution Explorer)



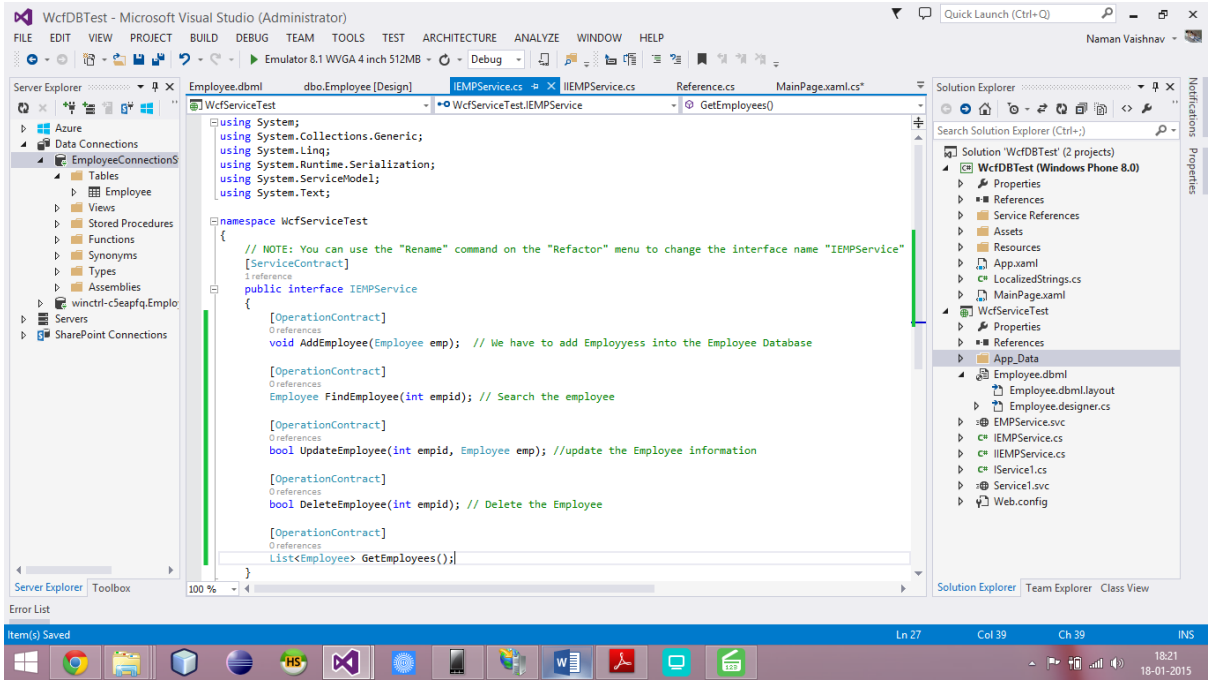


Now do the code into the **IEMPService.cs** File Execute the Contract and Declaire one Generic Type

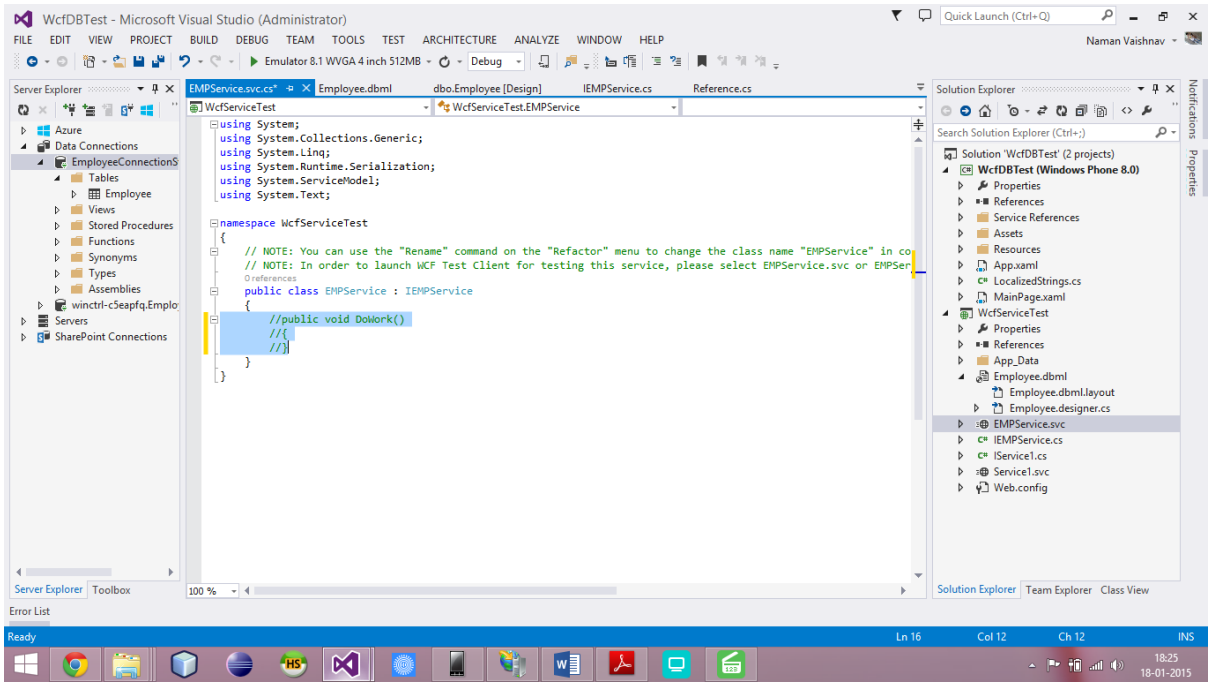
CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WcfServiceTest
{
```

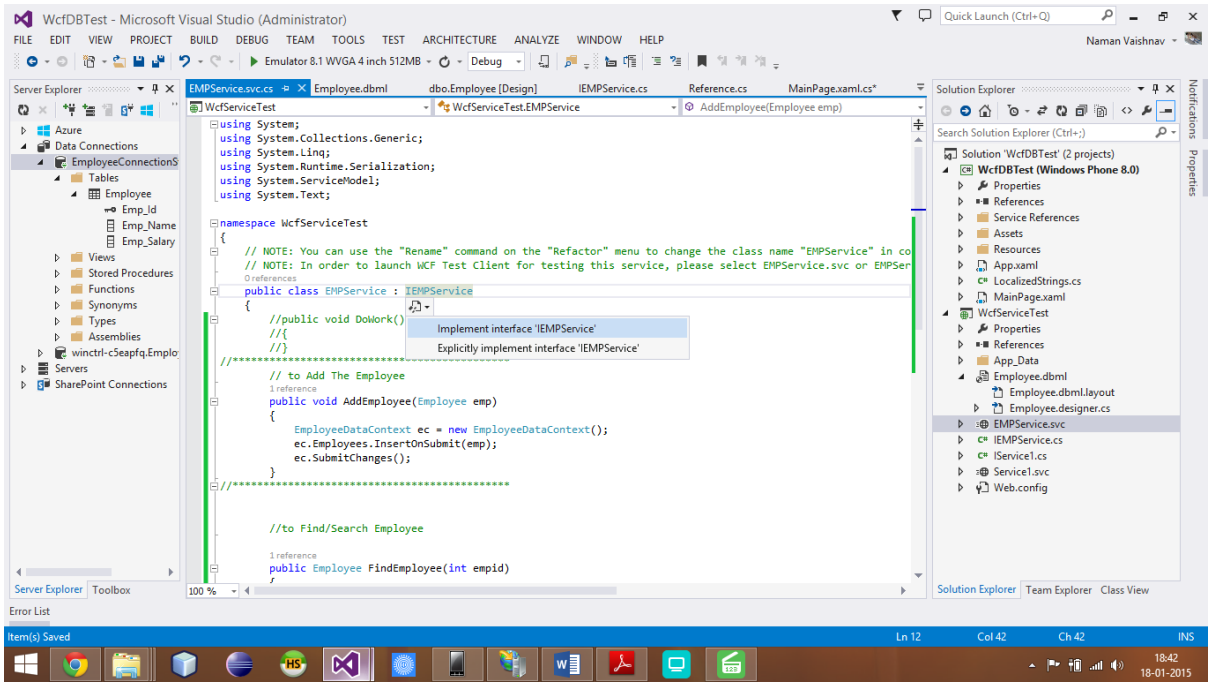
```
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the
interface name "IEMPService" in both code and config file together.
    [ServiceContract]
    public interface IEMPService
    {
        [OperationContract]
        void AddEmployee(Employee emp);  // We have to add Employyess into the
Employee Database

        [OperationContract]
        Employee FindEmployee(int empid); // Search the employee

        [OperationContract]
        bool UpdateEmployee(int empid, Employee emp); //update the Employee
information

        [OperationContract]
        bool DeleteEmployee(int empid); // Delete the Employee

        [OperationContract]
        List<Employee> GetEmployees();
    }
}
```



Now GO to the EMPService.svc.cs File And Do the Code



`public class EMPService : IEMPService` ←Click On the IEMPService And U Get The all Methods From the IEMPService File



See the code in **EmpService.svc.cs**

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

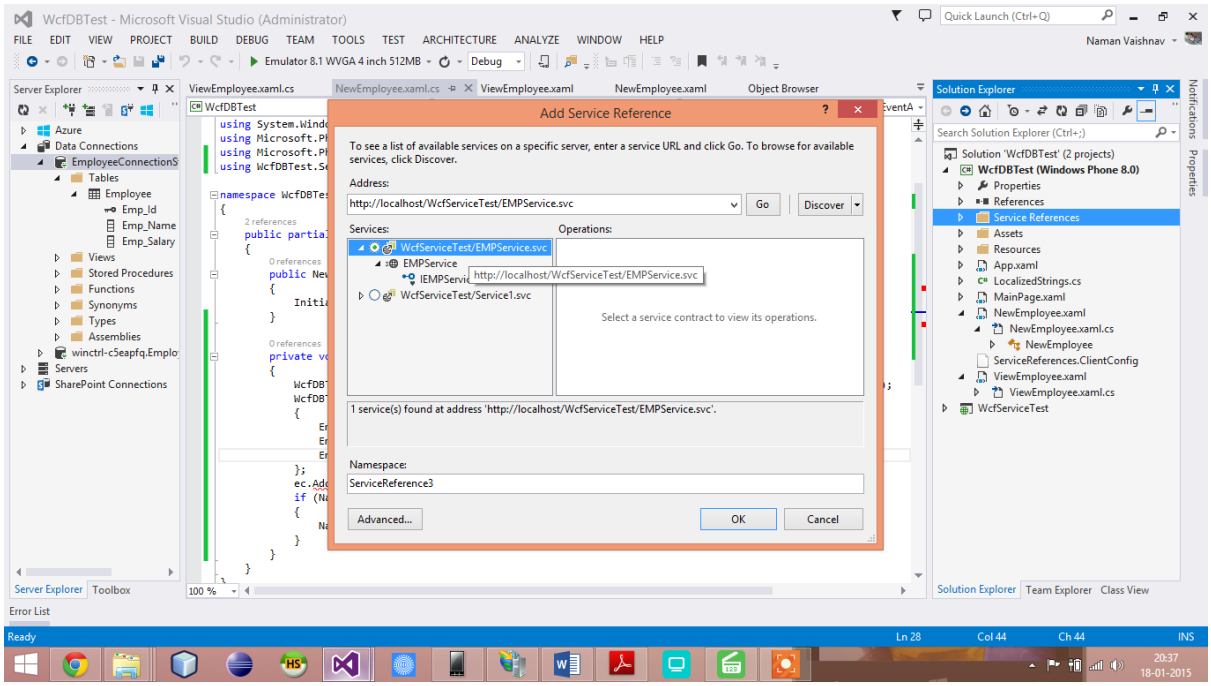
```
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WcfServiceTest
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the
class name "EMPService" in code, svc and config file together.
    // NOTE: In order to launch WCF Test Client for testing this service, please
select EMPService.svc or EMPService.svc.cs at the Solution Explorer and start
debugging.
    public class EMPService : IEMPService
    {
        //public void DoWork()
        //{
        //}
//*****************************************
        // to Add The Employee
        public void AddEmployee(Employee emp)
        {
            EmployeeDataContext ec = new EmployeeDataContext();
            ec.Employees.InsertOnSubmit(emp);
            ec.SubmitChanges();
        }
//*****************************************


        //to Find/Search Employee

        public Employee FindEmployee(int empid)
        {
            EmployeeDataContext ec = new EmployeeDataContext();
            IEnumerable<Employee> lemp = from e in ec.Employees where e.Emp_Id ==
empid select e; // LinQ for Locate Emp..
            Employee femp = lemp.FirstOrDefault();//  Find Emp? yes.. than return it
/ no than give Default Value
            return femp;
        }

//*****************************************

        // to update Emp

        public bool UpdateEmployee(int empid, Employee emp) // Bool <-- Means its
Value Is True Or False
        {
            //throw new NotImplementedException();

            EmployeeDataContext ec = new EmployeeDataContext();
            IEnumerable<Employee> lemp = from e in ec.Employees where e.Emp_Id ==
empid select e;
            Employee femp = lemp.FirstOrDefault();
            if (femp != null)
            {
                femp.Emp_Id = emp.Emp_Id;
                femp.Emp_Name = emp.Emp_Name;
                femp.Emp_Salary = emp.Emp_Salary;
                ec.SubmitChanges();
                return true; // Old Value Are Replace With The New One
            }
            else
```

```
            {
                return false;
            }
        }

        public bool DeleteEmployee(int empid)
        {
            //throw new NotImplementedException();
            EmployeeDataContext ec = new EmployeeDataContext();
            IEnumerable<Employee> lemp = from e in ec.Employees where e.Emp_Id ==
empid select e;
            Employee femp = lemp.FirstOrDefault();
            if (femp != null)
            {
                ec.Employees.DeleteOnSubmit(femp);
                ec.SubmitChanges();
                return true;
            }
            else
            {
                return false;
            }
        }

        public List<Employee> GetEmployees()
        {
            //throw new NotImplementedException();
            EmployeeDataContext ec = new EmployeeDataContext();
            return ec.Employees.ToList();     // jetla b changes thaya hashe ae badha
list ma add kari devama avshe ..
        }
    }
}
```

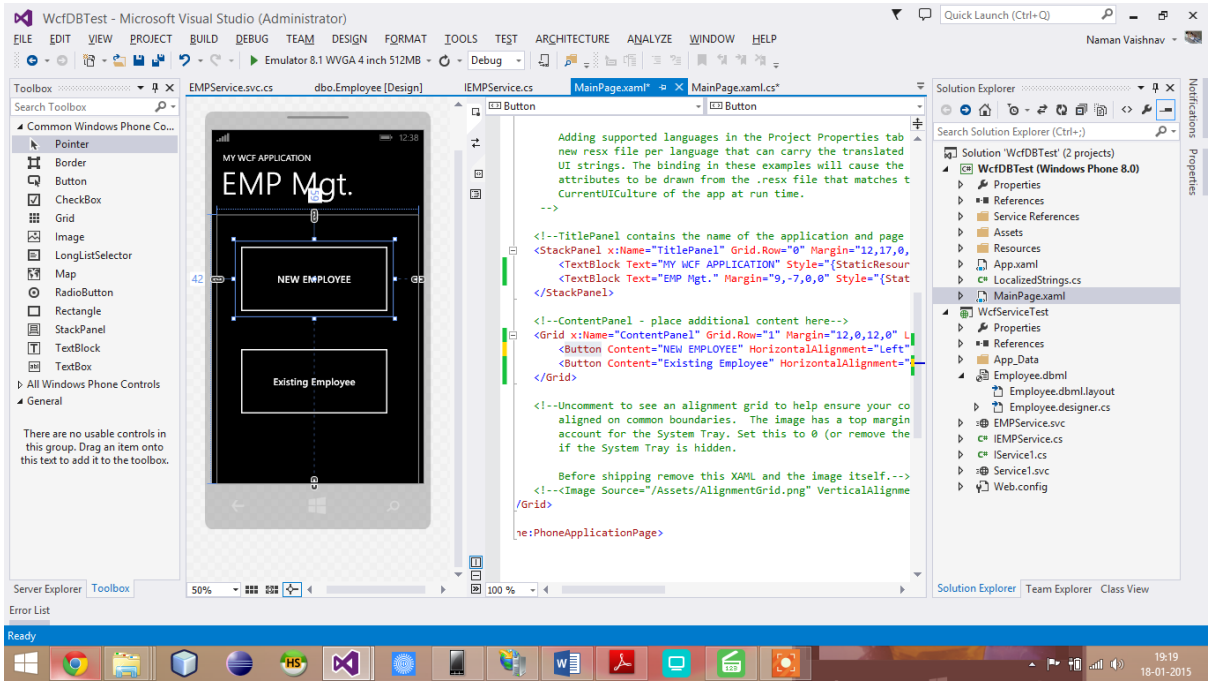
Now work on Service is Completed on other hand We have to see Phone App First Create Layout In the main Page Like this..

Make Sure that U have added Service Reference in this Side

This Step Watch out ←



Now Create Layout of Main Page



MAinPage.Xaml

```
<phone:PhoneApplicationPage
    x:Class="WcfDBTest.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
```

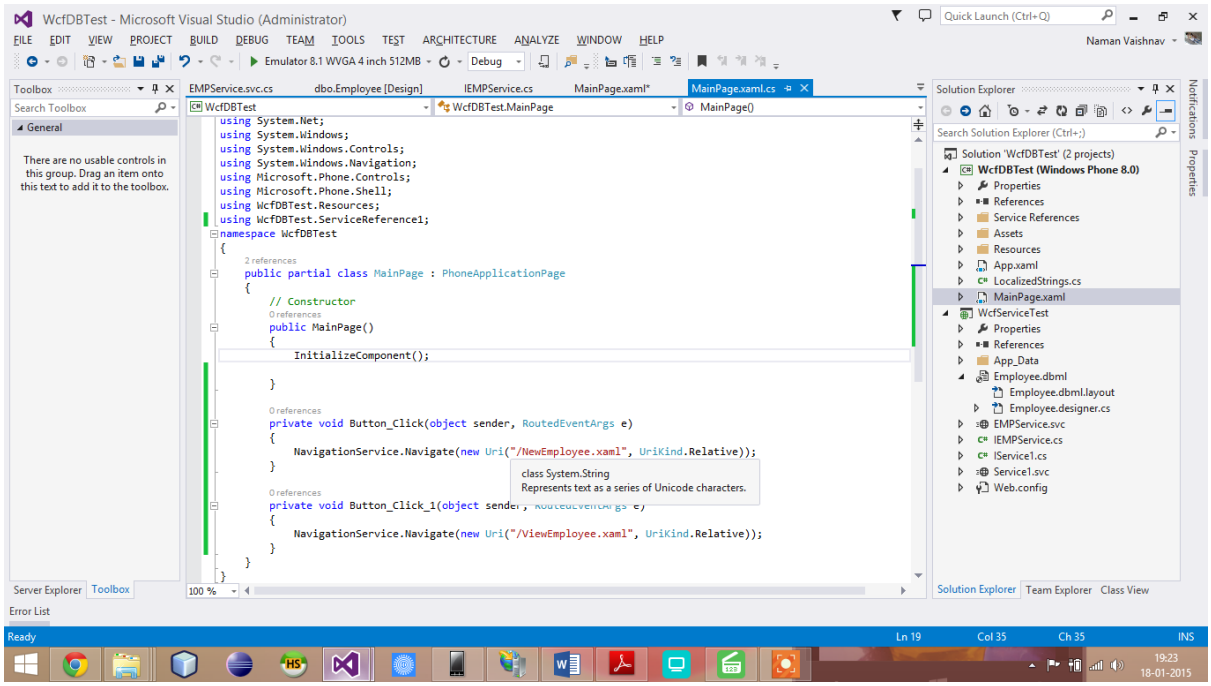
```
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!-- LOCALIZATION NOTE:
            To localize the displayed strings copy their values to appropriately named
            keys in the app's neutral language resource file (AppResources.resx) then
            replace the hard-coded text value between the attributes' quotation marks
            with the binding clause whose path points to that string name.

            For example:

                Text="{Binding Path=LocalizedResources.ApplicationTitle,
Source={StaticResource LocalizedStrings}}"

            This binding points to the template's string resource named
"ApplicationTitle".

            Adding supported languages in the Project Properties tab will create a
            new resx file per language that can carry the translated values of your
            UI strings. The binding in these examples will cause the value of the
            attributes to be drawn from the .resx file that matches the
            CurrentUICulture of the app at run time.
         -->

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
            <TextBlock Text="MY WCF APPLICATION" Style="{StaticResource
PhoneTextNormalStyle}" Margin="12,0"/>
            <TextBlock Text="EMP Mgt." Margin="9,-7,0,0" Style="{StaticResource
PhoneTextTitle1Style}"/>
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0"
Loaded="ContentPanel_Loaded">
            <Button Content="NEW EMPLOYEE" HorizontalAlignment="Left"
Margin="42,59,0,0" VerticalAlignment="Top" Height="170" Width="356"
Click="Button_Click"/>
            <Button Content="Existing Employee" HorizontalAlignment="Left"
Margin="42,291,0,0" VerticalAlignment="Top" Height="170" Width="356"/>
        </Grid>

        <!--Uncomment to see an alignment grid to help ensure your controls are
            aligned on common boundaries.  The image has a top margin of -32px to
            account for the System Tray. Set this to 0 (or remove the margin
altogether)
            if the System Tray is hidden.
```

```
            Before shipping remove this XAML and the image itself.-->
        <!--<Image Source="/Assets/AlignmentGrid.png" VerticalAlignment="Top"
Height="800" Width="480" Margin="0,-32,0,0" Grid.Row="0" Grid.RowSpan="2"
IsHitTestVisible="False" />-->
    </Grid>

</phone:PhoneApplicationPage>
```

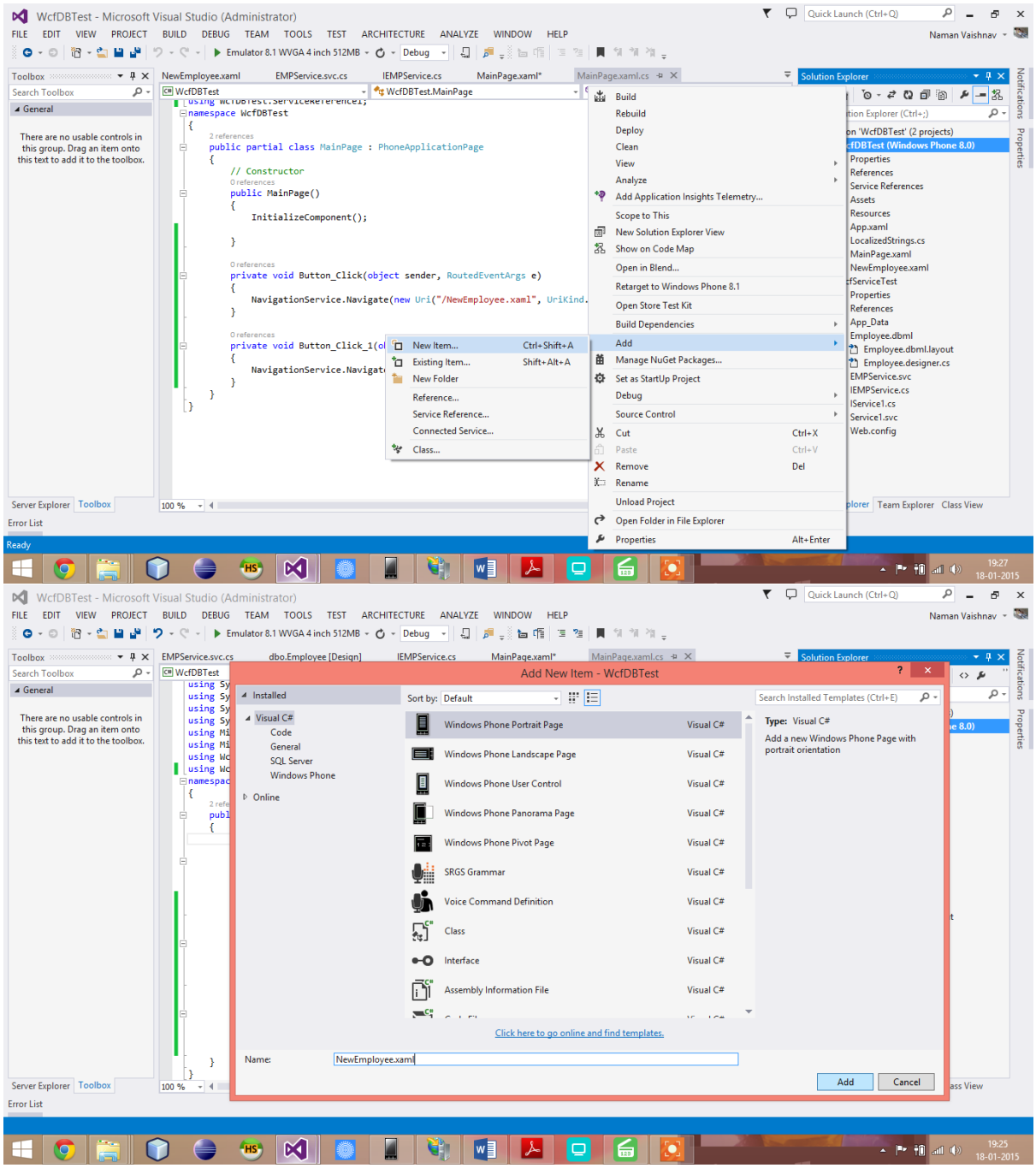
Now Go to this Button's Click Event

MainPage.Xaml.cs :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using WcfDBTest.Resources;
using WcfDBTest.ServiceReference1;
namespace WcfDBTest
{
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();

        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            NavigationService.Navigate(new Uri("/NewEmployee.xaml",
UriKind.Relative));
        }

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            NavigationService.Navigate(new Uri("/ViewEmployee.xaml",
UriKind.Relative));
        }
    }
}
```



Now Create new portrait page in the WcfDbTest Project named **NewEmployee.xaml** and **ViewEmployee.xaml**



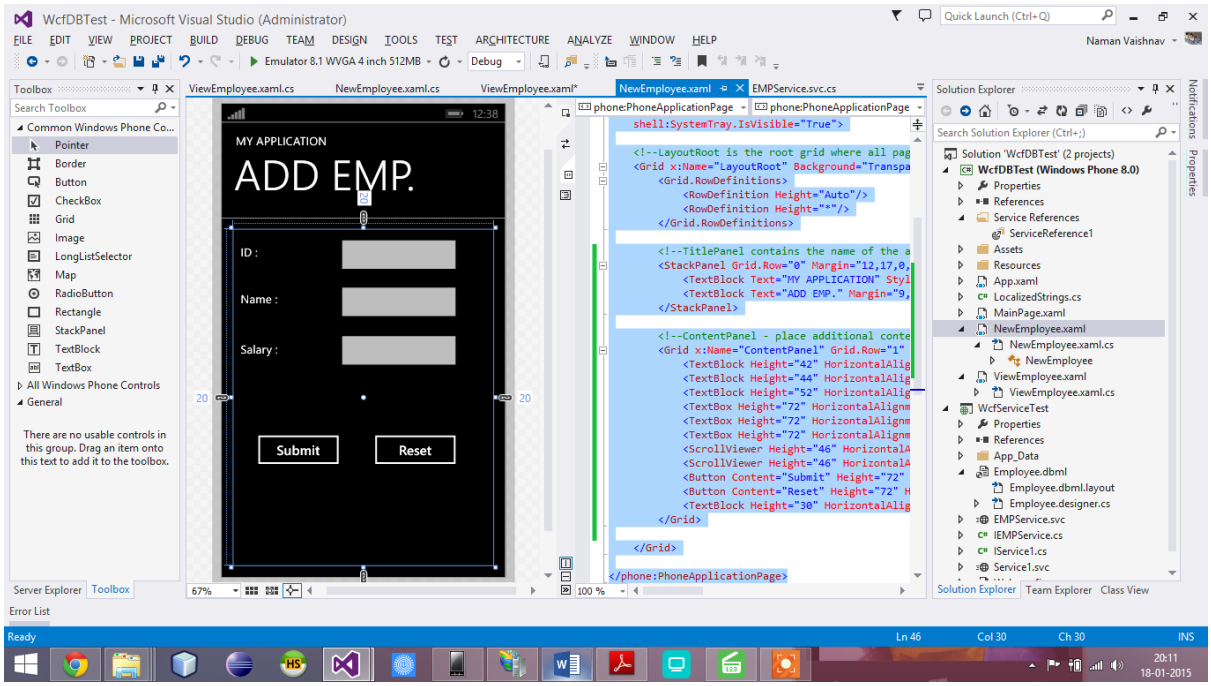
Now in the NewEmployee page

Create Layout First

NewEmployee.xaml

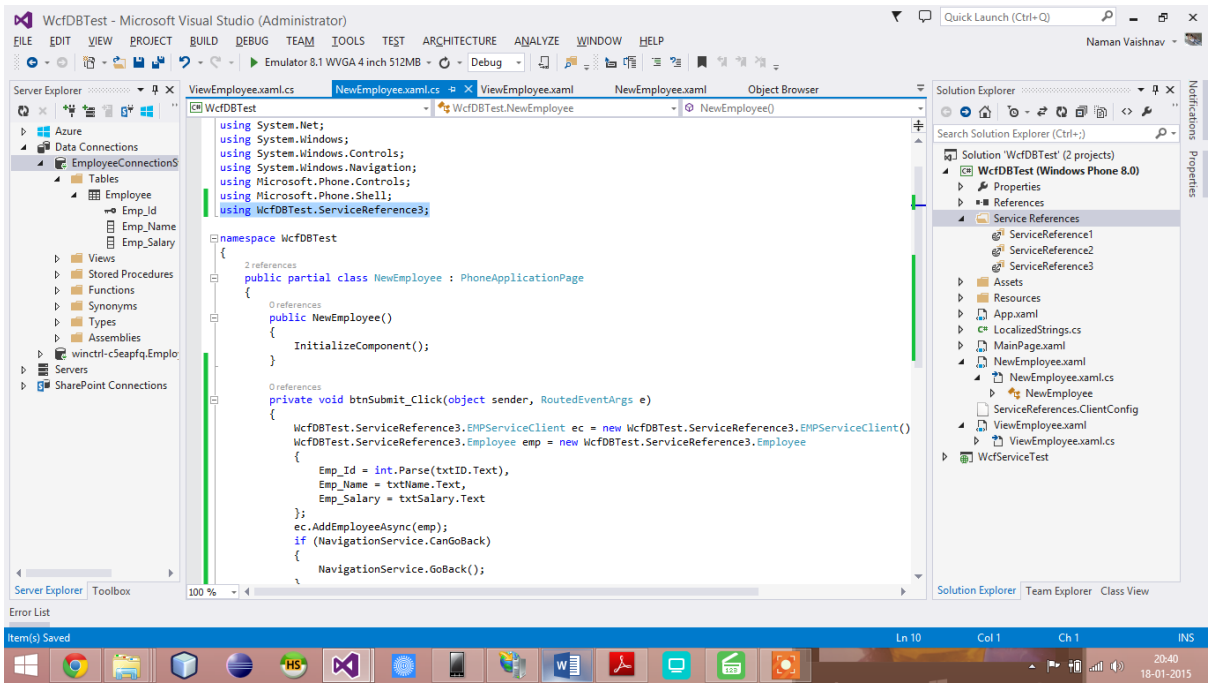
```
<phone:PhoneApplicationPage
    x:Class="WcfDBTest.NewEmployee"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```

```
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    mc:Ignorable="d"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel Grid.Row="0" Margin="12,17,0,28">
            <TextBlock Text="MY APPLICATION" Style="{StaticResource
PhoneTextNormalStyle}"/>
            <TextBlock Text="ADD EMP." Margin="9,-7,0,0" Style="{StaticResource
PhoneTextTitle1Style}"/>
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="20">
            <TextBlock Height="42" HorizontalAlignment="Left" Margin="12,24,0,0"
x:Name="textBlock1" Text="ID :" VerticalAlignment="Top" Width="77" />
            <TextBlock Height="44" HorizontalAlignment="Left" Margin="12,102,0,0"
x:Name="textBlock2" Text="Name :" VerticalAlignment="Top" Width="99" />
            <TextBlock Height="52" HorizontalAlignment="Left" Margin="12,188,0,0"
x:Name="textBlock3" Text="Salary : " VerticalAlignment="Top" Width="77" />
            <TextBox Height="72" HorizontalAlignment="Left" Margin="172,6,0,0"
x:Name="txtID" Text="" VerticalAlignment="Top" Width="216" />
            <TextBox Height="72" HorizontalAlignment="Left" Margin="172,86,0,0"
x:Name="txtName" Text="" VerticalAlignment="Top" Width="216" />
            <TextBox Height="72" HorizontalAlignment="Left" Margin="172,168,0,0"
x:Name="txtSalary" Text="" VerticalAlignment="Top" Width="216" />
            <ScrollViewer Height="46" HorizontalAlignment="Left" Margin="188,281,0,0"
x:Name="scrDept" VerticalAlignment="Top" Width="200"
VerticalScrollBarVisibility="Visible"/>
            <ScrollViewer Height="46" HorizontalAlignment="Left" Margin="188,367,0,0"
x:Name="scrDesg" VerticalAlignment="Top" Width="200"/>
            <Button Content="Submit" Height="72" HorizontalAlignment="Left"
Margin="30,336,0,0" x:Name="btnSubmit" VerticalAlignment="Top" Width="160"
Click="btnSubmit_Click" />
            <Button Content="Reset" Height="72" HorizontalAlignment="Left"
Margin="228,336,0,0" x:Name="btnReset" VerticalAlignment="Top" Width="160" />
            <TextBlock Height="30" HorizontalAlignment="Left" Margin="30,518,0,0"
x:Name="txtMessage" Text="" VerticalAlignment="Top" Width="385" Foreground="#FF2E88D4"
/>
        </Grid>

    </Grid>

</phone:PhoneApplicationPage>
```



Enter Submit buttons Click Event

Do Not Forgot To Add NameSpace Of Service Reference



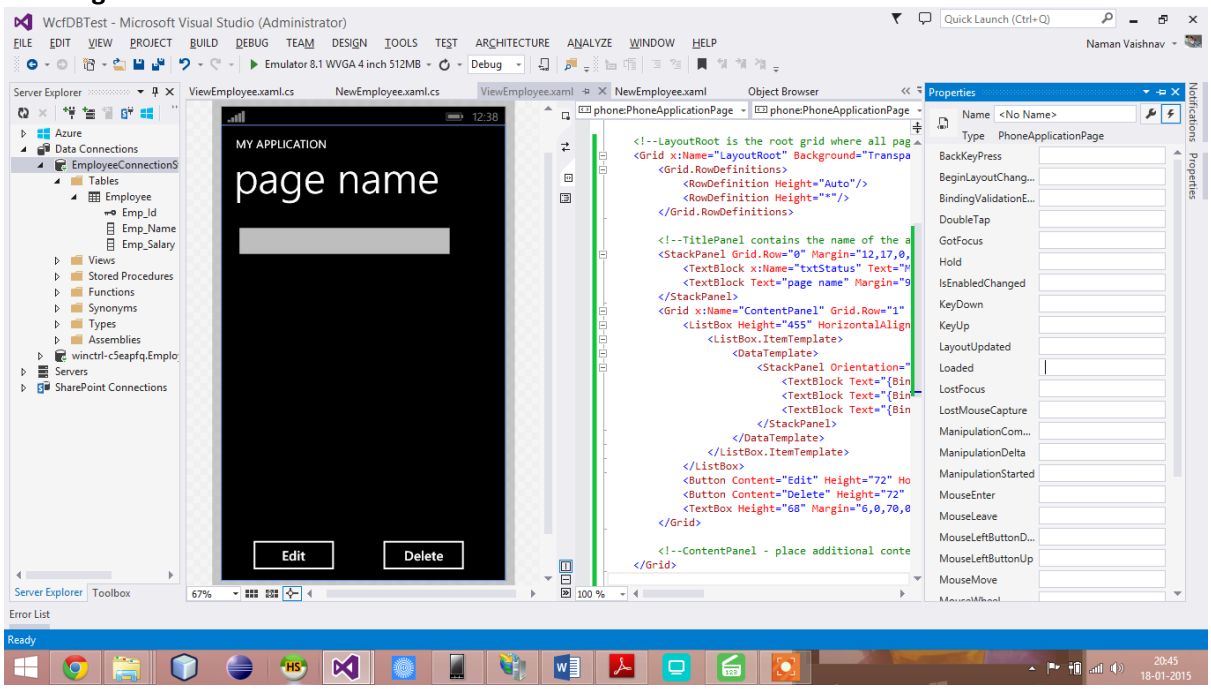
Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;
```

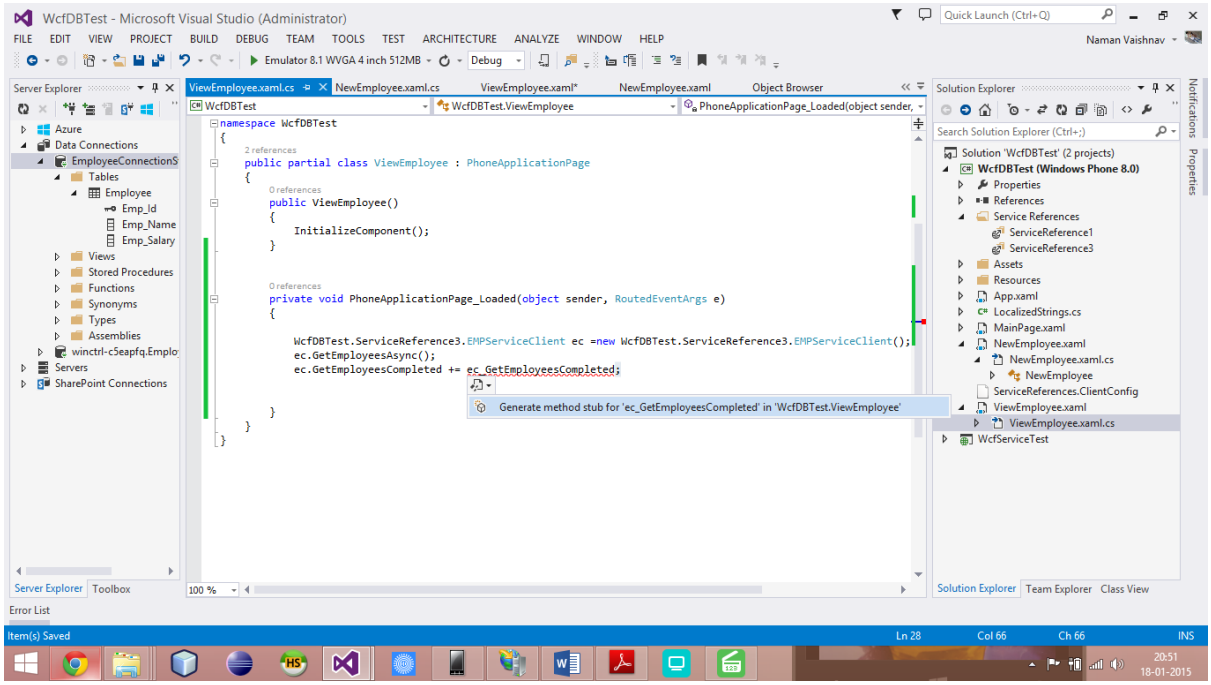
```
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using WcfDBTest.ServiceReference3;

namespace WcfDBTest
{
    public partial class NewEmployee : PhoneApplicationPage
    {
        public NewEmployee()
        {
            InitializeComponent();
        }

        private void btnSubmit_Click(object sender, RoutedEventArgs e)
        {
            WcfDBTest.ServiceReference3.EMPServiceClient ec = new
WcfDBTest.ServiceReference3.EMPServiceClient();
            WcfDBTest.ServiceReference3.Employee emp = new
WcfDBTest.ServiceReference3.Employee
            {
                Emp_Id = int.Parse(txtID.Text),
                Emp_Name = txtName.Text,
                Emp_Salary = txtSalary.Text
            };
            ec.AddEmployeeAsync(emp);
            if (NavigationService.CanGoBack)
            {
                NavigationService.GoBack();
            }
        }
    }
}
```

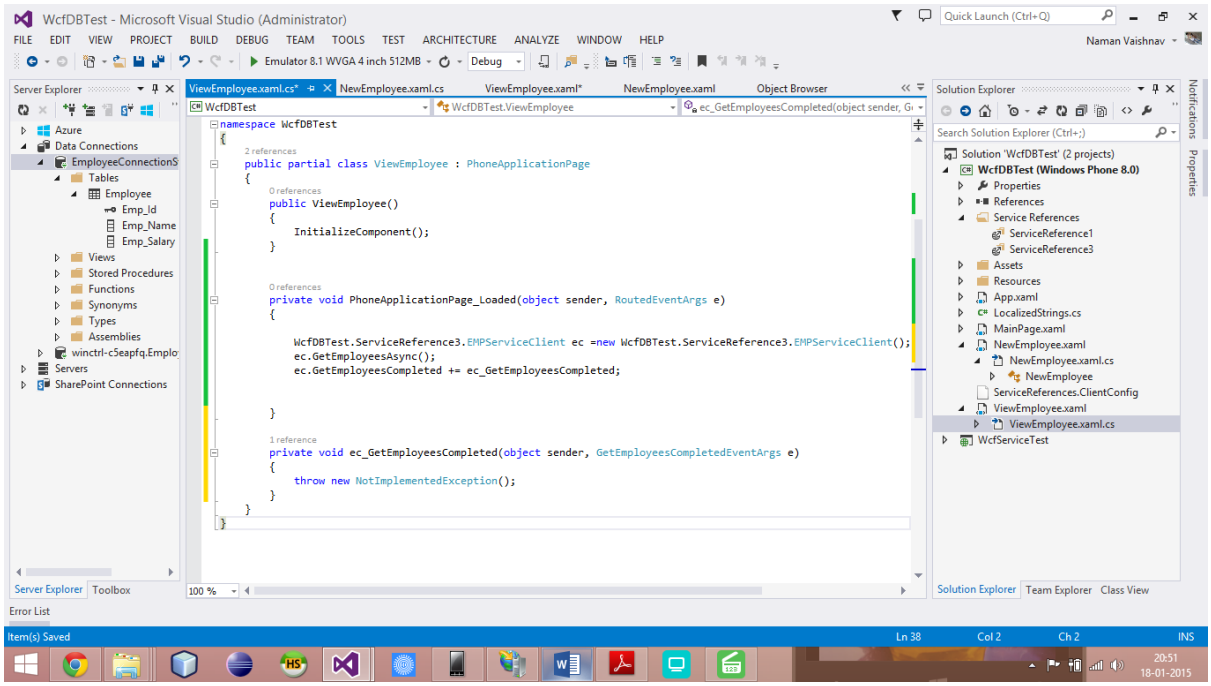
Now code in The **ViewEmployee.Xaml**

1st Design it

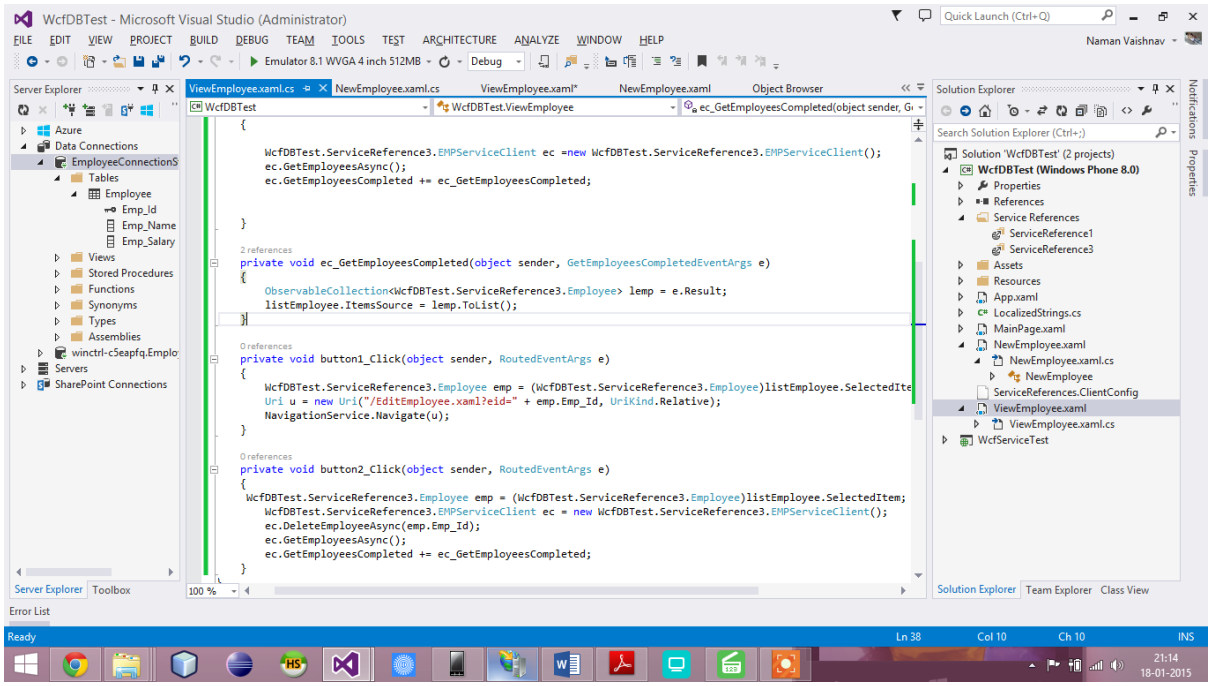


Now code into the Loded Event






```
        0 references
        private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)
        {

            WcfDBTest.ServiceReference3.EMPServiceClient ec =new WcfDBTest.ServiceReference3.EMPServiceCli
            ec.GetEmployeesAsync();
            ec.GetEmployeesCompleted += ec_GetEmployeesCompleted;


        }

                                THis Code Wants To import One Namespace
        1 reference
        private void ec_GetEmployeesCompleted(object sender, GetEmployeesCompletedEventArgs e)
        {
            ObservableCollection<WcfDBTest.ServiceReference3.Employee> lemp = e.Result;
            listEmployee.ItemsSource = lemp.ToList();
        }
    }
}
```

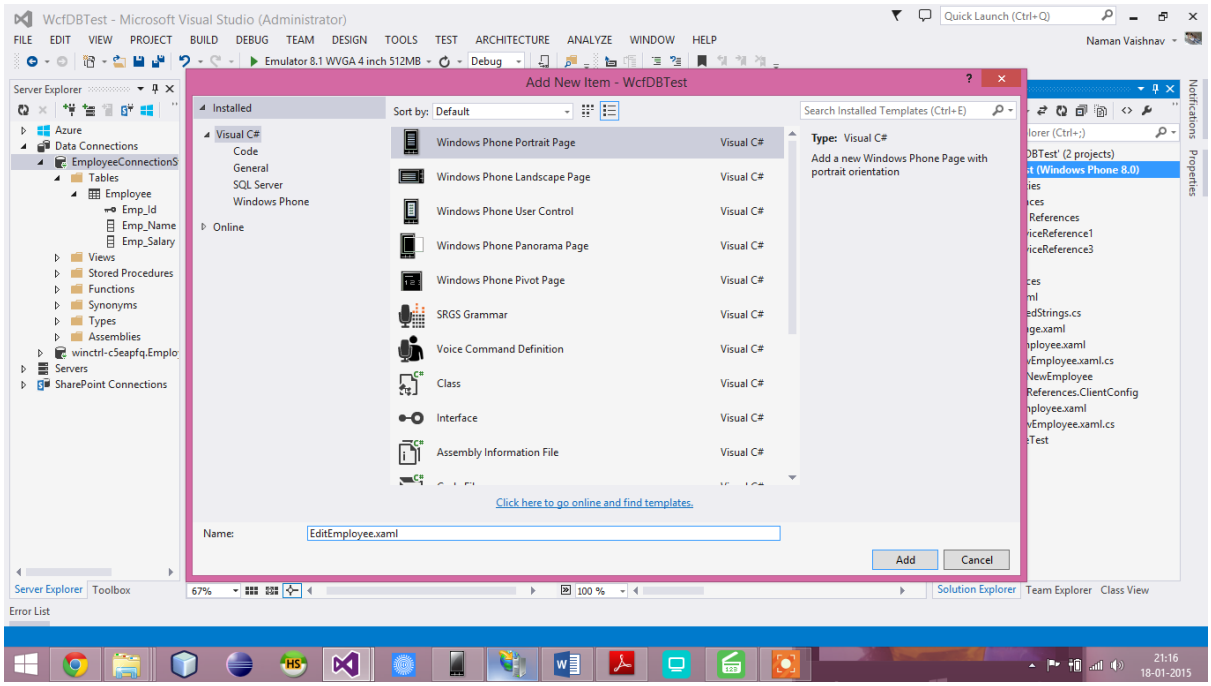


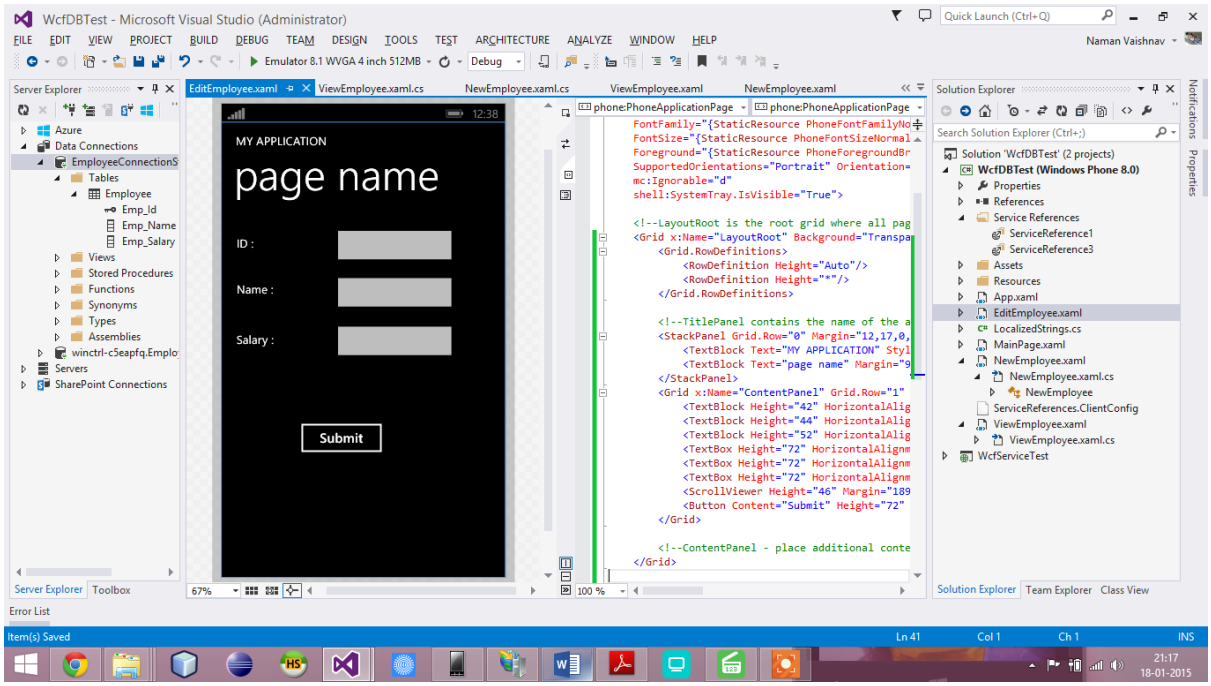
Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
//using WcfDBTest.ServiceReference1;
using WcfDBTest.ServiceReference3;
using System.Collections.ObjectModel;

namespace WcfDBTest
{
    public partial class ViewEmployee : PhoneApplicationPage
    {
        public ViewEmployee()
        {
            InitializeComponent();
        }


        private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)
        {

            WcfDBTest.ServiceReference3.EMPServiceClient ec =new
WcfDBTest.ServiceReference3.EMPServiceClient();
            ec.GetEmployeesAsync();
            ec.GetEmployeesCompleted += ec_GetEmployeesCompleted;


        }

        private void ec_GetEmployeesCompleted(object sender,
GetEmployeesCompletedEventArgs e)
```

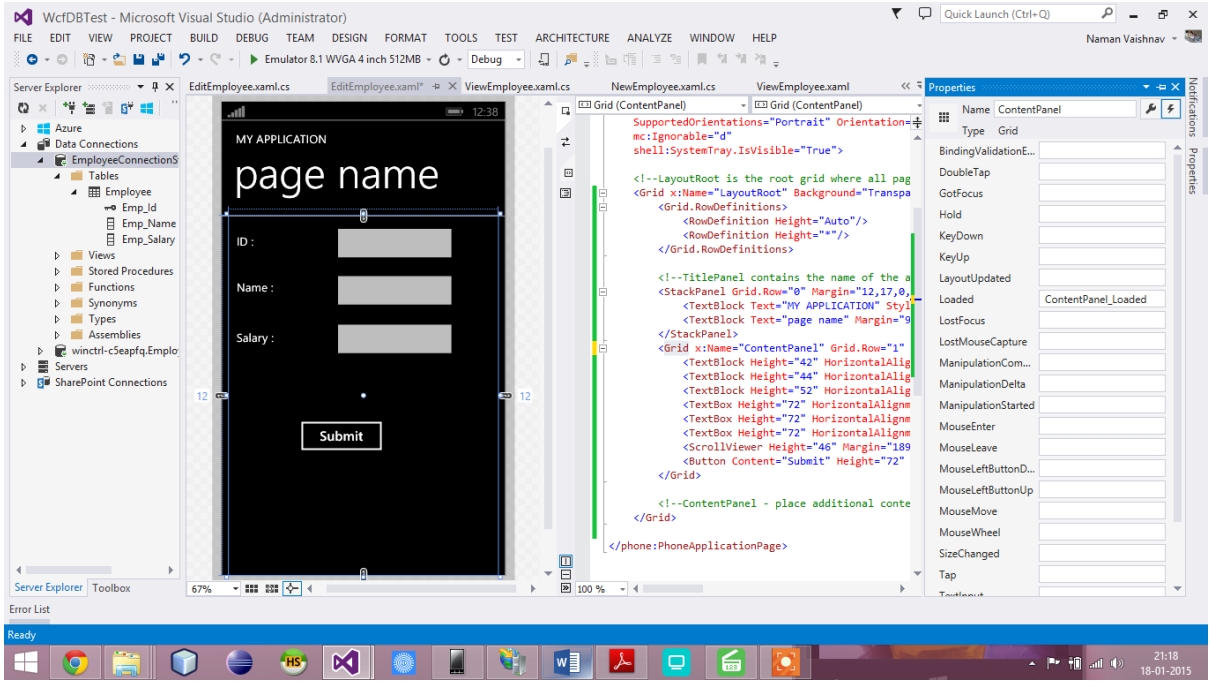
```
        {
            ObservableCollection<WcfDBTest.ServiceReference3.Employee> lemp =
e.Result;
            listEmployee.ItemsSource = lemp.ToList();
        }

        private void button1_Click(object sender, RoutedEventArgs e)
        {
            WcfDBTest.ServiceReference3.Employee emp =
(WcfDBTest.ServiceReference3.Employee)listEmployee.SelectedItem;
            Uri u = new Uri("/EditEmployee.xaml?eid=" + emp.Emp_Id, UriKind.Relative);
            NavigationService.Navigate(u);
        }

        private void button2_Click(object sender, RoutedEventArgs e)
        {
         WcfDBTest.ServiceReference3.Employee emp =
(WcfDBTest.ServiceReference3.Employee)listEmployee.SelectedItem;
            WcfDBTest.ServiceReference3.EMPServiceClient ec = new
WcfDBTest.ServiceReference3.EMPServiceClient();
            ec.DeleteEmployeeAsync(emp.Emp_Id);
            ec.GetEmployeesAsync();
            ec.GetEmployeesCompleted += ec_GetEmployeesCompleted;
        }
    }
}
```

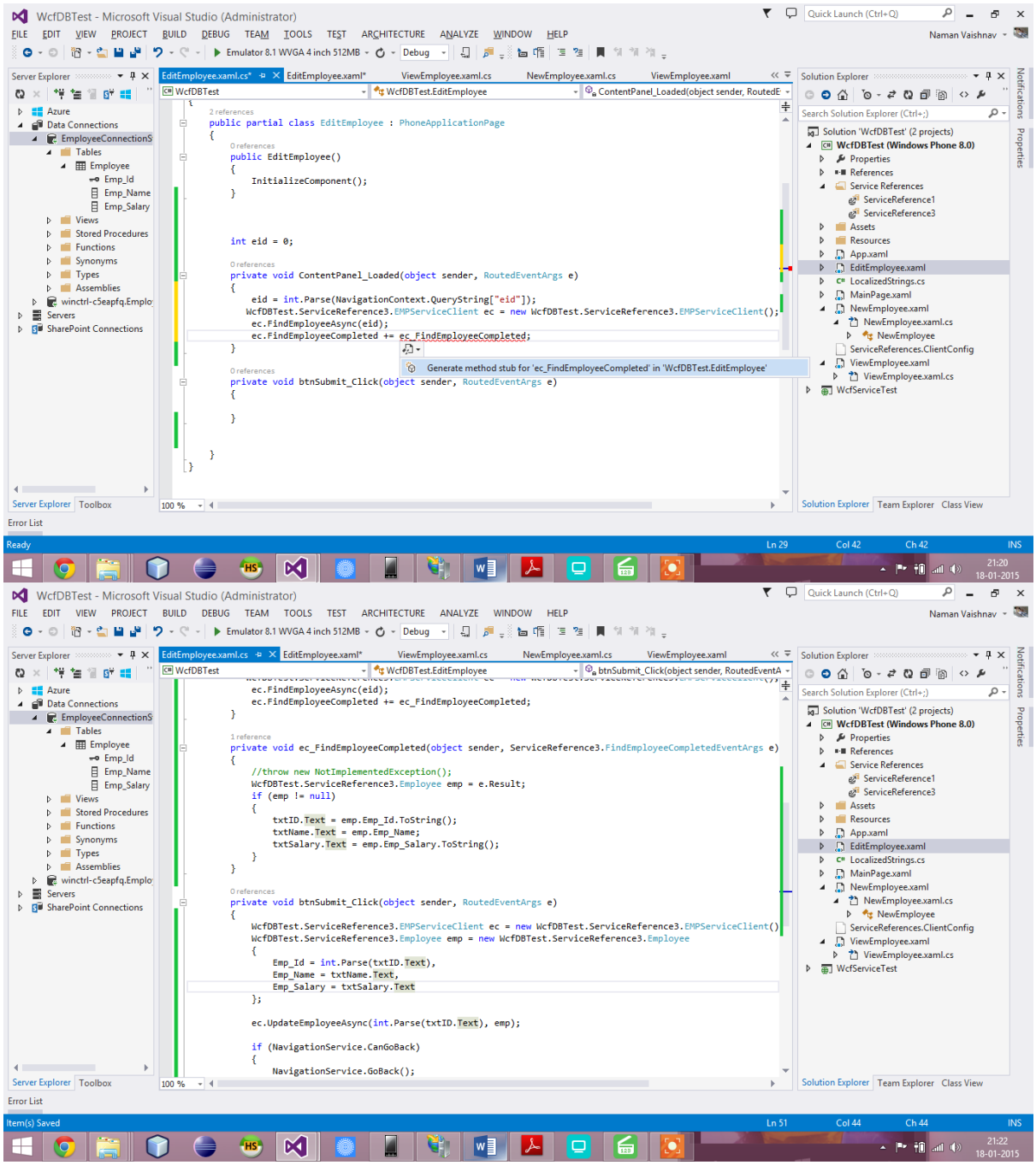
Now Add **EditEmployee.Xaml** Page





Into the Lode Event





Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;

namespace WcfDBTest
{
    public partial class EditEmployee : PhoneApplicationPage
```

```
    {
        public EditEmployee()
        {
            InitializeComponent();
        }



        int eid = 0;

        private void ContentPanel_Loaded(object sender, RoutedEventArgs e)
        {
            eid = int.Parse(NavigationContext.QueryString["eid"]);
           WcfDBTest.ServiceReference3.EMPServiceClient ec = new
WcfDBTest.ServiceReference3.EMPServiceClient();
            ec.FindEmployeeAsync(eid);
            ec.FindEmployeeCompleted += ec_FindEmployeeCompleted;
        }

        private void ec_FindEmployeeCompleted(object sender,
ServiceReference3.FindEmployeeCompletedEventArgs e)
        {
            //throw new NotImplementedException();
            WcfDBTest.ServiceReference3.Employee emp = e.Result;
            if (emp != null)
            {
                txtID.Text = emp.Emp_Id.ToString();
                txtName.Text = emp.Emp_Name;
                txtSalary.Text = emp.Emp_Salary.ToString();
            }
        }

        private void btnSubmit_Click(object sender, RoutedEventArgs e)
        {
            WcfDBTest.ServiceReference3.EMPServiceClient ec = new
WcfDBTest.ServiceReference3.EMPServiceClient();
            WcfDBTest.ServiceReference3.Employee emp = new
WcfDBTest.ServiceReference3.Employee
            {
                Emp_Id = int.Parse(txtID.Text),
                Emp_Name = txtName.Text,
                Emp_Salary = txtSalary.Text
            };

            ec.UpdateEmployeeAsync(int.Parse(txtID.Text), emp);

            if (NavigationService.CanGoBack)
            {
                NavigationService.GoBack();
            }
        }


    }
}
```