https://www.halvorsen.blog

Database Communication in Visual Studio/C# using Web Services

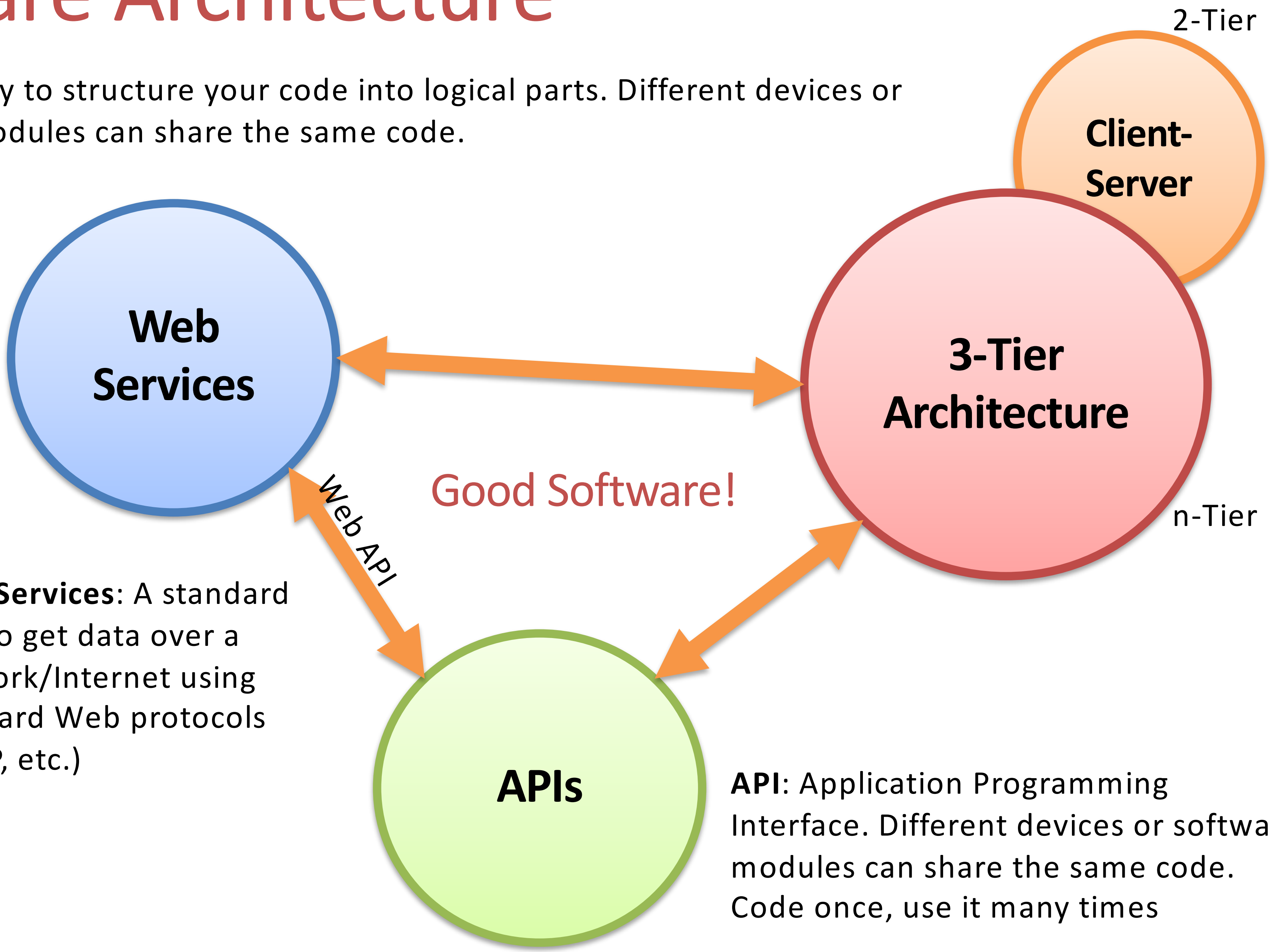
Hans-Petter Halvorsen

Background

- With Web Services you can easily get your data through Internet
- We will use **Web Services** because we assume that the the App should be used on Internet outside the Firewall).
- The Database is located on a Server that has no direct access to the Internet.

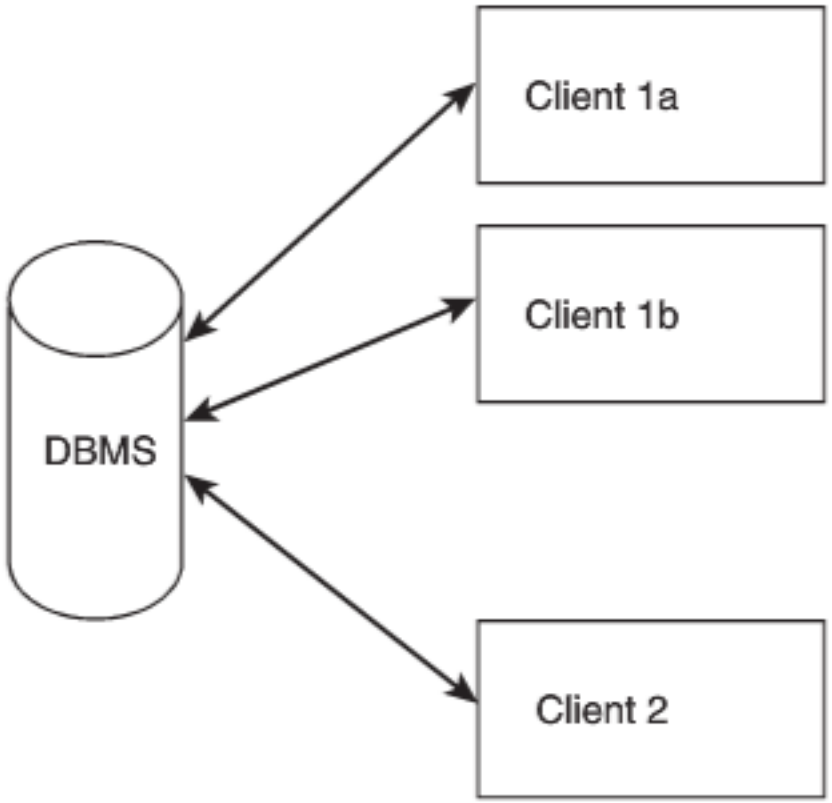
Software Architecture

3-Tier: A way to structure your code into logical parts. Different devices or software modules can share the same code.

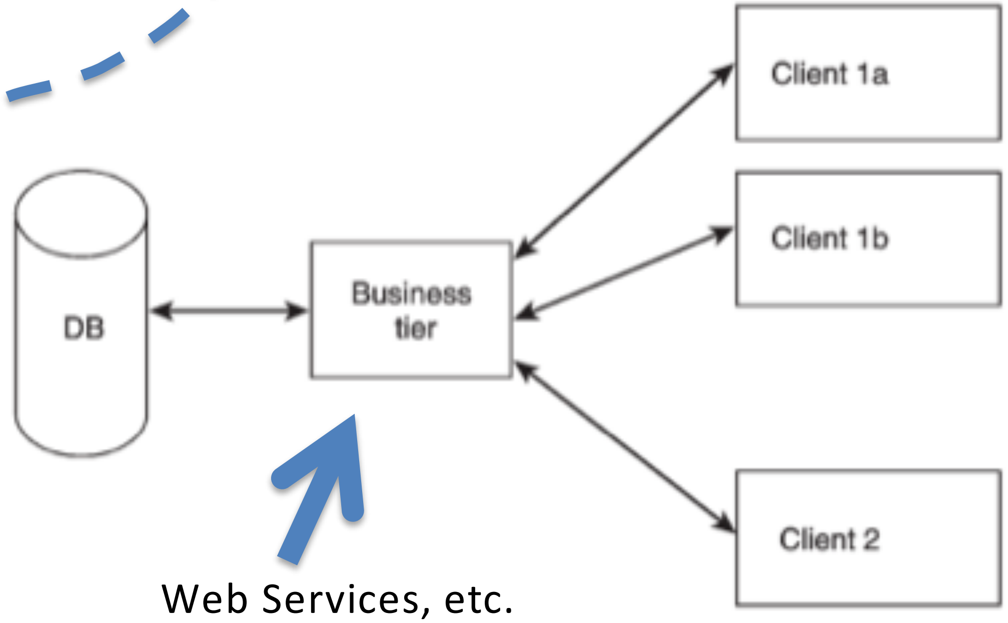


Web Services: A standard way to get data over a network/Internet using standard Web protocols (HTTP, etc.)

API: Application Programming Interface. Different devices or software modules can share the same code. Code once, use it many times



The database-centric style. Typically, the clients communicate directly with the database.

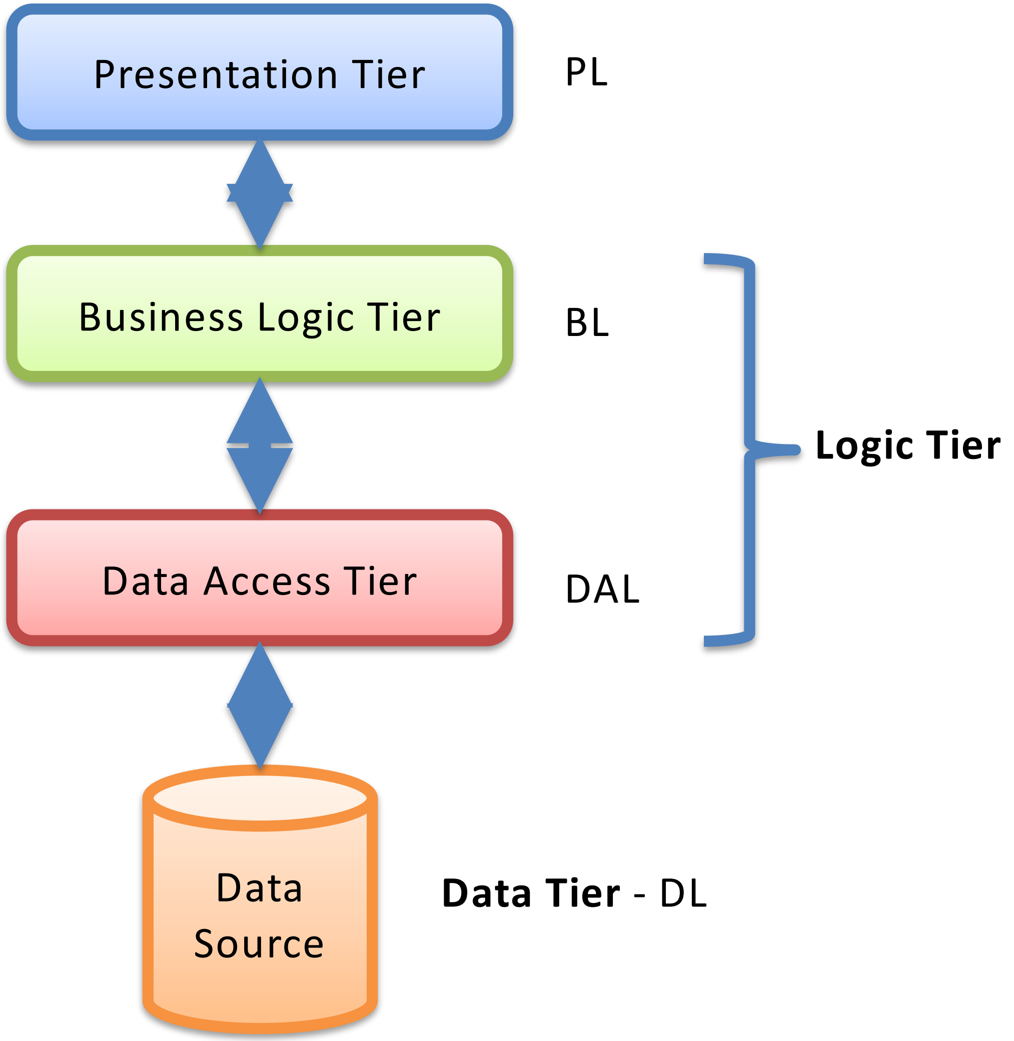


A three-tier style, in which clients do not connect directly to the database.

Web Services, etc.

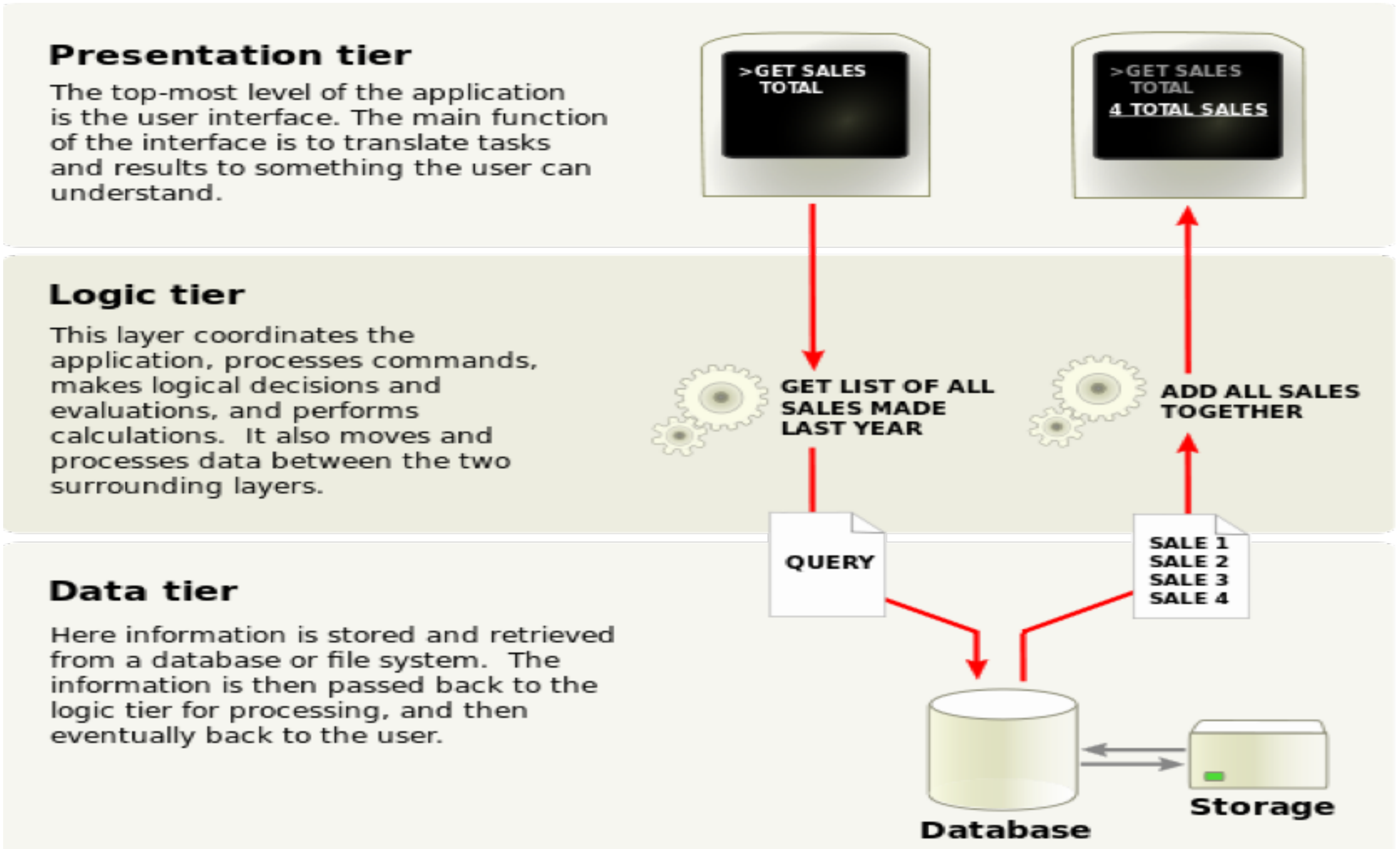
3-tier/layer Architecture

Note! The different layers can be on the same computer (Logic Layers) or on different Computers in a network (Physical Layers)



Why 3-Tier (N-Tier Architecture?)

- Flexible applications
- Reusable code
  - Code once, use many times
- Modularized
  - You need only to change part of the code
  - You can deploy only one part
  - You can Test only one part
  - Multiple Developers
- Different parts (Tiers) can be stored on different computers
- Different Platforms and Languages can be used
- etc.



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.

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.

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.

http://en.wikipedia.org/wiki/Multitier_architecture

3-tier/layer Architecture

Presentation Tier

- This is the topmost level of the application.
- The presentation tier displays information related to such services as browsing merchandise, purchasing and shopping cart contents.
- It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network.
- In simple terms it is a layer which users can access directly such as a web page, or an operating systems GUI

Application tier (business logic, logic tier, data access tier, or middle tier)

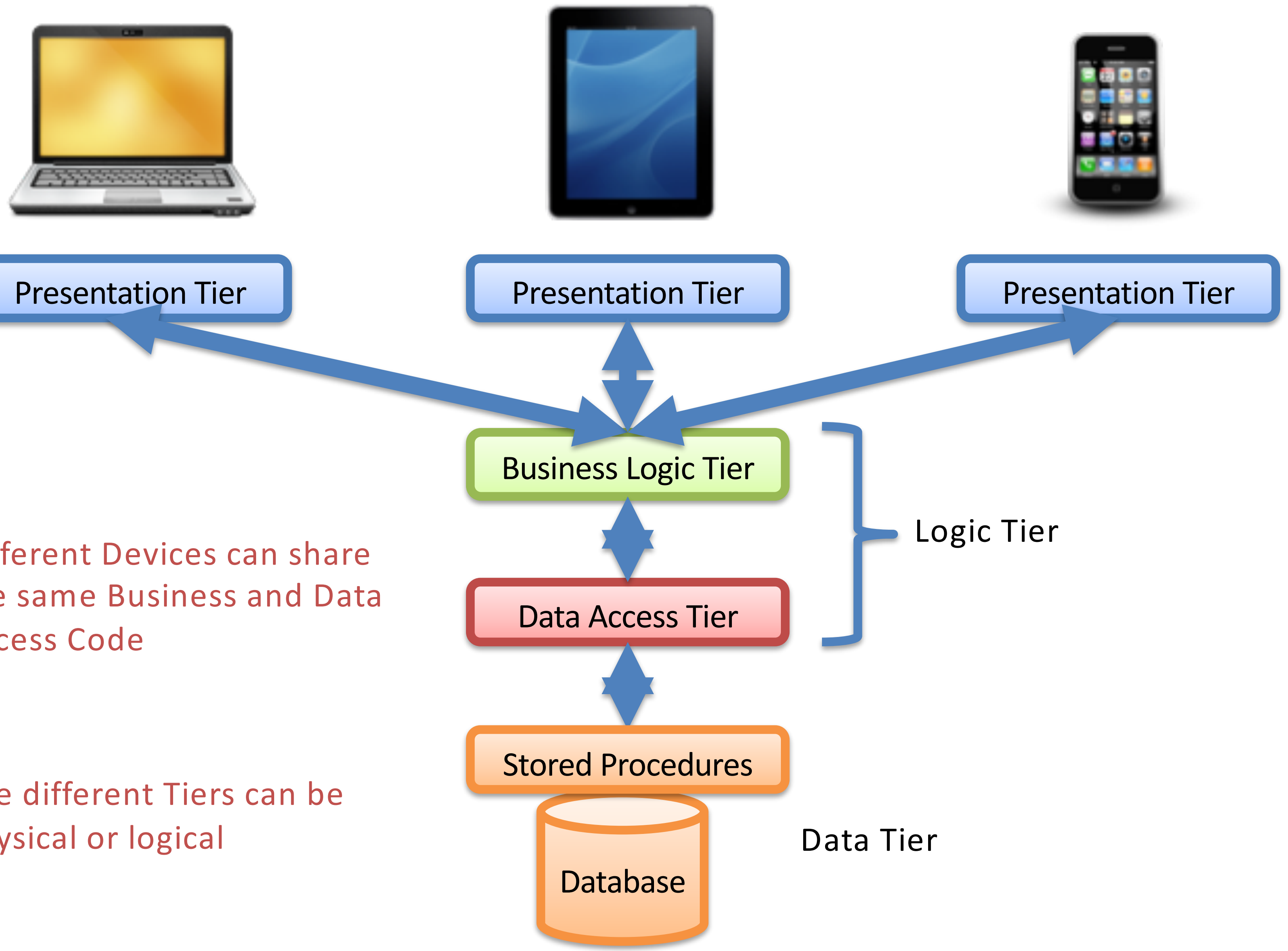
- The logical tier is pulled out from the presentation tier and, as its own layer.
- It controls an application's functionality by performing detailed processing.

Data tier

- This tier consists of database servers. Here information is stored and retrieved.
- This tier keeps data neutral and independent from application servers or business logic.
- Giving data its own tier also improves scalability and performance.

http://en.wikipedia.org/wiki/Multitier_architecture

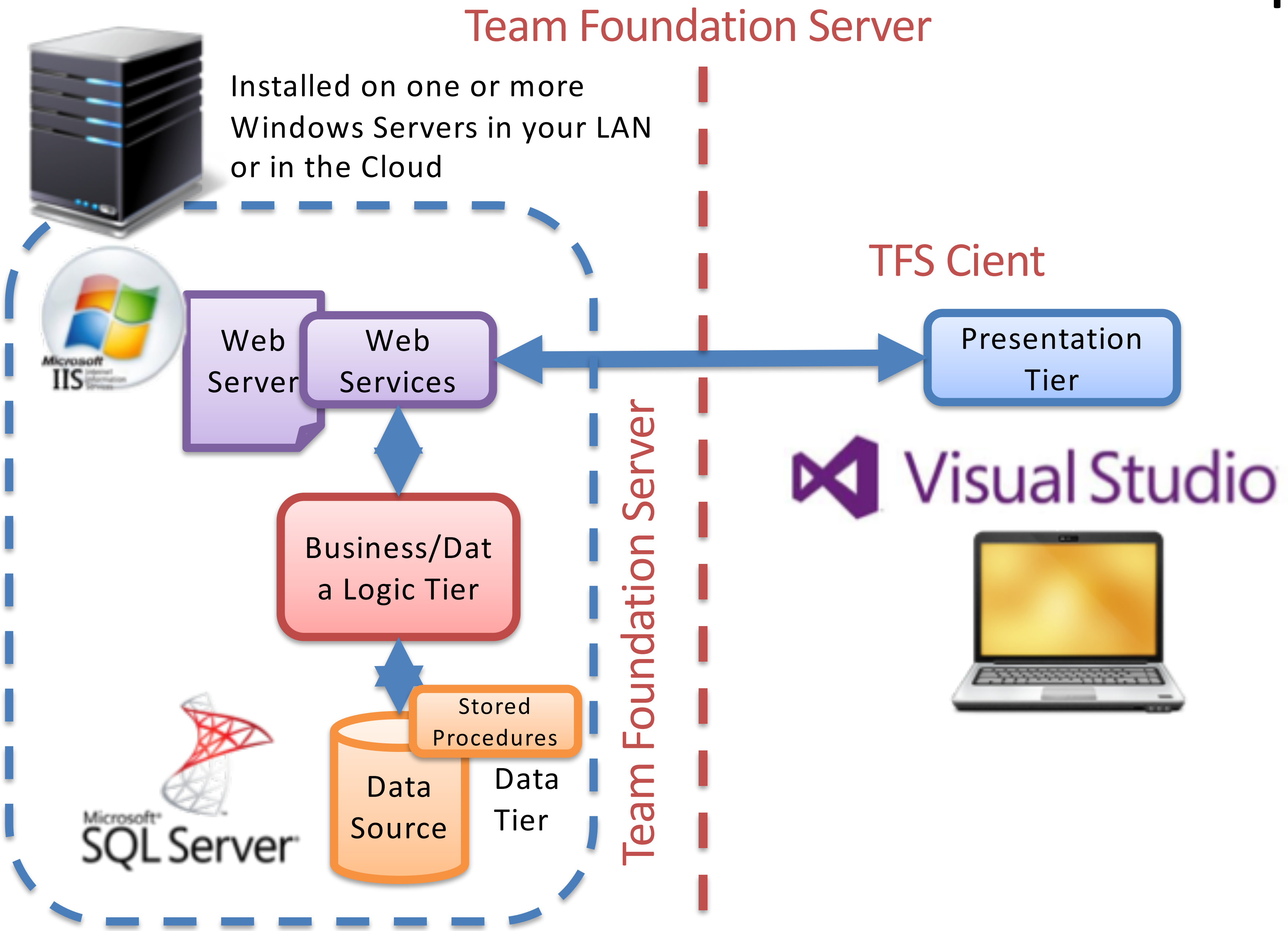
3-tier Architecture



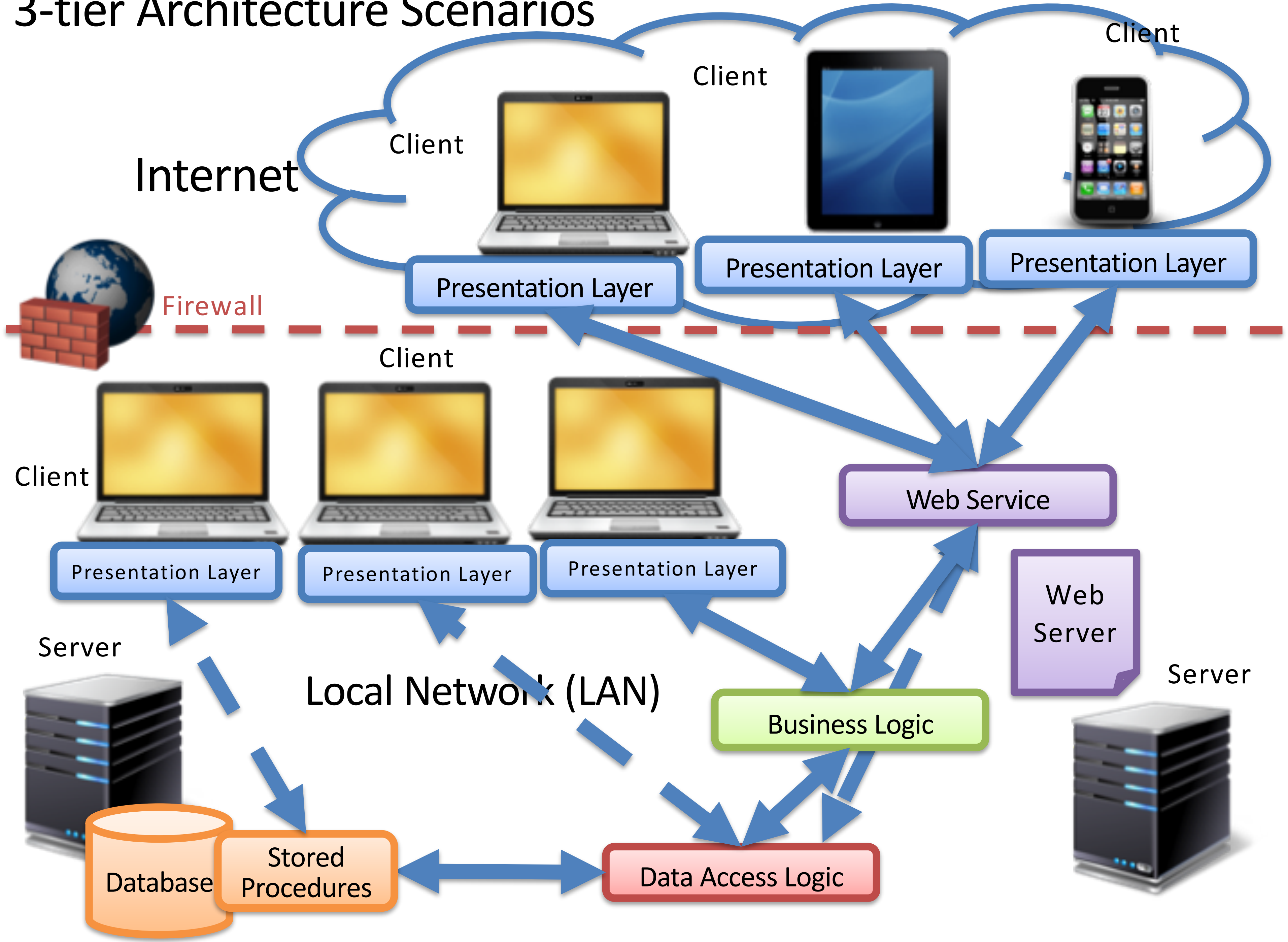
Different Devices can share the same Business and Data Access Code

The different Tiers can be physical or logical

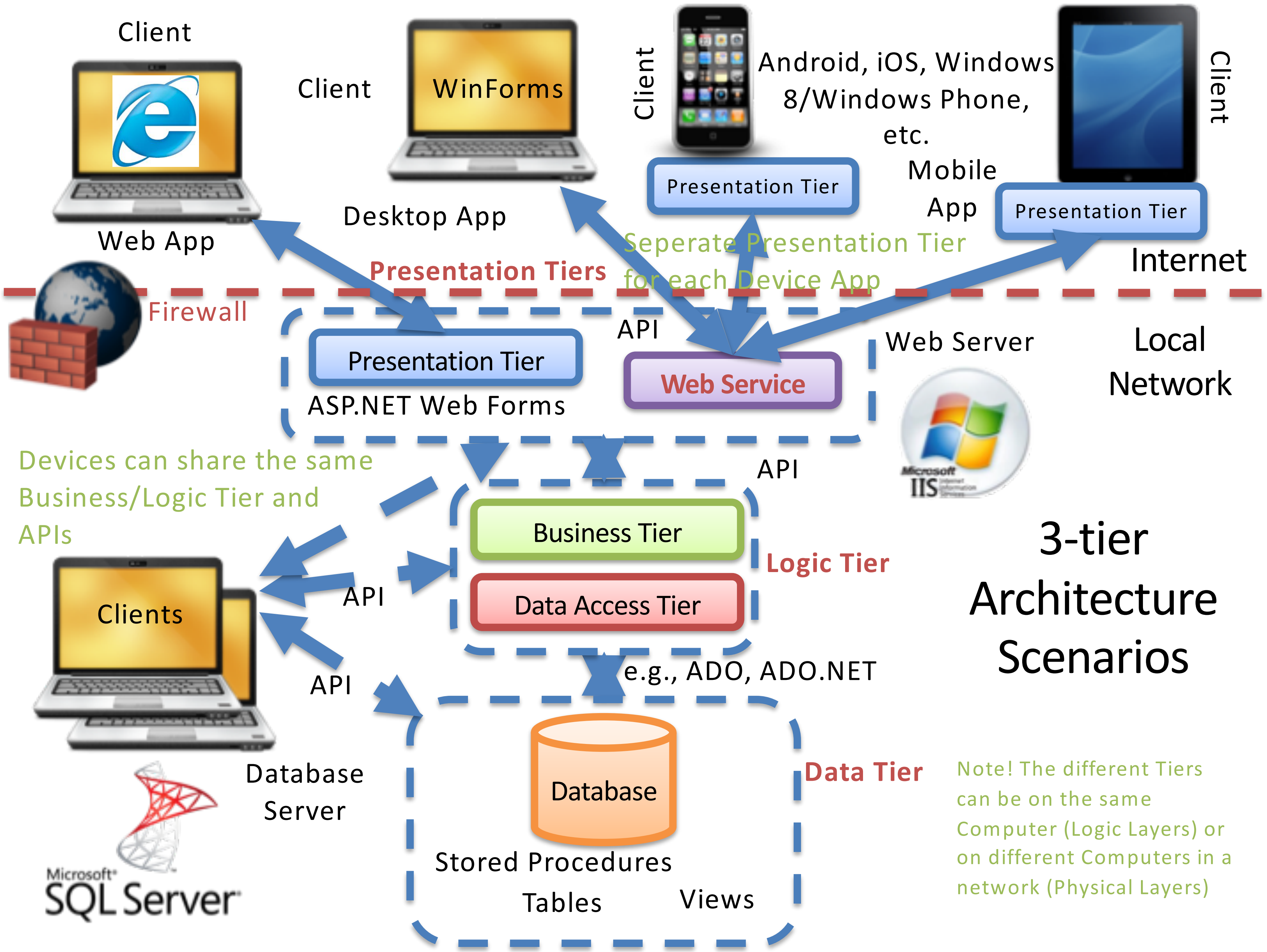
3-tier + WebService Architecture - Example



3-tier Architecture Scenarios



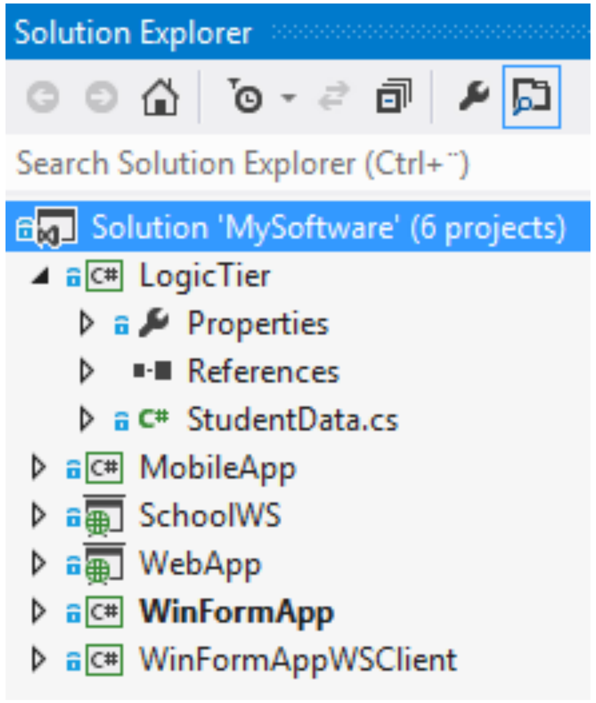
3-tier Architecture Scenarios



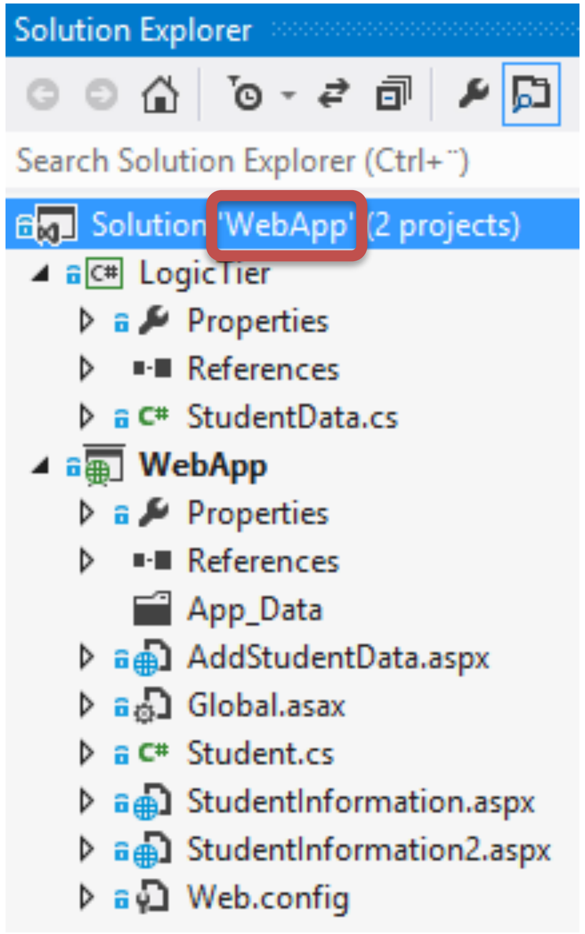
Note! The different Tiers can be on the same Computer (Logic Layers) or on different Computers in a network (Physical Layers)

Visual Studio Projects

Solution with all Projects (Logic Tier, Web Service, Desktop App, Web App, Mobile App)

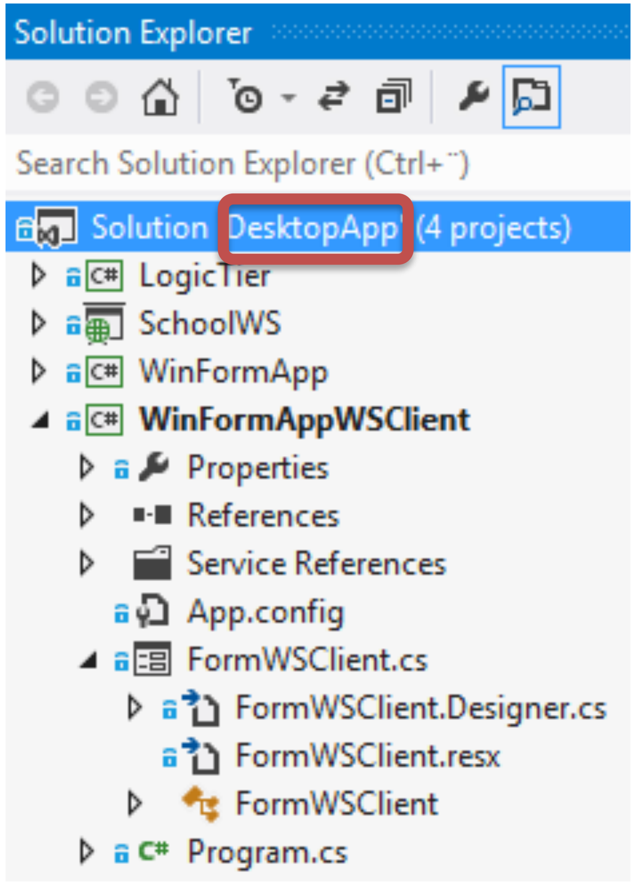


```
Solution Explorer
Search Solution Explorer (Ctrl+¨)
Solution 'MySoftware' (6 projects)
  LogicTier
    Properties
    References
    StudentData.cs
  MobileApp
  SchoolWS
  WebApp
  WinFormApp
  WinFormAppWSClient
```

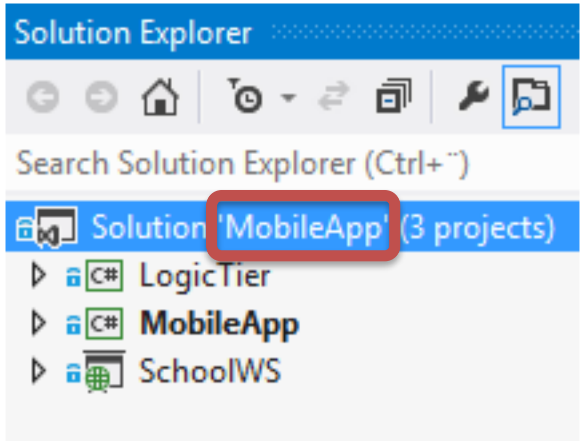


```
Solution Explorer
Search Solution Explorer (Ctrl+¨)
Solution 'WebApp' (2 projects)
  LogicTier
    Properties
    References
    StudentData.cs
  WebApp
    Properties
    References
    App_Data
    AddStudentData.aspx
    Global.asax
    Student.cs
    StudentInformation.aspx
    StudentInformation2.aspx
    Web.config
```

Solution with Projects used by Web App (Logic Tier, Web App)

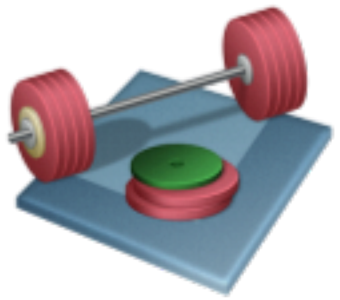


```
Solution Explorer
Search Solution Explorer (Ctrl+¨)
Solution 'DesktopApp' (4 projects)
  LogicTier
  SchoolWS
  WinFormApp
  WinFormAppWSClient
    Properties
    References
    Service References
    App.config
    FormWSClient.cs
      FormWSClient.Designer.cs
      FormWSClient.resx
      FormWSClient
    Program.cs
```



```
Solution Explorer
Search Solution Explorer (Ctrl+¨)
Solution 'MobileApp' (3 projects)
  LogicTier
  MobileApp
  SchoolWS
```

Data Tier



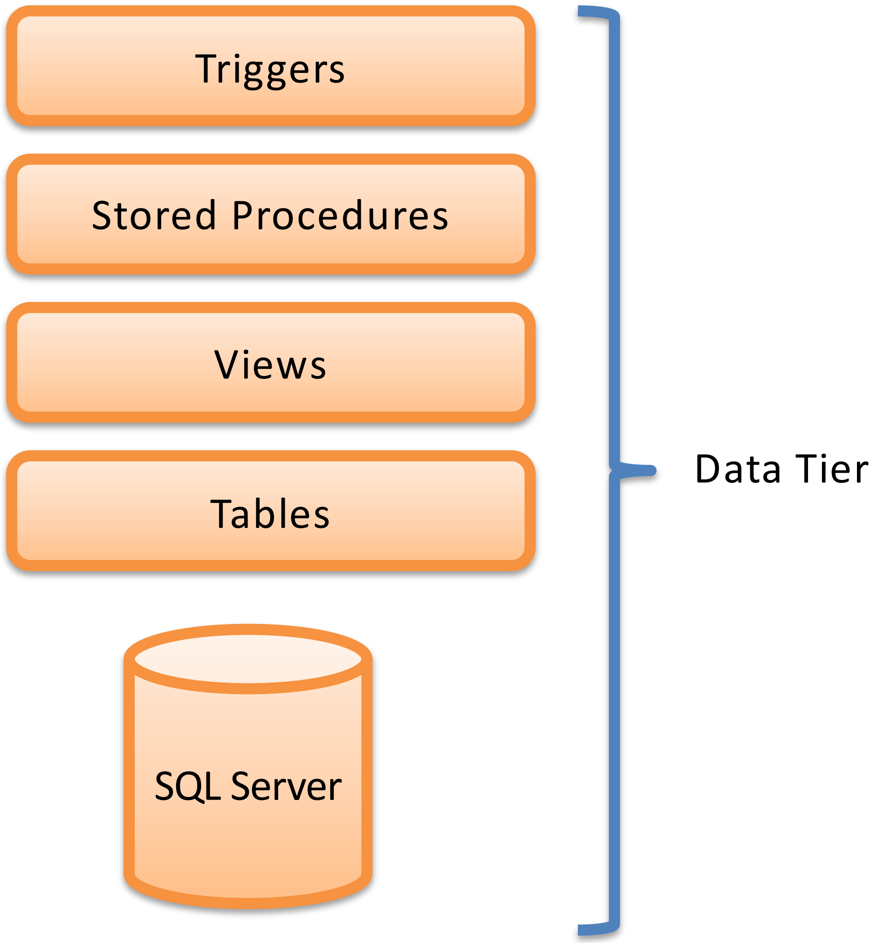
We are going to create the Database / Data Layer/Tier, including:

1. Tables
2. Views
3. Stored Procedures
4. Triggers
5. Script for some “Dummy” Data

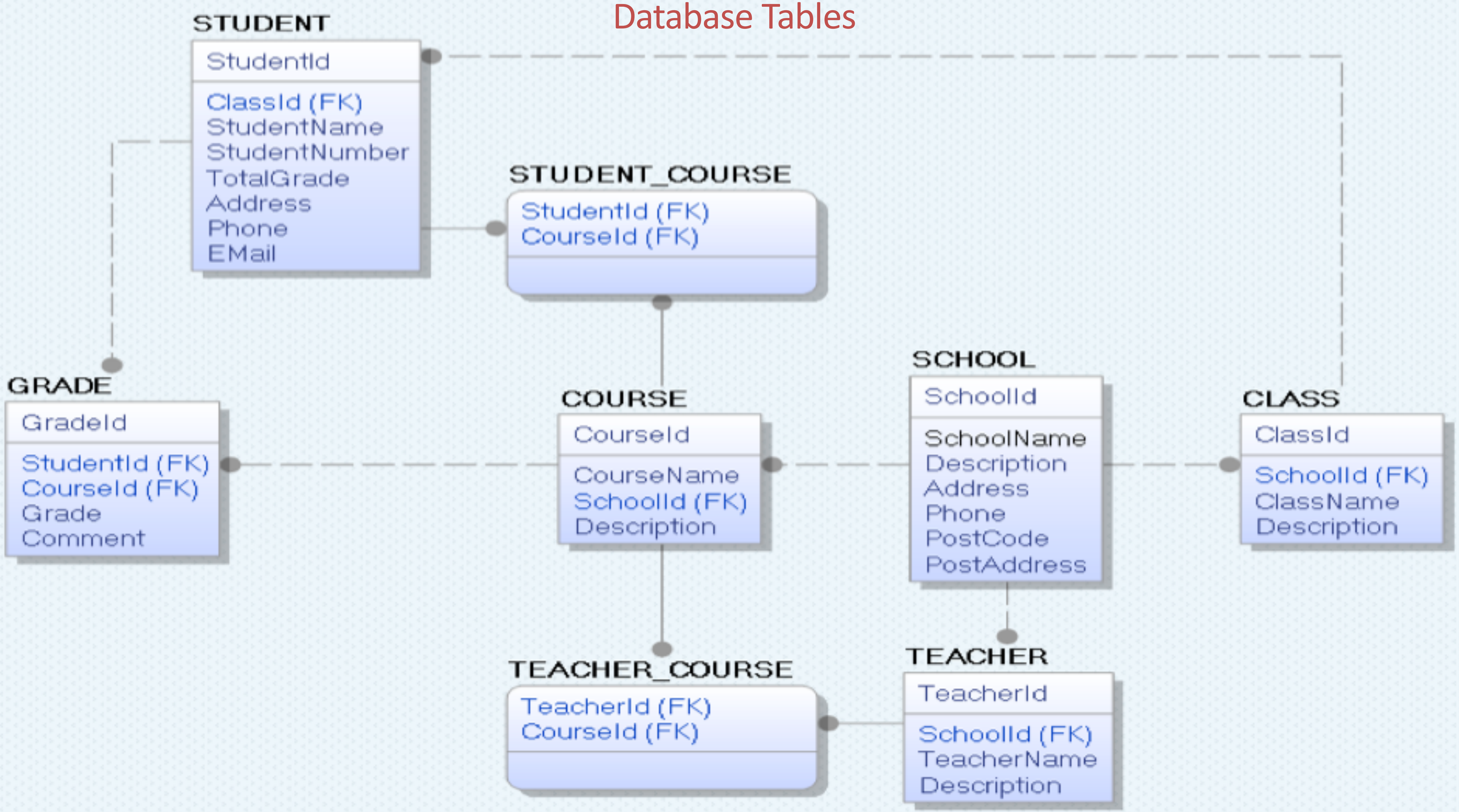
Note! Install them in this order

Download Zip Files with Tables, Views, Stored Procedures and Triggerse in order to create the Data Tier in SQL Server (The ZIP File is located on the same place as this File)

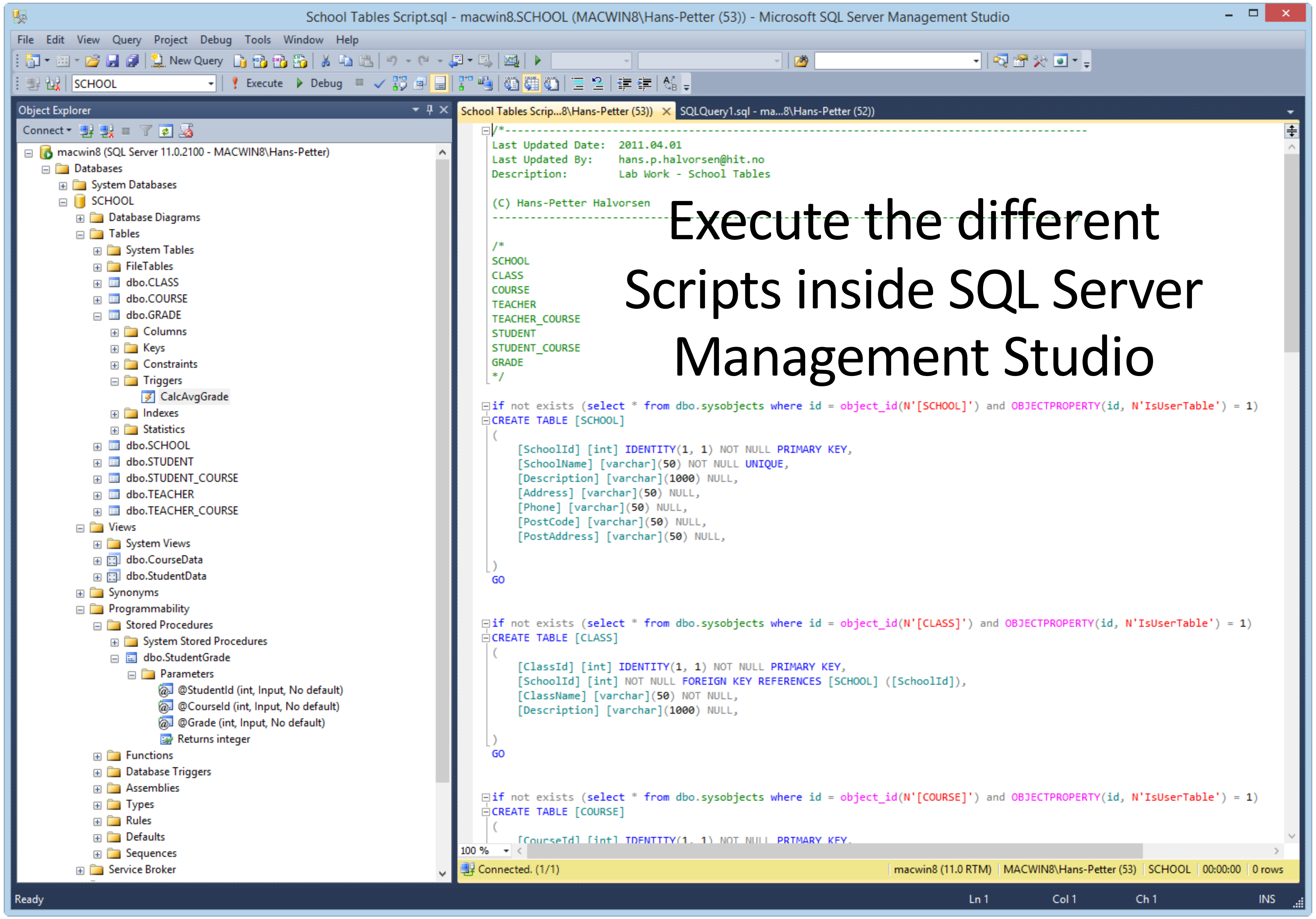
Data Tier


Triggers
Stored Procedures
Views
Tables
SQL Server
Data Tier

Database Tables



Execute the different Scripts inside SQL Server Management Studio


School Tables Script.sql - macwin8.SCHOOL (MACWIN8\Hans-Petter (53)) - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
SCHOOL
Execute
Debug
Object Explorer
Connect
School Tables Scrip...8\Hans-Petter (53))
SQLQuery1.sql - ma...8\Hans-Petter (52))
100 %
Ready
Ln 1
Col 1
Ch 1
INS

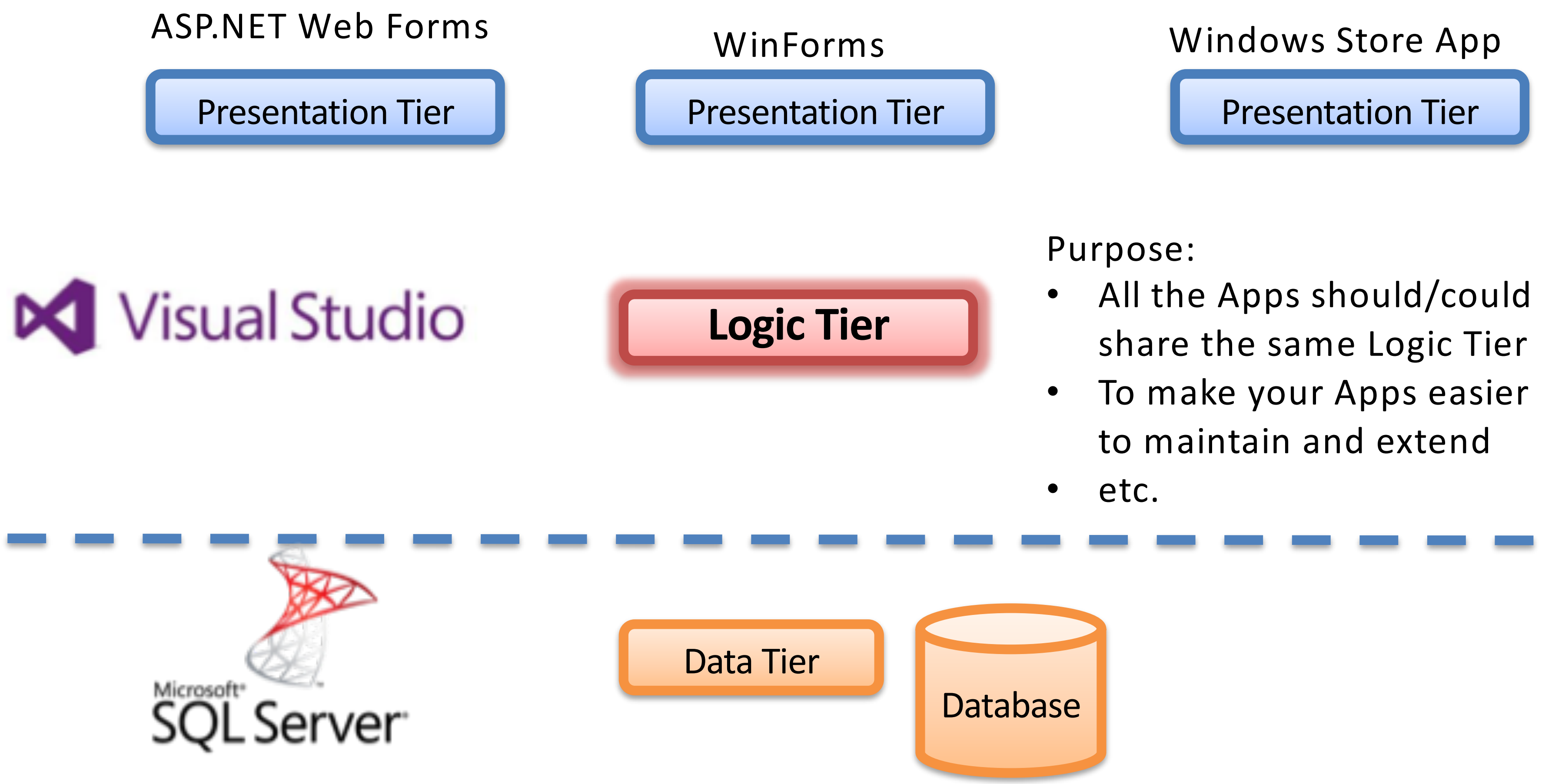
```
macwin8 (SQL Server 11.0.2100 - MACWIN8\Hans-Petter)
  Databases
    System Databases
    SCHOOL
      Database Diagrams
      Tables
        System Tables
        FileTables
        dbo.CLASS
        dbo.COURSE
        dbo.GRADE
          Columns
          Keys
          Constraints
          Triggers
            CalcAvgGrade
          Indexes
          Statistics
        dbo.SCHOOL
        dbo.STUDENT
        dbo.STUDENT_COURSE
        dbo.TEACHER
        dbo.TEACHER_COURSE
      Views
        System Views
        dbo.CourseData
        dbo.StudentData
      Synonyms
      Programmability
        Stored Procedures
          System Stored Procedures
          dbo.StudentGrade
            Parameters
              @StudentId (int, Input, No default)
              @CourseId (int, Input, No default)
              @Grade (int, Input, No default)
              Returns integer
        Functions
        Database Triggers
        Assemblies
        Types
        Rules
        Defaults
        Sequences
      Service Broker
```

```sql
/*------------------------------------------------------------------------------------------
Last Updated Date:  2011.04.01
Last Updated By:    hans.p.halvorsen@hit.no
Description:        Lab Work - School Tables

(C) Hans-Petter Halvorsen
------------------------------------------------------------------------------------------*/

/*
SCHOOL
CLASS
COURSE
TEACHER
TEACHER_COURSE
STUDENT
STUDENT_COURSE
GRADE
*/

if not exists (select * from dbo.sysobjects where id = object_id(N'[SCHOOL]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [SCHOOL]
(
    [SchoolId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    [SchoolName] [varchar](50) NOT NULL UNIQUE,
    [Description] [varchar](1000) NULL,
    [Address] [varchar](50) NULL,
    [Phone] [varchar](50) NULL,
    [PostCode] [varchar](50) NULL,
    [PostAddress] [varchar](50) NULL,

)
GO


if not exists (select * from dbo.sysobjects where id = object_id(N'[CLASS]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [CLASS]
(
    [ClassId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    [SchoolId] [int] NOT NULL FOREIGN KEY REFERENCES [SCHOOL] ([SchoolId]),
    [ClassName] [varchar](50) NOT NULL,
    [Description] [varchar](1000) NULL,

)
GO


if not exists (select * from dbo.sysobjects where id = object_id(N'[COURSE]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE [COURSE]
(
    [CourseId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
```

```
Connected. (1/1)    macwin8 (11.0 RTM)    MACWIN8\Hans-Petter (53)    SCHOOL    00:00:00    0 rows
```

You are finished with the Exercise

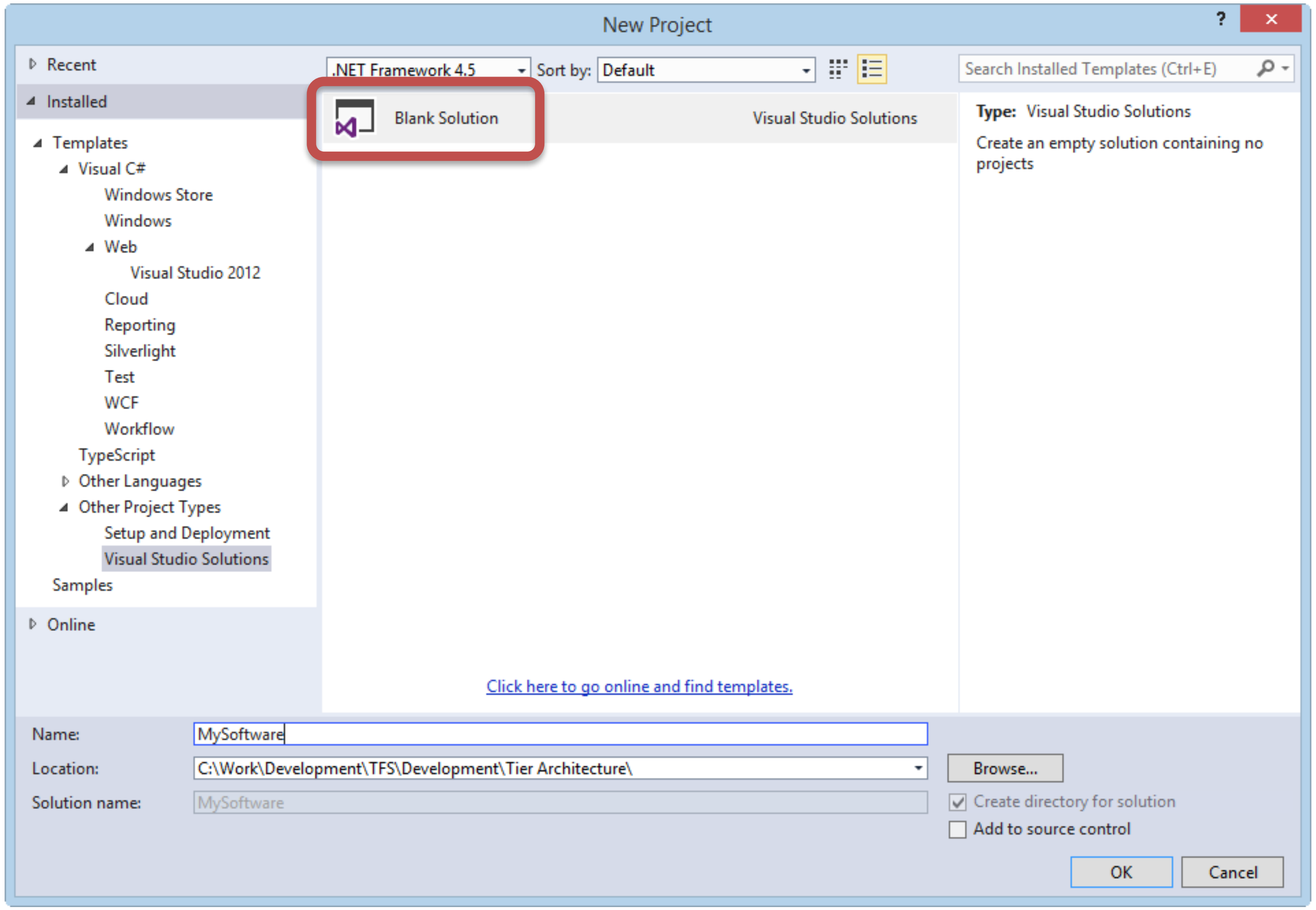
Logic Tier



Purpose:

- All the Apps should/could share the same Logic Tier
- To make your Apps easier to maintain and extend
- etc.

Create an Empty (Blank) **Solution** in Visual Studio



Type: Visual Studio Solutions

Create an empty solution containing no projects

Name: MySoftware

Location: C:\Work\Development\TFS\Development\Tier Architecture\

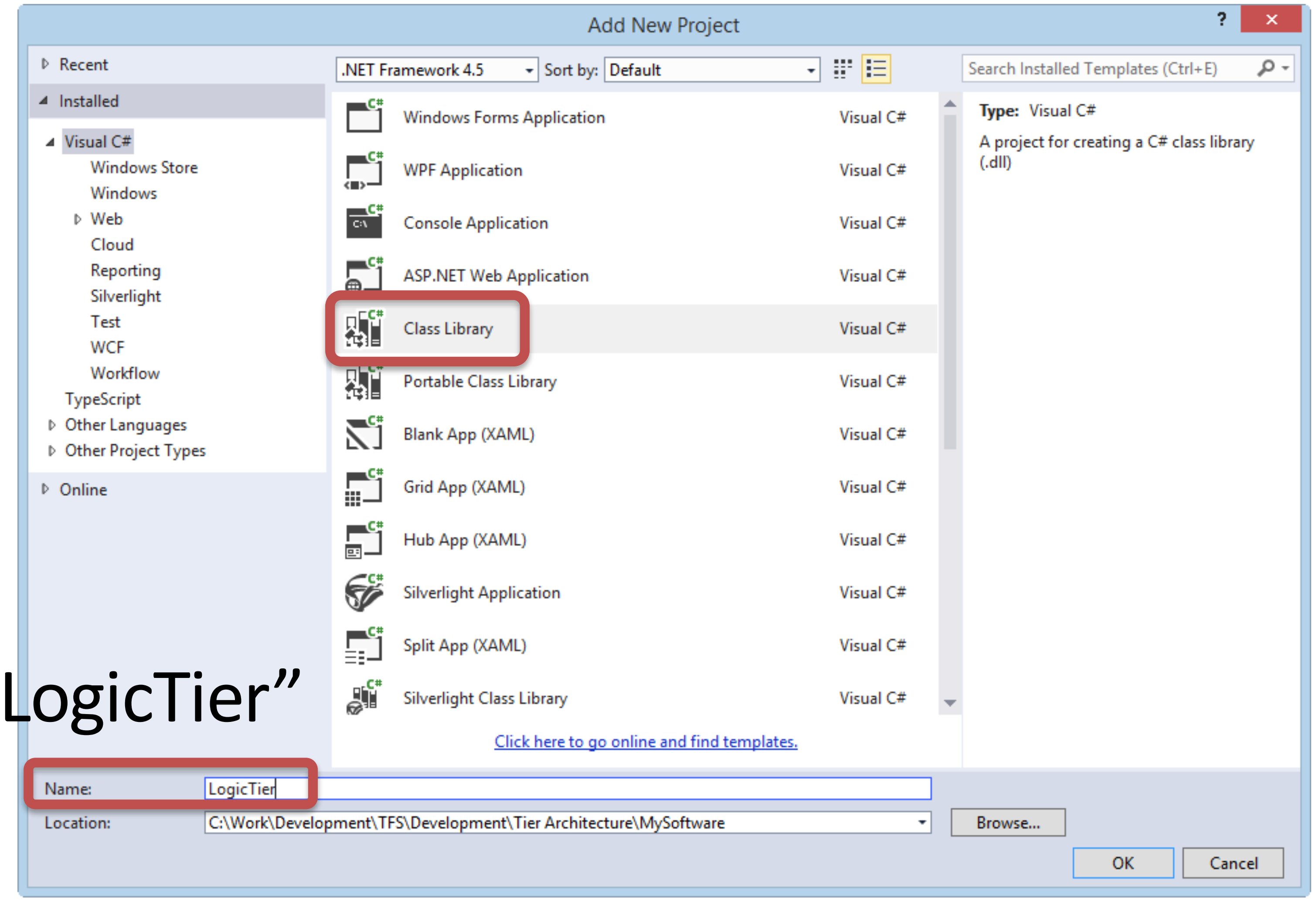
Solution name: MySoftware

[✓] Create directory for solution

[ ] Add to source control

Add **Project** for Logic Tier (Data Access)

Select a “**Class Library**” Project



Add New Project

- Recent
- Installed
  - Visual C#
    - Windows Store
    - Windows
    - Web
    - Cloud
    - Reporting
    - Silverlight
    - Test
    - WCF
    - Workflow
  - TypeScript
  - Other Languages
  - Other Project Types
- Online

.NET Framework 4.5 Sort by: Default

Search Installed Templates (Ctrl+E)

Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
ASP.NET Web Application	Visual C#
Class Library	Visual C#
Portable Class Library	Visual C#
Blank App (XAML)	Visual C#
Grid App (XAML)	Visual C#
Hub App (XAML)	Visual C#
Silverlight Application	Visual C#
Split App (XAML)	Visual C#
Silverlight Class Library	Visual C#

Click here to go online and find templates.

Type: Visual C#

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

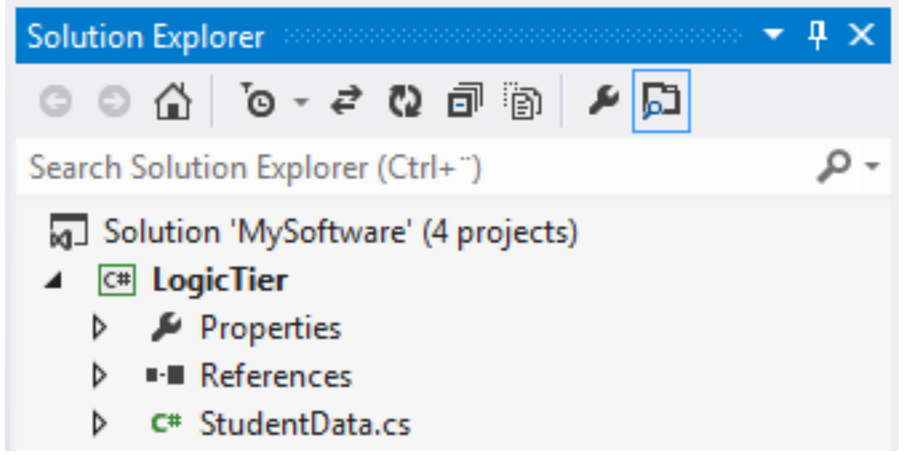
Name: LogicTier

Location: C:\Work\Development\TFS\Development\Tier Architecture\MySoftware

Browse... OK Cancel

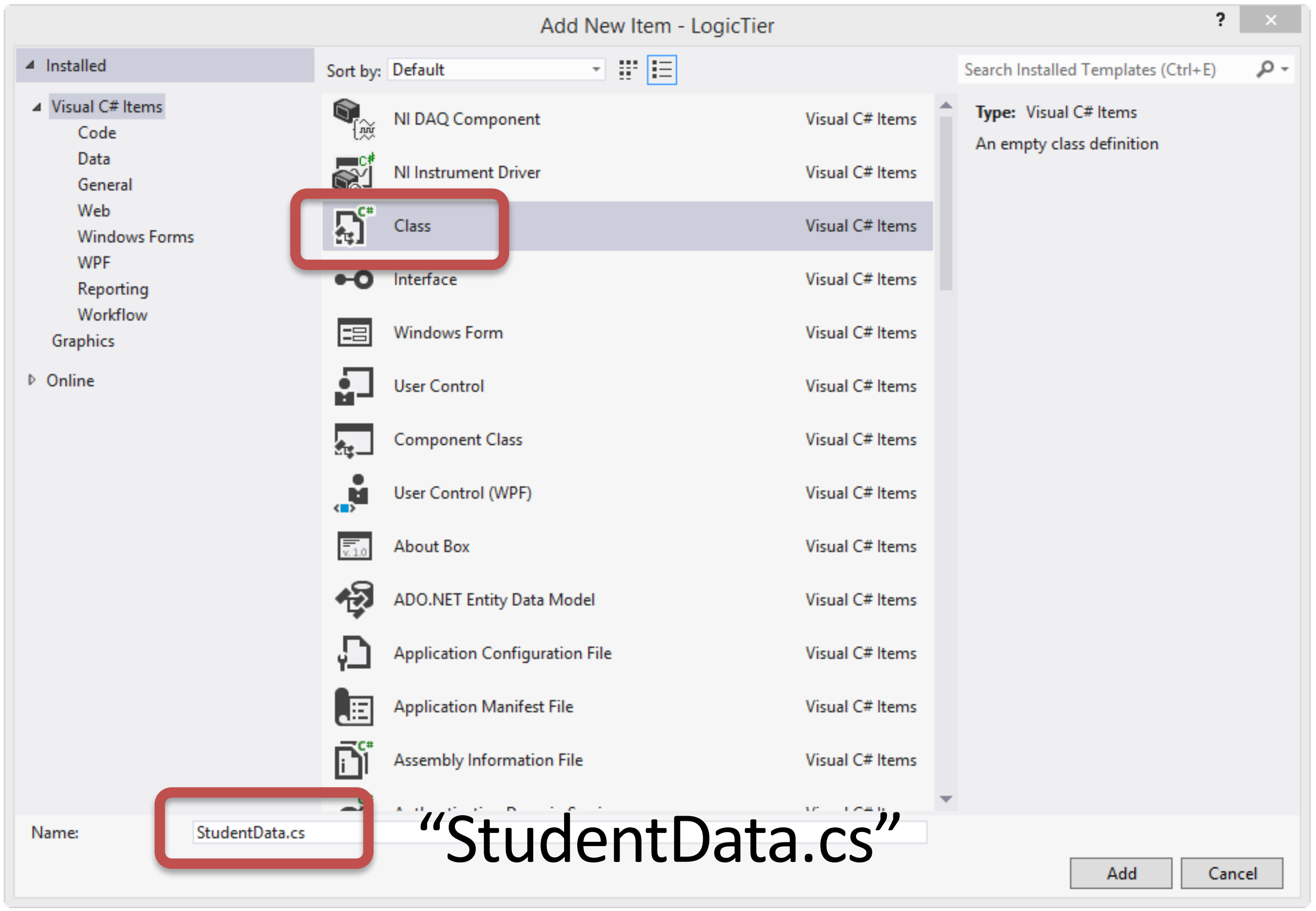
“LogicTier”

Add a New **Class** to the Project (“StudentData.cs”)


Solution Explorer

Search Solution Explorer (Ctrl+¨)

Solution 'MySoftware' (4 projects)
- LogicTier
  - Properties
  - References
  - StudentData.cs


Add New Item - LogicTier

Sort by:	Default

Search Installed Templates (Ctrl+E)

- Installed
  - Visual C# Items
    - Code
    - Data
    - General
    - Web
    - Windows Forms
    - WPF
    - Reporting
    - Workflow
  - Graphics
- Online

NI DAQ Component	Visual C# Items
NI Instrument Driver	Visual C# Items
Class	Visual C# Items
Interface	Visual C# Items
Windows Form	Visual C# Items
User Control	Visual C# Items
Component Class	Visual C# Items
User Control (WPF)	Visual C# Items
About Box	Visual C# Items
ADO.NET Entity Data Model	Visual C# Items
Application Configuration File	Visual C# Items
Application Manifest File	Visual C# Items
Assembly Information File	Visual C# Items

Type: Visual C# Items

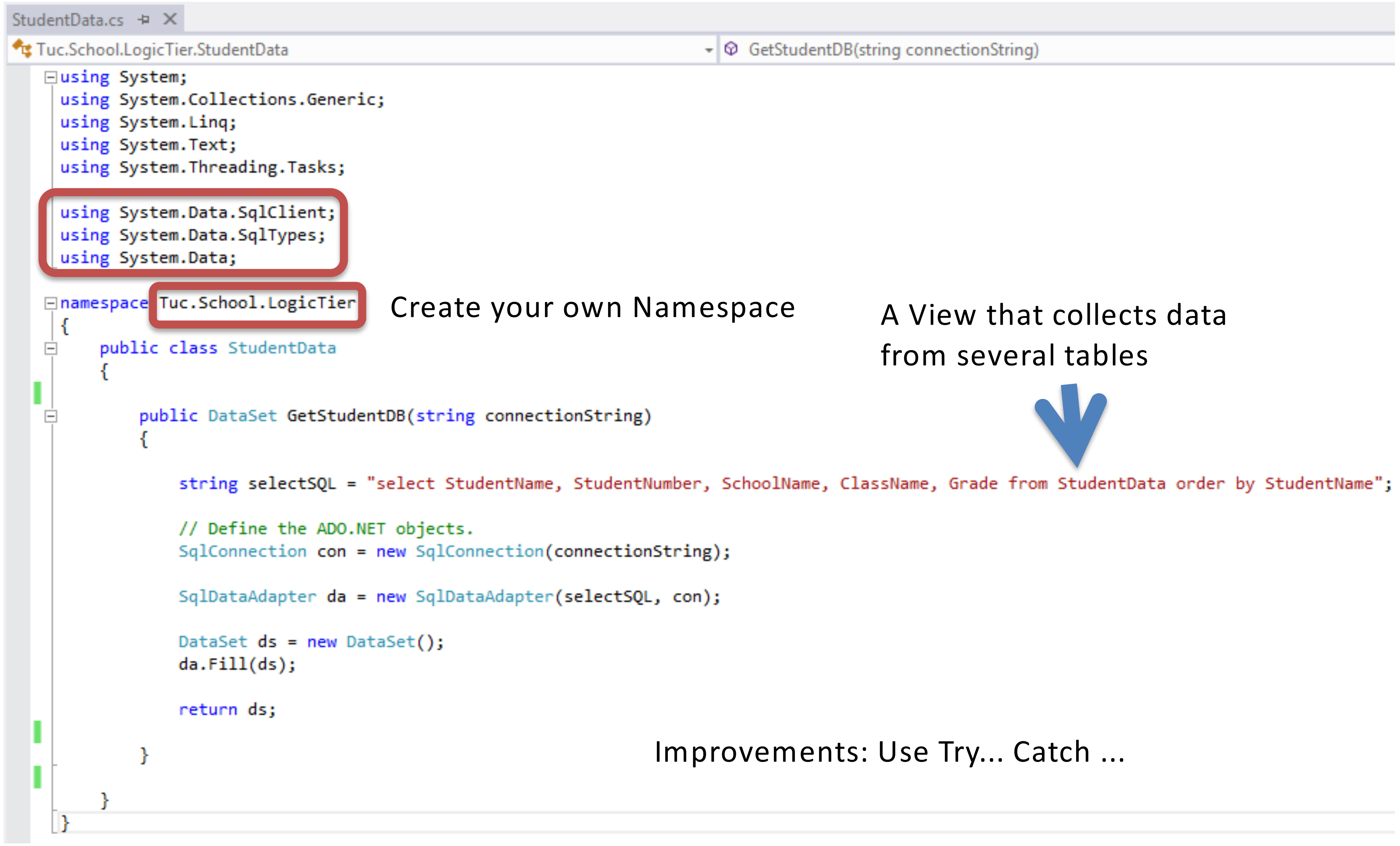
An empty class definition

Name: StudentData.cs

Add Cancel

“StudentData.cs”

Create the **Code**, e.g., like this ("StudentData.cs"):


StudentData.cs
Tuc.School.LogicTier.StudentData
GetStudentDB(string connectionString)

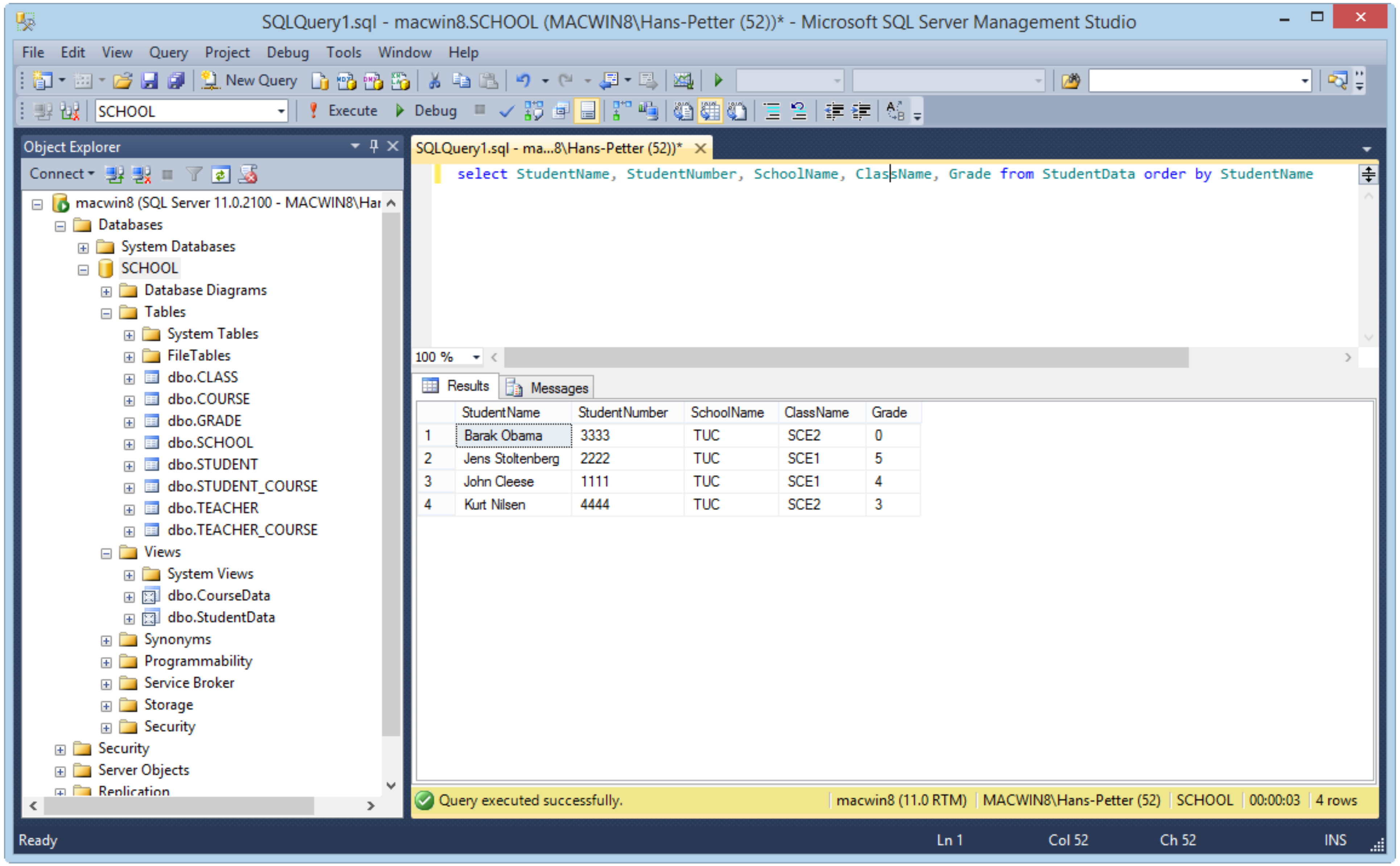
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Data;

namespace Tuc.School.LogicTier
{
    public class StudentData
    {

        public DataSet GetStudentDB(string connectionString)
        {

            string selectSQL = "select StudentName, StudentNumber, SchoolName, ClassName, Grade from StudentData order by StudentName";

            // Define the ADO.NET objects.
            SqlConnection con = new SqlConnection(connectionString);

            SqlDataAdapter da = new SqlDataAdapter(selectSQL, con);

            DataSet ds = new DataSet();
            da.Fill(ds);

            return ds;

        }

    }
}
```

Create your own Namespace

A View that collects data from several tables

Improvements: Use Try... Catch ...

You should test the SQL Query in the **SQL Server Management Studio** first



```
select StudentName, StudentNumber, SchoolName, ClassName, Grade from StudentData order by StudentName
```

	StudentName	StudentNumber	SchoolName	ClassName	Grade
1	Barak Obama	3333	TUC	SCE2	0
2	Jens Stoltenberg	2222	TUC	SCE1	5
3	John Cleese	1111	TUC	SCE1	4
4	Kurt Nilsen	4444	TUC	SCE2	3

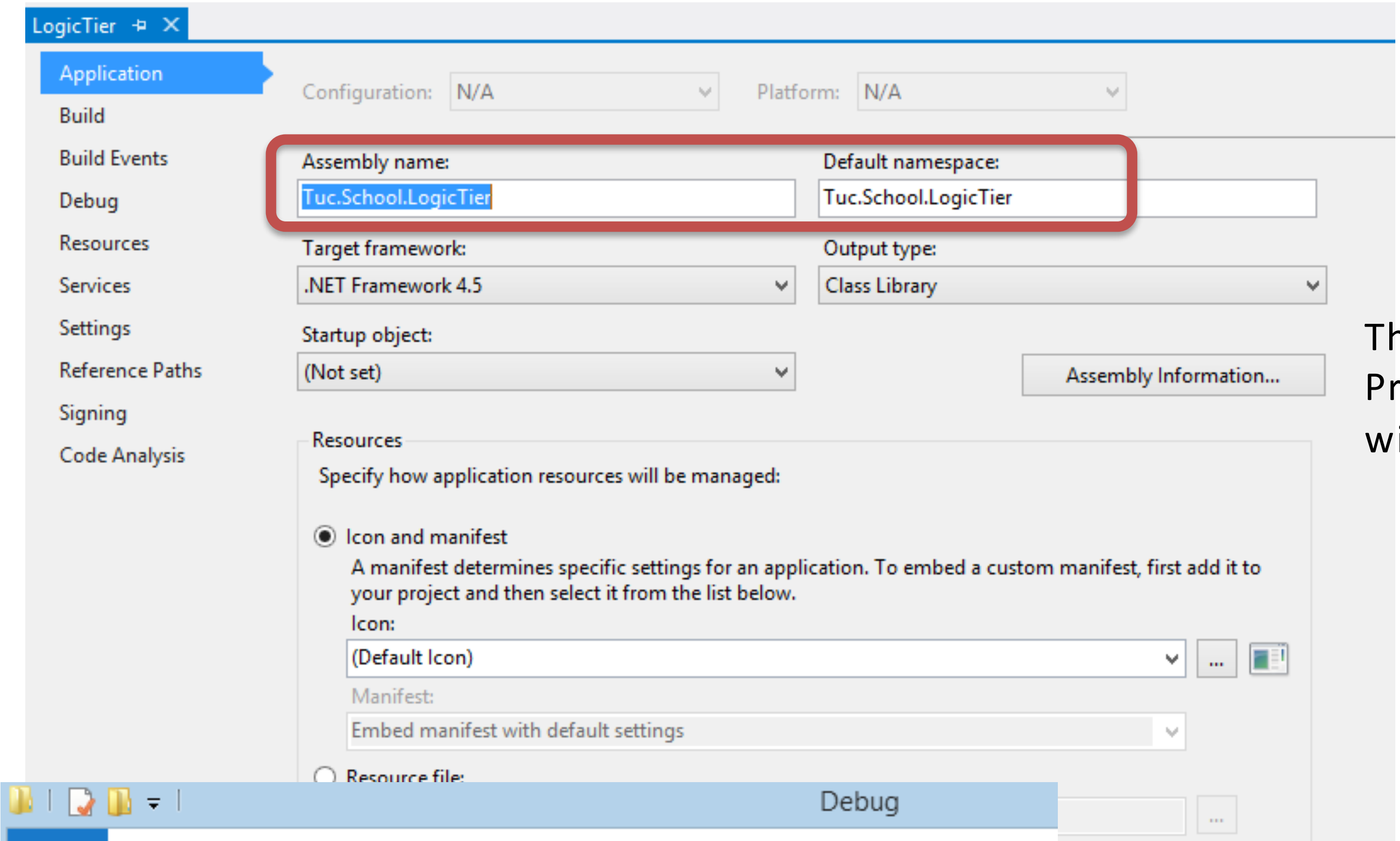
Query executed successfully. | macwin8 (11.0 RTM) | MACWIN8\Hans-Petter (52) | SCHOOL | 00:00:03 | 4 rows

Code (“StudentData.cs”):

```
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Data;

namespace Tuc.School.LogicTier
{
    public class StudentData
    {
        public DataSet GetStudentDB(string connectionString)
        {
            string selectSQL = "select StudentName, StudentNumber, SchoolName, 
ClassName,                                   Grade from StudentData order by StudentName";

            // Define the ADO.NET objects.
            SqlConnection con = new SqlConnection(connectionString);

            SqlDataAdapter da = new SqlDataAdapter(selectSQL, con);

            DataSet ds = new DataSet();
            da.Fill(ds);

            return ds;

        }
    }
}
```

Create a proper name for the **Assembly** (.dll File)

Right-click on the Project in the Solution Explorer and select Properties



Configuration: N/A Platform: N/A

Assembly name: Tuc.School.LogicTier

Default namespace: Tuc.School.LogicTier

Target framework: .NET Framework 4.5

Output type: Class Library

Startup object: (Not set)

Resources

Specify how application resources will be managed:

● Icon and manifest

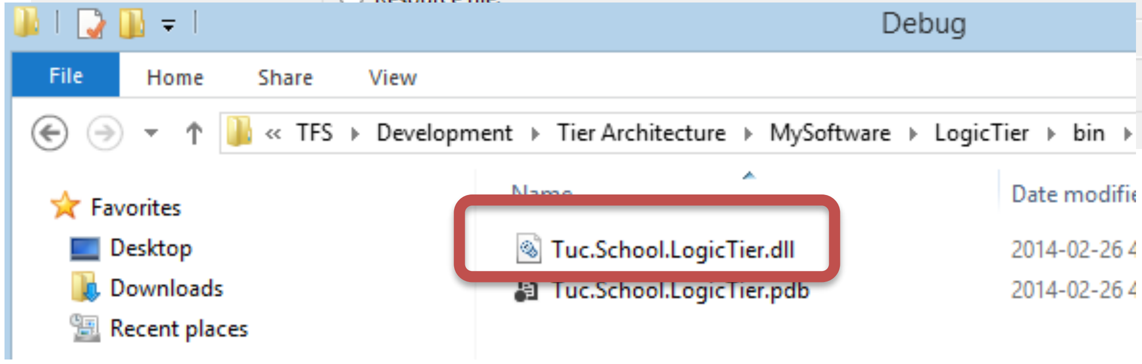
A manifest determines specific settings for an application. To embed a custom manifest, first add it to your project and then select it from the list below.

Icon: (Default Icon)

Manifest: Embed manifest with default settings

○ Resource file:

Then Build your Project (hopefully with no errors)



« TFS ▸ Development ▸ Tier Architecture ▸ MySoftware ▸ LogicTier ▸ bin ▸

Tuc.School.LogicTier.dll 2014-02-26

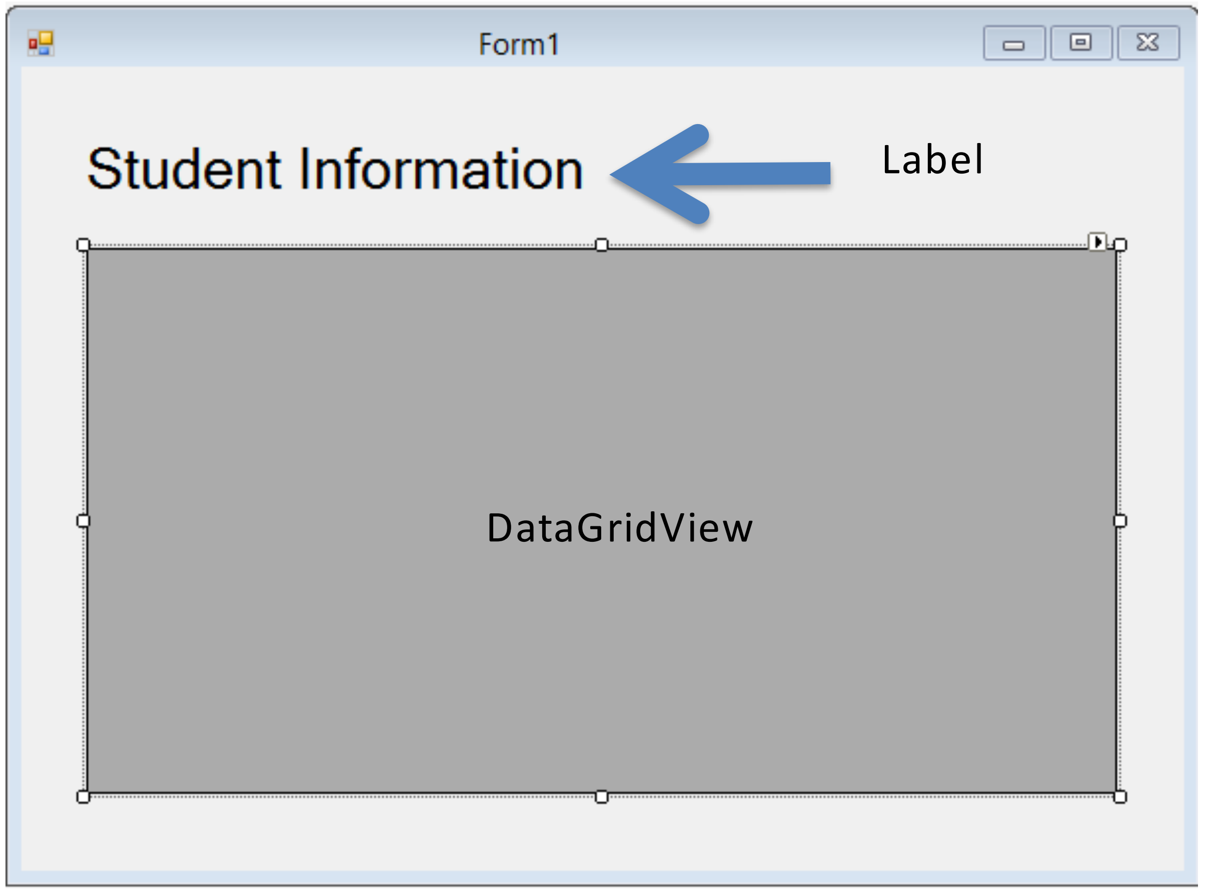
Tuc.School.LogicTier.pdb 2014-02-26

This will be the Assembly for your Logic Tier, that can be imported and used in other projects.
Create once – use it many times!!

You are finished with the Exercise

Presentation Layer Desktop App: WinForms

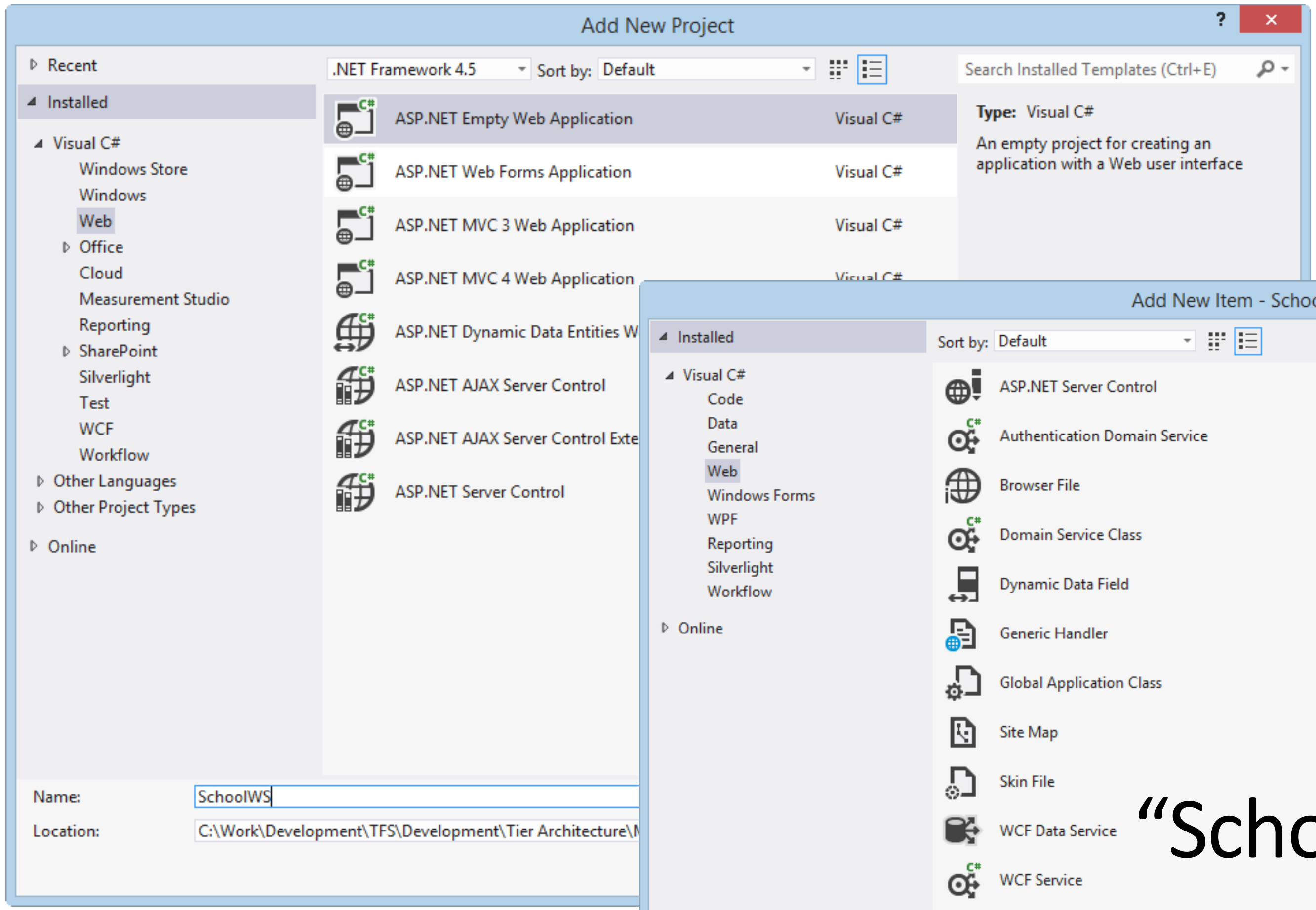
Using **Web Services** (we assume the The App should be used on Internet outside the Firewall)



Step 1: Create Web Service

"SchoolWS"

Create an ASP.NET Project:



Add New Project

Recent
Installed
Visual C#
Windows Store
Windows
Web
Office
Cloud
Measurement Studio
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Other Languages
Other Project Types
Online

.NET Framework 4.5 | Sort by: Default | Search Installed Templates (Ctrl+E)

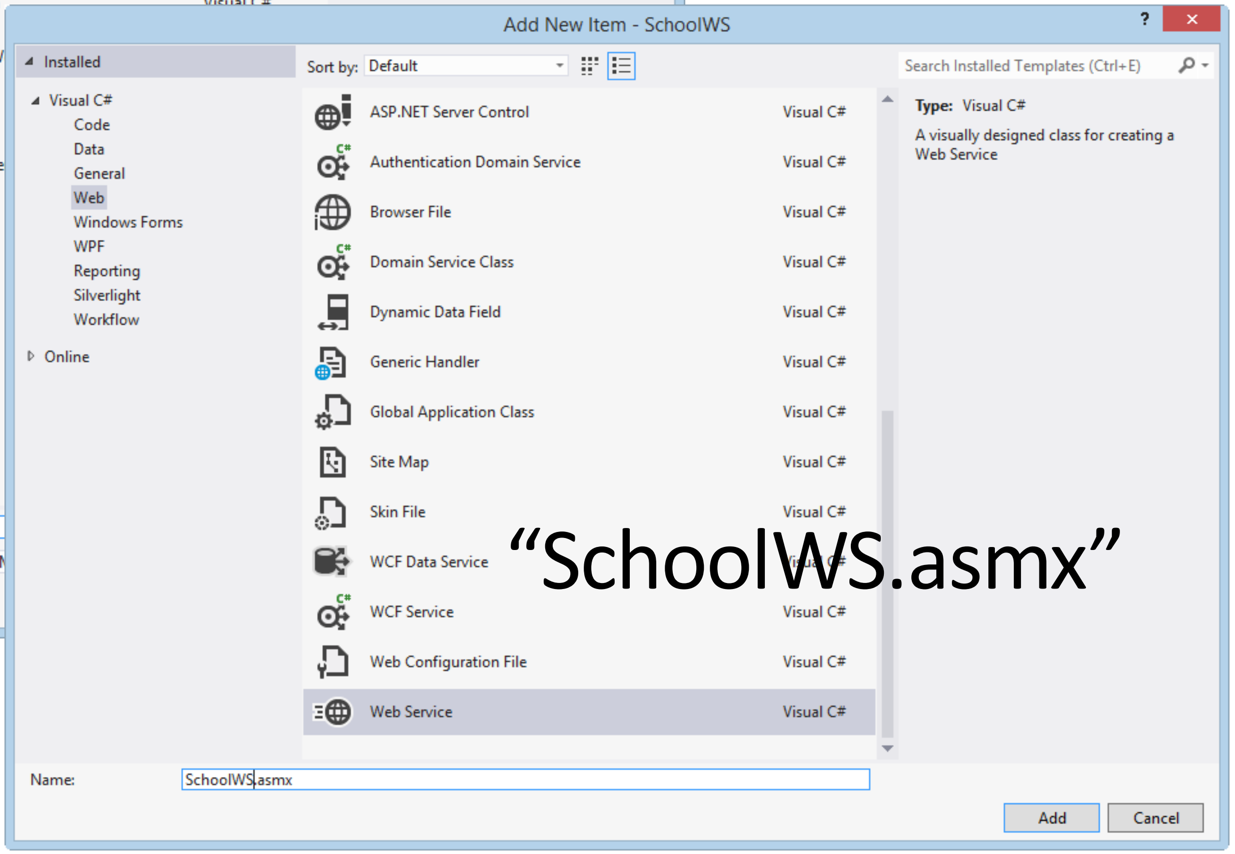
ASP.NET Empty Web Application	Visual C#
ASP.NET Web Forms Application	Visual C#
ASP.NET MVC 3 Web Application	Visual C#
ASP.NET MVC 4 Web Application	Visual C#
ASP.NET Dynamic Data Entities W	
ASP.NET AJAX Server Control	
ASP.NET AJAX Server Control Exte	
ASP.NET Server Control	

Type: Visual C#

An empty project for creating an application with a Web user interface

Name: SchoolWS

Location: C:\Work\Development\TFS\Development\Tier Architecture\N



Add New Item - SchoolWS

Installed
Visual C#
Code
Data
General
Web
Windows Forms
WPF
Reporting
Silverlight
Workflow
Online

Sort by: Default | Search Installed Templates (Ctrl+E)

ASP.NET Server Control	Visual C#
Authentication Domain Service	Visual C#
Browser File	Visual C#
Domain Service Class	Visual C#
Dynamic Data Field	Visual C#
Generic Handler	Visual C#
Global Application Class	Visual C#
Site Map	Visual C#
Skin File	Visual C#
WCF Data Service	Visual C#
WCF Service	Visual C#
Web Configuration File	Visual C#
Web Service	Visual C#

Type: Visual C#

A visually designed class for creating a Web Service

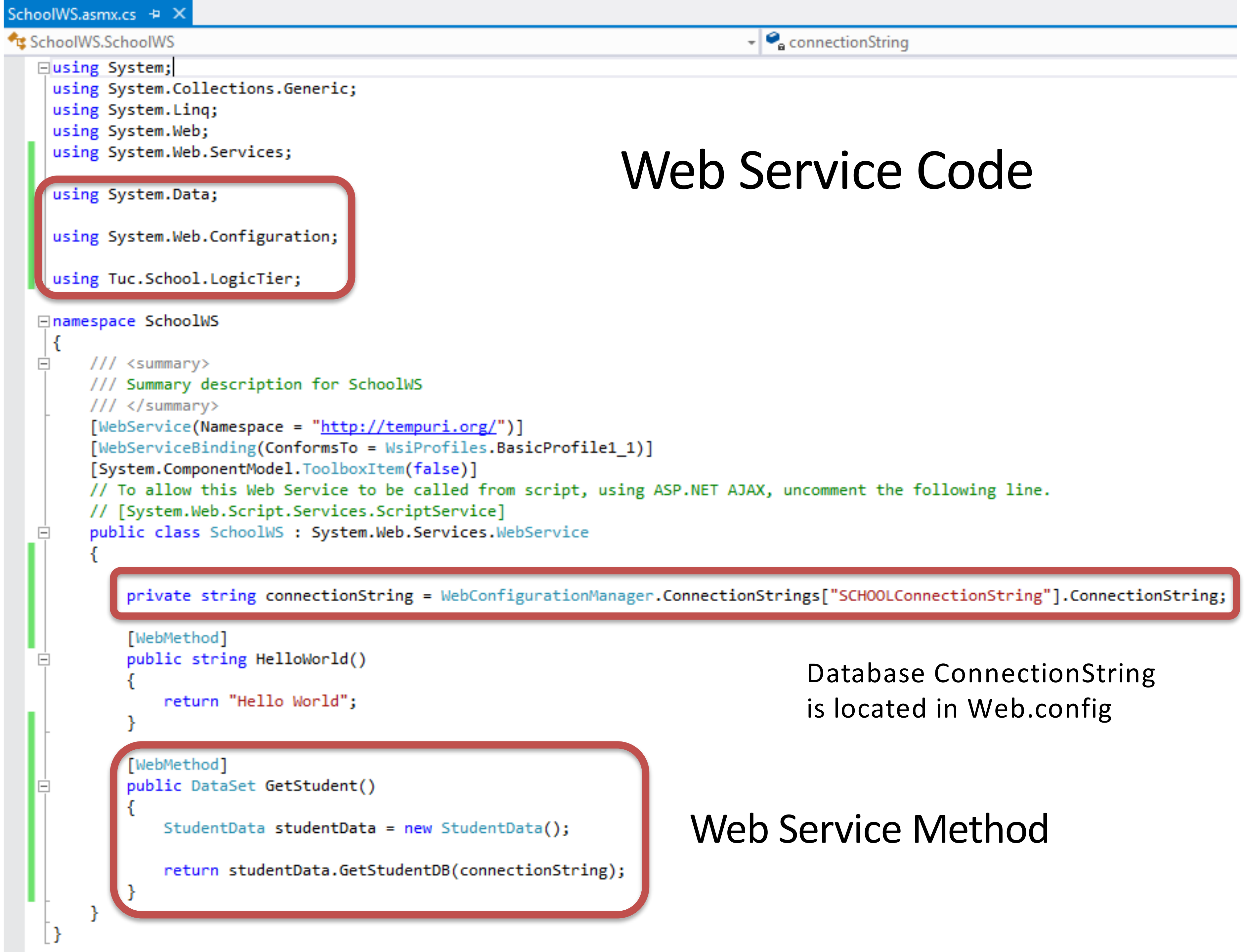
Name: SchoolWS.asmx

Add | Cancel

"SchoolWS.asmx"

Add Web Service:

Web Service Code

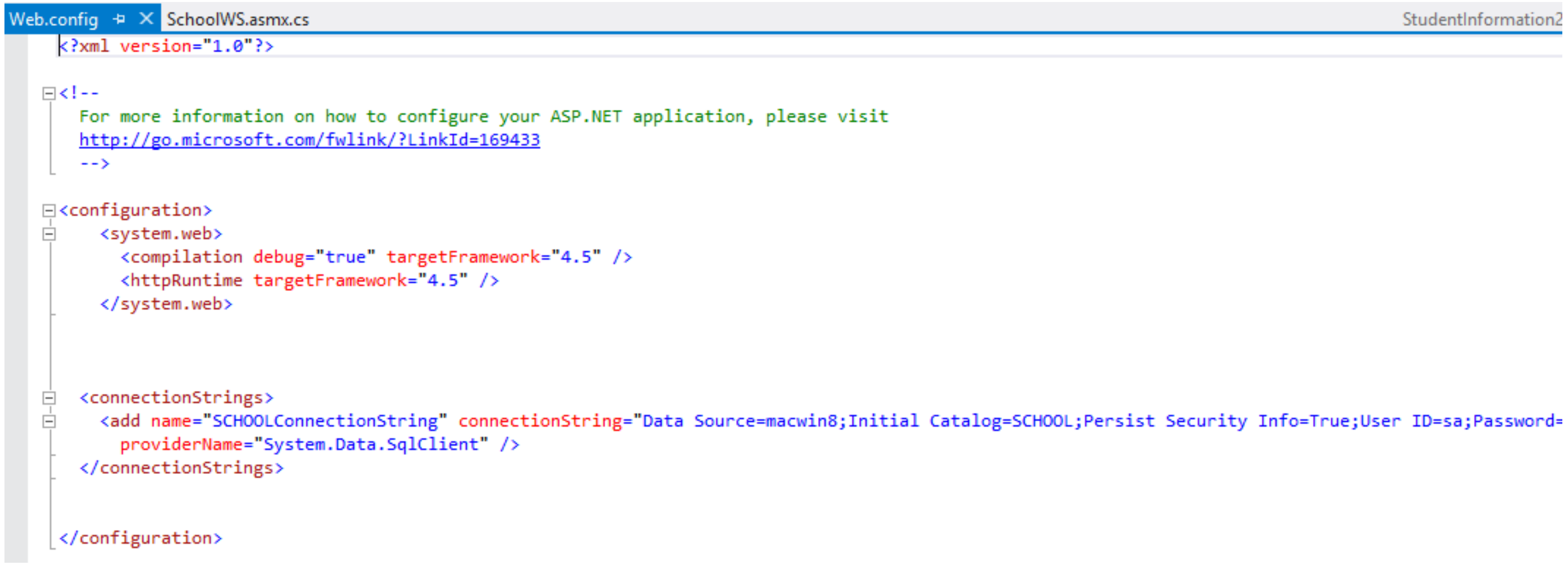


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

using System.Data;

using System.Web.Configuration;

using Tuc.School.LogicTier;

namespace SchoolWS
{
    /// <summary>
    /// Summary description for SchoolWS
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class SchoolWS : System.Web.Services.WebService
    {

        private string connectionString = WebConfigurationManager.ConnectionStrings["SCHOOLConnectionString"].ConnectionString;

        [WebMethod]
        public string HelloWorld()
        {
            return "Hello World";
        }

        [WebMethod]
        public DataSet GetStudent()
        {
            StudentData studentData = new StudentData();

            return studentData.GetStudentDB(connectionString);
        }
    }
}
```

Database ConnectionString is located in Web.config

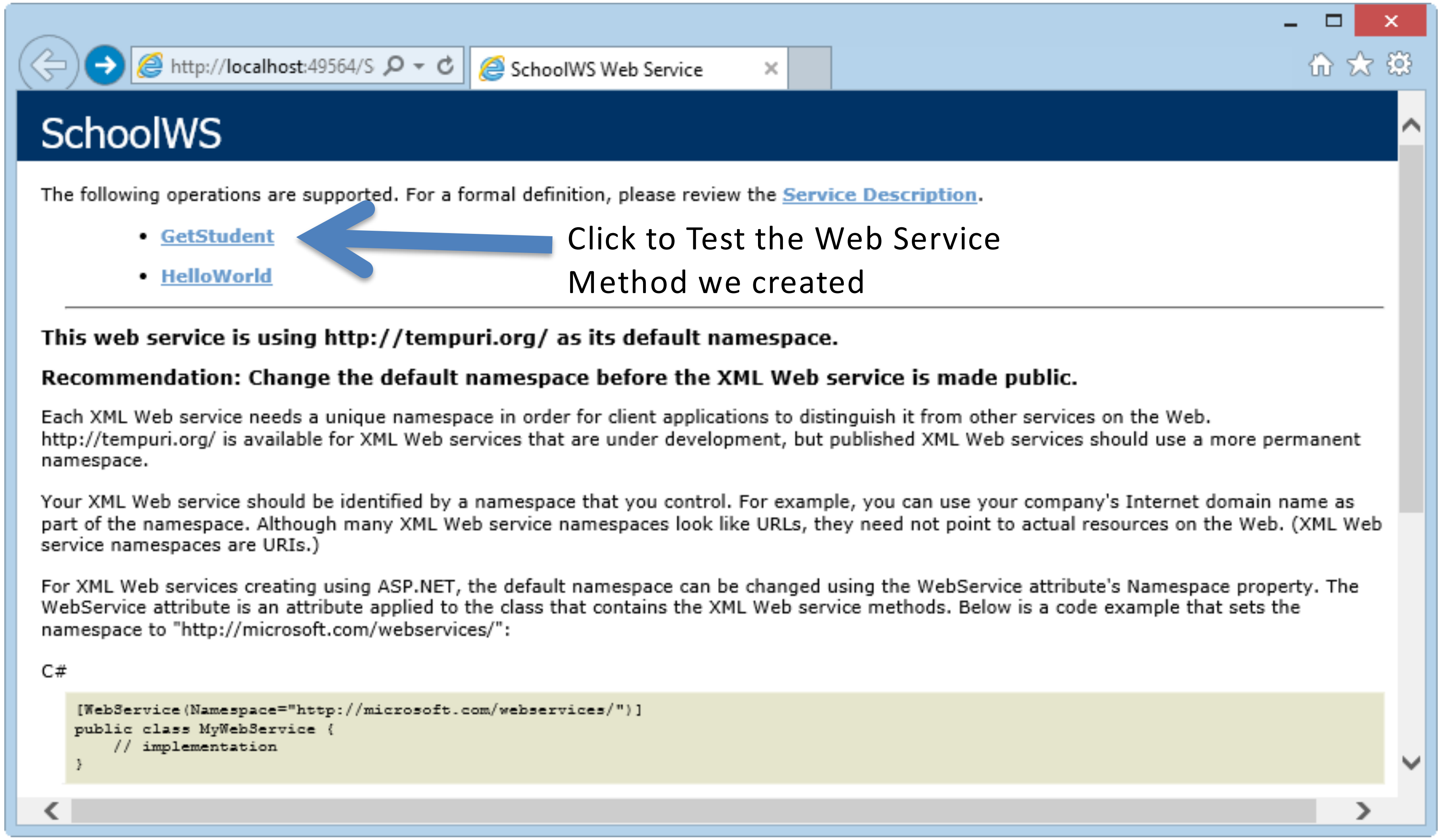
Web Service Method

Database ConnectionString is located in **Web.config**



```xml
<?xml version="1.0"?>

<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->

<configuration>
    <system.web>
      <compilation debug="true" targetFramework="4.5" />
      <httpRuntime targetFramework="4.5" />
    </system.web>



  <connectionStrings>
    <add name="SCHOOLConnectionString" connectionString="Data Source=macwin8;Initial Catalog=SCHOOL;Persist Security Info=True;User ID=sa;Password=
      providerName="System.Data.SqlClient" />
  </connectionStrings>


</configuration>
```

Test Web Service



SchoolWS

The following operations are supported. For a formal definition, please review the Service Description.

- GetStudent
- HelloWorld

Click to Test the Web Service Method we created

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

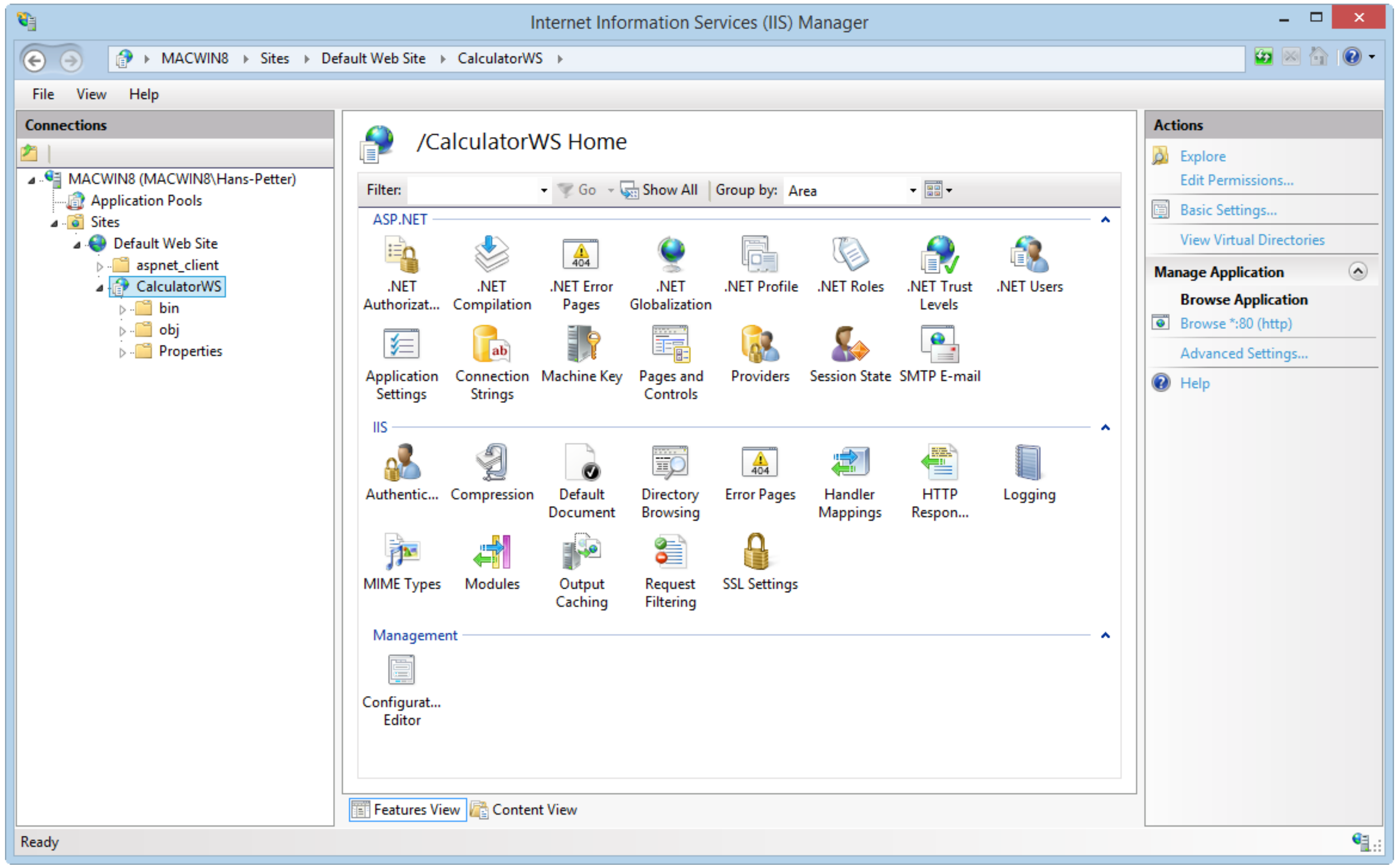
C#

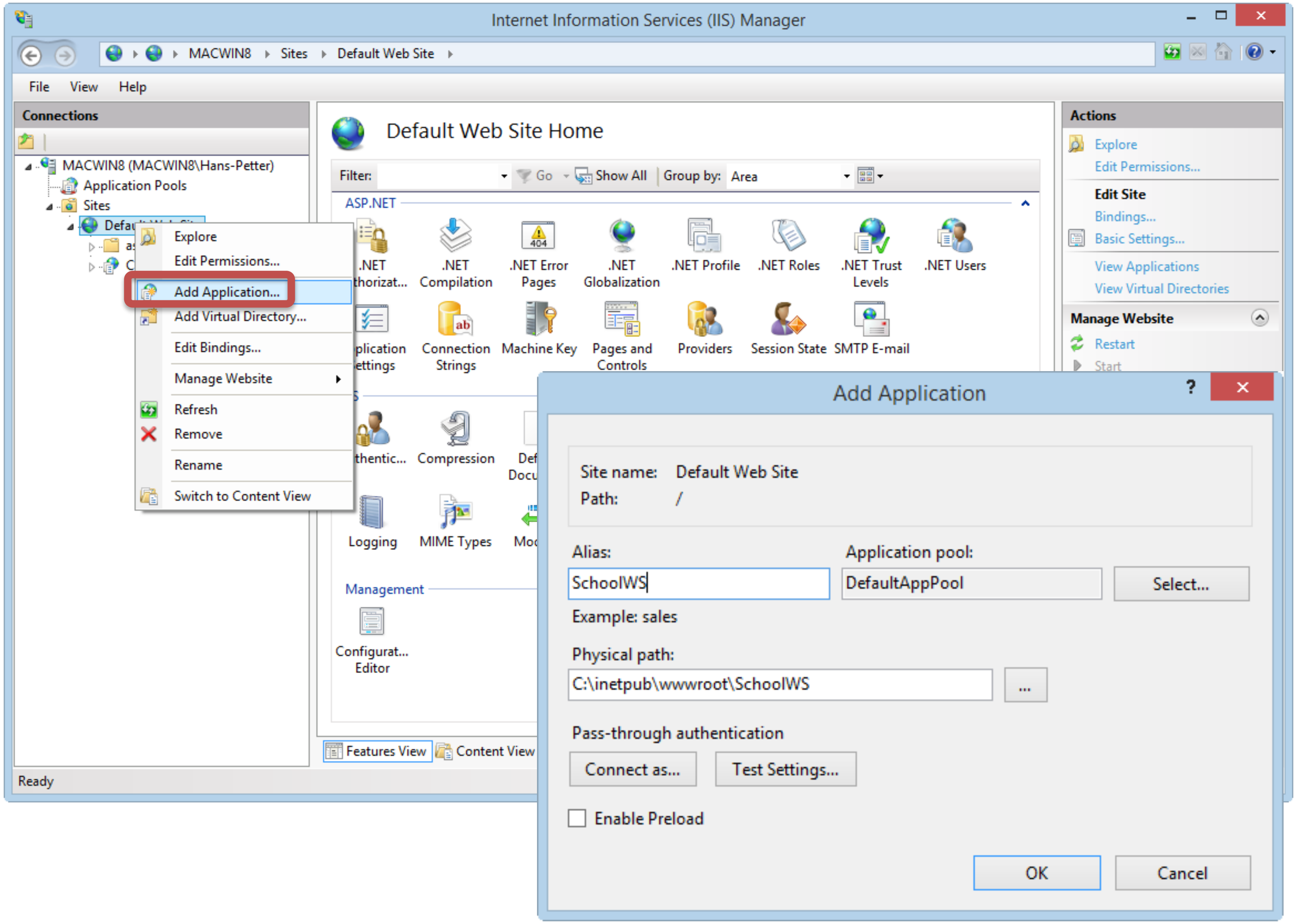
```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

It Works!!

Deploy/Publish Web Service to **IIS**

Copy Web Service Files (Project) to default IIS Directory: C:\inetpub\wwwroot





Add Application

Site name: Default Web Site
Path: /

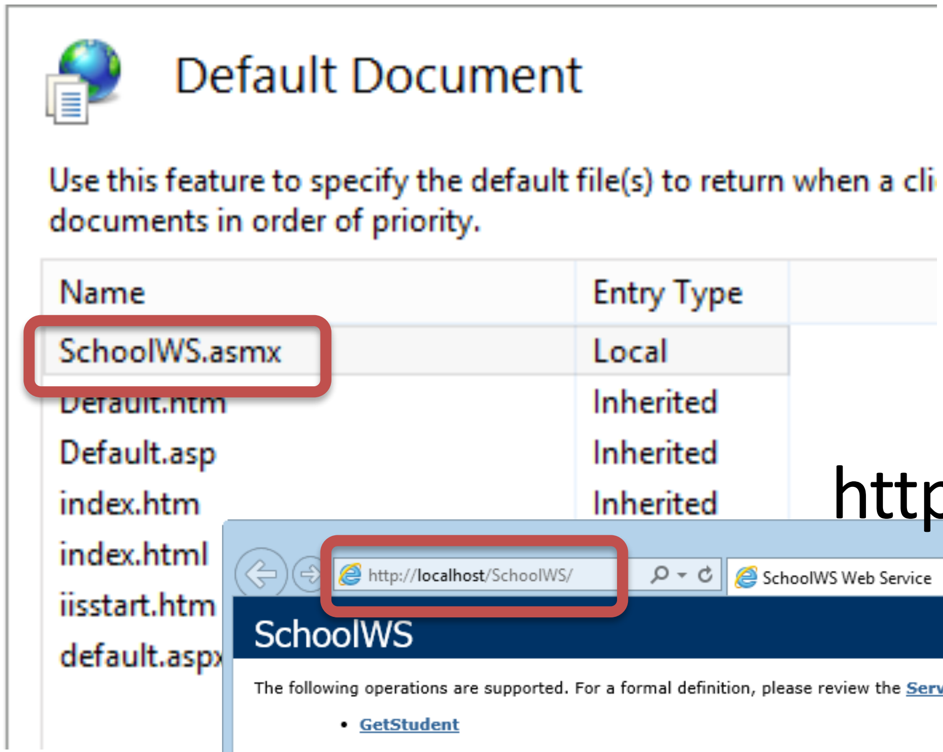
Alias: SchoolWS
Application pool: DefaultAppPool
Select...
Example: sales

Physical path: C:\inetpub\wwwroot\SchoolWS

Pass-through authentication
Connect as... Test Settings...

Enable Preload

OK Cancel



Default Document

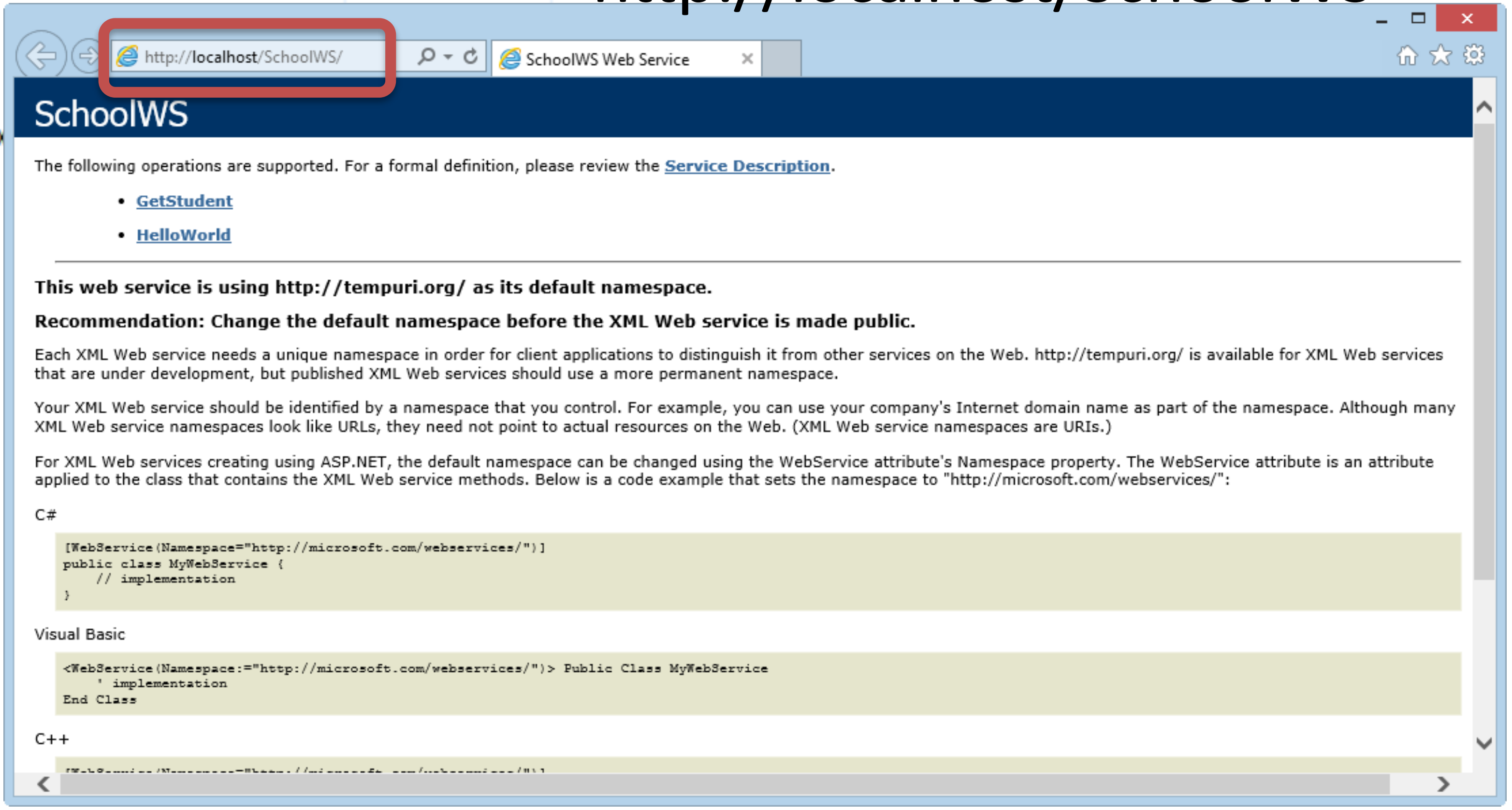
Use this feature to specify the default file(s) to return when a cli
documents in order of priority.

Name	Entry Type
SchoolWS.asmx	Local
Default.htm	Inherited
Default.asp	Inherited
index.htm	Inherited
index.html	
iisstart.htm	
default.aspx	

Default Document

Test if WS working:

http://localhost/SchoolWS



http://localhost/SchoolWS/

SchoolWS Web Service

SchoolWS

The following operations are supported. For a formal definition, please review the Service Description.

- GetStudent
- HelloWorld

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

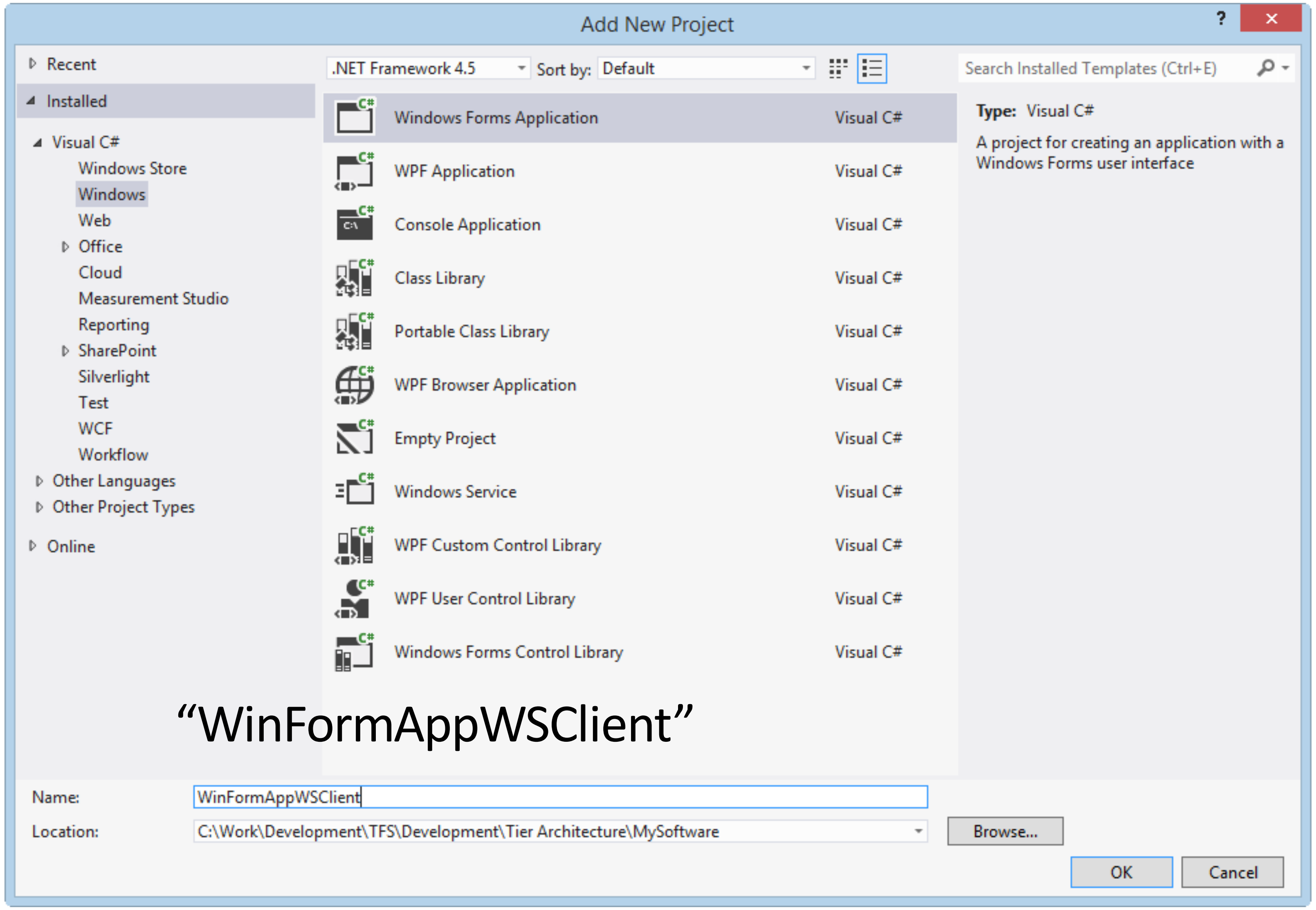
Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

Step 2: Use Web Service in WinForm

Create New WinForm Project:



Add New Project

Recent
Installed
Visual C#
Windows Store
Windows
Web
Office
Cloud
Measurement Studio
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Other Languages
Other Project Types
Online

.NET Framework 4.5 Sort by: Default

Search Installed Templates (Ctrl+E)

Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
Class Library	Visual C#
Portable Class Library	Visual C#
WPF Browser Application	Visual C#
Empty Project	Visual C#
Windows Service	Visual C#
WPF Custom Control Library	Visual C#
WPF User Control Library	Visual C#
Windows Forms Control Library	Visual C#

Type: Visual C#

A project for creating an application with a Windows Forms user interface

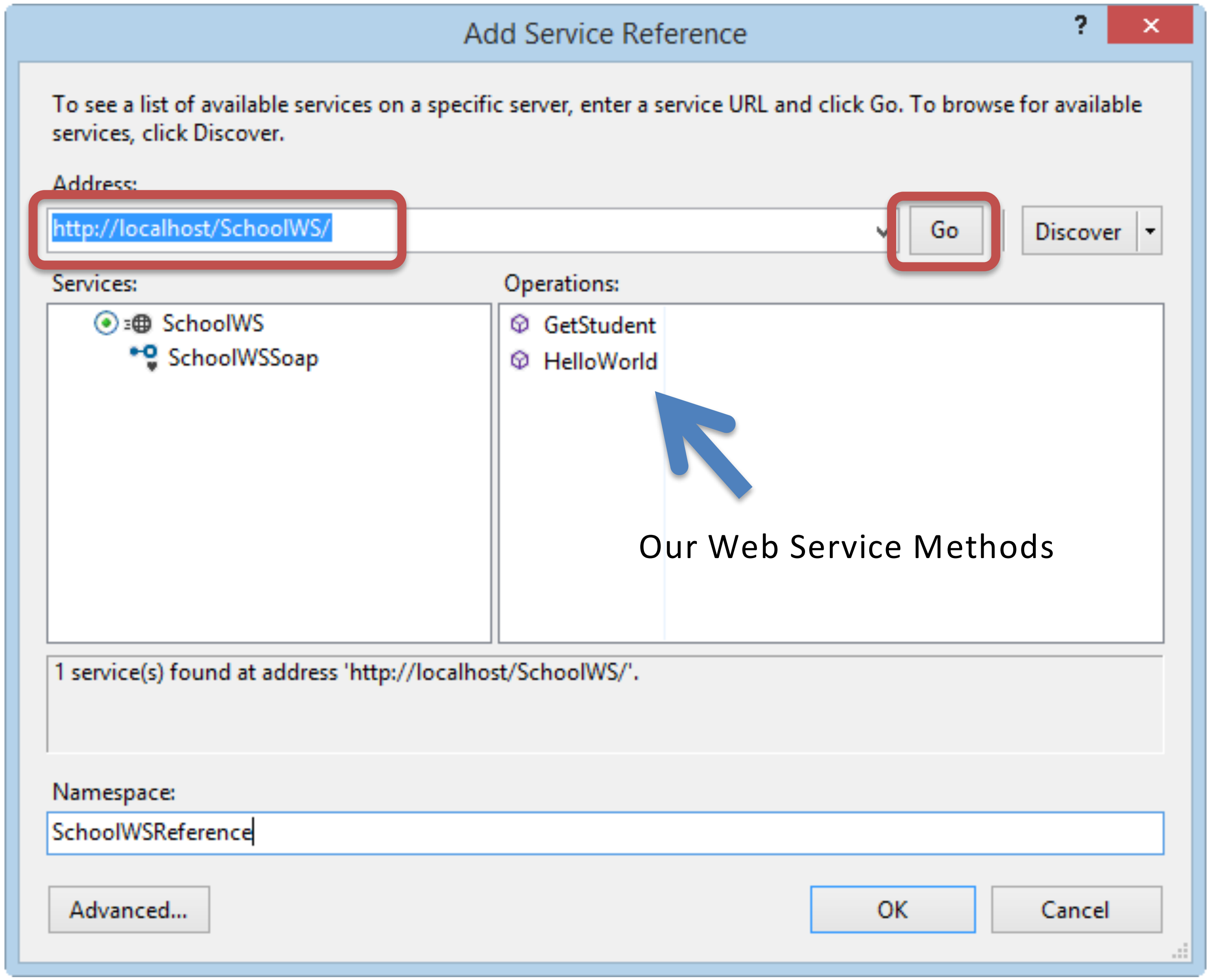
"WinFormAppWSClient"

Name: WinFormAppWSClient

Location: C:\Work\Development\TFS\Development\Tier Architecture\MySoftware

Browse... OK Cancel

Add Web Service Reference


Add Service Reference

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address:

http://localhost/SchoolWS/

Services:

- SchoolWS
  - SchoolWSSoap

Operations:

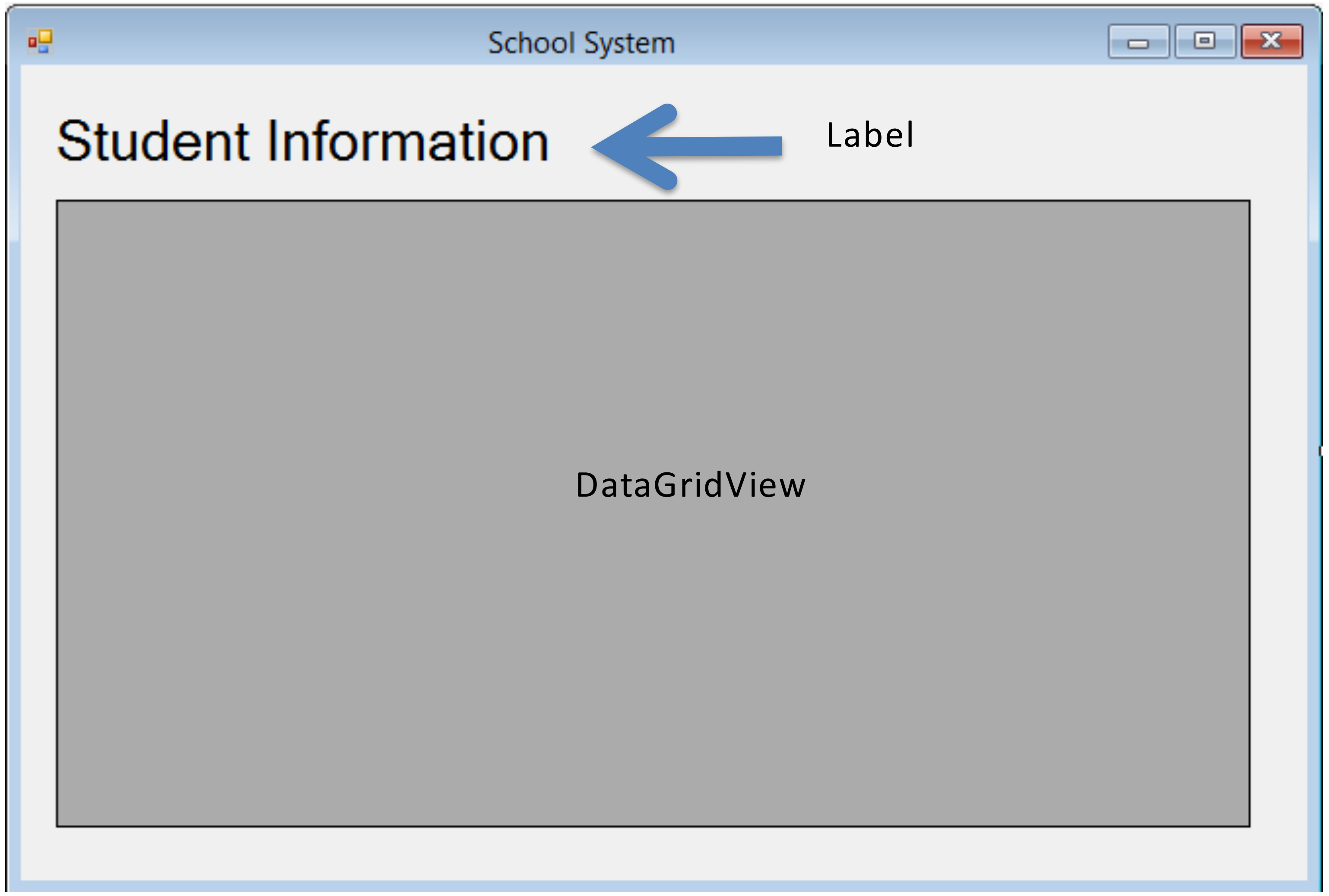
- GetStudent
- HelloWorld

1 service(s) found at address 'http://localhost/SchoolWS/'.

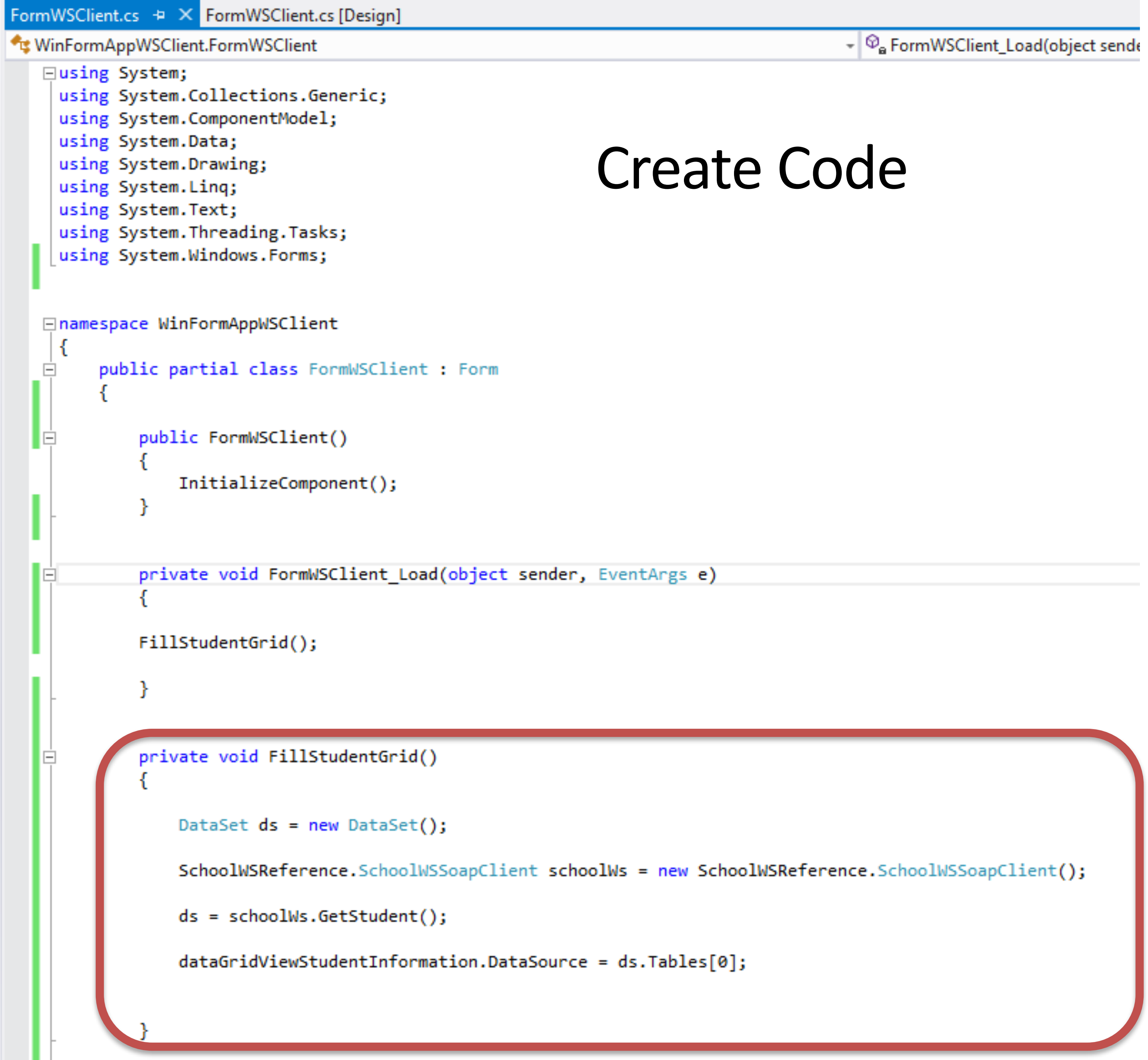
Namespace:

SchoolWSReference

Create **GUI**



Create Code


FormWSClient.cs
FormWSClient.cs [Design]
WinFormAppWSClient.FormWSClient
FormWSClient_Load(object sende

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormAppWSClient
{
    public partial class FormWSClient : Form
    {

        public FormWSClient()
        {
            InitializeComponent();
        }


        private void FormWSClient_Load(object sender, EventArgs e)
        {

        FillStudentGrid();

        }


        private void FillStudentGrid()
        {

            DataSet ds = new DataSet();

            SchoolWSReference.SchoolWSSoapClient schoolWs = new SchoolWSReference.SchoolWSSoapClient();

            ds = schoolWs.GetStudent();

            dataGridViewStudentInformation.DataSource = ds.Tables[0];


        }
```

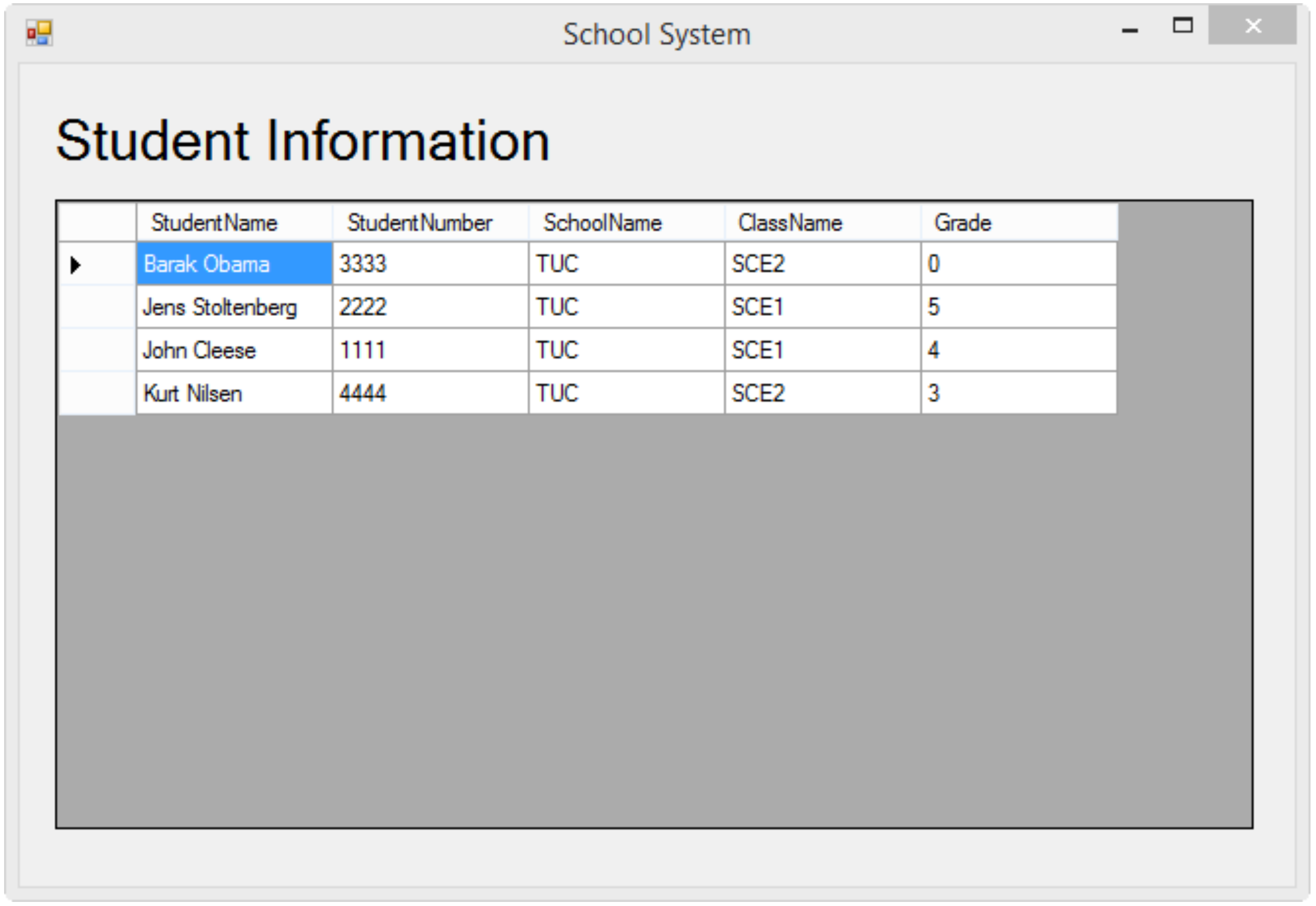
WinForm Code

```
using System.Windows.Forms;


namespace WinFormAppWSClient
{
    public partial class FormWSClient : Form
    {

        public FormWSClient()
        {
            InitializeComponent();
        }

        private void FormWSClient_Load(object sender, EventArgs e)
        {

        FillStudentGrid();
        }

        private void FillStudentGrid()
        {

            DataSet ds = new DataSet();

            SchoolWSReference.SchoolWSSoapClient schoolWs = new
SchoolWSReference.SchoolWSSoapClient();

            ds = schoolWs.GetStudent();

            dataGridViewStudentInformation.DataSource = ds.Tables[0];

        }
}
```

Call the Web Service method

Fill GridView

Test it:


School System

Student Information

	StudentName	StudentNumber	SchoolName	ClassName	Grade
▶	Barak Obama	3333	TUC	SCE2	0
	Jens Stoltenberg	2222	TUC	SCE1	5
	John Cleese	1111	TUC	SCE1	4
	Kurt Nilsen	4444	TUC	SCE2	3

It works!!!

You are finished with the Exercise

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog