


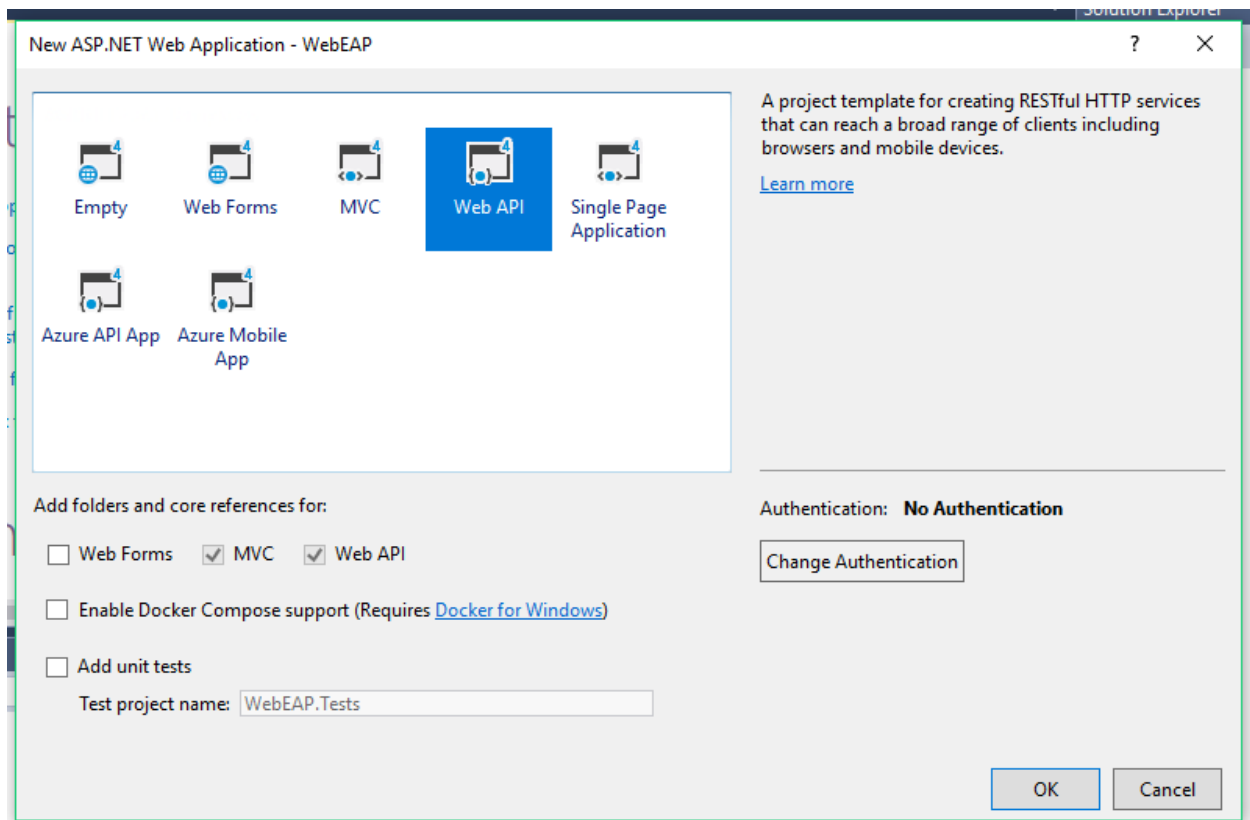
Reviews EAPs

I. CSDL (EAP) – tables Student

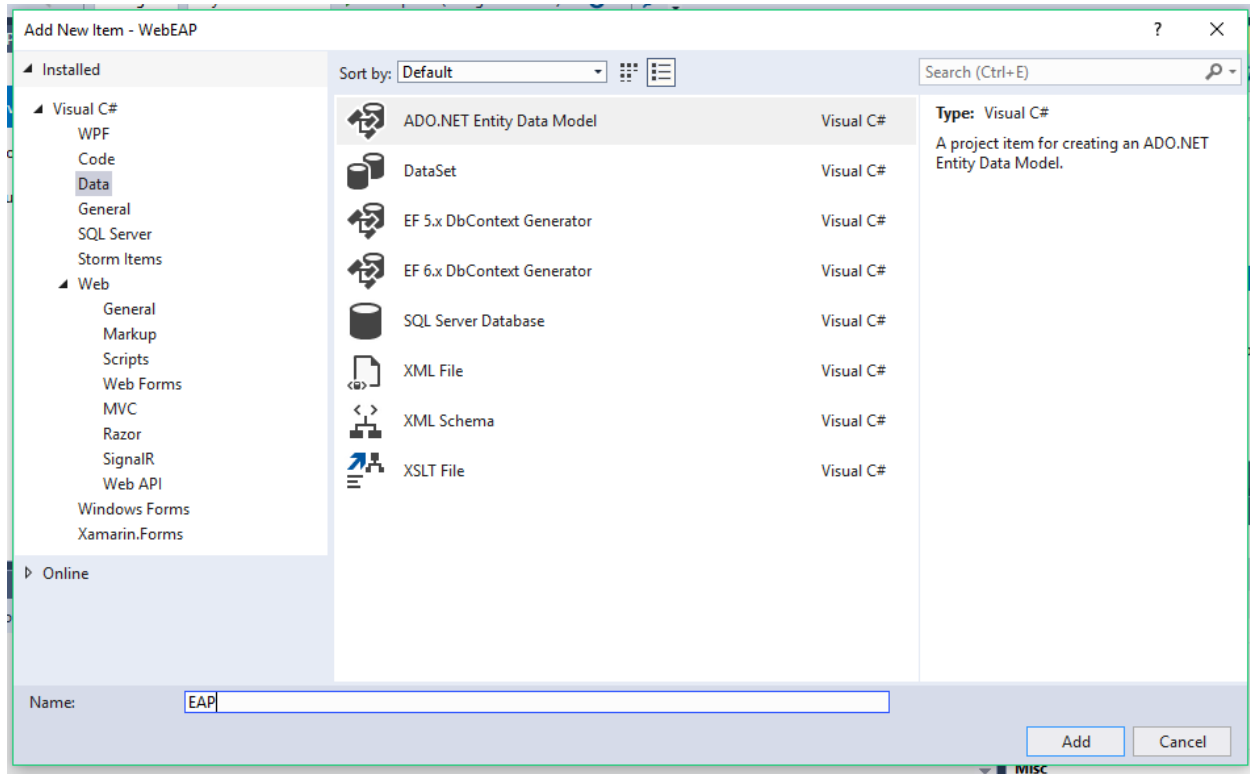
Student			
	Column Name	Data Type	Allow Nulls
	Student_Code	nvarchar(40)	<input type="checkbox"/>
	Student_Name	nvarchar(40)	<input checked="" type="checkbox"/>
	Student_Email	nvarchar(30)	<input checked="" type="checkbox"/>
	Student_Phone	nvarchar(30)	<input checked="" type="checkbox"/>
	Student_Details	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

II. Tạo API

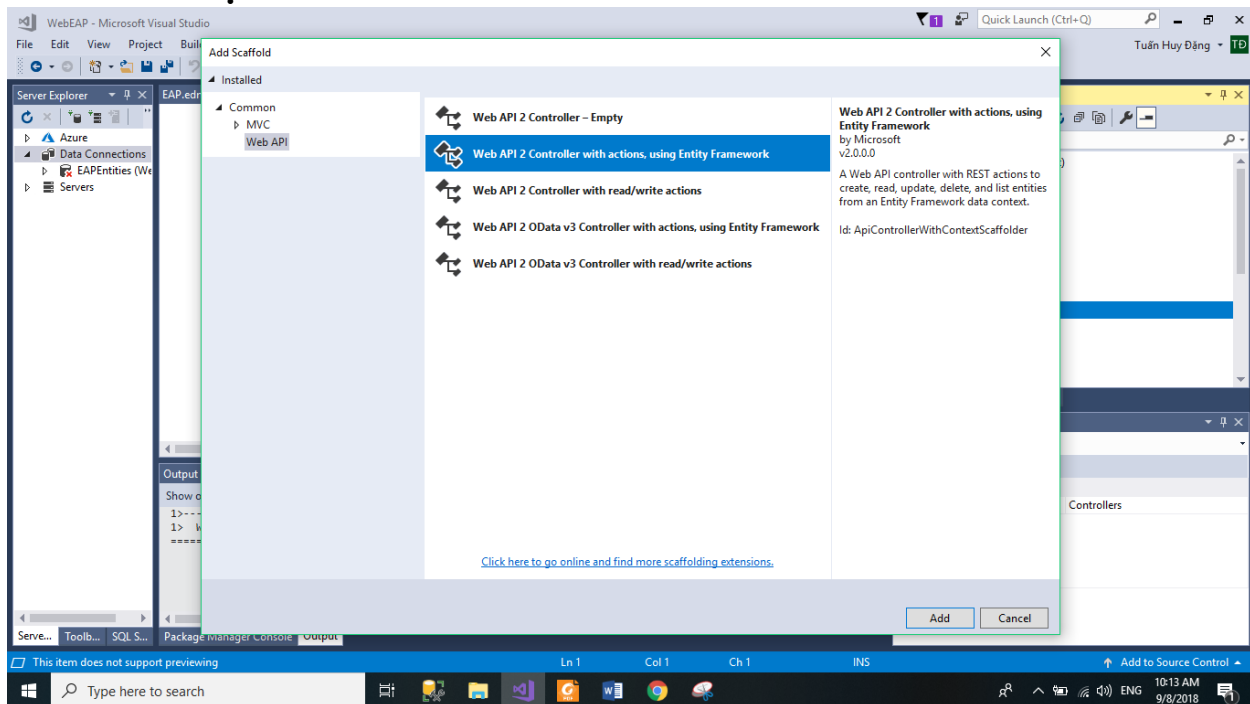
1. Tạo một Project API-Web



2. Sử dụng EntityFramework tạo Models



3. Tạo controller



4. Hiệu chỉnh

Trong `Application_Start()` bổ sung code convert xml->json

```
using System.Web.Http;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;
```

```
namespace WebEAP
{
    public class WebApiApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
```

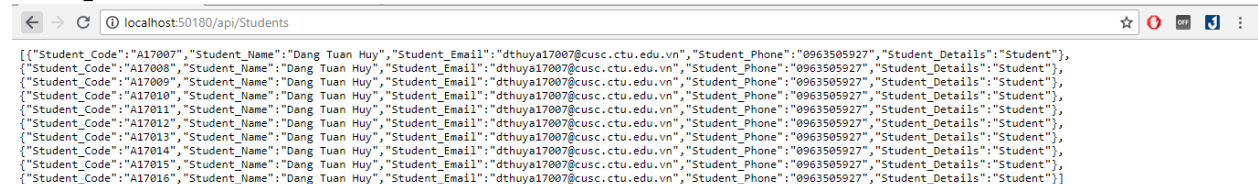
```
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.ReferenceLo
opHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;
```

```
GlobalConfiguration.Configuration.Formatters.Remove(GlobalConfiguration.Configuration.For
matters.XmlFormatter);
```

```
    AreaRegistration.RegisterAllAreas();
    GlobalConfiguration.Configure(WebApiConfig.Register);
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    BundleConfig.RegisterBundles(BundleTable.Bundles);
```

```
    }
}
}
```

Kết quả



The screenshot shows a web browser window with the address bar displaying `localhost:50180/api/Students`. The page content is a JSON array of 10 student objects. Each object contains fields for `Student_Code`, `Student_Name`, `Student_Email`, `Student_Phone`, and `Student_Details`. The data is as follows:

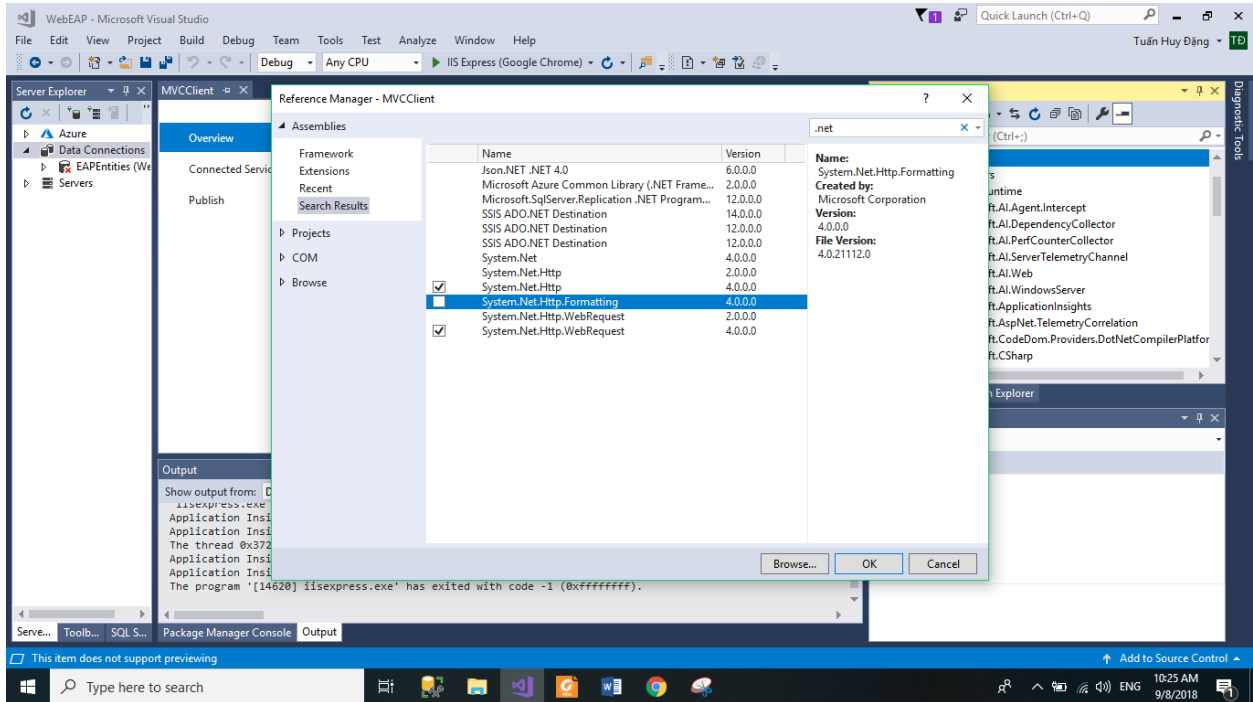
```
[{"Student_Code": "A17007", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17008", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17009", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17010", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17011", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17012", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17013", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17014", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17015", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"},
{"Student_Code": "A17016", "Student_Name": "Dang Tuan Huy", "Student_Email": "dthuya17007@cusc.ctu.edu.vn", "Student_Phone": "0963505927", "Student_Details": "Student"}]
```

III. API – Client

A. Web MVC – Client

Trong cùng Solution tạo một Project MVCClient

Trong References của MVCClient hiệu chỉnh



1. Trong model tạo một Class có tên là Student.cs

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

namespace MVCClient.Models
{
    public class Student
    {
        public string Student_Code { get; set; }
        public string Student_Name { get; set; }
        public string Student_Email { get; set; }
        public string Student_Phone { get; set; }
        public string Student_Details { get; set; }
    }
}
```

2. Tạo một controller emty có tên là StudentsController.cs Bổ sung code gọi API

```
using MVCClient.Models;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Threading.Tasks;
```

```

using System.Web;
using System.Web.Mvc;

namespace MVCCClient.Controllers
{
    public class StudentsController : Controller
    {
        HttpClient client;
        string url = "http://localhost:50180/api/Students";
        public StudentsController()
        {
            client = new HttpClient();
            client.BaseAddress = new Uri(url);
            client.DefaultRequestHeaders.Accept.Clear();
            client.DefaultRequestHeaders.Accept.Add(new
MediaTypewithQualityHeaderValue("application/json"));
        }
        // Get Student
        public async Task<ActionResult> Index()
        {
            HttpResponseMessage responseMessage = await client.GetAsync(url);

            if(responseMessage.IsSuccessStatusCode)
            {
                var responseData =
responseMessage.Content.ReadAsStringAsync().Result;

                var students =
JsonConvert.DeserializeObject<List<Student>>(responseData);
                return View(students);
            }
            return View("Error");
        }
        //Form Create
        public ActionResult Create()
        {
            return View(new Student());
        }
        [HttpPost]
        public async Task<ActionResult> Create(Student student)
        {
            HttpResponseMessage responseMessage = await
client.PostAsJsonAsync(url, student);
            if (responseMessage.IsSuccessStatusCode)
            {
                return RedirectToAction("Index");
            }
            return RedirectToAction("Error");
        }
        //Form Edit
        public async Task<ActionResult> Edit(string id)
        {
            HttpResponseMessage responseMessage = await client.GetAsync(url +
"/" + id);
            if (responseMessage.IsSuccessStatusCode)
            {
                var responseData =
responseMessage.Content.ReadAsStringAsync().Result;

```

```

        var student =
        JsonConvert.DeserializeObject<Student>(responseData);
        return View(student);
    }
    return View("Error");
}

[HttpPost]
public async Task<ActionResult> Edit(string id, Student student)
{
    HttpResponseMessage responseMessage = await
client.PutAsJsonAsync(url + "/" + id, student);
    if (responseMessage.IsSuccessStatusCode)
    {
        return RedirectToAction("Index");
    }
    return RedirectToAction("Error");
}
//Form Delete
public async Task<ActionResult> Delete(string id)
{
    HttpResponseMessage responseMessage = await client.GetAsync(url +
"/" + id);
    if (responseMessage.IsSuccessStatusCode)
    {
        var responseData =
responseMessage.Content.ReadAsStringAsync().Result;
        var student =
        JsonConvert.DeserializeObject<Student>(responseData);
        return View(student);
    }
    return View("Error");
}
[HttpPost]
public async Task<ActionResult> Delete(string id, Student student)
{
    HttpResponseMessage responseMessage = await client.DeleteAsync(url
+ "/" + id);
    if (responseMessage.IsSuccessStatusCode)
    {
        return RedirectToAction("Index");
    }
    return RedirectToAction("Error");
}
}
}

```

3. Tạo Views từ model Student

Code Index

```
@model IEnumerable<MVCCClient.Models.Student>
```

```
@{
```

```
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

```
}
```

```
<h2>List Student</h2>
```

```

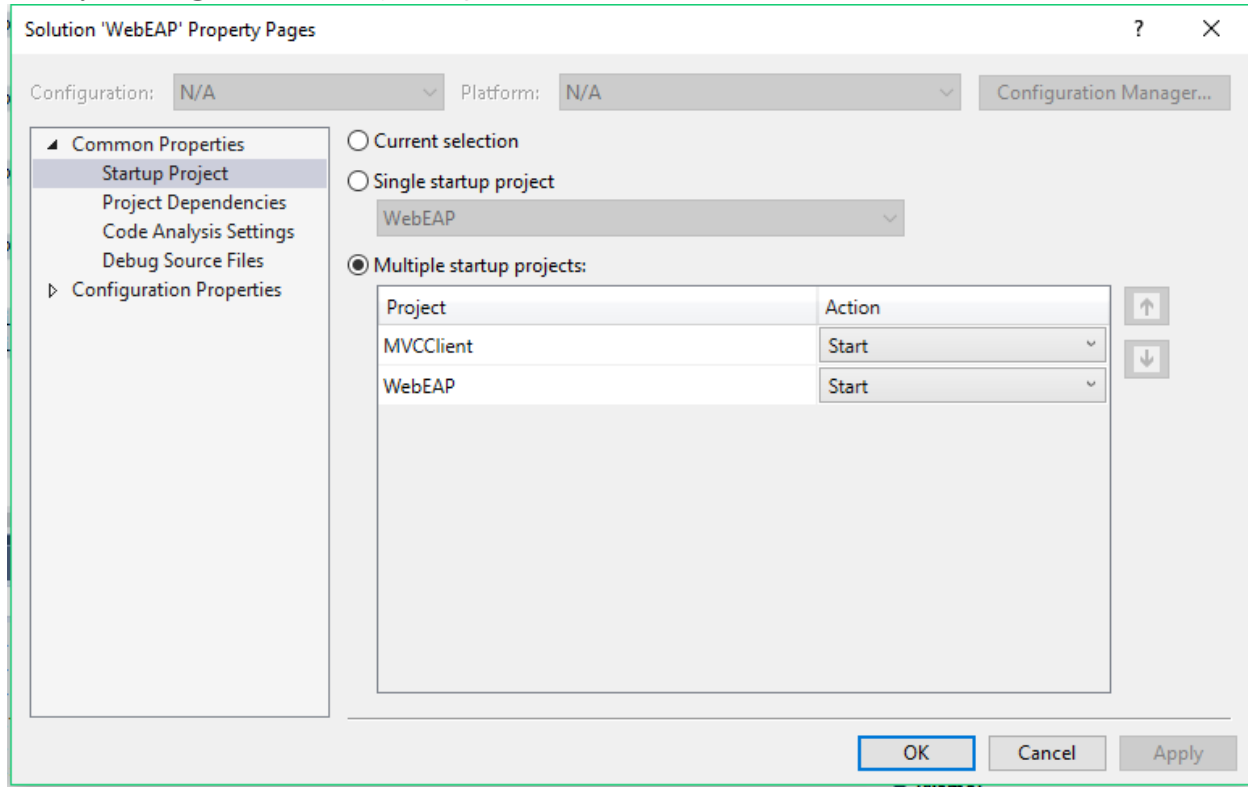
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table table-bordered">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Student_Code)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Student_Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Student_Email)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Student_Phone)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Student_Details)
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Student_Code)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Student_Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Student_Email)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Student_Phone)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Student_Details)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.Student_Code }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.Student_Code })
            </td>
        </tr>
    }
</table>

```

4. Kết quả

Lưu ý: config để API chạy cùng với MVC



○ Index:

The screenshot shows a web browser window displaying the 'Index' page of an ASP.NET application. The page title is 'List Student'. There is a 'Create New' link above the table. The table has the following columns: Student_Code, Student_Name, Student_Email, Student_Phone, Student_Details, and an action column with 'Edit | Delete' links.

Student_Code	Student_Name	Student_Email	Student_Phone	Student_Details	
A17007	Dang Tuan Huy	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17008	Nguyen Thi Yen Nhi	ntynhi16334@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17009	Nguyen Huu Hai	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17010	Tran Trung Tin	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17011	Thai Thien Tan	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17012	Nguyen Duc Trung	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17013	Tran Trung Phong	phonga18334@cusc.ctu.edu.vn	0123746984	Monitor	Edit Delete
A17014	Dang Tuan Huy	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17015	Dang Tuan Huy	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A17016	Dang Tuan Huy	dthuya17007@cusc.ctu.edu.vn	0963505927	Student	Edit Delete
A18008	Pham Nguyen Hoang Son	spnh@gmail.com	0925028451	Student	Edit Delete

© 2018 - My ASP.NET Application

○ Create

The screenshot shows a web browser window with the title 'Create - My ASP.NET App'. The address bar displays 'localhost:50307/Students/Create'. The page has a dark navigation bar with links for 'Application name', 'Home', 'About', and 'Contact'. The main content area is titled 'Create Student' and contains a form with five input fields: 'Student_Code', 'Student_Name', 'Student_Email', 'Student_Phone', and 'Student_Details'. A 'Create' button is positioned below the 'Student_Details' field. A 'Back to List' link is located at the bottom left of the form area. The footer of the page reads '© 2018 - My ASP.NET Application'.

Create Student

Student_Code

Student_Name

Student_Email

Student_Phone

Student_Details

Create

[Back to List](#)

© 2018 - My ASP.NET Application

Edit

Application name Home About Contact

Edit

Student

Student_Code	<input type="text" value="A17008"/>
Student_Name	<input type="text" value="Nguyen Thi Yen Nhi"/>
Student_Email	<input type="text" value="ntynhi16334@cusc.ctu.edu.vn"/>
Student_Phone	<input type="text" value="0963505927"/>
Student_Details	<input type="text" value="Student"/>

[Back to List](#)

© 2018 - My ASP.NET Application

Delete

Application name Home About Contact

Delete

Are you sure you want to delete this?

Student

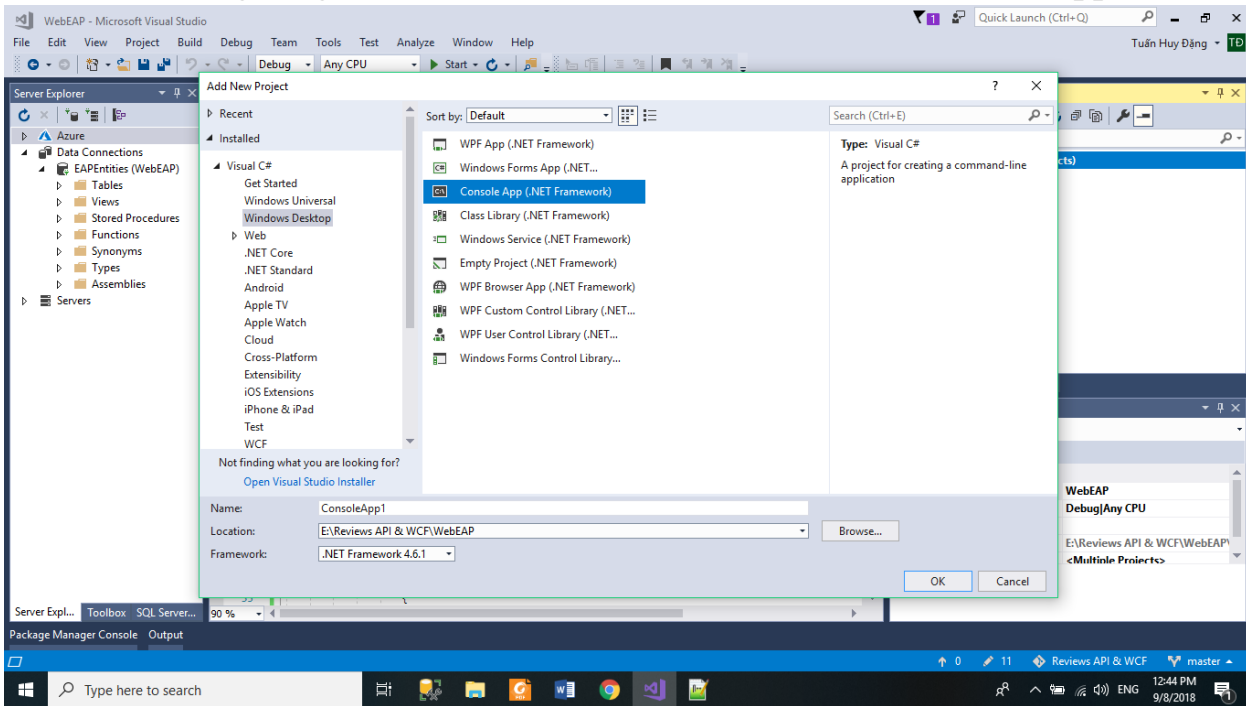
Student_Code	A17013
Student_Name	Tran Trung Phong
Student_Email	phonga18334@cusc.ctu.edu.vn
Student_Phone	0123746984
Student_Details	Monitor

| [Back to List](#)

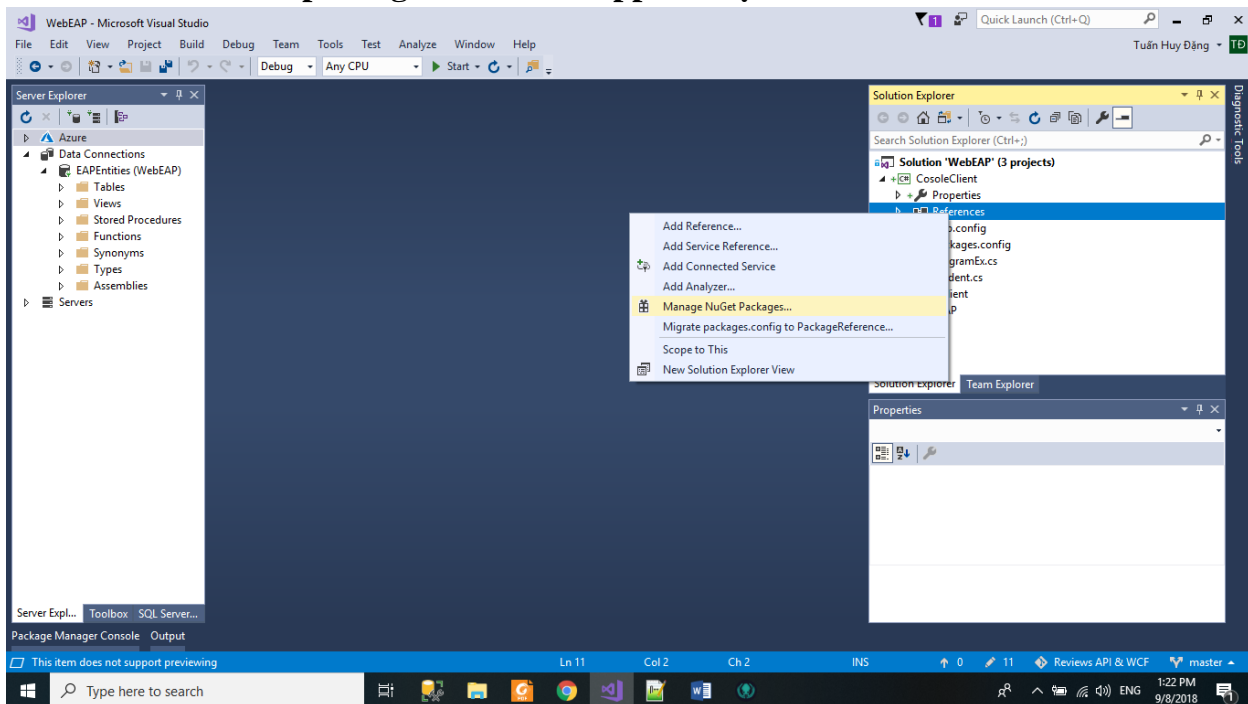
© 2018 - My ASP.NET Application

B. Console – Client

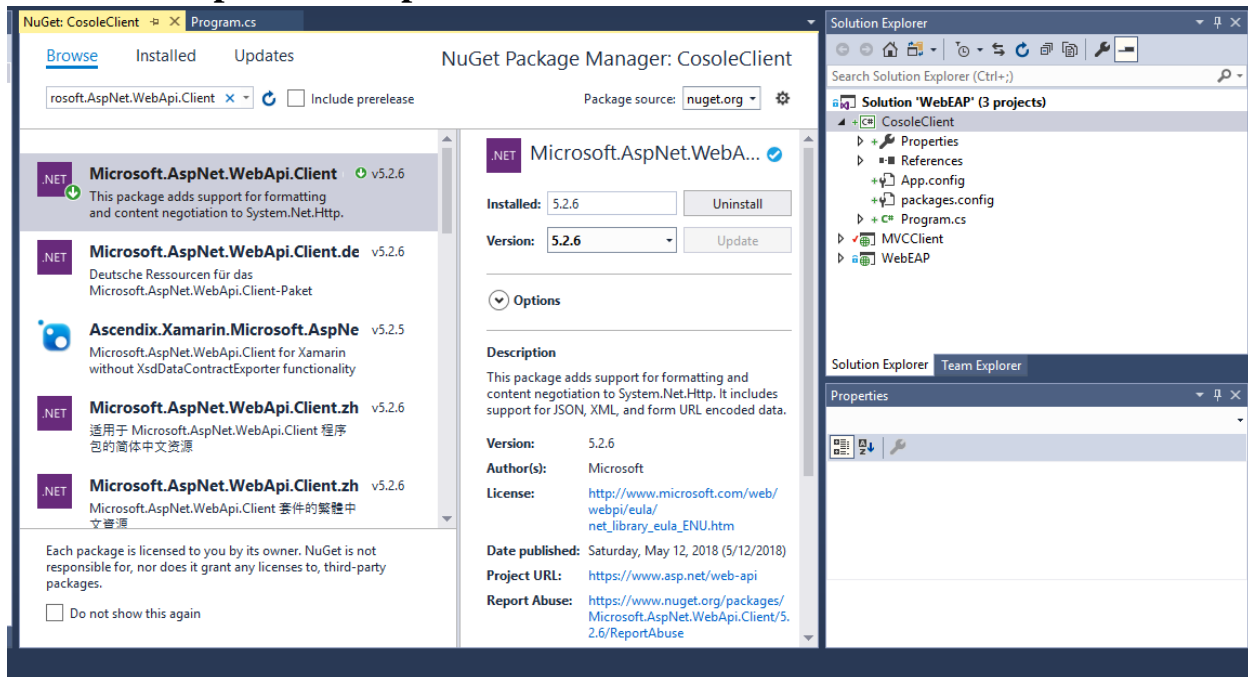
1. Trong cùng solution tạo một Project Console Client (Cosole Application)



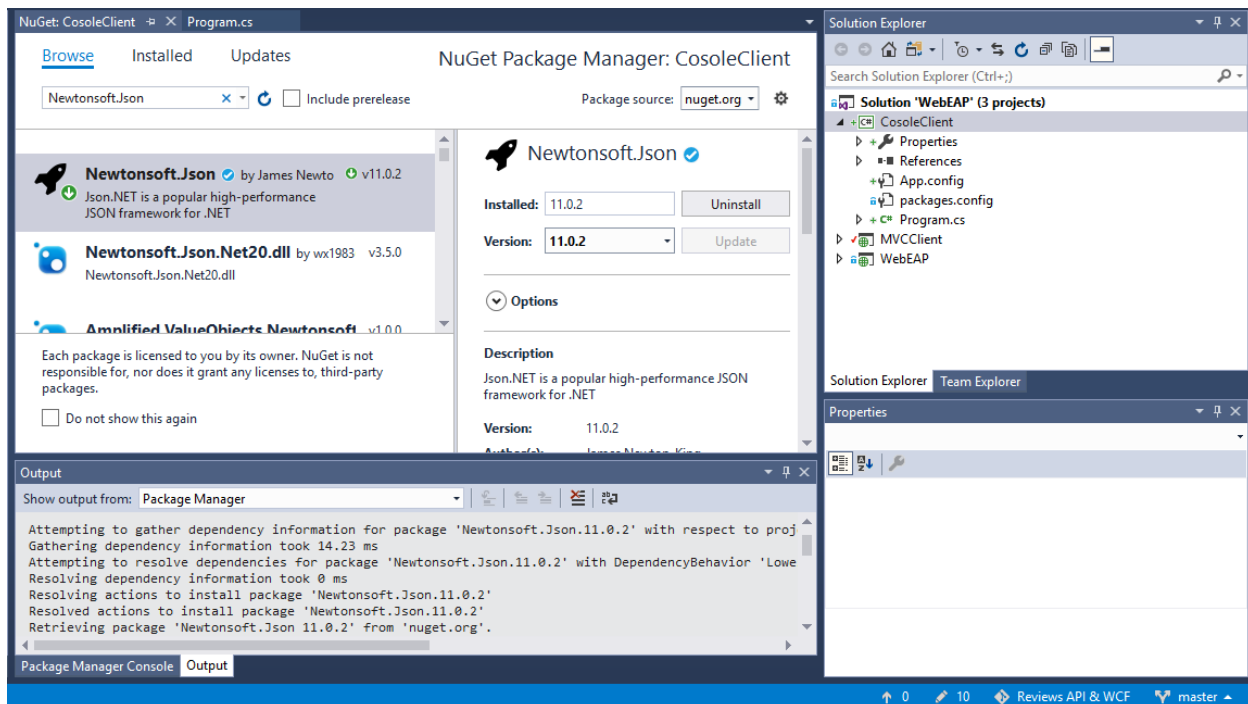
2. Thêm hai package cho cosele App vừa tạo



Microsoft.AspNet.WebApi.Client



Newtonsoft.Json



3. Tạo một Class có tên là Student.cs

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

namespace CosoleClient
```

```

{
    class Student
    {
        public string Student_Code { get; set; }
        public string Student_Name { get; set; }
        public string Student_Email { get; set; }
        public string Student_Phone { get; set; }
        public string Student_Details { get; set; }
    }
}

```

4. Đổi tên Programs.cs thành ProgramEx.cs và bổ sung code gọi API

```

using Newtonsoft.Json.Linq;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleClient
{
    class ProgramEx
    {
        // Lấy thông tin tất cả sinh viên
        static async Task GetAll()
        {
            string url = "http://localhost:50180/api/Students";
            var client = new HttpClient();
            client.BaseAddress = new Uri(url);
            client.DefaultRequestHeaders.Accept.Clear();
            client.DefaultRequestHeaders.Accept.Add(new
MediaTyewithQualityHeaderValue("application/json"));
            HttpResponseMessage response = await client.GetAsync(url);
            if (response.IsSuccessStatusCode)
            {
                var responseData = response.Content.ReadAsStringAsync().Result;

                JObject parsed = JObject.Parse(responseData.ToString());

                Console.WriteLine("-----List of Student '
Information-----");
                foreach (var pair in parsed)
                {
                    JObject obj = JObject.Parse(pair.ToString());
                    foreach (var s in obj)
                    {
                        Console.WriteLine("{0} : {1}", s.Key,
s.Value.ToString().Trim());
                    }
                    Console.WriteLine();
                }
            }
            // Tìm Kiếm sinh viên theo ID nè
            static async Task GetStudent(string id)
            {

```

```

        string url = "http://localhost:50180/api/Students";
        var client = new HttpClient();
        client.BaseAddress = new Uri(url);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
        HttpResponseMessage response = await client.GetAsync(url + "/" + id);
        if (response.IsSuccessStatusCode)
        {
            Student student = await response.Content.ReadAsAsync<Student>();
            Console.WriteLine("-----Lecturer information-----
-----");
            Console.WriteLine("Student Code: " + id);
            Console.WriteLine("Student Name: " + student.Student_Name);
            Console.WriteLine("Student Email: " + student.Student_Email);
            Console.WriteLine("Student Phone: " + student.Student_Phone);
            Console.WriteLine("Student Details: " +
student.Student_Details.ToString().Trim());
        }
        else
        {
            Console.WriteLine("Lecturer is not exist");
        }
    }

    static async Task AddNewStudent(string id, string name, string email, string
phone, string details)
    {
        string url = "http://localhost:50180/api/Students";
        var client = new HttpClient();
        client.BaseAddress = new Uri(url);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
        HttpResponseMessage response = await client.GetAsync(url);
        Student st = new Student()
        {
            Student_Code = id,
            Student_Name = name,
            Student_Email = email,
            Student_Phone = phone,
            Student_Details = details
        };
        response = await client.PostAsJsonAsync(url, st);
        if (response.IsSuccessStatusCode)
        {
            Console.WriteLine("-----Student's information is added-
-----");
        }
        else
        {
            Console.WriteLine("-----Student's information cannot
addto database-----");
        }
    }

    static async Task UpdateStudent(string id, string name, string email, string
phone, string details)
    {
        string url = "http://localhost:50180/api/Students";

```

```

var client = new HttpClient();
client.BaseAddress = new Uri(url);
client.DefaultRequestHeaders.Accept.Clear();
client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
HttpResponseMessage response = await client.GetAsync(url + "/" + id);
if (response.IsSuccessStatusCode)
{
    Student student = await response.Content.ReadAsAsync<Student>();
    if (name != "")
        student.Student_Name = name;
    if (email != "")
        student.Student_Email = email;
    if (phone != "")
        student.Student_Phone = phone;
    if (details != "")
        student.Student_Details = details;
    response = await client.PutAsJsonAsync(url + "/" + id, student);
    if (response.IsSuccessStatusCode)
    {
        Console.WriteLine("-----Update Student
Successfully-----");
    }
    else
    {
        Console.WriteLine("-----Update Failure-----
-----");
    }
}
}
static async Task DeleteStudent(string id)
{
    string url = "http://localhost:50180/api/Students";
    var client = new HttpClient();
    client.BaseAddress = new Uri(url);
    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
    HttpResponseMessage response = await client.GetAsync(url + "/" + id);
    response = await client.DeleteAsync(url + "/" + id);
    if (response.IsSuccessStatusCode)
    {
        Console.WriteLine("-----Delete Successfully-----
-----");
    }
    else
    {
        Console.WriteLine("-----Delete Failure-----
---");
    }
}
static void Main(string[] args)
{
    string i;
    do
    {
        Console.WriteLine("1. Get all Student' information");
        Console.WriteLine("2. Get only one student's information");
    }
}

```

```

Console.WriteLine("3. Add new student");
Console.WriteLine("4. Update student's information");
Console.WriteLine("5. Delete student from database");
Console.Write("Enter your choice: ");
int ans = Convert.ToInt32(Console.ReadLine());
if (ans == 1)
{
    GetAll().Wait();
}
else
{
    if (ans == 2)
    {
        Console.Write("Enter the student code: ");
        string id = Console.ReadLine();
        GetStudent(id).Wait();
    }
    else
    {
        if (ans == 3)
        {
            Console.WriteLine("-----Enter the
Student'sInformation-----");
            Console.Write("Enter Student Code:");
            string id = Console.ReadLine();
            Console.Write("Enter Student Name:");
            string name = Console.ReadLine();
            Console.Write("Enter Student Email:");
            string email = Console.ReadLine();
            Console.Write("Enter Student Phone:");
            string phone = Console.ReadLine();
            Console.Write("Enter Student Details:");
            string details = Console.ReadLine();
            AddNewStudent(id, name, email, phone, details).Wait();
        }
        else
        {
            if (ans == 4)
            {
                Console.Write("Enter the student Code: ");
                string id = Console.ReadLine();
                Console.WriteLine("-----Enter the
Student's Information-----");
                Console.Write("Enter student Name:");
                string name = "";
                name = Console.ReadLine();
                Console.Write("Enter studentEmail:");
                string email = "";
                email = Console.ReadLine();
                Console.Write("Enter student Phone:");
                string phone = "";
                phone = Console.ReadLine();
                Console.Write("Enter student details :");
                string details = "";
                details = Console.ReadLine();
                UpdateStudent(id, name, email, phone, details).Wait();
            }
            else

```

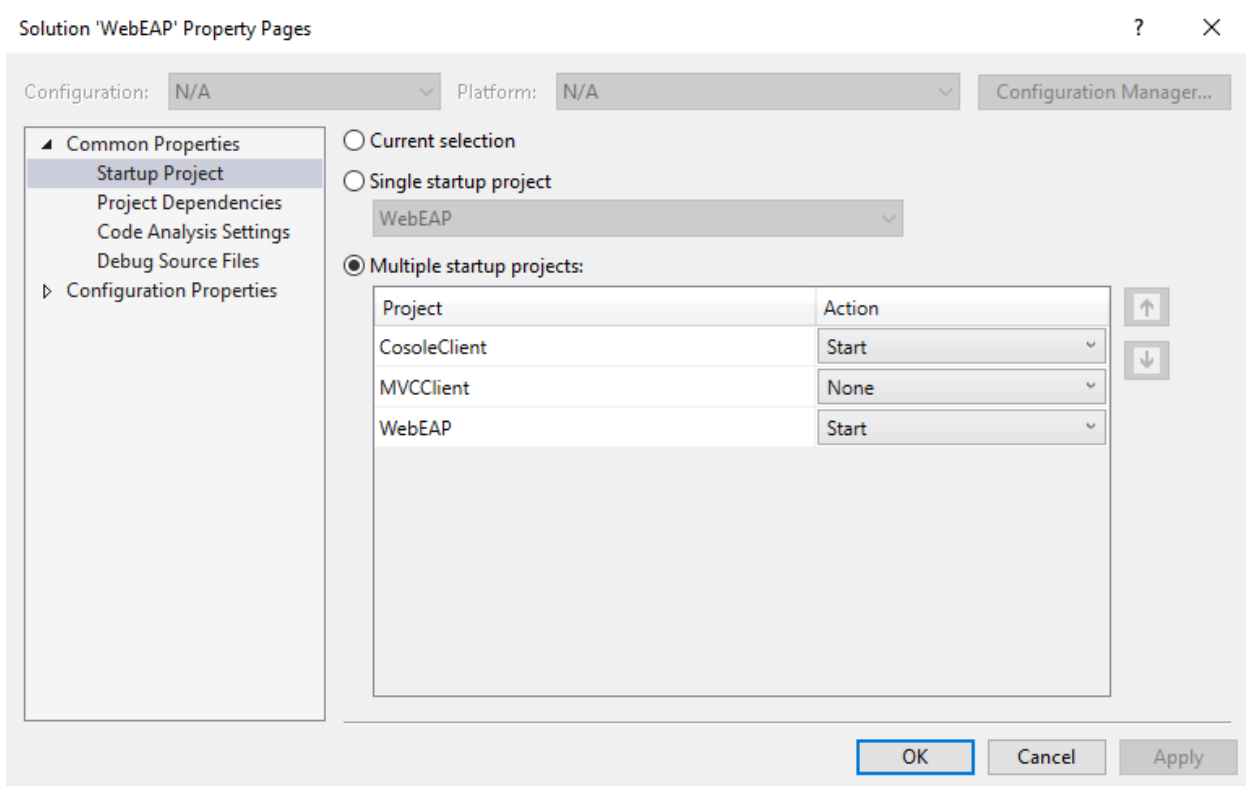


```

        {
            Console.WriteLine("Enter the student code: ");
            string id = Console.ReadLine();
            DeleteStudent(id).Wait();
        }
    }
}
Console.WriteLine("Do you want to continue: ");
i = Console.ReadLine();
} while (i == "y" || i == "Y");
Console.ReadLine();
}
}
}

```

5. Kết quả: Config App:



○ Danh sách toàn bộ

```

E:\Reviews API & WCF\WebEAP\CosoleClient\bin\Debug\CosoleClient.exe
Student_Details : Student
Student_Code : A17011
Student_Name : Thai Thien Tan
Student_Email : dthuya17007@cusc.ctu.edu.vn
Student_Phone : 0963505927
Student_Details : Student
Student_Code : A17012
Student_Name : Nguyen Duc Trung
Student_Email : dthuya17007@cusc.ctu.edu.vn
Student_Phone : 0963505927
Student_Details : Student
Student_Code : A17013
Student_Name : Tran Trung Phong
Student_Email : phongal8334@cusc.ctu.edu.vn
Student_Phone : 0123746984
Student_Details : Monitor
Student_Code : A17014
Student_Name : Ly Thuy Kieu
Student_Email : kieu@gmail.com
Student_Phone : 0963505987
Student_Details : Moniator
Student_Code : A17015
Student_Name : Dang Tuan Huy
Student_Email : dthuya17007@cusc.ctu.edu.vn
Student_Phone : 0963505927

```

○ Tìm theo ID

```

E:\Reviews API & WCF\WebEAP\CosoleClient\bin\Debug\CosoleClient.exe
Student_Name : Pham Nguyen Hoang Son
Student_Email : spnh@gmail.com
Student_Phone : 0925028451
Student_Details : Student
Do you want to continue: y
1. Get all Student' information
2. Get only one student's information
3. Add new student
4. Update student's information
5. Delete student from database
Enter your choice: 2
Enter the student code: A14567
-----Lecturer information-----
Student Code: A14567
Student Name: Nguyen Anh Tuan
Student Email: natuan@cusc.ctu.edu.vn
Student Phone: 0965456789
Student Details: Manager
Do you want to continue:

```

○ Thêm

```

E:\Reviews API & WCF\WebEAP\CosoleClient\bin\Debug\CosoleClient.exe
Student_Phone : 0925028451
Student_Details : Student
Do you want to continue: y
1. Get all Student' information
2. Get only one student's information
3. Add new student
4. Update student's information
5. Delete student from database
Enter your choice: 3
-----Enter the Student'sInformation-----
Enter Student Code:A18901
Enter Student Name:Nguyen Hai Au
Enter Student Email:nhau@gmail.com
Enter Student Phone:097653456
Enter Student Details:Managers
-----Student's information is added-----
Do you want to continue:

```

○ Sửa

```

E:\Reviews API & WCF\WebEAP\CosoleClient\bin\Debug\CosoleClient.exe
Enter Student Details:Managers
-----Student's information is added-----
Do you want to continue: y
1. Get all Student' information
2. Get only one student's information
3. Add new student
4. Update student's information
5. Delete student from database
Enter your choice: 4
Enter the student Code: A18901
-----Enter the Student's Information-----
Enter student Name:Nguyen Hai Au
Enter studentEmail:nhau@cusc.ctu.edu.vn
Enter student Phone:01236457890
Enter student details :No
-----Update Student Successfully-----
Do you want to continue:

```

○ Xóa

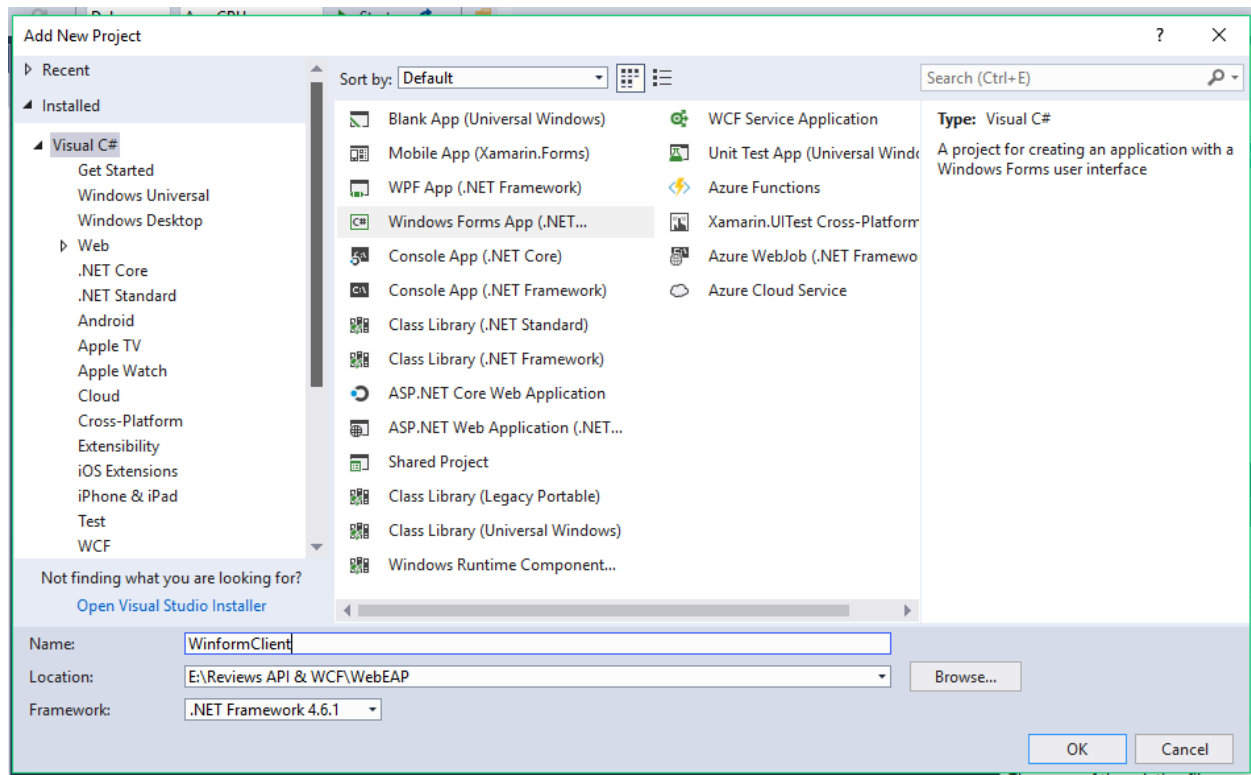
```

E:\Reviews API & WCF\WebEAP\CosoleClient\bin\Debug\CosoleClient.exe
-----Update Student Successfully-----
Do you want to continue: y
1. Get all Student' information
2. Get only one student's information
3. Add new student
4. Update student's information
5. Delete student from database
Enter your choice: 5
Enter the student code: A17015
-----Delete Successfully-----
Do you want to continue:

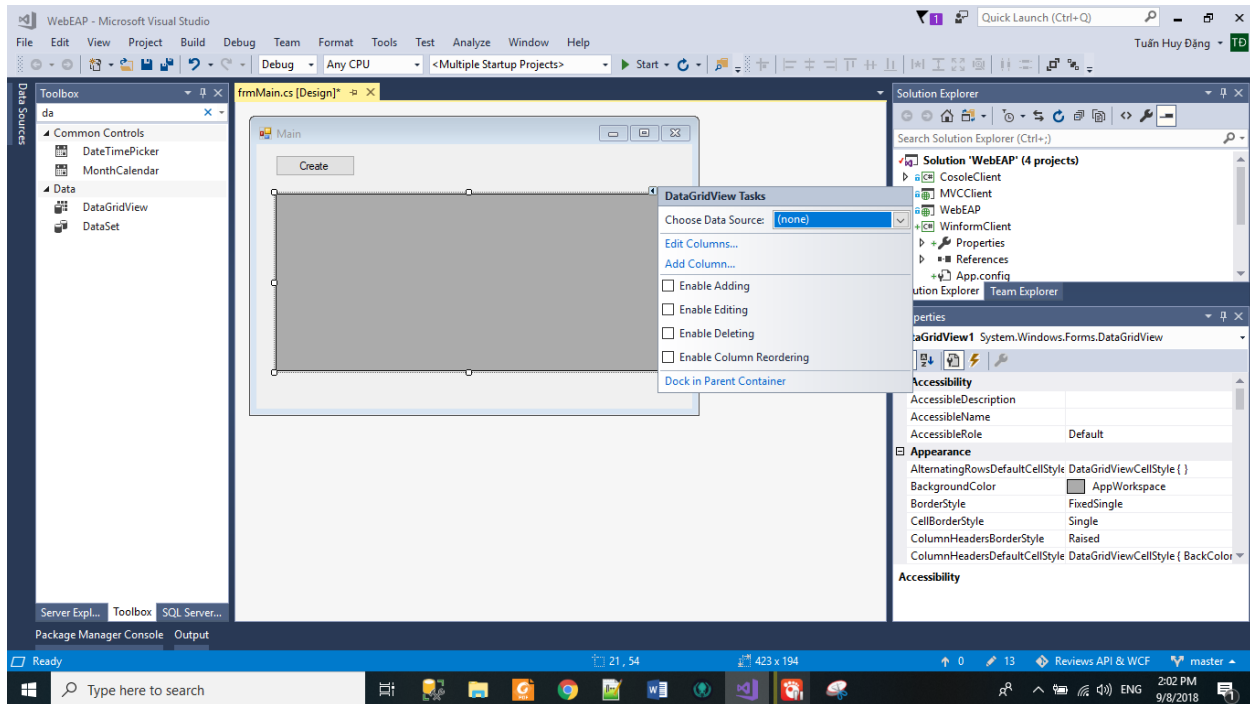
```

C. WindowForm – Client

1. Trong cùng một solution tạo một Project Window Form Application

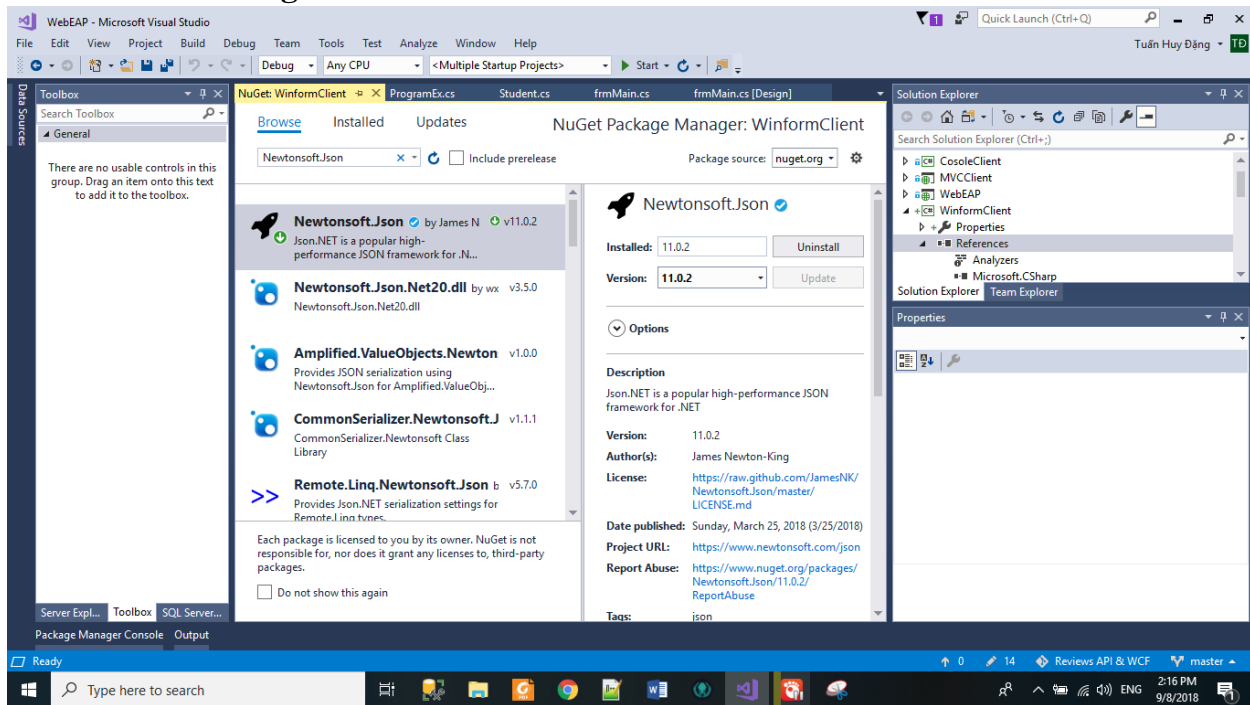


2. Tạo form sau

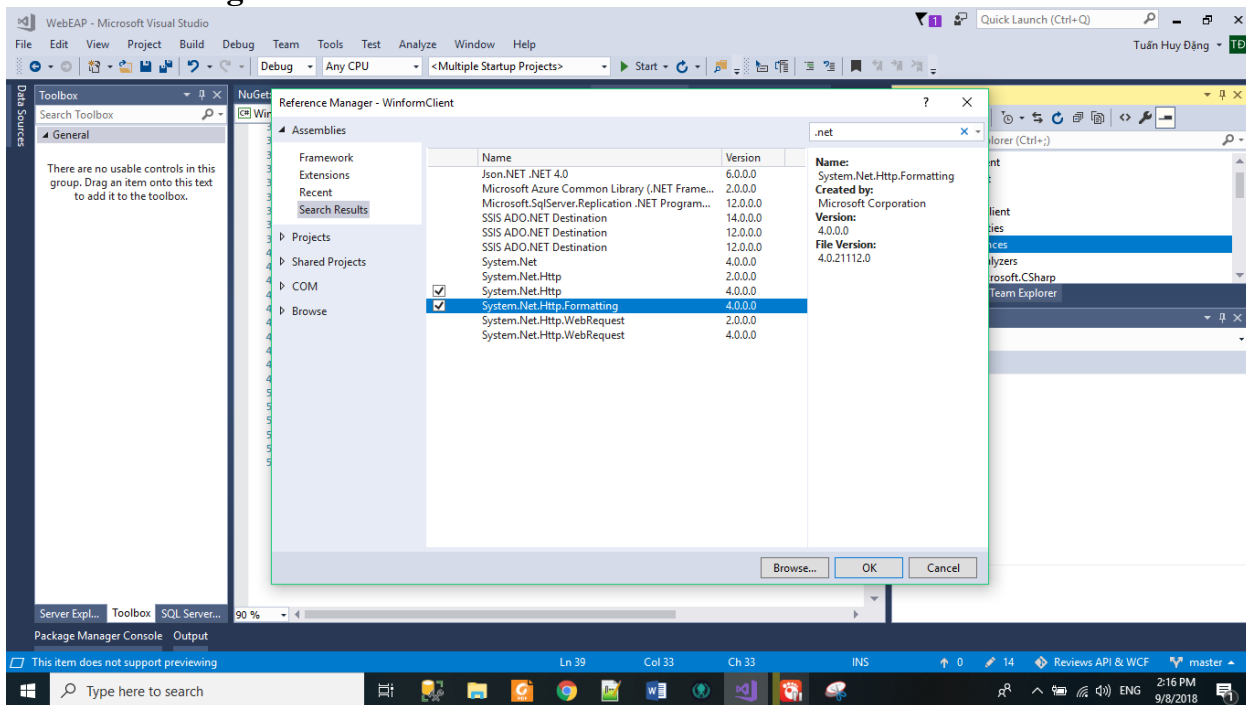


Button : Name btnCreate

3. Install Package



4. Config

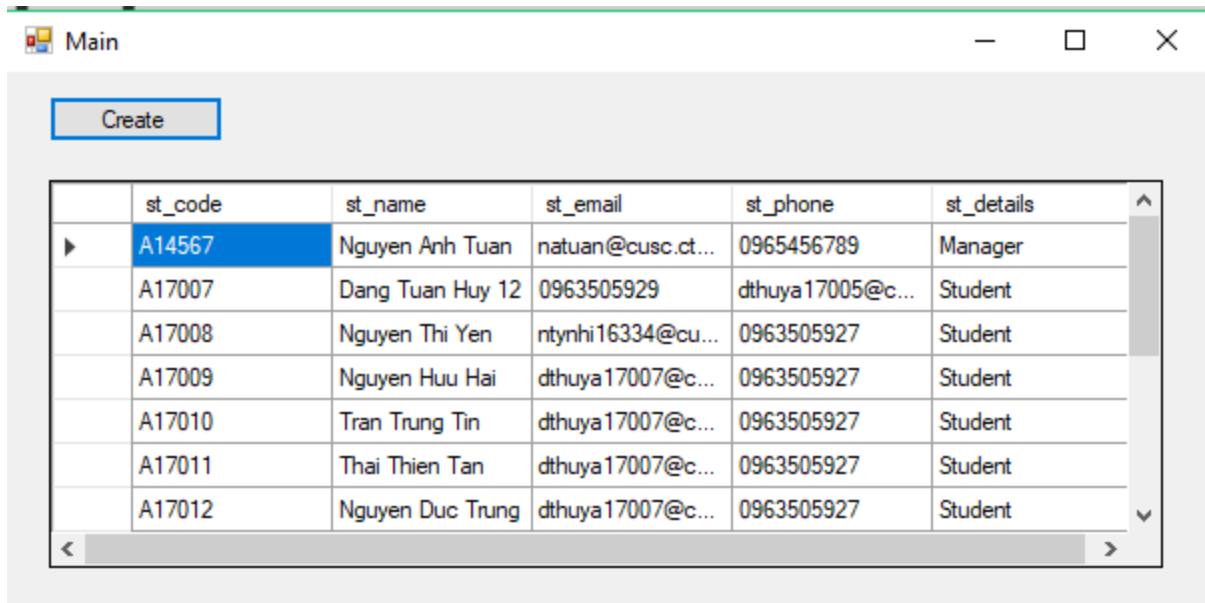


5. Trong Project Winform tạo Class Student.cs

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

namespace WinformClient
{
    public class Student
    {
        public string Student_Code { get; set; }
        public string Student_Name { get; set; }
        public string Student_Email { get; set; }
        public string Student_Phone { get; set; }
        public string Student_Details { get; set; }
    }
}
```

6. Tạo frmMain có giao diện như trong hình



7. Bổ sung code sự kiện và gọi API cho frmMain

```
using Newtonsoft.Json.Linq;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinformClient
{
    public partial class frmMain : Form
    {
        public frmMain()
        {
            InitializeComponent();
        }

        private void frmMain_Load(object sender, EventArgs e)
        {
            GetAll();
        }

        public async Task GetAll()
        {
            string url = "http://localhost:50180/api/Students";
            var client = new HttpClient();
            client.BaseAddress = new Uri(url);
            client.DefaultRequestHeaders.Accept.Clear();
            client.DefaultRequestHeaders.Accept.Add(new
            MediaTypeWithQualityHeaderValue("application/json"));
        }
    }
}
```

```

HttpResponseMessage response = await client.GetAsync(url);

// Chinh sửa và Fill Form
DataTable table = new DataTable();
table.Columns.Add("st_code");
table.Columns.Add("st_name");
table.Columns.Add("st_email");
table.Columns.Add("st_phone");
table.Columns.Add("st_details");
//Kết thúc
if (response.IsSuccessStatusCode)
{
    var responseData = response.Content.ReadAsStringAsync().Result;

    JArray parsed = JArray.Parse(responseData.ToString());

    Console.WriteLine("-----List of Student '
Information-----");
    foreach (var pair in parsed)
    {
        //dùng DataRow
        DataRow row = table.NewRow(); //QT
        JObject obj = JObject.Parse(pair.ToString());
        int count = 0; //QT
        foreach (var s in obj)
        {
            //Console.WriteLine("{0} : {1}", s.Key,
s.Value.ToString().Trim());
            row[count++] = s.Value.ToString();
        }
        // Console.WriteLine();
        table.Rows.Add(row);
    }
    dataGridView1.DataSource = table;
}

private void btnCreate_Click(object sender, EventArgs e)
{
    frmCreate frm = new frmCreate();
    frm.Show();
    this.Hide();
}

private void dataGridView1_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
{
    int index = dataGridView1.CurrentCell.RowIndex;
    Student st = new Student()
    {
        Student_Code = dataGridView1.Rows[index].Cells[0].Value.ToString(),
        Student_Name = dataGridView1.Rows[index].Cells[1].Value.ToString(),
        Student_Email = dataGridView1.Rows[index].Cells[2].Value.ToString(),
        Student_Phone = dataGridView1.Rows[index].Cells[3].Value.ToString(),
        Student_Details = dataGridView1.Rows[index].Cells[4].Value.ToString()
    };
    frmEdit frm = new frmEdit(st);
    frm.Show();
}

```



```

        this.Hide();
    }
}

```

8. Tạo frmCreate như giao diện

9. Bổ sung code gọi API và sự kiện cho frmCreate

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinformClient
{
    public partial class frmCreate : Form
    {
        public frmCreate()
        {
            InitializeComponent();
        }

        private void frmCreate_FormClosed(object sender, FormClosedEventArgs e)
        {
            frmMain frm = new frmMain();
            frm.Show();
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            txtStudentCode.Text = "";
        }
    }
}

```

```

        txtName.Text = "";
        txtPhone.Text = "";
        txtEmail.Text = "";
        txtDetails.Text = "";
    }

    private void btnAdd_Click(object sender, EventArgs e)
    {
        AddNewStudent(txtStudentCode.Text, txtName.Text, txtPhone.Text,
txtEmail.Text, txtDetails.Text);
    }
    static async Task AddNewStudent(string id, string name, string email, string
phone, string details)
    {
        string url = "http://localhost:50180/api/Students";
        var client = new HttpClient();
        client.BaseAddress = new Uri(url);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
        HttpResponseMessage response = await client.GetAsync(url);

        Student st = new Student()
        {
            Student_Code = id,
            Student_Name = name,
            Student_Email = email,
            Student_Phone = phone,
            Student_Details = details
        };
        response = await client.PostAsJsonAsync(url, st);
        if (response.IsSuccessStatusCode)
        {
            MessageBox.Show("Ok", "Infor", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            frmMain frm = new frmMain();
            frm.Show();
        }
        else
        {
            MessageBox.Show("Fail", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
}
}
}

```

10. Thiết kế formEdit

The screenshot shows a Windows Form titled "Edit" with the following fields and buttons:

Student Code:	A17016
Student Name:	Dang Tuan Huy
Student Email:	dthuya17007@cusc.ctu.edu.vn
Student Phone:	0963505927
Details:	Student

At the bottom of the form, there are two buttons: "Update" and "Reset".

11. Bổ sung code gọi API và sự kiện cho frmEdit

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

namespace WinformClient
{
    public partial class frmEdit : Form
    {
        public frmEdit()
        {
            InitializeComponent();
        }
        public frmEdit(Student st)
        {
            InitializeComponent();
            txtStudentCode.Text = st.Student_Code;
            txtName.Text = st.Student_Name;
            txtEmail.Text = st.Student_Email;
            txtPhone.Text = st.Student_Phone;
            txtDetails.Text = st.Student_Details;
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            txtName.Text = "";
            txtPhone.Text = "";
            txtEmail.Text = "";
        }
    }
}
```

```

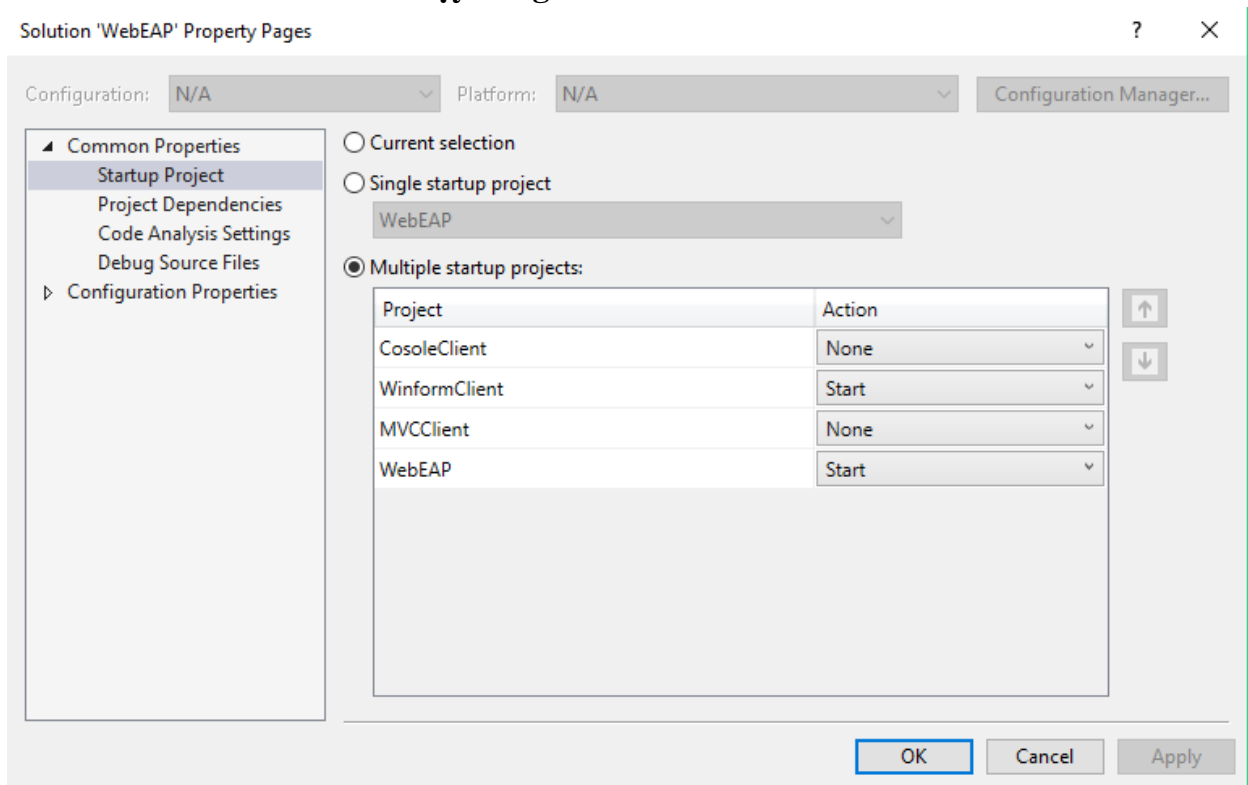
        txtDetails.Text = "";
    }
    static async Task UpdateStudent(string id, string name, string email, string
phone, string details)
    {
        string url = "http://localhost:50180/api/Students";
        var client = new HttpClient();
        client.BaseAddress = new Uri(url);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
        HttpResponseMessage response = await client.GetAsync(url + "/" + id);
        if (response.IsSuccessStatusCode)
        {
            Student student = await response.Content.ReadAsAsync<Student>();
            if (name != "")
                student.Student_Name = name;
            if (email != "")
                student.Student_Email = email;
            if (phone != "")
                student.Student_Phone = phone;
            if (details != "")
                student.Student_Details = details;
            response = await client.PutAsJsonAsync(url + "/" + id, student);
            if (response.IsSuccessStatusCode)
            {
                MessageBox.Show("Ok", "Information", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                frmMain frm = new frmMain();
                frm.Show();
            }
            else
            {
                MessageBox.Show("Ok", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
private void btnUpdate_Click(object sender, EventArgs e)
{
    UpdateStudent(txtStudentCode.Text, txtName.Text, txtPhone.Text,
txtEmail.Text, txtDetails.Text);
}

private void frmEdit_FormClosing(object sender, FormClosingEventArgs e)
{
    frmMain frm = new frmMain();
    frm.Show();
    this.Hide();
}
}
}

```

12. Config

Cấu hình API và winform chạy cùng lúc



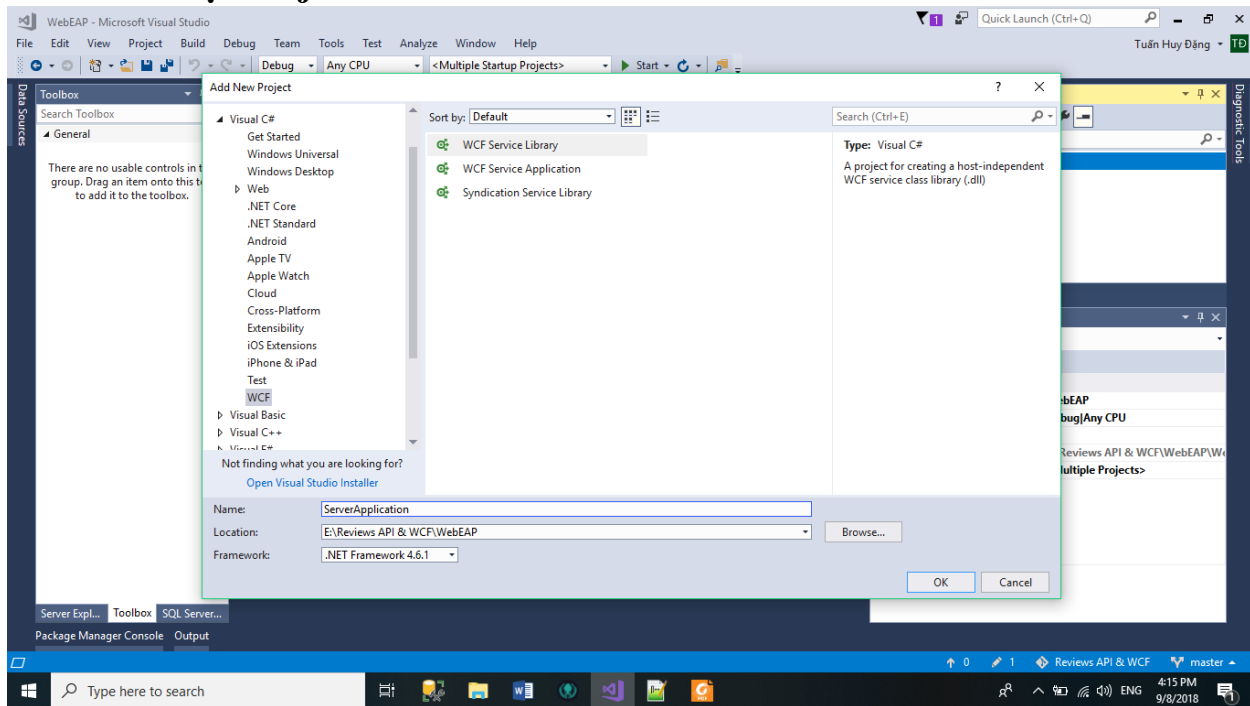
IV. WCF

Sử dụng lại CSDL

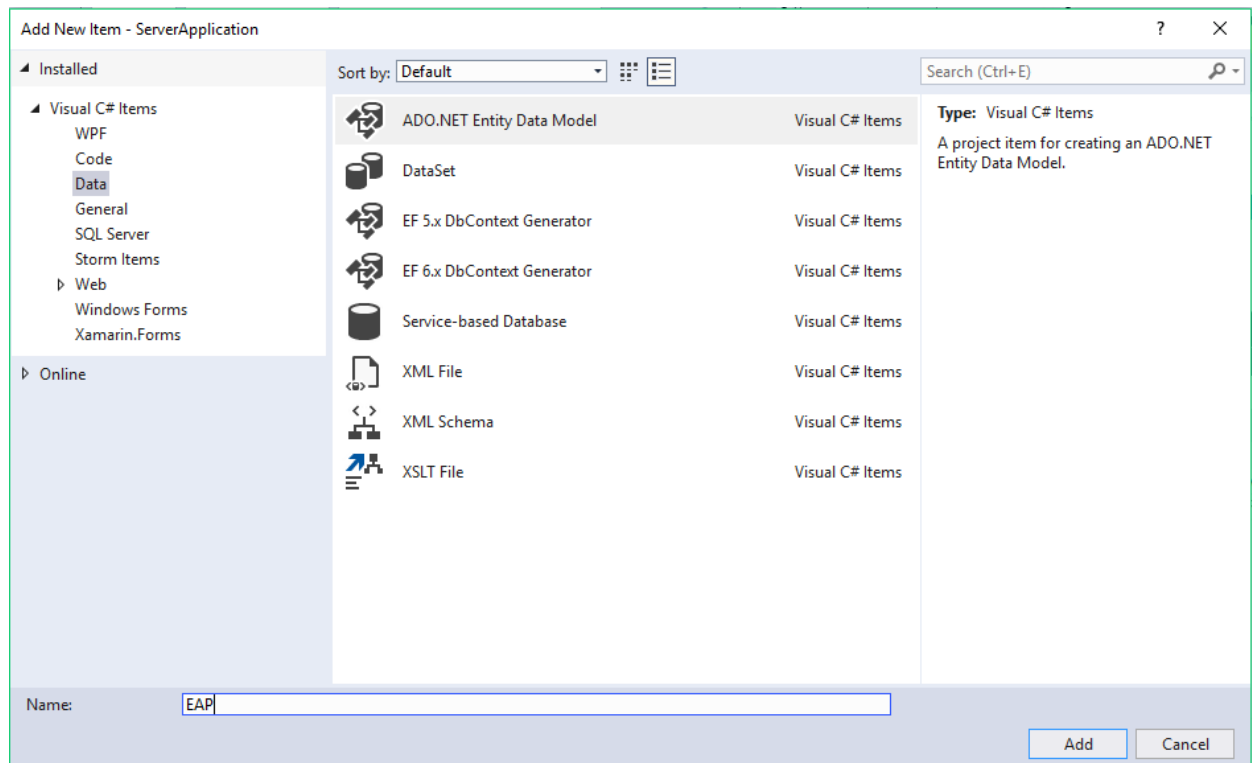
Student			
	Column Name	Data Type	Allow Nulls
🔑	Student_Code	nvarchar(40)	<input type="checkbox"/>
	Student_Name	nvarchar(40)	<input checked="" type="checkbox"/>
	Student_Email	nvarchar(30)	<input checked="" type="checkbox"/>
	Student_Phone	nvarchar(30)	<input checked="" type="checkbox"/>
	Student_Details	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tạo WCF Service

Bước 1: Tạo Project



Bước 2: Sử dụng Entity Framework



Bước 3: Bổ sung code trong IService1.cs

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

namespace ServerApplication
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the
    // interface name "IService1" in both code and config file together.
    [ServiceContract]
    public interface IService1
    {
        //Bổ sung
        [OperationContract]
        List<Student> GetStudents();
        [OperationContract]
        Student GetStudent(string id);
        [OperationContract]
        List<Student> FindStudent(string id);
        [OperationContract]
        bool AddNewStudent(Student student);
        [OperationContract]
        bool UpdateStudent(Student student);
        [OperationContract]
        bool DeleteStudent(string id);
    }
}
```

```
}
```

Bước 4: Implement Service1 và bổ sung code

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

namespace ServerApplication
{
    public class Service1 : IService1
    {
        EAPEntities db = new EAPEntities();
        public List<Student> GetStudents()
        {
            return db.Students.ToList();
        }
        public Student GetStudent(string id)
        {
            return db.Students.Find(id);
        }

        public List<Student> FindStudent(string name)
        {
            var students = from s in db.Students select s;
            students = students.Where(s => s.Student_Name.Contains(name));
            students = students.OrderBy(s => s.Student_Code);
            return students.ToList();
        }
        public bool AddNewStudent(Student student)
        {
            try
            {
                db.Students.Add(student);
                db.SaveChanges();
                return true;
            }
            catch (Exception)
            {
                return false;
            }
        }
        public bool UpdateStudent(Student student)
        {
            try
            {
                db.Entry(student).State = EntityState.Modified;
                db.SaveChanges();
                return true;
            }
            catch (Exception)
            {
                return false;
            }
        }
    }
}
```



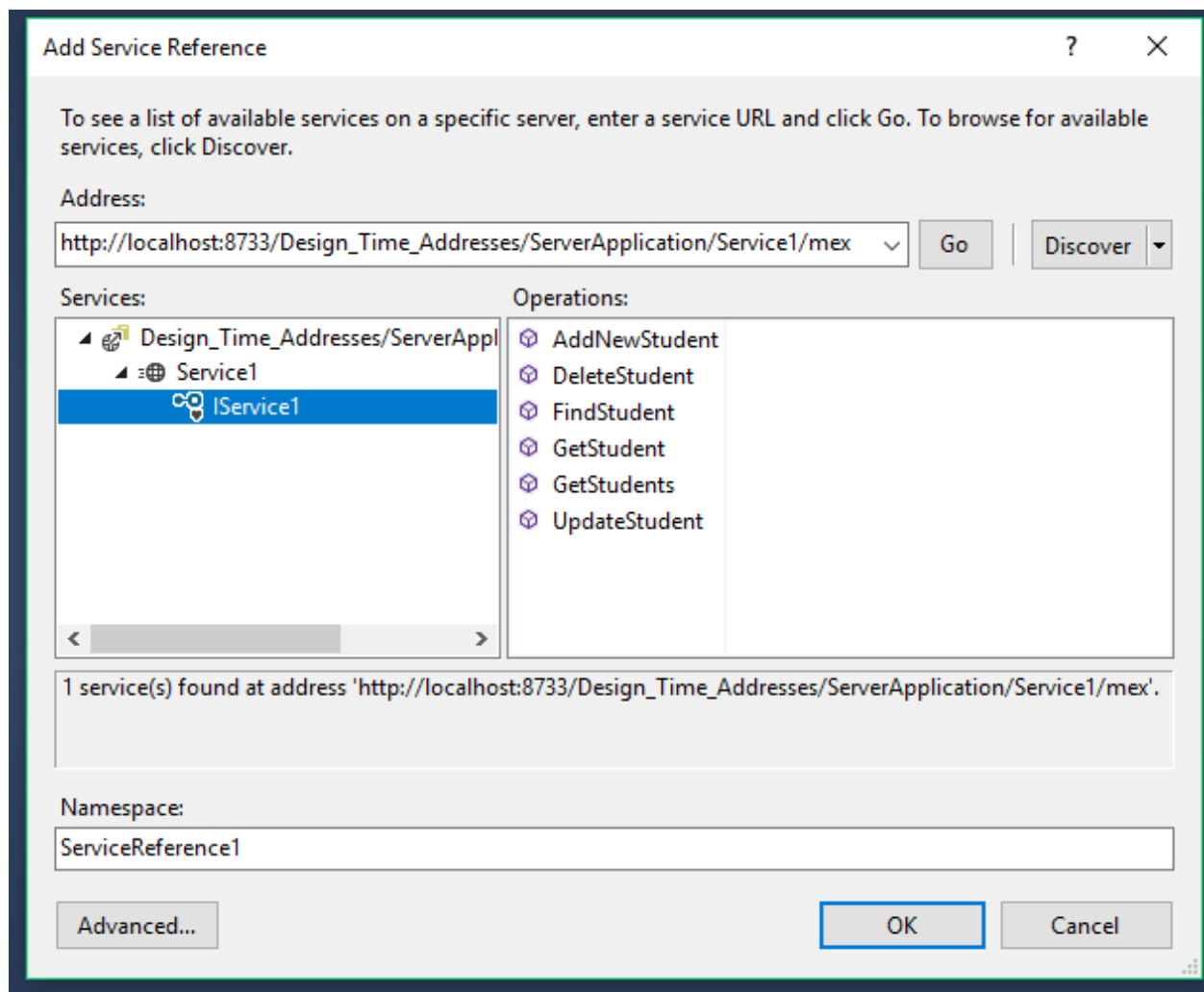
```

    }
    public bool DeleteStudent(string id)
    {
        try
        {
            Student student = db.Students.Find(id);
            db.Students.Remove(student);
            return true;
        }
        catch (Exception)
        {
            return false;
        }
    }
}
}

```

V. WCF – WebMVC

1. Trong cùng solution tạo Project ASP.NET MVC WCF_MVC
2. Discover Service



3. Tạo Empty Controller (StudentController)

4. Bổ sung code cho Controller vừa tạo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using WCF_MVC.ServiceReference1;

namespace WCF_MVC.Controllers
{
    public class StudentController : Controller
    {
        // GET: Student
        Service1Client db = new Service1Client();
        public ActionResult Index()
        {
            return View(db.GetStudents());
        }
        public ActionResult Edit(string id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Student stu = db.GetStudent(id);
            if (stu == null)
            {
                return HttpNotFound();
            }
            return View(stu);
        }
        public ActionResult Create()
        {
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"Student_Code,Student_Name,Student_Email,Student_Phone,Student_Details")] Student
student)
        {
            if (ModelState.IsValid)
            {
                bool b = db.AddNewStudent(student);
                if (b)
                {
                    return RedirectToAction("Index");
                }
                else
                {
                    return View(student);
                }
            }
            return View(student);
        }
    }
}
```

```

    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
"Student_Code,Student_Name,Student_Email,Student_Phone,Student_Details")] Student
student)
    {
        if (ModelState.IsValid)
        {
            bool b = db.UpdateStudent(student);
            if (b)
            {
                return RedirectToAction("Index");
            }
            else
            {
                return View(student);
            }
        }
        return View(student);
    }
    public ActionResult Delete(string id)
    {
        if(id==null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Student student = db.GetStudent(id);
        if(student == null)
        {
            return HttpNotFound();
        }
        return View(student);
    }
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(string id)
    {
        Student student = db.GetStudent(id);
        db.DeleteStudent(id);
        return RedirectToAction("Index");
    }
}
}

```

5. Generate View tương ứng với mỗi actionmethod và xem kết quả

○ Index

EAP

Index - My ASP.NET App

localhost:52682/Student

Application name Home About Contact

Index

[Create New](#)

Student_Code	Student_Details	Student_Email	Student_Name	Student_Phone	
A14567	Manager	natuan@cusc.ctu.edu.vn	Nguyen Anh Tuan 1	0965456789	Edit Delete
A17007	Student	0963505929	Dang Tuan Huy 12	dthuya17005@cusc.ctu.edu.vn	Edit Delete
A17008	Student	ntynhi16334@cusc.ctu.edu.vn	Nguyen Thi Yen Nhi	0963505927	Edit Delete
A17009	Student	dthuya17007@cusc.ctu.edu.vn	Nguyen Huu Hai	0963505927	Edit Delete
A17010	Student	dthuya17007@cusc.ctu.edu.vn	Tran Trung Tin	0963505927	Edit Delete
A17011	Student	dthuya17007@cusc.ctu.edu.vn	Thai Thien Tan	0963505927	Edit Delete
A17012	Student	dthuya17007@cusc.ctu.edu.vn	Nguyen Duc Trung	0963505927	Edit Delete
A17013	Monitor	phonga18334@cusc.ctu.edu.vn	Tran Trung Phong	0123746984	Edit Delete
A17014	Moniator	kieu@gmail.com	Ly Thuy Kieu	0963505987	Edit Delete
A17016	Student	dthuya17007@cusc.ctu.edu.vn	Dang Tuan Huy	0963505927	Edit Delete
A18008	Student	spnh@gmail.com	Pham Nguyen Hoang Son	0925028451	Edit Delete
A18901	No	nhau@cusc.ctu.edu.vn	Nguyen Hai Au	01236457690	Edit Delete
C16114	Null	0166457890	Nguyen Huu Hai	nhhai@cusc.ctu.edu.vn	Edit Delete

Create

Create - My ASP.NET App

localhost:52682/Student/Create

Application name Home About Contact

Create

Student

Student_Code

Student_Details

Student_Email

Student_Name

Student_Phone

[Back to List](#)

© 2018 - My ASP.NET Application

Edit

Application name Home About Contact

Edit

Student

Student_Code	<input type="text" value="A14567"/>
Student_Details	<input type="text" value="Manager"/>
Student_Email	<input type="text" value="natuan@cusc.ctu.edu.vn"/>
Student_Name	<input type="text" value="Nguyen Anh Tuan 1"/>
Student_Phone	<input type="text" value="0965456789"/>

[Back to List](#)

© 2018 - My ASP.NET Application

Delete

Application name Home About Contact

Delete

Are you sure you want to delete this?

Student

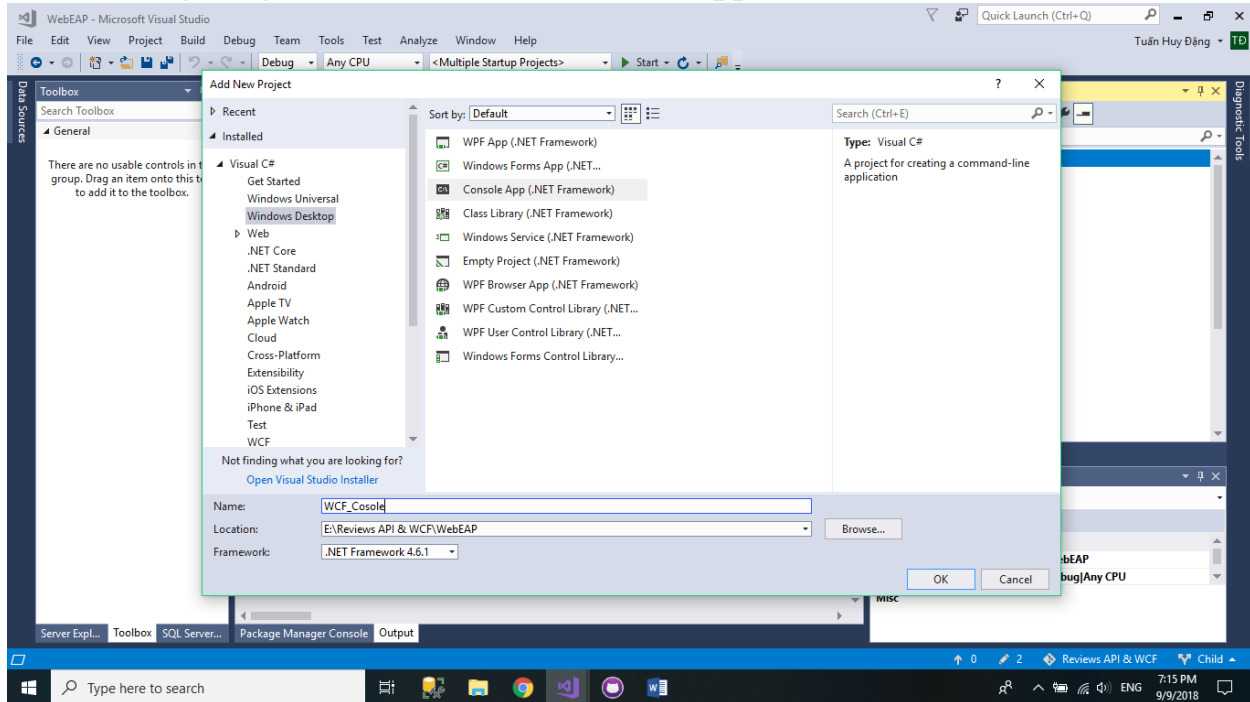
Student_Code	A17016
Student_Details	Student
Student_Email	dthuya17007@cusc.ctu.edu.vn
Student_Name	Dang Tuan Huy
Student_Phone	0963505927

| [Back to List](#)

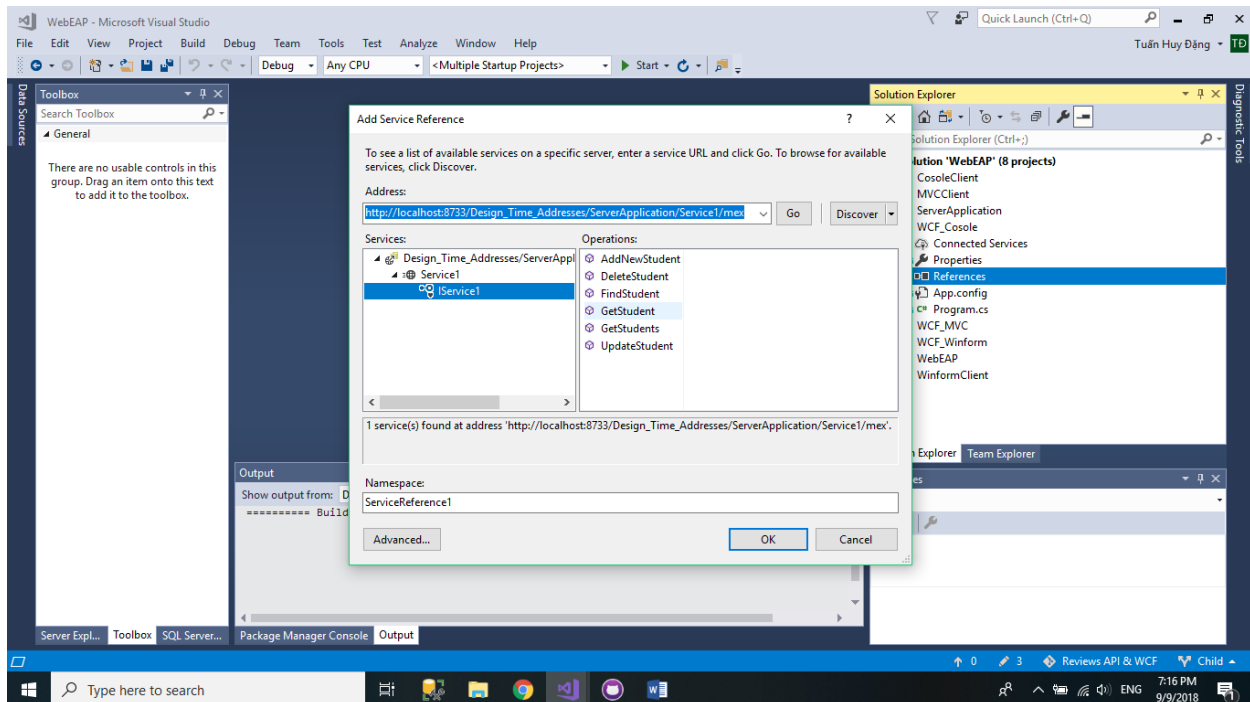
© 2018 - My ASP.NET Application

VI. WCF - Cosole

1. Trong cùng solution tạo Project Cosole App WCF_Cosole



2. Discover Service từ WCF Service



3. Trong Programs.cs bổ sung mã nguồn gọi service danh sách, thêm, sửa, xóa

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using WCF_Cosole.ServiceReference1;

namespace WCF_Cosole
{
    class Program
    {
        static Service1Client db = new Service1Client();

        //Hiển thị danh sách
        public static void GetAll()
        {
            var data = db.GetStudents();
            for (int i = 0; i < data.Length; i++)
            {
                Console.WriteLine("Students " + (i + 1) + ": ");
                Console.WriteLine("Student Code: " + data[i].Student_Code);
                Console.WriteLine("Student Name: " + data[i].Student_Name);
                Console.WriteLine("Student Email: " + data[i].Student_Email);
                Console.WriteLine("Student Phone: " + data[i].Student_Phone);
                Console.WriteLine("Student Details" + data[i].Student_Details);
            }
        }

        //Tìm Kiếm
        public static void GetStudent(string id)
        {
            var data = db.GetStudent(id);
            Console.WriteLine("Student Code: " + data.Student_Code);
            Console.WriteLine("Student Name: " + data.Student_Name);
            Console.WriteLine("Student Email: " + data.Student_Email);
            Console.WriteLine("Student Phone: " + data.Student_Phone);
            Console.WriteLine("Student Details: " + data.Student_Details);
        }

        public static void addStudent(ServiceReference1.Student student)
        {
            var add = db.AddNewStudent(student);
            if (add)
            {
                Console.WriteLine("true");
            }
            else
            {
                Console.WriteLine("false");
            }
        }

        public static void UpdateStudent(ServiceReference1.Student student)
        {
            var update = db.UpdateStudent(student);
            if (update)

```

```

        {
            Console.WriteLine("true");
        }
        else
        {
            Console.WriteLine("false");
        }
    }
}

public static void DeleteStudent(string id)
{
    var delete = db.DeleteStudent(id);
    if (delete)
    {
        Console.WriteLine("true");
    }
    else
    {
        Console.WriteLine("false");
    }
}

static void Main(string[] args)
{
    reset:
        Console.WriteLine("Select an option: ");
        Console.WriteLine("1. Get all student's information");
        Console.WriteLine("2. Get a student information");
        Console.WriteLine("3. Update student information");
        Console.WriteLine("4. Delete a student ");
        Console.WriteLine("5. Add new student");
        Console.Write("Your choice: ");

        int c = Convert.ToInt32(Console.ReadLine());
        switch (c)
        {
            case 1:
                GetAll();
                goto reset;
                break;
            case 2:
                Console.Write("Enter Student Code: ");
                // int i = Convert.ToInt32(Console.ReadLine());
                string i = Console.ReadLine();
                GetStudent(i);
                goto reset;
                break;
            case 3:
                ServiceReference1.Student students = new Student();
                Console.Write("Student Code update: ");
                //int id1 = Convert.ToInt32(Console.ReadLine());
                string id1 = Console.ReadLine();
                students.Student_Code = id1;

                Console.Write("Student Name update: ");
                string name1 = Console.ReadLine();
                students.Student_Name = name1;

                Console.Write("Student Phone update: ");

```



```

        string phone1 = Console.ReadLine();
        students.Student_Phone = phone1;

        Console.Write("Student Email update: ");
        string email1 = Console.ReadLine();
        students.Student_Email = email1;

        Console.Write("Student Details update: ");
        string details = Console.ReadLine();
        students.Student_Details = details;
        UpdateStudent(students);
        goto reset;
        break;
    case 4:
        Console.Write("Enter Student delete: ");
        //int i2 = Convert.ToInt32(Console.ReadLine());
        string i2 = Console.ReadLine();
        DeleteStudent(i2);
        goto reset;
        break;
    case 5:
        ServiceReference1.Student student = new Student();

        Console.Write("Student Code: ");
        //int id = Convert.ToInt32(Console.ReadLine());
        string id = Console.ReadLine();
        student.Student_Code = id;

        Console.Write("Lecturer Name: ");
        string name = Console.ReadLine();
        student.Student_Name = name;

        Console.Write("Student Phone: ");
        string phone = Console.ReadLine();
        student.Student_Phone = phone;

        Console.Write("Student Email: ");
        string email = Console.ReadLine();
        student.Student_Email = email;

        Console.Write("Lecturer details: ");
        string info = Console.ReadLine();
        student.Student_Details = info;

        addStudent(student);
        goto reset;
        break;
    default:
        break;
}
Console.ReadLine();
}
}

```

}

4. Kết quả

○ Danh sách

```
E:\Reviews API & WCF\WebEAP\WCF_Cosole\bin\Debug\WCF_Cosole.exe
Student Phone: dthuya17005@cusc.ctu.edu.vn
Student DetailsStudent
Students 3:
Student Code: A17008
Student Name: Nguyen Thi Yen Nhi
Student Email: ntyynhi16334@cusc.ctu.edu.vn
Student Phone: 0963505927
Student DetailsStudent
Students 4:
Student Code: A17009
Student Name: Nguyen Huu Hai
Student Email: dthuya17007@cusc.ctu.edu.vn
Student Phone: 0963505927
Student DetailsStudent
Students 5:
Student Code: A17010
Student Name: Tran Trung Tin
Student Email: dthuya17007@cusc.ctu.edu.vn
Student Phone: 0963505927
Student DetailsStudent
Students 6:
Student Code: A17011
Student Name: Thai Thien Tan
Student Email: dthuya17007@cusc.ctu.edu.vn
Student Phone: 0963505927
Student DetailsStudent
Students 7:
Student Code: A17012
Student Name: Nguyen Duc Trung
Student Email: dthuya17007@cusc.ctu.edu.vn
```

○ Thêm

```
E:\Reviews API & WCF\WebEAP\WCF_Cosole\bin\Debug\WCF_Cosole.exe
true
Select an option:
1. Get all student's information
2. Get a student information
3. Update student information
4. Delete a student
5. Add new student
Your choice: 5
Student Code: 112345
Lecturer Name: Nguyen Ngoc My
Student Phone: 0123456789
Student Email: my@gmail.com
Lecturer details: Lecture
true
Select an option:
1. Get all student's information
2. Get a student information
3. Update student information
4. Delete a student
5. Add new student
Your choice:
```

○ Sửa

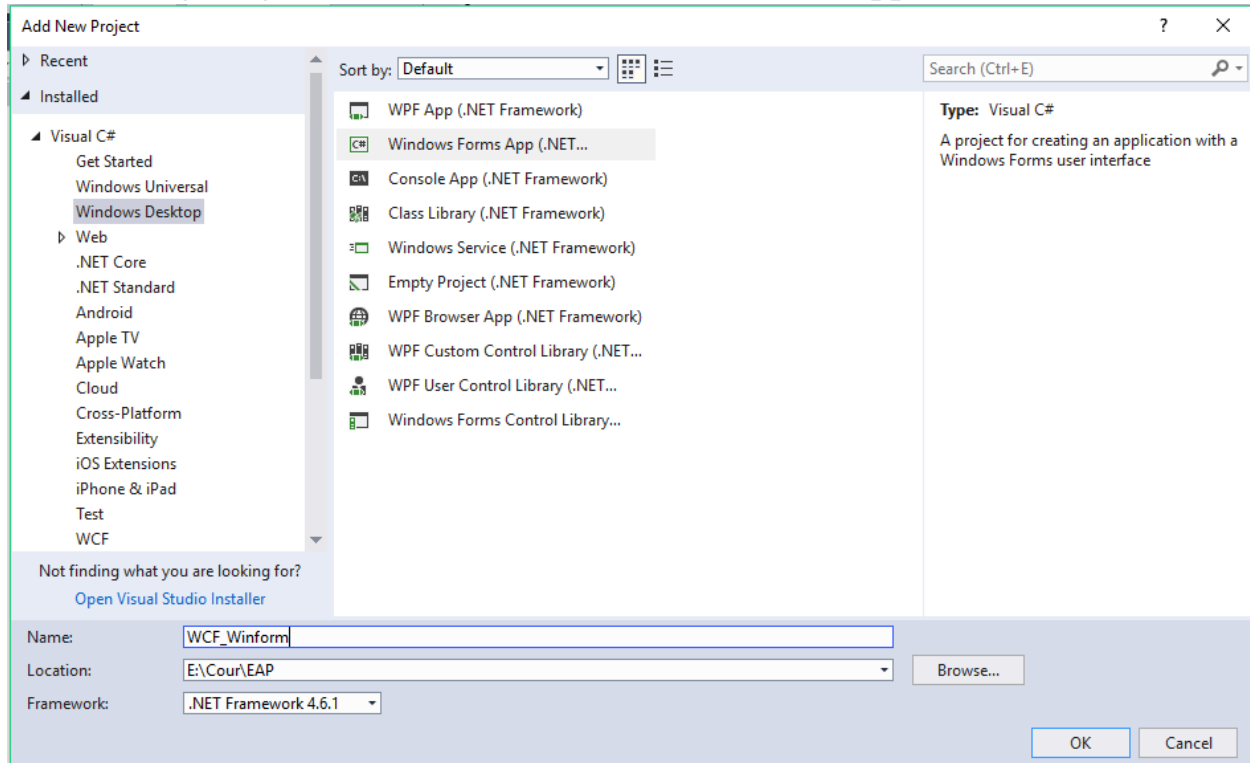
```
E:\Reviews API & WCF\WebEAP\WCF_Cosole\bin\Debug\WCF_Cosole.exe
Select an option:
1. Get all student's information
2. Get a student information
3. Update student information
4. Delete a student
5. Add new student
Your choice: 3
Student Code update: CP908
Student Name update: Hong Ha
Student Phone update: 0963505927
Student Email update: 123@gmail.com
Student Details update: Student
true
Select an option:
1. Get all student's information
2. Get a student information
3. Update student information
4. Delete a student
5. Add new student
Your choice:
```

○ Xóa

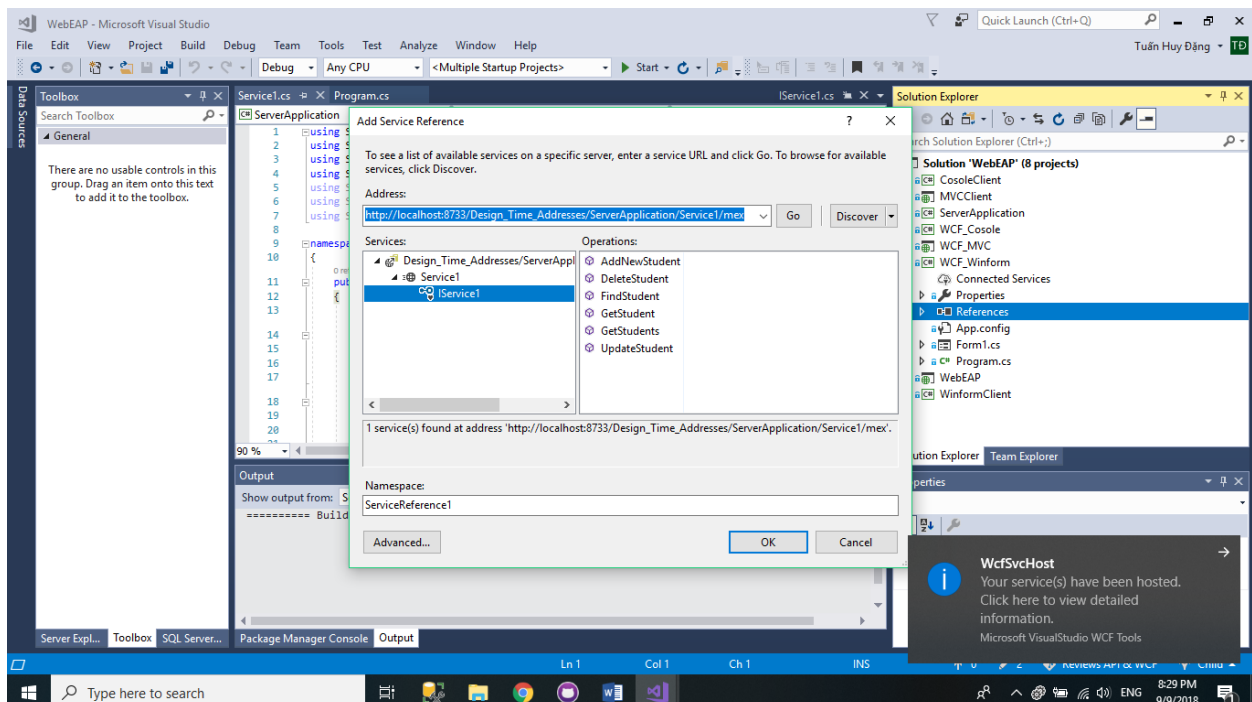
```
E:\Reviews API & WCF\WebEAP\WCF_Cosole\bin\Debug\WCF_Cosole.exe
1. Get all student's information
2. Get a student information
3. Update student information
4. Delete a student
5. Add new student
Your choice: 4
Enter Student delete: CP908
true
Select an option:
1. Get all student's information
2. Get a student information
3. Update student information
4. Delete a student
5. Add new student
Your choice:
```

VII. WCF – Winform

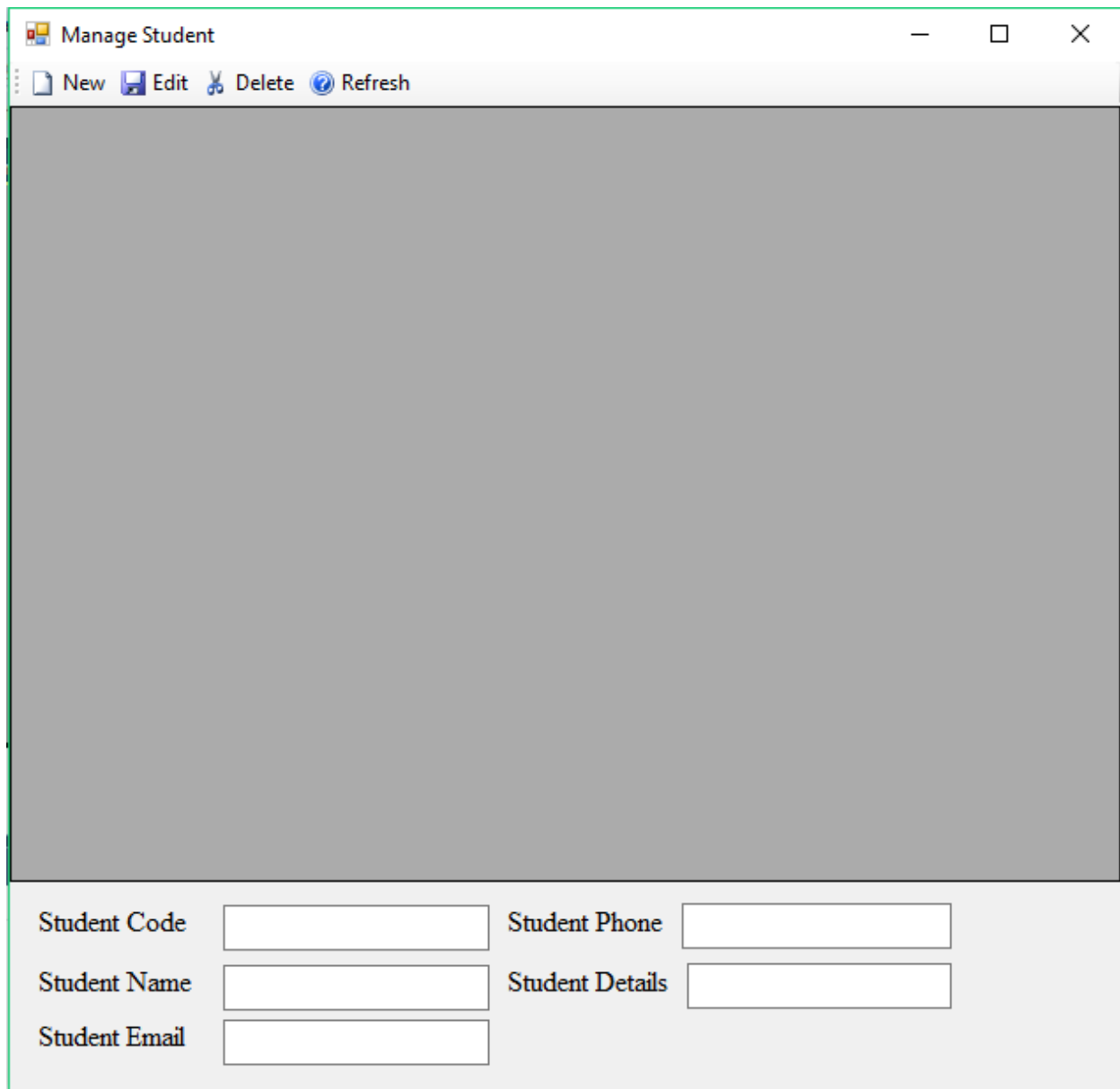
1. Trong cùng 1 solution tạo Project Windows Form App



2. Discover với WCF Service



3. Thiết Kế Form theo hình

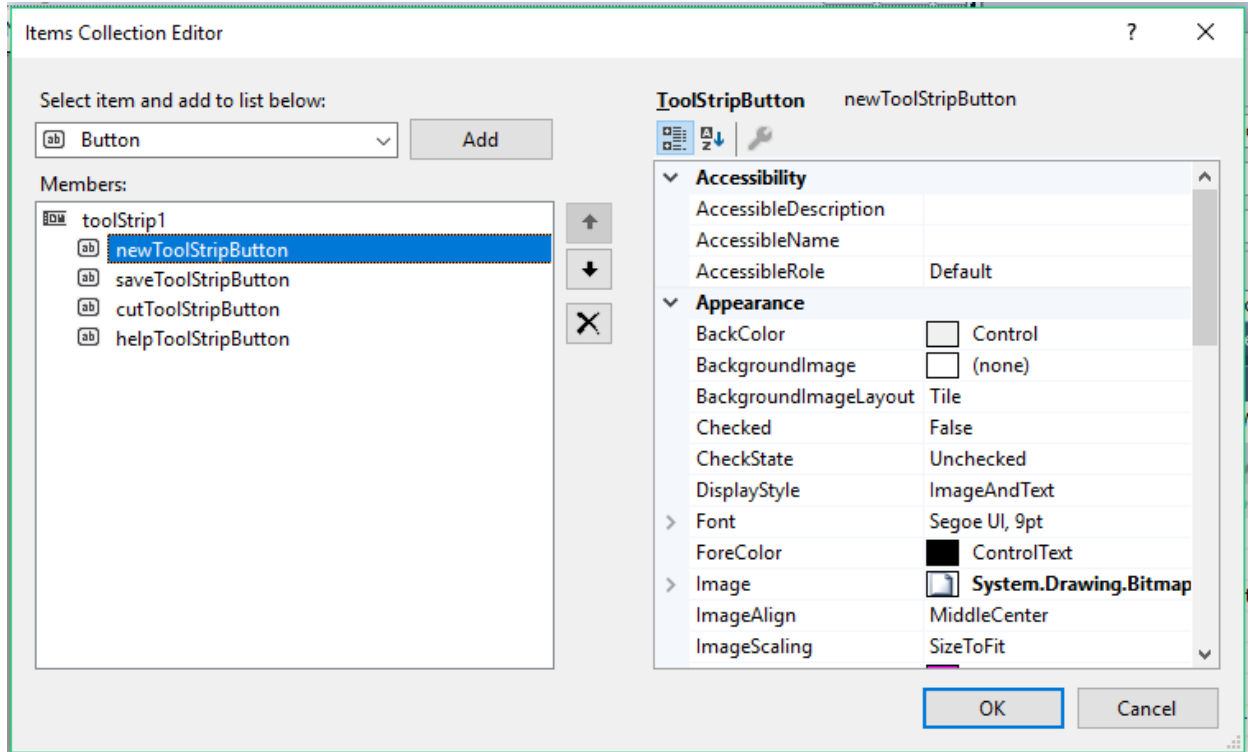


The screenshot shows a window titled "Manage Student" with a standard Windows-style title bar (minimize, maximize, close buttons). Below the title bar is a menu bar containing four items: "New" (with a document icon), "Edit" (with a pencil icon), "Delete" (with a scissors icon), and "Refresh" (with a circular arrow icon). The main area of the window is a large, empty gray rectangle. At the bottom of the window is a form with five input fields arranged in three rows:

Student Code	<input type="text"/>	Student Phone	<input type="text"/>
Student Name	<input type="text"/>	Student Details	<input type="text"/>
Student Email	<input type="text"/>		

4. Cấu hình Form

- Dùng ToolStrip



- Dùng SplitContainer

Horizontal Splitter Orientation

- DataGridView

5. Bắt sự kiện thêm sửa xóa, binding Data Source

```
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;
using WCF_Winform.ServiceReference1;

namespace WCF_Winform
{
    public partial class frmMain : Form
    {
        public frmMain()
        {
            InitializeComponent();
            try
            {
                ServiceReference1.Service1Client client = new
                ServiceReference1.Service1Client();
                dataGridView1.DataSource = client.GetStudents();
            }
        }
    }
}
```

```

        catch(Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void btnNew_Click(object sender, EventArgs e)
    {
        try
        {
            //Create new Student Demo
            txtCode.Text = txtName.Text = txtEmail.Text = txtPhone.Text =
txtDetails.Text = string.Empty;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void btnEdit_Click(object sender, EventArgs e)
    {
        try
        {
            DataGridViewRow selectedRow = dataGridView1.SelectedRows[0];
            if (selectedRow != null)
            {
                //Manual binding for demo purpose only
                txtCode.Text = selectedRow.Cells["Student_Code"].Value.ToString();
                txtName.Text = selectedRow.Cells["Student_Name"].Value.ToString();
                txtEmail.Text =
selectedRow.Cells["Student_Email"].Value.ToString();
                txtPhone.Text =
selectedRow.Cells["Student_Phone"].Value.ToString();
                txtDetails.Text =
selectedRow.Cells["Student_Details"].Value.ToString();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void btnDelete_Click(object sender, EventArgs e)
    {
        try
        {
            ServiceReference1.Service1Client client = new
ServiceReference1.Service1Client();
            if (MessageBox.Show("Are you sure you want to delete this student?",
"frmMain", MessageBoxButtons.OKCancel) == DialogResult.OK)
            {
                //int deleteStudentCode =
int.Parse(dataGridView1.SelectedRows[0].Cells["Student_Code"].Value.ToString());
                string deleteStudentCode =
dataGridView1.SelectedRows[0].Cells["Student_Code"].Value.ToString();
                if (client.DeleteStudent(deleteStudentCode))

```

```

        {
            dataGridView1.DataSource = client.GetStudents();
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void btnRefresh_Click(object sender, EventArgs e)
{
    try
    {
        ServiceReference1.Service1Client client = new
ServiceReference1.Service1Client();
        dataGridView1.DataSource = client.GetStudents();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        bool isNewStudent = true;
        foreach (DataGridViewRow item in dataGridView1.Rows)
        {
            if (item.Cells["Student_Code"].Value.ToString() == txtCode.Text)
            {
                isNewStudent = false;
                break;
            }
        }
        //Student_Code = int.Parse(txtid.Text)
        ServiceReference1.Service1Client client = new
ServiceReference1.Service1Client();
        Student student = new Student() { Student_Code = txtCode.Text,
Student_Name = txtName.Text, Student_Email = txtEmail.Text, Student_Phone =
txtPhone.Text, Student_Details = txtDetails.Text };
        if (isNewStudent ? client.AddNewStudent(student) :
client.UpdateStudent(student))
        {
            dataGridView1.DataSource = client.GetStudents();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

public void Reset1()
{

```



```

txtCode.Text = "";
txtDetails.Text = "";
txtEmail.Text = "";
txtPhone.Text = "";
txtName.Text = "";
}
private void btnReset_Click(object sender, EventArgs e)
{
    Reset1();
}
}
}

```

6. Giao diện complete

The screenshot shows a Windows application titled "Manage Student". It features a menu bar with "New", "Edit", "Delete", and "Refresh" options. Below the menu is a table with six columns: Student_Code, Student_Details, Student_Email, Student_Name, and Student_Phone. The table contains 20 rows of student data. The row with Student_Code "A17016" is selected. Below the table is a form with input fields for "Student Code", "Student Name", "Student Email", "Student Phone", and "Student Details". The "Save" and "Reset" buttons are located at the bottom right of the form.

Student_Code	Student_Details	Student_Email	Student_Name	Student_Phone
112345	Lecture_1	my@gmail.com	Nguyen Ngoc Mymr	0123456789
A14567	Manager1	natuan@cusc.ct...	Nguyen Anh Tua...	0965456789
A17007	Student	0963505929	Dang Tuan Huy 12	dthuya17005@c...
A17008	Student	ntynhi16334@cu...	Nguyen Thi Yen	0963505927
A17009	Student	dthuya17007@c...	Nguyen Huu Hai	0963505927
A17010	Student	dthuya17007@c...	Tran Trung Tin	0963505927
A17011	Student	dthuya17007@c...	Thai Thien Tan	0963505927
A17012	Student	dthuya17007@c...	Nguyen Duc Trung	0963505927
A17013	Monitor	phonga18334@c...	Tran Trung Phong	0123746984
A17014	Moniator	kieu@gmail.com	Ly Thuy Kieu	0963505987
A17016	Student	dthuya17007@c...	Dang Tuan Huy	0963505927
A18008	Teacher	spnh@gmail.com	Pham Nguyen H...	0925028451
A18901	No	nhau@cusc.ctu....	Nguyen Hai Au	01236457890
C16114	Null	0166457890	Nguyen Huu Hai	nhhai@cusc.ctu....
CP908	Student	123@gmail.com	Hong Ha	0963505927
R456731	WS	0963505925	Nguyen Hong	nhng@gmail.com
WCF	DF	huy@gmail.com	Dang Tuan Huy	0935028450

Student Code: A17016 Student Phone: 0963505927

Student Name: Dang Tuan Huy Student Details: Student

Student Email: dthuya17007@cusc.ctu

Save Reset

Thi tốt nhé ae

Link Source: https://github.com/dangtuanhuy/EAP_Reviews

Chạy được Project yêu cầu VS 2017 Pro