

Bài Thực hành môn
Chuyên đề Ngôn Ngữ Lập Trình KHMT
Biên soạn: Nguyễn Bá Diệp

I. Mục lục

Contents

I.	Mục lục	1
II.	Mục tiêu	2
III.	Nội dung.....	2
1.	Khởi động dự án	2
2.	Tạo form test	8
3.	Thêm menu chức năng.....	12
IV.	Database	13
1.	Kết nối đến máy chủ cơ sở dữ liệu Database Connection (Microsoft SQL Server)	13
2.	Kết nối cơ sở dữ liệu dạng file – không cần đến server - Database File Connection (Microsoft SQL Server Database File).....	16
3.	DataBinding	20
V.	Crystal Report	33
1.	Cài đặt crystal report plugin và control cho visual studio	33
2.	Sinh viên thực hiện kết nối đến cơ sở dữ liệu (hoặc file cơ sở dữ liệu – Mục IV) và tiến hành tạo đối tượng crystal report.....	36
3.	Hiển thị report trên window form	41
VI.	Source Control Version với Git	44
VII.	Bài tập 1 - Lập trình C# với Console Application	45
VIII.	Bài tập 2 - Ứng dụng Windows Form	50
IX.	Bài tập 3 - Ứng dụng Windows Form2.....	56
1.	Bài 1:.....	56
2.	Bài 2:.....	67
X.	Bài tập 4 Truy cập cơ sở dữ liệu với ADO.NET	70

1.	Bài 1:	70
2.	Bài 2:	79
3.	Bài 3:	81
XI.	Bài tập 5 Truy cập Cơ sở dữ liệu với ADO.NET (Tiếp theo)	83
XII.	Bài tập 6	88
XIII.	Bài tập 7: TẠO BÁO BIỂU VỚI CRYSTAL REPORT	91

II. Mục tiêu

Hướng dẫn và gợi ý sinh viên thực hành theo hướng dự án (Project base). Sau khi thực hành sinh viên có thể chỉnh sửa code, module cho phù hợp với các dự án sinh viên dự định thực hiện.

Trong quá trình thực hành, sinh viên nên thường xuyên lưu bài, lưu trữ các phiên bản khác nhau của dự án:

- Phương pháp thủ công: copy thư mục, paste và đổi tên thư mục, project

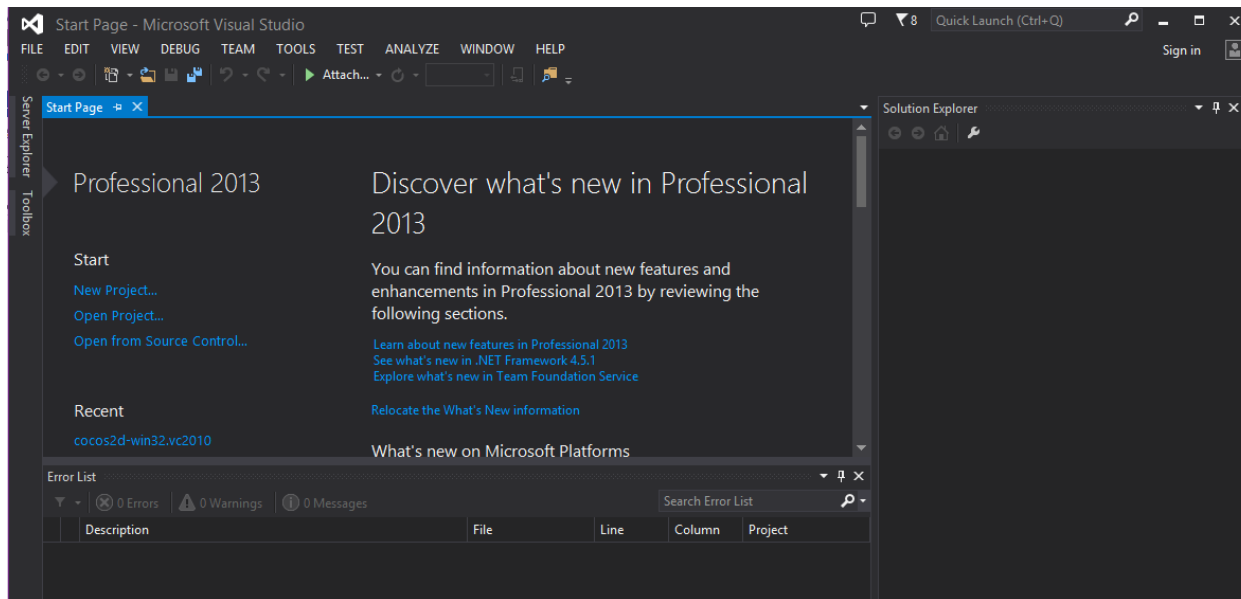
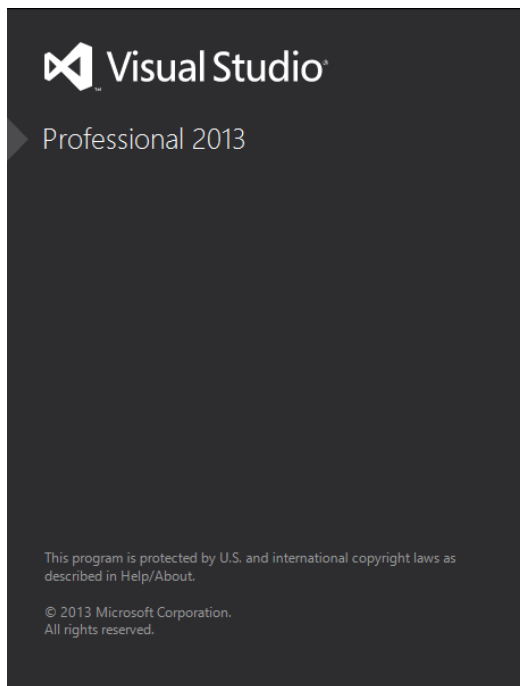
Phương pháp công cụ local/offline: như tortoisSVN (miễn phí – online/offline), Team Foundation Server (tích hợp trong visual studio – online/offline)

- Phương pháp công cụ online: visualSVN (bản quyền, trả phí) github tool – Github Server (miễn phí, online)

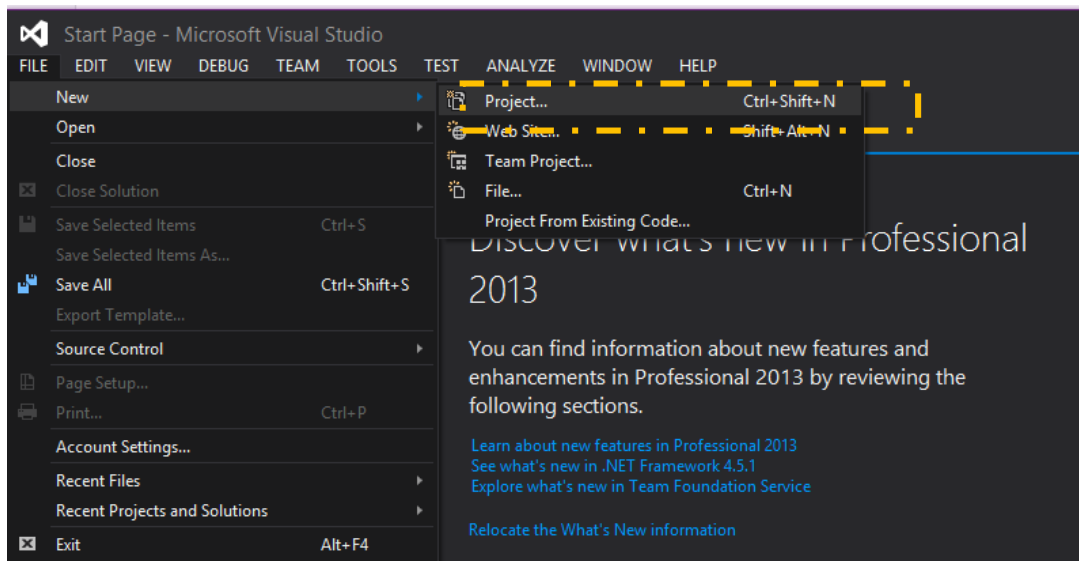
Nếu project làm việc theo nhóm với nhiều thành viên khác nhau thì Github Server/tool là giải pháp phù hợp với sinh viên do cộng đồng hỗ trợ lớn, miễn phí, phù hợp với việc chia sẻ code của các thành viên trong nhóm.

III. Nội dung

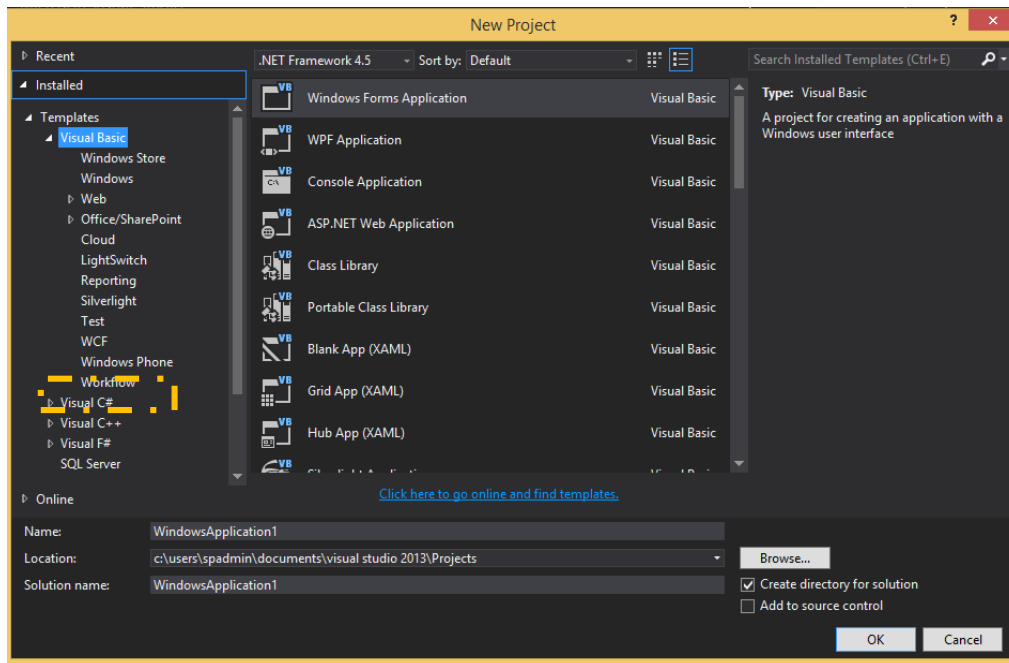
1. Khởi động dự án



Tạo mới dự án: File -> New -> Project



Chọn dự án của visual C#

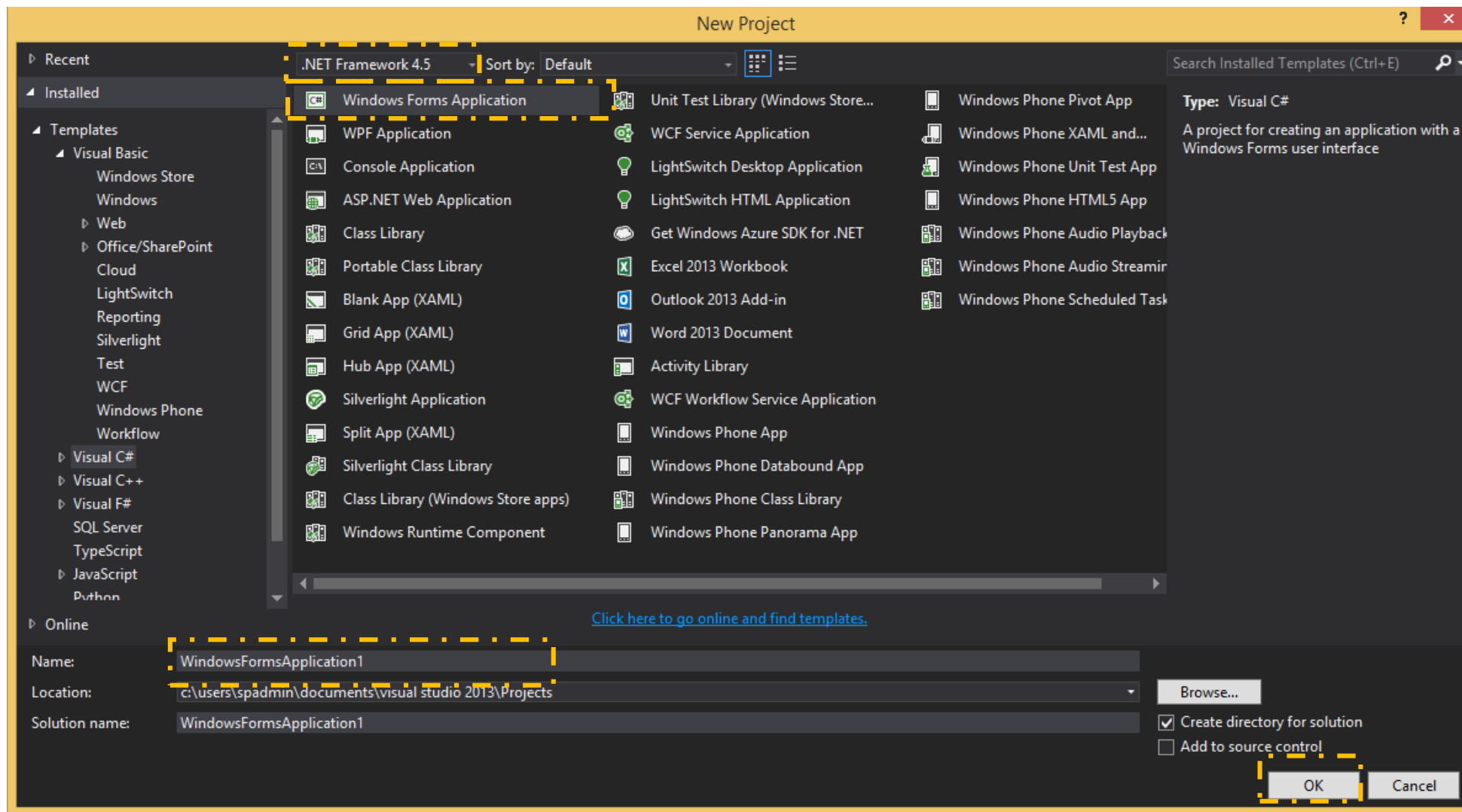


1: Chọn .NET Framework phù hợp

2: Chọn loại template cho dự án là Windows Form Application (hoặc Windows Application với các phiên bản visual studio nhỏ hơn 2013)

3: Đặt tên cho Project

4: Click OK hoặc thiết lập thêm các thuộc tính khác như đường dẫn, thêm vào công cụ quản lý phiên bản source code (source control)

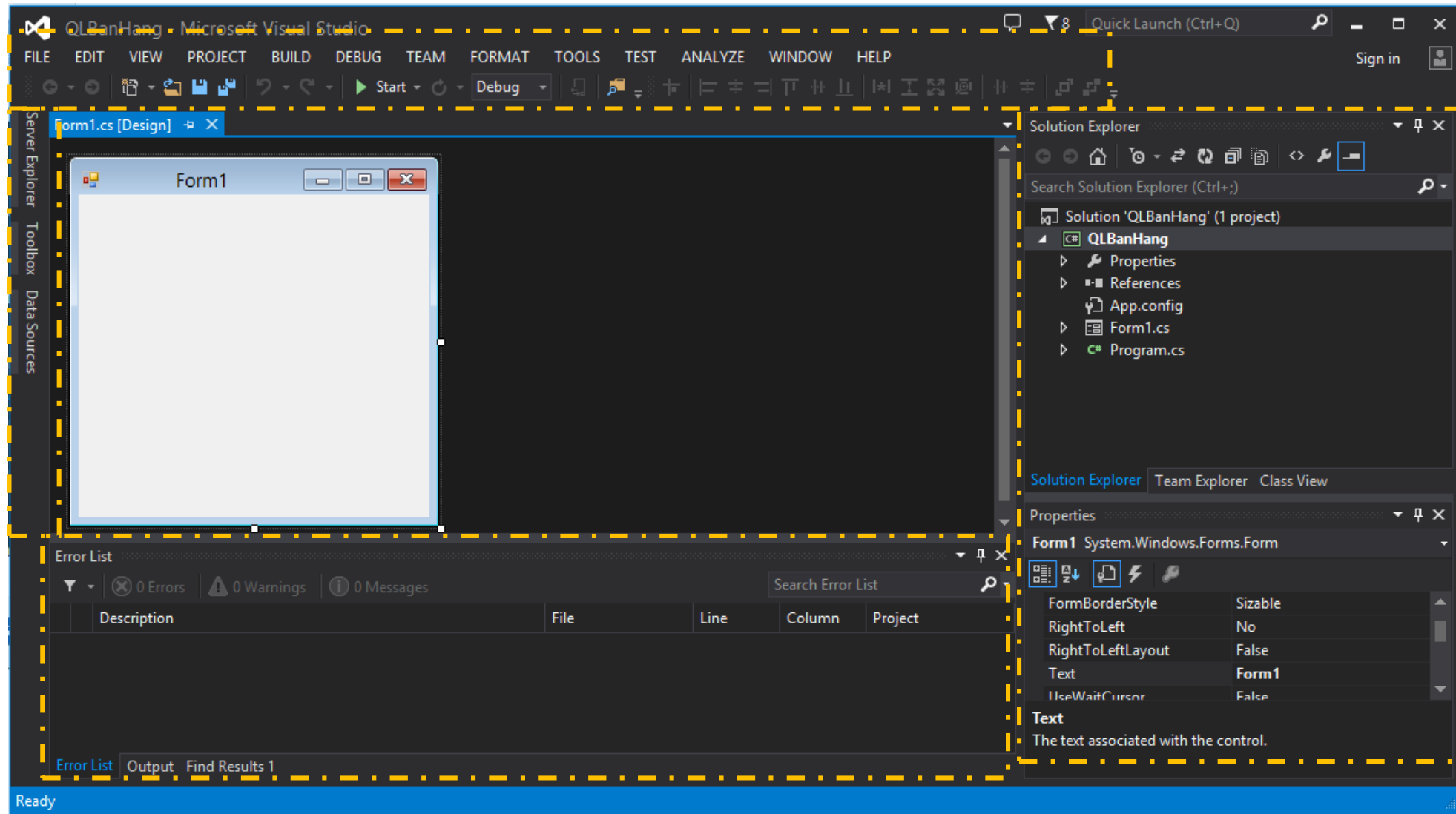


Cửa sổ chính của dự án bao gồm các thành phần:

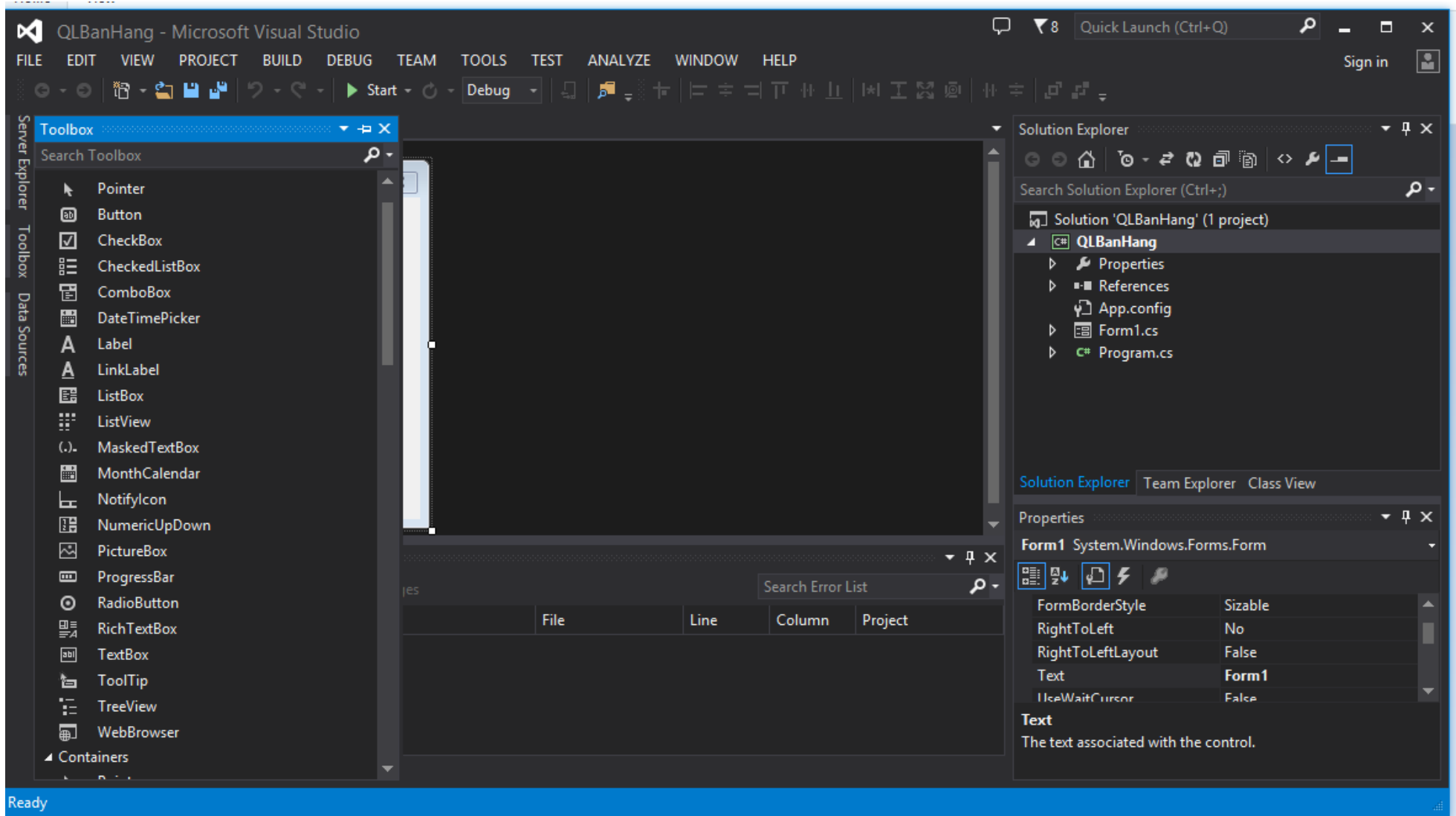
Phía trên: Menu chức năng và tool bar hỗ trợ, các mục của tool bar có thể thêm, di chuyển và xóa bớt

Phía phải, trái và bên dưới: Cửa sổ tool, hiển thị rõ hơn chức năng của tool hỗ trợ, có thể để chế độ tự động ẩn, ẩn, và luôn mở (kích hoạt bằng nút nhấn hình chuông), các tool này cũng có thể kết hợp với nhau thành các tab hoặc chia nhỏ khung nhìn trong cửa sổ (View).

Thông thường phía bên dưới dùng để hiển thị lỗi, quá trình biên dịch, các giao tiếp dòng lệnh (console/ terminal) và debug.

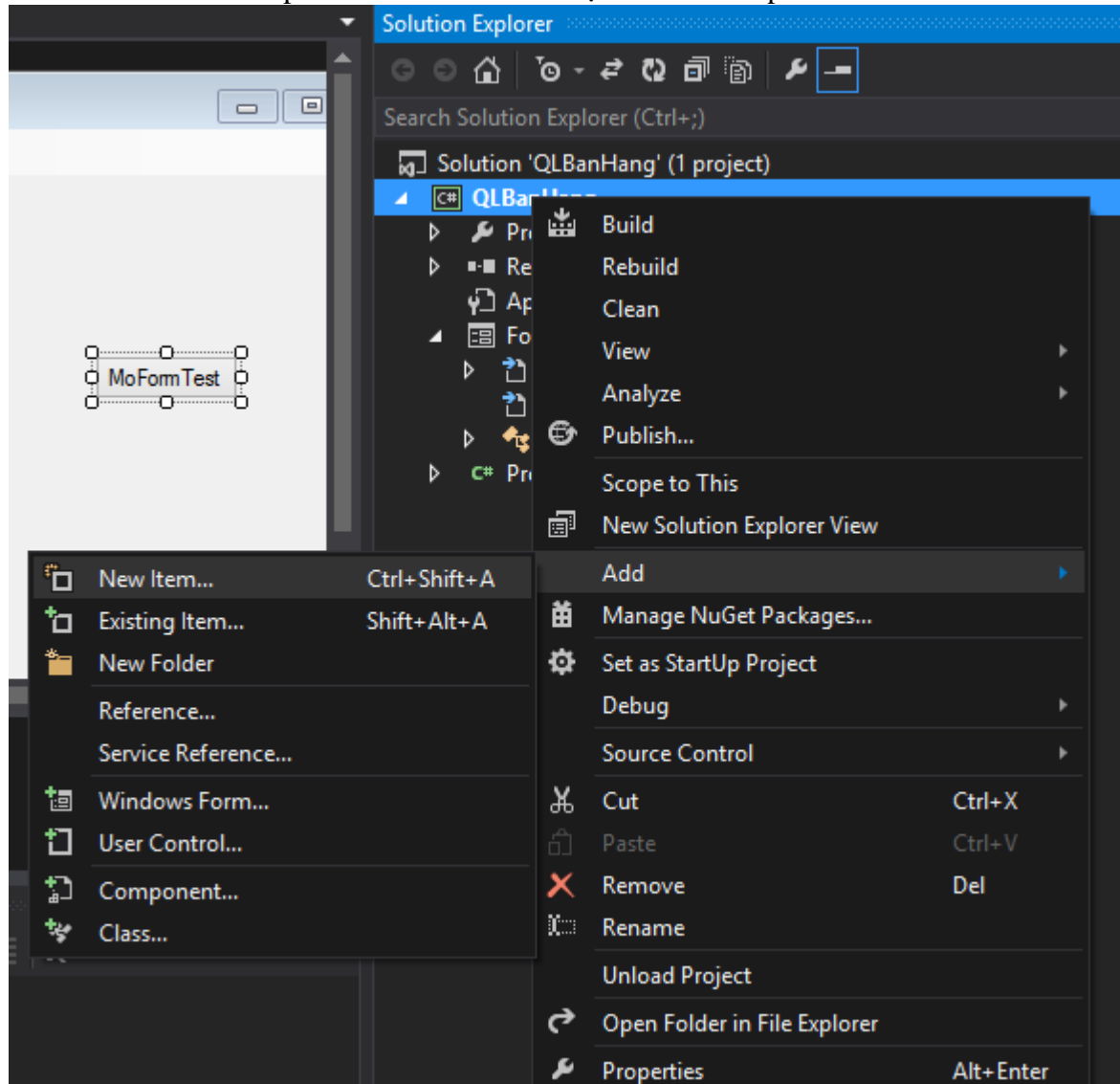


Khi các cửa sổ tool ẩn bị click vào sẽ phóng to kích thước phù hợp hơn (VD mở cửa sổ Toolbox như hình)

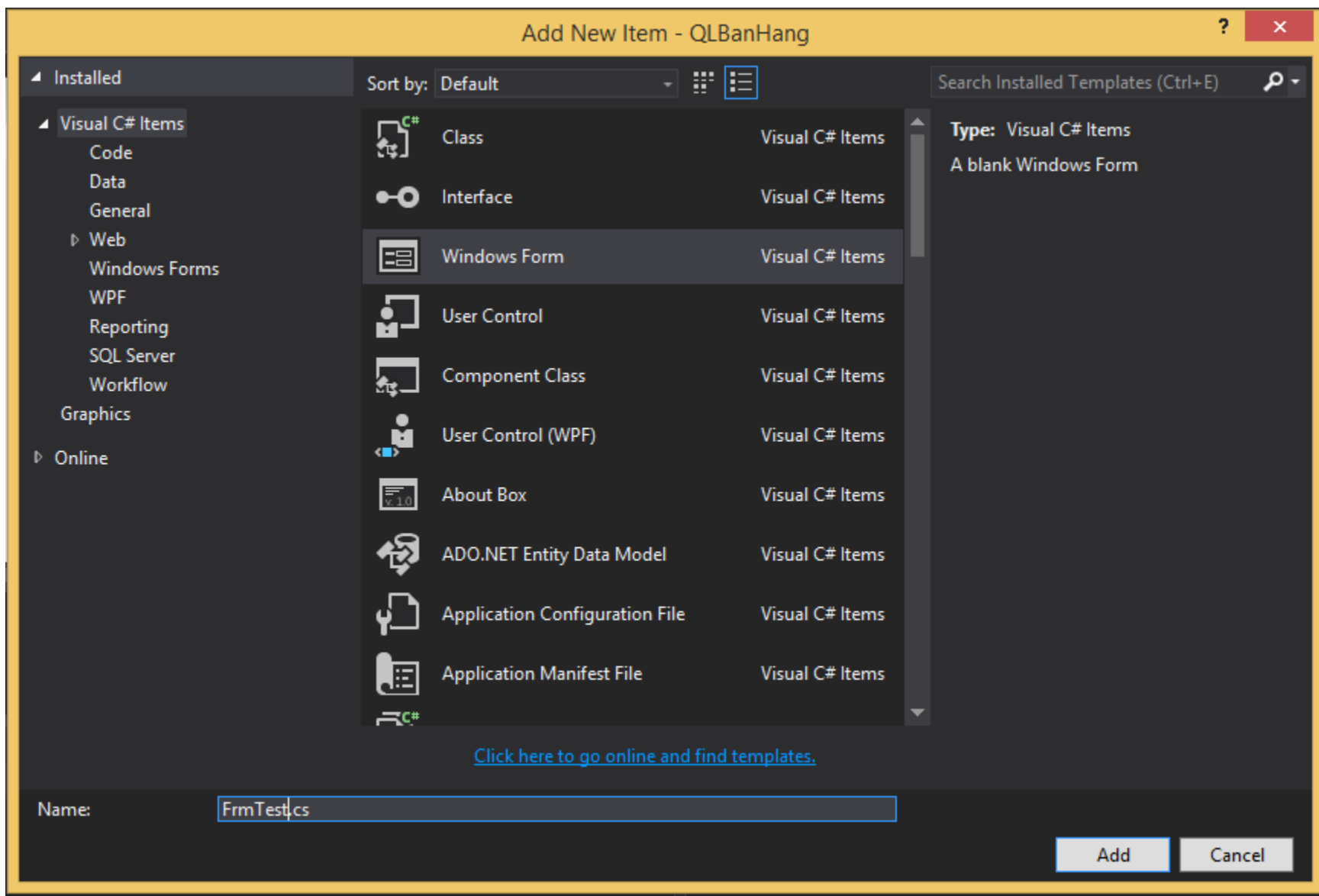


2. Tạo form test

Từ cửa sổ Solution Explorer -> Solution “tên dự án” -> Click phải -> Add -> New Item



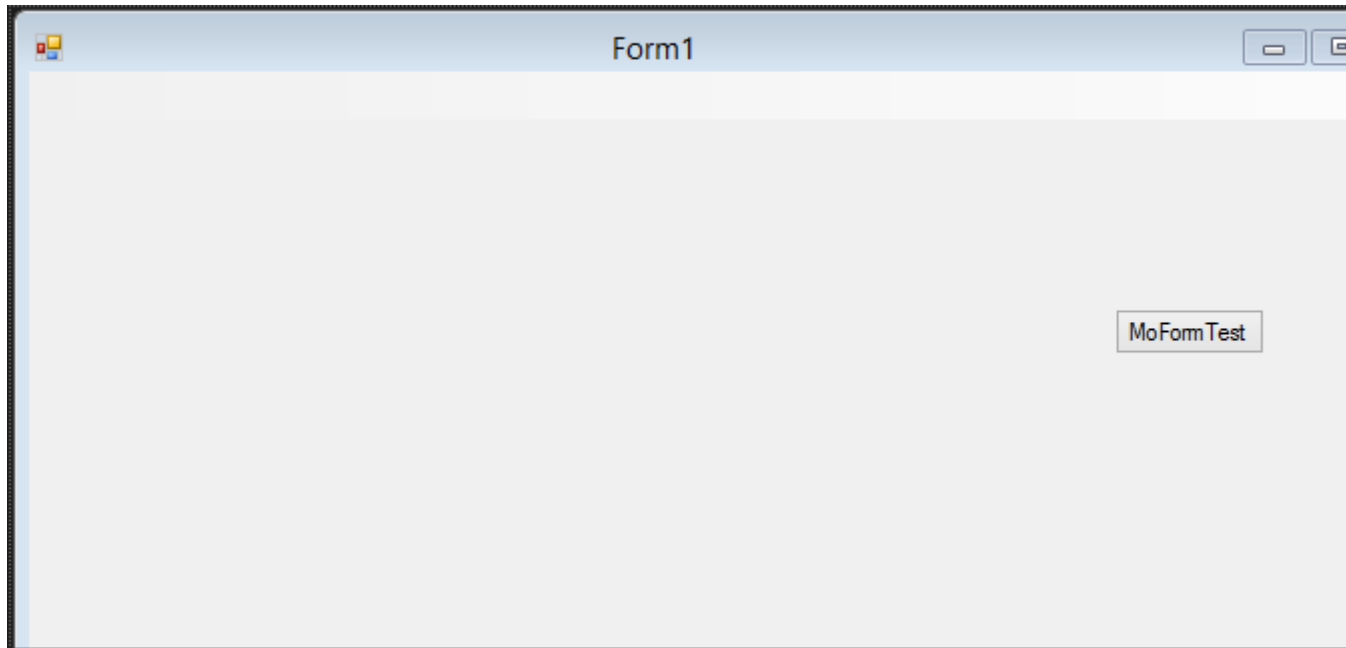
Chọn Loại Item là Windows Form-> đặt tên cho form mới-> Add

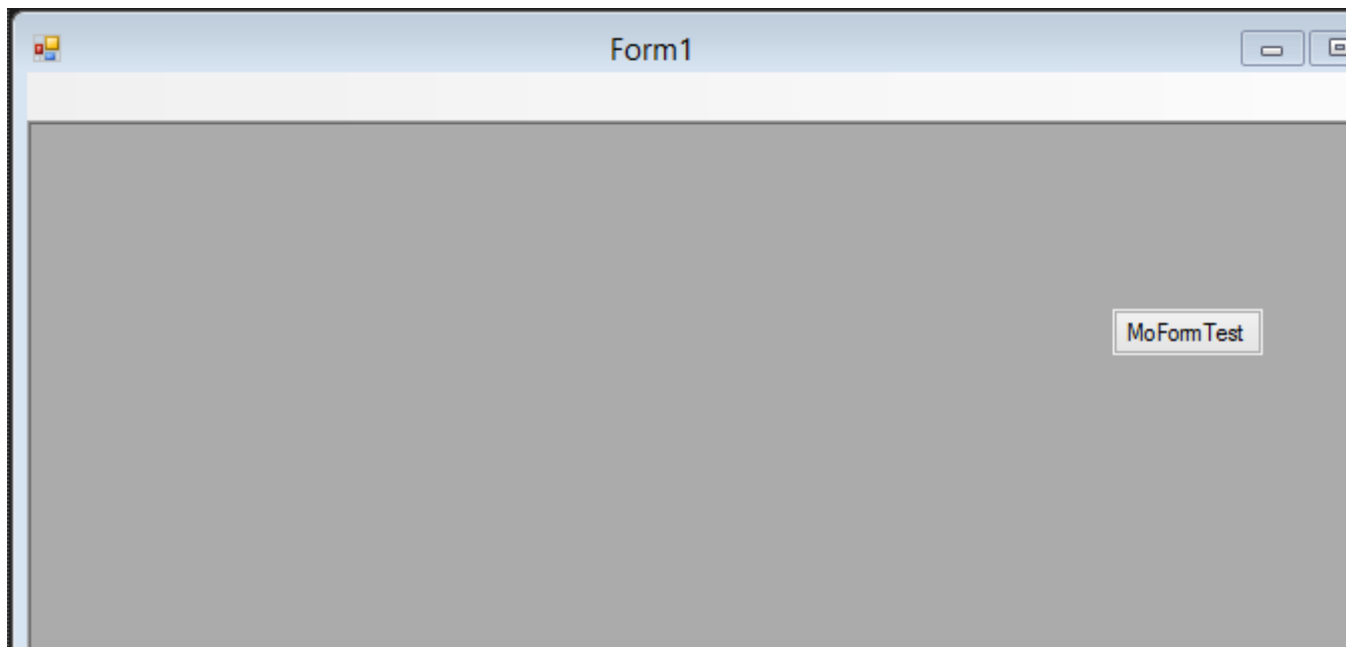


Chỉ thuộc tính của Form chính là **IsMdiContainer** = true

Icon	(Icon)
ImeMode	NoControl
IsMdiContainer	False

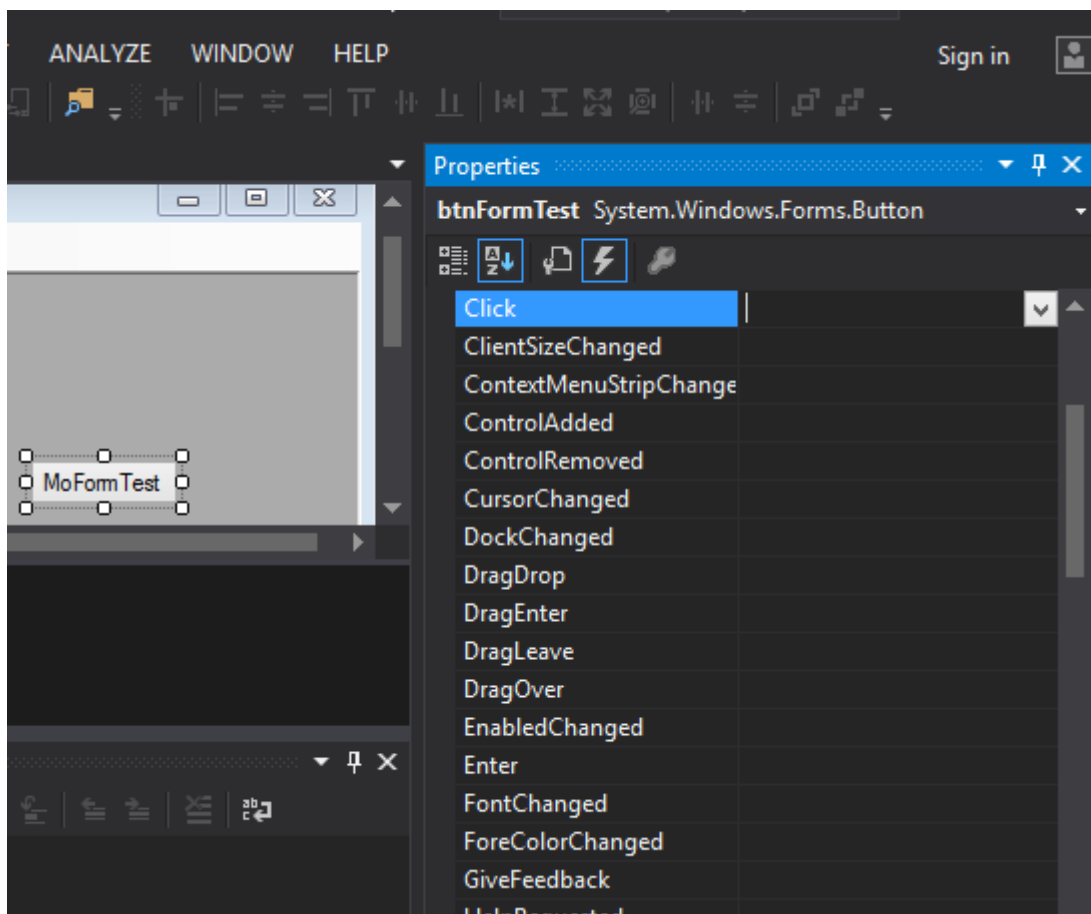
Form chính sẽ thay đổi





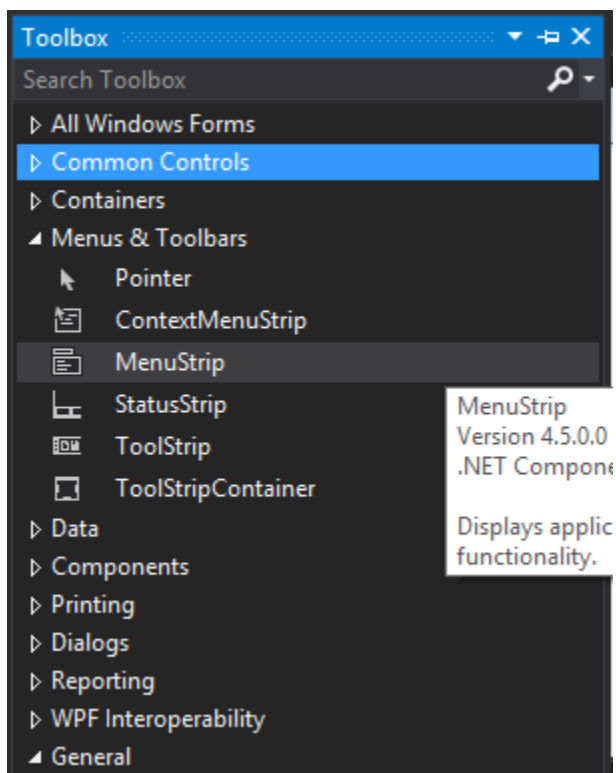
Tạo 1 button nhấn trong form chính

Add sự kiện click cho button và thêm các dòng code sau:



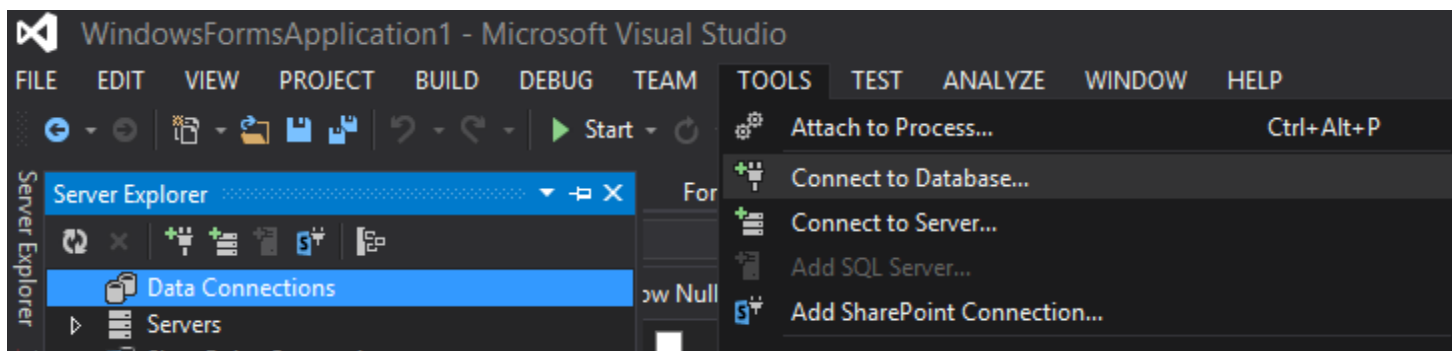
```
private void btnFormTest_Click(object sender, EventArgs e)
{
    FrmTest myFrmTest = new FrmTest();
    myFrmTest.MdiParent = this;
    myFrmTest.Show();
}
```

3. Thêm menu chức năng



IV. Database

1. Kết nối đến máy chủ cơ sở dữ liệu Database Connection (Microsoft SQL Server)



Change Data Source ? X

Data source:

- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Database File
- Oracle Database
- <other>

Description

Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.

Data provider:

.NET Framework Data Provider for SQL S ▼

☐ Always use this selection

OK Cancel

Add Connection?×

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Microsoft SQL Server (SqlClient)

Change...

Server name:

DELL

▼

Refresh

Log on to the server

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:

▼

☐ Attach a database file:

Browse...

Logical name:

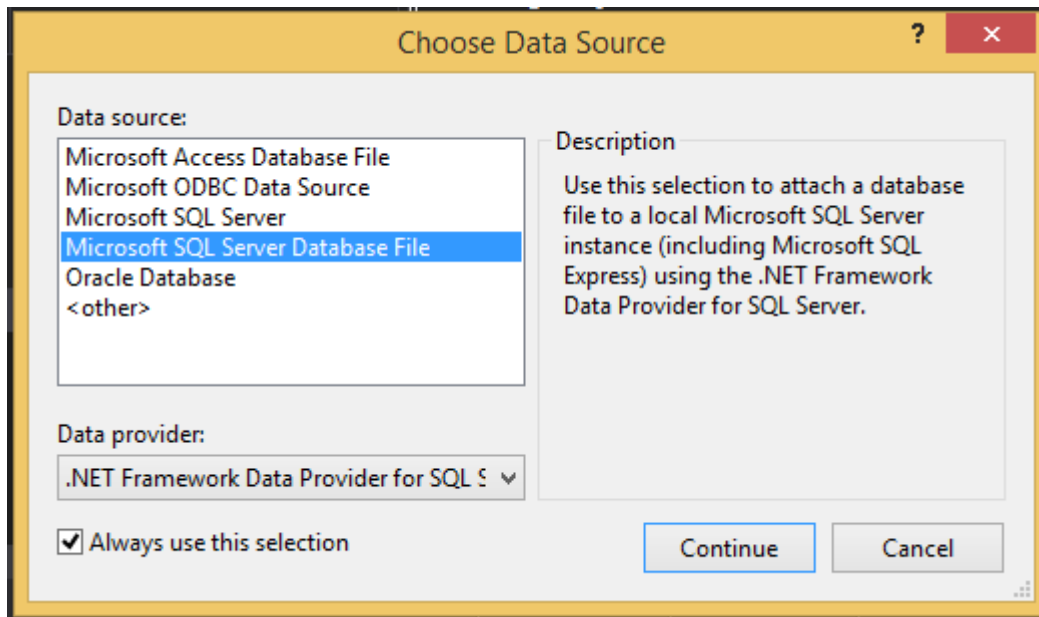
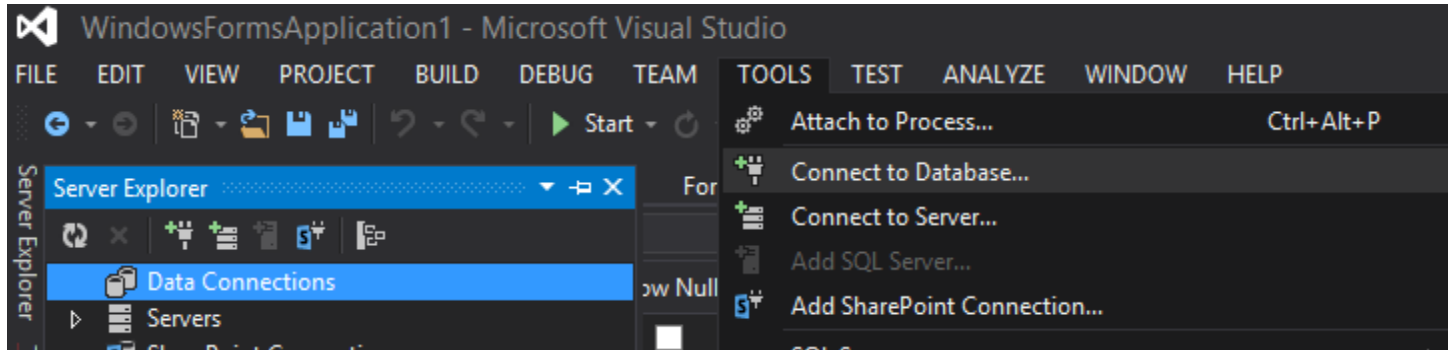
Advanced...

Test Connection

OK

Cancel

2. Kết nối cơ sở dữ liệu dạng file – không cần đến server - Database File Connection (Microsoft SQL Server Database File)



Chọn đường dẫn đến file database và Click nút OK

Add Connection ? x

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server Database File (SqlClient) Change...

Database file name (new or existing):
DBTest Browse...

Log on to the server

☒ Use Windows Authentication
☐ Use SQL Server Authentication

User name:
Password:
☐ Save my password

Advanced...

Test Connection OK Cancel

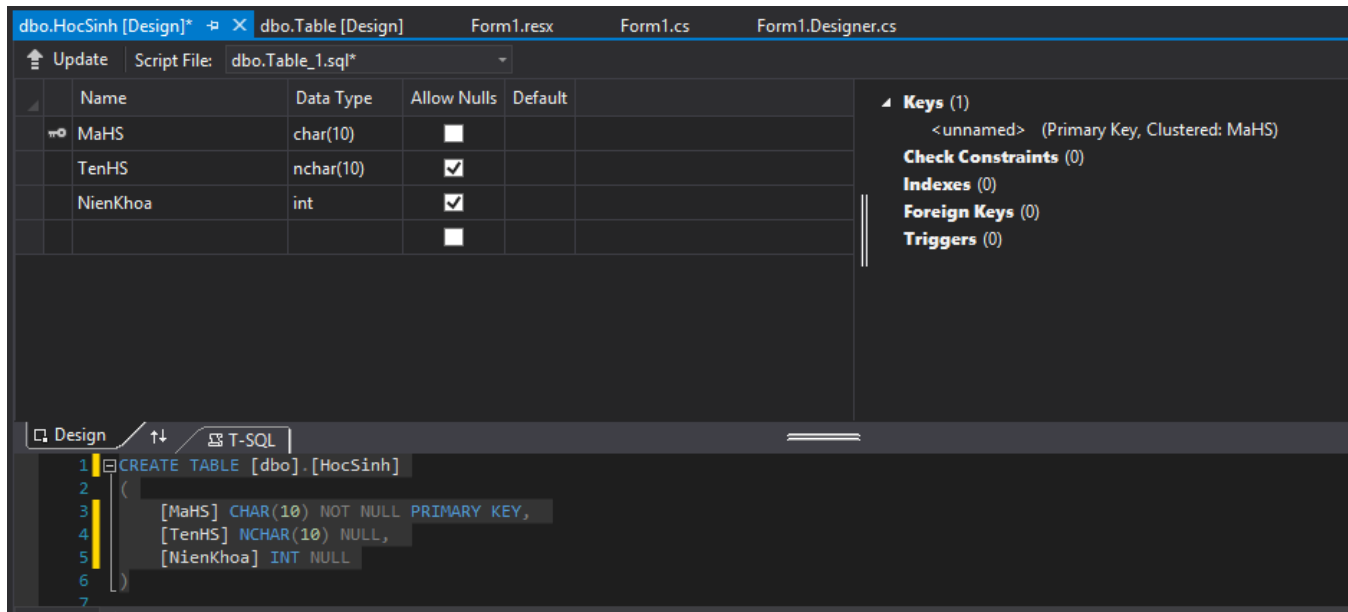
Nếu file database chưa có, chương trình sẽ hiện hộp thoại tạo mới

Microsoft Visual Studio x

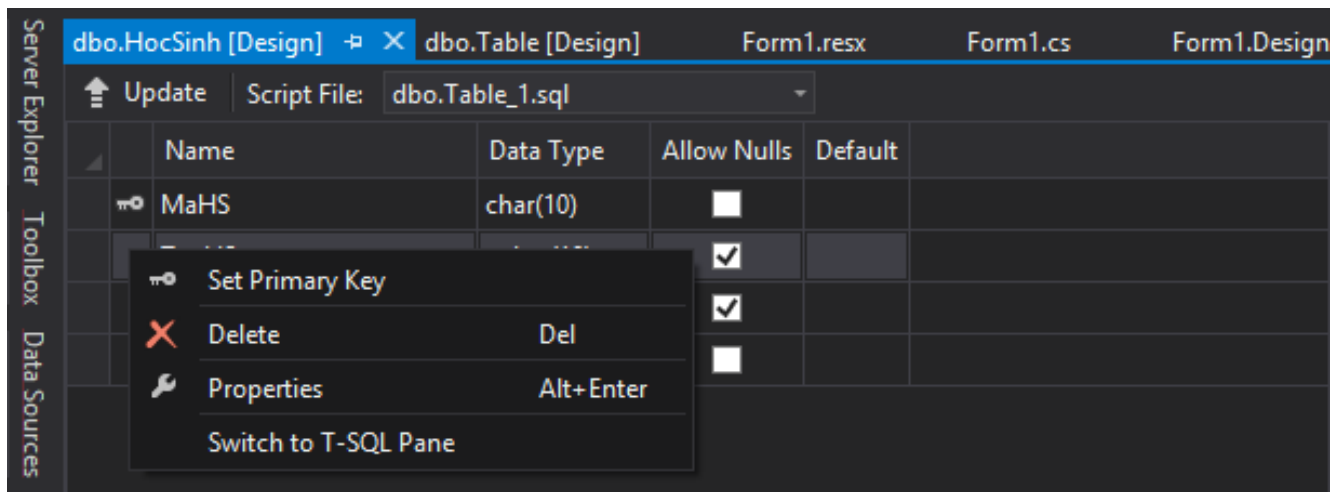
? The database file 'C:\Users\SPadmin\Documents\DBTest.mdf' does not exist.

Would you like to create it?

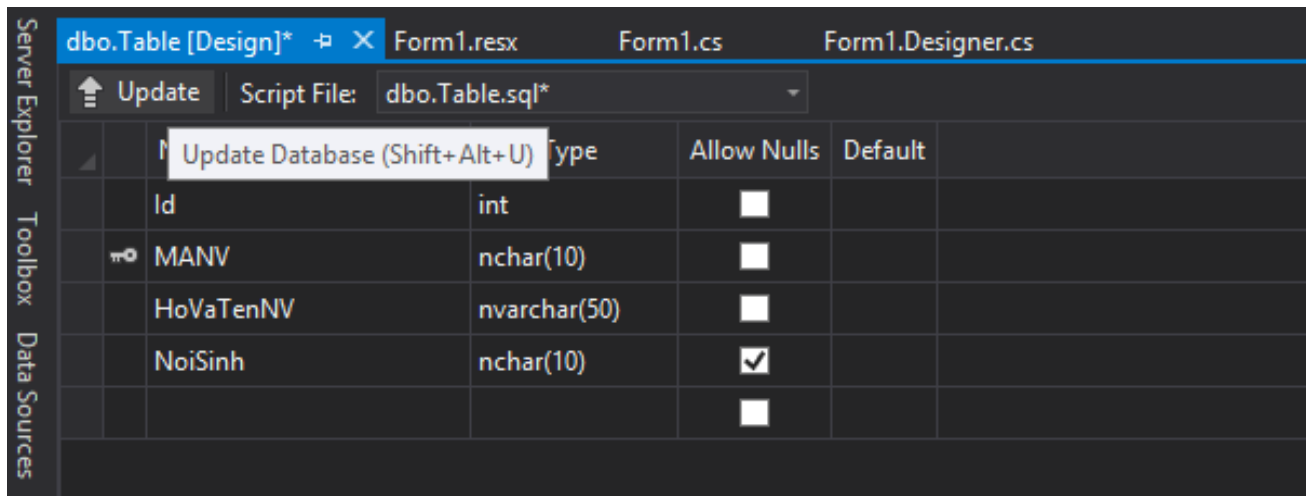
Yes No



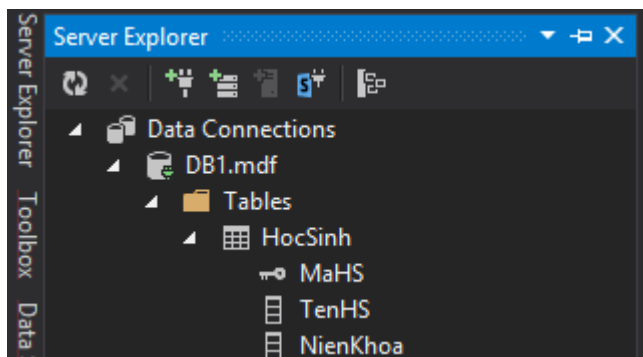
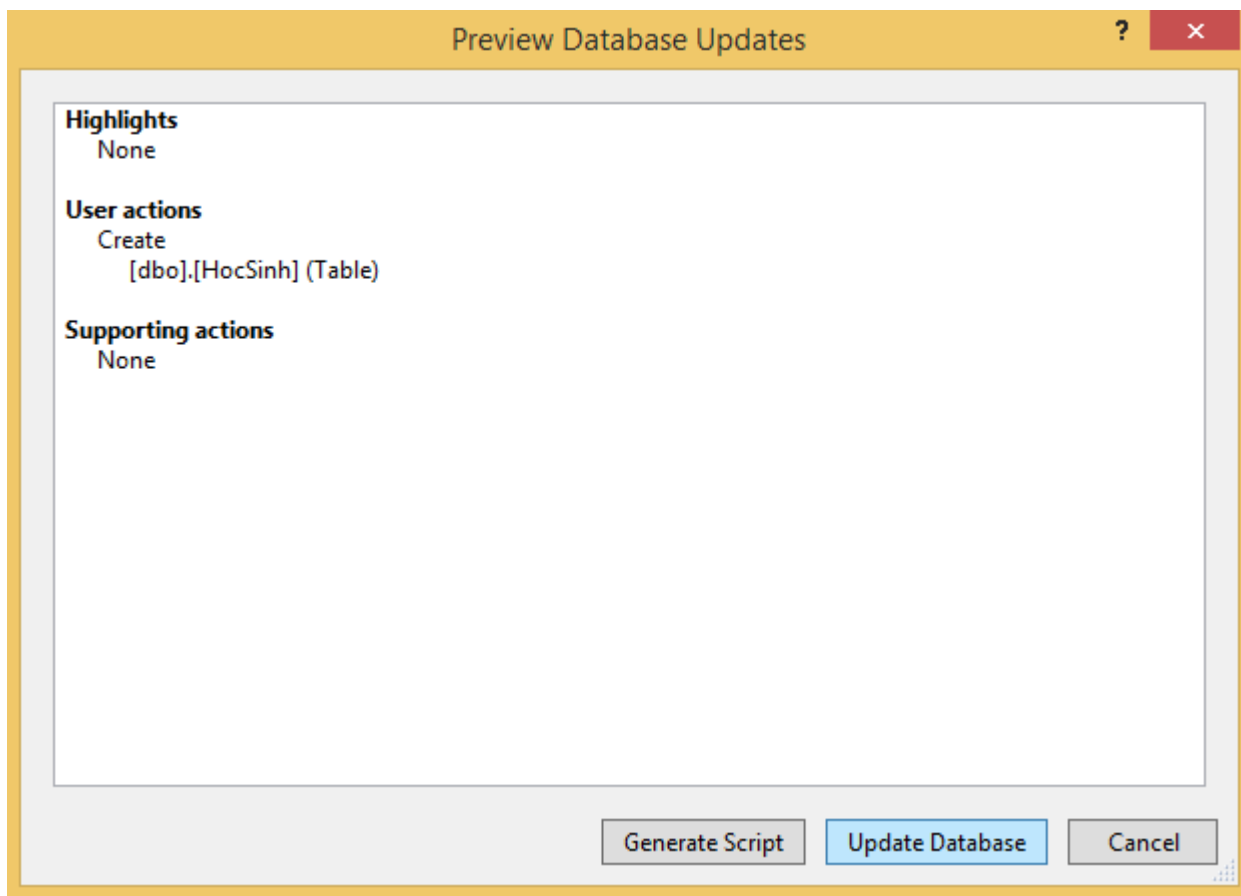
Thiết lập khoá chính cho bảng



Thiết lập xong → chọn Update

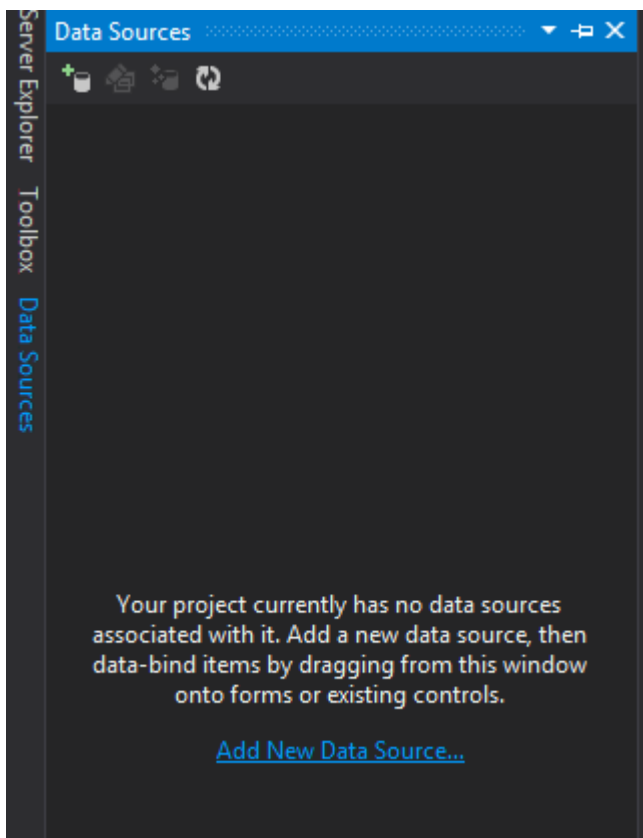


Chọn Update database



3. DataBinding

Thêm các kết nối DataSource





Choose a Data Source Type

Where will the application get data from?



Database



Service



Object



SharePoint

Lets you connect to a database and choose the database objects for your application.

< Previous

Next >

Finish

Cancel



Choose a Database Model

What type of database model do you want to use?



Dataset

The database model you choose determines the types of data objects your application code uses. A dataset file will be added to your project.

< Previous

Next >

Finish

Cancel

**Choose Your Data Connection**

Which data connection should your application use to connect to the database?

DB1.mdf



New Connection...

This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

- ☐ No, exclude sensitive data from the connection string. I will set this information in my application code.
- ☐ Yes, include sensitive data in the connection string.

☐ Connection string that you will save in the application (expand to see details)

Data Source=(LocalDB)\v11.0;AttachDbFilename="E:\CTU CIT\Giang day\CT170 CD NNLT\Project QLBH
\WindowsFormsApplication1\DB1.mdf";Integrated Security=True;Connect Timeout=30

< Previous

Next >

Finish

Cancel

**Save the Connection String to the Application Configuration File**

Storing connection strings in your application configuration file eases maintenance and deployment. To save the connection string in the application configuration file, enter a name in the box and then click Next.

Do you want to save the connection string to the application configuration file?

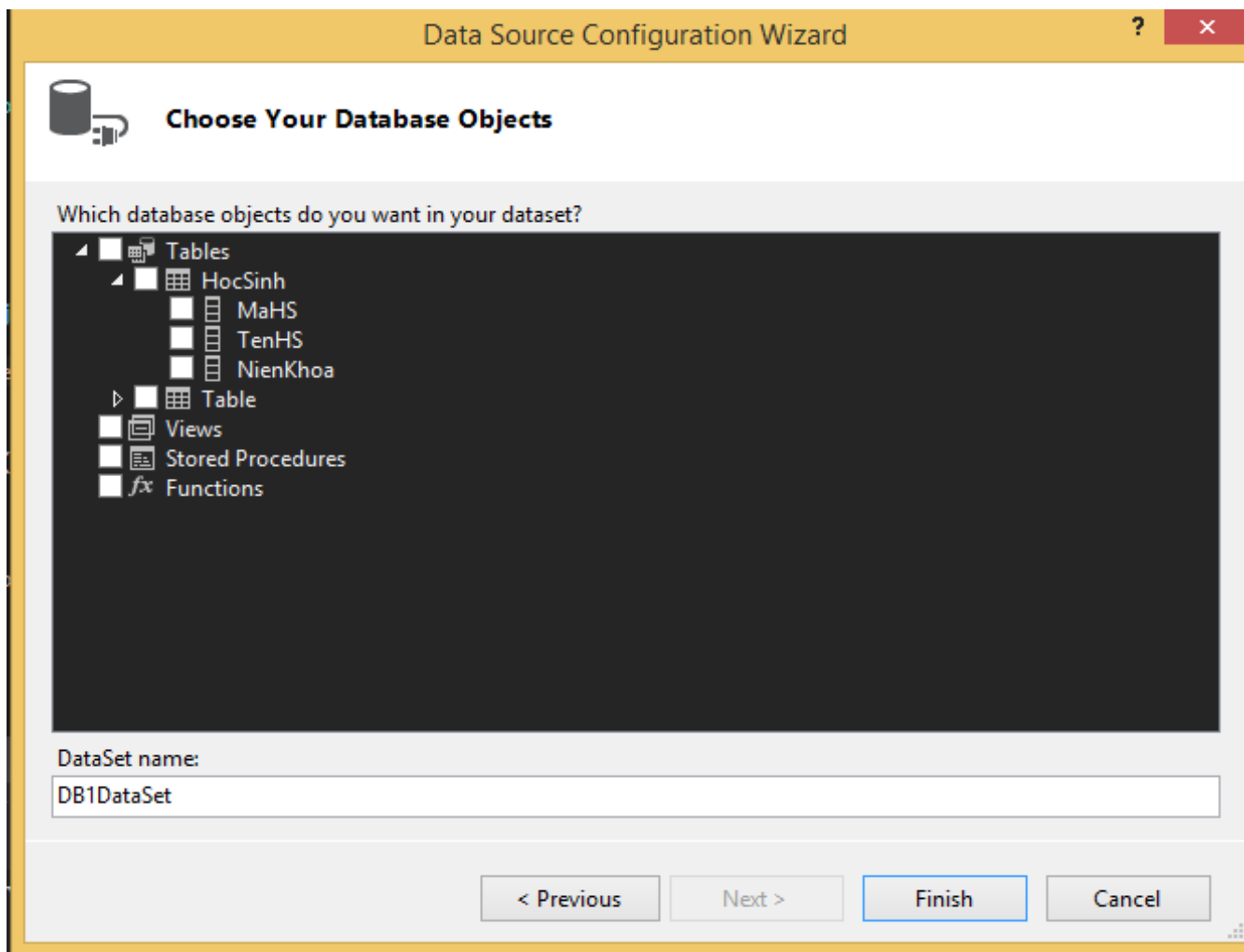
☒ Yes, save the connection as:

< Previous

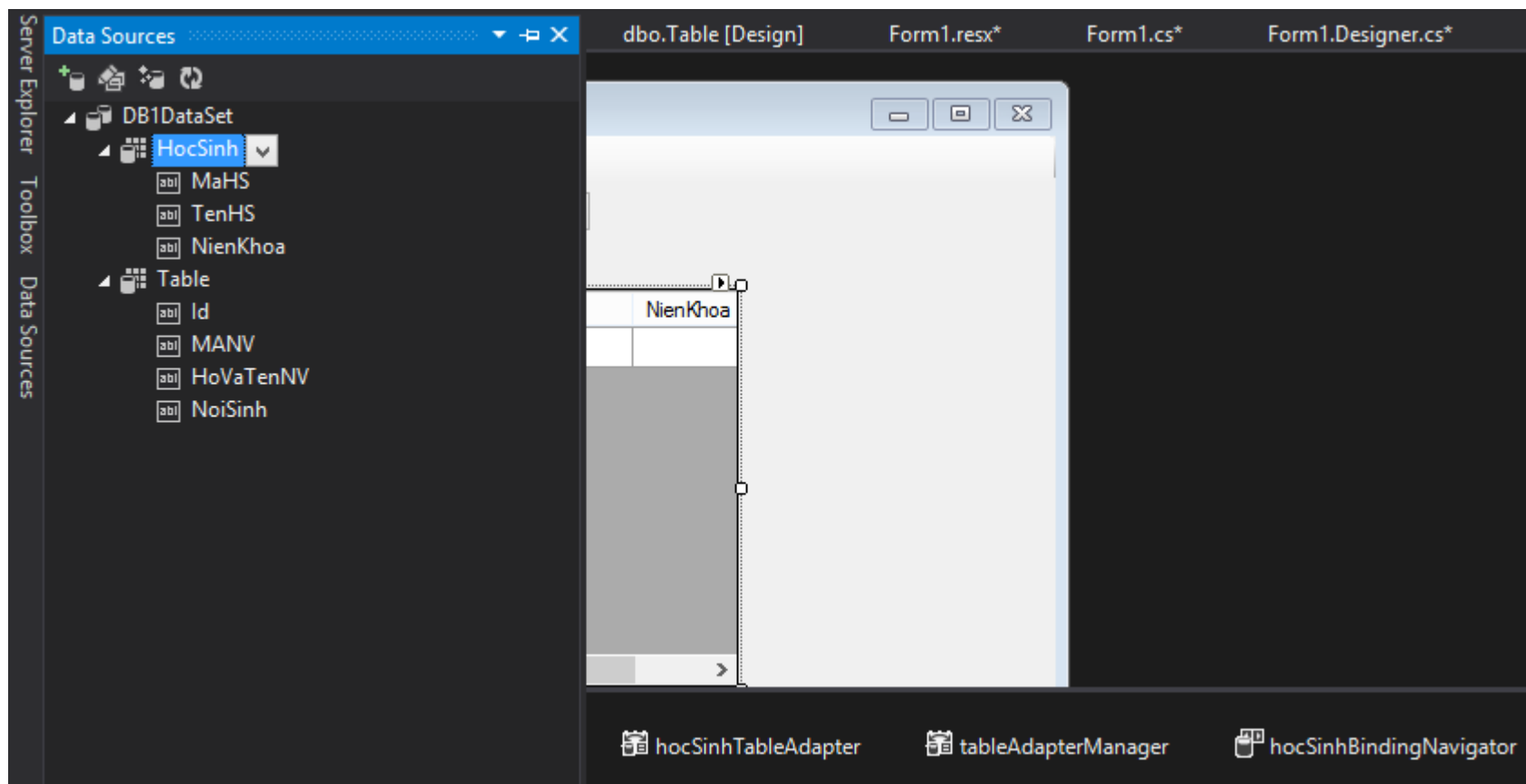
Next >

Finish

Cancel



Kéo thả bảng HocSinh từ cửa sổ Data Sources vào Form



Form sẽ tự tạo thêm DataGridView để hiển thị dữ liệu, các button di chuyển, các đối tượng phía dưới (dB1Dataset, hocSinhBindingSource, hocSinhTableAdapter ...)

dbo.HocSinh [Data] Form1.cs [Design]* x dbo.HocSinh [Design] dbo.Table [Design] Form1.resx* Form1.cs*

Form1

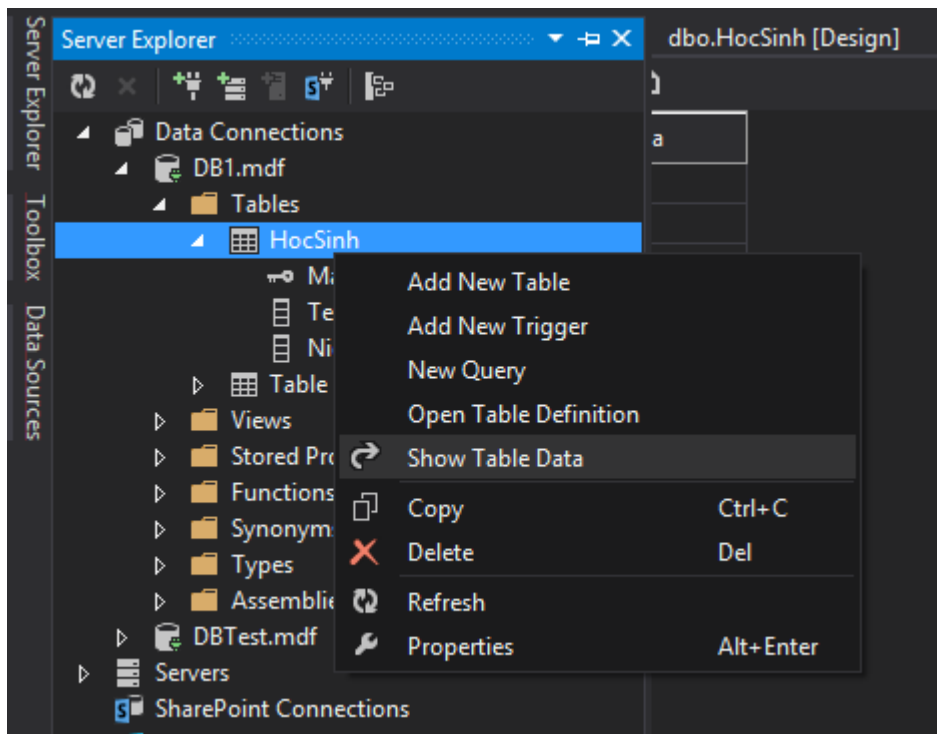
0 of {0}

MaSV Theem

	MaHS	TenHS	NienKhoa
*			

dB1DataSet hocSinhBindingSource hocSinhTableAdapter tableAdapterManager hocSinhBindingNavigator

Để thêm dữ liệu, từ bảng HocSinh click chuột phải chọn Show Table Data



dbo.HocSinh [Data] | Form1.cs [Design]* | dbo.HocS

Max Rows: 1000

	MaHS	TenHS	NienKhoa
	1050551	Ba Diep	31
	2012512	Van Teo	13
	B125125	Chi Pheo	38
	DI12Z6A1	Thi No	39
	NULL	NULL	NULL

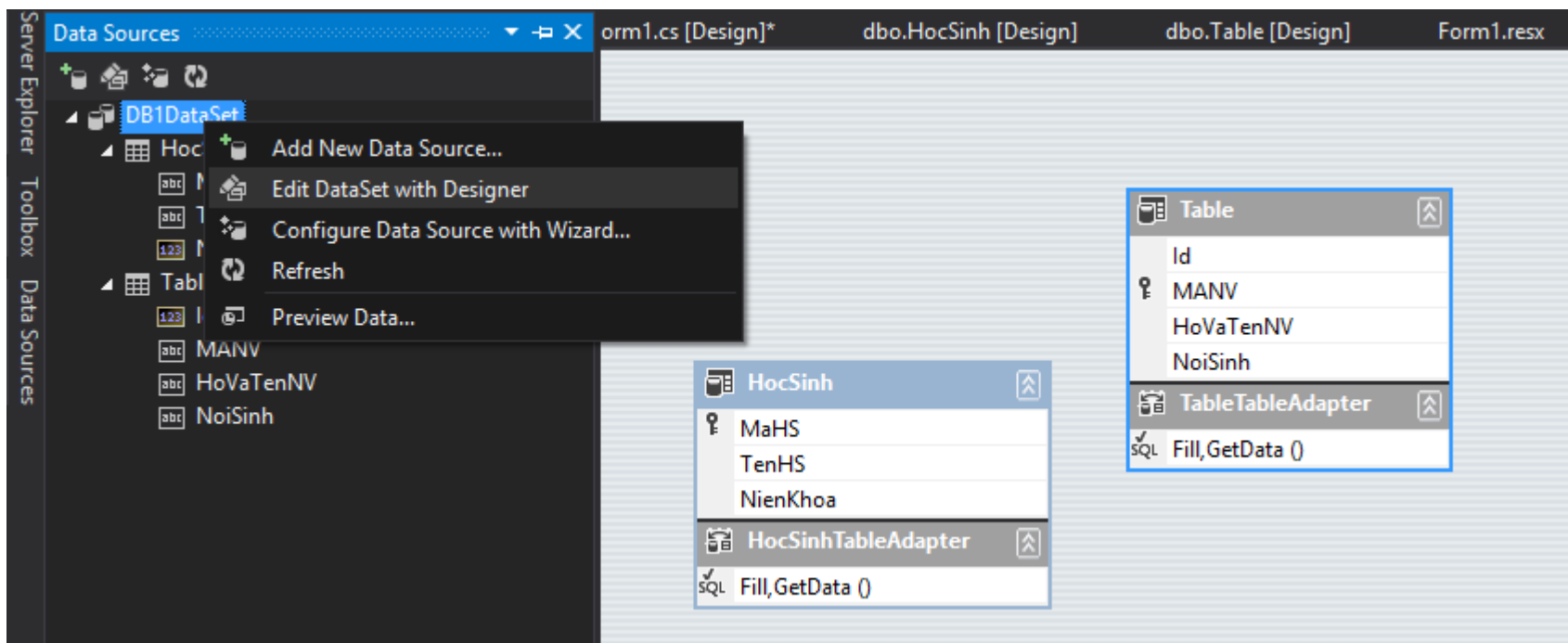
Form1

1 of 3

MaSV Theem

	MaHS	TenHS	NienKhoa
▶	1050551	Ba Diep	31
	2012512	Van Teo	13
	B125125	Chi Pheo	38
*			

Tuỳ chỉnh các thiết lập trong dataset qua công cụ “Edit DataSet with Designer”



Lập trình truy cập database

```
try
{
    SqlConnection conn = new SqlConnection("Data Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\DB1.mdf;Integrated Security=True;Connect Timeout=30");
    conn.Open();
    string sqltext = "SELECT COUNT(MaHS) FROM HOCSINH";
    SqlCommand com = new SqlCommand(sqltext, conn);
    int soHS = Convert.ToInt32(com.ExecuteScalar());
    MessageBox.Show("Tong so HS:" + soHS.ToString(), "Thong bao");
}
catch (SqlException ex)
{
    MessageBox.Show(ex.StackTrace);
}
```

Với ConnectionString được lấy từ lúc thiết lập dataSource và thay ký tự \ bởi ký tự \\



Choose Your Data Connection

Which data connection should your application use to connect to the database?

DB1Conn (Settings) ▼

New Connection...

This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

- ☐ No, exclude sensitive data from the connection string. I will set this information in my application code.
- ☐ Yes, include sensitive data in the connection string.

[-] Connection string that you will save in the application (expand to see details)

Data Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\DB1.mdf;Integrated Security=True;Connect Timeout=30

< Previous

Next >

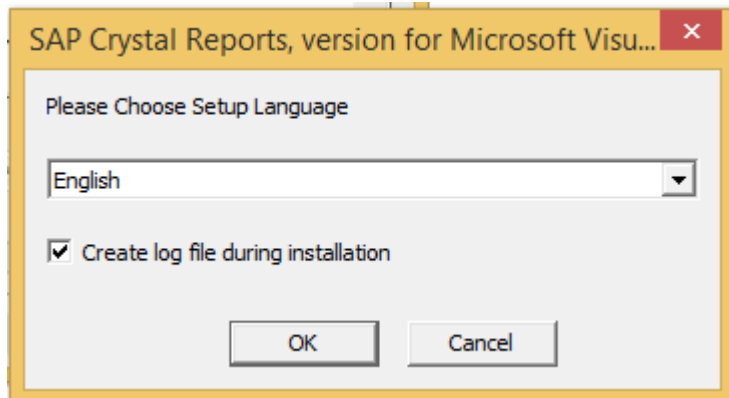
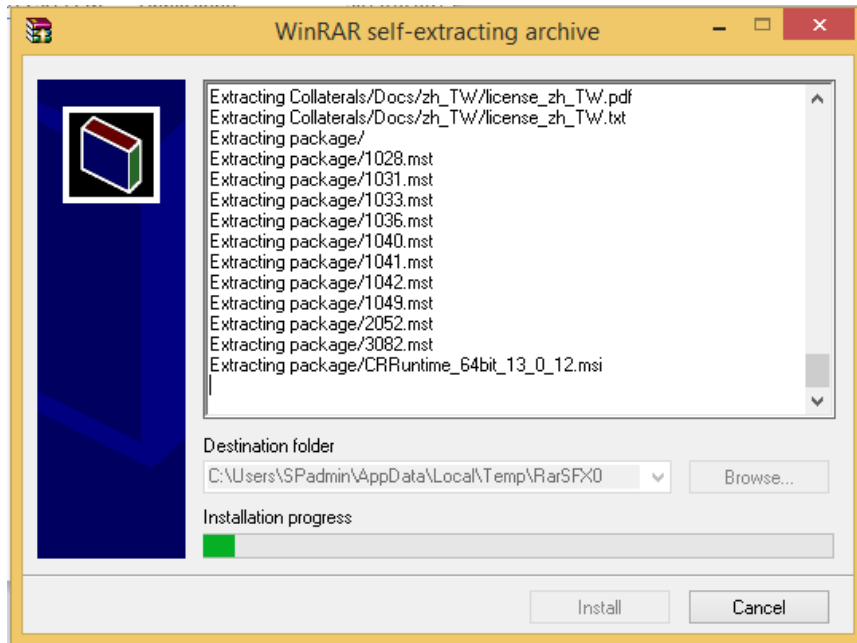
Finish

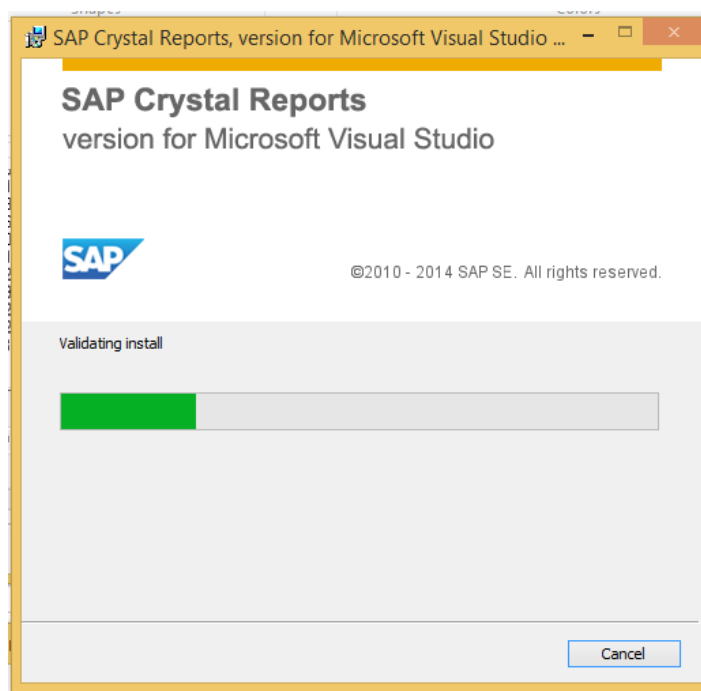
Cancel

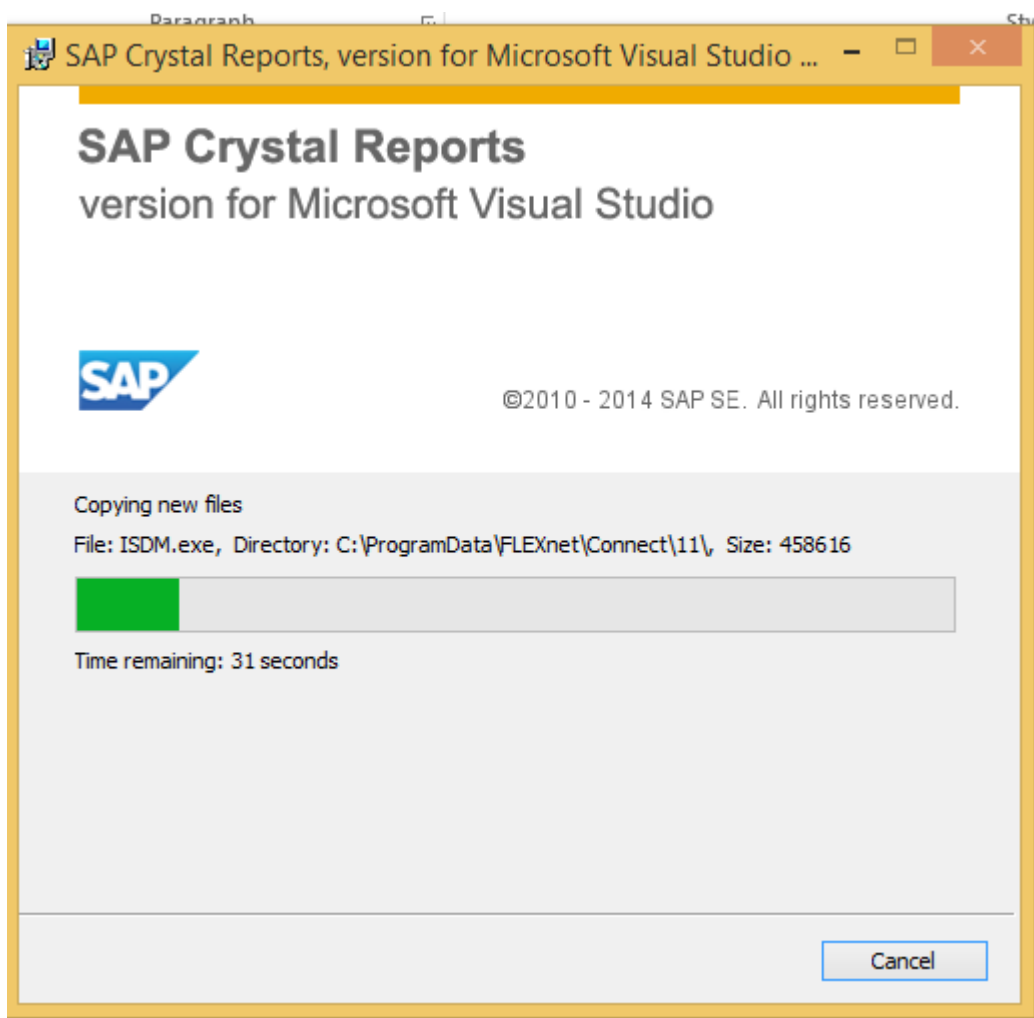
V. Crystal Report

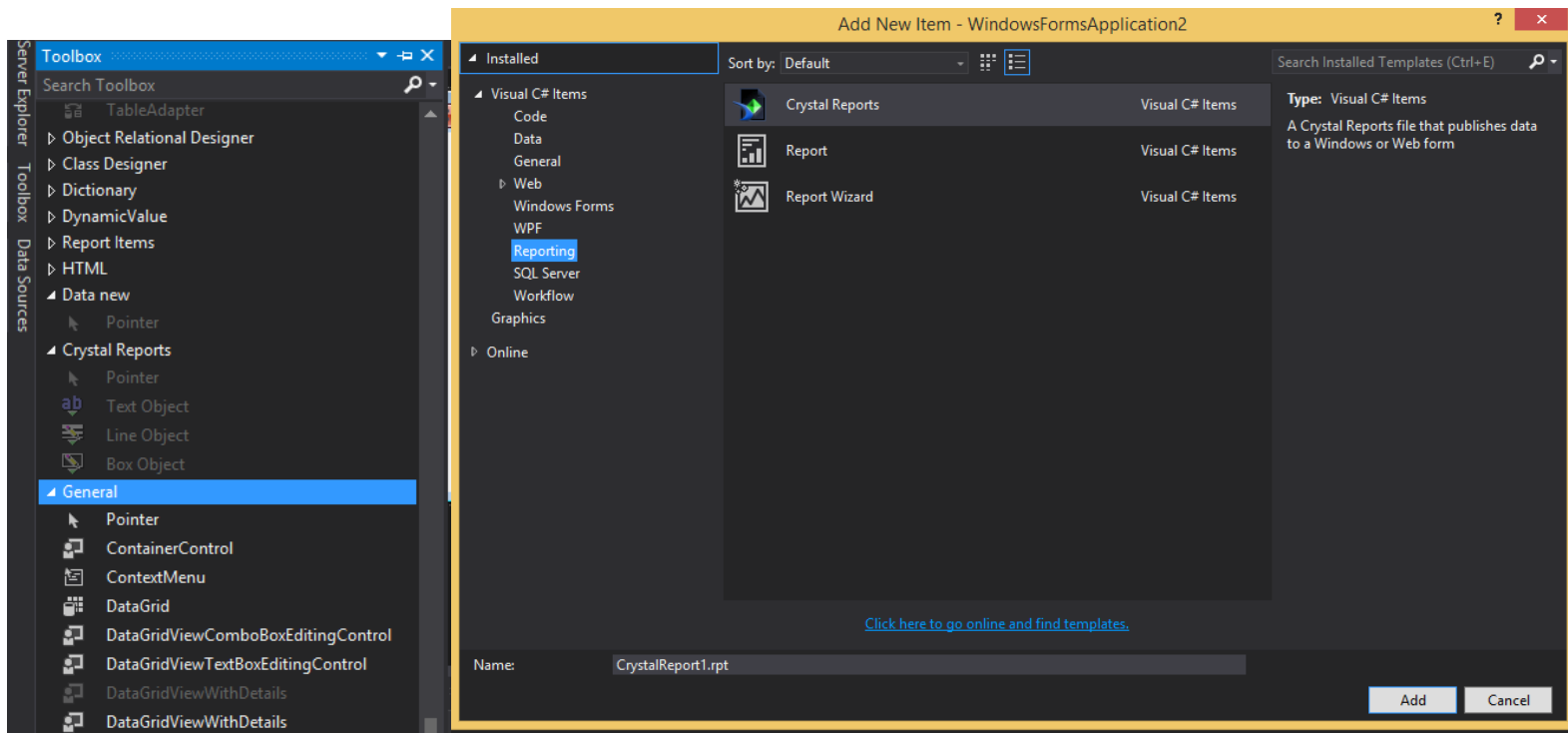
1. Cài đặt crystal report plugin và control cho visual studio

 CRforVS_13_0.exe	9/12/2015 3:18 PM	Application	284,686 KB
 CRforVS_13_0_12.exe	9/12/2015 3:14 PM	Application	301,818 KB

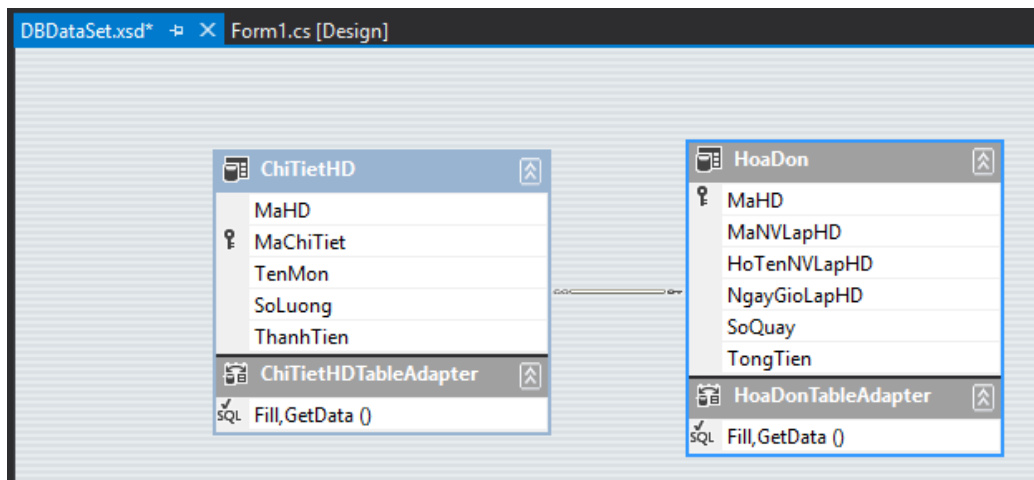


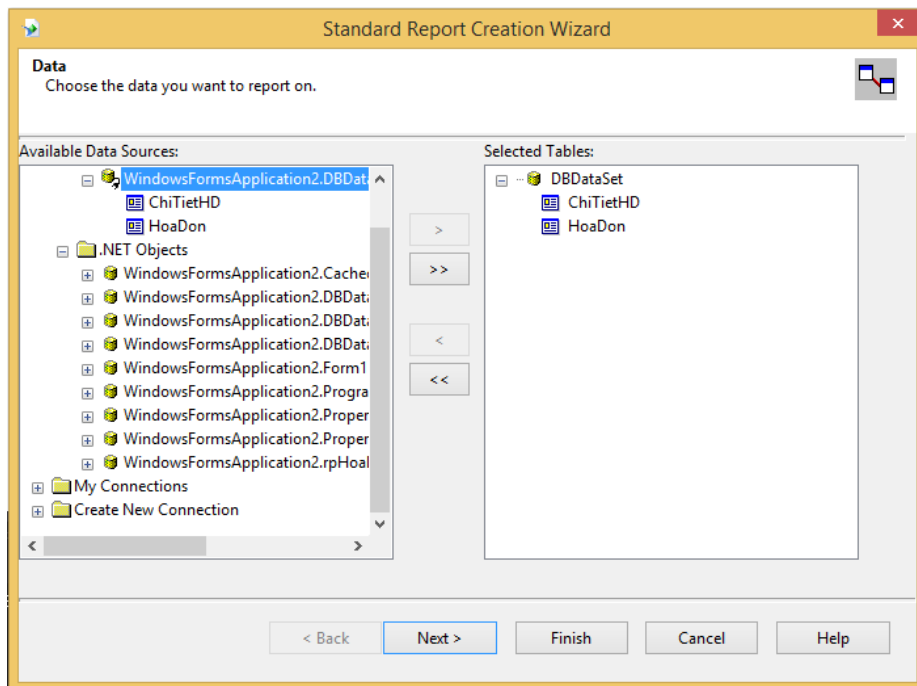
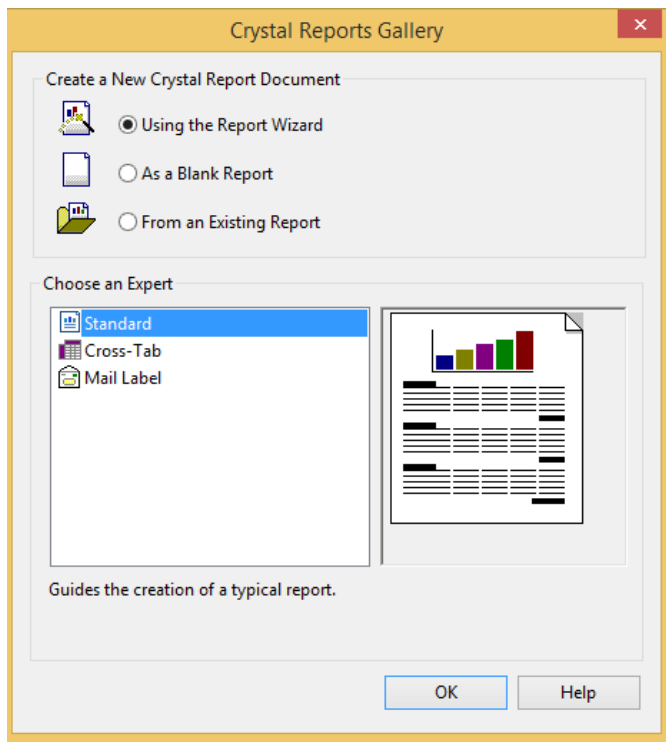






2. Sinh viên thực hiện kết nối đến cơ sở dữ liệu (hoặc file cơ sở dữ liệu – Mục IV) và tiến hành tạo đối tượng crystal report
Giả sử ta có DataSet sau:





Standard Report Creation Wizard

Link
Link together the tables you added to the report.

ChiTietHD

MaHD
MaChiTiet
TenMon
SoLuong

→

HoaDon

MaHD
MaNVLapHD
HoTenNVLapHD
NgayGioLapHD

Auto-Arrange

Auto-Link
☐ By Name
☒ By Key
Link

Order Links...
Clear Links
Delete Link
Link Options...
Index Legend...

< Back Next > Finish Cancel Help

Standard Report Creation Wizard

Fields
Choose the information to display on the report.

Available Fields:

ChiTietHD

MaHD
MaChiTiet
TenMon
SoLuong
ThanhTien

HoaDon

MaHD
MaNVLapHD
HoTenNVLapHD
NgayGioLapHD
SoQuay
TongTien

>
>>
<
<<

Fields to Display:

ChiTietHD.MaHD
ChiTietHD.MaChiTiet
ChiTietHD.TenMon
ChiTietHD.SoLuong
ChiTietHD.ThanhTien
HoaDon.MaHD
HoaDon.MaNVLapHD
HoaDon.HoTenNVLapHD
HoaDon.NgayGioLapHD
HoaDon.SoQuay
HoaDon.TongTien

Browse Data... Find Field...

< Back Next > Finish Cancel Help

Standard Report Creation Wizard

Grouping
(Optional) Group the information on the report.

Available Fields:

- Report Fields
 - ChiTietHD.MaHD
 - ChiTietHD.MaChiTiet
 - ChiTietHD.TenMon
 - ChiTietHD.SoLuong
 - ChiTietHD.ThanhTien
 - HoaDon.MaHD**
 - HoaDon.MaNVLapHD
 - HoaDon.HoTenNVLapHD
 - HoaDon.NgayGioLapHD
 - HoaDon.SoQuay
 - HoaDon.TongTien
- ChiTietHD
 - MaHD
 - MaChiTiet

Group By:

Browse Data... Find Field...

< Back Next > Finish Cancel Help

Standard Report Creation Wizard

Group Sorting
(Optional) Sort the groups based on the summarized totals.

Group that will be sorted:
HoaDon.MaHD

What kind of group ordering would you like to see?

☒ None
☐ Top 5 groups
☐ Bottom 5 groups

Comparing summary values for the Top or Bottom groups:

< Back Next > Finish Cancel Help

Standard Report Creation Wizard

Chart
(Optional) Include a chart on the report.

What kind of chart would you like to see?

☐ No Chart ☒ Bar Chart ☐ Line Chart ☐ Pie Chart

Chart title:
Count of SoLuong / MaHD

On change of:
HoaDon.MaHD

Show summary:
Count of ChiTietHD.SoLuong

< Back Next > Finish Cancel Help

Standard Report Creation Wizard

Record Selection
(Optional) Select a subset of information to display.

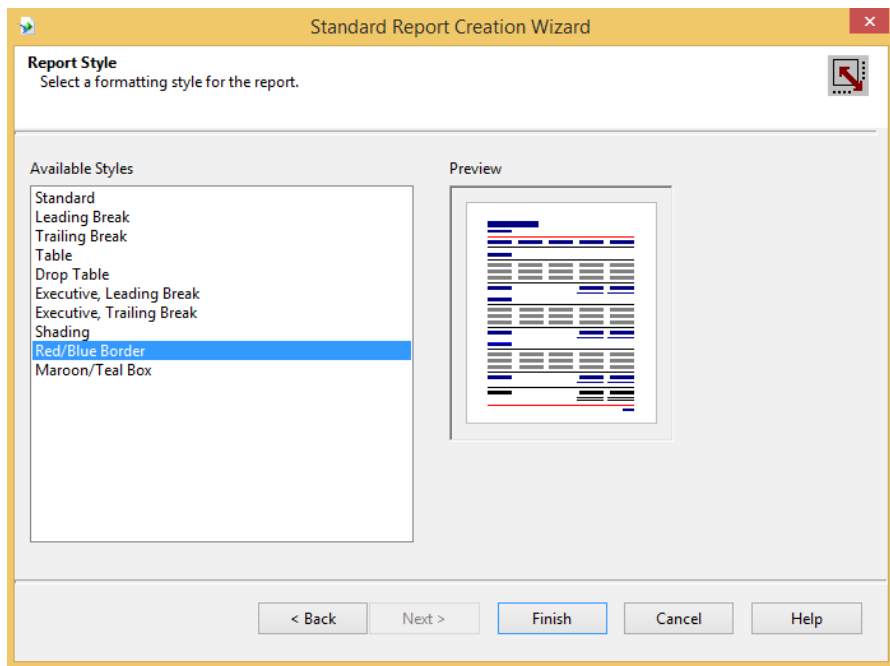
Available Fields:

- Report Fields
 - ChiTietHD.MaHD
 - ChiTietHD.MaChiTiet
 - ChiTietHD.TenMon
 - ChiTietHD.SoLuong
 - ChiTietHD.ThanhTien
 - HoaDon.MaHD
 - HoaDon.MaNVLapHD
 - HoaDon.HoTenNVLapHD
 - HoaDon.NgayGioLapHD
 - HoaDon.SoQuay
 - HoaDon.TongTien
- ChiTietHD
 - MaHD
 - MaChiTiet
 - TenMon

Filter Fields:

> <

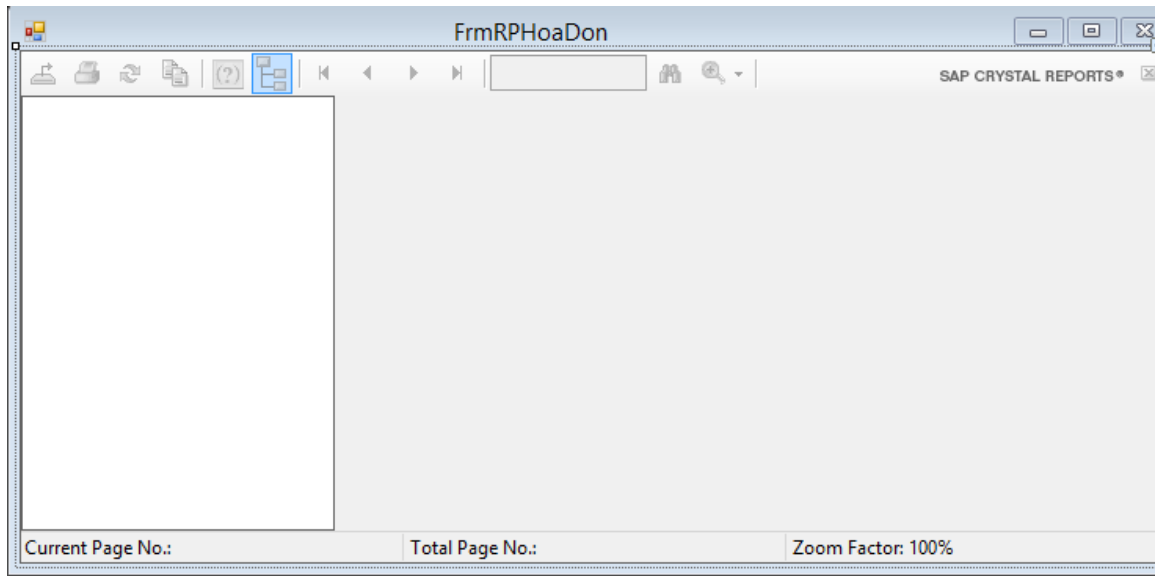
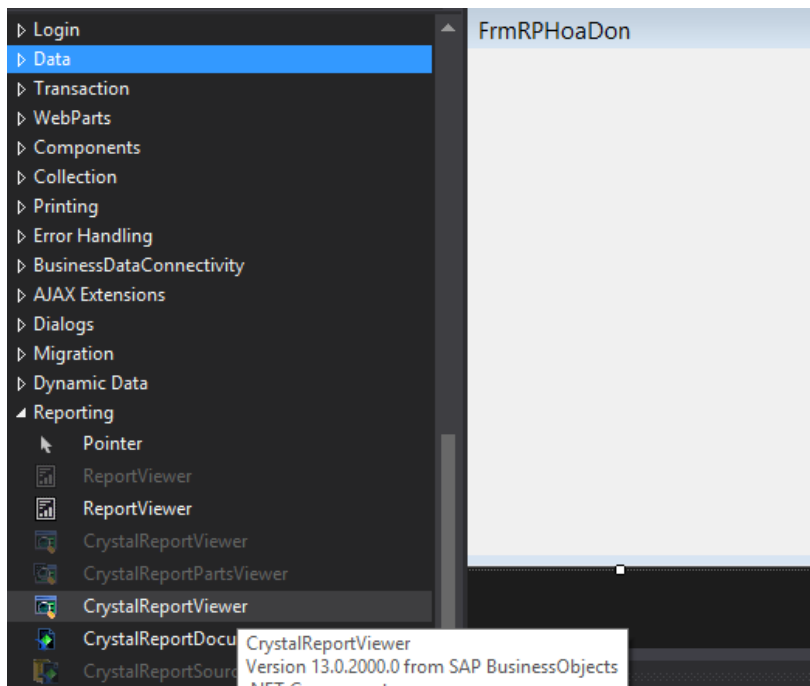
< Back Next > Finish Cancel Help



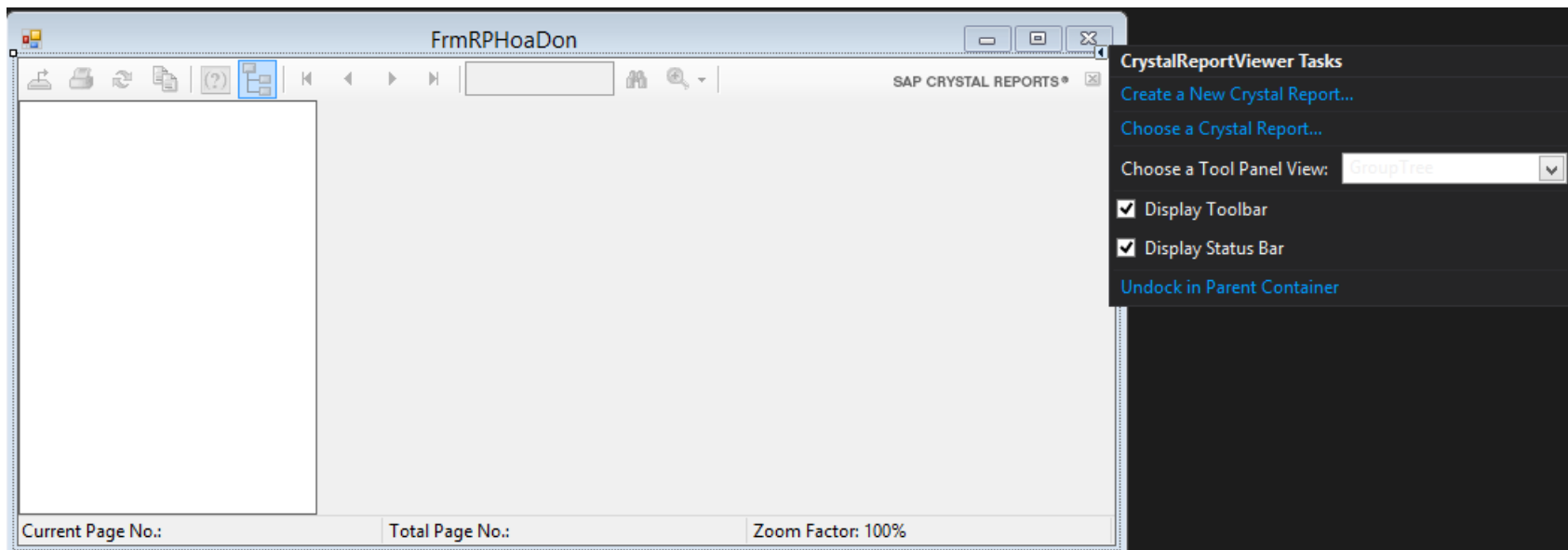
3. Hiện thị report trên window form

Tạo Form FrRpHoaDon để hiển thị report Hoá Đơn.

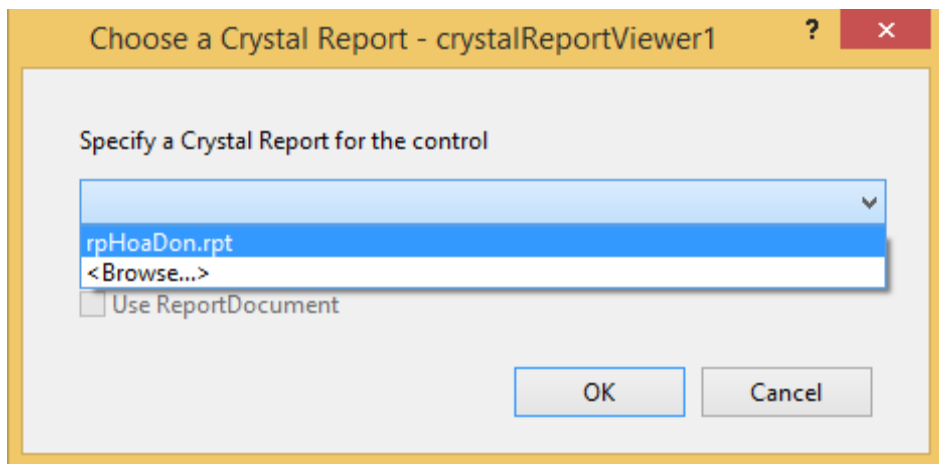
Từ toolbox kéo thả đối tượng CrystalReportViewer để hiển thị report lên form



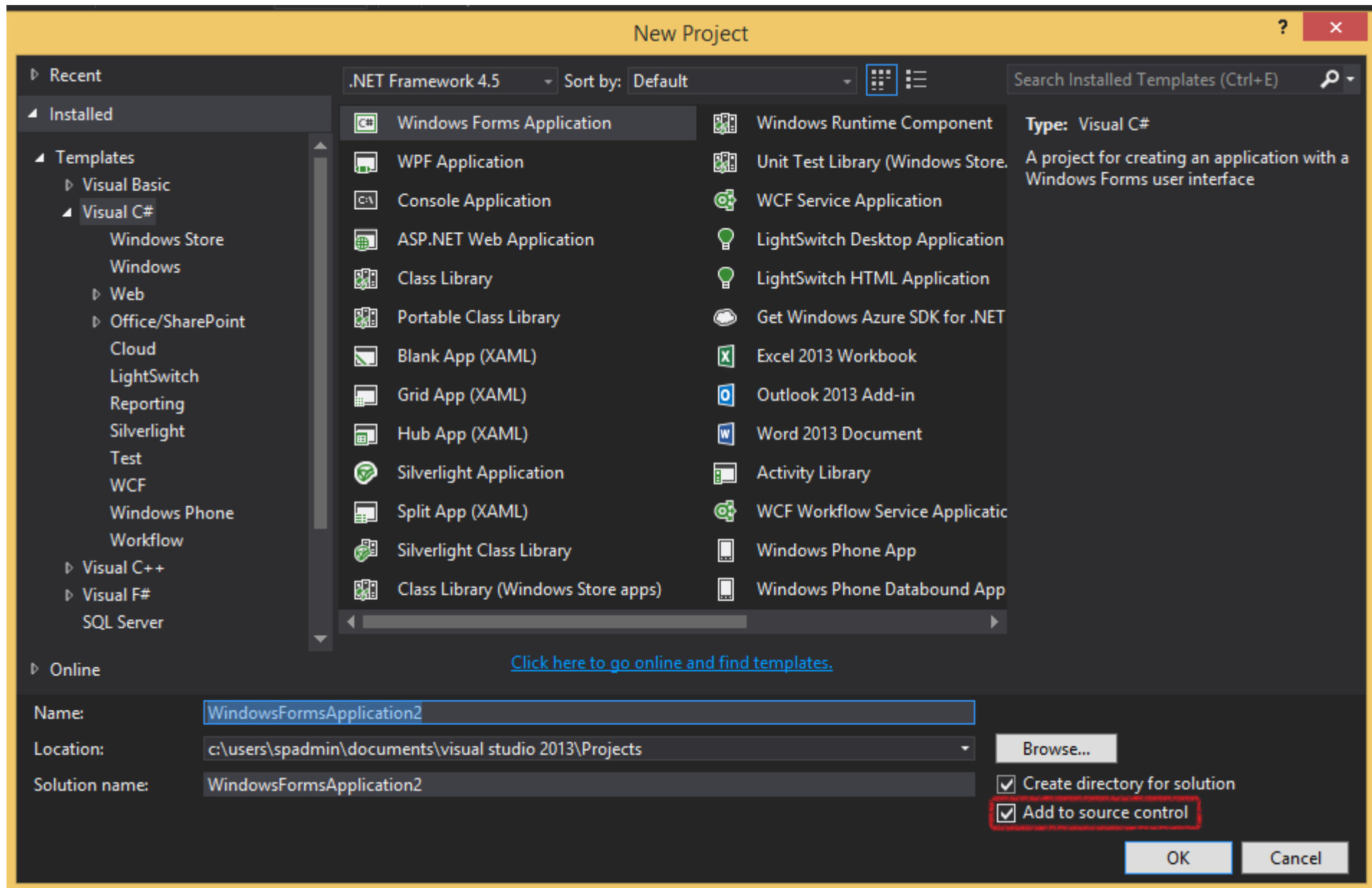
Chọn “Choose a Crystal Report” từ các report đã tạo

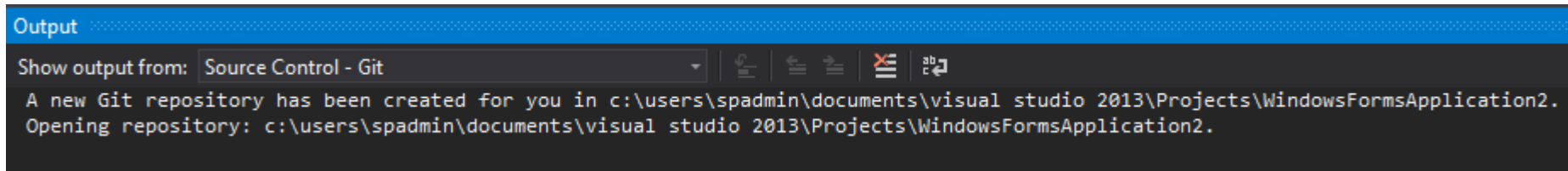
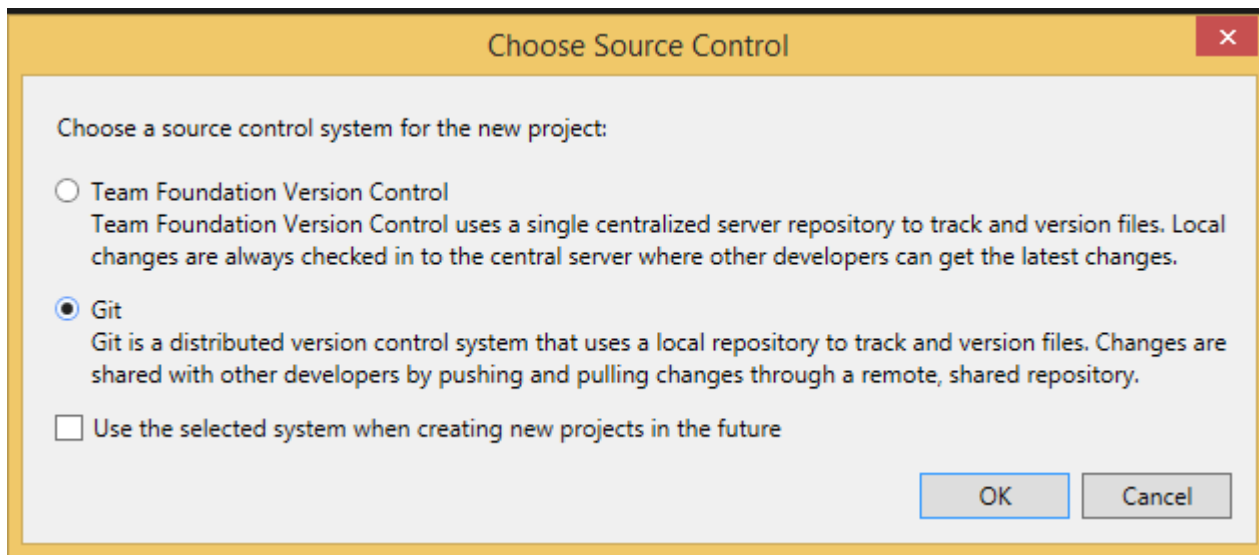


Chọn report Hoá Đơn (rpHoaDon) chúng ta đã tạo qua công cụ hỗ trợ tạo report



VI. Source Control Version với Git

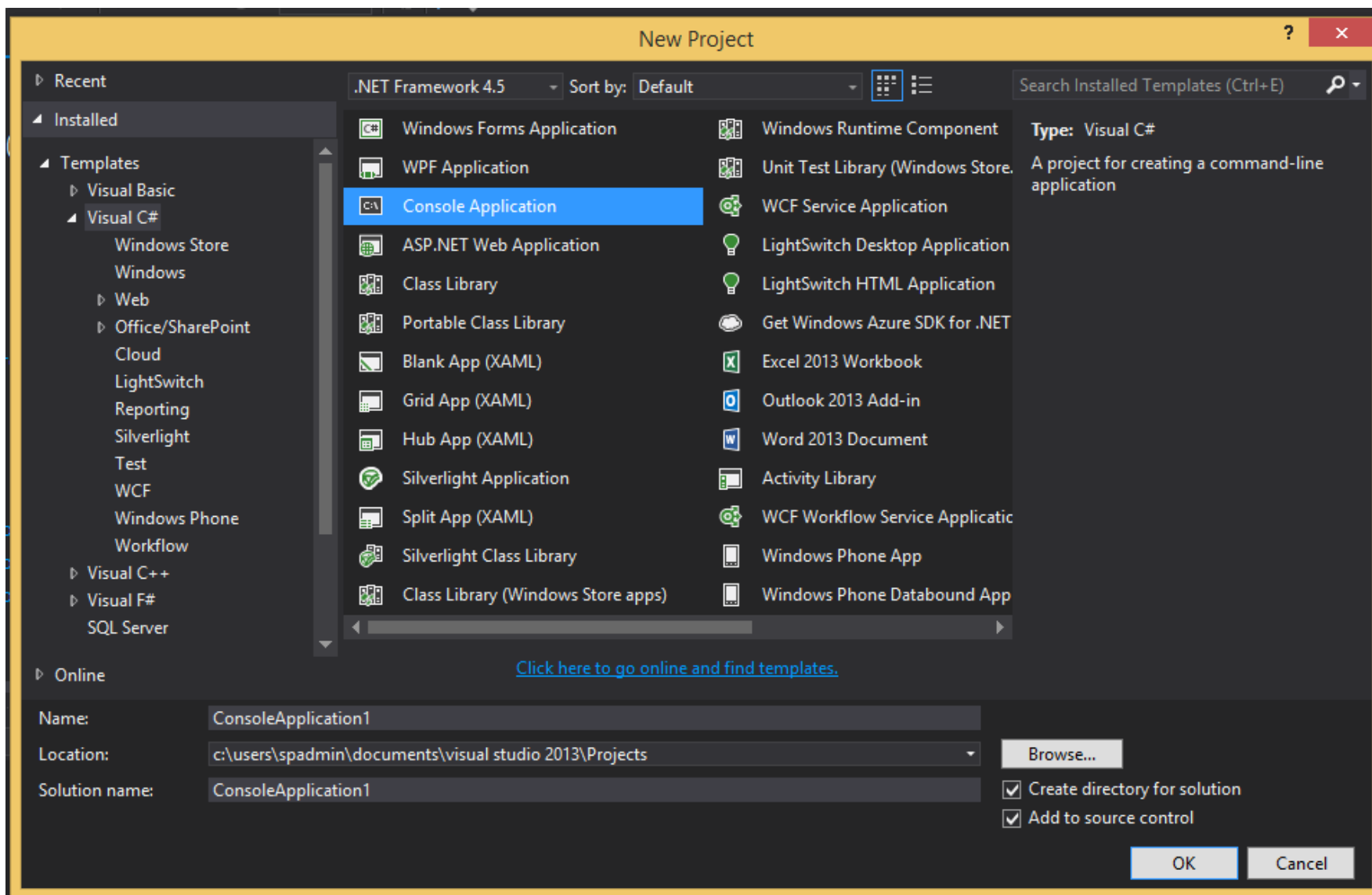




VII. Bài tập 1 - Lập trình C# với Console Application

Tạo ứng dụng Console Application (giao tiếp qua giao diện dòng lệnh, giống commandline hoặc terminal)

File-> New->Project -> Console Application



Sinh viên viết code trong hàm main, thông tin sẽ hiển thị ở cửa sổ debug hoặc output (thường ở phía dưới cửa sổ)

```
Program.cs* X
ConsoleApplication1.Program
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApplication1
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13
14         }
15     }
16 }
17
```

4 lệnh thường sử dụng là:

```
Console.Write();
Console.WriteLine();
Console.Read();
Console.ReadLine();
```

Quy tắc cộng chuỗi hiển thị VD: `System.Console.Write("gia tri bien a = " + a + "b= " + b);`

Quy tắc sử dụng {đối số} để giữ chỗ hiển thị VD: `System.Console.Write("gia tri bien a = {0} va b = {1}", a + 2, b + 6);`

Sinh viên có thể chuyển kiểu chuỗi về kiểu số qua 2 hình thức ép kiểu sau:

```
int a = int.Parse(Console.ReadLine());
int b = Convert.ToInt32(Console.ReadLine());
```

Khởi động: Viết chương trình cho phép hiển thị 'Hello World' lên màn hình.

1. Viết chương trình tính $n!$ với n nhập từ bàn phím.
2. Viết chương trình tính $1 + 2 + \dots + n$ với n nhập từ bàn phím.
3. Viết chương trình tính x^n với x và n là số nguyên được nhập từ bàn phím.
4. Viết chương trình tính ước số chung lớn nhất của hai số nguyên dương a và b được nhập từ bàn phím.
5. Viết chương trình in ra màn hình theo dạng sau (với n nhập từ bàn phím, nếu $n < 1$ thì in ra câu thông báo lỗi)

*	*	*
**	**	**
	***	***

n = 2	n = 3	n = 4

6. Viết chương trình in ra màn hình theo dạng sau (với n nhập từ bàn phím, nếu n < 1 thì in ra câu thông báo lỗi)

*	*	*
**	**	**
	***	***

n = 2	n = 3	n = 4

7. Viết hàm đệ quy fibo(n) tính số Fibonacci thứ n. Viết chương trình chính sử dụng hàm fibo(n) để tính số fibonacci thứ n với n nhập từ bàn phím.

8. Tạo lớp HìnhTron có 1 dữ liệu thành viên là bán kính cùng 2 phương thức tính chu vi (ChuVi()) và diện tích (DienTich()) của đường tròn.

- Định nghĩa phương thức xây dựng có 1 tham số với ý nghĩa là gán giá trị bán kính là giá trị của tham số trong phương thức xây dựng.
- Định nghĩa 1 thuộc tính R cho phép gán và lấy giá trị của dữ liệu thành viên bán kính.

Tạo lớp HìnhCau thừa kế từ lớp HìnhTron.

- Tính diện tích bề mặt của hình cầu. Phương thức tính diện tích bề mặt được định nghĩa bằng cách ghi đè phương thức DienTich() của lớp HìnhTron.

$$\text{Diện tích bề mặt} = 4 \cdot \pi \cdot R^2$$

- Tính thể tích của hình cầu. Thể tích = $\frac{4}{3} \cdot \pi \cdot R^3$

Viết phương thức Main cho phép nhập 1 giá trị bán kính r từ bàn phím. Dùng cơ chế xử lý ngoại lệ để xử lý các ngoại lệ phát sinh trong quá trình nhập giá trị từ bàn phím. Kế tiếp, tạo 1 đối tượng HìnhTron và 1 đối tượng HìnhCau. Cuối cùng, hiển thị chu vi và diện tích của đối tượng HìnhTron cùng diện tích bề mặt và thể tích của đối tượng HìnhCau.

9. Định nghĩa 1 giao tiếp Shape có 1 phương thức tính diện tích DienTich(). Kế tiếp

định nghĩa 3 lớp sau đây hiện thực giao tiếp Shape:

- Lớp HìnhTron có 1 dữ liệu thành viên là bán kính.
- Lớp HìnhChuNhat có 2 dữ liệu thành viên là chiều dài và chiều rộng.
- Lớp HìnhVuong có 1 dữ liệu thành viên là chiều dài cạnh.

Cả 3 lớp trên đều có các phương thức xây dựng có tham số nhằm gán giá trị cho dữ liệu thành viên của mỗi lớp. Để ý là cả 3 lớp này đều phải cài đặt phương thức tính diện tích của giao tiếp Shape.

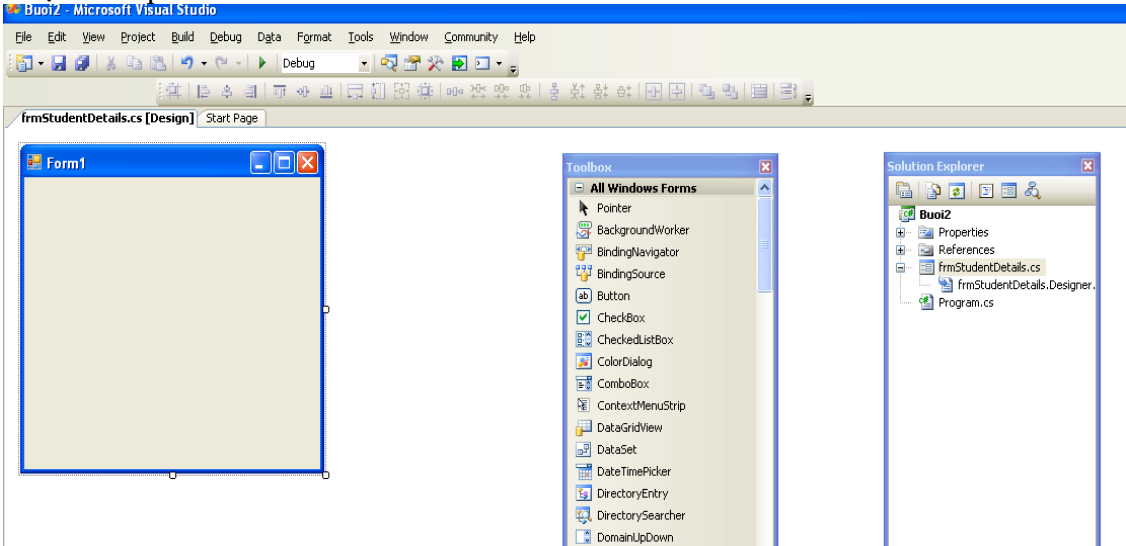
Trong phương thức Main(), tạo 3 đối tượng hình tròn, hình chữ nhật và hình vuông. 3 đối tượng này là các phần tử của 1 mảng các đối tượng Shape. Dùng vòng lặp foreach để lặp qua mảng này và hiển thị kết quả diện tích các hình phẳng lên màn hình.

10. Tạo 1 delegate tên HienThi nhằm tham chiếu đến các phương thức có 1 tham số là 1 số nguyên và kiểu trả về là void.

- Định nghĩa 1 event tên là TimSoLe gắn liền với delegate HienThi.
- Phát sinh sự kiện TimSoLe để liệt kê các giá trị là số nguyên lẻ trong khoảng từ 1 đến 50.

VIII. Bài tập 2 - Ứng dụng Windows Form

1. Sửa lại tên lớp của form Test thành StudentDetails.



2. Thêm các controls lên form để tạo giao diện như hình

3. Thiết lập các thuộc tính cho các controls:

Control	Thuộc tính	Giá trị
---------	------------	---------

TextBox	Name Text	txtName
TextBox	Name Text	txtYear
ListBox	Name Items	lstClass I II III IV V VI VII VIII IX X
ComboBox	Name Items	cboDiv A B C D
CheckedListBox	Name Items	lstHob Singing Dancing Swimming Reading Travelling
Button	Name Text	btnConfirm Confirm Student Details
Button	Name Text	btnClear Clear

4. Xác nhận thông tin người dùng nhập vào các controls trên form. Xử lý sự kiện Click của button btnConfirm:

```

private void btnConfirm_Click(object sender, EventArgs e)
{
    int cnt = lstHob.CheckedItems.Count;
    string hobby="";
    for(int index=0;index<=cnt-1;index++)
    {
        hobby = hobby + lstHob.CheckedItems[index];
        if (index<cnt-1)
            hobby = hobby + ",";
    }

    MessageBox.Show(txtName.Text + " born in " + txtYear.Text + " from Class " +
        lstClass.Text + ", Division " + cboDiv.Text +
        " has following hobbies " + hobby);
}

```

5. Button Clear sẽ xóa hết các giá trị người dùng đã nhập. Xử lý sự kiện Click của button btnClear:

```

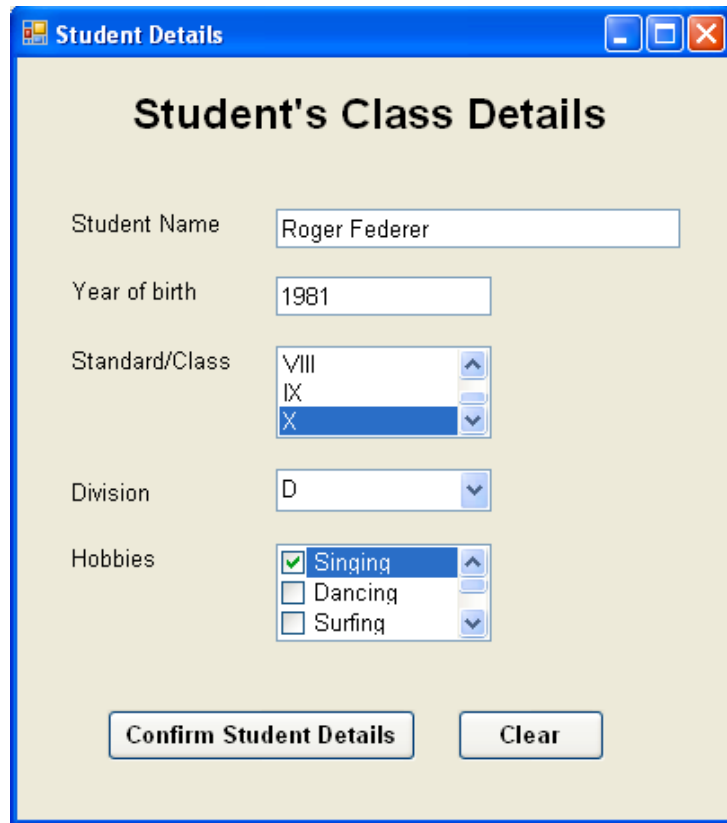
private void btnClear_Click(object sender, EventArgs e)
{
    lstClass.ClearSelected();
    cboDiv.Text="";
    lstHob.ClearSelected();
    for(int index=0;index<=lstHob.Items.Count-1;index++)
    {
        if(lstHob.GetItemChecked(index))
            lstHob.SetItemChecked(index,false);
    }
}

```

6. Lưu dự án rồi chọn Build→Build Solution

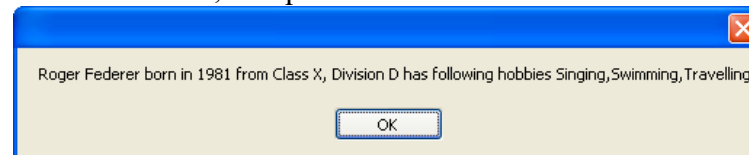
7. Thực thi chương trình bằng cách chọn Debug/Start

8. Nhập các thông tin vào các controls (hình bên dưới)



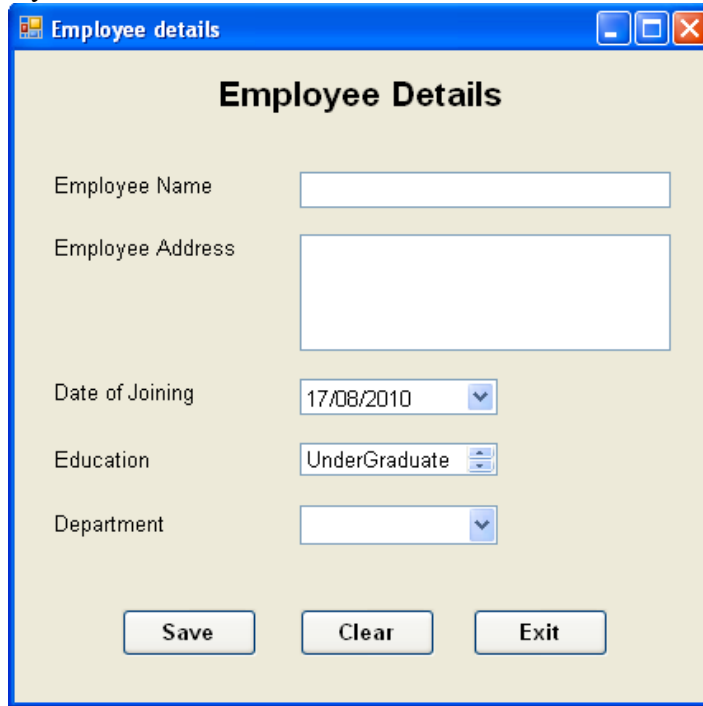
A screenshot of a Windows-style application window titled "Student Details". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a light beige background and is titled "Student's Class Details". It contains several input fields: "Student Name" with the text "Roger Federer", "Year of birth" with "1981", "Standard/Class" with a dropdown menu showing "VIII", "IX", and "X" (where "X" is selected), "Division" with a dropdown menu showing "D", and "Hobbies" with a list box containing "Singing" (checked), "Dancing", and "Surfing". At the bottom, there are two buttons: "Confirm Student Details" and "Clear".

9. Sau khi nhập thông tin, click button Confirm Student Details, kết quả sẽ là:



10. Xóa các giá trị đã nhập bằng cách click button Clear.

Bài 2: Tạo form Employee Details như sau:



Form này cho phép người dùng nhập vào họ tên, địa chỉ nhân viên, ngày vào làm. Bên cạnh đó, người dùng có thể lựa chọn mục Education bằng cách lựa chọn từ danh sách (UnderGraduate, PostGraduate). Danh sách Department cho phép người dùng lựa chọn phòng ban (Human Resource, Marketing, R&D) mà nhân viên đó làm việc. Một số yêu cầu khác của form này:

1. Địa chỉ nhân viên cho phép nhập nhiều hơn 1 dòng văn bản.
2. Button Save cho phép xác nhận thông tin người dùng nhập vào. (MessageBox)
3. Button Clear cho phép xóa các thông tin người dùng đã nhập trước đó.
4. Button Exit cho phép kết thúc chương trình

Bài 3: Tạo ứng dụng cho phép người dùng xác nhận thông tin đã nhập để giữ chỗ một chuyến bay. Một số thông tin về chương trình:

1. Chương trình cho phép người dùng nhập mã số vé, tên của hành khách, số hộ chiếu của hành khách và ngày bay.
2. Người dùng có thể lựa chọn thành phố xuất phát, thành phố đích và hạng vé (phổ thông, hạng nhất, thương gia).

3. Người dùng có thể lựa chọn thêm một số dịch vụ trên máy bay (chăm sóc trẻ nhỏ, y tá, ...).
4. Button Verify sẽ hiển thị thông báo lỗi nếu thành phố xuất phát và thành phố đích giống nhau.
5. Button Clear cho phép xóa các thông tin người dùng đã nhập. Form có thể có giao diện như sau:

Flight Reservation Details

Ticket number

Passenger Name

Passport number

Source
Ha Noi
Singapore

Destination
Ha Noi
Singapore

Flight Date

Class

Services ☐ Child Care
☐ Nurse

Verify **Clear**

IX. Bài tập 3 - Ứng dụng Windows Form2

1. Bài 1:

Một cửa hàng sách cần quản lý thông tin về các loại sách sẽ bán của mình. Một số thông tin cần quản lý là: tên sách, số lượng, giá bán. Hãy thiết kế lớp *sách bán* cho phép có những thuộc tính (có phạm vi truy cập là public) là các thông tin trên. Ngoài các thuộc tính lớp nên có phương thức tính tiền dựa vào số lượng sách và giá bán mỗi quyển.

Thí dụ: Quyển *Mật mã của Da Vinci* bán cho 1 khách hàng là 20 quyển, giá bán 1 quyển là 99000đ. Vậy số tiền phải trả là 1.980.000 đ.

1. Tạo ứng dụng loại Windows Application.
2. Thêm một lớp mới tên là clsBookSale. Lớp này có 03 dữ liệu thành viên: tên sách, số lượng, giá bán.

```
class clsBookSale
{
    string strTitle;
    int intQuantity;
    decimal decPrice;
}
```

3. Ứng với mỗi dữ liệu thành viên, một thuộc tính được tạo ra:


```

public string Title
{
    get
    {
        return strTitle;
    }
    set
    {
        strTitle=value;
    }
}

public int Quantity
{
    get
    {
        return intQuantity;
    }
    set
    {
        if (value>0)
            intQuantity = value;
        else throw new Exception();
    }
}

```

```

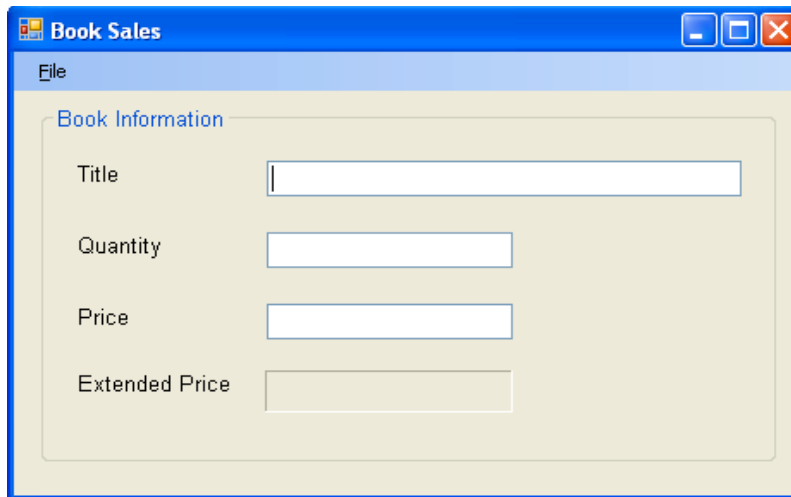
public decimal Price
{
    get
    {
        return decPrice;
    }
    set
    {
        if(value>=0)
            decPrice=value;
        else throw new Exception();
    }
}

```

4. Phương thức tính tiền cho phép tính tiền sau mỗi lần bán sách:

```
public decimal ExtendedPrice()
{
    decimal decExtendedPrice = intQuantity * decPrice;
    return decExtendedPrice;
}
```

5. Thiết kế form của ứng dụng:



Hình 3.1: Giao diện ban đầu

Các mục trên menu File như sau:

- New: cho phép xóa trắng các ô nhập.
- Calculate: Sau khi người dùng nhập các thông tin vào các ô Title, Quantity, Price; người dùng có thể chọn tính năng này để tính số tiền phải trả.
- Exit: chấm dứt ứng dụng.

6. Khai báo một đối tượng thuộc lớp clsBookSale trong phần khai báo của form ở hình 3.1:

```
public partial class frmBookSale : Form
{
    private clsBookSale mBookSale;
    public frmBookSale()
    {
        InitializeComponent();
    }
}
```

7. Sự kiện Click của mục menu mnuCalculate được xử lý:

```
private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsBookSale();
        mBookSale.Title = txtTitle.Text;
        mBookSale.Quantity = Convert.ToInt32(txtQuan.Text);
        mBookSale.Price = Convert.ToDecimal(txtPrice.Text);

        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

8. Sự kiện Click của mục menu New cần xử lý để cho phép xóa trắng các ô nhập.

9. Sự kiện Click của mục menu Exit cho phép chấm dứt chương trình. Để thực hiện được, ta sử dụng: **Application.Exit();**

10. Lưu dự án và thực thi ứng dụng

11. Giả sử, cần ghi nhận số lần bán sách trong ngày và tổng số tiền bán ra trong ngày; lúc đó lớp clsBookSale cần được bổ sung các thành viên để cho phép ghi nhận 02 giá trị này.

Giải pháp ở đây là thêm 02 dữ liệu thành viên dạng **static**; cứ mỗi lần bán sách, số lần tăng 1 và số tiền tăng lên bằng số tiền đã bán ở lần đó.

Thêm 02 dữ liệu thành viên **static** vào lớp `clsBookSale` và 02 thuộc tính tương ứng:

```
static decimal decSalesTotal=0;
static int intSalesCount=0;

public static decimal SalesTotal
{
    get
    {
        return decSalesTotal;
    }
}

public static int SalesCount
{
```

Phương thức `ExtendedPrice` được viết lại:

```
{
    return intSalesCount;
}

public decimal ExtendedPrice()
{
    decimal decExtendedPrice = intQuantity * decPrice;
    decSalesTotal += decExtendedPrice;
    intSalesCount += 1;

    return decExtendedPrice;
}
```

12. Menu File của form hình 3.1 được bổ sung mục Summary với sự kiện Click của mục này được xử lý:

```
private void mnuSum_Click(object sender, EventArgs e)
{
    string strMessage = "Sales Total: " +
        clsBookSale.SalesTotal.ToString("C") +
        "\nSales Count: " + clsBookSale.SalesCount.ToString();

    MessageBox.Show(strMessage, "Summary Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

13. Giả sử, cửa hàng sách cần giảm giá 10% cho một số khách hàng nào đó. Việc này có thể được thể hiện bằng 1 dữ liệu thành viên **bool** trong lớp clsBookSale:

```
const float sngDISCOUNT_RATE = 0.1F;

bool blnDiscount;
public bool Discount
{
    get
    {
        return blnDiscount;
    }
    set
    {
        blnDiscount=value;
    }
}
```

Lúc này, một phương thức tính số tiền giảm được định nghĩa thêm trong lớp clsBookSale:

```
public decimal DiscountAmount()
{
    decimal decDiscountAmount;
    if (blnDiscount)
        decDiscountAmount = intQuantity * decPrice *
            Convert.ToDecimal(sngDISCOUNT_RATE);
    else
        decDiscountAmount = 0M;

    return decDiscountAmount;
}
```

Phương thức NetDue() cho phép tính số tiền thực sự phải trả, số tiền này = Tổng số tiền – Số tiền giảm. Lưu ý là khi tính số tiền thực sự phải trả ta cần cập nhật lại các thuộc tính dùng chung (**static**) cho các đối tượng của lớp.

```

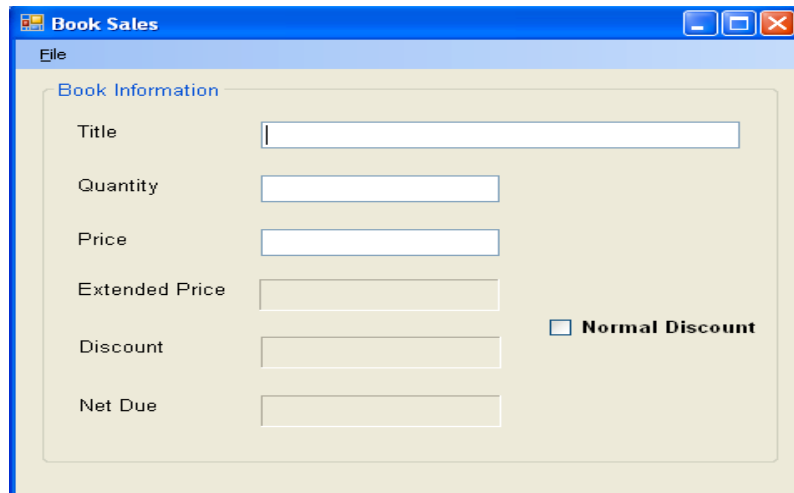
public decimal NetDue()
{
    decimal decNetDue = ExtendedPrice() - DiscountAmount();

    decSalesTotal += decNetDue;
    intSalesCount += 1;
    return decNetDue;
}

public decimal ExtendedPrice()
{
    decimal decExtendedPrice = intQuantity * decPrice;
    return decExtendedPrice;
}

```

14. Thêm một CheckBox vào form:



The screenshot shows a Windows-style application window titled "Book Sales". Inside, there's a menu bar with "File". Below it is a section titled "Book Information" containing several input fields: "Title" (a long text box), "Quantity", "Price", "Extended Price", "Discount", and "Net Due" (all shorter text boxes). To the right of the "Discount" field, there is a checkbox labeled "Normal Discount".

Hình 3.2: Form sau khi thêm CheckBox Normal Discount

15. Sự kiện `mnuCalculate_Click` được xử lý lại:

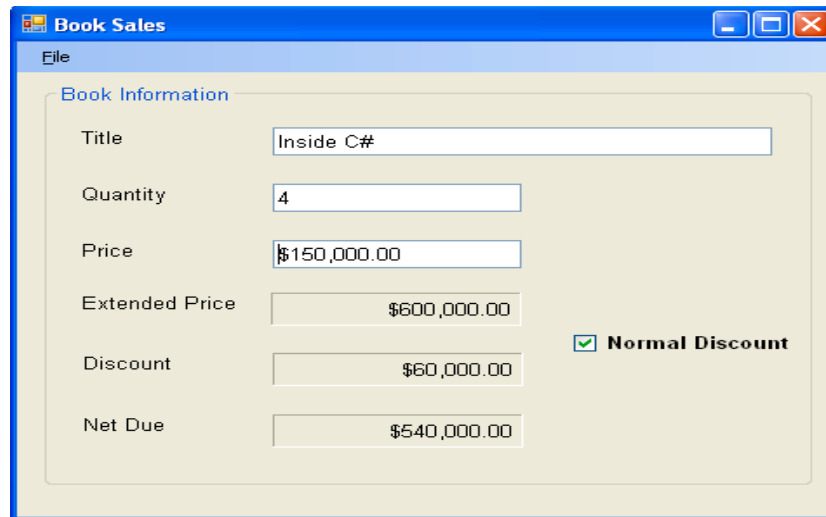
```

private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsBookSale();
        mBookSale.Title = txtTitle.Text;
        mBookSale.Quantity = Convert.ToInt32(txtQuan.Text);
        mBookSale.Price = Convert.ToDecimal(txtPrice.Text);
        mBookSale.Discount=chkDis.Checked;

        txtPrice.Text = mBookSale.Price.ToString("C");
        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
        lblDis.Text = mBookSale.DiscountAmount().ToString("C");
        lblNetDue.Text = mBookSale.NetDue().ToString("C");
    }
    catch(Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

```

16. Một kết quả thực thi chương trình:



The screenshot shows a Windows application window titled "Book Sales". Inside the window is a menu bar with "File" and a main area titled "Book Information". The form contains the following data:

Field	Value
Title	Inside C#
Quantity	4
Price	\$150,000.00
Extended Price	\$600,000.00
Discount	\$60,000.00
Net Due	\$540,000.00

There is also a checkbox labeled "Normal Discount" which is checked.

Hình 3.3: Một kết quả thực hiện chương trình

17. Cải tiến lớp `clsBookSale` bằng cách định nghĩa 1 phương thức xây dựng có tham số để gán trị cho các dữ liệu thành viên:

```
public clsBookSale(string title, int quality, decimal price, bool discount)
{
    this.Title=title;
    this.Quality=quality;
    this.Price=price;
    this.Discount=discount;
}
```

18. Sự kiện `mnuCalculate_Click` được viết lại:

```
private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsBookSale(txtTitle.Text, Convert.ToInt32(txtQuan.Text),
                                     Convert.ToDecimal(txtPrice.Text), chkDis.Checked);

        txtPrice.Text = mBookSale.Price.ToString("C");
        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
        lblDis.Text = mBookSale.DiscountAmount().ToString("C");
        lblNetDue.Text = mBookSale.NetDue().ToString("C");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

19. Thực thi chương trình và kiểm tra lại kết quả.

20. Đầu mỗi năm học (tháng 8, 9); nhà sách bán giảm giá 15% cho sinh viên giỏi. Điều đó dẫn tới giao diện của chương trình được thiết kế lại. Bên cạnh đó, lớp `clsBookSale` cũng được sửa đổi lại. Tuy nhiên ta có thể định nghĩa lớp mới `clsStudentBookSale` thừa kế từ lớp `clsBookSale`. Lớp mới này có thêm thành viên mới kiểu `bool` là `StudentDiscount` (true: sinh viên giỏi).

Lớp clsStudentBookSale:

```
class clsStudentBookSale:clsBookSale
{
    bool blnStudentDiscount;
    const float sngSTUDENT_DISCOUNT_RATE = 0.15F;

    public bool StudentDiscount
    {
        get
        {
            return blnStudentDiscount;
        }
        set
        {
            blnStudentDiscount=value;
        }
    }
}
```

Lớp clsStudentBookSale có phương thức xây dựng gọi lại phương thức xây dựng của lớp clsBookSale:

```
public clsStudentBookSale(string title, int quality, decimal price,
                          bool discount, bool student)
    :base(title,quality,price,discount)
{
    blnStudentDiscount=student;
}
```

21. Phương thức tính tiền giảm của lớp clsBookSale sẽ bị ghi đè:

```
public virtual decimal DiscountAmount()
```

22. Trong lớp clsStudentBookSale, phương thức DiscountAmount() được định nghĩa:

```

|
public override decimal DiscountAmount()
{
    decimal decDiscountAmount;

    decDiscountAmount=base.DiscountAmount();

    if (StudentDiscount)
        decDiscountAmount += ExtendedPrice() * Convert.ToDecimal(sngSTUDENT_DISCOUNT_RATE);

    return decDiscountAmount;
}

```

23. Giao diện chương trình được thiết kế lại:

Hình 3.4: Giao diện cuối cùng

24. Sự kiện `mnucalculate_Click` được định nghĩa lại:

```

private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsStudentBookSale(
            txtTitle.Text,
            Convert.ToInt32(txtQuan.Text),
            Convert.ToDecimal(txtPrice.Text),
            chkDis.Checked, chkDisStudent.Checked
        );

        txtPrice.Text = mBookSale.Price.ToString("C");
        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
        lblDis.Text = mBookSale.DiscountAmount().ToString("C");
        lblNetDue.Text = mBookSale.NetDue().ToString("C");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

```

25. Lưu dự án và thực thi ứng dụng.

2. Bài 2:

- Thiết kế lớp sinh viên với tối thiểu các thuộc tính: mã sinh viên, họ tên sinh viên, số tín chỉ (int), đơn giá mỗi tín chỉ (decimal), sinh viên có thuộc diện chính sách hay không (bool); bên cạnh những thuộc tính trên lớp sinh viên còn có phương thức tính tiền học phí phải đóng của sinh viên trong học kỳ đó, cách tính như sau: thành tiền = số tín chỉ * đơn giá 1 tín chỉ (với lưu ý rằng sinh viên thuộc diện chính sách sẽ được giảm 20% học phí).
- Thiết kế giao diện chính của ứng dụng là một MDI Form gồm các chức năng sau trên menu

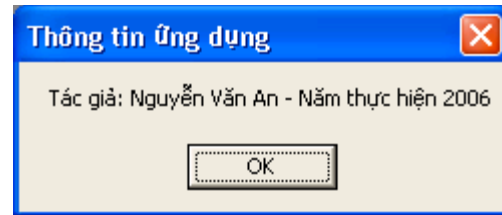
Học phí

Thông tin

- Thu học phí sinh viên
- Tổng học phí đã đóng
- Thoát
- Chương trình

Trong đó:

a. Thoát cho phép chấm dứt chương trình; *Chương trình* cho phép bật ra một hộp thông điệp ghi nhận thông tin về ứng dụng chẳng hạn như sau:



Hình 3.5: Thông tin ứng dụng

b. Thu học phí sinh viên cho phép bật ra form ghi nhận việc thu học phí của sinh viên như sau:

Học phí

Mã SV: 1040393

Tên SV: Nguyễn Văn An

Số tín chỉ: 28

Đơn giá: 48000

☐ Diện chính sách

Số tiền đóng: 1344000 đồng

Tính Xóa Thoát

Học phí

Mã SV: 1040394

Tên SV: Trần Thị Mỹ Hiền

Số tín chỉ: 30

Đơn giá: 48000

☒ Diện chính sách

Số tiền đóng: 1152000 đồng

Tính Xóa Thoát

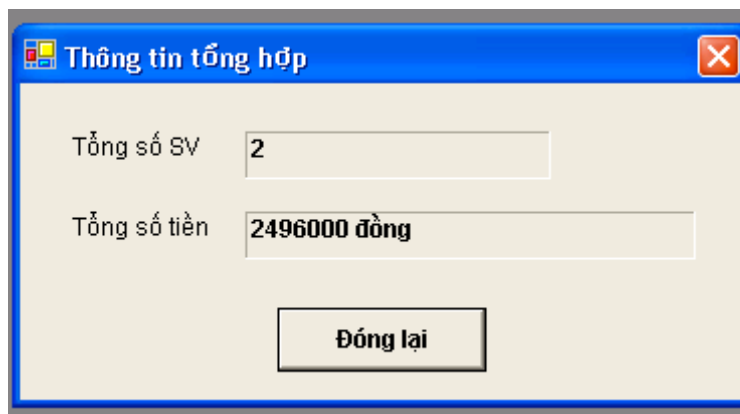
Hình 3.6: Giao diện thu học phí sinh viên

Nút *Tính* cho phép tính tổng học phí phải đóng. **Gợi ý:** Tạo đối tượng sinh viên và đặt các thuộc tính là các giá trị đã nhập.

Nút *Xóa* cho phép xóa trắng các ô nhập, các ô chứa kết quả và CheckBox *Diện chính sách* đặt lại trạng thái là *Không đánh dấu*.

Nút *Thoát* cho phép đóng form này lại.

c. *Tổng học phí đã đóng* cho phép hiển thị số lượng sinh viên đã đóng học phí trong ngày và tổng số tiền đã thu, form có thể có dạng sau:



The image shows a Windows-style dialog box with a blue title bar containing the text 'Thông tin tổng hợp' and a close button (X). The main area has a light beige background. It contains two labels with corresponding text boxes: 'Tổng số SV' followed by a text box containing the number '2', and 'Tổng số tiền' followed by a text box containing '2496000 đồng'. At the bottom center, there is a button with the text 'Đóng lại'.

Hình 3.7: Thông tin tổng hợp

Nút *Đóng lại* của form cho phép đóng lại form Thông tin tổng hợp.

Gợi ý: 1 Form là MDI form thì thuộc tính IsMdiContainer là **true**.

X. Bài tập 4 Truy cập cơ sở dữ liệu với ADO.NET

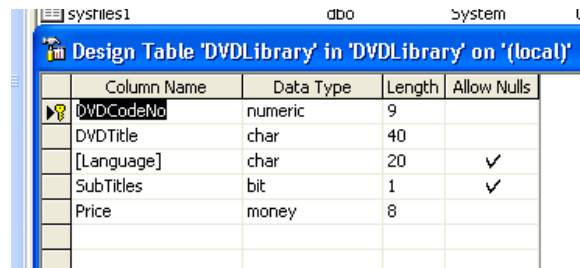
TRUY CẬP CƠ SỞ DỮ LIỆU VỚI ADO.NET

1. Bài 1:

Trong phần này chúng ta sẽ tìm hiểu cách thức làm việc với ADO.NET. Đây là một tập hợp các lớp cũng như các giao diện cho phép truy cập dữ liệu trong .NET Framework. Ở đây ta sẽ sử dụng ADO.NET để nối kết với cơ sở dữ liệu SQL Server, lấy dữ liệu và hiển thị chúng trên form.

Các bước thực hiện:

1. Trong SQL Server tạo một cơ sở dữ liệu tên là DVDLibrary.
2. Thêm một bảng tên DVDLibrary với cấu trúc như sau (Hình 4.1):



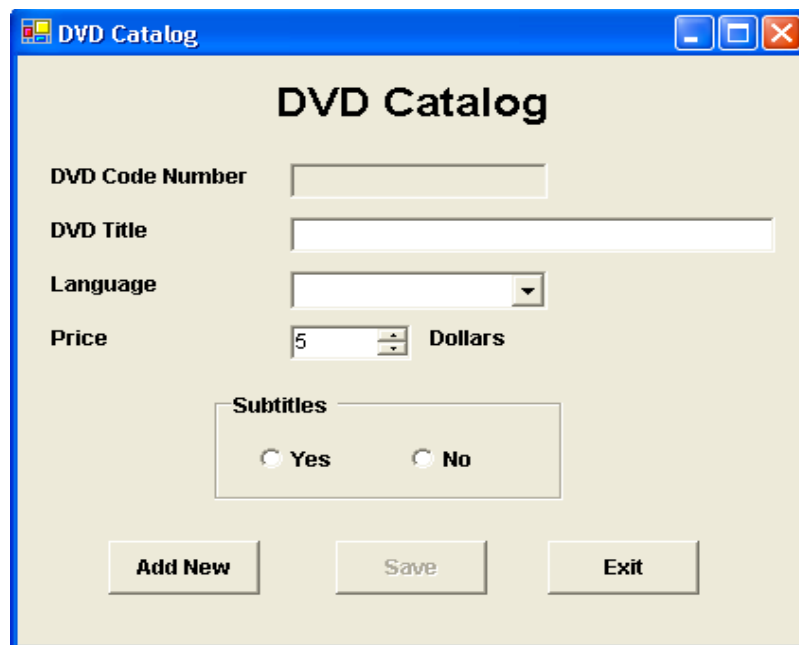
The screenshot shows the 'Design Table' window for 'DVDLibrary' in the 'DVDLibrary' database on '(local)'. The table has five columns: 'DVDCodeNo' (numeric, 9, no nulls), 'DVDTitle' (char, 40, no nulls), '[Language]' (char, 20, allow nulls), 'SubTitles' (bit, 1, allow nulls), and 'Price' (money, 8, no nulls). The 'DVDCodeNo' column is highlighted with a primary key icon.

Column Name	Data Type	Length	Allow Nulls
DVDCodeNo	numeric	9	
DVDTitle	char	40	
[Language]	char	20	✓
SubTitles	bit	1	✓
Price	money	8	

Hình 4.1: Cấu trúc bảng DVDLibrary

3. Thêm vài dòng dữ liệu vào bảng.
4. Tạo một ứng dụng WinForm tên prjDVD.

5. Đổi tên của form chính thành frmDVDCatalog.
6. Thêm một số control vào và sửa đổi một số thuộc tính tương ứng (bảng 4.3) sao cho ta được giao diện:



The screenshot shows a Windows-style application window titled "DVD Catalog". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a light beige background. At the top center, the text "DVD Catalog" is displayed in a large, bold, black font. Below this, there are four input fields arranged vertically on the left, each with a label and a corresponding control on the right: "DVD Code Number" with a single-line text box, "DVD Title" with a multi-line text box, "Language" with a dropdown menu, and "Price" with a numeric spinner box showing the value "5" and the text "Dollars" to its right. Below these fields is a "Subtitles" section containing two radio buttons labeled "Yes" and "No". At the bottom of the window, there are three buttons: "Add New", "Save", and "Exit", each in its own rectangular frame.

Hình 4.2: Giao diện ứng dụng

Control	Property	Value
Form	Name Text	frmDVDCatalog DVD Catalog
Label	Name Text	lblHeading DVD Catalog
Label	Name Text	lblDVDCoNo DVD Code Number
Label	Name Text	lblTitle DVD Title
Label	Name Text	lblLang Language
Label	Name Text	lblPrice Price
Label	Name Text	lblDollars Dollars
GroupBox	Name Text	grpSubTitles Subtitles
RadioButton	Name Text Checked	radYes Yes true
RadioButton	Name Text Checked	radNo No false
ComboBox	Name Collection	cboLang English Spanish French Chinese Russian

NumericUpDown	Name Maximum Minimum Value	updPrice 150 5 5
TextBox	Name ReadOnly Text	txtCodeNo true
TextBox	Name Text	txtTitle
Button	Name Text	btnAddNew AddNew
Button	Name Text	btnSave Save
Button	Name Text	btnExit Exit

Hình 4.3: Giá trị một số thuộc tính của các controls

7. Chuyển sang cửa sổ Code View, tham chiếu đến một số không gian tên:

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

8. Khai báo những biến sau trong phần khai báo của form:

```
private SqlConnection conn;
private SqlCommand commd;
private string sqlstr, lang;
private int codeno, subtitles;
decimal price;
```

9. Viết đoạn mã sau vào phương thức xử lý sự kiện Load của form:

```

private void frmCatalog_Load(object sender, System.EventArgs e)
{
    try
    {
        conn = new SqlConnection("Server=HoaiBao\\HoaiBao;Initial
        Catalog=DVDLibrary;User Id=user;pwd=user;");
        commd = new SqlCommand("Select * From DVDLibrary",conn);
        conn.Open();
        btnSave.Enabled = false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + ex.StackTrace);
    }
}

```

10. Khi button AddNew được click, một đối tượng SqlCommand được tạo ra thực hiện câu truy vấn lấy về mã số lớn nhất của DVD Code No từ bảng from the DVDLibrary. Nếu bảng rỗng, một ngoại lệ được tạo ra. Vì thế ta phải bắt lại ngoại lệ này nhằm kiểm tra khả năng bảng rỗng (lúc đó giá trị của biến codeno được gán là 0). Sau đó giá trị của codeno tăng 1 để có giá trị hiện hành của codeno. Kế tiếp các trường của form được xóa trắng nhờ vào việc gọi thực thi phương thức ResetFields() để một mẫu tin mới được thêm vào và kích hoạt button Save.

Sự kiện btnAddNew_Click:

```

private void btnAddNew_Click(object sender, System.EventArgs e)
{
    //calculate new Code
    commd = new SqlCommand("Select max(DVDCoNo) From
        DVDLibrary", conn);
    try
    {
        codeno = Convert.ToInt32(commd.ExecuteScalar());
        btnAddNew.Enabled = false;
        btnSave.Enabled=true;
    }
    catch(Exception ex)
    {
        codeno=0;
        MessageBox.Show(ex.Message);
    }

    codeno++;
    txtCodeNo.Text = codeno.ToString();
    ResetFields();
}

```

11. Sự kiện Click của button btnExit được xử lý để đóng lại nối kết dữ liệu và kết thúc ứng dụng:

```

private void btnExit_Click(object sender, System.EventArgs e)
{
    conn.Close();
    Application.Exit();
}

```

12. Sự kiện btnSave_Click nhằm thêm một dòng dữ liệu mới vào bảng. Bởi vì đây không phải là một câu truy vấn lấy về dữ liệu mà là một câu truy vấn hành động nên ta sử dụng phương thức ExecuteNonQuery của lớp SqlCommand:

```

private void btnSave_Click(object sender, System.EventArgs e)
{
    try
    {
        sqlstr = "Insert Into DVDLibrary Values(" +
            codeno.ToString() + "," + txtTitle.Text + ", '" +
            lang + "'," + subtitles.ToString() +
            "," + price.ToString() + ")";
        MessageBox.Show(sqlstr);
        commd = new SqlCommand(sqlstr, conn);
        commd.ExecuteNonQuery();
        MessageBox.Show("Record has been add successfully");
        ResetFields();
        btnExit.Focus();
        btnAddNew.Enabled = true;
        btnSave.Enabled=false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + ex.StackTrace);
    }
}

```

13. Sự kiện ValueChanged của NumericUpDown được xử lý:

```

private void updPrice_ValueChanged(object sender,
    System.EventArgs e)
{
    price = updPrice.Value;
}

```

14. Sự kiện SelectedIndexChanged của ComboBox được xử lý:

```
private void cboLang_SelectedIndexChanged(object sender,
    System.EventArgs e)
{
    lang = cboLang.SelectedItem.ToString();
}
```

15. Sự kiện CheckedChanged của RadioButton được xử lý:

```
private void radYes_CheckedChanged(object sender,
    System.EventArgs e)
{
    subtitles = 1;
}

private void radNo_CheckedChanged(object sender,
    System.EventArgs e)
{
    subtitles = 0;
}
```

16. Đoạn mã cho phương thức ResetFields:

```
private void ResetFields()
{
    txtTitle.Text = "";
    price = updPrice.Value=5;
    lang=cboLang.Items[0].ToString();
    cboLang.Text = "";
    radYes.Checked = true;
    radNo.Checked= false;
    btnSave.Enabled=true;
}
```

17. Lưu, biên dịch và thực thi dự án. Một thí dụ mẫu cho việc thêm vào một mẫu tin như trong hình 4.4:

The image shows a Windows-style application window titled "DVD Catalog". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a light beige background. At the top, the title "DVD Catalog" is centered in a large, bold, black font. Below the title, there are four input fields arranged vertically on the left, each with a label and a corresponding input area on the right. The first field is labeled "DVD Code Number" and contains the value "7". The second field is labeled "DVD Title" and contains the text "The Aviator". The third field is labeled "Language" and is a dropdown menu currently showing "English". The fourth field is labeled "Price" and contains the value "25", followed by the text "Dollars". Below these fields, there is a section titled "Subtitles" which contains two radio buttons: "Yes" (which is unselected) and "No" (which is selected). At the bottom of the window, there are three buttons: "Add New", "Save", and "Exit", each in its own rectangular box.

DVD Catalog

DVD Code Number: 7

DVD Title: The Aviator

Language: English

Price: 25 Dollars

Subtitles: ☐ Yes ☒ No

Add New Save Exit

Hình 4.4: Kết quả thực thi chương trình

2. Bài 2:

Sửa đổi ứng dụng ở bài 1 để thêm chức năng cập nhật và xóa các mẫu tin. Giao diện chương trình trở thành:



The screenshot shows a Windows-style application window titled "DVD Catalog". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is a light beige color. At the top, the title "DVD Catalog" is centered. Below it, there are several input fields: "DVD Code Number" (a single-line text box), "DVD Title" (a multi-line text box), "Language" (a dropdown menu), and "Price" (a numeric spinner box set to 5, followed by the text "Dollars"). Below these is a "Subtitles" section with two radio buttons, "Yes" (selected) and "No". At the bottom of the window, there is a row of six buttons: "Add New", "Update", "Delete", "Save", "Cancel", and "Exit".

Hình 4.5: Giao diện mới của chương trình

Ứng dụng sửa đổi có một số chức năng:

1. Ban đầu các buttons Save & Cancel bị vô hiệu hóa (disable).
2. Khi người dùng click buttons Update hay Delete, textbox DVD Code Number sẽ bị thay thế bởi 1 Combobox để người dùng có thể lựa chọn giá trị các mã DVD (DVDCode) từ ComboBox. Nếu buttons Update (AddNew hay Delete) được click; các buttons AddNew, Update, Delete bị vô hiệu hóa; các button Save & Cancel trở nên có hiệu lực (enable).
3. Dựa trên mã DVD (DVDCode) từ Combobox, thông tin chi tiết về đĩa DVD có mã được lựa chọn sẽ được hiển thị chi tiết trên form.
4. Khi người dùng thay đổi giá trị của đĩa DVD hiện hành và click Save, các thông tin sửa đổi sẽ được lưu vào cơ sở dữ liệu; các buttons AddNew và Delete trở nên có hiệu lực (buttons Save & Cancel mất hiệu lực).

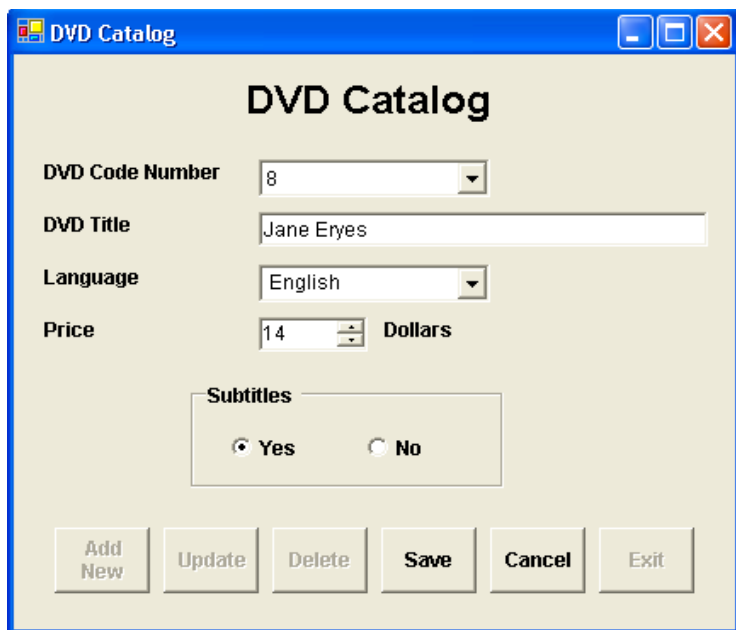
5. Tương tự khi người dùng click Delete, một thông điệp để người dùng xác nhận có muốn xóa thông tin của DVD hiện hành không? Nếu người dùng xác nhận là muốn xóa, thông tin về DVD hiện hành bị xóa khỏi cơ sở dữ liệu.

6. Khi người dùng click button AddNew sau khi đã thực hiện các thao tác như Update hay Delete; Combobox DVD Code No sẽ bị thay thế bởi textbox DVD Code Number.

7. Khi người dùng click button Cancel; không có sự thay đổi nào xảy ra; mọi thứ trở lại bình thường như cũ.

Gợi ý

- Tạo 1 Combobox và 1 Textbox cho mã DVD (DVD Code No) ở cùng 1 vị trí trên form để chúng chồng lấp lên nhau.
- Dùng thuộc tính Visible để thiết lập việc ẩn/hiện của 2 controls này.
- Dùng thuộc tính Enabled để cho phép các buttons có hiệu lực (enable) hay mất hiệu lực (disable).



The screenshot shows a Windows-style application window titled "DVD Catalog". Inside the window, there is a form with the following fields and controls:

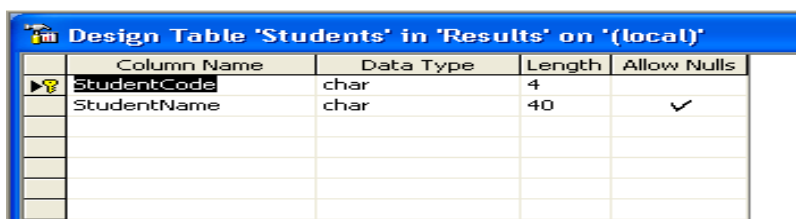
- DVD Code Number:** A dropdown menu currently showing the value "8".
- DVD Title:** A text box containing the text "Jane Eryes".
- Language:** A dropdown menu currently showing the value "English".
- Price:** A numeric spinner box showing "14", followed by the text "Dollars".
- Subtitles:** A section containing two radio buttons, "Yes" (which is selected) and "No".
- Buttons:** A row of six buttons at the bottom: "Add New", "Update", "Delete", "Save", "Cancel", and "Exit".

Hình 4.6: Kết quả mới của ứng dụng

Hình 4.6 là một thí dụ khi người dùng click button Update và chọn mã DVD từ combobox (Các buttons AddNew, Update, Delete bị mất hiệu lực và các buttons Save, Cancel có hiệu lực).

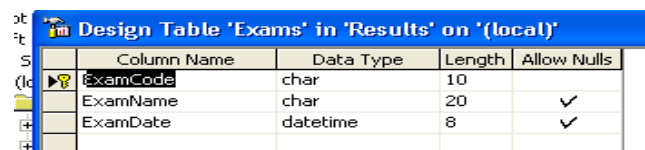
3. Bài 3:

Tạo một cơ sở dữ liệu tên Results trong SQLServer và 03 bảng có cấu trúc như trong hình 4.7, 4.8, 4.9. Tạo một ứng dụng Winforms để hiển thị thông tin từ các bảng này trong 1 DataGridView. Bên cạnh đó, ứng dụng còn có chức năng tìm kiếm thông tin của 1 sinh viên với mã sinh viên nhập từ bàn phím



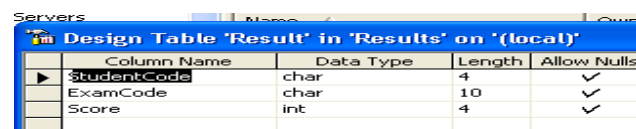
Column Name	Data Type	Length	Allow Nulls
StudentCode	char	4	
StudentName	char	40	✓

Hình 4.7: Cấu trúc bảng Students



Column Name	Data Type	Length	Allow Nulls
ExamCode	char	10	
ExamName	char	20	✓
ExamDate	datetime	8	✓

Hình 4.8: Cấu trúc bảng Exams



Column Name	Data Type	Length	Allow Nulls
StudentCode	char	4	✓
ExamCode	char	10	✓
Score	int	4	✓

Hình 4.9: Cấu trúc bảng Result

Giao diện của ứng dụng có thể như hình 4.10:

The screenshot shows a window titled "Exam Results" with a search bar labeled "Student Code" and a "Search" button. Below the search bar is a table with the following data:

	StudentCode	StudentName	ExamCode	ExamName	Score
	03	Alice	NET01	Introduction to NET	33
	03	Alice	NET02	Winform	55
	06	Zidane	JAV01	Java Basics	50
	03	Alice	JAV01	Java Basics	60
	06	Zidane	JAV02	Advanced Java	80
	06	Zidane	NET01	Introduction to NET	57
	06	Zidane	NET02	Winform	67
	07	Ballack	JAV01	Java Basics	50
	07	Ballack	JAV02	Advanced Java	45
	07	Ballack	NET01	Introduction to NET	55
	07	Ballack	NET02	Winform	75
*					

Hình 4.10: DataGridView hiển thị thông tin từ 03 bảng

Khi người dùng lựa chọn một mã sinh viên trong combobox, kết quả của sinh viên đó sẽ được hiển thị như hình 4.11

The screenshot shows the same "Exam Results" window, but the "Student Code" dropdown is now set to "07". The table displays only the results for that student:

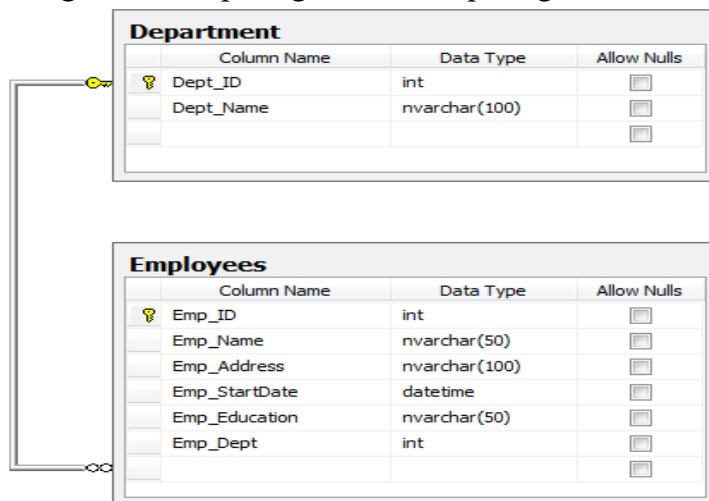
	StudentCode	StudentName	ExamCode	ExamName	Score
▶	07	Ballack	JAV01	Java Basics	50
	07	Ballack	JAV02	Advanced Java	45
	07	Ballack	NET01	Introduction to NET	55
	07	Ballack	NET02	Winform	75
*					

Hình 4.11: Tìm kiếm kết quả 1 sinh viên

Gợi ý: Dùng lớp DataAdapter và DataSet.

XI. Bài tập 5 Truy cập Cơ sở dữ liệu với ADO.NET (Tiếp theo)

Xây dựng form cho phép thêm mới nhân viên trong một công ty. Mỗi nhân viên có mã số, tên, địa chỉ, ngày bắt đầu làm việc, trình độ học vấn (Undergraduate, Postgraduate); mỗi nhân viên thuộc 1 phòng ban duy nhất. Mã số nhân viên là một số nguyên tự động được tạo ra bởi hệ thống. Mỗi phòng ban gồm mã số phòng ban và tên phòng ban.



Hình 5.1: CSDL Human

Bước 1: Tạo CSDL Human như hình 5.1.

Bước 2: Thêm dữ liệu vào bảng Department như sau:

Dept_ID	Dept_Name
1	Human Resources
2	R&D
3	Marketing

Hình 5.2: Dữ liệu trong bảng Department

Bước 3: Tạo ứng dụng WinForm có tên là Employees

Bước 4: Thêm một lớp quản lý việc mở và đóng kết nối đến CSDL của hệ SQL Server. Lớp này có các dữ liệu thành viên static: tên server,

tên CSDL, tài khoản, mật khẩu và 1 đối tượng SqlConnection. Ngoài ra lớp còn có 2 phương thức:

- bool CreateConnection(): Mở nối kết đến CSDL.
- bool CloseConnection(): Đóng nối kết đến CSDL.

Bước 4: Tạo form thêm mới nhân viên:

The screenshot shows a Windows application window titled "Employee details". Inside the window is a form titled "Employee Details" with the following fields and controls:

- Employee ID:** A single-line text input field.
- Employee Name:** A single-line text input field.
- Employee Address:** A multi-line text input field.
- Date of Joining:** A date-time picker control showing "24/07/2012".
- Education:** A list box with "UnderGraduate" selected.
- Department:** A combobox with "Human Resources" selected.
- Buttons:** "Add New", "Save", and "Exit" buttons at the bottom.

Annotations with arrows point to the following controls:

- Textbox:** Points to the Employee ID, Employee Name, and Employee Address fields.
- DateTimePicke:** Points to the Date of Joining date-time picker.
- Listbox:** Points to the Education list box.
- Combobox:** Points to the Department combobox.

Text next to the Listbox annotation specifies the items: *-UnderGraduate* and *-PostGraduate*.

Hình 5.3: Form thêm mới nhân viên

Bước 5: Mở mã lệnh của form thêm mới nhân viên, sử dụng 2 không gian tên: System.Data

System.Data.SqlClient

Bước 6: Trong lớp của form thêm mới nhân viên, thêm 2 dữ liệu thành viên: DataSet ds;

int ID; //Dùng cho việc tự tăng mã số

Bước 7: Mỗi khi form thêm mới được nạp lên màn hình, combobox Department được điền dữ liệu từ bảng Department của CSDL. Sự kiện load của form thêm mới được xử lý:

```
private void frmEmployee_FormClosed(object sender, FormClosedEventArgs e)
{
    clsConnection.CloseConnection();
    Application.Exit();
}

private void frmEmployee_Load(object sender, EventArgs e)
{
    clsConnection.CreateConnection();
    ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter("Select * From Department",clsConnection.con);
    da.Fill(ds,"Dept");
    cboDe.DataSource = ds.Tables["Dept"];
    cboDe.DisplayMember = "Dept_Name";
    cboDe.ValueMember = "Dept_ID";
}
```

Lưu ý: Khi đóng form lại, kết nối tới CSDL được đóng lại.

```

private void btnAdd_Click(object sender, EventArgs e)
{
    txtID.Clear();
    txtName.Clear();
    txtAdd.Clear();
    cboDe.SelectedIndex = -1;
    lstEdu.SelectedIndex = -1;
    dtpJoin.Value = DateTime.Now;
    btnAdd.Enabled = false;
    btnSave.Enabled = true;
    txtName.Focus();
    //
    try
    {
        SqlCommand com = new SqlCommand("Select Max(Emp_ID) From Employees", clsConnection.con);
        ID = Convert.ToInt32(com.ExecuteScalar()) + 1;
    }
    catch (Exception)
    {
        ID = 1;
    }
    txtID.Text = ID.ToString();
}

```

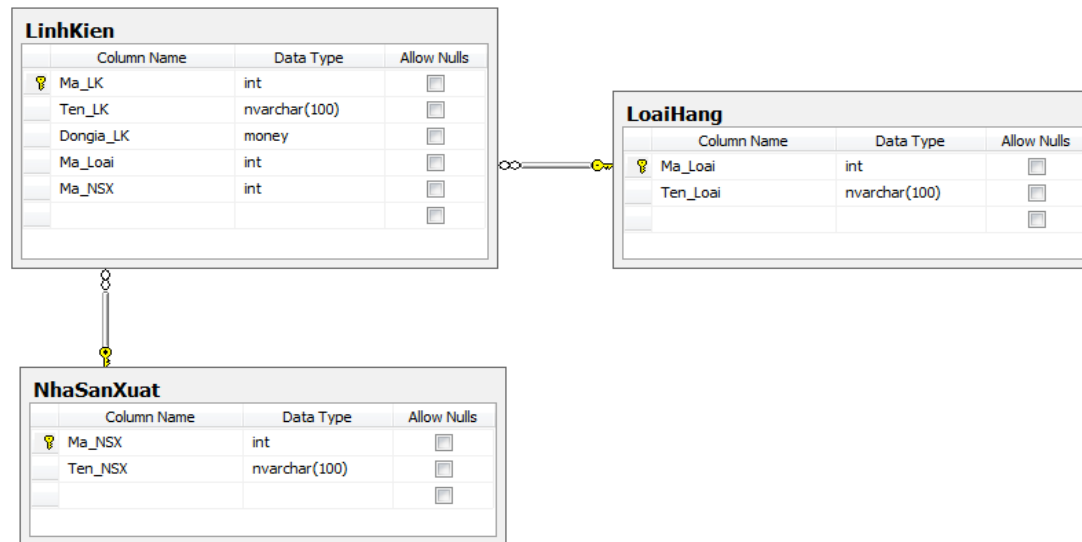
Bước 8: Khi người dùng click AddNew, các ô nhập bị xóa trắng, các danh sách về trạng thái không lựa chọn, ngày bắt đầu làm việc là ngày hiện hành, button AddNew ẩn, button Save bật. Một mã số mới được tạo ra cho nhân viên mới.

Bước 9: Khi người dùng nhập đầy đủ thông tin, nhấp Save, các thông tin nhập sẽ được lưu vào CSDL. Sự kiện click của button Save:

```
private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        string sql = "Insert Into Employees(Emp_ID, Emp_Name, Emp_Address, Emp_StartDate, Emp_Education, Emp_Dept) " +
            " Values(@empid, @name, @add, @date, @edu, @dept)";
        SqlCommand com = new SqlCommand(sql, clsConnection.con);
        SqlParameter p1 = new SqlParameter("@empid", SqlDbType.Int);
        p1.Value = Convert.ToInt32(txtID.Text);
        SqlParameter p2 = new SqlParameter("@name", SqlDbType.NVarChar);
        p2.Value = txtName.Text;
        SqlParameter p3 = new SqlParameter("@date", SqlDbType.DateTime);
        p3.Value = dtpJoin.Value;
        SqlParameter p4 = new SqlParameter("@edu", SqlDbType.NVarChar);
        p4.Value = lstEdu.Text;
        SqlParameter p5 = new SqlParameter("@dept", SqlDbType.Int);
        p5.Value = Convert.ToInt32(cboDe.SelectedValue);
        SqlParameter p6 = new SqlParameter("@add", SqlDbType.NVarChar);
        p6.Value = txtAdd.Text;
        com.Parameters.Add(p1);
        com.Parameters.Add(p2);
        com.Parameters.Add(p3);
        com.Parameters.Add(p4);
        com.Parameters.Add(p5);
        com.Parameters.Add(p6);
        com.ExecuteNonQuery();
        MessageBox.Show("Add successfully");
        btnAdd.Enabled=true;
        btnSave.Enabled=false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

XII. Bài tập 6

Xây dựng form cho phép thêm mới các linh kiện máy tính. Mỗi linh kiện có mã số, tên, đơn giá; mỗi linh kiện thuộc 1 loại linh kiện duy nhất và được cung cấp bởi 1 nhà cung cấp. Mã số linh kiện là một số nguyên tự động được tạo ra bởi hệ thống. Mỗi loại linh kiện có mã số và tên loại. Tương tự mỗi nhà cung cấp có mã nhà cung cấp và tên nhà cung cấp



Hình 6.1: CSDL QLLinhKien

Bước 1: Tạo CSDL QLLinhKien như hình 6.1.

Bước 2: Thêm dữ liệu vào bảng LoaiHang và NhaSanXuat như sau:

Ma_Loai	Ten_Loai
1	HDD
2	CPU
3	Mainboard
4	RAM
5	Mouse
6	Monitor
7	Keyboard

Ma_NSX	Ten_NSX
1	Intel
2	Samsung
3	CMC
4	Microsoft

Hình 6.2: Dữ liệu trong bảng LoaiHang và NhaSanXuat

Bước 3: Tạo ứng dụng WinForm có tên là QLLinhKien

Bước 4: Thêm một lớp quản lý việc mở và đóng kết nối đến CSDL của hệ SQL Server. Lớp này có các dữ liệu thành viên static: tên server, tên CSDL, tài khoản, mật khẩu và 1 đối tượng SqlConnection. Ngoài ra lớp còn có 2 phương thức:

- bool CreateConnection(): Mở nối kết đến CSDL.
- bool CloseConnection(): Đóng nối kết đến CSDL.

Bước 4: Tạo form thêm mới linh kiện:

Hình 6.3: Form thêm mới linh kiện

Bước 5: Mỗi khi form thêm mới được nạp lên màn hình, combobox loại linh kiện và nhà sản xuất được điền dữ liệu từ bảng LoaiHang và NhaSanXuat của CSDL. Button Thêm bật, Lưu ẩn.

Lưu ý: Khi đóng form lại, kết nối tới CSDL được đóng lại.

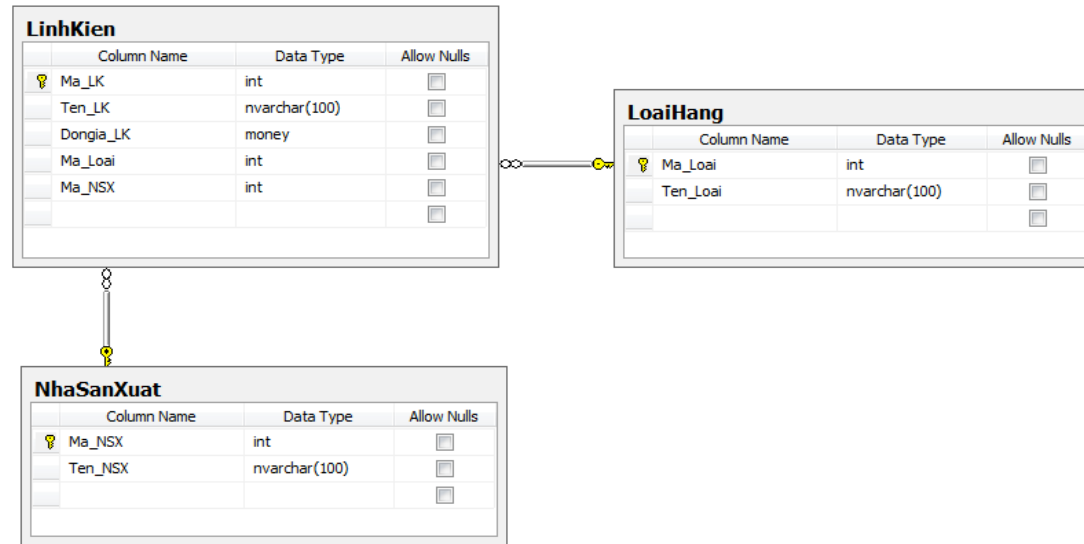
Bước 6: Khi người dùng click AddNew, các ô nhập bị xóa trắng, các danh sách về trạng thái không lựa chọn, button Thêm ẩn, button Lưu bật. Một mã số mới được tạo ra cho linh kiện mới.

Bước 7: Khi người dùng nhập đầy đủ thông tin, nhấp Lưu, các thông tin nhập sẽ được lưu vào CSDL. Các button trở về như ở bước 5.

Bước 8: Khi người dùng nhấp Thoát, chương trình kết thúc.

XIII. Bài tập 7: TẠO BÁO BIỂU VỚI CRYSTAL REPORT

Sử dụng lại chương trình và CSDL của buổi 6.



Hình 6.1: CSDL QLLinhKien

Bước 1: Tạo Báo biểu để in báo cáo linh kiện theo mẫu

BÁO CÁO (Mẫu 1)

Mã Linh kiện	Tên linh kiện	Đơn giá	Tên nhà Sản xuất
X	X	X	X
X	X	X	X

Bước 2: Tạo báo biểu để in báo cáo linh kiện theo Loại hàng

BÁO CÁO (Mẫu 2)

Loại hàng: xxxx

Mã Linh kiện	Tên linh kiện	Đơn giá
x	x	x
x	x	x
Tổng: xxxx		

Loại hàng: xxxx

Mã Linh kiện	Tên linh kiện	Đơn giá
x	x	x
x	x	x
Tổng: xxxx		

...

Bước 3. Điều chỉnh báo biểu ở bước 2 bằng cách thêm tham số: `loai_hang`

Thiết kế Form in báo biểu có thêm ô loại hàng để nhập loại hàng và truyền tham số này cho báo biểu.