

**THỰC HÀNH LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS**

Ấn bản 2021

# MỤC LỤC

MỤC LỤC .....	I
HƯỚNG DẪN.....	II
BÀI 1: ỨNG DỤNG CONSOLE NGÔN NGỮ C# .....	1
1.1 MỤC TIÊU .....	1
1.2 BÀI TẬP.....	2
BÀI 2: LẬP TRÌNH WINDOWS FORM VỚI CONTROLS CƠ BẢN.....	13
2.1 MỤC TIÊU .....	13
2.2 HƯỚNG DẪN LÀM QUEN VỚI WINDOWS FORM.....	14
2.3 BÀI TẬP.....	19
BÀI 3: LẬP TRÌNH WINDOWS FORM VỚI GIAO DIỆN MDI.....	27
3.1 MỤC TIÊU .....	27
3.2 BÀI TẬP.....	27
BÀI 4: LẬP TRÌNH VỚI CƠ SỞ DỮ LIỆU SỬ DỤNG ENTITY .....	36
4.1 MỤC TIÊU .....	36
4.2 BÀI TẬP.....	36
BÀI 5: TẠO BÁO CÁO THỐNG KÊ VỚI DATA REPORT.....	51
5.1 MỤC TIÊU .....	51
5.2 HƯỚNG DẪN TẠO DATA REPORT .....	51
5.3 BÀI TẬP.....	58
BÀI 6: ÔN TẬP VÀ KIỂM TRA .....	65

# HƯỚNG DẪN

## MÔ TẢ MÔN HỌC

Lập trình C# trên Windows là một trong những môn học nhằm cung cấp kiến thức cơ bản cho những ai muốn xây dựng được ứng dụng trên môi trường trên hệ điều hành Windows. Môn học này cung cấp những kiến thức cơ bản, tổng quan về công nghệ mới của Microsoft: .Net Framework, lập trình hướng đối tượng trên .Net, cũng như giới thiệu bộ thư viện .Net mà cung cấp sẵn (LINQ, Entity framework...), sinh viên có thể tạo giao diện ứng dụng, liên kết nối các hệ cơ sở dữ liệu. Từ đó sinh viên có thể bước đầu có thể xây dựng những ứng dụng tin học hóa, hỗ trợ các doanh nghiệp trong việc quản lý hàng hóa, nhân sự, mua bán, v.v.

## NỘI DUNG THỰC HÀNH

- Bài 1. Tổng quan ngôn ngữ C# trên ứng dụng Console: Bài này cung cấp cho học viên làm quen với ngôn ngữ C#. Với phương pháp lập trình hướng đối tượng, sử dụng thư viện LINQ để giúp truy vấn nhanh chóng.
- Bài 2. Lập trình với ứng dụng giao diện Windows Form với các control cơ bản. Học viên được thiết kế giao diện, làm quen với các controls như: TextBox, Button, ComboBox, RadioButton, DataGridView... Bên cạnh đó việc viết code cho các event trên giao diện người dùng.
- Bài 3. Sử dụng EntityFrameWork để kết nối với cơ sở dữ liệu SQL Server: Học viên được hướng dẫn cách sử dụng mô hình Code First trên EntityFrameWork với hướng tiếp cận có sẵn cơ sở dữ liệu. Biết cách sử dụng Entity để thực hiện các thao tác trên cơ sở dữ liệu
- Bài 4. Lập trình với giao diện MDI: Giúp học viên tìm hiểu thêm 1 số controls nâng cao trong Windows Form như MenuStrip, ToolStrip, Timer...
- Bài 5. Tạo báo cáo thông kê với Data Report: Hướng dẫn cách sử dụng Microsoft Report có sẵn trên VS.NET để tạo báo cáo thông kê. Bên cạnh đó cũng ôn tập lại cách sử dụng Entity FrameWork Code First để thể hiện dữ liệu trên report từ CSDL
- Bài 6: Bài thi kết thúc thực hành giúp đánh giá quá trình học tập của học viên.

## KIẾN THỨC TIỀN ĐỀ

Môn học Lập trình C# trên Windows đòi hỏi sinh viên có nền tảng về Lập trình C, và lập trình hướng đối tượng.

## YÊU CẦU MÔN HỌC

Người học phải dự học đầy đủ các buổi lên lớp và làm bài tập đầy đủ ở nhà.

## CÁCH TIẾP NHẬN NỘI DUNG MÔN HỌC

Để học tốt môn này, người học cần ôn tập các bài đã học, trả lời các câu hỏi và làm đầy đủ bài tập; đọc trước bài mới và tìm thêm các thông tin liên quan đến bài học.

Đối với mỗi bài học, người học đọc trước mục tiêu và tóm tắt bài học, sau đó đọc nội dung bài học. Kết thúc mỗi ý của bài học, người đọc trả lời câu hỏi ôn tập và kết thúc toàn bộ bài học, người đọc làm các bài tập.

## PHƯƠNG PHÁP ĐÁNH GIÁ THỰC HÀNH

Môn học được đánh giá gồm:

- Điểm thực hành hàng tuần (50%): Hình thức tham gia và làm bài tập trên các buổi thực hành theo yêu cầu của giáo viên.
- Điểm thi thực hành cuối kỳ (50%): Hình thức thi thực hành, phù hợp với quy chế đào tạo và tình hình thực tế tại nơi tổ chức học tập.



# BÀI 1: ỨNG DỤNG CONSOLE NGÔN NGỮ C#

## 1.1 MỤC TIÊU

- Hướng dẫn sinh viên làm quen với ngôn ngữ lập trình C#: qua việc viết các ứng dụng
- Console trong vs. NET 2015/ 2017/ 2019.
- Xây dựng các lớp, tạo đối tượng, truy xuất các phương thức, ...
- Soạn thảo mã nguồn, biên dịch, debug, thực thi chương trình...
- Kế thừa trong lập trình hướng đối tượng trên C#.
- Tìm hiểu về sử dụng thư viện LINQ của .NET
- Khuyến khích sinh viên sử dụng chuẩn viết code C# ( C# Coding Convetion)
- Ngôn ngữ C#

- Kiểu dữ liệu: `bool`, `decimal`, `double`, `float`, `int`, `string`, `DateTime`, `bool?`, `decimal?`, `double?`, `float?`, `int?`, `long?`, `DateTime?`, `object`, `var`, `dynamic`...
- Chuyển đổi/ép kiểu dữ liệu: `as`, `is`, Thư viện `Convert`
- Vòng lặp: `for`, `foreach`, `while`, `do...while` và Lệnh điều khiển `break`, `continue`, `return`
- Một số phương thức của thư viện **Console**

`Console.Write(<giá trị cần in ra màn hình>);`

`Console.WriteLine();` // Sử dụng lệnh in ra màn hình có xuống dòng

`Console.ReadLine();` // Đọc dữ liệu chuỗi từ bàn phím cho đến khi gặp ký tự xuống dòng

`Console.ReadKey();` // Dừng màn hình để xem kết quả.

- Coding Convention C#: Đưa ra các quy ước khi coding với ngôn ngữ lập trình C#, với các quy tắc này giúp tiết kiệm thời gian rất trong quá trình phát triển và bảo trì phần mềm.

Ví dụ: Đặt tên class dùng danh từ, đối với phương thức dùng động từ ...

Tên biến, tên phương thức thể hiện được ý nghĩa

Nên comment những đoạn code khó hiểu hoặc có chức năng đặc biệt

## 1.2 BÀI TẬP

**Bài tập 1A:** Viết chương trình cho phép người dùng nhập vào tổng số N sinh viên, mỗi sinh viên phải nhập vào các thông tin của sinh viên( Mã số sinh viên, Họ tên sinh viên, Điểm TB và tên Khoa) sau đó lần lượt tạo các đối tượng sinh viên và đưa vào mảng Student theo những thông tin do user nhập vào (dùng vòng lặp for). Cuối cùng xuất ra danh sách chi tiết thông tin sinh viên.

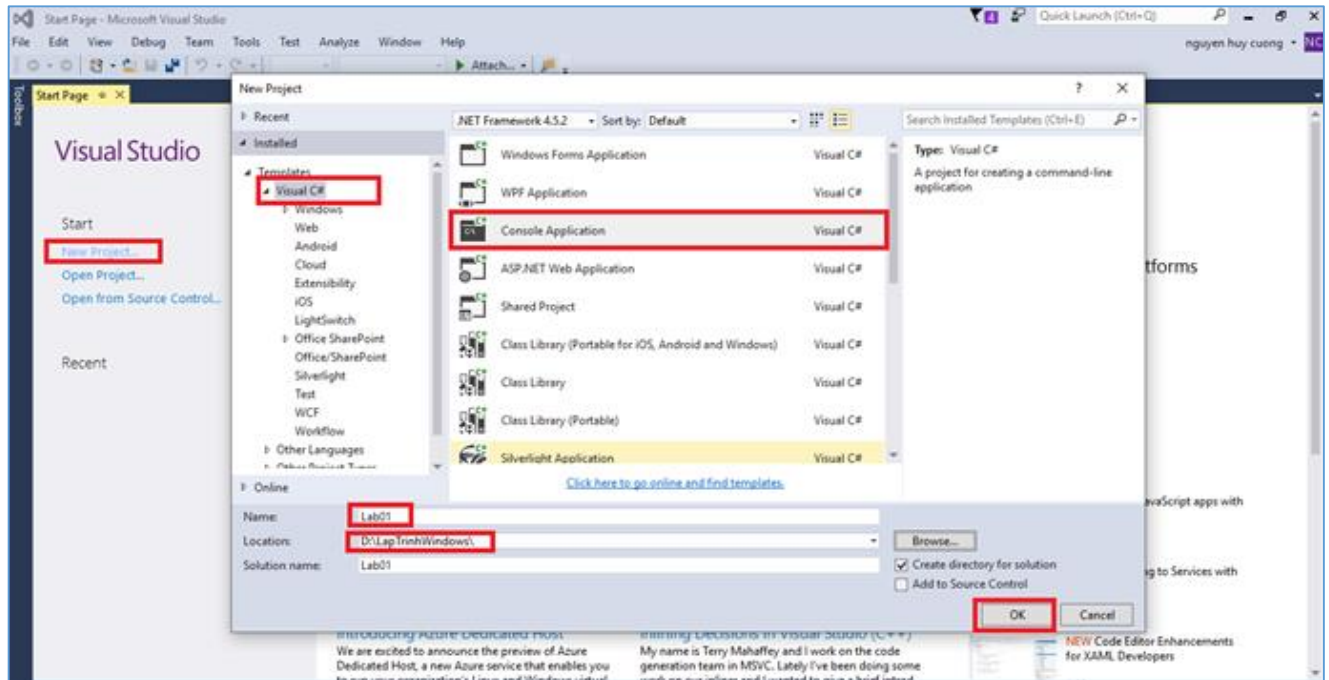
### Yêu Cầu

- ✓ Sinh viên xây dựng chương trình theo nội dung mô tả bên trên. Không viết lệnh nhập và xuất danh sách sinh viên trực tiếp trong hàm **main()** mà hãy viết hai phương thức nhập và xuất thông tin sinh viên ở lớp **Student**.
- ✓ Compile & Build chương trình.
- ✓ Run chương trình ở hai chế độ debug và không debug.
- ✓ Chạy từng bước chương trình trong chế độ debug: dùng breakpoint hoặc chạy từng dòng lệnh. Kiểm tra những giá trị của các biến trong chương trình ở cửa sổ Watch.

### Hướng Dẫn

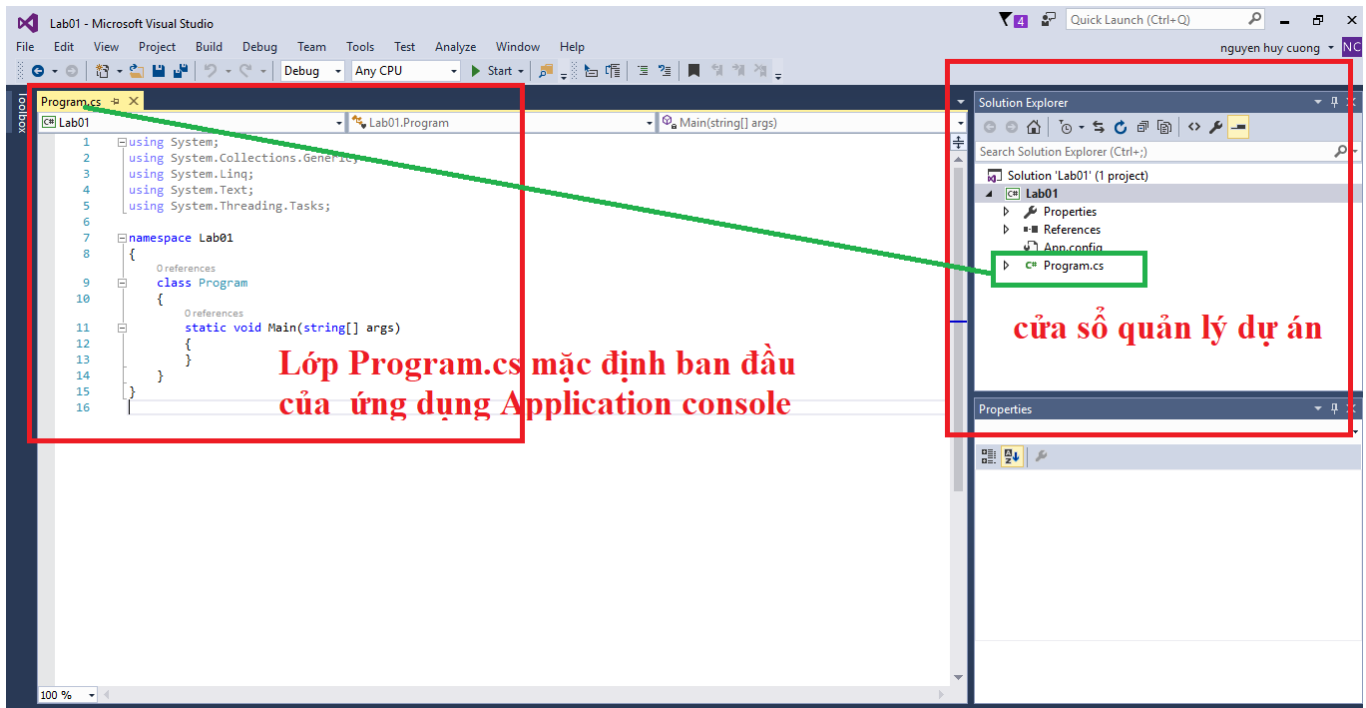
#### **Bước 1:** Tạo project mới trong VS 2015

- Mở Visual Studio 2015, chọn **New Project**, Chọn ngôn ngữ **Visual C#**, Loại project là **Console Application**
- Đặt tên project là **Lab01** và lưu



Hình 1: Tạo một project console c# mới trong VS .NET 2015

- Click OK để đồng ý tạo project, kết quả chúng ta được một ứng dụng console như sau:



Hình 2: Màn hình làm việc của Project

- Lưu lại dự Án File/Save All



**Bước 2:** Tạo lớp class Student, Khai báo thuộc tính, phương thức (Right Click vào Lab01 Solution, chọn Add/ Class đặt tên là Student.cs)

```
class Student
{
    //1. Tao thuoc tinh
    private string studentID; //mã số sinh viên
    private string fullName; // họ tên sinh viên
    private float averageScore; //điểm trung bình
    private string faculty; //khoa

    //2. Tao cac Property (tip: chọn Quick Actions And Refactorings từ thuộc tính)
    public string StudentID
    {
        get
        {
            return studentID;
        }
        set
        {
            studentID = value;
        }
    }

    public string FullName
    {
        get
        {
            return fullName;
        }

        set
        {
            fullName = value;
        }
    }

    public float AverageScore
    {
        get
        {
            return averageScore;
        }

        set
        {
            averageScore = value;
        }
    }
}
```

```
    }  
}  
public string Faculty  
{  
    get  
    {  
        return faculty;  
    }  
    set  
    {  
        faculty = value;  
    }  
}  
// 3. Tao constructor mặc định không tham số  
public Student()  
{  
}  
// 4. Tao constructor có tham số  
public Student(string id, string name, float score, string faculty)  
{  
    StudentID = id;  
    FullName = name;  
    AverageScore = score;  
    Faculty = faculty;  
}  
// 5. Viết các phương thức nhập, xuất sinh viên  
public void Input()  
{  
    Console.Write("Nhập MSSV:");  
    StudentID = Console.ReadLine();  
    Console.Write("Nhập Họ tên Sinh viên:");  
    FullName = Console.ReadLine();  
    Console.Write("Nhập Điểm TB:");  
    AverageScore = float.Parse(Console.ReadLine()); //ép sang kiểu float  
  
    Console.Write("Nhập Khoa:");  
    Faculty = Console.ReadLine();  
}  
public void Show()  
{
```

```

        Console.WriteLine("MSSV:{0} Họ Tên:{1} Khoa:{2} ĐiểmTB:{3}", this.StudentID,
            this.fullName, this.Faculty, this.AverageScore);
    }

```

**Bước 3:** Viết code trong hàm Main() để cho phép người dùng nhập vào tổng số N sinh viên

```

static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.Unicode; //Để sử dụng tiếng việt
    Console.InputEncoding = Encoding.Unicode;
    //Nhập tổng số sinh viên N, Convert kiểu dữ liệu sang biến N kiểu int
    Console.Write("Nhập tổng số sinh viên N =");
    int N = Convert.ToInt32(Console.ReadLine());
    Student[] arrStudents = new Student[N];
    Console.WriteLine("\n ====Nhập Danh Sách sinh viên====");
    for (int i = 0; i < N; i++)
    {
        Console.WriteLine("\n - Nhập sinh viên thứ {0}", i+1);
        arrStudents[i] = new Student();
        arrStudents[i].Input();
    }
    Console.WriteLine("\n ====Xuất Danh Sách sinh viên====");
    foreach (Student sv in arrStudents)
    {
        sv.Show();
    }
    Console.ReadKey(); //Dừng màn hình kiểm tra kết quả
}

```

**Bước 4:** Biên dịch và chạy chương trình

- ✓ Để biên dịch chương trình chọn menu **Build**, rồi chọn **Build Solution** (hoặc dùng phím tắt **F6** hoặc click phải vào Solution Explorer chọn **Build**). VS.NET sẽ thông báo biên dịch thành công hay gặp lỗi cú pháp.
- ✓ Để chạy chương trình click vào biểu tượng **Start** trên VS (hoặc chọn **Debug / Debug Without Debugging** hoặc dùng **Ctrl + F5**)

```
Nhập tổng số sinh viên N =2

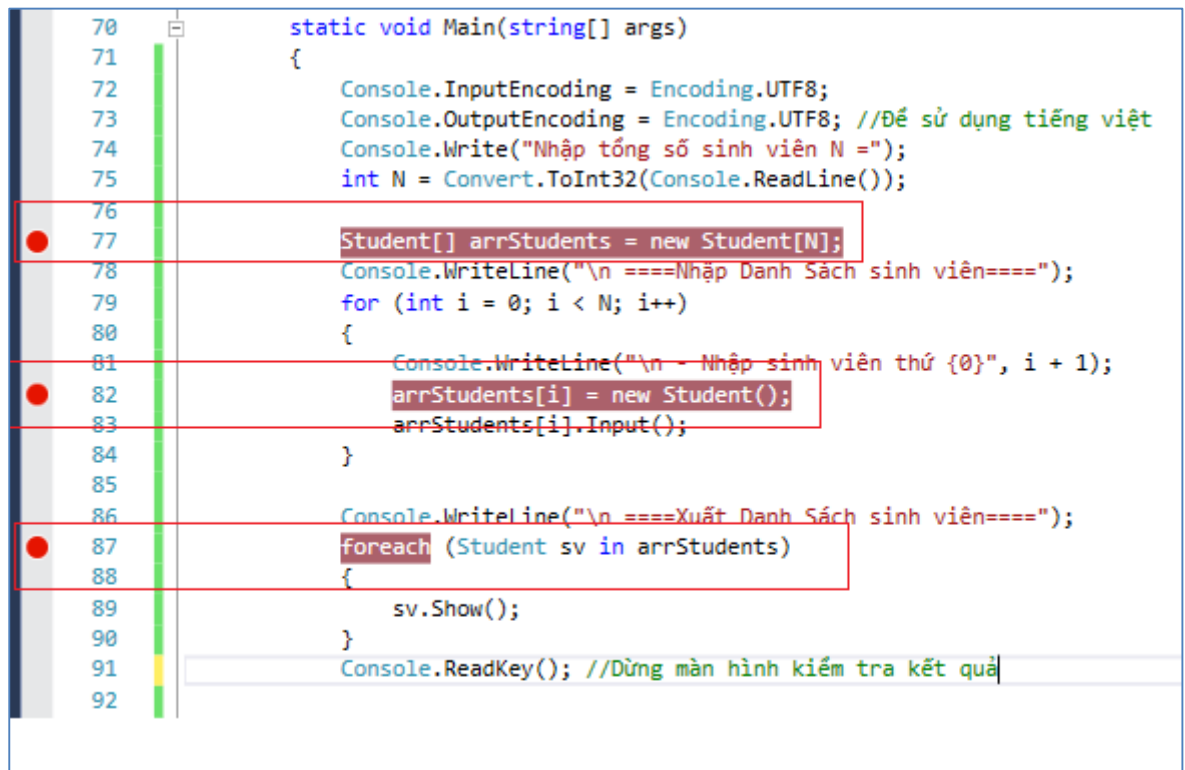
====Nhập Danh Sách sinh viên====

- Nhập sinh viên thứ 1
Nhập MSSV:1811030406
Nhập Họ tên Sinh viên:Nguyễn Văn A
Nhập Điểm TB:7.8
Nhập Khoa:CNTT

- Nhập sinh viên thứ 2
Nhập MSSV:1822363676
Nhập Họ tên Sinh viên:Nguyễn Văn B
Nhập Điểm TB:8.5
Nhập Khoa:QTKD

====Xuất Danh Sách sinh viên====
MSSV:1811030406 Họ Tên:Nguyễn Văn A Khoa:CNTT ĐiểmTB:7.8
MSSV:1822363676 Họ Tên:Nguyễn Văn B Khoa:QTKD ĐiểmTB:8.5
```

- ✓ Để Debug chương trình sử dụng F5(hoặc vào Debug / Start Debugging) kết hợp với việc đặt các breakpoint
  - Tạo breakpoint bằng cách đơn giản nhất là click chuột vào đầu dòng code (như trong hình). Để huỷ breakpoint, chỉ cần click chuột vào breakpoint đó một lần nữa. Ngoài ra các bạn cũng có thể tạo/huỷ breakpoint bằng phím F9.



- Nhấn F5 để bắt đầu Debug, Tại các vị trí BreakPoint chương trình sẽ dừng

lại. Muốn kiểm tra được sự thay đổi giá trị của các biến.

- Sử dụng Step Over / Step Into / Step Out trong quá trình Debug chương trình

**Step Over (F10):** Chạy step by step, lướt qua hàm (chỉ nhận giá trị return của hàm).

**Step Into (F11):** Chạy step by step, đi vào nội dung của các hàm con.

**Step Out (continue):** “Nhảy” đến breakpoint kế tiếp.

Nếu không còn breakpoint nào thì sẽ kết thúc debug. Ngoài ra nó còn có chức năng chạy lướt qua hàm con hiện tại.

### Bài tập 1B: Thực hiện tiếp ở bài tập 1A

- ✓ Sử dụng Collection là **List** để chứa danh sách sinh viên thay thế cho mảng sinh viên trong bài tập 1. Chạy lại chương trình theo yêu cầu trên.
- ✓ Viết tiếp chương trình thực hiện một số yêu cầu sau bằng cách sử dụng LINQ. *(Khuyến khích sinh viên viết ra các hàm cho dễ quản lý, những phần chung nên gom vào một hàm)*

1.1 Xuất ra thông tin của các SV đều thuộc khoa “CNTT” (nếu có)

1.2 Xuất ra thông tin sinh viên có điểm TB lớn hơn bằng 5 (nếu có).

1.3 Xuất ra danh sách sinh viên được sắp xếp theo điểm trung bình tăng dần

1.4 Xuất ra danh sách sinh viên có điểm TB lớn hơn bằng 5 và thuộc khoa “CNTT” (nếu có)

1.5 Xuất ra danh sách sinh viên có điểm trung bình cao nhất và thuộc khoa “CNTT” (nếu có)

#### **Hướng Dẫn**

1. Sử dụng **List** để thay thế mảng student ở bài tập 1A ( sử dụng phương thức add để đưa đối tượng vào list)

```

static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.Unicode;
    Console.InputEncoding = Encoding.Unicode; //Để sử dụng tiếng việt
    List<Student> listStudents = new List<Student>();
    Console.Write("Nhập tổng số sinh viên N =");
    int N = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("\n ====Nhập Danh Sách sinh viên====");
    for (int i = 0; i < N; i++)
    {
        Console.WriteLine("\n - Nhập sinh viên thứ {0}", i + 1);
        Student temp = new Student();
        temp.Input();
        listStudents.Add(temp);
    }

    Console.WriteLine("\n ====Xuất Danh Sách sinh viên====");
    foreach (Student sv in listStudents)
    {
        sv.Show();
    }
    Console.ReadKey(); //Dừng màn hình kiểm tra kết quả
}

```

2. Sử dụng Cú pháp truy vấn (query syntax) hoặc cú pháp phương thức (method syntax) trong LINQ. Danh sách truy vấn trong LINQ

Loại	Phương thức sử dụng
Lọc dữ liệu	Where
Chọn dữ liệu	Select, SelectMany
Phân vùng dữ liệu	Take, Skip, TakeWhile, SkipWhile
Kết hợp	Join, GroupJoin
Sắp Xếp	OrderBy / ThenBy, Reverse
Gom nhóm	GroupBy
Toán tử tập hợp	Distinct, Union, Intersect, Except
Chuyển đổi kiểu dữ liệu	ToSequence, ToArray, ToList, ToDictionary, ToLookup, OfType, Cast
Phần tử	First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault, ElementAt, ElementAtOrDefault, DefaultIfEmpty
Các hàm tính toán kết hợp	Count, LongCount, Sum, Min, Max, Average, Aggregate
Toán tử phát sinh	Range, Repeat, Empty
Toán tử lượng hóa	Any, All, Contains
Trộn	Concat

- Các yêu cầu 1.1 đến 1.5 đều xuất ra danh sách, nên chúng ta nên tách hàm nhập - xuất sinh viên riêng để tái sử dụng

```

private static List<Student> NhapDSSinhVien()
{
    List<Student> listStudents = new List<Student>();
    Console.Write("Nhập tổng số sinh viên N =");
    int N = Convert.ToInt32(Console.ReadLine());

    Console.WriteLine("\n ===Nhap Danh Sách sinh viên===");
    for (int i = 0; i < N; i++)
    {
        Console.WriteLine("\n - Nhập sinh viên thứ {0}", i + 1);
        Student temp = new Student();
        temp.Input();
        listStudents.Add(temp);
    }
    return listStudents;
}

4 references
private static void XuatDSSinhVien(List<Student> listStudent)
{
    Console.WriteLine("\n ===Xuất Danh Sách sinh viên===");
    foreach (Student sv in listStudent)
    {
        sv.Show();
    }
}

static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.Unicode;
    Console.InputEncoding = Encoding.Unicode; //Để sử dụng tiếng việt

    //Nhap Sinh Viên - xuất danh sách sinh viên
    List<Student> listStudent = NhapDSSinhVien();
    XuatDSSinhVien(listStudent);
    Console.ReadKey(); //Dừng màn hình kiểm tra kết quả
}

```

- Thực hiện các câu query theo yêu cầu bằng LINQ ( chỉ cần chọn 1 cách các bạn quen thuộc để thực hiện, trong các HD tiếp theo tôi sử dụng cú pháp phương thức)

```

//1.1 Xuất ra thông tin của các SV đều thuộc khoa "CNTT" (nếu có)
//cách 1: Sử dụng cú pháp truy vấn
List<Student> listStudentCNTT = (from s in listStudent
                                where s.Faculty == "CNTT"
                                select s).ToList();

//Cách 2: Sử dụng cú pháp phương thức
List<Student> listStudentCNTT = listStudent.Where(p => p.Faculty == "CNTT").ToList();

//xuất thông tin
if (listStudentCNTT.Count() == 0)
    Console.WriteLine("Không có sinh viên thuộc khoa CNTT");
else
    XuatDSSinhVien(listStudentCNTT);

```

- Thực hiện 1.2 Sử dụng Where Lấy danh sách sinh viên có điểm  $\geq 5$

```
// 1.2 Xuất ra thông tin các sinh viên có điểm TB lớn hơn bằng 5(nếu có).
Console.WriteLine("1.2 Xuất ra thông tin các sinh viên có điểm TB lớn hơn bằng 5");
List<Student> listStudentResult = listStudent.Where(p => p.AverageScore >=5).ToList();
if (listStudentResult.Count() == 0)
    Console.WriteLine("Không có sinh viên có điểm >=5");
else
    XuatDSSinhVien(listStudentResult);
```

- Thực hiện 1.3 Sử dụng OrderBy để sắp tăng dần

```
// 1.3 Xuất ra danh sách các sinh viên được sắp xếp theo điểm trung bình tăng dần
Console.WriteLine("1.3 Sắp xếp sinh viên có điểm tăng dần");
List<Student> listStudentSort = listStudent.OrderBy(p => p.AverageScore).ToList();
XuatDSSinhVien(listStudentSort);
```

-Thực hiện 1.4 và 1.5 (SV tự thực hiện)

**Bài tập 2:** Tạo thêm 1 project ở trong cùng solution Lab01 có tên là "Lab01-02".

- ✓ Viết lại chương trình từ bài tập 1 trên theo cách tạo thêm một lớp là **Person** làm lớp cơ sở cho lớp **Student**. Chọn Mã số, Họ tên làm field để đưa lên lớp **Person**.
- ✓ Thêm thông tin lớp giảng viên **Teacher** kế thừa từ lớp **Person**. Mỗi giảng viên đều có Mã số, Họ Tên và Địa chỉ.
- ✓ Ở hàm **Main()** cho phép nhập vào tổng số N đối tượng được đưa vào **List** danh sách **Person**, ở mỗi lần nhập liệu user có quyền chọn là nhập thông tin cho **Student** hoặc **Teacher**. Xuất danh sách thông tin vừa nhập liệu trên.
- ✓ Thực hiện một số yêu cầu sau trên List danh sách nhập liệu
  - 2.1 Tìm kiếm danh sách các Sinh Viên thuộc khoa "**CNTT**" nếu có.
  - 2.2 Xuất ra danh sách sinh viên có điểm trung bình nhỏ hơn 5 và thuộc khoa "**CNTT**".
  - 2.3 Xuất ra danh sách giáo viên có địa chỉ chứa thông tin "Quận 9" nếu có
  - 2.4 Tìm kiếm giáo viên có mã giảng viên là CHN060286. Xuất ra thông tin giáo viên tìm được nếu có.
  - 2.5 Tìm danh sách sinh viên có điểm trung bình cao nhất và thuộc khoa "**CNTT**"



**Bài tập 3:** Thêm 1 project ở trong cùng solution Lab01 có tên là "**Lab01-03**".

Công ty địa ốc D cần xây dựng chương trình quản lý thông tin về các khu đất do công ty cung ứng với các thông tin cần quản lý

- Khu Đất: Địa Điểm, Giá Bán (Đơn vị tính: VND) và Diện Tích (m2)

Thực hiện các yêu cầu sau:

- ✓ Xây dựng các lớp với chức năng thừa kế
- ✓ Nhập xuất danh sách các Khu Đất của công ty.
- ✓ Xuất ra danh sách thông tin các khu đất có diện tích được sắp xếp tăng dần.
- ✓ Xuất ra danh sách các khu đất có giá bán  $< 1$  tỷ và diện tích  $\geq 60\text{m}^2$  (nếu có).
- ✓ Tính đơn giá trung bình  $1\text{m}^2$  của tất cả các khu đất có diện tích lớn hơn  $1000\text{m}^2$  (nếu có).

#### Bài tập 4:

Công ty địa ốc D muốn mở rộng kinh doanh thêm về thị trường Nhà phố và Chung Cư. Với tất cả các thông tin cần quản lý như sau:

- Khu Đất: Địa Điểm, Giá Bán (Đơn vị tính: VND) và Diện Tích (m2)
- Nhà Phố: Địa Điểm, Giá Bán (Đơn vị tính: VND), Diện tích (m2), Năm Xây dựng, Số tầng
- Chung Cư: Địa Điểm, Giá Bán (Đơn Vị Tính: VND), Diện Tích (m2), Tầng

Thực hiện các yêu cầu sau:

- ✓ Xây dựng các lớp với chức năng thừa kế
- ✓ Nhập xuất danh sách các thông tin (Khu đất, Nhà phố, Chung Cư) cần quản lý.
- ✓ Xuất tổng giá bán cho 3 loại (Khu đất, Nhà phố, Chung Cư) của công ty D.
- ✓ Xuất danh sách các khu đất có diện tích  $> 100\text{m}^2$  hoặc là nhà phố mà có diện tích  $> 60\text{m}^2$  và năm xây dựng  $\geq 2020$  (nếu có).
- ✓ Nhập vào các thông tin cần tìm kiếm (địa điểm, giá, diện tích). Xuất thông tin danh sách tất cả các nhà phố hoặc chung cư phù hợp yêu cầu. (có địa điểm chứa chuỗi tìm kiếm không phân biệt hoa thường, có giá  $\leq$  giá tìm kiếm, và diện tích  $\geq$  diện tích cần tìm kiếm)

# **BÀI 2: LẬP TRÌNH WINDOWS FORM VỚI CONTROLS CƠ BẢN**

## **2.1 MỤC TIÊU**

---

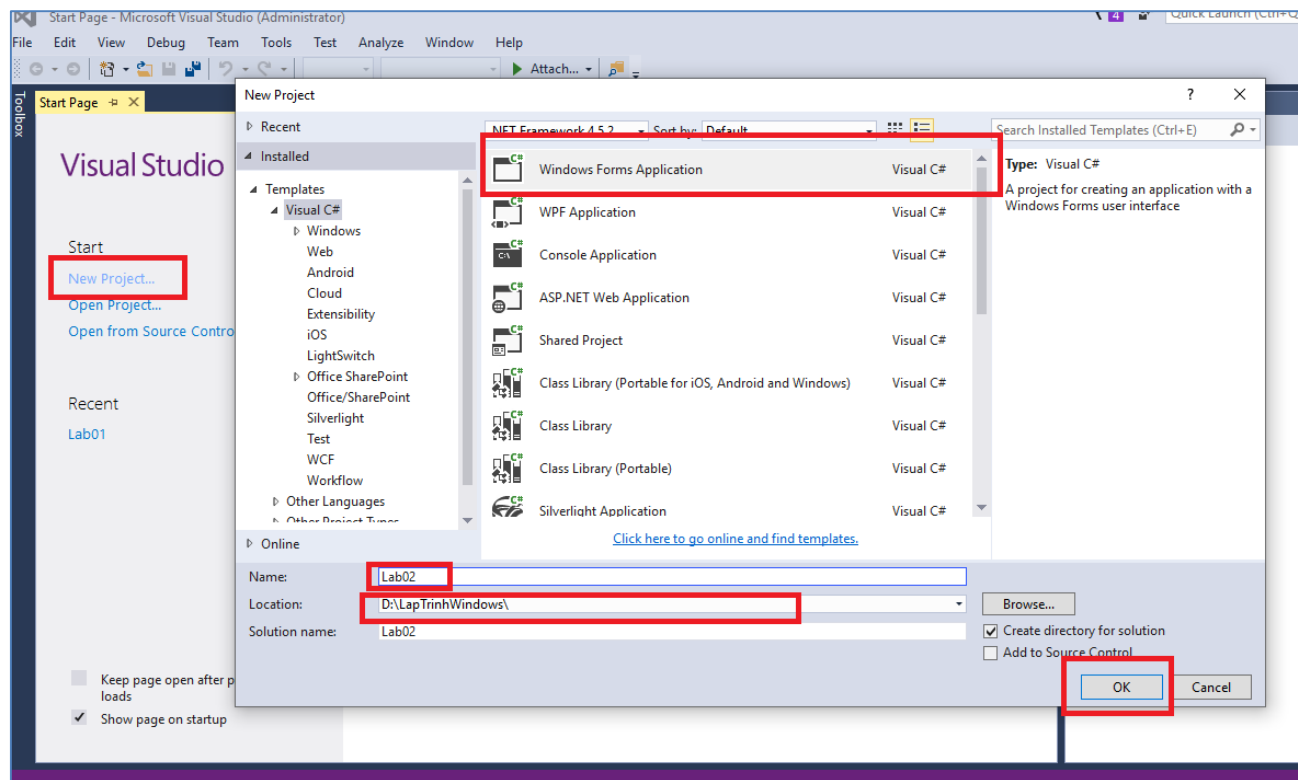
- Sử dụng Visual Studio .NET 2015/ 2017/ 2019 tạo ứng dụng dạng Windows Forms.
- Làm quen với việc sử dụng các control cơ bản trên form như:
  - o Label: Hiển thị các thông tin chỉ dẫn
  - o TextBox: Hộp nhập liệu thông tin
  - o Button: Cho phép user click chọn để thực hiện chức năng
  - o CheckBox: Cho phép user chọn một hoặc nhiều option
  - o Radio button: Cho phép user chọn duy nhất một option
  - o MessageBox: Hiển thị thông tin đến user
  - o DataGridView: Hiển thị danh sách thông tin trên bảng
  - o ListView: Hiển thị một danh sách các item với các biểu tượng
  - o ComboBox: Hộp chọn 1 giá trị trong danh sách giá trị
  - o ListBox: Danh sách các mục chọn, cho phép chọn 1 hoặc nhiều mục
  - o GroupBox: Nhóm các đối tượng về cùng nhóm
  - o Panel: Nhóm các đối tượng vào cùng 1 khung
- Tìm hiểu các thuộc tính trên control (Visible, Enable, Name, Text ...) và các phương thức là Event (Click, Text\_Change, Text\_Press...).
- Binding dữ liệu vào controls

## 2.2 HƯỚNG DẪN LÀM QUEN VỚI WINDOWS FORM

✓ Tạo Project Application, Giao diện màn hình thiết kế form

- Từ màn hình khởi động Microsoft Studio chọn Menu File - New – Project
- Language : Visual C#
- Loại ứng dụng: Windows Forms Application
- Name: Tên Project

Location: Đường dẫn lưu Project

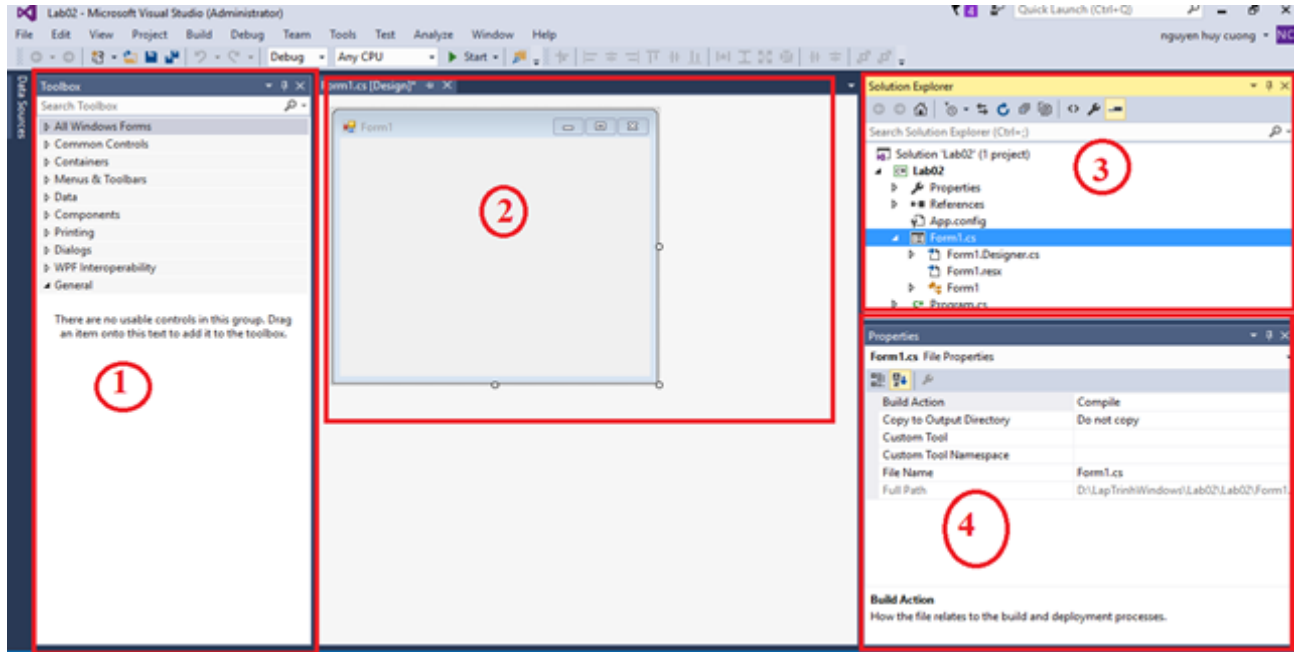


Hình 1: Tạo mới project Windows Form Application

Kết quả màn hình VS.NET cho ứng dụng Windows Form bao gồm các phần cơ bản

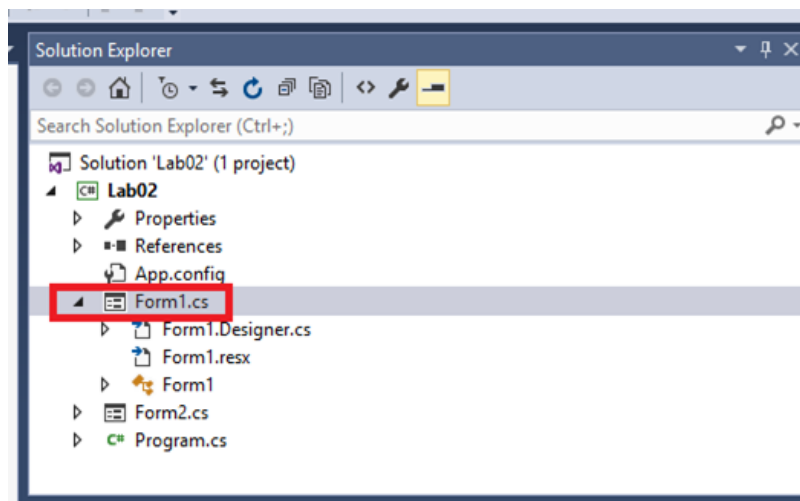
- (1): Toolbox: Chứa các control cho phép kéo thả vào Form
- (2): Màn hình thiết kế Form, có thể chuyển sang phần code editor...
- (3): Cửa sổ Solution Explorer: Cho phép người lập trình có thể quản lý các thành phần trong project, hỗ trợ định vị nhanh chóng đến các file mã nguồn.

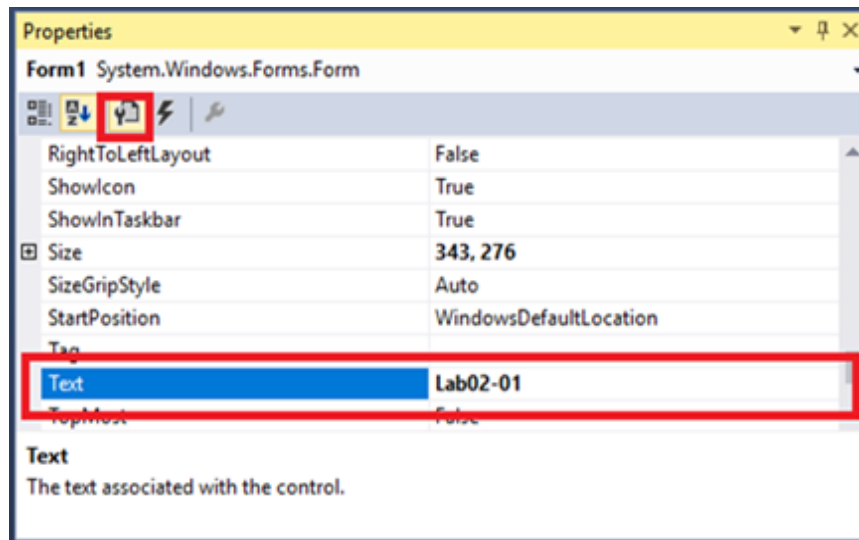
- (4): Cửa sổ property: cho phép user có thể custom lại các thành phần control trên form như: thiết lập các thuộc tính cho control, form, component, cho phép khai báo trình xử lý sự kiện của các control trên form...



Hình 2: Màn hình VS. NET phục vụ cho việc tạo project Windows Form

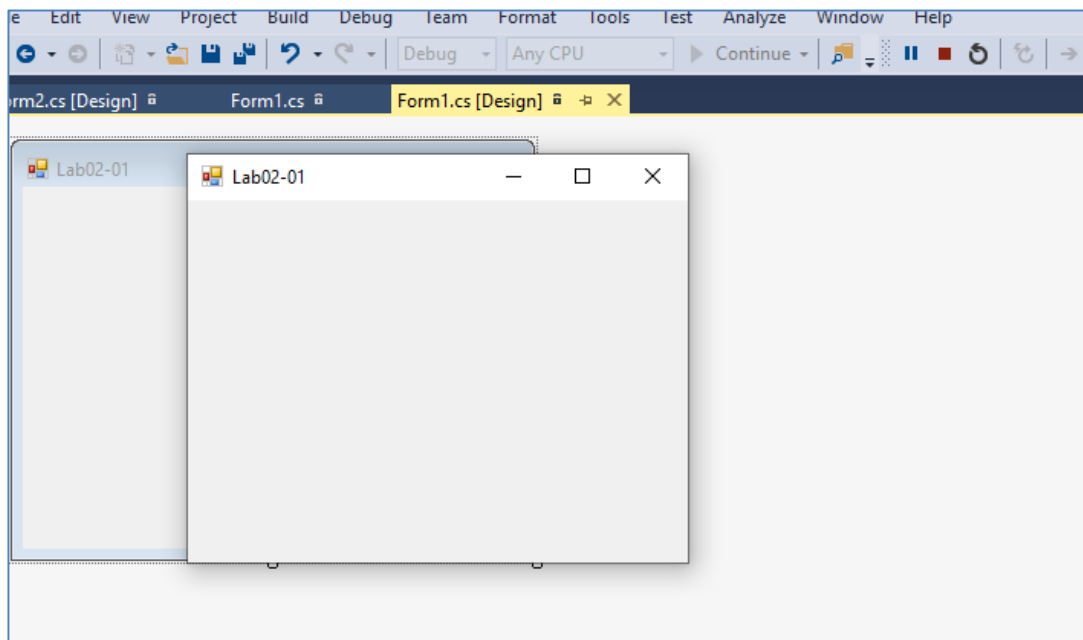
- Chú ý cửa sổ Toolbox chứa các công cụ để thiết kế: Nếu không thấy cửa sổ này, ta chọn menu View / **Toolbox**.
- Đổi tên form: Click lên Form1 ở cửa sổ Design, quan sát trên cửa sổ Properties, ta thấy có thuộc tính Text, giá trị mặc định là Form1, ta đổi thành Lab02-01





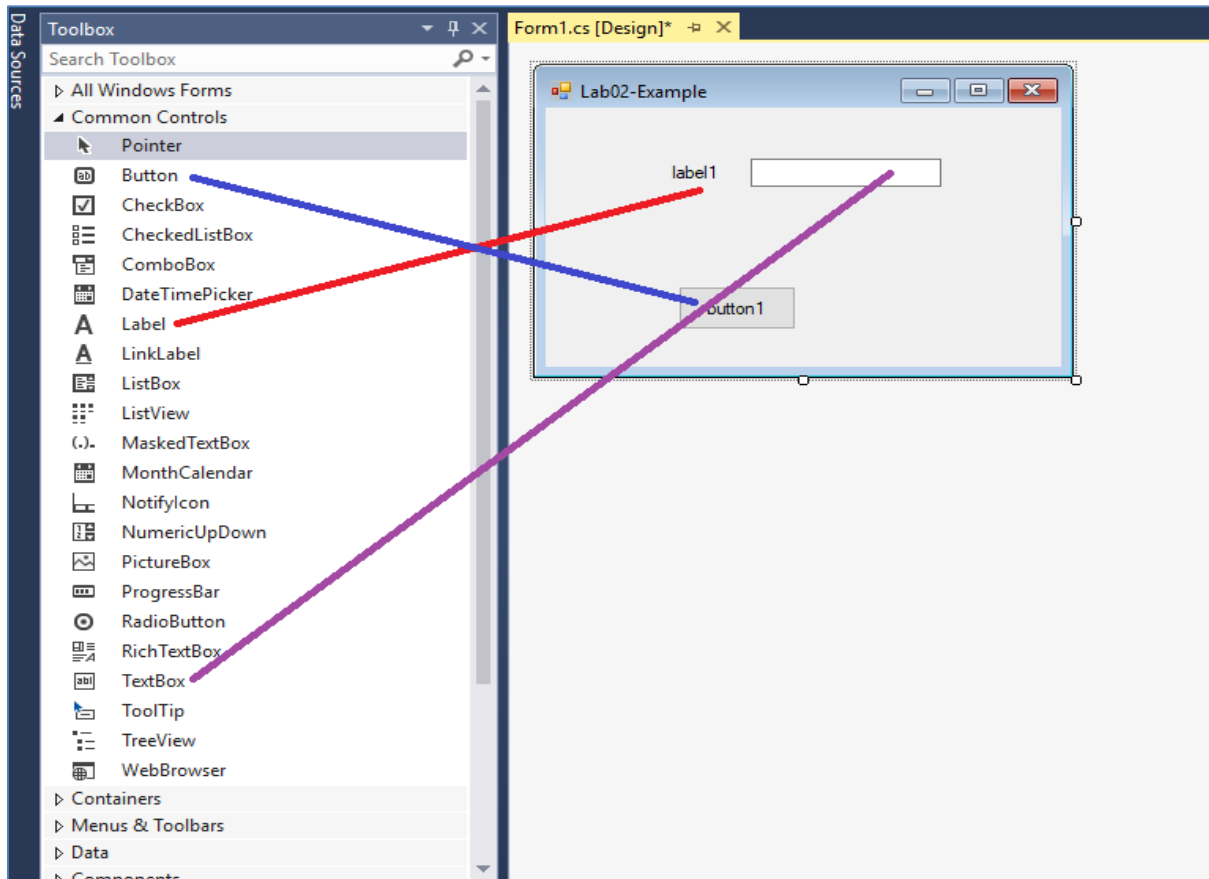
Hình 3: Property của đối tượng Form

- Chạy thử chương trình F5

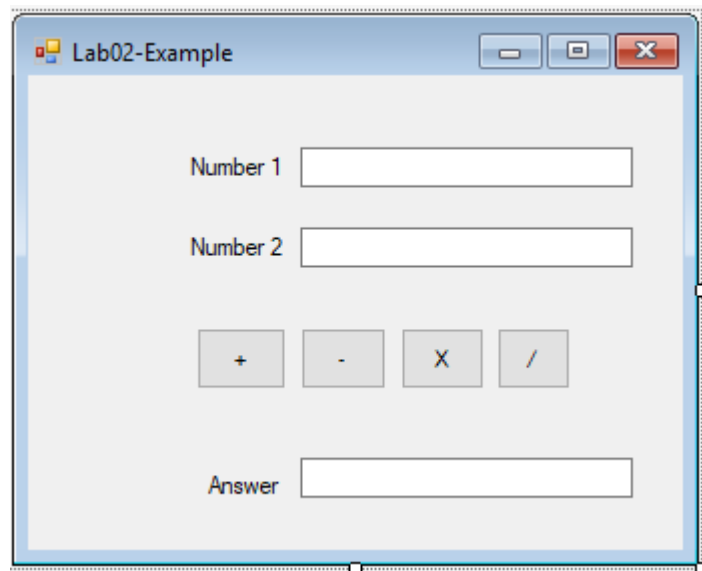


Hình 4: Chạy giao diện Forms đầu tiên

- Kéo thả các control từ toolbox vào Forms



- Thiết kế lại Form như hình dưới đây

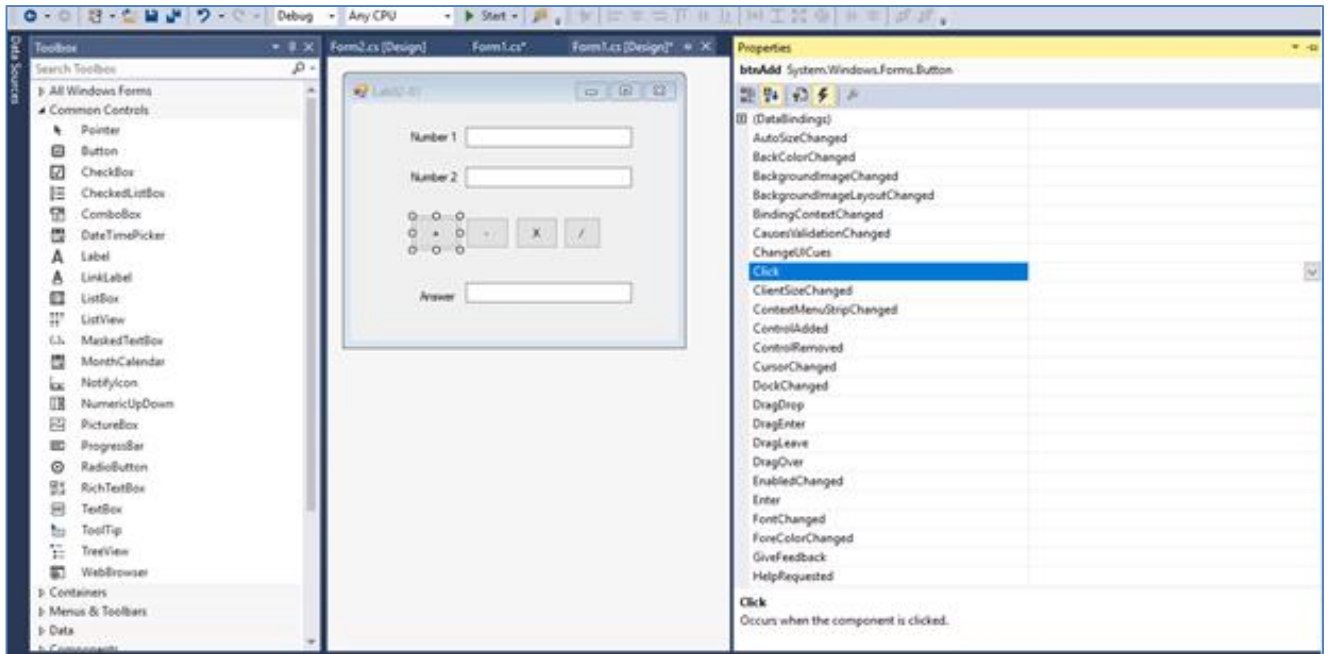


Hình 6: Giao diện sau khi thiết kế, chỉnh sửa Property

Đặt tên các **Controls** (thuộc tính **Name** trong **Property**) để dễ dàng quản lý. Đặt tên như sau

- Các **Label** ở trong form có tên: lblNumber1, lblNumber2, lblAnswer

- Các **TextBox** ở trong form có tên: txtNumber1, txtNumber2, txtAnswer
- Các **Button** ở trong form có tên: btnAdd, btnSub, btnMul, btnDiv
- Xử lý sự kiện trên Control: Vào **Property** chọn biểu tượng **Events**



Hình 7: Các Events trong buttonAdd

- ✓ Chọn tên Event cần xử lý và double click vào đó.

Đối với các button để chọn nhanh sự kiện click ta có thể double click trực tiếp vào button đó.

Double click vào Button **btnAdd** để tiến hành viết code cho sự kiện Mouse click.

VS .NET tự động sinh ra phương thức **btnAdd\_Click**. Sau đó tiến hành viết code tính tổng 2 số vừa nhập liệu

```
private void btnAdd_Click(object sender, EventArgs e)
{
    // Viết xử lý tính tổng trong sự kiện Add_click
    float number1 = float.Parse(txtNumber1.Text);
    float number2 = float.Parse(txtNumber2.Text);
    float result = number1 + number2;
    txtAnswer.Text = result.ToString();
}
```

## 2.3 BÀI TẬP

### Bài tập 1:

- ✓ Viết các sự kiện hoàn thành chương trình ở ví dụ trên ( các phép toán +,-,\*,/)
- ✓ Sử dụng **MessageBox** để đưa thông tin thêm cho người dùng khi gặp tất cả những lỗi ngoài mong muốn

**Hướng Dẫn:** Sử dụng cơ chế quản lý lỗi trong ngôn ngữ C#: try...catch...finally

```
private void btnAdd_Click(object sender, EventArgs e)
{
    try
    {
        //Xử lí sự kiện cộng 2 số
    }
    catch(Exception ex) //Khi gặp bất kì lỗi nào sẽ vào catch
    {
        MessageBox.Show(ex.Message);
    }
}
```

**Bài tập 2:** Tạo thêm Project “**Lab02-02**” trong cùng Solution Lab02 (Sử dụng CheckBox, ComboBox, DataGridView)

Quản lý thông tin sinh viên cần lưu trữ các thông tin sau: Mã số sinh viên, Họ Tên, Giới Tính, Điểm Trung Bình và Tên Khoa. Thiết kế chương trình quản lý thông tin sinh viên tương tự như sau:



Có 3 khoa được đưa vào comboBox (QTKD, CNTT, NNA).

### 2.1 Khi mới Load Form

- Khoa được chọn mặc định là QTKD. Giới tính Nữ mặc định được checked. Tổng số sinh viên Nam/Nữ đều là 0.

### 2.2 Khi nhấn vào nút "Thêm/Sửa"

- Kiểm tra các thông tin bắt buộc phải nhập liệu cho sinh viên như mã sinh viên, tên, và điểm trung bình. Nếu để trống sẽ xuất hiện thông báo lỗi "**Vui lòng nhập đầy đủ thông tin!**".
- Nếu mã số sinh viên vừa nhập chưa có ở trong DataGridView (bên phải) thì Thêm mới dữ liệu sinh viên vừa nhập vào DataGridView, và thông báo "**Thêm mới dữ liệu thành công!**"

Nếu đã có MSSV trong DataGridView thì Cập nhật dữ liệu sinh viên vào DataGridView, và thông báo "**Cập nhật dữ liệu thành công!**"

### 2.3 Khi nhấn vào nút "Xóa"

- Kiểm tra nếu MSSV cần xóa không tồn tại trong DataGridView thì thông báo lỗi "**Không tìm thấy MSSV cần xóa!**".
- Ngược lại thì xuất hiện cảnh báo YES/NO. Nhấn YES sẽ thực hiện xóa dòng dữ liệu sinh viên trong DataGridView và thông báo "**Xóa sinh viên thành công!**".

2.4 Viết code cho sự kiện ở DataGridView, người dùng chọn 1 dòng thì thể hiện ngược lại thông tin của các sinh viên đã chọn ở phần nhập liệu (bên trái).

2.5 Tính toán lại số sinh viên Nam, Nữ phù hợp với các ngữ cảnh khi thay đổi dữ liệu

### **Hướng Dẫn**

- Thiết kế giao diện tương tự như trên. Nên đặt các tên các Control để nhớ như: txtStudentID, txtFullName, optMale, optFemale, txtAverageScore, cmbFaculty, dgvStudent, btnUpdate, btnDelete...
- Ở ComboBox Khoa, để sẵn giá trị vào bằng cách vào Properties / Items rồi nhập các dòng (QTKD, CNTT, NNA)
- Thiết kế DataGridView có 5 cột (Columns) như yêu cầu. Đặt các tên column để nhớ. Điều chỉnh lại kích thước width cho phù hợp. Properties có SelectionMode = FullRowSelect
- Ở các sự kiện event, nên sử dụng try...catch để bắt lỗi.

- Trong sự kiện Form\_Load (double click vào Form để sinh ra). Để chọn mặc định khoa ban đầu

```
private void frmQuanLySinhVien_Load(object sender, EventArgs e)
{
    cmbKhoa.SelectedIndex = 0;
}
```

- Viết sự kiện Thêm/ Sửa. Double click vào button để tạo event btnUpdate\_Click

```
private int GetSelectedRow(string studentID)
{
    for (int i = 0; i < dgvStudent.Rows.Count; i++)
    {
        if( dgvStudent.Rows[i].Cells[0].Value.ToString() == studentID)
        {
            return i;
        }
    }
    return -1;
}

private void InsertUpdate(int selectedRow)
{
    dgvStudent.Rows[selectedRow].Cells[0].Value = txtStudentID.Text;
    dgvStudent.Rows[selectedRow].Cells[1].Value = txtFullName.Text;
    dgvStudent.Rows[selectedRow].Cells[2].Value = optFemale.Checked ? "Nữ" : "Nam";
    dgvStudent.Rows[selectedRow].Cells[3].Value =
float.Parse(txtAverageScore.Text).ToString();
    dgvStudent.Rows[selectedRow].Cells[4].Value = cmbFaculty.Text;
}
```

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    try
    {
        if (txtStudentID.Text == "" || txtFullName.Text == "" || txtAverageScore.Text == "")
            throw new Exception("Vui lòng nhập đầy đủ thông tin sinh viên!");

        int selectedRow = GetSelectedRow(txtStudentID.Text);
        if(selectedRow == -1)    //TH Thêm mới
        {
            selectedRow = dgvStudent.Rows.Add();
            InsertUpdate(selectedRow);
            MessageBox.Show("Thêm mới dữ liệu thành công!", "Thông Báo", MessageBoxButtons.OK);
        }
        else //TH cập nhật
        {
            InsertUpdate(selectedRow);
            MessageBox.Show("Cập nhật dữ liệu thành công!", "Thông Báo", MessageBoxButtons.OK);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

- Viết sự kiện Xóa

```

private void btnDelete_Click(object sender, EventArgs e)
{
    try
    {
        int selectedRow = GetSelectedRow(txtStudentID.Text);
        if (selectedRow == -1)
        {
            throw new Exception("Không tìm thấy MSSV cần xóa!");
        }
        else
        {
            DialogResult dr = MessageBox.Show("Bạn có muốn xóa ?", "YES/NO", MessageBoxButtons.YesNo);
            if (dr == DialogResult.Yes)
            {
                dgvStudent.Rows.RemoveAt(selectedRow);    //xóa thông tin sinh viên tại dòng tìm thấy
                MessageBox.Show("Xóa sinh viên thành công!", "Thông Báo", MessageBoxButtons.OK);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Sinh viên tự thực hiện yêu cầu còn lại 2.4 và 2.5

**Bài tập 3:** Tạo thêm Project “Lab02-03” trong cùng Solution Lab02

Tạo một project giúp cho rạp chiếu phim quản lý việc bán vé của mình



Rạp có 3 hàng ghế, mỗi hàng có 5 ghế, các ghế được đánh số từ 1 đến 15 và được phân thành 3 dãy như (hình trên):

Giá vé lô A 5000/vé

Giá vé lô B 6500/vé

Giá vé lô C 8000/vé

Trên Form trình bày một sơ đồ các chỗ ngồi để người sử dụng chọn vị trí muốn mua. Trên sơ đồ này cũng thể hiện những vị trí đã bán vé và những vị trí chưa bán vé bằng cách thể hiện màu khác nhau (ghế chưa bán vé màu trắng, ghế đã bán vé màu vàng).

Khi người sử dụng click chuột tại một vị trí trên sơ đồ thì:

- ✓ Nếu đây là vị trí chưa bán vé thì đổi màu của vị trí này sang màu xanh để cho biết đây là vị trí đang chọn.
- ✓ Nếu đây là vị trí đang chọn (có màu xanh) thì đổi màu của vị trí này trở về màu trắng
- ✓ Nếu đây là một vị trí đã bán vé (có màu vàng) thì xuất hiện một Message box thông báo cho người sử dụng biết vé ở vị trí này đã được bán. Sau khi đã chọn các vị trí người sử dụng có thể click chuột vào nút **CHỌN** hoặc **HỦY BỎ**

Nếu click vào nút **CHỌN** thì:

- Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu vàng (cho biết vị trí đã bán vé)
- Xuất lên một Label tổng số tiền phải trả cho số vé đã mua (phụ thuộc vào các

vị trí đã chọn)

Nếu click vào nút **HỦY BỎ** thì:

- Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu trắng trở lại
- Xuất lên Label giá trị 0

### **Hướng Dẫn:**

- Sử dụng GroupBox để gom nhóm (dễ dàng cho việc tìm kiếm, đổi màu, tính tiền...). Mặc định ban đầu các hàng ghế là trống BackColor= White;
- Sinh viên nên viết 1 sự kiện dùng chung cho tất cả các nút bấm (1 – 15)

```
private void btnChooseASeat(object sender, EventArgs e)
{
    Button btn = sender as Button;
    if (btn.BackColor == Color.White)
        btn.BackColor = Color.Blue;
    else if (btn.BackColor == Color.Blue)
        btn.BackColor = Color.White;
    else if (btn.BackColor == Color.Yellow)
        MessageBox.Show("Ghế đã được bán!!");
}
```

### **Bài tập 4:** Tạo thêm Project “**Lab02-04**” trong cùng Solution Lab02 (Sử dụng ListView)

Quản lý thông tin tài khoản cần lưu trữ các thông tin sau: số tài khoản, tên khách hàng, địa chỉ khách hàng và số tiền trong tài khoản. Thiết kế chương trình quản lý thông tin tài khoản cho một ngân hàng tương tự như sau:

Quản Lý Tài Khoản

# QUẢN LÝ THÔNG TIN TÀI KHOẢN

Số tài khoản

Tên khách hàng

Địa chỉ khách hàng

Số tiền trong tài khoản

Thêm / Cập Nhật Xóa Thoát

STT	Mã tài khoản	Tên khách hàng	Địa chỉ	Số tiền
-----	--------------	----------------	---------	---------

Tổng tiền:

#### 4.1 Khi nhấn vào nút "Thêm/Cập Nhật"

- Kiểm tra các thông tin bắt buộc phải nhập liệu cho số tài khoản, tên, địa chỉ và số tiền. Xuất hiện thông báo lỗi "Vui lòng nhập đầy đủ thông tin!".
- Nếu chưa có dữ liệu **số tài khoản** trong ListView thì Thêm mới dữ liệu nhập vào ListView, tính lại tổng tiền và thông báo "Thêm mới dữ liệu thành công!"

Nếu đã tồn tại số tài khoản trong ListView thì Cập nhật dữ liệu vào ListView và tính lại tổng tiền và thông báo "Cập nhật dữ liệu thành công!"

#### 4.2 Khi nhấn vào nút "Xóa"

- Kiểm tra nếu số tài khoản cần xóa tồn tại trong ListView, thì xuất hiện cảnh báo YES/NO

Nhấn YES sẽ thực hiện xóa dòng dữ liệu tài khoản trong ListView và thông báo "Xóa tài khoản thành công!".

- Nếu số tài khoản cần xóa không tồn tại thì thông báo lỗi "Không tìm thấy số tài khoản cần xóa!".

4.3 Viết code cho sự kiện ở ListView khi người dùng chọn 1 dòng thì thể hiện ngược lại ở phần nhập liệu đúng thông tin trên.

Quản Lý Tài Khoản

## QUẢN LÝ THÔNG TIN TÀI KHOẢN

Số tài khoản: 010292914

Tên khách hàng: Nguyễn Quốc Anh

Địa chỉ khách hàng: 219/21 Trần Thái Tông Q1

Số tiền trong tài khoản: 1000000

Thêm / Cập Nhật Xóa Thoát

STT	Mã tài khoản	Tên khách hàng	Địa chỉ	Số tiền
1	02130131	Nguyễn Thái Công	123 Trương Định P3 TP.HCM	2012012
2	010292914	Nguyễn Quốc Anh	219/21 Trần Thái Tông Q1	1000000

# BÀI 3: LẬP TRÌNH WINDOWS FORM VỚI GIAO DIỆN MDI

## 3.1 MỤC TIÊU

---

- Sử dụng Visual Studio .NET 2015/2017/2019 tạo ứng dụng dạng Windows Forms MDI ( Multiple Document Interface)
- Thực hành với các controls
  - ✓ *MenuStrip*
  - ✓ *ToolStrip*
  - ✓ *StatusStrip*
  - ✓ *Timer*
  - ✓ *Media*

## 3.2 BÀI TẬP

---

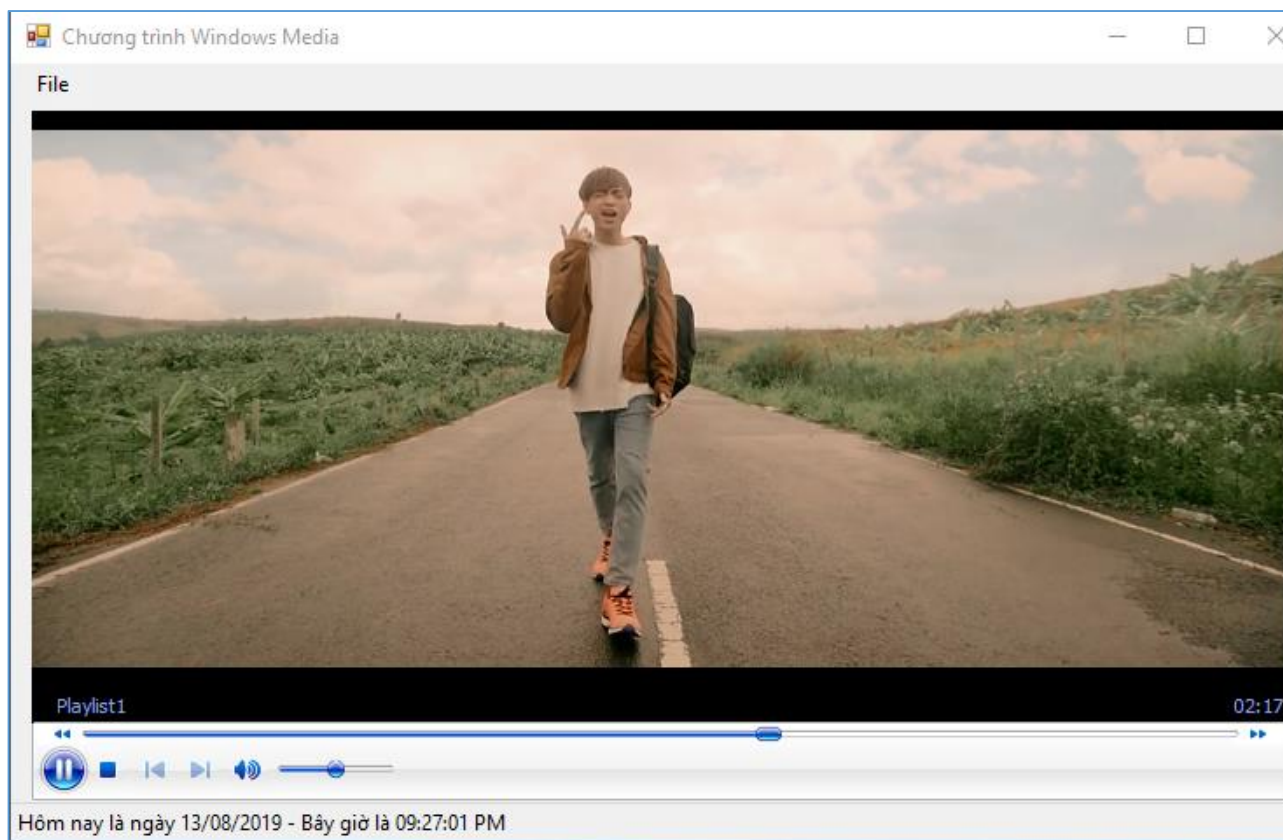
**Bài Tập 1:** Tạo Project **Lab04** mô phỏng điều khiển Windows media Player để phát các tập tin Audio và Video.

Yêu cầu:

- Ứng dụng chứa Windows Media Player control cho phép Play các file video/sound theo nhiều dạng format ( \*.avi \*.mpeg \*.wav \*.midi \*.mp4 \*.mp3 )
- Menu Bar có 1 Menu File và 2 Sub Menu Open và Exit
  - ✓ Khi chọn Sub Menu Exit sẽ thoát ứng dụng
  - ✓ Khi chọn Sub Menu Open sẽ mở hộp thoại Open File để chọn mở file Media - Control Multi Media sẽ phát file Media đã chọn.



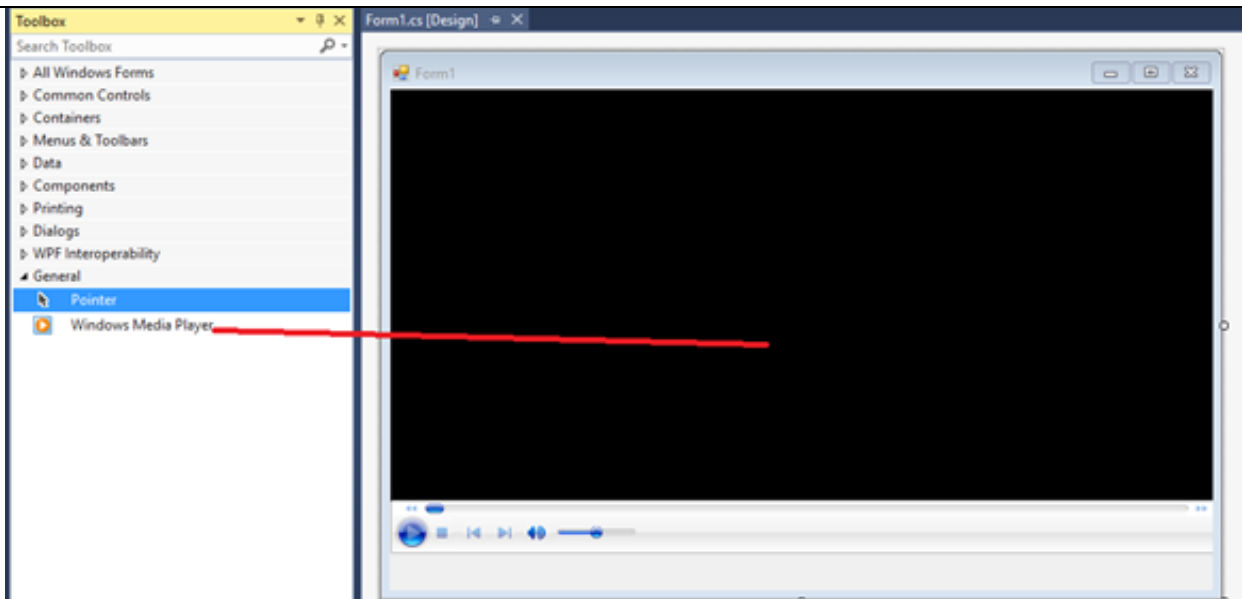
- StatusStrip hiển thị: Ngày giờ hệ thống và thay đổi giờ theo mỗi giây.



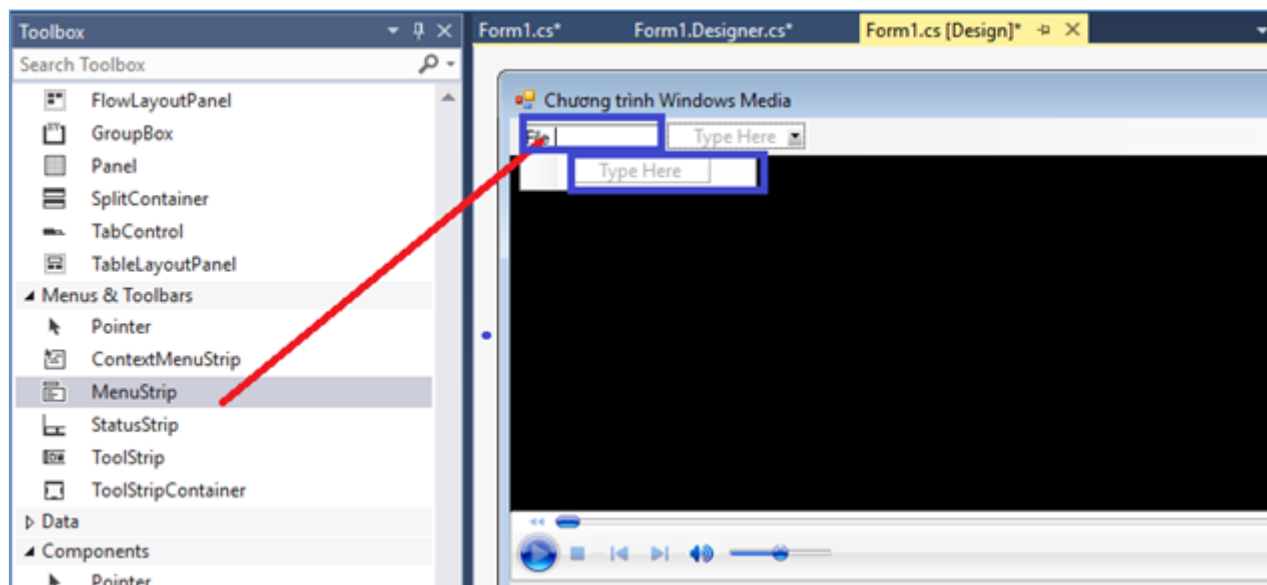
Hình 1: Thiết kế giao diện có windows media

### Hướng dẫn:

**Bước 1: Thiết kế giao diện:** Để sử dụng được điều khiển Windows Media Player click phải lên ToolBox và chọn Choose Items, trong cửa sổ Choose Toolbox Item, chọn thẻ COM Components và chọn thư viện Windows Media Player.

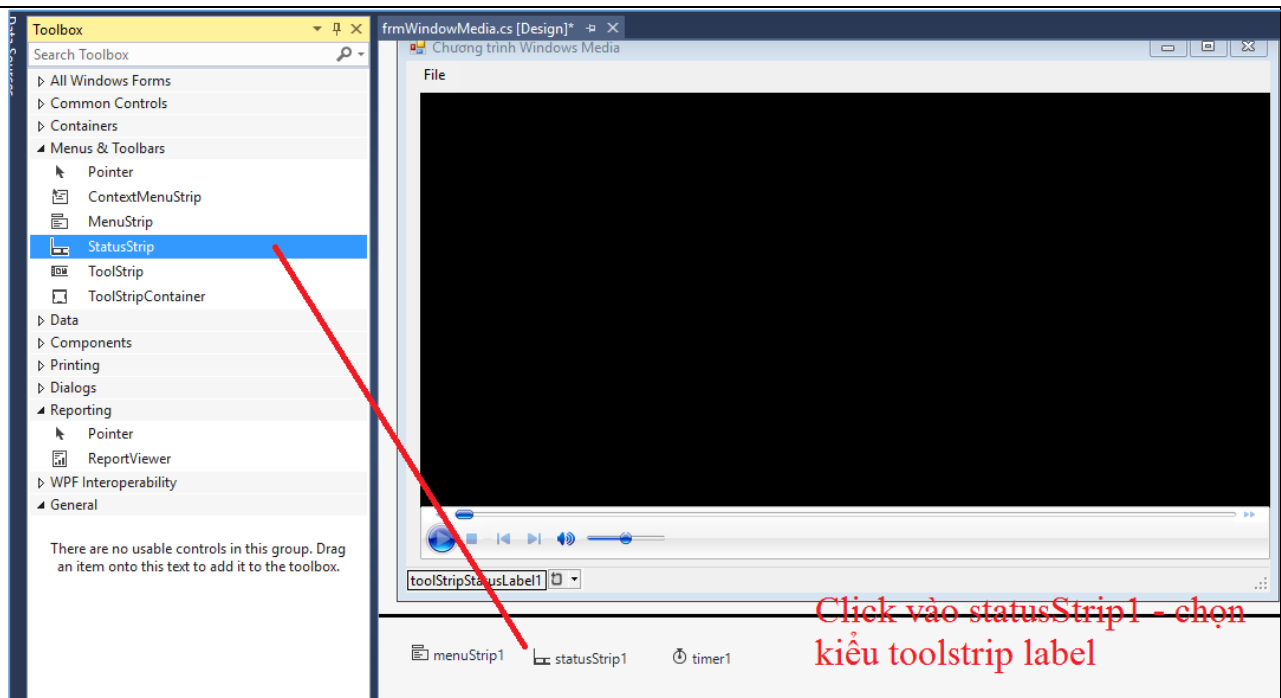


✓ Để tạo Menu **File**: Kéo thả MenuStrip trên toolbar ( Menu & ToolBars)

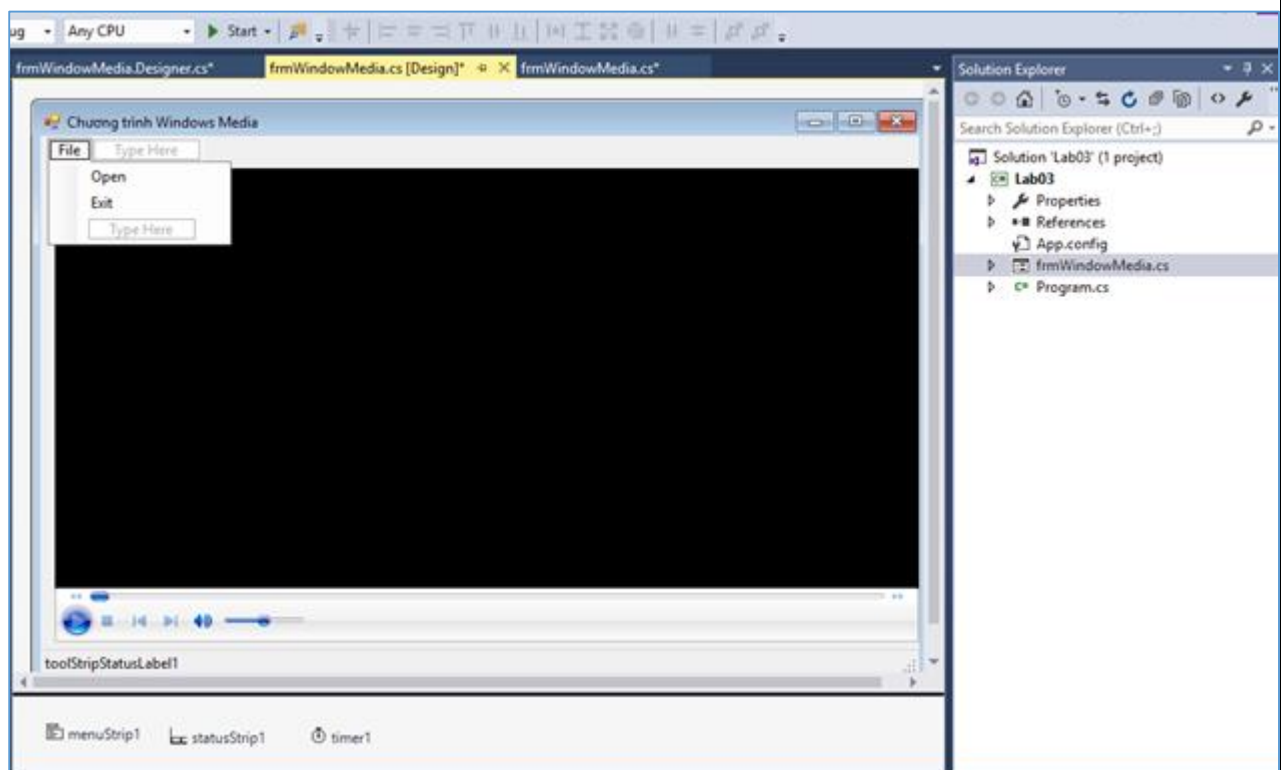


Hình 3: Kéo MenuStrip để tạo Menu

✓ Để tạo Status: Kéo thả StatusStrip



- ✓ Để thực hiện các sự kiện liên quan đến thời gian: Kéo **Timer** vào forms
- ✓ Giao diện sau thiết kế



**Bước 2: Viết các lệnh xử lý**✓ **Viết xử lý cho sự kiện Timer**

- Thiết lập thuộc tính của Timer **Interval:1000** (Đơn vị ms, cứ mỗi 1 giây sự kiện tick sẽ xảy ra)
- Chọn thuộc tính Enable = True để Timer được hoạt động
- Viết lệnh xử lý cho sự kiện **Timer1\_Tick()**

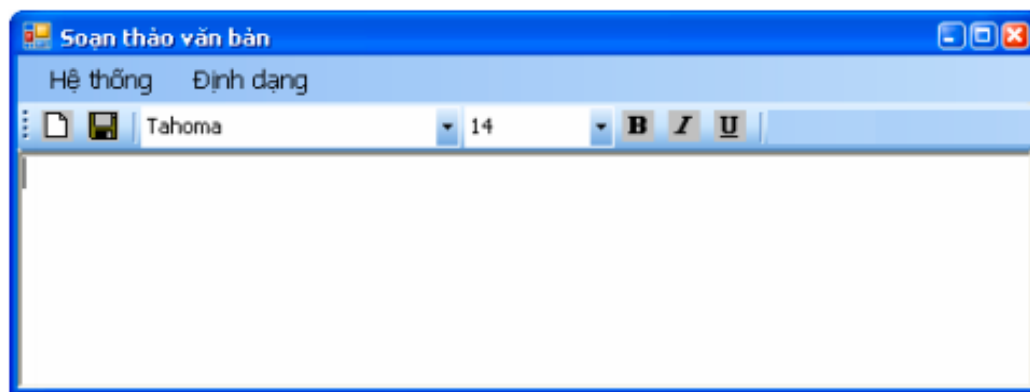
```
private void timer1_Tick(object sender, EventArgs e)
{
    this.toolStripStatusLabel1.Text = string.Format("Hôm nay là ngày {0} - Bây giờ là {1}",
    DateTime.Now.ToString("dd/MM/yyyy"), DateTime.Now.ToString("hh:mm:ss tt"));
}
```

✓ **Viết xử lý cho SubMenu Open**

```
// Xử lý menu Open Sử dụng OpenFileDialog
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Tạo hộp thoại mở file
    OpenFileDialog dlg = new OpenFileDialog();
    //lọc hiện thị các loại file
    dlg.Filter = "AVI file| *.avi | MPEG File | *.mpeg | Wav File | *.Wav | Midi File | *.midi | Mp4 File | *.mp4 | MP3 | *.mp3";
    //hien thi openFileDialog
    if (dlg.ShowDialog() == DialogResult.OK)
        axWindowsMediaPlayer1.URL = dlg.FileName; //Lấy tên file cần mở
}
```

**Bài tập 2: Soạn thảo văn bản**

Thêm project **Lab04-02** vào Solution Lab04 có giao diện như sau

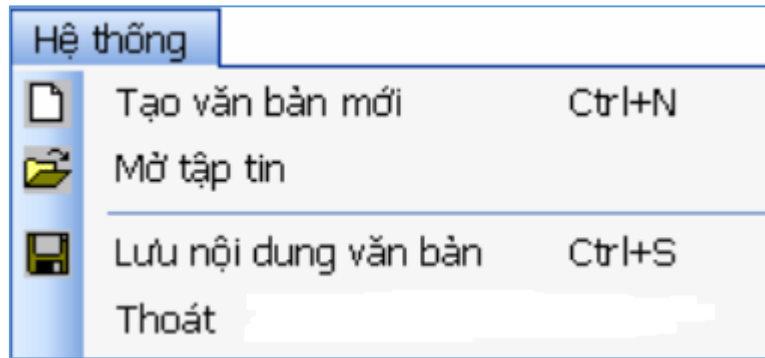


Sử dụng **RichTextBox** để thiết kế điều khiển hiển thị và nhập nội dung văn bản:

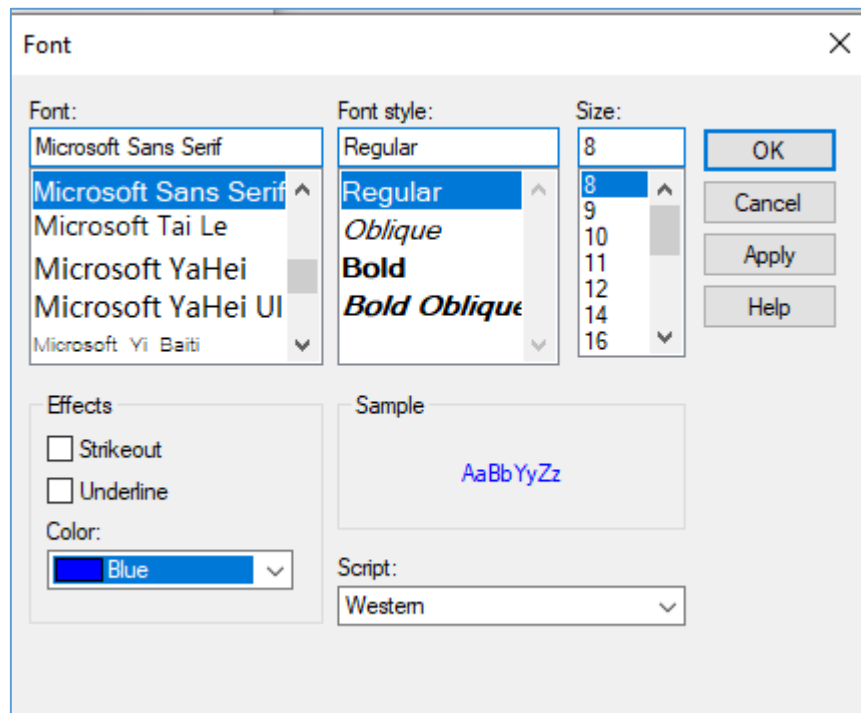
Sử dụng công cụ MenuStrip để tạo Hệ Thống và Định Dạng

Sử dụng công cụ **ToolStrip** để tạo thanh công cụ chứa các button và image như trên

✓ **Trong menu Hệ thống thiết kế như sau**



- ✓ Khi click vào menu định dạng gọi **FontDialog** có sẵn của windows



Hướng Dẫn cách gọi

```
FontDialog fontDlg = new FontDialog();
fontDlg.ShowColor = true;
fontDlg.ShowApply = true;
fontDlg.ShowEffects = true;
fontDlg.ShowHelp = true;
if (fontDlg.ShowDialog() != DialogResult.Cancel)
{
    richText.ForeColor= fontDlg.Color;
    richText.Font = fontDlg.Font;
}
```


- ✓ **Cài đặt xử lý cho các chức năng**


2.1 Khi mới mở Form, thực hiện:


- Tạo dữ liệu cho ComboBox Font: chứa tất cả các Font chữ của hệ thống.
- Tạo dữ liệu cho ComboBox Size: chứa các giá trị từ 8 → 72.

(8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72)


- Tạo giá trị mặc định ban đầu là Font **Tahoma**, Size **14**


2.2 Khi chọn Tạo văn bản mới (hoặc nhấn nút 


2.3 Khi chọn Mở tập tin (hoặc nhấn nút 

2.4 Khi chọn Lưu nội dung văn bản (hoặc nhấn nút 

- ✓ Nếu là văn bản mới và trước đó chưa lưu lần nào thì hiển thị hộp thoại lưu tập tin (SaveFileDialog) cho phép người dùng chọn thư mục cần lưu tập tin với kiểu tập tin cần lưu là \*.rtf.
- ✓ Nếu là văn bản đã được mở trước đó thì thông báo cho người dùng lưu văn bản thành công.

2.5 Nút : Khi chọn, tùy thuộc vào trạng thái của nút để xử lý nội dung của vùng văn bản đang được chọn có được in đậm hay không.

Nút : Khi chọn, tùy thuộc vào trạng thái của nút để xử lý nội dung của vùng văn bản đang được chọn có được in nghiêng hay không.

Nút : Khi chọn, tùy thuộc vào trạng thái của nút để xử lý nội dung của vùng văn bản đang được chọn có được gạch dưới hay không.

### Hướng Dẫn

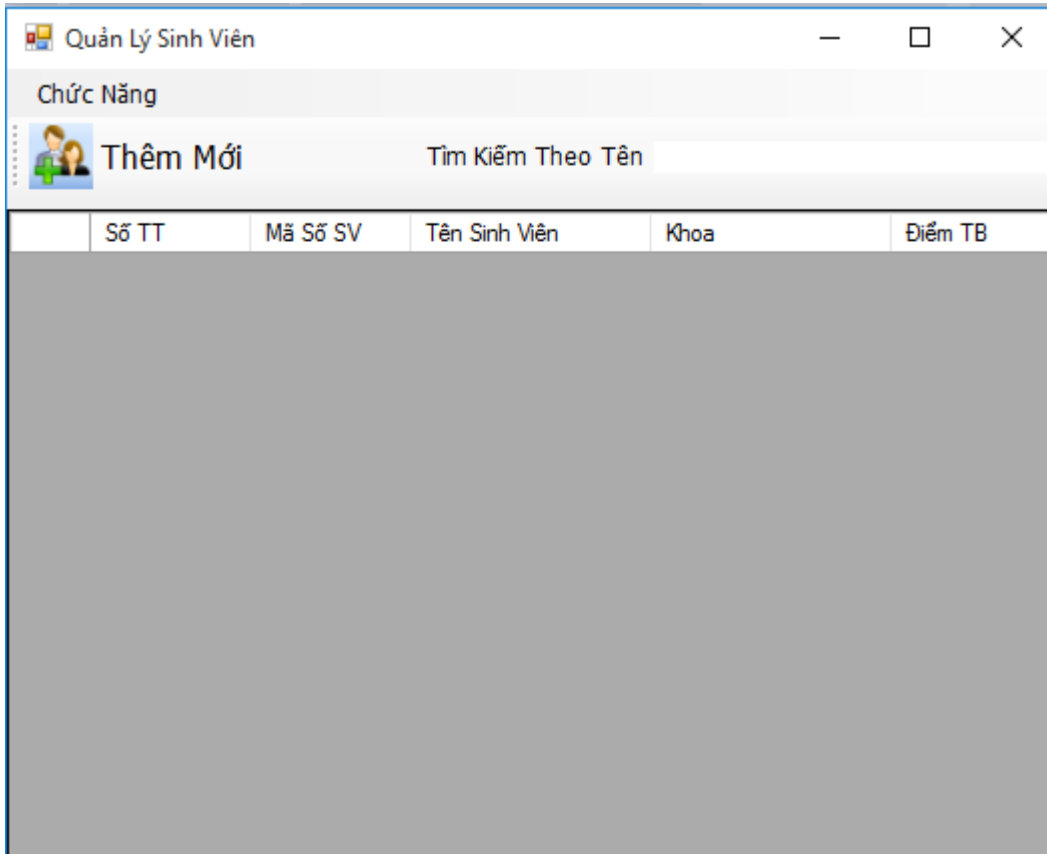
- Lấy tất cả các fonts từ hệ thống đưa vào combobox Fonts  
`foreach (FontFamily font in new InstalledFontCollection().Families)`  
`{`  
`cmbFonts.Items.Add(font.Name);`  
`}`
- Để load file sử dụng `richText.LoadFile(...);`
- Để Save file sử dụng `richText.SaveFile(...);`

**Bài tập 3: Viết Chương Trình Nhập Liệu Sinh Viên Đơn giản sau**

- Form Quản lý sinh viên có Menu Chức năng gồm 2 SubMenu là **Thêm mới** và **Thoát**

Sử dụng MenuStrip để tạo menu

Sử dụng ToolStrip để tạo hình và nút bấm Thêm mới, Tìm kiếm theo tên



Số TT	Mã Số SV	Tên Sinh Viên	Khoa	Điểm TB
-------	----------	---------------	------	---------

- Khi click vào menu thêm mới (Ctrl + N) hoặc ở icon hình thêm mới trên toolStrip thì gọi Form nhập liệu sinh viên như sau

Có 3 khoa được hiển thị trong Combobox (Style là DropDownList) gồm “Công nghệ thông tin”, “Ngôn ngữ Anh” và “Quản trị kinh doanh”

Khi click vào Nút Thêm Mới

- ✓ Thông báo cho người dùng các thông tin bắt buộc phải nhập liệu (Mã số, Tên Sinh Viên, Điểm)
- ✓ Thông báo cho người dùng thông tin mã số sinh viên nếu bị trùng trong DataGridView ở Form trước đó
- ✓ Thông báo cho người dùng dữ liệu nhập điểm trong phạm vi từ (0 -10)
- ✓ Trở về Form chính và đưa dữ liệu vào GridView hiển thị

Ở phần tìm kiếm theo tên sinh viên, khi Textbox tìm kiếm thay đổi thì luôn luôn tìm lại dữ liệu chứa tên tìm kiếm (không phân biệt hoa thường)

	Số TT	Mã Số SV	Tên Sinh Viên	Khoa	Điểm TB
▶	1	1	Nguyễn văn Bảo	Công nghệ thông tin	5.6
	2	BH030343	Phạm Chí Bình	Công nghệ thông tin	8.9



# BÀI 4: LẬP TRÌNH VỚI CƠ SỞ DỮ LIỆU SỬ DỤNG ENTITY

## 4.1 MỤC TIÊU


- Hướng dẫn sinh viên làm quen với việc xây dựng ứng dụng Windows Application có kết nối với CSDL SQL Server bằng Entity Framework của .NET.
- Sử dụng mô hình *Code First* trong EntityFramework với hướng tiếp cận đã có sẵn CSDL
- Thiết kế các Form nhập liệu cho các bảng trong cơ sở dữ liệu (hiện thị, thêm, xóa, sửa)


## 4.2 BÀI TẬP

Sử dụng SQL Server tạo cơ sở dữ liệu "QuanLySinhVien" đơn giản với 2 bảng: Sinh viên và Khoa như sau

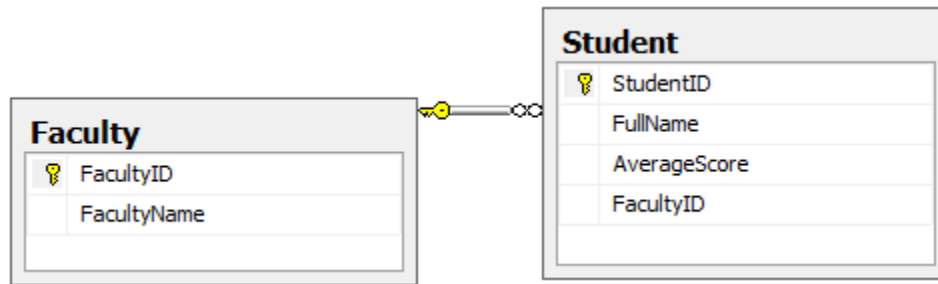
**Student** (**StudentID**, FullName, AverageScore, FacultyID)

**Faculty**(**FacultyID**, FacultyName)

DESKTOP-AA3CGL7...n - dbo.Student			
	Column Name	Data Type	Allow Nulls
	StudentID	nvarchar(20)	<input type="checkbox"/>
	FullName	nvarchar(200)	<input type="checkbox"/>
	AverageScore	float	<input type="checkbox"/>
	FacultyID	int	<input type="checkbox"/>

DESKTOP-AA3CGL7...en - dbo.Faculty			
	Column Name	Data Type	Allow Nulls
	FacultyID	int	<input type="checkbox"/>
	FacultyName	nvarchar(200)	<input type="checkbox"/>
			<input type="checkbox"/>

- ✓ Tạo mối quan hệ 2 bảng như sau:



- ✓ Nhập liệu sẵn vào cơ sở dữ liệu một số dòng

DESKTOP-AA3CGL7....en - dbo.Faculty		DESKTOP-AA3CGL7....n - dbo.Student			
FacultyID	FacultyName	StudentID	FullName	AverageScore	FacultyID
1	Công Nghệ Thông Tin	1611061916	Nguyễn Trần Hoàng Lan	4.5	1
2	Ngôn Ngữ Anh	1711060596	Đàm Minh Đức	2.5	1
3	Quản trị kinh doanh	1711061004	Nguyễn Quốc An	10	2

### Bài Tập 1 Sử dụng EntityFrameWork với mô hình Code First để kết nối CSDL

- ✓ Viết chương trình quản lý sinh viên có giao diện tương tự sau đây

**Quản lý thông tin Sinh viên**

**Thông Tin Sinh Viên**

Mã Số SV:

Họ Tên:

Khoa:

Điểm TB:

Mã Số SV	Họ Tên	Tên Khoa	Điểm TB
1611061916	Nguyễn Trần Hoàng Lan	Công Nghệ Thông Tin	4.5
1711060596	Đàm Minh Đức	Công Nghệ Thông Tin	2.5
1711061004	Nguyễn Quốc An	Ngôn Ngữ Anh	10

### Yêu Cầu Xử Lý

#### 1.1 Sự kiện Form\_load:

- Hiển thị danh sách sinh viên hiện có trong CSDL (Lấy từ bảng sinh viên)
- ComboBox Khoa lấy từ bảng **Faculty** và hiển thị tên khoa

#### 1.2 Khi nhấn vào nút "Thêm" Hoặc "Sửa"

- Kiểm tra các thông tin bắt buộc phải nhập liệu cho sinh viên như mã sinh viên, tên, và điểm trung bình. Nếu để trống sẽ xuất hiện thông báo lỗi "Vui lòng nhập đầy đủ thông tin!"

- Kiểm tra mã số sinh viên phải có 10 kí tự. Nếu không sẽ xuất thông báo "Mã số sinh viên phải có 10 kí tự!"
- Nếu trường hợp nhấn vào nút "Thêm" thì Thêm mới dữ liệu sinh viên vừa nhập vào CSDL, Load lại DataGridView, và thông báo "**Thêm mới dữ liệu thành công!**".
- Nếu trường hợp nhấn vào nút "Sửa". Nếu mã sinh viên đã tồn tại thì Cập nhật dữ liệu sinh viên vào CSDL, và thông báo "**Cập nhật dữ liệu thành công!**". Nếu mã sinh viên đó không tồn tại thì xuất thông báo "**Không tìm thấy MSSV cần sửa!**".
- Reset lại dữ liệu về giá trị ban đầu sau khi thêm/ sửa thành công

### 1.3 Khi nhấn vào nút "Xóa"

- Kiểm tra nếu MSSV cần xóa không tồn tại trong CSDL thì thông báo lỗi "**Không tìm thấy MSSV cần xóa!**".
- Ngược lại thì xuất hiện cảnh báo YES/NO. Nhấn YES sẽ thực hiện xóa dòng dữ liệu sinh viên trong DataGridView và thông báo "**Xóa sinh viên thành công!**".
- Reset lại dữ liệu về giá trị ban đầu sau khi xóa thành công

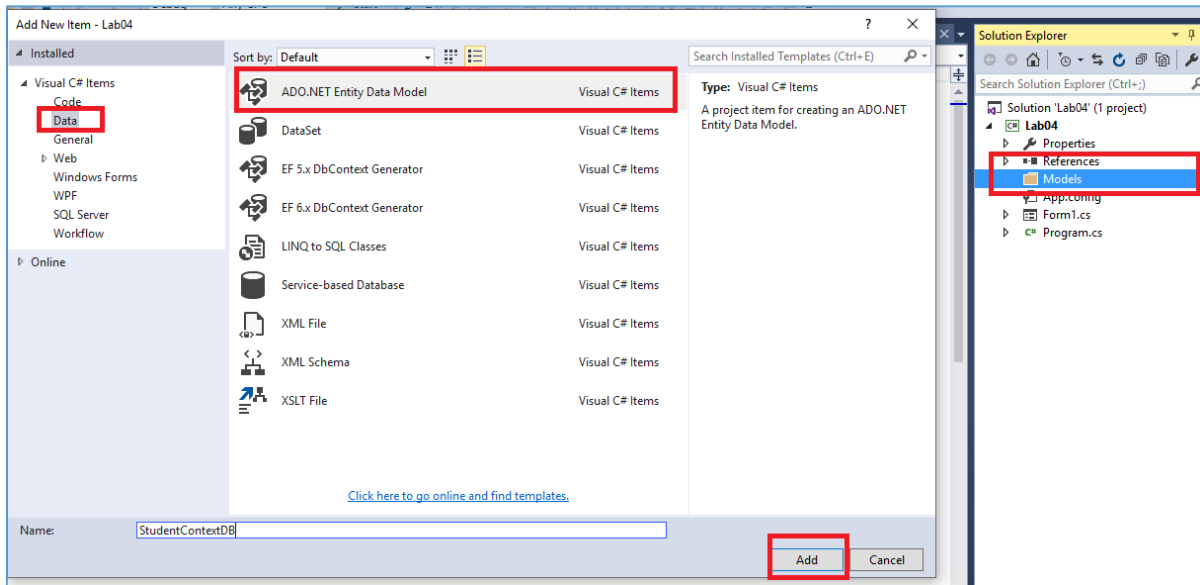
1.4 Viết code cho sự kiện ở DataGridView, người dùng chọn 1 dòng thì thể hiện ngược lại thông tin của các sinh viên đã chọn ở phần nhập liệu (bên trái).

### Hướng Dẫn

**Bước 1:** Entity Framework sinh ra các class chúng ta nên tạo trong 1 thư mục (**Models**) để dễ dàng quản lý.

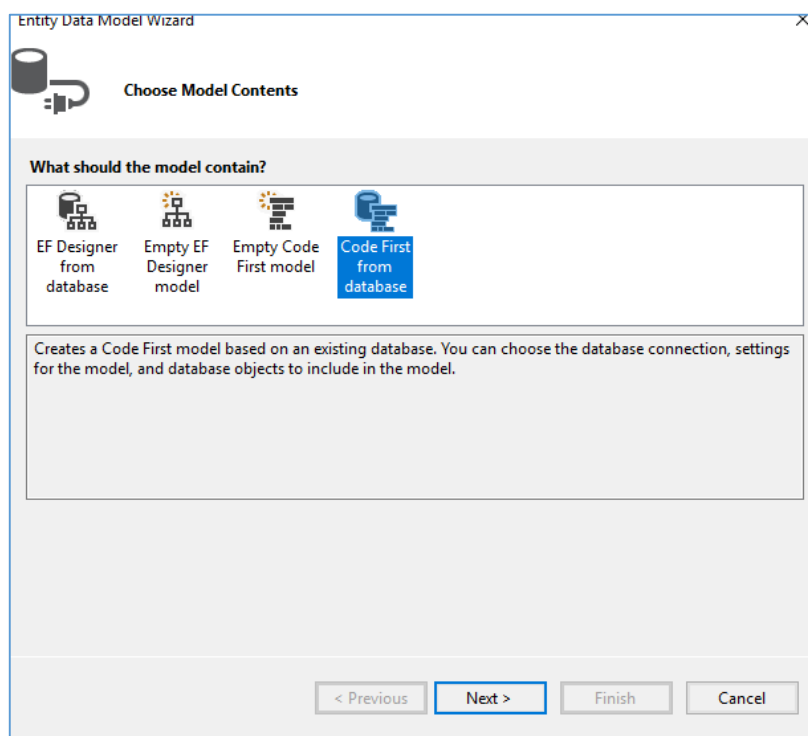
Click chuột phải vào Models chọn **New Item**. Chọn Loại **Data/ ADO.NET Entity Data Model**

Đặt tên context là "**StudentContextDB**" (mặc định là Model1). Và chọn **Add**

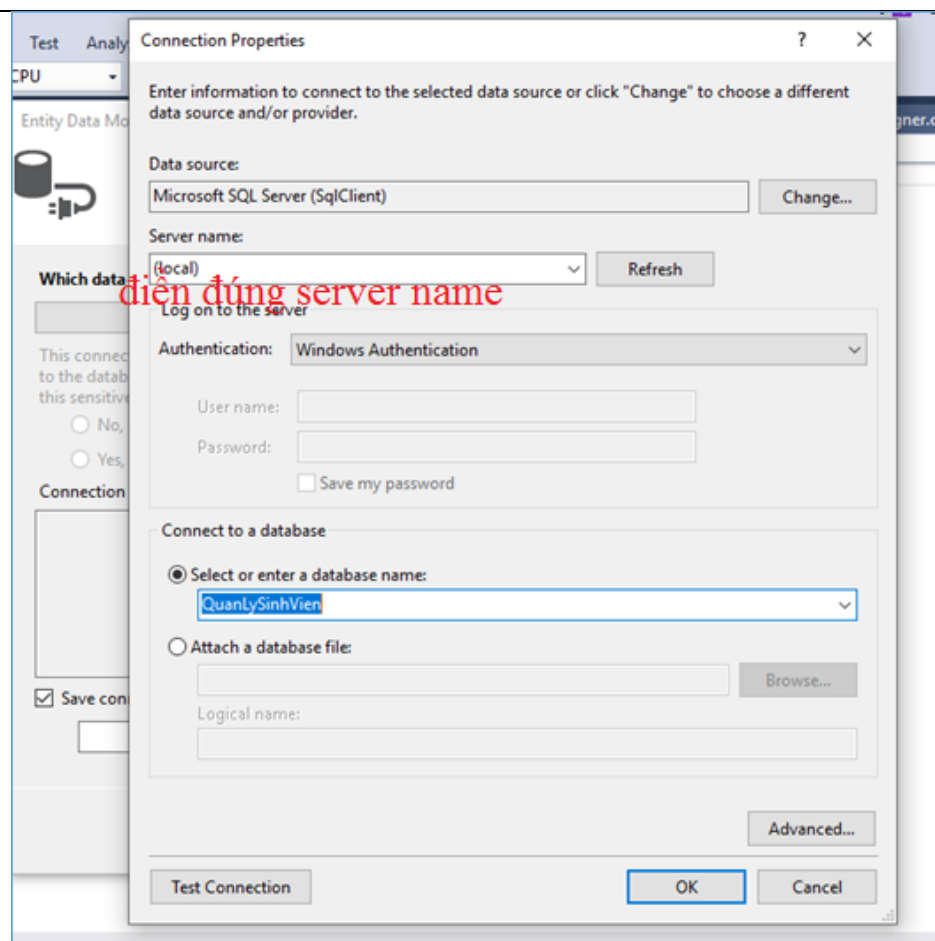


Sau khi chọn Add có 4 hướng loại Entity model để kết nối với cơ sở dữ liệu

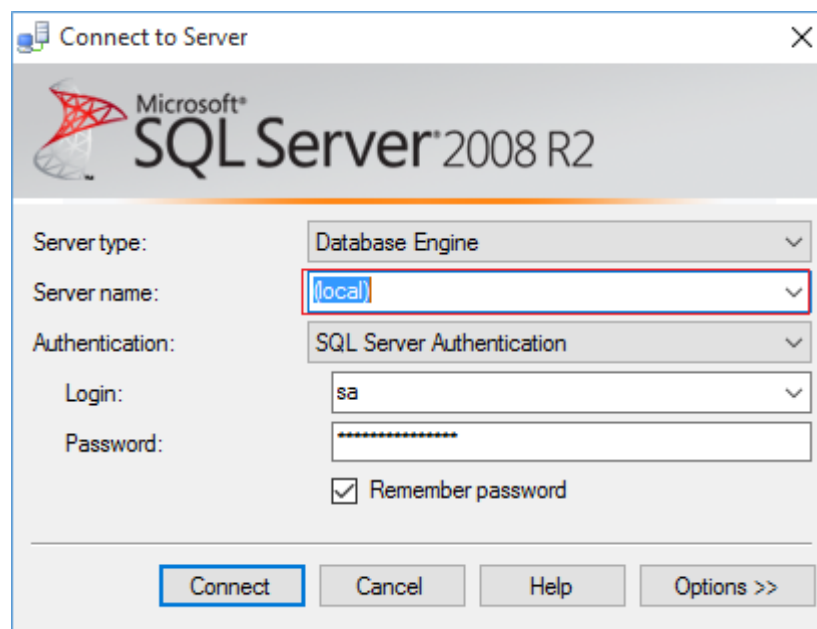
- Database First
- Model First
- Code First với hướng tiếp cận tạo ra cơ sở dữ liệu
- Code First với hướng tiếp cận đã có sẵn cơ sở dữ liệu



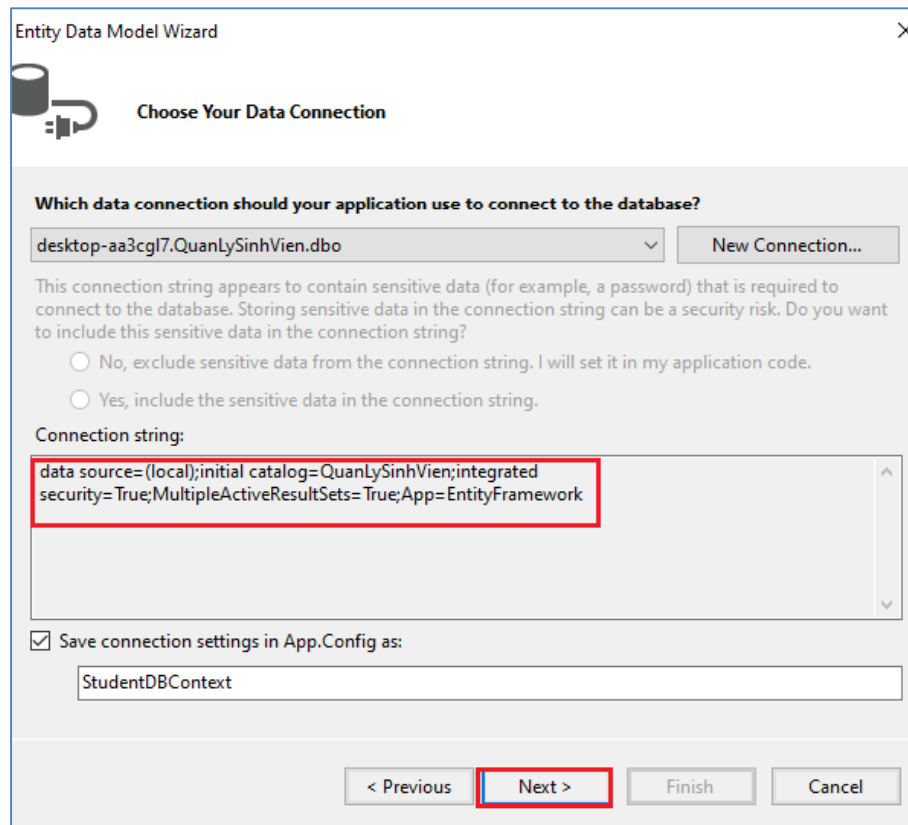
Ta chọn loại model là "**Code first from database**". Chọn **Next**  
Tìm đúng cơ sở dữ liệu Student ở SQL để trở database name vào



Chọn đúng tên server Name (trên từng máy có thể khác nhau - sinh viên có thể re-connect lại database để xem đúng tên Server Name. Ở ví dụ đây là trên local máy cá nhân)

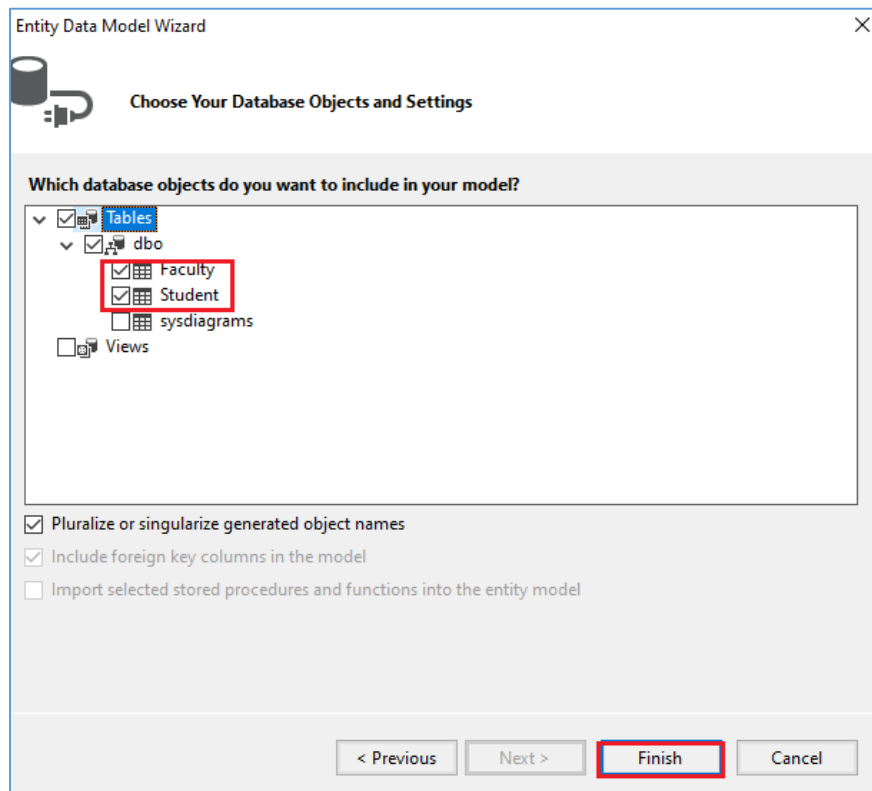


Chọn Next để tiếp tục tạo. Connect String sẽ được lưu ở file App.Config



The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The window title is 'Entity Data Model Wizard'. The main heading is 'Choose Your Data Connection'. Below this, it asks 'Which data connection should your application use to connect to the database?'. A dropdown menu shows 'desktop-aa3cgl7.QuanLySinhVien.dbo'. To the right is a 'New Connection...' button. Below the dropdown, a warning message states: 'This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.'. Below this, the 'Connection string:' field is highlighted with a red box and contains the text: 'data source=(local);initial catalog=QuanLySinhVien;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework'. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked. Below it, a text box contains 'StudentDBContext'. At the very bottom, there are four buttons: '< Previous', 'Next >' (highlighted with a red box), 'Finish', and 'Cancel'.

Sau đó chọn các bảng muốn tạo object (ở đây chọn 2 bảng từ CSDL)



The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Database Objects and Settings' step. The window title is 'Entity Data Model Wizard'. The main heading is 'Choose Your Database Objects and Settings'. Below this, it asks 'Which database objects do you want to include in your model?'. There is a tree view showing the database structure. Under 'Tables', the 'dbo' folder is expanded, and 'Faculty' and 'Student' are selected (highlighted with a red box). Other options like 'sysdiagrams' and 'Views' are also visible. Below the tree view, there are three checkboxes: 'Pluralize or singularize generated object names' (checked), 'Include foreign key columns in the model' (checked), and 'Import selected stored procedures and functions into the entity model' (unchecked). At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish' (highlighted with a red box), and 'Cancel'.

Sau khi Finish, Entity FrameWork đã tạo các class tương ứng như trong cơ sở dữ liệu



Một số thông tin cần lưu ý:

- App.Config: Sẽ lưu trữ connection String và tên của Context sử dụng
- StudentContextDB: Lớp chứa tập hợp DataSet cho các table được chọn.

VS.Net tự động sinh ra các file cs tương ứng map với CSDL. Sinh viên kiểm tra từng file (Student, Faculty) để hiểu cách mapping tương ứng với CSDL.

- ✓ Cách Sử dụng Entity để thao tác với cơ sở dữ liệu như: Lấy tất cả, thêm, xóa, sửa với CSDL (Sinh viên đọc kĩ và thử để thực hiện bài tập)

```
//luôn luôn sử dụng context để làm việc với các class
StudentContextDB context = new StudentContextDB();

//1. lấy tất cả các sinh viên từ bảng Student
List<Student> listStudent = context.Students.ToList();
//2. lấy sinh viên đầu tiên có StudentID = ID cho trước
Student db = context.Students.FirstOrDefault(p => p.StudentID == ID);
//3. insert 1 đối tượng sinh viên s vào database
Student s = new Student() { StudentID = "99", FullName = "test insert", AverageScore = 100 };
context.Students.Add(s);
context.SaveChanges();
//4. Update sinh viên -> lấy item ra và cần update thuộc tính nào thì set thuộc tính đó
Student dbUpdate = context.Students.FirstOrDefault(p => p.StudentID == ID);
if(dbUpdate!= null){
    dbUpdate.FullName = "Update FullName"; //....
    context.SaveChanges(); //lưu thay đổi
```

```
}  
//5. Xóa Student có ID cho trước , tương tự update  
Student dbDelete = context.Students.FirstOrDefault(p => p.StudentID == ID);  
if (dbDelete != null) {  
    context.Students.Remove(db);  
    context.SaveChanges(); // lưu thay đổi  
}  
//6. Lưu ý: Nếu sử dụng using System.Data.Entity.Migrations; có thể sử  
dụng hàm AddOrUpdate để thay thế Add và Update từ EntityFramework 6.0.0.0  
context.Students.AddOrUpdate(s); //Add or Update sinh viên s  
context.SaveChanges();
```

## Bước 2: Thiết kế và lập trình – Viết sự kiện Form-Load

```
private void frmStudentManagement_Load(object sender, EventArgs e)  
{  
    try  
    {  
        StudentContextDB context = new StudentContextDB();  
        List<Faculty> listFaculty = context.Faculties.ToList(); //lấy các khoa  
        List<Student> listStudent = context.Students.ToList(); //lấy sinh viên  
        FillFacultyCombobox(listFaculty);  
        BindGrid(listStudent);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}  
//Hàm binding list có tên hiển thị là tên khoa, giá trị là Mã khoa  
private void FillFacultyCombobox(List<Faculty> listFaculty)  
{  
    this.cmbFaculty.DataSource = listFaculty;  
    this.cmbFaculty.DisplayMember = "FacultyName";  
    this.cmbFaculty.ValueMember = "FacultyID";  
}  
//Hàm binding gridView từ list sinh viên  
private void BindGrid(List<Student> listStudent)  
{  
    dgvStudent.Rows.Clear();  
    foreach (var item in listStudent)  
    {  
        int index = dgvStudent.Rows.Add();  
        dgvStudent.Rows[index].Cells[0].Value = item.StudentID;
```



```

        dgvStudent.Rows[index].Cells[1].Value = item.FullName;
        dgvStudent.Rows[index].Cells[2].Value = item.Faculty.FacultyName ;
        dgvStudent.Rows[index].Cells[3].Value = item.AverageScore;
    }
}
// Sinh viên tự viết các sự kiện thêm, xóa, sửa sau khi đọc hướng dẫn sử dụng Entity để thao
tác với CSDL.

```

**Bài tập 2:** Tạo Form quản lý thông tin các khoa ở cùng project trong bài tập 1

- Thêm 1 cột **TotalProfessor** (tổng số giáo sư) kiểu INT cho phép *NULL* vào bảng **Faculty**

	Column Name	Data Type	Allow Nulls
🔑	FacultyID	int	<input type="checkbox"/>
	FacultyName	nvarchar(200)	<input type="checkbox"/>
▶	TotalProfessor	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- Thêm 1 form mới là **frmFaculty** có đủ các chức năng thêm, xóa, sửa, hiện thị thông tin khoa tương tự như quản lý Sinh viên ở bài tập 1.

	Mã Khoa	Tên Khoa	Tổng số GS
	1	Công Nghệ Thông Tin	
	2	Ngôn Ngữ Anh	
▶	3	Quản trị kinh doanh	

- Tạo 1 Button ở Form Quản lý sinh viên để khi click đó sẽ gọi sang Form quản lý thông tin các khoa (Hoặc sinh viên có thể dùng MenuStrip – tạo ra Sub menu)
- Thực hiện các yêu cầu trên form quản lý khoa: Lấy dữ liệu vào DataGridView, Thêm/Sửa, Xóa và Đóng form.

**Chú ý:** Khi CSDL có thay đổi sinh viên nên cập nhật lại phần models **được thay đổi** bằng cách tương tự như lúc tạo ra ban đầu (xóa đi – tạo lại hoặc đưa phần thay đổi chèn vào models hiện tại). Mục đích để đảm bảo CSDL đó đúng với Models.

### Bài tập 3: Tìm kiếm sinh viên

**Thiết kế chương trình quản lý sinh viên như giao diện sau** (Thiết kế thêm Chức năng, Quản lý khoa, tìm kiếm và form tìm kiếm sinh viên từ bài tập 1-2)

Mã Số SV	Họ Tên	Tên Khoa	Điểm TB
1611061916	Nguyễn Trần Hoàng Lan	Công Nghệ Thông Tin	4.5
1711060596	Đàm Minh Đức	Công Nghệ Thông Tin	2.5
1711061004	Nguyễn Quốc An	Ngôn Ngữ Anh	10

- Sử dụng ToolStrip, 2 button Quản lý khoa và Tìm kiếm
- Ở Menu chức năng: thể hiện các chức năng có phím tắt để tới các form: Quản lý khoa (F2) và Tìm Kiếm (Ctrl +F)

Chức năng

- Quản lý khoa F2
- Tìm Kiếm Ctrl+F
- Thoát

- Thiết kế form Tìm kiếm và thực hiện tìm kiếm thông tin sinh viên



**Bài tập 4:** Cho cơ sở dữ liệu quản lý sản phẩm và đơn hàng như sau

- ✓ Sử dụng cơ sở dữ liệu SQL server có 3 bảng **Product**, **Order**, **Invoice** lần lượt như sau

**Product:** Lưu trữ thông tin sản phẩm (**Mã sản phẩm**, Tên Sản phẩm, Đơn vị Tính, Giá Mua, Giá Bán)

**Order:** Lưu trữ chi tiết thông tin đơn hàng (**Số HĐ**, **Số TT**, Mã SP, Tên SP, ĐVT, Đơn giá, Số lượng)

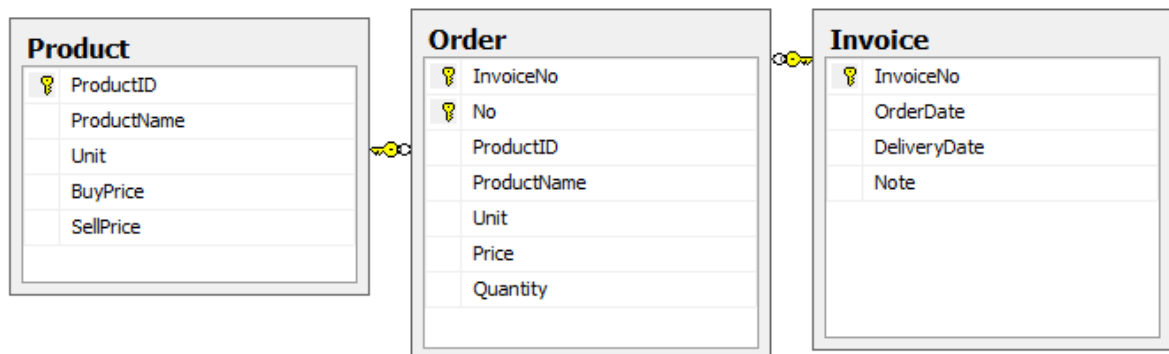
**Invoice:** Lưu trữ thông tin hóa đơn đặt hàng (**Số HĐ**, Ngày đặt hàng, ngày giao hàng, ghi chú)

Column Name	Data Type	Allow Nulls
ProductID	nvarchar(20)	<input type="checkbox"/>
ProductName	nvarchar(100)	<input type="checkbox"/>
Unit	nvarchar(20)	<input type="checkbox"/>
BuyPrice	decimal(18, 0)	<input checked="" type="checkbox"/>
SellPrice	decimal(18, 0)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
InvoiceNo	nvarchar(20)	<input type="checkbox"/>
No	int	<input type="checkbox"/>
ProductID	nvarchar(20)	<input type="checkbox"/>
ProductName	nvarchar(100)	<input checked="" type="checkbox"/>
Unit	nvarchar(20)	<input checked="" type="checkbox"/>
Price	decimal(18, 0)	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
InvoiceNo	nvarchar(20)	<input type="checkbox"/>
OrderDate	datetime	<input type="checkbox"/>
DeliveryDate	datetime	<input type="checkbox"/>
Note	nvarchar(255)	<input checked="" type="checkbox"/>

- ✓ Sơ đồ diagrams



Sử dụng script để tạo ra nhanh CSDL, và dữ liệu tương ứng

```

USE [ProductOrder]
GO
/***** Object:  Table [dbo].[Invoice]      Script Date: 07/04/2020 23:13:15
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Invoice] (
    [InvoiceNo] [nvarchar] (20) NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [DeliveryDate] [datetime] NOT NULL,
    [Note] [nvarchar] (255) NULL,
    CONSTRAINT [PK_Invoice] PRIMARY KEY CLUSTERED
  
```

```

(
    [InvoiceNo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Product]      Script Date: 07/04/2020 23:13:15
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Product] (
    [ProductID] [nvarchar](20) NOT NULL,
    [ProductName] [nvarchar](100) NOT NULL,
    [Unit] [nvarchar](20) NOT NULL,
    [BuyPrice] [decimal](18, 0) NULL,
    [SellPrice] [decimal](18, 0) NULL,
    CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
    [ProductID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Order]      Script Date: 07/04/2020 23:13:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Order] (
    [InvoiceNo] [nvarchar](20) NOT NULL,
    [No] [int] NOT NULL,
    [ProductID] [nvarchar](20) NOT NULL,
    [ProductName] [nvarchar](100) NULL,
    [Unit] [nvarchar](20) NULL,
    [Price] [decimal](18, 0) NOT NULL,
    [Quantity] [int] NOT NULL,
    CONSTRAINT [PK_Order] PRIMARY KEY CLUSTERED
(
    [InvoiceNo] ASC,
    [No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: ForeignKey [FK_Order_Invoice]      Script Date: 07/04/2020
23:13:15 *****/
ALTER TABLE [dbo].[Order] WITH CHECK ADD CONSTRAINT [FK_Order_Invoice] FOREIGN
KEY([InvoiceNo])
REFERENCES [dbo].[Invoice] ([InvoiceNo])
GO
ALTER TABLE [dbo].[Order] CHECK CONSTRAINT [FK_Order_Invoice]
GO
/***** Object: ForeignKey [FK_Order_Product]      Script Date: 07/04/2020
23:13:15 *****/
ALTER TABLE [dbo].[Order] WITH CHECK ADD CONSTRAINT [FK_Order_Product] FOREIGN
KEY([ProductID])
REFERENCES [dbo].[Product] ([ProductID])
GO

```

```
ALTER TABLE [dbo].[Order] CHECK CONSTRAINT [FK_Order_Product]
GO
```

### Thêm 1 số dữ liệu vào database như sau

```
USE [ProductOrder]
GO

/***** Object: Table [dbo].[Invoice]      Script Date: 07/04/2020 23:14:16
*****/

INSERT [dbo].[Invoice] ([InvoiceNo], [OrderDate], [DeliveryDate], [Note]) VALUES
(N'HDX001', CAST(0x0000AAD900000000 AS DateTime), CAST(0x0000AADA00000000 AS
DateTime), N'Giao hàng trước 9h')

INSERT [dbo].[Invoice] ([InvoiceNo], [OrderDate], [DeliveryDate], [Note]) VALUES
(N'HDX002', CAST(0x0000AADA00000000 AS DateTime), CAST(0x0000AADA00000000 AS
DateTime), N'Gọi điện trước khi giao')

INSERT [dbo].[Invoice] ([InvoiceNo], [OrderDate], [DeliveryDate], [Note]) VALUES
(N'HDX003', CAST(0x0000AADA00000000 AS DateTime), CAST(0x0000AADC00000000 AS
DateTime), N'giao tu 1-3h')

/***** Object: Table [dbo].[Product]      Script Date: 07/04/2020 23:14:16
*****/

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product1', N'Sản phẩm 1', N'Cái', CAST(100000 AS
Decimal(18, 0)), CAST(120000 AS Decimal(18, 0)))

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product2', N'Sản phẩm 2', N'Cái', CAST(90000 AS
Decimal(18, 0)), CAST(120000 AS Decimal(18, 0)))

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product3', N'Sản phẩm 3', N'Cái', CAST(40000 AS
Decimal(18, 0)), CAST(70000 AS Decimal(18, 0)))

INSERT [dbo].[Product] ([ProductID], [ProductName], [Unit], [BuyPrice],
[SellPrice]) VALUES (N'Product4', N'Sản phẩm 4', N'Hộp', CAST(200000 AS
Decimal(18, 0)), CAST(300000 AS Decimal(18, 0)))

/***** Object: Table [dbo].[Order]      Script Date: 07/04/2020 23:14:16 *****/

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX001', 1, N'Product1', N'Sản phẩm 1', N'Cái',
CAST(120000 AS Decimal(18, 0)), 20)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX001', 2, N'Product2', N'Sản phẩm 2', N'Cái',
CAST(120000 AS Decimal(18, 0)), 4)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX001', 3, N'Product4', N'Sản phẩm 4', N'Hộp',
CAST(300000 AS Decimal(18, 0)), 10)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX002', 1, N'Product4', N'Sản phẩm 1', N'Hộp',
CAST(300000 AS Decimal(18, 0)), 10)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX002', 2, N'Product2', N'Sản phẩm 3', N'Cái',
CAST(300000 AS Decimal(18, 0)), 12)

INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX003', 1, N'Product1', N'Sản phẩm 1', N'Cái',
CAST(120000 AS Decimal(18, 0)), 40)
```

```
INSERT [dbo].[Order] ([InvoiceNo], [No], [ProductID], [ProductName], [Unit],
[Price], [Quantity]) VALUES (N'HDX003', 4, N'Product2', N'Sản phẩm 2', N'Cái',
CAST(120000 AS Decimal(18, 0)), 60)
```

**Viết chương trình phần mềm xem thông tin đơn hàng như sau**

Thông Tin Đơn Hàng

☐ Xem tất cả trong tháng

Thời Gian Giao Hàng: 02/10/2019 ~ 02/10/2019

STT	Số HĐ	Ngày Đặt Hàng	Ngày Giao Hàng	Thành Tiền

Tổng Cộng: 0

### Khi load Form

- Thời gian giao hàng được thể hiện trong ngày hiện hành và tự động tìm kiếm dữ liệu có Hóa Đơn phát sinh trong ngày hiện hành này
- Người dùng có thể thay đổi thời gian giao hàng trong 1 khoảng thời gian bất kì, khi đó dữ liệu cũng được tự động thay đổi theo

Thông Tin Đơn Hàng

☒ Xem tất cả trong tháng

Thời Gian Giao Hàng: 02/10/2019 ~ 04/10/2019

STT	Số HĐ	Ngày Đặt Hàng	Ngày Giao Hàng	Thành Tiền
1	HDX001	01/10/2019	02/10/2019	5880000
2	HDX002	02/10/2019	02/10/2019	6600000

- Khi check vào CheckBox Xem tất cả trong tháng, thì thời gian giao hàng sẽ được thể hiện từ **ngày đầu tháng hiện hành** đến **cuối tháng** và hiện thị thông tin giao hàng trong thời gian đó.

# BÀI 5: TẠO BÁO CÁO THỐNG KÊ VỚI DATA REPORT

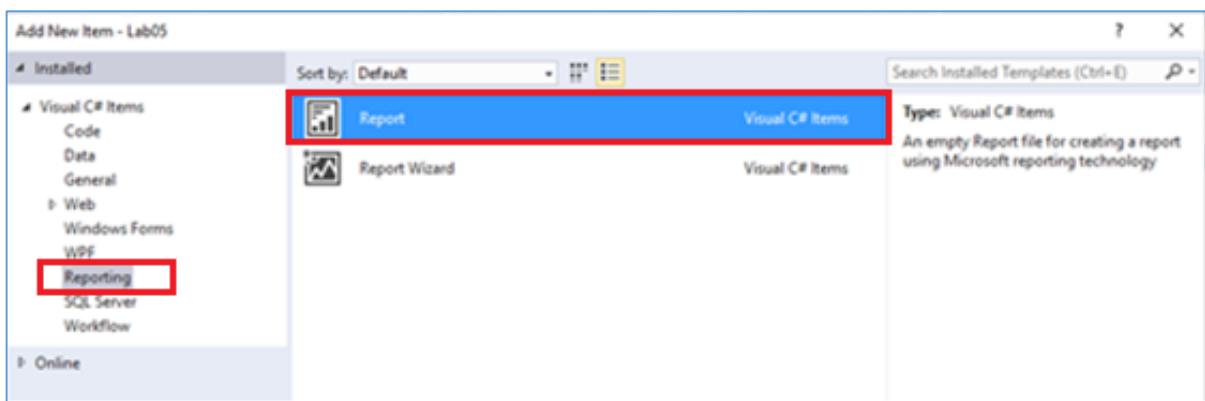
## 5.1 MỤC TIÊU

- Cài đặt, Thiết kế các báo cáo thống kê với Data Report: Sử dụng Report Data để thiết kế, Control Report Viewer để hiển thị các bản thiết kế báo cáo.
- Ôn tập lại các controls điều khiển trên windows form.
- Ôn tập kết nối cơ sở dữ liệu với Entity Framework Code First

## 5.2 HƯỚNG DẪN TẠO DATA REPORT

Tạo project Lab05 Windows Application để thực hiện. Xuất danh sách các sinh viên từ CSDL ra Report Data. ( Sử dụng cơ sở dữ liệu QuanLySinhVien Từ bài Lab04)

- ✓ Bước 1: Kiểm tra xem đã cài đặt Report cho VS hay chưa, nếu đã cài đặt rồi thì chuyển qua bước 3. Kiểm tra bằng cách sau *khi tạo 1 project Windows Form*, Nếu chọn Add/ New Item thấy đã có phần item **Reporting (chuyển qua b3 – máy ở trường đã cài đặt rồi)**



- ✓ Bước 2: Cài đặt **Report Viewer** Cho Visual Studio 2015 (nếu chưa cài đặt)
- Vào Control Panel chọn Programs And Feature , Chọn Visual Studio Installer , Click phải chọn Change

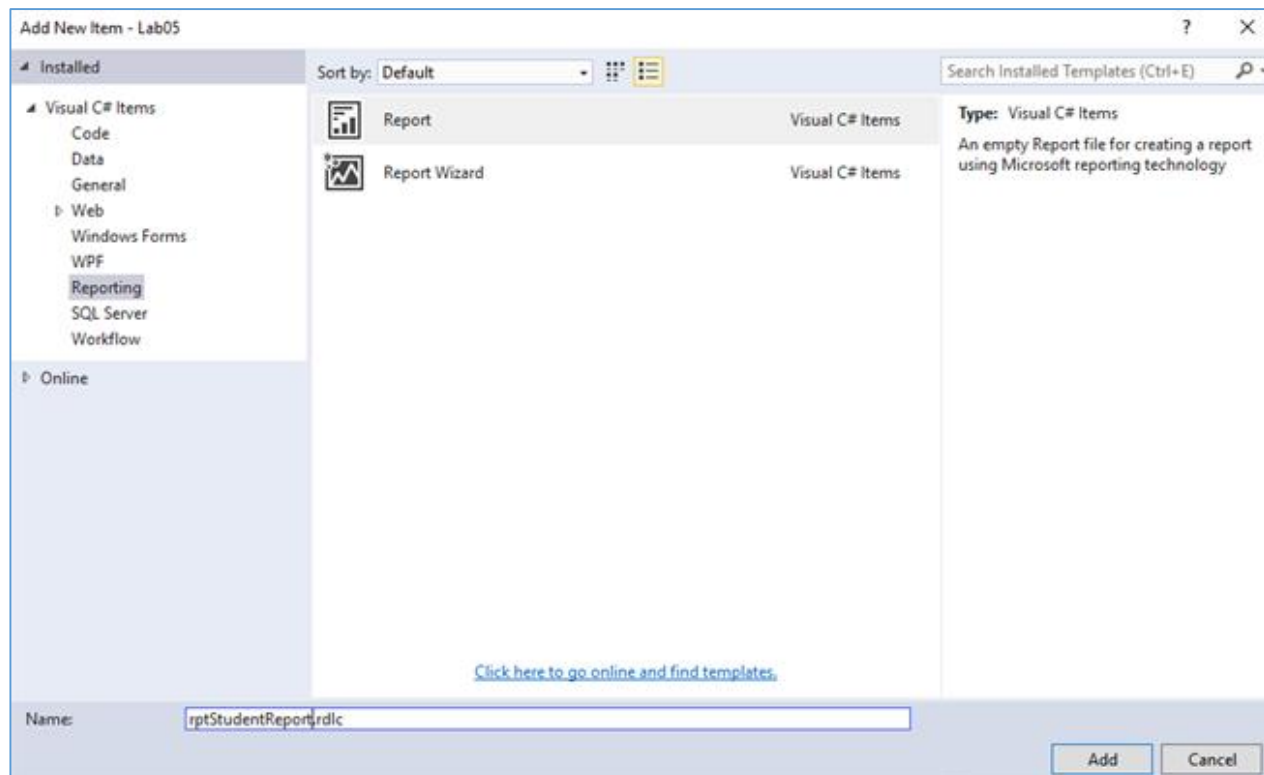


Microsoft Visual Studio 2008 Shell (integrated mode) ...	Microsoft Corporation	8/11/2019	224 MB	9.0.30729
Microsoft Visual Studio Installer	Microsoft Corporation	8/11/2019		2.2.3073.701
Microsoft Visual Studio 2015 with Updates	Microsoft Corporation	8/11/2019	6.15 GB	14.0.25420.1
Microsoft Visual Studio 2015 Tools - E...	Microsoft Corporation	8/11/2019	244 MB	9.0.35191
Microsoft Web Deploy 4.0	Microsoft Corporation	8/11/2019	9.55 MB	10.0.2606

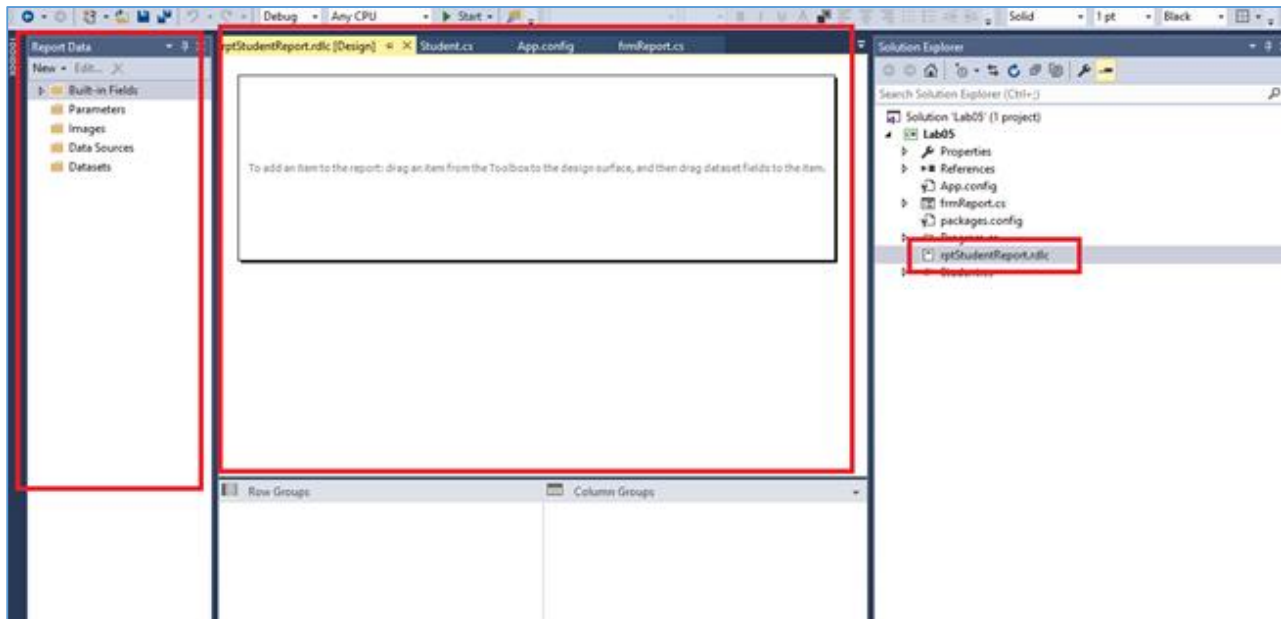
- Check vào Microsoft SQL Server Data Tools, Next để tiến hành hoàn tất



- ✓ Bước 3: Tạo bản thiết kế Report Data ( file .rdlc)



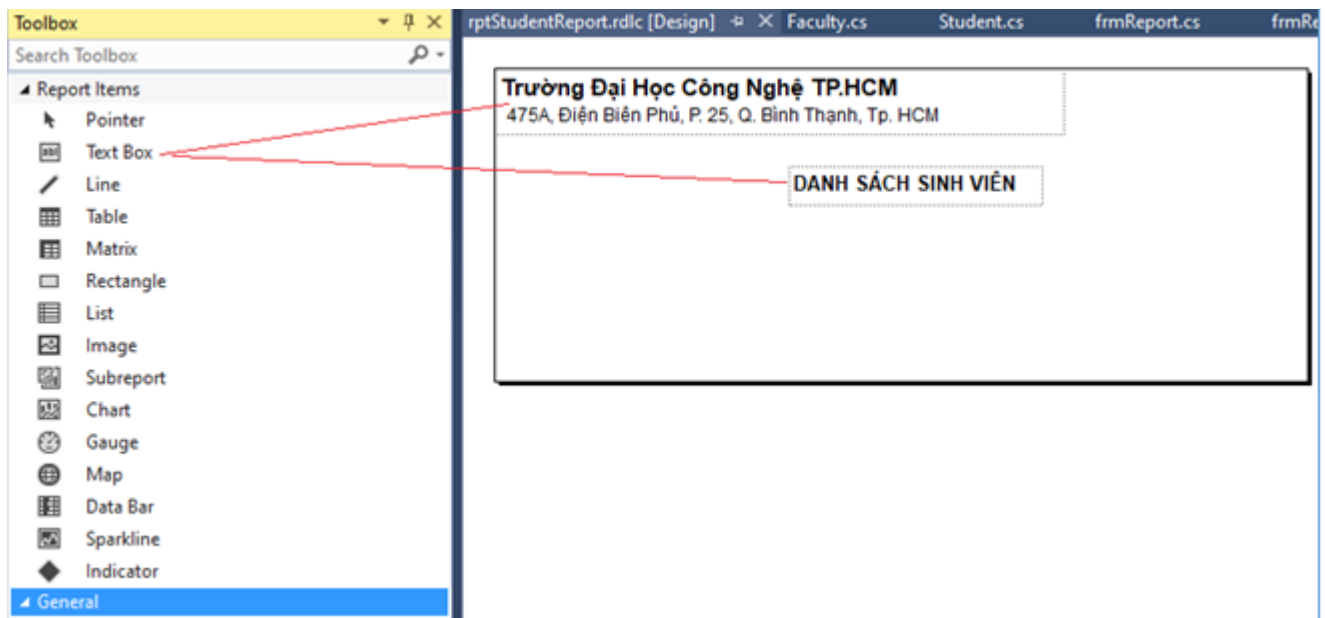
Thiết kế Giao diện của phần report design



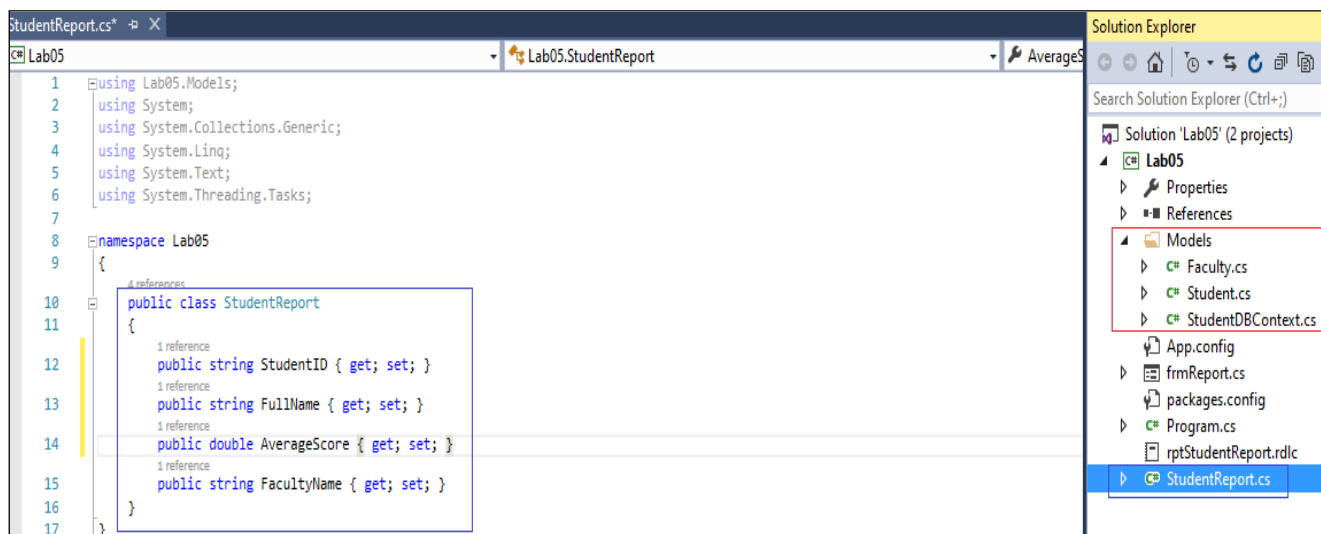
✓ Bước 4: Thiết kế mẫu Report Bằng cách sử dụng toolbox

- Sử dụng Text để hiển thị thông tin trên Report file.
- Sử dụng Table: Để lấy List dữ liệu từ dataset cụ thể (Database, Object, Sharepoint ...). Trong bài thực hành này sẽ mô tả lấy từ CSDL bảng Student được mô tả từ Lab3.

Thiết kế bảng báo cáo danh sách sinh viên như sau

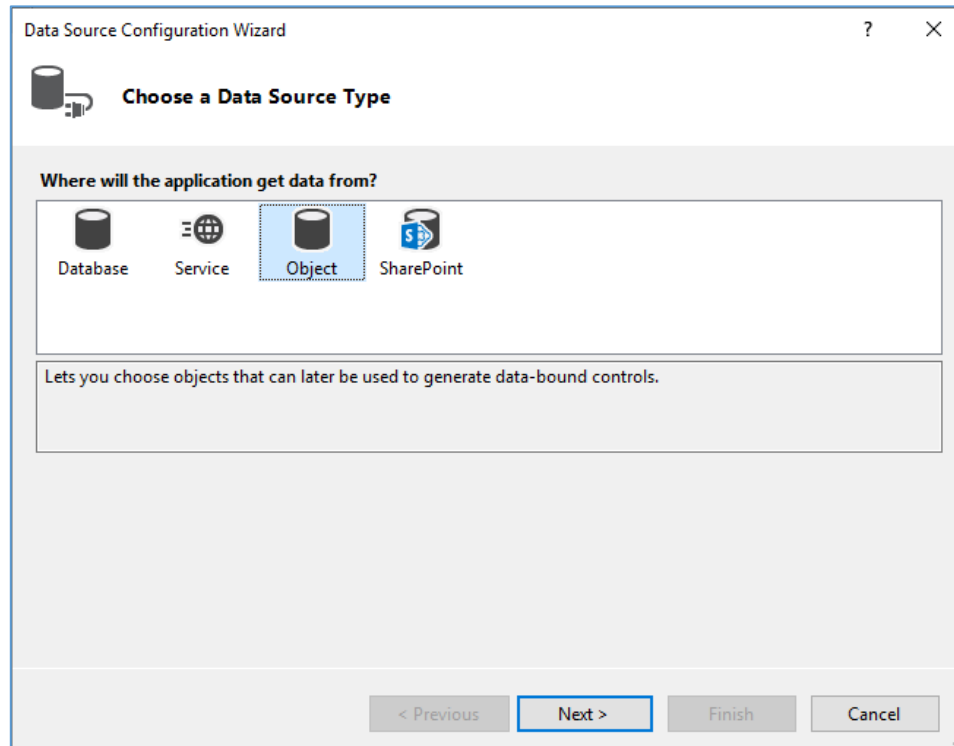


- ✓ Bước 5: Sử dụng Entity Framework Code First để kết nối với cơ sở dữ liệu “QuanLySinhVien” đã được mô tả ở Lab03 (folder models).
- Add thêm 1 class *StudentReport.cs* chứa các thông tin xuất báo cáo (mã sinh viên, họ tên, điểm, và tên khoa)

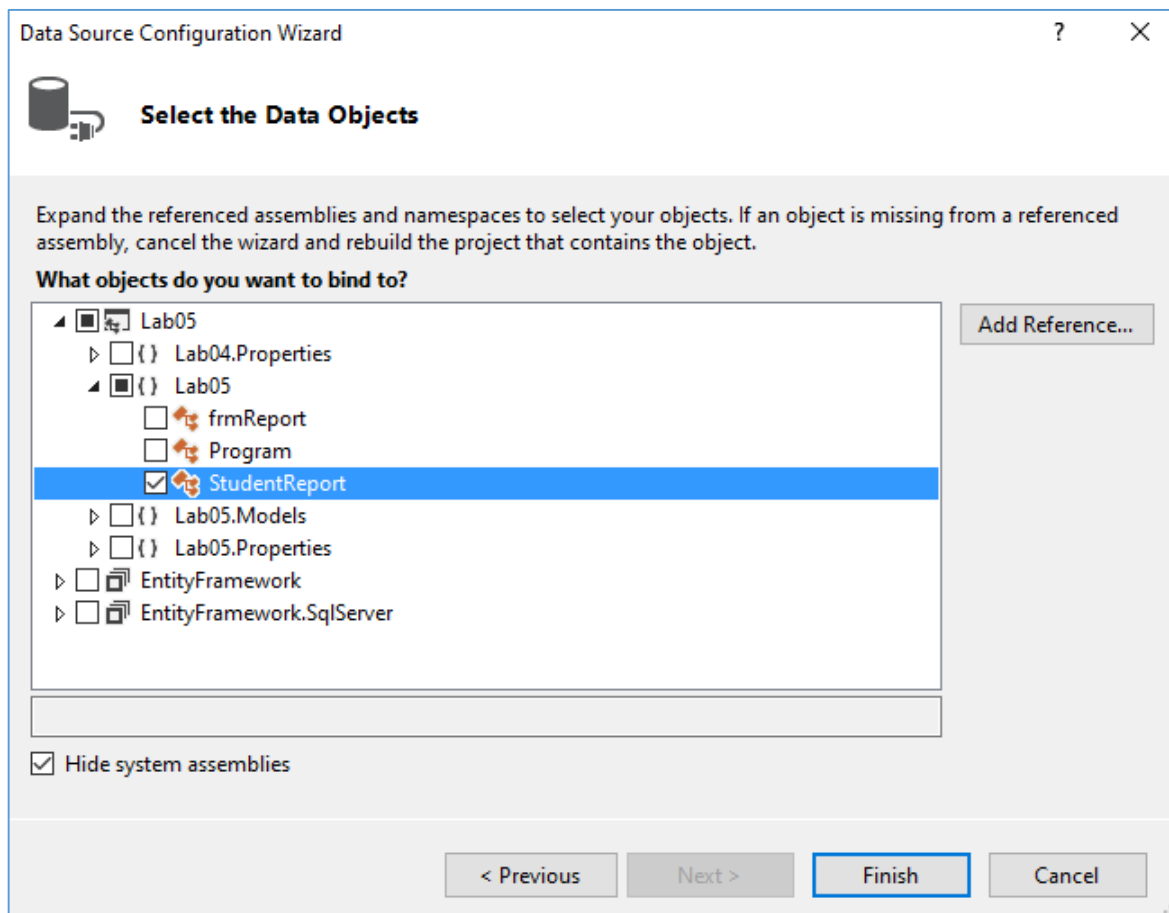


- Tạo DataSet và loại Object và trở tới class muốn lấy dữ liệu trong file report data

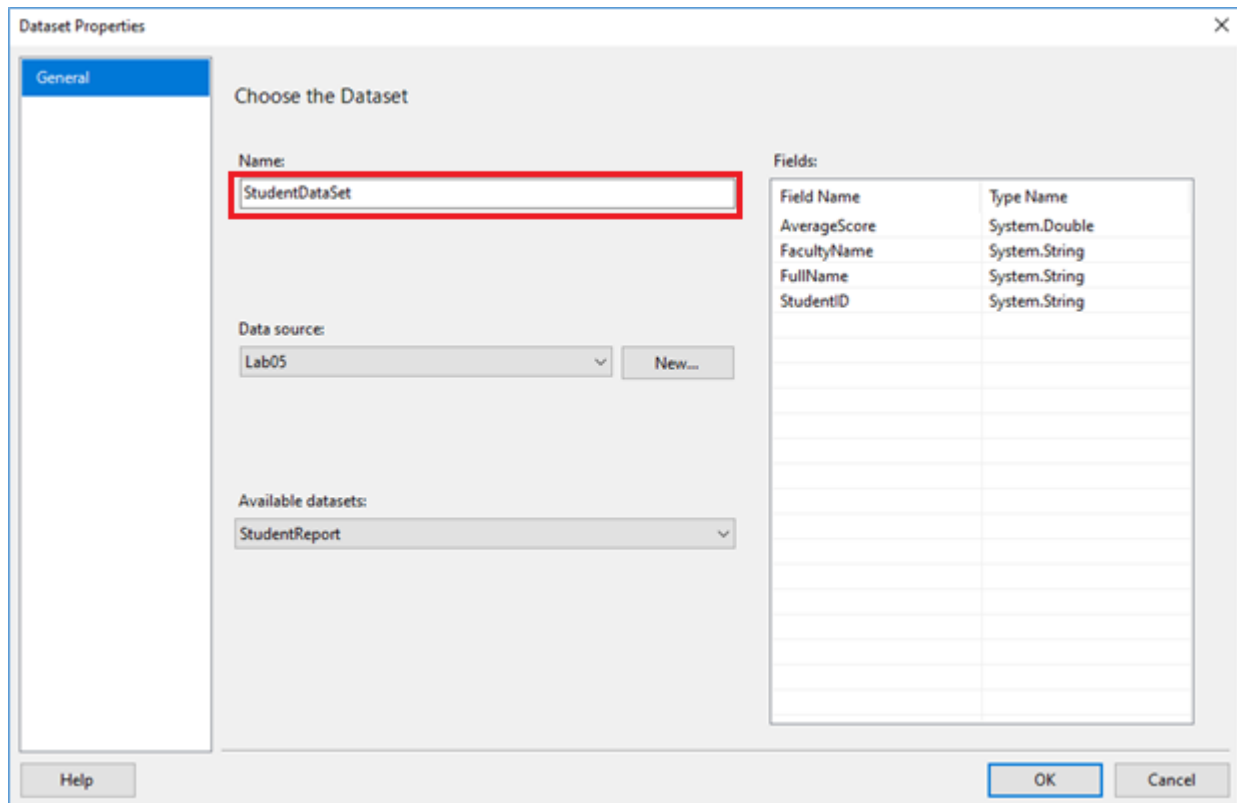




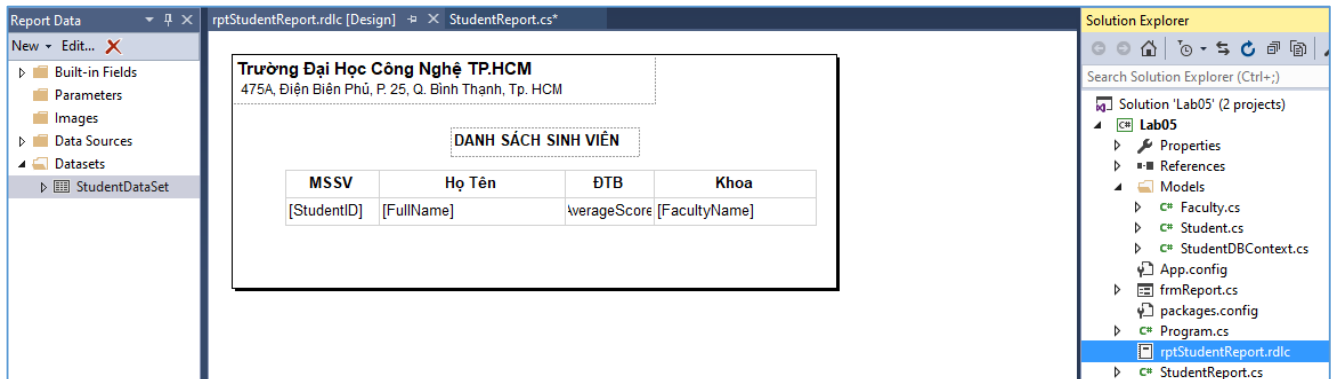
Chọn Next, và chọn StudentReport được tạo ra ở bước 5. (Nếu không thấy SV phải build lại project trước đó)



- Đặt tên cho DataSet chứa trong report file

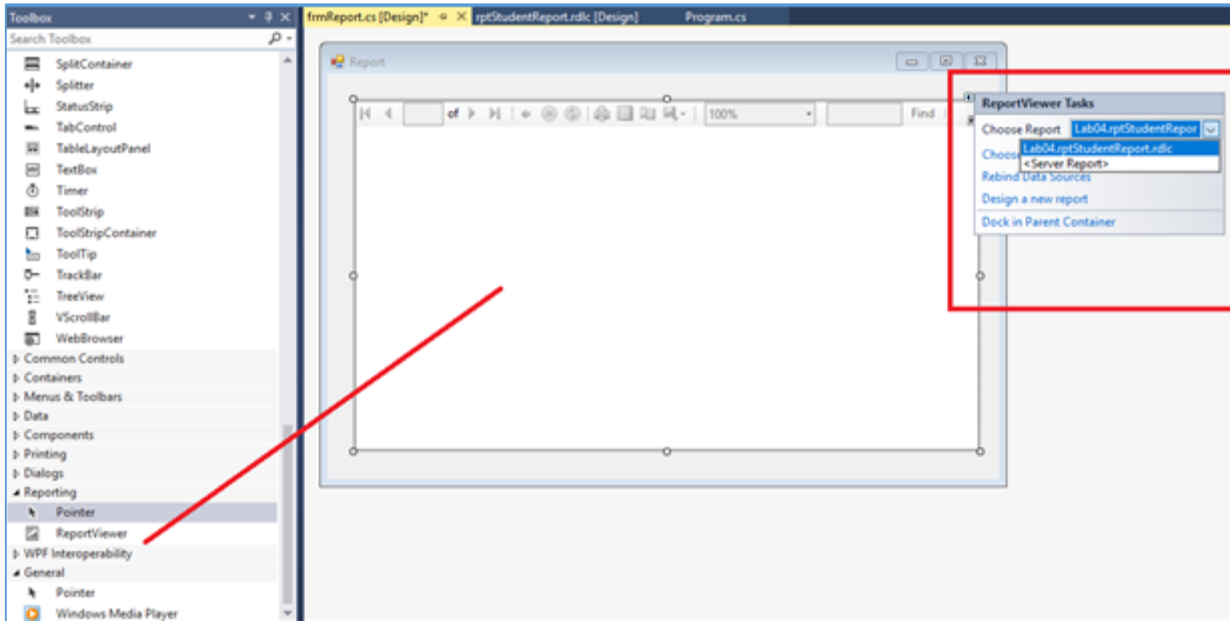


- Chọn các field tương ứng vào report và thiết kế lại kích thước, tên header tương ứng



- ✓ Bước 6: Sử dụng Control **ReportViewer** để gọi Report Data vừa thiết kế

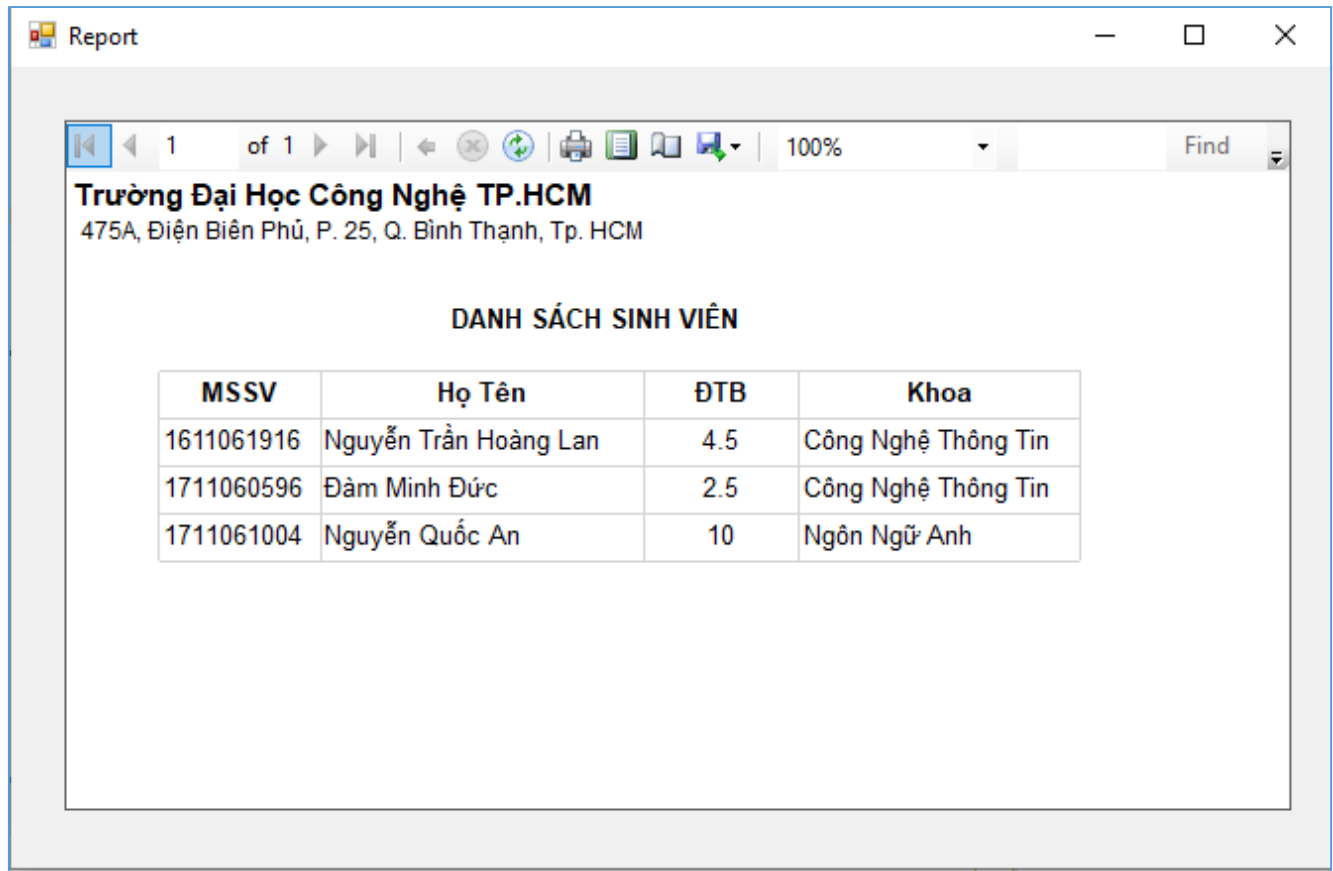
**ReportViewer:** là 1 control giúp hiện thị Data Report (bản thiết kế báo cáo). Có thể nhúng sẵn Report Data (rdlc) trong ReportViewer hoặc sử dụng code. Khi sử dụng bằng Code thì dễ dàng tùy chọn Report Data sẽ hiện thị.



✓ Bước 7: Viết sự kiện Form\_Load để hiển thị thông tin report

```
private void frmStudentManagement_Load(object sender, EventArgs e)
{
    StudentDBContext context = new StudentDBContext();
    List<Student> listStudent = context.Students.ToList(); //lấy tất cả sv
    List<StudentReport> listReport = new List<StudentReport>();
    foreach (Student i in listStudent)
    {
        StudentReport temp = new StudentReport();
        temp.StudentID = i.StudentID;
        temp.FullName = i.FullName;
        temp.AverageScore = i.AverageScore;
        temp.FacultyName = i.Faculty.FacultyName;
        listReport.Add(temp);
    }
    this.reportViewer1.LocalReport.ReportPath = "rptStudentReport.rdlc";
    var reportDataSource = new ReportDataSource("StudentDataSet", listReport); //đúng
    //tên dataset trong thiết kế
    this.reportViewer1.LocalReport.DataSources.Clear(); //clear
    this.reportViewer1.LocalReport.DataSources.Add(reportDataSource);
    this.reportViewer1.RefreshReport(); //chạy report
}
```

- Chạy chương trình để xem kết quả



- Như vậy Report Data có thể lấy dữ liệu từ **Object**, Entity Data Model cũng tạo ra các **Object** tương ứng. Trong Report Data chúng ta có thể sử dụng trực tiếp các object được tạo từ Entity Data Model ( SV làm bài tập 1 để thể hiện dữ liệu Report Data được lấy từ CSDL )

## 5.3 BÀI TẬP

**Bài tập 1:** Sử dụng lại cơ sở dữ liệu ở bài tập 4 – Lab4, Sinh Viên thêm dữ liệu giúp cho việc kiểm tra thông tin trên Data Report

- ✓ Sử dụng cơ sở dữ liệu "**ProductOrder**" có 3 bảng **Product**, **Order**, **Invoice** lần lượt như sau ( Xem mô tả lại ở Lab03-bài tập 3)

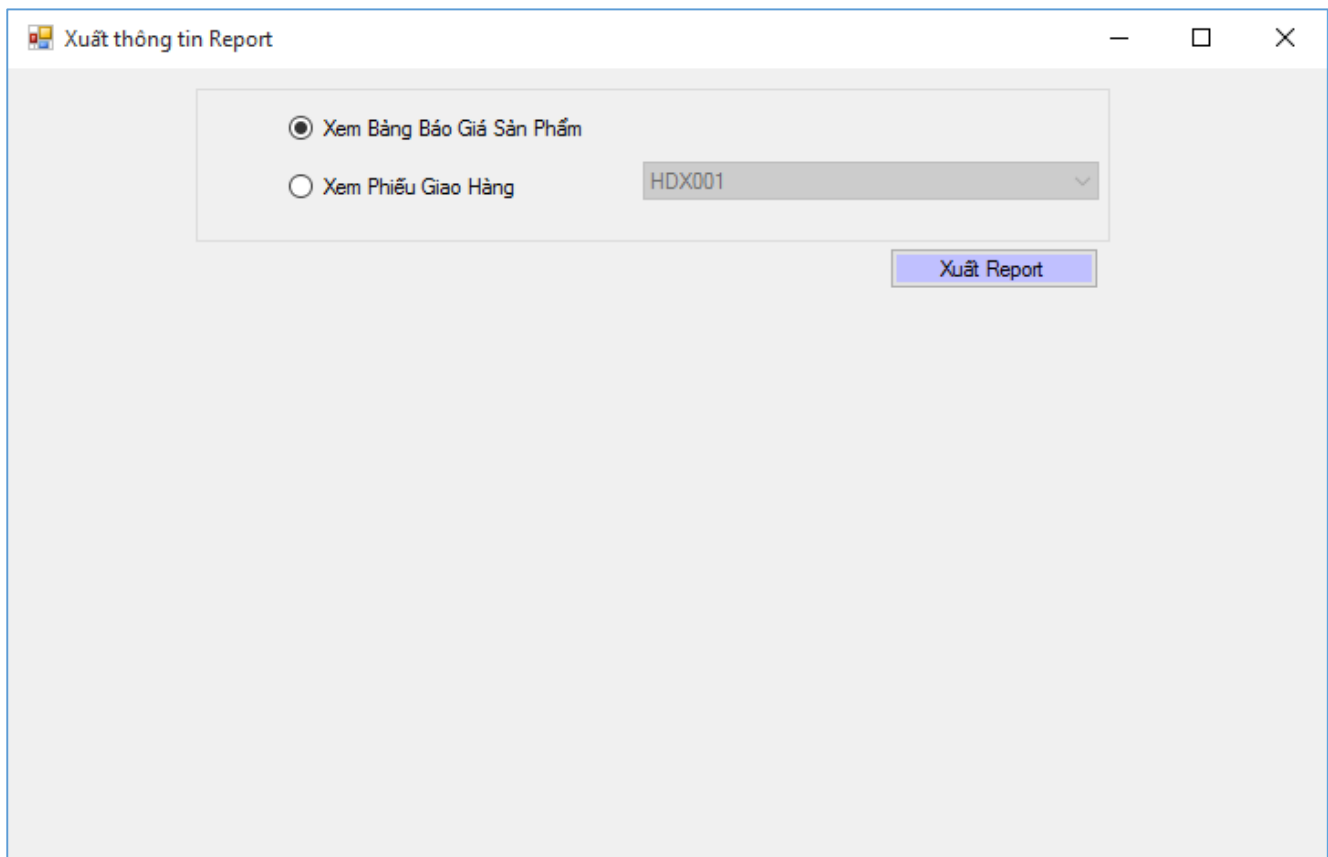
**Product:** Lưu trữ thông tin sản phẩm (**Mã sản phẩm**, Tên Sản phẩm, Đơn vị Tính, Giá Mua, Giá Bán)

**Order:** Lưu trữ chi tiết thông tin đơn hàng (Số HĐ, Số TT, Mã SP, Tên SP, ĐVT, Đơn giá, Số lượng)

**Invoice:** Lưu trữ thông tin hóa đơn đặt hàng (Số HĐ, Ngày đặt hàng, ngày giao hàng, ghi chú)

### Viết chương trình để xem thông tin 2 loại Data Report như sau

- Xem bảng báo giá sản phẩm: Lấy thông tin từ bảng Product và hiện thị report data
- Xem phiếu giao hàng của 1 hóa đơn: Kết hợp Invoice và Order



#### ✓ Khi Load Form

- ReportViewer ở phía dưới được ẩn đi (Visible = false)
- RadioButton Xem Bảng báo giá sản phẩm được chọn mặc định (Checked = True) sau khi mở Form (Disable ComboBox hóa đơn ở phần xem phiếu giao hàng)
- Khi chọn RadioButton Xem Phiếu Giao Hàng thì lấy tất cả các hóa đơn có ngày giao hàng được sắp giảm dần (Invoice) từ bảng Invoice lên ComboBox (Có DropDownStyle = DropDownList)



✓ **Khi Người dùng click vào Button “Xuất Report” (Trong TH chọn xem bảng báo giá)**

- Report Viewer được hiện thị (Visible = True)
- Hiện thị Bảng báo giá sản phẩm ở ReportViewer phía dưới và lấy tất cả thông tin từ bảng Product trong CSDL
- Report Bảng báo giá được hiện thị tương tự như hình dưới

**NHÀ PHÂN PHỐI SẢN PHẨM ABC**  
Số:475B Điện Biên Phủ, P.25, Q.Bình Thạnh, TP.HCM  
Email: [EMAIL@gmail.com](mailto:EMAIL@gmail.com)  
ĐT: 09XXXXXX

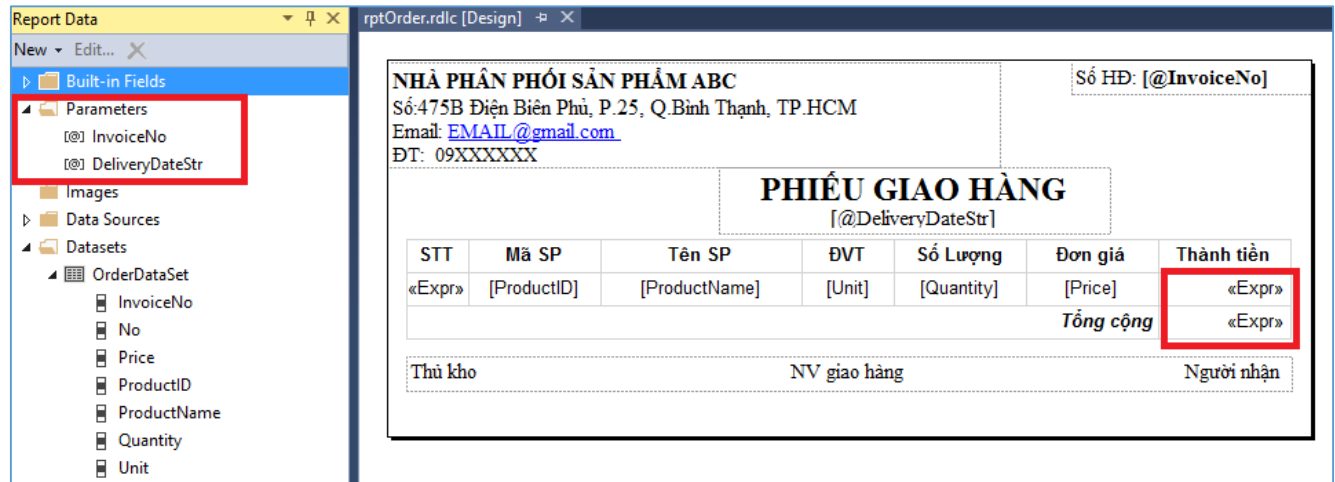
**BẢNG BÁO GIÁ**

STT	Mã Sản Phẩm	Tên Sản Phẩm	ĐVT	Giá Bán
1	Product1	Sản phẩm 1	Cái	120,000
2	Product2	Sản phẩm 2	Cái	120,000
3	Product3	Sản phẩm 3	Cái	70,000
4	Product4	Sản phẩm 4	Hộp	300,000

✓ **Khi Người dùng click vào Button “Xuất Report” ( Trong TH chọn xem Phiếu giao hàng)**

- Report Viewer được hiện thị (Visible = True)
- Hiện thị thông tin đơn hàng từ bảng **Order** mà có mã hóa đơn InvoiceNo = Mã hóa đơn được chọn ở ComboBox (DropDownList)
- Report Phiếu giao hàng được hiện thị tương tự như hình dưới:





**Tạo parameters:** invoiceNo, DeliveryDateStr: để truyền giá trị vào Report Data

**Thành tiền:** Sử dụng Expression = Số lượng \* Đơn giá

**Tổng cộng:** Sử dụng Expression = Sum( Số lượng \* Đơn giá)

#### - Viết code cho sự kiện click Report

```
ProductOrderModel context = new ProductOrderModel();
if (optBangBaoGia.Checked == true) //TH báo giá
{
    List<Product> listProduct = context.Products.ToList();
    this.reportViewer1.LocalReport.ReportPath = "rptProduct.rdlc"; //nhớ copy report debug
    var reportDataSource = new ReportDataSource("ProductDataSet", listProduct);
    this.reportViewer1.LocalReport.DataSources.Clear();
    this.reportViewer1.LocalReport.DataSources.Add(reportDataSource);
    this.reportViewer1.LocalReport.DisplayName = "Bảng Báo Giá";
}
else //TH chi tiết đơn hàng
{
    Invoice invoice = context.Invoices.FirstOrDefault(p => p.InvoiceNo == cmbInvoice.Text);
    List<Order> listOrder = context.Orders.Where(p => p.InvoiceNo == cmbInvoice.Text).ToList();
    if (invoice == null || listOrder.Count() == 0)
    {
        MessageBox.Show("Không tìm thấy đơn hàng");
        return;
    }
    //Sử dụng 2 tham số truyền vào cho report
    ReportParameter[] param = new ReportParameter[2];
    param[0] = new ReportParameter("InvoiceNo", invoice.InvoiceNo);
    param[1] = new ReportParameter("DeliveryDateStr", string.Format("Ngày " +
invoice.DeliveryDate.ToString("dd/MM/yyyy")));
    this.reportViewer1.LocalReport.ReportPath = "rptOrder.rdlc"; //nhớ copy report->debug
    this.reportViewer1.LocalReport.SetParameters(param);
    var reportDataSource = new ReportDataSource("OrderDataSet", listOrder);
    this.reportViewer1.LocalReport.DataSources.Clear();
    this.reportViewer1.LocalReport.DataSources.Add(reportDataSource);
    this.reportViewer1.LocalReport.DisplayName = "Phiếu giao hàng"; //tên hiển thị
}
this.reportViewer1.RefreshReport();
```

**Bài tập 2:** Sử dụng lại cơ sở dữ liệu ProductOrder ở bài tập 1, Sinh Viên thêm dữ liệu giúp cho việc kiểm tra thông tin trên Data Report.

**Viết chương trình xem báo cáo thống kê đơn hàng theo thời gian như sau**

STT	Ngày Giao Hàng	Số HĐ	Thành Tiền
1	02/10/2019	HDX001	5,880,000
2	02/10/2019	HDX002	6,600,000
3	04/10/2019	HDX003	12,000,000
Tổng cộng			24,480,000

Người sử dụng có thể chọn 1 trong 3 loại thời gian để xem thông tin

- Khi chọn xem theo ngày, thì các khoảng thời gian của tháng và theo khoảng được Disable
- Khi chọn xem theo tháng, thì cho phép chọn thời gian MM/yyyy (Custom) và Disable thời gian của xem theo ngày, xem theo khoảng thời gian
- Khi chọn xem theo 1 khoảng thời gian, thì các lựa chọn thời gian của xem theo ngày, xem theo tháng được Disable

Sử dụng một Data Report duy nhất để thực hiện, Thành tiền được tính từ công thức Thành tiền = Số lượng \* Đơn Giá (Price \* Quantity). Số lượng và đơn giá được lấy từ bảng Order trong CSDL.

Xem báo cáo thống kê đơn hàng

Thông Tin Đơn Hàng

Thời gian giao hàng

☒ Xem theo ngày
 02/10/2019

☐ Xem theo tháng
 10/2019

☐ Xem Từ Ngày
 06/10/2019 ~ 06/10/2019

In / Xem

1 of 1

100%

Find | Next

NHÀ PHÂN PHỐI SẢN PHẨM ABC

Số:475B Điện Biên Phủ, P.25, Q.Bình Thạnh, TP.HCM

Email: [EMAIL@gmail.com](mailto:EMAIL@gmail.com)

ĐT: 09XXXXXXX

THỐNG KÊ HÓA ĐƠN

Ngày 2 Tháng 10 Năm 2019

STT	Ngày Giao Hàng	Số HĐ	Thành Tiền
1	02/10/2019	HDX001	5,880,000
2	02/10/2019	HDX002	6,600,000
Tổng cộng			12,480,000

Trong trường hợp không có dữ liệu trong thời gian lựa chọn, Thông báo cho người dùng biết không có hóa đơn nào được tìm thấy.

# BÀI 6: ÔN TẬP VÀ KIỂM TRA

Sử dụng MS SQLServer tạo CSDL tên **QLSach** với 2 bảng sau:

LoaiSach ( **MaLoai** INT, TenLoai nvarchar(50) )

Sach( **MaSach** char(6) , TenSach nvarchar(150), NamXB INT, MaLoai INT)

DESKTOP-AA3CGL7...h - dbo.LoaiSach			
Column Name	Data Type	Allow Nulls	
MaLoai	int	<input type="checkbox"/>	
TenLoai	nvarchar(50)	<input type="checkbox"/>	

DESKTOP-AA3CGL7.QLSach - dbo.Sach			
Column Name	Data Type	Allow Nulls	
MaSach	char(6)	<input type="checkbox"/>	
TenSach	nvarchar(200)	<input type="checkbox"/>	
NamXB	int	<input type="checkbox"/>	
MaLoai	int	<input type="checkbox"/>	

Mã Loại: FK tới LoaiSach.MaLoai

2.1 Thiết kế CSDL và Nhập 1 số dữ liệu vào sẵn như sau (1đ)

DESKTOP-AA3CGL7...h - dbo.LoaiSach		
MaLoai	TenLoai	
1	Khoa Học	
2	Đời sống	
3	Y học	

DESKTOP-AA3CGL7.QLSach - dbo.Sach				
MaSach	TenSach	NamXB	MaLoai	
KH0001	Khám phá sự sống	2018	1	
KH0002	Hải dương học	2018	1	
YH0001	Chẩn đoán và điều trị	2020	3	

2.2 Thiết kế giao diện quản lý sách tương tự như sau (1đ)

Quản Lý Sách

Thống kê

Quản Lý Thông Tin Sách

Thông tin sách

Mã sách

Tên Sách

Năm XB

Thể loại

Khoa Học

Thêm

Sửa

Xóa

Mã Sách	Tên Sách	Năm XB	Thể loại
KH0001	Khám phá sự sống	2018	Khoa Học
KH0002	Hải dương học	2018	Khoa Học
YH0001	Chẩn đoán và điều trị	2020	Y học
YH0002	Dược Học Cổ Truyền	2019	Y học

### 2.3 Khi Load Form (1.5 đ)

- Hiển thị các giá trị Thể loại sách vào Combobox (lấy dữ liệu ở bảng LoaiSach và hiển thị TenLoai) (1đ)

- Hiển thị các sách tìm thấy bên phải (Lấy dữ liệu từ bảng Sách và Loại Sách) (0.5đ)

### 2.4 Chọn 1 dòng ở DataGridView, hiển thị lại thông tin sách phía bên trái (1đ)

### 2.5 Khi Click vào nút xóa (1đ)

- Nếu mã sách muốn xóa đã tồn tại trong CSDL, Hiển thị cảnh báo YES/NO “**Bạn có muốn xóa không?**” (0.25đ). Nhấn YES: Xóa dữ liệu sách đã chọn (0.25đ) và cập nhật lại DataGridView (0.25đ)

- Ngược lại: Thông báo “**Sách cần xóa không tồn tại!**” (0.25)

### 2.6 Khi Click vào nút thêm mới / sửa (3đ)

- Kiểm tra tất cả thông tin bắt buộc phải nhập cho sách. Nếu không xuất hiện thông báo “**Vui lòng nhập đầy đủ thông tin sách!**” (0.5đ)

- Kiểm tra số kí tự mã sách vừa nhập phải là 6. Nếu không xuất thông báo “**Mã sách phải có 6 kí tự!**”. (0.5đ)

- Thêm mới / Cập nhật dữ liệu nhập vào CSDL (1đ).

- Xuất thông báo “**thêm mới/ cập nhật thành công**” và cập nhật lại DataGridView (0.5đ)

- Reset các thông tin nhập liệu sách. Giá trị mặc định như lúc Load Form ban đầu (0.5đ)

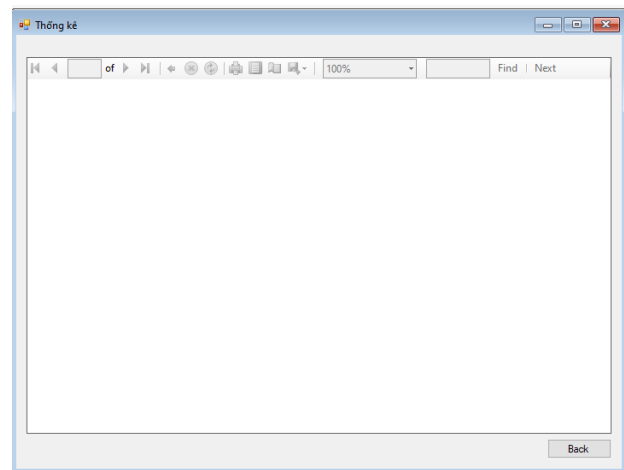
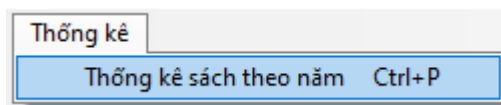
– các control nhập liệu bằng Empty, Combobox thể hiện thông tin các loại sách)

### 2.7 Tìm kiếm theo mã sách, tên sách hoặc năm xuất bản nếu có chứa chuỗi tìm kiếm vừa nhập liệu (0.5 đ)

	Mã Sách	Tên Sách	Năm XB	Thể loại
▶	KH0002	Hải dương học	2018	Khoa Học
	YH0002	Dược Học Cổ Truyền	2019	Y học

## 2.8 Tạo menuStrip thống kê có submenu là “Thống kê sách theo năm” (1đ)

- Tạo thêm form mới có chứa ReportViewer để chứa báo cáo thống kê tương tự sau.



- Khi người dùng click vào menu xem báo cáo thống kê hoặc Ctrl + P, sẽ xuất danh sách thông tin danh sách các sách được *sắp xếp giảm dần* năm XB trên ReportViewer.



Thống kê sách theo năm

Năm XB	Mã Sách	Tên Sách	Mã Loại
2020	YH0001	Chuẩn đoán và điều trị	3
2019	YH0002	Dược Học Cổ Truyền	3
2018	KH0001	Khám phá sự sống	1
2018	KH0002	Hải dương học	1

Điểm cộng (0.5đ): khi thể hiện được **Tên Loại** (thay vì mã loại) trong Report.

-----HẾT-----