

BÀI 5: TỔ CHỨC DỰ ÁN VỚI ENTITY FRAMEWORK

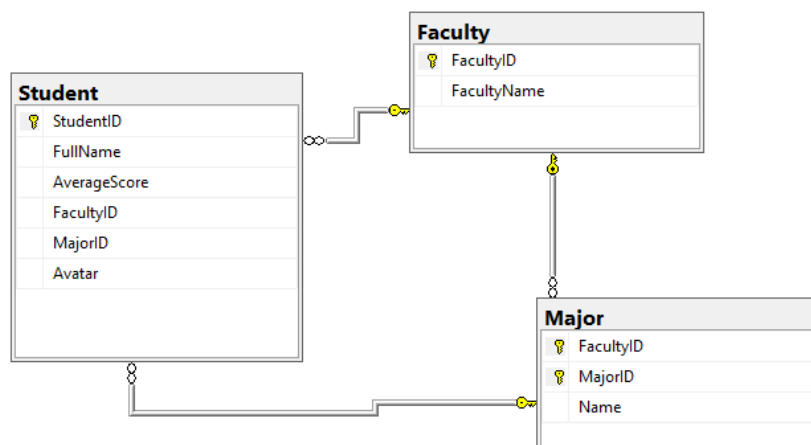
5.1 MỤC TIÊU

- Thiết kế ứng dụng với mô hình 3-layer.
- Kết hợp mô hình 3-layer và entity framework
- Truyền dữ liệu giữa các Form
- SV tự tìm hiểu cách triển khai các mẫu: Repository, Unit of work để phát triển các dự án tốt hơn.

5.2 BÀI TẬP

Thực hiện quản lý sinh viên và quản lý đăng ký chuyên ngành

Cho CSDL như Sau:



HUYCUONG\SQLEXP...05 - dbo.Student		
Column Name	Data Type	Allow Nulls
StudentID	nvarchar(10)	<input type="checkbox"/>
FullName	nvarchar(255)	<input type="checkbox"/>
AverageScore	float	<input type="checkbox"/>
FacultyID	int	<input checked="" type="checkbox"/>
MajorID	int	<input checked="" type="checkbox"/>
Avatar	nvarchar(255)	<input checked="" type="checkbox"/>

HUYCUONG\SQLEXP...05 - dbo.Faculty		
Column Name	Data Type	Allow Nulls
FacultyID	int	<input type="checkbox"/>
FacultyName	nvarchar(255)	<input type="checkbox"/>
		<input type="checkbox"/>

HUYCUONG\SQLEXP...ab05 - dbo.Major		
Column Name	Data Type	Allow Nulls
FacultyID	int	<input type="checkbox"/>
MajorID	int	<input type="checkbox"/>
Name	nvarchar(255)	<input type="checkbox"/>
		<input type="checkbox"/>

Thêm vào ban đầu 1 số dữ liệu cho bảng Faculty, Major, Student

HUYCUONG\SQLEXP...05 - dbo.Faculty	
FacultyID	FacultyName
1	Cong nghe thong tin
2	Ngon Ngu Anh
3	Quan Tri Kinh Doanh
NULL	NULL

HUYCUONG\SQLEXP...ab05 - dbo.Major		
FacultyID	MajorID	Name
1	1	Công nghệ phần mềm
2	1	Tiếng Anh Thương Mại
1	2	Hệ thống thông tin
2	2	Tiếng Anh Truyền Thông
1	3	An toàn thông tin
NULL	NULL	NULL

1. Giao diện quản lý sinh viên

Quản Lý Sinh Viên

Chức Năng

Thông Tin Sinh Viên

Ma Sinh Viên

1234567890

Họ Tên

Nguyen Van B

Khoa

Cong nghe thong tin

Diem Trung Binh

7.5

Ảnh đại diện

Add / Update

Delete

Quản lý sinh viên

☐ Chưa ĐK chuyển ngành

MSSV	Họ Tên	Khoa	DTB	Chuyên ngành
1234567890	Nguyen Van B	Cong nghe thong tin	7.5	Công nghệ phần mềm
1234567891	Nguyen Van A	Cong nghe thong tin	4.5	Hệ thống thông tin
2345643213	Nguyen Van C	Ngon Ngu Anh	10	

Các chức năng:

- Check / Uncheck CheckBox "Chưa đăng ký chuyên ngành":
 - ✓ *Checked = False (mặc định)*: Hiển thị tất cả sinh viên, kể ra sinh viên chưa có chuyên ngành và đã có chuyên ngành
 - ✓ *Checked = True*: Hiển thị các sinh viên chưa có chuyên ngành (MajorID = NULL)
- Load Ảnh Avatar / Show avatar
 - ✓ Tạo 1 folder "**Images**" để lưu hình ảnh, trong CSDL "Avatar" chỉ lưu tên hình
 - ✓ Khi thêm mới, nếu có hình ảnh sẽ được lưu vào thư mục Images với tên là {studentID}.{typeFile}.

Ví dụ 1: với mã sv là 1234567890 thì nếu người dùng upload hình file .jpg sẽ được lưu **Images/1234567890.jpg**

Ví dụ 2: với mã sv là 1234567891 thì nếu người dùng upload hình file .png sẽ được lưu **Images/ 1234567891.png**
- Các chức năng Add/Update và Xóa tương tự bài 4.

2. Giao diện đăng ký chuyên ngành

The screenshot shows a Windows application window titled 'frmRegister'. The main heading is 'Đăng ký chuyên ngành' in red. Below it are two dropdown menus: 'Khoa' with 'Ngon Ngu Anh' selected, and 'Chuyên ngành' with 'Tiếng Anh Thương Mại' selected. Below these is a table with the following data:

	Chọn	MSSV	Họ Tên	Khoa	DTB
✎	✓	2345643213	Nguyen Van C	Ngon Ngu Anh	10

Below the table is a large grey rectangular area. At the bottom of the window is a 'Register' button.

Khi chọn các DropDownList Khoa sẽ lấy các chuyên ngành tương ứng và hiển thị tất cả các sinh viên của Khoa mà chưa có chuyên ngành.

Có thể check chọn các Sinh viên ở DataGridView, khi click vào "register" thì sẽ chọn chuyên ngành cho sinh viên (được checked)

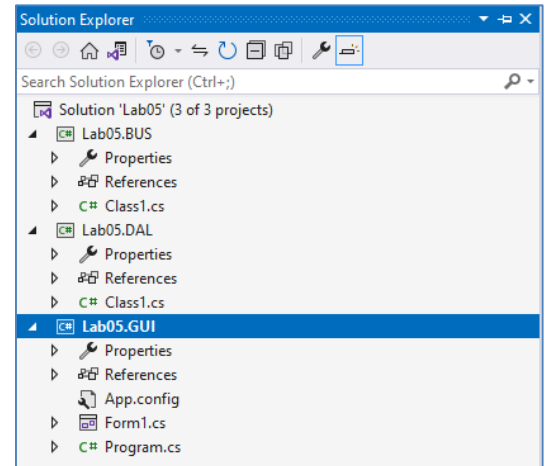
HƯỚNG DẪN:

Bước 1: Tạo dự án theo cấu trúc:

Lab05.GUI: là project Windows Form App (.NET Framework)

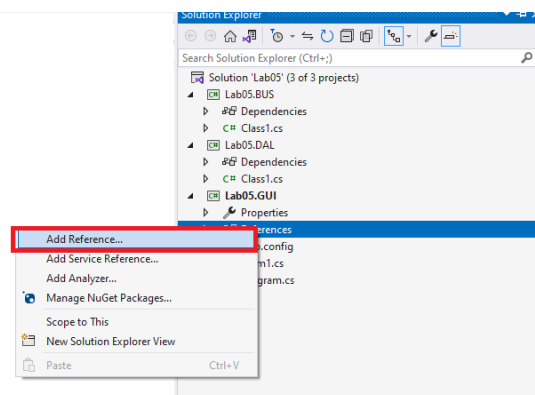
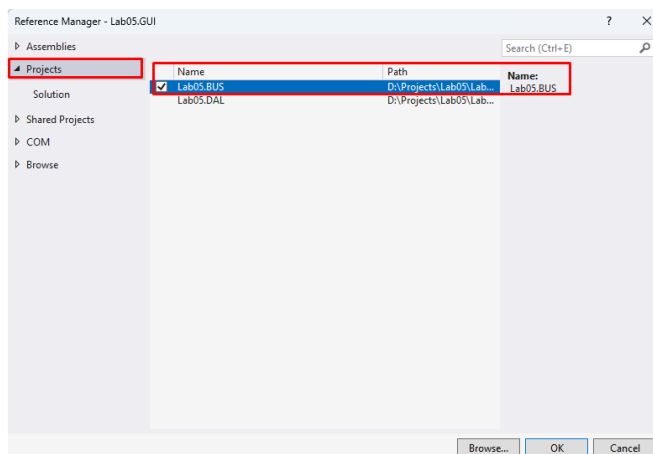
Lab05.BUS: là project loại Class Library (.NET Framework)

Lab05.DAL: là project loại Class Library (.NET Framework)



Cần liên kết các project bằng cách sử dụng tính năng References => Add reference để liên kết

✓ Lab05.GUI liên kết tới Lab05.BUS



Ở project đơn giản này, ta có thể sử dụng DTO là các Entities ở tầng DAL nên liên kết GUI tới DAL (Hoặc có thể tạo thêm tầng DTO để làm rõ mô hình này hơn và có các kiểu đối tượng DTO như: StudentDTO, FacultyDTO, MajorDTO)

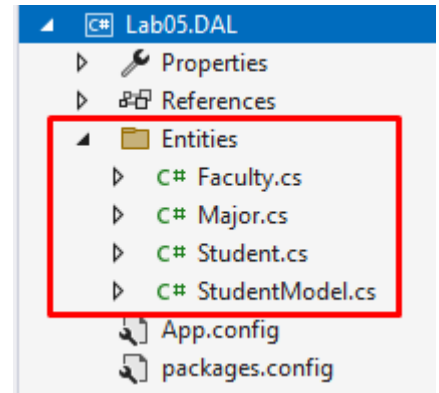
✓ Lab05.BUS liên kết tới Lab05.DAL (tự thực hiện)

Bước 2: Thực hiện ở tầng DAL (kết nối CSDL)

Xem lại cách kết nối CSDL sử dụng mô hình

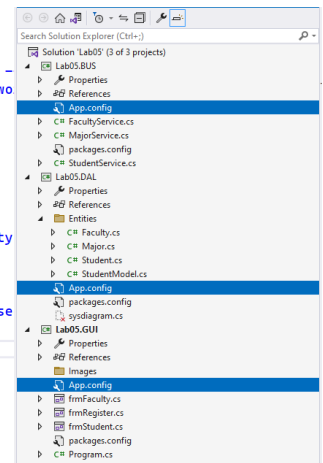
Entity Framework Code First theo hướng tiếp cận đã

Có cơ sở dữ liệu



Chú ý: Thông tin kết nối CSDL được lưu ở **App.Config** trong tầng DAL. Tuy nhiên khi gọi từ các tầng khác chúng ta cũng cần phải có file **App.Config** tương ứng

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework" />
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
  <connectionStrings>
    <add name="StudentModel" connectionString="data source=HUVCUONG\SQLEXPRESS;initial catalog=StudentLab05;persist security info=True;multiple active result sets=True;multiple result sets=True;encrypt=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

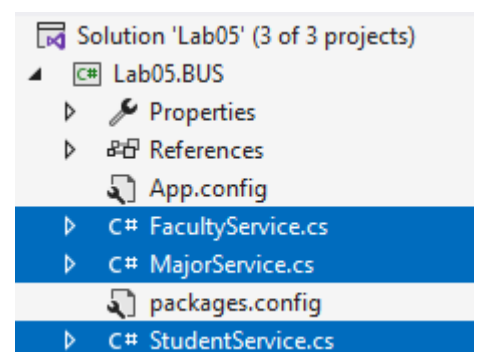


Bước 3: Thực hiện ở tầng BUS

Implement các hàm cần sử dụng

(có thể đặt tên với các ý nghĩa xử lý logic ở tầng BUS này như: StudentDAO, MajorDAO, FacultyDAO...)

Một số hàm thực hiện theo yêu cầu đề bài:



- ✓ **StudentService:** Thực hiện các xử lý logic với Student như: Lấy danh sách sinh viên, Lấy ds sinh viên chưa đăng ký chuyên ngành, Tìm sinh viên theo ID, Thực hiện update cho cả insert/update...

```

public class StudentService
{
    2 references
    public List<Student> GetAll()
    {
        StudentModel context = new StudentModel();
        return context.Students.ToList();
    }
    1 reference
    public List<Student> GetAllHasNoMajor()
    {
        StudentModel context = new StudentModel();
        return context.Students.Where(p=>p.MajorID == null).ToList();
    }
    1 reference
    public List<Student> GetAllHasNoMajor(int facultyID)
    {
        StudentModel context = new StudentModel();
        return context.Students.Where(p => p.MajorID == null && p.FacultyID == facultyID).ToList();
    }
    1 reference
    public Student FindById(string studentId)
    {
        StudentModel context = new StudentModel();
        return context.Students.FirstOrDefault(p => p.StudentID == studentId);
    }
    1 reference
    public void InsertUpdate(Student s)
    {
        StudentModel context = new StudentModel();
        context.Students.AddOrUpdate(s);
        context.SaveChanges();
    }
}

```

- ✓ **MajorService:** Thực hiện các xử lý logic về chuyên ngành như: Lấy danh sách chuyên ngành từ CSDL,...

```

namespace Lab05.BUS
{
    2 references
    public class MajorService
    {
        1 reference
        public List<Major> GetAllByFaculty(int facultyID)
        {
            StudentModel context = new StudentModel();
            return context.Majors.Where(p=>p.FacultyID == facultyID).ToList();
        }
    }
}

```

- ✓ **FacultyService:** Thực hiện các xử lý logic về Khoa như: Lấy danh sách Khoa từ CSDL, thêm, xóa, sửa...

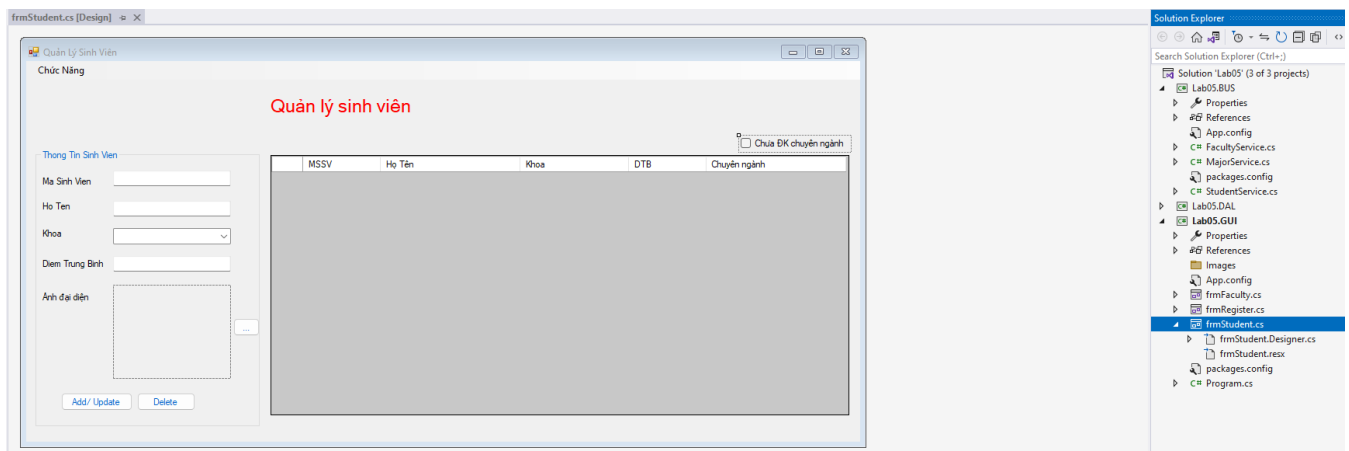
```

namespace Lab05.BUS
{
    4 references
    public class FacultyService
    {
        2 references
        public List<Faculty> GetAll()
        {
            StudentModel context = new StudentModel();
            return context.Faculties.ToList();
        }
    }
}

```

Bước 4: Thực hiện ở GUI và điều chỉnh cần thiết

4.1 Quản lý sinh viên



- ✓ Load Form: Lấy ds sinh viên từ CSDL để fill vào DataGridView, các thông tin Khoa để đưa vào Combobox

```

public partial class frmStudent : Form
{
    private readonly StudentService studentService = new StudentService();
    private readonly FacultyService facultyService = new FacultyService();
    public frmStudent()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        try
        {
            setGridViewStyle(dgvStudent);
            var listFaculty = facultyService.GetAll();
            var listStudents = studentService.GetAll();
            FillFacultyCombobox(listFaculty);
            BindGrid(listStudents);
        }
        catch (Exception ex)
        {
        }
    }
}

```

```

        MessageBox.Show(ex.Message);
    }
}
//Hàm binding list dữ liệu khoa vào combobox có tên hiển thị là tên khoa,
giá trị là Mã khoa
private void FillFacultyCombobox(List<Faculty> listFacultys)
{
    listFacultys.Insert(0, new Faculty());
    this.cmbFaculty.DataSource = listFacultys;
    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;
        if(item.Faculty != null)
            dgvStudent.Rows[index].Cells[2].Value =
item.Faculty.FacultyName;
        dgvStudent.Rows[index].Cells[3].Value = item.AverageScore + "";

        if (item.MajorID != null)
            dgvStudent.Rows[index].Cells[4].Value = item.Major.Name + "";
        ShowAvatar(item.Avatar);
    }
}
private void ShowAvatar(string ImageName)
{
    if (string.IsNullOrEmpty(ImageName))
    {
        picAvatar.Image = null;
    }
    else
    {
        string parentDirectory =
Directory.GetParent(AppDomain.CurrentDomain.BaseDirectory).Parent.Parent.FullName
;
        string imagePath = Path.Combine(parentDirectory, "Images",
ImageName);
        picAvatar.Image = Image.FromFile(imagePath);
        picAvatar.Refresh();
    }
}

public void setGridViewStyle(DataGridView dgvview)
{
    dgvview.BorderStyle = BorderStyle.None;
    dgvview.DefaultCellStyle.SelectionBackColor = Color.DarkTurquoise;
    dgvview.CellBorderStyle =
DataGridViewCellBorderStyle.SingleHorizontal;
    dgvview.BackgroundColor = Color.White;
    dgvview.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
}

```

- ✓ Đăng ký sự kiện **chkUnregisterMajor_CheckedChanged** ở checkbox cho việc lọc các sinh viên chưa đăng ký chuyên ngành


```
private void chkUnregisterMajor_CheckedChanged(object sender, EventArgs e)
{
    var listStudents = new List<Student>();
    if (this.chkUnregisterMajor.Checked)
        listStudents = studentService.GetAllHasNoMajor();
    else
        listStudents = studentService.GetAll();
    BindGrid(listStudents);
}
```

4.2 Đăng ký chuyên ngành

Sau khi chọn khoa

- Hiện thị ds sinh viên của Khoa mà chưa có chuyên ngành
(MajorId = NULL trong Student)
- Hiện thị chuyên ngành trong Khoa (mục đích để đăng ký)
- Việc đăng ký sinh viên khi checked vào danh sách sinh viên ở DataGridView

The screenshot shows a Windows form titled 'frmRegister' with the main heading 'Đăng ký chuyên ngành' in red. There are two dropdown menus: 'Khoa' and 'Chuyên ngành'. Below these is a DataGridView with the following columns: 'Chọn', 'MSSV', 'Họ Tên', 'Khoa', and 'DTB'. The grid area is currently empty. At the bottom of the form is a 'Register' button.

Hướng dẫn:

- ✓ Form Load: Đổ dữ liệu vào Khoa

```
public partial class frmRegister : Form
{
    private readonly StudentService studentService = new StudentService();
    private readonly FacultyService facultyService = new FacultyService();
    private readonly MajorService majorService = new MajorService();

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

```

private void frmRegister_Load(object sender, EventArgs e)
{
    try
    {
        var listFacultys = facultyService.GetAll();
        FillFacultyCombobox(listFacultys);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
//Hàm binding list dữ liệu khoa vào combobox có tên hiển thị là tên khoa, giá
trị là Mã khoa
private void FillFacultyCombobox(List<Faculty> listFacultys)
{
    this.cmbFaculty.DataSource = listFacultys;
    this.cmbFaculty.DisplayMember = "FacultyName";
    this.cmbFaculty.ValueMember = "FacultyID";
}
}

```

✓ cmbFaculty_SelectedIndexChanged: Khi thay đổi index lựa chọn cho Khoa

```

private void cmbFaculty_SelectedIndexChanged(object sender, EventArgs e)
{
    Faculty selectedFaculty = cmbFaculty.SelectedItem as Faculty;
    if(selectedFaculty != null)
    {
        var listMajor = majorService.GetAllByFaculty(selectedFaculty.FacultyID);
        FillMajorCombobox(listMajor);
        var listStudents = studentService.GetAllHasNoMajor(selectedFaculty.FacultyID);
        BindGrid(listStudents);
    }
}
1 reference
private void BindGrid(List<Student> listStudent)
{
    dgvStudent.Rows.Clear();
    foreach (var item in listStudent)
    {
        int index = dgvStudent.Rows.Add();
        dgvStudent.Rows[index].Cells[1].Value = item.StudentID;
        dgvStudent.Rows[index].Cells[2].Value = item.FullName;
        if (item.Faculty != null)
        {
            dgvStudent.Rows[index].Cells[3].Value = item.Faculty.FacultyName;
            dgvStudent.Rows[index].Cells[4].Value = item.AverageScore + "";
            if (item.MajorID != null)
            {
                dgvStudent.Rows[index].Cells[5].Value = item.Major.Name + "";
            }
        }
    }
}

```

Còn lại SV tự thực hiện để có thể đăng ký được chuyên ngành có các sinh viên được chọn (checked) ở DataGridView.