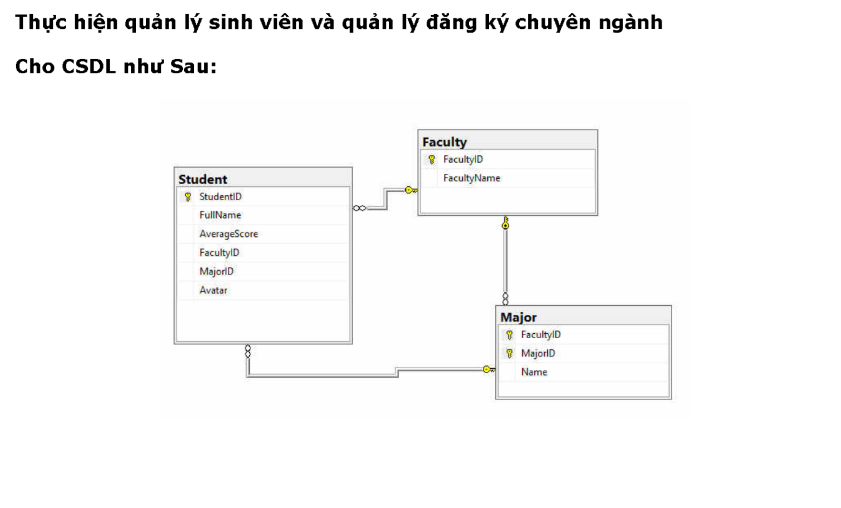
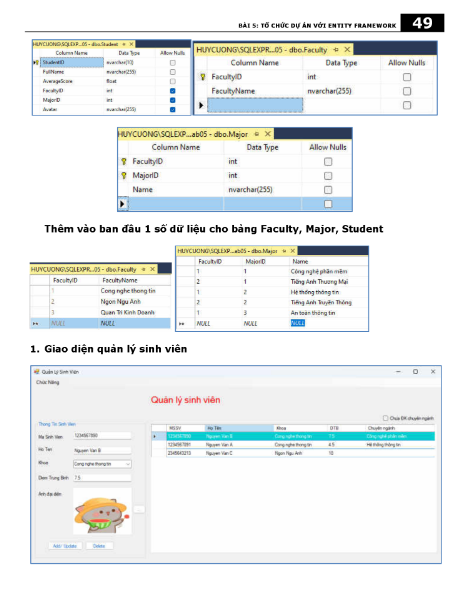
buổi 7 - TỔ CHỨC DỰ ÁN VỚI ENTITY FRAMEWORK

Sinh viên thực hiện : Trịnh Tuấn Đan

MSSV: 2180602115 Lớp: 21DTHA4





* **BUS.csproj**
  + **FacultyServices.cs**

|  |
| --- |
| * using DAL.Entities; * using System; * using System.Collections.Generic; * using System.Linq; * using System.Text; * using System.Threading.Tasks; * namespace BUS * { * public class FacultyServices * { * public List<Faculty> GetAll() * { * StudentModel context = new StudentModel(); * return context.Faculties.ToList(); * } * } * } |

* + **StudentServices.cs**

|  |
| --- |
| * using DAL.Entities; * using System; * using System.Collections.Generic; * using System.Linq; * using System.Text; * using System.Threading.Tasks; * namespace BUS * { * public class StudentServices * { * public List<Student> GetAll() * { * StudentModel context = new StudentModel(); * return context.Students.ToList(); * } * public List<Student> GetAllHasNoMajor() * { * StudentModel context = new StudentModel(); * return context.Students.Where(p=>p.MajorID == null).ToList(); * } * public List<Student> GetAllHasNoMajor(int facultyID) * { * StudentModel context = new StudentModel(); * return context.Students.Where(p => p.MajorID == null && p.FacultyID == facultyID).ToList(); * } * } * } |

* **Code Form Quản Lý Nhân Viên**

|  |
| --- |
| * using BUS; * using DAL.Entities; * using System; * using System.Collections.Generic; * using System.ComponentModel; * using System.Data; * using System.Data.Entity; * using System.Drawing; * using System.IO; * using System.Linq; * using System.Runtime.Remoting.Contexts; * using System.Text; * using System.Threading.Tasks; * using System.Windows.Forms; * namespace GUI * { * public partial class frmStudent : Form * { * private readonly StudentServices studentService = new StudentServices(); * private readonly FacultyServices facultyService = new FacultyServices(); * public frmStudent() * { * InitializeComponent(); * } * //Phương thức BindGrid được sử dụng để hiển thị danh sách sinh viên lên DataGridView * public 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 + ""; * dgvStudent.Rows[index].Cells[5].Value = item.Avatar; * } * } * // FillFalcultyCombobox được sử dụng để đổ danh sách khoa vào combobox cmdKhoa. * private void FillFalcultyCombobox(List<Faculty> listFacultys) * { * listFacultys.Insert(0, new Faculty()); * this.cmbFaculty.DataSource = listFacultys; * this.cmbFaculty.DisplayMember = "FacultyName"; * this.cmbFaculty.ValueMember = "FacultyID"; * } * private void Form1\_Load(object sender, EventArgs e) * { * setGridViewStyle(dgvStudent); * var listFacultys = facultyService.GetAll(); * var listStudent = studentService.GetAll(); * FillFalcultyCombobox(listFacultys); * BindGrid(listStudent); * } * 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 dgview) * { * dgview.BorderStyle = BorderStyle.None; * dgview.DefaultCellStyle.SelectionBackColor = Color.DarkTurquoise; * dgview.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal; * dgview.BackgroundColor = Color.White; * dgview.SelectionMode = DataGridViewSelectionMode.FullRowSelect; * } * 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); * } * //click DataGridView * private void dgvStudent\_CellClick(object sender, DataGridViewCellEventArgs e) * { * try * { * if (dgvStudent.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null) * { * dgvStudent.CurrentRow.Selected = true; * txtStudentID.Text = dgvStudent.Rows[e.RowIndex].Cells[0].FormattedValue.ToString(); * txtFullName.Text = dgvStudent.Rows[e.RowIndex].Cells[1].FormattedValue.ToString(); * cmbFaculty.SelectedIndex = cmbFaculty.FindString(dgvStudent.Rows[e.RowIndex].Cells[2].FormattedValue.ToString()); * txtAverageScore.Text = dgvStudent.Rows[e.RowIndex].Cells[3].FormattedValue.ToString(); * ShowAvatar(dgvStudent.Rows[e.RowIndex].Cells[5].FormattedValue.ToString()); * } * } * catch (Exception ex) * { * MessageBox.Show(ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error); * } * } * //=============================================================================================================================================== * //Load lại thông tin trên dataGridview * private void reloadDGV() * { * var listStudents = studentService.GetAll(); * BindGrid(listStudents); * } * // Refresh các ô nhập dữ liệu * private void clean() * { * txtAverageScore.Text = ""; * txtStudentID.Text = ""; * txtFullName.Text = ""; * cmbFaculty.SelectedIndex = 0; * } * // Thêm Sinh Viên * private void themSV() * { * StudentModel context = new StudentModel(); * Student s = new Student() * { * StudentID = txtStudentID.Text, * FullName = txtFullName.Text, * AverageScore = float.Parse(txtAverageScore.Text), * FacultyID = (int)cmbFaculty.SelectedValue * }; * context.Students.Add(s); * context.SaveChanges(); * reloadDGV(); * clean(); * MessageBox.Show("Thêm dữ liệu thành công!", "Thông báo", MessageBoxButtons.OK); * } * //Thay Đổi thông tin Sinh Viên * private void Sua() * { * using (StudentModel context = new StudentModel()) * { * Student sua = context.Students.FirstOrDefault(p => p.StudentID == txtStudentID.Text); * if (sua != null) * { * sua.FullName = txtFullName.Text; * sua.FacultyID = (int)cmbFaculty.SelectedValue; * sua.AverageScore = float.Parse(txtAverageScore.Text); * context.Entry(sua).State = EntityState.Modified; // Đánh dấu đối tượng để cập nhật. * context.SaveChanges(); * List<Student> list = context.Students.ToList(); * BindGrid(list); * MessageBox.Show("Cập nhật dữ liệu thành công!", "Thông báo", MessageBoxButtons.OK); * clean(); * } * else * { * MessageBox.Show("Không tìm thấy MSSV cần Sửa", "Thông báo", MessageBoxButtons.OK); * } * } * } * private void btnAddUpdate\_Click(object sender, EventArgs e) * { * try * { * if (txtStudentID.Text == "" || txtFullName.Text == "" || txtAverageScore.Text == "") * { * throw new Exception("Vui lòng nhập đầy đủ thông tin"); * } * else if (txtStudentID.Text.Length != 10) * { * MessageBox.Show("MSSV phải có 10 kí tự!", "Thông báo", MessageBoxButtons.OK); * } * else * { * using (StudentModel context = new StudentModel()) * { * // Kiểm tra xem sinh viên đã tồn tại trong cơ sở dữ liệu hay chưa * Student existingStudent = context.Students.FirstOrDefault(s => s.StudentID == txtStudentID.Text); * if (existingStudent == null) * { * // Sinh viên chưa tồn tại, vì vậy hãy thêm một sinh viên mới * themSV(); * } * else * { * // Sinh viên đã tồn tại, vì vậy hãy cập nhật thông tin của họ * Sua(); * } * } * } * } * catch (Exception ex) * { * MessageBox.Show(ex.Message); * } * } * //=============================================================================================================================================== * private void xoaSV() * { * StudentModel context = new StudentModel(); * Student xoa = context.Students.FirstOrDefault(p => p.StudentID == txtStudentID.Text); * if (xoa != null) * { * context.Students.Remove(xoa); * context.SaveChanges(); * List<Student> list = context.Students.ToList(); * BindGrid(list); * MessageBox.Show("Xóa SV thành công!", "Thông báo", MessageBoxButtons.OK); * clean(); * } * } * private void btnDelete\_Click(object sender, EventArgs e) * { * DialogResult dl = MessageBox.Show("Xóa SV", "YES/NO", MessageBoxButtons.YesNo); * if (dl == DialogResult.Yes) * { * xoaSV(); * } * else * { * MessageBox.Show("Không tìm thấy MSSV cần xóa", "Thông báo", MessageBoxButtons.OK); * } * } * } * } |

* **Ảnh Kết quả Code**

|  |
| --- |
|  |
|  |
|  |
|  |

 Link Source Code: