

## Mục tiêu

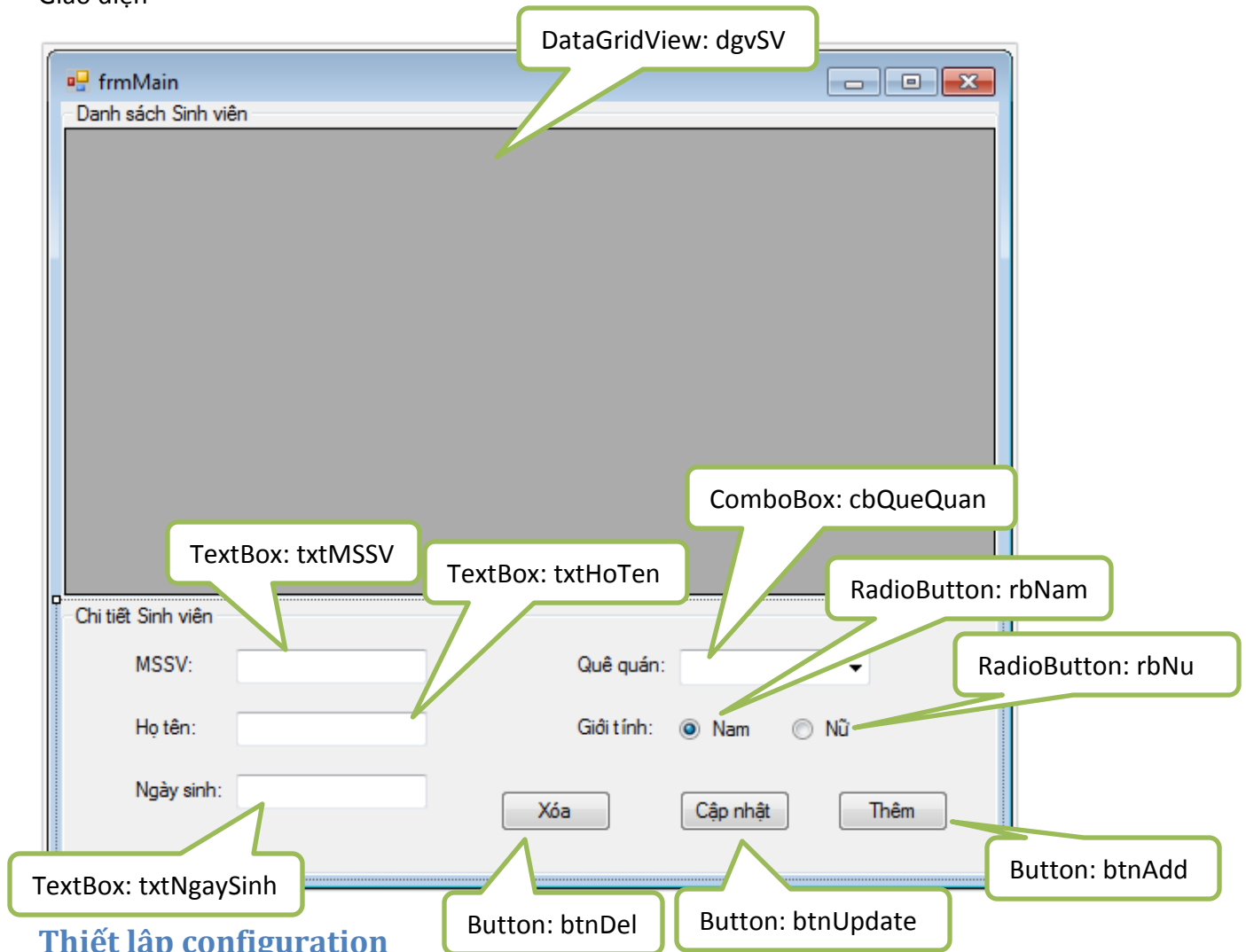
- Thao tác xử lý connect database
- Thao tác với dữ liệu với ADO.NET

## Nội dung

Xây dựng ứng dụng đơn giản với database QLSV

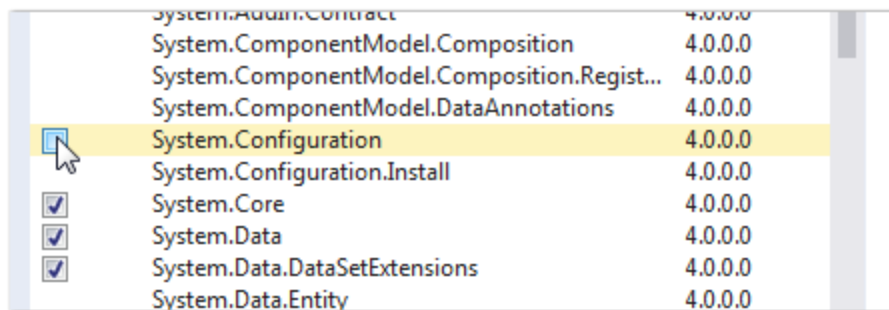
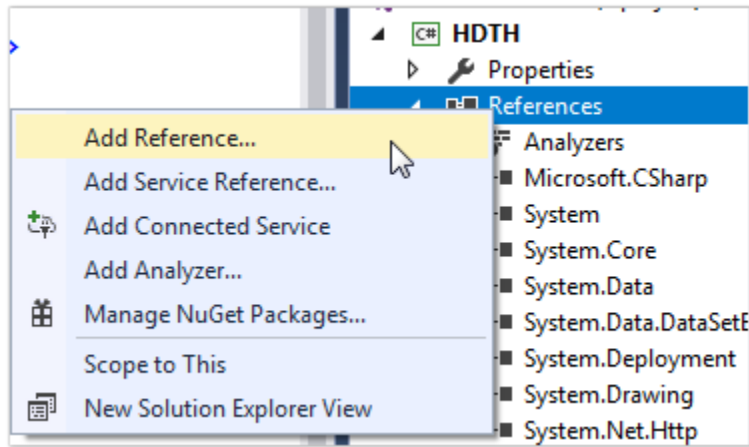
## Tạo project winform

Giao diện



## Thiết lập configuration

Thêm reference



Thêm dữ liệu cho **app.config**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name="sqlPC_QLSV" connectionString="Data Source=.\sqlexpress;Initial
      Catalog=QLSV;Integrated Security=True"/>
  </connectionStrings>
</configuration>
```

## Xử lý kết nối database

✓ class **DataUlti**

```
class DataUlti
{
    // connection dùng chung
    static string cnString;
    static string nameCS = "sqlPC_QLSV";
    static SqlConnection cn;
    static DataUlti()
    {
        // khởi tạo connection từ config
        cnString = ConfigurationManager.ConnectionStrings[nameCS].ConnectionString;
        cn = new SqlConnection(cnString);
    }
}
```

```

// Phương thức tạo Command
static SqlCommand CreateCommand()
{
    // mở connect
    if (cn.State == ConnectionState.Closed)
    {
        cn.Open();
    }
    // tạo command
    var cm = new SqlCommand();
    cm.Connection = cn;
    return cm;
}
// Phương thức lấy danh sách Quê quán
public static List<QueQuan> DSQueQuan()
{
    var lqq = new List<QueQuan>();
    var cm = CreateCommand();
    cm.CommandText = @"SELECT * FROM QueQuan";
    // lấy reader đọc dữ liệu
    var reader = cm.ExecuteReader();
    while (reader.Read())
    {
        lqq.Add(QueQuan.ReadQueQuan(reader));
    }
    // đóng connect
    cn.Close();
    return lqq;
}
// Phương thức lấy danh sách Sinh viên
public static List<SinhVien> DSSinhVien()
{
    var lsv = new List<SinhVien>();
    var cm = CreateCommand();
    cm.CommandText = @"SELECT * FROM SinhVien";
    var reader = cm.ExecuteReader();
    while (reader.Read())
    {
        lsv.Add(SinhVien.ReadSinhVien(reader));
    }
    cn.Close();
    return lsv;
}

```

```

// Phương thức thêm Sinh viên
public static bool ThemSinhVien(SinhVien sv)
{
    var cm = CreateCommand();
    cm.CommandText = @"INSERT INTO SinhVien (MSSV, HoTen, NgaySinh, QueQuanID,
GioiTinh) VALUES (@MSSV, @HoTen, @NgaySinh, @QueQuanID, @GioiTinh)";
    cm.Parameters.Add(new SqlParameter("@MSSV", sv.MSSV));
    cm.Parameters.Add(new SqlParameter("@HoTen", sv.HoTen));
    cm.Parameters.Add(new SqlParameter("@NgaySinh", sv.NgaySinh));
    cm.Parameters.Add(new SqlParameter("@QueQuanID", sv.QueQuanID));
    cm.Parameters.Add(new SqlParameter("@GioiTinh", sv.GioiTinh));
    int rs = cm.ExecuteNonQuery();
    cn.Close();
    // số lượng record thêm thành công
    return rs > 0;
}
// Phương thức sửa Sinh viên

```

```

public static bool CapNhatSinhVien(SinhVien sv)
{
    var cm = CreateCommand();
    cm.CommandText = @"UPDATE SinhVien SET HoTen=@HoTen, NgaySinh=@NgaySinh,
QueQuanID=@QueQuanID, GioiTinh=@GioiTinh WHERE MSSV=@MSSV";
    cm.Parameters.Add(new SqlParameter("@MSSV", sv.MSSV));
    cm.Parameters.Add(new SqlParameter("@HoTen", sv.HoTen));
    cm.Parameters.Add(new SqlParameter("@NgaySinh", sv.NgaySinh));
    cm.Parameters.Add(new SqlParameter("@QueQuanID", sv.QueQuanID));
    cm.Parameters.Add(new SqlParameter("@GioiTinh", sv.GioiTinh));
    int rs = cm.ExecuteNonQuery();
    cn.Close();
    return rs > 0;
}
// Phương thức xóa Sinh viên
public static bool XoaSinhVien(SinhVien sv)
{
    var cm = CreateCommand();
    cm.CommandText = @"DELETE FROM SinhVien WHERE MSSV=@MSSV";
    cm.Parameters.Add(new SqlParameter("@MSSV", sv.MSSV));
    int rs = cm.ExecuteNonQuery();
    cn.Close();
    return rs > 0;
}
}

```

✓ class **SinhVien**

```

public class SinhVien
{
    public string MSSV { get; set; }
    public string HoTen { get; set; }
    public DateTime NgaySinh { get; set; }
    public int QueQuanID { get; set; }
    public bool GioiTinh { get; set; }
    // phương thức đọc dữ liệu sinh viên
    static int iMSSV = 0, iHoTen = 1, iNgaySinh = 2, iQueQuanID = 3, iGioiTinh = 4;
    public static SinhVien ReadSinhVien(SqlDataReader reader)
    {
        var sv = new SinhVien {
            MSSV = reader.GetString(iMSSV),
            HoTen = reader.GetString(iHoTen),
            NgaySinh = reader.GetDateTime(iNgaySinh),
            QueQuanID = reader.GetInt32(iQueQuanID),
            GioiTinh = reader.GetBoolean(iGioiTinh)
        };
        return sv;
    }
    // phương thức tạo mssv thích hợp
    public static string CreateMSSV()
    {
        var ds = DataUlti.DSSinhVien();
        // lấy mssv cao nhất
        ds.Sort((sv1, sv2) => -sv1.MSSV.CompareTo(sv2.MSSV));
        long ms = long.Parse(ds[0].MSSV);
        ms++;
        return ms.ToString();
    }
}

```

✓ class **QueQuan**

```

public class QueQuan
{
    public int ID { get; set; }
    public string TenDiaPhuong { get; set; }

    static int iID = 0, iTenDiaPhuong = 1;
    public static QueQuan ReadQueQuan(SqlDataReader reader)
    {
        var qq = new QueQuan {
            ID = reader.GetInt32(iID),
            TenDiaPhuong = reader.GetString(iTenDiaPhuong)
        };
        return qq;
    }
}

```

## Xử lý form

- ✓ Tạo cột thích hợp cho DataGridView

```

// phương thức tạo cột cho datagridview
void CreateColumn()
{
    // xóa cột trước
    dgvSV.Columns.Clear();
    // cột MSSV
    var dgvctb = new DataGridViewTextBoxColumn();
    dgvctb.HeaderText = "MSSV";
    dgvSV.Columns.Add(dgvctb);
    // cột Họ tên
    dgvctb = new DataGridViewTextBoxColumn();
    dgvctb.HeaderText = "Họ tên";
    dgvSV.Columns.Add(dgvctb);
    // cột Ngày sinh
    dgvctb = new DataGridViewTextBoxColumn();
    dgvctb.HeaderText = "Ngày sinh";
    dgvSV.Columns.Add(dgvctb);
    // cột Quê quán
    var dgvCb = new DataGridViewComboBoxColumn();
    dgvCb.HeaderText = "Quê quán";
    dgvSV.Columns.Add(dgvCb);
    var dgvcchk = new DataGridViewCheckBoxColumn();
    dgvcchk.HeaderText = "Giới tính Nam";
    dgvSV.Columns.Add(dgvcchk);
}

```

- ✓ Load dữ liệu cho DataGridView

```

void LoadDGV()
{
    // xóa dòng trước
    dgvSV.Rows.Clear();
    // lấy dữ liệu cho combobox
    var lqq = DataUlti.DSQueQuan();
    DataGridViewRow row = null;
}

```

```

DataGridViewCell cell = null;

var ds = DataUlti.DSSinhVien();
foreach (var sv in ds)
{
    row = new DataGridViewRow();
    // giữ lại dữ liệu gốc
    row.Tag = sv;
    cell = new DataGridViewTextBoxCell();
    cell.Value = sv.MSSV;
    row.Cells.Add(cell);

    cell = new DataGridViewTextBoxCell();
    cell.Value = sv.HoTen;
    row.Cells.Add(cell);

    cell = new DataGridViewTextBoxCell();
    cell.Value = sv.NgaySinh.ToShortDateString();
    row.Cells.Add(cell);

    cell = new DataGridViewComboBoxCell();
    cell.Value = sv.QueQuanID;
    // gán datasource
    ((DataGridViewComboBoxCell)cell).DataSource = lqq;
    // gán property tương ứng
    ((DataGridViewComboBoxCell)cell).DisplayMember = "TenDiaPhuong";
    ((DataGridViewComboBoxCell)cell).ValueMember = "ID";
    row.Cells.Add(cell);

    cell = new DataGridViewCheckBoxCell();
    cell.Value = sv.GioiTinh;
    row.Cells.Add(cell);

    dgvSV.Rows.Add(row);
}
}

```

✓ Khởi tạo khi form load

```

// phương thức khởi tạo (gọi trong event form load)
void Init()
{
    // không cho phép thêm row
    dgvSV.AllowUserToAddRows = false;
    // không hiển thị row header
    dgvSV.RowHeadersVisible = false;
    // lựa chọn theo row
    dgvSV.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    // chỉ cho phép chọn 1
    dgvSV.MultiSelect = false;
    // lấy dữ liệu cho combobox
    var lqq = DataUlti.DSQueQuan();
    cbQueQuan.DataSource = lqq;
    // gán property thích hợp
    cbQueQuan.DisplayMember = "TenDiaPhuong";
    cbQueQuan.ValueMember = "ID";
    // không cho phép thay đổi MSSV
    txtMSSV.Enabled = false;

    CreateColumn();
    LoadDGV();
}

```

- ✓ Gán dữ liệu cho các control input

```
void FillSVDetail(SinhVien sv)
{
    txtMSSV.Text = sv.MSSV;
    txtHoTen.Text = sv.HoTen;
    txtNgaySinh.Text = sv.NgaySinh.ToShortDateString();
    cbQueQuan.SelectedValue = sv.QueQuanID;
    rbNam.Checked = sv.GioiTinh;
}
```

- ✓ Xử lý select trên DataGridView

```
// gắn kết event khi lựa chọn trên datagridview
dgvSV.SelectionChanged += DgvSV_SelectionChanged;
```

```
private void DgvSV_SelectionChanged(object sender, EventArgs e)
{
    if (dgvSV.SelectedRows.Count > 0)
    {
        // lấy lại dữ liệu được giữ
        var sv = dgvSV.SelectedRows[0].Tag as SinhVien;
        FillSVDetail(sv);
    }
}
```

- ✓ Xử lý RadioButton giới tính

```
// gắn kết event xử lý radiobutton
rbNam.CheckedChanged += RbNam_CheckedChanged;
```

```
private void RbNam_CheckedChanged(object sender, EventArgs e)
{
    rbNu.Checked = !rbNam.Checked;
}
```

- ✓ Xử lý thêm, xóa, cập nhật

```
// gắn kết event xử lý button
btnUpdate.Click += BtnUpdate_Click;
btnDel.Click += BtnDel_Click;
btnAdd.Click += BtnAdd_Click;
```

```
private void BtnAdd_Click(object sender, EventArgs e)
{
    // tạo đối tượng sinh viên
    var sv = new SinhVien {
        MSSV = SinhVien.CreateMSSV(),
        HoTen = txtHoTen.Text,
        NgaySinh = DateTime.Parse(txtNgaySinh.Text),
        QueQuanID = (int)cbQueQuan.SelectedValue,
        GioiTinh = rbNam.Checked
    };
    // thông báo kết quả
    if (DataUlti.ThemSinhVien(sv))
```

```

    {
        MessageBox.Show("Đã thêm thành công");
        // load lại dữ liệu
        LoadDGV();
    }
    else
    {
        MessageBox.Show("Không thể thêm");
    }
}

private void BtnDel_Click(object sender, EventArgs e)
{
    // lấy sinh viên đang chọn
    if (dgvSV.SelectedRows.Count > 0)
    {
        var sv = dgvSV.SelectedRows[0].Tag as SinhVien;
        // confirm việc xóa
        if (MessageBox.Show(string.Format("Xác nhận xóa sinh viên {0}", sv.MSSV),
            "Xác nhận xóa", MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            // thông báo kết quả
            if(DataUlti.XoaSinhVien(sv))
            {
                MessageBox.Show("Đã xóa thành công");
                // load lại dữ liệu
                LoadDGV();
            }
            else
            {
                MessageBox.Show("Không thể xóa");
            }
        }
    }
}

```

```

private void BtnUpdate_Click(object sender, EventArgs e)
{
    // lấy sinh viên đang chọn
    if (dgvSV.SelectedRows.Count > 0)
    {
        var sv = dgvSV.SelectedRows[0].Tag as SinhVien;
        // lấy dữ liệu mới
        sv.HoTen = txtHoTen.Text;
        sv.QueQuanID = (int)cbQueQuan.SelectedValue;
        sv.GioiTinh = rbNam.Checked;
        DateTime ns;
        if(DateTime.TryParse(txtNgaySinh.Text, out ns))
        {
            sv.NgaySinh = ns;
        }
        if (DataUlti.CapNhatSinhVien(sv))
        {
            MessageBox.Show("Đã cập nhật thành công");
            // load lại dữ liệu
            LoadDGV();
        }
        else
        {
            MessageBox.Show("Không thể cập nhật");
        }
    }
}

```



## Bài tập

Hoàn thành ví dụ trên.