Hướng dẫn thiết kế giao diện phần mềm có nhiều chức năng

Mỗi chức năng nên để trên một giao diện riêng.

Ví dụ bài quản lý sinh viên có các chức năng:

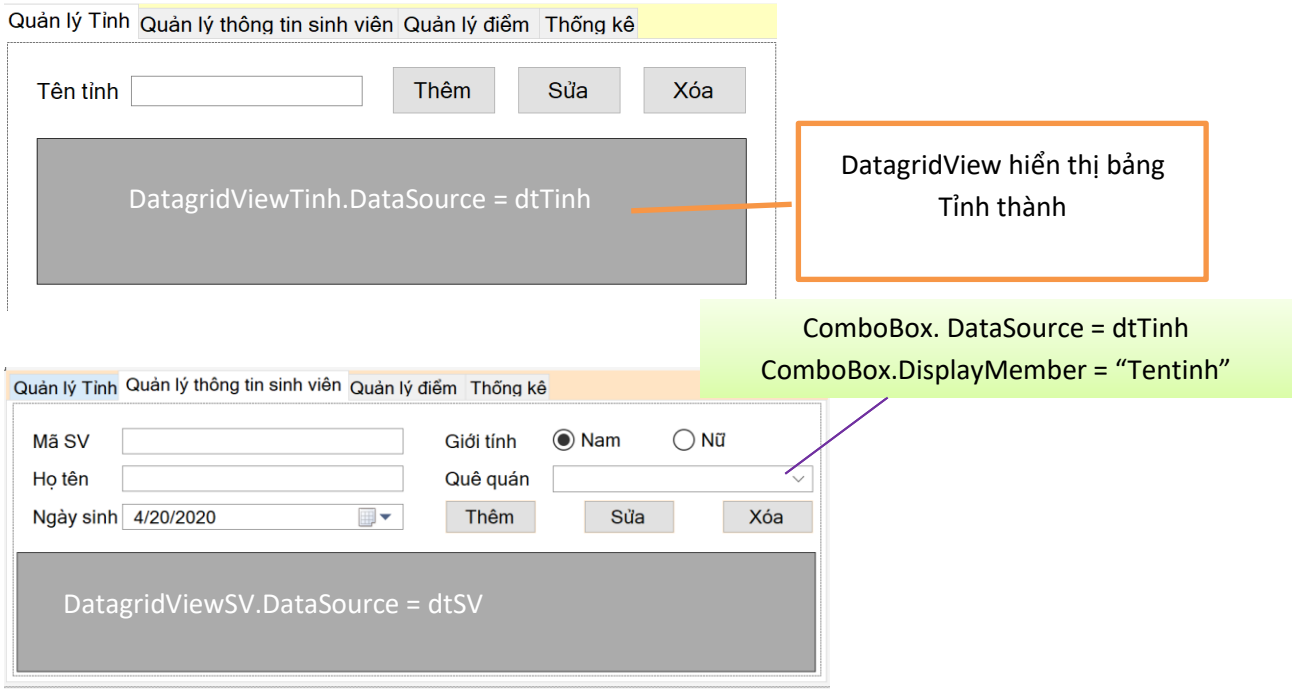
- Thêm, sửa, xóa tỉnh thành
- Thêm sửa xóa thông tin sinh viên
- Cập nhật điểm cho sinh viên
- Thống kê sinh viên theo một tiêu chí nào đó

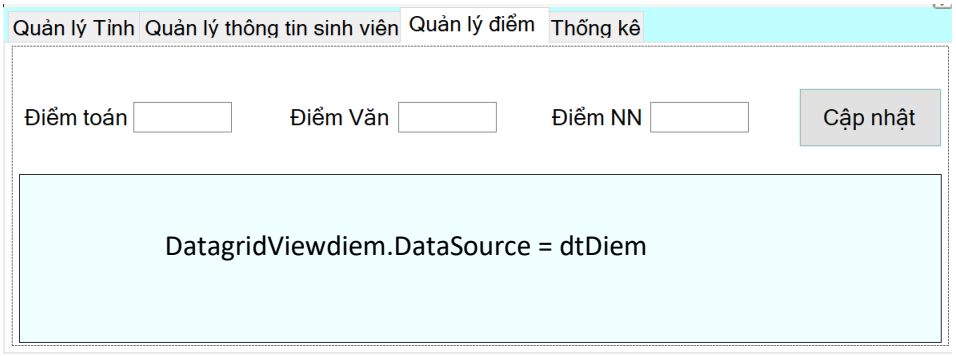
Như vậy, không thể để tất cả các chức năng trên vào một giao diện.

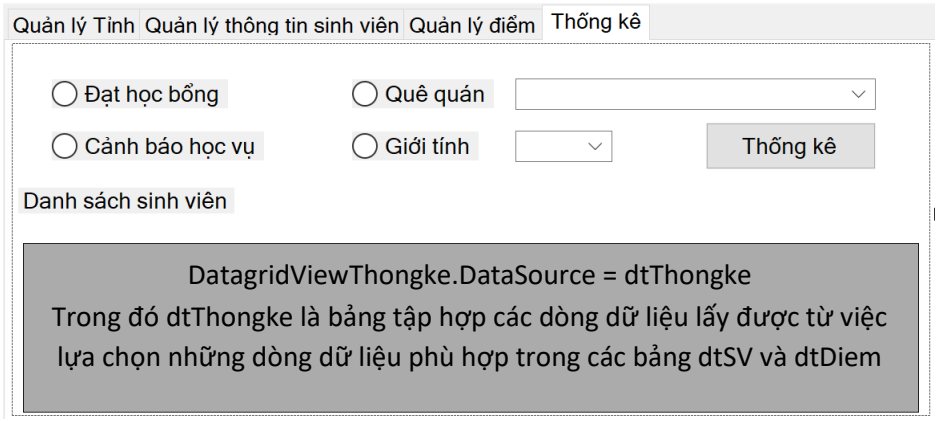
Để thiết kế mỗi chức năng trên một giao diện khác nhau có thể sử dụng một trong các cách sau:

Cách 1. Sử dụng TabControl, thiết kế mỗi chức năng trên một TabPage của TabControl

VD:







Cách 2. Sử dụng Panel, thiết kế mỗi chức năng trên một Panel. Các Panel được thiết kế chồng lên nhau. Khi gọi đến chức năng nào thì Panel tương ứng với chức năng đó được gọi BringToFront() để hiển thị lên trên cùng.

VD khi gọi vào chức năng quản lý điểm thì panel điểm được gọi lên trên cùng

```
private void mnQLDiem_Click(object sender, EventArgs e)
{
    pnDiem.BringToFront();
}
```

Cách 3. Sử dụng các form phụ, mỗi chức năng được thiết kế trên một form phụ. Khi gọi đến chức năng nào thì gọi form phụ đó ShowDialog() lên.

VD khi gọi chức năng quản lý thông tin sinh viên thì hiển thị form quản lý thông tin sinh viên lên.

```
private void mnQLSV_Click(object sender, EventArgs e)
{
    //tạo đối tượng của form Quản lý TT SV
    FormQLTTSV fQLSV = new FormQLTTSV();
    //đưa dữ liệu của bảng SV lên form
    fQLSV.dtsv = dtSV;
    //Hiển thị form lên để làm việc
    fQLSV.ShowDialog();
}
```

Cách 4. Sử dụng các UserControl, cái này giống như panel, nhưng được phân thành các file riêng biệt. Khi sử dụng đến chức năng nào thì gọi UserControl đó hiển thị lên form.

VD: khi gọi chức năng quản lý sinh viên thì tạo đối tượng của usercontrol QLSV rồi hiển thị nó lên trên cùng

```
//ucSinhVien là một UserControl thiết kế chức năng Quản lý thông tin SV
ucSinhVien ucSV = new ucSinhVien();
private void mnQLSV_Click(object sender, EventArgs e)
{
    if (!pnMain.Contains(ucSV))
    {
        ucSV.dtsv = dtSV;
        ucSV.dttinh = dtTinh;
        ucSV.dtdiem = dtDiem;
        this.Controls.Add(ucSV);
        ucSV.Dock = DockStyle.Fill;
    }
    ucSV.BringToFront();
}
```

Với cách 1 và cách 2, việc viết code sẽ trở nên rối khi có quá nhiều code trong cùng 1 file (vì tất cả đang cùng nằm trên form chính). Trong khi cách 3 và cách 4 sẽ tạo ra các file riêng biệt nên việc viết code trên mỗi file sẽ ít hơn, dễ theo dõi hơn.

Tuy nhiên, cách nào cũng cần chú ý việc đọc/ghi dữ liệu từ file json. Nếu lưu dữ liệu không đầy đủ thì khi đọc ra sẽ không giống cấu trúc bảng hoặc lớp đã tạo. Do vậy khi lấy dữ liệu đọc được từ file cần chú ý kiểm tra; nếu một cấu trúc nào đó chưa có dữ liệu thì không dùng cấu trúc đã đọc được mà phải dùng cấu trúc mới tạo.

Ví dụ: đoạn mã sau dùng để tạo cấu trúc bảng trước khi lấy dữ liệu

```
//Tạo bảng Tinh
dtTinh.Columns.Add("ID");
dtTinh.Columns.Add("Tentinh");
//Tạo bảng sinh viên
dtSV.Columns.Add("MaSV");
dtSV.Columns.Add("TenSV");
dtSV.Columns.Add("NgaySinh");
dtSV.Columns.Add("Gioitinh");
dtSV.Columns.Add("Quequan");
//Tạo bảng điểm
dtDiem.Columns.Add("MaSV");
dtDiem.Columns.Add("TenSV");
dtDiem.Columns.Add("DiemToan", System.Type.GetType("System.Double"));
dtDiem.Columns.Add("DiemVan", System.Type.GetType("System.Double"));
dtDiem.Columns.Add("DiemNN", System.Type.GetType("System.Double"));
dtDiem.Columns.Add("DiemTB", System.Type.GetType("System.Double"));
```

Đoạn mã sau dùng để lấy dữ liệu vào các bảng (nếu dữ liệu đã có)

```
if (System.IO.File.Exists("dulieu.json"))
{
    //nếu đã có thì sử dụng streamreader để đọc dl từ file ra chuỗi jsonstr
    System.IO.StreamReader reader = new System.IO.StreamReader("dulieu.json");
    string jsonstr = reader.ReadToEnd();
    reader.Close();//đọc xong đóng file để khỏi lỗi khi lưu lại
    //chuyển đổi dữ liệu từ chuỗi jsonstr thành đối tượng tương ứng lúc lưu
    dsDulieu = JsonConvert.DeserializeObject<DataSet>(jsonstr);
    //kiểm tra dữ liệu bảng Tỉnh có ko? Nếu có thì chuyển vào bảng dtTinh
    //nếu không thì bảng dtTinh lấy cấu trúc từ lúc tạo
    if (dsDulieu.Tables["Tinh"].Rows.Count > 0)
        dtTinh = dsDulieu.Tables["Tinh"];
    //kiểm tra dữ liệu bảng SV có ko? Nếu có thì chuyển vào bảng dtSV
    if (dsDulieu.Tables["Sinhvien"].Rows.Count > 0)
        dtSV = dsDulieu.Tables["Sinhvien"];
    //kiểm tra dữ liệu bảng Điểm có ko? Nếu có thì chuyển vào bảng dtDiem
    if (dsDulieu.Tables["Diem"].Rows.Count > 0)
        dtDiem = dsDulieu.Tables["Diem"];
}
```

Đoạn mã sau dùng để lưu tất cả các bảng dữ liệu vào 1 file JSON

```
//sử dụng DataSet dsDulieu để gom tất cả các bảng vào một tập hợp
dsDulieu.Tables.Clear();
dsDulieu.Tables.Add(dtTinh);
dsDulieu.Tables.Add(dtSV);
dsDulieu.Tables.Add(dtDiem);
string jsonstr;
//chuyển đổi tập hợp các bảng trong DataSet thành chuỗi JSON
jsonstr = JsonConvert.SerializeObject(dsDulieu);
//lưu chuỗi JSON vào file
System.IO.File.WriteAllText("dulieu.json", jsonstr);
```