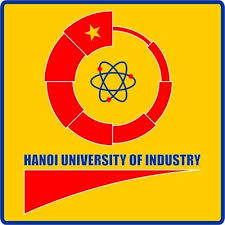
**BỘ CÔNG NGHIỆP**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**Báo cáo bài tập lớn**

**LẬP TRÌNH WINDOWS**

**ĐỀ TÀI: Phần mềm quản lý người tìm việc – việc tìm người**

**Giáo Viên:** Thầy Phạm Văn Hà

**Nhóm -** **Lớp:** Nhóm 16 - KTPM3\_K11

**Thành Viên:** Chu Tuấn Anh 1141360180

Nguyễn Văn Công 1141360237

Đào Hữu Dương 1141360220

**Hà Nội, 2018**

**Mục Lục**

[***Mở Đầu*** 4](#_Toc532594243)

[**Chương 1: Khảo sát bài toán và phân tích yêu cầu** 5](#_Toc532594244)

[**1.1** **Khảo sát yêu cầu:** 5](#_Toc532594245)

[**1.2. Phân Tích Yêu Cầu:** 6](#_Toc532594246)

[**1.2.1.Giao Diện:** 6](#_Toc532594247)

[**1.2.2. Dữ Liệu:** 6](#_Toc532594248)

[**Chương 2: Thiết kế hệ thống (Thiết kế cấu trúc, giao diện, CSDL)** 8](#_Toc532594249)

[**2.1. Thiết Kế Cấu Trúc** 8](#_Toc532594250)

[**2.2. Thiết Kế Giao Diện** 9](#_Toc532594251)

[**2.2.1.Giao diện Quản Lý Người Tìm Việc** 9](#_Toc532594252)

[**2.2.2. Giao diện Quản Lý Tuyển Dụng** 15](#_Toc532594253)

[**2.2.3. Giao diện Tìm Kiếm** 19](#_Toc532594254)

[**2.2.4. Thêm Xóa Tài Khoản (Chỉ hiển thị khi quyền truy cập là ADMIN)** 22](#_Toc532594255)

[**2.3. Thiết Kế Cơ Sở Dữ Liệu** 22](#_Toc532594256)

[**Chương 3: Chương trình và các kết quả đạt được** 24](#_Toc532594257)

[**3.1.Quản Lý Người Tìm Việc** 32](#_Toc532594258)

[**3.1.1. Graphics User Interface** 32](#_Toc532594259)

[3.1.1.1.Thêm người tìm việc 32](#_Toc532594260)

[3.1.1.2. Duyệt Tin Mới 33](#_Toc532594261)

[3.1.1.3.Quản Lý Tin Đã Duyệt 34](#_Toc532594262)

[**3.1.2. Business Logic Layer** 35](#_Toc532594263)

[**3.1.3.Data Access Layer** 36](#_Toc532594264)

[**3.2.Quản Lý Tuyển Dụng** 36](#_Toc532594265)

[**3.2.1. Graphics User Interface** 36](#_Toc532594266)

[3.2.1.1.Thêm tin tuyển dụng 36](#_Toc532594267)

[3.2.1.2. Duyệt Tin Mới 37](#_Toc532594268)

[3.2.1.3.Quản Lý Tin Đã Duyệt 38](#_Toc532594269)

[**3.2.2. Business Logic Layer** 39](#_Toc532594270)

[**3.2.3.Data Access Layer** 39](#_Toc532594271)

[**3.3.Thêm, Xóa Tài Khoản** 40](#_Toc532594272)

[**3.3.1. Graphics User Interface** 40](#_Toc532594273)

[**3.3.2. Business Logic Layer** 41](#_Toc532594274)

[**3.3.3.Data Access Layer** 41](#_Toc532594275)

[**3.4.Tìm Kiếm** 42](#_Toc532594276)

[**3.4.1. Graphics User Interface** 42](#_Toc532594277)

[**3.4.2. Business Logic Layer** 43](#_Toc532594278)

[**3.4.3.Data Access Layer** 44](#_Toc532594279)

[***Kết Luận*** 45](#_Toc532594280)

# **Mở Đầu**

Sau một thời gian học tập và tìm hiểu bộ môn Lập Trình Windows, được sự chỉ bảo của thầy Phạm Văn Hà, người phụ trách bộ môn Lập Trình Windows. Chúng em đã hoàn thành bài tập lớn với đề tài : “Phần mềm quản lý người tìm việc – việc tìm người”

Bài tập lớn này gồm 3 chương:

Chương 1: Khảo sát bài toán và phân tích yêu cầu

Chương 2: Thiết kế hệ thống (Thiết kế cấu trúc, giao diện, CSDL)

Chương 3: Chương trình và các kết quả đạt được

Chúng em xin bày tỏ lòng biết ơn chân thành tới Thầy Phạm Văn Hà – thầy phụ trách và chỉ bảo, góp ý cho chúng em để hoàn thành bài tập lớn này !

Trong quá trình tìm hiểu, biên soạn, bài tập lớn này khó tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp thêm để có nhiều kinh nghiệm hơn nữa trong cách làm bài tiểu luận, bài tập lớn, bài tốt nghiệp,....

*Trân trọng cảm ơn !*

**Nhóm sinh viên thực hiện**

# **Chương 1: Khảo sát bài toán và phân tích yêu cầu**

* 1. **Khảo sát yêu cầu:**

Bảng câu hỏi khảo sát:

|  |  |
| --- | --- |
| **Câu hỏi** | **Câu trả lời của khách hàng** |
| 1.Anh chị có thể cho chúng tôi biết mục đích anh chị muốn có phần mềm không ạ? | Trung tâm chúng tôi muốn có phần mềm để dễ duyệt, quản lý các hồ sơ của người tìm việc và các công ty tuyển dụng mục đính giảm thiểu các hồ sơ giấy tờ. |
| 2.Anh chị có yêu cầu cơ bản nào về phần mềm quản lý này ko? | Tôi muốn phần mêm phải có các chức năng cơ bản như: Sửa, Xóa và Tìm Kiếm dữ liệu cho cả phần Quản Lý Tìm Việc và Quản Lý Tìm Người. Các tin phải được duyệt bởi nhân viên của chúng tôi. |
| 3.Anh chị muốn có tài khoản để quản lý chứ?? | Vâng, tôi muốn có 2 loại tài khoản với 2 mức, loại thứ 1 gồm các chức năng của loại thứ 2 và có thể thêm hoặc xóa loại tài khoản thứ 2. |
| 4.Anh chị có muốn thêm các hồ sơ thủ công không? Hay chỉ nhận và duyệt các Hồ sơ online được người dùng gửi về | Có, Tôi muốn có chức năng Thêm thủ công nữa. |
| 5. Anh chị muốn Người Tìm Việc gồm những thông tin cụ thể nảo? | Tôi muốn thông tin Người Tìm Việc phải có đủ Họ Tên, Ảnh, Ngành Nghề, Năm Kinh Nghiệm, Ngày Sinh, Giới Tính, Địa chỉ, Trình độ, Mức lương mong muốn, Vị trí mong muốn, Lương, Loại hình công việc và Thông tin liên hệ. |
| 6. Anh chị muốn phía Tuyển Dụng gồm các thông tin cụ thể nào? | Còn phía thông tin Công ty tuyển dụng thì phải có các thông tin Tên công ty, Địa điểm tuyển dụng, Mức lương, Hình thức làm việc, Số lượng tuyển, Yêu cầu giới tính, Ngành nghề, Yêu cầu bằng cấp, Lương, Mô tả công việc, Yêu cầu hồ sơ và Thông tin liên hệ. |
| 7. Anh chị muốn chức năng Tìm Kiếm ra sao thì có thể đáp ứng yêu cầu anh chị? | Chúng tôi làm phần mềm với mục đích là lưu thông tin mà không cần hồ sơ giấy, vậy nên khi có người đến Trung Tâm để tư vấn việc làm hoặc nhà tuyển dụng tới tìm nhân sự thì chúng tôi có thể truy xuất được các thông tin. |
| 8. Vậy chúng tôi sẽ tạo 2 loại tìm kiếm, loại 1 là tìm kiếm đơn giản là nhập ngành nghề rồi tìm kiếm. loại 2 là tìm kiếm nâng cao, sẽ lọc các yêu cầu một cách chi tiết để tìm kiếm hiệu quả hơn. Ý anh chị ra sao? | Vâng, được ạ. |
| 9. Anh chị còn muốn nói hay còn yêu cầu gì nữa không? | Không ạ. |
|  |  |

**1.2. Phân Tích Yêu Cầu:**

**1.2.1.Giao Diện:**

-Có Đăng Nhập(với 2 loại tài khoản(Chủ và nhân viên))

-Quản Lý Người Tìm Việc:

+Thêm Người Tìm Việc(Thủ công)

+Mục duyệt hoặc xóa các tin mới(tin chưa duyệt từ dữ liệu chung gửi về)

+Mục quản lý(Xem,Sửa,Xóa) các tin đã duyệt

-Quản Lý Tuyển Dụng:

+Thêm tin Tuyển Dụng(Thủ Công)

+Mục duyệt hoặc xóa các tin mới(tin chưa duyệt từ dữ liệu chung gửi về)

+Mục quản lý(Xem, Sửa, Xóa) các tin đã duyệt

-Tìm kiếm

+Tìm kiếm gồm: Tìm Việc hay Tìm Người, Tên công việc, Hình thức làm việc, Yêu cầu bằng cấp, Địa điểm làm việc, Yêu cầu giới tính, Mức lương.

-Thêm/Xóa tài khoản(dành cho Chủ)

**1.2.2. Dữ Liệu:**

-Tài Khoản: +Loại 1: có thể Thêm/Xóa tài khoản mới để quản lý, và có thể sử dụng tất cả tính năng của phần mềm.

+Loại 2(Được thêm bởi loại 1): Có thể sử dụng tất cả tính năng của phần mềm, Ngoại trừ việc tạo Thêm/Xóa tài khoản.

-Thông tin về Người Tìm Việc:

+Họ Tên, Ảnh, Ngành Nghề, Năm Kinh Nghiệm, Ngày Sinh, Giới Tính, Địa chỉ, Trình độ, Mức lương mong muốn, Vị trí mong muốn, Lương, Loại hình công việc và Thông tin liên hệ.

-Thông tin về Người Tuyển Dụng:

+Tên công ty, Địa điểm tuyển dụng, Mức lương, Hình thức làm việc, Số lượng tuyển, Vị Trí, Yêu cầu Kinh Nghiệm, Yêu cầu giới tính, Ngành nghề, Yêu cầu bằng cấp, Lương, Mô tả công việc, Yêu cầu hồ sơ và Thông tin liên hệ.

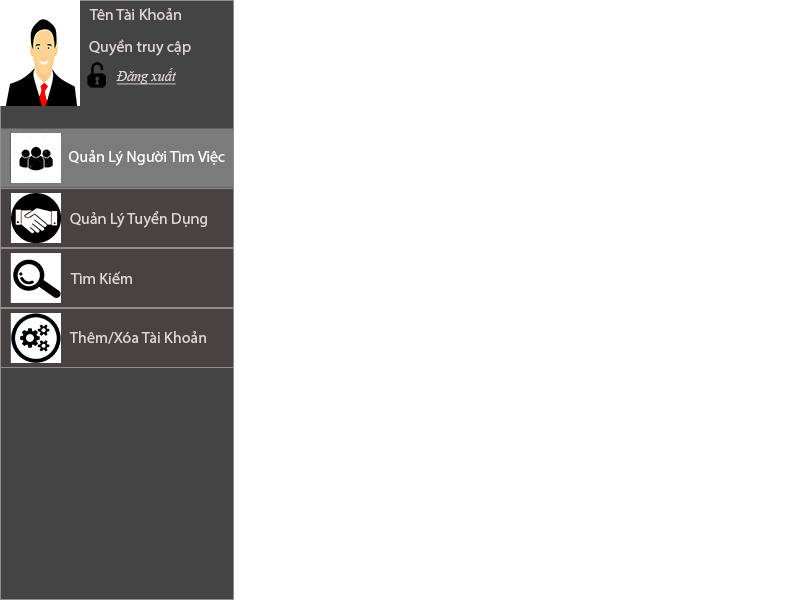
# **Chương 2: Thiết kế hệ thống (Thiết kế cấu trúc, giao diện, CSDL)**

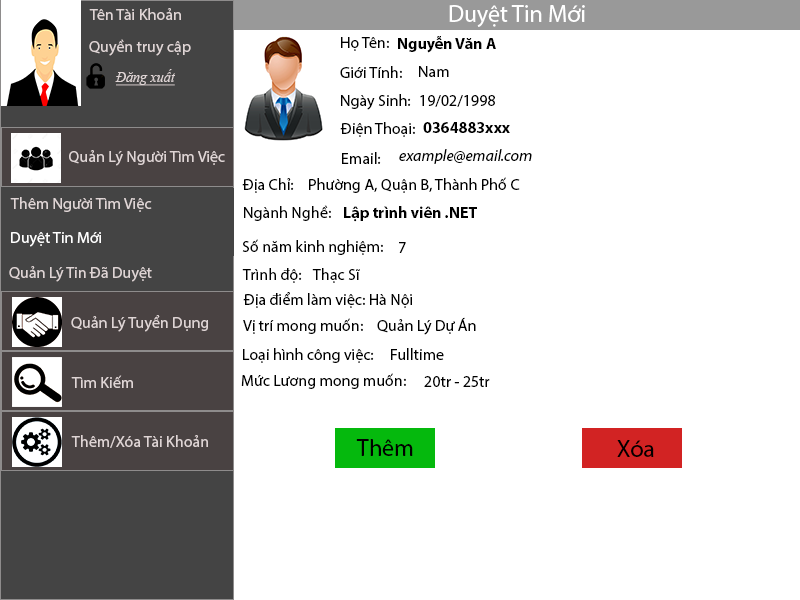
## **2.1. Thiết Kế Cấu Trúc**

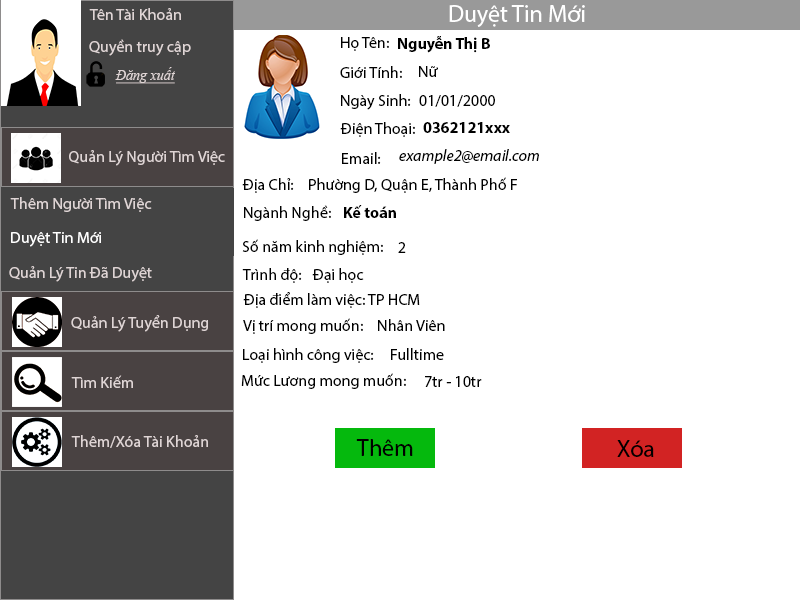
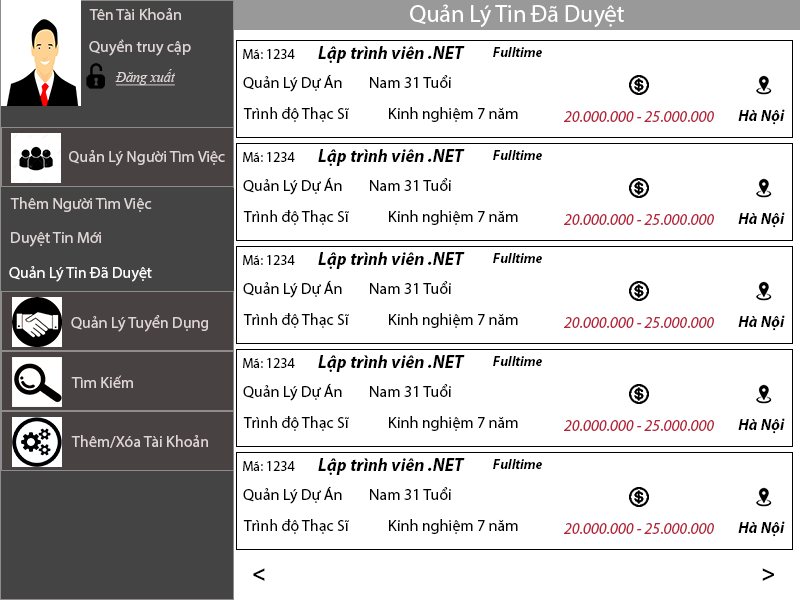
## **2.2. Thiết Kế Giao Diện**

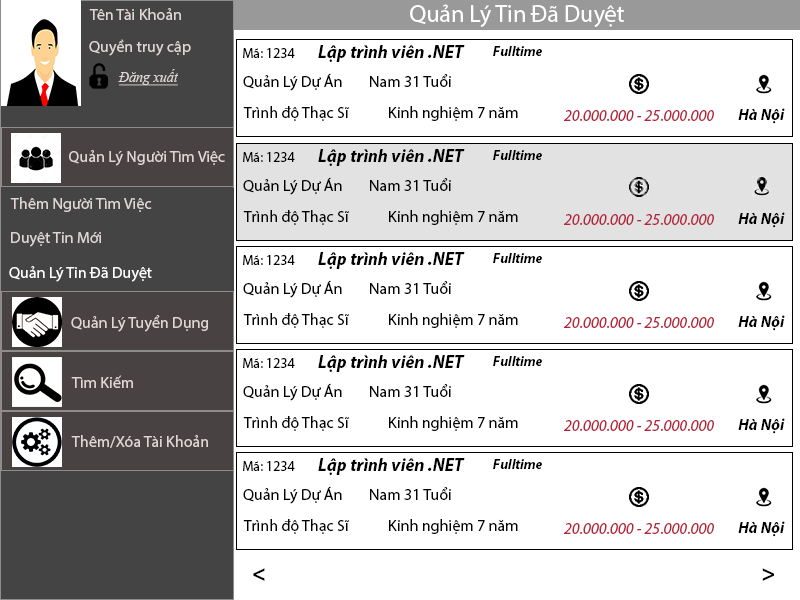
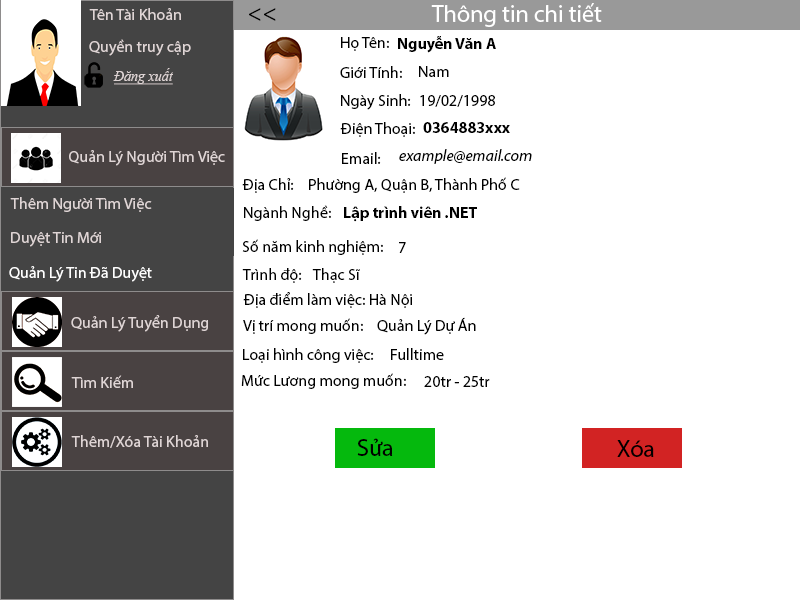
-Giao diện Đăng Nhập

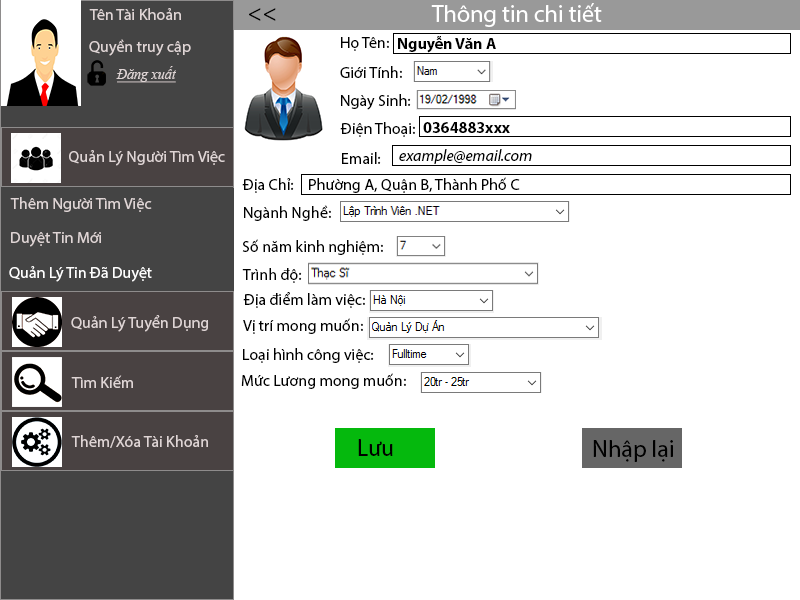
### **2.2.1.Giao diện Quản Lý Người Tìm Việc**

**-**Khi mới đăng nhập****

-Thêm người tìm việc****-Duyệt tin mới

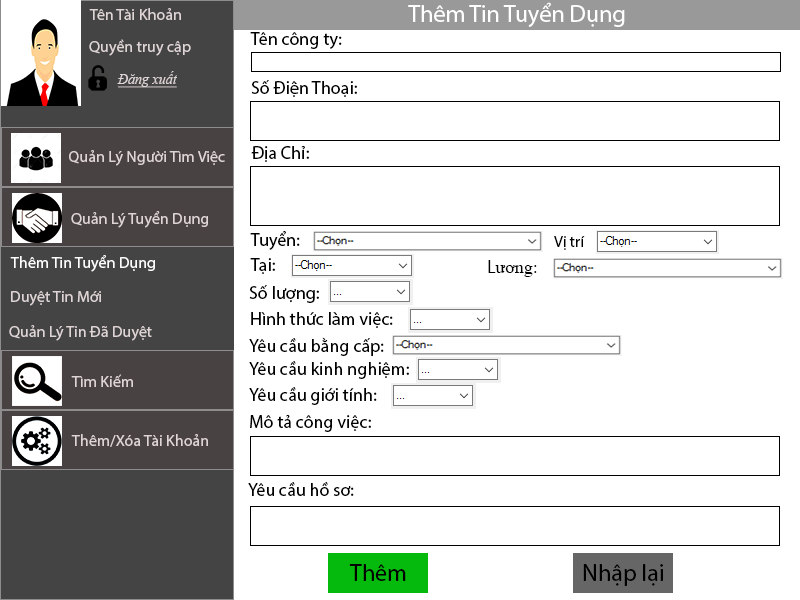
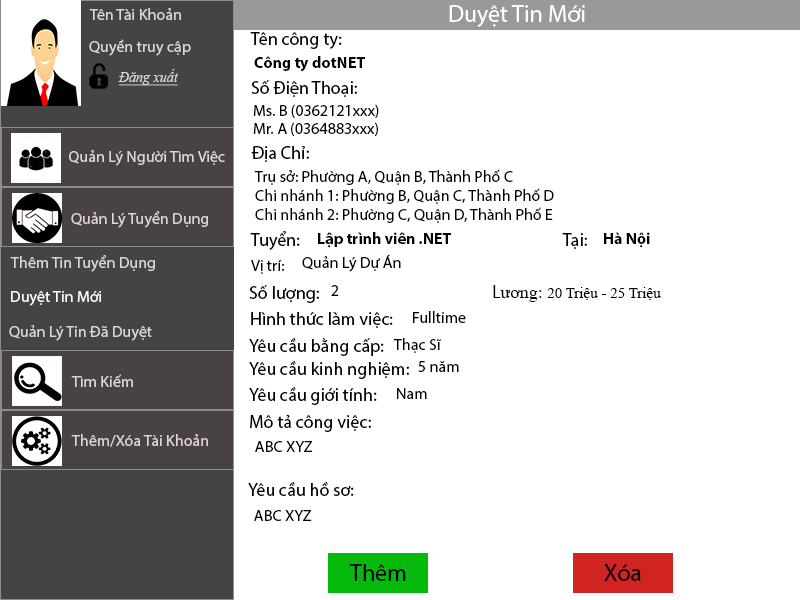
-Duyệt Tin Mới (Sau khi ấn Thêm hoăc Xóa [Hiển Thị Tin Tiếp Theo])- Quản Lý Tin Đã Duyệt

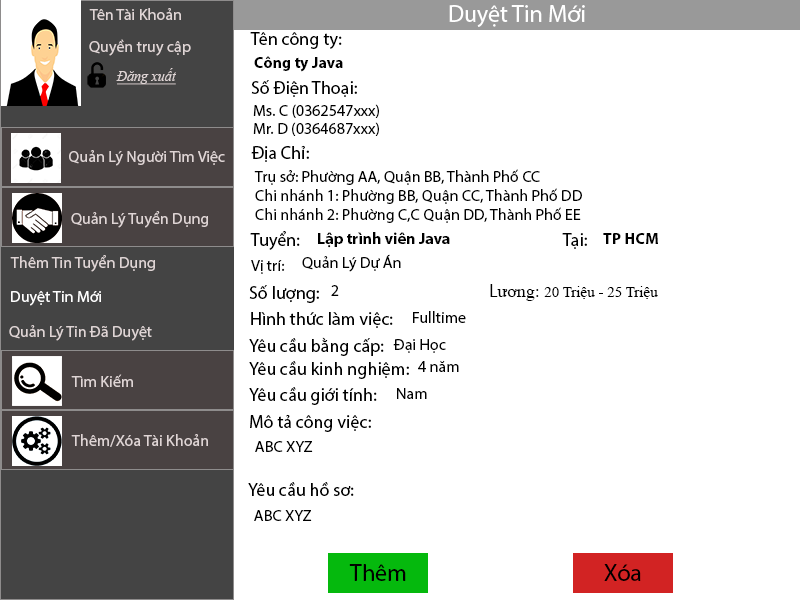
-Quản Lý Tin Đã Duyệt (Khi Trỏ Chuột vào Tin)- Quản Lý Tin Đã Duyệt - Sau Khi Click Chọn 1 Tin Bất Kỳ (Hiển thị chi tiết)

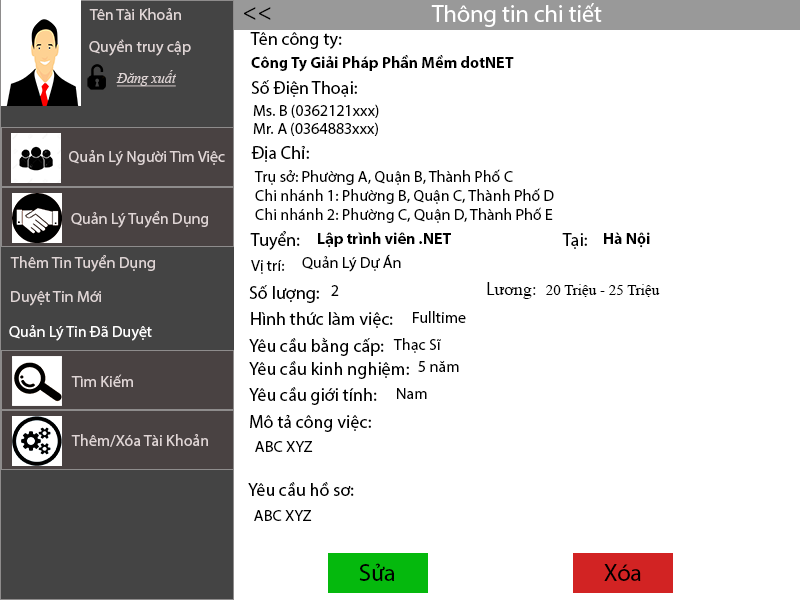
-Quản Lý Tin Đã Duyệt - Sau khi Click chọn Sửa (các label visible=false, textbox,combobox,… visible=true)

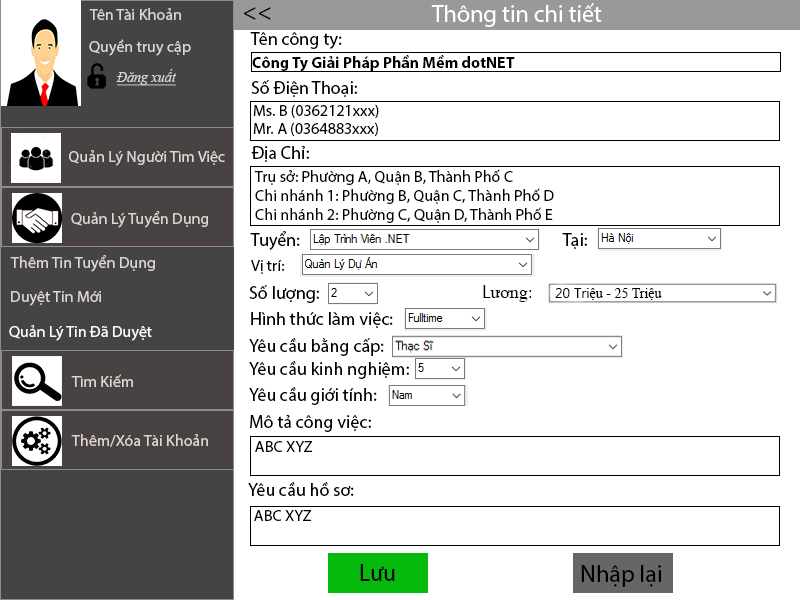
**\*\*\*Chú ý: Nếu Quyền Truy Cập là Nhân Viên thì Nút Thêm/Xóa Tài Khoản *KHÔNG* hiển thị !**

### **2.2.2. Giao diện Quản Lý Tuyển Dụng**

-Thêm Tin Tuyển Dụng-Duyệt Tin Mới

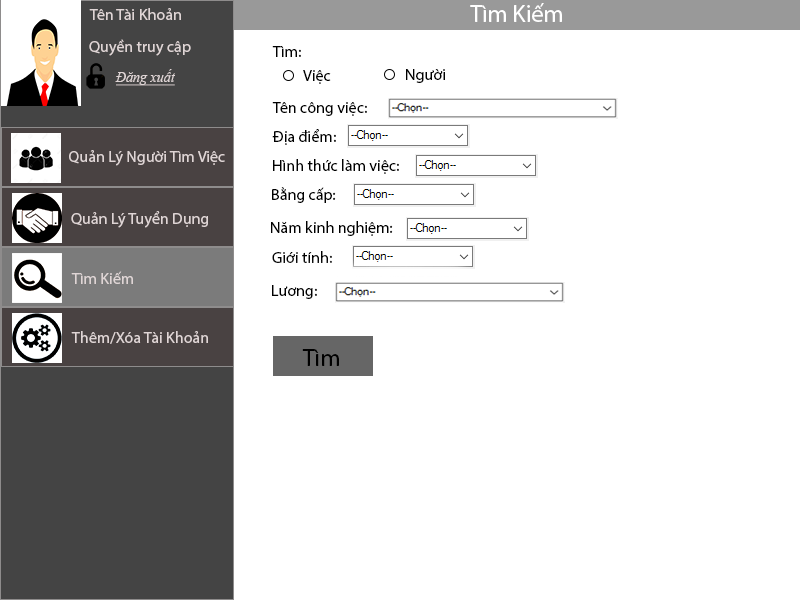
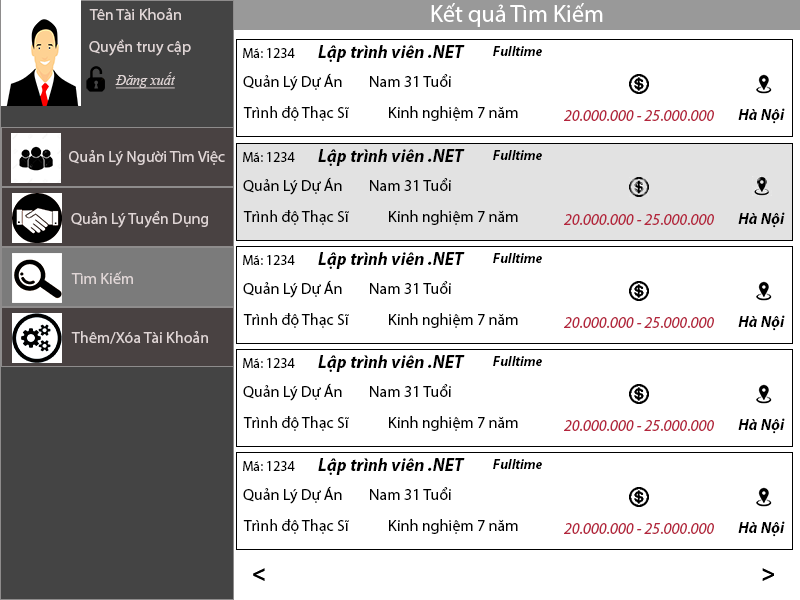
-Duyệt Tin Mới(Sau Khi Nhấn Button[Hiển thị Tin Tiếp Theo])****- Quản Lý Tin Đã Duyệt****

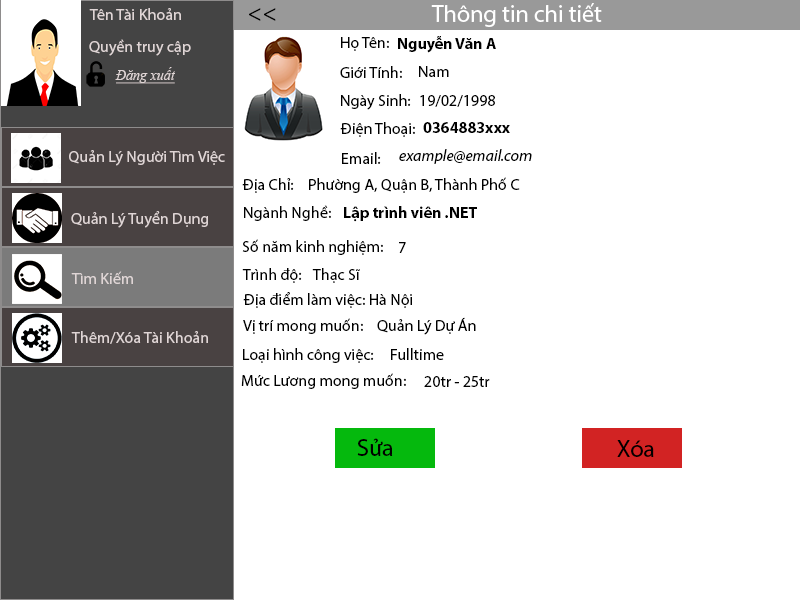
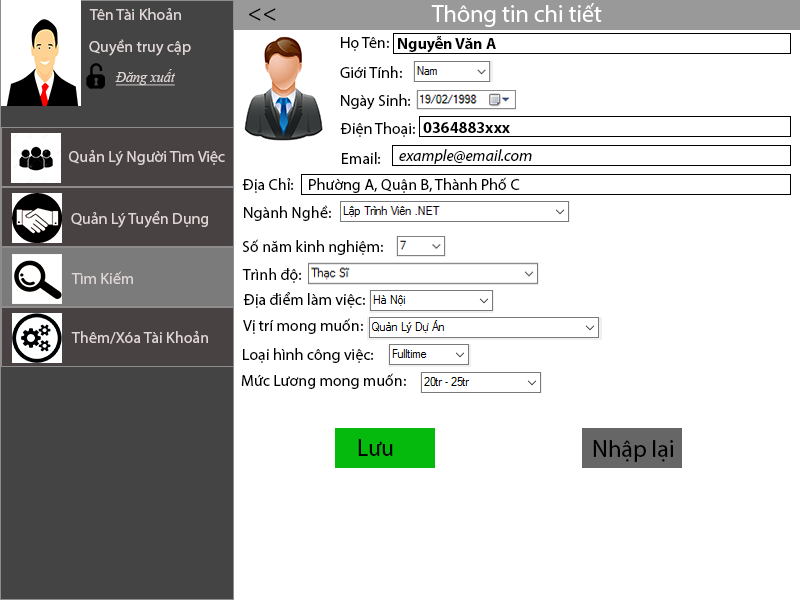
-Quản Lý Tin Đã Duyệt (Khi Trỏ Chuột vào Tin)****- Quản Lý Tin Đã Duyệt - Sau Khi Click Chọn 1 Tin Bất Kỳ (Hiển thị chi tiết)****

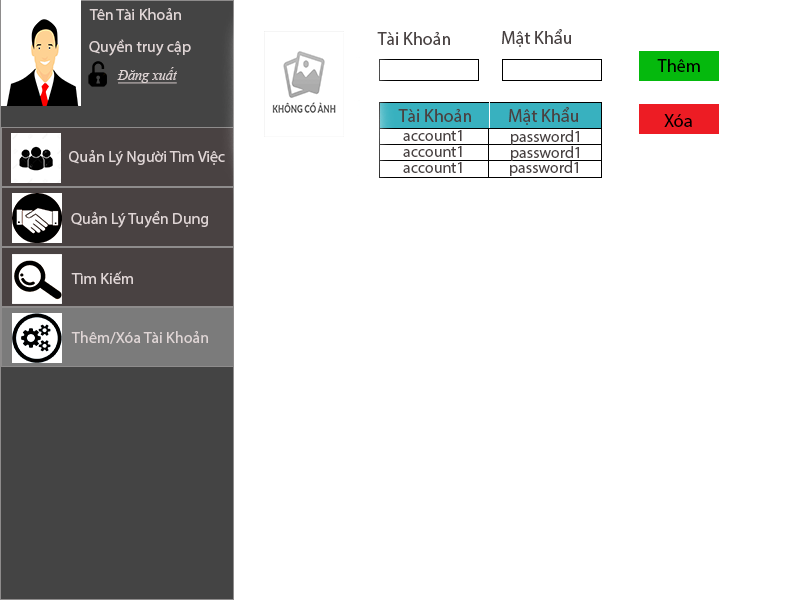
-Quản Lý Tin Đã Duyệt - Sau khi Click chọn Sửa (các label visible=false, textbox,combobox,… visible=true)****

**\*\*\*Chú ý: Nếu Quyền Truy Cập là Nhân Viên thì Nút Thêm/Xóa Tài Khoản *KHÔNG* hiển thị !**

**2.2.3. Giao diện Tìm Kiếm**

-Sau khi chọn Tìm Kiếm-Kết quả Tìm Kiếm

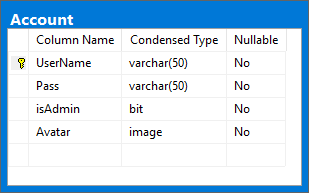
-Sau khi Click Chọn 1 tin bất kỳ (Hiển thị chi tiết)- Tìm Kiếm - Sau khi Click chọn Sửa (ẩn Label; hiện textbox,combobox)

**2.2.4. Thêm Xóa Tài Khoản (Chỉ hiển thị khi quyền truy cập là ADMIN)**

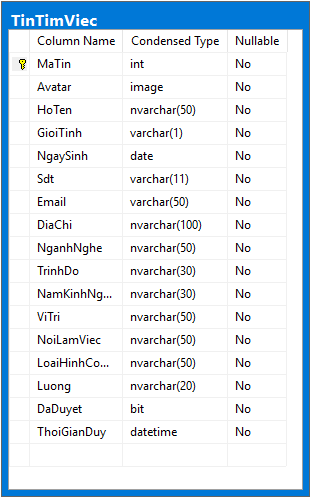
**2.3. Thiết Kế Cơ Sở Dữ Liệu**

**-**Tên Cơ Sở Dữ Liệu: “NguoiTimViec\_ViecTimNguoi”

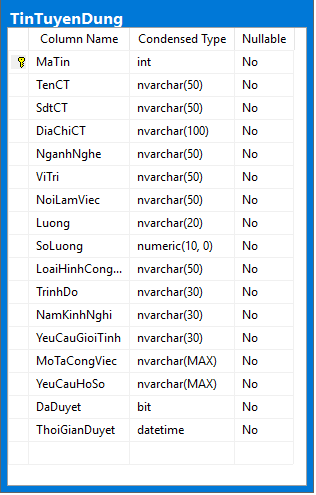
-Bảng Account



-Bảng TinTimViec

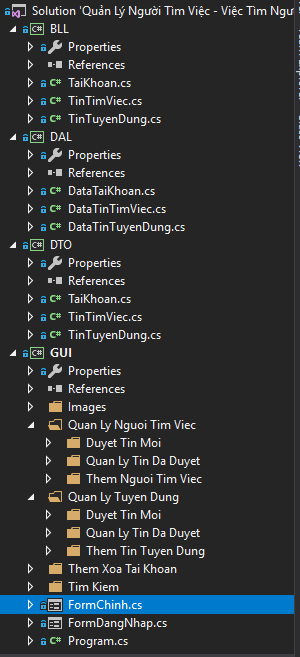
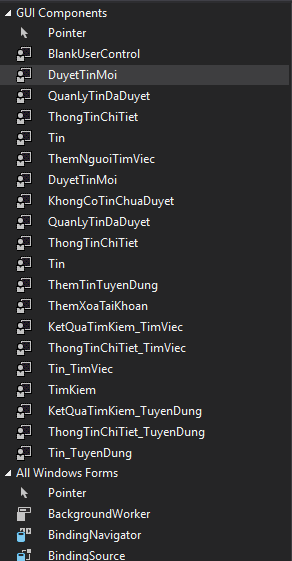


-Bảng TinTuyenDung

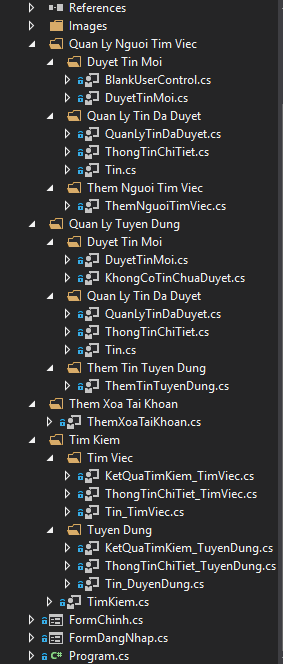


# **Chương 3: Chương trình và các kết quả đạt được**

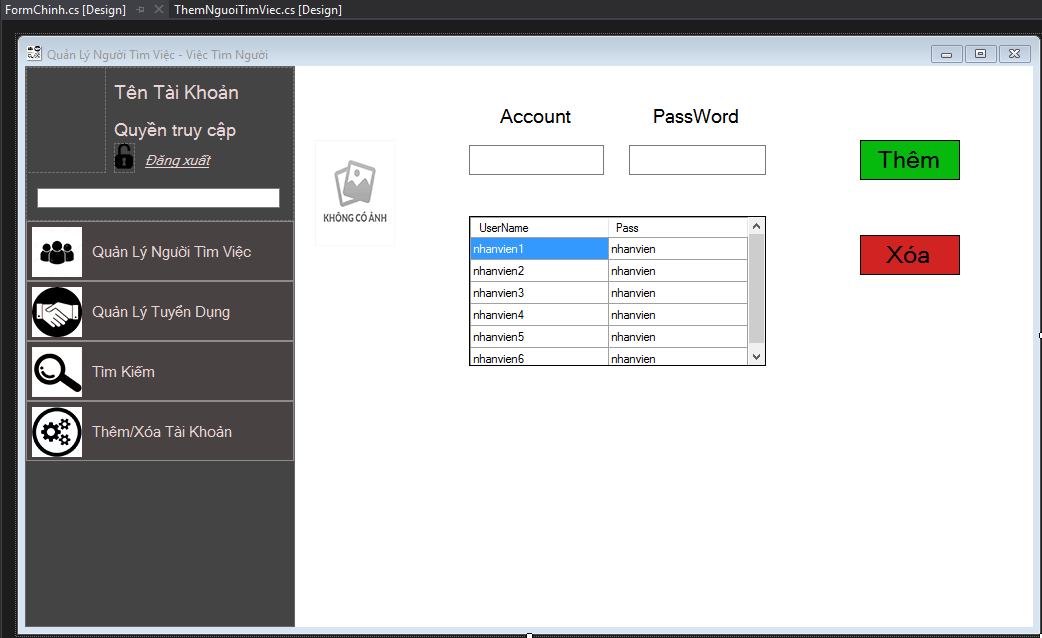
-Phân chia theo mô hình 3 lớp và các chức năng(mỗi chức năng là các UserControl)



-Các UserControl của từng chức năng

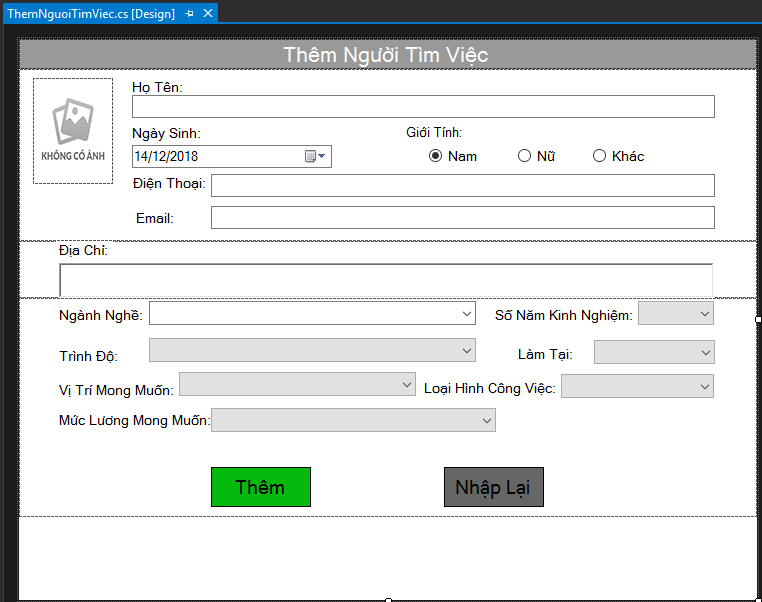


-Xây dựng FormChinh.cs chứa 2 panel (pnlNavigation chứa Menu và thông tin Đăng nhập, pnlContent chứa các UserControl của các chức năng)

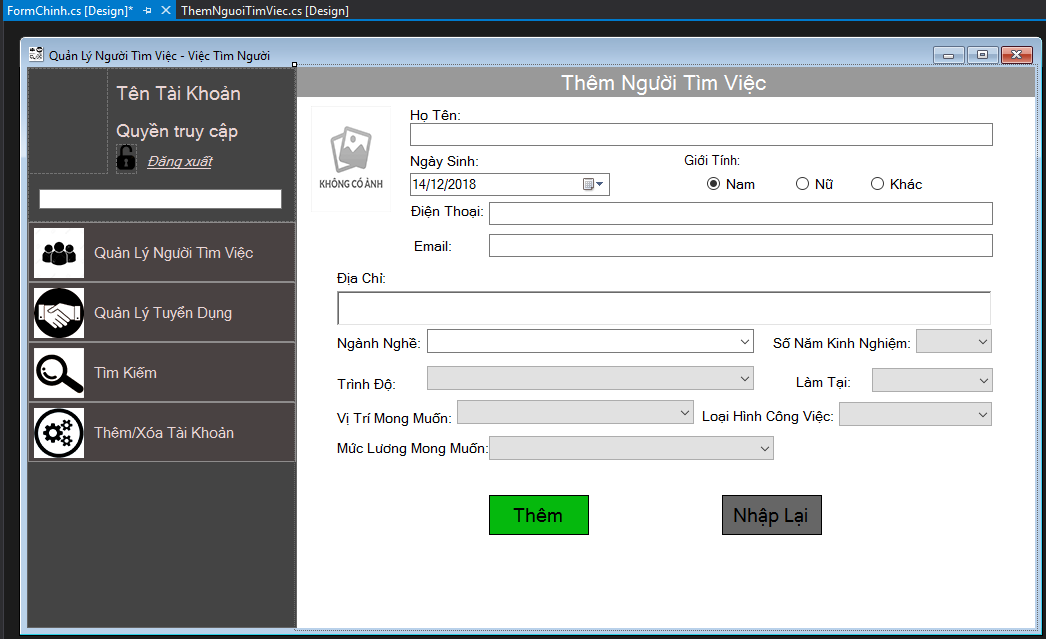


-Mỗi Thành Viên xây dựng 5-7 UserControl như sau

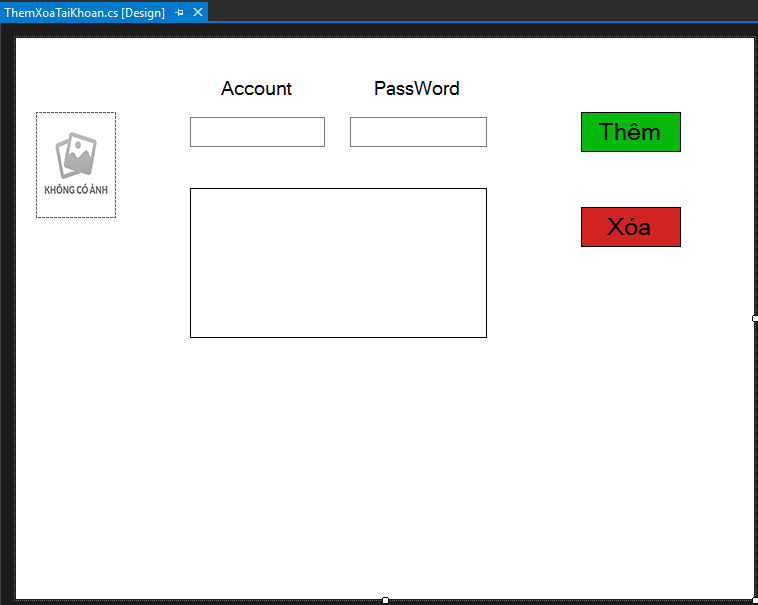
+Xây dựng UserControl Thêm Người Tìm Việc



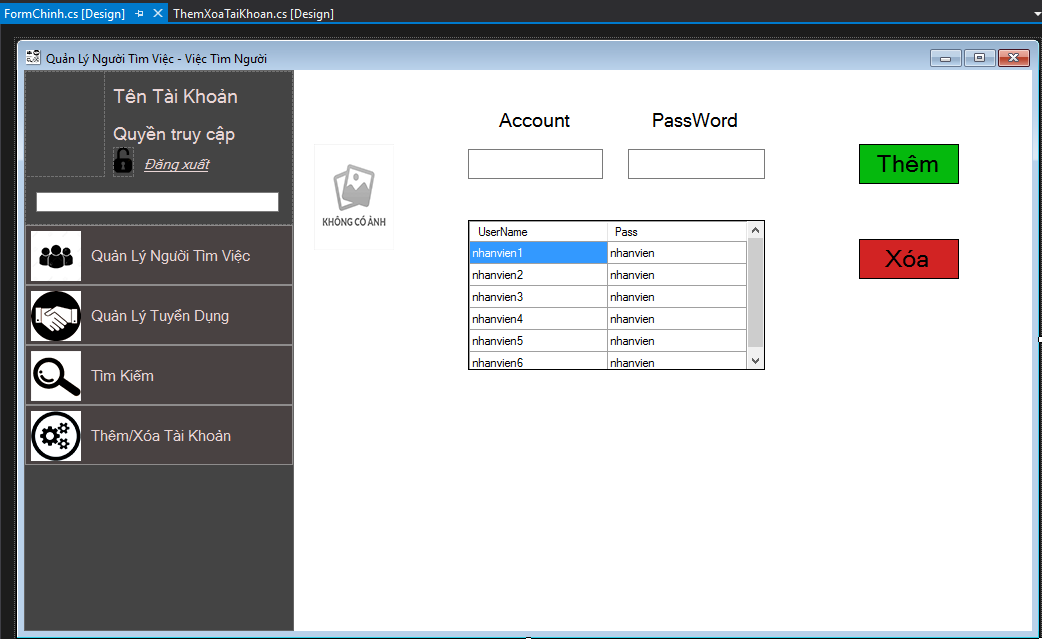
+Ghép UserControl Thêm Người Tìm Việc vào pnlContent của FormChinh.cs



+Xây dựng UserControl Thêm xóa Tài Khoản



-Ghép Thêm Xóa Tài Khoản vào pnlContent của FormChinh.cs



+Các Các UserControl khác xây dựng và ghép vào pnlContent của FormChinh.cs tương tự như ví dụ trên.

-Xây dựng Menu

+Mỗi nút Chức năng chính trên Menu được bao bởi 1 panel.

+Chiều cao panel bằng chiều cao của nút chính

+Khi click nút chính thì hiển thị các nút con bên dưới bằng cách tăng thuộc tính chiều cao Height của panel bọc các nút. Nếu Chiều cao của panel nhỏ hơn tổng chiều cao của các nút trong panel thì cho chiều cao panel bằng chiều cao các nút mà panel chứa.

+Khi click lần nữa thì chiều cao của Panel giảm đến khi bằng chiều cao của nút chính.

-Gọi khi click vào các nút con (các chức năng thì hiển thị UserControl tương ứng như đoạn code ví dụ sau)

private void btnDuyetTinMoi\_TimViec\_Click(object sender, EventArgs e)

{

duyetTinMoi\_TimViec.Show();

duyetTinMoi\_TimViec.BringToFront();

setHolderButton(sender);

duyetTinMoi\_TimViec.Active();

}

private void btnQuanLyTinDaDuyet\_TimViec\_Click(object sender, EventArgs e)

{

quanLyTinDaDuyet\_TimViec.Show();

quanLyTinDaDuyet\_TimViec.BringToFront();

setHolderButton(sender);

quanLyTinDaDuyet\_TimViec.Active(sender, e);

}

private void btnThemTinTuyenDung\_Click(object sender, EventArgs e)

{

themTinTuyenDung.Show();

themTinTuyenDung.BringToFront();

setHolderButton(sender);

}

private void btnDuyetTinMoi\_TuyenDung\_Click(object sender, EventArgs e)

{

duyetTinMoi\_TuyenDung.Show();

duyetTinMoi\_TuyenDung.Active();

duyetTinMoi\_TuyenDung.BringToFront();

setHolderButton(sender);

}

-Khi Chạy chương trình, hiển thị giao diện đăng nhập đầu tiên

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new formDangNhap());

}

}



-Khi đăng nhập thành công thì gọi Constructor của FormChinh.cs và truyền thông tin tài khoản đăng nhập qua DTO.TaiKhoan.cs

private void btnLogin\_Click(object sender, EventArgs e)

{

try

{

string username = txtUsername.Text;

string pass = txtPassword.Text;

if (TaiKhoanBLL.Instance.check(username, pass))

{

DTO.TaiKhoan tk = new DTO.TaiKhoan();

tk = TaiKhoanBLL.Instance.getInfoByID(username);

FormChinh mainForm = new FormChinh(tk);

mainForm.Show();

this.Hide();

}

else MessageBox.Show("Sai tên tài khoản hoặc mật khẩu");

}

catch

{

MessageBox.Show("Lỗi, đăng nhập k thành công ");

}

}

## **3.1.Quản Lý Người Tìm Việc**

### **3.1.1. Graphics User Interface**

### 3.1.1.1.Thêm người tìm việc

**-**Kiểm tra hợp lệ của các textbox

//txtDienThoai chi chap nhan so 0-9

private void txtDienThoai\_TextChanged(object sender, EventArgs e)

{

if (txtDienThoai.Text.Length > 0

&& System.Text.RegularExpressions.Regex.IsMatch(txtDienThoai.Text.ElementAt(txtDienThoai.Text.Length - 1).ToString(), "[^0-9]"))

{

txtDienThoai.Text = txtDienThoai.Text.Remove(txtDienThoai.TextLength - 1, 1);

txtDienThoai.Select(txtDienThoai.TextLength, 0);

}

}

//txtEmail màu đỏ nếu sai định dạng

private void txtEmail\_Leave(object sender, EventArgs e)

{

if (txtEmail.Text.Length > 0

&& !System.Text.RegularExpressions.Regex.IsMatch(txtEmail.Text.ToString(), @"[a-zA-Z0-9\_\.]@[a-zA-Z0-9]{2,}(\.[a-zA-Z0-9]{2,4}){1,2}$"))

{

txtEmail.ForeColor = Color.Red;

}

}

//Họ Tên KO chấp nhận ký tự hoặc số

private void txtHoTen\_TextChanged(object sender, EventArgs e)

{

if(txtHoTen.Text.Length > 0

&& System.Text.RegularExpressions.Regex.IsMatch(txtHoTen.Text.ToString(), "[^a-zA-ZáàảãạâấầẩẵậăắằẳẵặđéèẻẽẹêếềểễệíìỉĩịóòỏõọôốồổỗộơớờởỡợuúùủũụưứừửữựÁÀẢÃẠÂẤẦẨẴẬĂẮẰẲẴẶĐÉÈẺẼẸÊẾỀỂỄỆÍÌỈĨỊÓÒỎÕỌÔỐỒỔỖỘƠỚỜỞỠỢUÚÙỦŨỤƯỨỪỬỮỰ ]+$"))

{

txtHoTen.Text = txtHoTen.Text.Remove(txtHoTen.TextLength - 1, 1);

txtHoTen.Select(txtHoTen.TextLength, 0);

}

}

-Lọc và lưu ảnh Avatar

//Chọn ảnh và lưu ảnh (Byte[]) vào DTO.TinTimViec

private void ptbAvatar\_Click(object sender, EventArgs e)

{

OpenFileDialog openFile = new OpenFileDialog();

try

{

openFile.Filter = "Bitmap(\*.bmp;\*.dib) |\*.bmp;\*.dib " +

"| JPEG(\*.jpg;\*.jpeg;\*.jpe;\*.jfif) | \*.jpg;\*.jpeg;\*.jpe;\*.jfif " +

"| GIF(\*.gif) | \*.gif " +

"| TIFF(\*.tif;\*.tiff) | \*.tif;\*.tiff" +

"| PNG(\*.png) | \*.png" +

"| ICO(\*.ico) | \*.ico" +

"| All Picture Files |\*.bmp;\*.dib;\*.jpg;\*.jpeg;\*.jpe;\*.jfif;\*.gif;\*.tif;\*.tiff;\*.png;\*.ico" +

"| All Files | \*.\*";

if (openFile.ShowDialog() == DialogResult.OK)

{

Image img = new Bitmap(openFile.FileName);

MemoryStream memoryStream = new MemoryStream();

img.Save(memoryStream, System.Drawing.Imaging.ImageFormat.Jpeg);

tinDTO.Img = memoryStream.ToArray();

ptbAvatar.Image = img;

if (tinDTO.Img.Length > 1000000) // >1MB = 1000 000 Bytes

{

ptbAvatar.Image = Properties.Resources.image\_PictureBox;

MessageBox.Show("Chọn ảnh nhỏ hơn 1MB !");

}

}

}

catch(ArgumentException)

{

MessageBox.Show(openFile.FileName.ToString() + "\nFILE vừa chọn không phải là ảnh hoặc định dạng ảnh chưa được hỗ trợ.", "Error Box", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

catch(Exception er)

{

MessageBox.Show(er.ToString());

}

}

### 3.1.1.2. Duyệt Tin Mới

-Load tin lên UserControl

private void LoadData()

{

DTO.TinTimViec tinDTO = BLL.TinTimViec.Instance.getTinChuaDuyetCuNhat();

if (tinDTO != null)

{

ptbAvatar.Image = Image.FromStream(new MemoryStream(tinDTO.Img));

lblHoTen.Text = tinDTO.HoTen;

if (tinDTO.GioiTinh == "M")

lblGioiTinh.Text = "Nam";

if (tinDTO.GioiTinh == "F")

lblGioiTinh.Text = "Nữ";

if (tinDTO.GioiTinh == "O")

lblGioiTinh.Text = "Khác";

lblNgaySinh.Text = Convert.ToDateTime(tinDTO.NgaySinh).ToString("dd-MM-yyyy");

lblSoDienThoai.Text = tinDTO.SoDienThoai;

lblEmail.Text = tinDTO.Email;

lblDiaChi.Text = tinDTO.DiaChi;

lblNganhNghe.Text = tinDTO.NganhNghe;

lblSoNamKinhNghiem.Text = tinDTO.NamKinhNghiem;

lblTrinhDo.Text = tinDTO.TrinhDo;

lblDiaDiemLamViec.Text = tinDTO.NoiLamViec;

lblViTriMongMuon.Text = tinDTO.ViTri;

lblLoaiHinhCongViec.Text = tinDTO.LoaiHinhCongViec;

lblMucLuongMongMuon.Text = tinDTO.Luong;

blankUserControl.SendToBack();

blankUserControl.Height = panel1.Height + panel2.Height + panel3.Height;

}

else

{

blankUserControl.BringToFront();

}

}

-Duyệt, Xóa Tin

private void btnDuyet\_Click(object sender, EventArgs e)

{

string msg = "";

//msg = tin.DuyetTin();

msg = BLL.TinTimViec.Instance.DuyetTin();

if (msg != "")

MessageBox.Show(msg);

LoadData();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

string msg = "";

msg = BLL.TinTimViec.Instance.XoaTin();

if (msg != "")

MessageBox.Show(msg);

LoadData();

}

### 3.1.1.3.Quản Lý Tin Đã Duyệt

-Load các Tin (thêm UserControl Tin.cs vào flowTablePanel)

Int32 tinMin = tinHienTai;

Int32 tinMax = tinHienTai + soTinTrenMotTrang;

dt.Clear();

dt = BLL.TinTimViec.Instance.UpdateTableDaDuyet();

if (dt != null && dt.Rows.Count > 0)

{

for (Int32 i = tinMin; i < tinMax; i++)

{

if (i > dt.Rows.Count - 1)

break;

Tin tinUC = new Tin();

tinUC.Click += showDetail;

tinUC.Width = ftpContainer.Width - 7;

tinUC.Name = dt.Rows[i]["MaTin"].ToString();

ftpContainer.Controls.Add(tinUC);

tinHienTai++;

}

}

}

-Hiển thị chi tiết khi click vào 1 Tin

tinUC.Click += showDetail;

//hàm Click Event cho Mỗi Tin

private void showDetail(object sender, EventArgs e)

{

thongTinChiTiet.Show();

thongTinChiTiet.BringToFront();

Tin shortTin = sender as Tin;

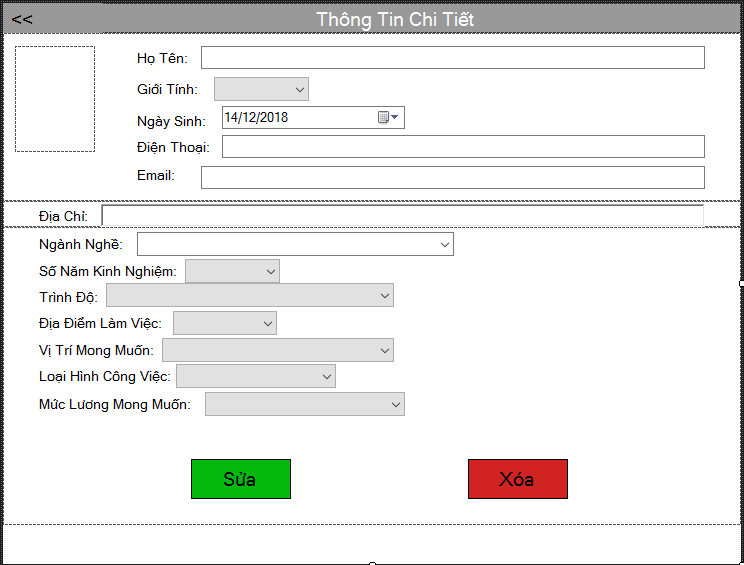
tinDTO = BLL.TinTimViec.Instance.getTinByMaTin(shortTin.Name);

thongTinChiTiet.LoadData(tinDTO);

}

-UserControl Tin.cs và thongTinChiTiet.cs





### **3.1.2. Business Logic Layer**

-Gồm các hàm và Singleton để gọi trực tiếp hàm, phục vụ cho lớp GUI

private static TinTimViec instance;

public static TinTimViec Instance

{

get

{

if (instance == null)

instance = new TinTimViec();

return instance;

}

}

public string KiemTraKetNoi()

public string Insert(DTO.TinTimViec tin)

public DataTable getTableFollowSearch(DTO.TinTimViec search)

public void UpdateTableChuaDuyet()

public DataTable UpdateTableDaDuyet()

public DTO.TinTimViec getTinChuaDuyetCuNhat()

public DTO.TinTimViec getTinDaDuyetMoiNhat()

public DTO.TinTimViec getTinByMaTin(string maTin)

public string DuyetTin()

public string XoaTin()

public string XoaMaTin(string maTin)

public string UpDate(DTO.TinTimViec tin)

### **3.1.3.Data Access Layer**

-Gồm các hàm và Singleton để gọi trực tiếp hàm, phục vụ cho lớp BLL

private static DataTinTimViec instance;

public static DataTinTimViec Instance

{

get

{

if (instance == null)

instance = new DataTinTimViec();

return instance;

}

}

public SqlConnection getConnect()

public DataTable getTinDaDuyet()

public DataTable getTinChuaDuyet()

public DataTable getTinByMaTin(string maTin)

public DataTable timKiem(SqlCommand cmd)

public string ExcuteNonQuery(SqlCommand command)

## **3.2.Quản Lý Tuyển Dụng**

### **3.2.1. Graphics User Interface**

### 3.2.1.1.Thêm tin tuyển dụng

-Kiểm tra textbox, chỉnh chiều cao RichtextBox khi gõ chữ

//ten cong ty chi duoc nhap chu

private void txtTenCongTy\_TextChanged(object sender, EventArgs e)

{

if (txtTenCongTy.Text.Length > 0 && System.Text.RegularExpressions.Regex.IsMatch(txtTenCongTy.Text.ToString(), "[^a-zA-ZáàảãạâấầẩẵậăắằẳẵặđéèẻẽẹêếềểễệíìỉĩịóòỏõọôốồổỗộơớờởỡợuúùủũụưứừửữựÁÀẢÃẠÂẤẦẨẴẬĂẮẰẲẴẶĐÉÈẺẼẸÊẾỀỂỄỆÍÌỈĨỊÓÒỎÕỌÔỐỒỔỖỘƠỚỜỞỠỢUÚÙỦŨỤƯỨỪỬỮỰ ]+$"))

{

txtTenCongTy.Text = txtTenCongTy.Text.Remove(txtTenCongTy.TextLength - 1, 1);

txtTenCongTy.Select(txtTenCongTy.TextLength, 0);

}

}

//thay doi kich thuoc cua cac RichTextBox khi nhap nhieu dong

private void rtbDiaChi\_TextChanged(object sender, EventArgs e)

{

rtbDiaChi.Height = (rtbDiaChi.GetLineFromCharIndex(rtbDiaChi.Text.Length) + 2) \* rtbDiaChi.Font.Height + rtbDiaChi.Margin.Vertical + 5;

}

private void rtbSDT\_TextChanged(object sender, EventArgs e)

{

rtbSDT.Height = (rtbSDT.GetLineFromCharIndex(rtbSDT.Text.Length) + 2) \* rtbSDT.Font.Height + rtbSDT.Margin.Vertical + 5;

}

private void rtbMoTaCongViec\_TextChanged(object sender, EventArgs e)

{

rtbMoTaCongViec.Height = (rtbMoTaCongViec.GetLineFromCharIndex(rtbMoTaCongViec.Text.Length) + 2) \* rtbMoTaCongViec.Font.Height + rtbMoTaCongViec.Margin.Vertical + 5;

}

private void rtbYeuCauHoSo\_TextChanged(object sender, EventArgs e)

{

rtbYeuCauHoSo.Height = (rtbYeuCauHoSo.GetLineFromCharIndex(rtbYeuCauHoSo.Text.Length) + 2) \* rtbYeuCauHoSo.Font.Height + rtbYeuCauHoSo.Margin.Vertical + 5;

}

### 3.2.1.2. Duyệt Tin Mới

-Load Tin

public void loadtin()

{

DTO.TinTuyenDung tinDTO = tin.getTinChuaDuyetCuNhat();

if (tinDTO != null)

{

lblTenCongTy.Text = tinDTO.TenCT;

lblSDT.Text = tinDTO.SdtCT;

lblDiaChi.Text = tinDTO.DiaChiCT;

lblTuyen.Text = tinDTO.NganhNghe;

lblNoiLamViec.Text = tinDTO.NoiLamViec;

lblViTri.Text = tinDTO.ViTri;

lblLuong.Text = tinDTO.Luong;

lblSoLuong.Text = tinDTO.SoLuong.ToString();

lblHinhThucLamViec.Text = tinDTO.LoaiHinhCongViec;

lblYeuCauBangCap.Text = tinDTO.TrinhDo;

lblYeuCauKinhNghiem.Text = tinDTO.NamKinhNghiem;

lblYeuCauGioiTinh.Text = tinDTO.YeuCauGioiTinh;

lblMoTaCongViec.Text = tinDTO.MoTaCongViec;

lblYeuCauHoSo.Text = tinDTO.YeuCauHoSo;

khongCoTinChuaDuyet.SendToBack();

khongCoTinChuaDuyet.Height = panelMot.Height + panelHai.Height + panelBa.Height + panelBon.Height + panelNam.Height + panelSau.Height;

}

else

{

khongCoTinChuaDuyet.BringToFront();

}

}

-Thêm, Xóa Tin

private void btnThem\_Click(object sender, System.EventArgs e)

{

string them = null;

them = tin.DuyetTin();

if (them != "")

MessageBox.Show("Thêm tin thành công");

loadtin();

}

private void btnXoa\_Click(object sender, System.EventArgs e)

{

string xoa;

xoa = tin.XoaTin();

if (xoa != "")

{

MessageBox.Show("Xóa tin thành công");

}

loadtin();

}

### 3.2.1.3.Quản Lý Tin Đã Duyệt

-Load các Tin (thêm UserControl Tin.cs vào flowTablePanel)

private void LoadData()

{

Int32 tinMin = tinHienTai;

Int32 tinMax = tinHienTai + soTinTrenMotTrang;

dt.Clear();

dt = BLL.TinTuyenDung.Tin.updateTinDaDuyet();

if (dt != null && dt.Rows.Count > 0)

{

for (Int32 i = tinMin; i < tinMax; i++)

{

if (i > dt.Rows.Count - 1)

break;

Tin tinUC = new Tin();

tinUC.Click += showDetail;

tinUC.Width = flpConten.Width - 7;

tinUC.Name = dt.Rows[i]["MaTin"].ToString();

flpConten.Controls.Add(tinUC);

tinHienTai++;

}

}

}

-Hiển thị chi tiết khi click vào 1 Tin

tinUC.Click += showDetail;

private void showDetail(object sender, EventArgs e)

{

Tin shortTin = sender as Tin;

tinDTO = BLL.TinTuyenDung.Tin.getTinByMaTin(shortTin.Name);

thongTinChiTiet.loadTin(tinDTO);

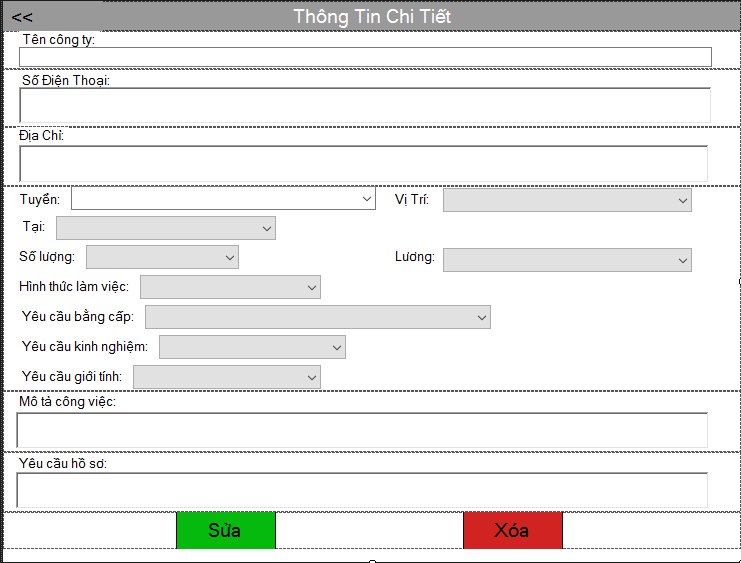
thongTinChiTiet.Show();

thongTinChiTiet.BringToFront();

}

-UserControl Tin.cs và thongTinChiTiet.cs





### **3.2.2. Business Logic Layer**

-Gồm các hàm và Singleton để gọi trực tiếp, phục vụ cho lớp GUI

private static TinTuyenDung tin;

public static TinTuyenDung Tin

{

get

{

if (tin == null)

tin = new TinTuyenDung();

return tin;

}

}

public string themtintuyendung(DTO.TinTuyenDung tin)

public void updateTinChuaDuyet()

public DataTable updateTinDaDuyet()

public DTO.TinTuyenDung getTinChuaDuyetCuNhat()

public string DuyetTin()

public string XoaTin()

public DTO.TinTuyenDung getTinByMaTin(string maTin)

public DTO.TinTuyenDung getTinDaDuyetMoiNhat()

public string XoaTinByMaTin(string maTin)

public string Update(DTO.TinTuyenDung tin)

public DataTable TimKiem(DTO.TinTuyenDung search)

### **3.2.3.Data Access Layer**

-Gồm các hàm và Singleton để gọi trực tiếp hàm, phục vụ cho lớp BLL

private static DataTinTuyenDung data;

public static DataTinTuyenDung DaTa

{

get

{

if(data == null)

{

data = new DataTinTuyenDung();

}

return data;

}

}

public SqlConnection getConnect()

public DataTable getTable()

public DataTable getTinDaDuyet()

public DataTable getTinChuaDuyet()

public bool ExecuteNonQuery(SqlCommand cmd)

public DataTable getTinByMaTin(string maTin)

public DataTable TimKiemTinTuyenDung(string sql)

## **3.3.Thêm, Xóa Tài Khoản**

### **3.3.1. Graphics User Interface**

-Thêm xóa tài khoản

private void btnAddAcc\_Click(object sender, EventArgs e)

{

if (txtAcc.Text == "" || txtPass.Text == "")

{

MessageBox.Show("Vui lòng nhập tài khoản và mật khẩu");

}

else

try

{

tk.Acc = txtAcc.Text;

tk.Pass = txtPass.Text;

Byte[] imgArr = new Byte[1000000];

Image img = new Bitmap(ptbAvatar.Image);

MemoryStream memStream = new MemoryStream();

img.Save(memStream, System.Drawing.Imaging.ImageFormat.Jpeg);

imgArr = memStream.ToArray();

tk.Avatar = imgArr;

TaiKhoanBLL.Instance.addAccount(tk);

ThemXoaTaiKhoan\_Load(sender, e);

ptbAvatar.Image = Properties.Resources.image\_PictureBox;

txtAcc.Text = "";

txtPass.Text = "";

}

catch (Exception er)

{

MessageBox.Show("Trùng Tài Khoản");

}

}

private void btnDelAcc\_Click(object sender, EventArgs e)

{

try

{

if (txtAcc.Text == "" || txtPass.Text == "")

{

MessageBox.Show("Vui lòng chọn tài khoản xóa");

}

else

{

tk.Acc = txtAcc.Text;

tk.Pass = txtPass.Text;

TaiKhoanBLL.Instance.delAccount(tk);

MessageBox.Show("Xóa thành công");

ThemXoaTaiKhoan\_Load(sender, e);

}

}

catch

{

MessageBox.Show("Error!!!");

}

}

### **3.3.2. Business Logic Layer**

-Gồm các hàm và Singleton để gọi trực tiếp hàm, phục vụ cho lớp GUI

private static TaiKhoanBLL instance;

public static TaiKhoanBLL Instance

{

get

{

if (instance == null)

instance = new TaiKhoanBLL();

return instance;

}

}

public DataTable getAccount()

public void addAccount(TaiKhoan tk)

public void delAccount(TaiKhoan tk)

public TaiKhoan getInfoByID(string id)

public bool check(string username, string pass)

### **3.3.3.Data Access Layer**

-Gồm các hàm và Singleton để gọi trực tiếp hàm, phục vụ cho lớp BLL

private static DataTaiKhoan instance;

public static DataTaiKhoan Instance

{

get

{

if (instance == null)

instance = new DataTaiKhoan();

return instance;

}

private set

{

DataTaiKhoan.instance = value;

}

}

public DataTable ExcuteQuery(string query)

public DataTable ExcuteQuery(SqlCommand cmd)

public void ExcuteNonQuery(string query)

public void ExcuteNonQuery(SqlCommand sql)

public DataTable getAccount()

public void addAccount(TaiKhoan tk)

public void delAccount(TaiKhoan tk)

public TaiKhoan getInfoByID(string id)

public bool checkAcc(string username, string pass)

**3.4.Tìm Kiếm**

**3.4.1. Graphics User Interface**

-Gửi thông tin tìm kiếm để load lên UserControl KetQuaTimKiem.cs để load các tin phù hợp

private void btnTim\_Click(object sender, EventArgs e)

{

if(rdoViec.Checked)

{

//load tim viec

DTO.TinTimViec tin = new DTO.TinTimViec();

tin.NganhNghe = cmbNganhNghe.Text == "" || cmbNganhNghe.Text == "Tất Cả Ngành Nghề" ? "%" : cmbNganhNghe.Text;

tin.LoaiHinhCongViec = cmbLoaiHinhCongViec.Text == "" || cmbLoaiHinhCongViec.Text == "Tất cả" ? "%" : cmbLoaiHinhCongViec.Text;

tin.NoiLamViec = cmbNoiLamViec.Text == "" || cmbNoiLamViec.Text == "Tất Cả" ? "%" : cmbNoiLamViec.Text;

tin.TrinhDo = cmbTrinhDo.Text == "" || cmbTrinhDo.Text == "Tất Cả Trình Độ" ? "%" : cmbTrinhDo.Text;

tin.NamKinhNghiem = cmbNamKinhNghiem.Text == "" || cmbNamKinhNghiem.Text == "Tất Cả" ? "%" : cmbNamKinhNghiem.Text;

tin.GioiTinh = cmbGioiTinh.Text == "" || cmbGioiTinh.Text == "Không yêu cầu giới tính" ? "%" : (cmbGioiTinh.Text == "Nam"? "M" : "F");

tin.Luong = cmbLuong.Text == "" || cmbLuong.Text == "Tất cả" ? "%" : cmbLuong.Text;

ketQuaTimKiem\_TimViec.sendData(tin);

ketQuaTimKiem\_TimViec.Active(sender, e);

ketQuaTimKiem\_TimViec.BringToFront();

}

if (rdoNguoi.Checked)

{

//load tuyen dung

DTO.TinTuyenDung tin = new DTO.TinTuyenDung();

tin.NganhNghe = cmbNganhNghe.Text == "" || cmbNganhNghe.Text == "Tất Cả Ngành Nghề" ? "%" : cmbNganhNghe.Text;

tin.LoaiHinhCongViec = cmbLoaiHinhCongViec.Text == "" || cmbLoaiHinhCongViec.Text == "Tất cả" ? "%" : cmbLoaiHinhCongViec.Text;

tin.NoiLamViec = cmbNoiLamViec.Text == "" || cmbNoiLamViec.Text == "Tất Cả" ? "%" : cmbNoiLamViec.Text;

tin.TrinhDo = cmbTrinhDo.Text == "" || cmbTrinhDo.Text == "Tất Cả Trình Độ" ? "%" : cmbTrinhDo.Text;

tin.NamKinhNghiem = cmbNamKinhNghiem.Text == "" || cmbNamKinhNghiem.Text == "Tất Cả" ? "%" : cmbNamKinhNghiem.Text;

tin.YeuCauGioiTinh = cmbGioiTinh.Text == "" || cmbGioiTinh.Text == "Không yêu cầu giới tính" ? "%" : cmbGioiTinh.Text;

tin.Luong = cmbLuong.Text == "" || cmbLuong.Text == "Tất cả" ? "%" : cmbLuong.Text;

ketQuaTimKiem\_TuyenDung.sendData(tin);

ketQuaTimKiem\_TuyenDung.Active(sender, e);

ketQuaTimKiem\_TuyenDung.BringToFront();

}

}

**3.4.2. Business Logic Layer**

public DataTable getTableFollowSearch(DTO.TinTimViec search)

{

try

{

string cmdString = "SELECT \* FROM TinTimViec " +

"WHERE NganhNghe LIKE @NganhNghe " +

"AND NoiLamViec LIKE @NoiLamViec " +

"AND LoaiHinhCongViec LIKE @LoaiHinhCongViec " +

"AND TrinhDo LIKE @TrinhDo " +

"AND NamKinhNghiem LIKE @NamKinhNghiem " +

"AND GioiTinh LIKE @GioiTinh " +

"AND Luong LIKE @Luong " +

"AND DaDuyet = 'true'";

SqlCommand cmd = new SqlCommand(cmdString);

cmd.Parameters.AddWithValue("@NganhNghe", search.NganhNghe);

cmd.Parameters.AddWithValue("@NoiLamViec", search.NoiLamViec);

cmd.Parameters.AddWithValue("@LoaiHinhCongViec", search.LoaiHinhCongViec);

cmd.Parameters.AddWithValue("@TrinhDo", search.TrinhDo);

cmd.Parameters.AddWithValue("@NamKinhNghiem", search.NamKinhNghiem);

cmd.Parameters.AddWithValue("@GioiTinh", search.GioiTinh);

cmd.Parameters.AddWithValue("@Luong", search.Luong);

DataTable table = new DataTable();

table = DataTinTimViec.Instance.timKiem(cmd);

return table;

}

catch(Exception)

{

return null;

}

}

public DataTable TimKiem(DTO.TinTuyenDung search)

{

try

{

string cmdtext = "select \* from TinTuyenDung where NganhNghe LIKE N'" + search.NganhNghe + "' AND NoiLamViec LIKE N'" + search.NoiLamViec + "' AND LoaiHinhCongViec LIKE N'" + search.LoaiHinhCongViec + "' AND TrinhDo LIKE N'" + search.TrinhDo + "' AND NamKinhNghiem LIKE N'" + search.NamKinhNghiem + "' AND Luong LIKE N'" + search.Luong + "'AND YeuCauGioiTinh LIKE N'" + search.YeuCauGioiTinh + "' ";

DataTable table = new DataTable();

dt = DAL.DataTinTuyenDung.DaTa.TimKiemTinTuyenDung(cmdtext);

return dt;

}

catch (Exception e)

{

return null;

}

}

**3.4.3.Data Access Layer**

public DataTable timKiem(SqlCommand cmd)

{

cmd.Connection = getConnect();

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataView dv = new DataView();

DataTable dt = new DataTable();

da.Fill(dt);

dv = dt.DefaultView;

dv.Sort = "ThoiGianDuyet DESC"; // Xem Tin Moi Nhat Truoc

dt = dv.ToTable();

return dt;

}

public DataTable TimKiemTinTuyenDung(string sql)

{

SqlDataAdapter da = new SqlDataAdapter(sql, getConnect());

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

# **Kết Luận**

* **Ưu điểm**
* Chương trình có giao diện thân thiện , thuận lợi cho người sử dụng.
* Các chức năng như tìm kiếm , thêm , sửa , xóa… được thực hiện nhanh chóng và thuận tiện .
* Phù hợp với các ứng dụng triển khai trên diện rộng.
* Dễ dàng sử dụng và quản lý.
* **Nhược điêm**
* Do thời gian nghiên cứu có hạn và khả năng còn hạn chế nên một số vẫn còn trên ý tưởng mà chưa được thực thi.
* Chương trình có tính chuyên nghiệp chưa cao ,chức năng còn ít .
* Chưa giải quyết trọn vẹn các vẫn đề nảy sinh trong quá trình quản lý.
* Giao diện tuy thân thiện nhưng tính thẩm mỹ chưa cao.
* **Hướng phát triển của phần mềm**
* Tìm hiểu và hoàn thiện một số chức năng còn chưa được thực hiện trong bài tập lớn.
* Hỗ trợ cơ sở dữ liệu SQL để khả năng bảo mật , tốc độ truy vấn , khả năng quản lý dữ liệu tốt hơn

*Đây là sản phẩm của các thành viên nhóm em. Nếu có gì sai sót mong thầy góp ý để sản phẩm của chúng em được hoàn thiện hơn ạ .*

*Em xin chân thành cảm ơn !*