



## BÀI 4

### WINDOWS FORM NÂNG CAO

#### A. MỤC TIÊU:

- Xây dựng Module Quản lý đăng nhập

#### B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

#### C. NỘI DUNG THỰC HÀNH

1. Chạy File Scrip **1.LTWNC** đính kèm, thực hiện nhập dữ liệu cho mỗi bảng 3 mẫu.

DM\_ManHinh(**MaManHinh**, TenManHinh)

QL\_NguoiDung (**TenDangNhap**, MatKhau, HoatDong)

QL\_NhomNguoiDung (**MaNhom**, TenNhom, GhiChu)

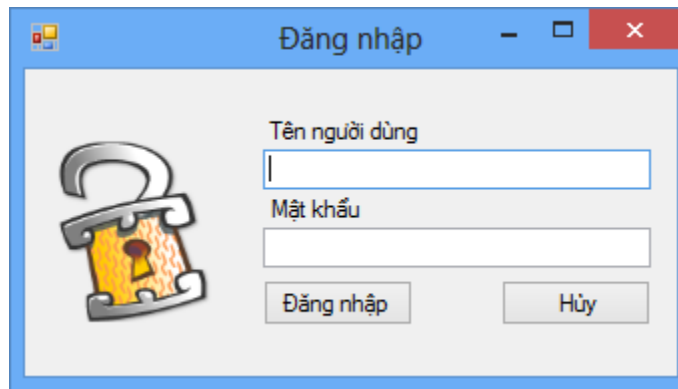
QL\_NguoiDungNhomNguoiDung(**TenDangNhap**,**MaNhomNguoiDung**, Ghi Chu).

QL\_PhanQuyen (**MaNhomNguoiDung**, **MaManHinh**, CoQuyen)

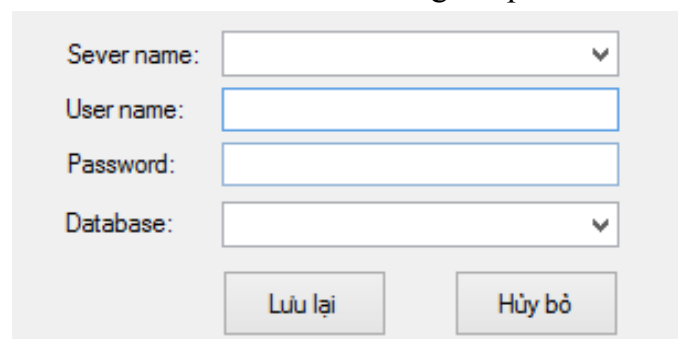
<b>QL_NguoiDung</b> TenDangNhap MatKhau HoatDong	<b>QL_NhomNguoiDung</b> MaNhom TenNhom GhiChu
<b>QL_PhanQuyen</b> MaNhomNguoiDung MaManHinh CoQuyen	<b>QL_NguoiDungNhomNguoiDung</b> TenDangNhap MaNhomNguoiDung GhiChu
	<b>DM_ManHinh</b> MaManHinh TenManHinh

2. Thực hiện Form Đăng nhập và Form Cấu Hình chuỗi kết nối

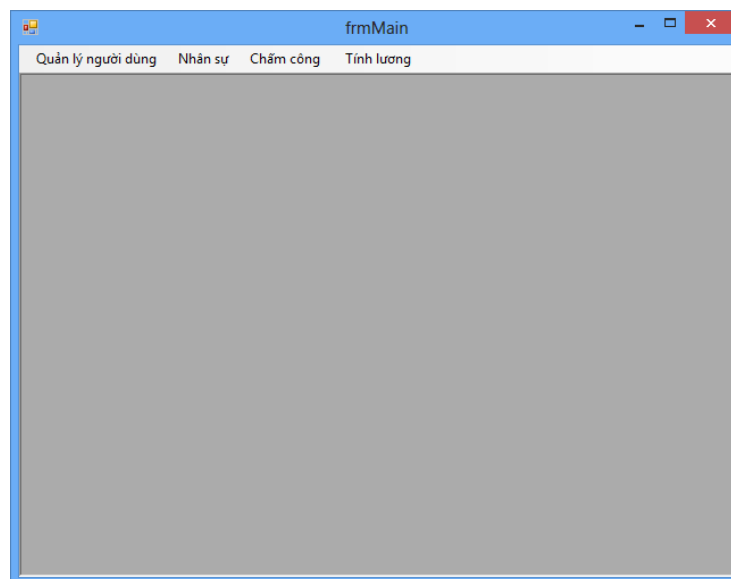
## 2.1 Xây dựng giao diện 3 Form như sau:



Hình 1. Form Đăng nhập



Hình 2. Form Cấu hình



Hình 3. Form Main

## 2.2 Quá trình xử lý thông tin đăng nhập cho một ứng dụng

1. Kiểm tra thông tin đăng nhập. (Nếu textbox Tên đăng nhập hoặc Mật khẩu bỏ trống thì bắt nhập lại).
2. Kiểm tra chuỗi thông tin kết nối trong tập tin “App.config”.
  - 0// Kết nối thành công chuỗi cấu hình hợp lệ
  - 1// Chuỗi cấu hình không tồn tại
  - 2// Chuỗi cấu hình không phù hợp.
3. Cấu hình chuỗi kết nối phù hợp Xử lý đăng nhập trường hợp 0.
  - Kiểm tra sự tồn tại của Tên người dùng và mật khẩu
  - Tên người dùng còn hoạt động hay không ?
4. Cấu hình chuỗi kết nối không phù hợp Xử lý tạo mới cấu hình nếu là trường hợp 1,2

### 2.3 Gợi ý thực hiện cho Form Đăng nhập

**B1:** Thực hiện sự kiện cho Hành động đăng nhập

```
if (string.IsNullOrEmpty(txtUsername.Text.Trim()))
{
    MessageBox.Show("Không được bỏ trống" + lblUsername.Text.ToLower());
    this.txtUsername.Focus();
    return;
}
if (string.IsNullOrEmpty(this.txtPassword.Text))
{
    MessageBox.Show("Không được bỏ trống" + lblPassword.Text.ToLower());
    this.txtPassword.Focus();
    return;
}
int kq=CauHinh.Check_Config(); //hàm Check_Config() thuộc Class QL_NguoiDung
if (kq == 0)
{
    ProcessLogin();// Cấu hình phù hợp xử lý đăng nhập
}
if (kq == 1)
{
    MessageBox.Show("Chuỗi cấu hình không tồn tại");// Xử lý cấu hình
    ProcessConfig();
}
if (kq == 2)
{
    MessageBox.Show("Chuỗi cấu hình không phù hợp");// Xử lý cấu hình
    ProcessConfig();
}
```

**B2:** Xây dựng Class `QL_NguoiDung` Kiểm tra chuỗi thông tin kết nối trong tập tin “App.config”:

```
Class QL_NguoiDung
public int Check_Config()
{
    if (Properties.Settings.Default.LTWNCConn == string.Empty)
        return 1;// Chuỗi cấu hình không tồn tại
    SqlConnection _Sqlconn = new
    SqlConnection(Properties.Settings.Default.LTWNCConn);
    try
```

```

    {
        if (_Sqlconn.State == System.Data.ConnectionState.Closed)
            _Sqlconn.Open();
        return 0; // Kết nối thành công chuỗi cấu hình hợp lệ
    }
    catch
    {
        return 2; // Chuỗi cấu hình không phù hợp.
    }
}

```

**B3:** Cấu hình chuỗi kết nối phù hợp Xử lý đăng nhập trường hợp 0:

Xây dựng enum `LoginResult`

```

public enum LoginResult
{
    /// <summary>
    /// Wrong username or password
    /// </summary>
    Invalid,
    /// <summary>
    /// User had been disabled
    /// </summary>
    Disabled,
    /// <summary>
    /// Logging success
    /// </summary>
    Success
}

```

Class Program chúng ta điều chỉnh như sau:

```

static class Program
{
    public static frmMain mainForm = null;
    public static LoginForm loginForm = null;
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        loginForm = new LoginForm();
        Application.Run( loginForm);
    }
}

```

Viết hàm `ProcessLogin()`

```

public void ProcessLogin()
{
    LoginResult result;
    result = CauHinh.Check_User(txtUsername.Text, txtPassword.Text); //
    Check_User viết trong Class QL_NguoiDung
    // Wrong username or pass
}

```

```

        if (result == LoginResult.Invalid)
        {
            MessageBox.Show("Sai " + lblUsername.Text + " Hoặc " +
lblPassword.Text);
            return;
        }
        // Account had been disabled
        else if (result == LoginResult.Disabled)
        {
            MessageBox.Show("Tài khoản bị khóa");
            return;
        }
        if (Program.mainForm == null || Program.mainForm.IsDisposed)
        {
            Program.mainForm = new frmMain();
        }
        this.Visible = false;
        Program.mainForm.Show();
    }
}

```

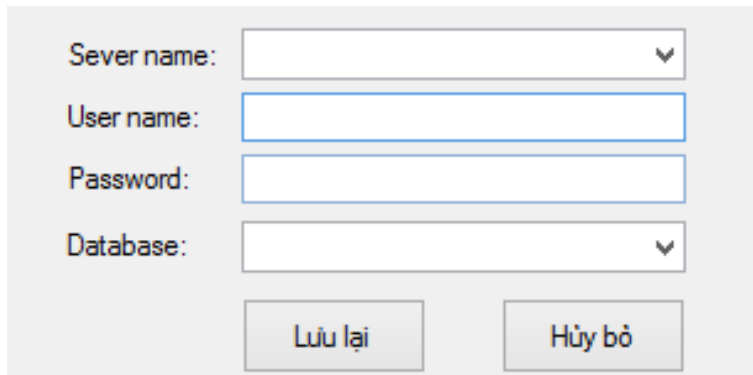
Viết hàm `Check_User` trong Class `QL_NguoiDung`

```

public LoginResult Check_User(string pUser, string pPass)
{
    SqlDataAdapter daUser = new SqlDataAdapter("select * from QL_NguoiDung
where TenDangNhap='" + pUser + "' and MatKhai='" + pPass + "'",
        Properties.Settings.Default.LTWNCConn);
    DataTable dt = new DataTable();
    daUser.Fill(dt);
    if (dt.Rows.Count == 0)
        return LoginResult.Invalid; // User không tồn tại
    else if (dt.Rows[0][2] == null || dt.Rows[0][2].ToString() == "False")
    {
        return LoginResult.Disabled; // Không hoạt động
    }
    return LoginResult.Success; // Đăng nhập thành công
}

```

**B4:** Cấu hình chuỗi kết nối không phù hợp Xử lý tạo mới cấu hình nếu là trường hợp 1 và 2:



The image shows a Windows Forms application window with a light gray background. It contains four labels on the left: "Sever name:", "User name:", "Password:", and "Database:". To the right of each label is a control: a dropdown menu for "Sever name", a text box for "User name", a text box for "Password", and a dropdown menu for "Database". At the bottom of the form are two buttons: "Lưu lại" (Save) and "Hủy bỏ" (Cancel).

Quá trình thực trên Form Cấu hình:

1. Tìm kiếm Server Name
2. Nhập User Sql
3. Nhập Password tương ứng bước 2
4. Tìm kiếm Database tương ứng với các thông tin hợp lệ trên
5. Tất cả hợp lệ lưu lại chuỗi cấu hình.

### **Tìm kiếm Server Name**

```
private void cboServer_DropDown(object sender, EventArgs e)
{
    cboServer.DataSource = CauHinh.GetServerName();
    cboServer.DisplayMember = "ServerName";
}
```

```
public DataTable GetServerName()
{
    DataTable dt = new DataTable();
    dt = SqlConnectionEnumerator.Instance.GetDataSources();
    return dt;
}
```

### **Nhập User Sql**

### **Nhập Password tương ứng bước 2**

### **Tìm kiếm Database tương ứng với các thông tin hợp lệ trên**

```
private void cboDataBase_DropDown(object sender, EventArgs e)
{
    cboDataBase.DataSource =
    CauHinh.GetDBName(cboServer.Text,txtUsername.Text,txtPassword.Text);
    cboDataBase.DisplayMember = "name";
}
```

```
public DataTable GetDBName(string pServer, string pUser, string pPass)
{
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter("select name from sys.Databases",
    "Data Source=" + pServer + ";Initial Catalog=master;User ID=" + pUser + ";pwd = " +
    pPass + "");
    da.Fill(dt);
    return dt;
}
```

### **Tất cả hợp lệ lưu lại chuỗi cấu hình**

```
private void btnLuu_Click(object sender, EventArgs e)
{
    CauHinh.SaveConfig(cboServer.Text, txtUsername.Text, txtPassword.Text,
    cboDataBase.Text);
    this.Close();
}
```

```
public void SaveConfig(string pServer, string pUser, string pPass, string pDBname)
{
    DemoDN.Properties.Settings.Default.LTWNCConn = "Data Source=" + pServer +
```

```
";Initial Catalog=" + pDBname + ";User ID=" + pUser + ";pwd = " + pPass + "";  
    DemoDN.Properties.Settings.Default.Save();  
}
```

### Mã hóa Password dùng hàm sau:

```
public const string PUBLIC_KEY = "05DHLTH";  
  
    /// <summary>  
    /// Encrypts the specified value.  
    /// </summary>  
    /// <param name="value">The value.</param>  
    /// <returns></returns>  
    public static string Encrypt(string value, string publickey)  
    {  
        if (string.IsNullOrEmpty(value))  
        {  
            return string.Empty;  
        }  
  
        byte[] bytesIn = Encoding.UTF8.GetBytes(value);  
  
        DESCryptoServiceProvider des = new DESCryptoServiceProvider();  
  
        byte[] bytesKey = Encoding.UTF8.GetBytes(publickey);  
        Array.Resize(ref bytesKey, des.Key.Length);  
        Array.Resize(ref bytesKey, des.IV.Length);  
  
        des.Key = bytesKey;  
        des.IV = bytesKey;  
  
        MemoryStream msOut = new MemoryStream();  
        ICryptoTransform desdecrypt = des.CreateEncryptor();  
  
        CryptoStream cryptStream = new CryptoStream(msOut, desdecrypt,  
        CryptoStreamMode.Write);  
  
        cryptStream.Write(bytesIn, 0, bytesIn.Length);  
        cryptStream.FlushFinalBlock();  
        byte[] bytesOut = msOut.ToArray();  
  
        cryptStream.Close();  
        msOut.Close();  
  
        return Convert.ToBase64String(bytesOut);  
    }
```