

Thực hành chức năng đăng nhập

---o0o---

1. Câu chuyện người dùng (User Story)

Người dùng muốn:

- Có thể đăng nhập (Login), đăng xuất Logout hệ thống
- Có thể đăng nhập bằng nhiều User, **Mỗi User có quyền khác nhau (Admin, User)**
- Có thể thay đổi mật khẩu của User (Admin có thể reset mật khẩu của User, User có thể đổi lại mật khẩu của tài khoản của mình)
- Admin quản lý được User (Thêm, Sửa, Xóa thông tin của tài khoản)
- Có thể phân quyền cho 2 nhóm người dùng: Admin và User theo chức năng.

2. Phân tích chức năng từ User Story

- Chức năng Đăng nhập (login), Đăng xuất (Logout)
- Chức năng Reset, Thay đổi Password.
- Chức năng Quản lý Người dùng
- Chức năng phân quyền

3. Thiết kế Data

- User(ID, UserName, Password, FullName, Remember, #IDType)
- Function(IDFunc, NameFunc, Alias)
- UserType(IDType, NameType)
- Permission(IDFunc, IDType, Total) phân quyền theo kỹ thuật bit field (1,2,4,8=15). Ktra: tổng quyền & quyền=gia trị quyền

4. Thiết kế giao diện

- Form Login
- Form Quản lý Người dùng
- Form Thay đổi Mật khẩu
- Form Phân quyền

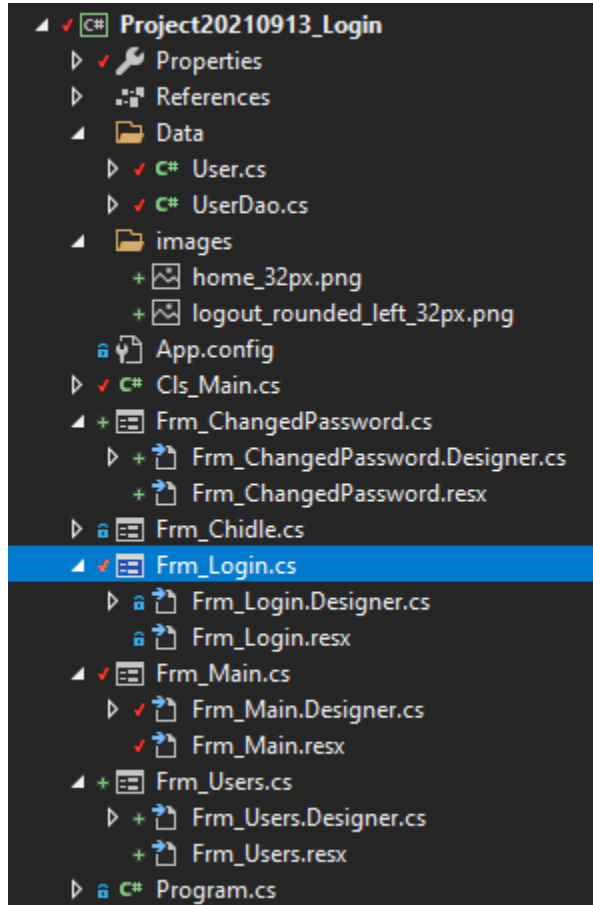
5. Yêu cầu Kiến trúc

- String, File
- Cấu trúc (if, vòng lặp)
- Collections (List, Hashtable)
- Controls (TextBox, Label, CheckBox, ComboBox, Button, Form, Menustrip, StatusStrip, MessageBox).

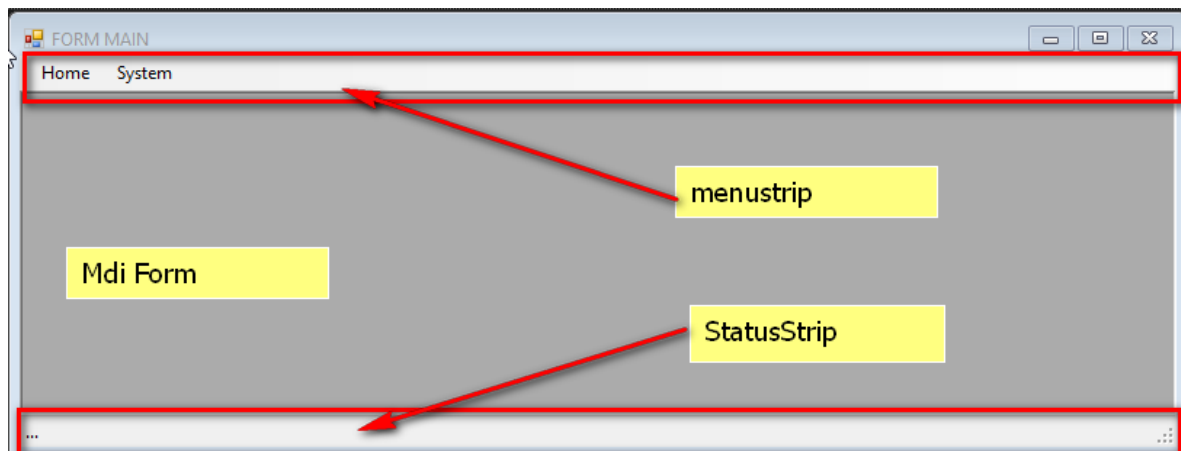
Hướng dẫn thực hiện

1. Thiết kế giao diện

Cấu trúc file và thư mục của project



Thiết kế form main



Trong đó:

STT	Tên control	Tên	Mô tả
01	Form	FrmMain	isMdiParent=true
02	MenuStrip	mnuMain	Menu của chương trình. Chứa: <ul style="list-style-type: none"> - toolStripMenuItem: Home/Logout - System/Changed Password - System/Users List
03	StatusStrip	stsMain	Thành trạng thái của form. Thêm 1 lable có tên lblErr vào thành StatusStrip

Thiết kế form login

The screenshot shows a Windows-style window titled "Frm_Login". Inside the window, there are two text input fields. The first is labeled "UserName:" and the second is labeled "Password:". Below the "Password:" field, there is a checkbox followed by the text "Remember". At the bottom of the form, there are two buttons: "Login" and "Exit". The "Login" button has a blue border, while the "Exit" button is a standard gray button.

Trong đó:

STT	Control	Tên	Mô tả
01	form	FrmLogin	Form login
02	2 label	null	Label hiển thị UserName, Password
03	TextBox	txtUserName	Textbox nhập userName
04	textBox	txtPassword	Textbox nhập password.
05	checkbox	ckbRemember	Trạng thái ghi nhớ password

06	button	btnLogin	Đăng nhập
07	button	btnExit	Button Thoát khỏi chương trình
08	StatusStrip	lblErr	Thêm label trong statusstrip để hiển thị lỗi

Thiết kế form changed password

The screenshot shows a Windows form titled "Frm_ChangedPassword". At the top, there is a label "User: N/A" with an annotation "label: lblUserName". Below this, the form is divided into two main sections: "Admin-Reset Password" and "User-Changed password".

The "Admin-Reset Password" section contains a "Choise User" dropdown menu (annotated as "ComboBox: cboUser") and a "Reset Password" button (annotated as "Button: btnResetPassword"). This section is enclosed in a "GroupBox: grbAdmin".

The "User-Changed password" section contains two password input fields: "Password New" (annotated as "TextBox: txtPasswordNewOne") and "Password New pre" (annotated as "TextBox: txtPasswordNewTwo"). Below these fields is a "Changed Password" button (annotated as "Button: btnChangedPassword"). This section is enclosed in a "GroupBox: grbUser".

2. Viết code

Tạo những lớp entity: Thep phân tích ở trên để làm chức năng đăng nhập và thay đổi mật khẩu cần tạo những class đối tượng như sau:

Tạo Class User: (Data/User.cs)

lớp này chứa các thuộc tính cần lưu trữ cho 1 user

```

17 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
public class User
{
    // field - Những thuộc tính cần lưu trữ của User
    private int id;
    private string userName;
    private string passWord;
    private bool remember;
    private int idUserType;
    //Constructor- hàm tạo cho đối tượng User
    2 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public User() { }
    0 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public User(int id, string userName, string passWord, bool remeber)
    {
        this.id = id;
        this.userName = userName;
        this.passWord = passWord;
        this.remember = remeber;
    }
    // Properties - Tạo thuộc tính cho User theo snippet: RClick vào
    // field tương ứng --> Chọn Quick Action &... (Ctrl +)--> Encapsulate Field...
    7 references | Nguyen Minh Phuc, 6 days ago | 1 author, 1 change
    public int ID { get { return id; } set { id = value; } }

    8 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public string UserName { get => userName; set => userName = value; }
    7 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public string Password { get => passWord; set => passWord = value; }
    5 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public bool Remember { get => remember; set => remember = value; }
    4 references | 0 changes | 0 authors, 0 changes
    public int IdUserType { get => idUserType; set => idUserType = value; }
    //Phương thức Lấy chuỗi định dạng của đối tượng User
    1 reference | 0 changes | 0 authors, 0 changes
    public string GetString()
    {
        return string.Format("{0},{1},{2},{3},{4}", ID, UserName, Password, Remember.ToString(), IdUserType);
    }
}

```

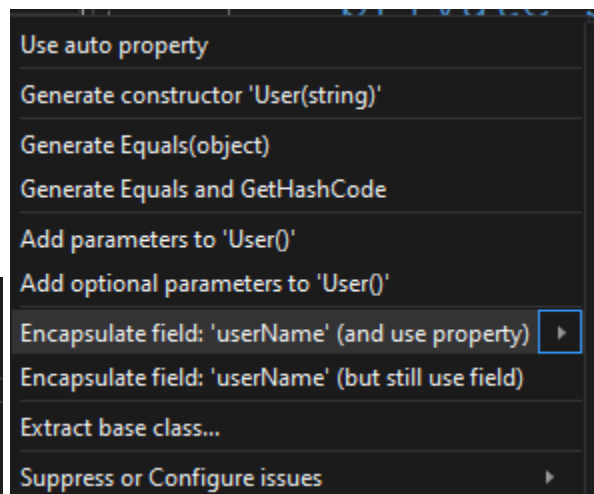
Trong đó:

- Để tạo properties cho các field (biến thành viên thực hiện: R_click vào field-->chọn theo hình.

```

// field - Những thuộc tính cần lưu trữ của User
private int id;
private string userName;
private string passWord;
private bool remem
private int idUser
//Constructor- hàm
2 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1
0 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1

```



Tạo lớp UserDao: (Data/UserDao.cs)

Lớp này thực hiện trao đổi dữ liệu từ file text (database) với phần Giao diện xử lý. Trong lớp này khai báo :

- Thuộc tính kiểu danh sách `List<User>` để lưu trữ những nội dung đọc được từ file text (listUser.ini)

- Phương thức **GetUsers(string path)** : Đọc nội dung từ file listUser.ini đưa vào List<User>.
- Phương thức **WriteUses(string path)**: Ghi nội dung từ danh sách vào file listUser.ini

```
1 reference | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
public class UserDao
{
    //Khái báo đối tượng List<User>
    public List<User> listUser;
    //Viết hàm tạo để khởi tạo đối tượng listUser.
    public UserDao()
    {
        listUser = new List<User>();
        khoá code
    }

    //Phương thức ghi danh sách listUser vào trong file text theo đường dẫn file Path
    public void WriteUsers(string path)
    {
        //Sử dụng cấu trúc using để khai báo đối tượng FileStream
        using (FileStream fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Write, FileShare.Write))
        {
            //Sử dụng class StreamWriter để làm việc với file text (ini, txt)
            using (StreamWriter sw = new StreamWriter(fs))
            {
                foreach (User item in listUser)
                {
                    //Phương thức WriteLine dùng để ghi một đoạn string lên file text. sau đó xuống dòng
                    sw.WriteLine(string.Format("{0},{1},{2},{3},{4}", item.ID, item.UserName, item.PassWord, item.Remember.ToString(),
                        item.IdUserType));
                }
            }
        }
    }
}
```

```
1 reference | 0 changes | 0 authors, 0 changes
public void GetUsers(string path)
{
    using (FileStream fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read, FileShare.Read))
    {
        //Dùng Lớp StreamReader để đọc file theo đường dẫn được khai báo trong FileStream (phía trên).
        using (StreamReader sr = new StreamReader(fs))
        {
            string line = string.Empty;
            User user;
            //Readline() phương thức đọc từng dòng trong file.
            //Vòng Lặp while cho phép thực hiện việc này liên tục khi đọc được giá trị null
            //Sau khi đọc xong 1 dòng, data sẽ được lưu vào trong biến line.
            while ((line = sr.ReadLine()) != null)
            {
                //1,admin,admin,True,1
                //Kiểm tra line có giá trị hay không (null hoặc empty)
                if (!string.IsNullOrEmpty(line))
                {
                    //Dùng hàm Split('char') để tách chuỗi line thành mảng theo ký tự (char) ngăn cách
                    string[] userArray = line.Split(',');
                    //Tạo một đối tượng User và thêm giá trị từ mảng đọc được vào
                    user = new User();
                    user.ID = Convert.ToInt32(userArray[0]);
                    user.UserName = userArray[1];
                    user.PassWord = userArray[2];
                    user.Remember = Convert.ToBoolean(userArray[3].ToString());
                    user.IdUserType = Convert.ToInt32(userArray[4]);
                    //Them User vào trong danh sách listUser.
                    listUser.Add(user);
                }
            }
        }
    }
}
```

Tạo lớp Cls_Main:

Chứa những biến dùng chung trong project, những biến này được dùng theo dạng static (biến toàn cục). Tạo Class trong project và đặt tên là **Cls_Main.cs**

```

namespace Project20210913_Login
{
    20 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public class Cls_Main
    {
        //Biến lưu User sau khi đăng nhập thành công
        public static User _staticUser=new User();
        //Đối tượng danh sách User danh sách này được khởi tạo khi mở chương trình. giá
        trị được đọc từ file text.
        public static List<User> _listUser = new UserDao().listUser;
        //Biến lưu đường dẫn file listUser.ini trong thư mục cài đặt của project
        public static string pathfile = string.Format(@"{0}\listUser.ini",
            Application.StartupPath);
    }
}

```

Chú ý:

- **Application.StartupPath**: là thuộc tính lấy đường dẫn đến thư mục bin\debug của project. Class này trong namespace: **System.Windows.Forms**;
- **String.format**: Phương thức này là phương thức định dạng và trả về một chuỗi sau khi định dạng.
- **List<User>**: Kiểu dữ liệu danh sách, chứa danh sách những item kiểu User.

Viết code cho form main

Tại giao diện form Main sau khi thiết kế xong, tiến hành mở sự kiện LoadForm của Form này (thực hiện bằng Double Click vào thanh tiêu đề của form), sau đó tiến hành code cho sự kiện này. (giống hình bên dưới)

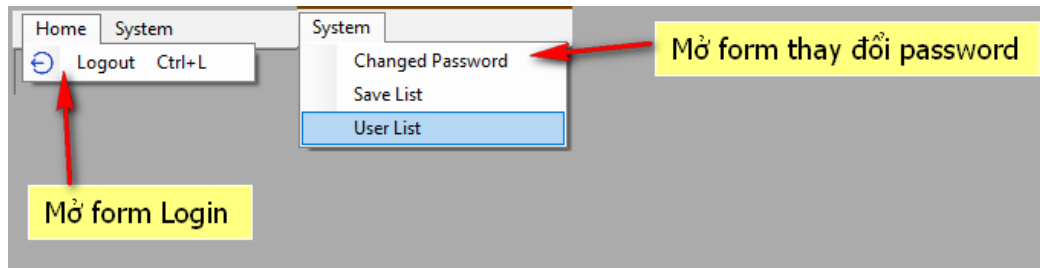
```

3 references | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
public partial class Frm_Main : Form
{
    1 reference | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    public Frm_Main()
    {
        InitializeComponent();
    }
    //Tạo đối tượng của lớp UserDao.
    UserDao _userDao;
    //Sự kiện load của form. sự kiện này sẽ được thực hiện khi form main được mở lên.
    1 reference | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
    private void Frm_Main_Load(object sender, EventArgs e)
    {
        //Khởi tạo đối tượng UserDao để tiến hành gọi phương thức lấy danh sách User trong file listUser.ini
        _userDao = new UserDao();
        _userDao.GetUsers(Cls_Main.pathfile);
        //Gán danh sách vừa được đọc từ file vào cho biến toàn cục _listUser trong Cls_Main.
        //giá trị này dc dùng trong các form của chương trình.
        Cls_Main._listUser = _userDao.listUser;

        //Thực hiện mở form Login
        Frm_Login frm_Login = new Frm_Login();
        frm_Login.ShowDialog();
        //Sau khi đăng nhập thành công sẽ hiển thị thông tin người đăng nhập trên lblUserInfor tại StatusStrip
        lblUserInfor.Text = string.Format("Login by {0}-{1}", Cls_Main._staticUser.ID, Cls_Main._staticUser.UserName);
    }
}

```

Code cho sự kiện click của menu trong menustrip



Tạo sự kiện này bằng cách double click vào menuItem Logout trên menu

```
1 reference | 0 changes | 0 authors, 0 changes
private void mnuLogout_Click(object sender, EventArgs e)
{
    Cls_Main._staticUser = null;
    Frm_Login frm_Login = new Frm_Login();
    frm_Login.ShowDialog();
}
```

Khi logout cần xoá thông tin User được lưu trong lần đăng nhập trước

Nhập code vào sự kiện Click

tạo sự kiện Click cho MenuItem ChangedPassword trên menu

```
1 reference | 0 changes | 0 authors, 0 changes
private void mnuChangedPassword_Click(object sender, EventArgs e)
{
    //Khái báo đối tượng form Frm_ChangedPassword
    Frm_ChangedPassword frm_ChangedPassword = new Frm_ChangedPassword();
    //Xác định form cha của form ChangedPassword là Form hiện tại (FrmMain)
    //Để mở form này theo hình thức form cha, form con
    frm_ChangedPassword.MdiParent = this;
    //Xác định vị trí hiển thị ngay giữa màn hình của form ChangedPassword khi hiển thị
    frm_ChangedPassword.StartPosition = FormStartPosition.CenterScreen;
    //Sử dụng phương thức Show() trong trường hợp này, không được dùng ShowDialog
    frm_ChangedPassword.Show();
}
```

Thực hiện code để mở form Frm_ChangedPassword

Viết code cho chức năng đăng nhập

Với giao diện form FrmLogin được thiết kế ở trên ta tiến hành thực hiện các dòng code để xử lý những yêu cầu trên form FrmLogin như sau:


```

1 reference | Nguyễn Minh Phúc, 7 days ago | 1 author, 1 change
public Frm_Login()
{
    InitializeComponent();
}

//Khai báo biến lưu danh sách User, Biến này sẽ nhận giá trị từ biến toàn cục trong lớp Cls_Main khi Load chương trình
//Biến này dùng để lưu trữ danh sách User và thực hiện thao tác kiểm tra userName và Password trên danh sách này
private List<User> listUser;
//Sự kiện Load của Form Login
1 reference | Nguyễn Minh Phúc, 7 days ago | 1 author, 1 change
private void Frm_Login_Load(object sender, EventArgs e)
{
    //Khởi tạo giá trị cho đối tượng listUser
    listUser = Cls_Main._listUser;
}

```

Sự kiện Click Của button Login:

```

1 reference | Nguyễn Minh Phúc, 7 days ago | 1 author, 1 change
private void btnLogin_Click(object sender, EventArgs e)
{
    // void Frm_Login.btnLogin_Click(object sender, EventArgs e)
    //Trước khi kiểm tra username và password có trùng khớp với user trong listUser hay không phải tiến hành kiểm tra giao diện trước
    //Kiểm tra đã nhập text trên 2 textbox txtUserName và txtPassword hay chưa.
    if(!string.IsNullOrEmpty(txtUserName.Text))
    {
        if (!string.IsNullOrEmpty(txtPassWord.Text))
        {
            //kiem tra xem txt vừa nhập có trùng với username và pass hay không--> if...else...
            //Viết phương thức kiểm tra đăng nhập với 2 tham số truyền vào là text trên 2 textbox txtUserName và txtPassword
            if( KiemTraDangNhap(txtUserName.Text, txtPassWord.Text))
            {
                //Khi kiểm tra trùng khớp username và password thì sẽ cho tắt form Login và form Main sẽ được sử dụng bình thường
                this.Close();
            }
            else
            {
                //Nếu kiểm tra không trùng khớp xuất thông báo trên lblErr trên StatusStrip của form Login
                lblErr.Text = "UserName or Password wrong";
            }
        }
    }
}

```

Phương thức Kiểm tra đăng nhập (KiemTraDangNhap(string UserName,string Password)
Phần chữ màu xanh (green) là phần ghi chú khi nhập không cần nhập. Phần này Thầy ghi chú để các em hiểu code.

```

1 reference | Nguyễn Minh Phúc, 7 days ago | 1 author, 1 change
/// <summary>
/// Phương thức Kiểm tra đăng nhập với 2 tham số
/// Trong phương thức để kiểm tra, sử dụng 1 vòng lặp để lặp trên từng user của danh sách User.
/// Trong phương thức này sử dụng Foreach để duyệt list.
/// </summary>
/// <param name="userName">Text trên txtUserName</param>
/// <param name="passWord">Text trên txtPassword</param>
/// <returns>Trả về True nếu trùng khớp, False nếu không trùng khớp UserName và Password</returns>
1 reference | Nguyễn Minh Phúc, 7 days ago | 1 author, 1 change
private bool KiemTraDangNhap(string userName, string passWord)
{
    foreach (User item in listUser)
    {
        //Kiểm tra nếu trùng cả Username và Password thì sẽ return True đồng thời lưu lại thông tin User đăng nhập vào trong biến tĩnh
        //StaticUser trong lớp Cls_Main. Ngược lại trả về False
        if (item.UserName.Equals(userName) && item.Password.Equals(passWord))
        {
            item.Remember = ckbRemember.Checked;
            Cls_Main._staticUser = item;
            return true;
        }
    }
    return false;
}

```

Phần code cho button Exit

```

1 reference | Nguyễn Minh Phúc, 7 days ago | 1 author, 1 change
private void btnExit_Click(object sender, EventArgs e)
{
    //Để thoát khỏi chương trình các form đang mở sẽ đóng lại hết (kết thúc chương trình)
    //Thực hiện việc này sử dụng phương thức Exit() trong Lớp Application.
    Application.Exit();
}

```

Phần code kiểm tra việc nhớ mật khẩu của user khi thực hiện thao tác nhập tên User

Việc nhớ này được lưu trong biến remember của đối tượng User Khi Remember là true thì sau khi nhập tên user trong txtUserName thì password sẽ tự động hiển thị trong txtPassword.

tạo sự kiện Leave của txtUserName: (chọn txtUserName trên giao diện, vào Properties, chọn tab Event (dấu sét) và tìm đến sự kiện tên là Leave) --> Double Click vào--> sẽ tạo 1 sự kiện trong code

```
1 reference | Nguyen Minh Phuc, 7 days ago | 1 author, 1 change
private void txtUserName_Leave(object sender, EventArgs e)
{
    foreach (User item in listUser)
    {
        if (item.UserName==txtUserName.Text && item.Remember)
        {
            ckbRemember.Checked = true;

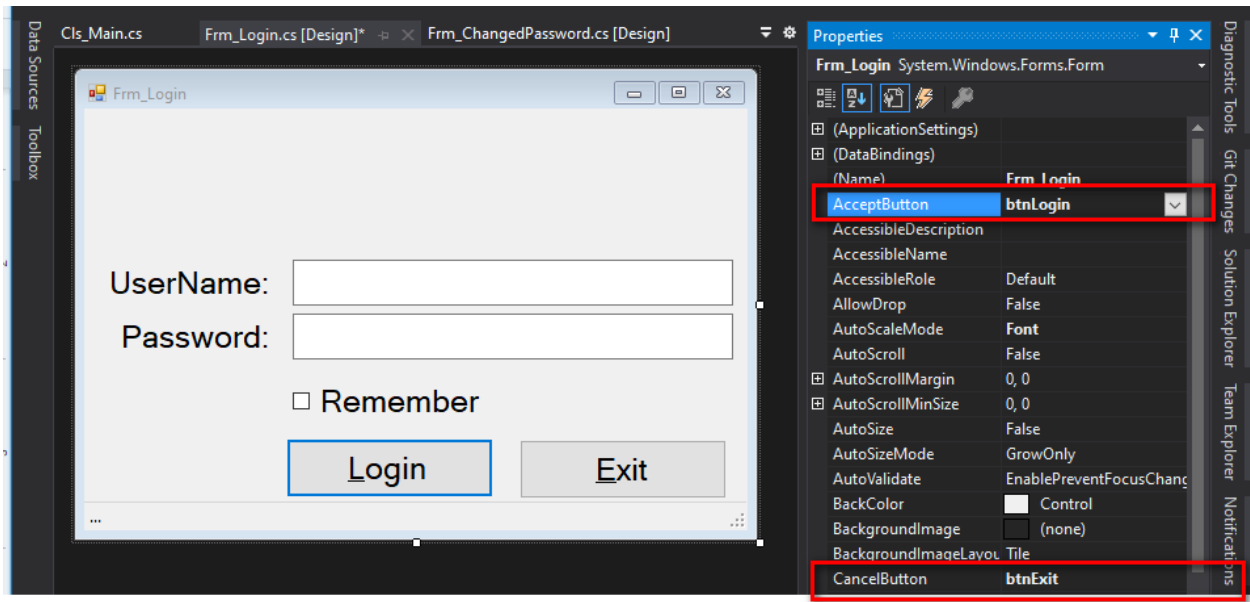
            txtPassWord.Text = item.PassWord;
            btnLogin.Focus();
        }
    }
}
```

Nhập code trong khung đỏ vào trong sự kiện Leave

Nếu userName trùng và biến Remember =true thì sẽ tự động hiện password

Chú ý. Nếu muốn chọn phím tắt cho button login và Exit. tiến hành như sau

- Chọn form Login trên giao diện. → Mở Properties → Tìm đến thuộc tính AcceptButton, CancelButton.



AcceptButton: cho phép bạn dùng phím Enter để thực hiện lệnh Click trên button
 CancelButton: cho phép bạn dùng phím Esc để thực hiện lệnh Click trên button

Viết code cho chức năng đăng Xuất:

Chức năng này thực hiện việc mở lại form FrmLogin.

Viết code cho chức năng thay đổi password

Khai báo những biến thành viên trong form Frm_ChangedPassword

```

4 references | 0 changes | 0 authors, 0 changes
public partial class Frm_ChangedPassword : Form
{
    1 reference | 0 changes | 0 authors, 0 changes
    public Frm_ChangedPassword()
    {
        InitializeComponent();
    }
    //Khai báo một số biến để điều khiển luồng của chương trình.
    //Biến kiểu danh sách List<User> lưu danh sách User hiện tại. Danh sách này được lấy từ danh sách listUser trong
    lớp Cls_Main
    List<User> _listUser;
    //Biến lưu trạng thái Load của comboBox, khi đổ data vào comboBox User cboUsers. false là chưa load xong, true: đã
    load xong.
    bool statusLoad = false;

    //Biến IdUserType để lưu loại User : 1 là Admin; 2 là User
    private int IdUserType = 0;
    //Lưu ID của User hiện tại, User đăng nhập vào chương trình
    int idUser = 0;
    //Khai báo đối tượng UserDao để thực hiện thao tác Lưu thay đổi của danh sách listUser vào file listUser.ini
    UserDao userDao;

```

Sự kiện load của Form Frm_ChangedPassword

Khi tiến hành load Form lên sẽ khởi tạo một số giá trị như sau:

```

/// <summary>
/// Khi load form Frm_ChangedPassword nếu là Tài khoản Admin sẽ cho hiển thị cả phần Reset Password,
/// Còn nếu là loại tại khoản User chỉ cho phép hiển thị phần ChangedPassword (xem trong giao diện)
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 reference | 0 changes | 0 authors, 0 changes
private void Frm_ChangedPassword_Load(object sender, EventArgs e)
{
    //Khởi tạo đối tượng UserDao được khai báo trước đó.
    userDao = new UserDao();
    //lấy loại người dùng từ biến toàn cục trong lớp Cls_Main là Cls_Main._staticUser.
    IdUserType = Cls_Main._staticUser.IdUserType;
    //Lấy ID của người dùng
    idUser = Cls_Main._staticUser.ID;
    //hiển thị id và tên người dùng lên lblUserName
    lblUserName.Text = string.Format("{0}-{1}", idUser.ToString(), Cls_Main._staticUser.UserName.ToUpper());
    //kiểm tra loại người dùng để hiển thị đúng kịch bản giao diện đưa ra
    switch (IdUserType)
    {
        case 1://admin
            grbAdmin.Enabled = true;//thuộc tính mất focus/co focus
            // grbAdmin.Visible = false;/**/Thuộc tính an/hien
            //Lấy danh sách User đổ vào ComboBox cboUser
            _listUser = Cls_Main._listUser;
            LoadComboUser();
            break;
        case 2://User
            grbAdmin.Enabled = false;
            break;
        default:
            grbAdmin.Enabled = false;
            break;
    }
}

```

Phương thức LoadComboUser()

```

1 reference | 0 changes | 0 authors, 0 changes
private void LoadComboUser()
{
    //Lấy data trong danh sách User (listUser) gán cho thuộc tính datasource của comboBox
    cboUser.DataSource = _listUser;
    //Gán giá trị cho thuộc tính DisplayMember. giá trị của thuộc tính này sẽ được nhìn thấy trên ComboBox.
    cboUser.DisplayMember = "UserName";\
    //Gán giá trị cho thuộc tính ValueMember của comboBox. giá trị này sẽ được lấy ra khi chọn comboBox. qua
    thuộc tính SelectedValue.
    cboUser.ValueMember = "ID";
    //Thực hiện chuyển index về -1 và hiển thị text trên cboUser.
    cboUser.SelectedIndex = -1;
    cboUser.Text = "--- Select user ---";
    //Gán trạng thái Load Xong comboBox
    statusLoad = true;
}

```

Sự kiện khi thực hiện chọn item trong cboUser: sự kiện này sẽ được thực hiện khi load giao diện và khi chọn item trên comboBox.

Để sử dụng cần kiểm tra index bằng thuộc tính SelectedIndex và kiểm tra trạng thái load của ComboBox bằng biến StatusLoad (biến được khai báo ở đầu).

Trong sự kiện này thực hiện 1 việc là lấy id của User được chọn.

1 reference | 0 changes | 0 authors, 0 changes

```
private void cboUser_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cboUser.SelectedIndex >= -1 && statusLoad)
    {
        //idUser = (int)cboUser.SelectedValue;
        //idUser = int.Parse(cboUser.SelectedValue.ToString());
        idUser = Convert.ToInt32(cboUser.SelectedValue);
        //MessageBox.Show(cboUser.SelectedValue.ToString());
        //inboxing -- uninboxing
    }
}
```

Sự kiện ResetPassword. ResetPassword thực hiện việc cấp lại mật khẩu cho User chọn trong comboBox (cboUser). Sẽ đổi lại password cho user theo tên của UserName. Sau đó tiến hành lưu lại vào trong file listUser.

1 reference | 0 changes | 0 authors, 0 changes

```
private void btnResetPassword_Click(object sender, EventArgs e)
{
    foreach (User item in Cls_Main._listUser)
    {
        if (item.ID == idUser)
        {
            item.Password = item.UserName;
            userDao.listUser = Cls_Main._listUser;
            userDao.WriteUsers(Cls_Main.pathfile);
        }
    }
}
```

Sự kiện ChangedPassword (button). Yêu cầu nhập Password mới 2 lần và tiến hành cập nhật password cho user.

```

1 reference | 0 changes | 0 authors, 0 changes
private void btnChangedPassword_Click(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(txtPasswordNewOne.Text))
    {
        if (!string.IsNullOrEmpty(txtPasswordTwo.Text))
        {
            if (txtPasswordNewOne.Text.Equals(txtPasswordTwo.Text))
            {
                if (ChangedPassword(txtPasswordNewOne.Text))
                {
                    MessageBox.Show("Changed pass successfull", "Information", MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
                }
                else
                {
                    MessageBox.Show("Changed password Error", "Error", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("Not Same", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("Not input Password 2", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
    else
    {
        MessageBox.Show("Not input Password 1", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

Phương thức ChangedPassword(string userName)

```

1 reference | 0 changes | 0 authors, 0 changes
private bool ChangedPassword(string passNew)
{
    foreach (User item in Cls_Main._listUser)
    {
        if (item.ID == idUser)
        {
            item.PassWord = passNew;
            userDao.listUser = Cls_Main._listUser;
            userDao.WriteUsers(Cls_Main.pathfile);
            return true;
        }
    }
    return false;
}

```

Thực hiện chức năng quản lý tài khoản người dùng (2021-09-27)

Thiết kế giao diện