# **Thực hành chức năng đăng nhập**

---o0o---

## **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, Xoá thông tin của tài khoản)
* Có thể phân quyền cho 2 nhóm người dùng: Admin và User theo chức năng.

## **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

## **Thiết kế Data**

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

## **Thiết kế giao diện**

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

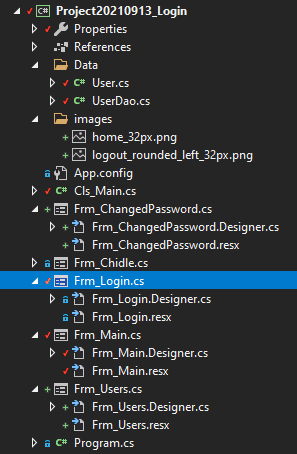
## **Yêu cầu Kiến thức**

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

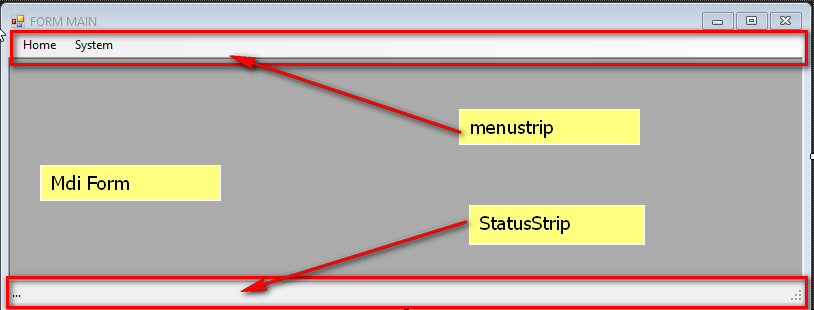
Hướng dẫn thực hiện

## Thiết kế giao diện

Cấu trúc file và thư mục của project



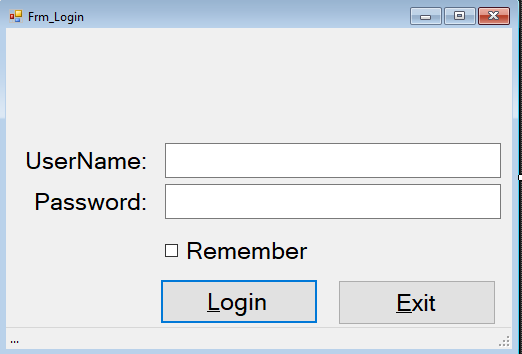
Thiết kế form main



Trong đó:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên control | Tên | Mô tả |
| 01 | Form | FrmMain | isMdiParent=true |
| 02 | MenuStrip | mnuMain | Menu của chương trình. Chứa:   * toolStripmenuItem: Home/Logout * System/Changed Password * System/Users List |
| 03 | StatusStrip | stsMain | Thành trang thái của form.  Thêm 1 lable có tên lblErr vào thành StatusStrip |

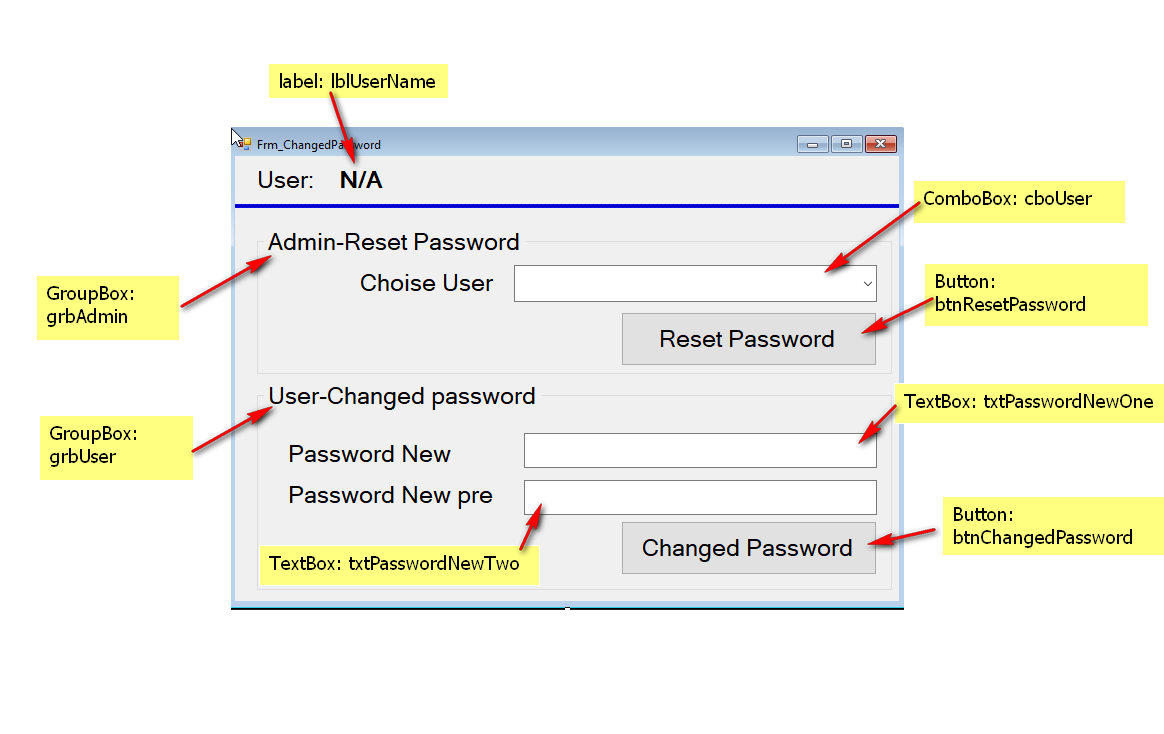
Thiết kế form login



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 nghi 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

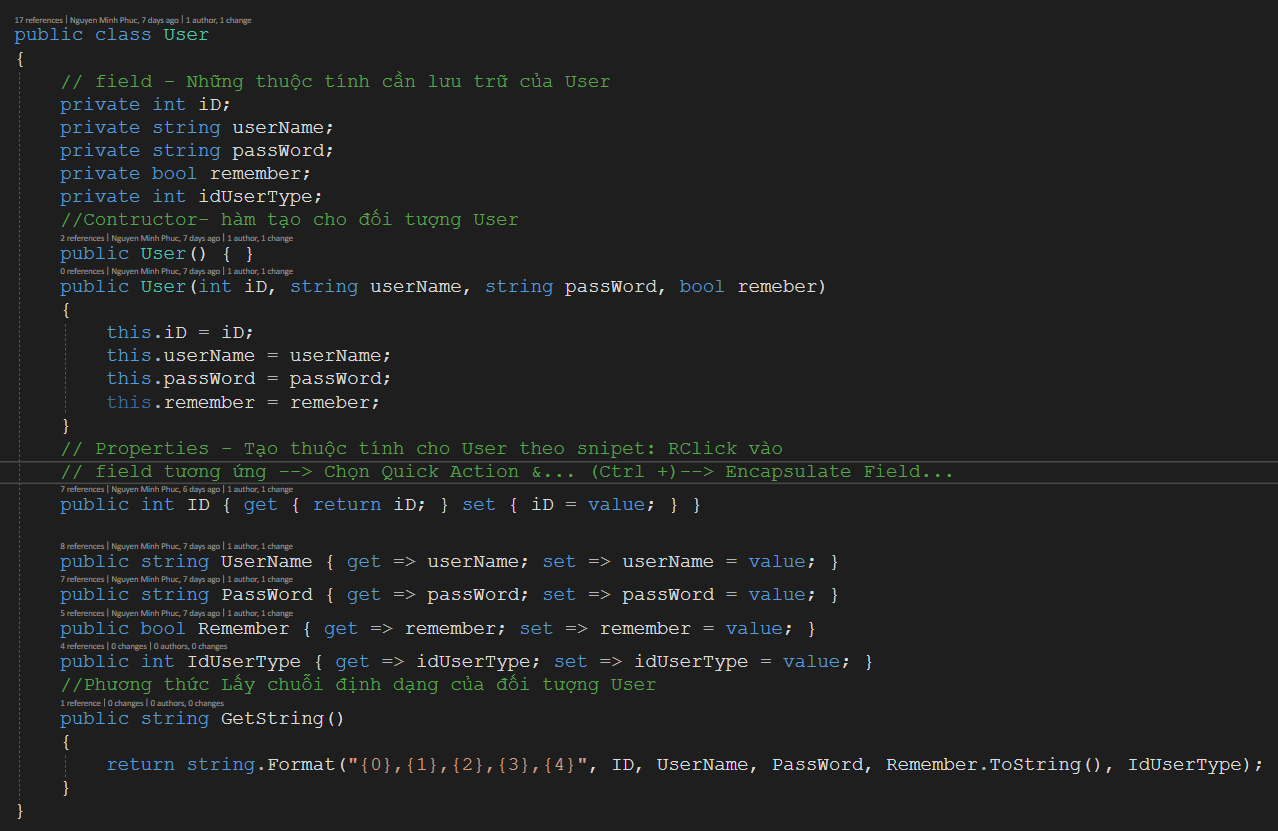


## Viết code

**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:

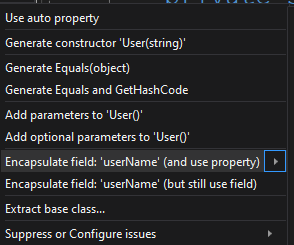
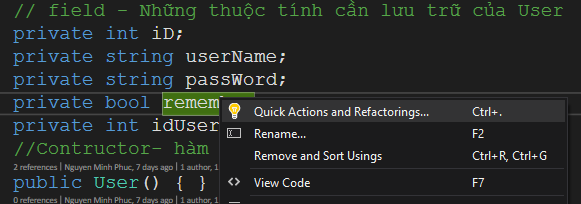
### **Tạo Class User:** (Data/User.cs)

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



Trong đó:

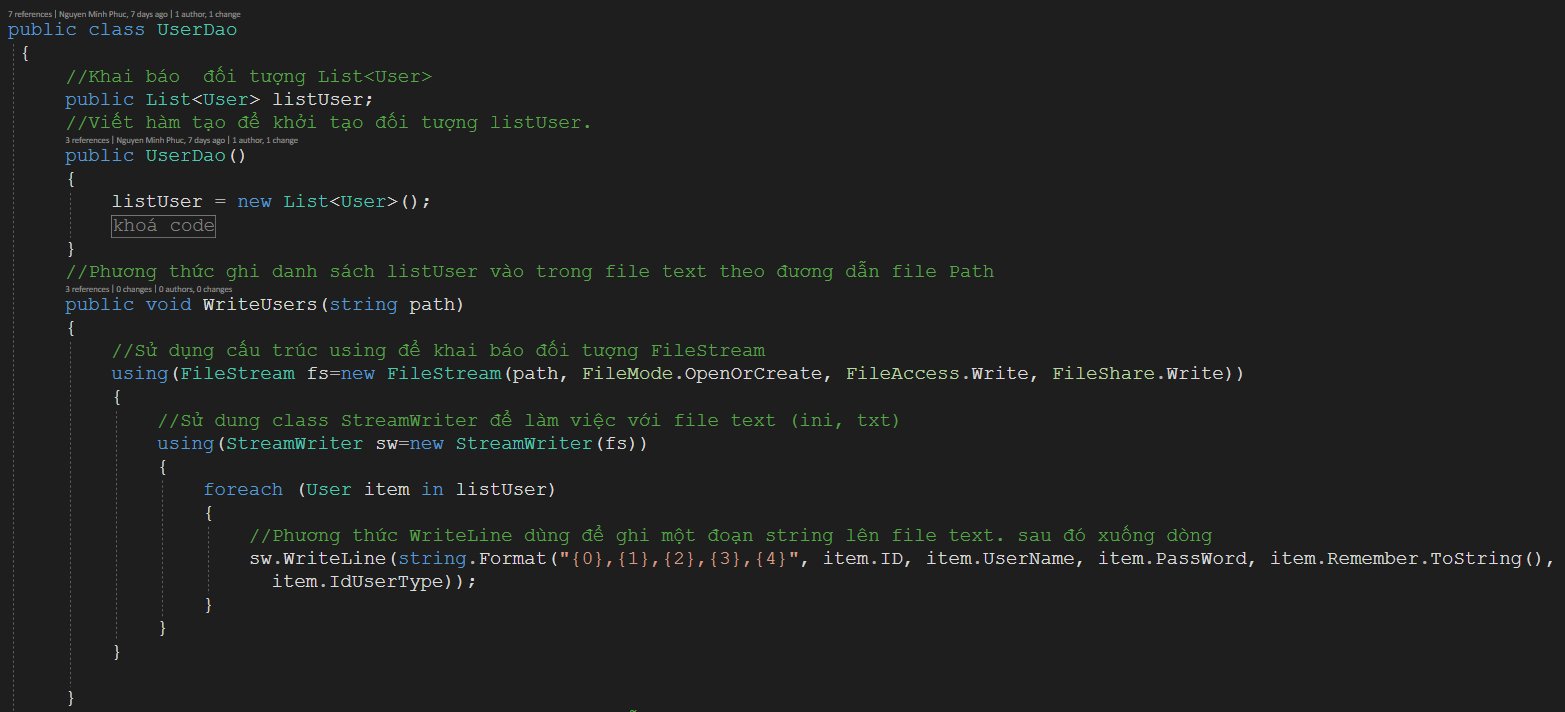
* Để tạo properties cho các field (biến thành viên thực hiện: R\_click vào field-->chọn theo hình.

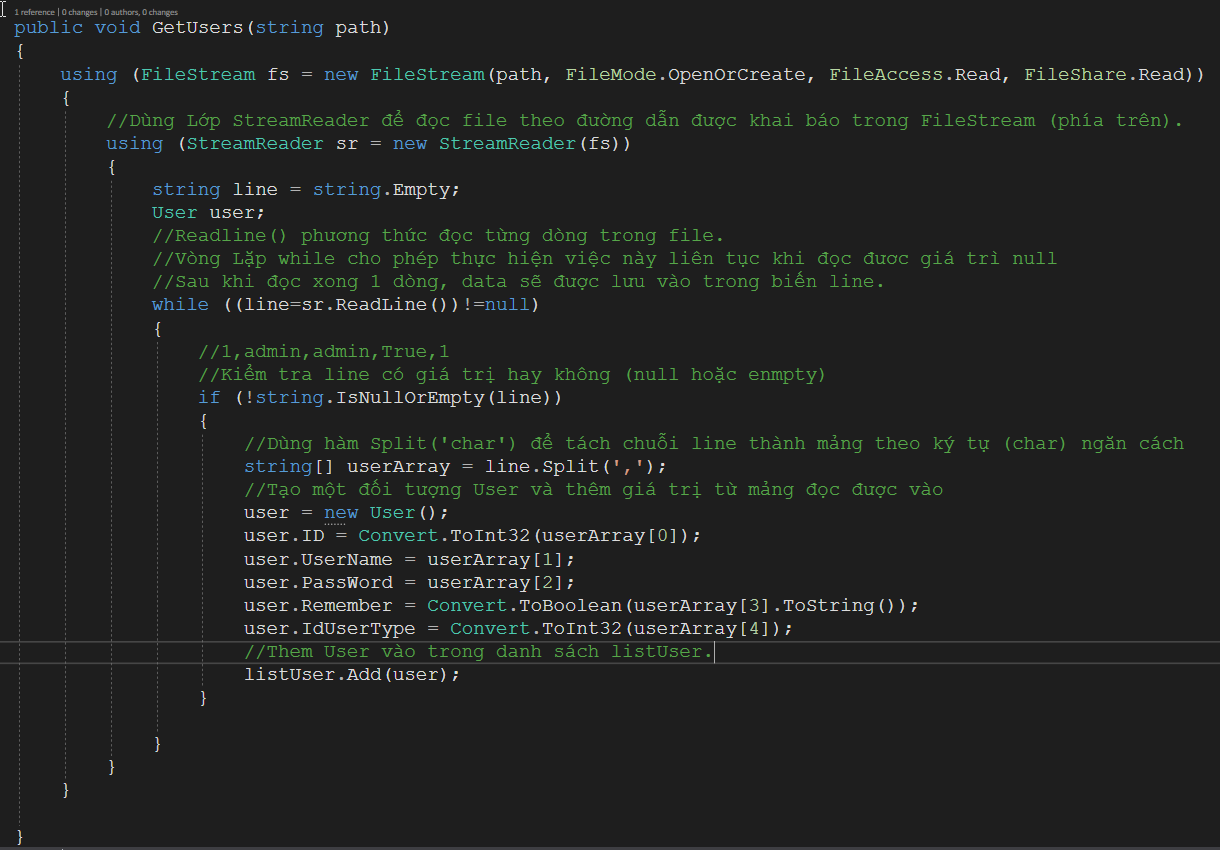


#### **Tạo lớp UserDao: (Data/UserDao.cs)**

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

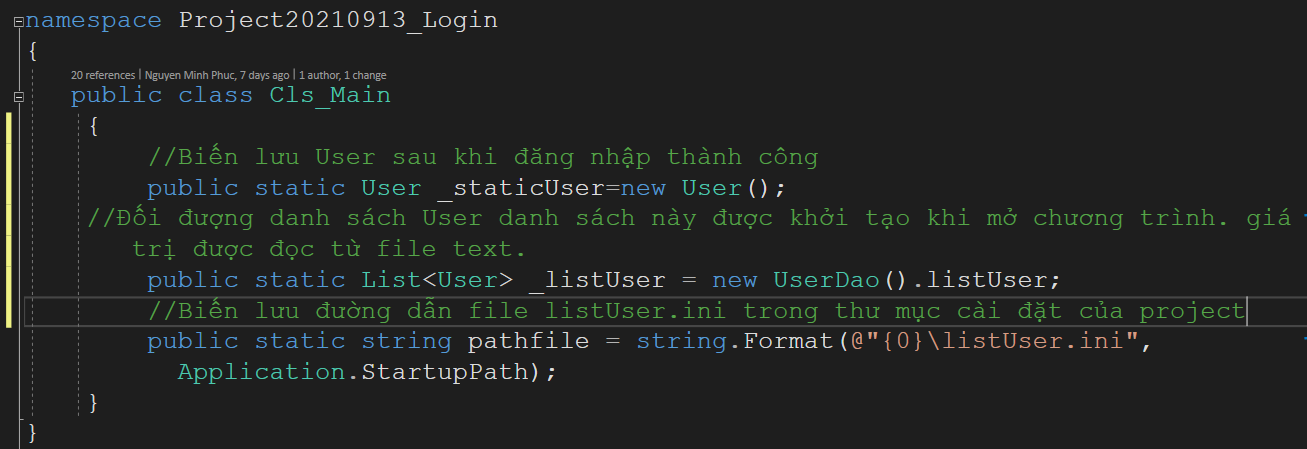
* Thuộc tính kiểu danh sách List<User> để lưu trữ những nội dung đọc được từ file text (listUser.ini)
* Phương thức GetUsers(string path) : Đọc nội dung từ file listUser.ini đưa vào List<User>,
* Phương thức WriteUses(string path): Ghi nội dung từ danh sách vào file listUser.ini





#### **Tạo lớp Cls\_Main:**

Chứa nhưng 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

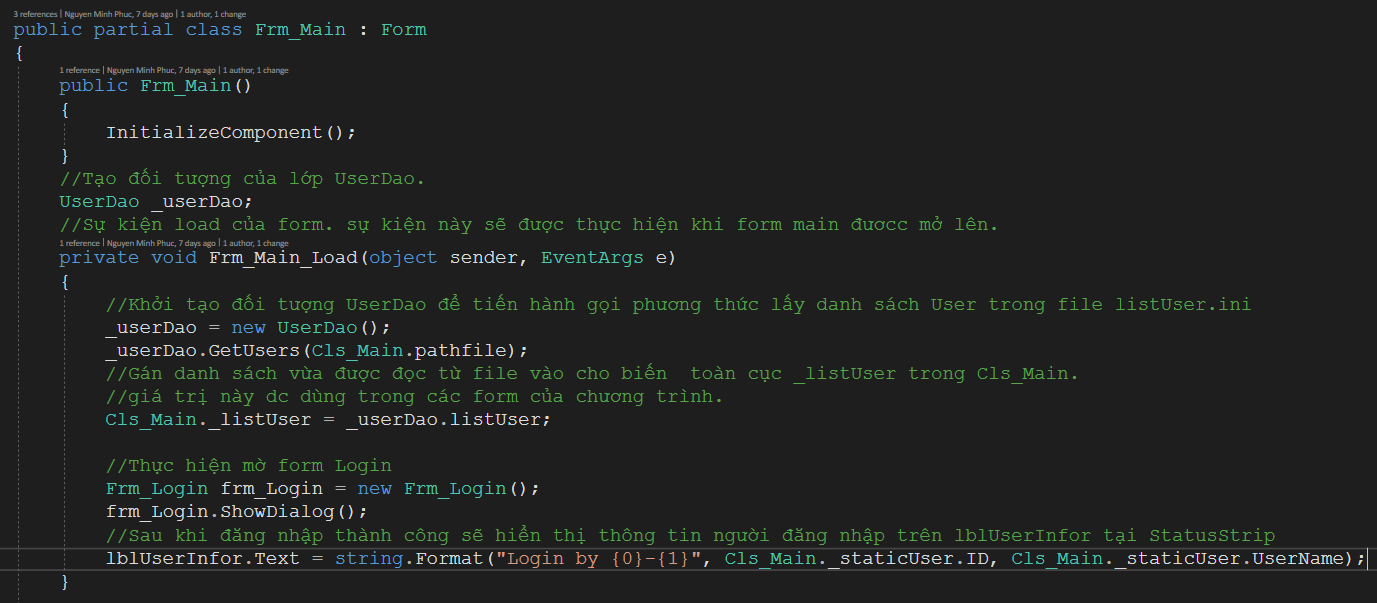


Chú ý:

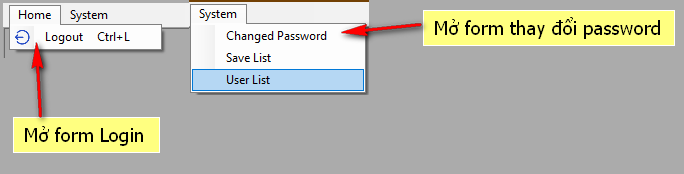
* Application.StartupPath: là thuộc tính lấy đường dẫn đến thư mục bin\debug của project. Class này trong namespace: System.Windows.Forms;
* **String.format**: Phương thức này là phương thức định dạng và trả về một chuỗi sau khi định dạng.
* List<User>: Kiểu dữ liệu danh sách, chứa danh sách những item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

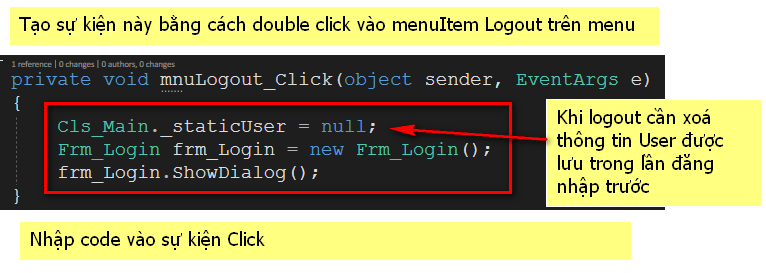
#### **Viết code cho form main**

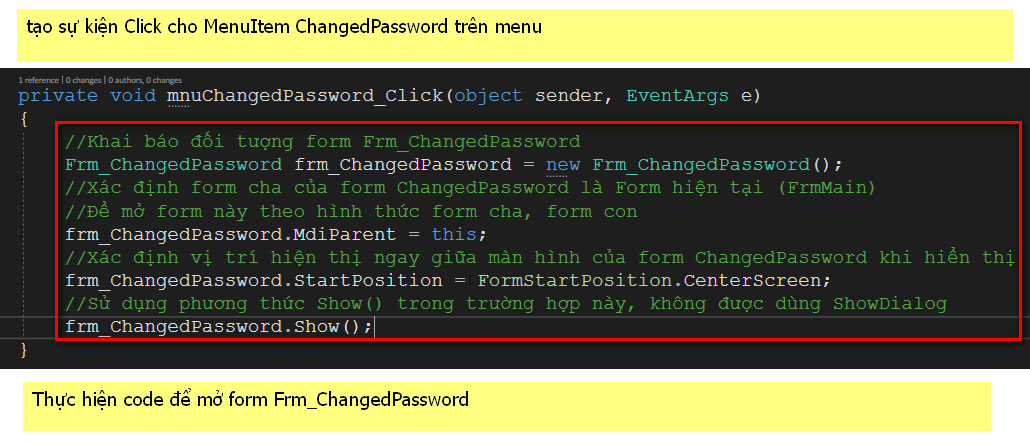
Tại giao diện form Main sau khi thiết kế xong, tiến hành mở sự kiện LoadForm của Form này (thực hiện bằng Double Click vào thanh tiêu đề của form), sau đó tiến hành code cho sự kiện này. (giống hình bên dưới)



Code cho sự kiện click của menu trong menustrip

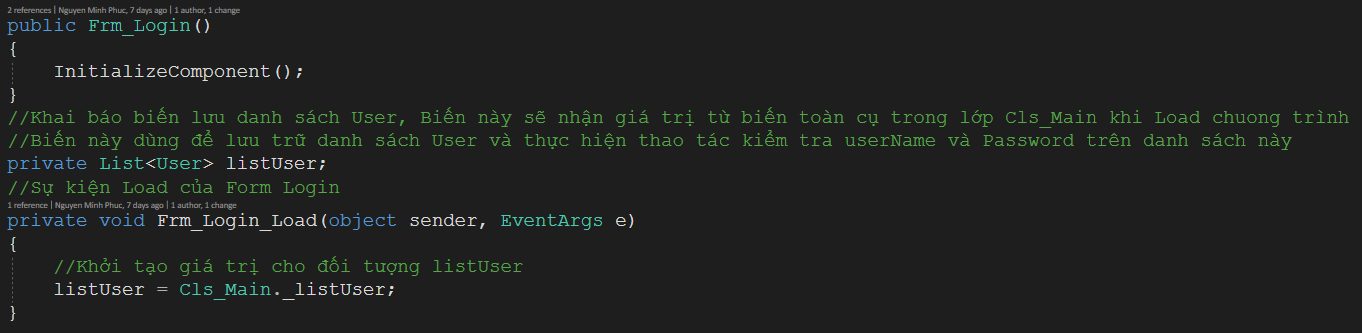




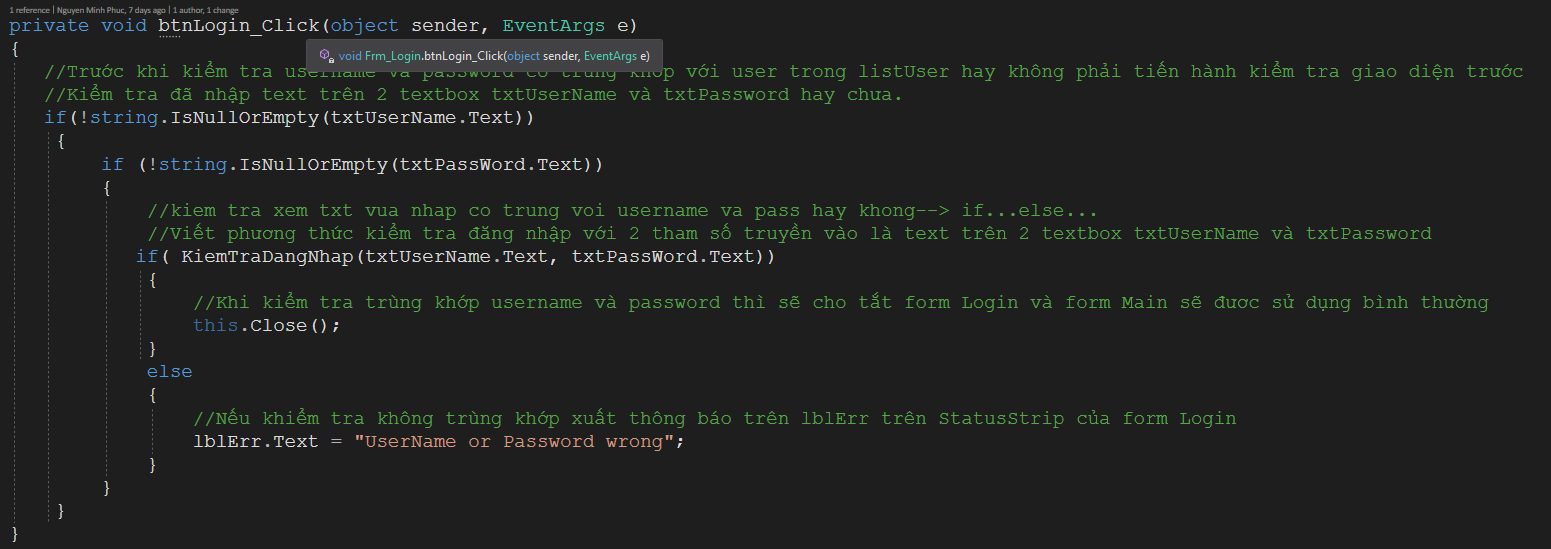


#### **Viết code cho chức năng đăng nhập**

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:

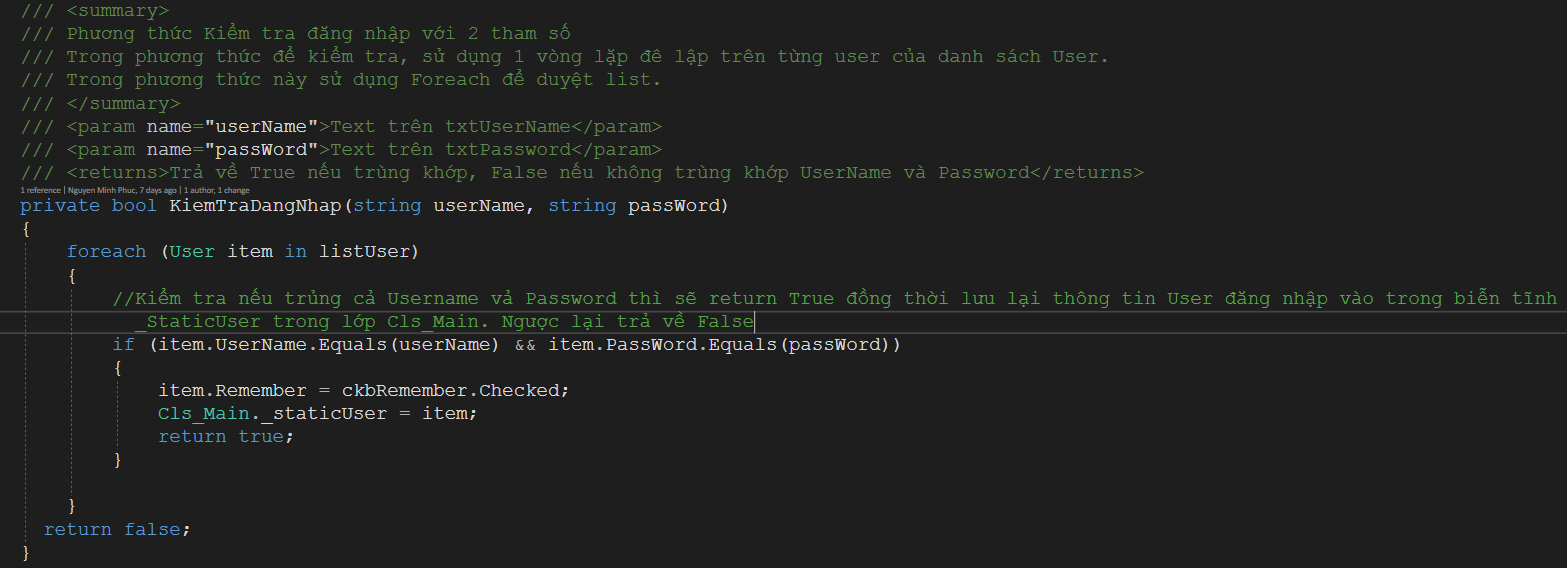


Sự kiện Click Của button Login:

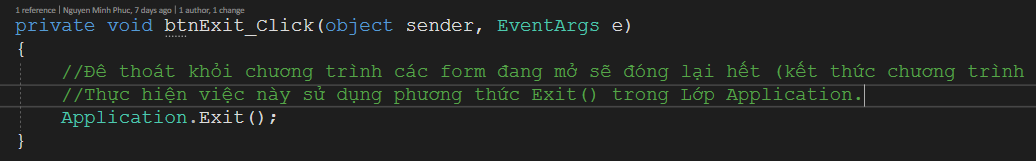


Phương thưc Kiểm tra đăng nhập (KiemTraDangNhap(string UserName,string Password)

Phần chữ màu xanh (green) là phần ghi chú khi nhập không cần nhập. Phần này Thầy ghi chú để các em hiểu code.

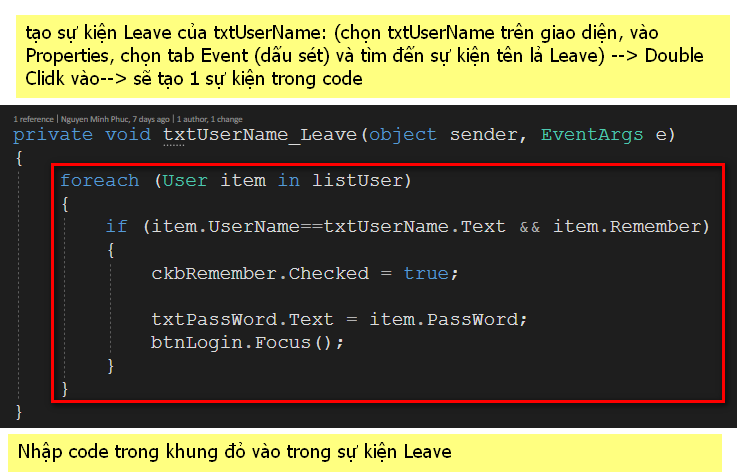


Phần code cho button Exit



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

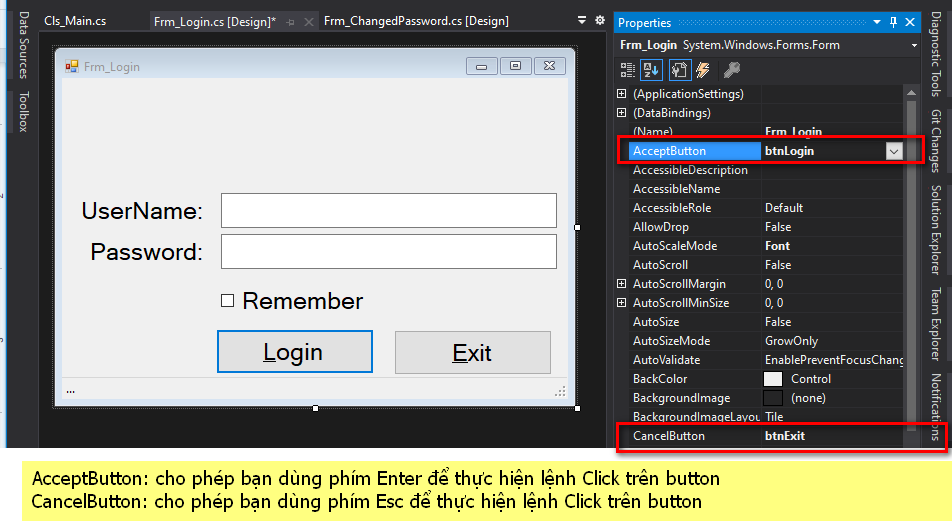
Việc nhớ này được lưu trong biên remember của đối tượng User Khi Remember là true thì sau khi nhập tên user trong txtUserName thì password sẽ tự động hiển thị trong txtPassword.



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.

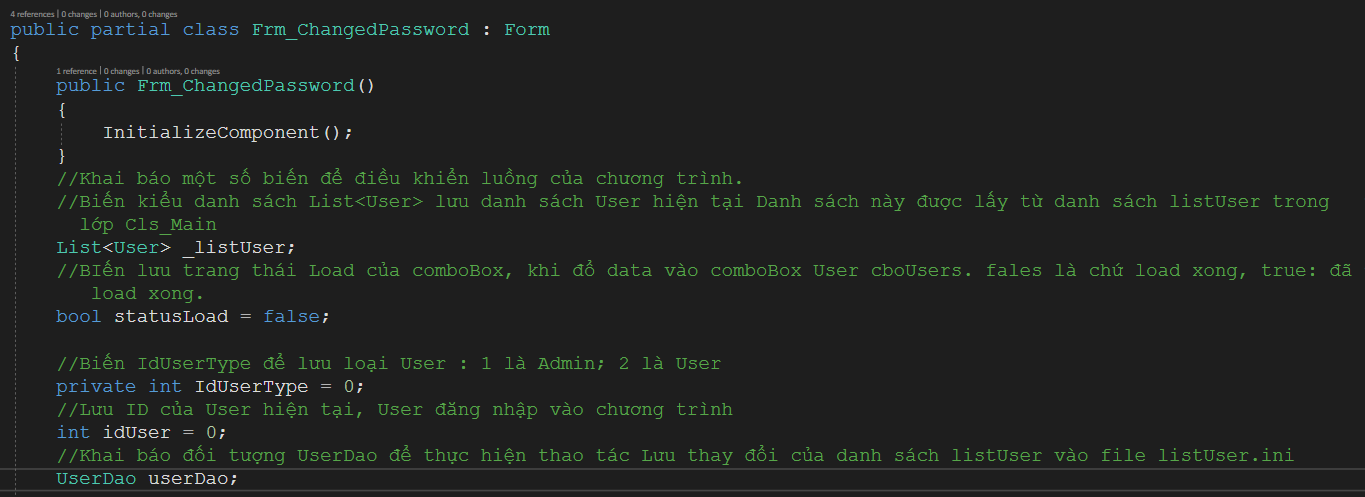


#### **Viết code cho chức năng đăng Xuất:**

Chức năng này thực hiện việc mở lại form FrmLogin.

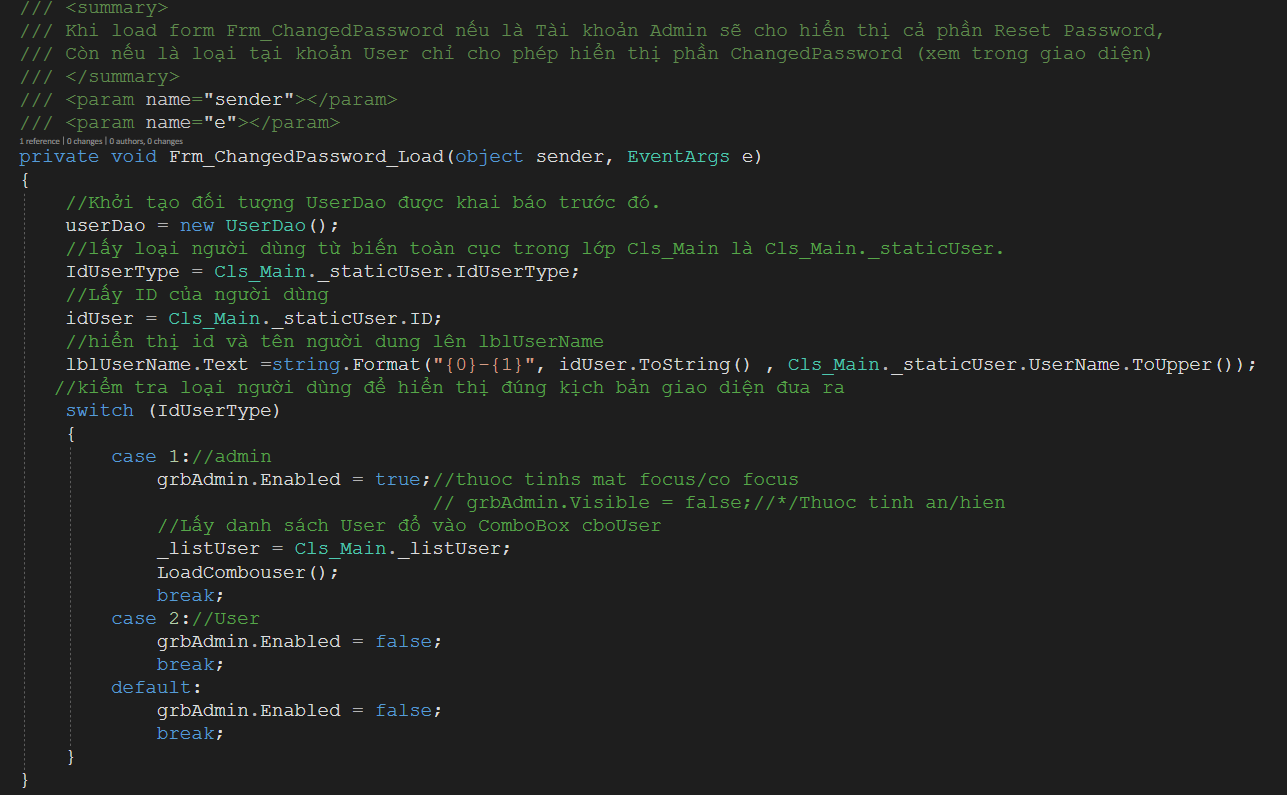
#### **Viết code cho chức năng thay đổi password**

Khai báo những biến thành viên trong form Frm\_ChangedPassword

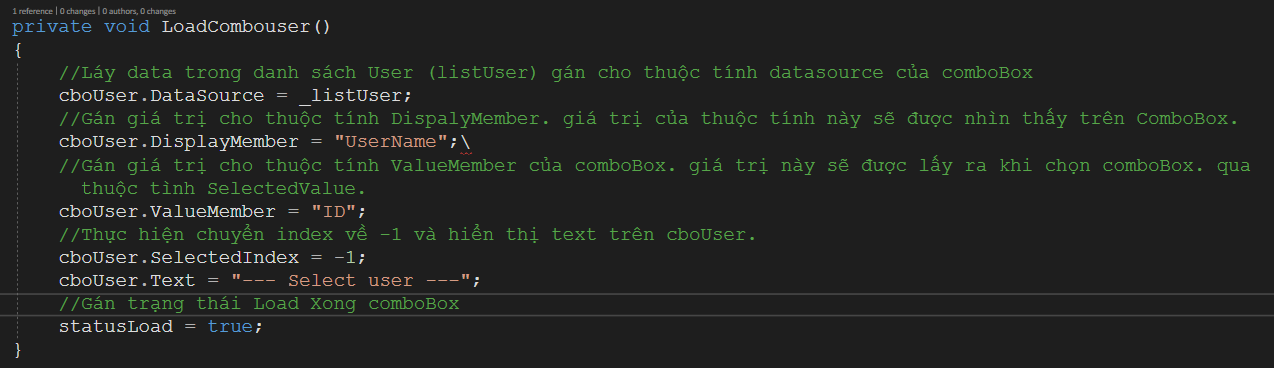


Sự kiện load của Form Frm\_ChangedPassword

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



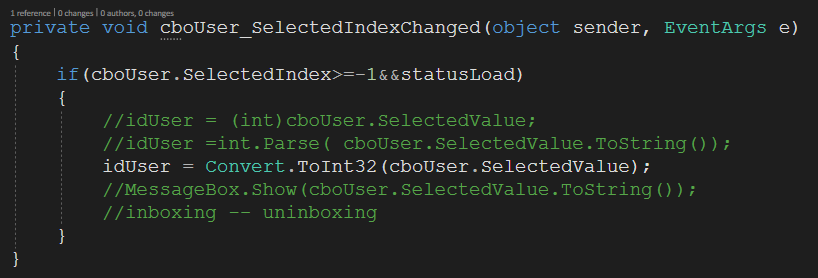
Phương thưc LoadComboUser()



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.

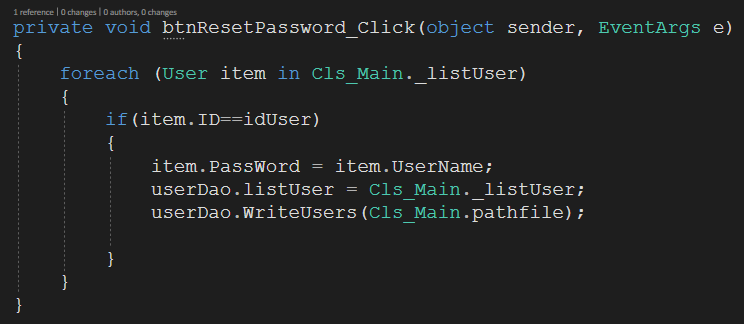
Để sử dụng cần kiểm tra index bẳng thuộc tính SelectedIndex và kiểm tra trạng thái load của ComboBox bằng biến StatusLoad (biến được khái báo ở đầu.

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

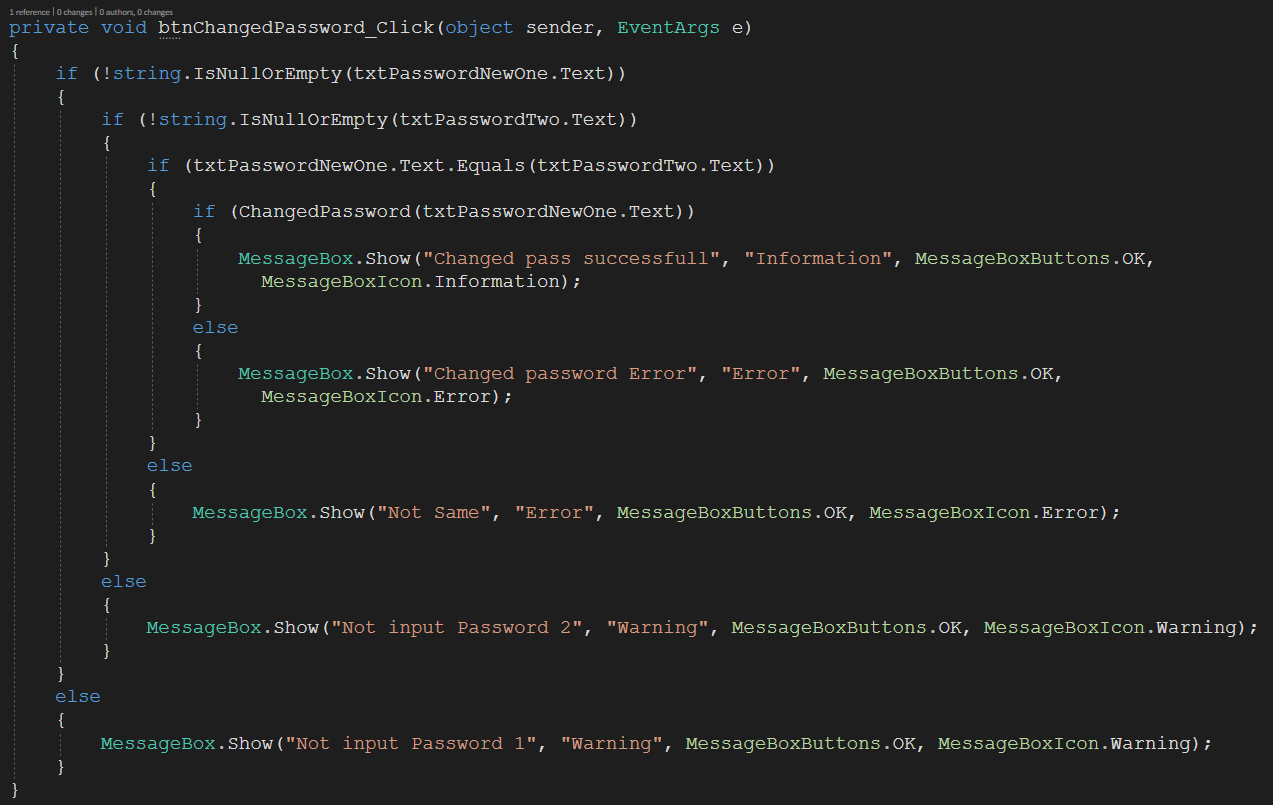


Sự kiện ResetPassword. ResetPassword thực hiện việc cấp lại mật khẩu cho User chọn trong comboBox (cboUser). Sẽ đổi lại password cho user theo tên của UserName.

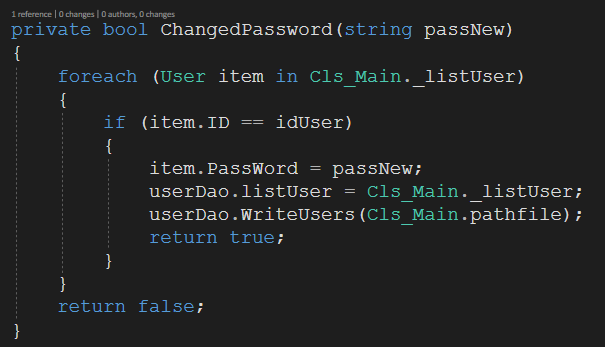
Sau đó tiến hành lưu lại vào trong file listUser.



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



Phương thưc ChangedPassword(string userName)



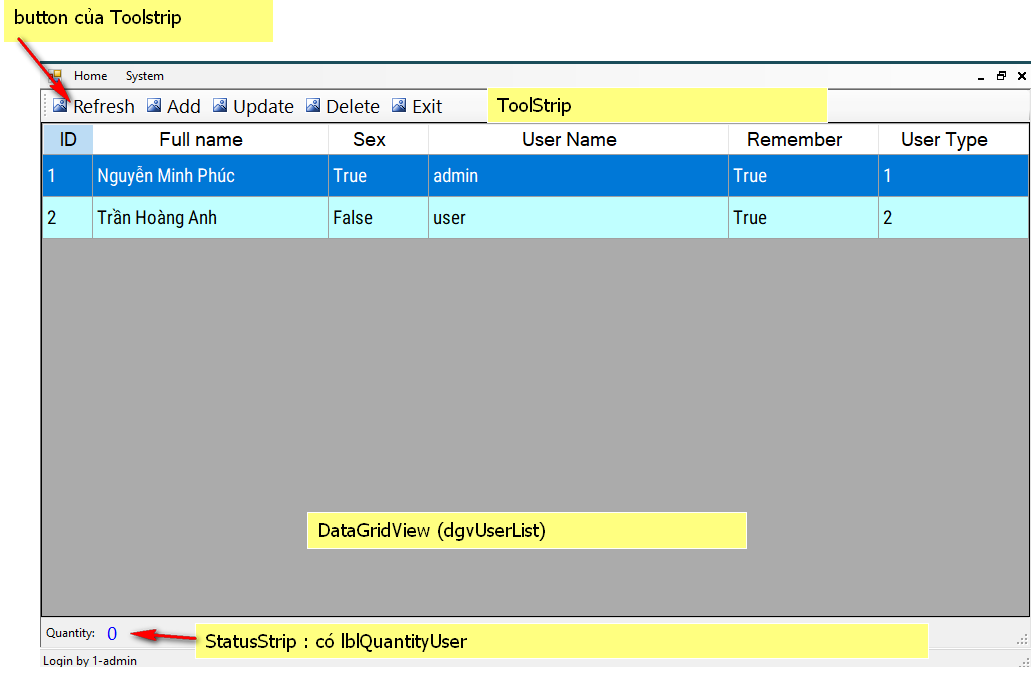
#### 

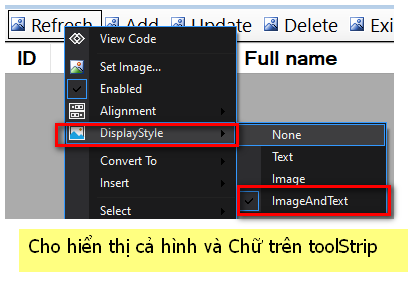
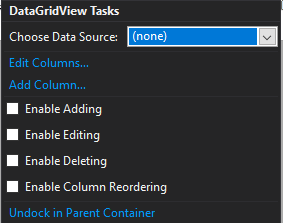
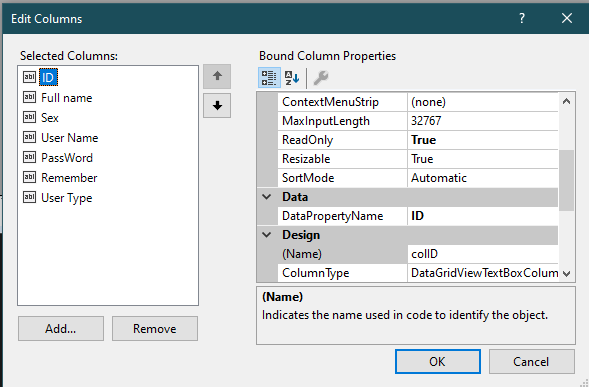
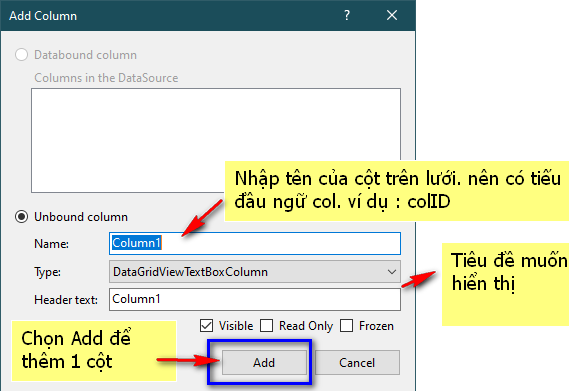
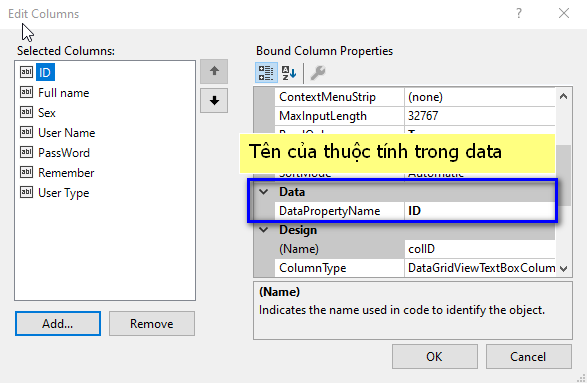
# Thực hiện chức năng quản lý tài khoản người dụng (2021-09-27)

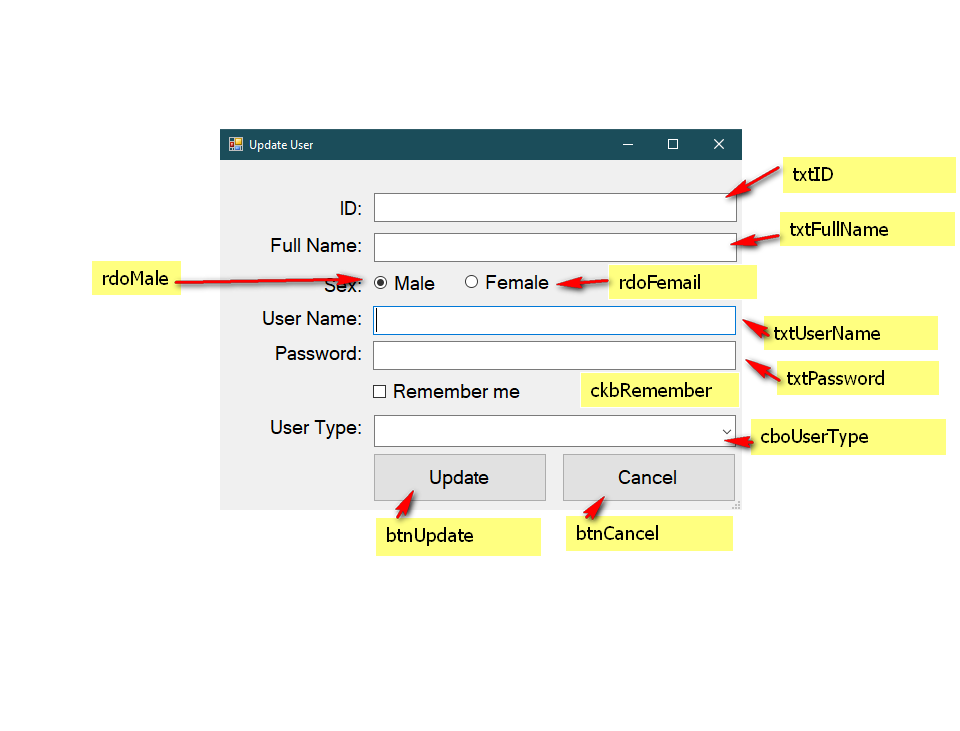
## Thiết kế giao diện

Thiết kế chức năng này, cần thiết kế 2 form

* Form Chính (Frm\_User\_Main): hiển thị danh sách người dùng hiện có trong chương trình.



* Thiết kế **toolStrip**. Tại toolStrip, chọn thêm 1 button và tiến hành đặt tên cho những button này. Để hiển thị như hình hày thực hiện như sau:
* R\_Click vào biểu tượng button và chọn   
  
* Vào Properities để đổi tên (Name) và chữ hiển thị (Text) cho ToolStrip.  
  btnRefresh, btnAdd, bntUpdate, btnDelete, btnExit.
* Để tạo được cột trên dataGridView bằng giao diện, tiến hành thực hiện như sau:
* Tại góc trên bên phải của DataGridView () chọn vào cái hình tam giác.  và chọn **Edit column**. (chú ý trong bước này bỏ kết các check bên dưới luôn)
* Một hộp hoại sẽ xuất hiện.  
    
  Tiến hành thêm các cột (column) cho dataGridView bằng cách chọn vào **Add** và nhập tên các cột (**chú ý:** Thực hiện hết các thuộc tính trong lớp **User**)  
  
* Sau khi thực hiện hết các cột (những thuộc tính trong bảng User) Tiến thành link với data qua thuộc tính **DataPropertyName** : thuộc tính này có giá trị chính là tên của Thuộc tính trong lớp User . ví dụ cột id trên dataGridView muốn link được với thuộc tính ID trong lớp User thì chỗ này nhập ID.  
  
* Những thuộc tính thiết lập cho DataGridView hiển thị: Chọn DataGridView và vào Properties chọn các thuộc tính sau:
* **AlternatingRowsDefaultCellStyle**: Thiết lập định dạng cho dòng chẵn khác định dạng dòng lẻ
* **columnHeadersDefaultCellStyle**: Định dạng cho tiêu đề cột của DataGridView.
* **RowHeadersVisible**: Ẩn đi tiêu đề dòng
* **SelectionMode**: Lựa chọn FullRowSelect (chọn hết dòng khi click lên ô).
* **RowTemplate**: chỉnh thuộc tính Height để chiều cao của dòng lớn hơn, không be chữ.
* Form thêm và chỉnh sửa thông tin người dùng (Frm\_User\_Modifiled): dùng để thêm và sửa thông tin.



**Hướng dẫn viết Code:**

Chỉnh sửa code bài cũ:

Vì trong bài này chúng ta có thêm 2 thuộc tính cho class **User** là FullName và Sex, nên để cho code có thể thực hiện được ta tiến hành thêm 2 thuộc tính này vào Class **User**

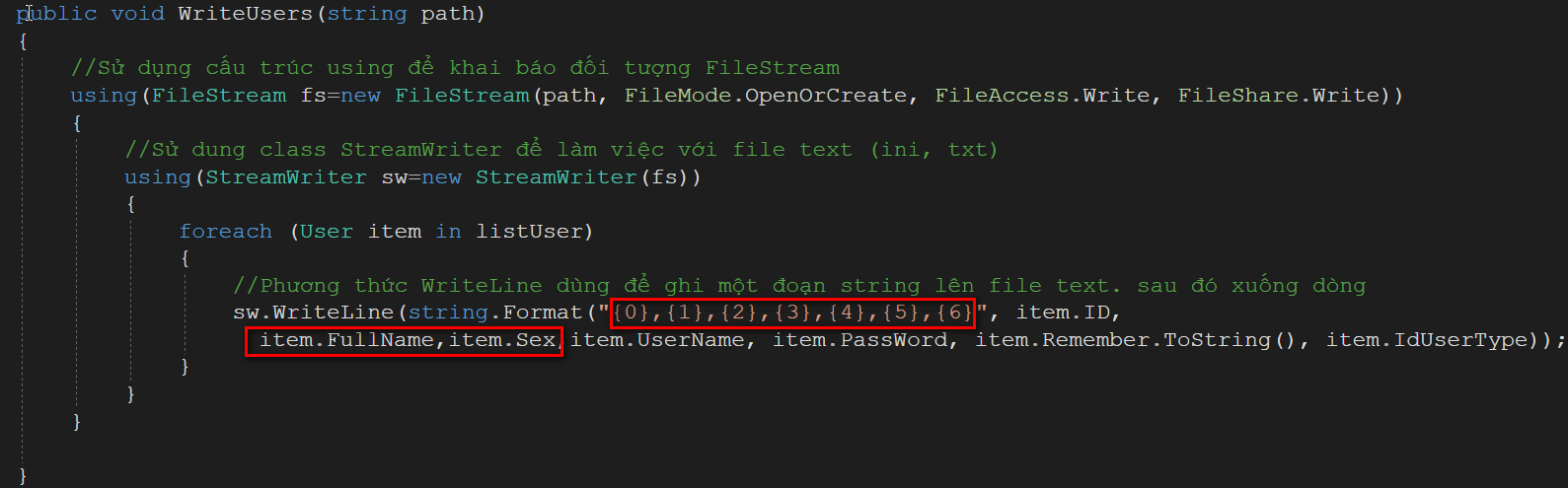
private string fullName;

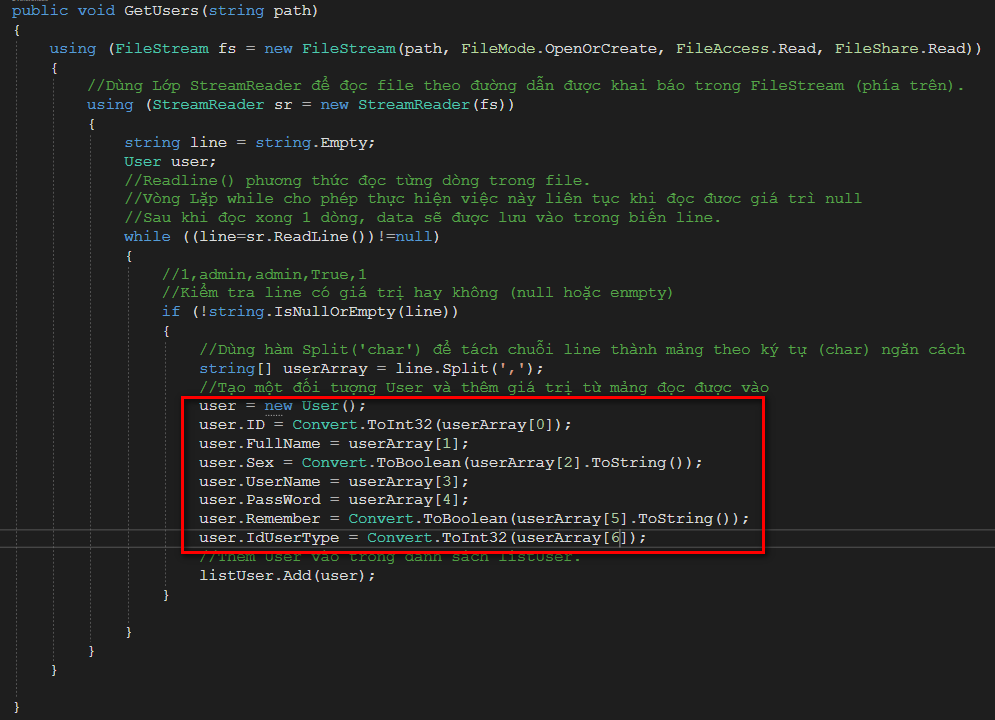
private bool sex;

public string FullName { get => fullName; set => fullName = value; }

public bool Sex { get => sex; set => sex = value; }

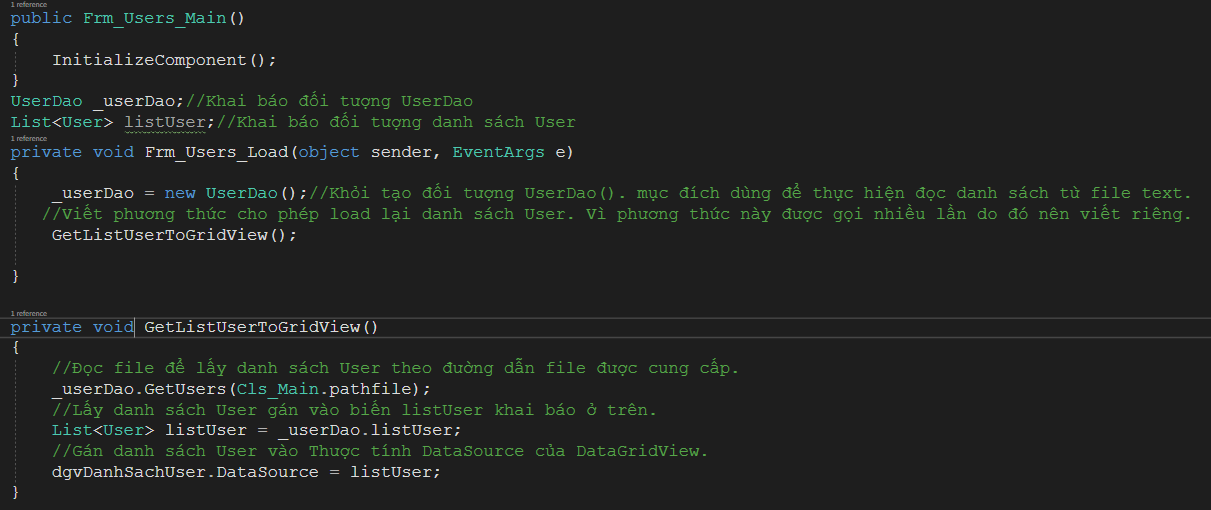
Trong Class UserDao.cs, chỉnh lại hàm GetUsers(),



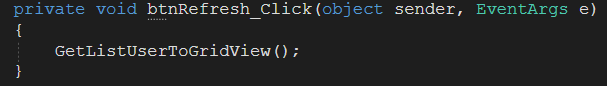


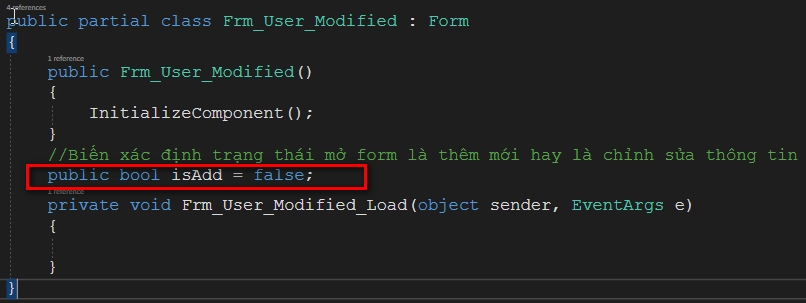
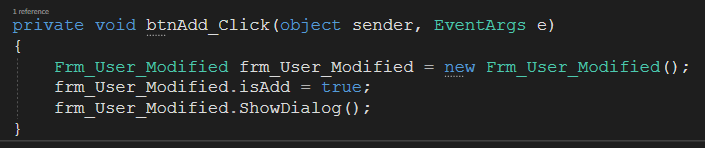
## Hướng dẫn code quản lý User

* Thực hiện chức năng **Load Data** lên DataGridView trong Form **Frm\_User\_Main:**



* Thực hiện chức năng cho btnRefresh: Gọi lại danh sách lên lưới.



* Thực hiện chức năng cho btnAdd: Khi nhấn vào button này sẽ cho mở form Frm\_User\_Modified. Khi nhấn vào btnUpdate cũng cho mở Frm này, do đó trên form Frm\_User\_Modified cần phải khai báo 1 biến public để phân biệt khi mở Form này.  
    
  Tại sự kiện Click của **btnAdd** ta tiến hành code như sau:  
    
  Tại sự kiện lick của btnUpdate ta tiến hành code như sau:  
  
* Thực hiện code cho chức năng thêm mới một User sẽ bao gồm các chức năng sau:
  + ID của User phải tăng tự động khi tiến hành thêm User mới (không trùng với các user đã có.
  + Kiểm tra userName trùng trong danh sách.
  + Kiểm tra nhập liệu bắt buộc cho các thuộc tính (FullName, UserName, Password và UserType) những thuộc tính khác khi không chọn sẽ lấy giá trị mặc đinh.
  + Tiến hành Lưu thông tin vào file. Và đóng form.