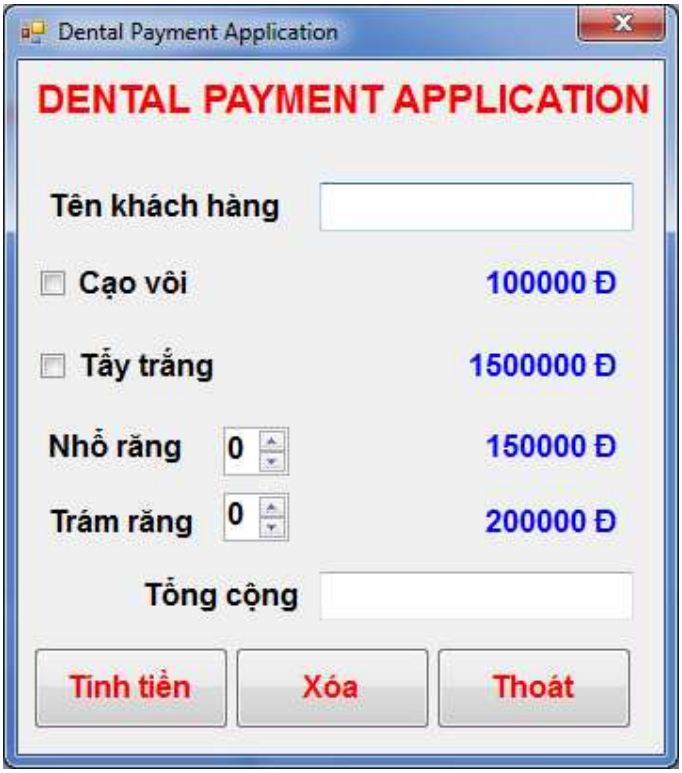
BÀI TẬP THỰC HÀNH LẬP TRÌNH WINDOWS – C#

1. Phần 2: Làm quen với ứng dụng Form (tt)

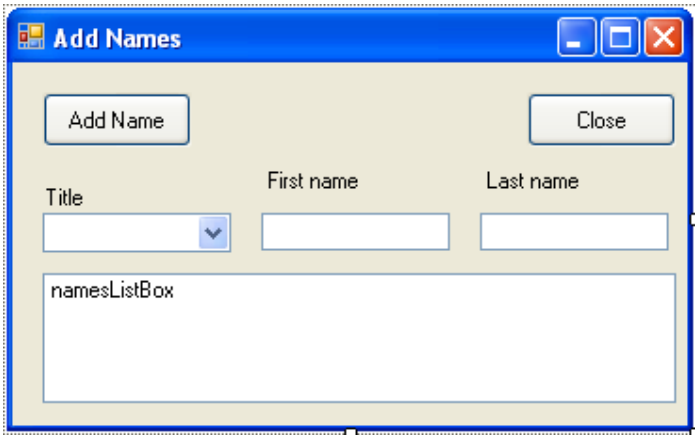
1.1. Bài 12: Viết chương trình tính tiền dịch vụ nha khoa với yêu cầu sau:



1. Khi chạy chương trình (F5) sẽ xuất hiện màn hình như trên, trong đó cấm thao tác trên tất cà các đối tượng trừ đối tượng textbox lưu trữ tên khách hàng và nút thoát.
2. Nếu người dùng nhập tên khách hàng thì mới cho phép thao tác trên các control khác (trừ textbox tổng cộng).
3. Nhấn vào nút "**Tính tiền**", thực hiện tính tổng tiền theo lựa chọn của khách hàng và hiển thị thông báo dạng "Khách hàng: XYZ cần trả số tiền ABC" với XYZ là tên nhập vào, ABC là số tiền cần trả.
4. Nhấn vào nút "**Xóa**": đưa màn hình trở lại trạng thái lúc khởi tạo.
5. Nhấn vào nút "**Thoát**": Xuất hiện thông báo "***Bạn muốn có đóng chương trình không?***"
   - Nếu người dùng chọn **Yes**, đóng chương trình.
   - Nếu người dùng chọn **No**, trở lại chức năng trước đó.

1.2. Bài 13: Danh sách tên

1.2.1. Tạo Form theo mẫu:



1.2.2. Các thuộc tính:

1.2.2.1.Không cho phép resize form:

Property	New Value
FormBorderStyle	FixedSingle
Size	370, 225
Text	Add Names

1.2.2.2. Các Control:

Control Type	Property	New Value
Button	Name	closeButton
	Location	273, 16
	Text	Close
Button	Name	addButton
	Location	16, 16
	Text	Add
Label	Text	Title
Label	Text	First Name
	Location	132, 56
Label	Text	Last Name
	Location	248, 56
TextBox	Name	firstTextBox
	Location	132, 80
	Text	(blank)
TextBox	Name	lastTextBox
	Location	248, 80
	Text	(blank)
ComboBox	Name	titleComboBox
	Items	Mr.;Mrs.;Ms.;Miss.;Dr.
	Location	16, 80
	MaxDropDownItems	4
	Size	100,21
	Text	(blank)
ListBox	Name	namesListBox
	Location	16, 112
	Size	336, 69

1.2.3. Các sự kiện:

1.2.3.1.Sự kiện cho nút Close:

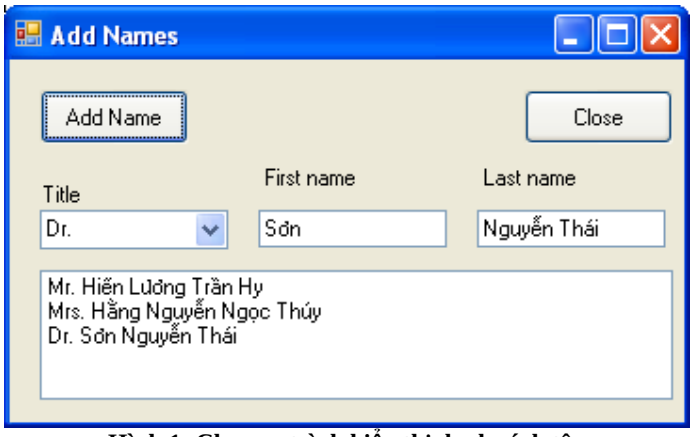
```
private void closeButton_Click(object sender, EventArgs e)
{
      this.Close();//thêm code này
}
```

1.2.3.2.Sự kiện cho nút Add:

```
private void addButton_Click(object sender, EventArgs e)
  {
      //add the following code
      string first = firstTextBox.Text;
      string title = titleComboBox.Text;
      string last = lastTextBox.Text;
```

```
        string nameToAdd = title + " " + first + " " + last;
        namesListBox.Items.Add(nameToAdd);
    }
```

1.2.4. Kết quả chương trình:

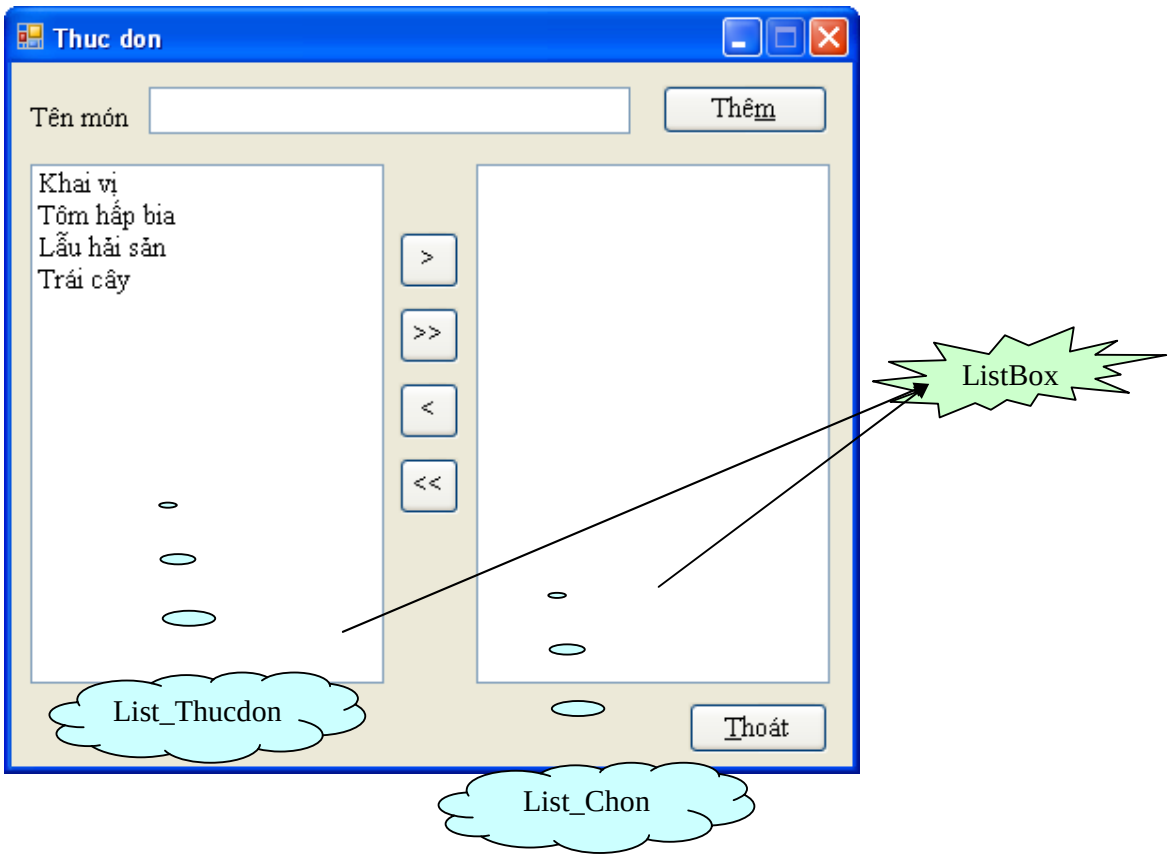


Hình 1: Chương trình hiển thị danh sách tên

Ghi chú: *Bài tập này sinh viên tự thực hiện*

1.3. Bài 14: Xây dựng chương trình ứng dụng: Danh mục món ăn

1.3.1. Thiết kế giao diện như hình sau:



- Đặt lại thuộc tính và tên cho form.
- Đặt lại thuộc tính và tên cho các control trên form.

1.3.2. Đăng ký và xử lý sự kiện Button "Thêm"

- Thêm một item mới vào List_Thucdon:

```
List_Thucdon.Items.Add(txt_Mon.Text);
```

- Sau khi thêm xong thì clear txt_Mon

```
txt_Mon.Text = "";
```

1.3.3. Đăng ký và xử lý sự kiện cho Button >

- Thêm một item được chọn trong List_Thucdon vào List_Chon

```
List_Chon.Items.Add(List_Thucdon.SelectedItem);
```

- Tuy nhiên, sẽ xảy ra trường hợp một item được add nhiều lần. Vì vậy, phải kiểm tra xem trong List_Chon đã tồn tại item đó hay chưa?

1.3.4. Tương tự, đăng ký và xử lý sự kiện cho Button >>

- Sử dụng vòng lặp để duyệt tất cả các item trong ListBox
- ???

1.3.5. Đăng ký và xử lý sự kiện cho Button <

- Xóa một item ra khỏi ListBox

```
List_Chon.Items.Remove(List_Chon.SelectedItem);
```

hoặc:

```
List_Chon.Items.RemoveAt(List_Chon.SelectedIndex);
```

- Xóa tất cả các item trong ListBox

```
List_Chon.Items.Clear();
```

1.4. Bài 15: Sercurity Panel

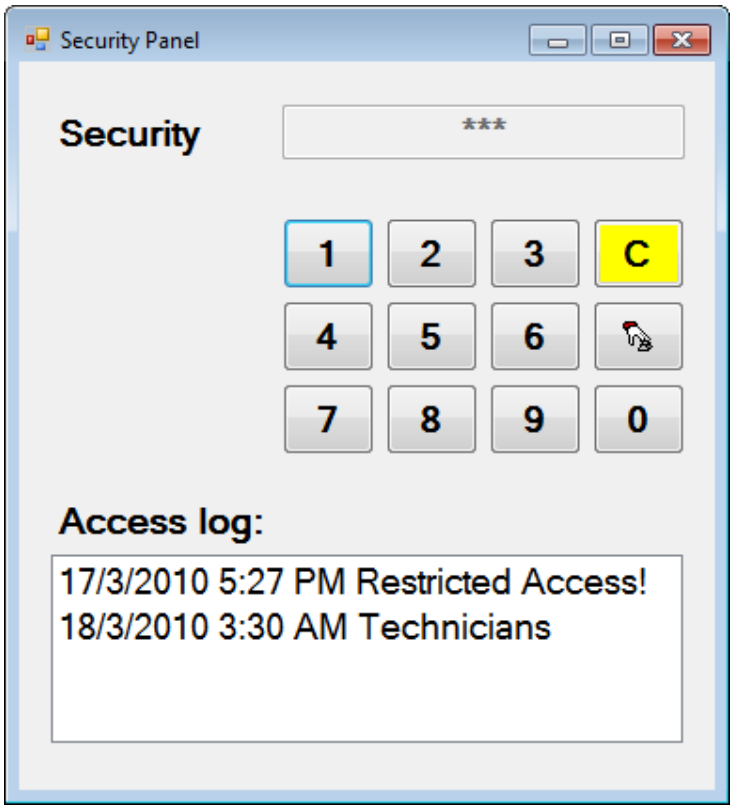
1.4.1. Xây dựng ứng dụng theo yêu cầu sau:

Một phòng lab muốn thiết lập một Sercurity Panel đặt bên ngoài cửa. Chỉ cho phép những các cá nhân có trách nhiệm mới được vào và mỗi lần đăng nhập họ phải nhập các sercurity code được cung cấp. Những sercurity code (hay còn gọi là access code) sau được xem là hợp lệ và cung cấp cho các nhóm nhân viên như bảng sau:

Value	Group
6990, 6999	Technicians
0100	Custodians
9998, 1006 – 1008	Scientist

Một khi nhập access code thì sẽ có hai trạng thái: granted hoặc denied. Tất cả thông tin truy cập sẽ được hiển thị trong một khung thông tin bên dưới keypad. Nếu access là granted thì ngày, thời gian, group (technician, custodian, scientist) sẽ được hiển thị ở khung thông tin. Trường hợp access là denied thì ngày, giờ và thông tin "Access denied" sẽ hiển thị ở khung bên dưới. Ngoài ra user nếu chỉ nhấn một con số sercurity code thì sẽ hiển thị ra thông báo là ngày, giờ và "Restricted Access".

1.4.2. Giao diện chương trình:



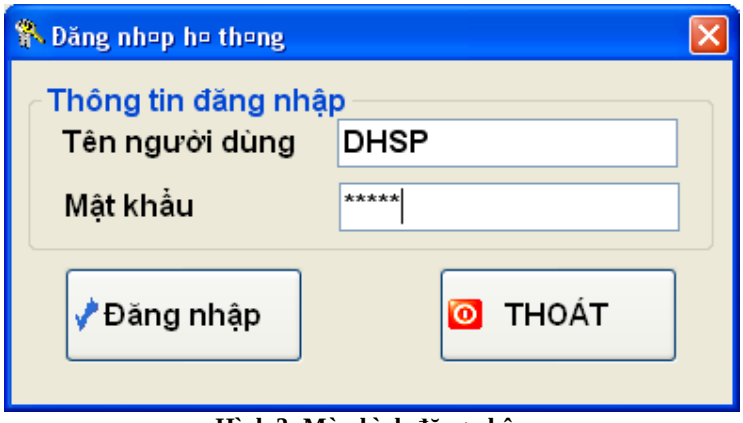
Hình 2: Minh họa giao diện của ứng dụng Security Panel

1.4.3. Yêu cầu nâng cao:[1]

- Bổ sung chức năng log file: tất cả thông tin login dù access granted hay denied đều được ghi nhận vào file dạng text. File này được lưu trữ cùng với thư mục của ứng dụng.

2. Phần 3: Làm quen với các control Toolbar, Menu

2.1. Bài 01 – Viết chương trình thể hiện màn hình "Đăng nhập" (login) như sau:



Hình 3: Màn hình đăng nhập

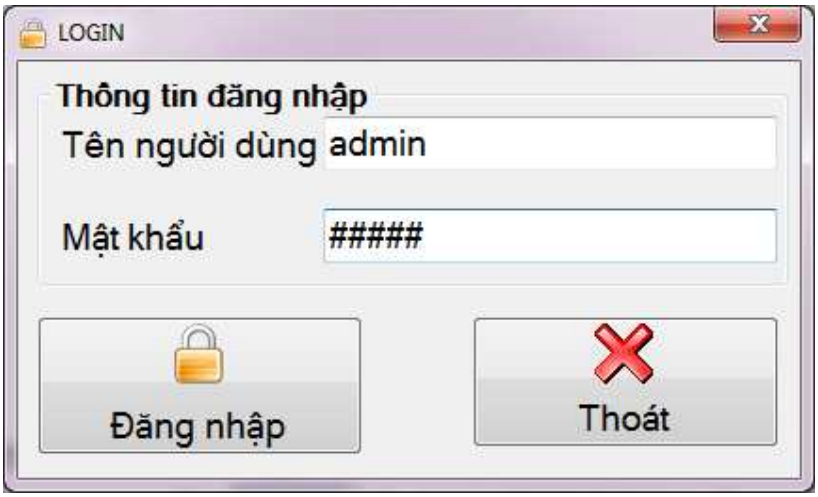
- Khi nhấn vào nút "Đăng nhập", thực hiện kiểm tra
  - Nếu **tên người dùng = "DHSP"** và **mật khẩu = "tinhoc"** thì hiển thị thông báo (MessageBox) **"Đăng nhập thành công**

[1] Sinh viên tự thực hiện sau khi đã đọc bài làm việc với file.

- Ngược lại: thông báo **"Không đúng tên người dùng/mật khẩu"**

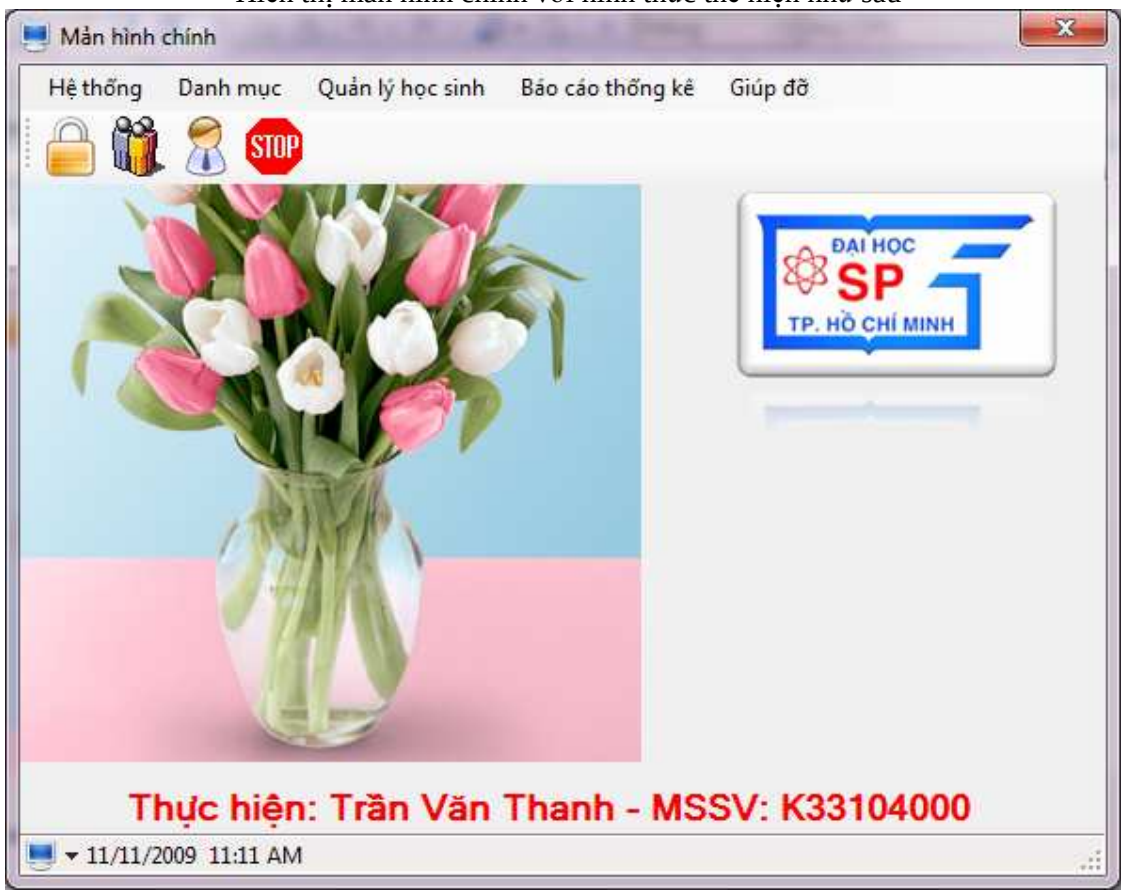
- Khi nhấn vào nút "Thoát"→ Kết thúc chương trình

2.2. Bài 02 – Viết chương trình thể hiện màn hình "Đăng nhập" như sau:



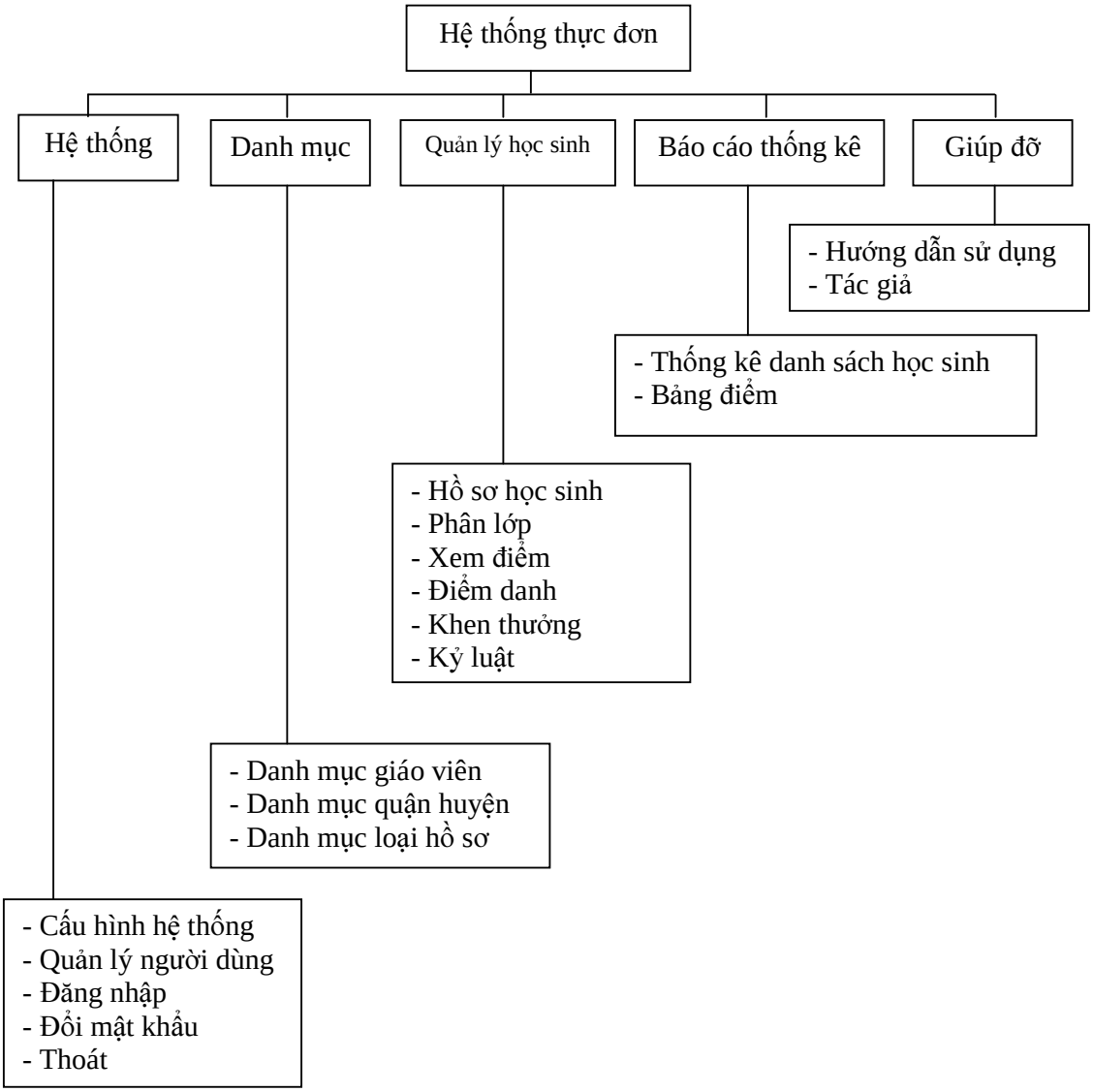
Hình 4: Màn hình đăng nhập

- Khi nhấn vào nút "Đăng nhập", thực hiện kiểm tra
  - Nếu **tên người dùng = "DHSP"** và **mật khẩu = "tinhoc"** thì
    - Ẩn màn hình "Đăng nhập"
    - Hiển thị màn hình chính với hình thức thể hiện như sau



Hình 5: Màn hình chính

  - Ngược lại: thông báo **"Không đúng tên người dùng/mật khẩu"**
- Khi nhấn vào nút "Thoát" hiển thị thong báo "Bạn có muốn thoát chương trình không ?"
  - Nếu chọn "Yes"→ Kết thúc chương trình
  - Ngược lại: trở lại màn hình login
- Hệ thống Menu màn hình chínhnhư sau:



- Nhấn vào chức năng **"Hệ thống\Đăng nhập"** trong **menu** hoặc nút hình **"Chùm khóa"** trên **ToolBar**, hiển thị màn hình login như trong **bài tập 01**.
- Nhấn vào chức năng **"Hệ thống\Thoát":** Kết thúc chương trình
- Nhấn vào chức năng **"Hệ thống\Quản lý người dùng"** trong **menu** hoặc nút hình **"3 người"** trên **ToolBar**, hiển thị màn hình quản lý người dùng như hình bên dưới.

Hình 6: Màn hình quản lý người dùng

- Nhấn vào chức năng **"Quản lý học sinh\Thông tin học sinh"** trong **menu** hoặc nút hình **"thằng người có cà vạt"** trên **ToolBar**, hiển thị màn hình thông tin sinh viên.

Hình 7: Màn hình quản lý học sinh

Hình 8: Màn hình quản lý học sinh

3. Thiết kế chương trình tựa WordPad: RichTextEditor

3.1. Mục tiêu

- Phát triển ứng dụng Single Document Interface
- Làm quen với MenuStrip, ToolStrip, StatusStrip.
- Sử dụng các Common Dialog: OpenFileDialog, SaveFileDialog, ColorDialog…
- Sử dụng control RichTextBox.

3.2. Nội dung

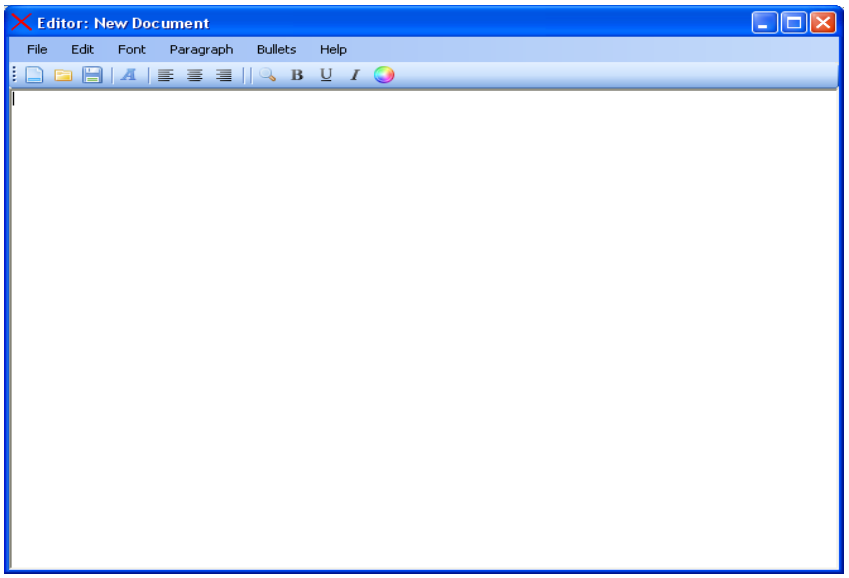
Xây dựng một ứng dụng soạn thảo văn bản có các chức năng sau:

Nhóm chức năng cơ bản:

- Cho phép soạn thảo trên hai định dạng
  - Cơ bản: text File (*.txt)
  - Nâng cao: Rich Text Format (*.rtf)
- Cho phép định dạng các đoạn text
  - Align: left, right và center.
  - Format: bold, italic, underline
  - Chọn font chữ
  - Chọn màu chữ
  - Chọn màu nền của document
- Chức năng Cut, Copy và Paste.

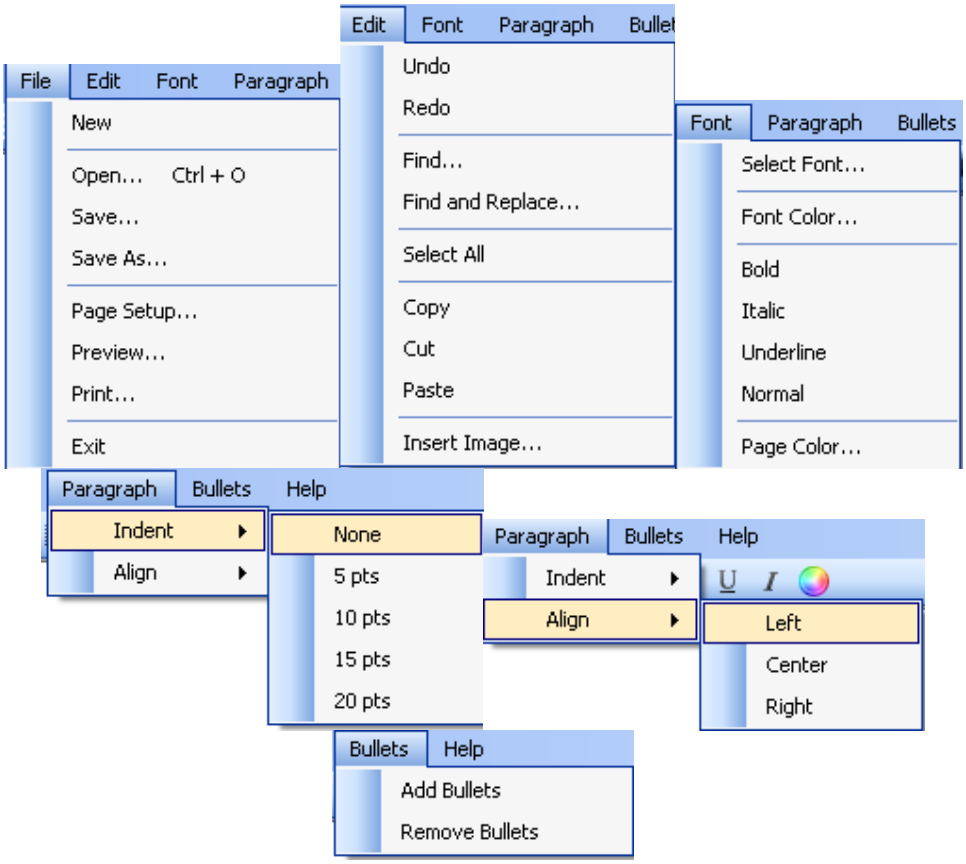
Nhóm chức năng nâng cao:

- Các chức năng như sau
  - Open file: mở file RTF hoặc TXT
  - Save file: lưu document
  - New file: tạo mới document
  - Chức năng chèn ảnh
  - Chức năng chèn bullet
  - Indent
  - Zoom
  - ...



Hình 9: Giao diện chương trình RichTextEditor

Chi tiết các Menu:



3.3. Lưu ý:

- Sinh viên chỉ cần làm nhóm chức năng cơ bản là đạt yêu cầu
- Chức năng nâng cao tùy chọn, sinh viên khá giỏi nên làm.

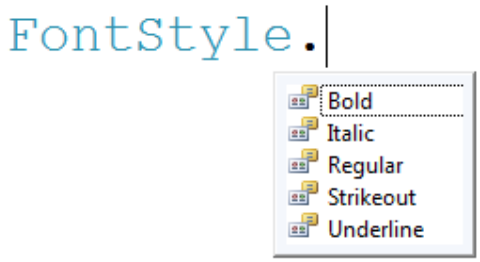
3.4. Mở rộng:

Sinh viên có thể mở rộng bất cứ tính năng nào cho ứng dụng của mình, sự sáng tạo luôn được đánh giá cao.

Hướng dẫn cơ bản:

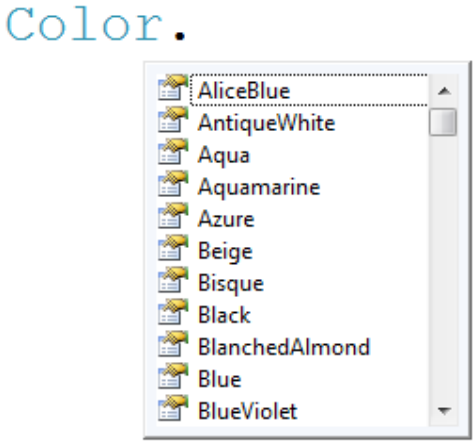
Sinh viên tìm hiểu các thuộc tính cơ bản của RichTextBox control như:

- **SelectionFont**
- **SelectionFont.Style**: mô tả các kiểu chữ
  - Gồm có các kiểu mô tả trong enumeration FontStyle như sau



- **SelectionAlignment**: thiết lập thuộc tính canh lề
  - Gồm có các kiểu mô tả trong enumeration **HorizontalAlignment** như sau

- **SelectionColor**: thiết lập màu của text, sử dụng struct Color để mô tả màu



- **BackColor**: thiết lập màu nền cho document chứa text, sử dụng struct Color

Các phương thức của RichTextBox như:

- **Cut()**: cut đoạn text được chọn vào Clipboard
- **Copy()**: copy đoạn text được chọn vào Clipboard
- **Paste()**: dán đoạn text trong Clipboard vào document.

Hướng dẫn chức năng Bold trên ToolStrip:

```
private void btnBold_Click(object sender, EventArgs e)
{
    //Lấy style hiện hành của RichTextBox có tên rtbDoc
    FontStyle newstyle = rtbDoc.SelectionFont.Style;

    //Thiết lập hoặc tắt kiểu style
    newstyle = rtbDoc.SelectionFont.Style ^ FontStyle.Bold;

    //Tạo font có style mới cho rtbDoc
    rtbDoc.SelectionFont = new Font(rtbDoc.Font, newstyle);

    //Thiết lập trạng thái check hoặc không check cho button Bold
    btnBold.Check = (rtbDoc.SelectionFont.Style & FontStyle.Bold) > 0;
}
```

Sinh viên làm tương tự với các chức năng: Italic, Underline…

Hướng dẫn chức năng canh lề

Thiết lập thuộc tính canh lề trái cho RichTextBox tên rtbDoc

```
rtbDoc.SelectionAlignment = HorizontalAlignment.Left;
```

Hướng dẫn chức năng cut, copy và paste cho RichTextBox tên rtbDoc

```
rtbDoc.Cut();    // cut đoạn text được chọn vào clipboard
```

tương tự cho copy và paste

Hướng dẫn chức năng nâng cao Insert ảnh vào document: Sinh viên tự nghiên cứu đoạn code sau:

```
OpenFileDialog dlg = new OpenFileDialog();
dlg.Filter = "Bitmap File (*.bmp)| *.bmp"; //Mở rộng thêm
if (dlg.ShowDialog() == DialogResult.OK)
{
    Image img = Image.FromFile(dlg.FileName);
    Clipboard.SetDataObject(img);
    DataFormats.Format df;
    df = DataFormats.GetFormat(DataFormats.Bitmap);
    if (this.rtbDoc.CanPaste(df))
        rtbDoc.Paste(df);
}
```