Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Phát Triển Phần Mềm Và Ứng Dụng Thông Minh	**BÀI 2** **THIẾT KẾ GIAO DIỆN**	

A. MỤC TIÊU:

- Thiết kế Control.

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. Components

Phân loại: Có 2 loại

- Thành phần xây dựng sẵn của .NET
- Thành phần do người dùng định nghĩa.

1.1 Thành phần xây dựng sẵn của .NET

Tham chiếu đến thành phần xây dựng sẵn:

- Thành phần không nhìn thấy trên thanh công cụ: sử dụng hộp thoại Project/ Add reference ….
- Thành phần nhìn thấy trên thanh công cụ:kích phải trên tab trên Toolbox, chọn Choose Items, danh sách các thành phần hiển thị ở Tab COM Components hay .NET Framework Components….

Ví dụ thành phần xây dựng sẵn:

1. Kích phải trên tab trên Toolbox, chọn Choose Items
2. Chọn điểu khiển Browse Button ở tab COM Components
3. Kích nút OK, biểu tượng Browse Button sẽ xuất hiện trên Toolbox.
4. Kéo lê điều khiển mong muốn vào Winform, sẽ có các assemblies .NET tương ứng tự động tạo và thêm vào ứng dụng, xem ở mục References trong cửa sổ Solution Explorer.

1.2 Thành phần người dùng định nghĩa

a. Kế thừa tự control đã có

Controls thừa kế từ control đã có

1. File/ New Project, Windows Forms Control Library
2. Viết mã kế thừa lớp điều khiển đã có:
3. Ví dụ: public class NumericTextBox: System.Windows.Forms.TextBox
4. Viết mã bổ sung hàm, sự kiện, thuộc tính cho lớp
5. Biên dịch để tạo file DLL trong thư mục <Application Folder>\bin\Debug

b. Custom không giao diện (tạo thư viện sử dụng)

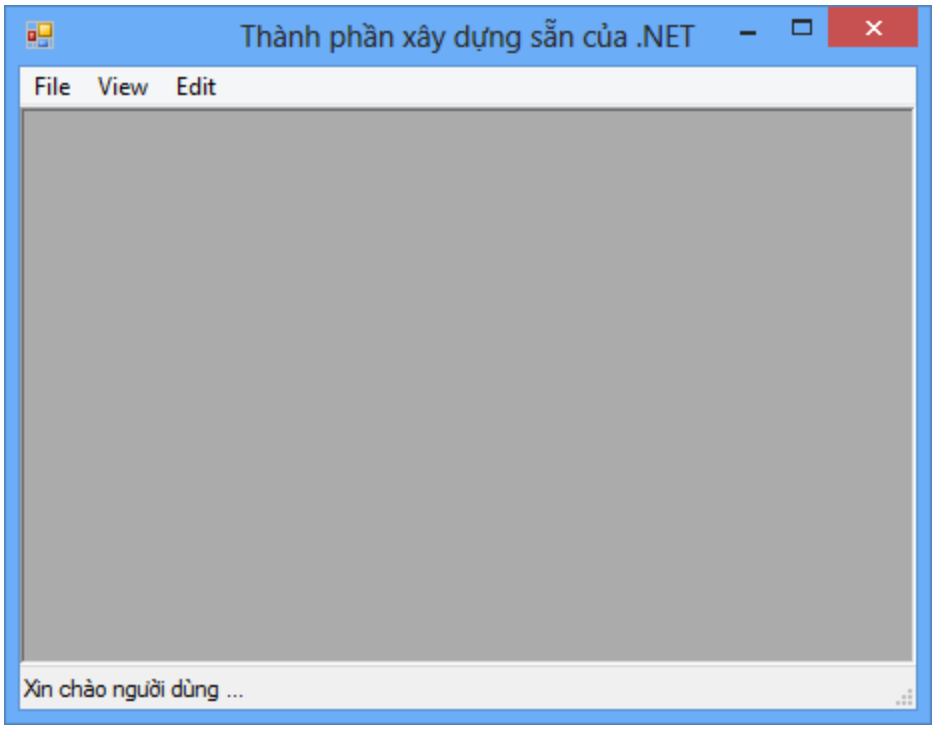
1. File/ New Project, Windows Forms Control Library(Có giao diện) hay Class Library (Không có giao diện).
2. Nếu custom control có giao diện, kích phải trên Project, chọn Add/ Custom Control: public class CardValidator: Control
3. Viết mã thuộc tính, phương thức...
4. Biên dịch ứng dụng để tạo file DLL trong thư mục <Application Folder>\bin\Debug

c. Custom controls có giao diện

1. File/ New Project, Windows Forms Control Library
2. Thiết kế giao diện người dùng của user control
3. Viết mã để thêm phương thức, property cho control.
4. Biên dịch ứng dụng, sẽ tạo tập tin DLL trong thư mục <Application Folder>\bin\Debug

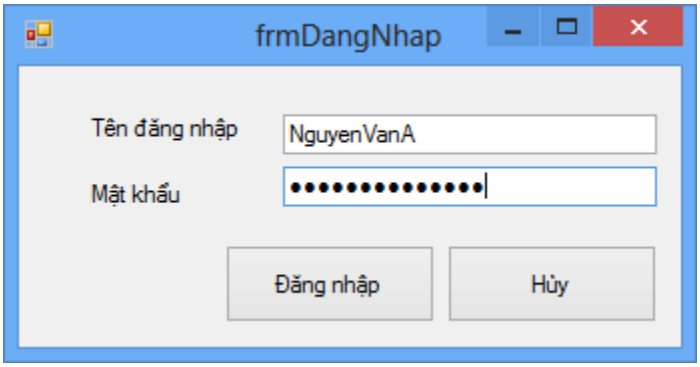
2. Giới thiệu bài tập mẫu

2.1 Sử dụng thành phần xây dựng sẵn của .NET: StatusBar, MainMenu xây dựng Form Main giao diện như sau:

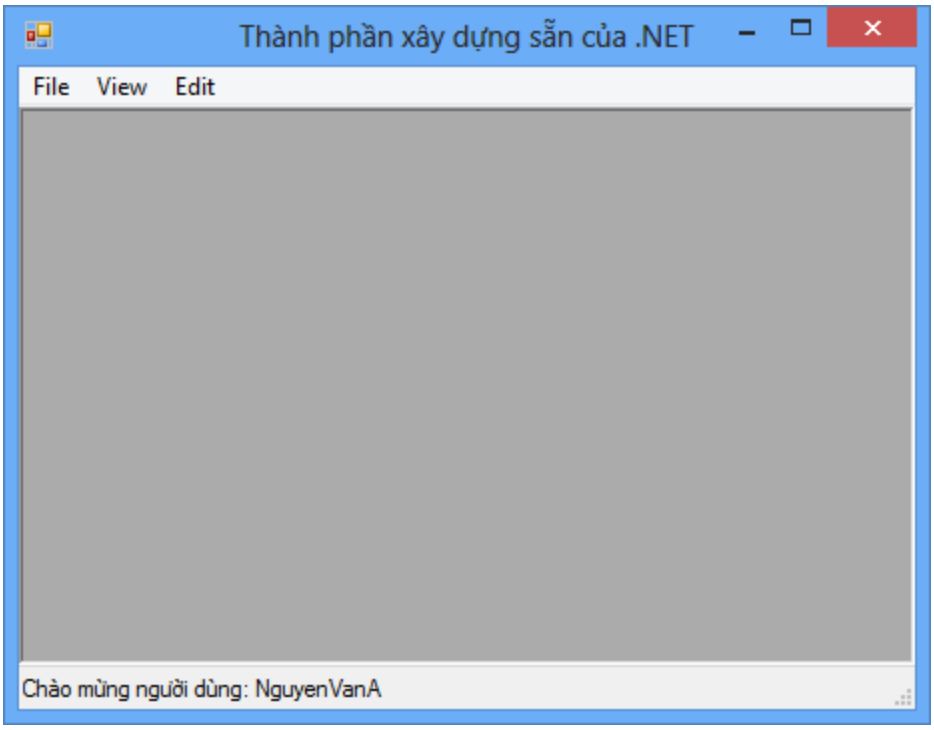


B1: Tham chiếu đến thành phần xây dựng sẵn để đưa 2 control trên hiển thị trên Toolbox.

B2: Tạo Form đăng nhập như sau:



B3: Sau khi thực hiện đăng nhập hiển thị thông tin Tên đăng nhập lên StatusBar của From main như sau:



2.2 Kế thừa tự control đã có: Tạo Control NumericTextBox chỉ chấp nhận nhập giá trị số.

1. File/New Project, Class Library
2. Kích phải trên dự án, chọn Add/ User Control
3. Thay mã thừa kế:

public partial class NumericTextBox : TextBox

4. Thêm mã sau:

```
public partial class NumericTextBox : TextBox
    {
        public NumericTextBox()
        {
            InitializeComponent();
        }
private void NumericTextBox_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar))
            {
                e.Handled = true;
            }
        }
    }
```

Tạo ứng dụng sử dụng control

1. File/ New Project, Windows Forms Application

2. Kích phải trên All Windows Forms trong Toolbox, chọn Choose Items, ở tab .NET Framework Components, chọn Browse để duyệt đến file NumericTextBox.dll, OK
3. Tạo form, đặt điều khiển NumericTextBox vào form
4. Chạy ứng dụng

2.3 Custom không giao diện (tạo thư viện sử dụng): Kiểm tra độ dài của Tên Card nếu Tên Card có độ dài khác 5 là không hợp lệ.

File/ New Project, Class Library, tạo Project Card_Validator chứa lớp CardValidator.cs

```
public class CardValidator
        {
            public string Name { get; set; }
            public string CardNo { get; set; }
            public CardValidator(string Name, string CardNo)
            {
                this.Name = Name;
                this.CardNo = CardNo;
            }
            public bool Validate()
            {
                int CardLength = CardNo.Length;
                if (CardLength == 5)
                    return true;
                else
                return false;
            }
        }
```

Sử dụng custom controls không giao diện

1. File/ New/ Project, chọn Windows Forms Application
2. Để thêm một tham chiếu đến lớp CardValidator, kích phải trên mục Reference trong cửa sổ Solution Explorer, chọn mục Add Reference, chọn Browse để duyệt đến lớp thư viện Card_Validator.dll

Tạo một sự kiện thực thi lệnh sau:

```
    CardValidator Validator = new CardValidator("Name", "1234567");
            if (Validator.Validate())
                this.lblThongBao.Text = "Valid card Number" +
Validator.CardNo;
            else
            {
                this.lblThongBao.Text = " Invalid Card Number";
```

```
                MessageBox.Show("Invalid Card Number");
            }
```

2.4 Custom có giao diện: Xây dựng Control mới kết hợp Label và Timer

1. File/ New Project, Windows Form Control Library, gõ tên Project là Digital_Clock
2. Add/ User Control với tên DigitalClock, thiết kế User Control như sau:

➢ Bổ sung điều khiển Label

➢ Bổ sung điều khiển Timer với thuộc tính:

- Interval: 1000
- Enabled: True

3. Kích đúp vào điều khiển Timer để mở hàm sự kiện Tick, bổ sung mã sau: label1.Text = DateTime.Now.ToString();
4. Biên dịch ứng dụng: Build/Build Solution

Sử dụng Custom có giao diện

1. File/ New Project, Windows Forms Application
2. Kích phải trên tab All Windows Forms trên Toolbox, Choose Items, tab .NET Framework Components, Browse, duyệt đến DigitalClock.dll
3. Tạo form, bổ sung Control DigitalClock vào form
4. Chạy ứng dụng

3. Bài tập tự làm

Kế thừa từ control

1 Tạo control UpperTextBox chỉ chấp nhận chữ hoa.

2 Tạo control MailTextBox chỉ chấp nhận khi có ký tự @ và '.com'.Nếu không tồn tại 2 điều kiện trên thì thông báo lỗi qua ErrorProvider.

3 Tạo control UserTextBox không chấp nhận khi có ký tự đặc biệt. Nếu tồn tại ký tự đặc biệt thì thông báo lỗi qua ErrorProvider.

4 Tạo control PassTextBox chỉ chấp nhận khi có ký tự đặc biệt. Nếu không tồn tại ký tự đặc biệt hoặc ít hơn 6 ký tự thì thông báo lỗi qua ErrorProvider.

5 Tạo control DataGridView có dòng chẵn 1 màu dòng lẻ 1 màu.

6 Xây dựng Button sao cho khi Hover chuột qua thì Button đổi màu.

Sử dụng custom controls không giao diện

Tạo thư viện SqlClass với các yêu cầu sau:

7 Hàm tạo đối tượng `SqlConnection` với mẫu hàm như sau:

```
public void CreateConnection(string pConnectionString)
```

```
        {
        }
```

8 Hàm `TestConnection()`

```
public bool TestConnection()
        {
        }
```

9 Hàm thực thi 1 câu truy vấn trả về danh sách kết quả

```
public DataTable ExcuteQuery(string pQuery)
        {
        }
```

10 Hàm thực thi thêm, xóa, sửa

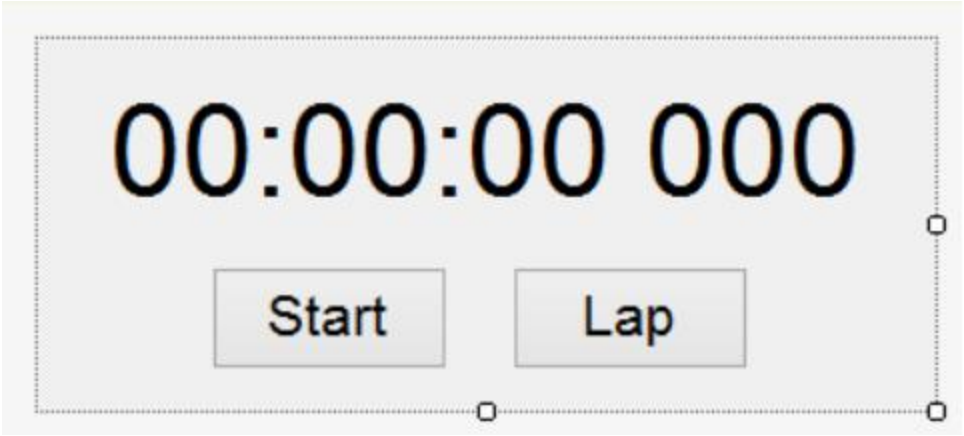
```
public bool Insert(string pQuery, SqlConnection pConnection)
        {
        }
public bool Update(string pQuery, SqlConnection pConnection)
        {
        }
public bool Delete(string pQuery, SqlConnection pConnection)
        {
        }
```

Sử dụng Custom controls có giao diện

11 Thiết kế user controls sau với: MSSV truyền từ biến ngoài. Thông tin sinh viên được lưu trên file txt

Mã sinh viên:	**3951140202067**	Ngày sinh:	29/06/1999
Tên sinh viên:	Huỳnh Ngọc Bích	Giới tính:	Nữ
Mã lớp:	04DHTH1	CMND:	

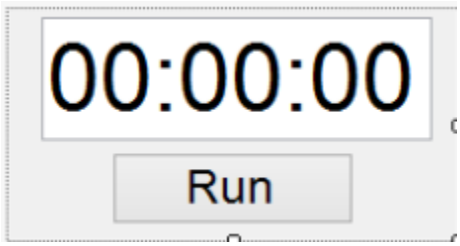
12 Đồng hồ bấm giờ



Yêu cầu:

- Bấm Start thì cho đồng hồ chạy đồng thời đổi text thành Stop.
- Bấm Stop thì cho đồng hồ dừng và đổi text thành Start.
- Bấm Lap (ghi giờ) thực hiện ghi xuống file số lần cho trước.

13 Đồng hồ đếm ngược



Yêu cầu:

- Thiết kế và thực hiện coding đồng hồ điếm ngược cho phép người dùng nhập giá trị vào khi bấm "**Run**" thì tiến hành đếm ngược.