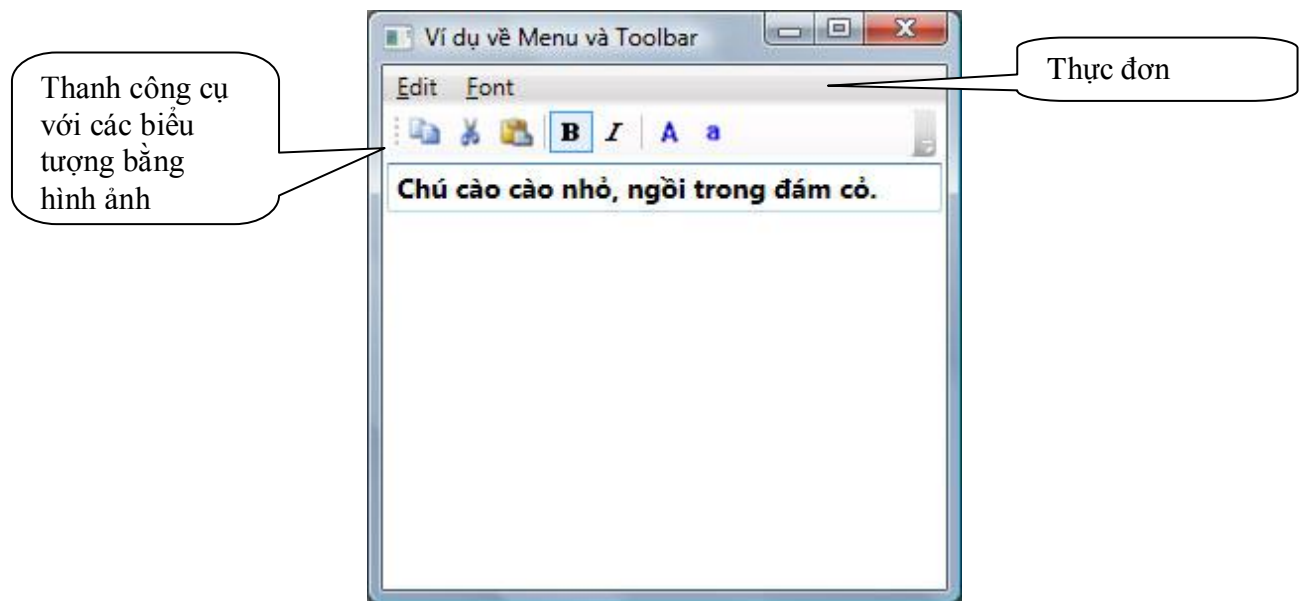


Bài 4

THỰC ĐƠN (MENU) VÀ THANH CÔNG CỤ (TOOLBAR) - WPF

Thực đơn (Menu) và thanh công cụ (Toolbar) là một trong những thành phần quan trọng của cửa sổ, chúng chứa đựng các chức năng chính của chương trình mà người dùng có thể thực hiện. Thanh thực đơn chứa hầu hết tất cả chức năng chính của chương trình, tổ chức theo dạng phân cấp, trong khi thanh công cụ thường chứa một số chức năng thiết yếu mà người dùng hay quan tâm dưới dạng các biểu tượng hình ảnh để người dùng có thể thao tác một cách nhanh chóng. Hình 4.1 minh họa cửa sổ chương trình với thanh thực đơn và thanh công cụ.

Bài này giới thiệu phương pháp xây dựng và sử dụng Thực đơn và Thanh công cụ bằng ngôn ngữ XAML trong ứng dụng WPF từ những ví dụ đơn giản là các mục menu (Menu Item) thông thường (không có biểu tượng và trạng thái) đến những ví Menu Item với biểu tượng hình ảnh (Icon) và các trạng thái (Checked và UnChecked).



Hình 4.1. Một ví dụ về cửa sổ với thực đơn và thanh công cụ

1. Xây dựng thực đơn và sử dụng thực đơn

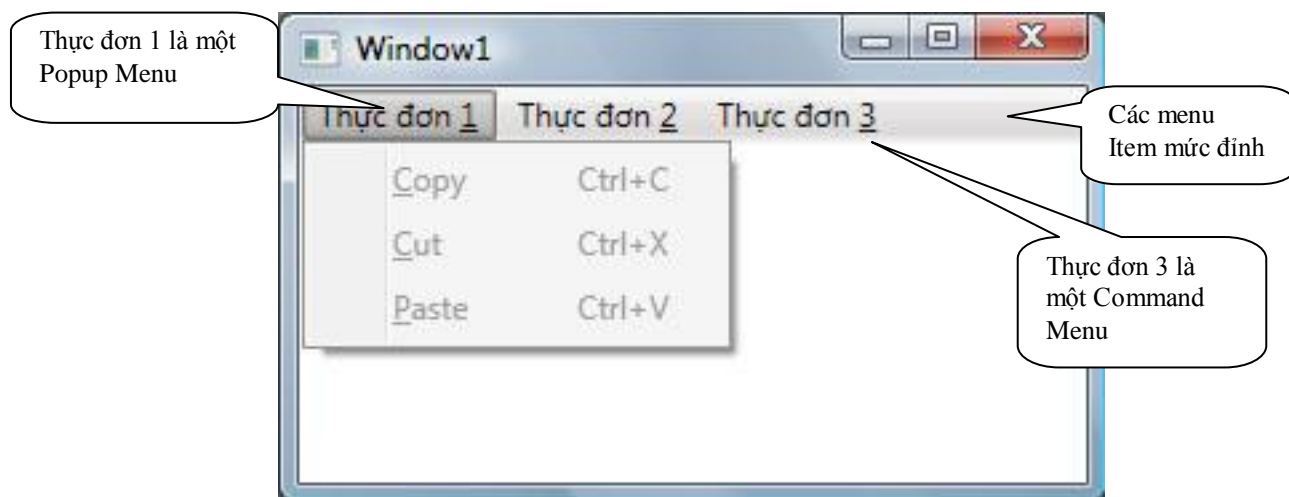
Thực đơn (**Menu**) là điều khiển gồm nhiều phần tử được tổ chức dưới dạng phân cấp. Thanh thực đơn thường nằm trên đỉnh cửa sổ (dưới thanh tiêu đề). Các phần tử thực đơn (**Menu Item**) xuất hiện trên thanh thực đơn còn được gọi là Menu Item mức đỉnh. Mỗi Menu Item mức đỉnh có thể chứa nhiều Menu Item cấp dưới (Sub Menu) hoặc được gắn trực tiếp với các bộ quản lý sự kiện (Event handler) như sự kiện **Click** hay các lệnh của hệ thống được xây dựng sẵn (như Copy, Cut, Paste,...). Tương tự như vậy, mỗi Menu Item cấp dưới lại có thể chứa nhiều Menu Item cấp dưới của chính nó.

Khi một Menu Item chứa các Menu Item cấp dưới thì thường được gọi là Popup Menu, các Menu Item cấp dưới sẽ xuất hiện khi người dùng nhấn chuột lên Popup Menu. Nếu Menu Item được gắn trực tiếp với bộ quản lý sự kiện hay một lệnh của hệ thống thì được gọi là Command Menu, nó sẽ thực thi một câu lệnh mong muốn khi người dùng nhấn chuột hoặc nhấn phím tắt (ký tự trên bàn phím gắn với Menu Item) để chọn nó.

Ta sẽ tìm hiểu từng bước xây dựng và sử dụng menu bắt đầu từ Menu với các Menu Item đơn giản, tiếp đến là các Menu Item có trạng thái (Checked, UnChecked) và Menu Item có biểu tượng hình ảnh.

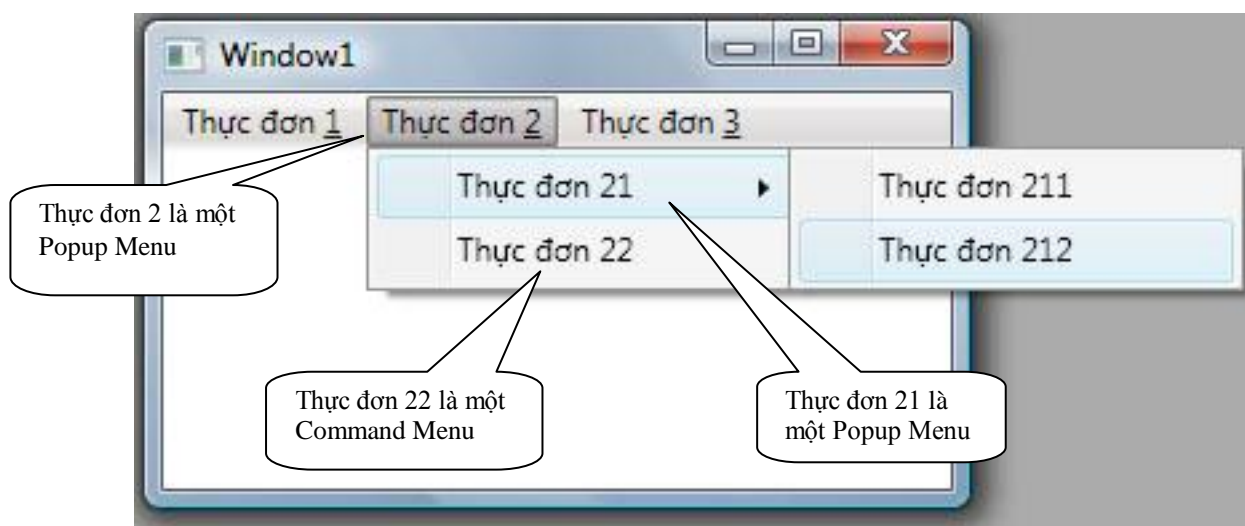
1.1 Xây dựng thực đơn với các Menu Item đơn giản

Trong phần này ta tìm hiểu từng bước xây dựng một thanh menu đơn giản gồm 3 Menu Item mức đỉnh là Thực đơn 1, Thực đơn 2 và Thực đơn 3. Trong đó, Thực đơn 1 và Thực đơn 2 là các Menu Popup (có chứa các menu con), Thực đơn 3 là loại Command Menu Item, nó không chứa Menu con mà sẽ thực thi một câu lệnh khi ta nhấn chuột vào nó như minh họa ở Hình 4.2.



Hình 4.2. Một ví dụ về cửa sổ với thanh thực đơn

Hình 4.3 minh họa Thực đơn 2 có hai Menu Item cấp dưới là Thực đơn 21 và Thực đơn 22. Trong đó, Thực đơn 21 là Popup Menu chứa hai thực đơn cấp dưới của nó là Thực đơn 211 và Thực đơn 212, Thực đơn 22 là Command Menu Item.

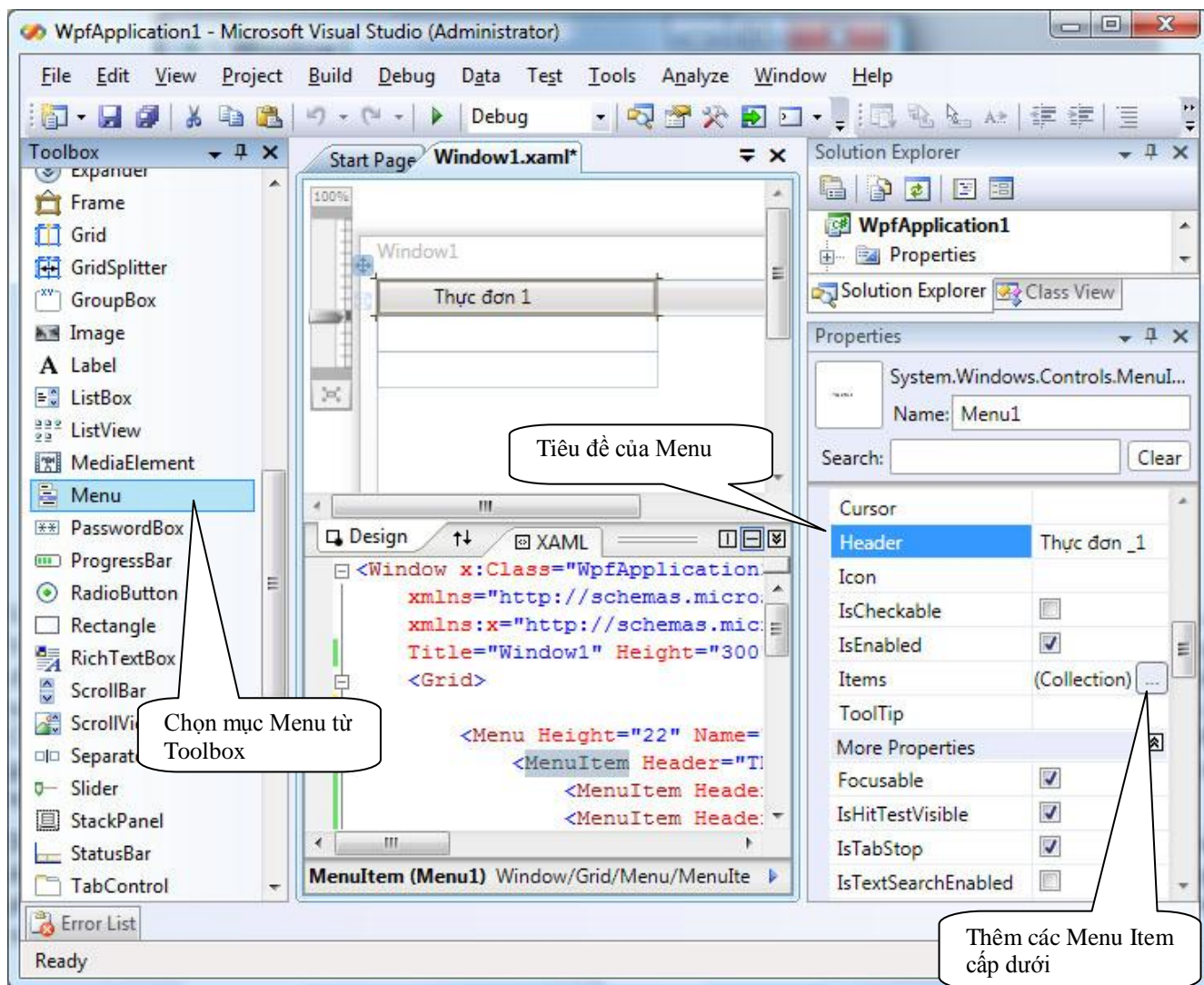


Hình 4.3. Ví dụ về thực đơn cấp dưới là một Popup Menu

Ta có thể tạo menu bằng công cụ trực quan hoặc gõ trực tiếp mã lệnh XAML.

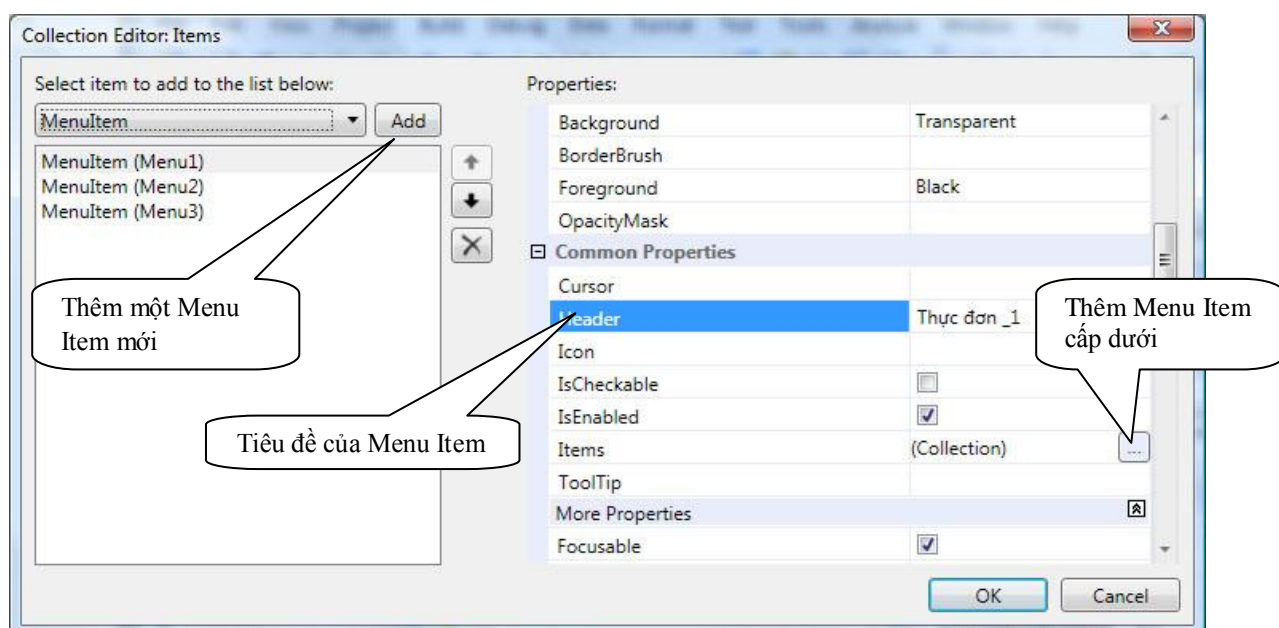
Tạo Menu bằng công cụ trực quan.

Xem hình 4.4 minh họa công cụ tạo Menu cho ứng dụng WPF bằng Visual Studio 2008. Chọn menu từ ToolBox, sau đó Nhấn chuột lên cửa sổ để tạo Menu mới. Tiếp đến nhấn nút **(Collection) ...** trên thanh Properties của Menu để mở hộp thoại quản lý các Menu Item như hình 4.5.



Hình 4.4 Công cụ soạn thảo Menu cho ứng dụng WPF trong Visual Studio 2008

Hình 4.5 là hộp thoại quản lý các Menu Item, nhấn nút **Add** để thêm Menu Item và nhập tiêu đề cho Menu Item ở mục Header. Nếu muốn thêm các Menu Item cấp dưới của Menu Item hiện tại, nhấn nút **(Collection) ...**.



Hình 4.5. Hộp thoại Quản lý các Menu Item

Sử dụng mã XAML để tạo thực đơn

Đoạn mã trình Menu trên bằng XAML:

```
<Grid>

    <Menu Height="22" Name="menu1" VerticalAlignment="Top" >
        <MenuItem Header="Thực đơn _1" Name="Menu1">
            <MenuItem Header="_Copy" Command="ApplicationCommands.Copy"
ToolTip="Copy văn bản"/>
            <MenuItem Header="_Cut" Command="ApplicationCommands.Cut" ToolTip="Cắt
văn bản"/>
            <MenuItem Header="_Paste" Command="ApplicationCommands.Paste"
ToolTip="Dán văn bản"/>
        </MenuItem>
        <MenuItem Header="Thực đơn _2" Name="Menu2">
            <MenuItem Header="Thực đơn 21">
                <MenuItem Header="Thực đơn 211" Click="MenuItem211_Click" />
                <MenuItem Header="Thực đơn 212" Click="MenuItem212_Click" />
            </MenuItem>
            <MenuItem Header="Thực đơn 22" Click="MenuItem22_Click" />
        </MenuItem>
        <MenuItem Header="Thực đơn _3" Click="MenuItem3_Click" Name="Menu3" />
    </Menu>
</Grid>
```

Thanh Menu được bắt đầu bằng thẻ `<Menu>` và kết thúc bằng thẻ đóng `</Menu>`. Có nhiều thuộc tính của thẻ này, trong ví dụ trên thì

`Height="22"` : Chiều cao menu là 22 pixel.

`Name="menu1"` : Tên của menu là menu1. Tên menu được mã trình C# sử dụng để quản lý nó.

`VerticalAlignment="Top"` : Menu được căn để nằm bên trên của Grid chứa nó.

Các **Popup Menu** được tạo bởi thẻ `<MenuItem>` và kết thúc bằng thẻ đóng `</MenuItem>`. Giữa cặp thẻ này là các thẻ `<MenuItem>` khác để tạo nên các Menu Item cấp dưới của nó.

Các **Command Menu** thì được tạo bởi thẻ `<MenuItem/>`, không có thẻ đóng. Một số thuộc tính cơ bản của Menu Item bao gồm

`Header="Thực đơn _1"`: Tiêu đề hay nhãn của Menu Item. Dấu gạch dưới đặt trước ký tự sẽ được sử dụng làm **phím tắt khi kết hợp với phím Alt** để gọi Menu Item bằng bàn phím. Trong ví dụ này thì ký tự 1 được dùng làm phím tắt cho Menu Item “Thực đơn 1”, ký tự được dùng làm phím tắt sẽ được hiển thị với dấu gạch chân khi người dùng nhấn phím Alt để mở Menu.

`Name="Menu1"`: Tên của Menu Item, cần thiết cho mã trình C# có thể can thiệp vào Menu Item.

`ToolTip="Cắt văn bản"`: Lời chú thích cho Menu Item khi di chuột qua.

Đối với các Command Menu, có hai cơ chế thực thi lệnh khi chọn Menu.

Nếu muốn gắn Command Menu với các lệnh có sẵn của hệ thống như: Copy, Cut, Paste,... thì ta sử dụng thuộc tính **Command** của Menu Item. Ví dụ, lệnh `<MenuItem Header="_Copy" Command="ApplicationCommands.Copy"/>` làm cho Menu Item Copy này sẽ thực hiện công việc copy dòng văn bản đang được chọn trong cửa sổ vào bộ nhớ đệm. Chú ý, các lệnh của hệ thống bắt đầu bằng `ApplicationCommands`.

Nếu muốn gắn Command Menu với các hàm xử lý sự kiện tự định nghĩa thì sử dụng thuộc tính **Click** của Menu Item. Ví dụ, `<MenuItem Header="Thực đơn 211" Click="MenuItem211_Click" />` để yêu cầu khi chọn "Thực đơn 211" thì sẽ gọi hàm `MenuItem211_Click`.

Dưới đây là đoạn mã trình C# khai báo các hàm xử lý sự kiện khi nhấn vào các Menu khai báo bằng XAML trên. Khi các menu được chọn thì các hàm tương ứng được khai báo trong thuộc tính **Click** sẽ được gọi và làm nhiệm vụ đơn giản làm hiển thị hộp thông báo.

```
namespace WpfApplication1
{
    /// <summary>
    /// Interaction logic for Window1.xaml
```



```

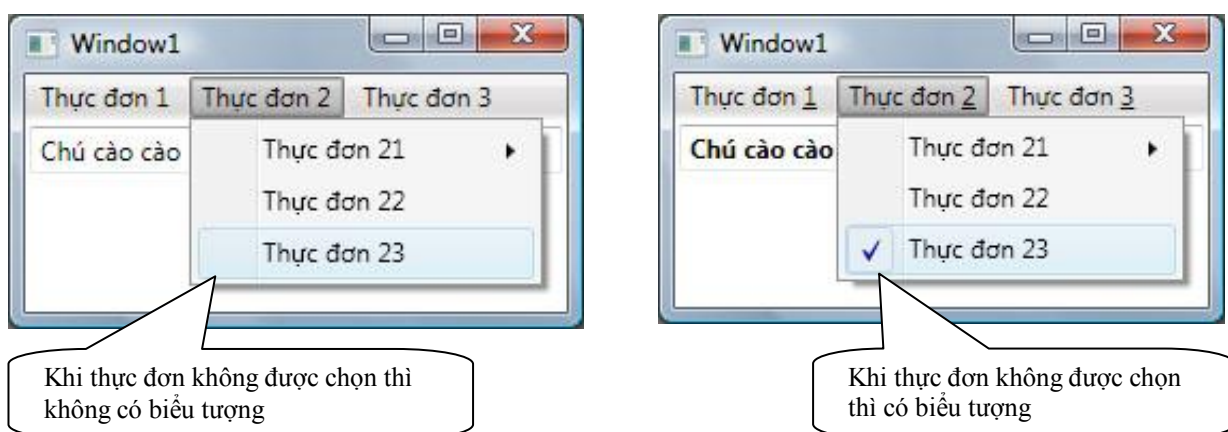
/// </summary>
public partial class Window1 : Window
{
    public Window1()
    {
        InitializeComponent();
    }

    //Hàm xử lý sự kiện nhấn Menu "Thực đơn 211"
    private void MenuItem211_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Bạn chọn Menu 211");
    }
    //Hàm xử lý sự kiện nhấn Menu "Thực đơn 212"
    private void MenuItem212_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Bạn chọn Menu 212");
    }
    //Hàm xử lý sự kiện nhấn Menu "Thực đơn 22"
    private void MenuItem22_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Bạn chọn Menu 22");
    }
    //Hàm xử lý sự kiện nhấn Menu "Thực đơn 3"
    private void MenuItem3_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Bạn chọn Menu 3");
    }
}
}

```

1.2 Menu Item với trạng thái Checked và Unchecked

Khi làm việc với Menu, đôi khi ta có những chức năng với đặc thù có hai trạng thái On/Off. Ví dụ như chương cần có một Menu Item để làm cho một Textbox hiển thị ở dạng chữ đậm và chữ thường, người dùng mong muốn Menu Item thể hiện được trạng thái On/Off tương ứng với kiểu chữ (chữ đậm/chữ thường) trên Textbox. Điều khiển Menu của WPF cung cấp cho chúng ta loại Menu Item với hai trạng thái Checked và UnChecked.



Hình 4.6. Minh họa Menu Item với trạng thái Checked và UnChecked

Để tạo ra Menu Item có trạng thái, ta sử dụng thuộc tính `IsCheckable="True"` của Menu Item.

Mã lệnh tạo "Thực đơn 23" như sau:

```

<MenuItem Header="Thực đơn _2" Name="Menu2">
    <MenuItem Header="Thực đơn 21">
        <MenuItem Header="Thực đơn 211" Click="MenuItem211_Click" />
    </MenuItem>
    <MenuItem Header="Thực đơn 22">
        <MenuItem Header="Thực đơn 221" Click="MenuItem221_Click" />
    </MenuItem>
    <MenuItem Header="Thực đơn 23" Click="MenuItem23_Click" />
</MenuItem>

```

```

        <MenuItem Header="Thực đơn 212" Click="MenuItem212_Click" />
    </MenuItem>
    <MenuItem Header="Thực đơn 22" Click="MenuItem22_Click" />
    <!--Thực đơn có trạng thái Checked và UnChecked-->
    <MenuItem Header="Thực đơn 23" IsCheckable="True" Checked="Menu23_Checked"
UnChecked="Menu23_Unchecked"/>
</MenuItem>

```

Đối với Menu Item có trạng thái, mỗi khi Menu được chọn sẽ phát sinh một trong hai sự kiện Checked và Unchecked tương ứng. Ta sử dụng các thuộc tính **Checked** và **UnChecked** để gắn các hàm xử lý sự kiện cần được thực thi. Ở ví dụ trên, hàm **Menu23_Checked** sẽ được gọi khi Menu "Thực đơn 23" được đánh dấu, và hàm **Menu23_Unchecked** được gọi khi Menu "Thực đơn 23" được bỏ dấu chọn.

Mã nguồn minh họa của hai hàm trên như sau:

```

public partial class Window1 : Window
{
    public Window1()
    {
        InitializeComponent();
    }
    //.....
    //Hàm xử lý sự kiện khi Menu "Thực đơn 23" được đánh dấu chọn
    private void Menu23_Checked(object sender, RoutedEventArgs e)
    {
        //Thiết lập thuộc tính chữ đậm cho textBox1
        textBox1.FontWeight = FontWeights.Bold;
    }
    //Hàm xử lý sự kiện khi Menu "Thực đơn 23" được bỏ dấu chọn
    private void Menu23_Unchecked(object sender, RoutedEventArgs e)
    {
        //Thiết lập thuộc tính chữ thường cho textBox1
        textBox1.FontWeight = FontWeights.Normal;
    }
}

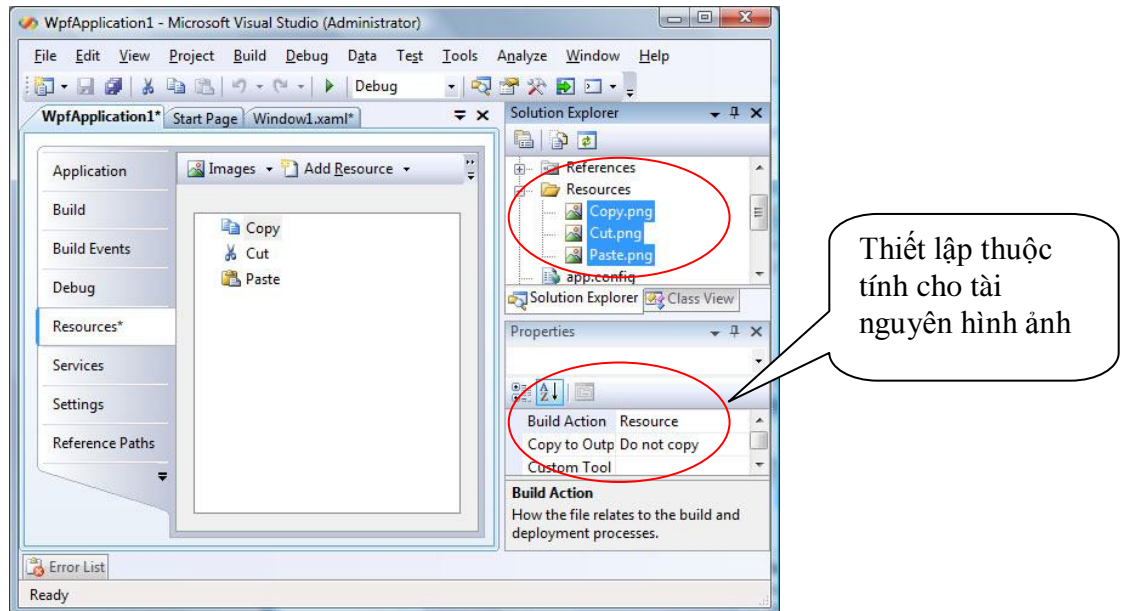
```

1.3 Menu Item với biểu tượng hình ảnh

Với các ứng dụng xây dựng trên nền .Net 2.0, công việc xây dựng Thực đơn và Thanh công cụ với biểu tượng hình ảnh là khá đơn giản, có thể sử dụng công cụ thiết kế giao diện trực quan. Ví dụ, muốn tạo menu có biểu tượng hình ảnh (icon), chúng ta thêm Menu Strip vào form, sau đó thêm các mục cho thực đơn (Menu Item). Nhấn chuột phải lên từng mục và chọn "**Set Image**" là thành công. Tuy nhiên, khi xây dựng ứng dụng WPF, không có mục chọn "**Set Image**" khi nhấn chuột phải vào một mục trong thực đơn. Chúng ta phải nạp các tệp hình ảnh, biểu tượng và tài nguyên (Resource) của ứng dụng và viết một số lệnh XAML để gắn biểu tượng cho Menu Item.

Nạp các tệp hình ảnh, biểu tượng vào tài nguyên của ứng dụng.

- i. Trên thanh thực đơn Visual Studio, chọn Project → Properties sẽ hiện ra bảng cài đặt các thông số cài đặt cho ứng dụng.
- ii. Chọn mục resources.
- iii. Trong mục Add Resource chọn Add Existing File nếu đã có sẵn File biểu tượng hình ảnh trên máy hoặc chọn New Images hay Add New Icon tùy ý.
- iv. Chú ý, sau khi thêm được các File hình ảnh biểu tượng vào tài nguyên, để các điều khiển trên cửa sổ như Menu, Toolbar sử dụng được chúng, ta phải thiết lập thuộc tính '**Build Action : Resource**' và '**Copy to Output Directory : Do not copy**'.



Hình 4.7. Thêm tài nguyên và thiết lập thuộc tính cho tài nguyên ảnh

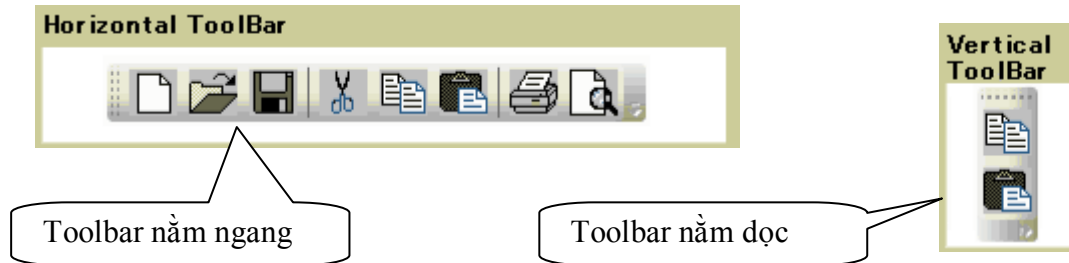
Viết mã lệnh XAML gắn biểu tượng cho các Menu Item.

```
<StackPanel>
<!--Khai báo Tài nguyên cần sử dụng-->
<StackPanel.Resources>
<Image x:Key="copy" Source="Resources/Copy.png" Width="16" Height="16"></Image>
<Image x:Key="cut" Source="Resources/Cut.png" Width="16" Height="16"></Image>
<Image x:Key="paste" Source="Resources/Paste.png" Width="16" Height="16"></Image>
</StackPanel.Resources>
<Menu Height="22" Name="menu1" VerticalAlignment="Top" >
  <MenuItem Header="Thực đơn _1" Name="Menu1">
    <!--Ba Menu Copy, Cut, Paste được gắn biểu tượng-->
    <MenuItem Header="_Copy" Command="ApplicationCommands.Copy" ToolTip="Copy văn bản" Icon="{StaticResource copy}" />
    <MenuItem Header="_Cut" Command="ApplicationCommands.Cut" ToolTip="Cắt văn bản" Icon="{StaticResource cut}" />
    <MenuItem Header="_Paste" Command="ApplicationCommands.Paste" ToolTip="Dán văn bản" Icon="{StaticResource paste}" />
  </MenuItem>
  <MenuItem Header="Thực đơn _2" Name="Menu2">
    <MenuItem Header="Thực đơn 21">
      <MenuItem Header="Thực đơn 211" Click="MenuItem211_Click" />
      <MenuItem Header="Thực đơn 212" Click="MenuItem212_Click" />
    </MenuItem>
    <MenuItem Header="Thực đơn 22" Click="MenuItem22_Click" />
    <!--Thực đơn có trạng thái Checked và UnChecked-->
    <MenuItem Header="Thực đơn 23" IsCheckable="True" Checked="Menu23_Checked" Unchecked="Menu23_Unchecked"/>
  </MenuItem>
  <MenuItem Header="Thực đơn _3" Click="MenuItem3_Click" Name="Menu3" />
</Menu>
<TextBox Name="textBox1" TextWrapping="Wrap" Margin="2">
  Chú cào cào nhỏ, ngồi trong đám cỏ.
</TextBox>
</StackPanel>
```

Như vậy, chúng ta đã xây dựng thành công thanh menu với các biểu tượng đẹp mắt hay menu với trạng thái Checked/UnChecked cũng như biết cách gắn các hàm xử lý sự kiện cho các menu. Phần tiếp theo ta sẽ tìm hiểu về thanh công cụ.

2. Xây dựng và sử dụng thanh công cụ (Toolbar)

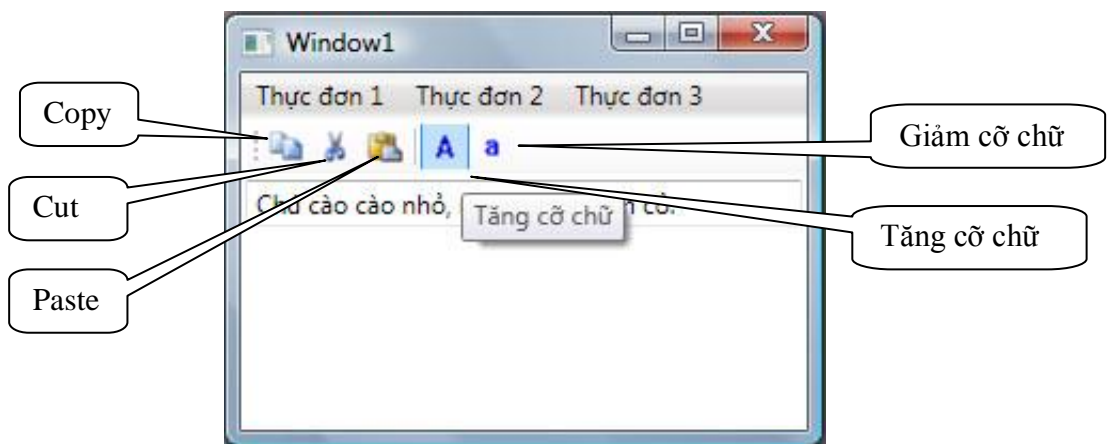
Thanh công cụ (Toolbar) là thanh chứa các chức năng dưới dạng các dãy hình ảnh biểu tượng, mỗi biểu tượng gắn với một mục chức năng cụ thể. Thông thường các Toolbar chứa những chức năng thiết yếu mà người dùng hay quan tâm nhất, bởi vì thanh Toolbar có ưu điểm là dễ dàng thao tác. Một cửa sổ có thể có một hoặc nhiều thanh Toolbar. Trong phần này ta tìm hiểu phương pháp xây dựng thanh Toolbar với các nút chức năng thông thường và các nút chức năng có trạng thái (Checked/UnChecked).



Hình 4.8. Thanh công cụ - Toolbar

2.1. Nút công cụ thông thường

Chúng ta bắt đầu tìm hiểu các bước xây dựng thanh Toolbar với các nút bấm đơn giản như ví dụ minh họa ở hình 4.9. Thanh công cụ bao gồm năm nút: Copy, Cut, Paste thực hiện các chức năng có sẵn của hệ thống, Nút **A** và **a** gắn với hàm xử lý sự kiện tự xây dựng, làm nhiệm vụ tăng/giảm cỡ chữ của Textbox bên dưới.



Hình 4.9. Ví dụ minh họa Toolbar

Thanh công cụ được xây dựng bằng đoạn mã XAML sau:

```
<StackPanel>
<ToolBar Height="26" Name="toolBar1" Width="280" HorizontalAlignment="Left" >
    <Button Height="23" Name="button1" Width="23"
Command="ApplicationCommands.Copy" ToolTip="Copy văn bản">
        <Image Source="Resources/Copy.png" Width="16" Height="16"
HorizontalAlignment="Left" />
    </Button>
    <Button Height="23" Name="button2" Width="23" Command="ApplicationCommands.Cut"
ToolTip="Cắt văn bản">
        <Image Source="Resources/Cut.png" Width="16" Height="16"
HorizontalAlignment="Left" />
    </Button>
    <Button Height="23" Name="button3" Width="23"
Command="ApplicationCommands.Paste" ToolTip="Dán văn bản">
        <Image Source="Resources/Paste.png" Width="16" Height="16"
HorizontalAlignment="Left" />
    </Button>
```



```

        <Separator/>
        <Button Height="23" Name="button4" ToolTip="Tăng cỡ chữ" Width="23"
Click="IncreaseFont_Click">
            <Image Source="Resources/inc.ico" Width="16" Height="16"
HorizontalAlignment="Left" />
        </Button>
        <Button Height="23" Name="button5" ToolTip="Giảm cỡ chữ" Width="23"
Click="DecreaseFont_Click">
            <Image Source="Resources/Dec.ico" Width="16" Height="16"
HorizontalAlignment="Left" />
        </Button>
    </ToolBar>
</StackPanel>

```

Mã XAML tạo thanh công cụ được bắt đầu bằng thẻ `<ToolBar>` và kết thúc bằng thẻ đóng `</ToolBar>`.

Các nút lệnh (Button) của thanh công cụ được tạo bởi thẻ `<Button>` và kết thúc bằng thẻ đóng `</Button>`.

`Name=" button1"`: Tên của Button, cần thiết cho mã trình C# có thể can thiệp vào Button.

`ToolTip="Copy văn bản "`: Lời chú thích cho Button khi di chuột qua.

Có hai cơ chế thực thi lệnh khi chọn nút lệnh trong Toolbar.

Nếu muốn gắn nút lệnh với các lệnh có sẵn của hệ thống như: Copy, Cut, Paste,.. thì ta sử dụng thuộc tính **Command** của Button. Ví dụ, lệnh `<Button Height="23" Name="button1" Width="23" Command="ApplicationCommands.Copy" ToolTip="Copy văn bản">` làm cho nút lệnh Copy này sẽ thực hiện công việc copy dòng văn bản đang được chọn trong cửa sổ vào bộ nhớ đệm. Chú ý, các lệnh của hệ thống bắt đầu bằng `ApplicationCommands`.

Nếu muốn gắn nút lệnh với các hàm xử lý sự kiện tự định nghĩa thì sử dụng thuộc tính **Click** của Button. Ví dụ, `<Button Height="23" Name="button4" ToolTip="Tăng cỡ chữ" Width="23" Click="IncreaseFont_Click">` để yêu cầu khi chọn "button4" thì sẽ gọi hàm `IncreaseFont_Click`.

Giữa cặp thẻ `<Button>` và `</Button>` là thẻ `<Image Source="Resources/Copy.png" Width="16" Height="16" HorizontalAlignment="Left" />` để định nghĩa hình ảnh biểu tượng của nút bấm.

Thẻ `<Separator/>` dùng để tạo ra vạch phân cách giữa các nút bấm.

Mã nguồn minh họa của hai hàm xử lý sự kiện nhấn nút `button4` và `button5` như sau:

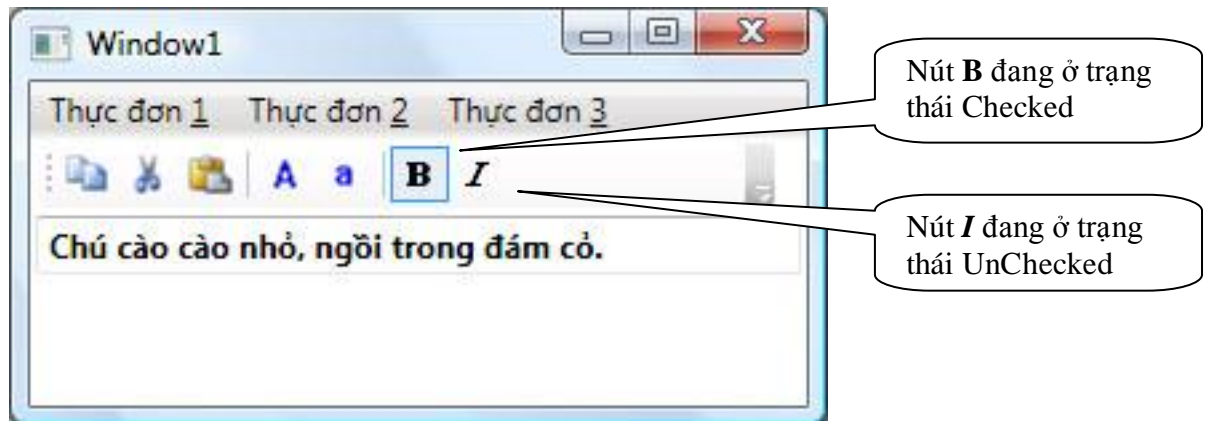
```

public partial class Window1 : Window
{
    public Window1()
    {
        InitializeComponent();
    }
    //.....
    //Hàm xử lý sự kiện khi button4 được nhấn
    private void IncreaseFont_Click(object sender, RoutedEventArgs e)
    {
        if (textBox1.FontSize < 18)
        {
            textBox1.FontSize += 2; //Tăng cỡ font chữ của textBox1 lên 2 đơn vị
        }
    }
    //Hàm xử lý sự kiện khi button5 được nhấn
    private void DecreaseFont_Click(object sender, RoutedEventArgs e)
    {
        if (textBox1.FontSize > 10)
        {
            textBox1.FontSize -= 2; //Giảm cỡ font chữ của textBox1 lên 2 đơn vị
        }
    }
}

```

2.2. Nút công cụ có trạng thái

Ngoài các nút bấm thông thường, thanh công cụ còn cho phép tạo ra các nút bấm có trạng thái, khi ở trạng thái được chọn (Checked) thì sẽ có màu nền khác và có đường viền để người dùng có thể nhận biết được trạng thái của nút đó (xem minh họa hình 4.10).



Hình 4.10. Ví dụ minh họa Toolbar với trạng thái Checked và UnChecked

Thanh công cụ với các nút có trạng thái được xây dựng bằng đoạn mã XAML sau:

```
<StackPanel>
<ToolBar Height="26" Name="toolBar1" Width="280" HorizontalAlignment="Left" >
    <Button Height="23" Name="button1" Width="23"
    Command="ApplicationCommands.Copy" ToolTip="Copy văn bản">
        <Image Source="Resources/Copy.png" Width="16" Height="16"
        HorizontalAlignment="Left" />
    </Button>
    <Button Height="23" Name="button2" Width="23" Command="ApplicationCommands.Cut"
    ToolTip="Cắt văn bản">
        <Image Source="Resources/Cut.png" Width="16" Height="16"
        HorizontalAlignment="Left" />
    </Button>
    <Button Height="23" Name="button3" Width="23"
    Command="ApplicationCommands.Paste" ToolTip="Dán văn bản">
        <Image Source="Resources/Paste.png" Width="16" Height="16"
        HorizontalAlignment="Left" />
    </Button>

    <Separator/>
    <Button Height="23" Name="button4" ToolTip="Tăng cỡ chữ" Width="23"
    Click="IncreaseFont_Click">
        <Image Source="Resources/inc.ico" Width="16" Height="16"
        HorizontalAlignment="Left" />
    </Button>
    <Button Height="23" Name="button5" ToolTip="Giảm cỡ chữ" Width="23"
    Click="DecreaseFont_Click">
        <Image Source="Resources/Dec.ico" Width="16" Height="16"
        HorizontalAlignment="Left" />
    </Button>
    <Separator/>
    <!--Nút bấm với trạng thái Checked và UnChecked-->
    <CheckBox Name="check1" ToolTip="Chữ đậm" Checked="Bold_Checked"
    Unchecked="Bold_Unchecked">
        <Image Source="Resources/bold.png" Width="16" Height="16"
        HorizontalAlignment="Left" />
    </CheckBox>
</ToolBar>
</StackPanel>
```

```

        </CheckBox>
        <CheckBox Name="check2" ToolTip="Chữ nghiêng" Checked="Italic_Checked"
Unchecked="Italic_Unchecked">
            <Image Source="Resources/Italic.png" Width="16" Height="16"
HorizontalAlignment="Left" />
        </CheckBox>
    </ToolBar>

</StackPanel>

```

Khác với các nút lệnh thông thường được tạo bởi thẻ `<Button>` và kết thúc bằng thẻ đóng `</Button>`, các nút lệnh có trạng thái được tạo nên bởi thẻ `<CheckBox Name="check1" ToolTip="Chữ đậm" Checked="Bold_Checked" Unchecked="Bold_Unchecked">` và kết thúc bằng thẻ đóng `</CheckBox>`.

Nút lệnh có trạng thái phát sinh hai sự kiện `Checked` và `Unchecked`, tương ứng với trạng thái của nút là được chọn hay bỏ chọn khi người dùng nhấn nút. Trong ví dụ trên, khi nút `check1` được chọn thì hàm `Bold_Checked` được gọi và nút `check1` bỏ chọn thì hàm `Bold_Unchecked` được gọi. Hai hàm này do ta tự xây dựng với mã lệnh như sau.

Mã nguồn minh họa của hai hàm xử lý sự kiện nhấn nút `check1`:

```

public partial class Window1 : Window
{
    public Window1()
    {
        InitializeComponent();
    }
    //.....
    private void Bold_Checked(object sender, RoutedEventArgs e)
    {
        //Thiết lập thuộc tính chữ đậm cho textBox1
        textBox1.FontWeight = FontWeights.Bold;
    }

    private void Bold_Unchecked(object sender, RoutedEventArgs e)
    {
        //Thiết lập thuộc tính chữ thường cho textBox1
        textBox1.FontWeight = FontWeights.Normal;
    }

    private void Italic_Checked(object sender, RoutedEventArgs e)
    {
        //Thiết lập thuộc tính chữ nghiêng cho textBox1
        textBox1.FontStyle = FontStyles.Italic;
    }

    private void Italic_Unchecked(object sender, RoutedEventArgs e)
    {
        //Thiết lập thuộc tính chữ thẳng đứng cho textBox1
        textBox1.FontStyle = FontStyles.Normal;
    }
}

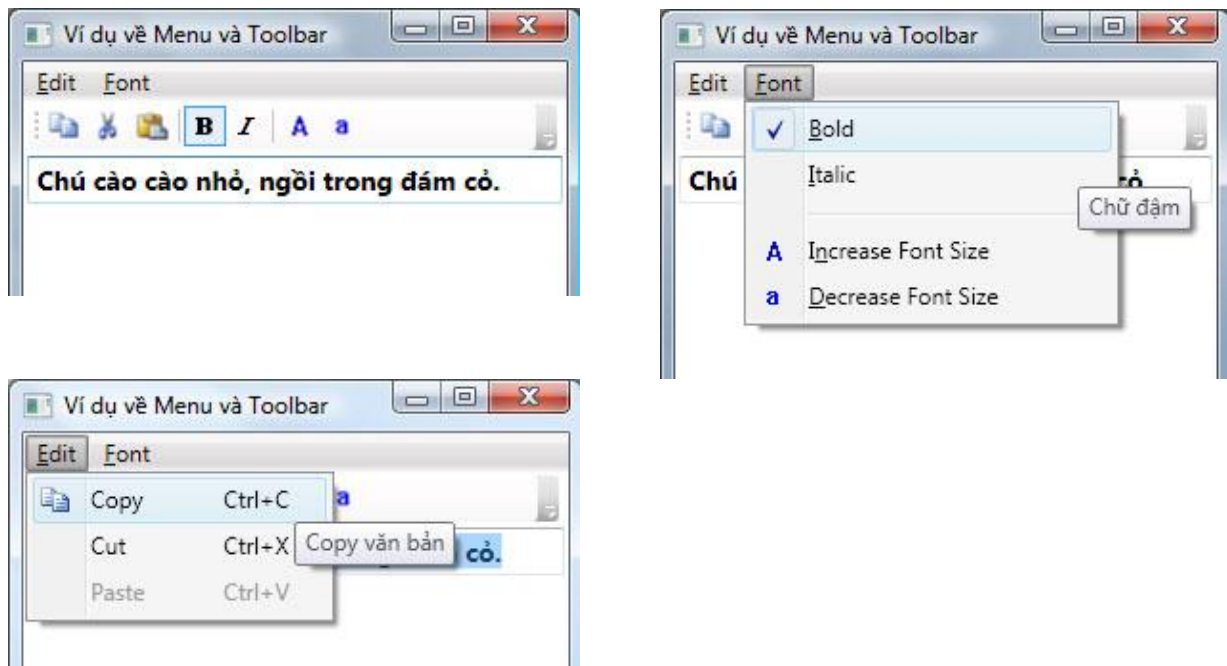
```

Ngoài các nút bấm có trạng thái kiểu này, thanh công cụ còn cho phép tạo ra các nút bấm có trạng thái loại trừ nhau bằng cách sử dụng thẻ `<RadioButton>` để tạo ra nút bấm.

Như vậy, chúng ta đã hoàn tất bài học xây dựng và quản lý thanh Thực đơn và thanh công cụ. Tiếp theo là ví dụ tổng hợp lại các kiến thức trên.

3. Ví dụ tổng hợp về xây dựng Menu và Toolbar

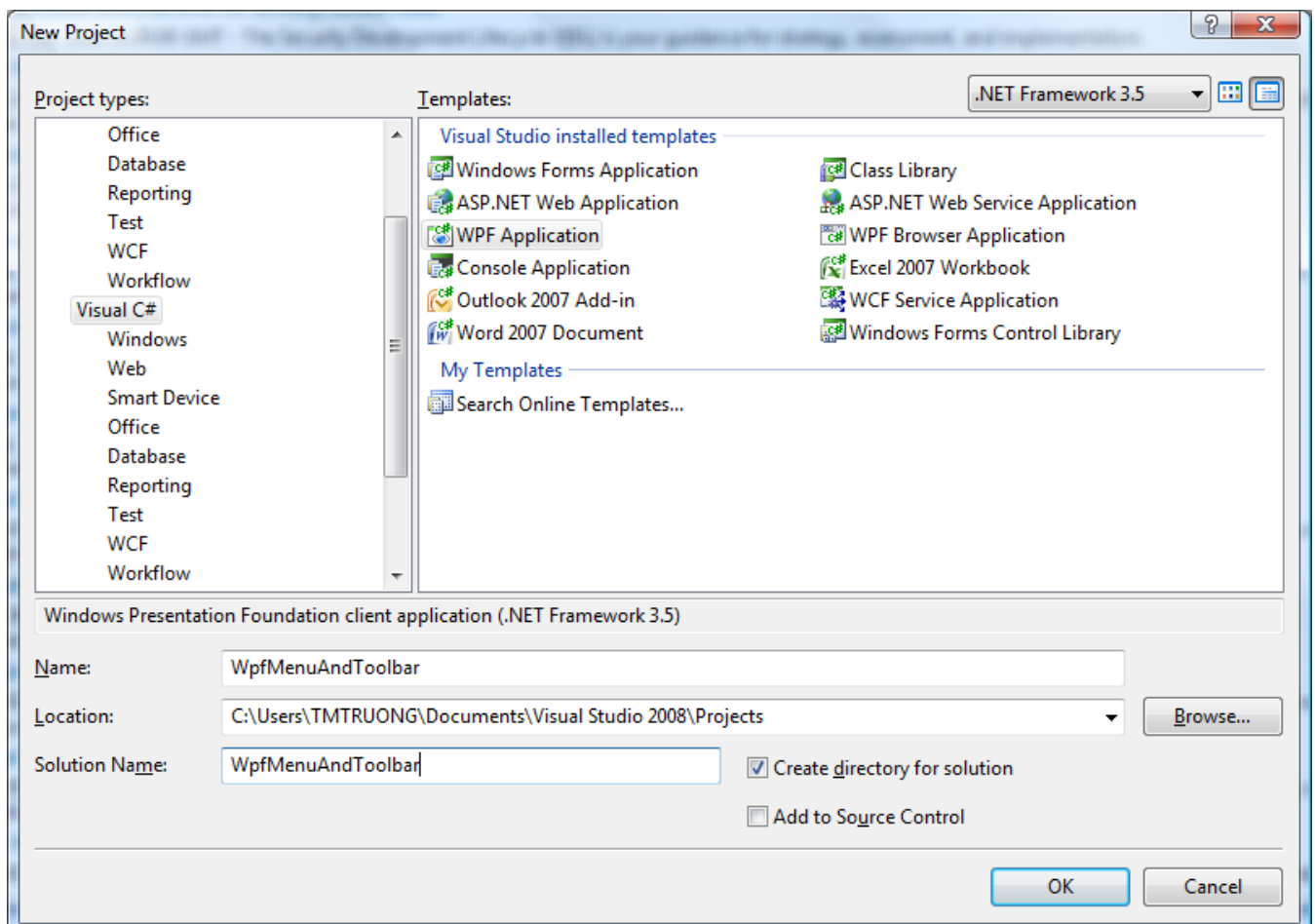
Xây dựng cửa sổ bao gồm một thanh thực đơn, một thanh công cụ và một hộp soạn thảo như minh họa ở hình 4.11



Hình 4.11. Ví dụ minh họa tổng hợp về Menu và Toolbar

Các bước thực hiện như sau:

3.1 Tạo ứng dụng WPF



Ở đây, ta chú ý chọn .Net Framework 3.5.

3.2 Mã XAML của giao diện

Mở file Window1.xaml tương ứng với file code Window1.xaml.cs.

```
<Window x:Class="WpfMenuAndToolbar.Window1"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Ví dụ về Menu và Toolbar" Height="300" Width="300"
>
    <StackPanel >
        <!--Khai báo Tài nguyên cần sử dụng-->
        <StackPanel.Resources>
            <Image x:Key="inc" Source="Resources/inc.ico" Width="16"
Height="16"/></Image>
            <Image x:Key="dec" Source="Resources/dec.ico" Width="16"
Height="16"/></Image>
            <Image x:Key="copy" Source="Resources/Copy.png" Width="16"
Height="16"/></Image>
        </StackPanel.Resources>
        <!--Khai báo thanh thực đơn-->
        <Menu Name="Menu_main">
            <MenuItem Header="_Edit" >
                <MenuItem Command="ApplicationCommands.Copy" ToolTip="Copy văn bản"
Icon="{StaticResource copy}"/>
                <MenuItem Command="ApplicationCommands.Cut" ToolTip="Cắt văn bản"/>
                <MenuItem Command="ApplicationCommands.Paste" ToolTip="Dán văn
bản"/>
            </MenuItem>
            <MenuItem Header="_Font" >
                <MenuItem Header="_Bold" Name="Menu21" ToolTip="Chữ đậm"
IsCheckable="True"
                Checked="Bold_Checked"
                Unchecked="Bold_Unchecked"/>
                <MenuItem Header="_Italic" Name="Menu22" ToolTip="Chữ nghiêng"
IsCheckable="True"
                Checked="Italic_Checked"
                Unchecked="Italic_Unchecked"/>
                <Separator/>
                <MenuItem Header="I_ncrease Font Size" ToolTip="Tăng cỡ chữ"
Icon="{StaticResource inc}"
                Click="IncreaseFont_Click"/>
                <MenuItem Header="_Decrease Font Size" ToolTip="Giảm cỡ chữ"
Icon="{StaticResource dec}"
                Click="DecreaseFont_Click"/>
            </MenuItem>
        </Menu>
        <!--Khai báo thanh công cụ-->
        <ToolBar Height="26" Name="toolBar1" Width="280" HorizontalAlignment="Left"
>
            <Button Height="23" Name="button1" Width="23"
Command="ApplicationCommands.Copy" ToolTip="Copy văn bản">
                <Image Source="Resources/Copy.png" Width="16" Height="16"
HorizontalAlignment="Left" />
            </Button>
            <Button Height="23" Name="button2" Width="23"
Command="ApplicationCommands.Cut" ToolTip="Cắt văn bản">
                <Image Source="Resources/Cut.png" Width="16" Height="16"
HorizontalAlignment="Left" />
            </Button>
            <Button Height="23" Name="button3" Width="23"
Command="ApplicationCommands.Paste" ToolTip="Dán văn bản">
                <Image Source="Resources/Paste.png" Width="16" Height="16"
HorizontalAlignment="Left" />
            </Button>
```



```

        <Separator/>
        <CheckBox Name="check1" ToolTip="Chữ đậm" Checked="Bold_Checked"
Unchecked="Bold_Unchecked">
            <Image Source="Resources/bold.png" Width="16" Height="16"
HorizontalAlignment="Left" />
        </CheckBox>
        <CheckBox Name="check2" ToolTip="Chữ nghiêng" Checked="Italic_Checked"
Unchecked="Italic_Unchecked">
            <Image Source="Resources/Italic.png" Width="16" Height="16"
HorizontalAlignment="Left" />
        </CheckBox>
        <Separator/>
        <Button Height="23" Name="button6" ToolTip="Tăng cỡ chữ" Width="23"
Click="IncreaseFont_Click">
            <Image Source="Resources/inc.ico" Width="16" Height="16"
HorizontalAlignment="Left" />
        </Button>
        <Button Height="23" Name="button7" ToolTip="Giảm cỡ chữ" Width="23"
Click="DecreaseFont_Click">
            <Image Source="Resources/Dec.ico" Width="16" Height="16"
HorizontalAlignment="Left" />
        </Button>
    </ToolBar>
    <!--Khai báo hộp soạn thảo-->
    <TextBox Name="textBox1" TextWrapping="Wrap"
        Margin="2">
        Chú cào cào nhỏ, ngồi trong đám cỏ.
    </TextBox>

</StackPanel>
</Window>

```

3.3 Mã lệnh C# xử lý các sự kiện

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace WpfMenuAndToolbar
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>

    public partial class Window1 : System.Windows.Window
    {
        public Window1()
        {
            InitializeComponent();
        }

        private void Bold_Checked(object sender, RoutedEventArgs e)
        {
            textBox1.FontWeight = FontWeights.Bold;
            check1.IsChecked = true;
        }
    }
}

```

```

        Menu21.IsChecked = true;
    }

    private void Bold_Unchecked(object sender, RoutedEventArgs e)
    {
        textBox1.FontWeight = FontWeights.Normal;
        check1.IsChecked = false;
        Menu21.IsChecked = false;
    }

    private void Italic_Checked(object sender, RoutedEventArgs e)
    {
        textBox1.FontStyle = FontStyles.Italic;
        check2.IsChecked = true;
        Menu22.IsChecked = true;
    }

    private void Italic_Unchecked(object sender, RoutedEventArgs e)
    {
        textBox1.FontStyle = FontStyles.Normal;
        check2.IsChecked = false;
        Menu22.IsChecked = false;
    }

    private void IncreaseFont_Click(object sender, RoutedEventArgs e)
    {
        if (textBox1.FontSize < 18)
        {
            textBox1.FontSize += 2;
        }
    }

    private void DecreaseFont_Click(object sender, RoutedEventArgs e)
    {
        if (textBox1.FontSize > 10)
        {
            textBox1.FontSize -= 2;
        }
    }
}

```

F5 để chạy ứng dụng. Ta sẽ thu được kết quả tương tự như yêu cầu.

4. Câu hỏi ôn tập

1. Trên thanh Menu, các Menu Item mức đỉnh chỉ có thể là Popup Menu ?

- A. Đúng
- B. Sai

Câu trả lời: B

2. Menu Item mức dưới của một Popup Menu cũng có thể là một Popup Menu?

- A. Đúng
- B. Sai

Câu trả lời: A

3. Thuộc tính nào sau đây của Menu được dùng để gán nhãn (tiêu đề) cho Menu?

- A. Title
- B. Header
- C. Text
- D. Tooltip

Câu trả lời: B

4. Các thuộc tính được có thể được sử dụng để gán lệnh cho một Menu Item (Chọn nhiều)

- A. Command
- B. Click
- C. OnClick
- D. Checked và UnChecked

Câu trả lời: A, B và D

5. Thanh công cụ được phép nằm ở vị trí nào trên cửa sổ

- A. Nằm ngang.
- B. Nằm dọc
- C. Được phép nằm cả theo chiều dọc và chiều ngang.

Câu trả lời: C

6. Những thẻ nào sau đây được dùng để tạo các nút trên thực đơn (Chọn nhiều):

- A. <Button>
- B. <Checkbox>
- C. <ToolBar Item>

Câu trả lời: A và B

5. Tài liệu tham khảo

1. Windows Presentation Foundation, URL: <http://msdn.microsoft.com/en-us/library/ms754130.aspx>.
2. WPF - XAML Introduction, URL: <http://homepage.ntlworld.com/herring1/xamlintro.html> .
3. Menu Icons in WPF, URL: <http://anuraj.wordpress.com/2008/06/23/menu-icons-in-wpf/> .
4. WPF Sample Series, URL: <http://karlshifflett.wordpress.com/2008/01/23/wpf-sample-series-stretch-toolbar-width-of-window/> .