XAML UI Controls Lab - WPF

Perform these labs on your own computer using Visual Studio 2022 or later to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Check Box

Open the **Views\UserDetailView.xaml** file and add a new row before the button row.

Add the following.

```
<Label Content="Employee Perks"
        Grid.Row="6" />
<StackPanel Orientation="Horizontal"
            VerticalAlignment="Center"
            Grid.Row="6"
            Grid.Column="1">
  <CheckBox Content="401k" />
  <CheckBox Content="Health Care" />
  <CheckBox Content="HSA" />
  <CheckBox Content="Flex-Time" />
</StackPanel>
```

Be sure to go back to the <RowDefinition> elements and change the new row definition you added from "80" to "Auto".

Try It Out

Run the application and click on the **Users** menu to view the results.

Notice that the check box is not aligned.

Fix Check Box Alignment

Open the **Resources\Styles\StandardStyles.xaml** file and add a new style.

```
<Style TargetType="CheckBox"
        BasedOn="{StaticResource BaseFrameworkElement}">
  <Setter Property="VerticalAlignment"
          Value="Center" />
  <Setter Property="VerticalContentAlignment"
          Value="Center" />
</Style>
```

Try It Out

Run the application and click on the **Users** menu to view the results.

Demo 2: Radio Button

Open the **Views\UserDetailView.xaml** file and add a new row before the button row.

Add the following.

```
<Label Content="Still Employed?"
        Grid.Row="7" />
<StackPanel Orientation="Horizontal"
            Grid.Row="7"
            Grid.Column="1">
  <RadioButton Content="Yes" IsChecked="True" />
  <RadioButton Content="No" />
</StackPanel>
```

Be sure to go back to the <RowDefinition> elements and change the new row definition you added from "80" to "Auto".

Try It Out

Run the application and click on the **Users** menu to view the results.

Notice that the radio button is not aligned.

Fix Radio Button Alignment

Open the **Resources\Styles\StandardStyles.xaml** file and add a new style.

```
<Style TargetType="RadioButton"
        BasedOn="{StaticResource BaseFrameworkElement}">
  <Setter Property="VerticalAlignment"
          Value="Center" />
  <Setter Property="VerticalContentAlignment"
          Value="Center" />
</Style>
```

Try It Out

Run the application and click on the **Users** menu to view the results.

Demo 3: ComboBox

Open the **Views\UserDetailView.xaml** file and add a new row before the button row.

Add the following.

```
<Label Content="Phone"
        Grid.Row="8" />
<Grid Grid.Row="8"
      Grid.Column="1">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="Auto" />
  </Grid.ColumnDefinitions>
  <TextBox HorizontalAlignment="Stretch" />
  <ComboBox Grid.Column="1">
    <ComboBoxItem IsSelected="True">Home</ComboBoxItem>
    <ComboBoxItem>Mobile</ComboBoxItem>
    <ComboBoxItem>Other</ComboBoxItem>
  </ComboBox>
</Grid>
```

Be sure to go back to the <RowDefinition> elements and change the new row definition you added from "80" to "Auto".

Open the **Resources\Styles\StandardStyles.xaml** file and add a new style.

```
<Style TargetType="ComboBox"
       BasedOn="{StaticResource BaseControl}">
</Style>
```

Try It Out

Run the application and click on the **Users** menu to view the results.

Demo 4: Date Picker

Open the **Views\UserDetailView.xaml** file and add a new row before the button row.

Add the following.

```
<Label Content="Birth Date"
        Grid.Row="9" />
<DatePicker Grid.Row="9"
            Grid.Column="1"
            HorizontalAlignment="Left" />
```

Be sure to go back to the <RowDefinition> elements and change the new row definition you added from "80" to "Auto".

Open the **Resources\Styles\StandardStyles.xaml** file and add a new style.

```
<Style TargetType="DatePicker"
       BasedOn="{StaticResource BaseControl}">
</Style>
```

Try It Out

Run the application and click on the **Users** menu to view the results.

Demo 5: Border

Open the **Views\HomeView.xaml** file and add a <Border> around the <Label>.

```
<StackPanel VerticalAlignment="Center"
            HorizontalAlignment="Stretch">
  <Border BorderBrush="Black"
          HorizontalAlignment="Center"
          BorderThickness="2">
    <Label FontSize="48"
           HorizontalAlignment="Center"
           Content="{StaticResource ApplicationTitle}"
/>
  </Border>
</StackPanel>
```

Try It Out

Run the application and view the border around the application title.

Demo 6: Add a Border Around all Views

Let's adjust all the views so they look more like a normal screen.

Open the **StandardStyles.xaml** file and add a new keyed style.

```
<Style TargetType="Border"
       x:Key="Screen.Border">
  <Setter Property="BorderBrush"
          Value="Black" />
  <Setter Property="BorderThickness"
          Value="2" />
  <Setter Property="Margin"
          Value="5" />
  <Setter Property="HorizontalAlignment"
          Value="Left" />
  <Setter Property="VerticalAlignment"
          Value="Top" />
  <Setter Property="MinWidth"
          Value="600" />
</Style>
```

Open the **Views\LoginView.xaml** file and wrap a <Border> around the <Grid>.

```
<Border Style="{StaticResource Screen.Border}">
```

Open the **Views\ProductDetail.xaml** file and wrap a <Border> around the <Grid>.

```
<Border Style="{StaticResource Screen.Border}">
```

Open the **Views\UserDetailView.xaml** file and wrap a <Border> around the <Grid>.

```
<Border Style="{StaticResource Screen.Border}">
```

Try It Out

Run the application and click on each menu item to view the border around each screen.

Demo 7: ScrollViewer

Run the application and click on the **Products** menu.

Reduce the main window to show how the screen gets cut off.

Stop the application.

Open the **MainWindow.xaml** file and replace the <Grid x:Name="ContentArea"> element with the following code.

```
<ScrollViewer Grid.Row="1"
              HorizontalScrollBarVisibility="Auto"
              VerticalScrollBarVisibility="Auto">
  <Grid x:Name="ContentArea" />
</ScrollViewer>
```

Try It Out

Run the application and click on the **Products** menu, reduce the main window size to show that it now displays a scroll bar.

Demo 8: GroupBox

Open the **Views\ProductDetailView.xaml** file.

Locate the **Start Selling Date** row.

Delete the <Label> and <TextBox> controls for Selling Start Date, Selling End Date, and the Discontinued Date fields.

Add the following code where this row is.

```
<GroupBox Grid.Row="10"
          Grid.ColumnSpan="2"
          Header="Product Dates">
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto" />
      <RowDefinition Height="Auto" />
      <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto" />
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Label Content="Selling Start Date" />
    <TextBox Grid.Column="1" />
    <Label Content="Selling End Date"
           Grid.Row="1" />
    <TextBox Grid.Row="1"
             Grid.Column="1" />
    <Label Content="Discontinued Date"
           Grid.Row="2" />
    <TextBox Grid.Row="2"
             Grid.Column="1" />
  </Grid>
</GroupBox>
```

Change the Grid.Row property on the <StackPanel> following the </GroupBox> to the number "**11**".

You can now remove two <RowDefinition> elements from the <Grid>.

Open the **Resources\Styles\StandardStyles.xaml** file and add a new style.

```
<Style TargetType="GroupBox"
       BasedOn="{StaticResource BaseControl}">
  <Setter Property="BorderThickness"
          Value="2" />
</Style>
```

Try It Out

Run the application and click on the **Products** menu to view the result.

Demo 9: Expander

Open the **Views\ProductDetailView.xaml** file.

Change the GroupBox from the previous demo to an <Expander>

Try It Out

Run the application and click on the **Products** menu to view the result.

Demo 10: Tab

Open the **Views\ProductDetailView.xaml** file.

Remove the <Grid.ColumnDefinitions> from the <Grid>.

Modify the <Grid.RowDefinitions> to the following:

```
<Grid.RowDefinitions>
  <RowDefinition Height="Auto" />
  <RowDefinition Height="*" />
  <RowDefinition Height="Auto" />
</Grid.RowDefinitions>
```

Remove **Grid.ColumnSpan="2"** from the <ReusableUI:HeaderView> control.

Just after the <ReusableUI:HeaderView> control, add the following.

```
<TabControl Grid.Row="1">
  <TabItem Header="Main Product Info">
  </TabItem>
  <TabItem Header="Additional Info">
  </TabItem>
  <TabItem Header="Product Dates">
  </TabItem>
</TabControl>
```

"Main Product Info" TabItem

In the <TabItem Header="Main Product Info"> element, add the following <Grid> definition.

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>
</Grid>
```

Now, cut the controls in Grid.Row's "1" through "5" and paste them just after the </Grid.RowDefinition>.

Renumber the rows from 0 through 5.

"Additional Info" TabItem

In the <TabItem Header="Additional Info"> element, add the following <Grid> definition.

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>
</Grid>
```

Now, cut the controls in Grid.Row's "6" through "9" and paste them just after the </Grid.RowDefinition>.

Renumber the rows from 0 through 3.

Within the "Product Dates" TabItem

Cut the <Grid> out from within the <Expander> and place it into this last TabItem.

Delete the <Expander>

Modify the <StackPanel> to be in **Grid.Row="2"**.

Add Style for Tab Control

Open the **StandardStyles.xaml** file and add a new style.

```
<Style TargetType="TabControl"
       BasedOn="{StaticResource BaseFrameworkElement}">
</Style>
```

Try It Out

Run the application and click on the **Products** menu to view the result.