

Designing the layout

To complete the app's layout, design attributes can be added to the XAML element tags to specify what they will display and precisely where in the interface they will appear:



Universal
(continued)



A single **Margin** value sets all four margins around that component. You can specify two values to set left & right, top & bottom margins, e.g. **Margin = "10,30"**. Alternatively, you can specify four values to set left,top,right,bottom margins individually, e.g. **Margin = "10,30,10,50"**.

- 1 Open **MainPage.xaml**, then add two attributes to the outer **<StackPanel>** element to fix its position
HorizontalAlignment = "Left" VerticalAlignment = "Top"
- 2 Next, edit the **<Image>** element by modifying the initial assigned value of 200 – to increase its width
Width = "300"
- 3 Now, add an attribute to the nested **<StackPanel>** element to fix its position
VerticalAlignment = "Center"
- 4 Then, edit all six **<TextBlock>** elements alike, to specify their initial content, width, and margin on all four sides
Text = "..." **Width = "20"** **Margin = "15"**
Text = "..." **Width = "20"** **Margin = "15"**
Text = "..." **Width = "20"** **Margin = "15"**
Text = "..." **Width = "20"** **Margin = "15"**
Text = "..." **Width = "20"** **Margin = "15"**
Text = "..." **Width = "20"** **Margin = "15"**
- 5 Edit the first **<Button>** element to rename it, specify its button label content and margin on all four sides
x:Name = "BtnPick" Content = "Get My Lucky Numbers"
Margin = "15"
- 6 Edit the second **<Button>** element to rename it and specify its button label content
x:Name = "BtnReset" Content = "Reset"
- 7 Finally, add an attribute to each respective **<Button>** element to specify their initial state
IsEnabled = "True"
IsEnabled = "True"

...cont'd

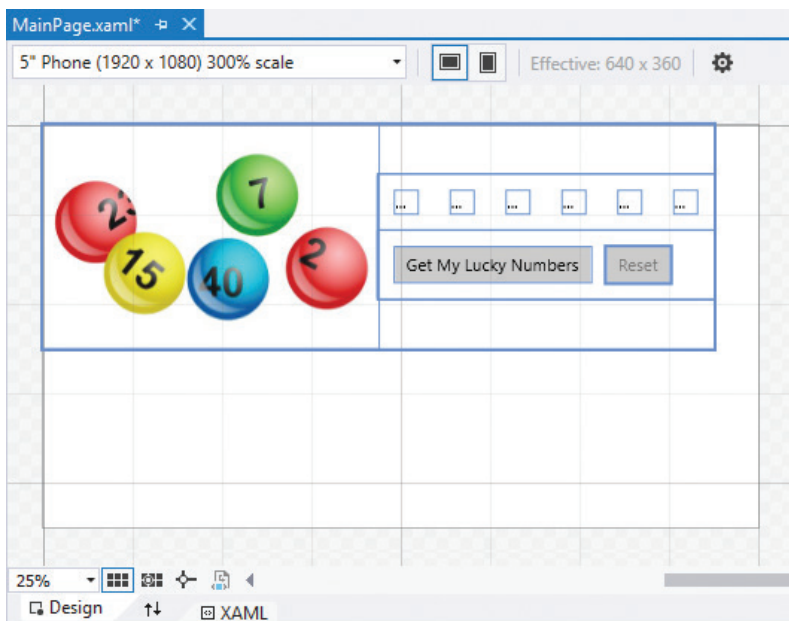
The order in which the attributes appear in each element is unimportant, but the elements within the **MainPage.xaml** file should now look similar to the screenshot below:

```
11 <StackPanel x:Name="MainStack" HorizontalAlignment="Left" VerticalAlignment="top" Orientation="Horizontal" >
12
13 <Image x:Name="Image" Width="300" Height="200" Source="Assets/lotto.jpg" />
14
15 <StackPanel x:Name="Controls" Orientation="Vertical" VerticalAlignment="Center">
16
17 <StackPanel x:Name="Labels" Orientation="Horizontal">
18 <TextBlock x:Name="textBlock1" Text="..." Width="20" Margin="15"/>
19 <TextBlock x:Name="textBlock2" Text="..." Width="20" Margin="15"/>
20 <TextBlock x:Name="textBlock3" Text="..." Width="20" Margin="15"/>
21 <TextBlock x:Name="textBlock4" Text="..." Width="20" Margin="15"/>
22 <TextBlock x:Name="textBlock5" Text="..." Width="20" Margin="15"/>
23 <TextBlock x:Name="textBlock6" Text="..." Width="20" Margin="15"/>
24 </StackPanel>
25
26 <StackPanel x:Name="Buttons" Orientation="Horizontal">
27 <Button x:Name="BtnPick" Content="Get My Lucky Numbers" Margin="15" IsEnabled="True"/>
28 <Button x:Name="BtnReset" Content="Reset" IsEnabled="False"/>
29 </StackPanel>
30 </StackPanel>
31
32 </StackPanel>
33
34 </StackPanel>
```



You can optionally add **Margin = "0"** attributes to explicitly require elements to have no margin width.

As you make changes to the **XAML** view code, the component layout gets changed accordingly in the **Design** view and should now look like this:



The **Design** view shows the components' initial state – the Reset button appears grayed out, as it is not enabled.