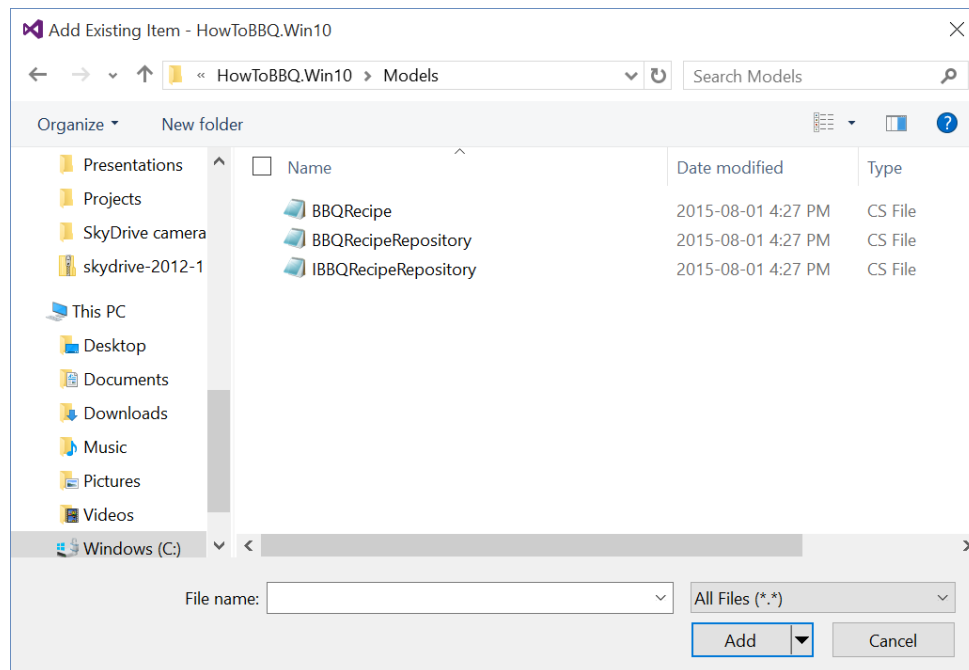


## Windows 10 UWP – Hands on Lab

### Lab 2:

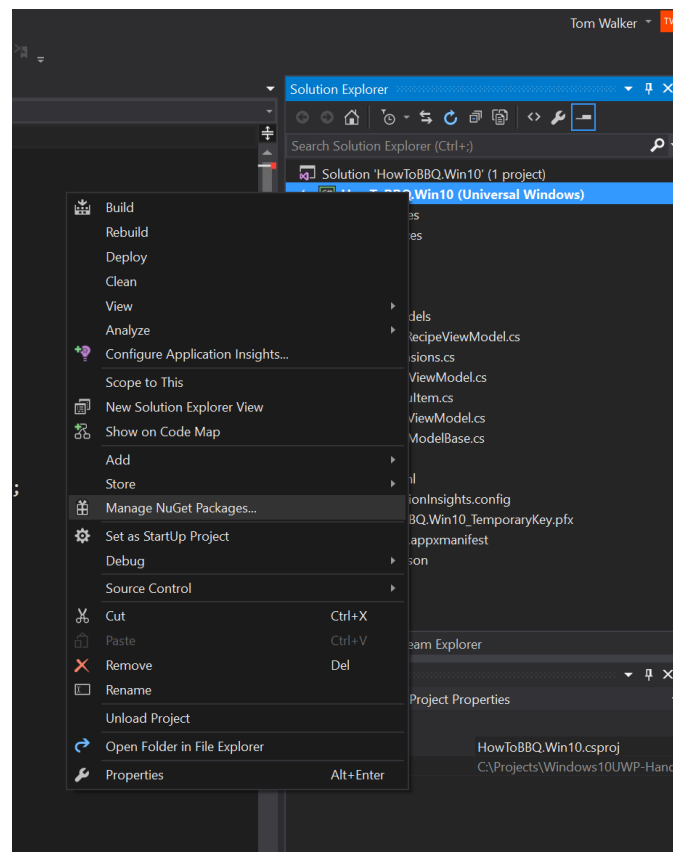
**Note:** This demo app created for this lab uses the Visual Studio 2015 RTM and Windows Tools SDK ver 10240.

1. Select the “Models” folder and bring up the popup menu and select “Add->Existing Item”. This will bring up Windows Explorer. Navigate to the folder (i.e. C:\Projects\Windows10UWP-HandsOnLab\Lab2\Completed) where the lab files have been downloaded. Add the following files located in the “Models” folder (BBQRecipe.cs, BBQRecipeRepository.cs, IBBRecipeRepository.cs)

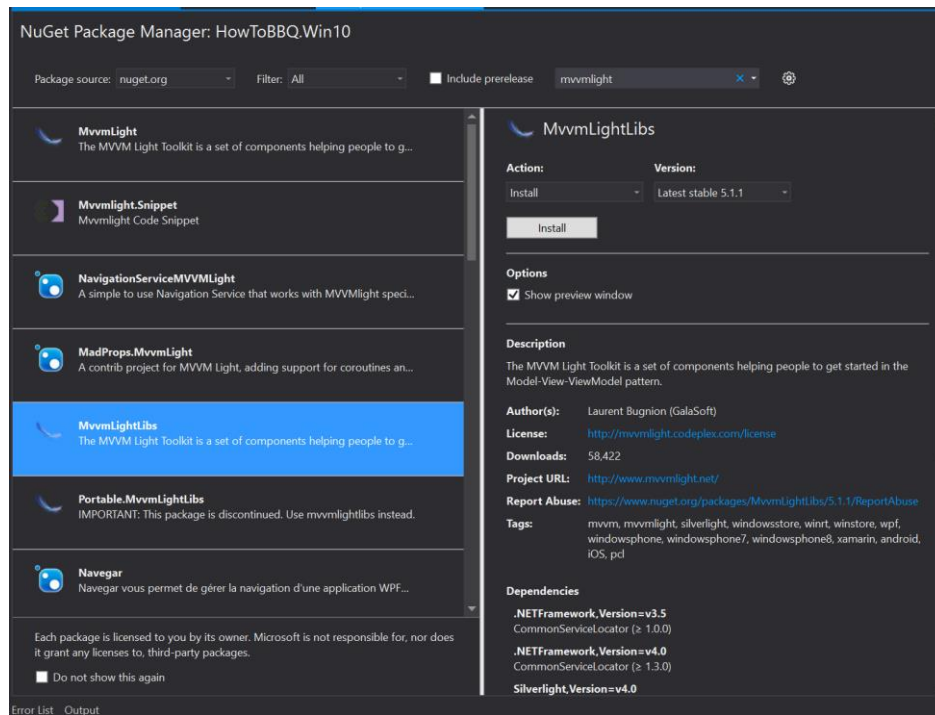


2. Select the “Mvvm” folder and bring up the popup menu and select “Add->Existing Item”. This will bring up Windows Explorer. Navigate to the folder (i.e. C:\Projects\Windows10UWP-HandsOnLab\Lab2\Completed) where the lab files have been downloaded. Add the following files located in the “Mvvm” folder (NavigationService.cs, INavigationService.cs)

3. Select the “ViewModels” folder and bring up the popup menu and select “Add->Existing Item”. This will bring up Windows Explorer. Navigate to the folder (i.e. C:\Projects\Windows10UWP-HandsOnLab\Lab2\Completed) where the lab files have been downloaded. Add the following files located in the “ViewModels” folder (BBQRecipeViewModel.cs, MainViewModel.cs, Extensions.cs).
4. Select the “Assets” folder and bring up the popup menu and select “Add->Existing Item”. This will bring up Windows Explorer. Navigate to the folder (i.e. C:\Projects\Windows10UWP-HandsOnLab\Lab2\Completed) where the lab files have been downloaded. Add the following files located in the “Assets” folder (BBQRecipeViewModel.cs, MainViewModel.cs, Extensions.cs)
5. To use the Navigation service in the pages we need to add the reference to Mvvmlight to use Simpleloc to manage registering and resolve class instances. Bring up the pop menu and select “Manage NuGet Packages”.



6. In the NuGet Package Manager screen, enter “Mvmlight” to begin the search locate “MvvmLightLibs” to install “MvvmLight Library” to the project. Click “install” button to add the library.



7. Open the App.xaml file and change the app theme from “Light” to “Dark”

```
1 <Application
2   x:Class="HowToBBQ.Win10.App"
3   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5   xmlns:local="using:HowToBBQ.Win10"
6   RequestedTheme="Light">
7
8   <Application.Resources>
9     <SolidColorBrush x:Key="JustOneShadeOfGray"
10      Color="LightSlateGray" />
11
12   </Application.Resources>
13
14 </Application>
```

```
1 <Application
2   x:Class="HowToBBQ.Win10.App"
3   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5   xmlns:local="using:HowToBBQ.Win10"
6   RequestedTheme="Dark">
7
8   <Application.Resources>
9     <SolidColorBrush x:Key="JustOneShadeOfGray"
10      Color="LightSlateGray" />
11
12   </Application.Resources>
13
14 </Application>
```

8. Open the App.xaml.cs and add the following code onto line 31 before the App() constructor.

```
public static MainViewModel MainViewModel;
```

```
15 using Windows.UI.Xaml.Input;
16 using Windows.UI.Xaml.Media;
17 using Windows.UI.Xaml.Navigation;
18
19 namespace HowToBBQ.Win10
20 {
21     /// <summary>
22     /// Provides application-specific behavior to supplement the default Application
23     /// </summary>
24     6 references | 0 changes | 0 authors, 0 changes
25     sealed partial class App : Application
26     {
27         /// <summary>
28         /// Initializes the singleton application object. This is the first line of
29         /// executed, and as such is the logical equivalent of main() or WinMain().
30         /// </summary>
31         public static MainViewModel MainViewModel;
32
33         1 reference | 0 changes | 0 authors, 0 changes
34         public App()
35         {
36             Microsoft.ApplicationInsights.WindowsAppInitializer.InitializeAsync(
37                 Microsoft.ApplicationInsights.WindowsCollectors.Metadata |
38                 Microsoft.ApplicationInsights.WindowsCollectors.Session);
39             this.InitializeComponent();
40             this.Suspending += OnSuspending;
41         }
42     }
43 }
```

9. In the App.xaml.cs file and add the following code on to line 68.

```
SimpleIoc.Default.Register<INavigationService>(() => { return new
NavigationService(rootFrame); });
```

```
56     }
57 #endif
58
59     Frame rootFrame = Window.Current.Content as Frame;
60
61     // Do not repeat app initialization when the Window already has content,
62     // just ensure that the window is active
63     if (rootFrame == null)
64     {
65         // Create a Frame to act as the navigation context and navigate to the first page
66         rootFrame = new Frame();
67
68         SimpleIoc.Default.Register<INavigationService>(() => { return new NavigationService(rootFrame); });
69
70         rootFrame.NavigationFailed += OnNavigationFailed;
71
72         if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)
73         {
74             //TODO: Load state from previously suspended application
75         }
76
77         // Place the frame in the current Window
78         Window.Current.Content = rootFrame;
79
80         MainViewModel = new MainViewModel();
81         MainViewModel.IsDataLoaded = false;
82     }
83 }
```

10. In the App.xaml.cs file and add the following code on to line 68.

```
MainViewModel = new MainViewModel();  
MainViewModel.IsDataLoaded = false;
```

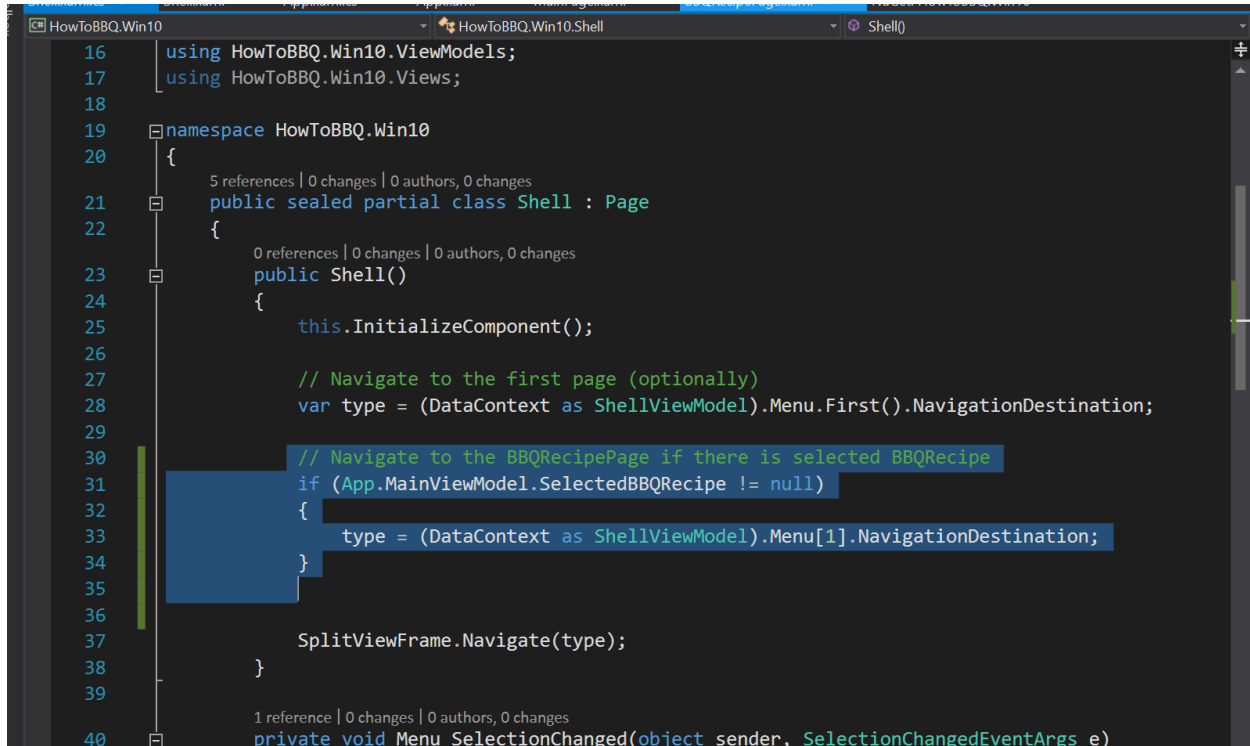
```
69  
70         rootFrame.NavigationFailed += OnNavigationFailed;  
71  
72         if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)  
73         {  
74             //TODO: Load state from previously suspended application  
75         }  
76  
77         // Place the frame in the current Window  
78         Window.Current.Content = rootFrame;  
79  
80         MainViewModel = new MainViewModel();  
81         MainViewModel.IsDataLoaded = false;  
82     }  
83  
84     if (rootFrame.Content == null)  
85     {  
86         // When the navigation stack isn't restored navigate to the first page,  
87         // configuring the new page by passing required information as a navigation  
88         // parameter  
89         rootFrame.Navigate(typeof(Shell), e.Arguments);  
90     }  
91     // Ensure the current window is active  
92     Window.Current.Activate();  
93 }
```

11. In the App.xaml.cs file, add the following code on to the top of the file.

```
using HowToBBQ.Win10.Mvvm;  
using HowToBBQ.Win10.ViewModels;
```

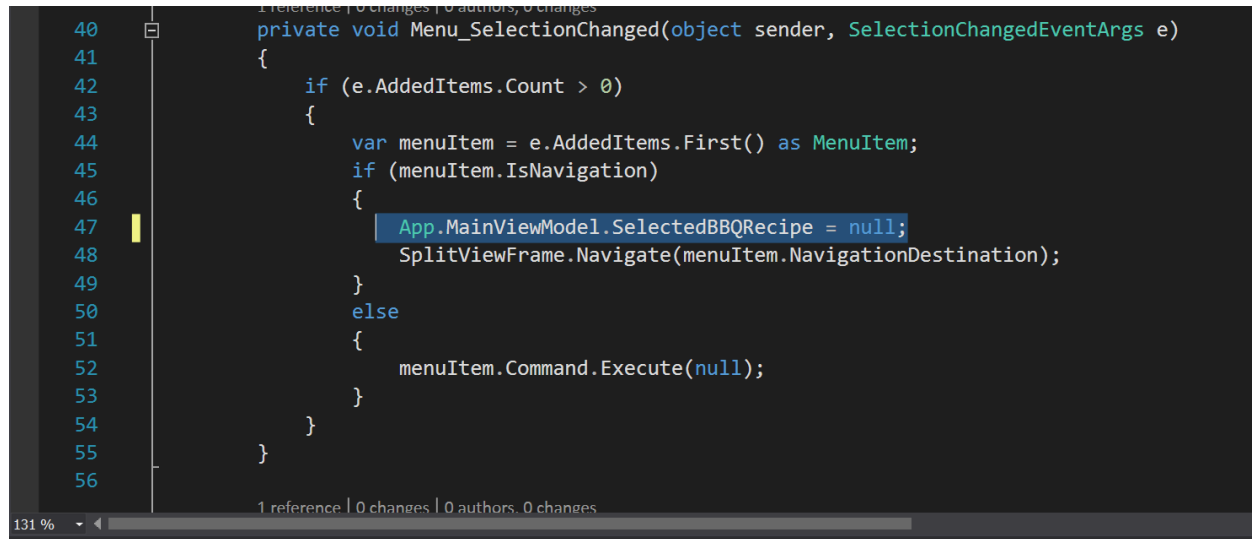
12. Open the file Shell.xaml.cs and add the following lines of code to line 30.

```
// Navigate to the BBQRecipePage if there is selected BBQRecipe
if (App.MainViewModel.SelectedBBQRecipe != null)
{
    type = (DataContext as ShellViewModel).Menu[1].NavigationDestination;
}
```



13. In the file Shell.xaml.cs and add the following lines of code to line 47

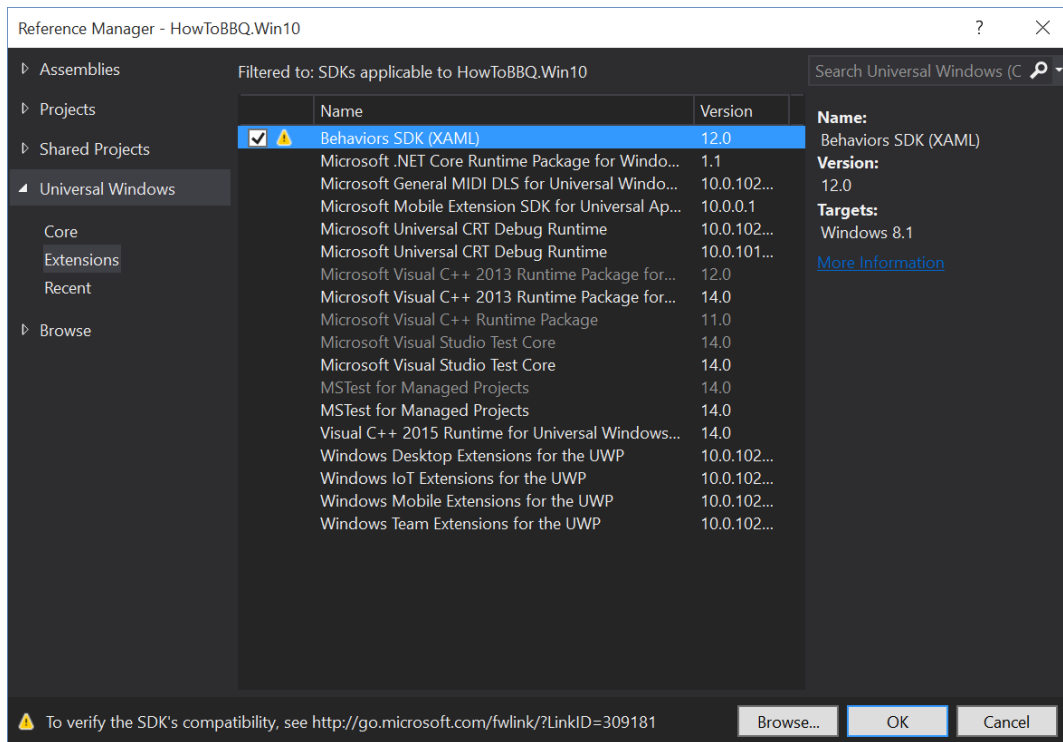
```
App.MainViewModel.SelectedBBQRecipe = null;
```



```
1 reference | 0 changes | 0 authors, 0 changes
40 private void Menu_SelectionChanged(object sender, SelectionChangedEventArgs e)
41 {
42     if (e.AddedItems.Count > 0)
43     {
44         var menuItem = e.AddedItems.First() as MenuItem;
45         if (menuItem.IsNavigation)
46         {
47             App.MainViewModel.SelectedBBQRecipe = null;
48             SplitViewFrame.Navigate(menuItem.NavigationDestination);
49         }
50         else
51         {
52             menuItem.Command.Execute(null);
53         }
54     }
55 }
56
1 reference | 0 changes | 0 authors, 0 changes
131 %
```

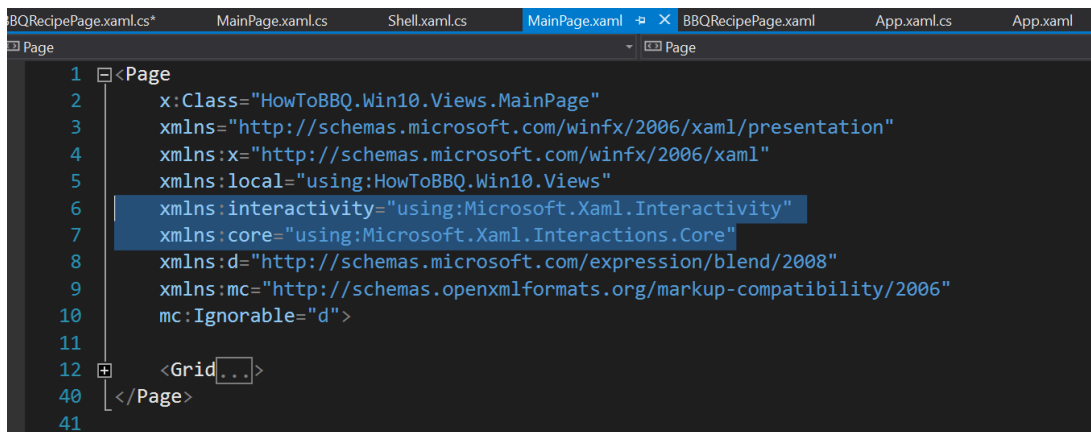


14. Add the following “Behaviors SDK(XAML)” references to the project.



15. Open the MainPage.xaml, at the top of the file add the following lines of code.

```
xmlns:interactivity="using:Microsoft.Xaml.Interactivity"
xmlns:core="using:Microsoft.Xaml.Interactions.Core"
```

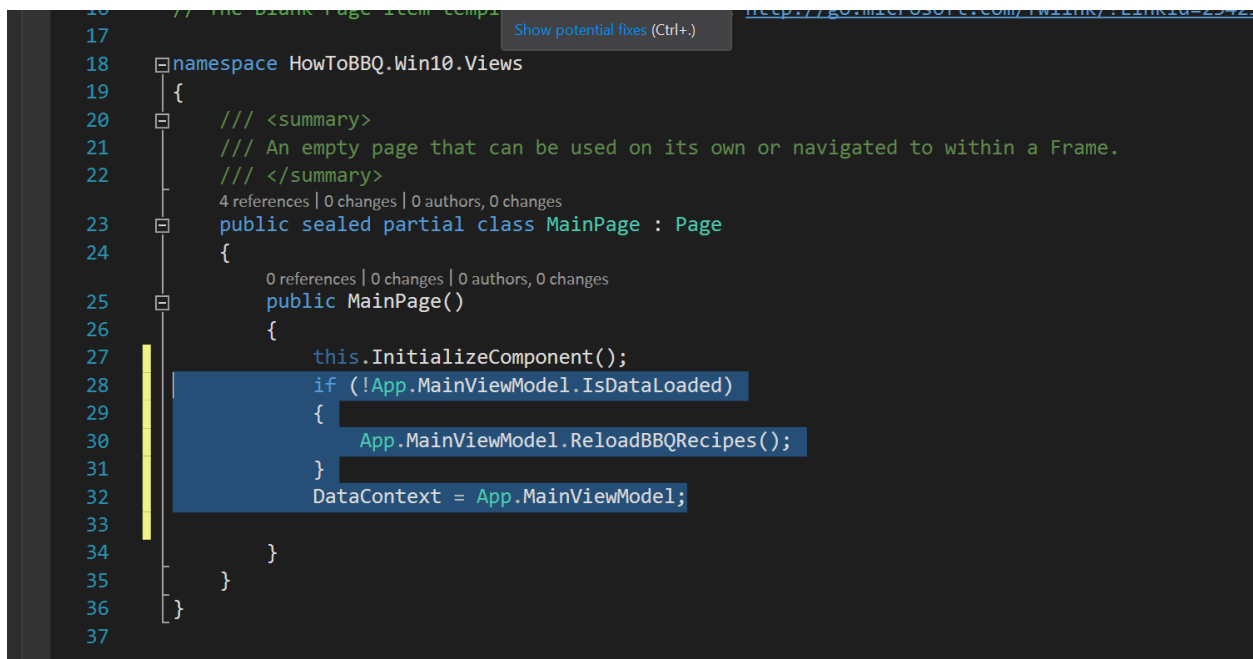


1. In the file MainPage.xaml and add the following lines of xaml code replace the lines of code between the Grid tags.

```
<Grid>
    <GridView x:Name="RecipesGridView" ItemsSource="{Binding Recipes}"
    SelectionMode="Single" IsItemClickEnabled="True" Margin="40,20,30,0"
    HorizontalAlignment="Right">
        <interactivity:Interaction.Behaviors>
            <core:EventTriggerBehavior EventName="ItemClick">
                <core:CallMethodAction MethodName="BBQRecipeTapped"
                TargetObject="{Binding Mode=OneWay}" />
            </core:EventTriggerBehavior>
        </interactivity:Interaction.Behaviors>
        <GridView.ItemContainerStyle>
            <Style TargetType="GridViewItem">
                <Setter Property="Width" Value="600"/>
                <Setter Property="Height" Value="600"/>
                <Setter Property="Margin" Value="5"/>
            </Style>
        </GridView.ItemContainerStyle>
        <GridView.ItemTemplate>
            <DataTemplate>
                <Grid>
                    <Image Source="{Binding ImageSource}"/>
                    <Border Background="#99000000"
    VerticalAlignment="Bottom">
                        <TextBlock Grid.Column="1" Text="{Binding
    Name}"
                                FontSize="30" Margin="10"
    VerticalAlignment="Center"/>
                    </Border>
                </Grid>
            </DataTemplate>
        </GridView.ItemTemplate>
    </GridView>
</Grid>
```

2. Open the file MainPage.xaml.cs and add the following lines of code to line 28.

```
if (!App.MainViewModel.IsDataLoaded)
{
    App.MainViewModel.ReloadBBQRecipes();
}
DataContext = App.MainViewModel;
```



3. Open the file BBQRecipe.xaml and add the following lines of xaml code replace the lines of code between the Grid tags.

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="22*" />
        <ColumnDefinition Width="661*" />
    </Grid.ColumnDefinitions>
    <Grid.ChildrenTransitions>
        <TransitionCollection>
            <EntranceThemeTransition />
        </TransitionCollection>
    </Grid.ChildrenTransitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="140" />
        <RowDefinition Height="577*" />
        <RowDefinition Height="51*" />
    </Grid.RowDefinitions>

    <Grid Grid.Row="1" x:Name="contentRegion" Grid.ColumnSpan="2" Grid.RowSpan="2"
Margin="46,0,50,0">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="500" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <StackPanel Grid.Column="0" Orientation="Vertical">
            <Image x:Name="BBQImage" Source="{Binding ImageSource, Mode=TwoWay}"
Canvas.ZIndex="-3" Height="593" Stretch="UniformToFill" />
        </StackPanel>
        <StackPanel Grid.Column="1" Orientation="Vertical" >
            <StackPanel Margin="50,50,50,25">
                <TextBlock Text="Name" FontSize="14"/>
                <TextBox x:Name="TextBoxTitle" Text="{Binding Name, Mode=TwoWay}"/>
            </StackPanel>
            <StackPanel Margin="50,0,50,25">
                <TextBlock Text="Prep Time" FontSize="14"/>
                <TextBox Text="{Binding PrepTime, Mode=TwoWay}"/>
            </StackPanel>
            <StackPanel Margin="50,0,50,25">
                <TextBlock Text="Total Time" FontSize="14"/>
                <TextBox Text="{Binding TotalTime, Mode=TwoWay}"/>
            </StackPanel>
            <StackPanel Margin="50,0,50,25">
                <TextBlock Text="Serves" FontSize="14"/>
                <TextBox Text="{Binding Serves, Mode=TwoWay}"/>
            </StackPanel>
            <StackPanel Margin="50,0,50,25">
                <TextBlock Text="Ingredients" FontSize="14"/>
                <TextBox Text="{Binding Ingredients, Mode=TwoWay}"/>
            </StackPanel>
            <StackPanel Margin="50,0">
                <TextBlock Text="Directions" FontSize="14"/>
                <TextBox Text="{Binding Directions, Mode=TwoWay}" Height="150"
TextWrapping="Wrap"/>
            </StackPanel>
        </StackPanel>
    </Grid>
</Grid>
```

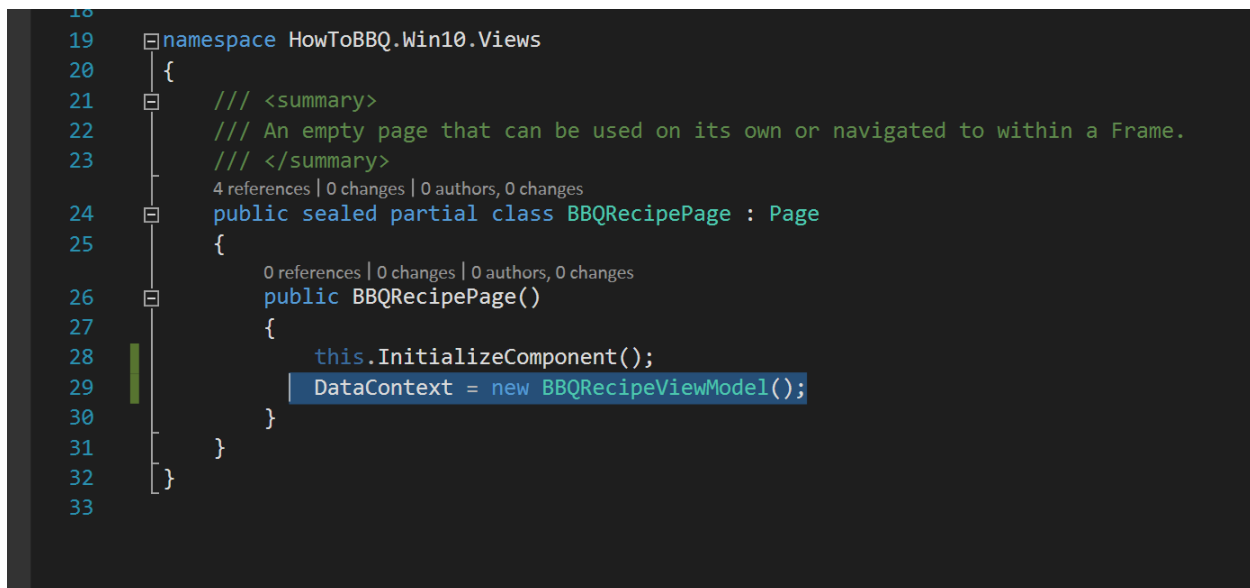
```

        </StackPanel>
    </Grid>
    <StackPanel Grid.ColumnSpan="2" Margin="46,20,0,0">
        <TextBlock x:Name="PageTitle" Text="{Binding Name, Mode=TwoWay}"
Style="{StaticResource HeaderTextBlockStyle}"
IsHitTestVisible="false" TextWrapping="NoWrap"
VerticalAlignment="Bottom"/>
    </StackPanel>
</Grid>

```

4. Open the file BBQRecipe.xaml.cs and add the following line to line 29.

```
DataContext = new BBQRecipeViewModel();
```



5. In the file BBQRecipe.xaml.cs and add the following lines to top of the file.

```

using HowToBBQ.Win10.Models;
using HowToBBQ.Win10.ViewModels;

```