

Windows 10 UWP – Hands on Lab

Lab 6:

Note: The demo app created for this lab uses Visual Studio 2015 with the latest Windows Tools SDK version 10240. To help with our coding efforts with the boiler code and page navigation we use a UWP app extension called Template 10.

1. In the \ViewModels\DetailPageViewModel.cs file and add the following lines after the constructor.

```
private WriteableBitmap bitmap;

private async Task<WriteableBitmap> SelectImageFromPicker()
{
    var picker = new FileOpenPicker();
    picker.ViewMode = PickerViewMode.Thumbnail;
    picker.SuggestedStartLocation = PickerLocationId.PicturesLibrary;
    picker.FileTypeFilter.Add(".jpg");
    picker.FileTypeFilter.Add(".jpeg");
    picker.FileTypeFilter.Add(".png");
    StorageFile file = await picker.PickSingleFileAsync();

    if (file != null)
    {
        ImageProperties imgProp = await file.Properties.GetImagePropertiesAsync();
        var savedPictureStream = await file.OpenAsync(FileAccessMode.Read);

        //set image properties and show the taken photo
        bitmap = new WriteableBitmap((int)imgProp.Width, (int)imgProp.Height);
        await bitmap.SetSourceAsync(savedPictureStream);
        bitmap.Invalidate();

        SaveImageAsync(file);

        return bitmap;
    }
    else return null;
}

private async void SaveImageAsync(StorageFile file)
{
    if (file != null)
    {
        StorageFile newImageFile = await file.CopyAsync(ApplicationData.Current.LocalFolder, Guid.NewGuid().ToString());
        recipe.ImagePath = newImageFile.Path;
    }
}

private async Task<WriteableBitmap> TakePicture()
{
    CameraCaptureUI captureUI = new CameraCaptureUI();
    captureUI.PhotoSettings.Format = CameraCaptureUIPhotoFormat.Jpeg;
    captureUI.PhotoSettings.CroppedSizeInPixels = new Size(600, 600);

    StorageFile photo = await captureUI.CaptureFileAsync(CameraCaptureUIMode.Photo);
    if (photo != null)
    {
        WriteableBitmap bitmap = new WriteableBitmap(600, 600);
        IRandomAccessStream stream = await photo.OpenAsync(FileAccessMode.Read);
        bitmap.SetSource(stream);

        SaveImageAsync(photo);

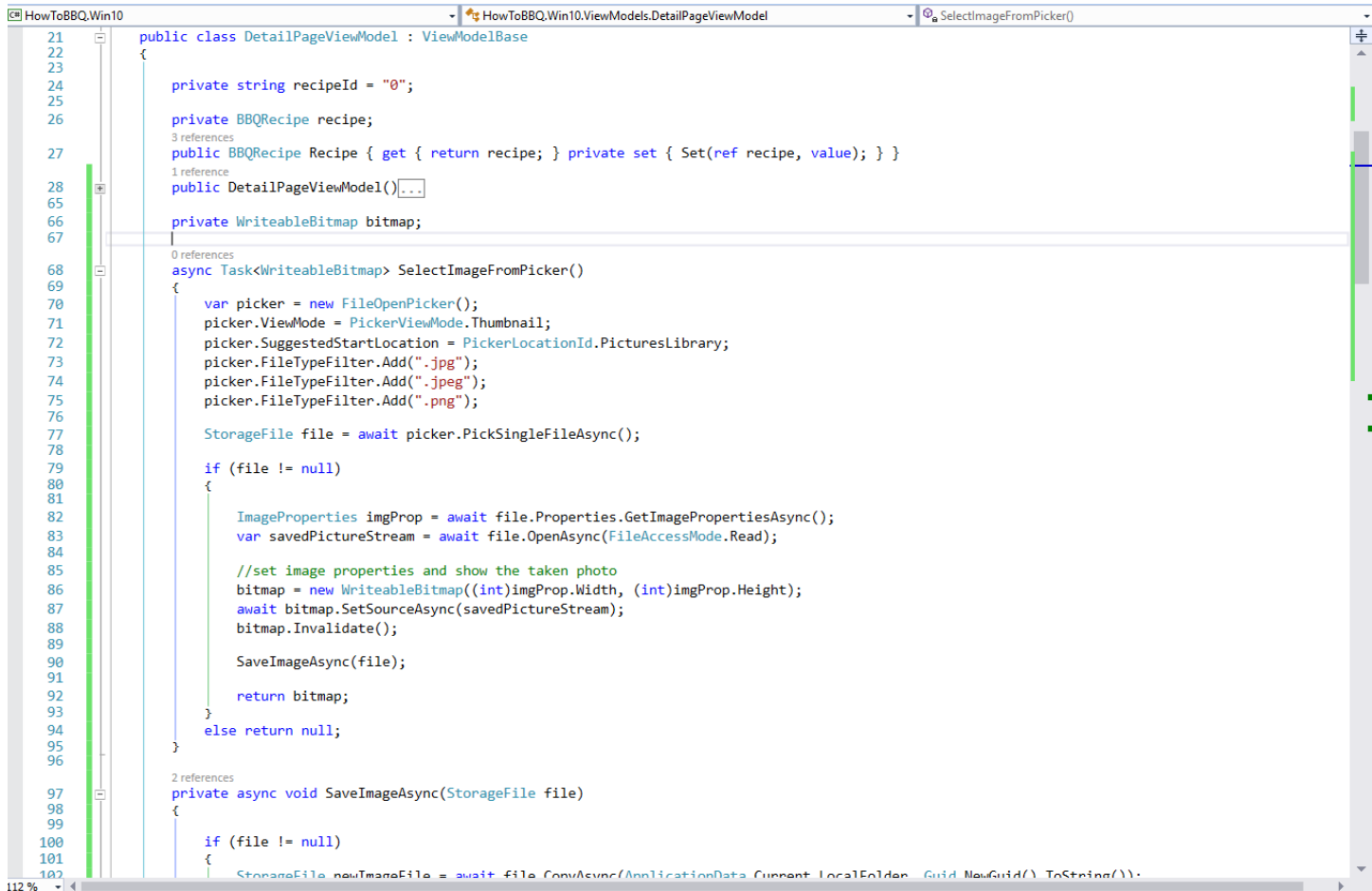
        return bitmap;
    }

    return null;
}
```

```

public async Task<WriteableBitmap> SelectImage(bool useCamera)
{
    if (useCamera) return await TakePicture();
    else return await SelectImageFromPicker();
}

```



- Open the \ViewModels\DetailPageViewModel.cs file and make sure the following using statements match the ones below. Add the missing using statements.

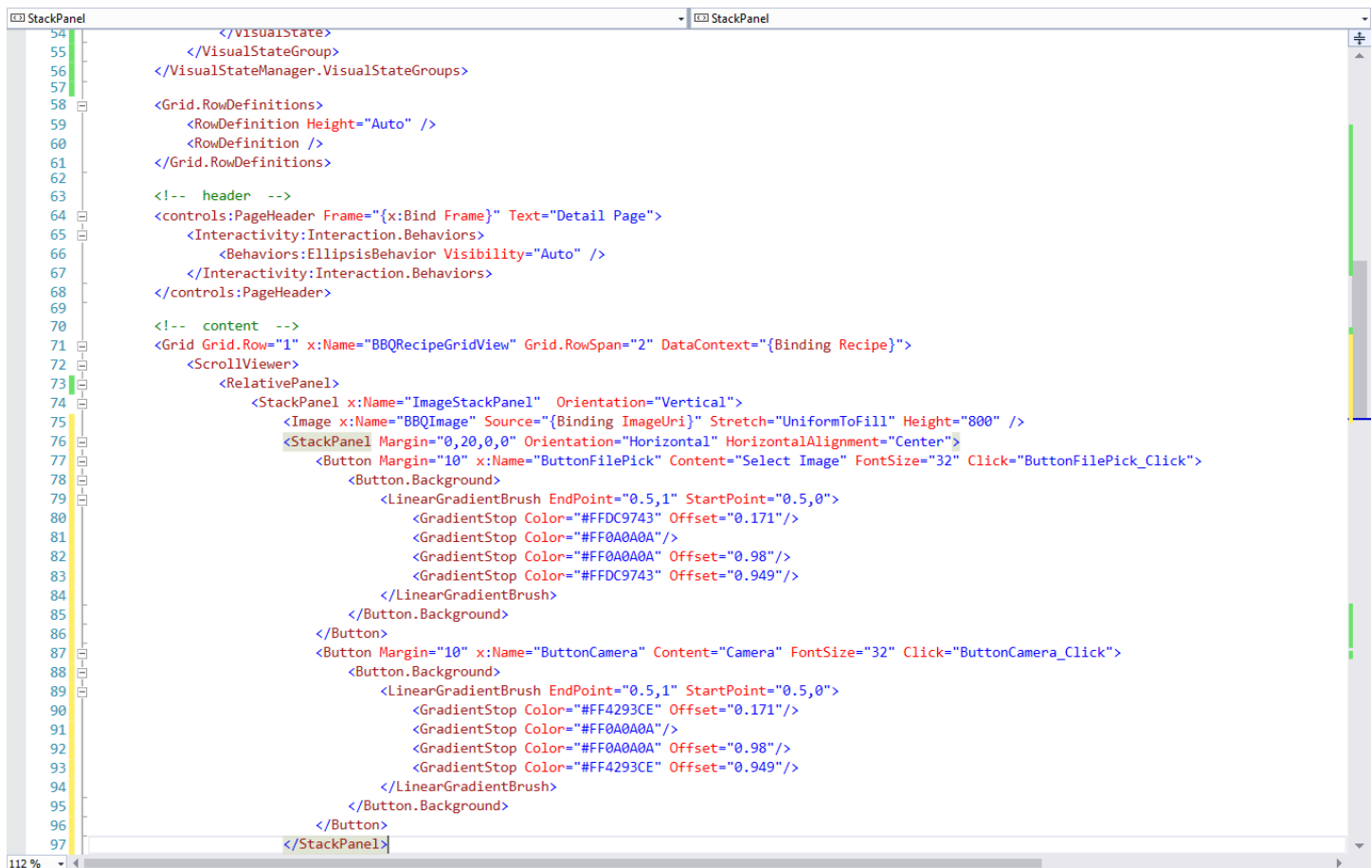
```

using Windows.UI.Xaml.Media.Imaging;
using Windows.Storage.FileProperties;
using Windows.Storage.Pickers;
using Windows.Media.Capture;
using Windows.Foundation;

```

- Open the \View\DetailPage.xaml file and add the following lines of Xaml code on line 76. We are adding 2 new buttons within the StackPanel labelled "ImageStackPanel".

```
<StackPanel Margin="0,20,0,0" Orientation="Horizontal" HorizontalAlignment="Center">
    <Button Margin="10" x:Name="ButtonFilePick" Content="Select Image" FontSize="32" Click="Button
FilePick_Click">
        <Button.Background>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop Color="#FFDC9743" Offset="0.171"/>
                <GradientStop Color="#FF0A0A0A"/>
                <GradientStop Color="#FF0A0A0A" Offset="0.98"/>
                <GradientStop Color="#FFDC9743" Offset="0.949"/>
            </LinearGradientBrush>
        </Button.Background>
    </Button>
    <Button Margin="10" x:Name="ButtonCamera" Content="Camera" FontSize="32" Click="ButtonCamera_C
lick">
        <Button.Background>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop Color="#FF4293CE" Offset="0.171"/>
                <GradientStop Color="#FF0A0A0A"/>
                <GradientStop Color="#FF0A0A0A" Offset="0.98"/>
                <GradientStop Color="#FF4293CE" Offset="0.949"/>
            </LinearGradientBrush>
        </Button.Background>
    </Button>
</StackPanel>
```



4. Open the \View\DetailPage.xaml,cs file and add the following lines of Xaml code on line 15 or after the page constructor.

```
private async void ButtonFilePick_Click(object sender, RoutedEventArgs e)
{
    await GetImage(false);
}

private async void ButtonCamera_Click(object sender, RoutedEventArgs e)
{
    await GetImage(true);
}

async Task GetImage(bool useCamera)
{
    try
    {
        var detailViewModel = this.DataContext as DetailPageViewModel;
        BBQImage.Source = await detailViewModel.SelectImage(useCamera);
        BBQImage.Visibility = (BBQImage.Source == null) ? Visibility.Visible : Visibility.Visible;
    }
    catch (Exception ex)
    {
        throw;
    }
}
```

5. Make sure to include the following using statements within the \View\DetailPage.xaml,cs file.

```
using Windows.UI.Xaml;
using System.Threading.Tasks;
using System;
```

6. Double click on the file “Package.appxmanifest” to open the App Manifest widow. Make sure to enable “Video Library”, “Picture Library”.

