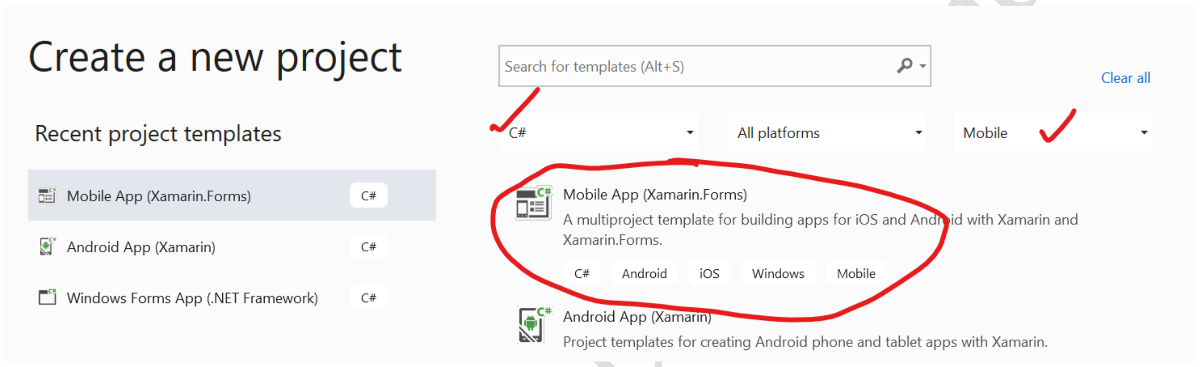
HANDS-ON EXERCISE FOR MID 3

OBJECTIVES:
- Learn how to read from text file
- Learn how to search any string from text file

DIRECTIONS:

Create a mobile app that can read and search for any string from a text file. The text file should be saved on the Assets folder of Android. This app is useful to provide quick backend of data as opposed to setting up a relational database.

1. Create a new project in Visual Studio 2019 and select Mobile App (Xamarin.Forms) on the template options. **Save it as Mid3_YOURFULLNAME**.



Create a new project

Recent project templates
- Mobile App (Xamarin.Forms) C#
- Android App (Xamarin) C#
- Windows Forms App (.NET Framework) C#

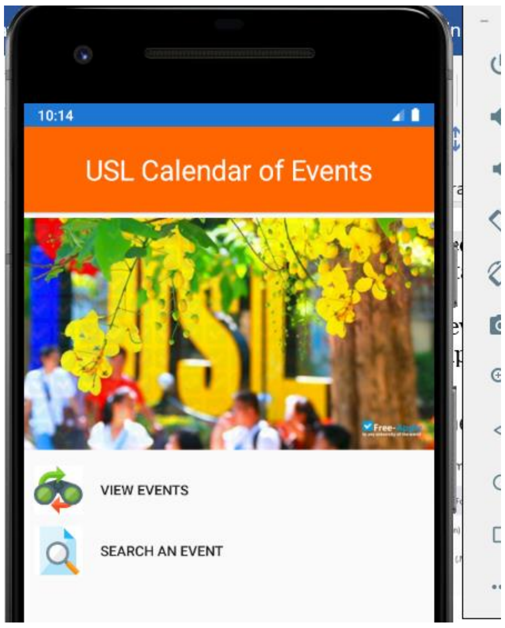
Search for templates (Alt+S) | Clear all

C# | All platforms | Mobile

Mobile App (Xamarin.Forms)
A multiproject template for building apps for iOS and Android with Xamarin and Xamarin.Forms.
C# Android iOS Windows Mobile

Android App (Xamarin)
Project templates for creating Android phone and tablet apps with Xamarin.

2. Design your MainPage.xaml to contain a header frame, any USL related background photo, and two buttons with photos for: (1.) viewing of events, and (2.) search an event.



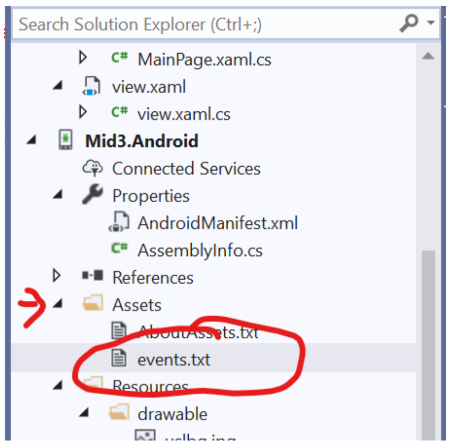
You may use the following icons for the view and search event:
- https://d1nhio0ox7pgb.cloudfront.net/_img/g_collection_png/standard/128x128/find_replace.png
- https://www.iconexperience.com/_img/g_collection_png/standard/128x128/document_view.png

Once the user will click on the VIEW EVENTS button, redirect the user to a page that you will create and name it as **view.xaml**. On the other hand, if the user will click on the SEARCH AN EVENT button, redirect the user to a page that you will create and name it as **search.xaml**.

On the screenshot above, I made use of Grid. You may adopt the sample codes of my **mainpage.xaml** as shown below.

```
<StackLayout>
    <Frame BackgroundColor=🟧"#ff6600" Padding="24" CornerRadius="0">
        <Label Text="USL Calendar of Events" HorizontalTextAlignment="Center" TextColor=□"White" FontSize="28"/>
    </Frame>
    <Image Source="uslbg.jpg" Aspect="AspectFit" />
    <Grid Margin="10">
        <Grid.RowDefinitions>
            <RowDefinition Height="auto" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="auto" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Image Source="https://d1nhio0ox7pgb.cloudfront.net/_img/g_collection_png/standard/128x128/find_replace.png"
               HorizontalOptions="Start" />
        <Button Grid.Column="1" Text="View Events" BackgroundColor=□"Transparent" HorizontalOptions="Start" Clicked="View_Clicked"/>
        <Image Grid.Row="2" Source="https://www.iconexperience.com/_img/g_collection_png/standard/128x128/document_view.png"
               HorizontalOptions="Start" />
        <Button Grid.Row="2" Grid.Column="1" Text="Search an Event" BackgroundColor=□"Transparent" HorizontalOptions="Start"
                Clicked="Search_Clicked" />
    </Grid>
</StackLayout>
```

3. You should download the file **events.txt** that has been attached on this problem. Save the file into your Android/Assets folder. Your mobile app will read and search text from this text file.



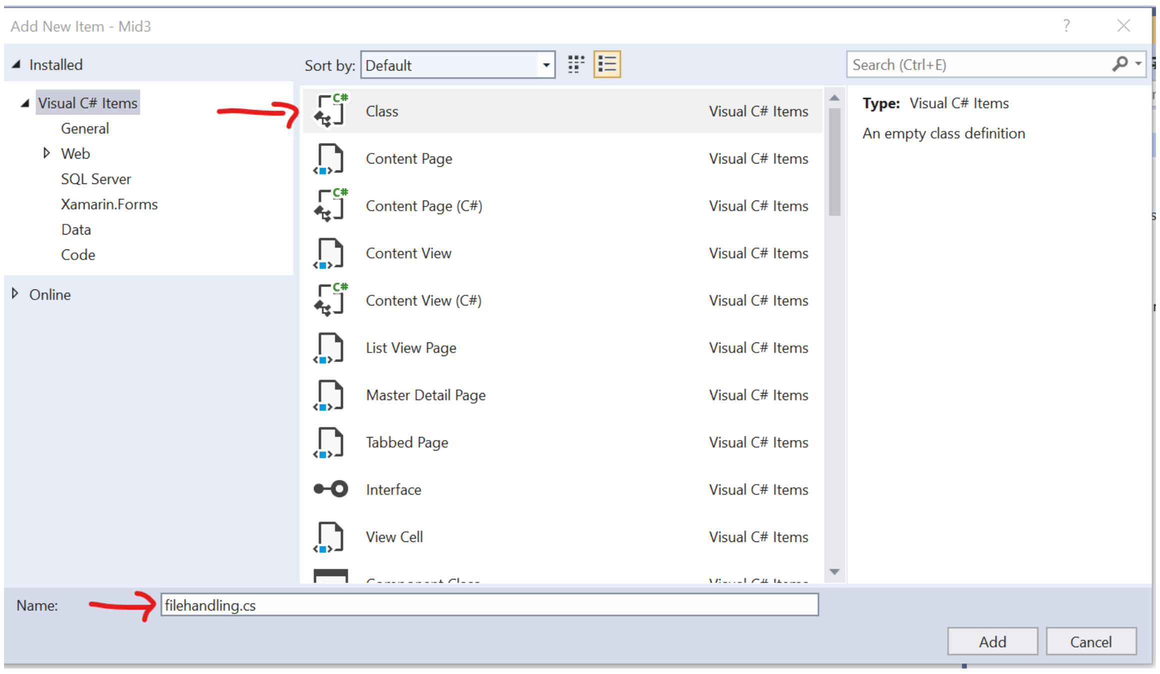
4. Create a new class on the root folder of your project (no need to create Model subfolder) and name it as **filehandling.cs**. This class is where we are going to reuse codes in reading a text file. Add the following libraries on this file.

```
using Android.Content.Res; //for AssetManager
using System.IO; //for StreamReader
```

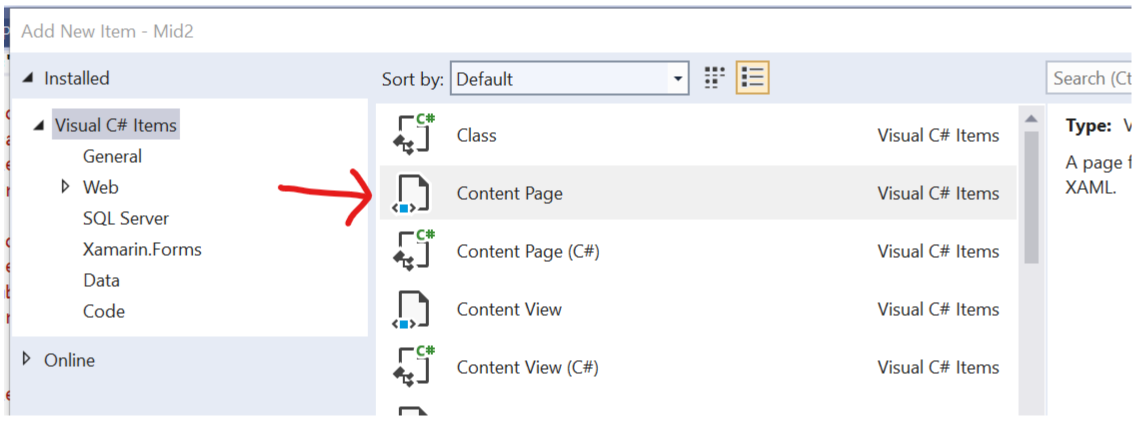
Then add the following methods inside the class.

```
public string readFile(string fname)
{
    string text = string.Empty;
    string filename = fname;

    AssetManager assets = Android.App.Application.Context.Assets;
    using (StreamReader reader = new StreamReader(assets.Open(filename)))
    {
        text = reader.ReadToEnd();
    }
    return text;
}
```


Add New Item - Mid3
Installed
Sort by: Default
Search (Ctrl+E)
Visual C# Items
General
Web
SQL Server
Xamarin.Forms
Data
Code
Online
Class Visual C# Items
Content Page Visual C# Items
Content Page (C#) Visual C# Items
Content View Visual C# Items
Content View (C#) Visual C# Items
List View Page Visual C# Items
Master Detail Page Visual C# Items
Tabbed Page Visual C# Items
Interface Visual C# Items
View Cell Visual C# Items
Type: Visual C# Items
An empty class definition
Name: filehandling.cs
Add
Cancel

5. Create a content page and save it as **view.xaml**. This page should contain a header frame, USL related background, and a scroll view to display all the contents of the text file. Also, there should be a HOME button found at the bottom of this plage.

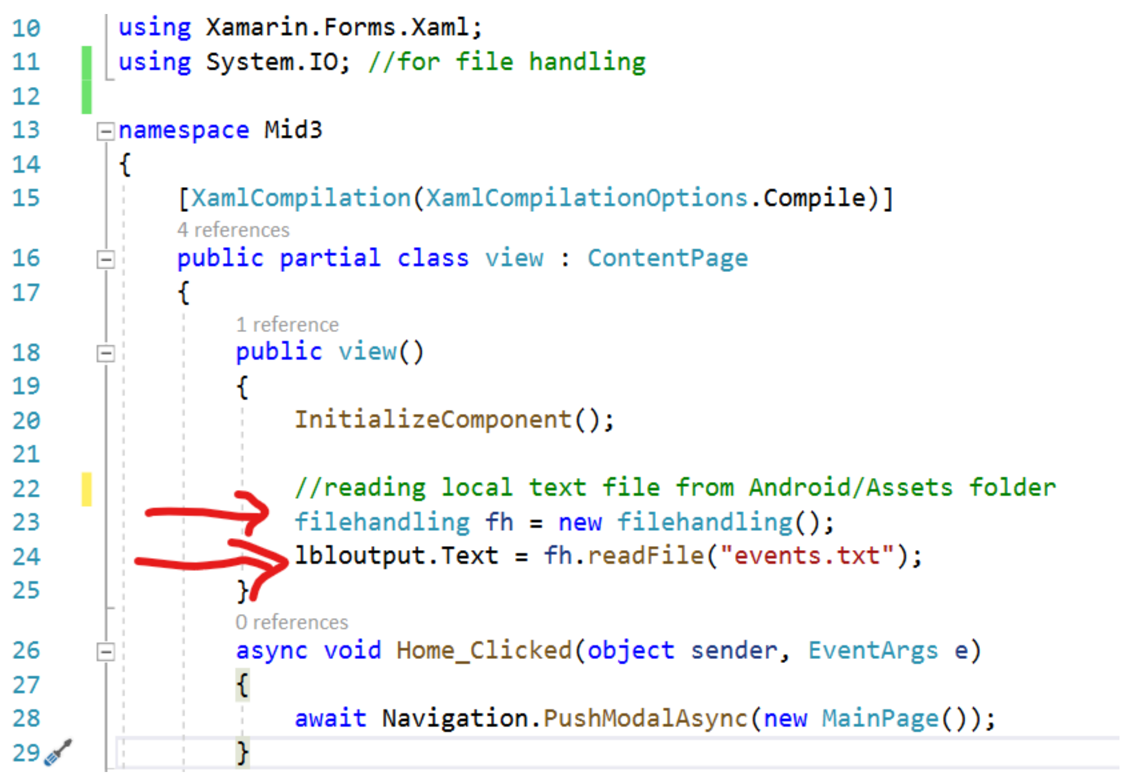


You may refer to the sample codes below. The scroll view should be scrollable both horizontally and vertically. This is to ensure that contents of the text file are viewable on the app regardless of the volume of its contents.

Contents of the text file are going to be dumped on the label named as **lbloutput**.

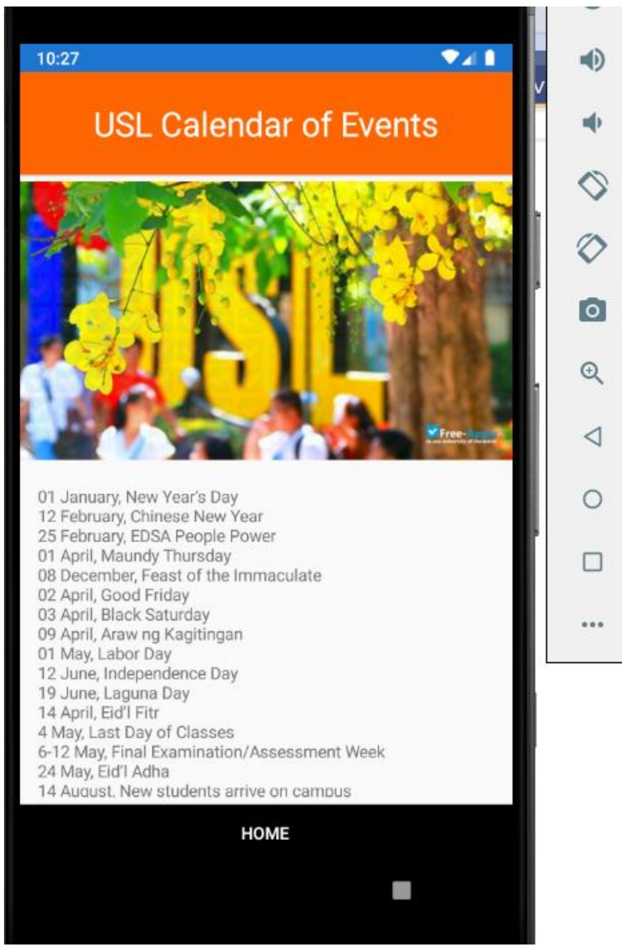
```
<StackLayout>
    <Frame BackgroundColor="#ff6600" Padding="24" CornerRadius="0">
        <Label Text="USL Calendar of Events" HorizontalTextAlignment="Center" TextColor="White" FontSize="28"/>
    </Frame>
    <Image Source="uslbg.jpg" Aspect="AspectFit" />
    <ScrollView VerticalOptions="FillAndExpand" Orientation="Both">
        <Label x:Name="lbloutput" Margin="15"/>
    </ScrollView>
    <StackLayout BackgroundColor="Black">
       <Button Clicked ="Home_Clicked" HorizontalOptions="CenterAndExpand" VerticalOptions="EndAndExpand"
            Text="HOME" TextColor="White" BackgroundColor="Black"/>
    </StackLayout>
</StackLayout>
```

In your **view.xaml.cs** file, add the following highlighted codes. Line 23 creates an object reference to the **filehandling.cs** class we created earlier. It contains a method **readFile** that receives a filename of the text file and returns the contents of the file in string format as shown in Line 24.



```
10  using Xamarin.Forms.Xaml;
11  using System.IO; //for file handling
12
13  namespace Mid3
14  {
15      [XamlCompilation(XamlCompilationOptions.Compile)]
        4 references
16      public partial class view : ContentPage
17      {
            1 reference
18          public view()
19          {
20              InitializeComponent();
21
22              //reading local text file from Android/Assets folder
23              filehandling fh = new filehandling();
24              lbloutput.Text = fh.readFile("events.txt");
25          }
            0 references
26          async void Home_Clicked(object sender, EventArgs e)
27          {
28              await Navigation.PushModalAsync(new MainPage());
29          }
```

Try to run your application. Click on VIEW EVENTS, and it should redirect you to your **view.xaml** page containing the data inside your text file.



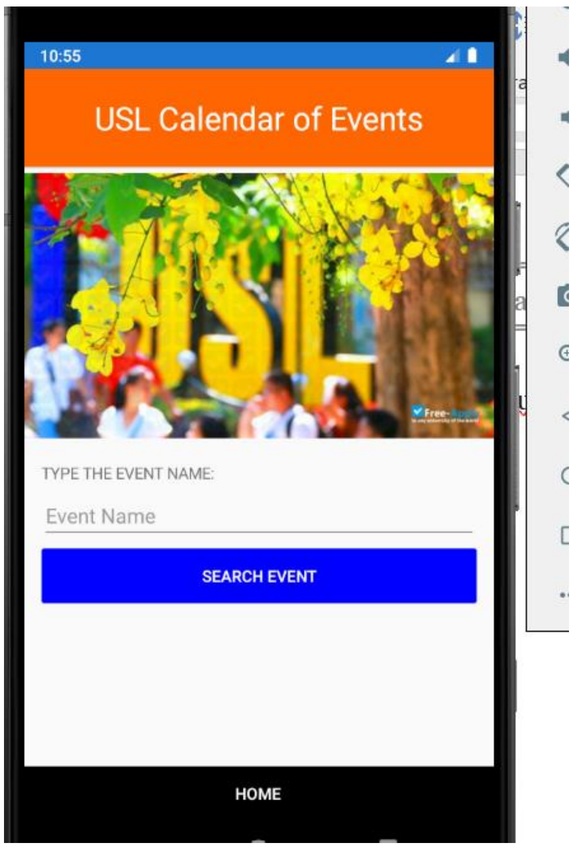
10:27

USL Calendar of Events

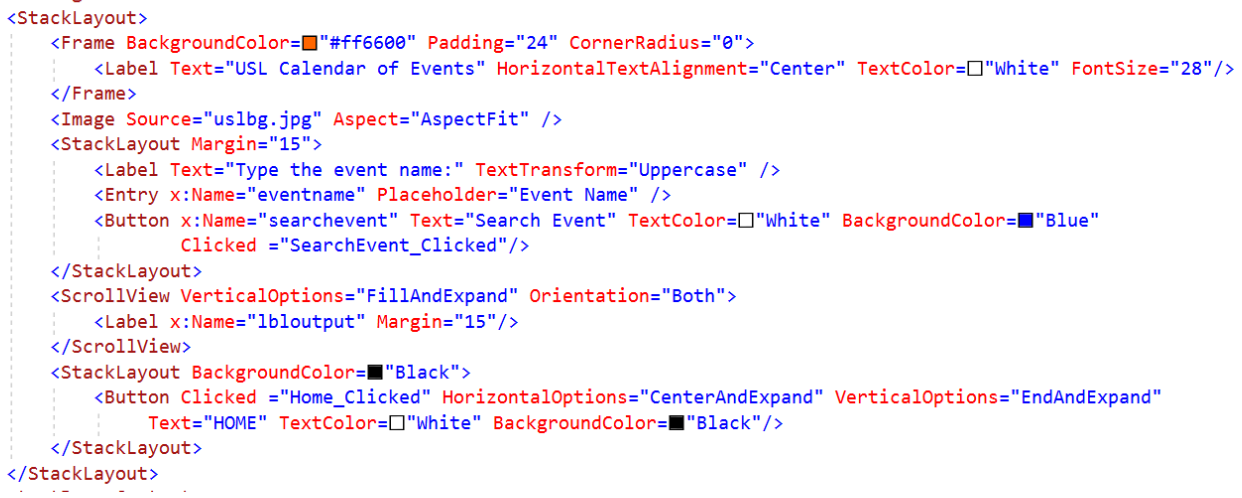
```
01 January, New Year's Day
12 February, Chinese New Year
25 February, EDSA People Power
01 April, Maundy Thursday
08 December, Feast of the Immaculate
02 April, Good Friday
03 April, Black Saturday
09 April, Araw ng Kagitingan
01 May, Labor Day
12 June, Independence Day
19 June, Laguna Day
14 April, Eid'l Fitr
4 May, Last Day of Classes
6-12 May, Final Examination/Assessment Week
24 May, Eid'l Adha
14 August, New students arrive on campus
```

HOME

6. Create your **search.xaml** page, and it should contain the same frame header, USL related photo, a label, an entry to receive the name of the event that needs to be searched, and a button to display results. Similarly, a home button should be found at the bottom of this page.

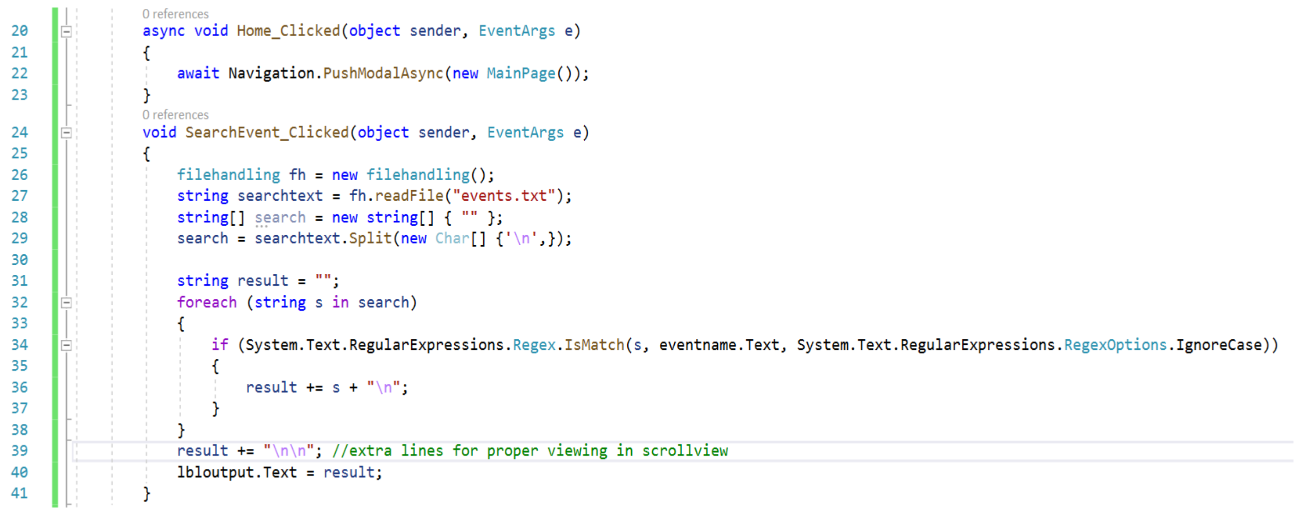


You may adopt the same code below.



```
<StackLayout>
    <Frame BackgroundColor=🟧"#ff6600" Padding="24" CornerRadius="0">
        <Label Text="USL Calendar of Events" HorizontalTextAlignment="Center" TextColor=□"White" FontSize="28"/>
    </Frame>
    <Image Source="uslbg.jpg" Aspect="AspectFit" />
    <StackLayout Margin="15">
        <Label Text="Type the event name:" TextTransform="Uppercase" />
        <Entry x:Name="eventname" Placeholder="Event Name" />
        <Button x:Name="searchevent" Text="Search Event" TextColor=□"White" BackgroundColor=🟦"Blue"
                Clicked ="SearchEvent_Clicked"/>
    </StackLayout>
    <ScrollView VerticalOptions="FillAndExpand" Orientation="Both">
        <Label x:Name="lbloutput" Margin="15"/>
    </ScrollView>
    <StackLayout BackgroundColor=■"Black">
        <Button Clicked ="Home_Clicked" HorizontalOptions="CenterAndExpand" VerticalOptions="EndAndExpand"
             Text="HOME" TextColor=□"White" BackgroundColor=■"Black"/>
    </StackLayout>
</StackLayout>
```

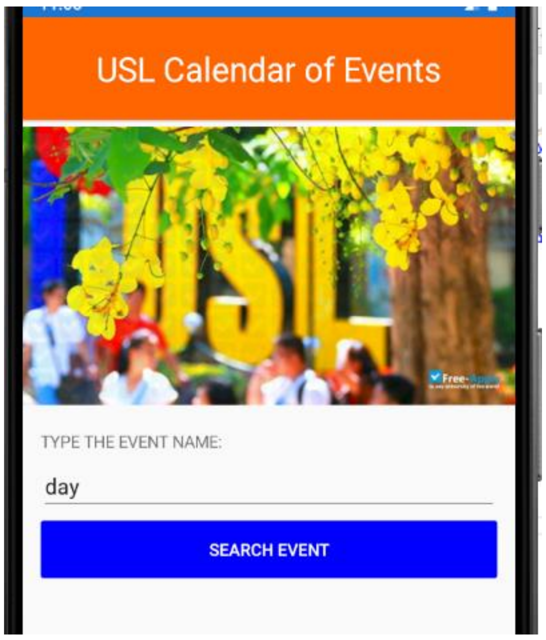
7. Open your **search.xaml.cs** and add the following codes. When the user search for an event, the text is compared line-by-line with the content of the text file as shown in Lines 32-38 using a regular expression class.

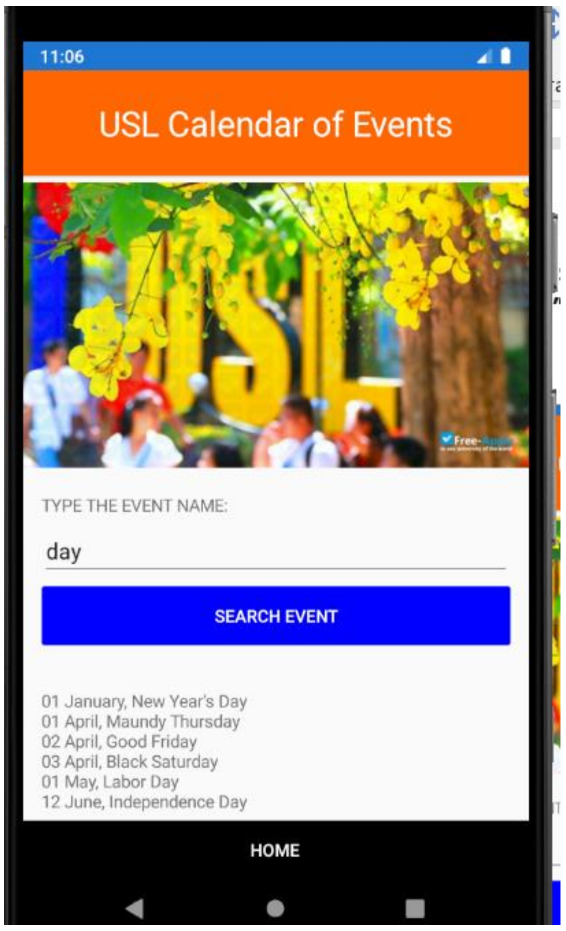


```
        0 references
        async void Home_Clicked(object sender, EventArgs e)
        {
            await Navigation.PushModalAsync(new MainPage());
        }
        0 references
        void SearchEvent_Clicked(object sender, EventArgs e)
        {
            filehandling fh = new filehandling();
            string searchtext = fh.readFile("events.txt");
            string[] search = new string[] { "" };
            search = searchtext.Split(new Char[] {'\n',});

            string result = "";
            foreach (string s in search)
            {
                if (System.Text.RegularExpressions.Regex.IsMatch(s, eventname.Text, System.Text.RegularExpressions.RegexOptions.IgnoreCase))
                {
                    result += s + "\n";
                }
            }
            result += "\n\n"; //extra lines for proper viewing in scrollview
            lbloutput.Text = result;
        }
```

Only those text that appear on each line of the text files are going to be displayed as search output. For instance, if I search for any appearance of the word **"day"** on the text file, then the output should be filtered only to these criteria.





```
01 January, New Year's Day
01 April, Maundy Thursday
02 April, Good Friday
03 April, Black Saturday
01 May, Labor Day
12 June, Independence Day
```