Xamarin – Cross Platform Development

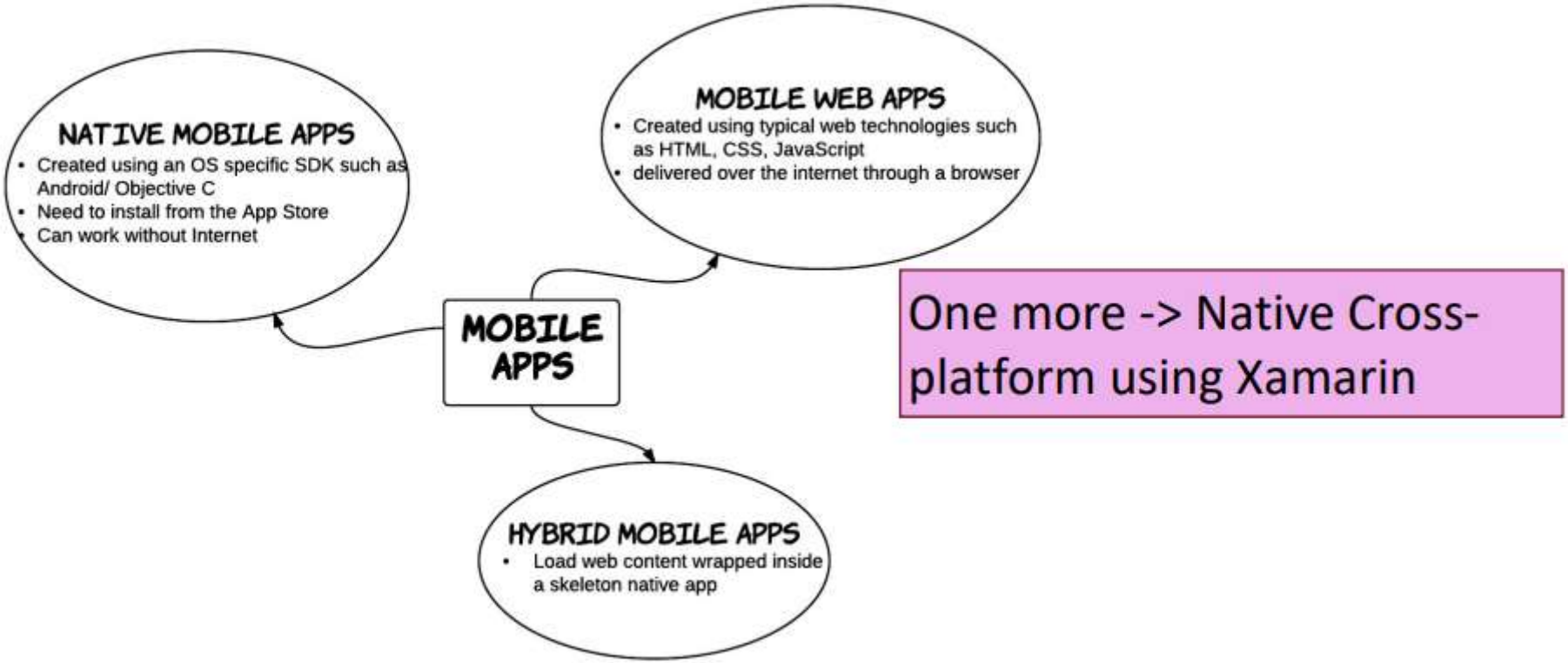
CSE2MAD LECTURE 8

Introduction to cross-platform mobile app development

Outline

- Types of Apps – recap
- Native Cross-Platform using Xamarin
- Why Native Cross-Platform?
- How Does Xamarin Work?
- Xamarin - Platform SDK
- Xamarin Available IDEs
- Application Output
- Xamarin Emulators/ Simulators
- Xamarin Forms
- Other Cross Platform Development Technolgies

Recap: Types of mobile apps

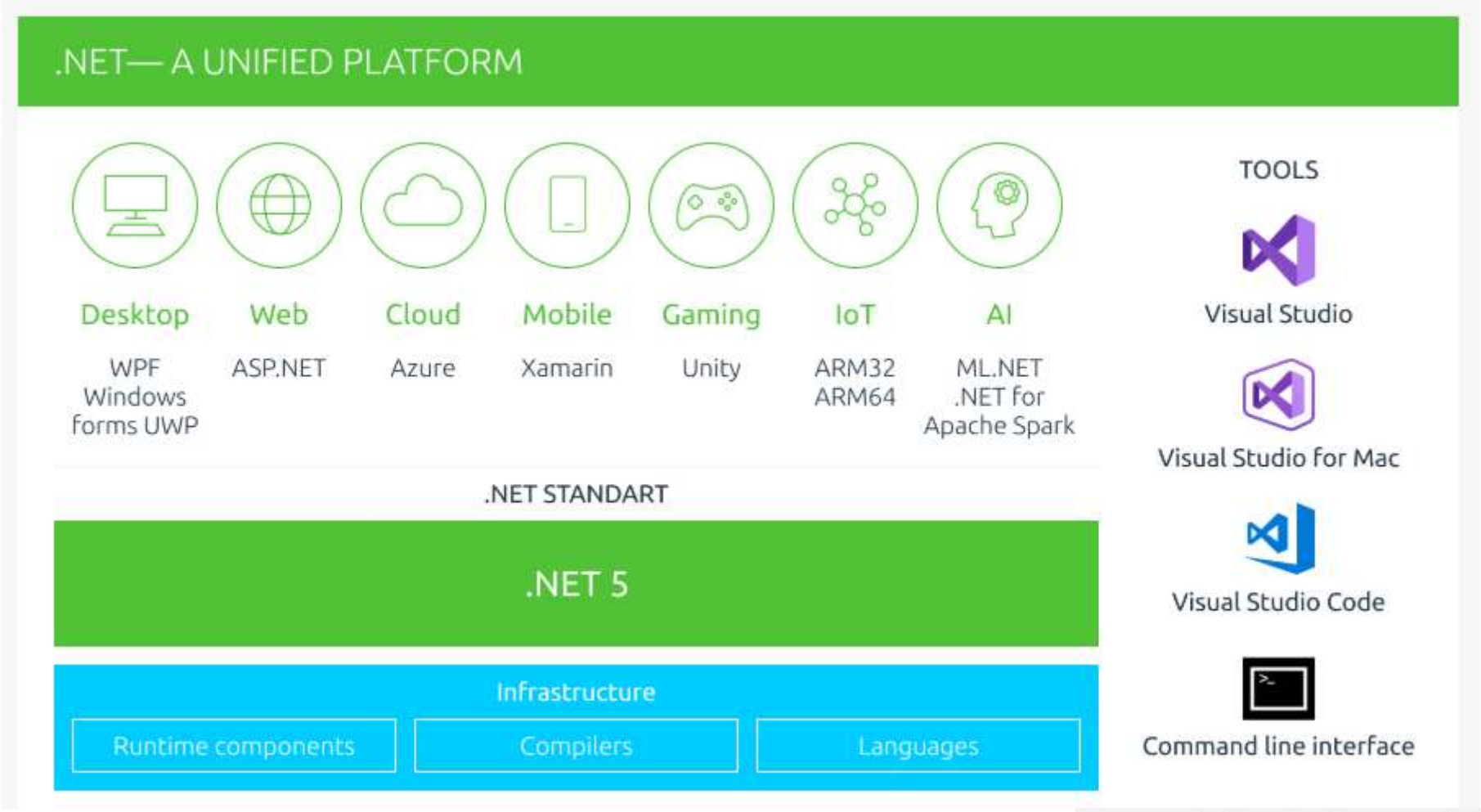


.NET

- What is .NET? https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet
  - Platform consisting of languages, libraries and tools for developing many application types of many devices and operating systems
  - C# is the most popular programming language
    - Need help → see https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/
    - Also use C++, F#, Visual Basic
  - Contains a common set of libraries to implement common functionality across the languages it supports (.NET Core)
  - .NET Framework → Windows based support
  - **Xamarin/Mono → For cross platform mobile app support**
    - **What is Mono?**
  - Excellent tool support: Visual Studio Windows/Mac/Linux, Docker

For More Information : https://dotnet.microsoft.com/learn/xamarin/what-is-xamarin

.NET Direction



Data source: devblogs.microsoft.com—Introducing .NET 5, 2019

Native cross-platform using Xamarin

- Native Cross-Platform (Android/iOS/Windows Phone) Mobile Apps
- Programs in C# for iOS, Android, tvOS, watchOS, macOS and Windows while still compiling native apps
- IDE: Visual Studio (windows & mac)

Why Native cross-platform?

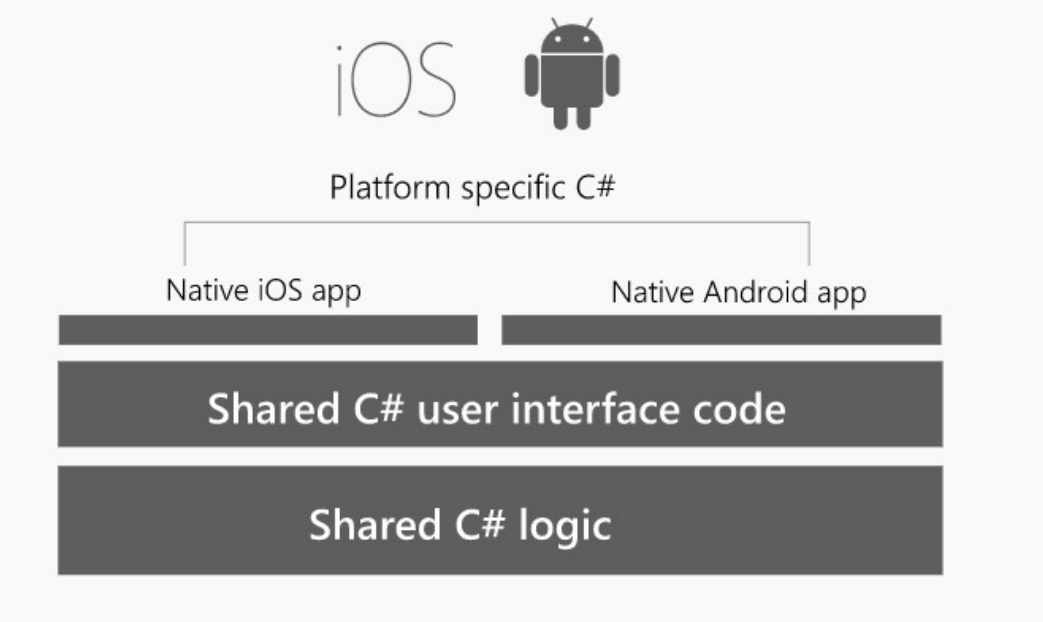
- Use one language, C# to develop for iOS and Android
- Code application logic once and then share it across both iOS and Android
- Support unique features of native platforms

▶…Xamarin also has drawbacks

- Problems with Xamarin ecosystem: small-medium community
- Delayed support for platform updates
- Currently a bit unstable, but this will improve…

Exposure to Xamarin will be an advantage in the job market!

How Does Xamarin Work?



https://visualstudio.microsoft.com/xamarin/

Xamarin Available IDEs

WE NOW HAVE Visual studio for MAC and it's replacing Xamarin Studio

		Develop on Mac OS X	Develop on Windows	
	Xamarin IDE	**Xamarin Studio**	**Xamarin Studio**	**Visual Studio**
Mobile Platform	iOS	YES	NO	YES
	Android	YES	YES	YES
	Windows	NO	NO	YES

Xamarin App Properties

- **Native UI look and feel (Important to maintain user expectations)**
- **Native API Access (full features of the target platform)**
- **Native Performance (critical for the user experience)**

Application Output

- Android -> .apk file
- iOS -> .app file

- Same as native & deployed the same.

Xamarin Emulators/ Simulators

- Android testing -> on an emulator
- iOS testing -> on a simulator

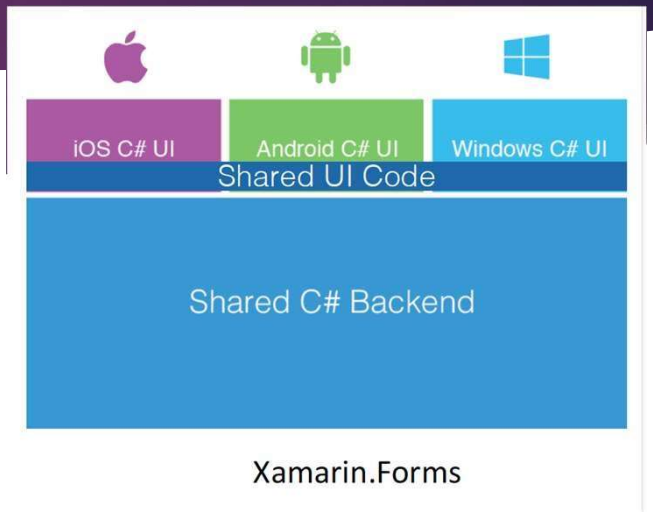
- **Emulator vs Simulator?**

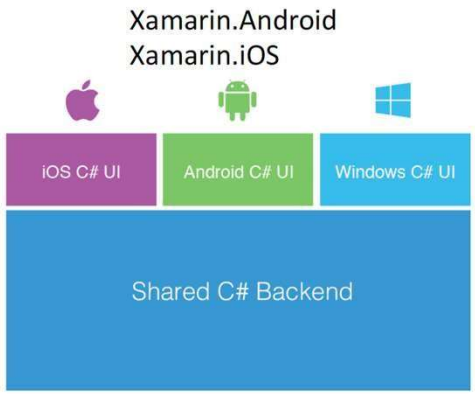
- Emulators mimic the software & hardware found on an actual device.
  - Eg: if the device being emulated has 2 GM RAM, the emulator will also only have 2 GM RAM
- Simulators mimic the software found on an actual device.
  - Eg: even if the device being emulated has 2 GM RAM in reality, simulator will have access to full resources of the computer its running on.

Xamarin Forms

- A way to write UI once and get it compiled to native UI on iOS, Android, and Windows Phone.
- Doesn't replace Xamarin.iOS and Xamarin.Android; rather, integrates with them.
- **Motivation:** Case where platform coverage is crucial but it's not possible to maintain several native applications

Native vs Xamarin.Native vs Xamarin.Forms





Xamarin Forms vs Xamarin Native

- **Use Xamarin forms for:**
- Data entry apps
- Prototypes and proofs-of-concept
- Apps that require little platform-specific functionality
- Apps where code sharing is more important than custom UI
- Apps must look the same

Use Xamarin.iOS & Xamarin.Android for:

- Apps that require specialized interactions
- Apps with highly polished design
- Apps that use many platform-specific APIs

Inside a Xamarin.Forms Application

Each screen corresponds to a ***Page***

A Page is ..
In Android -> an *Activity*

In iOS -> a *View Controller*

In the Windows Universal Platform (UWP)-> *Page*

Let's try some code

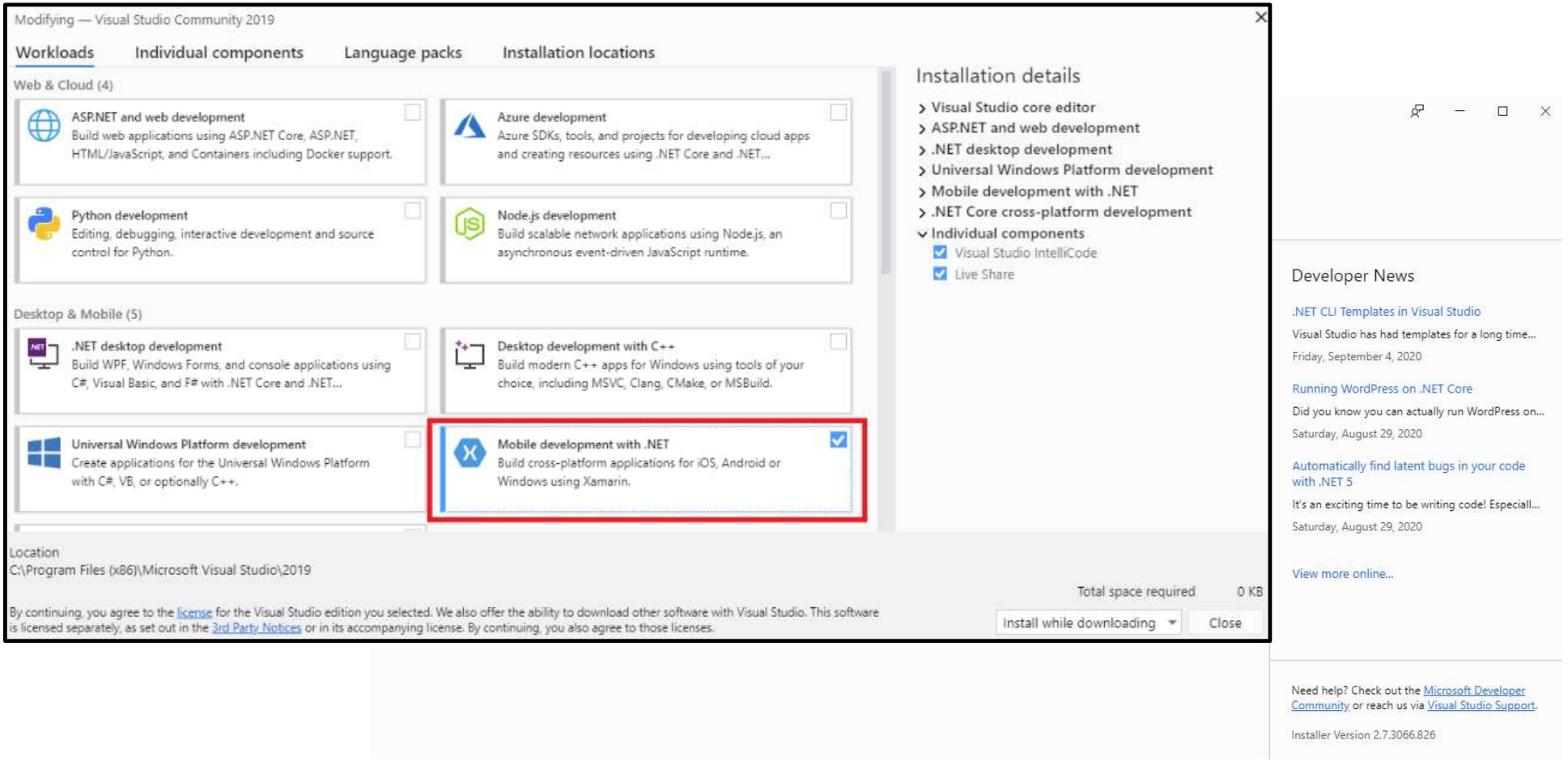
You can try alongside, let's explore an Xamarin Forms app

Code: https://docs.microsoft.com/en-us/xamarin/get-started/quickstarts/single-page

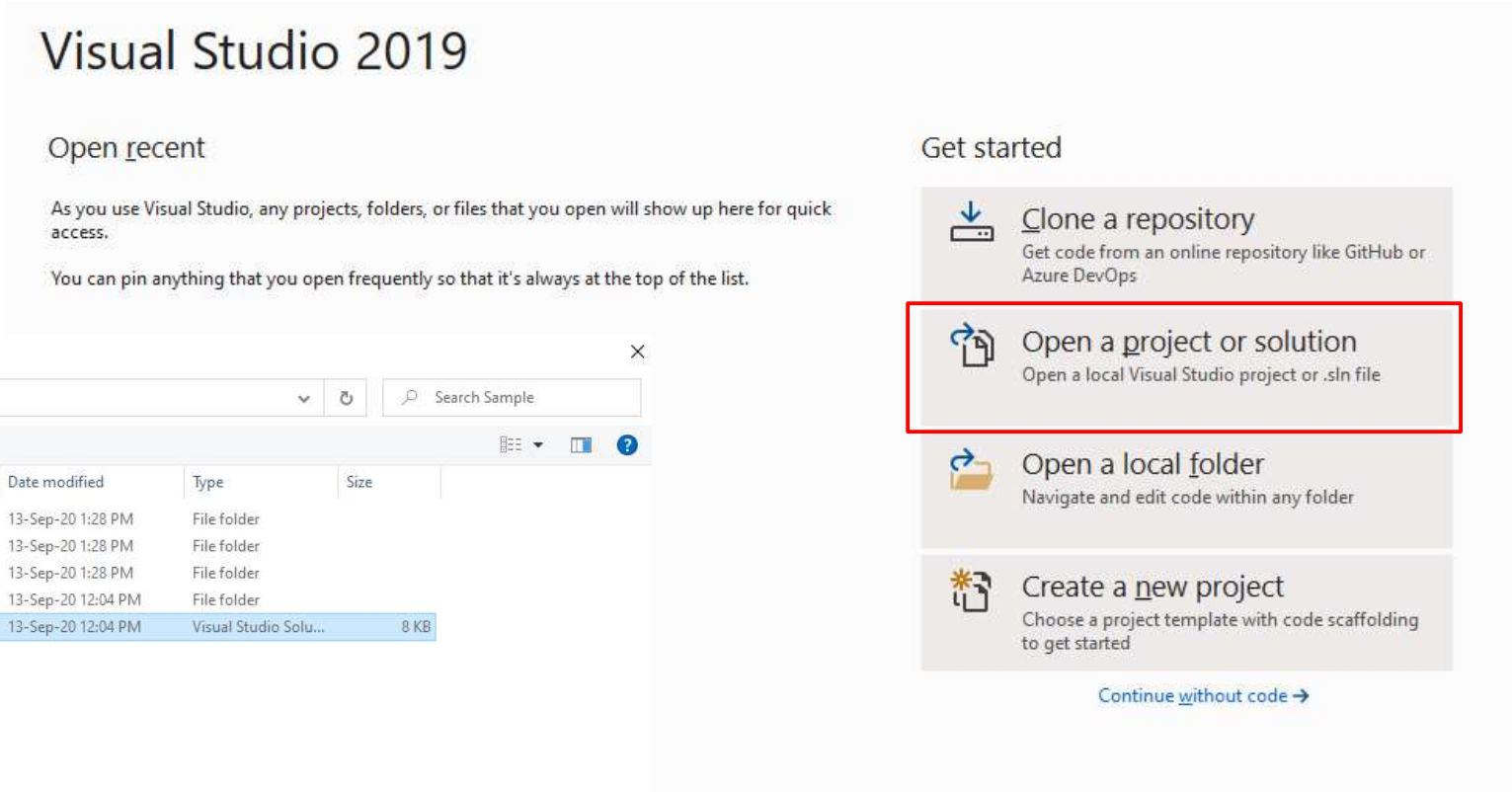
Pre-Req's (free)

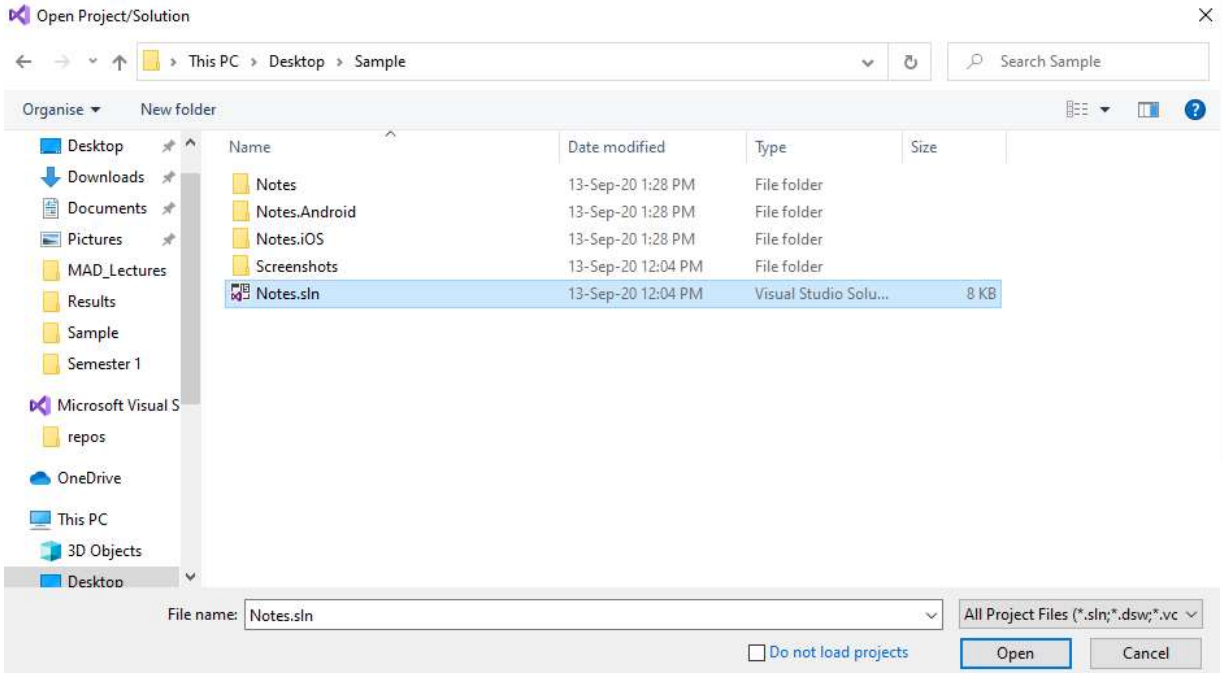
Visual Studio Community Edition

Step 1 – Download and Install VS 2019 Community Edition



Step 2 – Open the solution file and set up the dev. environment



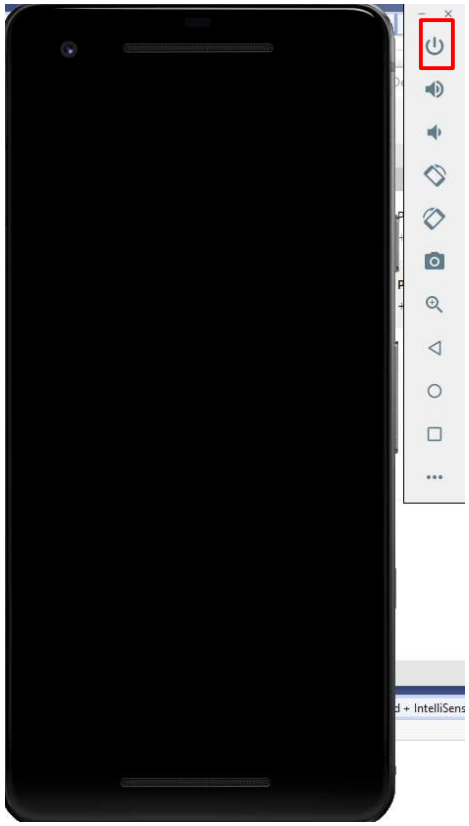


Step 2 – Open the solution file and set up the dev. environment

- Set up the Android Emulator

Step 3 – Start the emulator

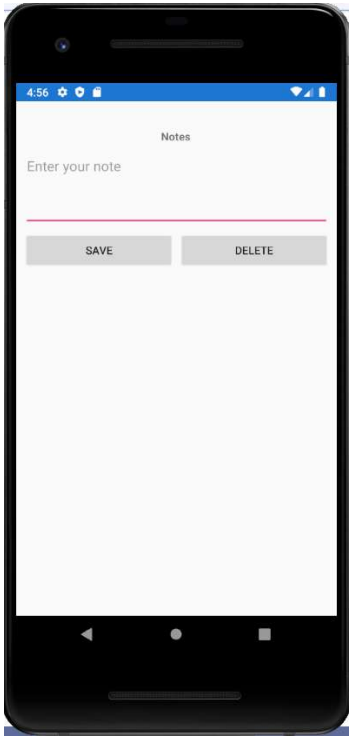
- Turn it on



We'll now see the emulator in VS

Run the app

Output

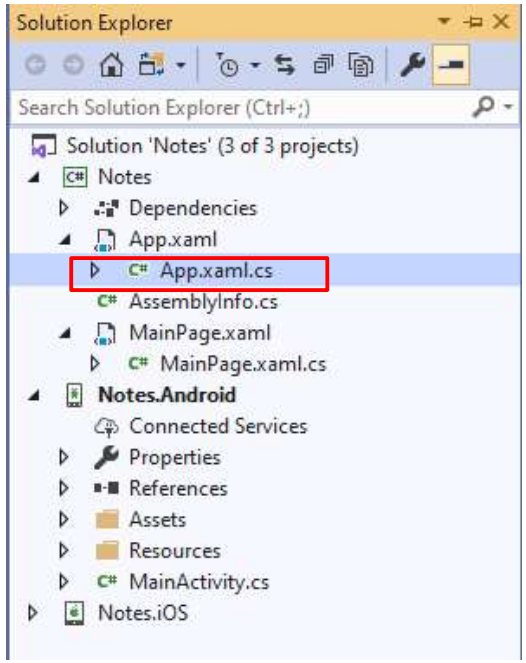


Let’s look at the code

Inside a Xamarin.Forms Application

```
using Xamarin.Forms;

namespace Notes
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();

            MainPage = new MainPage();
        }

        protected override void OnStart()
        {
            // Handle when your app starts
        }

        protected override void OnSleep()
        {
            // Handle when your app sleeps
        }

        protected override void OnResume()
        {
            // Handle when your app resumes
        }
    }
}
```

This code defines the starting class for the application and it instantiates a new MainPage



Inside a Xamarin.Forms Application

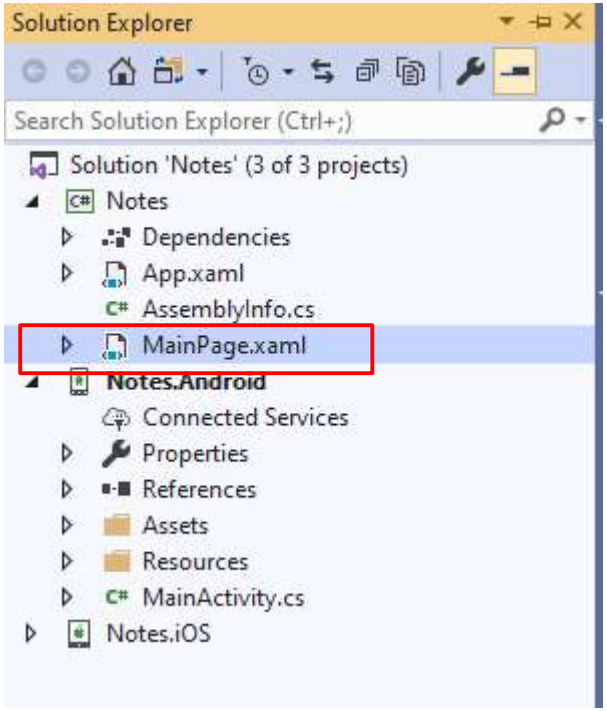
- Xamarin.Forms applications have a single class App.
- App class has MainPage– in the constructor. This where you set your main page
- ContentPage class has a Content Property
- Also, we have OnStart, OnSleep, OnResume, etc.

App.xaml

```
<?xml version="1.0" encoding="utf-8"?>
<Application
xmlns="http://xamarin.com/schemas/2014/forms"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xam
l" x:Class="Notes.App">
    <Application.Resources>
    </Application.Resources>
</Application>
```

MainPage.xaml

```
<?xml version="1.0" encoding="utf-8"?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Notes.MainPage">
    <StackLayout Margin="10,35,10,10">
        <Label Text="Notes"
               HorizontalOptions="Center"
               FontAttributes="Bold" />
        <Editor x:Name="_editor"
                Placeholder="Enter your note"
                HeightRequest="100" />
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*" />
                <ColumnDefinition Width="*" />
            </Grid.ColumnDefinitions>
            <Button Text="Save"
                    Clicked="OnSaveButtonClicked" />
            <Button Grid.Column="1"
                    Text="Delete"
                    Clicked="OnDeleteButtonClicked"/>
        </Grid>
    </StackLayout>
</ContentPage>
```



How is this similar to Android XML?

eXtensible Application Markup Language (XAML)

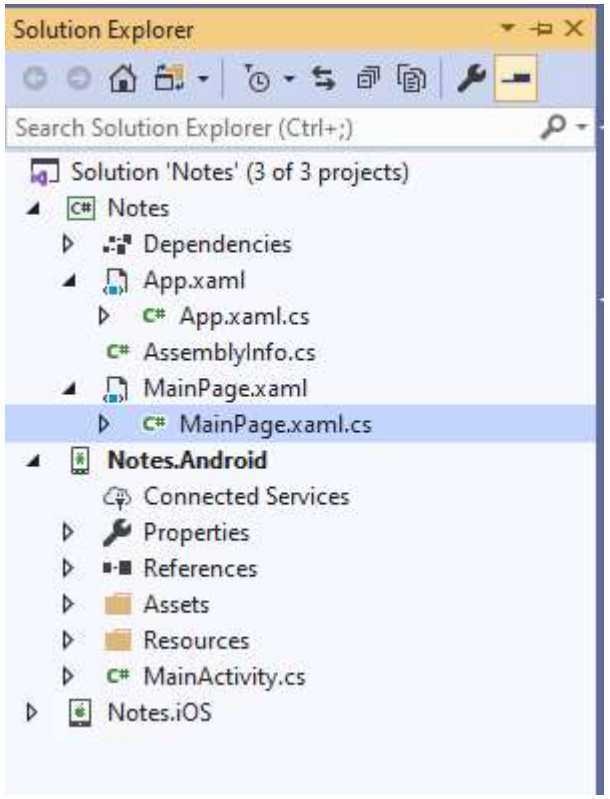
- **Used to define Xamarin.Forms user interfaces**
- XAML is basically XML, but XAML has some unique syntax features.
- The most important are:
  - Property elements
  - Attached properties
  - Markup extensions

.XAML used for native layouts for Android

All XAML documents are also valid XML documents, but not vice-versa.

MainPage.xaml.cs

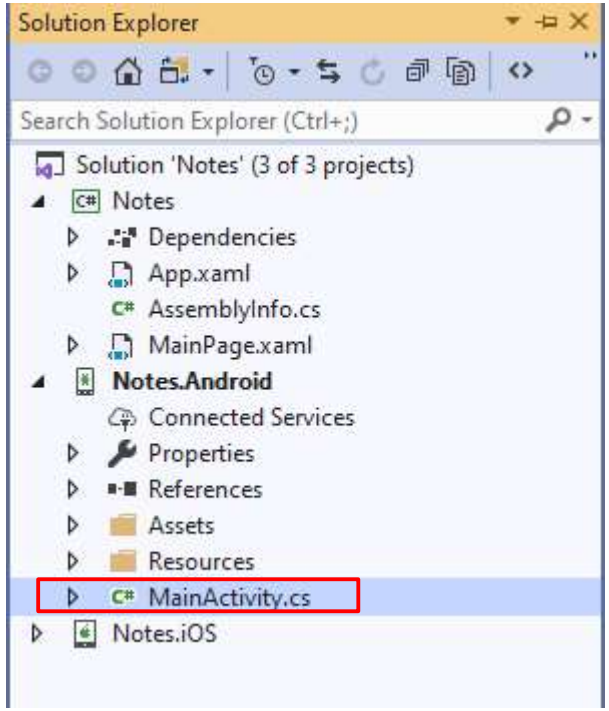
```
using System;
using System.IO;
using Xamarin.Forms;

namespace Notes
{
    public partial class MainPage : ContentPage
    {
        string _fileName =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplication
Data), "notes.txt");

        public MainPage()
        {
            InitializeComponent();

            if (File.Exists(_fileName))
            {
                _editor.Text = File.ReadAllText(_fileName);
            }
        }

        void OnSaveButtonClicked(object sender, EventArgs e)
        {
            File.WriteAllText(_fileName, _editor.Text);
        }

        void OnDeleteButtonClicked(object sender, EventArgs e)
        {
            if (File.Exists(_fileName))
            {
                File.Delete(_fileName);
            }
            _editor.Text = string.Empty;
        }
    }
}
```



Can you see similarities to Android?

MainActivity.cs

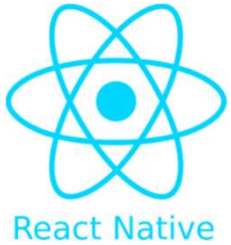
```
using System;

using Android.App;
using Android.Content.PM;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.OS;

namespace Notes.Droid
{
    [Activity(Label = "Notes", Icon = "@mipmap/icon", Theme = "@style/MainTheme",
MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize |
ConfigChanges.Orientation)]
    public class MainActivity :
global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
    {
        protected override void OnCreate(Bundle savedInstanceState)
        {
            TabLayoutResource = Resource.Layout.Tabbar;
            ToolbarResource = Resource.Layout.Toolbar;

            base.OnCreate(savedInstanceState);
            global::Xamarin.Forms.Forms.Init(this, savedInstanceState);
            LoadApplication(new App());
        }
    }
}
```



How is this similar to Native Android?

Multi-Platform alternatives

- React Native



Written in JavaScript—rendered with native code

- Ionic

Open source SDK
uses web frameworks to build apps on top of Cordova

- Cordova

Mobile apps with HTML, CSS & JS
Free and open source

- Flutter

Written in Dart, open-source SDK & Framework