# Creating Dialog Fragment Using Xamarin Android

# Introduction

Xamarin is a development platform that allows us to code native, cross-platform iOS, Android, and Windows Phone apps in C#.

# UI Designers

The tools we use to create mobile user interfaces are called designers. These generate Extensible Markup Language (XML) files in their respective proprietary file formats. Two designers are available from Xamarin - "Xamarin Designer for Android" and "Xamarin Designer for iOS". With the availability of these designers, the need for the original, native XML editors has diminished. Anything you might need to build Android or iOS UIs can be found in Xamarin’s tools. However, iOS developers still frequently use the Xcode Interface Builder, and Android developers (less frequently) use XML editors, such as the Android Development Tools (ADT) plug-in for Eclipse. An XML layout is a layout provided by XML , and the tool is largely a matter of taste and personal preference, even the decision to use a designer tool at all.

Some Xamarin developers are opting to code UIs by hand in C# for all platforms with no designer used whatsoever. I recommend the designers to help learn the file formats, UI elements, and their properties. At the very least, use a designer tool like training wheels until you’re ready to ride freestyle.

# Requirements

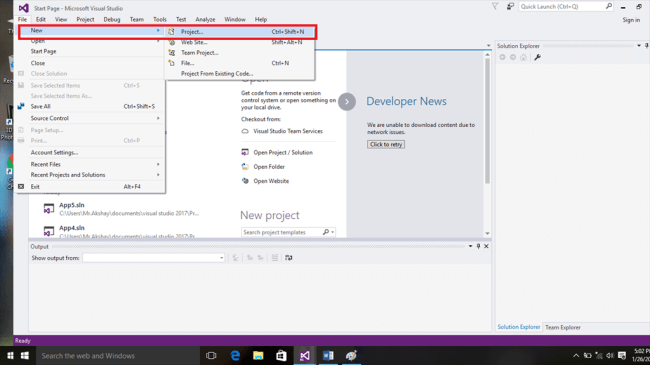
* Visual Studio RC 2017
* Xamarin Studio Packages should be installed
* Self confidence with creativity

# Steps should be followed

Please follow my steps to create Dialog Fragment and use this tutorial effectively.

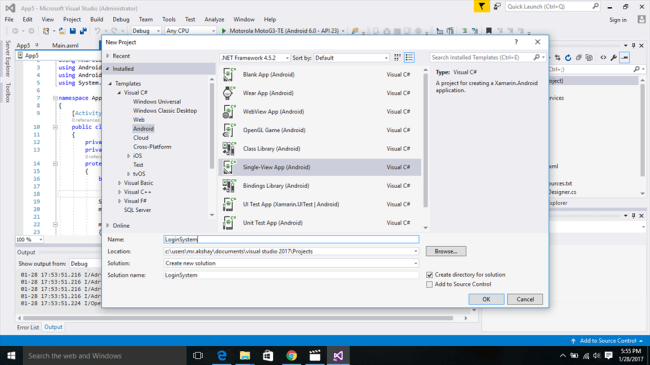
# Step 1

Launch the Visual Studio RC 2017 and go to File >> New >> Project.



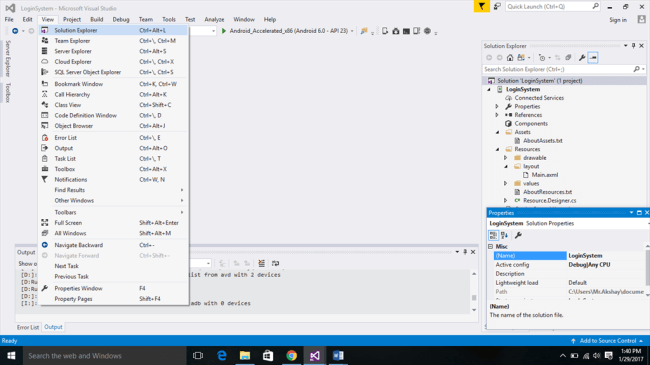
# Step 2

Then, we need to select the Android templatefrom the list, name the application, create the Project, and select where to save the project.



# Step 3

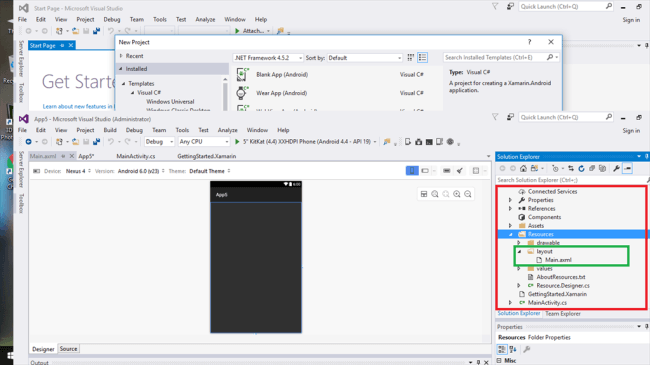
Here we go. Our new project has been created successfully.  Now, we need to navigate to Designer Page. To open the designer page, you need to open *Solution Explorer View >> Solution Explorer*.



# Step 4

Then, to open "Designer" page, we need to select the following,

Solution Explorer >> Resources >> Layout >>Main.Xaml.



# Step 5

Add the following code in Main.Xaml source page. This is our designer page and here, our Login System UI design will be displayed. We need to add the Message Icon in left side of our button. So, download any message icon.png from the internet and add that to drawable folder in Solution Explorer.

Go to properties of "Sign Up" Button and search for and choose the icon from "drawable" folder.

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3. android:orientation="vertical"
4. android:layout\_width="match\_parent"
5. android:layout\_height="match\_parent"
6. android:background="@android:color/background\_light">
7. <TextView
8. android:text="Create An Account"
9. android:textAppearance="?android:attr/textAppearanceLarge"
10. android:layout\_width="match\_parent"
11. android:layout\_height="0dp"
12. android:id="@+id/textView1"
13. android:layout\_weight="1"
14. android:background="@android:color/darker\_gray"
15. android:textStyle="bold"
16. android:textSize="25sp"
17. android:gravity="center" />
18. <Button
19. android:text="Sign IN"
20. android:layout\_width="match\_parent"
21. android:layout\_height="0dp"
22. android:id="@+id/button1"
23. android:layout\_weight="1"
24. android:textSize="25sp"
25. android:background="@android:color/holo\_blue\_light" />
26. <TextView
27. android:text="OR"
28. android:textAppearance="?android:attr/textAppearanceLarge"
29. android:layout\_width="match\_parent"
30. android:layout\_height="0dp"
31. android:id="@+id/textView2"
32. android:layout\_weight="1"
33. android:background="@android:color/darker\_gray"
34. android:textSize="25sp"
35. android:textColor="@android:color/background\_light"
36. android:gravity="center" />
37. <Button
38. android:text="Sign Up"
39. android:layout\_width="match\_parent"
40. android:layout\_height="0dp"
41. android:id="@+id/sinup"
42. android:layout\_weight="1"
43. android:textSize="30sp"
44. android:background="@drawable/buttonsignupstyle"
45. android:drawableLeft="@drawable/images" />
46. <ProgressBar
47. android:layout\_width="match\_parent"
48. android:layout\_height="wrap\_content"
49. android:id="@+id/progressBar1"
50. android:layout\_weight="1" />
51. </LinearLayout>

# Step 6

Add the following code in MainActivity.cs. This is the main activity which plays the most important role and makes your app work correctly by customizing the buttons with background activity.

***MainActivity.cs***

1. **using** System;
2. **using** Android.App;
3. **using** Android.Content;
4. **using** Android.Runtime;
5. **using** Android.Views;
6. **using** Android.Widget;
7. **using** Android.OS;
9. **namespace** LoginSystem
10. {
11. [Activity(Label = "LoginSystem", MainLauncher = **true**, Icon = "@drawable/icon")]
12. **public** **class** MainActivity : Activity
13. {
15. **private** Button mButtnSignup;
16. **protected** **override** **void** OnCreate(Bundle bundle)
17. {
18. **base**.OnCreate(bundle);
20. // Set our view from the "main" layout resource
21. SetContentView(Resource.Layout.Main);
23. mButtnSignup = FindViewById<Button>(Resource.Id.sinup);
24. mButtnSignup.Click += (**object** sender, EventArgs args) =>
25. {
26. FragmentTransaction transcation = FragmentManager.BeginTransaction();
27. Dialogclass signup = **new** Dialogclass();
28. signup.Show(transcation, "Dialog Fragment");
29. };

32. }

35. }
36. }

# Step 7

Then, we need to add the drawable objects in the "drawable" folder and create Buttonsignupstyle.xml and edittextstyle.xml files to add the following code. This gives style to our button and makes the text color.

***ButtonSignup.xml***

1. <selector xmlns:android="http://schemas.android.com/apk/res/android">
3. <item android:state\_pressed="false">
5. <layer-list>
6. <item android:right="5dp" android:top="5dp">
7. <shape>
8. <corners android:radius="2dp"/>
9. <solid android:color="#D6D6D6"/>
10. </shape>
11. </item>

14. <item android:bottom="2dp" android:left="2dp">
15. <shape>
16. <gradient android:angle="270" android:endColor="#F57847" android:startColor="#F57847"/>
17. <stroke android:width="1dp" android:color="#BABABA"/>
18. <corners android:radius="4dp"/>
19. <padding android:bottom="10dp" android:left="10dp" android:right="10dp" android:top="10dp"/>
20. </shape>
21. </item>
22. </layer-list>
23. </item>
25. <item android:state\_pressed="true">
27. <layer-list>
28. <item android:right="5dp" android:top="5dp">
29. <shape>
30. <corners android:radius="2dp"/>
31. <solid android:color="#D6D6D6"/>
32. </shape>
33. </item>

36. <item android:bottom="2dp" android:left="2dp">
37. <shape>
38. <gradient android:angle="270" android:endColor="#E67E55" android:startColor="#F57847"/>
39. <stroke android:width="1dp" android:color="#BABABA"/>
40. <corners android:radius="4dp"/>
41. <padding android:bottom="10dp" android:left="10dp" android:right="10dp" android:top="10dp"/>
42. </shape>
43. </item>
44. </layer-list>
45. </item>
47. </selector>

***edittextstyle.xml***

1. <?xml version="1.0" encoding="utf-8"?>
2. <selector xmlns:android="http://schemas.android.com/apk/res/android">
4. <item android:state\_focused="true">
6. <shape android:shape="rectangle">
8. <gradient
9. android:startColor="#FFF"
10. android:endColor="#FFF">
11. </gradient>

14. <stroke
15. android:width="3dp"
16. android:color="#4A6EA9">
17. </stroke>
19. <corners
20. android:radius="2dp">
21. </corners>
23. <padding
24. android:left="10dp">
25. </padding>
27. </shape>
28. </item>
30. <item>
31. <shape>
33. <gradient
34. android:startColor="#FFF"
35. android:endColor="#FFF">
36. </gradient>
38. <corners
39. android:radius="2dp">
40. </corners>
42. <padding
43. android:left="10dp"
44. android:top="6dp"
45. android:right="10dp"
46. android:bottom="6dp">
47. </padding>
49. </shape>
50. </item>
52. </selector>

# Step 8

Then, we need to create the Dialog.cs file to our project. Here, it must be defined as class and add the following code in it.

***Dialog.Cs***

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Text;
6. **using** Android.App;
7. **using** Android.Content;
8. **using** Android.OS;
9. **using** Android.Runtime;
10. **using** Android.Views;
11. **using** Android.Widget;
13. **namespace** LoginSystem
14. {
15. **class** Dialogclass: DialogFragment
16. {
17. **public** **override** View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
18. {
19. **base**.OnCreateView(inflater, container, savedInstanceState);
20. var view = inflater.Inflate(Resource.Layout.dialog\_sign\_up, container, **false**);
21. **return** view;
23. }
24. }
25. }

# Step 9

Add the following code in dialog\_sign\_up.Xaml source page. This is our designer page and here, our dialog\_sign\_up UI design will be displayed. So, download any message icon.png from the internet and add that to drawable folder in Solution Explorer.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@android:color/background\_light">

<EditText

android:id="@+id/Login\_FName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="FIRST NAME"

android:textColor="#000000"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textSize="25sp"

android:gravity="center"

android:layout\_weight="1"

android:background="@android:color/darker\_gray"

android:textStyle="bold" />

<EditText

android:id="@+id/Login\_Email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="EMAIL"

android:textColor="#000000"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textSize="25sp"

android:gravity="center"

android:layout\_weight="1"

android:background="@android:color/darker\_gray"

android:textStyle="bold" />

<EditText

android:id="@+id/Login\_Password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:password="true"

android:hint="Password"

android:textColor="#000000"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textSize="25sp"

android:gravity="center"

android:layout\_weight="1"

android:background="@android:color/darker\_gray"

android:textStyle="bold" />

<Button

android:id="@+id/Login\_Sinup"

android:text="Sign Up"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="30sp"

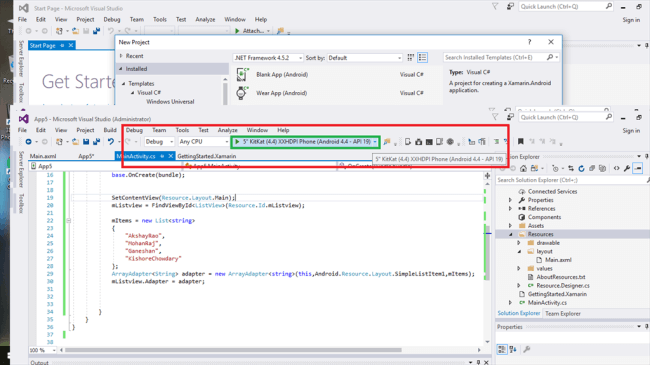
android:background="@drawable/buttonsignupstyle"

android:drawableLeft="@drawable/Kajan\_Twitter" />

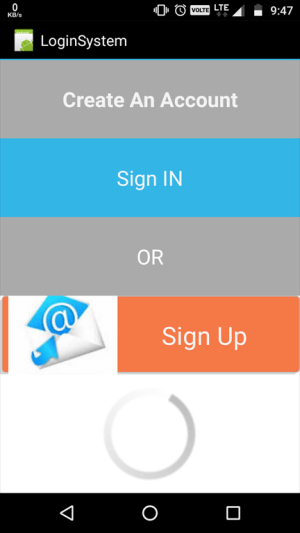
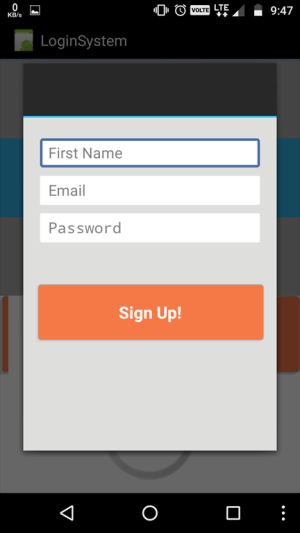
</LinearLayout>

# Final Step

Then, run the app using Visual Studio Android Emulator and here we go.



Our app gets executed.

**Conclusion**

If you have any doubts or queries, please comment below. In the next tutorial, I will show you how to create Sliding Tab Interface.