**Xamarin 101 Workshop**

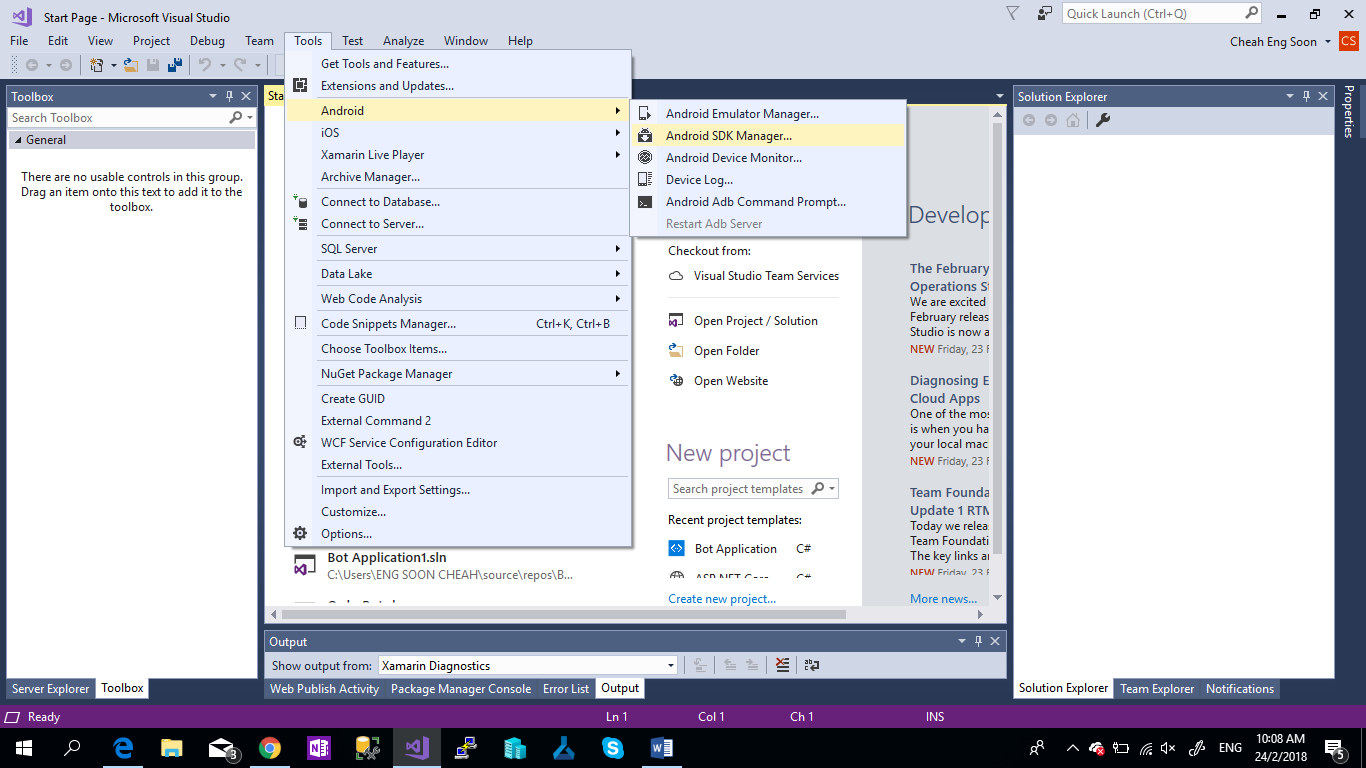
**Objective:** Build your first mobile app for iOS, Android and Windows with Xamarin.Forms.

**Requirements:**

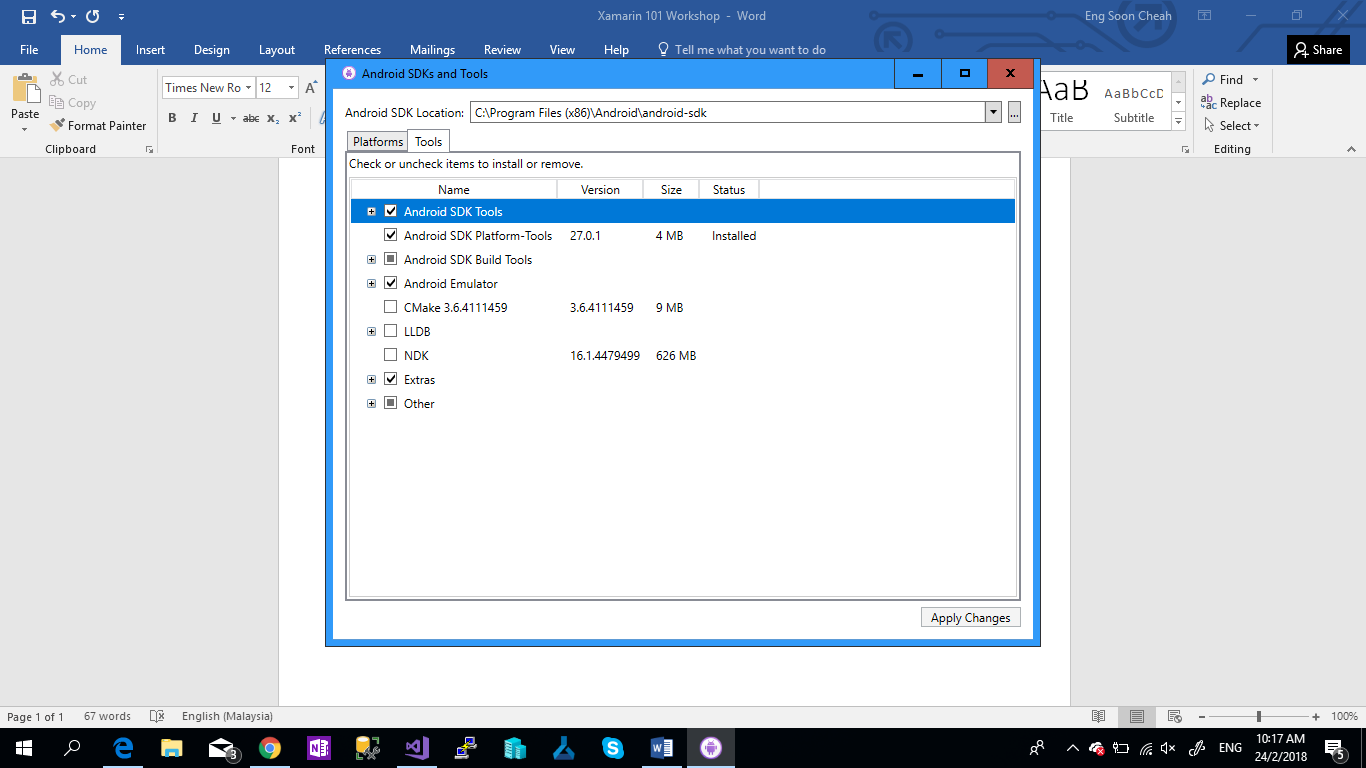
1. Install Visual Studio 2017 with Xamarin
2. Install Android SDK Manager for Tools and Extras (Last Step)

**How to install Android SDK Manager for Tools and Extras**

1. Go to **Tools** > **Android** > **Android SDK Manager**



1. Click **Yes** for Launch Android SDK Manager.
2. Select **Tools** Tab, Check [ /] Android SDK Tools and Check [ /] Extras. Lastly, click Apply Changes.



**Content**

**Chapter 1:** Create Your First Project

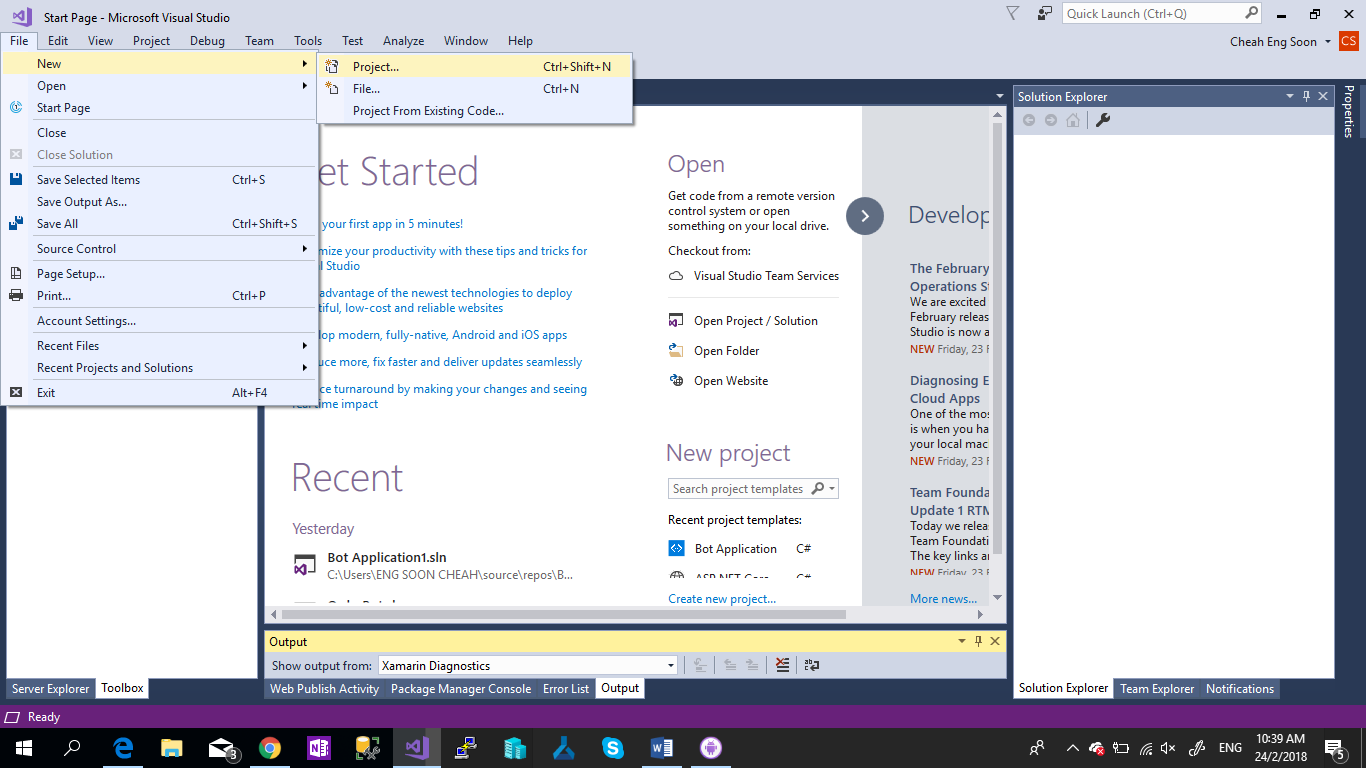
**Chapter 2:** Layout & Control & Event Handler

**Chapter 3:** Plugin

**References**

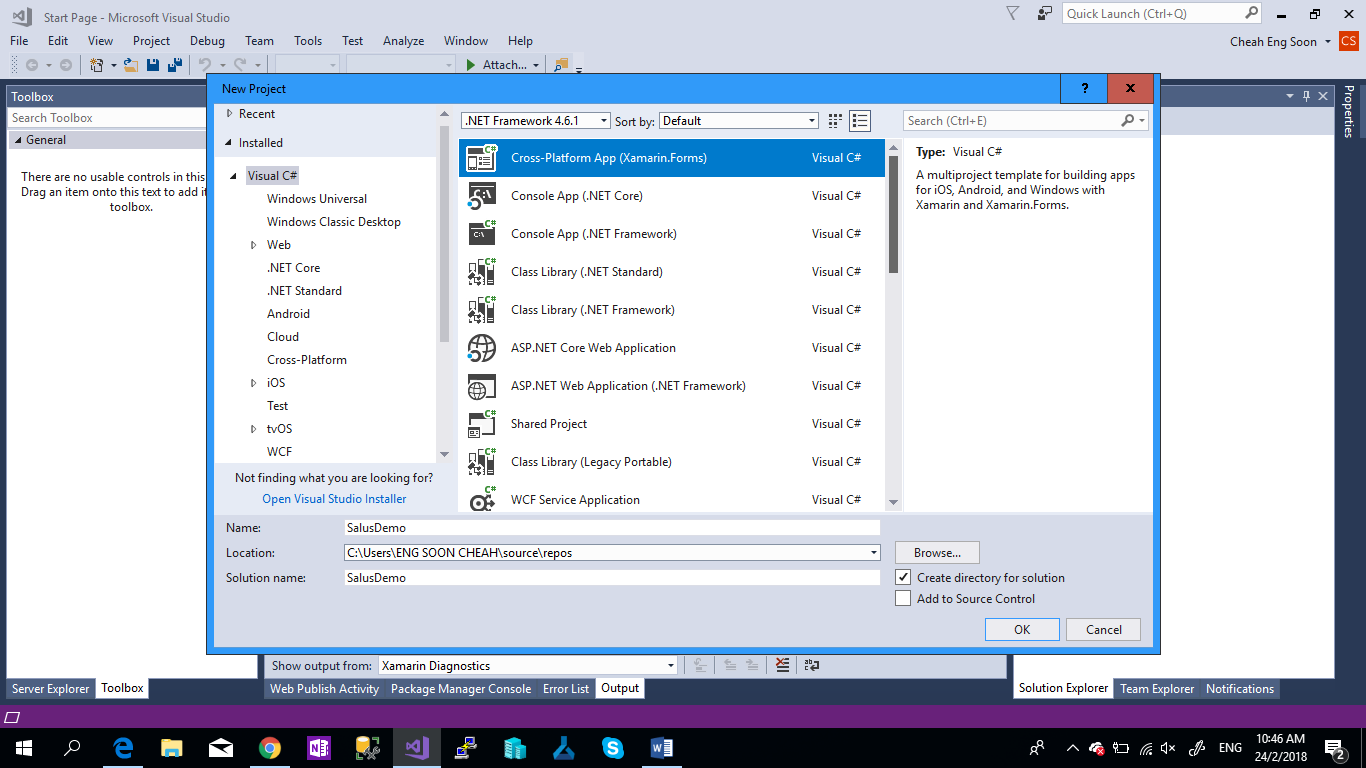
**Chapter 1: Create Your First Project**

1. Open your Visual Studio 2017.
2. Go to **File** > **New** > **Project**

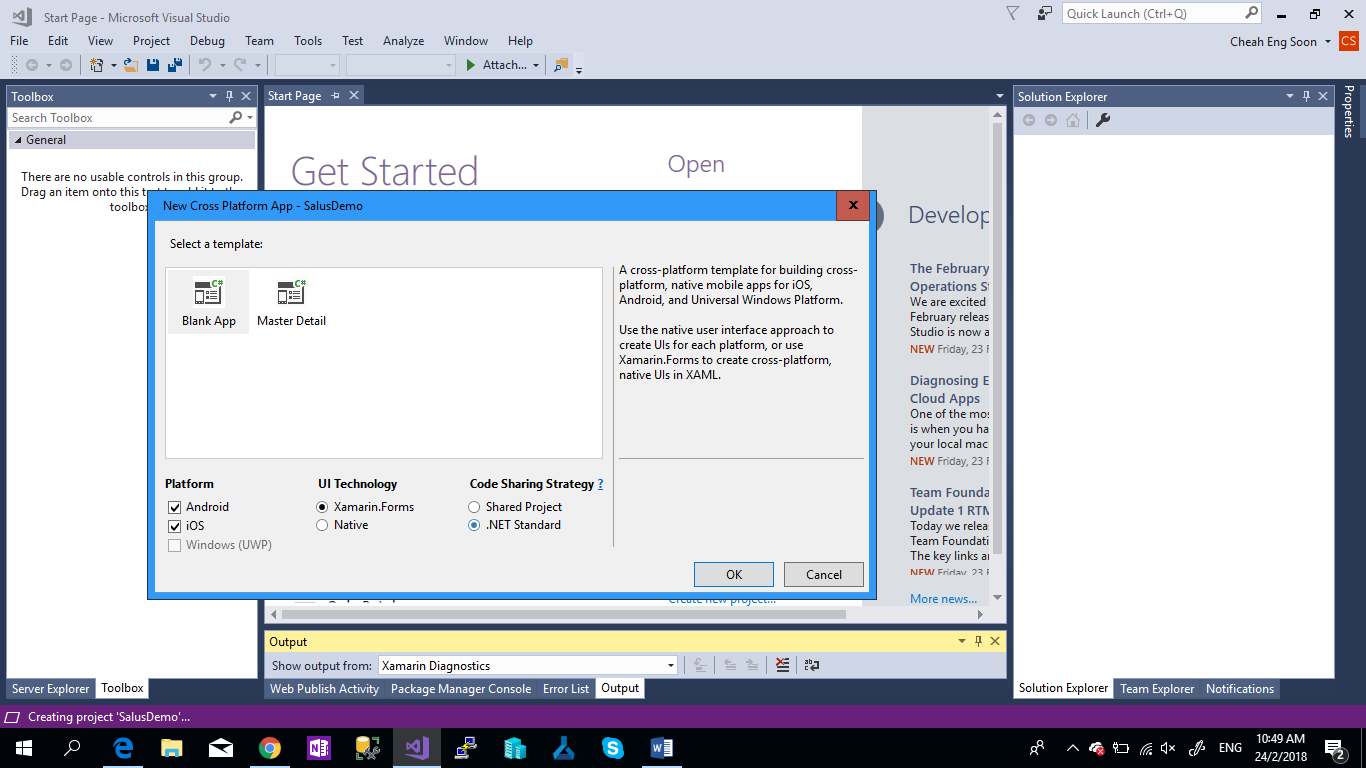


1. Select **Visual C#** > Select **Cross-Platform App (Xamarin.Forms)** > Name the Project

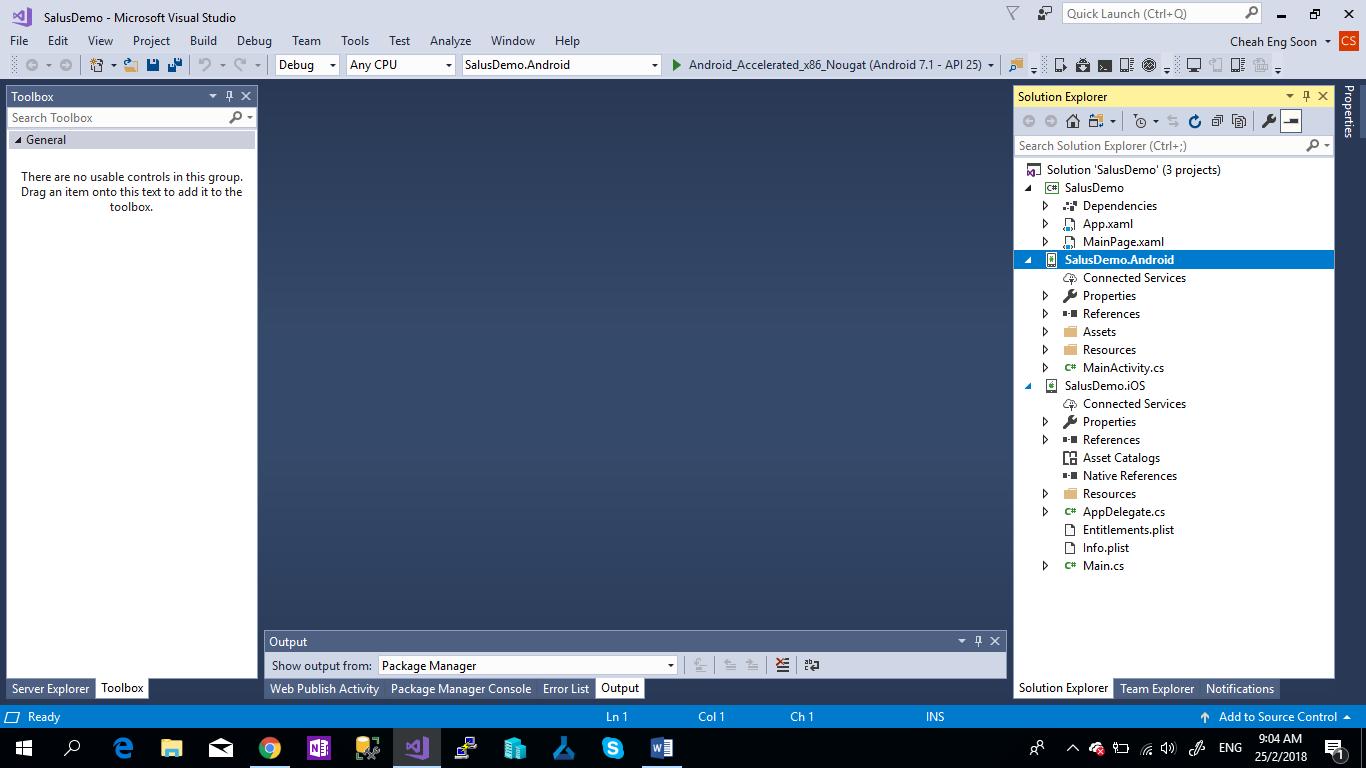
As **SalusDemo >** Click **OK**



1. At the **Code Sharing Strategy**, select **.Net Standard** and Click **OK**.



1. Now your Xamarin Mobile application success created.



**Chapter 2: Layout & Control & Event Handler**

1. Go to “**MainPage.xaml**” and Layout and Control Event.

|  |
| --- |
| <ContentPage.Content> |
|  | <!--Grid Layout--> |
|  | <Grid VerticalOptions="FillAndExpand" HorizontalOptions="FillAndExpand" > |
|  | <Grid.RowDefinitions> |
|  | <RowDefinition Height="\*" /> |
|  | <RowDefinition Height="2\*"/> |
|  | <RowDefinition Height="2\*" /> |
|  | <RowDefinition Height="\*" /> |
|  | </Grid.RowDefinitions> |
|  | <Grid.ColumnDefinitions> |
|  | <ColumnDefinition Width="\*" /> |
|  | <ColumnDefinition Width="\*" /> |
|  | </Grid.ColumnDefinitions> |
|  | <!--Controls--> |
|  | <Label x:Name="lbllocation" Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" BackgroundColor="Blue" TextColor="White" FontSize="35"/> |
|  | <Button x:Name="btnmessage" Grid.Row="1" Grid.Column="0" Image="sms.png" Clicked="SendSMS" /> |
|  | <Button x:Name="btnalarm" Grid.Row="1" Grid.Column="1" Image="alarm.png" Clicked="OnAlertYesNoClicked"/> |
|  | <Button x:Name="btnPolice" Grid.Row="2" Grid.Column="0" Image="police.png" Clicked="CallPolice" /> |
|  | <Button x:Name="btnAmbulance" Grid.Row="2" Grid.Column="1" Image="hospital.png" Clicked="CallHospital" /> |
|  |  |
|  | <Entry x:Name="txtPhoneNumber" Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" |
|  | Placeholder="Phone Number" TextChanged="PhoneNo\_TextChanged" Keyboard="Numeric" TextColor="Red" FontAttributes="Bold" FontSize="40" /> <!--aka Textbox--> |
|  | </Grid> |
|  | </ContentPage.Content> |

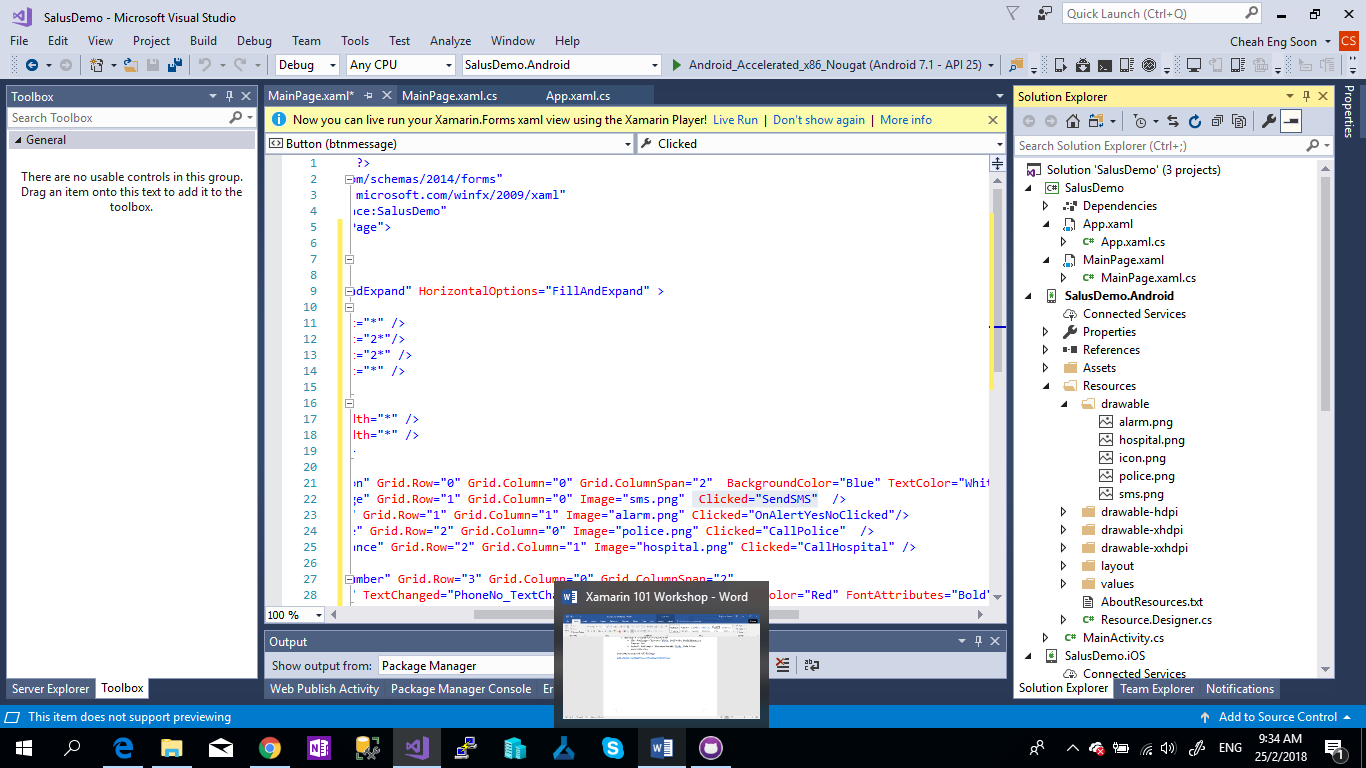
1. Add Image to each Platform (iOS and Android)

* iOS – Add Image to “**Resources**” Folder , **Build Action: Bundle Resource** in Properties Tab.
* Android – Add Image to “**Resources/drawable**” Folder, **Build Action: Android Resource**

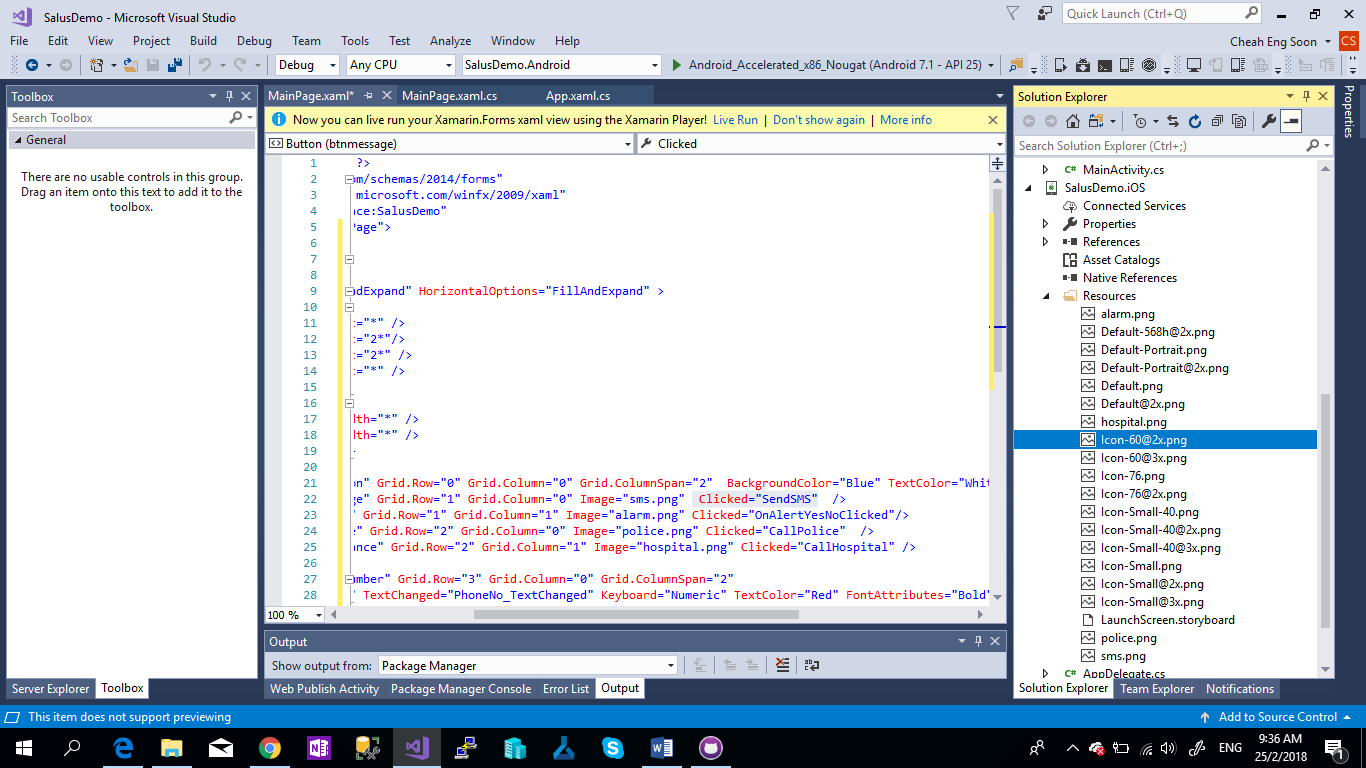
Download the image and MP3 file From:

<https://github.com/cheahengsoon/FirstXamarinFormsApp>

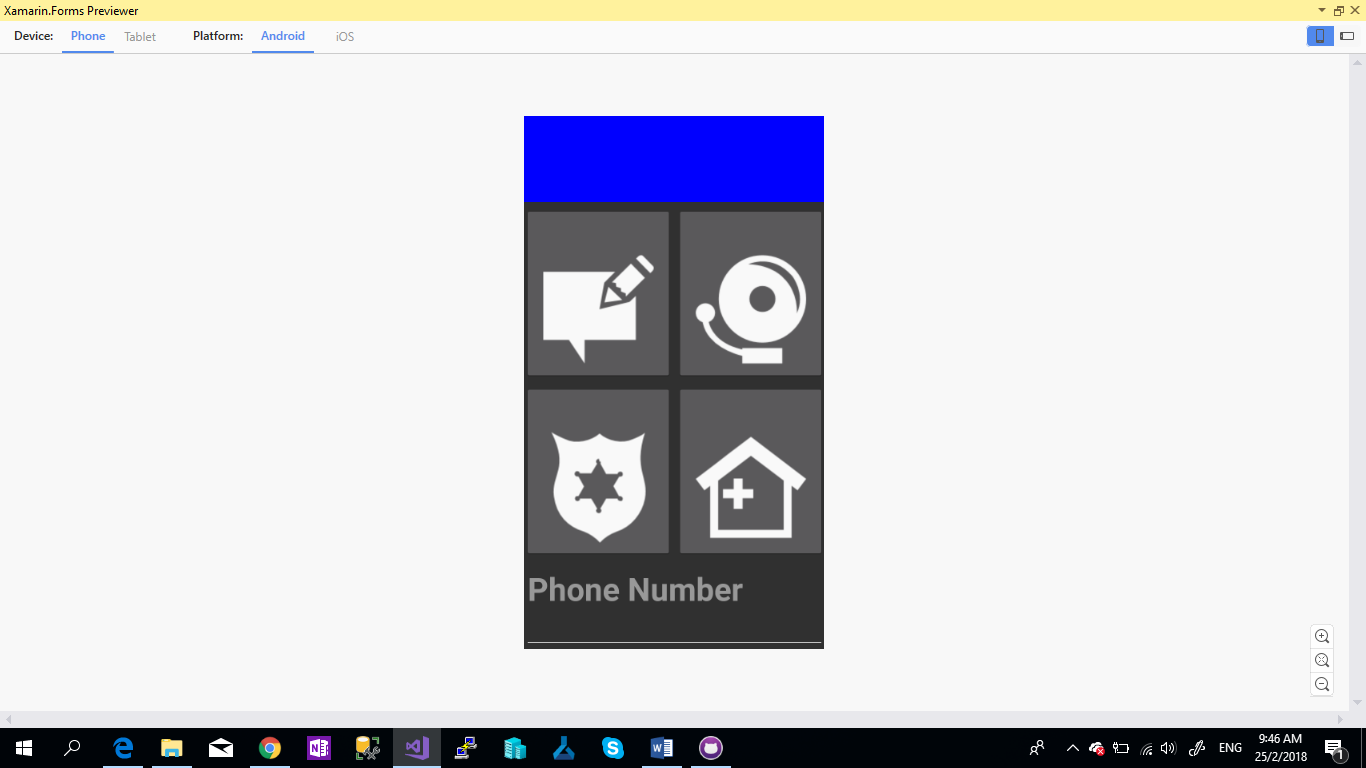
As shown in Xamarin.Android



As shown in Xamarin.iOS



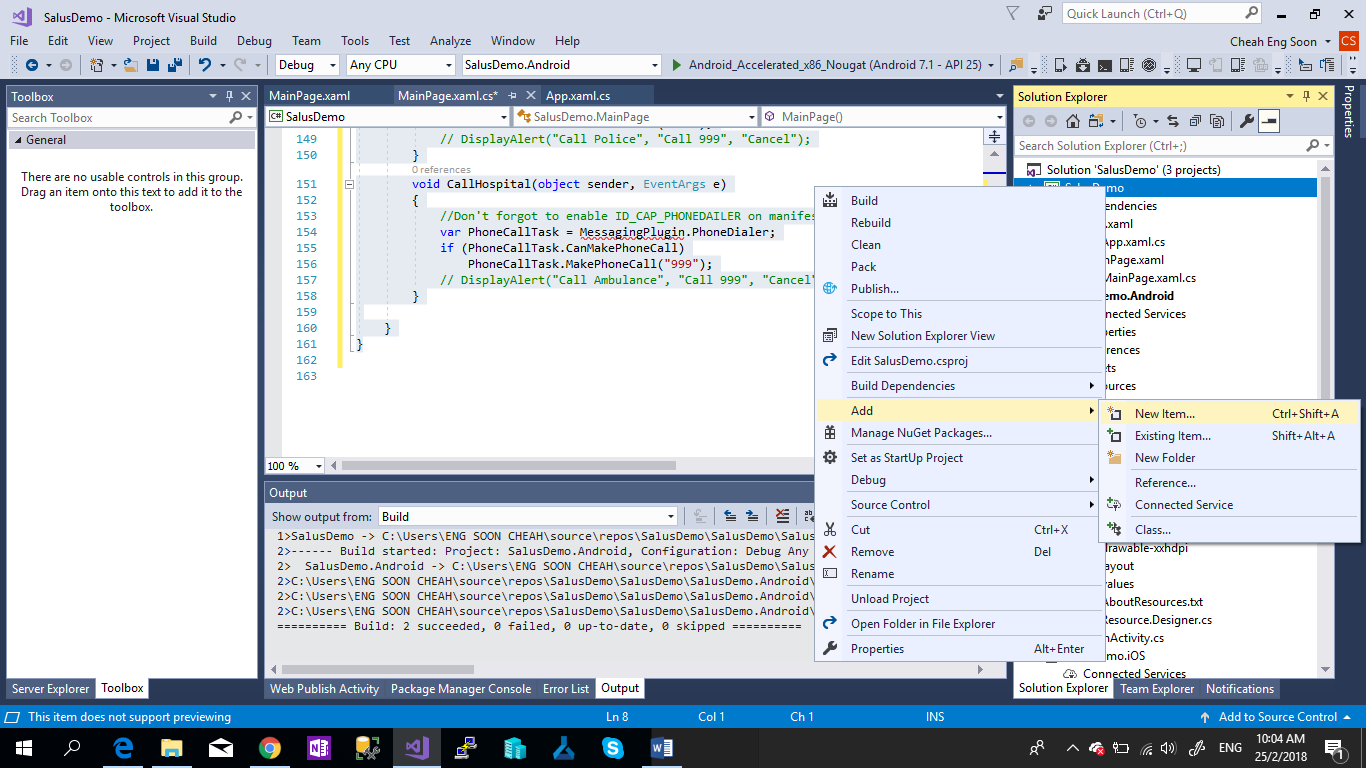
As result shown in xaml previewer but still not had any functions?



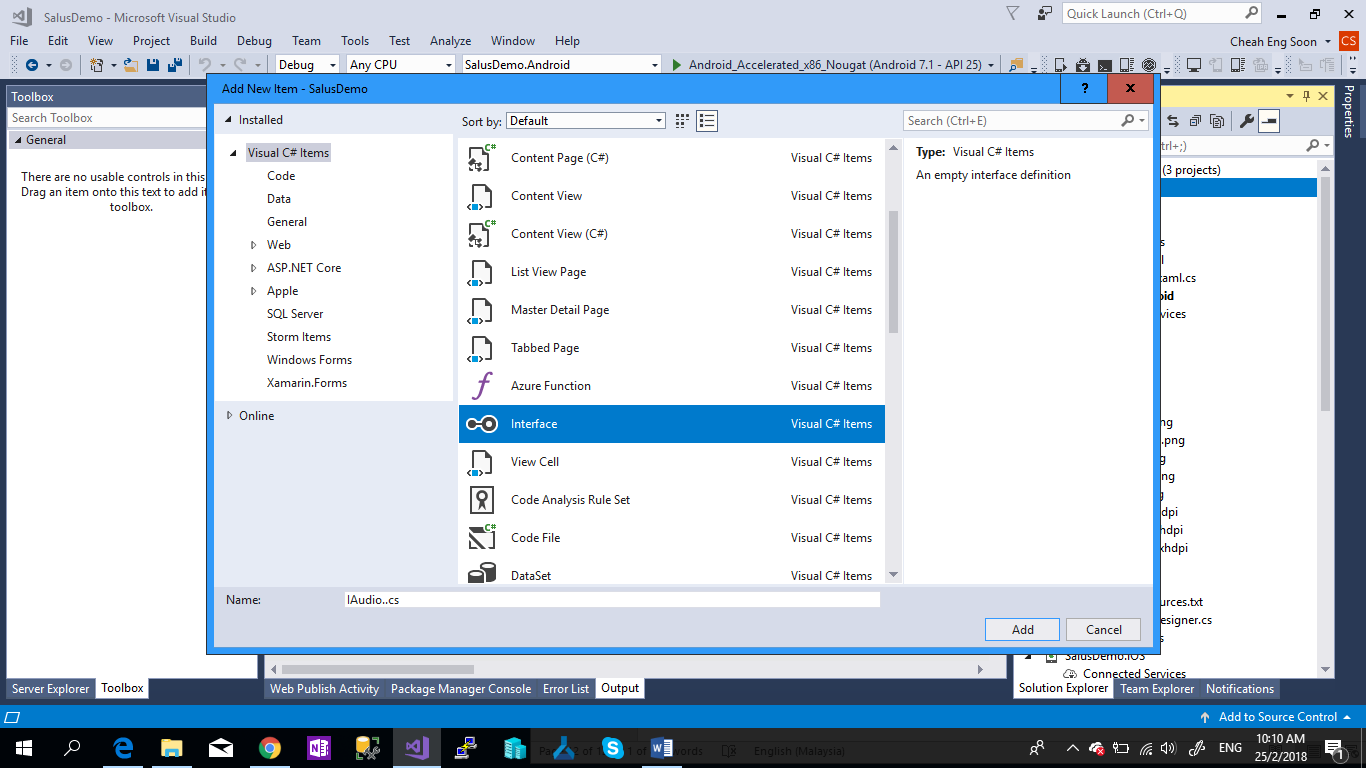
1. Add Event handler to MainPage.xaml.cs , add the C# Code

|  |  |
| --- | --- |
|  | public partial class MainPage : ContentPage |
|  | { |
|  | public MainPage() |
|  | { |
|  | InitializeComponent(); |
|  |  |
|  | //Read the Phone Number from Temporary Storage |
|  | if (Application.Current.Properties.ContainsKey("PhoneNo")) |
|  | { |
|  | var phonenumber = (string)Application.Current.Properties["PhoneNo"]; |
|  | txtPhoneNumber.Text = phonenumber; |
|  | } |
|  | // AskPermission(); |
|  |  |
|  | GetGPS(); |
|  | GetTrack(); |
|  | } |
|  |  |
|  | //private async void AskPermission() |
|  | //{ |
|  | // var status = await CrossPermissions.Current.CheckPermissionStatusAsync(Permission.Location); |
|  | // if (status != PermissionStatus.Granted) |
|  | // { |
|  | // if (await CrossPermissions.Current.ShouldShowRequestPermissionRationaleAsync(Permission.Location)) |
|  | // { |
|  | // await DisplayAlert("Need location", "Gunna need that location", "OK"); |
|  | // } |
|  |  |
|  | // var results = await CrossPermissions.Current.RequestPermissionsAsync(Permission.Location); |
|  | // status = results[Permission.Location]; |
|  | // } |
|  | //} |
|  |  |
|  | private async void GetTrack() |
|  | { |
|  | try |
|  | { |
|  | if (CrossGeolocator.Current.IsListening) |
|  | { |
|  | await CrossGeolocator.Current.StopListeningAsync(); |
|  | lbllocation.Text = "Stopped tracking"; |
|  |  |
|  | } |
|  | else |
|  | { |
|  | if (await CrossGeolocator.Current.StartListeningAsync(30000, 0)) |
|  | { |
|  | lbllocation.Text = "Started tracking"; |
|  |  |
|  | } |
|  | } |
|  | } |
|  | catch //(Exception ex) |
|  | { |
|  | //Xamarin.Insights.Report(ex); |
|  | // await DisplayAlert("Uh oh", "Something went wrong, but don't worry we captured it in Xamarin Insights! Thanks.", "OK"); |
|  | } |
|  | } |
|  |  |
|  | public async void GetGPS() |
|  | { |
|  | try |
|  | { |
|  | var locator = CrossGeolocator.Current; |
|  | locator.DesiredAccuracy = 1000; |
|  | lbllocation.Text = "Getting gps"; |
|  |  |
|  | var position = await locator.GetPositionAsync(timeoutMilliseconds: 10000); |
|  |  |
|  | if (position == null) |
|  | { |
|  | lbllocation.Text = "null gps :("; |
|  | return; |
|  | } |
|  | lbllocation.Text = string.Format("Time: {0} \nLat: {1} \nLong: {2}"+ |
|  | " \nAltitude: {3} \nAltitude Accuracy: {4} \nAccuracy: {5} \nHeading: {6} \nSpeed: {7}", |
|  | position.Timestamp, position.Latitude, position.Longitude, |
|  | position.Altitude, position.AltitudeAccuracy, position.Accuracy, position.Heading, position.Speed); |
|  |  |
|  | } |
|  | catch //(Exception ex) |
|  | { |
|  | // Xamarin.Insights.Report(ex); |
|  | // await DisplayAlert("Uh oh", "Something went wrong, but don't worry we captured it in Xamarin Insights! Thanks.", "OK"); |
|  | } |
|  | } |
|  | protected override void OnAppearing() |
|  | { |
|  | base.OnAppearing(); |
|  | try |
|  | { |
|  | CrossGeolocator.Current.PositionChanged += CrossGeolocator\_Current\_PositionChanged; |
|  | CrossGeolocator.Current.PositionError += CrossGeolocator\_Current\_PositionError; |
|  | } |
|  | catch |
|  | { |
|  | } |
|  | } |
|  |  |
|  | void CrossGeolocator\_Current\_PositionError(object sender, Plugin.Geolocator.Abstractions.PositionErrorEventArgs e) |
|  | { |
|  |  |
|  | lbllocation.Text = "Location error: " + e.Error.ToString(); |
|  | } |
|  |  |
|  | void CrossGeolocator\_Current\_PositionChanged(object sender, Plugin.Geolocator.Abstractions.PositionEventArgs e) |
|  | { |
|  | var position = e.Position; |
|  | lbllocation.Text = string.Format("Time: {0} \nLat: {1} \nLong: {2} \nAltitude: " |
|  | + "{3} \nAltitude Accuracy: {4} \nAccuracy: {5} \nHeading: {6} \nSpeed: {7}", |
|  | position.Timestamp, position.Latitude, position.Longitude, |
|  | position.Altitude, position.AltitudeAccuracy, position.Accuracy, position.Heading, position.Speed); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | protected override void OnDisappearing() |
|  | { |
|  | base.OnDisappearing(); |
|  | try |
|  | { |
|  | CrossGeolocator.Current.PositionChanged -= CrossGeolocator\_Current\_PositionChanged; |
|  | CrossGeolocator.Current.PositionError -= CrossGeolocator\_Current\_PositionError; |
|  | } |
|  | catch |
|  | { |
|  | } |
|  | } |
|  |  |
|  | private void PhoneNo\_TextChanged(object sender,TextChangedEventArgs e) |
|  | { |
|  | //Temporary Storage for Phone Number from txtxPhoneNumber |
|  | var phonenumber = txtPhoneNumber.Text; |
|  | Application.Current.Properties["PhoneNo"] = phonenumber; |
|  | } |
|  |  |
|  | void OnAlertYesNoClicked(object sender, EventArgs e) |
|  | { |
|  | DependencyService.Get<IAudio>().PlayAudioFile("policesiren.mp3"); |
|  | } |
|  |  |
|  | void SendSMS(object sender, EventArgs e) |
|  | { |
|  | var SmsTask = MessagingPlugin.SmsMessenger; |
|  |  |
|  | if (SmsTask.CanSendSms) |
|  | SmsTask.SendSms(txtPhoneNumber.Text, "I'm in Danger,My Location: "+ lbllocation.Text); |
|  | } |
|  | void CallPolice(object sender,EventArgs e) |
|  | { |
|  | //Don't forgot to enable ID\_CAP\_PHONEDAILER on manifest file |
|  | var PhoneCallTask = MessagingPlugin.PhoneDialer; |
|  | if (PhoneCallTask.CanMakePhoneCall) |
|  | PhoneCallTask.MakePhoneCall("999"); |
|  | // DisplayAlert("Call Police", "Call 999", "Cancel"); |
|  | } |
|  | void CallHospital(object sender, EventArgs e) |
|  | { |
|  | //Don't forgot to enable ID\_CAP\_PHONEDAILER on manifest file |
|  | var PhoneCallTask = MessagingPlugin.PhoneDialer; |
|  | if (PhoneCallTask.CanMakePhoneCall) |
|  | PhoneCallTask.MakePhoneCall("999"); |
|  | // DisplayAlert("Call Ambulance", "Call 999", "Cancel"); |
|  | } |

1. Now Create a Interface class for Play Audio for Alarm
2. Right Click Project “SalusDemo” , Select “Add” > Select “New item”.

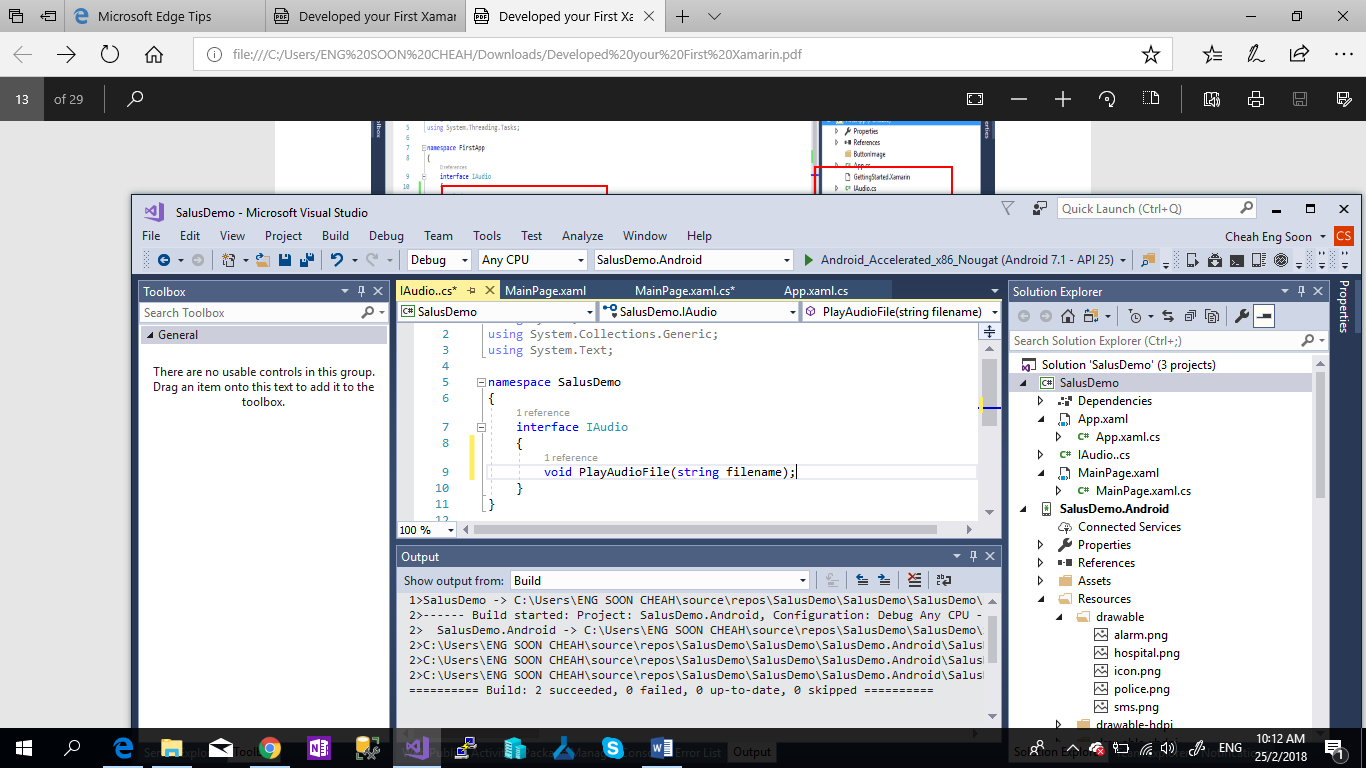


1. Select “Interface” and name the class as “IAudio.cs”. and click “Add”.



1. “IAudio.cs” Success created, and insert this line of code as below.

**void PlayAudioFile(string filename);**

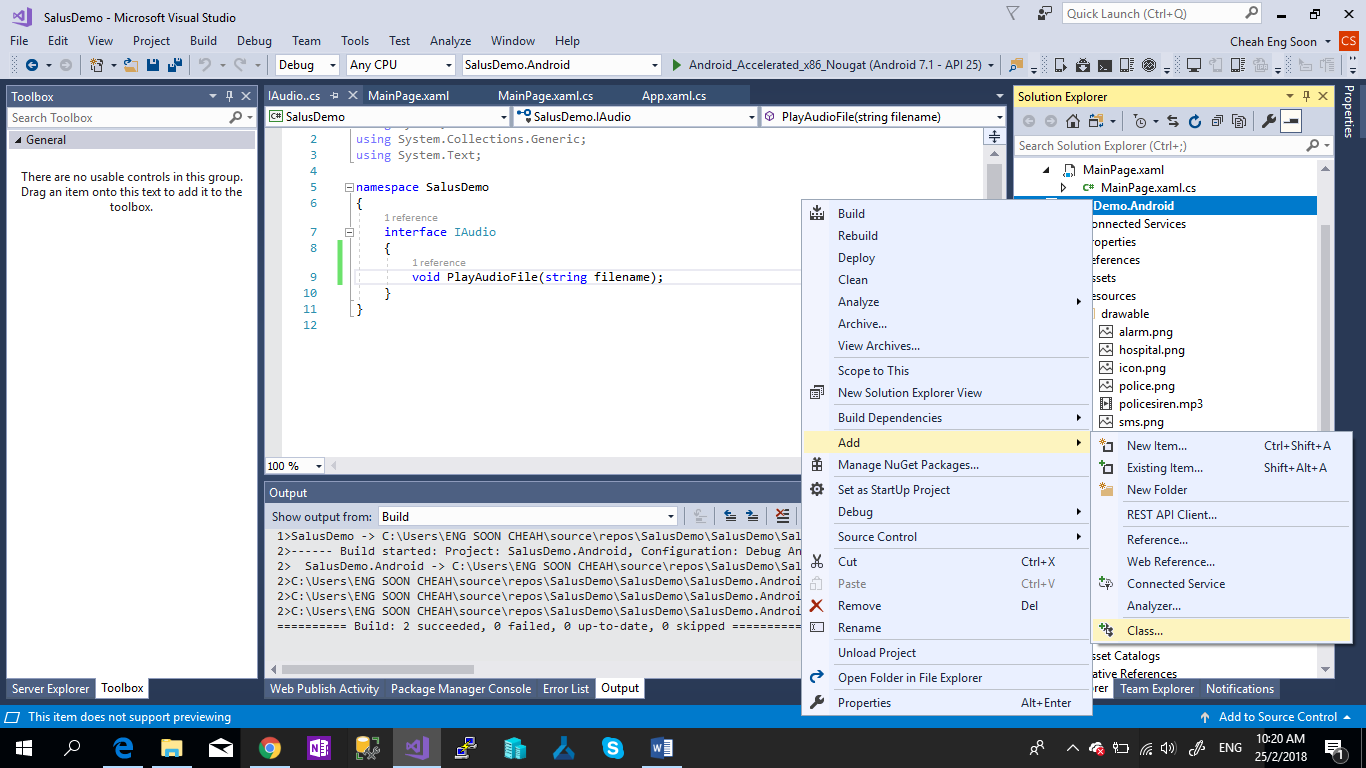


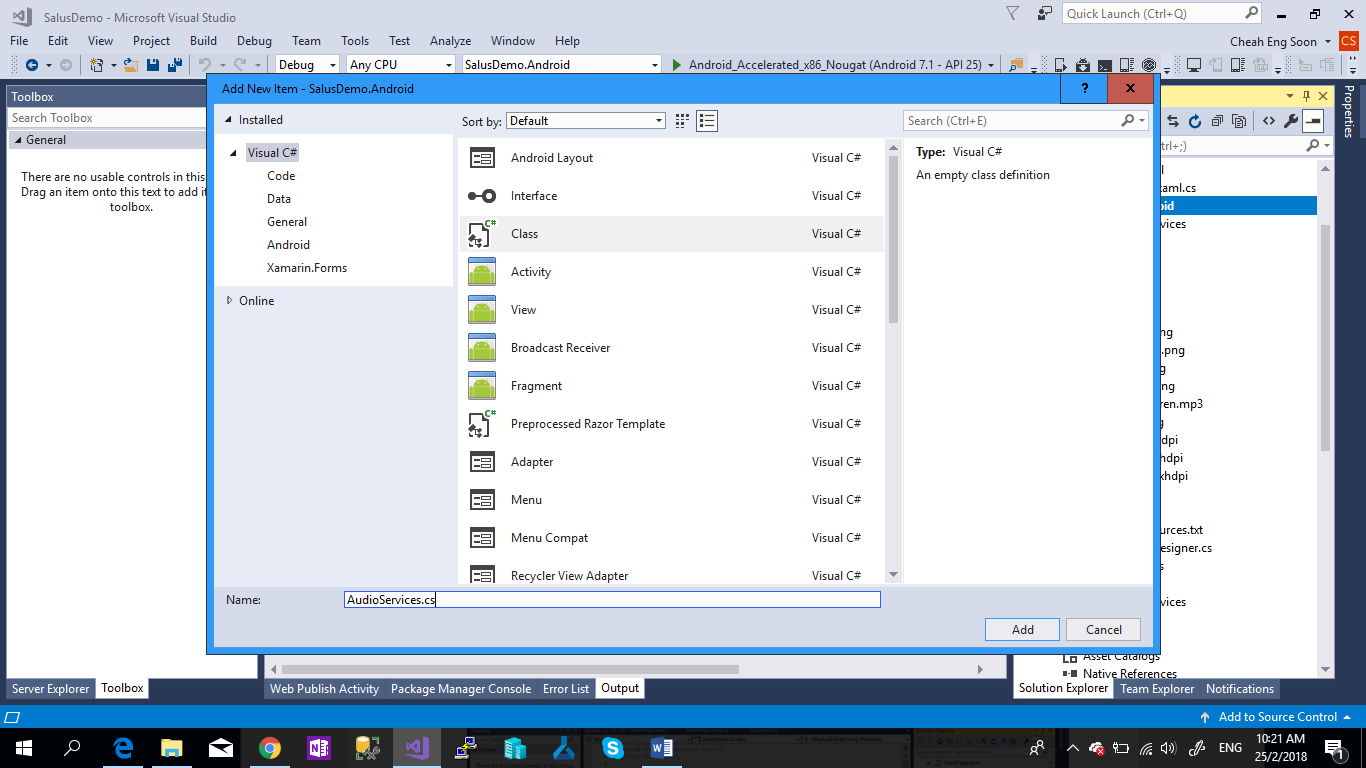
1. Now add the **policesiren.mp3** to **Xamarin.Android** and **Xamarin.iOS** same as **Step 2**.

\*\*iOS: file should be added to the Resource Folder.

\*\*Android: file should be added to the Assets Folder.

1. Add Class “**AudioServices.cs**” to **SalusDemo.Android**.





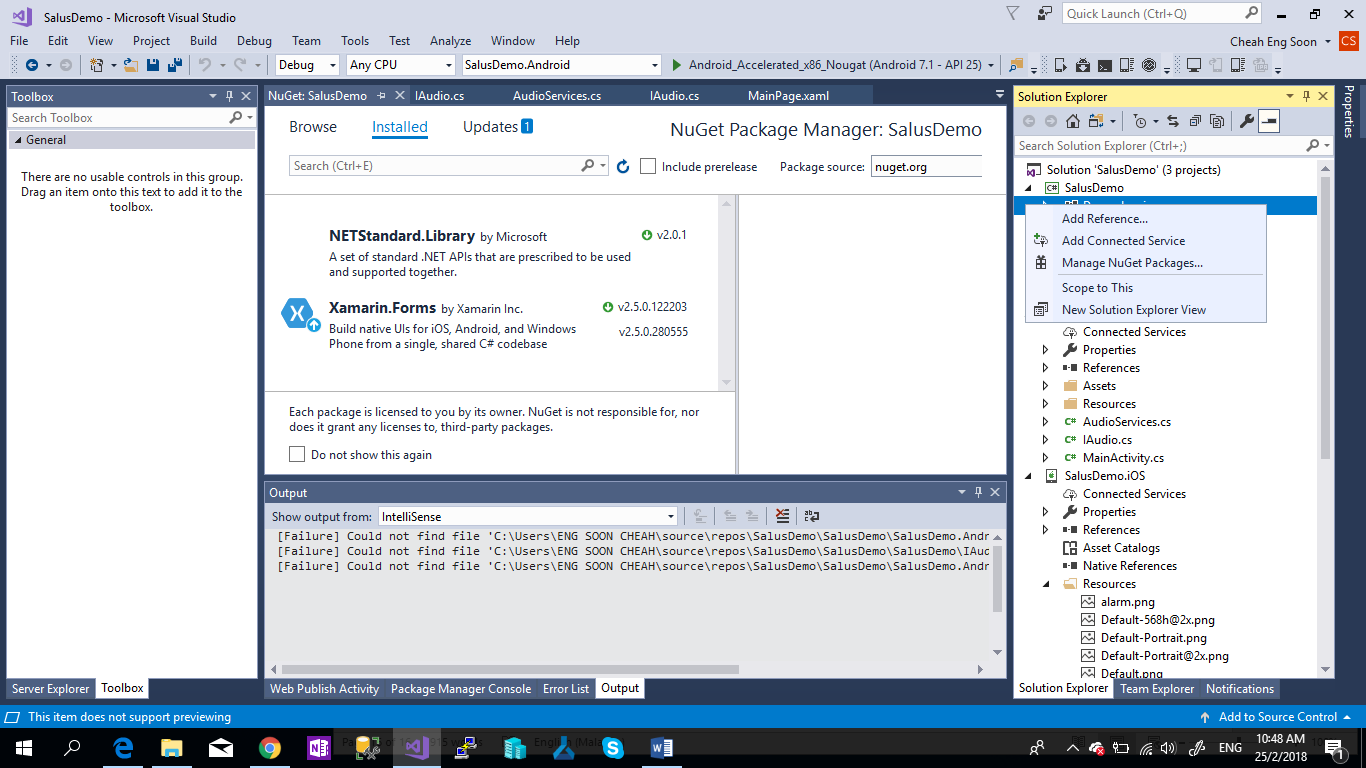
1. Add the Code to “**AudioServices.cs**”

|  |
| --- |
| [assembly: Dependency(typeof(AudioService))] |
|  |  |
|  | Namespace SalusDemo.Android |
|  | { |
|  | class AudioService: IAudio |
|  | { |
|  | public AudioService()  { |
|  | public void PlayAudioFile(string fileName) |
|  | { |
|  | var player = new MediaPlayer(); |
|  | var fd = global::Android.App.Application.Context.Assets.OpenFd(fileName); |
|  | player.Prepared += (s, e) => |
|  | { |
|  | player.Start(); |
|  | }; |
|  | player.SetDataSource(fd.FileDescriptor, fd.StartOffset, fd.Length); |
|  | player.Prepare(); |
|  | } |
|  | } |
|  |  |
|  |  |
|  |  |
|  | } |

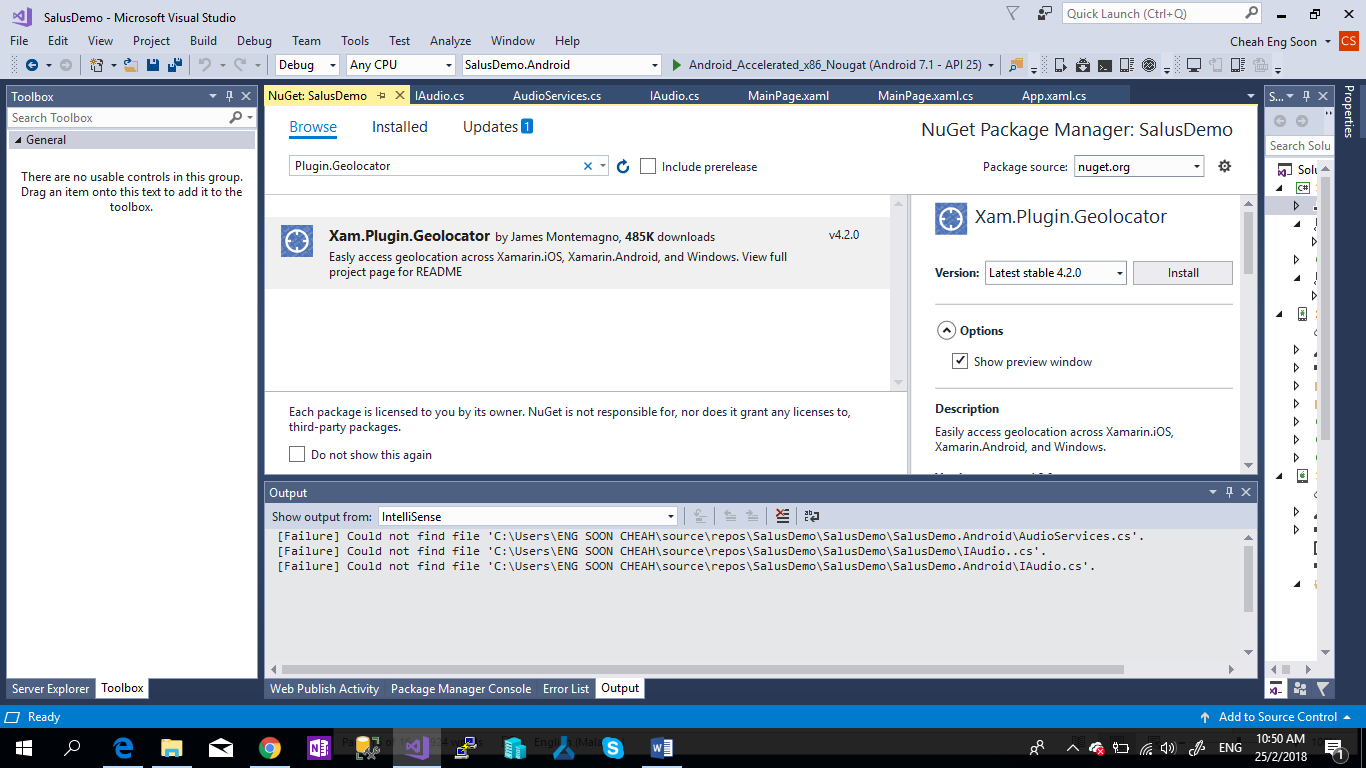
1. Repeat the Step 5 for iOS .

**Chapter 3 - Plugin**

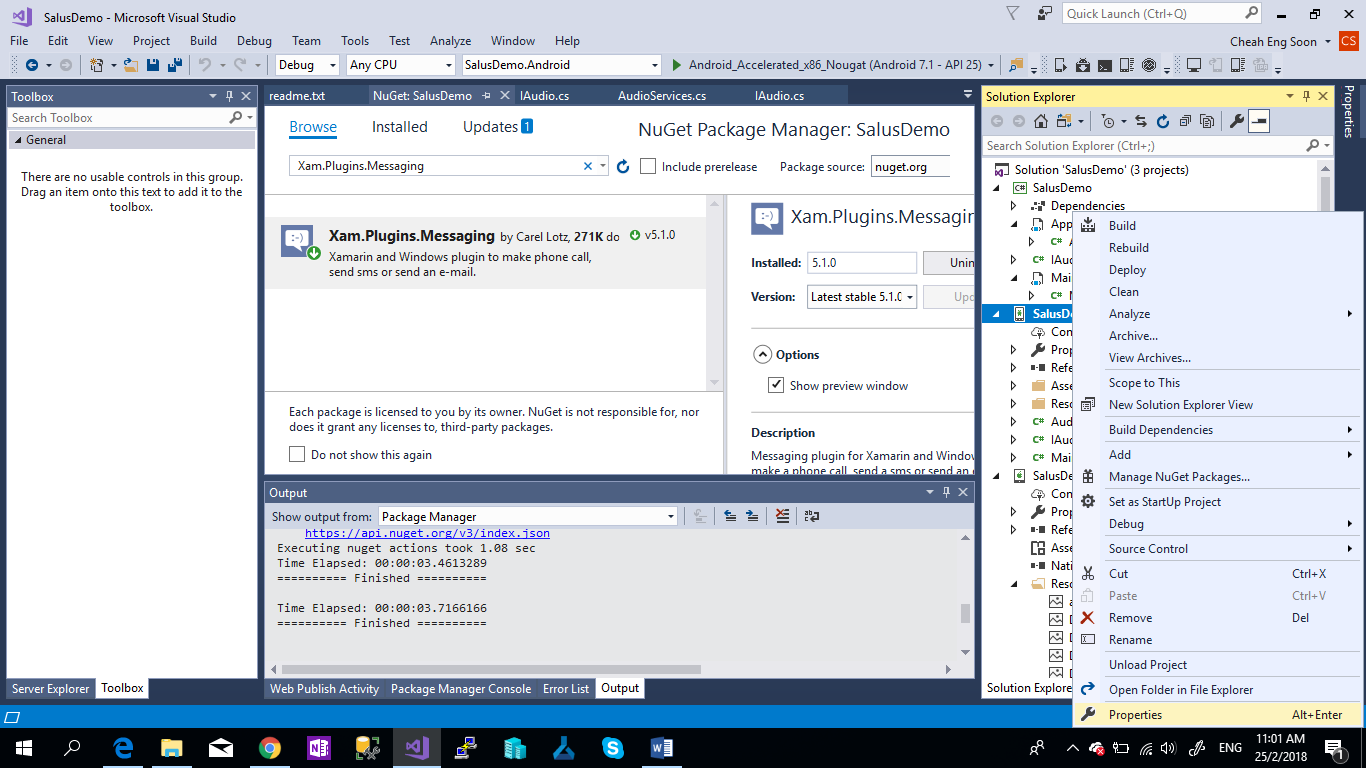
1. Right Click **Dependencies** > Select “**Manage Nuget Package**s”.



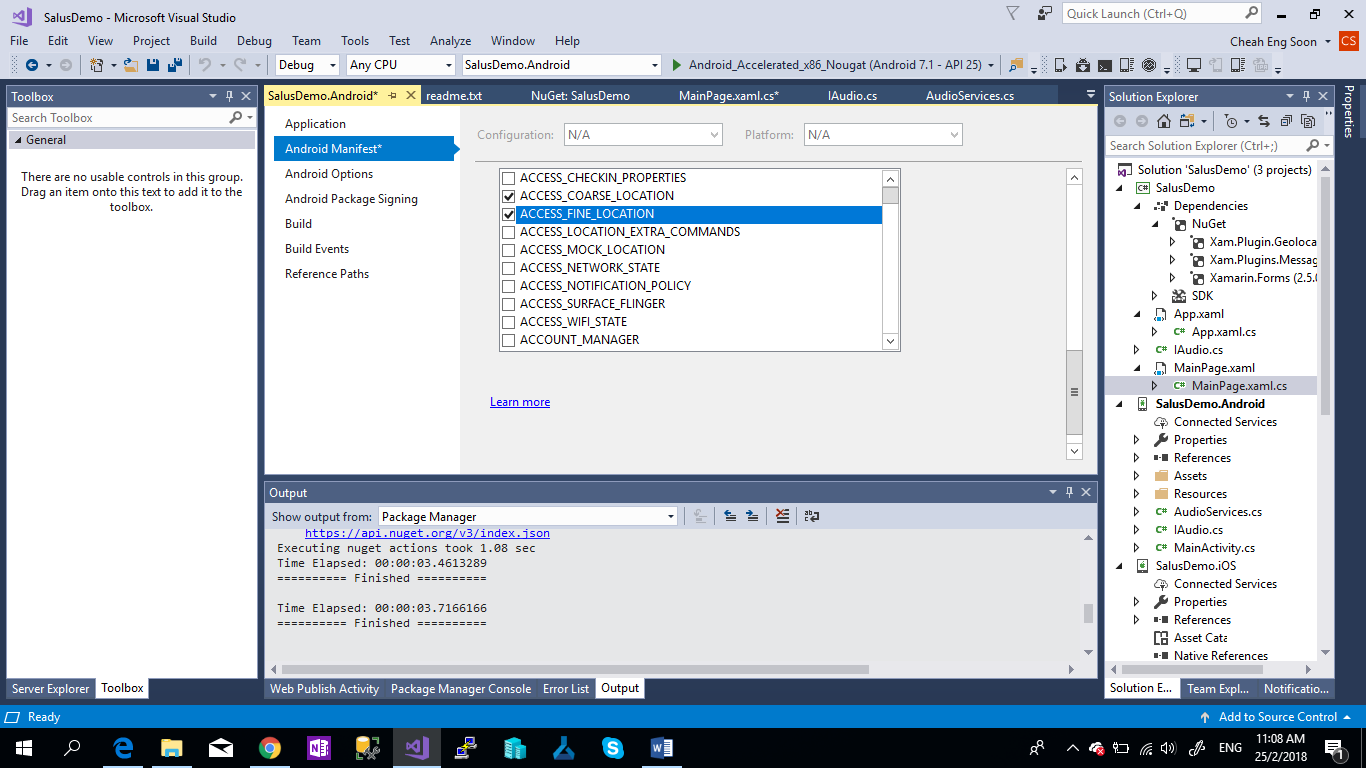
1. Search for **Plugin.Geolocator** and Select and Click **Install**.



1. Repeat the **Step 2** for **Xam.Plugins.Messaging**
2. Now Setup the Permission for Android
3. Right Click SalusDemo.Android > Select Properties



1. Select the Android Manifest Tab , and Enable the Location permission



1. Repeat the **Step 4b**. for **CALL\_PHONE** and **SEND\_SMS**.
2. Debug your application.