## Cross-Platform Mobile Application Development with Xamarin: Module 3, Lesson 6

## Build a Basic iOS app using Xamarin.iOS Lab

## Overview

Xamarin is a powerful tool that allows you to develop mobile applications across all major mobile platforms. This lab will cover how to build a very basic iOS application using Xamarin.iOS.

Objectives

In this hands-on lab you will learn how to:

* Interact with Xamarin, specifically in the context of building an iOS app
* Create a text input field
* Display a dialog box on the click of a button

Prerequisites

The following are required to complete this hands-on lab:

* Visual Studio with Xamarin installed or Xamarin Studio for OS X
* A mac with remote login enabled or access to a mac server
* Completion of [Module 3 Lessons 1-6](https://github.com/MSFTImagine/computerscience/tree/master/Instructor-Led/Module3/Lessons) and [Module 5 Lesson 5 Lab](https://github.com/MSFTImagine/computerscience/tree/master/Instructor-Led/Module3/Labs)

Exercises

This hands-on lab includes the following exercises:

* Exercise 1: Create a First Name Entry UI

Exercise 1: Create a First Name Entry UI

Create a First Name Entry UI that displays the name in a popup when the button is clicked.

1. Create a Visual Studio Visual C#> iOS > Universal > Blank App (iOS) Project.
2. Create a new UI page by right clicking the project and selecting Add > New Item. Choose Apple > iOS > ViewController and name the file ViewControllerName. (Tip: Multi-page iOS apps often use the Storyboards but since this example only uses one page, we’ll just use a UIViewController and .xib file.)
3. This action automatically creates the ViewControllerName.xib file and the ViewControllerName class inherited from UIViewController. The two are linked in the view controller class by a parameter in the base class, which refers to the .xib name.

public partial class ViewControllerName : UIViewController

{

public ViewControllerName() : base("ViewControllerName", null)

{

}

1. Set the startup page as the new ViewController. In AppDelegate.cs FinishedLaunching method, instantiate the ViewControllerName class and assign to the Window.RootViewController property.

public override bool FinishedLaunching(UIApplication application, NSDictionary launchOptions)

{

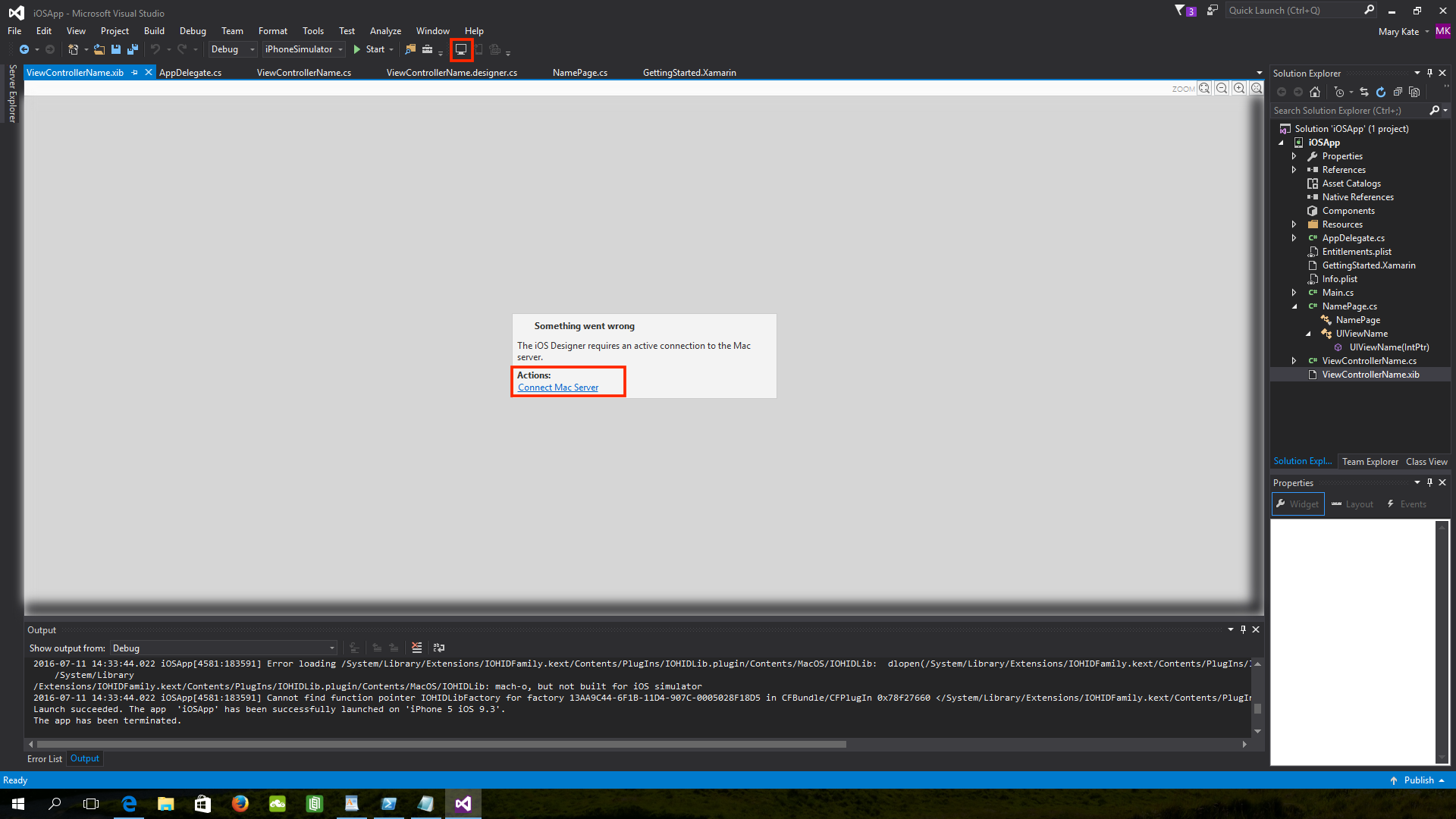
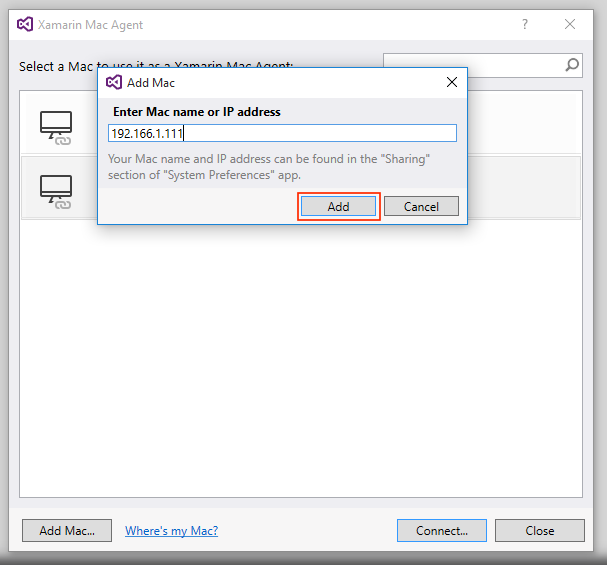
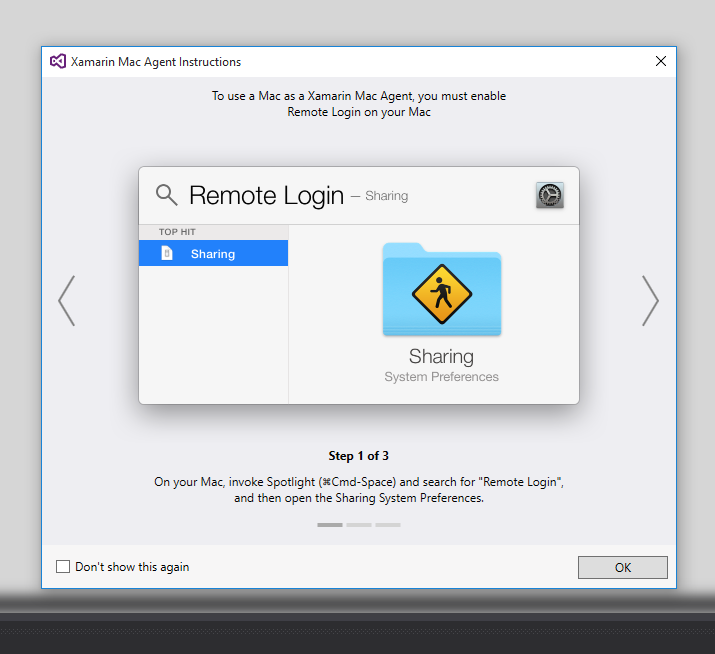
Window = new UIWindow(UIScreen.MainScreen.Bounds);

Window.RootViewController = new ViewControllerName();

Window.MakeKeyAndVisible();

return true;

}

1. Double click on ViewControllerName.xib to open it the iOS Designer. Here you will connect to a mac or mac server. You will see the following screens.
2. Create a custom page in the Xamarin.Forms core project and set it to be the app’s main page. Create a class inherited from ContentPage and call it NamePage.

using System;

using UIKit;

namespace Mobile3\_Lab2\_iOS

{

public partial class UIViewName : UIView

{

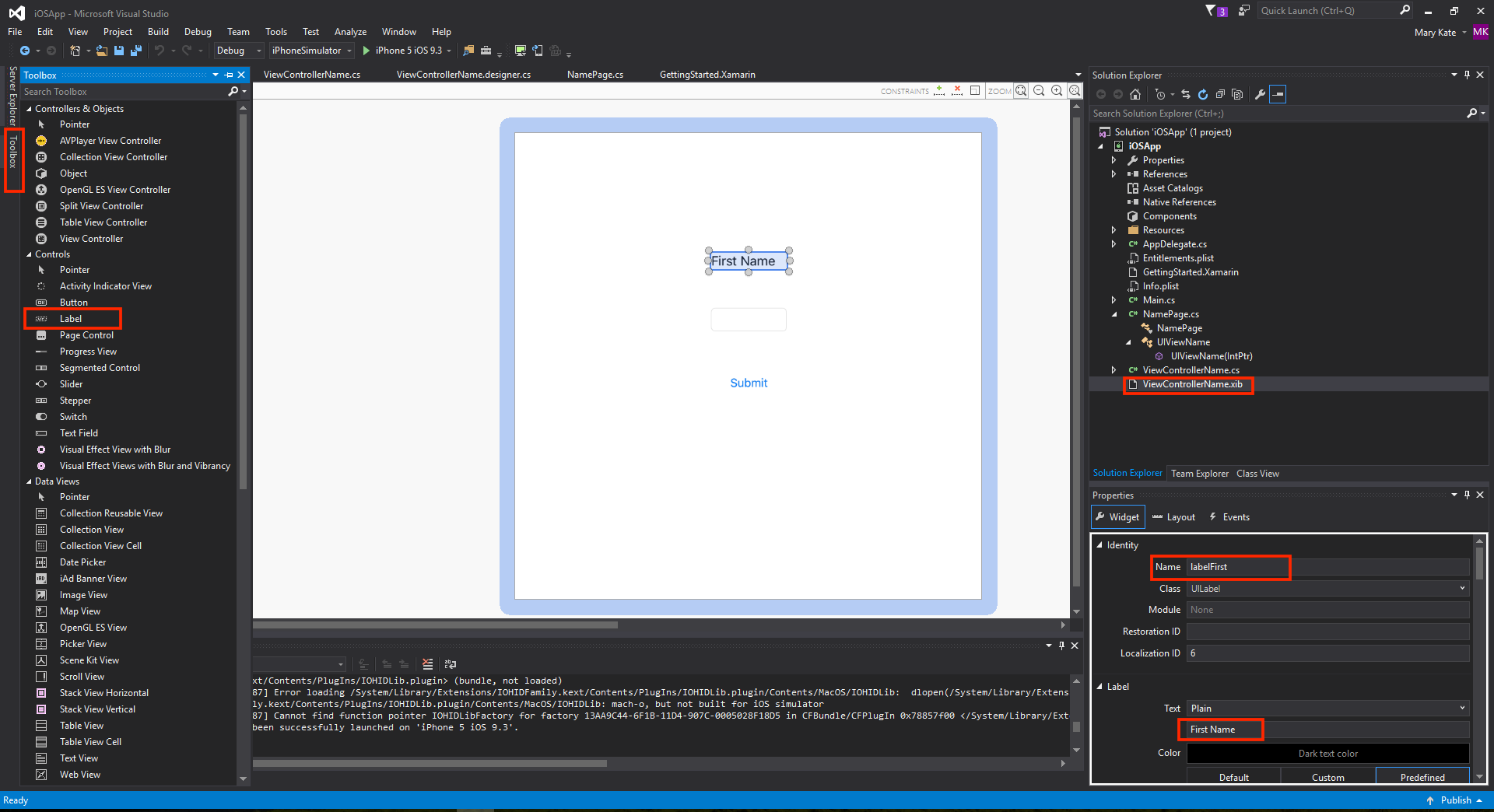
public UIViewName (IntPtr handle) : base (handle)

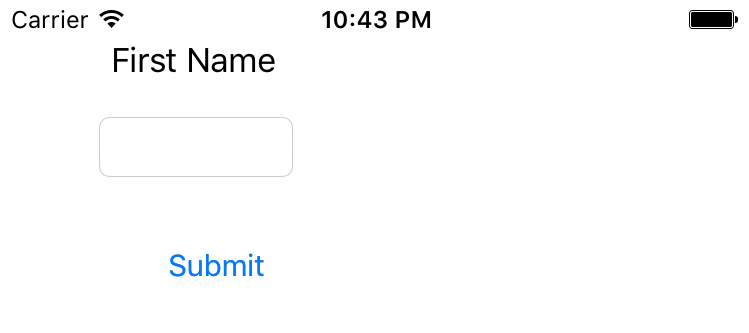
{

}

}

}

1. Add views to the ViewName.xib using the iOS Designer by dragging the view icons from the Toolbox on the left onto the canvas. Configure views by clicking on each control and editing their properties in the Properties box in the lower right of the IDE.
   * Add a Label called labelFirst with a with a text value of “First Name”. Expand the width of the label to accommodate the text by dragging on a control handle.
   * Add a Text Field called firstName with the text empty.
   * Add a Button called submitName with Text of “Submit”.

The views look like this on the page and tapping the entry field will invoke a keyboard.

1. Make the button click event show the firstName value in a popup.

Handle the button’s Click event. In the ViewControllerName.xib, double click on the submitName button to create click event-handling code in the view controller.

partial void SubmitName\_TouchUpInside(UIButton sender)

{

throw new NotImplementedException();

}

1. Inside that handler, invoke a popup to display the entered name.

partial void SubmiName\_TouchUpInside(UIButton sender)

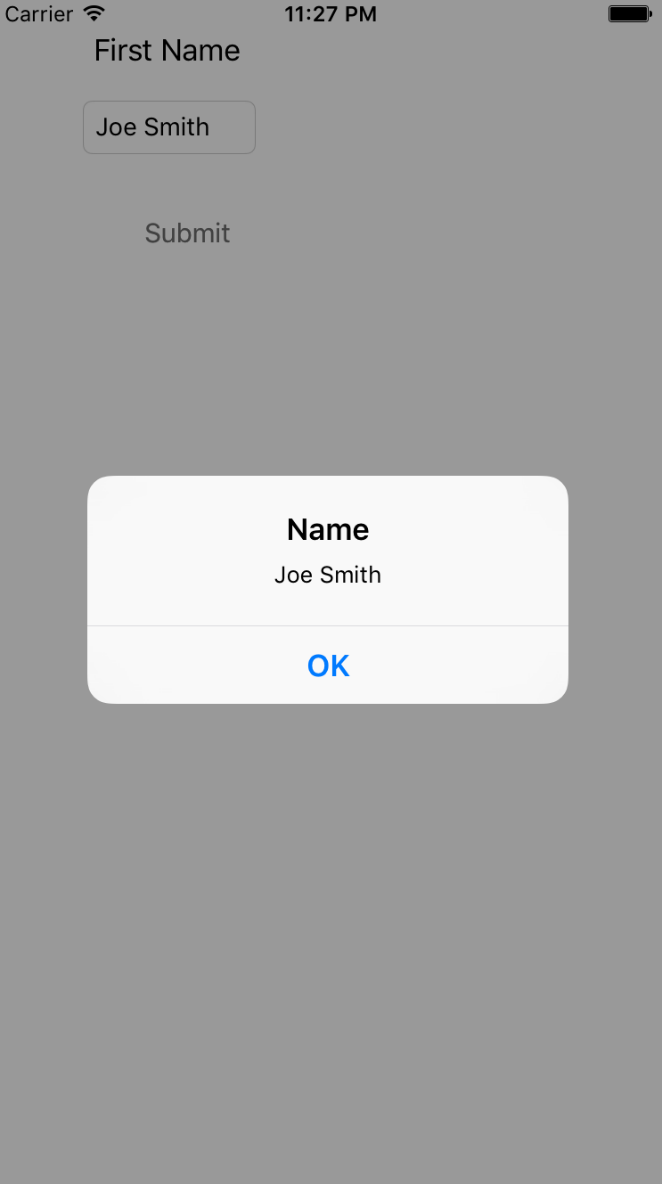
{

var alert = UIAlertController.Create("Name", firstName.Text ,UIAlertControllerStyle.Alert);

alert.AddAction(UIAlertAction.Create("OK", UIAlertActionStyle.Cancel, alertAction => { }));

this.PresentViewController(alert, true, null);

}

Entering a name and tapping the submit button raises the alert: