Lab 02: Tab Navigation

Prerequisites

Remember that if you are using Windows, you will still need an accompanying Mac on your network with XCode and the Xamarin tools to build and run the code.

See the Xamarin.iOS setup documentation if you need help getting your iOS environment setup:

http://docs.xamarin.com/guides/ios/getting_started/installation/

Downloads

There are several projects in a single solution that we will use to explore the various mobile navigation patterns on iOS, which are included with this exercise and can be found at:

Content/Exercises/NavigationiOS/Lab Resources

Lab Goals

The goal of this lab will be to get an understanding of how iOS and Android handle the tab navigation pattern. By completing this lab, you will learn about:

- The iOS and Android variants of tab navigation – how they are constructed.
- How tab selection is handled on iOS or Android (or both).

Steps

Open the Starting Solution

1. Launch Xamarin Studio and open MobileNavigationPatterns solution file included in your lab resources.

iOS Tab Navigation

Creating the Tab UI

1. Within the iOSTabs project, open AppDelegate.cs.
2. To present a tab navigation system on iOS, you can use the provided UITabBarController class. We create one of those for a class-level field in `FinishedLaunching`. and set it to be the window's `RootViewController`.

```
EvolveTabBarController evolveTabController;

public override bool FinishedLaunching (UIApplication app, NSDictionar
y options)
```

```
{
    // ...
    evolveTabController = new EvolveTabBarController ();
    // ...
    window.RootViewController = evolveTabController;
}
```

3. Tab bar controllers need to be provided a list of view controllers to present as tabs. In this case, the code to do so is isolated in a subclass of `UITabBarController`. Open Navigation/EvolveTabBarController.cs.
4. Inside `ViewDidLoad`, we create the controllers that will make up the tab choices. For each controller, we set up its `TabBarItem` to customize the tab's appearance. In this case, we set the title and give the tab an icon to use.

```
var vc1 = new UINavigationController (new SessionsViewController ());
vc1.TabBarItem = new UITabBarItem ("Sessions", UIImage.FromBundle ("im
ages/tabsession"), 0);
```

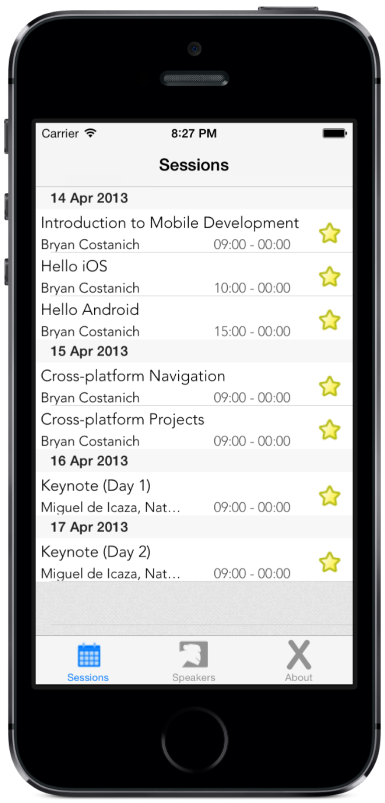
5. With all of the controllers created, we hand them over the tab bar controller's ViewControllers property.

```
ViewControllers = new UIViewController[] { vc1, vc2, vc3 };
```

6. Optionally, you can set the selected tab. It will default to the first tab.

```
SelectedIndex = 0;
```

7. Run the application in the simulator and switch between tabs. In iOS, this switching is handled for us by the `UITabBarController`.



Sessions

14 Apr 2013

Introduction to Mobile Development
Bryan Costanich 09:00 - 00:00

Hello iOS
Bryan Costanich 10:00 - 00:00

Hello Android
Bryan Costanich 15:00 - 00:00

15 Apr 2013

Cross-platform Navigation
Bryan Costanich 09:00 - 00:00

Cross-platform Projects
Bryan Costanich 09:00 - 00:00

16 Apr 2013

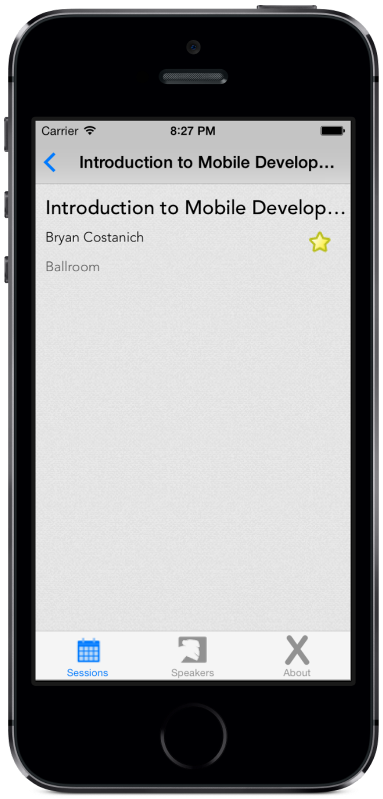
Keynote (Day 1)
Miguel de Icaza, Nat… 09:00 - 00:00

17 Apr 2013

Keynote (Day 2)
Miguel de Icaza, Nat… 09:00 - 00:00

Stack Within Separate Tabs

8. With the application running, click to the Sessions tab and then navigate to any session details.

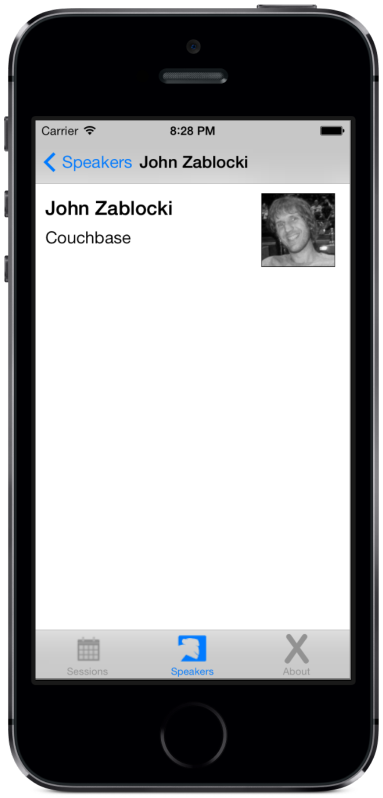


Introduction to Mobile Develop…

Bryan Costanich

Ballroom

9. Now click to the Speakers tab and navigate to any speaker details.



John Zablocki

Couchbase

10. Flip back to the Sessions tab and notice iOS has maintained the user back stack for each tab independently for you, if this is what your app needs.

Summary

In this lab, we saw how iOS offers tab navigation solutions for applications. We reviewed how to create tab systems and saw how to handle tab switches by the user. As well, we saw how back stack is maintained (or not) within tabs.