Alyas *et al.LGU (IJECI)* 2018

LGU (IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

LGU International Journal for
Electronic Crime Investigation

Research Article

Vol. 2 issue 2 April-June 2018

Advanced User Interfaces from iOS To Windows 8

Syeda Binish Zahra[1], Syed M. Shabih-Ul-Hassan[2]

[1]Department of Computer Science, Lahore Garrison University, Lahore, Pakistan.

[2]Professional Academy of Commerce, Lahore, Pakistan
binishzahra@lgu.edu.pk[1] shabih@pac.edu.pk[2]

Abstract:

This article discusses porting advanced user interface features from an iOS app to a Windows Store app. We use an electronic medical record (EMR) application for this case study. This paper can give new direction for windows 8 to windows 10 porting.

Keywords: Hypertext Markup Language (HTML), integrated development environment (IDE), User-Interface (UI).

1. Introduction

In recent years, tablets, as new forms of mobile computing platforms, have quickly moved from the consumer electronics spaces into the business and enterprise computing domains. After the release of Windows 8 operating systems earlier this year, we felt there was a need to provide some quick start tutorials for developers on how to port their existing apps from other platforms such as iOS to Windows 8, and start developing new apps on Intel UltrabookTM devices, tablets, and other Intel architecture-based devices running Windows 8. This article serves this objective and focuses on the advanced user interface topics.

On iOS, natively Objective-C is the main development language. For Windows Store apps, you have multiple choices available, including Visual C#, HTML / JavaScript*, and others. In this case study, we use Visual C#* as the development language of choice.

2. From Xcode* 4 to Visual Studio* 2012

Like the Xcode tools package on OS X* for iOS application developers, Visual Studio 2012 provides an integrated development environment (IDE) for Windows Store app developers [2]. Also like the Interface Building design tool on Xcode 4, which supports storyboarding (Figure 1), Visual Studio 2012 includes a XAML Designer tool (Figure 2).

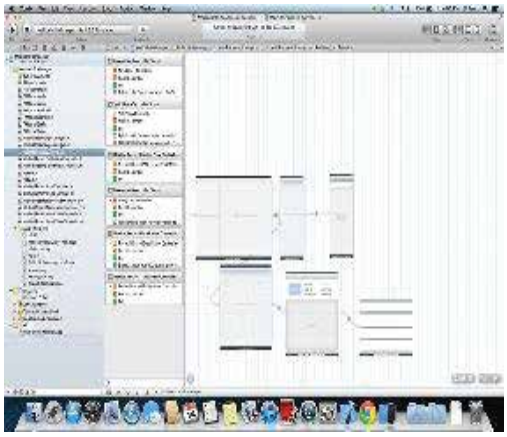

Figure.1: The Interface Builder in Xcode 4

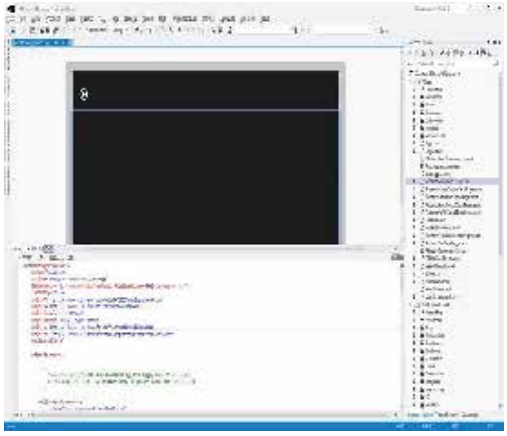

Figure.2: Visual Studio 2012 XAML Designer

3. The Case Study Project

This article is one of a series based on a case study project. In the project (link to the folder of other articles based on this project), we ported a medical record management application from iOS to Windows 8. The basic requirements of the application include:

- Show a list of patients
- Show the personal and medical information of a specific patient, which includes identity, billing, vitals, lab tests, medical images, etc.
- Display detailed graphs and images when selected

This article will cover the advanced UI features of the project.

4. High Level UI Design and Navigation Patterns

On iOS, we can use the split-view controller to present a master view and a detailed view on the screen. We can use table views or tab bar views to group different categories of information on the view. Figure 3 shows the split view along with the master table view and the detailed table view [3]. The left pane of the split view shows the scrollable patient list. The right pane shows the medical records associated with the selected patient in the list. We use a table view to put the medical record categories on the same view. We can also use the tab bar view, with each tab view displaying a specific medical record category. Figure 4 shows how this view is created in Xcode 4 storyboard.

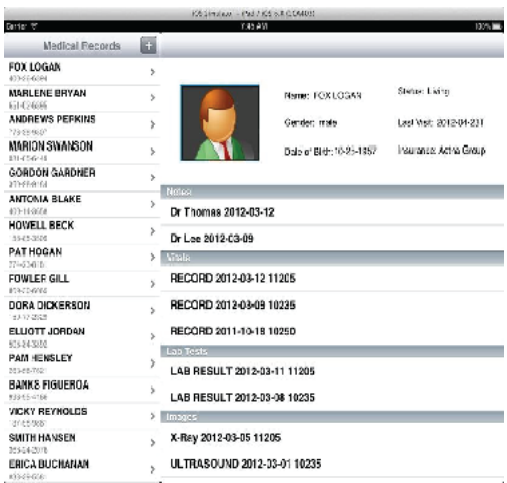

Figure.3: On Ios, A Split View Controller And Its Master Table View And Detailed Table View

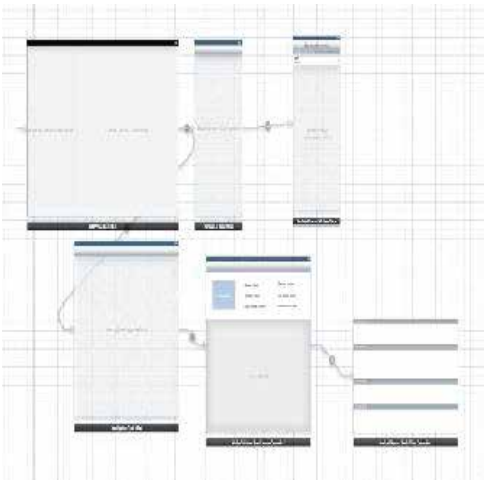

Figure.4: Use Xcode storyboard to create a split view and its master and detailed table views

In a Windows Store app, we can accommodate this design by following the Windows Store app hierarchical system of navigation pattern (Figure 5). The first level page shows a grid view that includes a tile for each patient (Figure 6). The second level page is a grouped item page that includes the medical records for the patient selected from the first level page (Figure 7). The third level page is a group detail page that shows the specific category of medical records selected from the second level page (Figure 8). We can also have a fourth level page that shows the item details, for example, the actual X-ray image selected from the third level page.

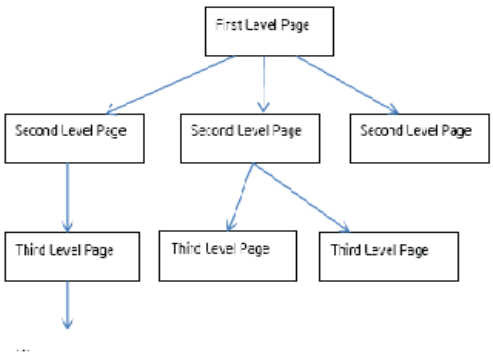

Figure.5: Windows Store app hierarchical system of navigation.

Figure.6: In the Windows Store app, the root level grid view includes tiles for the patient list.

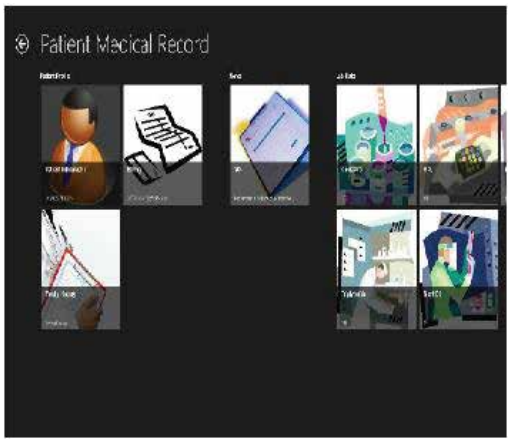

Figure.7: In the Windows Store app, the second level page shows the medical records associated with the selected patient.

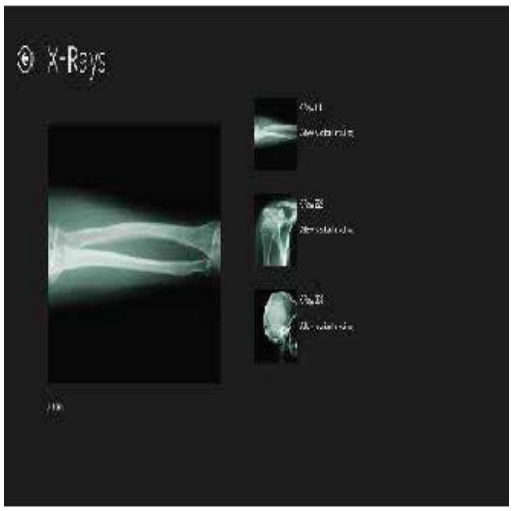

Figure.8: In the Windows Store app, the third level page shows the group selected from the second level page.

Visual Studio 2012 projects, the UI page is defined with a "XAML" file and a C #implementation file (.cs) associated with it. Because the transitions from one page to another page usually originate from user actions, for example, when a grid view item is pressed, naturally the event listeners are the places used to handle the navigations.

5. Windows Project Templates and Data Binding

In Figure 7 and Figure 8, items are grouped and shown nicely in the grid views. Visual Studio 2012 Windows Store project templates provide a powerful basis to construct these user interface pages. The predefined project templates include grouped items page, group detail page, item detail page, etc. We use the X-rays group detail page as an example here.

In Visual Studio 2012's Solution Explorer window, right click the project name and select "Add -> New Item…" from the pop-up menu. Select "Visual C#" on the left pane. On the center pane, we see the list of the predefined page templates. Among those templates, we select "Group Item Page." A preview of the template is shown on the right pane. We also enter a name for the page in the text box at the bottom of the dialog (Figure 9), and press the "Add" button. Visual Studio 2012 now generates a file named "XRayImagesGroup DetailPage.xaml" file and a file named "XRayImagesGroupDetailPage.xaml.cs" in the project.

Figure.9: Add New Item dialog shows the Window Store project templates (**)

In Visual Studio 2012, if we expand the "Common" folder generated under the project (Figure 10), we can see Visual Studio has generated a group of files under it. Among these files, LayoutAwarePage.cs contains the class that we derive the XRay Images GroupDetail Page from.

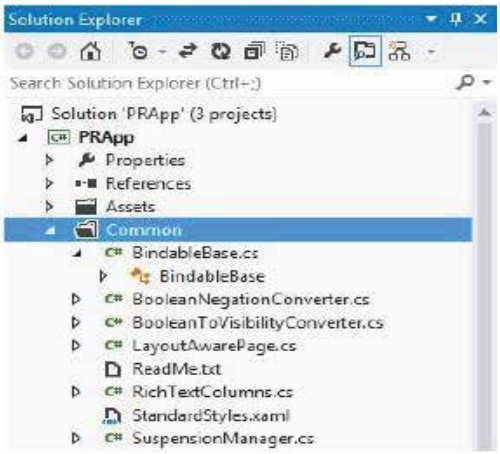

Figure.10: The "Common" folder in the project.

The “Common” folder also includes the “BindableBase.cs” file. We derive a data model for the view from this class.

6. Conclusion

Advanced user interface with IOS are more helpful and secure when we compare its functionality and performance with windows OS. The case study of electronic medical record (EMR)is showing the reliability of advanced user interface we add or modify some features of IOS ,then this operating system will become more reliable than any operating system that are used in market.

7. References

1. Heuristic evaluation- Nielsen, Jakob. Useit. How to conduct a heuristic evaluation. http://www.useit.com/papers/heuristic/heuristic_evaluation.html (Retrieved 2011-05-19)

2. Kumar, Minu, and Garg, Nitika. Aesthetic principles and cognitive emotion appraisals: How much of the beauty lies in the eye of the beholder? Journal of Consumer Psychology. Vol. 20, No. 4. (October 2010) E-Journals, EBSCOhost (Retrieved 2011-05-18)

3. Kurki, Ilmari, and Repokari, Lauri, and Saarela, Toni. Visual search on a mobile phone display. SAICSIT '02. (2002):253-253.