iPhone Application Development using Xcode

Pavithra Ramakrishnan
Computer Science & Information Systems Department
Bradley University
Peoria, Illinois, U.S.A.
1-309-677-2460
jiangbo@bradley.edu

Jiang B. Liu
Computer Science & Information Systems Department
Bradley University
Peoria, Illinois, U.S.A.
1-309-677-2386
jiangbo@bradley.edu

ABSTRACT
In this research, an iPhone application has been developed using the latest IOS 5 operating system with the new feature storyboards, object-c, JSON, and Hpple parser using Xcode. The app will launch a splash screen for a web site where a list of icons will take the user to different screens. Some added features on the app are Map view, link to Sakai and feedback icons. The rich user interface in this app makes it more interesting with different gestures like swipe, tap and pinch. This app will be of a great use to the University students with the smart phones. It provides users easy access to the web site and the online materials who can view them anywhere with a handhold device.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**]: Miscellaneous.

General Terms
Design, Experimentation.

Keywords
Service Orientation, iPhone Application, Web Access Framework.

1. INTRODUCTION
There has been an exponential growth in the use of handheld electronic devices and the web developers are finding a plethora of opportunities in delivering the web content to these devices. The trend suggests that the cellular phones, PDA's, digital cameras, camcorders, laptops and other personal utility devices are converging into fewer sets of devices like smart phones and tablets. There are both advantages and disadvantages with the emergence of these devices and the pros clearly outnumber the cons [1-2]. These new devices are often equipped with touch interfaces, voice recognition, gyroscopes, compass, GPS, sound and visual devices. This opens up immense opportunities for web developers to design creative applications that can use these features and help the user have a personal experience of the web. It is also imperative that the web designers have to invent newer ways to tailor the web to make the information more usable, accessible and accurate for the user keeping in mind the privacy and security concerns.

iPhone is one of the most popular smart phones used all over the world. The iPhone appstore has thousands of apps in various categories. Users have shown their interests in purchasing these apps, they prefer to view the web info in the form of apps rather than native websites. The reason is the rich user interface provided by these apps. iPhone apps are developed using Xcode which is Apple's development environment [3-9].

This research project is about learning how iPhone applications are programmed and developing innovative ideas in the application, which will be useful for the users.

2. DEVELOPMENT METHODS
In this project, an application has been developed for a professor from computer science department. It was developed for the latest IOS 5 operating system using the latest feature called storyboards. The programming language used to create the app is Objective–C. A splash screen is launched at the beginning of the app, and then a grid view of icons has been displayed, where each icon takes the user to different views when touched. The courses handled in every semester, research works, and Services are few icons represented as in the website. Three added features to the app are Map view, link to Sakai and a feedback icon to get feedback from students. The Map view displays the location of Bradley University and the professor's office in specific. Sakai icon provides students a direct link to Sakai where they can login and access the class notes and calendar. Finally, the feedback icon lets students type their feedback and submit it to the professor. The courses icon consists of general content from the website in the detail view and a HTML Parser code was written to extract specific text from the website and dynamically display it in the application using table view. The parser used was Hpple downloaded from an open source project, which is a very good HTML parser. Then the code was written in Objective–C to parse the content from the website using Hpple. Hpple is a nice and open source Objective-C wrapper on the Xpath query library for parsing HTML content. Hpple allows easy searching by Xpath and allows easy access to tags, divs and attributes.

Out of all the parsers HTML parser is the most difficult one which requires developers to write their own code according to the website they extract from. When HTML websites are updated according to the latest trend the structure changes and hence the parsing code for that app has to be changed otherwise it will provide wrong or no results to the users. So, the website will be redesigned using JSON in order to write a web service API. This enables an easier way to extract the required content from the website and it is a very reliable method which adapts according to updates. JSON is a lightweight data-interchange format. Like XML, it is human-readable, platform independent, and enjoys a wide availability of implementations. JSON is a subset of the object literal notation of JavaScript. Data represented in JSON can

be parsed by JavaScript easily, so it is ideal for AJAX based web applications. If a Web Service returns XML data, then the data to be passed from the server to the client is larger than what it would be in the case of JSON. So, using JSON, we only need to pass fewer amounts of data. Also, XML parsing on client side is cumbersome and costly in the perspective of processing. Certain changes will be made in the web service to make it JSON enabled, a JSON serializer will be enabled in the web service config file. All the objects are made serializable and then the web service is called using JQuery. Video lectures from the class, news feeds from Bradley web site, Chat applications will be embedded in the app.

With the latest updates of IOS, the features of its development environment are also becoming more users friendly and with more interesting and rich user interfaces. The LLMV compiler points out the mistakes in code then and there and also corrects the mistakes for developers. Storyboards in Xcode are graphic illustrations of views and images which serves the purpose of pre-visualizing a motion picture or animation. The controls, transitions and triggers from every view can be done manually instead of doing it programmatically. The libraries, frameworks and objects required for our app are all provided in Xcode framework itself. It's even easier to connect to a table view and customize a UITableview using storyboards.

3. DEVELOPING THE IPHONE APP

We used a Mac computer and Xcode framework to develop IOS applications. First, open Xcode and choose create a new project and choose storyboard option (Fig. 1). Once a new project is created, Xcode will create the classes, storyboard and necessary frameworks for the project.
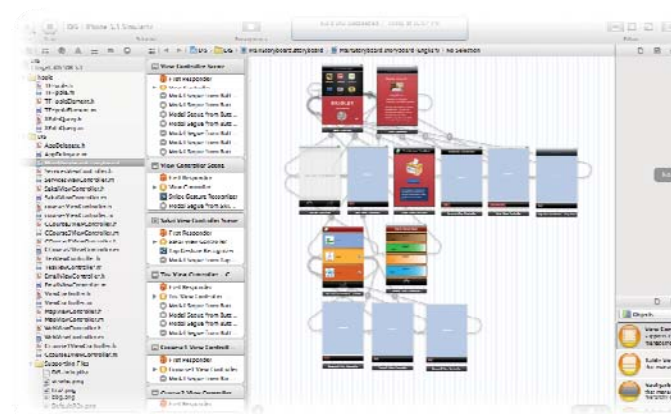

Figure 1. Xcode storyboard.

Storyboard is the latest feature provided by apple in Xcode. It shows the entire scene of the project in one single view. Users can directly navigate from one view to another via storyboards. This is the first view of the project. It has a grid view of icons and each icon lets the users to navigate to the detail view (Figure 2 and 3). The info icon at the bottom provides information about the application.

The info page for the app has a swipe gesture to return back to the main view. The swipe gesture is available in the object library.

The Tab bar view is shown in Figure 4, where the user can click the teachings icon in the main view to redirect it to the tab bar

view, which has the recent courses and current courses. It has a back button to return back to the main page.
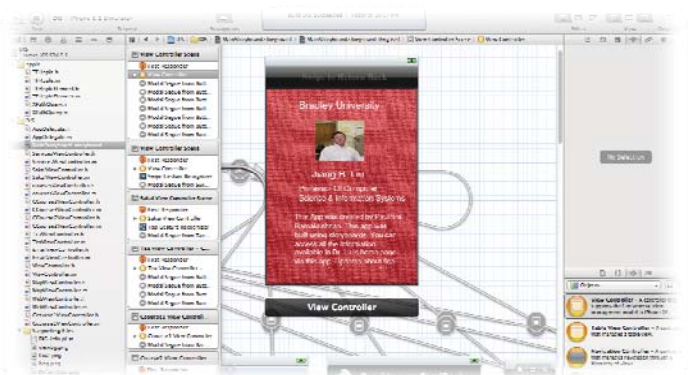

Figure 2. Top-level page of the Application.

Figure 3. Second-level page of the Application.

Figure 4. Xcode Tab bar view.

Figure 5 illustrated the current courses view controller. It will display the list of courses taken in the current semester and the disclosure button navigates to the detail view about the course. It has a back button to return back to the main page.

The header file imports the hpple parser and declares a set of labels to display the course names from the web page. The hpple parser is a HTML parser, which acts as a search tree and the xpath query extracts the required text from the website and displays it.

In the implementation file (Figure 6), the variables are declared as strings in the ViewDidLoad method and using NSDATA method the URL from which data has to be extracted is specified. The

Xpath syntax is used to specify which data needs to be extracted. The label is appended to the string declared and the data from the string will be displayed via the label.
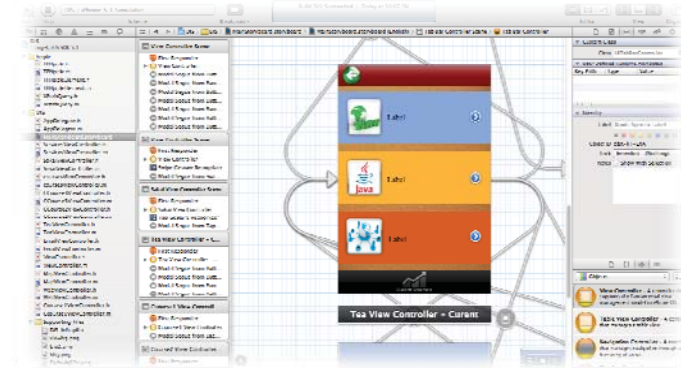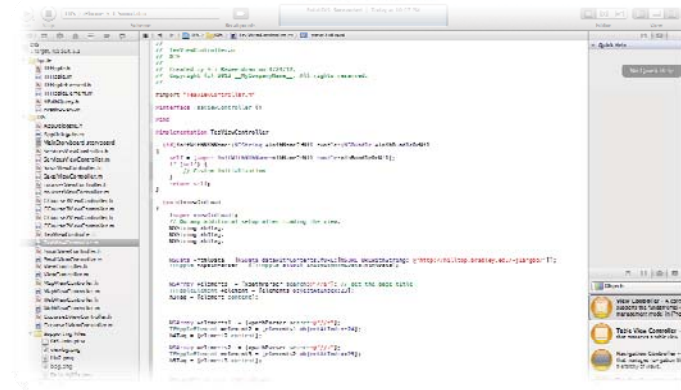

Figure 5. Xcode view controller.

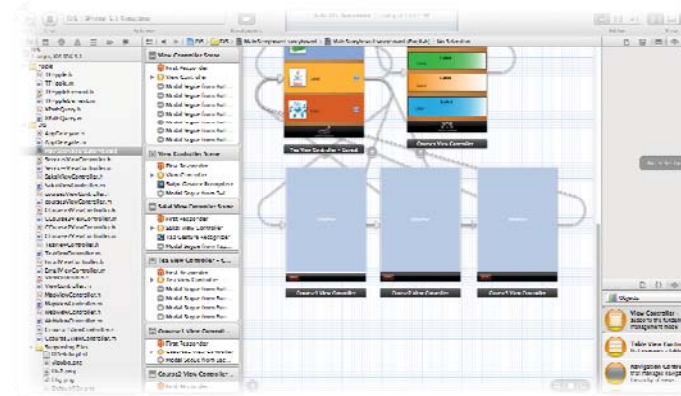**Figure 6. Xcode Object-C Implementation.**

Figure 7. Xcode Courses detailed view.

The detail view of each of the current courses displayed and the information regarding assignments, final exam, and other course details are provided in Figure 7.

In the header file of each detail view a property of the string is declared. In the implementation file (Figure 8) it is important to synthesize the property declared otherwise a warning will be issued on building the project. The required URL is specified under the ViewDidLoad method and using NSURL Request method the URL is loaded in the detail view.

The recent courses view (Figure 9) displays a set of recent courses from the website. It again uses the Hpple parser to extract the required text. It has a tap gesture to return back to the main menu.
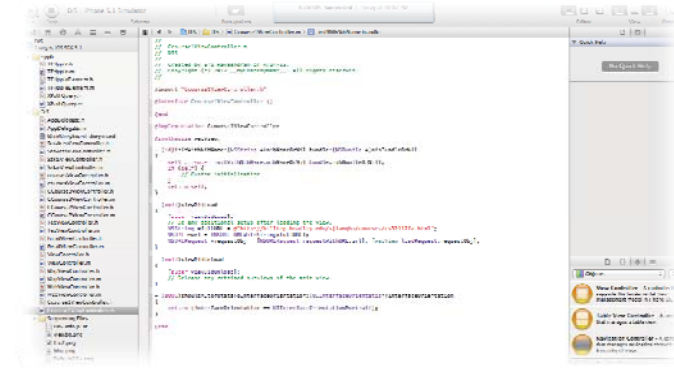

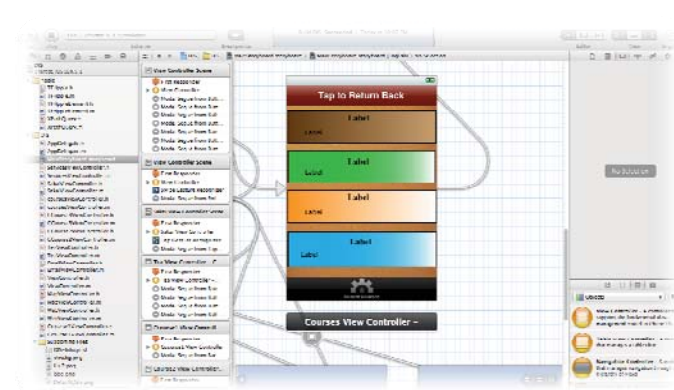**Figure 8. Xcode implementation of a detail view.**

Figure 9. Xcode recent courses view.

When the user clicks on the research icon in the research view controller (Figure 10), it navigates to this view controller. It displays about the research activities to the user.
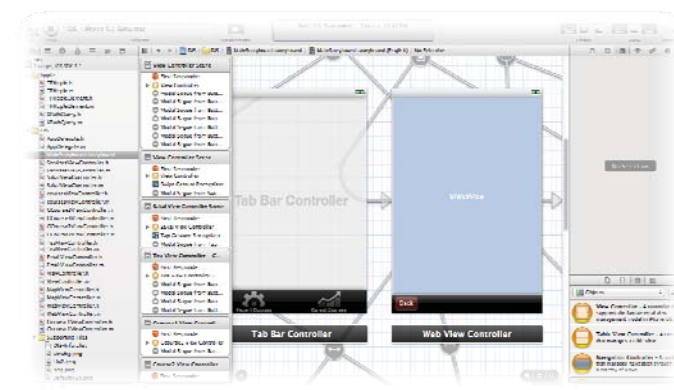

Figure 10. Xcode view controller.

The feedback view controller is shown in Figure 11. When the user clicks on submit feedback it will automatically connect to the email page and the user can enter their email address, type the feedback and send it to the instructor.
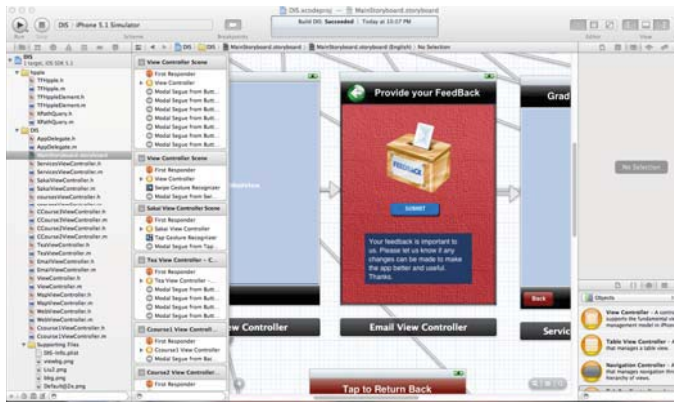
Figure 11. Xcode feedback view controller.

In the header file a mailcomposerviewdelegate method is set and the IBAction is set to the submit button. In the implementation file (Figure 12) the mail composer methods are set and an array is declared for recipient address, cc, bcc and email body. The message delivery and failure notifications are also set and to automatically save the message to draft folder.
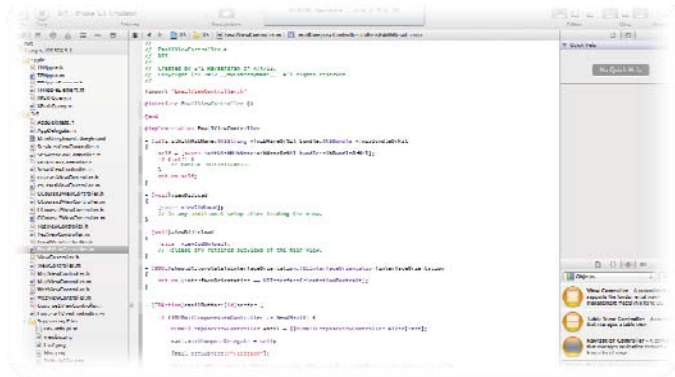

Figure 12. Xcode implementation of a feedback view controller.

The services view controller displays the services provided by the instructor to the Department (Figure 13). Using the web view property the required URL is specified and loaded.
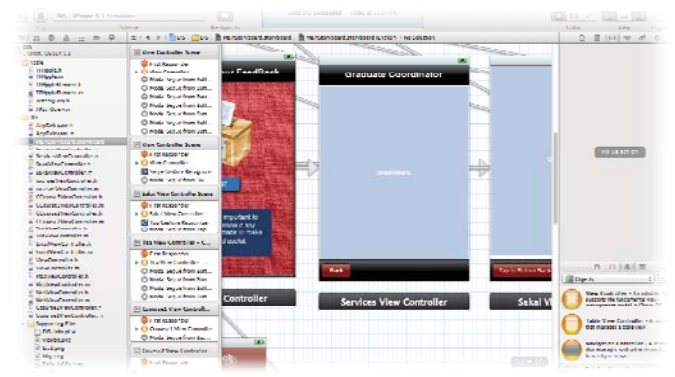

Figure 13. Xcode service view controller.

The Sakai View Controller displays the class notes and list of information in a table view (Figure 14). It displays the website in the mobile view format. The users can login and check their notes, events etc.
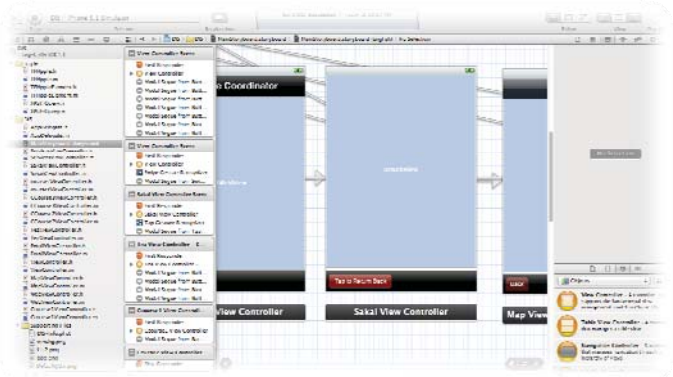

Figure 14. Xcode Sakai view controller.

The Map View Controller (Figure 15) provides the Google map view. An annotation is placed on the office address of the instructor where users can zoom and view the address.
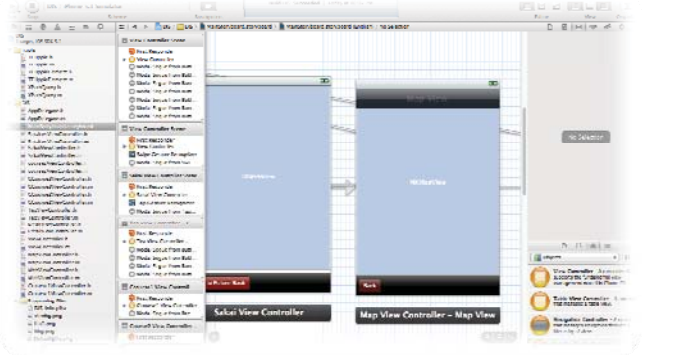

Figure 15. Xcode map view controller.

A map kit framework has to be included first. In the header file location co-ordinates are added and IBAction and IBOutlet are created for map view and to show the address. Address annotation is also declared.

In the implementation file (Figure 16) a title and subtitle has to be set to display the name and address of the instructor. In the IBAction the latitude and longitude values are set to display the exact address. Annotations with titles and subtitles are added to place the pin on the exact location.
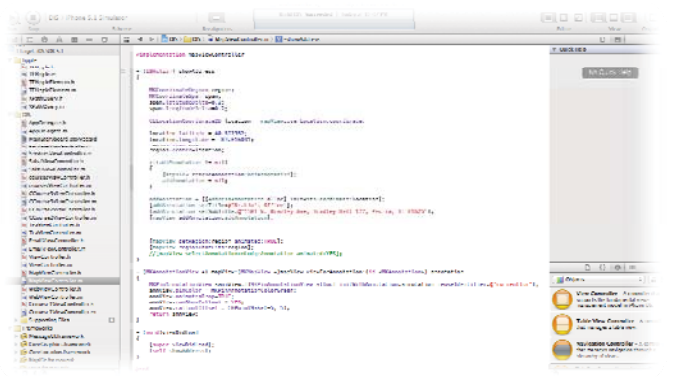

Figure 16. Xcode implementation of the map view controller.

This completes the code for the project. The Figure 17 shows the screen shots after building and running the project.
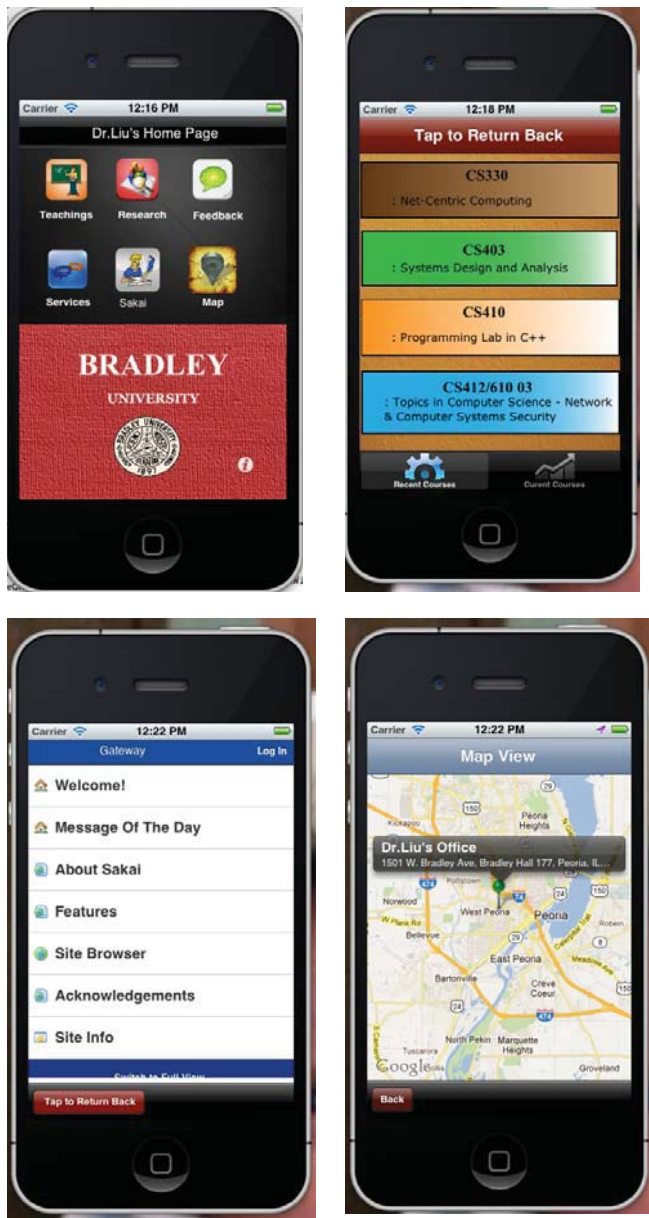

Figure 17. Xcode iPhone app screenshots.

The courses, map view, feedback, research, services and Sakai icons have been implemented in the app currently. The rest of the features like video lectures, chat application, web service creation and news feeds are still in progress and some changes with the

design has to be made. So far the features implemented works fine with the app. All the issues and errors in the code have been solved. The HTML parser used works just fine and extracts the required text without any issues. Any bug fixes and issues to this app in the future can be easily updated.

4. CONCLUSION

Almost every university has its own apps in online education. iPhone users are increasing day by day and the usefulness of an app has also increased. This app will be of a great use to the students with the iPhone. It provides students easy access to the courses and the class materials. Even if a student misses his/her class he/she can makeup everything with the help of video lectures and clear doubts then and there using chat application. The students can view assignments, notes, presentations via the app easily. They don't have to enter the web url and wait for it to load every time. The rich user interface in this app makes it more interesting with different gestures like swipe, tap and pinch. The map view helps them locate the school and their departments more easily. Overall this app will serve a good purpose to students.

5. REFERENCES

[1]   Neil Smyth, iPhone iOS 5 Development Essentials, CreateSpace independent publishing, 2012.

[2]   Jiang B. Liu, Nithya Vasudevan, and Kumar Macherla, "Develop Web Services for Reliable Cross-Platform Web Applications," Proceedings of IEEE 10th International Conference on Quality Software, pp 429-434, Zhangjiajie, China, July 2010.

[3]   Apple Developer, https://developer.apple.com/

[4]   Xcode, http://www.xcode-tutorials.com/

[5]   Xcode Tutorials, http://www.geekylemon.com/xcodetutorials.htm

[6]   Xcode Snippets, http://www.icodeblog.com/2011/12/06/using-xcode-4-snippets/

[7]   The Appcode, http://www.theappcodeblog.com/

[8]   Xcode Guide, http://maniacdev.com/xcode-4-tutorial-and-guide/

[9]   Object-C Essential, http://www.lynda.com/Xcode-4-tutorials/Objective-C-Essential-Training/77860-2.html.