WMU Mobile for iPhone

Christopher A. Ashbay
*Western Michigan University*, christopher.a.ashbay@gmail.com

Justuce Reule
*Western Michigan University*

Tim Wickey
*Western Michigan University*

Follow this and additional works at: https://scholarworks.wmich.edu/honors_theses

Part of the Computer Sciences Commons

**Recommended Citation**

Ashbay, Christopher A.; Reule, Justuce; and Wickey, Tim, "WMU Mobile for iPhone" (2011). *Honors Theses*. 63.

https://scholarworks.wmich.edu/honors_theses/63
WMU Mobile for iPhone

Chris Ashbay
Justus Reule
Tim Wickey

Senior Design 4910
Instructor: Dr. Kapenga
April 28, 2011
Abstract

Western Michigan University does not currently have a smart phone application. *WMU Mobile for iPhone* was created for WMU students as a mobile source for campus news, weather information, and athletics highlights. It also provides connections to various sources of WMU media, and allows users to access GPS maps of campus buildings. The app is designed to allow for future releases which may include allowing the user to view campus events by department and search campus maps. The *WMU Mobile for iPhone* is accessible on the iPhone, iPod Touch, and iPad, and is available for free through the Apple App Store.
# Table of Contents

1. Introduction............................................................................................................... 4  
2. Background............................................................................................................... 6  
   2.1 History of the Mobile app.............................................................................. 6  
   2.2 Other Universities....................................................................................... 7  
   2.3 The Client...................................................................................................... 8  
3. Design Decisions..................................................................................................... 11  
4. Design..................................................................................................................... 13  
5. Implementation........................................................................................................ 14  
6. Testing..................................................................................................................... 25  
7. Summary................................................................................................................ 26  
8. Appendices............................................................................................................. 27  
   8.1 Appendix A: Spikes...................................................................................... 27  
   8.2 Appendix B: Stories...................................................................................... 30  
   8.3 Appendix C: NDA and Project Ownership.............................................. 32  
   8.4 Appendix D: Resources.............................................................................. 33  
   8.5 Appendix E: Programming Standards.................................................. 34  
   8.6 Appendix F: Version Control Guide......................................................... 35  
   8.7 Appendix G: Unit Testing........................................................................... 39  
   8.8 Appendix H: The Release Process......................................................... 42  
9. References............................................................................................................... 43  
10. Glossary................................................................................................................ 44
1. Introduction

This paper introduces and describes *WMU Mobile for iPhone*. This application is intended as a resource for Western Michigan University, and as a public showcase of its technological capabilities. The Background, Design Decisions, Design, Implementation, Testing, and Summary sections will further detail the project. Additional details including Spikes, Stories, Ownership, Resources, Standards, Version Control, Unit Testing, and the Release Process are included in the respective Appendices.

The Background section introduces the client and explains which functions and features that the client requested for the *WMU Mobile for iPhone* app. It examines the apps of other universities. Later, a condensed history of smart phones - specifically the Apple iPhone and apps - will be summarized.

The Design Decisions and Design segments explain the design of the *WMU Mobile for iPhone* application. In the Design Decisions section, some major design choices are explained, including the choice of development platform and programming language. In the Design section, the overall application design structure and architecture are discussed.

The Implementation section explains the execution of the *WMU Mobile for iPhone* app and demonstrates its functionality. It also has a series of screenshots showing what the finished application does.

The Testing section covers the types of testing used for the development of *WMU Mobile for iPhone*. These include unit testing and system testing. Unit testing is the practice of testing small “units” of code to ensure all the individual pieces work the way they are expected to work. System testing is testing the program as a whole. Thorough testing provides reasonable confidence that the app will function properly when deployed.

The Appendices include Spikes, Stories, Resources, and also Programming Standards, Version Control, and Unit Testing. Spikes are specific examples of experiments and research that were performed while investigating whether it was feasible to develop the *WMU Mobile for iPhone* app. Stories are a list of functional behaviors that are agreed upon with the client. Resources are a description of the materials, equipment, and skills that were required to complete the *WMU Mobile for iPhone* app. This includes a list of any hardware and software necessary for designing an iPhone application.
WMU Mobile for iPhone is a complex application that required much debate and planning to build. The rest of this paper discusses in detail the process from start to finish.
2. Background

This section briefly discusses the history of Apple Inc's iPhone, and the current market for iPhones and apps. It then examines examples of iPhone apps of from several other universities. Finally, it introduces the client, along with the features and functions that they requested for the WMU Mobile for iPhone app.

2.1 History of the Mobile App

The mobile app has a particularly short history. The mobile phones of the recent past were expensive and inconvenient behemoths, with only one single dedicated purpose. Today, smart phones are capable of much more than simply making and receiving phone calls. Smart phones run applications for entertainment, games, news, sports, weather, social networking, travel, email, Internet browsers, and a variety of creative, productivity and business tools. Instead of being simple communication devices, smart phones have become fast, generic computing devices, with full fledged operating systems. [*1]

The Apple iPhone also has a very short and dynamic history. The first Apple iPhone was released in 2007. It ran iOS 1, the first release of the iPhone operating system. The screens were scratch resistant LCD screens that measured 480 by 320 pixels. First generation iPhones had Samsung 32 bit, 620 MHz RISC ARM processors. The amount of RAM in the first and second generation iPhones was 128 Mb. The tiny camera that was placed in the 1st generation iPhones was a 2 mega-pixel camera.

The current iPhone runs iOS 4, and has a 960 by 640 pixel resolution, and have fingerprint resistant screens with higher resolutions. It has a ARM cortex A8 Apple A4 1 GHz processor, almost twice as fast as the original processor. The RAM in the fourth generation iPhones has grown to 512 Mb. The current iPhones have 5 mega-pixel cameras that can record high definition video at a rate of 30 frames per second. [*2]

The Apple iPhone evolved from a more traditional phone model to the current model which is based nearly completely on apps. Originally, the only available apps for the iPhone were the ones that shipped with the device and were created by Apple Inc. However, Apple envisioned a market where the creative power of countless independent developers could be harnessed to develop applications for its product. A year after the launch of the iPhone, Apple released the iPhone SDK, and allowed independent developers to build applications for the iPhone. As soon as one year later there were 15,000 apps available on Apple’s App Store, and 500,000,000 apps downloaded. (See Figure 2.1.1). Now there are over 100 million Apple devices in use.
The App Store boasts more than 300,000 apps and 5 billion downloads, and it is not yet 2 1/2 years old. [*3]

Figure 2.1.1

![Apple iTunes App Store Apps and Downloads](image)

The iPhone is not the only choice in smartphones. It is, however, the most widely used smartphone by a significant margin. In North America during the month of May 2010, 54% of all web requests to the AdMob network came from iOS and 33% came from Android. [*4] Blackberry’s RIM OS came in 3rd with 7%. (See Figure 2.1.2) If we only were to develop a mobile app for one device, the obvious choice would be the iPhone. Since a second senior design team is simultaneously developing a similar app for Android, we will have 87% of the market covered after release.
2.2 Other Universities

Some time was spent researching iPhone apps published by other universities. Apps from 15 other universities from around the country were downloaded. Particular attention was paid to the apps from schools in the surrounding area. The purpose was to get ideas for features and functions that could be implemented on the WMU Mobile for iPhone app. Also it was important to see the general state of the competition.

One local app was the Central Michigan University iPhone App. It features campus specific news, calendar events, and check-lists. One of the things that stand out on this app is the Rolodex style browse view that lets you choose your target by flipping through the choices. (See Figure 2.2.1) The app is designed to display information quickly and conveniently, including having campus weather right on the home screen. You can also see complete weather details by opening the weather feature.

The University of Michigan also has an iPhone App. (See Figure 2.2.2) Its Magic Bus feature displays real-time bus tracking which students can use to see exactly where each bus is on a Google style map. Students can also access daily lunch menus for on-
campus dining, as well as read news, view calendar events, and find people in the university directory.

2.3 The Client

The client is Dr. Keith Hearit, and Scott Puckett. Dr. Hearit is the Vice Provost of Strategic Enrollment Planning at Western Michigan University and Mr. Puckett is the Web Developer for Enrollment Management and the Provost Office. The users of the app will be Western Michigan University students who have an iPhone or iPod Touch. Until the spring semester of 2011, Western Michigan University did not have an application for any mobile devices.

The iPhone team met with Dr. Hearit and Mr. Puckett in the fall of 2011 to discuss the possibility of designing and creating a mobile application for Western Michigan University students. Dr. Hearit and Mr. Puckett were both enthusiastic about this project, and had a number of requests for content and functionality of the app. They
requested that the app provide content about athletic events such as schedules, scores and game highlights. Sporting events are an important part of college life for most students, so this content is considered a high priority. In addition, they suggested that the app should contain the WMU news feed, weather and school closing information, and event schedules by department. The app should provide additional content, like streaming the college radio stations, WMUK and WIDR, and photos of campus and students. They suggested that the app provide maps of campus with interactive GPS for locating events, parking lots, buildings, and classrooms. Information on how to apply to WMU should also be included. Future releases could have functionality and content that are more geared toward parents and teachers.

Soon after that, the iPhone team met with Scott Puckett again to further refine which features and design elements should be included in the iPhone app. Mr. Puckett provided a list of functions that he and Dr. Hearit would like on the iPhone app. It reiterated that they wanted the app to provide campus news, campus events, sports news, game schedules, scores, and multimedia content. As before, they requested that the app provide campus maps with GPS, and weather information. Additionally, they would like the app to be able to play the school Fight Song, and have a key-shaking sound effect - activated by shaking the iPhone - for “key plays” at games. They also suggested that the app provides access to dorm and cafeteria menus, and access to Western Michigan University’s Facebook and Twitter feeds. Additionally, they would like to include Bronco Transit GPS maps and schedules, made available via Auxiliary Enterprises at www.broncotransit.com.

Mr. Puckett constantly addressed his concerns in order to resolve some issues. He said that a non-disclosure agreement would not be necessary for this project. He said that, while the iPhone team would be credited with the design and creation of the app, the Vice Provost’s office would own it, and be in charge of maintaining it. He said that they would like the apps to have the same branding as the official website. To accomplish this, he will provide the official University logos and the RGB colors that the website uses. He also looked into securing data feeds for news and menus and he stated that it looks unlikely that he will be able to obtain sources for polls that are regularly updated.
3. Design Decisions

The Design Decisions section outlines the choices made during the designing and building of the application. This includes implementation decisions such as the programming language and the development platform, as well as data sources and user-interface layout.

3.1 Development Platform

XCode was chosen as the development platform. XCode is the primary software suite of tools that is used for developing iPhone apps. XCode is bundled free with Mac OS X v10.6. XCode includes useful development tools such as Interface Builder, which is an application for designing graphical user interfaces, iPhone Simulator which generates a virtual iPhone to run iPhone apps, and many different performance tools called Instruments.

3.2 Programming Language

Objective-C was chosen as the programming language. XCode only allows programming in C and Objective-C. Since most code libraries supplied with the iPhone SDK are written in Objective-C, using the same made it easier to interface. Objective-C is a superset of the C language with Smalltalk-style messaging added. All of the syntax for non-object oriented code is identical to C, while the object oriented features use Smalltalk-style messaging. Messaging allows various components to listen for messages from other components and unlike a call, there is no guarantee that a message will be passed. They are used extensively in the iPhone for messages regarding device orientation and movement, screen touches, and virtually all other data that might be retrieved from the device’s operating system.

At the beginning of the project, XCode and Objective-C were the only choices afforded by Apple, though this restriction has been lifted since.[*14]

3.3 User Interface

The user interface of the app was designed based upon Apple’s specific guidelines known as the HIG (Human Interface Guidelines). Each app should have the same feel as any other app and should not require the user to have to familiarize themselves with it.

The app’s appearance and handling abide with the following rules:
• Splash-Screen.
  ○ The app launches with a splash-screen resembling what the app will look like when it opens. This reduces the appearance of load time.
• Rounded Buttons
  ○ The main screen has evenly spaced buttons with rounded corners that lead to each of its features. Rounded corners on buttons alert the user that they can be pressed.
• Transition
  ○ Main features transition with a fade, while the information button flips the screen so it appears as though you are reading the back of the main window. This is not required but common and encouraged.
• Control Layout
  ○ Each feature has a navigation bar at its top and a toolbar at its bottom. All its controls reside within these bars and always in the same position. For example, exit is always in the top left corner since users will press without even looking.
• Visual Representation
  ○ Information is displayed with the attempt of displaying relevant information clearly and without clutter or confusion. This results in showing less, but larger and clearer data, that can be interpreted at a glance.

3.4 Data Source

The primary data source for content was decided to be a special University website created exclusively to serve data to the app. It can take more than a week to post an update to the app and much longer than that for users to download the update. The University wanted the ability to modify content or update information remotely without having to submit an update to Apple.

Instead of building various URL’s into the app itself, as may be common practice, the app communicates with a URL in the data website which then redirects to the appropriate link. If a link gets moved, the data website can be updated immediately, and the app will continue to function as expected.
4. Design

The Design section describes the layout and directory organization of the *WMU Mobile for iPhone* App. It describes how the Three Tier Architecture approach was used in this app. Three Tier Architecture helps developers to design a flexible and reusable application. [*15] The three tiers are the user interface layer, application layer, and the data layer. The *WMU Mobile for iPhone* App is designed with a main core which launches the Main View (see Figure 5.1). The Main View has buttons that launch each of the apps separate features. Each view interfaces with the main controller such that it can launch them. This makes the app flexible; new features can easily be added and obsolete functions can easily be removed.

The user interface tier of the *WMU Mobile for iPhone* App is composed primarily of the .xib files. The XCode Interface Builder allows creation of individual Views that can be linked to specified code. For example, in the case of the Weather View, the display of the conditions and the forecast do not depend on where the data is retrieved from. This will prove useful if the source it currently uses becomes unavailable.

The application tier of the app includes the classes and libraries that are used to perform the logic and calculations for each of the features. Each view has an associated view controller. This preserves order while keeping the interface layer and application layer separate. Several libraries and classes used to manage and manipulate app behavior, web data, and media files.

The data tier of the app is primarily web data that is maintained by various Western Michigan University sources. A few features rely on freely available web content. The “Bus” feature pulls its data from www.broncotransit.com. The “Contacts”, “Events”, “News”, “Social”, and “Sports” features point to redirect pages that will be maintained by Scott Puckett at the Vice Provosts office. The “Radio” feature streams audio from www.wmuk.org. The “Map” and “Weather” features make use of Google’s Map API and Weather API, respectively.
5. Implementation

The Implementation section shows the solution that the *WMU Mobile for iPhone* app provides for Western Michigan University students. The screen shots below demonstrate the solutions for bus schedules, contact information, event schedules, maps, news feeds, radio connections, school social networking, WMU audio media, athletics information and local weather forecasts.

When the user launches the *WMU Mobile for iPhone* app from his or her device, the home page in Figure 5.1 is launched. The user can then select from any of the listed features. Also, at the bottom is an info button that displays credits to the programmers.

Figure 5.1

![Carrier 12:51 PM](image)
When the user selects the “Bus” button, the view in Figure 5.2 is displayed. From there, the user can select one of two links: schedule and bus location. The schedule link takes the user to the bus schedule to find out when a particular bus arrives where. The bus location button takes the user to a screen where they can click on a particular bus and see its exact location on a map.

Figure 5.2
When the user selects the “Contacts” button, the view in Figure 5.3 is displayed. This view allows the user to see a few main Western Michigan University phone numbers, and if using an iPhone, the user can actually click on a number to call it.

Figure 5.3

Contact WMU

Admissions
(269) 387-2000

Account Services
(269) 387-6000
For billing questions

Emergency Information
(269) 387-1001
For open/closed status and emergency notices

Public Safety
(269) 387-5555
24-hours a day

University Operator
(269) 387-1000
When the user selects the “Events” button, the view in Figure 5.4 is displayed. This view displays upcoming events around Western Michigan University, with times.

Figure 5.4
When the user selects the “Maps” button, the view in Figure 5.5.1 is displayed. The user can view the satellite image of the map by selecting the “Hybrid” toggle at the bottom right of the view. This will display Figure 5.5.2. The user can also click on the search button to see a predefined list of locations on the Western Michigan University campus.
When the user selects the “News” button, the view in Figure 5.6 is displayed. What the user sees are little news clips about Western Michigan University, and usually a link is included.

Figure 5.6

- **wmunews** Francophone Fest at WMU March 16-20 packs rare films and visit from noted Algerian filmmaker Abdelkrim Bahloul:
  - about 2 hours ago

- **wmunews** Flags lowered Friday for Sgt. Kristopher Gould, 25, of Frankenlust Township, Mich., who was killed in Afghanistan:
  - about 21 hours ago

When the user selects the “Radio” button, the view in Figure 5.7 is displayed. The user can click on either channel, HD1 or HD2 and then click the “play” button to start streaming WMUK. There is also a volume control located at the bottom of the screen.

Figure 5.7
When the user selects the “Social” button, the view in Figure 5.8 is displayed. From there the user can click on any social media link to direct them to the respective Western Michigan University social page.

Figure 5.8
When the user selects the “Sounds” button, the view in Figure 5.9 is displayed. The user can then click on any of the play buttons to make a song play and at the bottom can enable or disable the key shake module. If the user shakes the iPhone, a sound of keys is played.

Figure 5.9
When the user selects the “Sports” button, the view in Figure 5.10 is displayed. The user can then click on any link to see details on Western Michigan University athletics.

Figure 5.10
When the user selects the “Weather” button, the view in Figure 5.11 is displayed. This displays the current weather up to three days into the future.

Figure 5.11
6. Testing

The Testing section discusses the methods used for ensuring that the *WMU Mobile for iPhone* app is reliable and sturdy. Major software development projects require thorough testing procedures including Unit Testing and User Testing.

Unit Tests are tests that check functionality of small discrete sections of source code. The goal is to test all expected input to see if the code produced valid output. Unit tests verify that code is fit for use. A change will not be committed to the version control repository unless the code passes all unit tests. Every time something is added or modified, the entire project should be retested. This process is called regression testing. The Unit Tests for this project are automated for continuous regression testing for the entire project. Unit Testing for the *WMU Mobile for iPhone* app is performed by OCUnit. (For more detailed information on how OCUnit is used for the app, please see Appendix G.)

For User Testing, Scott Puckett has arranged for a group of twenty-four individuals with access to an iPhone, iPod Touch, or iPad device to test the *WMU Mobile for iPhone* app. After the app has been approved by Apple for release, the individuals will be able to download and use the app on their devices. A survey is being prepared that will gage the users interest in the app, the functionality of the app, and satisfaction of the app. The survey consist of a series of statements which the user can select from a range of responses from “highly disagree” to “highly agree”. Free response questions will also be provided. The data collected from the survey will be a valuable resource for catching any bugs and for adding future functionality.
7. Summary

In summary, the *WMU Mobile for iPhone* app was designed to fill a need for Western Michigan University Students. Previously, WMU did not have a general application for smart phone devices. The *WMU Mobile for iPhone* app provides a mobile source for campus news, weather, and athletics highlights, along with access to various sources of WMU media, and GPS maps of campus buildings. It is available for free through the Apple App Store for the iPhone, iPod Touch, and iPad devices.
8. Appendices

8.1 Appendix A: Spikes

This section is a list of the experiments performed to learn about iPhone and iPod Touch apps, the Objective-C language, XCode, Interface Builder, and Subversion. We will also include a list of experiments that we intend to perform as the project progresses.

Other Apps

Initially, apps from other universities were examined to find examples similar to this project. Several apps were downloaded, specifically from major and surrounding Universities. The apps were used to determine which features may be implemented for this project. Examples of good visual design and intuitive layout were particularly noted.

Objective-C and Apple Cocoa

The largest of the initial spikes was to become acquainted with the Objective-C language and Apple Cocoa API. Familiarization with Mac hardware Objective-C syntax is especially important for this project. There are several excellent online PDFs for understanding and writing Objective-C. [*8] [*9] [*10]

XCode and Interface Builder

Several YouTube tutorials were used to design several simple iPhone apps. The aim was to become comfortable using XCode and Interface Builder for designing attractive and functional interfaces. The process involved experimenting with background images, controllers such as buttons, and navigation between views and subviews. This process demonstrated what was necessary for GUI design and gave some experience with views and XCode. Below is a snapshot of the XCode IDE from Apple.
View Switcher

With the knowledge from the previous spikes, it was possible to design and implement a simple view-switching app. This app featured a more elegant and instinctive design, smoother transitions between windows, and cleaner code than previous attempts in the last spike. This provided more experience with XCode and Interface Builder, and helped to decide the programming style and initial visual appearance for the core.

Quad Program

The Quad Program is an in-class project designed to familiarize students with Subversion and version control in general. The program is a simple solver for quadratic equations. This spike allowed experimentation with checking out a project, updating it, and committing it using Subversion commands. This spike also taught basic organization strategies for setting up project directories and maintaining source code,
documentation, and other necessary files.

**Data Sources**

Acquiring data sources could potentially be the biggest spike for this project. Scott Puckett is currently working to determine which data sources are available for news, sports, polls and menus and if permission can be granted to use them. The next meeting with Mr. Puckett should hopefully provide answers to the questions in this spike.

**Surveys of Students**

Statistical information on smart phone usage, within the Western Michigan University student population, would be a very valuable addition to this project. It will likely be necessary to design a survey that will determine which students use smart phones, which types of smart phones (device and OS) they use, and how they use their smart phones for University related activities. It would also be very useful to know how many students plan on obtaining a smart phone within the next few years. Mr. Puckett said that a survey of this nature would be easy to create, but he expressed concern that the data might be incomplete due to limited student participation.
In Extreme Programming a story is a software requirement from a high level perspective. It is written in the everyday business language of the customer and represents the feature the customer wants to achieve. The story is only a sentence or two long and clear in meaning. It is easy for the customer to confirm that it has been implemented after the development team says it is complete. [*12]

The following are stories for this project.

<table>
<thead>
<tr>
<th>Release</th>
<th>Description</th>
<th>Time (Weeks)</th>
<th>Risk (1-10)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>A runnable iPhone application that launches a blank window with 2 buttons on it. The first button is an information button that flips to show a blank information window when it is clicked. The second button is a generic “Story 1” button that transitions into a blank window when it is clicked.</td>
<td>0</td>
<td>1</td>
<td>The “Story 1” button will be transformed into the real story 1.</td>
</tr>
<tr>
<td></td>
<td>Add a WMU News window that displays headlines about WMU and allows the user to read each story when they click on the headline.</td>
<td>2</td>
<td>3</td>
<td>Need a source for news</td>
</tr>
<tr>
<td></td>
<td>Add a Map view that uses a google map or similar to show WMU. Overlay map with WMU color-coded parking. WMU buildings and halls already exist on google maps.</td>
<td>2</td>
<td>4</td>
<td>Will be able to use google maps but may need to write in our own overlay for parking.</td>
</tr>
<tr>
<td></td>
<td>Alert users to extreme weather and school closings via Push notifications. Allow users to opt in and opt out of service.</td>
<td>3</td>
<td>8</td>
<td>WMU already has an emergency cell phone service. Could possibly use</td>
</tr>
<tr>
<td>Release 1</td>
<td>Add a page with WMU information for prospective students. Give steps on how to apply and show important phone numbers.</td>
<td>2</td>
<td>3</td>
<td>Static data. If this needs to be updatable, then we need a source.</td>
</tr>
<tr>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>Release 2</td>
<td>Add an events page. Allow user to view events by department.</td>
<td>2</td>
<td>7</td>
<td>Need a source.</td>
</tr>
<tr>
<td>Release 3</td>
<td>Add search to Map. If someone types “Rood”, a pin drops on Rood Hall and map zooms in.</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Release 4</td>
<td>Add a Poll page which displays polls and allows users to vote.</td>
<td>3</td>
<td>7</td>
<td>Need a source.</td>
</tr>
</tbody>
</table>
8.3 Appendix C: NDA and Project Ownership

Western Michigan University will fully own the *WMU Mobile for iPhone* app with the agreement that the developers are listed in the credits.

A non-disclosure agreement is not believed to be needed by Scott Pucket but the matter is currently under review by WMU's legal department.
8.4 Appendix D: Resources

To create the *WMU Mobile for iPhone* application each of the following was required:

- An Intel-based Mac computer running OS X 10.5 or later.
- The iPhone SDK.
- A membership to the Apple iPhone Developer Program ($99/yr). [*13]
- A subversion account for team based work on the project.
- An iPhone/iPod Touch for testing.
- And at least 14 weeks to release the first version.

Each member of the iPhone team owns or has access to a compatible Mac computer. These Macs all have the latest iPhone SDK. One team member is a current member of the Apple iPhone Developer Program. Western Michigan University also needed to join the developer program, so that it could list itself as the creator of the application.

The team has a working Subversion account and written standard procedures for using it. The team has access to at least one iPhone and several iPod Touches. Finally, there is a minimum twenty-four weeks to work on the project before graduation in the spring. Since the first release is tentatively scheduled to be finished in fourteen weeks, this allows plenty of time to release, and probably also do several subsequent releases.
8.5 Appendix E: Programming Standards

Style Guide
The iPhone app is being written in the style detailed in “Cocoa Style for Objective-C” style guide published by CocoaDevCentral.com. The guide can be viewed or downloaded at http://www.cocoadevcentral.com/articles/000082.php.

Unit Testing
The iPhone app project is being tested with the unit testing framework OCUnit. The software is open source and recommended by Apple for XCode projects. It is fully integrated into XCode. Tests can be configured to automatically run on each build. OCUnit can be downloaded here: http://www.sente.ch/software/ocunit/#Download. More information can be found here: http://developer.apple.com/tools/unittest.html.

System Testing
We will refer to system testing as the process of ensuring the iPhone app does what it is supposed to do. This will first be done by comparing the user stories to the iPhone app’s actual functionality. This will first be done by the development team, then by the customer. When both groups are satisfied the system tests will be considered passed.

Automation
All unit testing is done automatically on every compile. Code that has not been compiled is not committed to the project.
8.6 Appendix F: Version Control Guide

Any major software development project should include some type of version control. Software developers often have a need to be able to manage the current version of their software, as well as be able to retrieve a historical version of it. Not all clients will automatically upgrade to the latest version of a particular software package. So, to support earlier versions of a product, a developer will need to be able to view the exact source code of each historical release. The version control system that will be used for this project is Apache Subversion. Subversion has built-in support for multiple developers to work simultaneously. At each software check in, Subversion will look for conflicts with other checked in work. If conflicts are found, it will alert the developer and allow him or her to try to merge the differences. The following is the iPhone team’s guide for using Subversion with XCode. Assuming SVN is installed and turned on:

Initial Set-up:
Go to
XCode/Preferences/SCM/Repositories
Under Options make the following selections.

Configuration:  
- Configure SCM automatically
- Save files before SCM operations

Operations:
- Configure SCM automatically
- Save files before SCM operations

Comparison Handling:
- View comparisons using: FileMerge
- Display local files as they are: Right

In Xcode/SCM/Repositories
1. Select WMUAPP directory and click Checkout.
2. Save. If the directory already exists locally and SVN prompts to overwrite it, skip to step 12.
3. After Checkout complete, open project.
4. Now when a change is made to file, an "M" for modified, will show up in left hand column.
5. In the SCM group there will be a pbxproj file that also shows modified. This is because something within the project changed. A build directory with a “?” mark will probably be displayed. The “?” means the directory is not part of SVN repository, but the build should not ever be part of it, so just ignore.
6. To add a new file to the project: After adding, the “?” marks are on the new file. Right click the file and add to repository, Now the “?” turns to an A(dd).
7. SCM/Refresh Entire Project whenever you want to force an refresh between the local copy and the repository. This doesn’t change any files, just will show things like the “?” or “A” or “U”, etc.
8. If on refreshed the project a “U” appears to signify an updated copy of that file has been committed. Update the entire project to merge the changes onto your local copy.
9. Now run Unit tests.
10. If everything passes, run update again to see if anyone else committed in the meantime.
11. Commit the entire project and comment the commit.
12. Finally delete the local copy. This is important due to big problems overwriting the local copy when checking out the project later.
// StoryViewController.m
// WMUAPP
//
// Created by Tim Wickey on 10/
// Copyright 2010 WickeyWare. #
//
#import "StoryViewController.h"

@implementation StoryViewController

-(IBAction)backToMain{
    [self dismissModalViewController];
}

NSInteger addedfordemonstration;

@end
8.7 Appendix G: Unit Testing

Unit testing on the WMU Mobile for iPhone app is performed with OCUnit. It is an open-source framework integrated directly into XCode and recommended by Apple. It can be downloaded here: [http://www.sente.ch/software/ocunit/](http://www.sente.ch/software/ocunit/).

The Unit test bundle can be found in WMUAPP/Source/UnitTests/. It is a wholly separate project and has its own target, but no executable and can only be ran as a dependant of the project. Note also that tests do not have a header file. In reality they do, but are simply appended to the beginning of the main file. This is standard practice as test case headers are almost always empty.

Adding a Unit Test:
1. Add a test method to the appropriate TestCase.m
   a. The method MUST begin with “test” followed by a name.
      i. Example: -(void)testMethod{ }
   b. Use various STAssert macros to test such as STAssertNotNil, STAssterTrue, STAssertEquals, etc.
      i. STAssertTrue((1+1)==2, @"1 + 1 does not equal true");
2. Running Unit Tests:
   1. Right-click and select info on the WMUAPP target.
   2. In General, UnitTestBundle should be a direct dependant of WMUAPP.
      a. This will not change.
   3. Simply Build the project. At the beginning of the Build Results window you can see all your unit tests and their status. If a test fails you will get a failed build message with the test that failed and its message as you wrote it (1 + 1 does not equal true).

Important:
- Since the unit tests rely on your classes you must always import your headers files into your test before using it. Because of this, every class, file, and framework you add to your project MUST also be added to the UnitTestBundle.
- In example, when adding an new class, you must always check both WMUAPP and UnitTestBundle as targets. (See screenshot). When adding a framework to WMUAPP, remember to immediately after go add that framework to UnitTestBundle.
This screenshot shows the beginning of the unit test file. Each of the objects to test must be included. Then each of the methods that begin with “test” are automatically ran on build and results displayed to Build Results.
window.

#include "MainViewController.h"
#include "NewsViewController.h"
#include "InfoViewController.h"
#include "WMUAPPAppDelegate.h"
#include "WeatherViewController.h"

// only run on the simulator
#ifndef TARGET_CONDITIONALS_H
#define TARGET_CONDITIONALS_H

#include <SenTestingKit/SenTestingKit.h>
@interface GeneralTestCase : SenTestCase
@end

@end

@implementation GeneralTestCase

#ifdef USE_APPLICATION_UNIT_TEST  // all code under test is in the iPhone Application

-(void) testAppDelegate {
    id yourApplicationDelegate = [[UIApplication sharedApplication] delegate];
    XCTAssertNotNil(yourApplicationDelegate, "UIApplication failed to find the AppDelegate");
}

@end

#endif
8.8 Appendix H: The Release Process

When a version of an iPhone app is ready to be released, there are several steps to take with regards to version control. A separate directory will need to be created in the versioning repository. This is called a branch. The branch is a dead end that essentially holds the released version. Any future bug fixes should be applied to the branch, as well as the main source code, where applicable. This main source code is called the trunk. The development process will continue on the trunk code.

Apple Inc has a specific submission procedure for iPhone apps. To submit an app, a developer must be a member of the iPhone Developer Program. A membership costs $99 per year. Apple requires that prospective apps do not duplicate an app that Apple already provides. Apps cannot use copyrighted music, images, or other intellectual property without permission. When submitting an app, a developer must write a 700 character or less description of the app, provide screenshots for display in Apple's app Store and set the price. After submission, approval can take two to four weeks after which the app appears in the App Store, and anyone can find and download it to their iPhone. [*7]
9. References

10. Glossary

- AdMob - One of the world's largest mobile advertising networks.
- Android - A mobile device operating system created by Google.
- API - Application Programming Interface
- App - An executable application for a mobile device.
- Apple Inc - An American corporation that designs and markets consumer electronics, software, and personal computers.
- App Store - The online store where people can browse and download apps for Apple devices.
- ARM Processor - The processor used in the Apple iPhone. Stands for Advanced RISC Machine developed by ARM Holdings.
- C Language - A general purpose low level programming language developed by Denis Ritchie in 1972 in Bell Labs for use in the Unix Operating System.
- Client - The person or entity at whose requests the software is being developed for.
- Cocoa - One of Apple’s native object-oriented APIs for Mac OS X
- Compiler - A compiler is one or more programs that transforms program code such as C, C# or Java into a computer language, often known as binary or object code so that an executable could be constructed.
- GCC - GCC stands for GNU Compiler Collection. It is a compiler that was made for compiling the C, Objective-C, C++, FORTRAN, and Java programming languages.
- Ghz - 1,000,000,000 cycles per second.
- GNU - A worldwide project that was announced in 1983 that allows everybody to have unrestricted access to the code and libraries to the project whose goal is to create free software to work with software that is not free.
- Google - An American corporation that is primarily known for their Internet search engine, also known as Google.
- Google style map - A map that can show a regular street map, a satellite imagery map, or a hybrid which places street names on a satellite map.
- GUI - Graphical User Interface.
- HIG - Human Interface Guidelines
- IDE - Integrated development environment. A platform that provides all the tools necessary to build applications.
- iOS - Apple Inc’s Operating System for the iPhone, iPod Touch, and iPad.
- Interface Builder - XCode’s GUI design tool.
- iPad - Apple Inc’s 9.7 inch tablet computer.
- iPhone - Apple Inc’s popular mobile phone.
- iPhone Simulator - A feature built into XCode that allows an iPhone application to be ran on a virtual iPhone.
• iPod Touch - A portable media player which runs the same operating system as the iPhone.
• LCD - Liquid Crystal Display
• Mac OS X - The tenth series of Apple Inc’s operating system, called simply OS X.
• MB - Mega Byte. 1,000,000 bytes.
• Mhz - 1,000,000 cycles per second.
• Objective-C - The programming language required to develop applications for the iPhone.
• RIM OS - The operating system for Blackberry phones.
• RISC Architecture - Stands for Reduced Instruction Set Computer. It is based on the thought that simple instructions are faster computed so the architecture attempts to reduce the instructions first then execute.
• SDK - Software Development Kit.
• Smalltalk - An object oriented language made for educational purposes.
• Subversion - A versioning and revision control software that keeps incremental backups of software under development in a code repository. Subversion allows programmers to keep current and historical versions of the files that comprise their projects.
• Superset - A set that completely contains a second set is a superset of the second set. “The set of animals is a superset of the set of dogs.”
• Spike - An experimental program made to test the programmer’s theories and to learn.
• Story - A functional behavior of software, agreed on by the client and programmer. A story is a description of what the User will see. A story will not include the details of implementation.
• WMU - Western Michigan University
• XCode - Apple Inc’s proprietary IDE for building software for Mac OS X and iOS.