**CS3714 iOS Mobile Software Development**

Fall 2013

**Weather Panorama**

**Alex J Norton**

Department of Computer Science

Virginia Tech

Blacksburg, VA 24061

Date: 12/15/2013

Email: ajn123@vt.edu

Submitted to: Prof. Osman Balci

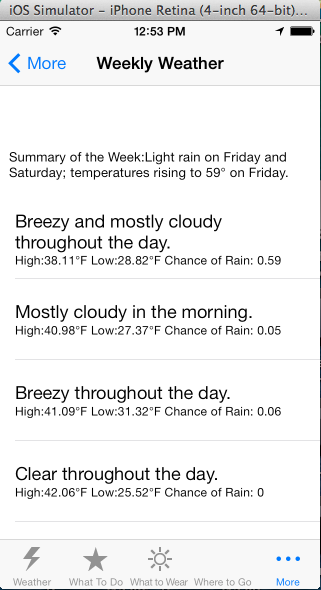
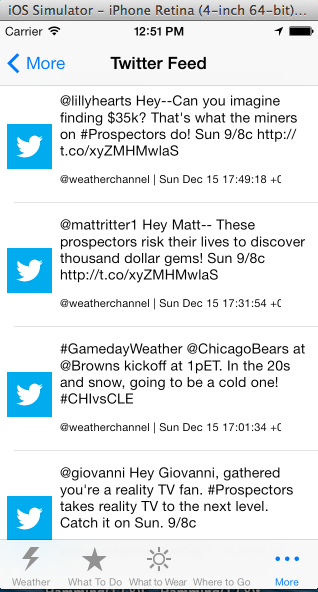
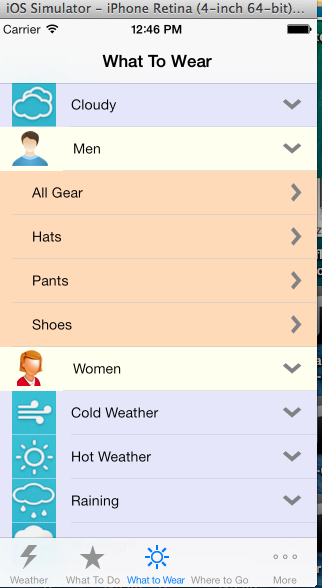
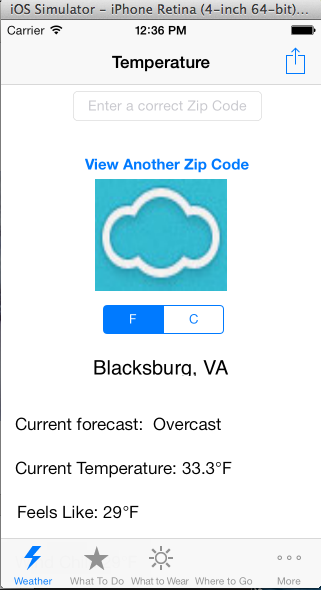
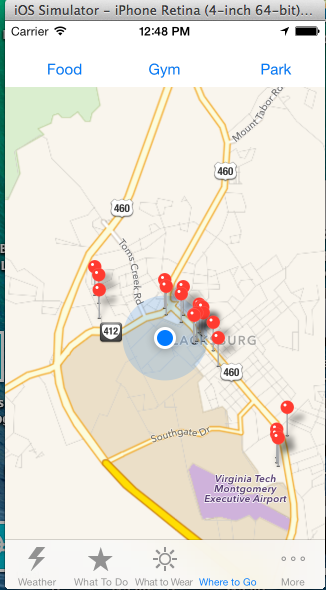
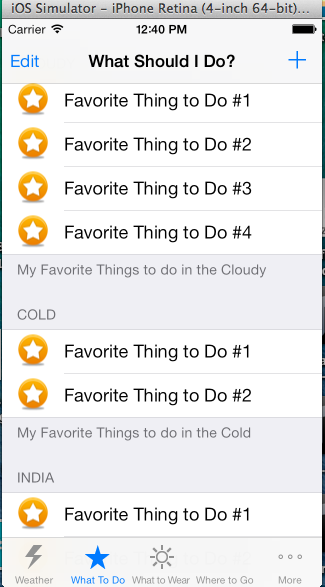
EXECUTIVE SUMMARY

This weather Panorama application allows for a user to easy access their current weather. Let the user see what they should wear in the weather and where they should go in certain weather conditions. The user can also view different places and where they are based on the weather such as a restaurant, park or gym. The user can also get a weekly update of the weather and swipe to view other information.

A twitter feed is also available to know what is happening with the latest weather on a national stage and how the world is feeling about it. They can also find out more by clicking the tiles and seeing what the latest information of the weather channel is.

The user can also see what they should wear in a current weather situation and look at links to buy those kinds of clothes. This list expands and grows and has categories for guys and girls.

The user can also save their favorite things to do during certain weather conditions in a list that organize them and ranks them by favorites. Allowing the user to change their number one of delete any entry they are tired of.



**TABLE OF CONTENTS**

EXECUTIVE SUMMARY ii

1. Problem Specification 1

1.1 What is the problem? 1

1.2 Why is the problem important to solve by engineering an iOS application? 2

1.3 Description of Your App’s Expected Functionality 2

1.4 Learned iOS Features My App Will Implement 2

1.5 New iOS Features My App Will Implement 4

1.6 Complexity of My iOS App 4

2. Requirements Specification 5

3. Architecture Specification 5

4. Design Specification 5

5. Delivered Software 5

6. Conclusions 6

References 6

# Problem Specification

Many weather applications only tell you about a 7 day forecast but what about the ability to see the weather from multiple cities, possibly the ones you are traveling too.

## What is the problem?

Weather is very important in society, it dictates what you do, and when you do it.

## Why is the problem important to solve by engineering an iOS application?

My weather app will display the user with a 7 day forecast of their current location. A user needs to know all this detail to be able to figure out how to plan their week as well as what can they do if the weather is bad or good.

## Description of Your App’s Expected Functionality

My tabbed application will display the weather of the user in a weekly (with a dynamic table view). The user will also be able to view the weather on weather.com will they can view HTML 5 videos in a UI web view. The user will also be able to browse different locations (scroll view) and view their respective weather forecasts. If the user long presses a forecast the respective sound will play depending on the forecast (long press) ie thunder for storms rain for rain and sunny for a sunny day.

I also want to have an additional app that displays activates that can be done in that area depending on the weather (you want to see a movie if it’s raining or go to the park if it’s sunny). Whatever the weather is the appropriate activities will show up for that weather. So possibly a query the location of the users activities on Google, with a UI web view display.

## Learned iOS Features My App Will Implement

Color each iOS feature your app will implement with red in the table below listing the iOS features learned in the course.

|  |
| --- |
| Accelerometer [[HokieCheer](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/HokieCheer/HokieCheer.html)] |
| Animation |
| * Jigsaw Puzzle [[Gestures](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Gestures/Gestures.html)] |
| Audio |
| * Play a short sound file upon shaking the device [[HokieCheer](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/HokieCheer/HokieCheer.html)] * Play/pause/stop a long sound file [[HokieCheer](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/HokieCheer/HokieCheer.html)] |
| CSV Files |
| * Read in and process a Unix OS file in Comma-Separated Values (CSV) text file format [[GeoTracker](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/GeoTracker/GeoTracker.html)] |
| Design Patterns |
| * Model View Controller [[Temperature](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Temperature/Temperature.html)] * Delegation [[Temperature](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Temperature/Temperature.html)] * Target-Action [[Temperature](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Temperature/Temperature.html)] * Pass data to a downstream view controller [Many of the tutorial apps] * Pass data to an upstream view controller using Delegation [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)], [[SongsILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/SongsILike/SongsILike.html)] |
| Geo-location-based iOS app |
| * Using Google Maps API [Google Maps Query String Parameters] [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] * Using Apple's MapKit Framework [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)], [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)], [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] |
| Gestures |
| * Detect and process user gestures: tap, swipe, rotate, pinch, pan (drag), and long press [[Gestures](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Gestures/Gestures.html)] |
| Global Data |
| * Create Global Data and use it in any class in your project [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)], [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] |
| Maps |
| * Current Location [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)], [[WhereAmI](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/WhereAmI/WhereAmI.html)] * Directions from address 1 to address 2 [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] * Create and display a map annotation object on a map in the form of a colored pin, which displays an annotation when tapped on [[GeoTracker](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/GeoTracker/GeoTracker.html)] * Create and display a polyline on a map showing a track consisting of geo-locations [[GeoTracker](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/GeoTracker/GeoTracker.html)] * Apple's MapKit map system [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)], [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)] * Google Maps JavaScript API [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] |
| Property List (XML) Files |
| * Read from application's main bundle [[ACC](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACC/ACC.html)], [[ACCSports](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACCSports/ACCSports.html)] * Read from a server computer over the network [[ACC](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACC/ACC.html)], [[ACCSports](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACCSports/ACCSports.html)], [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)], [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] * Written to documents directory on the iOS device [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)] |
| Scroll Menus |
| * Horizontally scrollable menu with selectable buttons as text, rounded rectangle, icon, and icon with text [[Autos](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Autos/Autos.html)] * Vertically scrollable menu with selectable buttons as icon with text [[Autos](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Autos/Autos.html)] |
| Split View-based iPad application [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)], [[NavalMines](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/NavalMines/NavalMines.html)] |
| Storage and Retrieval of User-specific Data |
| * By using property list (XML) files [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)] * By using Core Data / SQLite Relational Database [[SongsILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/SongsILike/SongsILike.html)] |
| Tab Bar-based Navigation [[TourGuide](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/TourGuide/TourGuide.html)] |
| Table Views |
| * Dynamic Table View [[ACC](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACC/ACC.html)], [[ACCSports](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACCSports/ACCSports.html)], [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)] * Expand / Shrink Table View Rows [[ACC](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACC/ACC.html)], [[ACCSports](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACCSports/ACCSports.html)] * Indexed Table View [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] * Custom-built table view rows (cells [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)] * Move rows within a section in a table view [[CitiesILike](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/CitiesILike/CitiesILike.html)] |
| Universal iOS application |
| * Develop a universal iOS application to run on all iOS devices [[ACCSports](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACCSports/ACCSports.html)], [[Countries](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Countries/Countries.html)], [[GeoTracker](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/GeoTracker/GeoTracker.html)] |
| User Interface |
| * Sliding View [[ACCSports](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/ACCSports/ACCSports.html)] * Spinning Wheel (UIPickerView) [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] * Create UI objects at run time [[Autos](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Autos/Autos.html)] * Resize and reposition UI objects at run time [[Autos](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Autos/Autos.html)] * Hiding the Status Bar [[FlashLight](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/FlashLight/FlashLight.html)] * Remove the keyboard by touching anywhere on the background [[VTQuest](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/VTQuest/VTQuest.html)] |
| Video |
| * Play a video file embedded in an HTML page using the HTML5 video tag [[NavalMines](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/NavalMines/NavalMines.html)] |
| Web Browser |
| * Make a portion of the UI or the whole UI provide the functionality of a web browser [[Web](http://manta.cs.vt.edu/cs3714/StudentsOnly/TutorialApps/Web/Web.html)] |

## New iOS Features My App Will Implement

I want to implement a twitter feed which pulls tweets from weather related tweeters and show them to the user so that they can better understand the weather. I also want the user to be able to share the weather with twitter if desired (have a share button).

I will also be using the wunderground API to get the weather information in an XML format which can be used to display the weather.

## Complexity of My iOS App

What will be complex about this app is allowing users to select any desired location (spinning wheel) for the weather and then clearing and updating the table list with the info. Also, knowing what the weather the user tapping on and playing the correct sound can cause some difficulty. Pulling the correct information from the wunderground and Twitter API can be tough too. I also want the user to access weather.com with their location to view up to date news about what is going on their area that is weather related. I also need to figure out the user’s location, query for it, and display it in a day format.

I also want to have an additional app that displays activates that can be done in that area depending on the weather (you want to see a movie if it’s raining or go to the park if it’s sunny). Whatever the weather is the appropriate activities will show up for that weather. So possibly a query the location of the user’s activities on Google, with a UI web view display.

# Requirements Specification

This app shall show the location of the user.

This app shall need Internet access.

This app shall need access to API such as twitter and Google places (and forecast API)

This app shall allow the user to update a list that is saved to disk

This app shall show user the closest locations of restaurants, gyms and parks.

This app shall use gestures to update information.

This app shall make a short sound clip upon shaking the device.

This app shall use a custom cell view.

This app shall use a custom annotation.

This app shall use a plist and save it to disk.

This app shall use a segmented view to update information.

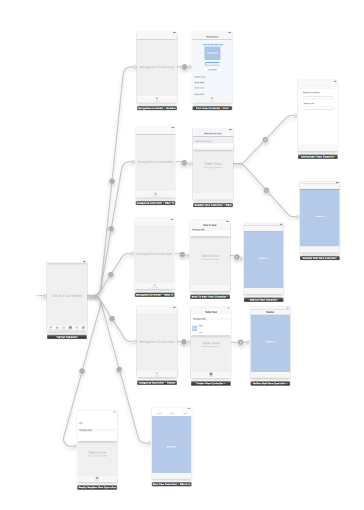
This app shall be able to edit a plist.

This App shall pass information to other view controllers.

# Architecture Specification

Two clients Server Architecture will be used, one to access the twitter weather feed. The Twitter API will be used. The other one will be the Google Place API to display. Also the Forecast API and Weather Underground API are used to get the weekly and current weather reports respectively.

# Design Specification



This tabbed application will have 6 main tabs with 10 total screens the user can visit. The user will be able to view their current weather and see any other US zip codes weather as well as their own locations weekly weather. There will be a “what should I do” page outlining links of what someone should do in a current weather situation, the user can add or delete links as necessary.

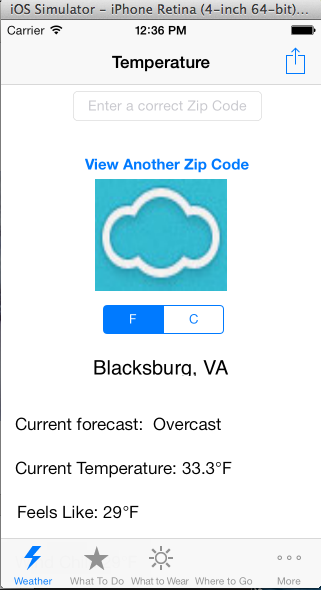
There will be a “what to wear” section which will be an expandable table view showing men’s and women’s section of what they should wear.

There will be a where to go tab that allows the user to select different types of places that will be annotated in the map below.

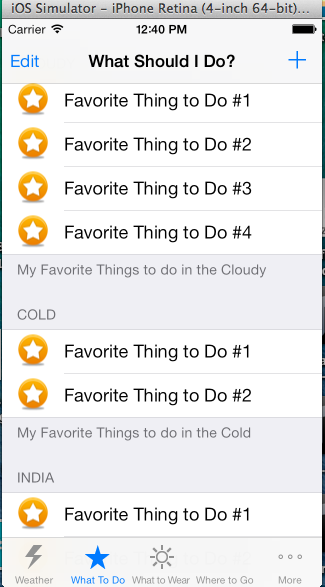
A weather twitter feed will be available that displays the latest tweets from the weather channel, the user can click them and go to the weather channel site to learn more.

There will be a weekly weather table that will allow for swiping gestures to update the main information. The weekly weather report will be in a table updated by the forecast API.

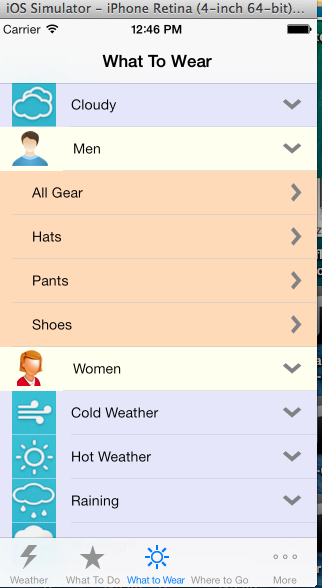
# Delivered Software



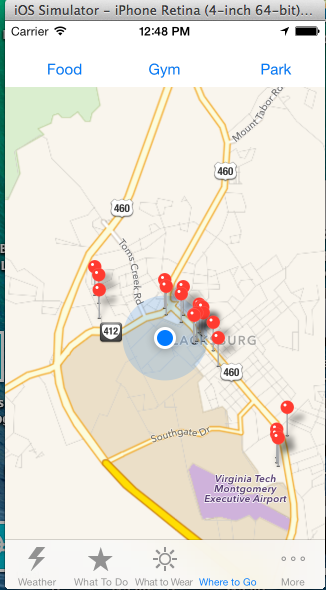
This first view gets the current users location and updates the labels on what the temperature is and which city they are closes too, what wind chill feels like, and a clip of text explaining the weather conditions, the picture updates to look like the closest weather condition. Touching the segmented control alternates between Fahrenheit and Celsius. In the text box above you can enter another US zip code and get the zip codes weather as well by clicking the button below it. The share button in the top right allows the user to share the weather instantly with their choice (Facebook, Twitter, email, etc). Shaking this screen also provides a different sound effect based on the weather.



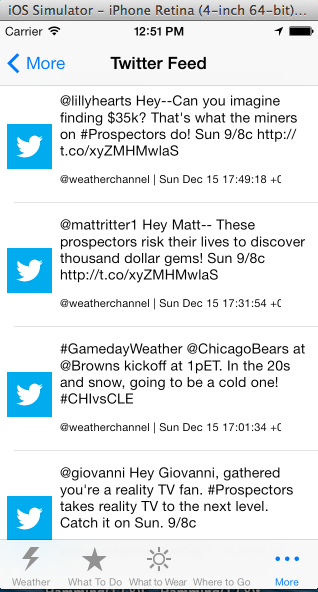
This list keeps track of things you should do in the weather as well as your favorite things to do, you can edit your favorites by moving the rows, or delete something if you do not like it(edit button). You can also add favorites by clicking the add button (+). If you click any of the rows a web view will be display of what the link that you set.



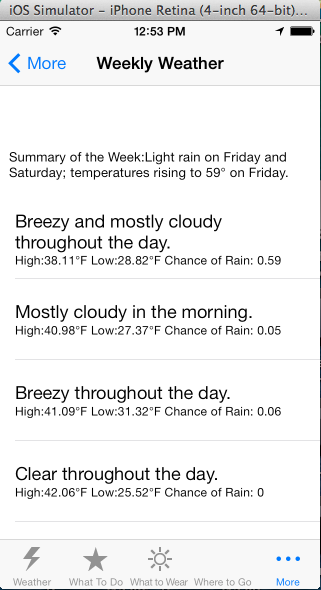
This screen keeps track of what you should wear in the weather, you can select what you want to wear based on the current weather conditions and click on the corresponding clothing and you will be linked to a site that shows you what you can wear for the particular weather condition.



This maps takes the user location and zooms in on it. (Location must be enabled) The user can then select either food gym or park to get the nearest locations of those places. This screen uses Google Places API to find these locations and places markers around them. Custom annotations are used and placed to display the user desired locations and are removed when the user selects another one.



This tap displays the twitter feed of the weather channel made with custom table rows. The user can click each item to be brought to the weather channel to watch videos and learn more about their current weather situation.



This tap displays a weekly update of the weather. The top displays a label summary of the weekly forecast. This uses the forecast API and the users location. You can swipe the top text box (up, down, left, or right) to view other weekly information, such as the nearest storm or coordinates.

# Conclusions

I enjoyed making this application. The use of the API’s and parsing JSON was a great tool to learn. The table views allow for an easy way to set information apart. Implementing the different methods for a table view allows flexibility with how you can show the table.

I learned how to make a table view with a hierarchy to allow for separation of clothing and women and men for a nice table view that can expand and detract. Selecting an item brought up a Webview, the website was passed information down to another view controller.

I also learned how to load a plist and save it to disk so the user can edit it as will and it will always be there for them. I learned how to use a map view as well as the Google Maps API to find update, and zoom in on the users location. I also learned how to make custom annotations and display them as pins on a map as well as show other pins based on what the user selects.

I made a custom table view cell for the twitter feed and allowed users to click to go o n weatherchannel.com and view more weather information.

The user can also view the weekly weather, which takes advantage of the Forecast API. A summary box at the top detects swipe directional gestures and displays different information depending on the gesture.

I learned how to interact with the twitter API and display tweets from any user such as @weatherchannel. I could then take the user their website based on a cell they selected.

References

Apple, Inc. (2012), “iOS Developer Library,” <https://developer.apple.com/library/ios/navigation/index.html>

Balci, O. (2012), “CS3714 iOS Mobile Software Development Course Website,” <http://manta.cs.vt.edu/cs3714>

"Google Places API Feedback on This Document." *Getting Started*. N.p., n.d. Web. 15 Dec. 2013. <https://developers.google.com/places/documentation/>.

"V2 Forecast API." *Forecast API Docs*. N.p., n.d. Web. 15 Dec. 2013. <https://developer.forecast.io/docs/v2>.

"A WEATHER API DESIGNED FOR DEVELOPERS." *API*. N.p., n.d. Web. 15 Dec. 2013. <http://www.wunderground.com/weather/api/>.