

Homework Assignment #3

WeatherTracker Part 1

Summary

In this assignment you we will start remaking your weather app from the Mac OS X course, re-imagined for iOS. This project will be broken up into multiple assignments. In this first assignment, all you need to do is create an app that shows a fixed list of weather stations, and lets you edit the name, code, latitude and longitude of those stations.

Demonstrate Your Understanding Of:

1. How to set up a UITableViewController.
2. How to populate a UITableView using its data source protocol and your model.
3. How to “drill-down” to a detail controller using UINavigationController.
4. How to populate a user interface with values from the model.
5. How to keep your fields out from under the keyboard.
6. How to update the model using the latest values from the user interface.

Functional Requirements

1. The app must display a list of weather stations (using a table view). I suggest creating a few stations in code and adding them to your list, but the more adventurous might want to load their stations from a property list. You do not have to use Core Data (we'll convert to Core Data in a subsequent assignment).
2. Each row must show the station's name, latitude, and longitude. Latitude and longitude should be shown on a separate line.
3. Tapping on a station must navigate to a detail controller for that station.
4. The detail controller must have text fields for the station's “name”, “code”, “latitude” and “longitude”. These fields must be populated with the current values from the selected station upon becoming visible.
5. The user must be able to tap each field and change its value.
6. You should keep the selected field from being lost underneath the keyboard (this is highly desired, but if you struggle with it, it would be acceptable at this point to arrange the fields so that they can never be below the keyboard).
7. Leaving the detail controller (navigating back) should save the current value for each field to the selected station, and the changed values for that station should be visible in the table view.

General Requirements

1. The app must not crash.
2. You must not use any private API.
3. You must follow Apple's Coding Guidelines for Cocoa (See: <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/CodingGuidelines/CodingGuidelines.html>).
4. The app must not leak memory or other resources (such as sockets, file handles, ports, etc).
5. The app must compile with ZERO compiler warnings and ZERO warnings from the static analyzer.

Suggestions

The "Homepwner" exercises in the Conway & Hillegass book (chapters 10-12) are similar to this exercise and should be helpful in meeting most of the requirements.

For help keeping the keyboard from covering your controls, read Apple's "Text, Web, and Editing Programming Guide for iOS", in the section entitled, "Moving Content That is Located Under the Keyboard".

BONUS Opportunities

1. +10 bonus points for implementing an "Add Station" feature that allows the user to add new stations.
2. +5 bonus points for implementing a way to delete one or more stations.
3. +5 bonus points for implementing an edit mode where stations can be re-ordered by dragging them up or down (like on the back-side of the Stocks app).

Submitting Homework & Due Date

You must submit your project source code, including the Xcode project file, all nibs, and all other resources, in a single ZIP archive **BY WEDNESDAY, FEB 2 at 11:00 PM Pacific Time**. Homework should be submitted to the CollectIt Dropbox linked from the course homepage: <https://catalyst.uw.edu/workspace/luke1/18578/>