Homework: Keychain

The purpose of this assignment is to demonstrate your understanding of the iOS keychain API, including Core Foundation concepts, keychain items and attributes, and the SecItem functions.

Note that the keychain is somewhat limited in the iOS Simulator. You may find it easier to work on this assignment using an actual iOS device.

In order to successfully complete this assignment, you must successfully implement all of the Functional Requirements below. As with all apps that you submit for this class, you should also:

- Ensure that your app does not crash or behave in an unstable manner
- Make use of good design patterns (e.g. MVC) and structure your app accordingly
- Follow the Human Interface Guidelines when designing your app's UI
- Test your app on an actual device running the latest version of iOS

Functional Requirements

For this assignment, you will create a basic password manager for Internet sites. Your app should open to an empty table view (the password list) with a basic navigation bar header; in the header, provide a simple title (such as "Passwords") and a system-standard Add button in the upper right.

Upon pressing the Add button, the app should display a password editor view with three text fields, all labeled: one for the hostname, one for the username, and one for the password. For new items, all three fields should be blank. This editor view should also display a navigation bar with a Cancel button in the upper left and a Save button in the upper right.

If the user taps Cancel, make no changes and hide the password editor. If the user taps Save, however, your app should create a new keychain item of the Internet Password class. This item should save the hostname and username as attributes and the password as data. Upon saving, the app should hide the password editor and update the password list.

Once there are items in the keychain, the password list should, for each keychain item, display the hostname along the left and the username along the right. The password list should not display the password itself. When the user taps a row in the password list, the app should show the password editor with its data (hostname, username, and password) all populated in the editor's text fields.

When the user is editing an existing keychain item, the editor should show the same Cancel and Save items. Upon canceling, the app should discard any changes made to the keychain item; upon saving, the app should update the keychain item being edited. Either way, the app should return to the list.

You may assume that the app is only storing a single username/password combination per hostname. In other words, the hostname may be used as a uniqued attribute for your keychain queries. (You may decide what to do if the user attempts to add a new keychain item for an existing hostname.)

Bonus Opportunities

You can receive up to 50 points for implementing Touch ID on your keychain items. Add a switch to the password editor for whether Touch ID is required for a password; then, if required, prompt for Touch ID when displaying the editor. Avoid prompting for Touch ID on the primary password list.

Submitting Your Work

To submit your work, upload a .zip file to the appropriate drop box that contains your entire Xcode project directory, including:

- Your .xcodeproj bundle and all its contents
- All your source files, including code, .xib files, and any image resources
- Any additional files that your app requires to run

Name this zip file "UWHWKeychain_*<UW NetID>*.zip", where *<UW NetID>* is the username assigned to you by UW. (For example, the instructor's submission would be named UWHWKeychain_tekl.zip.)

Your submission should compile cleanly on the first try, throwing absolutely no errors, warnings, or static analyzer problems. You may lose points if your solution does not compile cleanly.

Your submission should, once compiled, run well on an actual device running the latest version of iOS. You may choose to support past versions of iOS, but all testing will be done on the newest version available on the due date.