**[Assignment 2](https://bblearn.griffith.edu.au/webapps/blackboard/content/listContent.jsp?course_id=_48251_1&content_id=_2071246_1)**

The purpose of this assignment is to write more advanced iOS and Swift applications. The assignment consists of a programming task and a number of exercises that will be made available incrementally as milestones (so check the milestones inside this folder regularly to make sure you won't miss out on your next milestone!) Milestones are due every week and need to be submitted in or before the lab they are due in.

**Some Hints**

* Read the [UI Testing documentation](https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/RecordingUITests.html) (also available directly from within Xcode) on how to record UI interactions for testing.
* Some nice hints on user interface testing can be found on the [UI Testing Cheat Sheet](http://masilotti.com/ui-testing-cheat-sheet/)!
* Use UIAlertController with preferredStyle: .ActionSheet to display action sheet alerts.
* Use WKWebView (or UIWebView) to display web content (you can load a web page through the loadRequest() method)!  If you are using a UIWebView, make sure you set "Allow Arbitrary Loads" to "YES" for your app under "App Transport Security Settings" in order to be able to load non-secure web pages!
* In order to use WKWebView, you need to import WebKit in your project and link it against the WebKit framework!
* To create a URL request from a string, you can use the NSURLRequest(URL:) convenience initialiser, e.g.
  + NSURLRequest(URL: NSURL(string: "http://griffith.edu.au/")!)
* Saving data in property list format is easy, e.g. if you have an array of strings, you can save it using
  + array.writeToFile(fileName, atomically: true)
* Loading such data back is equally easy, e.g. if you want to create a mutable array from the saved array above, use
  + optionalArray = NSMutableArray(contentsOfFile: fileName);
* iOS uses a sandbox, so you can only load and save files from certain locations.  E.g. if you want to save to a file called data.plist, you can use

        let documentsDirectory = NSSearchPathForDirectoriesInDomains(.DocumentDirectory, .UserDomainMask, true)[0] as NSString

        let fileName = documentsDirectory.stringByAppendingPathComponent("data.plist")

* If the above does not seem to work for you for writing files, check that you have used .DocumentDirectory and **not** .DocumentationDirectory (which is read-only — this is a common mistake)!
* Core Data documents can be managed using the [managedObjectContext](http://developer.apple.com/library/ios/documentation/uikit/reference/UIManagedDocument_Class/Reference/Reference.html#//apple_ref/occ/instp/UIManagedDocument/managedObjectContext) property of [UIManagedDocument](http://developer.apple.com/library/ios/#documentation/uikit/reference/UIManagedDocument_Class/Reference/Reference.html)
* The [NSFetchedResultsController](http://developer.apple.com/library/ios/#documentation/CoreData/Reference/NSFetchedResultsController_Class/Reference/Reference.html) is a controller class that provides all the necessary methods you can use to update a table from a [UITableViewController](http://developer.apple.com/library/ios/#documentation/uikit/reference/UITableViewController_Class/Reference/Reference.html) subclass!
* more hints to come -- check back regularly!