iOS Development Accelerator
Week 2 Day 4

- UISearchBar
- UIWebview
- NSUserDefaults
- Intro to CoreData

UISearchBar

- Very similar to UITextField
- Delegate methods for when the search and cancel buttons are clicked
- also delegate methods for whenever the text in the search bar is changed if you want to be validating it.
- Same didBeginEditing and didEndEditing methods like the textfield
- Can be embedded inside a tableview for easy interface layout

UIWebView

- “Use the UIWebView class to embed web content in your application”
- The simple workflow of a web view:
  1. add web view to the view hierarchy
  2. send it a request to load web content
- Can have a delegate that tracks loading of content, this will come in handy when we look into OAuth
- It’s sort of like a mini browser in your app, and you can customize it to not allow the users to go back or forward.

NSUserDefaults

- “NSUserDefaults allows an app to customize its behavior based on user preferences”
- Think of it as an automatically persisting plist that is easily modified in code.
- Use the standardUserDefaults class method to return the shared defaults object.
- Setting values inside of it is as easy as these methods:
  - setBool:ForKey:
  - setObject:ForKey:
  - SetInteger:ForKey:

CoreData

- Core Data is a framework designed to generalize and automate common tasks associated with object life-cycle and persistence.
- Why use Core Data? Apple claims app your model layer will have 50% to 70% less code when using core data
- Core Data itself is not a database, its a way to easily allow your application to harness the power of a database. Please refer to Kirby for all database related questions.

NSManagedObjectModel

- Describes a Core Data database schema:
  - Entities (objects)
  - Attributes (object properties)
  - Relationships (has_many, belongs_to, etc.)
  - Validation (e.g. regex for email address)
  - Storage rules (e.g. separate file for binary data)
- Take special consideration when updating an app’s schema

NSManagedObjectContext

- The link between your code and the database
- Manages a collection of `NSManagedObjects`
- `NSManagedObjects` exist in a single `NSManagedObjectContext`
- Best practice is to create a primary (mainQueue) and a secondary (background queue) context

NSPersistentStoreCoordinator

- Sits between `managedObjectContext` & persistent store (on disk)
- Persists objects to disk, reads objects from disk
- Has it’s own version of the managedObjectModel
- Can automatically migrate your existing database to a new schema*

*sometimes

UIManagedDocument

- Creates a managed document with a built-in:
  - NSManagedObjectContext
  - NSPersistentStoreCoordinator
  - Document container file (written to disk at a file path of your choice)
- Uses a background thread MOC by default
- Can create a mainQueue MOC/PSC with configuration options