GENERAL ASSEMBLY

MOBILE APP DEVELOPMENT CRASH COURSE

Tripta Gupta, Web Development Instructor at General Assembly
Stephanie Szeto, Associate Product Manager at LiveRail
David Ladowitz, Software Engineer at Omada Health

AGENDA

- Introductions
- The Apple Way
- App Design Overview
- Model-View-Controller Structure
- X-Code Walkthrough
- Objective-C Overview
- Storyboards
- TableViews, Segues, WebViews
- Building the Social Links Tab
- Buttons, APIs, AlertViews
- Building the Contact Me Tab
- Build an App!

INTRODUCTIONS

INTRODUCTIONS

- Name
- Tech Background - Programming Language
- What do you want to learn from this class?
- Favorite App
- App Idea

THE APPLE WAY

THE APPLE WAY

- Why develop for iOS?
- Objective-C & Swift
- Closed Source
- XCode, Interface Builder
- iOS Human Interface Guidelines - https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html#//apple_ref/doc/uid/TP40006556
- Apple Approval Needed - App Submission

IOS DEVICES

- **Devices** – http://en.wikipedia.org/wiki/List_of_iOS_devices

IOS HUMAN INTERFACE GUIDELINES

- https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html#//apple_ref/doc/uid/TP40006556

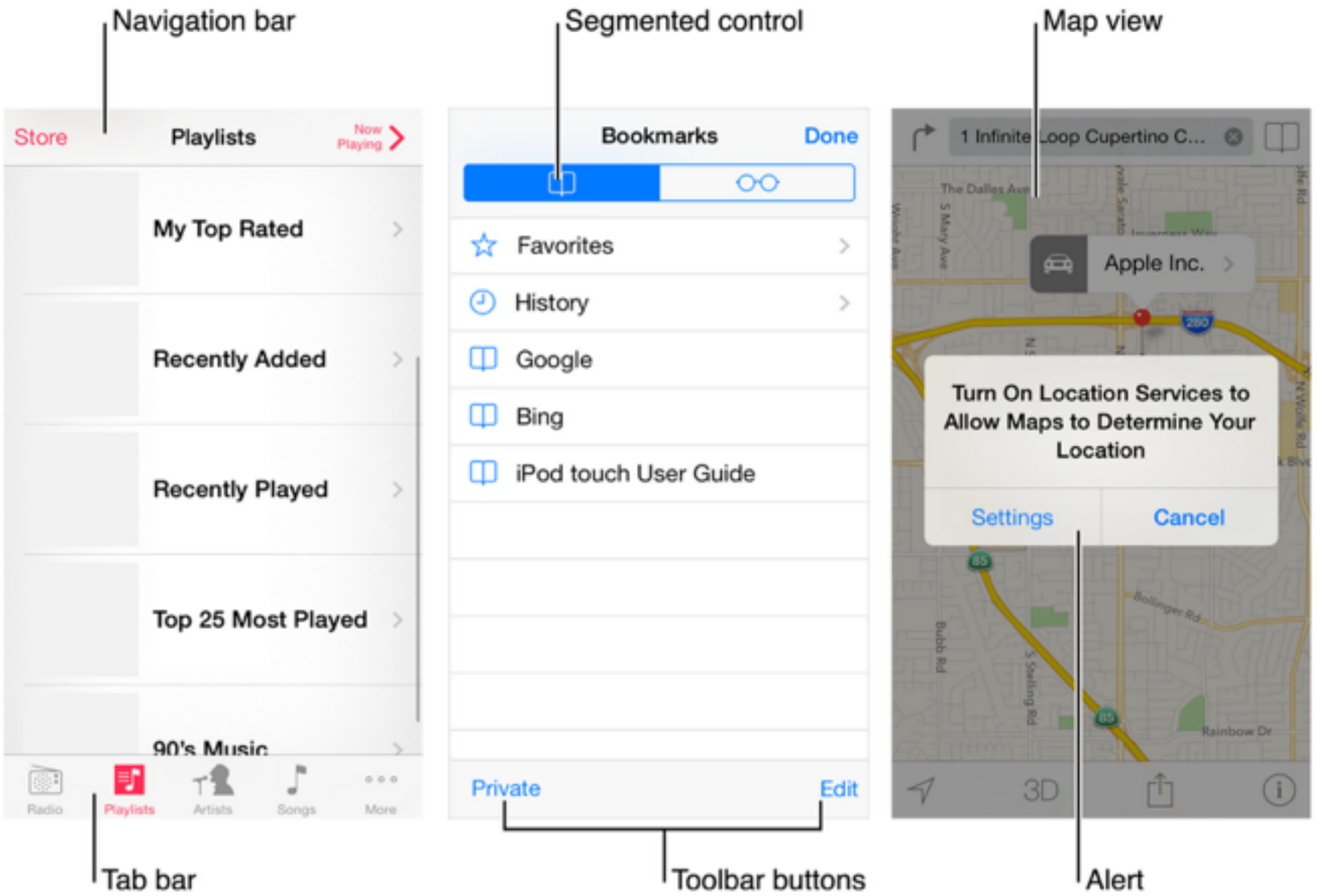
  iOS 7 embodies the following themes:

  1) **Deference.** The UI helps users understand and interact with the content, but never competes with it.

  2) **Clarity.** Text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design.

  3) **Depth.** Visual layers and realistic motion impart vitality and heighten

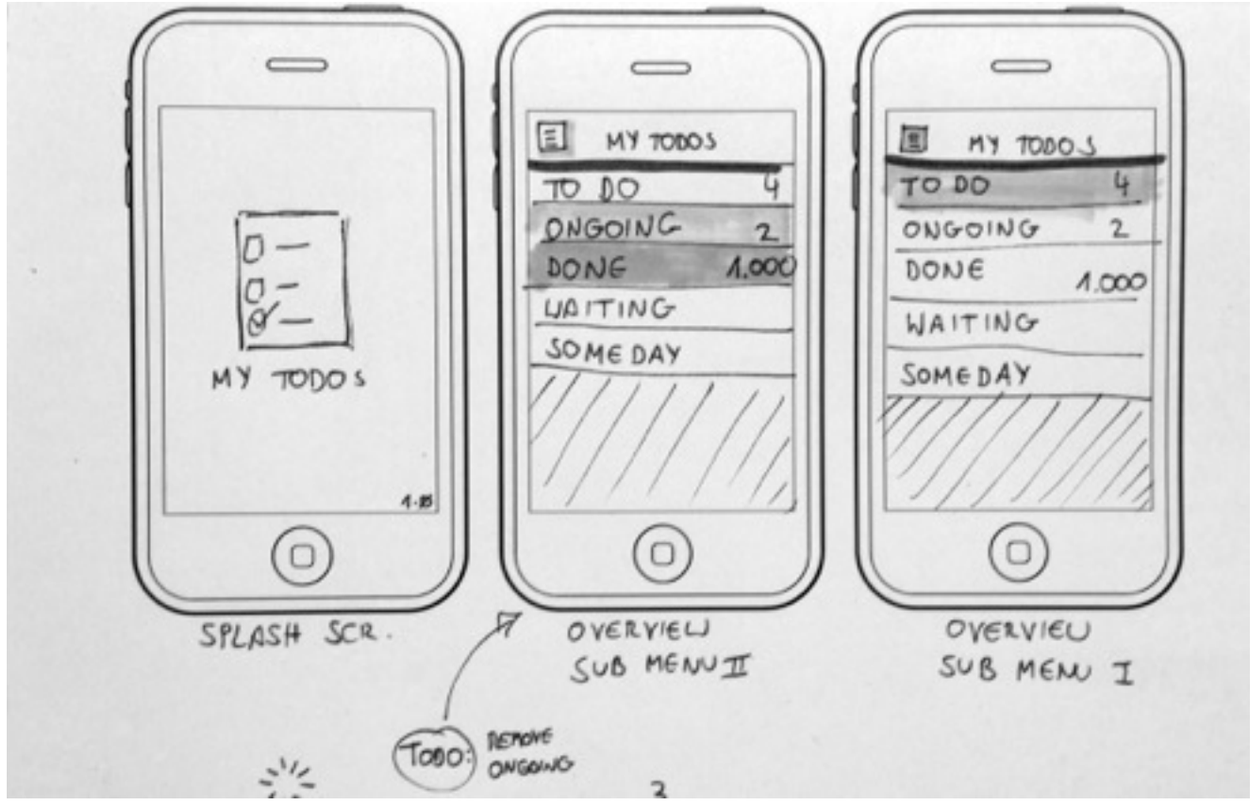
IOS APP ANATOMY



APP DESIGN OVERVIEW

APP DESIGN OVERVIEW

- Multi-faceted – graphic design, UI, UX, software engineering
- Design phase – sketches, storyboards
- Translate designs into software architecture



APPLE FRAMEWORKS

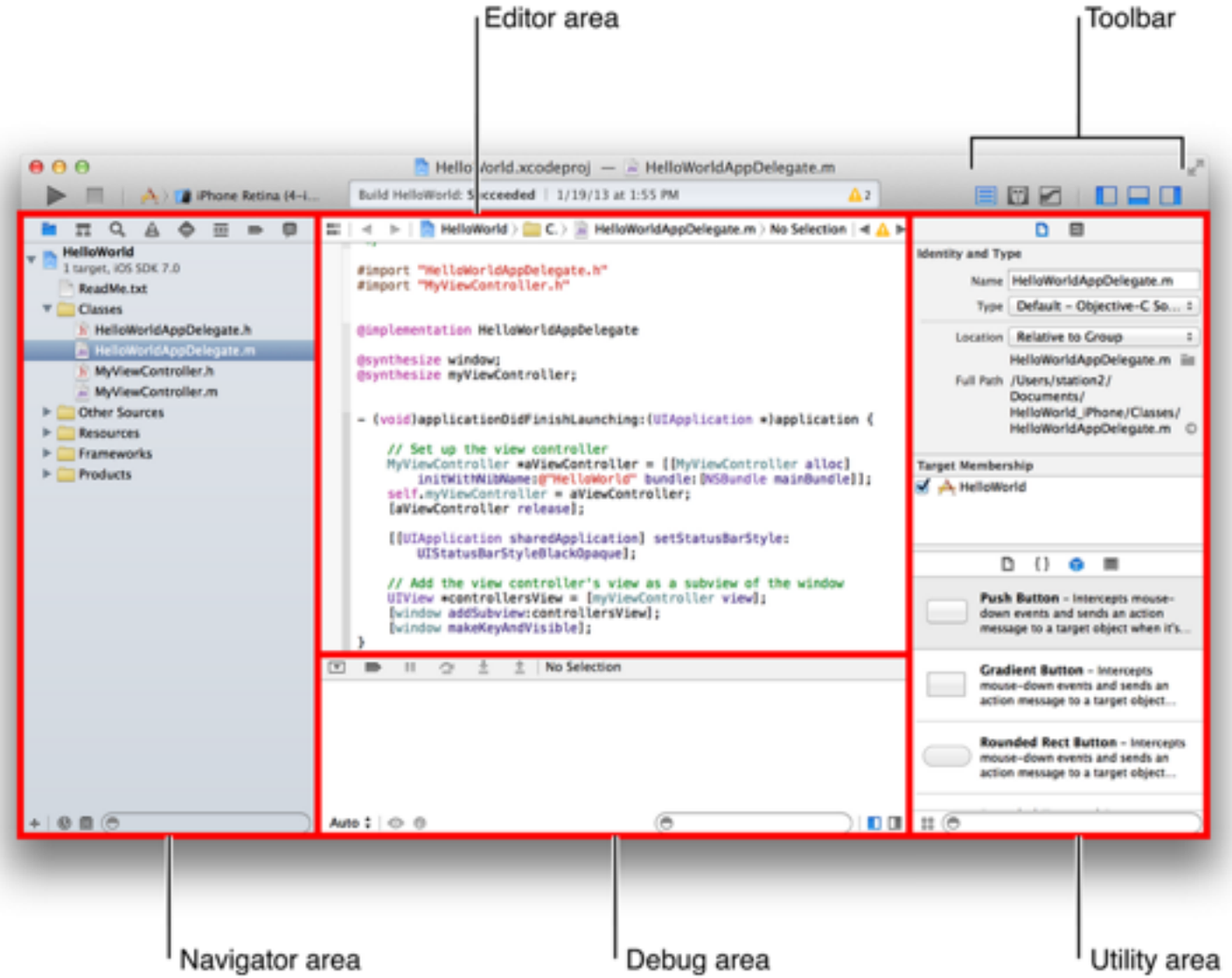
APPLE'S FRAMEWORKS/LIBRARIES

Framework	Description
Foundation	Defines core object-oriented data types like strings, arrays, dictionaries, etc. We'll explore the essential aspects of this framework in the Data Types module.
UIKit	Provides dozens of classes for creating and controlling the user interface on iOS devices.
AppKit	Same as UIKit, but for OS X devices.
CoreData	Provides a convenient API for managing object relationships, supporting undo/redo functionality, and interacting with persistent storage.
MediaPlayer	Defines a high-level API for playing music, presenting videos, and accessing the user's iTunes library.
AVFoundation	Provides lower-level support for playing, recording, and integrating audio/video into custom applications.
QuartzCore	Contains two sub-frameworks for manipulating images. The `CoreAnimation` framework lets you animate UI components, and `CoreImage` provides image and video processing capabilities (e.g., filters).
CoreGraphics	Provides low-level 2D drawing support. Handles path-based drawing, transformations, image creation, etc.

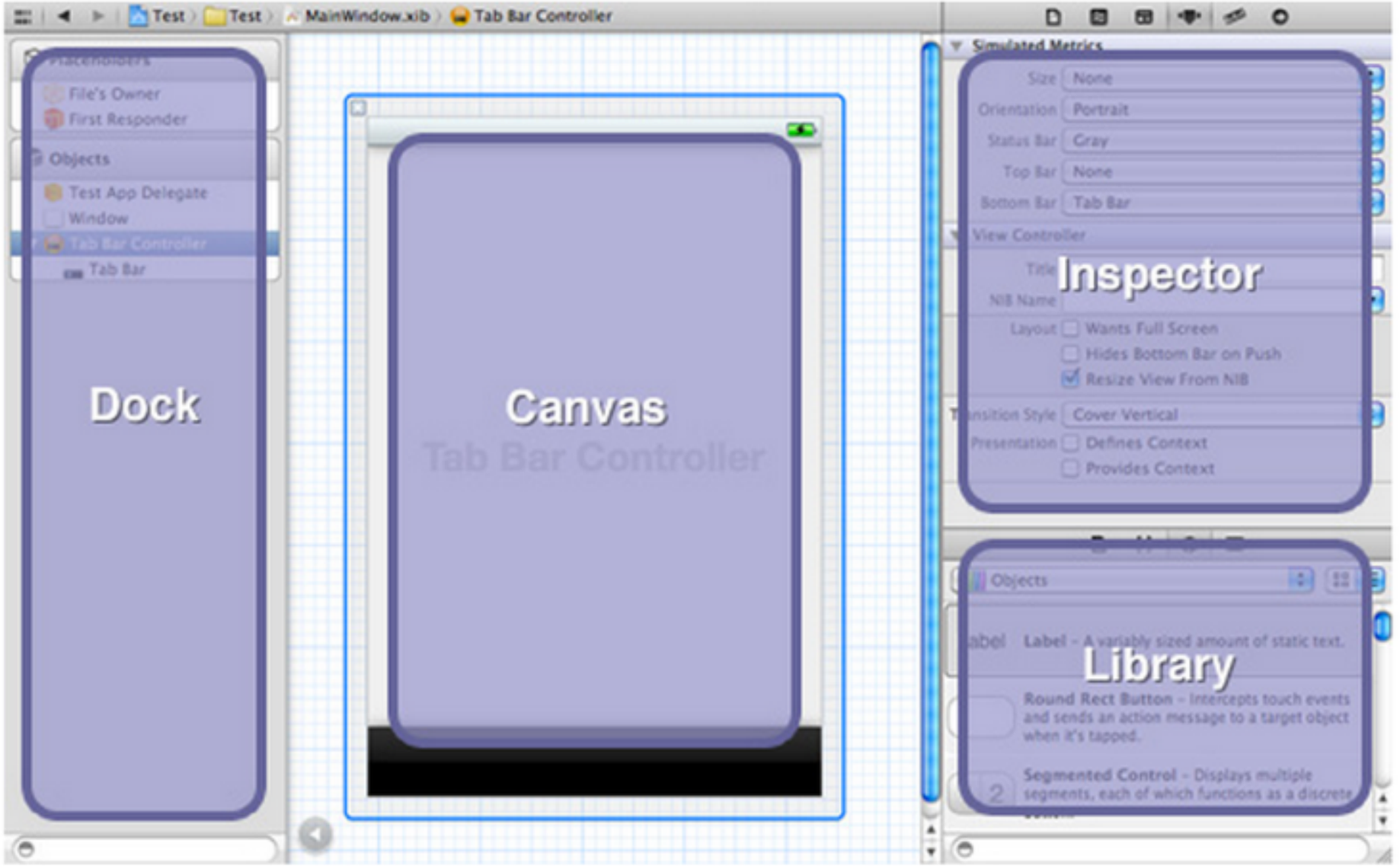
XCODE WALKTHROUGH

XCODE WALKTHROUGH

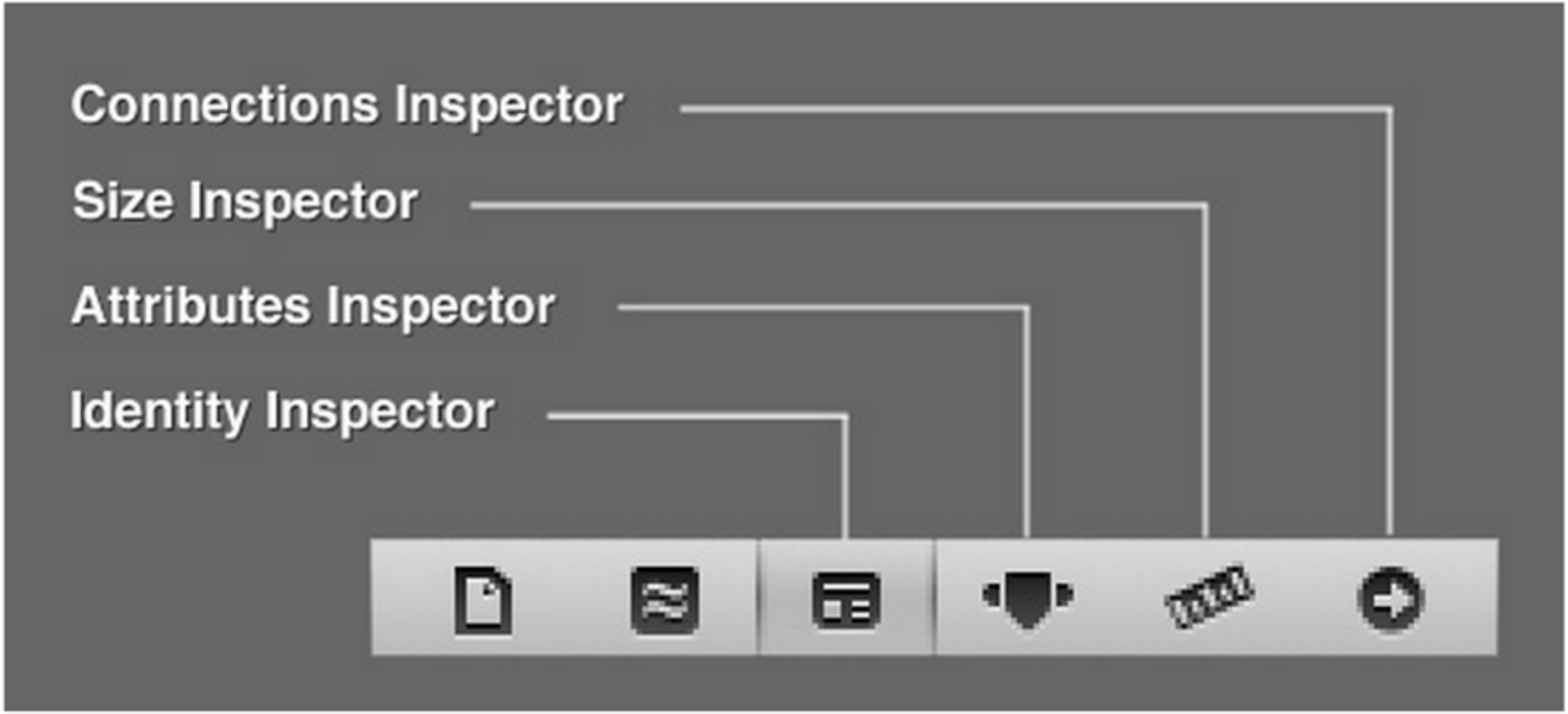
- https://developer.apple.com/library/mac/recipes/xcode_help-general/_index.html



XCODE WALKTHROUGH



XCODE WALKTHROUGH

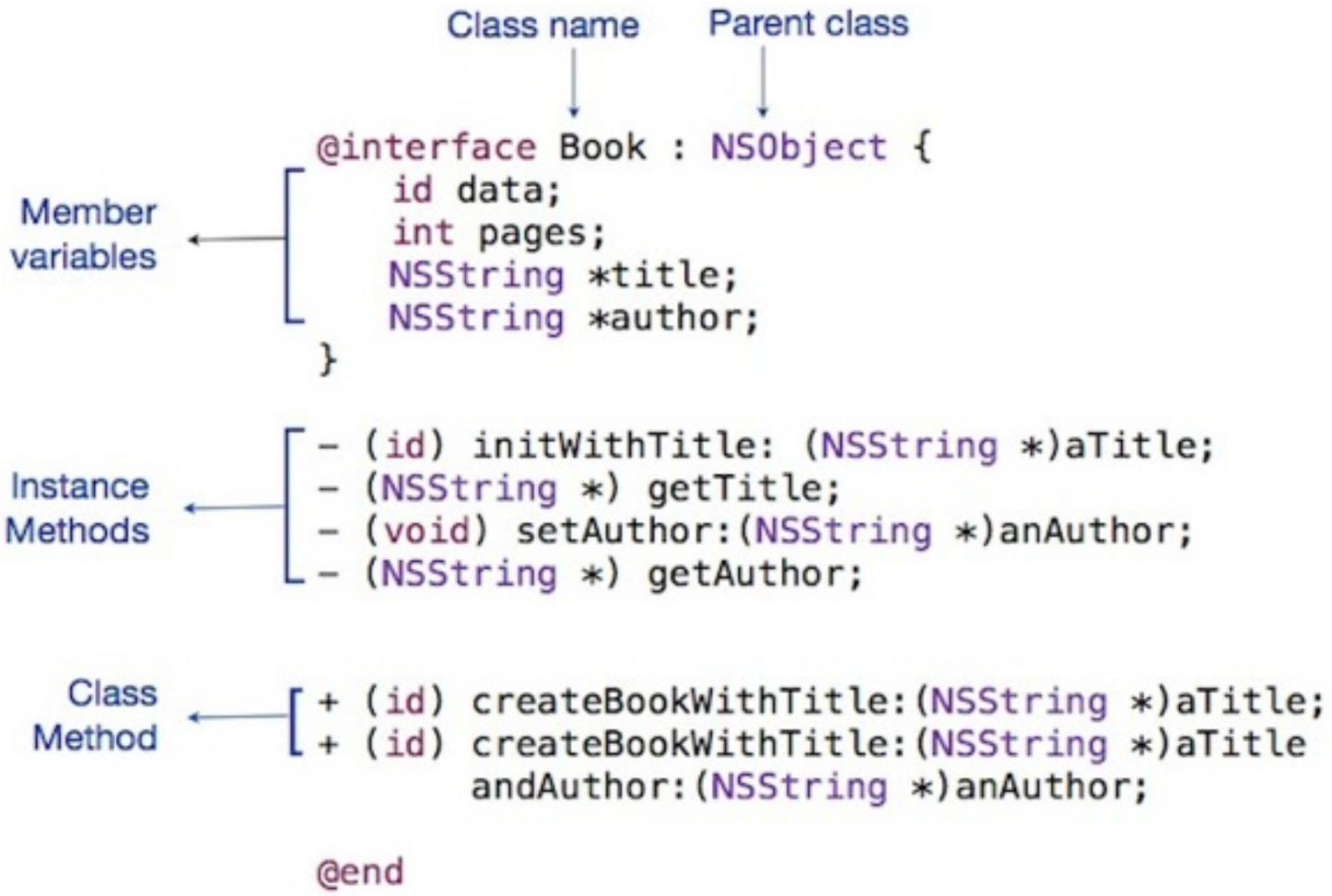


OBJECTIVE-C OVERVIEW

OBJECTIVE-C OVERVIEW

- Objective-C Syntax
- Objective-C Language Structure - Classes, Properties, etc
- Objective-C Data Types - NSString, NSArray, NSObject, NSDictionary
- Cheat Sheet - http://cdn1.raywenderlich.com/downloads/RW-Objective-C-Cheatsheet-v-1-5.pdf

OBJECTIVE-C OVERVIEW



STORYBOARDS

STORYBOARDS

- A **storyboard** is a visual representation of the app's user interface, showing screens of content and the transition between them.

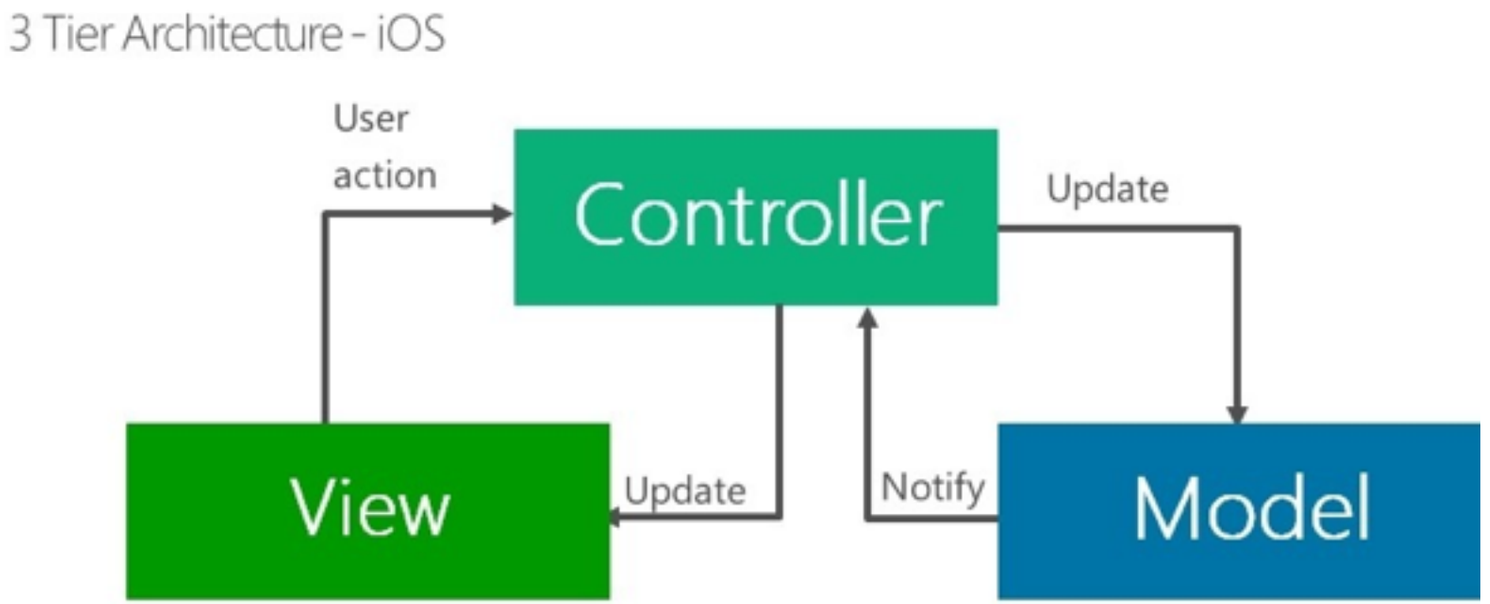
MODEL-VIEW-CONTROLLER

MODEL-VIEW-CONTROLLER

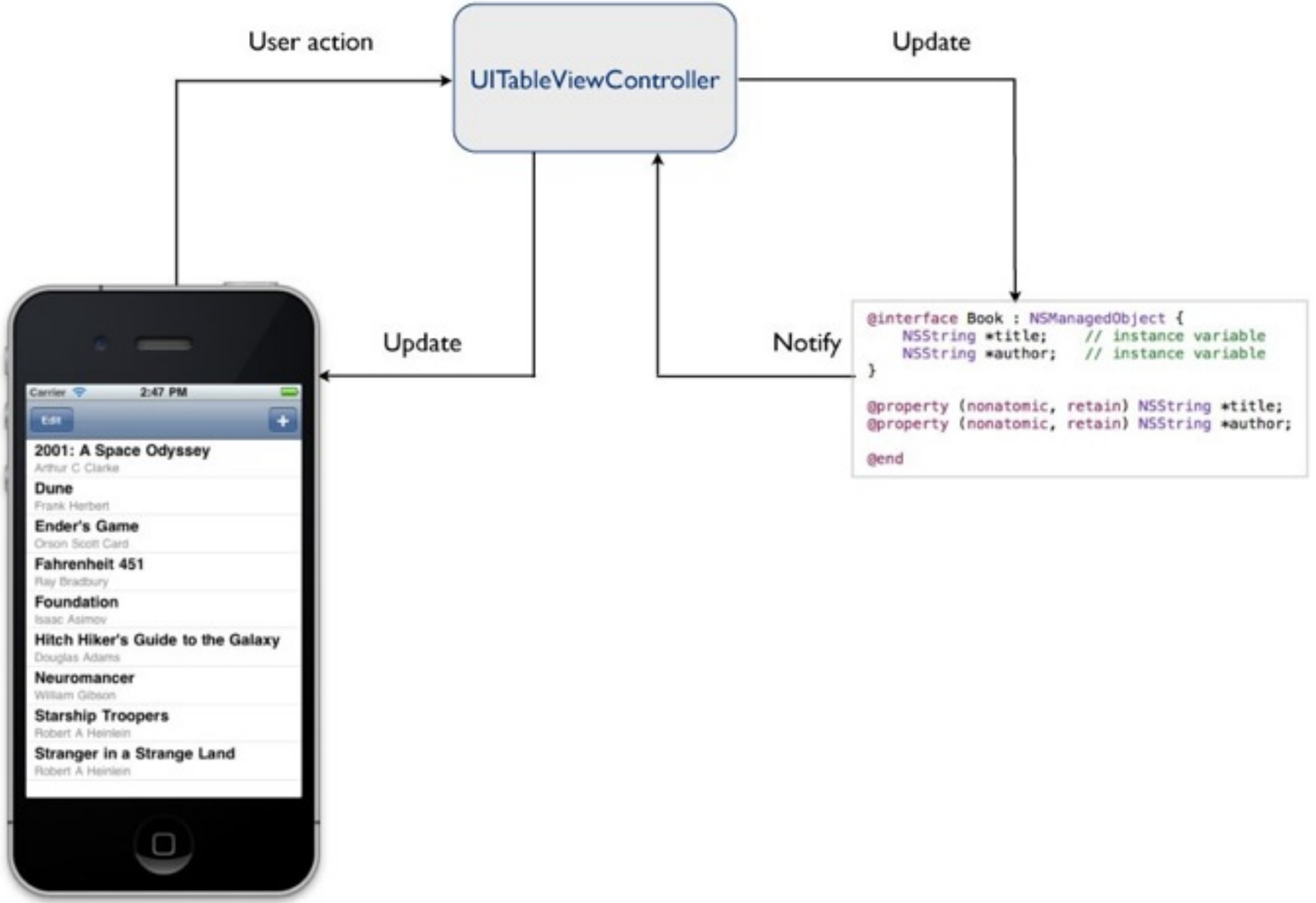
- User interfaces are comprised of **views** - https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Anatomy.html#//apple_ref/doc/uid/TP40006556-CH24-SW1
- Views can contain subviews
- Views are managed by **view controllers**
- View controllers can contain children view controllers
- View controllers interface with the app's **model**

MODEL-VIEW-CONTROLLER

- Model-View-Controller (MVC) design pattern assigns objects in an application one of 3 roles: model, view, or controller
- **Model** objects encapsulate the data specific
- A **view** object is an object in an application that users can see
- A **controller** objects acts an intermediary between one or more of an application's view objects and one or more of its model objects

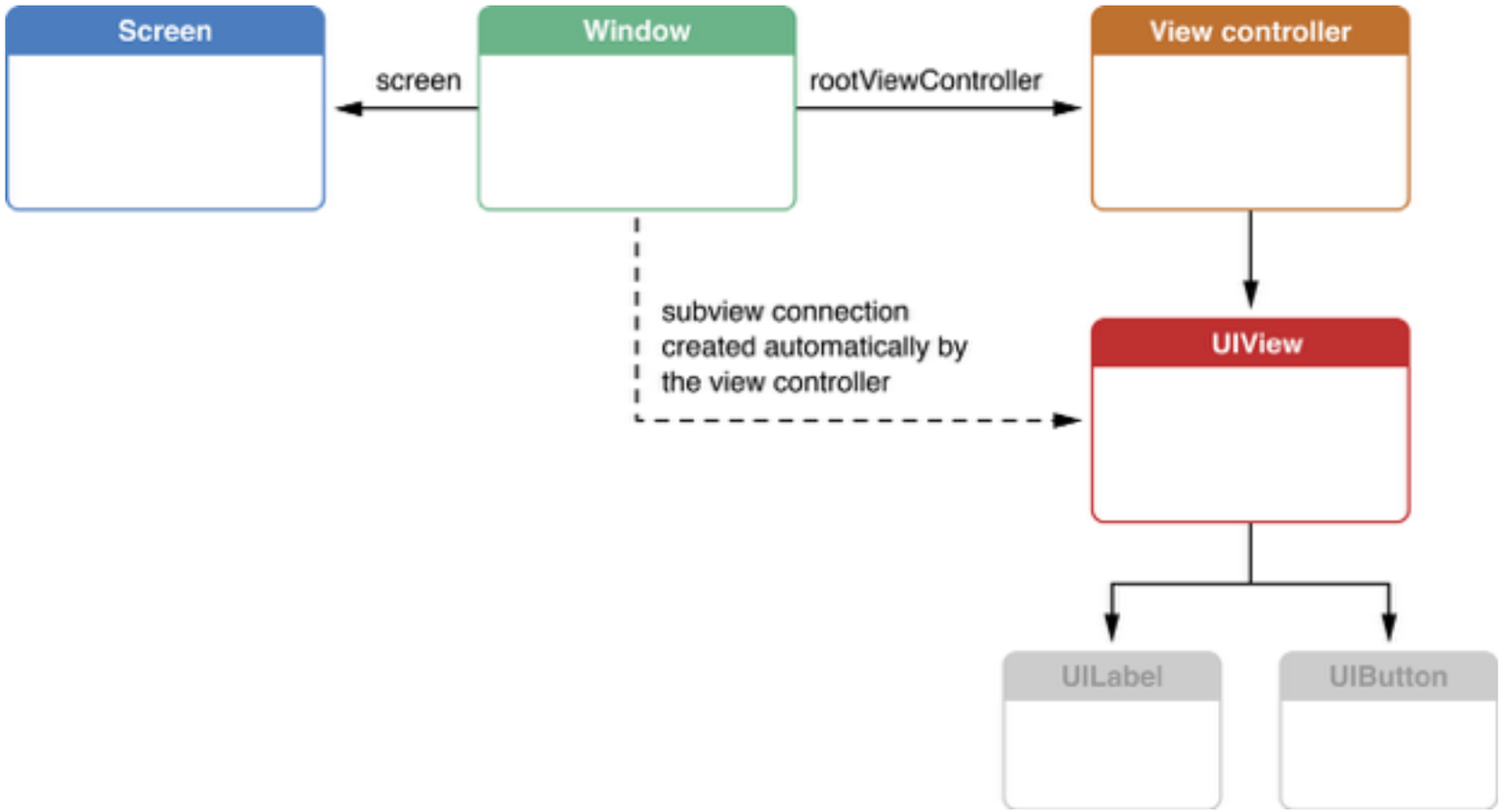


MODEL–VIEW–CONTROLLER



MODEL-VIEW-CONTROLLER

- In iOS, each view controller organizes and controls a view; this view is often the root view of a root hierarchy



VIEW CONTROLLER

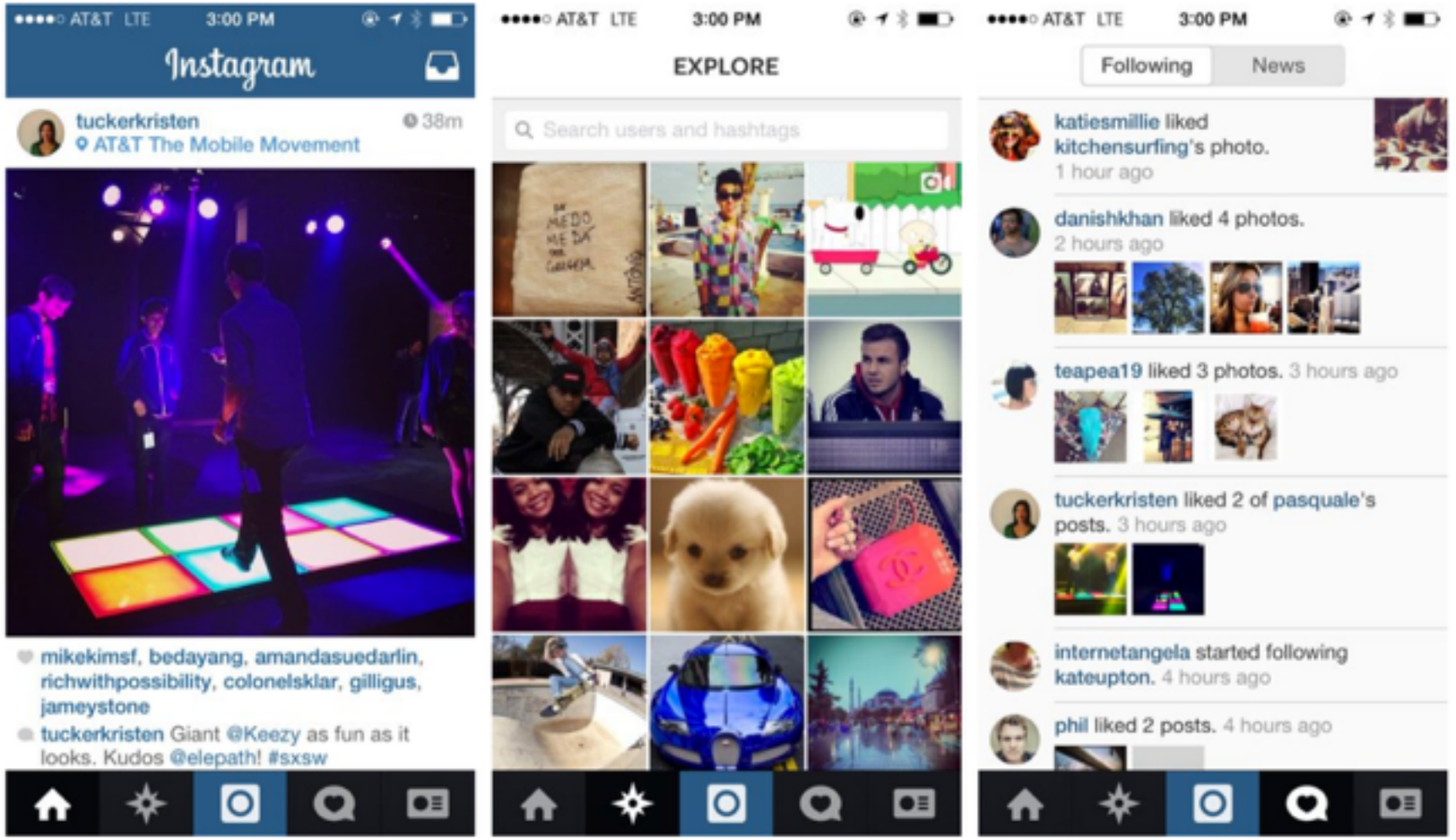
- View controllers are the backbone of an iOS application. For any given screen of an iPhone, there is generally one view controller.

- The view controller is responsible for creating the view that is displayed on the screen, as well as handling events network requests associated with that screen.

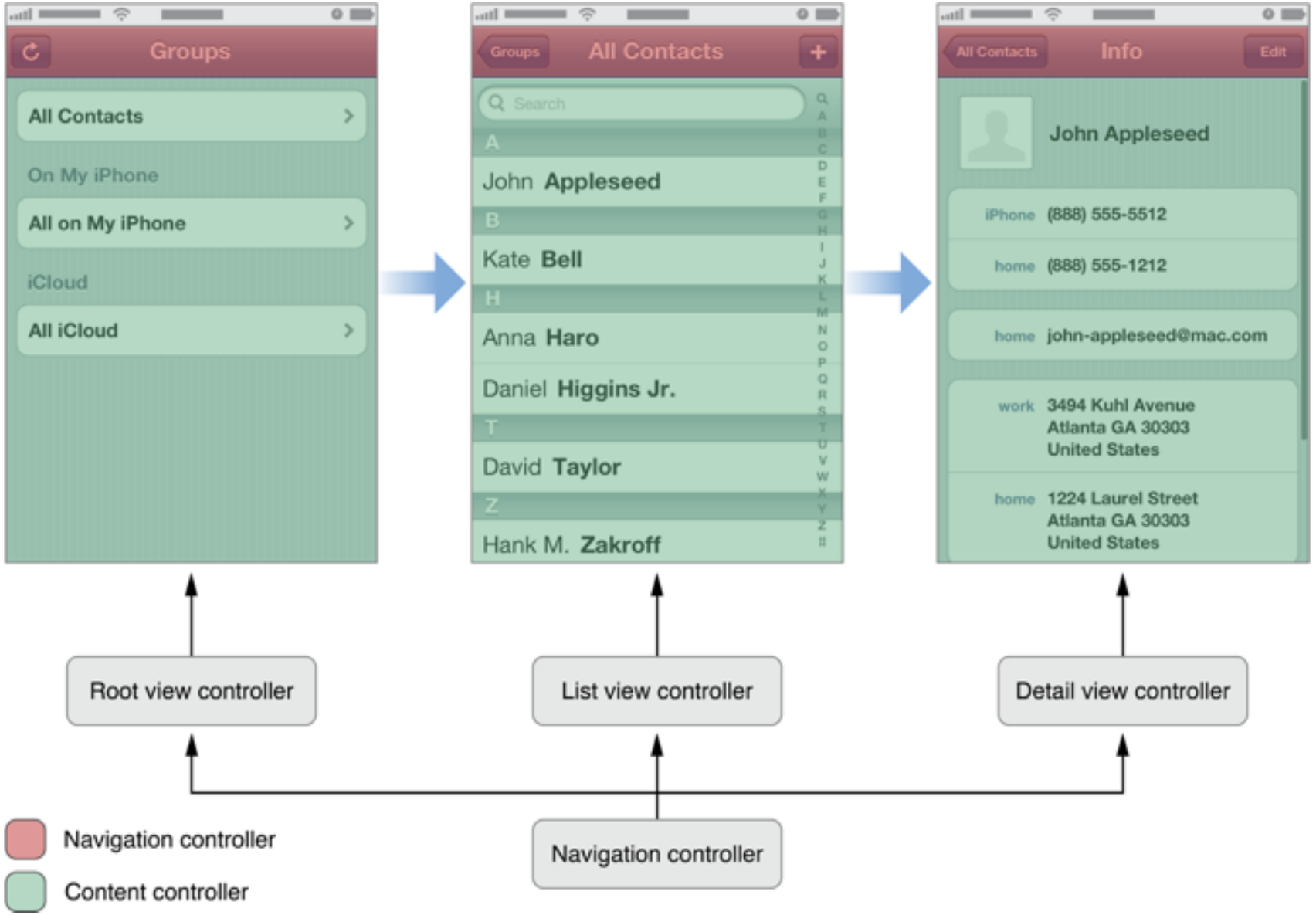
VIEW CONTROLLER

Instagram Example:

HomeViewController, TrendingViewController, & NewsViewController



VIEW CONTROLLER



APP PROJECT OVERVIEW

APP PROJECT OVERVIEW

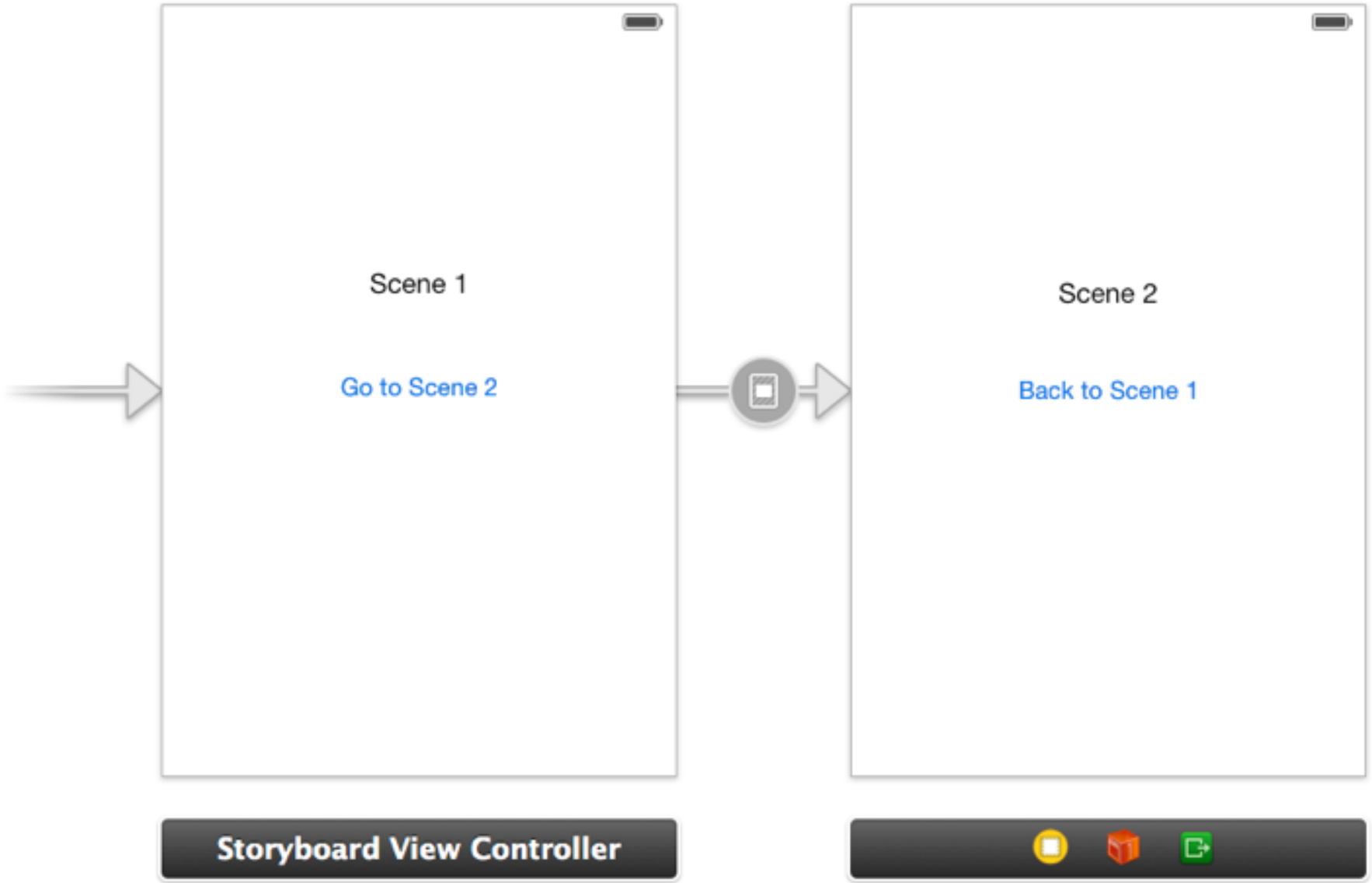
- We’re going to start by building a Resume app.

  The Resume app will consist of three sections:

  - About Me: Share your story
  - Social Links: Make it easy for others to follow you on LinkedIn, Twitter, Facebook, Github, and more
  - Projects: Impress employers with projects you’ve worked on

TABLEVIEWS, SEGUES, WEBVIEWS

SEGUES


Scene 1
Go to Scene 2
Scene 2
Back to Scene 1
Storyboard View Controller

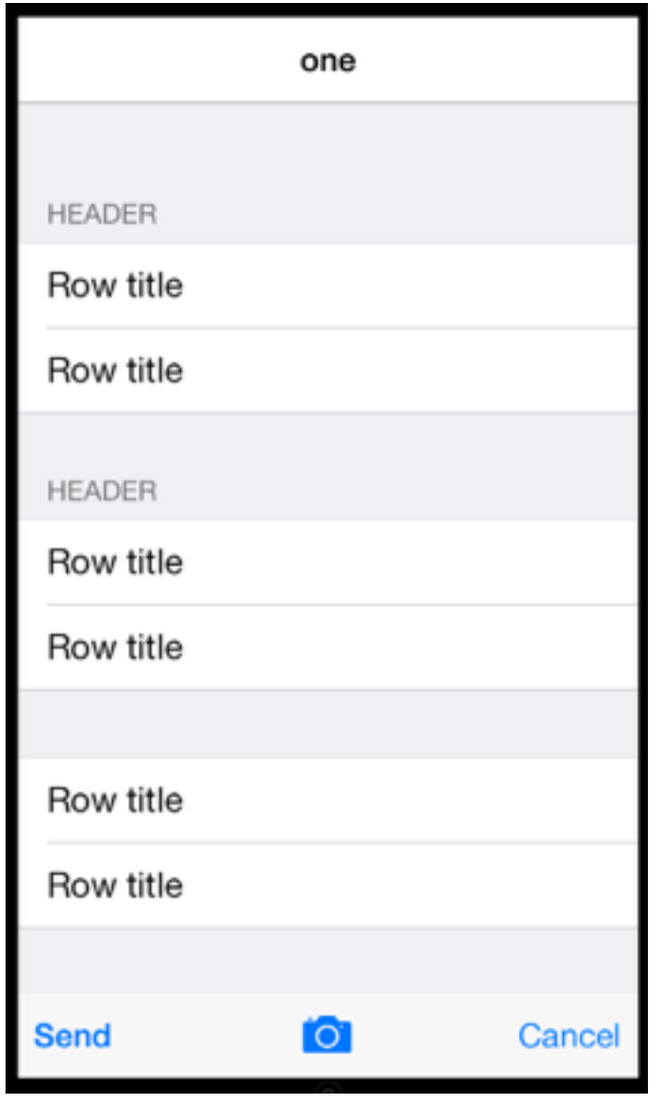
BUILDING THE SOCIAL LINKS TAB

BUILDING SOCIAL LINKS - OBJECTIVES

- Continue practicing your Storyboard skills
- Practice using TableViews
- Practice using Segues to pass information between two ViewControllers
- Practice loading URLs in a WebView

TABLE VIEW

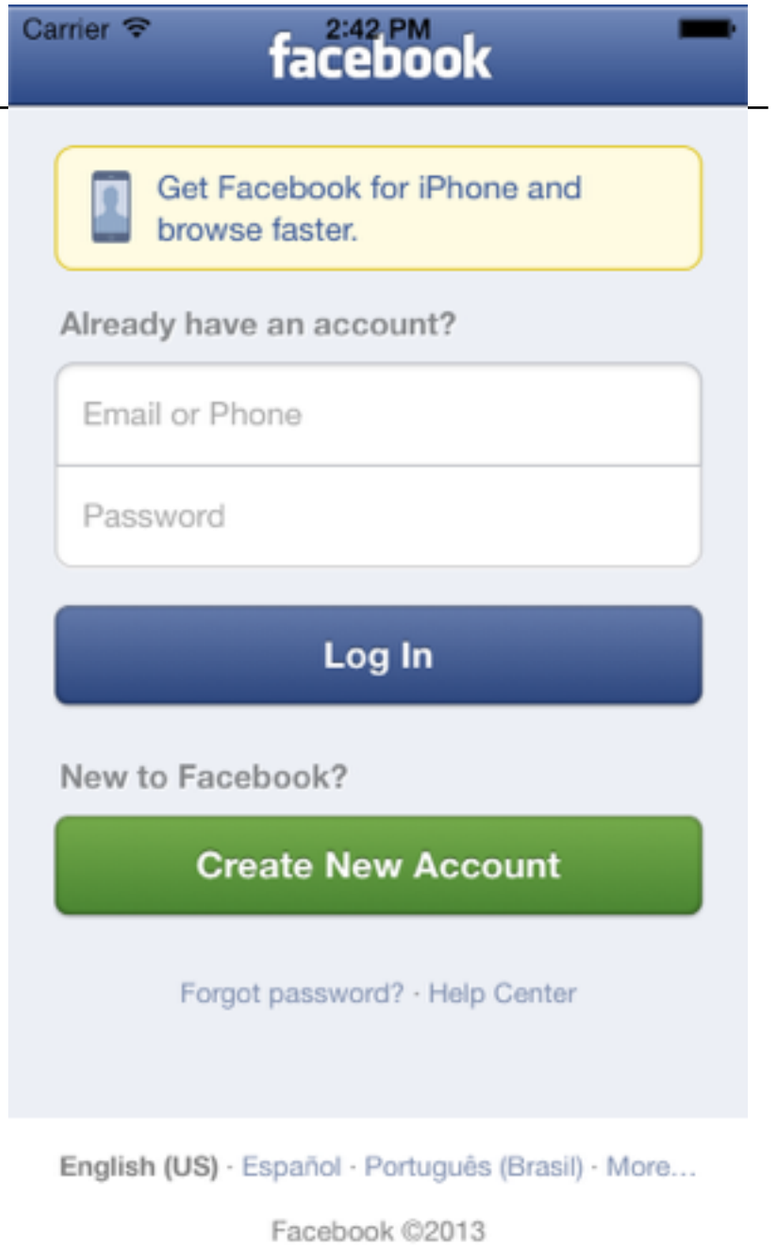
- A table view presents data in a single-column list of multiple rows.
- Can have plain or grouped table views.



WEB VIEW

- A web view is a region that can display rich HTML content.

- E.g. any app that uses authentication



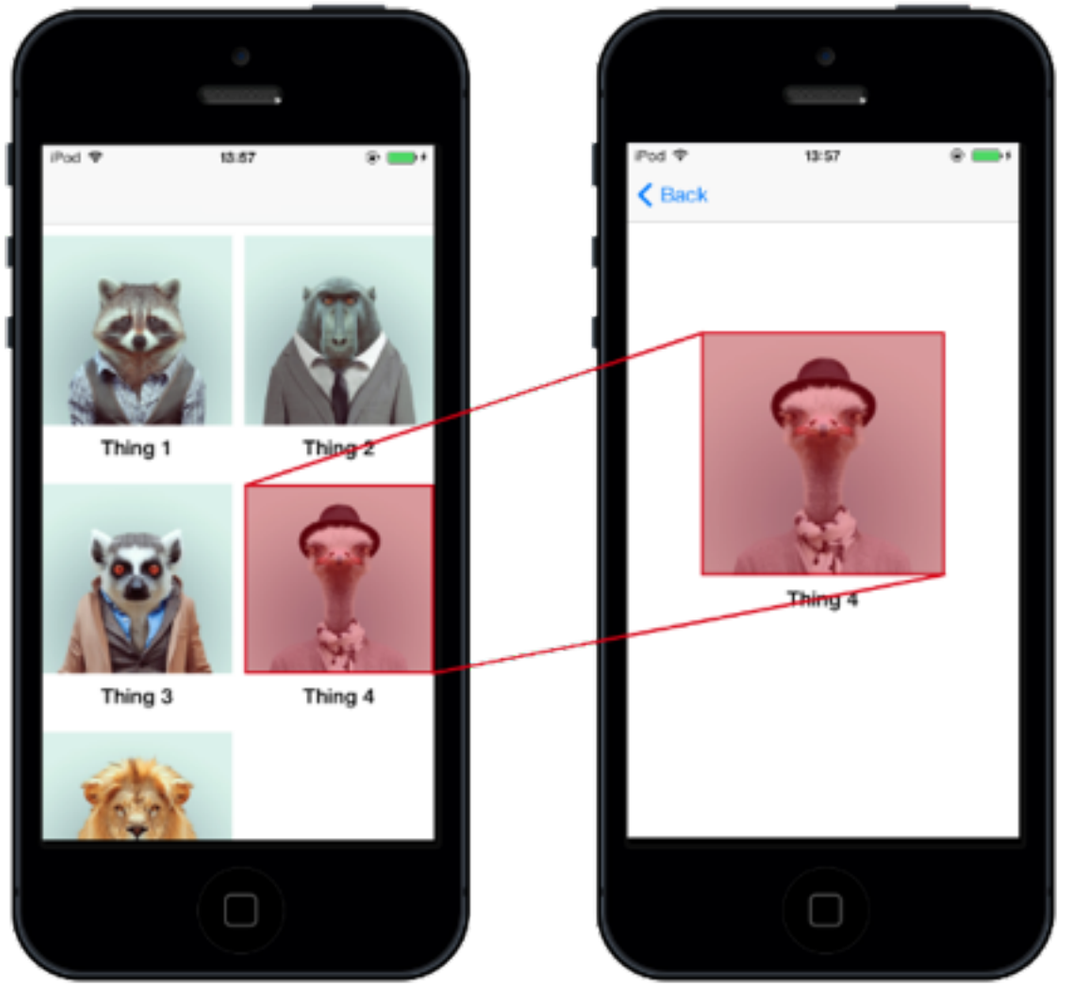
BUILDING THE PROJECT TAB

BUILDING PROJECT TAB - OBJECTIVES

- Continue practicing your Storyboard skills
- Collection Views
- Practice connecting Buttons to Actions

COLLECTION VIEW

- A collection view is a way to present an ordered set of data items using a flexible and changeable layout. The most common use for collection views is to present items in a grid-like arrangement, but collection views in iOS are capable of more than just rows and columns.
- With Collection views, you can change elements dynamically, so you can implement grids, stacks, circular layouts, dynamically changing layouts, or any type of arrangement you can imagine.



BUILD YOUR APP!!

WRAP UP

WRAP UP

- How to use Xcode
- How to use Storyboards
- Connecting Visual Storyboard Views with ViewController Code
- Reading Documentation

RESOURCES

RESOURCES

Coding:

- StackOverflow
- Ray Wenderlich Tutorials - http://www.raywenderlich.com/
- Tutsplus - http://code.tutsplus.com/categories/ios-sdk
- Apple Developer Resources - https://developer.apple.com/library/ios/navigation/

RESOURCES

Design:

- iOS 7 Design Cheat Sheet - http://ivomynttinen.com/blog/the-ios-7-design-cheat-sheet/
- iOS 7 Guides - https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html
- iOS Patterns
  - http://www.pttrns.com/
  - http://inspired-ui.com/
- Smashing Magazine - http://www.smashingmagazine.com/category/uxdesign/

Q&A

THANKS!

CONTACT INFO:

- Tripta Gupta: tripta@ga.co
- Stephanie Szeto: stephanie@liverail.com
- David Ladowitz: david.ladowitz@omadahealth.com