

# MOBILE APP DEVELOPMENT CRASH COURSE

Tripta Gupta  
Anuj Bhatia

---

# 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
- Build an App

# INTRODUCTIONS

---

---

# INTRODUCTIONS

---

- Name
- Tech Background – Programming Languages
- What do you want to learn from this class?
- Favorite App (iOS)
- App Ideas

# 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](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html#//apple_ref/doc/uid/TP40006556)
- iOS Developer Program — \$99/year
- Apple Approval Needed – App Submission

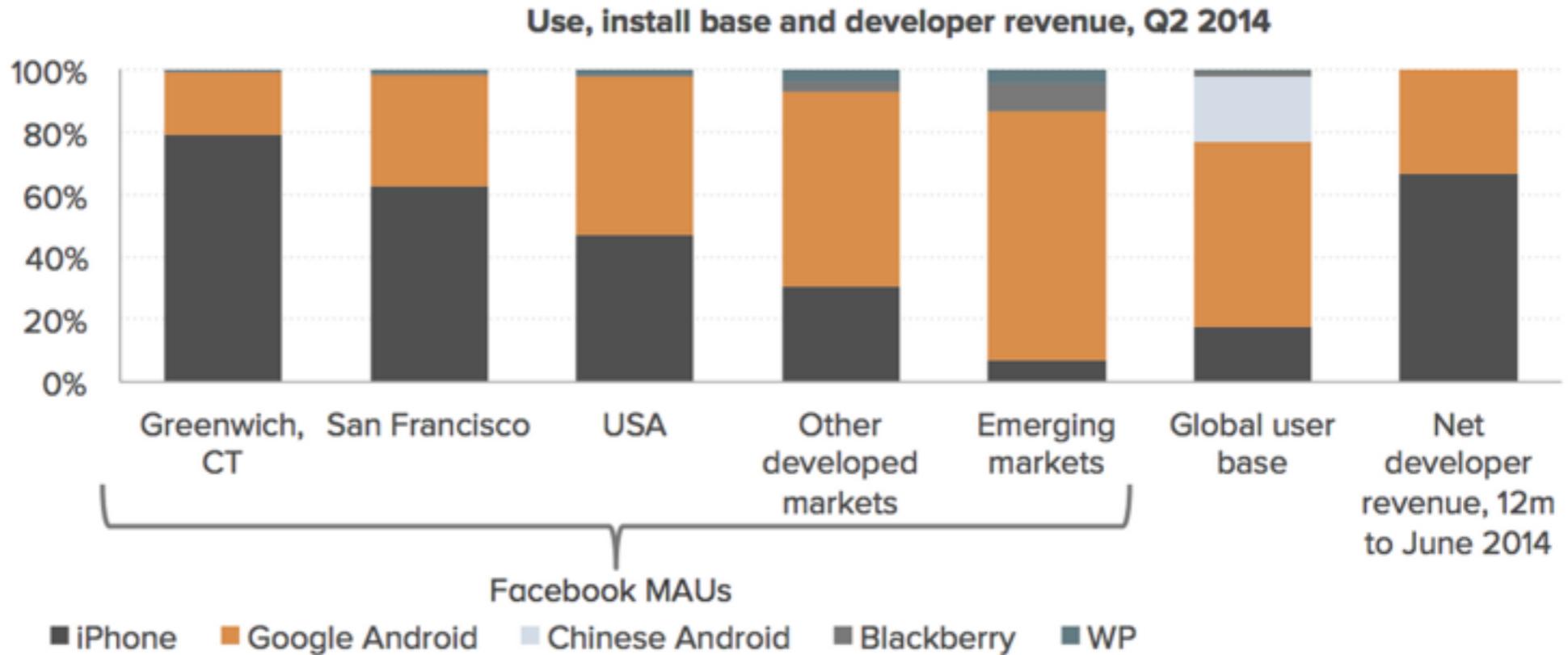
# WHY DEVELOP FOR IOS?

Android/iOS market share in the US smartphone market:



# WHY DEVELOP FOR IOS?

Android/iOS market share in the US smartphone market:

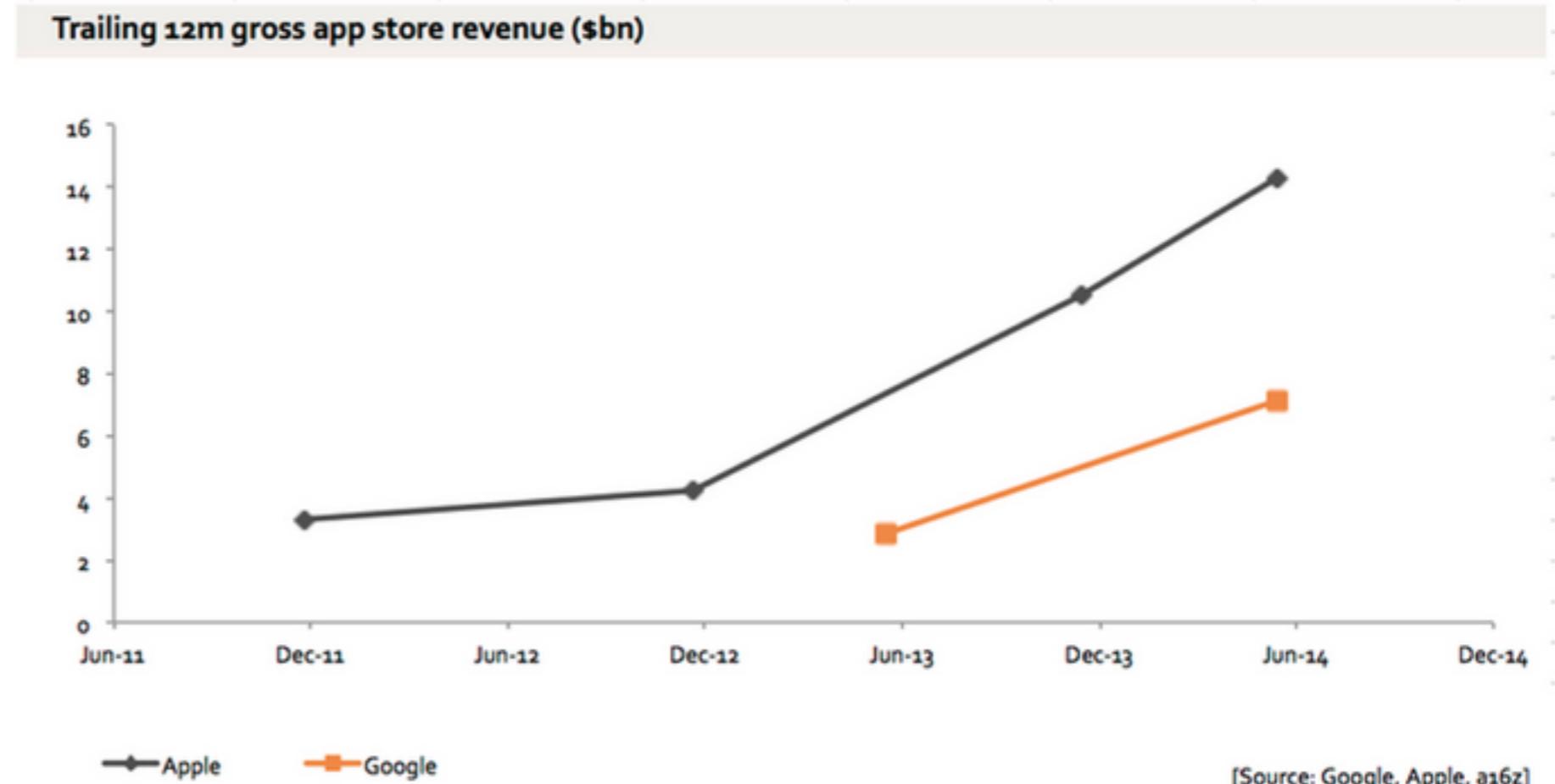


Source: Facebook, Google, Apple, a16z

# WHY DEVELOP FOR IOS?

“Apple brought in twice the **app revenue** on a little over than half the users.”

– Benedict Evans



## CROSS PLATFORM MOBILE DEVELOPMENT

---



Xamarin



Sencha



titanium™



unity



PhoneGap

# IOS DEVICES

---



iOS 8 will be compatible with:



iPhone 4s



iPhone 5



iPhone 5c



iPhone 5s



iPod touch  
5th generation



iPad 2



iPad with  
Retina display



iPad Air



iPad mini



iPad mini with  
Retina display

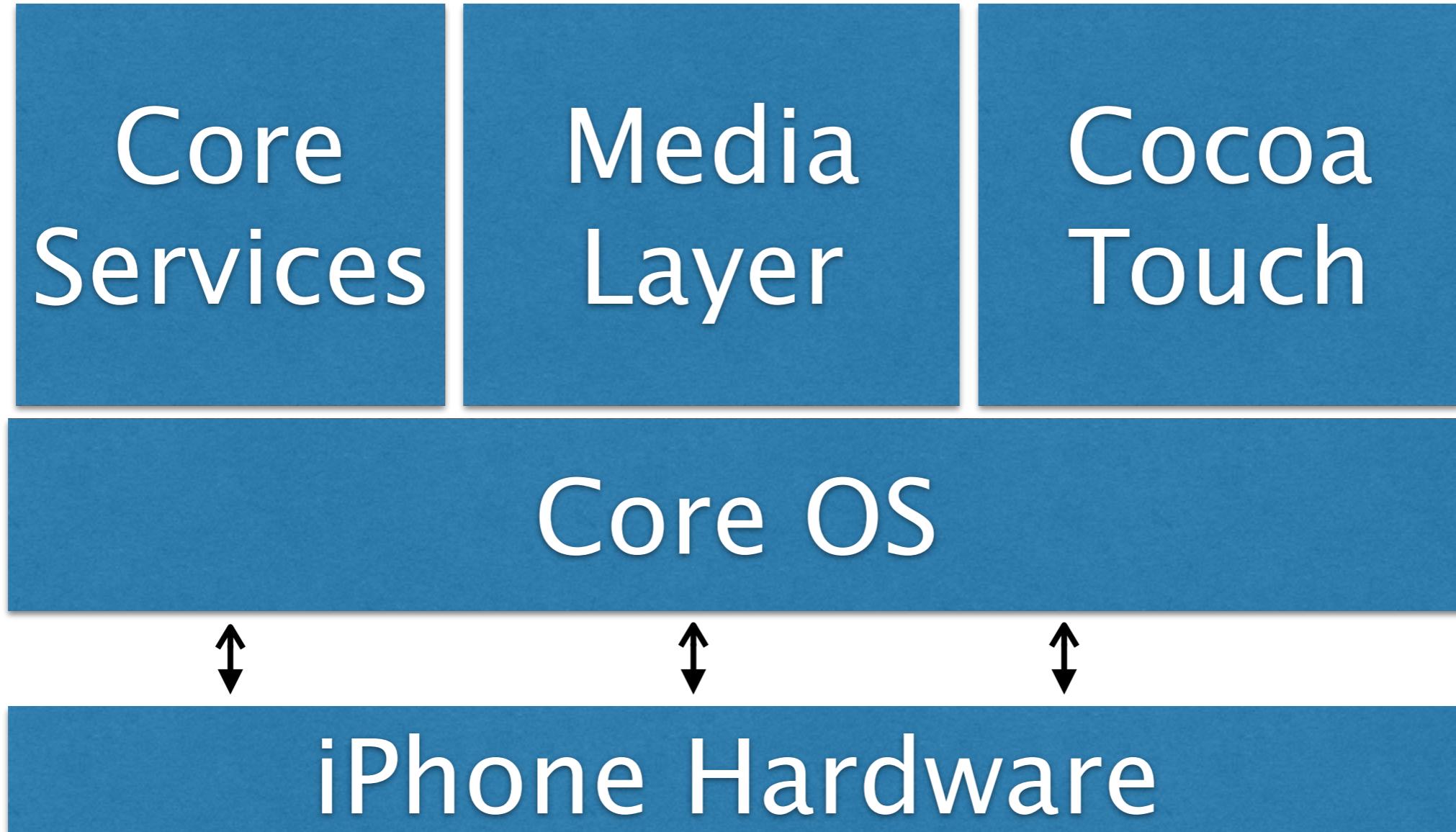
# IOS HUMAN INTERFACE GUIDELINES

---



## IOS SDK

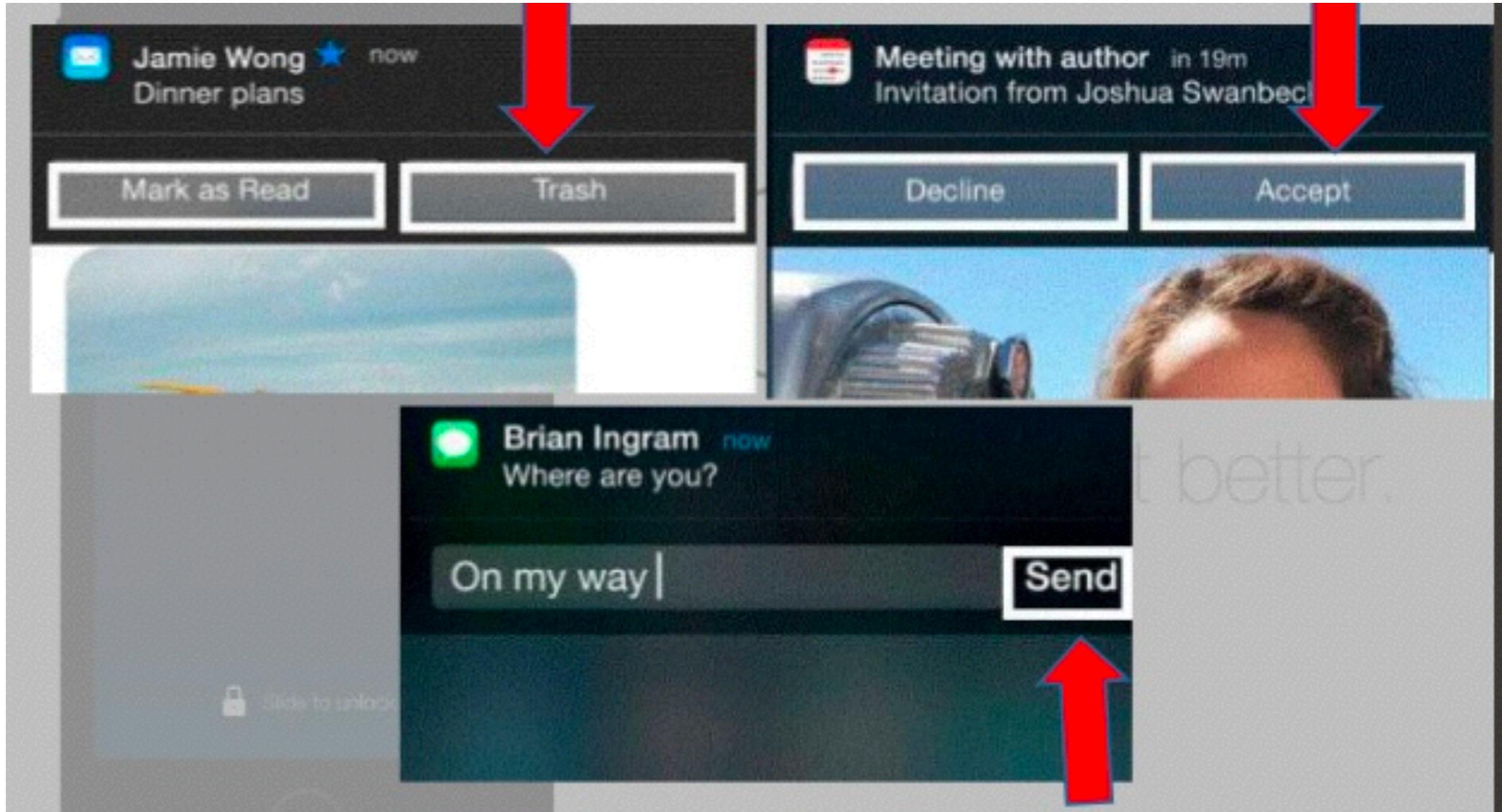
---



# NEW FEATURES IN IOS 8

---

# NEW FEATURES IN IOS 8



# NEW FEATURES IN IOS 8

---



# APP DESIGN OVERVIEW

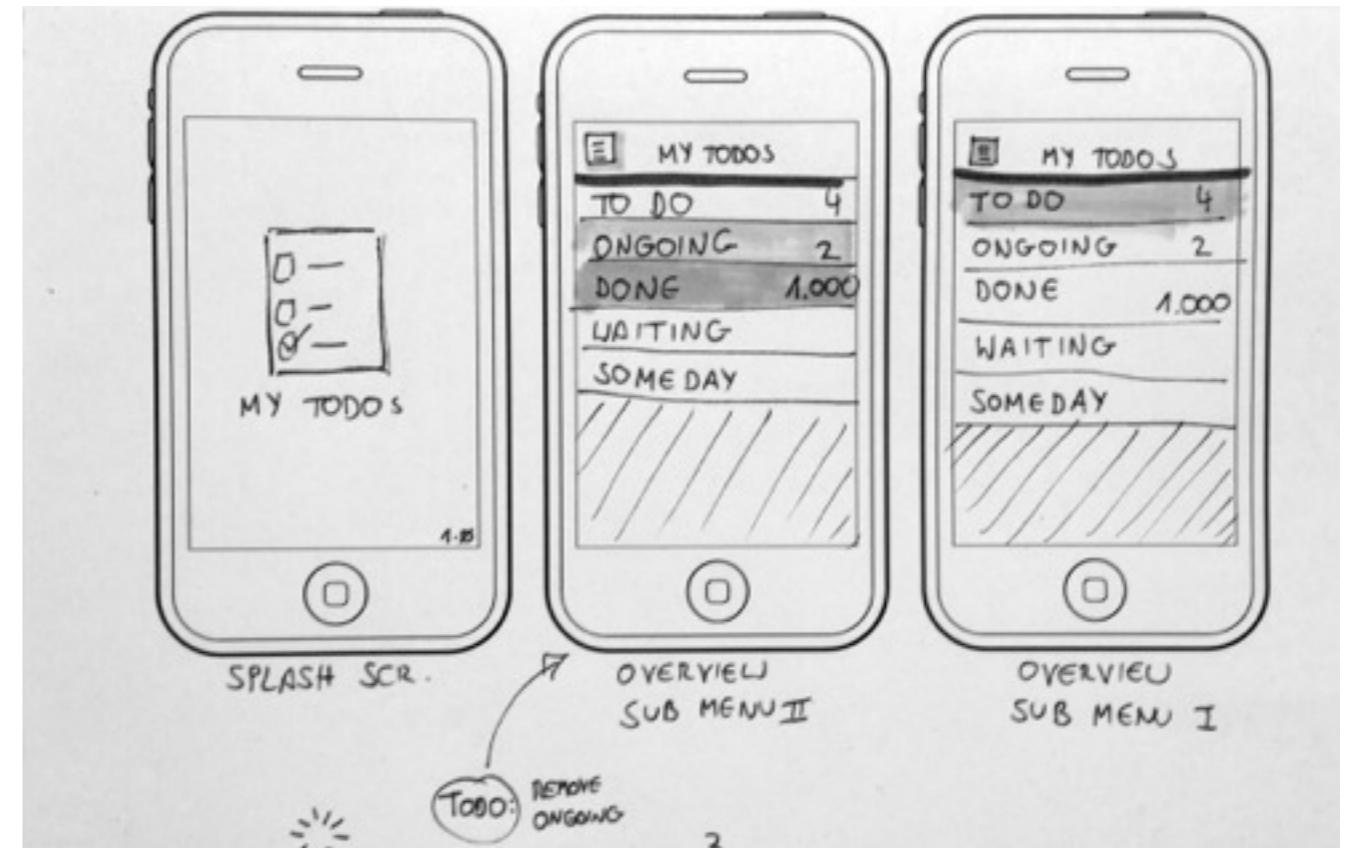
---

# APP DESIGN OVERVIEW

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

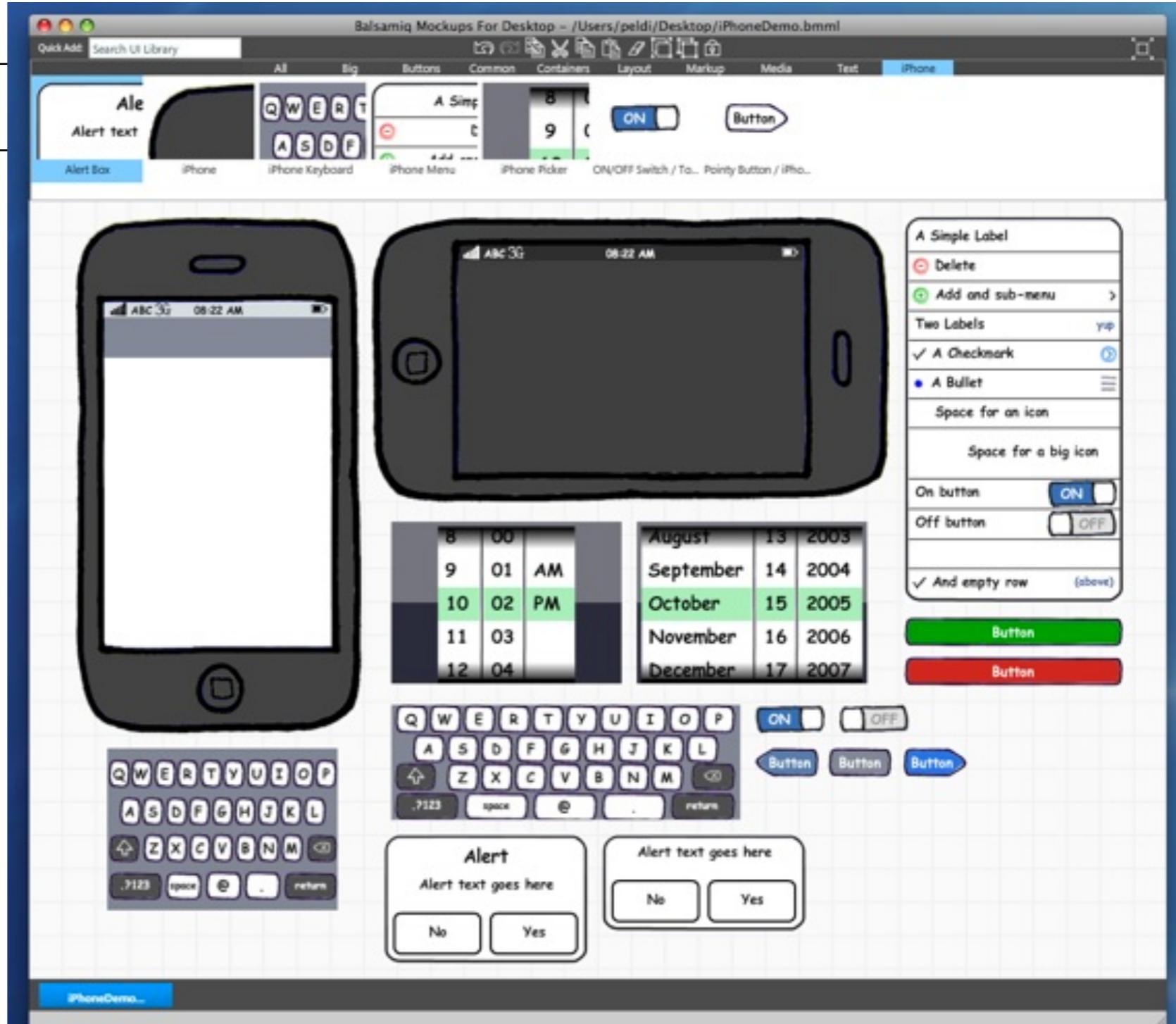
Use tools like:

- Balsamiq
- Sketch
- Spark Inspector



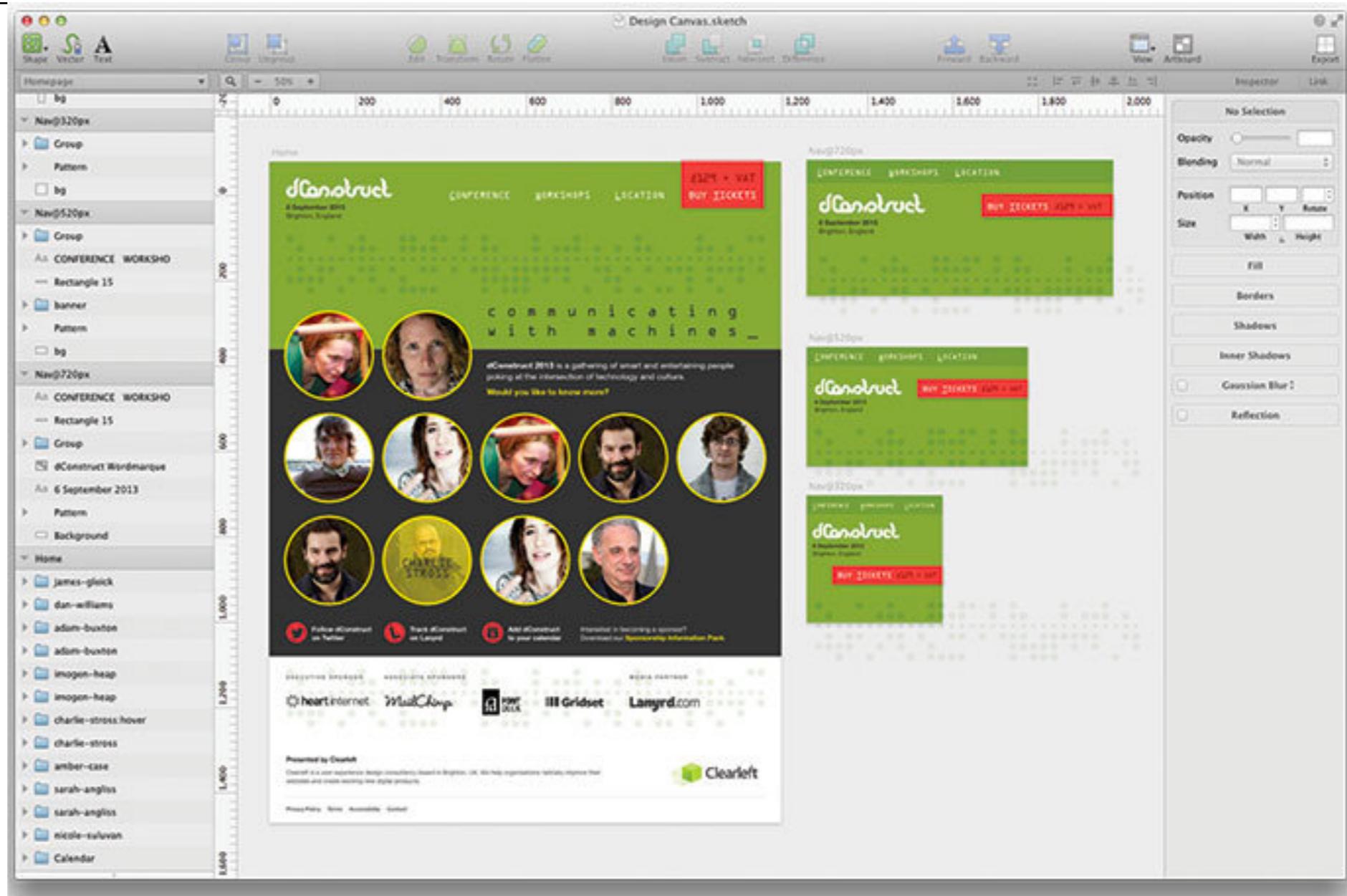
# APP DESIGN OVERVIEW

- ▶ Prototyping Tools:
  - Balsamiq

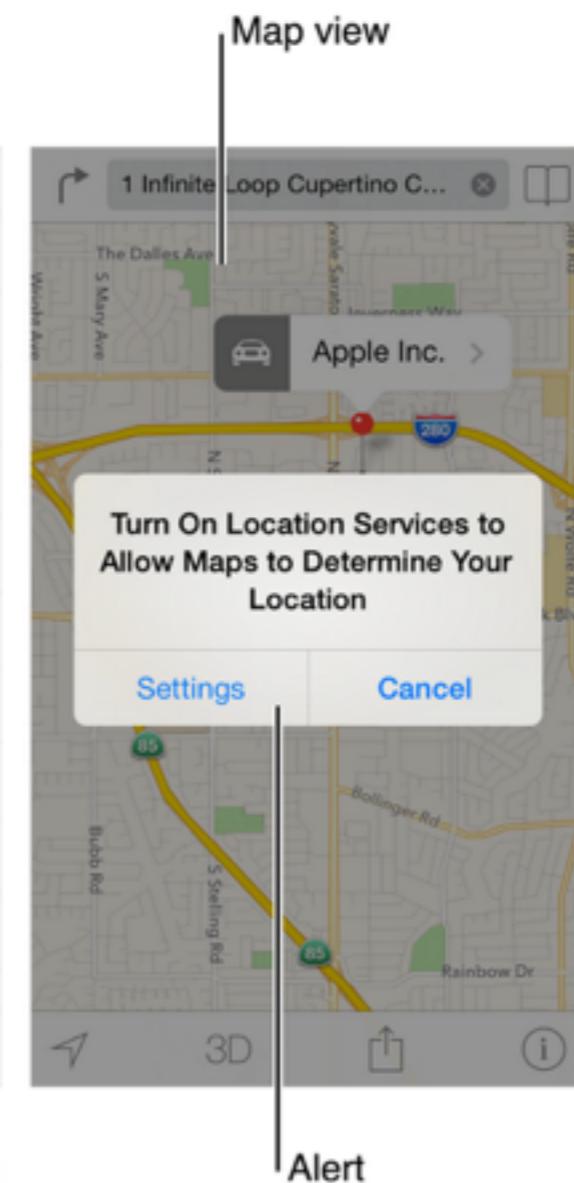
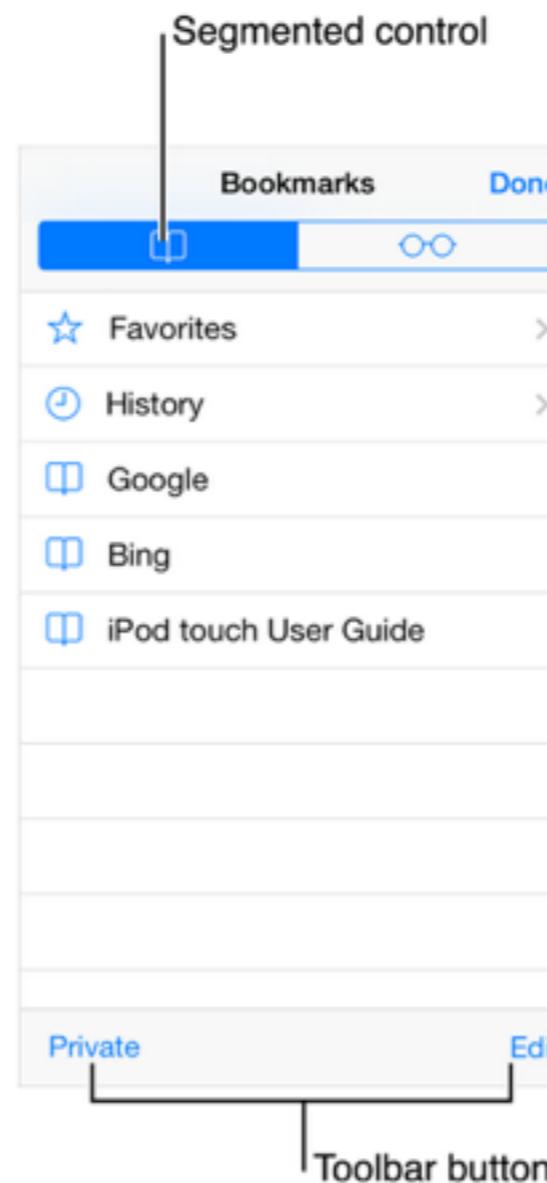
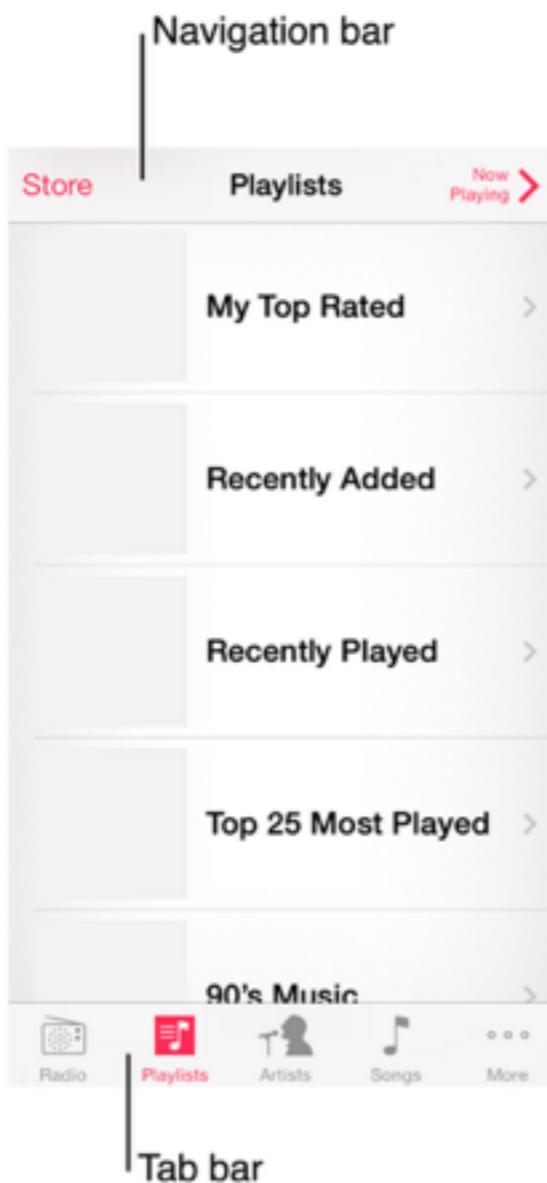


# APP DESIGN OVERVIEW

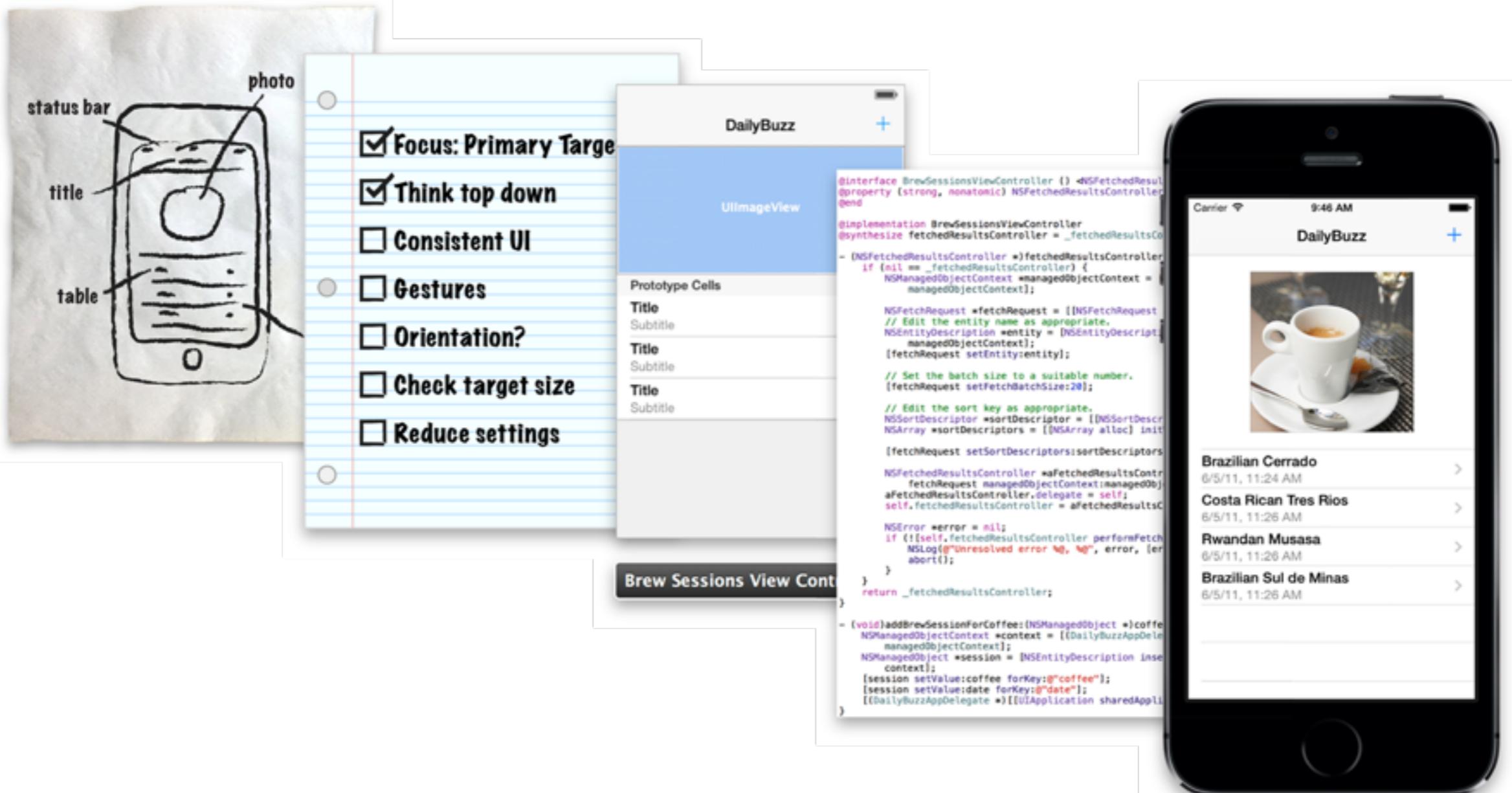
- ▶ Illustration Tools:
  - Sketch – ~\$80



# IOS APP ANATOMY



# APPLICATION DEVELOPMENT CYCLE



# APPLE FRAMEWORKS

---

# APPLE'S FRAMEWORKS/LIBRARIES

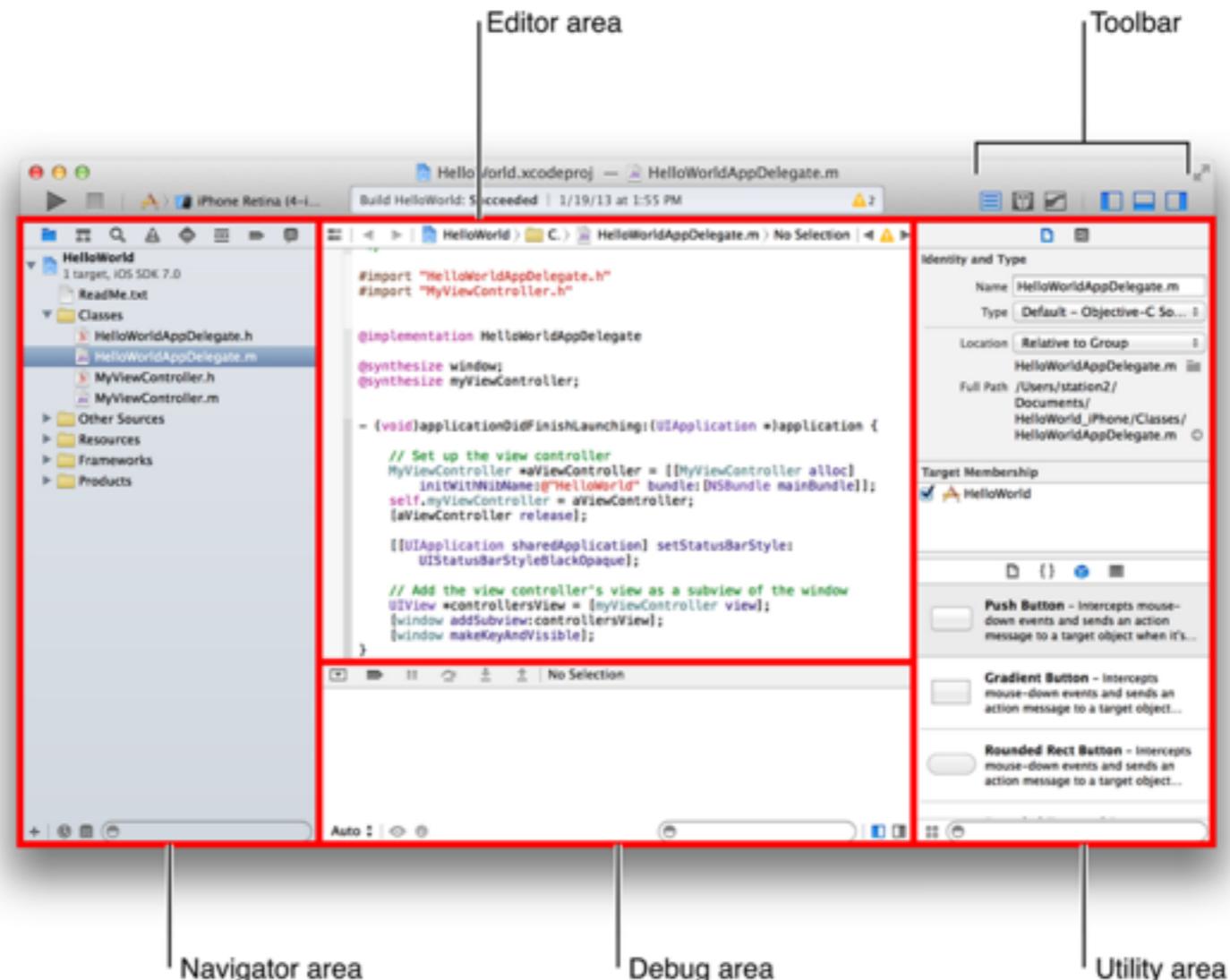
Framework	Description
Foundation	Defines core object-oriented data types like strings, arrays, dictionaries, etc. We'll explore the essential aspects of this framework in the <a href="#">Data Types</a> module.
UIKit	Provides dozens of classes for creating and controlling the user interface on iOS devices.
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

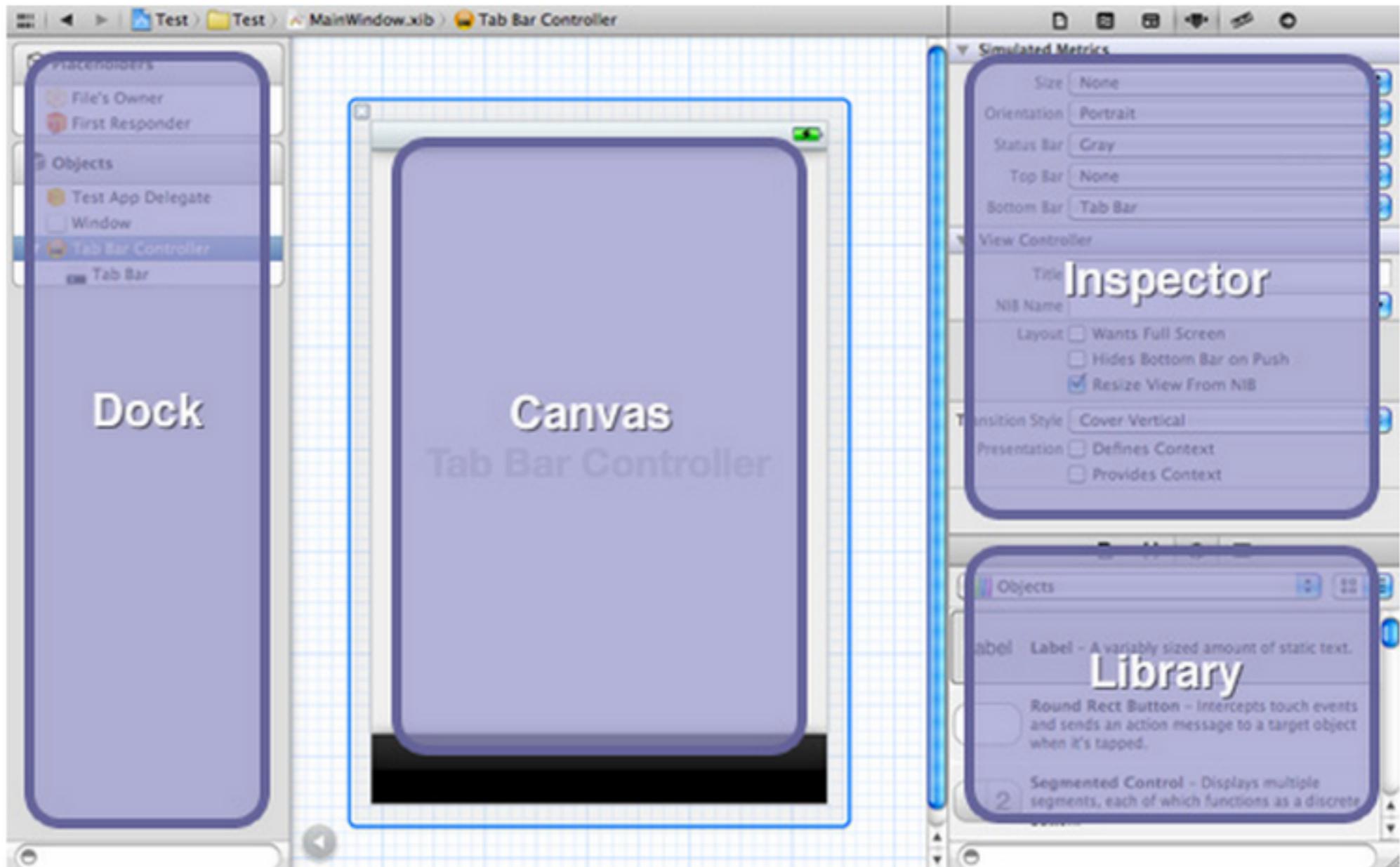
---

# XCODE WALKTHROUGH

- [https://developer.apple.com/library/mac/recipes/xcode\\_help-general/index.html](https://developer.apple.com/library/mac/recipes/xcode_help-general/index.html)

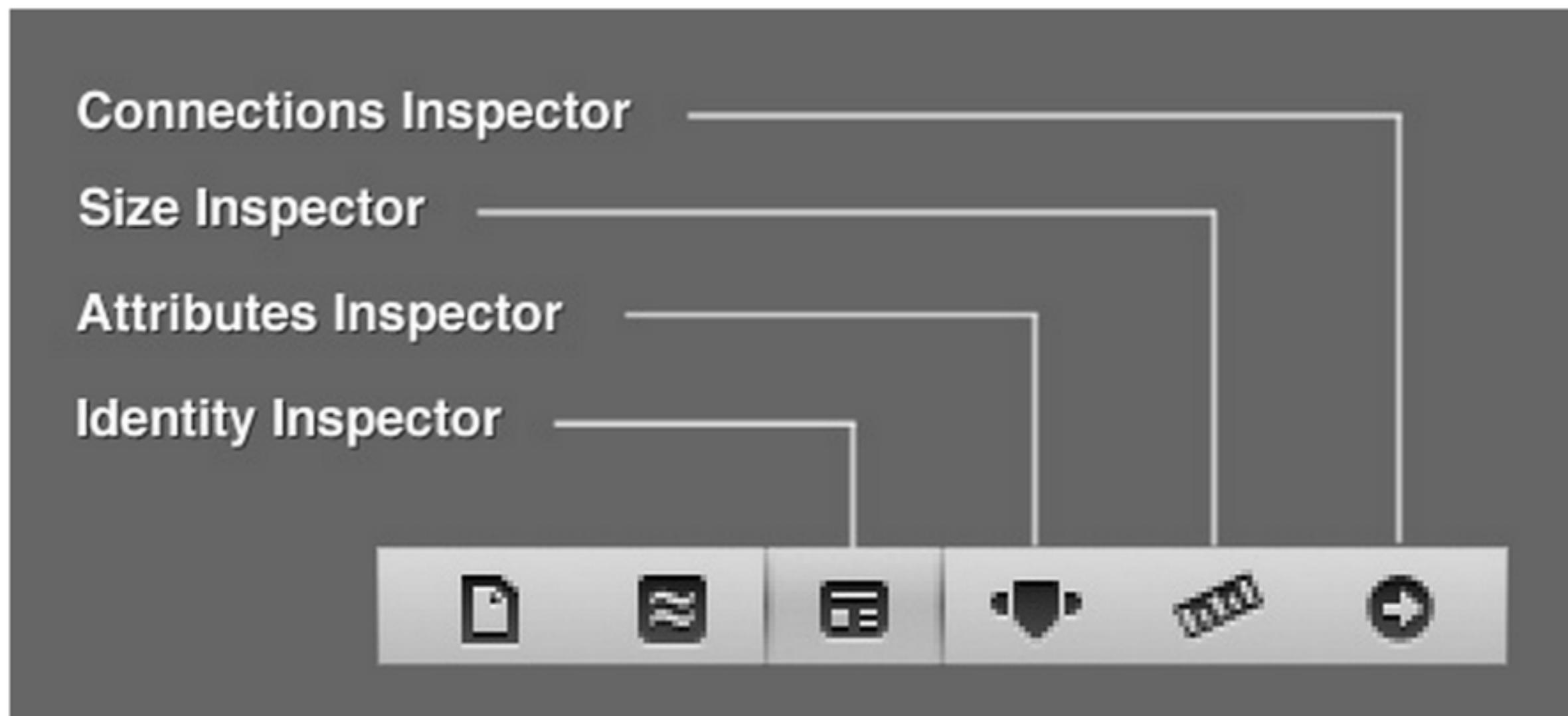


# XCODE WALKTHROUGH



# XCODE WALKTHROUGH

---



# OBJECTIVE-C OVERVIEW

---

---

## OBJECTIVE-C VS SWIFT

---

- Most current code bases are going to be in Objective-C
  - Most current third-party libraries are in Objective-C (e.g. AFNetworking)
  - Transition period will be a slow-but-steady conversion from Obj-C to Swift
- 
- You'll be translating Objective-C (think Stack Overflow posts)
  - Job market: can't get away with only knowing Swift

---

# 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

---

## OBJECTIVE-C VS SWIFT

---

- Most current code bases are going to be in Objective-C
  - Most current third-party libraries are in Objective-C (e.g. AFNetworking)
  - Transition period will be a slow-but-steady conversion from Obj-C to Swift
- 
- You'll be translating Objective-C (think Stack Overflow posts)
  - Job market: can't get away with only knowing Swift

# 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

---

---

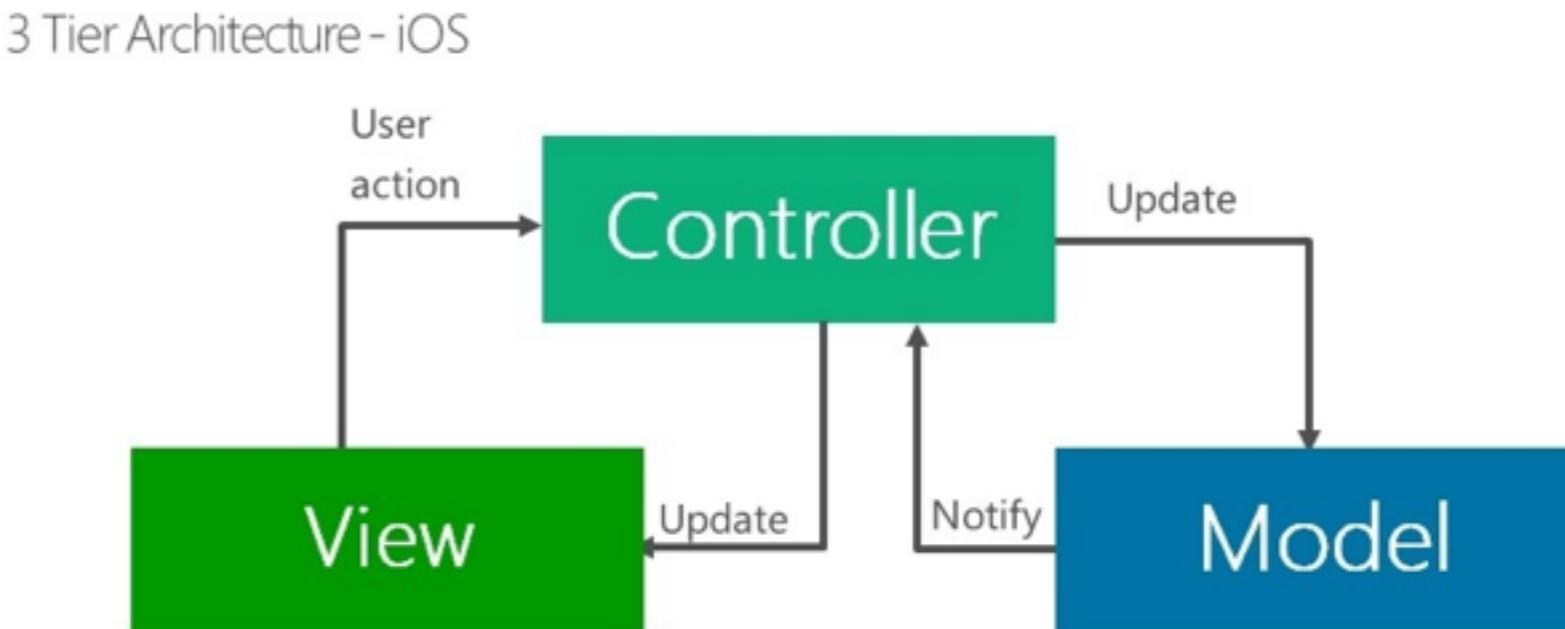
# MODEL–VIEW–CONTROLLER

---

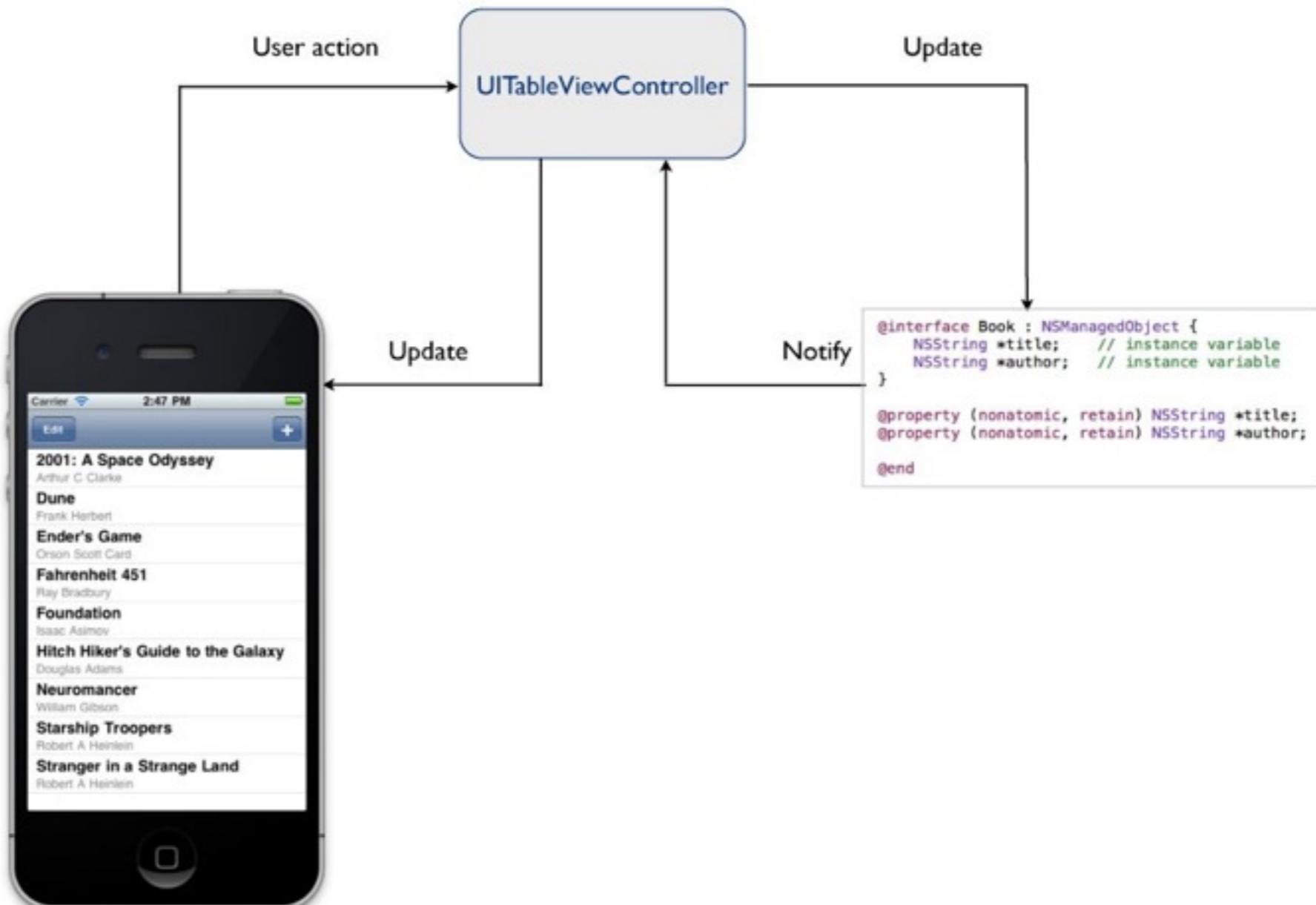
- User interfaces are comprised of **views** – [https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Anatomy.html#/apple\\_ref/doc/uid/TP40006556-CH24-SW1](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Anatomy.html#/apple_ref/doc/uid/TP40006556-CH24-SW1)
- 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



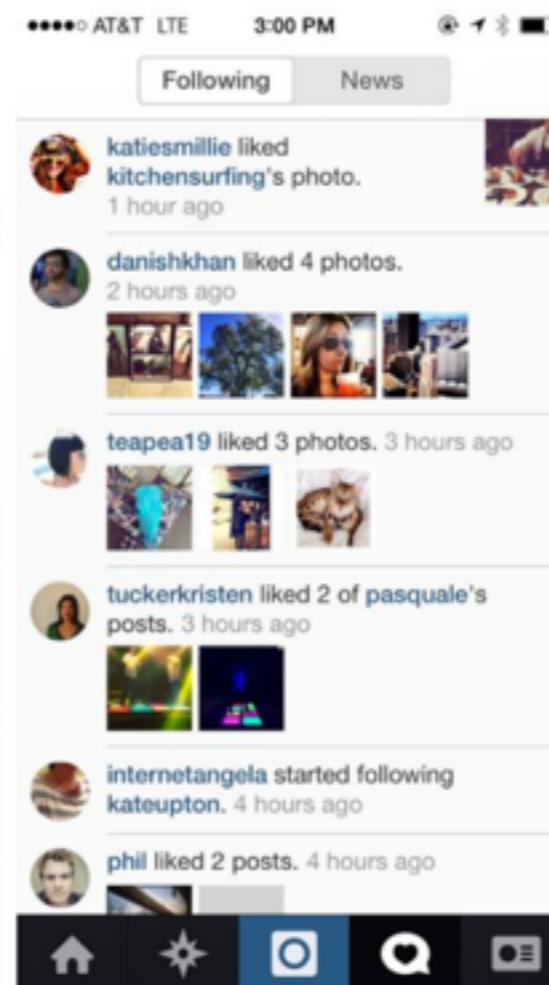
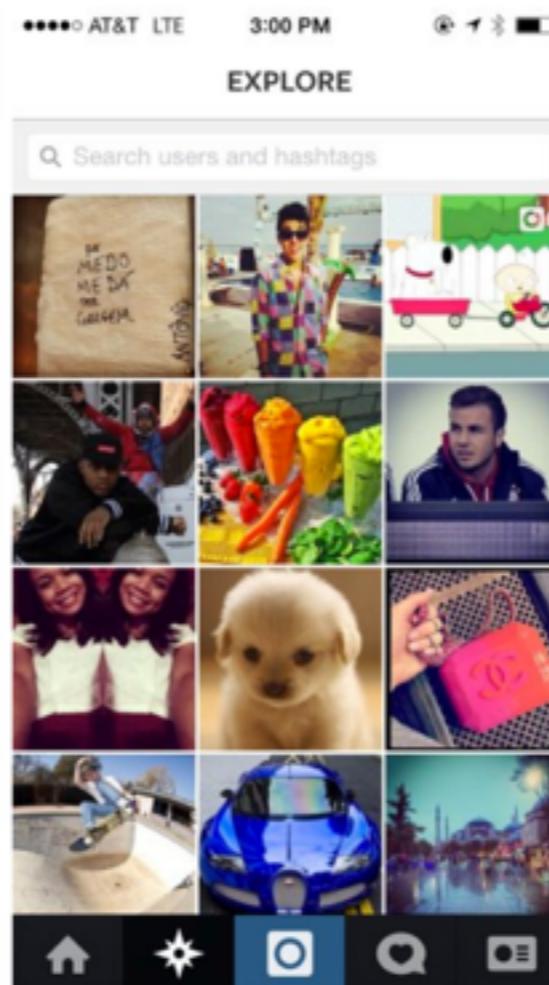
# MODEL-VIEW-CONTROLLER



# VIEW CONTROLLER

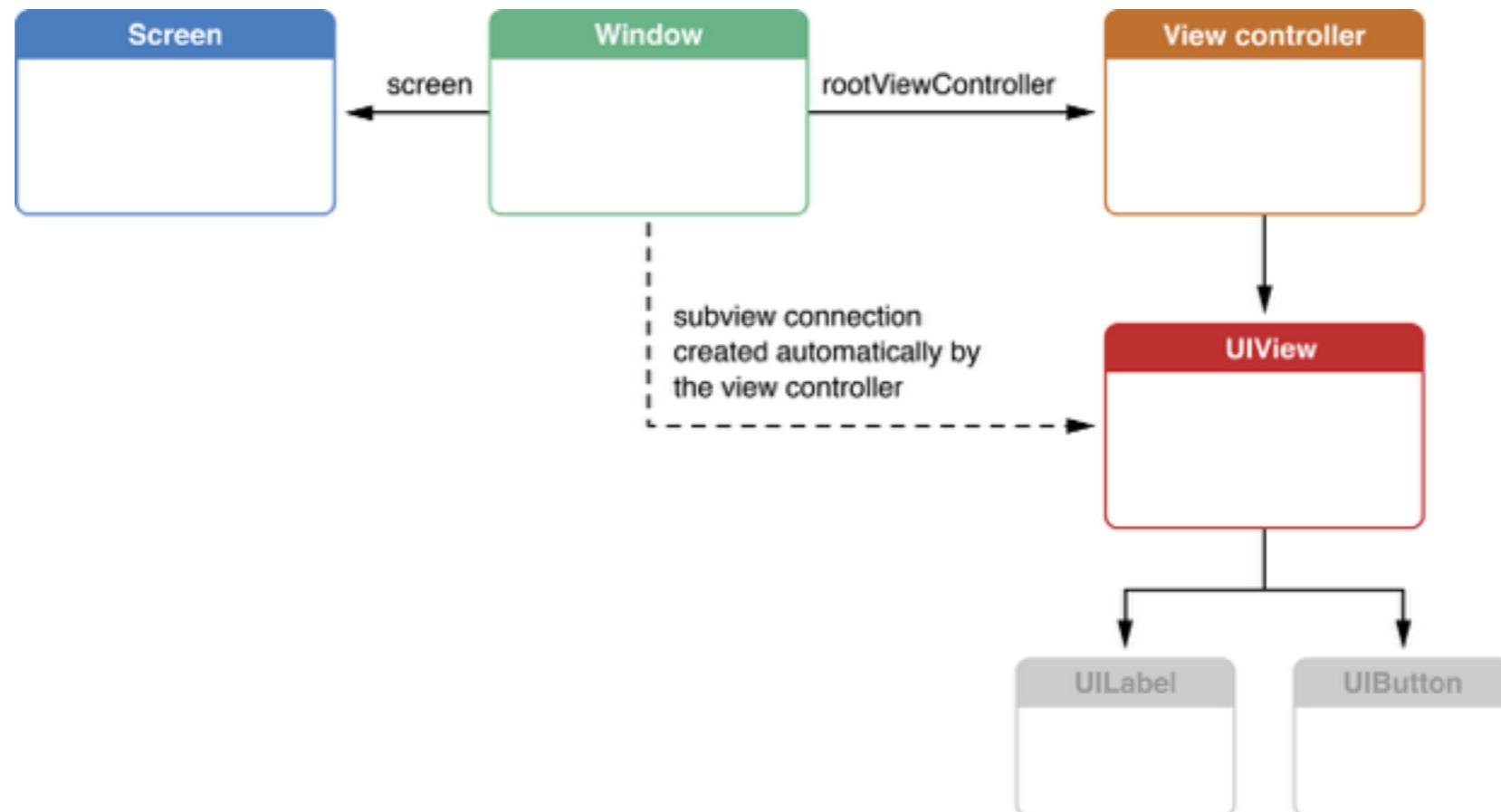
## Instagram Example:

HomeViewController, TrendingViewController, & NewsViewController



# MODEL–VIEW–CONTROLLER

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



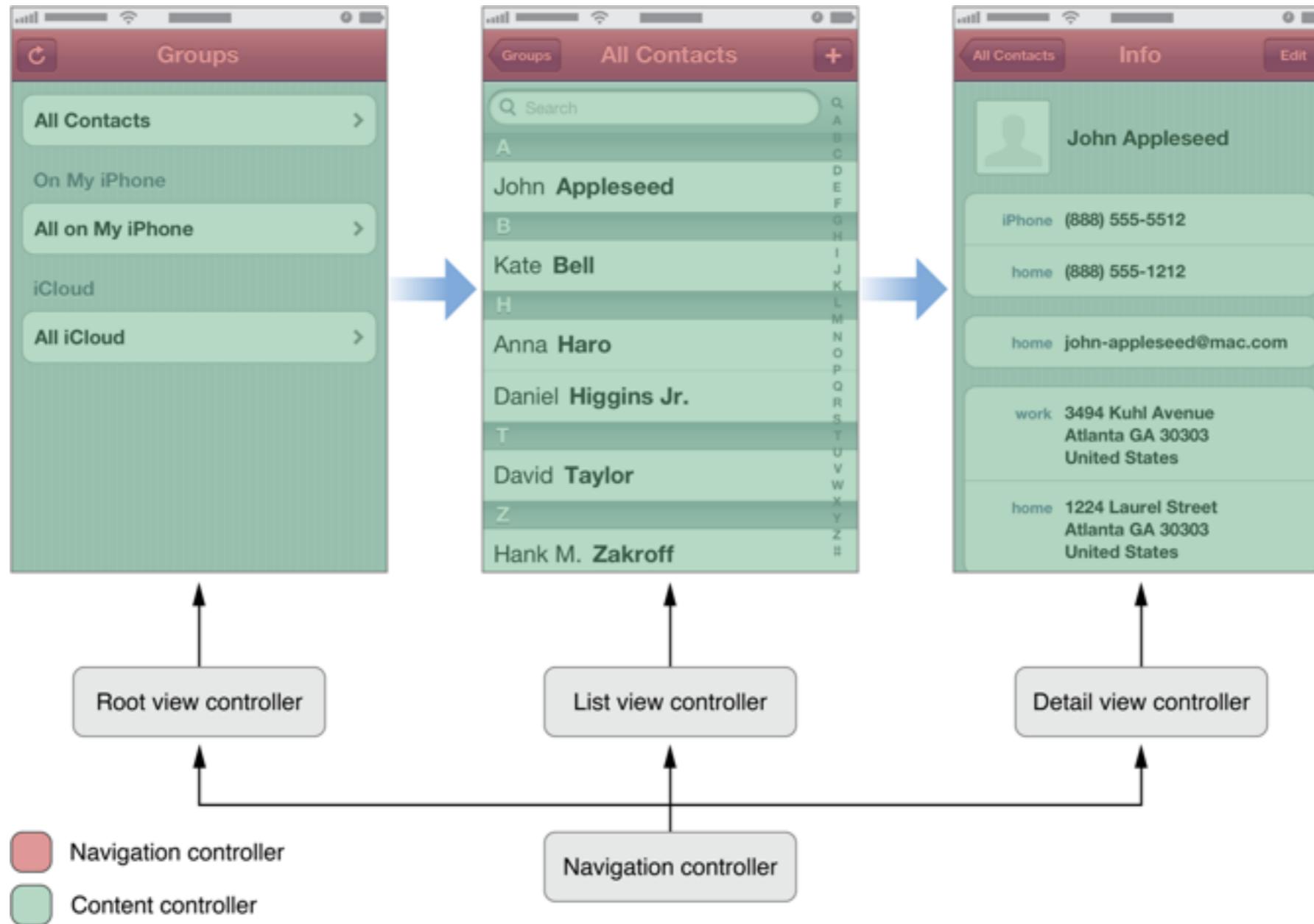
---

## **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



## DATABASE OPTIONS

---



Amazon Mobile Services

# 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

iOS Simulator – iPhone 6 – iPhone 6 / iOS 8.0 (12A365)

Carrier ⚡ 4:27 PM



Tripta Gupta

Web & iOS Engineer

*Placeholder text for the bio section.*

*Placeholder text for the first paragraph of the content area.*

*Placeholder text for the second paragraph of the content area.*

*Placeholder text for the footer area.*

---

---

# **START BUILDING YOUR APP!!**

---

# **IBACTION AND IBOUTLETS**

---

- **IB** stands for **Interface Builder**

- **IBOutlet — Interface Builder Outlet:**

- an outlet is a link from code to UI
- If you want to show or hide an UI element, if you want to get the text of a textfield or enable/disable an element, need to define an outlet of that object in the sources and link that outlet through the “interface object” to the UI element

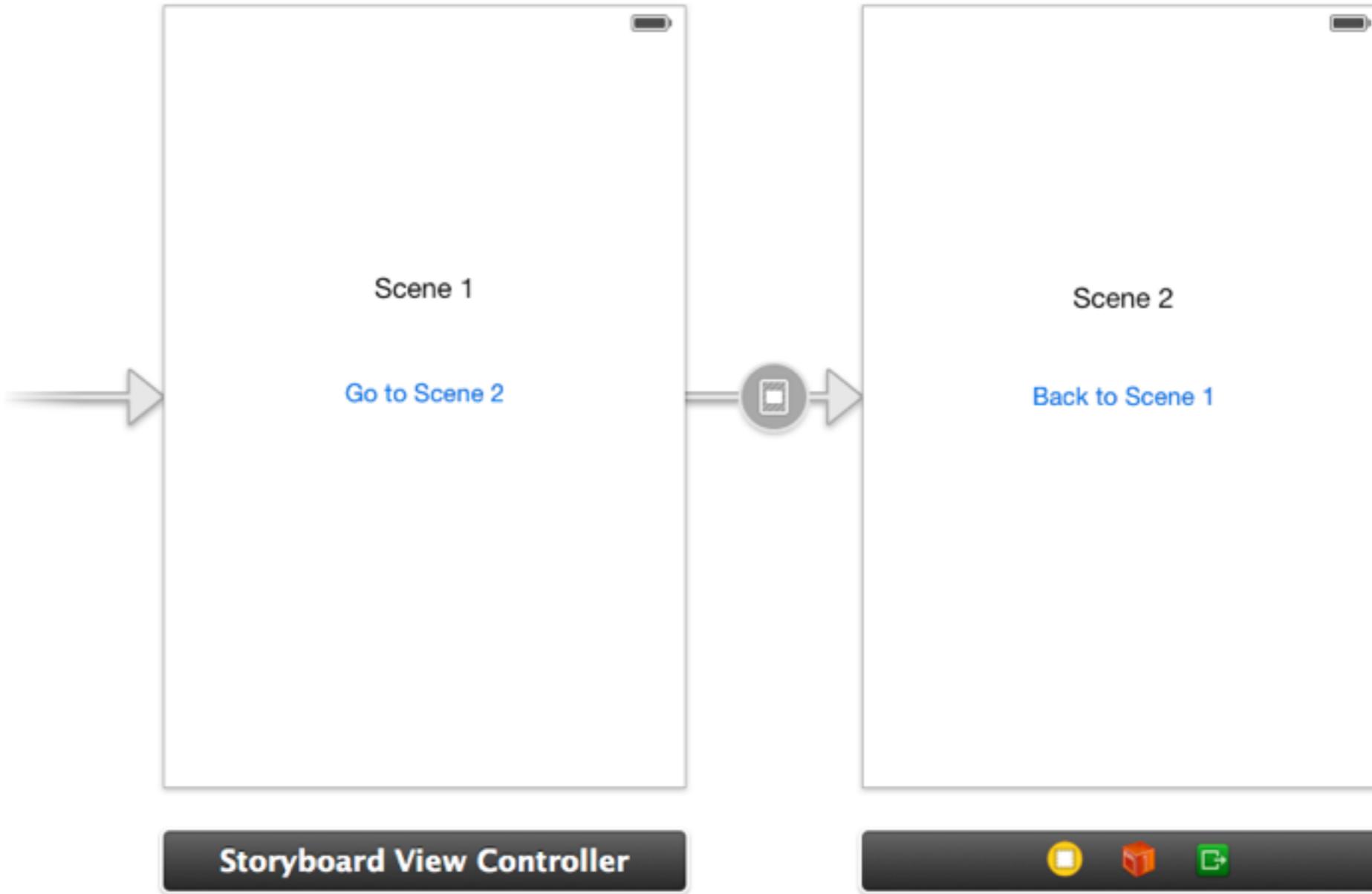
- **IBAction — Interface Builder Outlet:**

- special method triggered when some action is taken
  - e.g. when a button is pressed, it should call a method in your code
- IBAction connections are made from UI object

# TABLEVIEWS, SEGUES, WEBVIEWS

---

# SEGUES



---

---

# BUILDING THE SOCIAL LINKS TAB

---

## **BUILDING SOCIAL LINKS – OBJECTIVES**

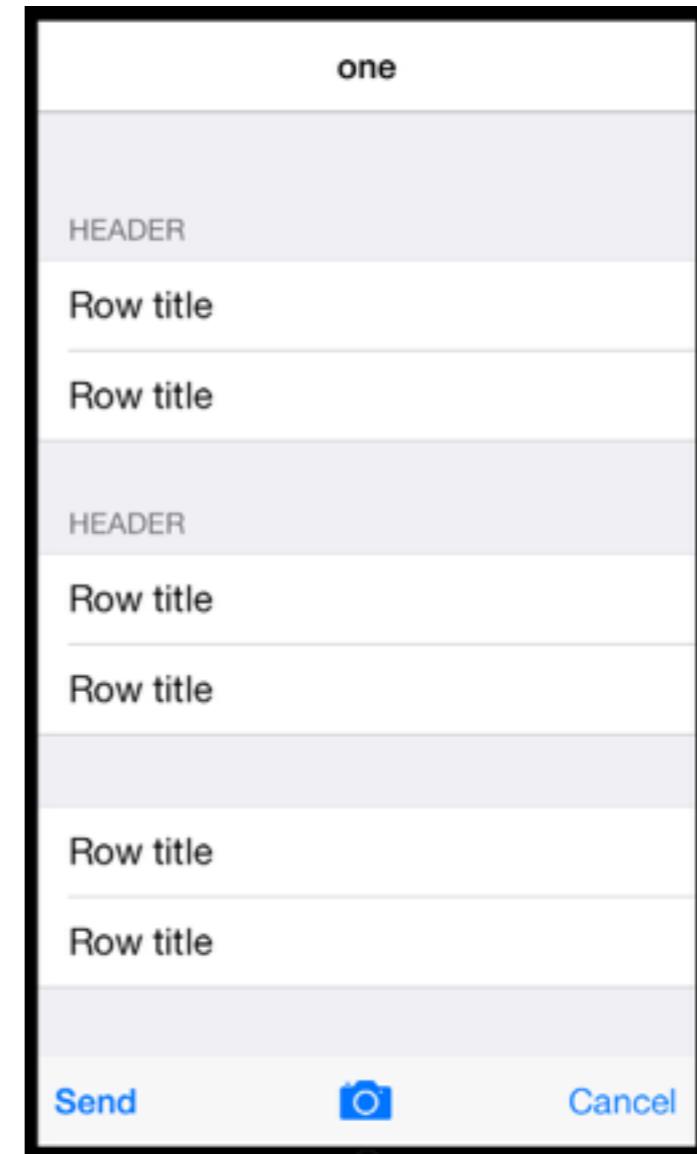
---

- 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

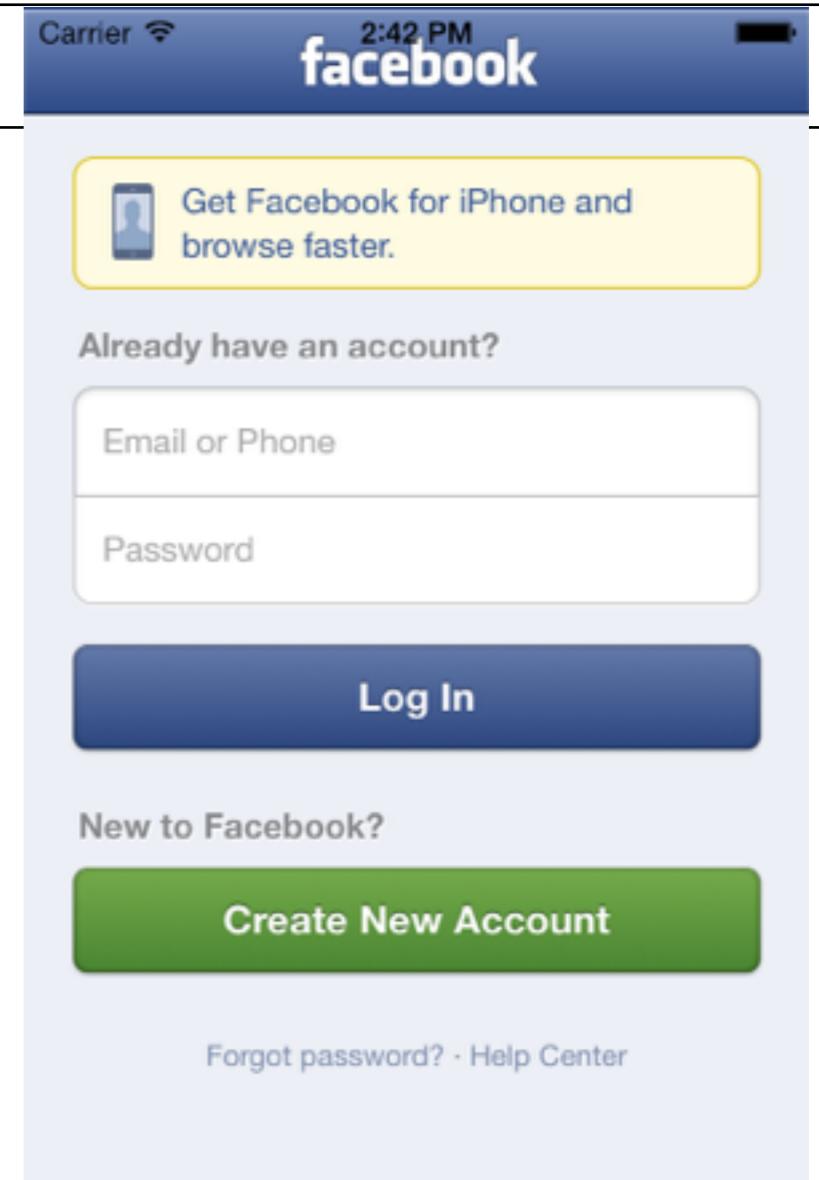
---

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



# UIWEBVIEW

- A web view is a region that can display rich HTML content.
- E.g. any app that uses authentication



[English \(US\)](#) · [Español](#) · [Português \(Brasil\)](#) · [More...](#)

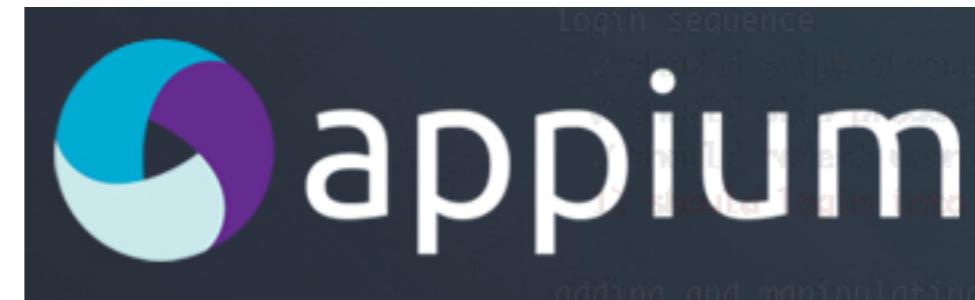
Facebook ©2013

# APPLICATION DEVELOPMENT CYCLE

---

# APPLICATION TESTING

---



## EXAMPLE CALABASH TEST

Feature: Rating a stand

Scenario: Find and rate a stand from the list

Given I am on the foodstand list

Then I should see a "rating" button

And I should not see "Dixie Burger & Gumbo Soup"

When I touch the "rating" button

Then I should see "Dixie Burger & Gumbo Soup"

When I touch "Dixie Burger & Gumbo Soup"

Then I should see details for "Dixie Burger & Gumbo Soup"

When I touch the "rate\_it" button

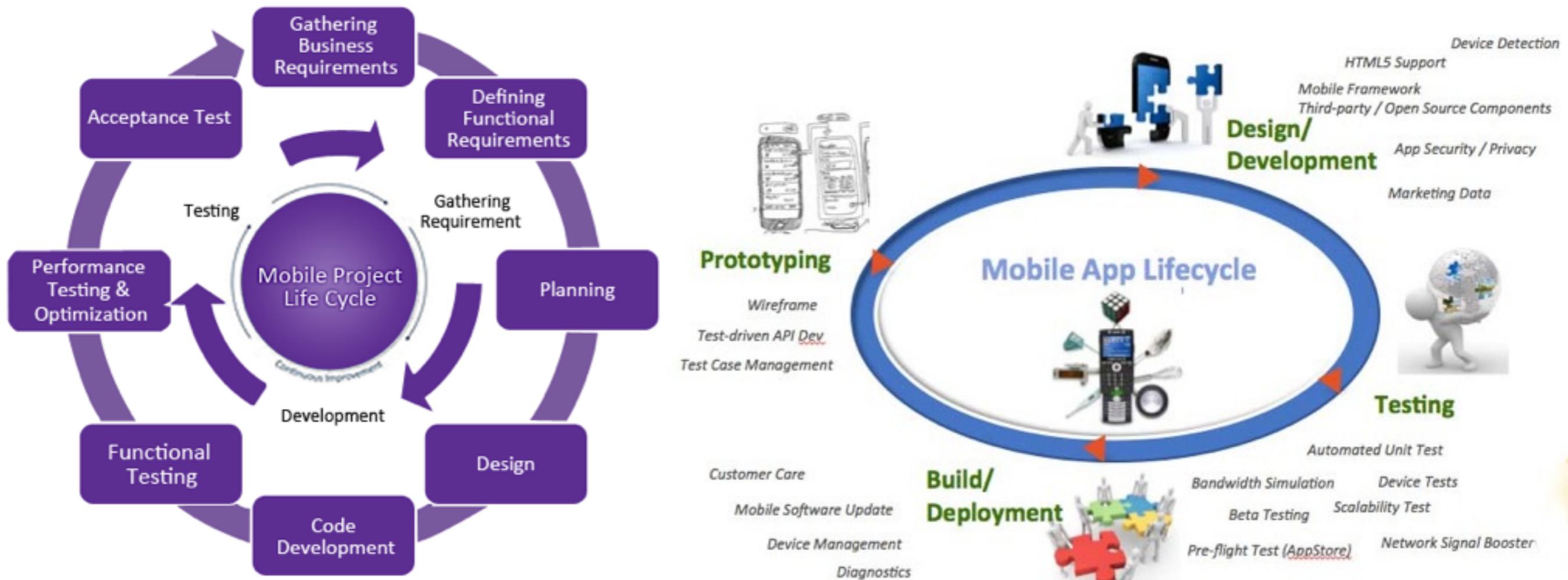
Then I should see the rating panel

When I touch "star5"

And I touch "rate"

Then "Dixie Burger & Gumbo Soup" should be rated 5 stars

# APPLICATION DEVELOPMENT CYCLE



# APP STORE REVIEW GUIDELINES

---

# APP STORE REVIEW GUIDELINES

---

- <https://developer.apple.com/appstore/resources/approval/guidelines.html>
  
- Functionality
- Metadata (name, description, ratings)
- Push notifications
- User Interface
- Purchasing
- ... and so much more!

# IOS DESIGN CHEAT SHEET

‣ <http://ivomynttinen.com/blog/the-ios-7-design-cheat-sheet/>

-Resolutions & Display Specifications

- Rounded Corners

- Grid System

- User Interface — e.g. flat UI

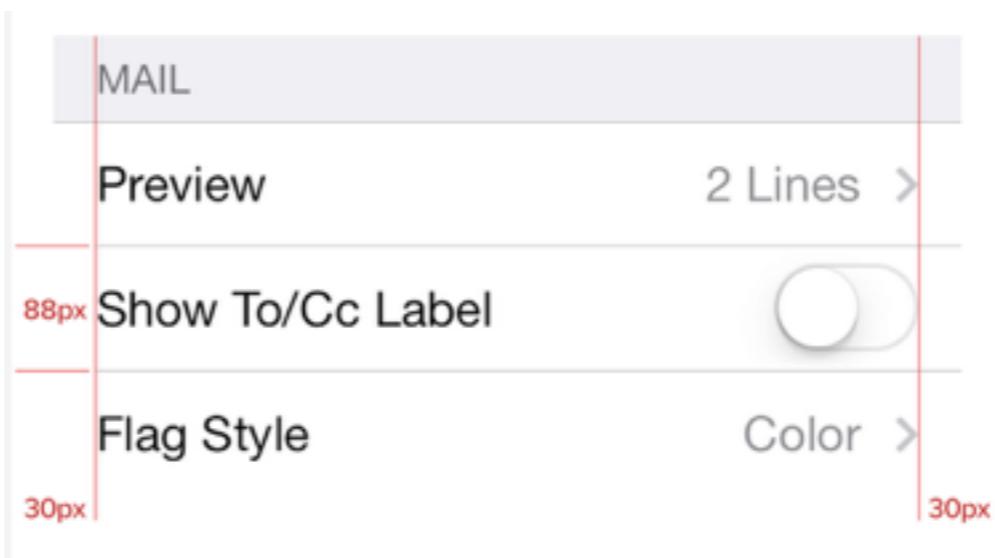
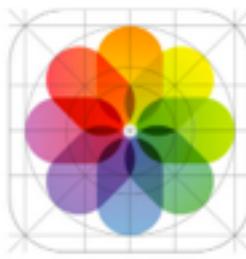
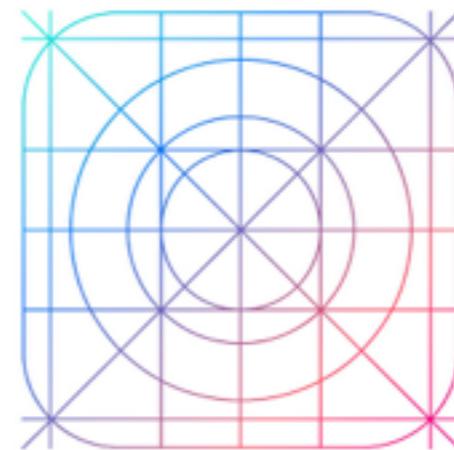
- Status Bar

- Typography

Rounded corners



Grid system



---

---

# BUILDING THE PROJECT

## TAB

---

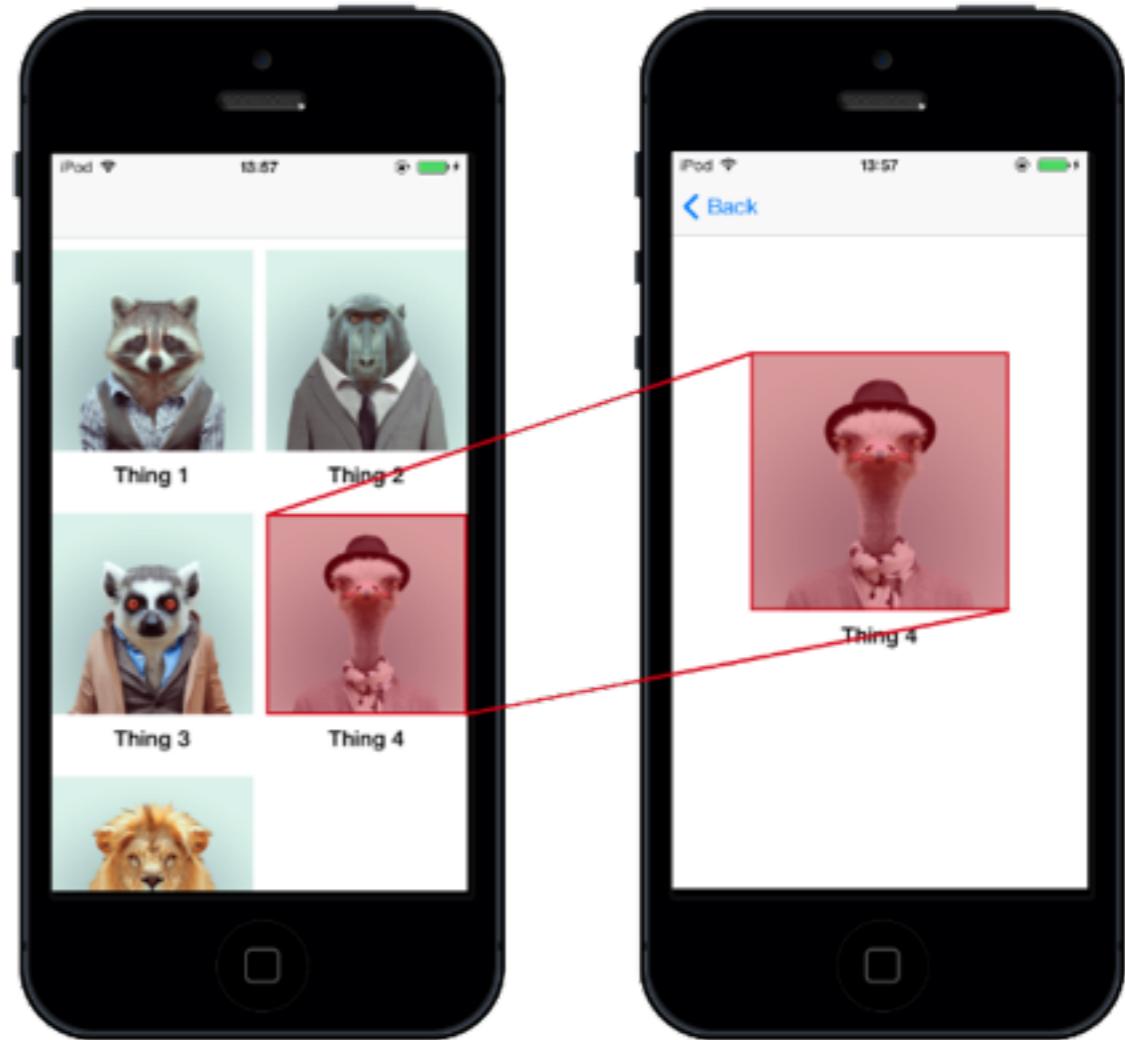
## **BUILDING PROJECT TAB – OBJECTIVES**

---

- 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.



---

## DELEGATES

---

- Delegation is a way of allowing objects to interact with each other without creating strong interdependencies between them, since this makes the design of your application less flexible.
- Instead of objects controlling one another, they can have a delegate which they send (or delegate) messages to, and the delegate does whatever they do, in order to respond and act to this message, and then usually return something back to the other object.

# WRAP UP

---

---

## WRAP UP

---

- How to use Xcode
- How to use Storyboards
- Connecting Visual Storyboard Views with ViewController Code
- Reading Documentation
- <https://developer.apple.com/library/ios/navigation/>

---

---

# 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/>
- Swift Apple Resources - <https://developer.apple.com/swift/resources/>
- WWDC 2014 Session Videos - <https://developer.apple.com/videos/wwdc/2014/>

---

# RESOURCES

---

## Design:

- iOS 8 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/>

---

# RESOURCES

---

## Books:

- Big Nerd Ranch Guide – iOS, 4th Edition
- Big Nerd Ranch Guide – Objective-C
- Functional Programming in Swift – [objc.io/books](http://objc.io/books)

---

# RESOURCES

---

## Blogs:

- NSHipster — advanced topics — <http://nshipster.com/>
- Objc.io — journal — <http://www.objc.io/>
- Mike Ash Blog — <https://www.mikeash.com/pyblog/>
- Dave Verwer's iOS Dev Weekly - <http://iosdevweekly.com/>
-

---

# RESOURCES

---

## Schools:

- Mobile Makers Academy — <http://www.mobilemakers.co/>
- Flatiron School — <http://flatironschool.com/iOScurriculum.html>
- Big Nerd Ranch — <https://training.bignerdranch.com/classes>
- CodePath — <http://codepath.com/iosbootcamp>

---

---

# Q&A

---

# THANKS!

---

## CONTACT INFO:

- Tripta Gupta: [tripta@google.com](mailto:tripta@google.com)
- Stephanie Szeto: [stephanie@liverail.com](mailto:stephanie@liverail.com)
- David Ladowitz: [david.ladowitz@omadahealth.com](mailto:david.ladowitz@omadahealth.com)