



iOS DEVELOPMENT BOOTCAMP '24

ORIENTATION SESSION



Tools

Technologies

Languages



AGENDA FOR TODAY'S SESSION

Brief Introduction

How to become an **iOS Developer?**

Salary Packages and Market Demand for **iOS Developers** in
the local and international market

Best Resources to Learn **iOS Development**

iOS Development Course Outline

Advice for an aspiring **iOS Developer**

Q/A Sessions



TRAINER'S INTRO

MBA Batch '21, IBA Karachi

BCS Batch '17, DHA SUFFA UNIVERSITY

I'm a U.S Consulate General Certified iOS Trainer

I'm currently working as a Senior Engineer - iOS @ Gaditek

I'm currently working as a Software Engineer - iOS @ MMC Global Inc (Texas, Houston)

Previously, I worked as an iOS Engineer for Hudisoft, Time Based Software, LLC

Previously worked as an iOS Trainer for DigiPakistan, Zeal International Training Center, and Apptron Technologies.

Also worked as an Assistant Project Manager - CEE Solutions (Pvt) Limited

5+ years of working experience in iOS Native Mobile Application Development

I have more than 10+ years of training experience.



You can find me on YouTube @ <https://www.youtube.com/c/FarhajAhmed>



FARHAJ AHMED
Tech Education, Personal Growth & Entrepreneurship

Farhaj Ahmed
@FarhajAhmed · 734 subscribers · 43 videos
This is Farhaj Ahmed. I possess a passion for teaching and training. I create iOS Developm... >
instagram.com/farhaj.ahmed1 and 4 more links

[Customise channel](#) [Manage videos](#)

Environment Setup for iOS Development
iOS Development Course - Module #1: iOS Basics

Farhaj Ahmed
Public
31 videos 1,733 views Updated today

[Play all](#) [Shuffle](#)

Senior iOS Engineer | iOS Specialist |
Tech Educator | Avid Reader

As a seasoned Senior Engineer-iOS at
the Cyber Security Department of
PureSquare, I contribute to the mission
of creating a safer digital world for
millions of users across 150+ countries.
Our suite of cutting-edge cybersecurity
products ensures privacy, security and

Lesson 7: SWIFT LOOPS - PART 1 | URDU | FARHAJ AHMED
Farhaj Ahmed · 165 views · 3 years ago
SWIFT 17:12

Lesson 8: SWIFT LOOPS - PART 2 | URDU | FARHAJ AHMED
Farhaj Ahmed · 117 views · 3 years ago
SWIFT 20:31

Lesson 9: SWIFT SWITCH STATEMENTS | URDU | FARHAJ AHMED
Farhaj Ahmed · 119 views · 3 years ago
SWIFT SWITCH STAT 23:14

Lesson 10: SWIFT FUNCTIONS - PART 1 | URDU | FARHAJ AHMED
Farhaj Ahmed · 238 views · 3 years ago
SWIFT FUN 18:31

Lesson 11: SWIFT FUNCTIONS - PART 2 | URDU | FARHAJ AHMED
Farhaj Ahmed · 144 views · 3 years ago
SWIFT FUN 28:23

Lesson 12: MY FIRST iOS APP - PART 1 | ONE BUTTON APP | URDU | FARHAJ
Farhaj Ahmed · 428 views · 3 years ago
One Button App MY FIRST iOS APP PART 1 21:12

Lesson 13: MY FIRST iOS APP - PART 2 | SLIDER & LABELS | URDU | FARHAJ
Farhaj Ahmed · 199 views · 3 years ago
Slider & Labels MY FIRST iOS APP PART 2 17:57

⋮

Progress from my past iOS Training Batches

- Out of 70+, our 40 students are currently being employed for iOS Developer Positions in the local market
- It resulted in the startup of 10 students
- 20 students are currently earning through freelance platforms





SAYING OF THE SESSION!

Be a Master of one instead of
being a Jack of all Trades!



Most Frequently Asked Questions regarding iOS Development

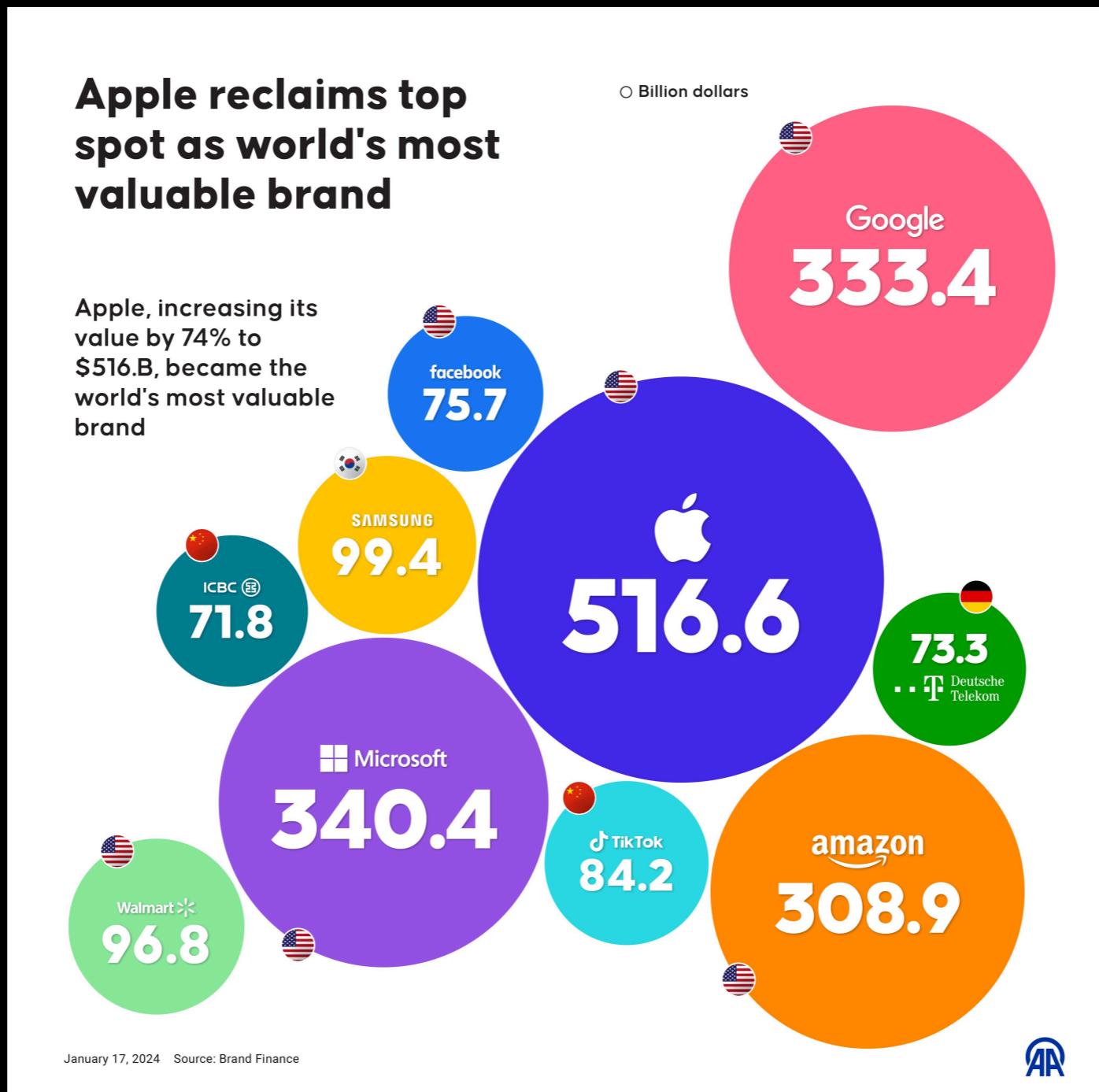


- Why iOS Development?
- Mac Required or Not?
- Pay Scale or Salary Packages for iOS?
- Do I Stick to Single Platform or Multiple?
- When do I start iOS Development?
- How to get Started with iOS?
- How to set up MAC OS X without Mac?
- Native or Hybrid Platforms?



Why iOS Development?

“Apple just made history by becoming the first company with a \$3 trillion market value—'and its lock on the consumer is only getting stronger”
Apple CEO Tim Cook



MAC NEEDED OR NOT?

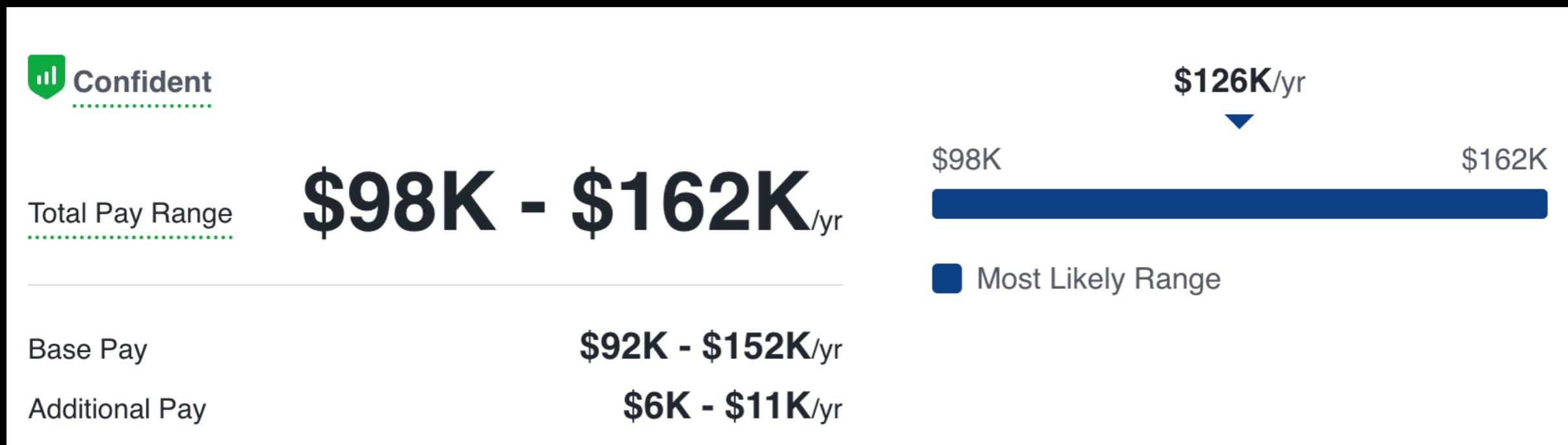
- ▶ **Not Needed for training purpose!**
- ▶ Minimum Requirements without Mac
 - ▶ 2.5 or above GHz Processor
 - ▶ 16 GB RAM or above
 - ▶ 500 GB Hard drive
- ▶ Minimum Requirements with Mac for iOS Development
 - ▶ 2.5 or above GHz Processor
 - ▶ 16 GB RAM or above
 - ▶ 500 GB Hard drive or above



Salary Expectation for iOS Development - International Market - U.S Stats



glassdoor



Salary Expectation for iOS Development - Freelance Market

The hourly rates of freelance iOS developers depends on the following factors:

- **Experience in years**
- **Client list** (if you only have Fortune 500 companies as clients, then your hourly rate will be much higher)
- **Technology focus** (if you are an expert in iOS, than your rate will be higher, than if you are a generalist)
- **Region** (an iOS developer in the USA, will be charging much higher rates, than a developer in India)
- **Industry** (the car manufacturing industry pays higher hourly rates, than a local coffee shop, which needs an app)

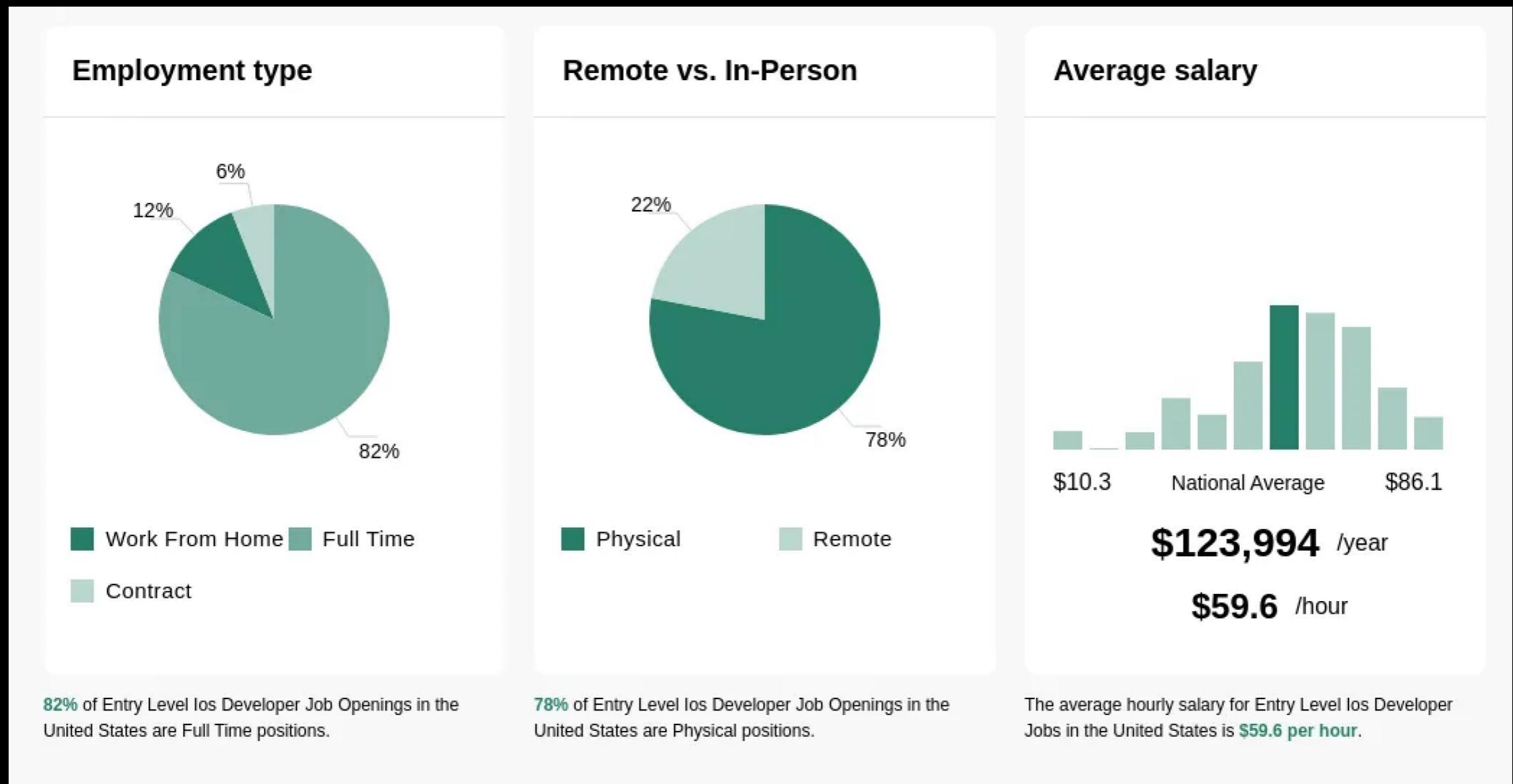
Salary Expectation for iOS Development - Freelance Market



Source: <https://arc.dev/freelance-developer-rates/ios>

Salary Expectation for iOS Development

As of January 2024, there are many openings in the United States for the iOS Development Position



Salary Expectation for iOS Development - International Market - U.S Stats

Junior iOS Developer

\$ 6, 000 / month - \$ 6, 500 / month

Mid Level iOS Developer:

\$ 8, 500 / month - \$ 10, 000 / month

Senior Level iOS Developer:

Salary > \$ 10, 000 / month

Salary Expectation for iOS Development - Local Market

Junior iOS Developer:

80,000 PKR / month - 100,000 PKR / month

Mid Level iOS Developer:

100,000 PKR / month - 300,000 PKR / month

Senior iOS Developer:

300,000 PKR / month - 500,000 PKR / month & above!

Desired SKILL for a fresh iOS Developer in the market?

 **Farhaj Ahmed**
iOS Engineer at Time Based Software, LLC
3w

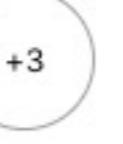
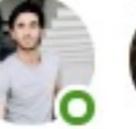
I'm collecting some reviews, help needed. Thanks!

Considering yourself as a Senior iOS Developer, I would like to ask what requirements do you look for a fresh graduate who's going to pursue their career for iOS Platform as a fresh start? What do they need to know and what knowledge is required to get his/her development work started?

11 Likes · 8 Comments

 Like  Comment  Share Top Comments ▾

Likes



 4,585 views of your post in the feed

Desired SKILL for a fresh iOS Developer in the market?

What is a desired skill set expected from a fresh iOS Developer in the market?



Hassan Khan • 1st

3w ...

Associate Architect (iOS) at Axact Technologies

Basic oop concepts should be strong and attitude towards a complex problem.

Doesn't mean he/she should know the correct answer to a complex problem. It all depends on how his/her approach towards the answer is.

[Like](#) [Reply](#) | 4 Likes



Husnain Ali • 1st

3w ...

Senior Software Engineer - (iOS | Laravel) at Novatore Solutions (Pvt) ...

The first thing i always notice is the confidence of that guy and the second must skill for a fresh graduate is the creativity of doing or solving problems then a senior software engineer i expect the fresh graduate to must have the oops & data structures concepts... cause the SWIFT or OBJECTIVE C purely based on oop and data structures last but not the least is the communication skills to pursue someone to hire him or why he is better resource then someone else in cooperate world.

[Like](#) [Reply](#) | 3 Likes

Desired Skill for a fresh iOS Developer in the market?

Muhammad Haris Bin Abid • 1st
iOS Developer | Android Developer 3w ...

Most important thing is candidate should not be dull. Problem solving is the key. Just check basic programming concepts.
If a person is sharp. He can learn any technology. iOS is no different.

But If you hire someone dummy who can't even google his problems after experience of year or two. He can cost you a lot.

[Like](#) [Reply](#) | 2 Likes

Syed Saad Rehman • 2nd
Senior Software Engineer at Vadion pvt Ltd. 2w ...

when interviewing a fresh grad for the post of ios or any other technology.
Good interviewers judges if how flexible a developer is to immerse himself with new things and how a good learner he is..

[Like](#) [Reply](#) | 1 Like

Desired Skill for a fresh iOS Developer in the market?

Mukesh Lokare •1st
Team Lead at NeoSOFT Technologies International.
3w ...

Opps concept really necessary, Basics in Swift, App/Controller lifecycle, Protocol Delegate, Adaptive design must know, Should know the coredata or realm. Should aware about establishing connection in front end & backend.

Like · 3 Likes | Comment

Mehedi Hasan •1st
Lead iOS Developer at Ergo Ventures Ltd | RxSwift | MVVM | Realm | ...
3w ...

Good at Data structure, Algorithm and Have some knowledge in Rest API, Core data & Table view and other UI element.

Like · 2 Likes | Comment

TRAITS NEEDED TO BECOME A GOOD DEVELOPER?

PATIENCE:

IF YOU ARE IMPATIENT ABOUT LEARNING AND EXPECT TO MASTER EVERYTHING QUICKLY AND EFFORTLESSLY, YOU WILL NEVER TRULY BE SUCCESSFUL AT PROGRAMMING

PERSISTENCE (NEVER GIVE UP!)

LOTS AND LOTS OF FOCUS!

**FOCUSED THINKING AS A CHORE.
WITHOUT IT, YOU WILL NEVER TRULY BE SUCCESSFUL AT PROGRAMMING**

PROBLEM SOLVING SKILLS (PRACTICE ON HACKER RANK, LEET CODE & ALGO EXPERT)

What is needed to be a fully fledged iOS Developer?

- Strong hold over SWIFT Programming Language
- Expertise to implement Table Views, Split Views, and Collection Views
- Knowledge of Auto Layout
- Knowledge of Databases and Protocol Oriented Programming
- Knowledge of Database Core Concepts and Core Data
- Knowledge of Networking and RESTful Web Services
- Knowledge of Design Patterns like Singleton, MVC (Model, View Controller), and MVVM (Model–View–ViewModel)
- Knowledge of Git and other development tools like JIRA, Bitbucket, SourceTree Sequel Pro, Postman, and Charles.



What is needed to be a fully fledged iOS Developer?

iOS Developer

Skills Matrix

	Junior	Middle	Senior
Ability	can make a news app talking to a JSON API	can timely design and deliver fast and reliable chat module	can establish team behaviour standards with non-fanatical ideas
Code Integration	uses git to move code around	has an opinion about GitFlow	has ideas how to build a CI process and automate chores
Paradigms	got the idea of OOP	got hands dirty with FRP	brings value from other platforms and paradigms
Dependencies	knows how to use Cocoapods	knows why apps can't use SPM, but Carthage is an option	knows why it is essential to own/reduce dependencies
Platform	uses Array, Dictionary and Set	knows Value/Reference types and Equitable/Hashable	knows the details of method dispatch of both Swift and Obj-C
Client-Server Protocol	getting that JSON from the Internet is a piece of cake!	making a WebSocket-based real-time chat is feasible	building a video chat is an achievable challenge
Reference	uses StackOverflow as a single source of truth	often uses official documentation	asks platform developers and can reverse engineer
Memory	knows how to avoid and fix a memory leak	knows NSPointerArray and why structs increase binary size	has a strategy to reduce out-of-memory crashes
UI	can build basic UI in the Interface Builder	has reasons to make UI in code	can take layout and diff calculation to non-main thread
Multithreading	asyncAfter is a friend, @synchronize all the things	asyncAfter is an enemy, and thread synchronisation is a problem	understands multithreading problems beyond the deadlock
Attitude	expects others to teach and guide during development	discusses design with the team to share decision	takes responsibility for design decision made by teammates
Design Patterns	understands Delegation, Target-Action and MVC idea	is proficient at Observer, Facade and Mediator patterns	knows the sweet spot between dependency injection and service locator
Product Quality	the app is tested if it works on my phone	writes unit tests and tried TDD and UI tests	defines a maintainable test pyramid with non-overlapping coverage areas

JUNIOR

0 - 1.5 YEARS

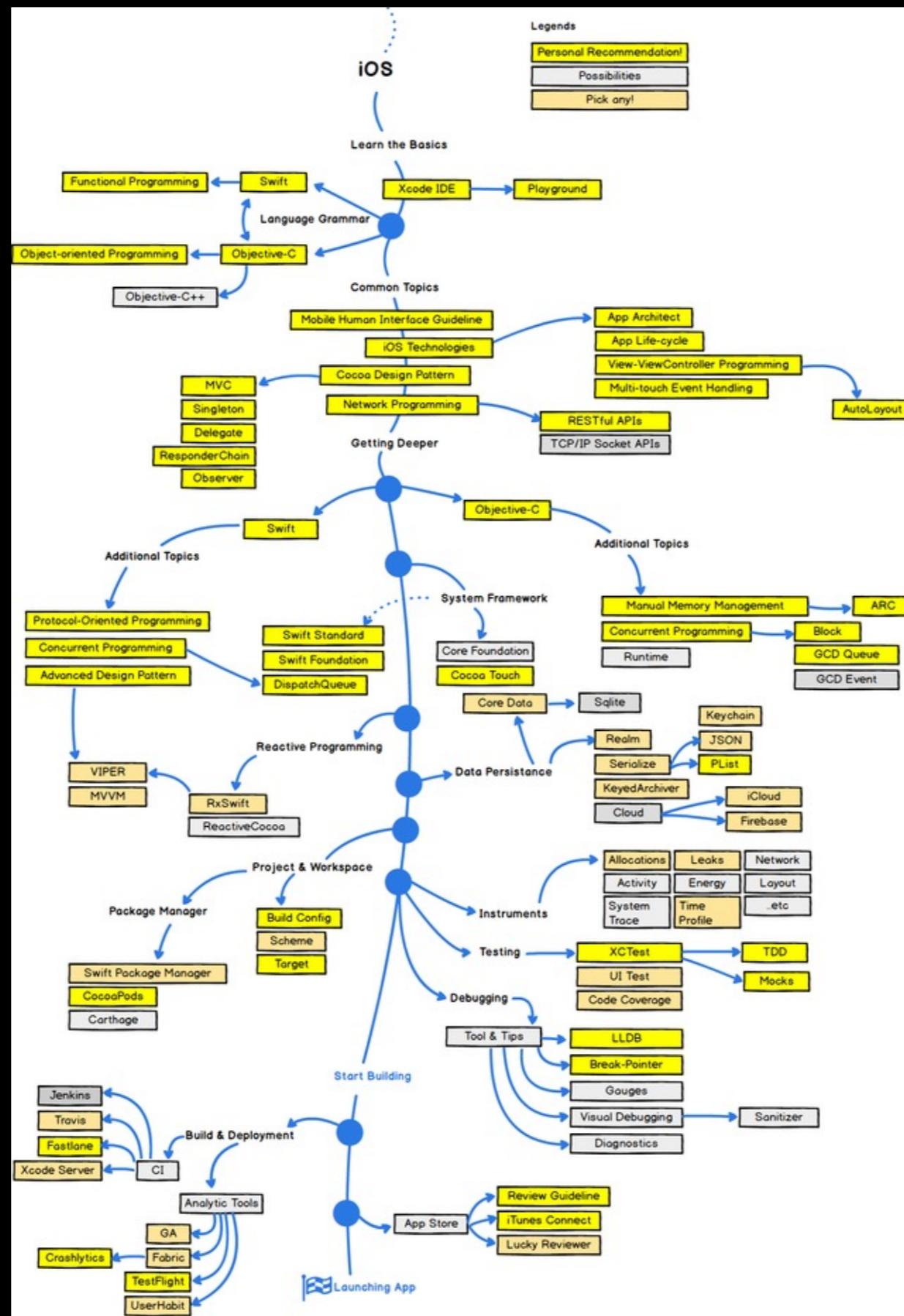
MIDDLE

1.5 - 4 YEARS

SENIOR

5+ YEARS

What is needed to be a fully fledged iOS Developer?



- Objective-C is the primary programming language you use when writing software for OS X and iOS.
- It's a superset of the C programming language and provides object-oriented capabilities and a dynamic runtime.
- Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods.
- The programming language Objective-C was originally developed in the early 1980s.

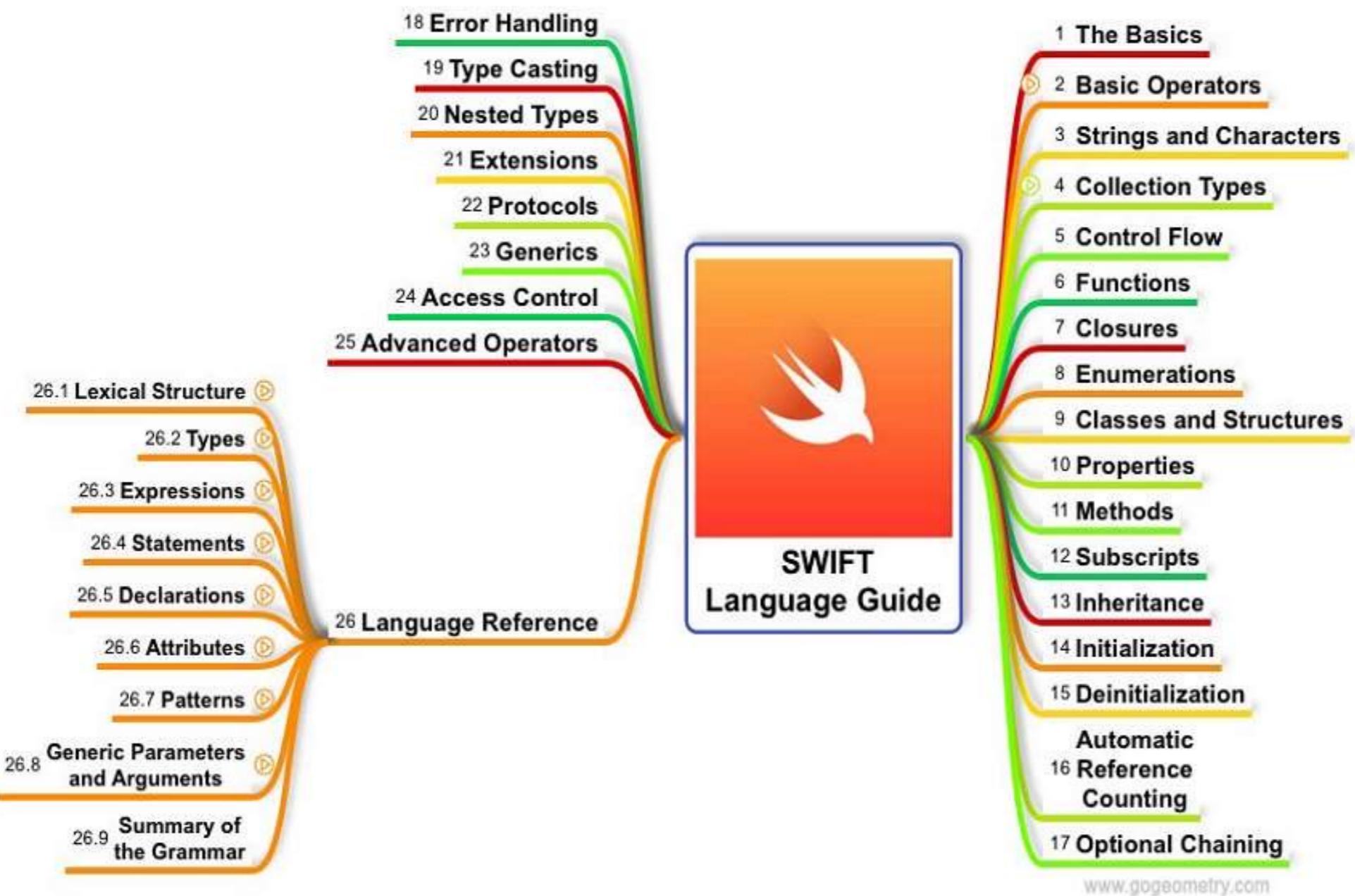


Objective-C

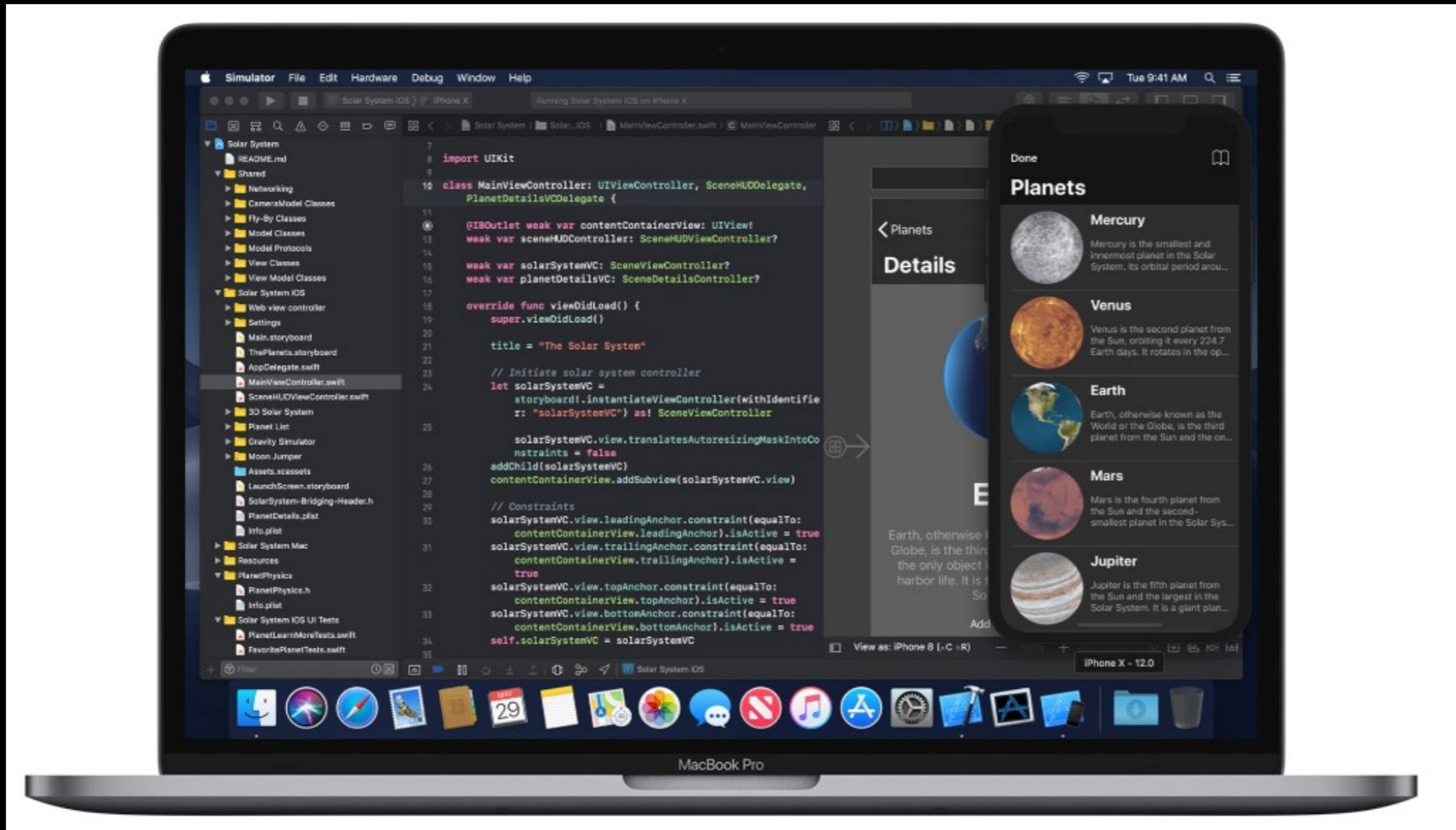
- Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for iOS, macOS, watchOS, tvOS, and Linux.
- Swift was introduced at Apple's 2014 Worldwide Developers Conference.



SWIFT PROGRAMMING LANGUAGE ROADMAP – 2024



Xcode – Tool for iOS Application Development



Xcode is an integrated development environment (IDE) for macOS containing suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS.

First released in 2003, the latest stable release is version 12.5 and is available via the Mac App Store free of charge for macOS High Sierra and macOS Mojave users

MODULE # 1: iOS BASICS

LEARNING OBJECTIVE:

TO MAKE YOU COMFORTABLE WITH SWIFT 5 BASICS

TOPICS TO BE COVERED:

- ▶ Expressions, Variables & Constants
- ▶ Types and Operations
- ▶ Basic Control Flow
- ▶ Advanced Control Flow
- ▶ Functions and Optionals



iOS BASICS

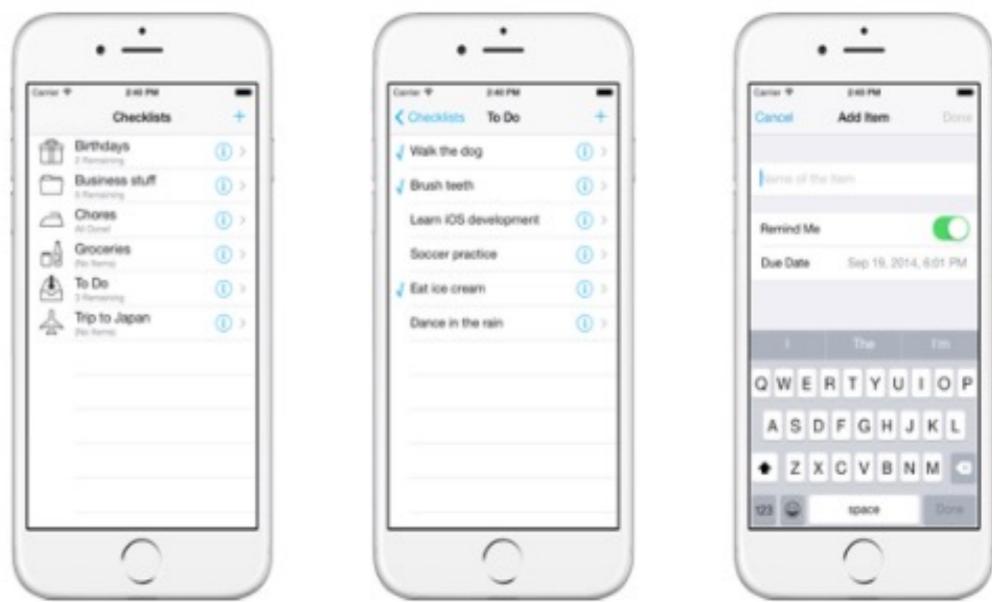
LEARNING OBJECTIVES:

- ▶ To make you comfortable with Xcode, Interface Builder, and Swift 4 in an easygoing manner.
- ▶ To make you comfortable with standard UIKit components
- ▶ To enable yourself to customize your own app to make them look good!
- ▶ To make you an expert with the fundamental design patterns that all iOS apps use and about table views, navigation controllers and delegates.



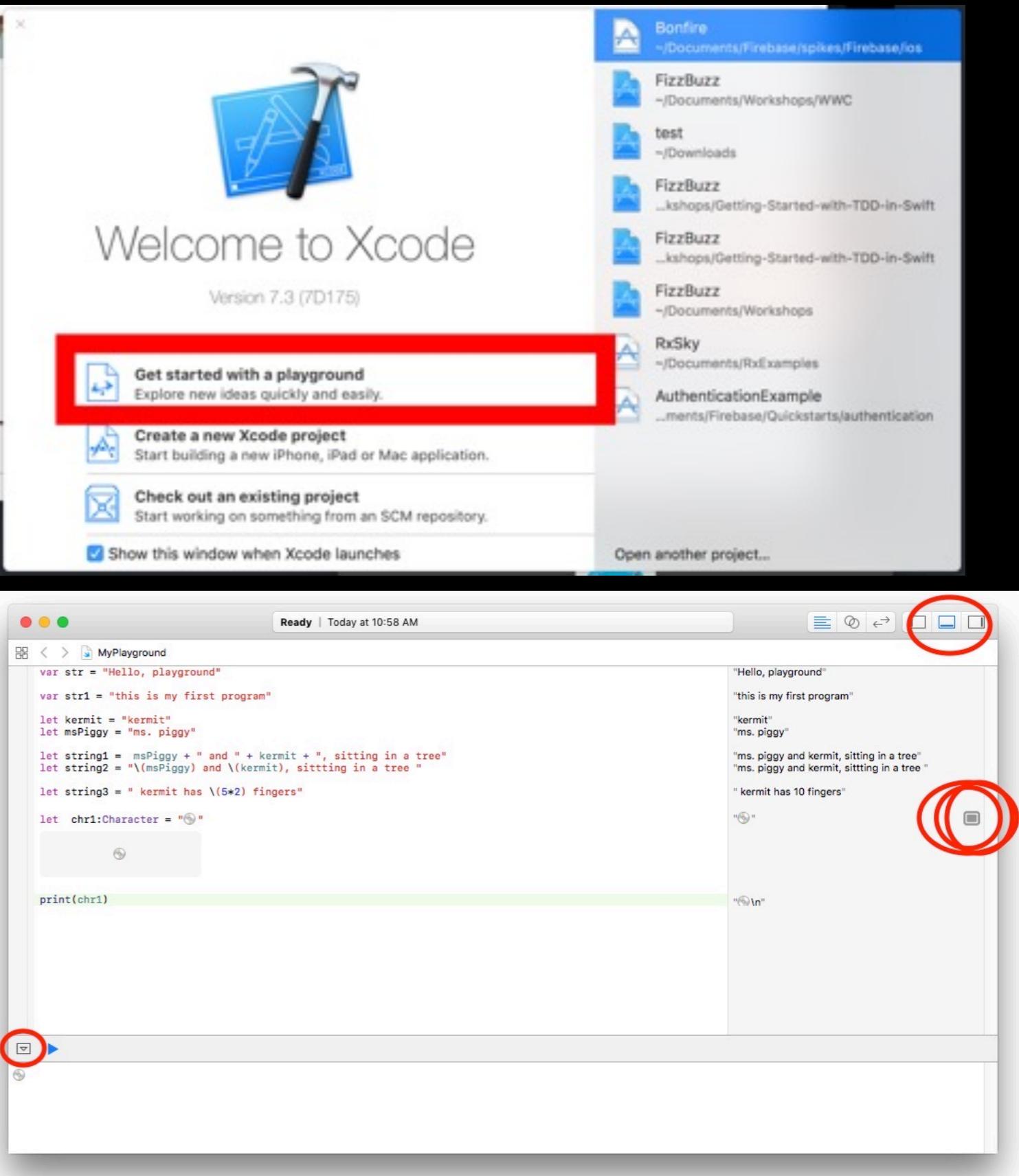
TOPICS TO BE COVERED:

- ▶ How to use Storyboards to design user interfaces
- ▶ How the Model-View-Controller design pattern works in iOS
- ▶ How to use table views, including the new prototype cells and static cells capability
- ▶ How to create your own data model objects
- ▶ What refactoring is, why you should do it, and how to do it
- ▶ How to use Navigation Controllers
- ▶ Using text fields and the keyboard
- ▶ Sending data between view controllers using delegates
- ▶ Saving your app's data into files in the app's Documents folder
- ▶ Using NSUserDefaults to store application settings
- ▶ How to use arrays and dictionaries
- ▶ How to set reminders using local notifications



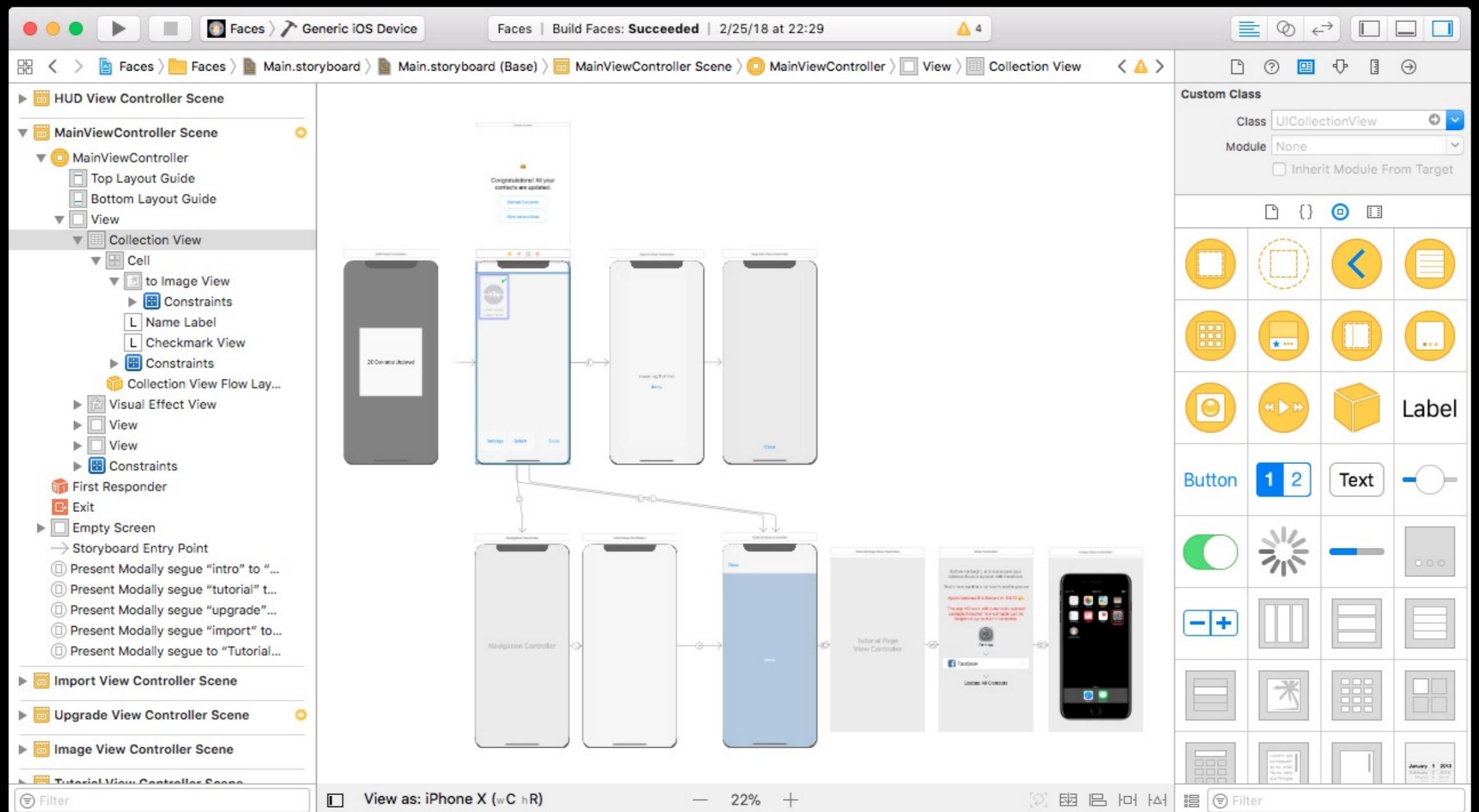
SWIFT Playground

- In Swift Playgrounds you create small programs called “playgrounds” that instantly show the results of the code that you write



Storyboards

- A storyboard is a visual representation of the user interface of an iOS application, showing screens of content and the connections between those screens.





IBActions and IBOOutlets

- **IBAction** and **IBOutlet** are macros defined to denote variables and methods that can be referred to in Interface Builder.

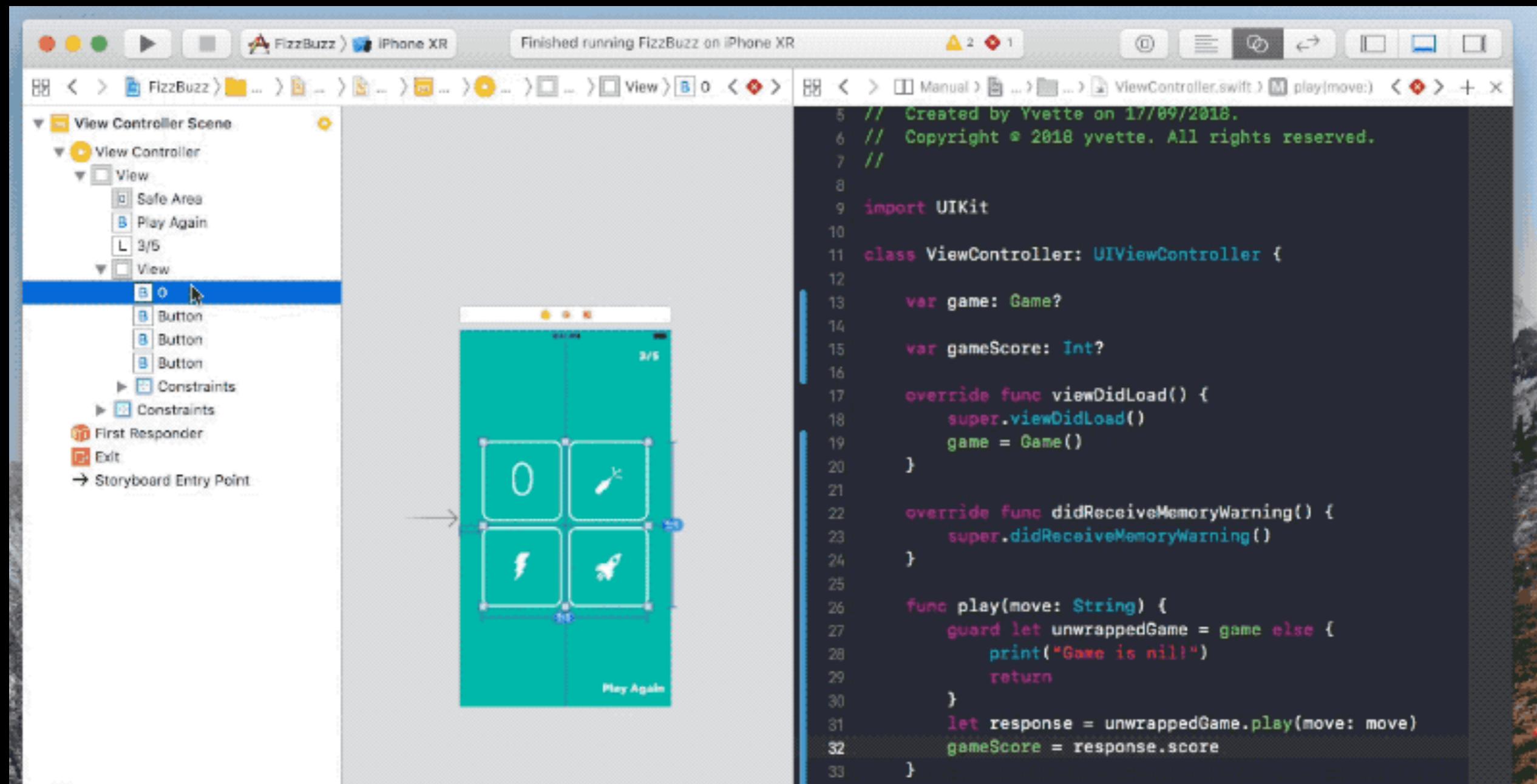
```
class MyViewController: UIViewController {  
    @IBOutlet weak var likeButton: UIButton?  
    @IBOutlet weak var instruction: UILabel?  
  
    @IBAction func likedThis(sender: UIButton) {  
        ...  
    }  
}
```



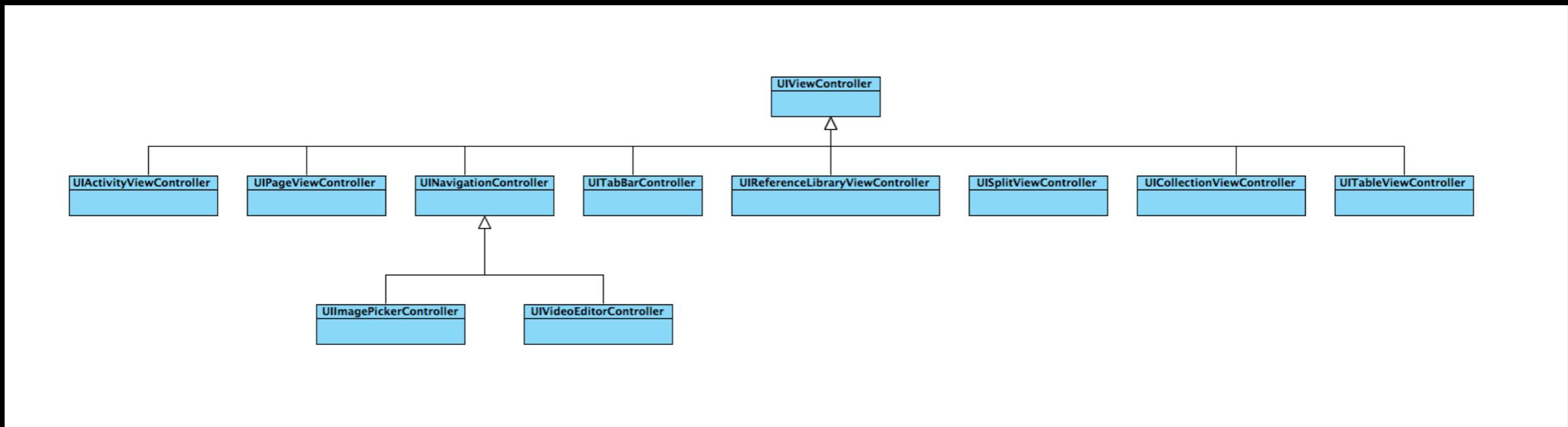
IBActions and IBOUtlets



- **IBAction** and **IBOutlet** are macros defined to denote variables and methods that can be referred to in Interface Builder.



UIViewController



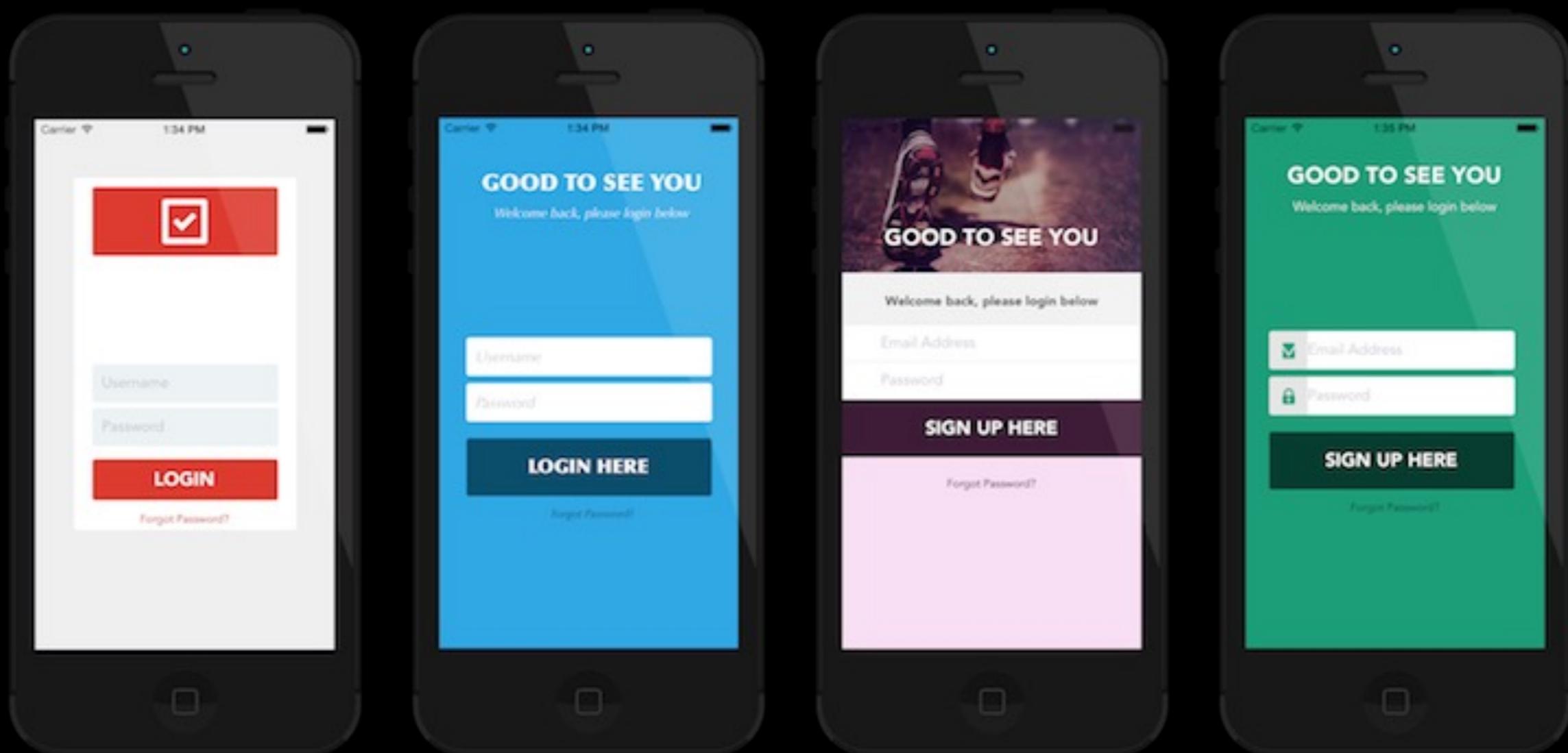
- The **UIViewController** class defines the shared behavior that is common to all view controllers.
- You rarely create instances of the **UIViewController** class directly. Instead, you subclass **UIViewController** and add the methods and properties needed to manage the view controller's view hierarchy.

UIViewController



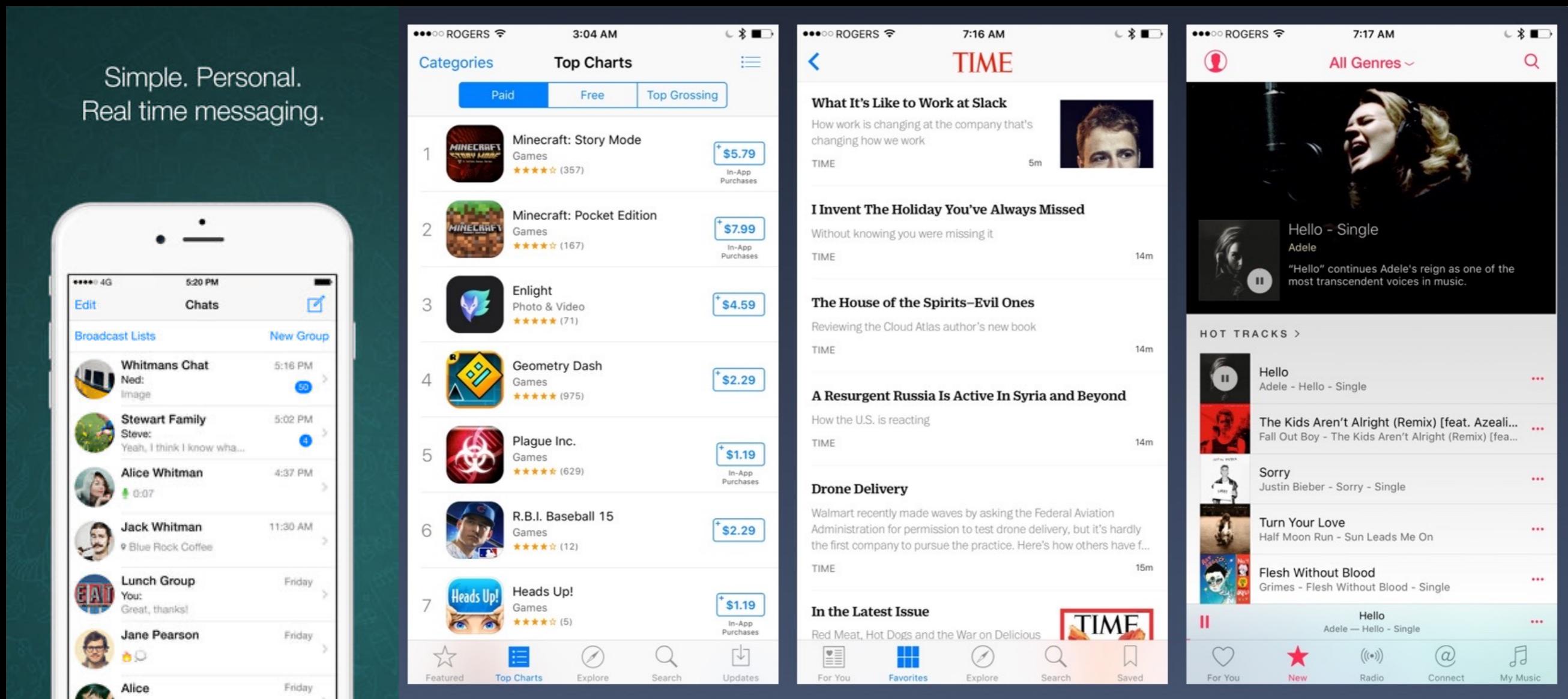
A **view controller's** main responsibilities include the following:

- Updating the contents of the views, usually in response to changes to the underlying data.
- Responding to **user interactions** with views.
- Resizing views and managing the layout of the overall interface.
- Coordinating with other objects—including other **view controllers**—in your app.



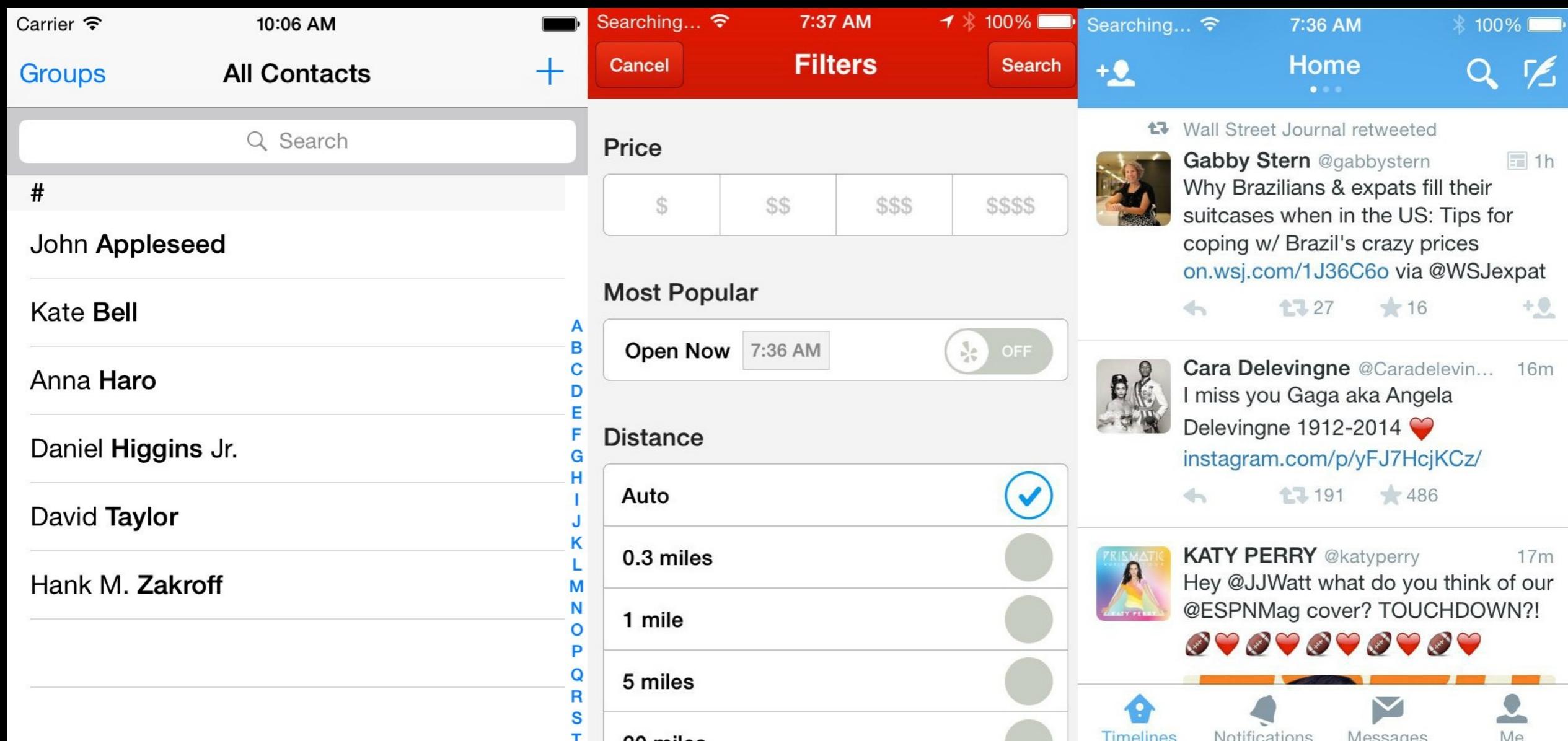
UITableView

- A view that presents data using rows arranged in a single column.
- A table view is made up of zero or more sections, each with its own rows. Sections are identified by their index number within the table view, and rows are identified by their index number within a section.



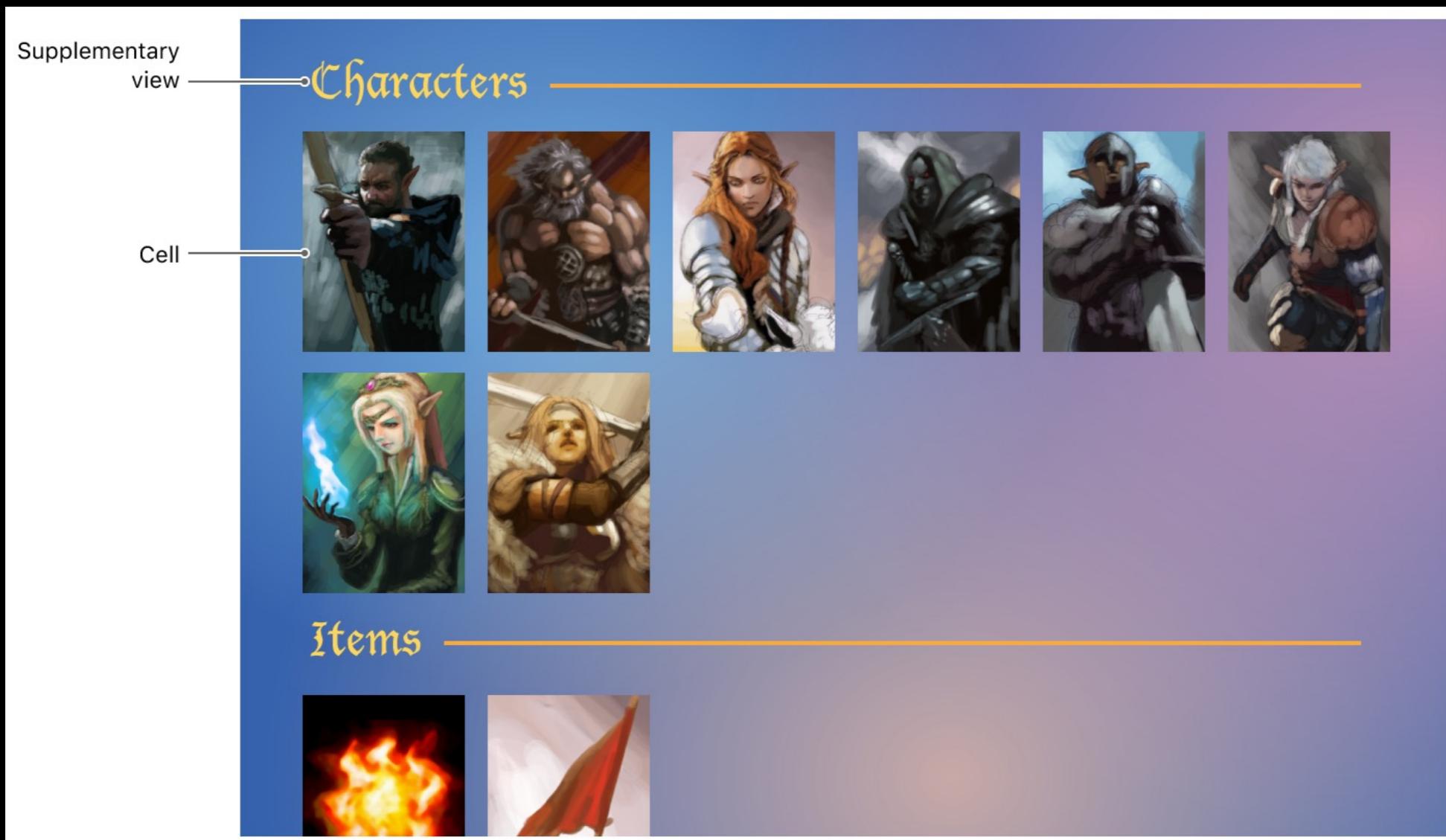
UITableView

- Table Views are everywhere in iOS Design

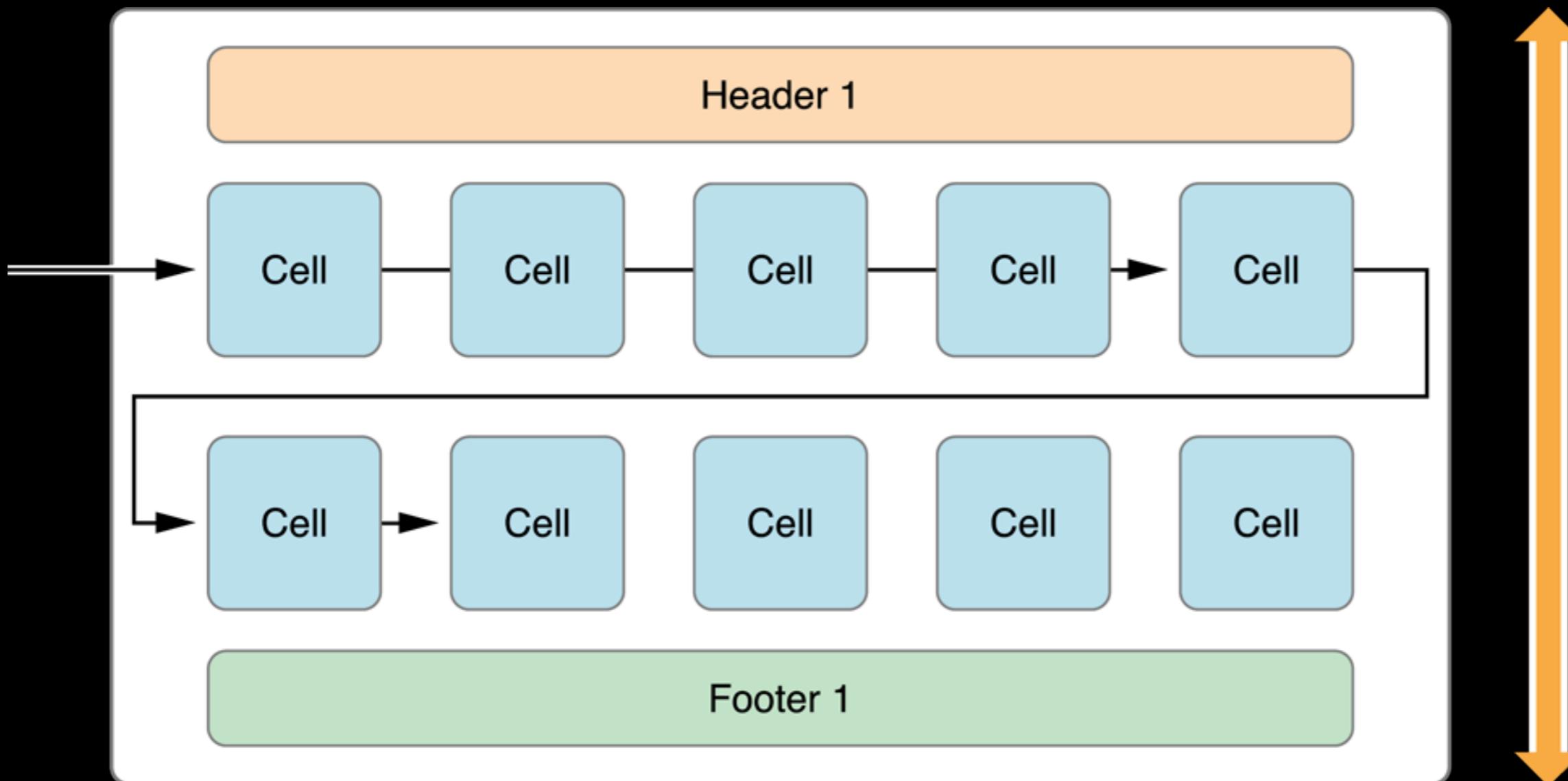


UICollectionView

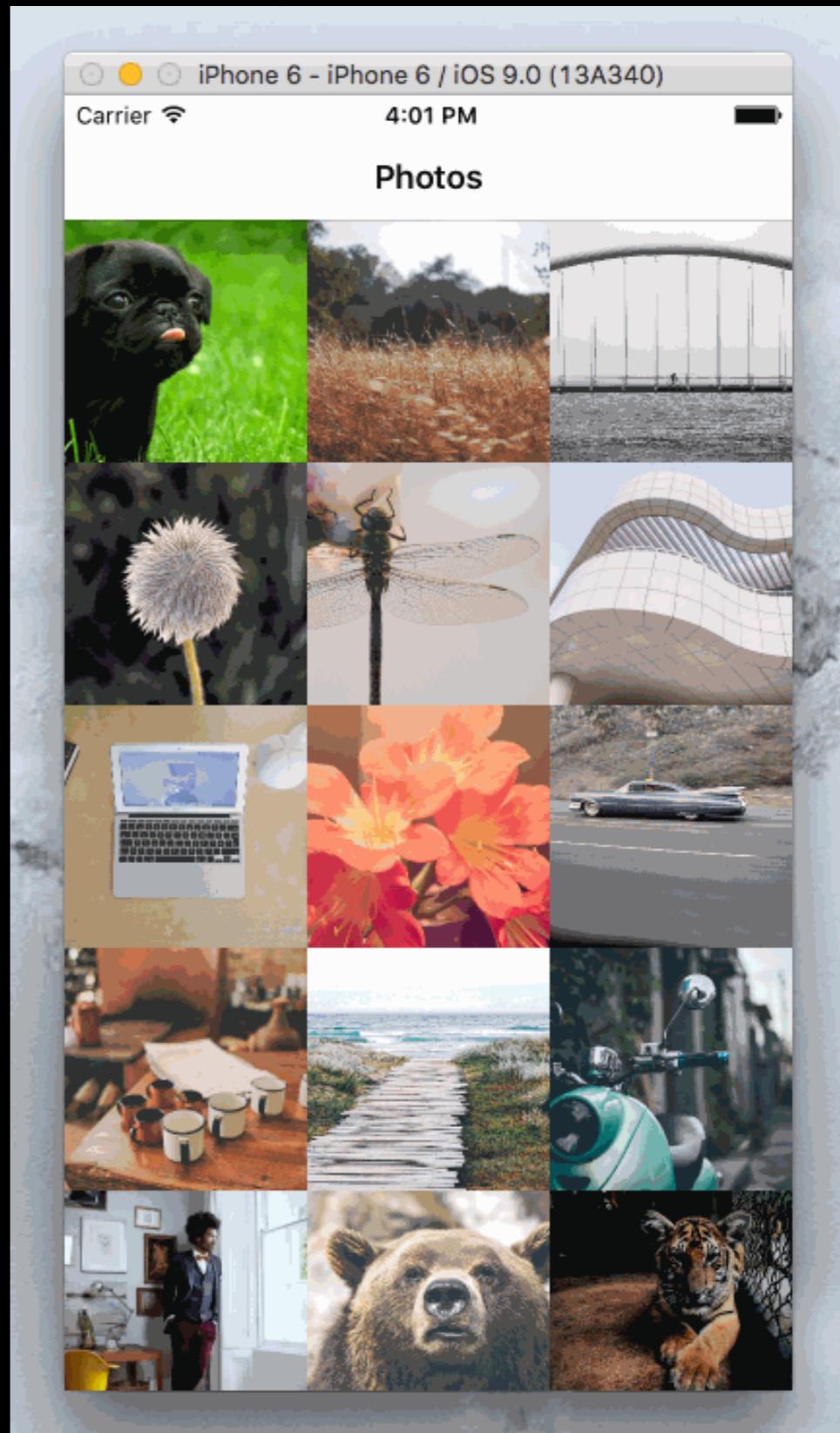
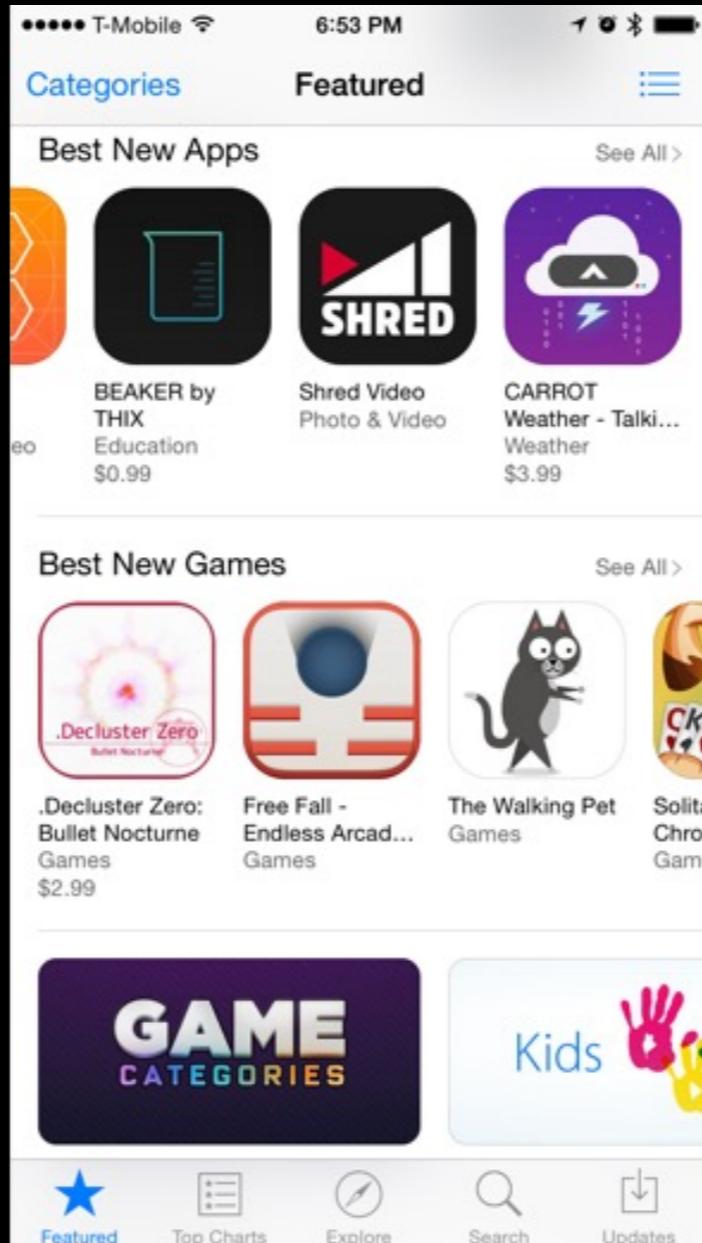
- An object that manages an ordered collection of data items and presents them using customizable layouts.
- The collection view's data source object provides both the content for items and the views used to present that content



UICollectionView



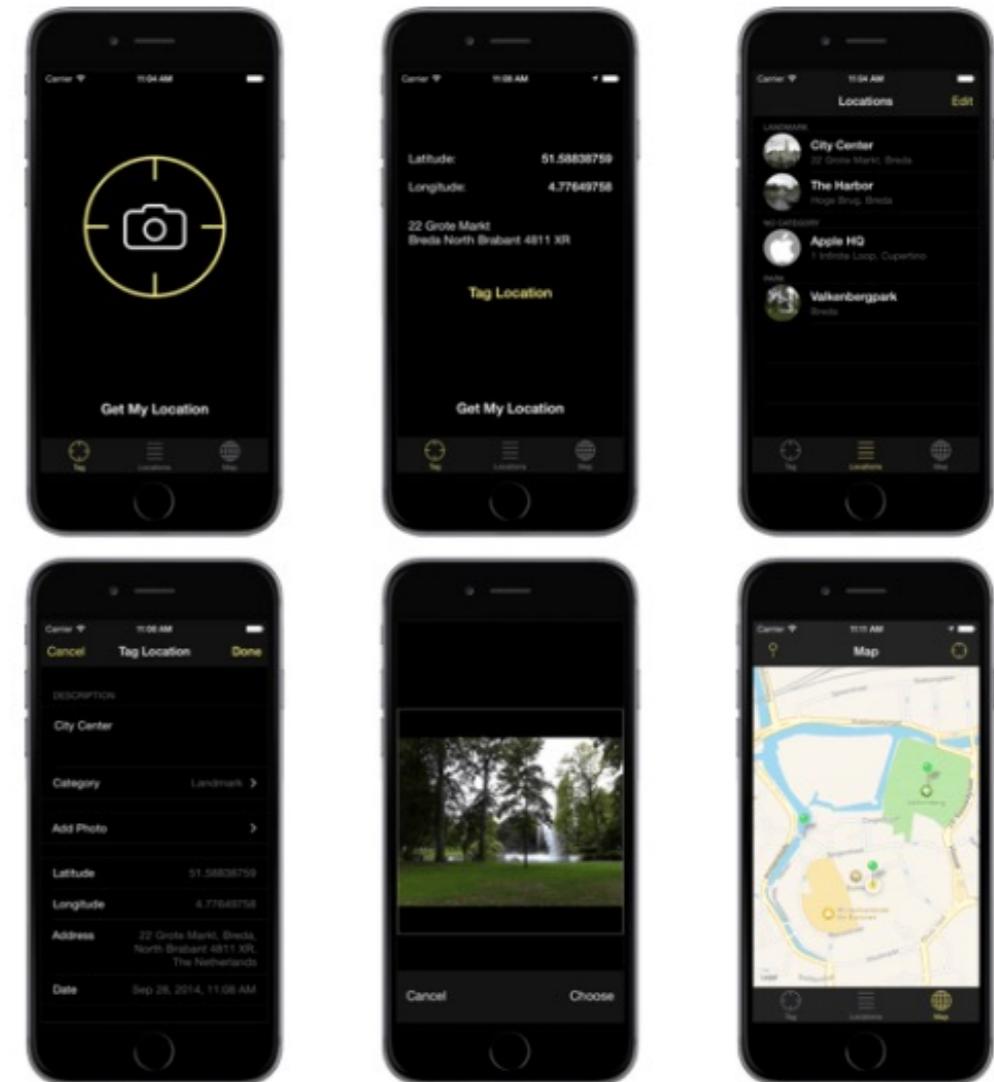
UICollectionView



Module # 2 : iOS INTERMEDIATE

TOPICS TO BE COVERED FOR APP 3 – IOS MAPKIT FOCUSED

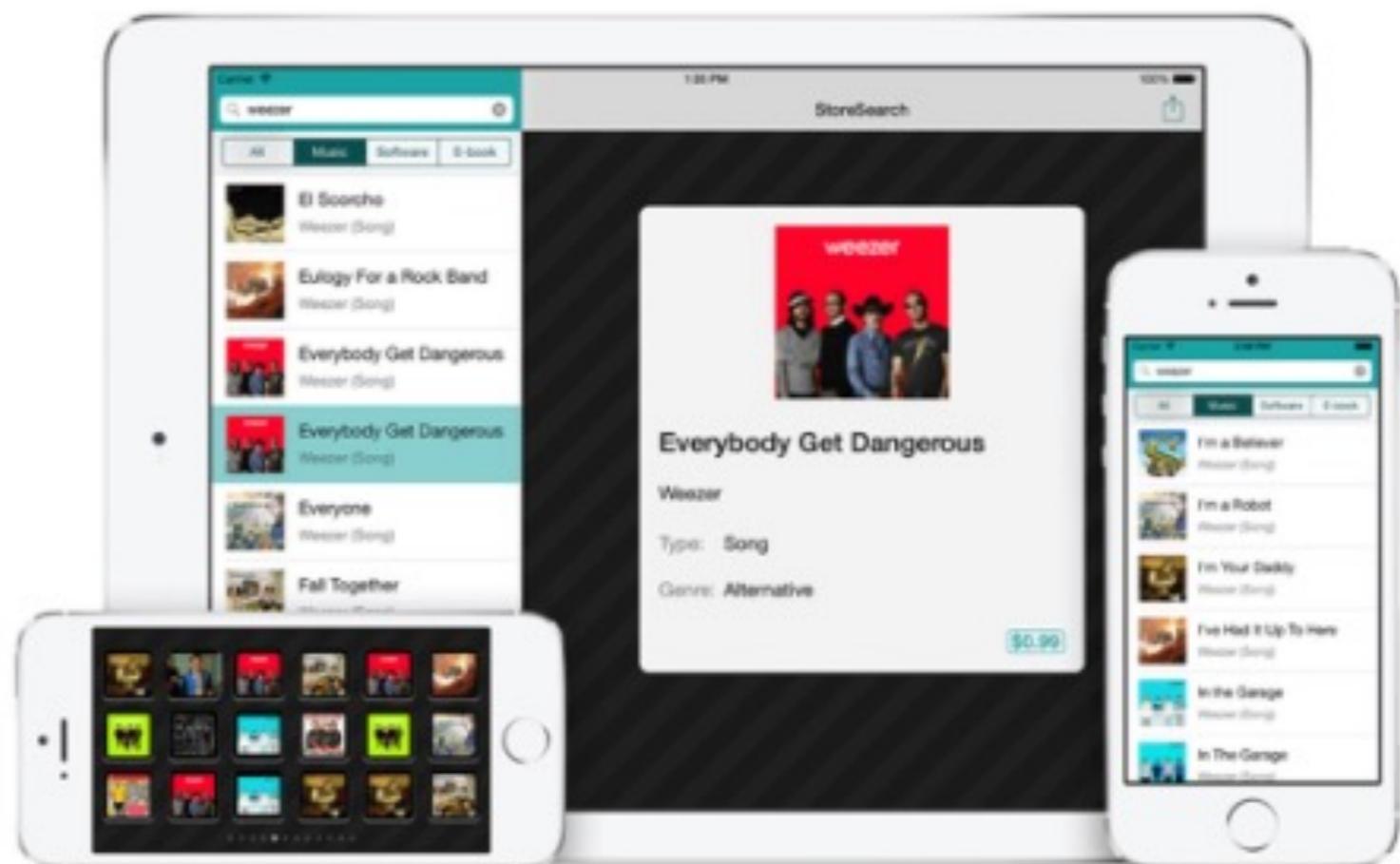
- ▶ How to use the Tab Bar Controller
- ▶ Using the Core Location framework to obtain GPS coordinates and do reverse geocoding
- ▶ How to make your own UIView subclasses and do custom drawing
- ▶ How to use Core Data to persist your objects
- ▶ How to make your table view cell objects
- ▶ How to embed the Map View into your app
- ▶ How to use NSNotificationCenter
- ▶ How to use the camera and photo library
- ▶ How to use “lazy loading” to improve the responsiveness and memory usage of your apps
- ▶ How to play basic sound effects
- ▶ How to make your app look more impressive with UIView-based animations and Core Animation
- ▶ How to use a web service from your apps and how to download images
- ▶ View controller containment: how to embed one view controller inside another
- ▶ Showing a completely different UI after rotating to landscape
- ▶ Cool effects with keyframe animations
- ▶ How to use scroll views and the paging control
- ▶ Internationalization and supporting multiple languages
- ▶ Changing the look of navigation bars and other UI elements
- ▶ Making iPad apps with split-view controllers and popovers
- ▶ Using Ad Hoc distribution for beta testing
- ▶ And finally, submit your apps to the App Store!



Module # 2 : iOS INTERMEDIATE

LEARNING OUTCOMES FOR APP 4 – iOS NETWORKING FOCUSED

- ▶ You will be able to develop a location-aware app that lets you keep a list of spots that you find interesting. In the process, you'll learn about Core Location, Core Data, Map Kit, and much more!
- ▶ Mobile apps often need to talk to web services. You will be able to make a stylish app that lets you search for products on the iTunes store using HTTP requests and JSON.



iOS INTERMEDIATE

LEARNING OBJECTIVE:

TO MAKE YOU COMFORTABLE WITH SWIFT 5 Intermediate Topics

TOPICS TO BE COVERED:

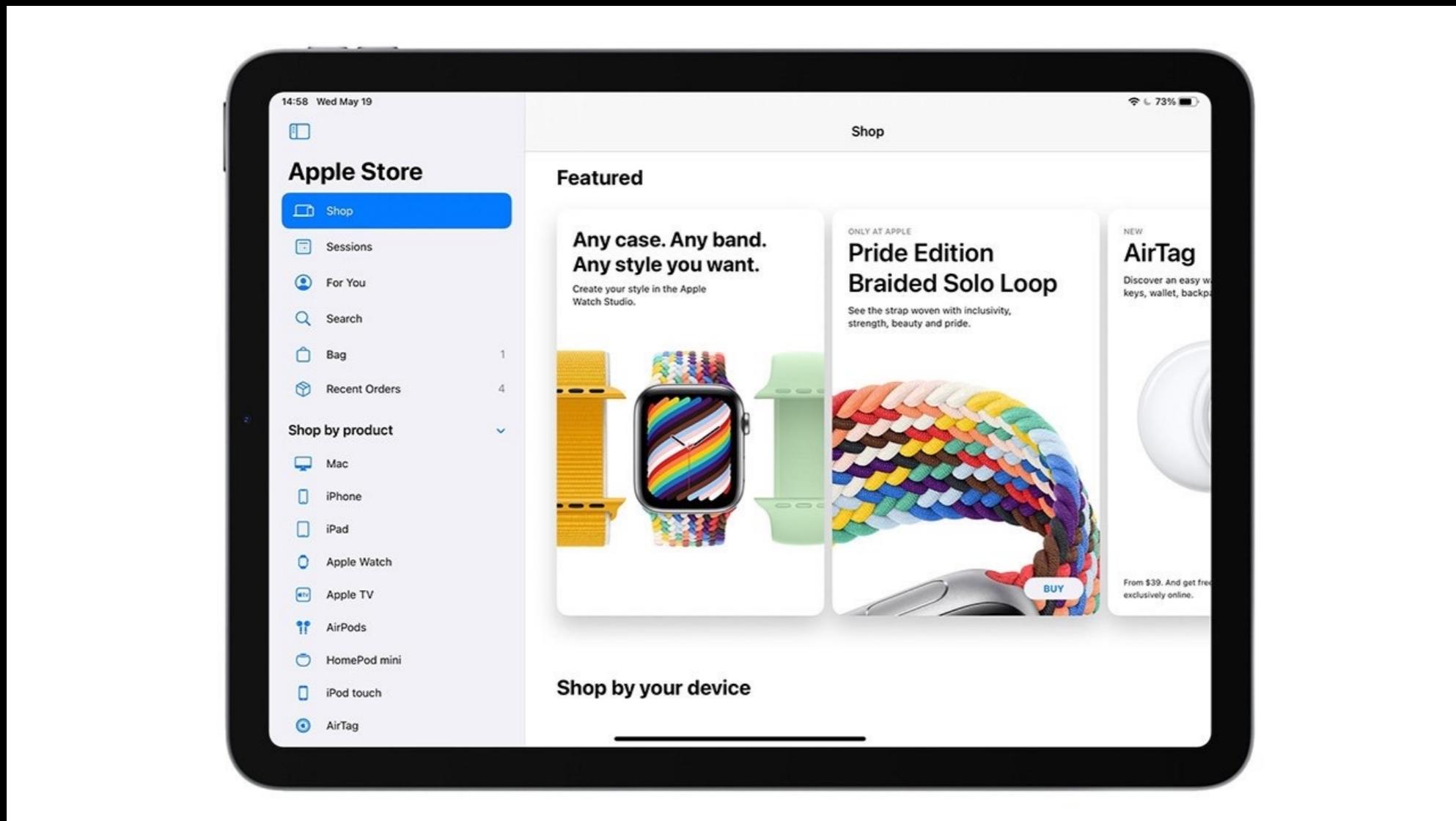
- Sets and Collection Iteration with Closures
- Protocol Oriented Programming in Swift
- Enumerations
- Advanced Classes
- Optionals & Protocols
- Generics



UI Split View Controller

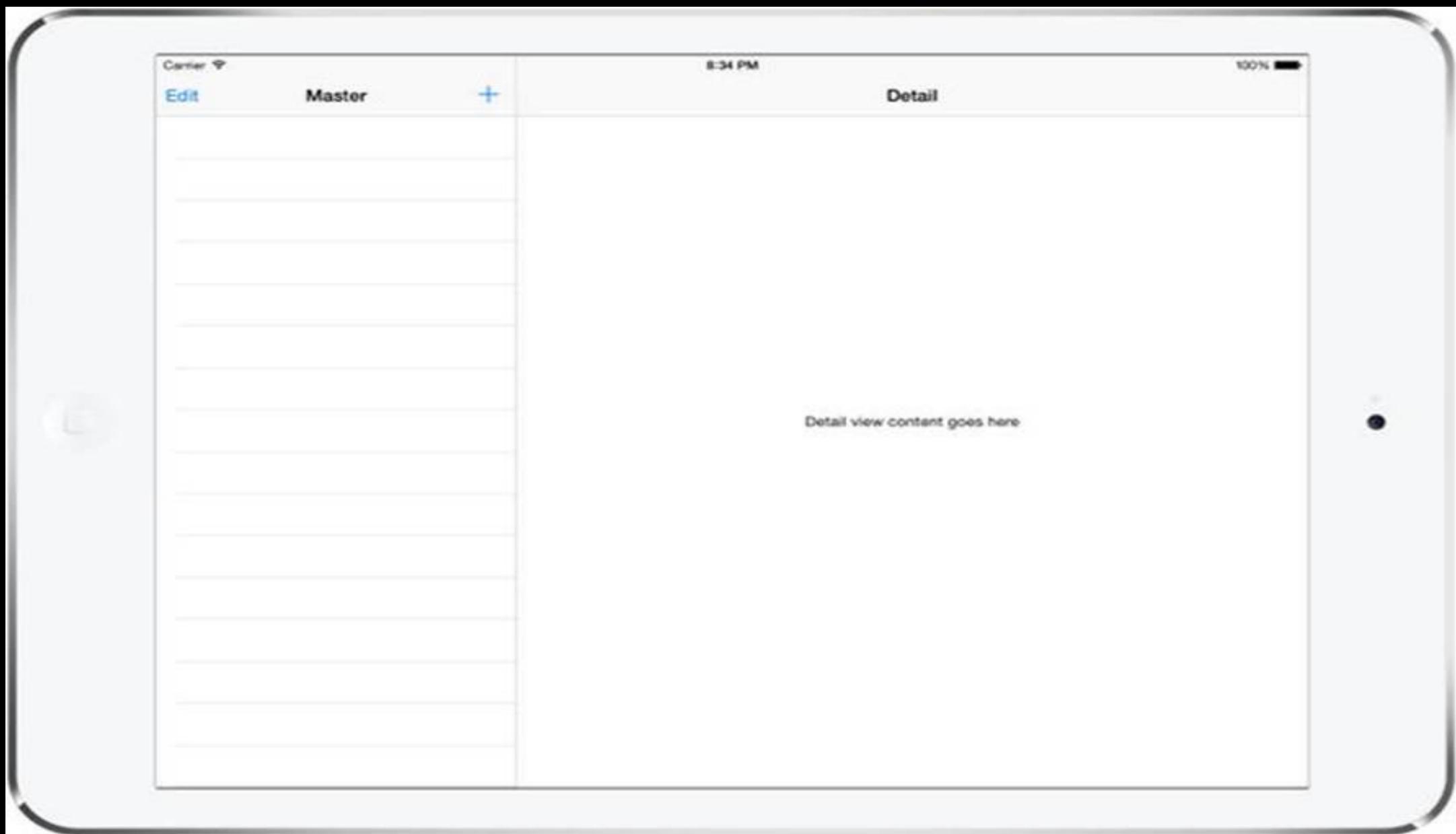


- A container view controller that implements a master-detail interface
- A split view controller is a container view controller that manages two child view controllers in a master-detail interface



UI Split View

The split view lets you carve up the screen into two sections and display a view controller on each side



Module 3: iOS Advance

LEARNING OBJECTIVE:

TO MAKE YOU COMFORTABLE WITH SWIFT 5 Advance Topics

TOPICS TO BE COVERED:

- ▶ Access Control and Code Organization
- ▶ Error Handling
- ▶ Asynchronous Closures and Memory Management
- ▶ Value Types and Reference Types
- ▶ Protocol-Oriented Programming



iOS Advance

LEARNING OBJECTIVES

- ▶ You will learn how to take control of your data in iOS apps using Core Data, Apple's powerful object graph and persistence framework.
- ▶ To make you an expert to deal with Auto layouts

TOPICS TO BE COVERED:

- ▶ Your First Core Data App
- ▶ NSManagedObject Subclasses
- ▶ The Core Data Stack
- ▶ Intermediate Fetching
- ▶ NSFetchedResultsController
- ▶ Versioning and Migration
- ▶ Unit Tests
- ▶ Measuring and Boosting Performance
- ▶ Multiple Managed Object Contexts
- ▶ AutoLayout Basics
- ▶ Size Classes
- ▶ Stack Views
- ▶ Advanced Constraints



iOS Development Tools Hands on Experience

- SourceTree



- BitBucket

- JIRA

- MIRO

- Postman

- Charles

- Zeplin

- Crashlytics / Fabric

- SequelPro

- Xcode

- Confluence

- Proxyman

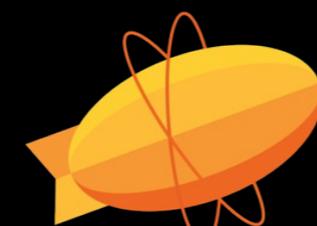


Bitbucket



Atlassian

SourceTree

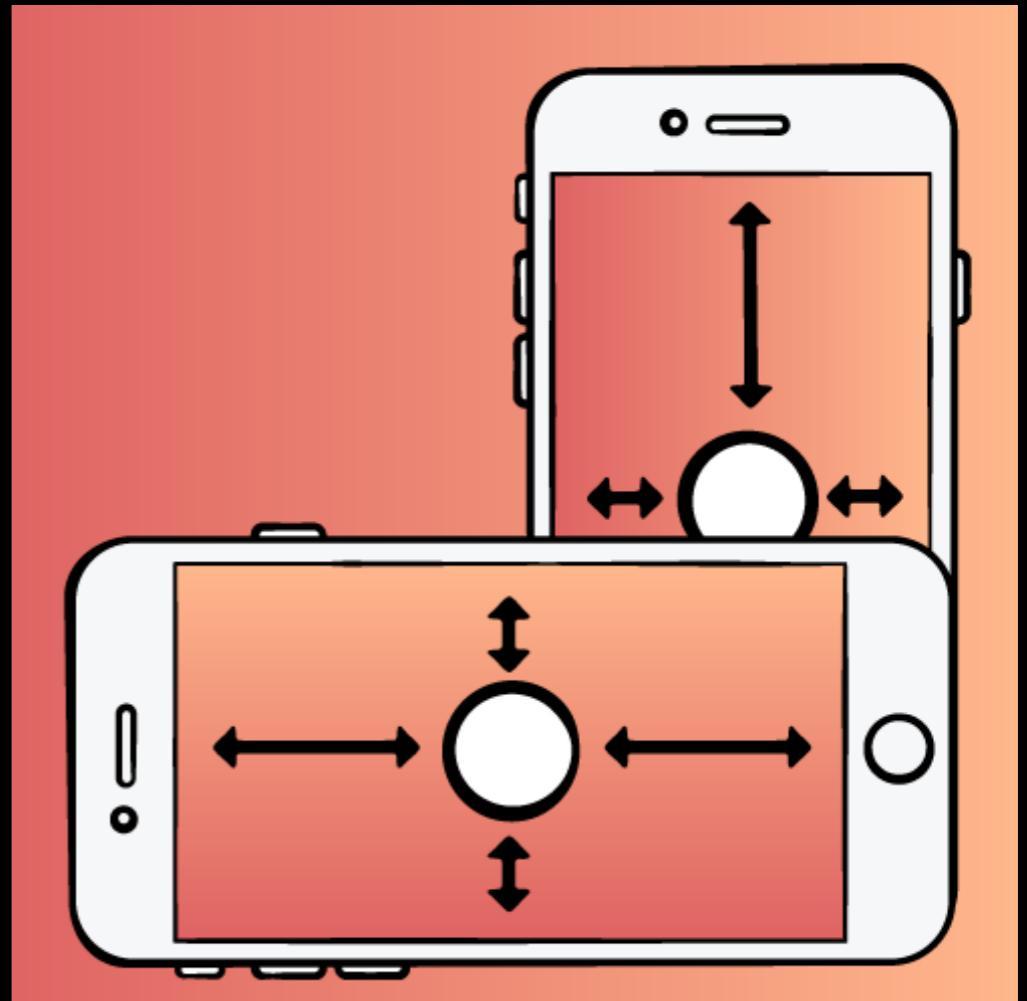
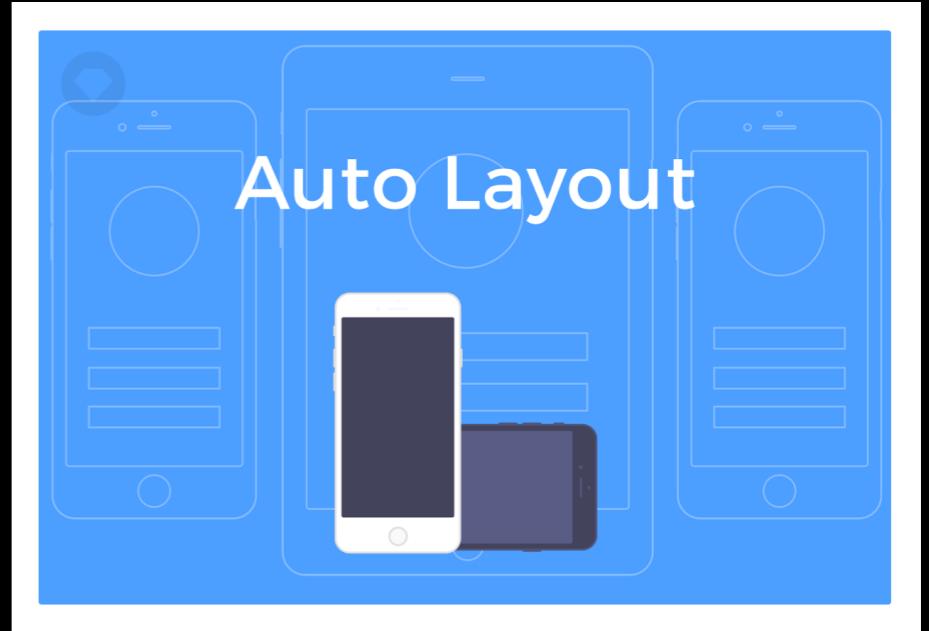


POSTMAN



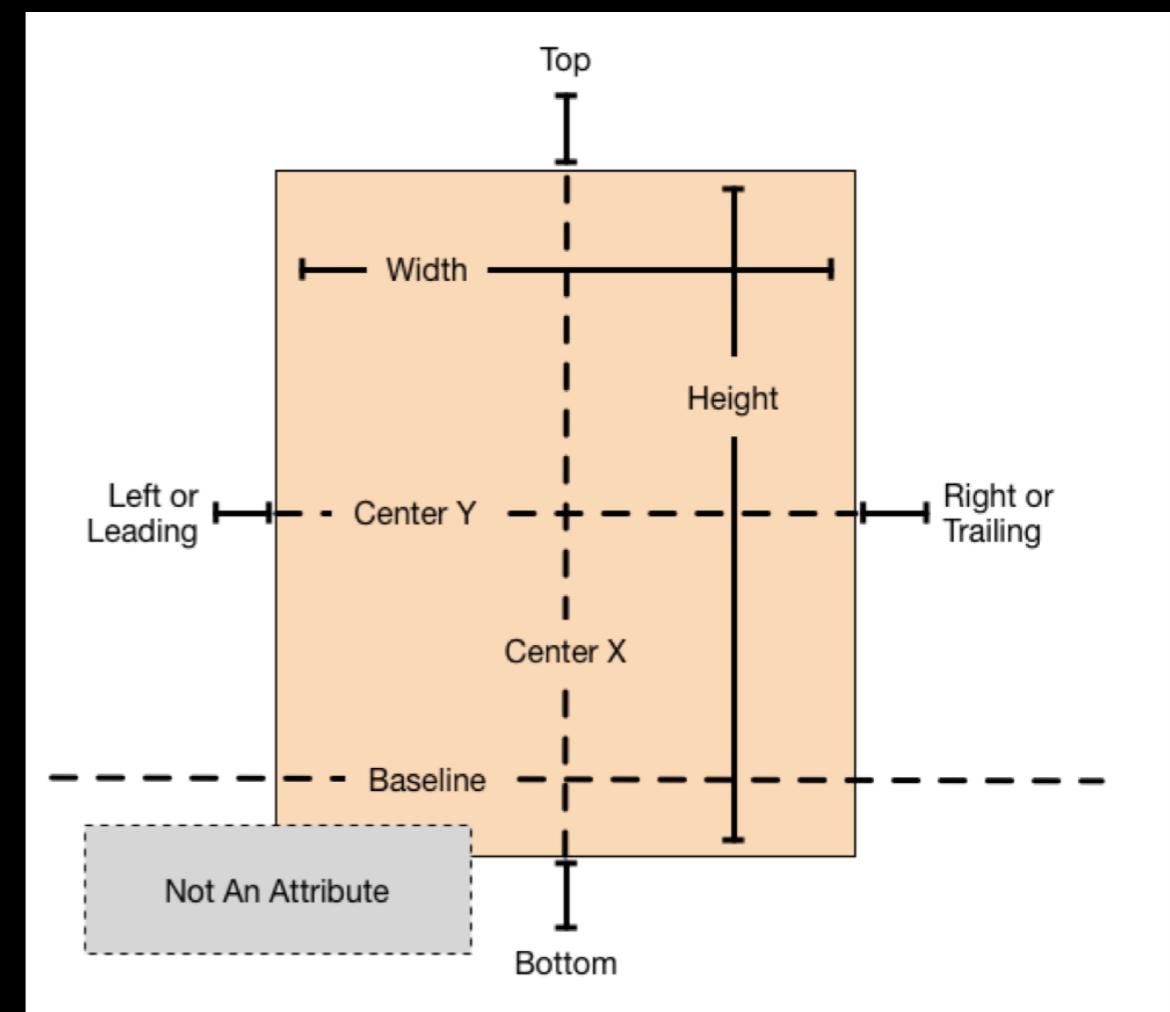
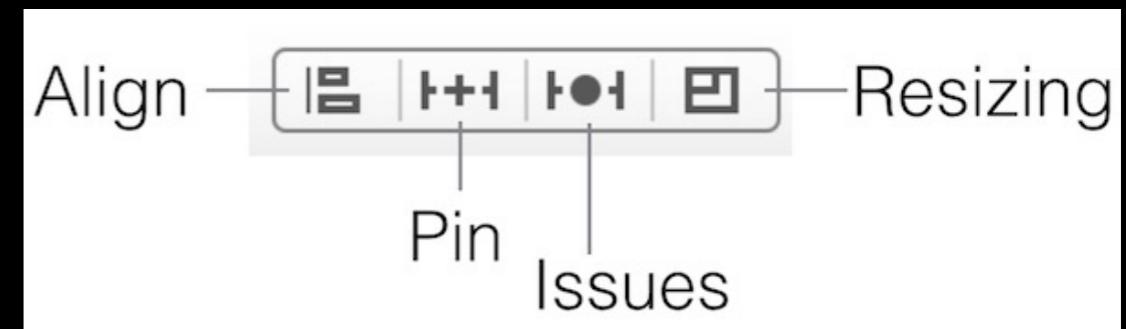
AutoLayout

- Auto Layout is a constraint-based layout system. It allows developers to create an adaptive interface that responds appropriately to changes in screen size and device orientation
- Auto Layout is compatible with many of Apple's most exciting application programming interfaces (APIs), including animations, motion effects, and sprites.



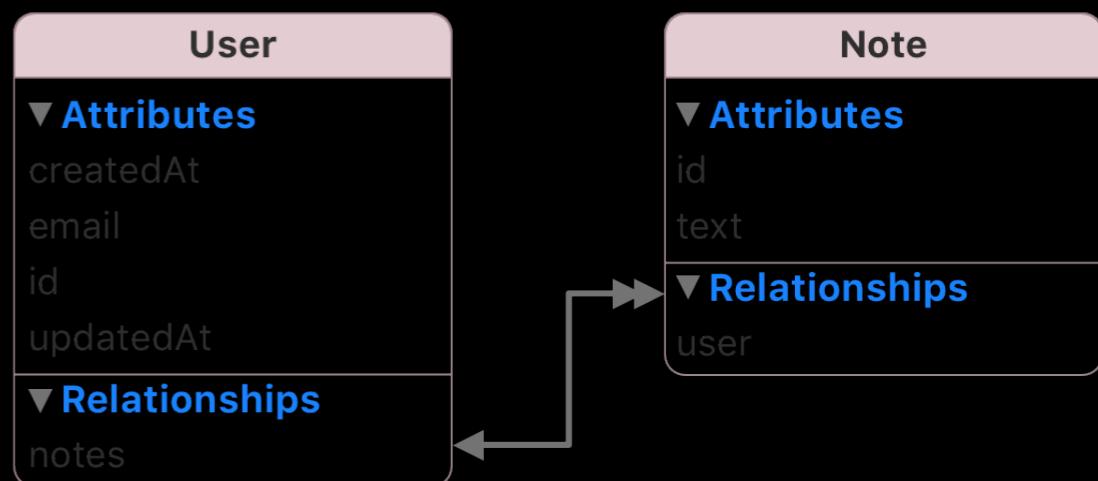
AutoLayout

- **Align** – Create alignment constraints, such as aligning the left edges of two views.
- **Pin** – Create spacing constraints, such as defining the width of a UI control.
- **Issues** – Resolve layout issues.
- **Resizing** – Specify how resizing affects constraints.



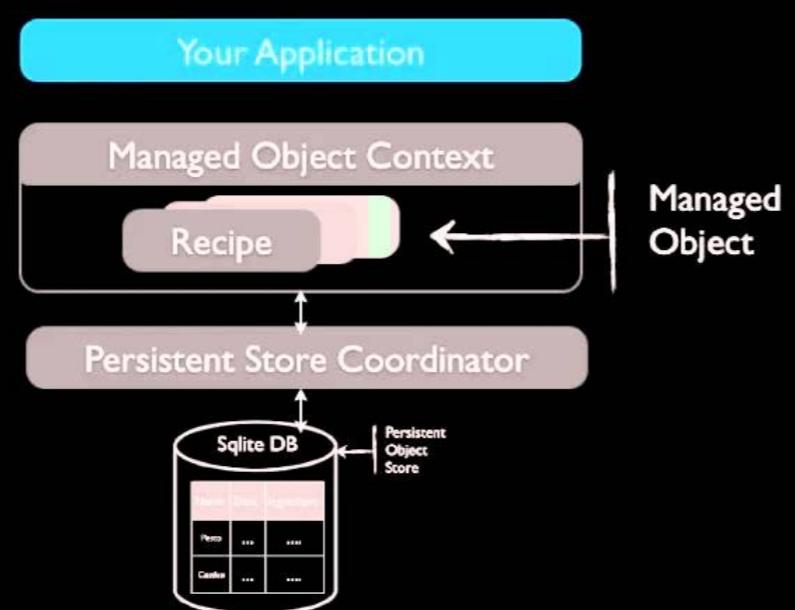
What is Core Data?

Core Data is a framework that you use to manage the model layer objects in your application. It provides generalized and automated solutions to common tasks associated with object life cycle and object graph management, including persistence.

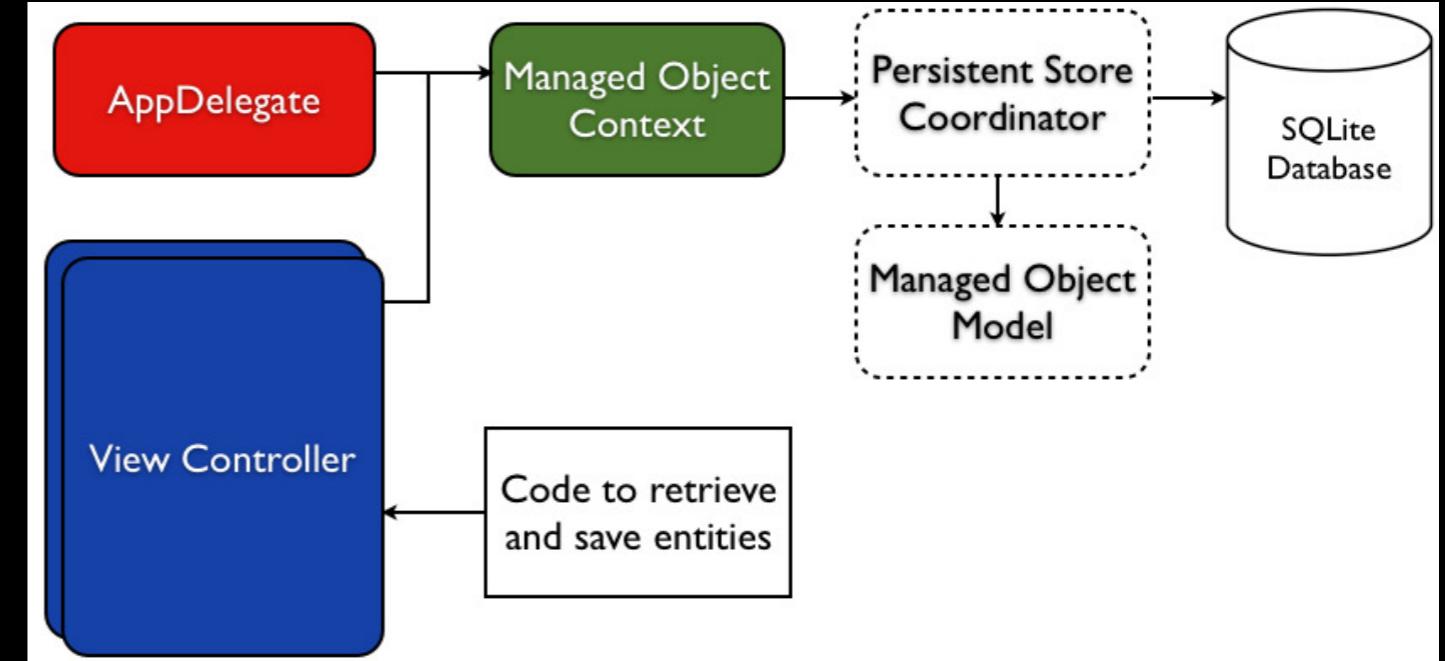


CoreData

Accessing the Persistent Store



Core Data



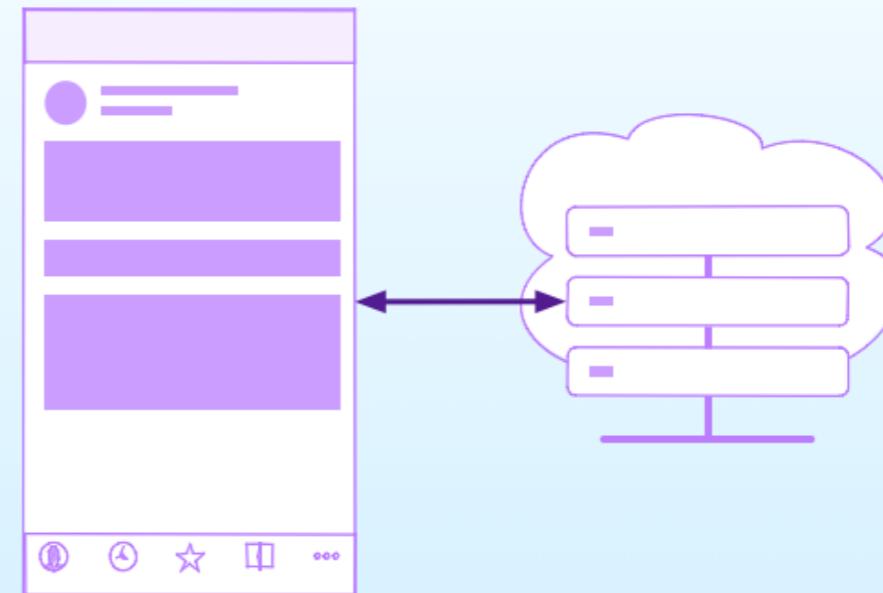
Core Data typically **decreases** by 50 to 70 percent the amount of code you write to support the **model layer**. This is primarily due to the following built-in features that you do not have to implement, test, or optimize:

- Lazy loading of objects, partially materialized futures (faulting), and copy-on-write data sharing to reduce overhead.
- Automatic validation of property values. Managed objects extend the standard key-value coding validation methods to ensure that individual values lie within acceptable ranges, so that combinations of values make sense.
- Grouping, filtering, and organizing data in memory and in the user interface.
- Automatic support for storing objects in external data repositories.
- Sophisticated query compilation. Instead of writing SQL, you can create complex queries by associating an **NSPredicate** object with a fetch request.

iOS Networking

Network Requests and REST APIs in iOS with Swift

A protocol-oriented approach (URLSession, JSON parsing, networking libraries, generics and architectural principles)



In a REST API call we use only a subset of the HTTP methods for the actions we need to perform:

- GET
- POST
- PUT
- DELETE

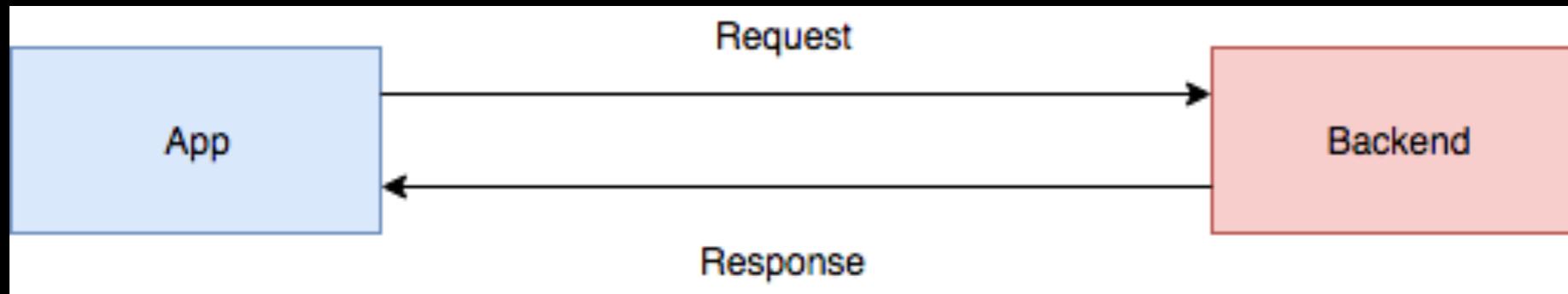
iOS Networking

Most apps require networking to connect to external services and data sources. ‘Networking’ means exchanging information via ‘http’ (Hypertext Transfer Protocol), one of the most used protocols.

Every time you open your browser and retrieve or send data, you are using HTTP as the protocol.

- **NSURLSession:** Networking without third party libraries
- **Alamofire:** Networking with a third-party library to simplify the process

iOS Networking



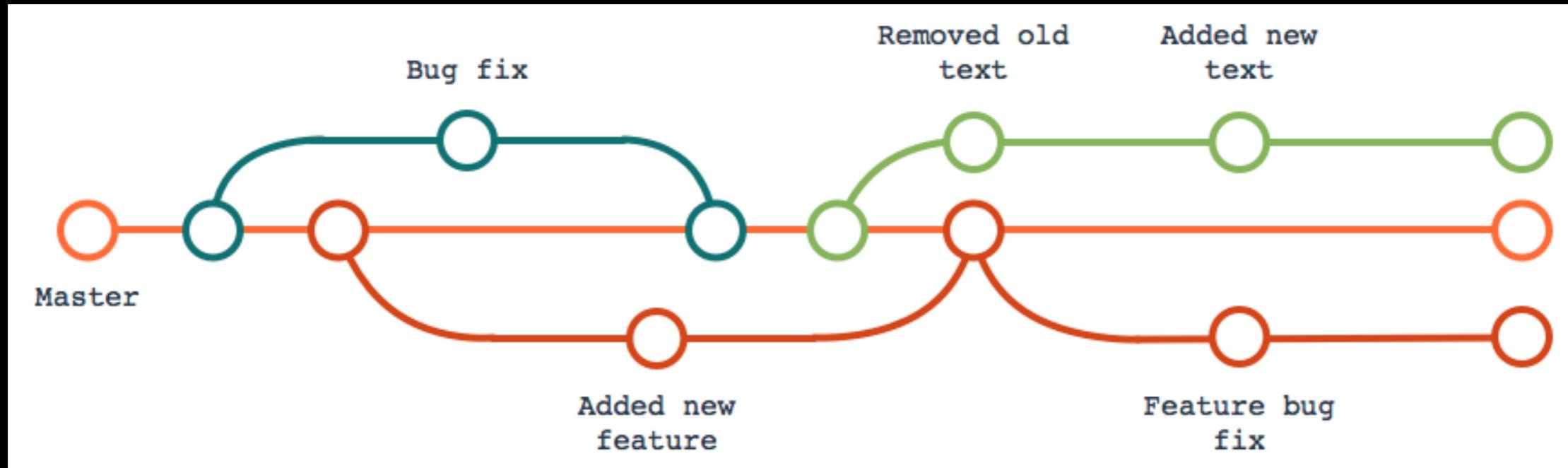
NSURLSession:

NSURLSession is the ‘native’ way to work with remote data sets in Swift. Alamofire is built upon this technology.

Alamofire:

Alamofire is an opinionated library that aims to make networking with iOS simpler. If you are planning to use more custom networking, for example using a different protocol, this library may not suit you. But if you are planning to use traditional REST APIs and simple networking, then I recommend trying Alamofire.

WHAT IS VERSION CONTROL?



- A system that keeps records of your changes
- Allows for collaborative development
- Allows you to know who made what changes
- Allows you to revert any changes and go back to a previous state
- User keep entire code and history on their location machine

HOW DOES GIT WORK?



- Commit
- Repositories
- Branches: Master, Develop
- Pull Request

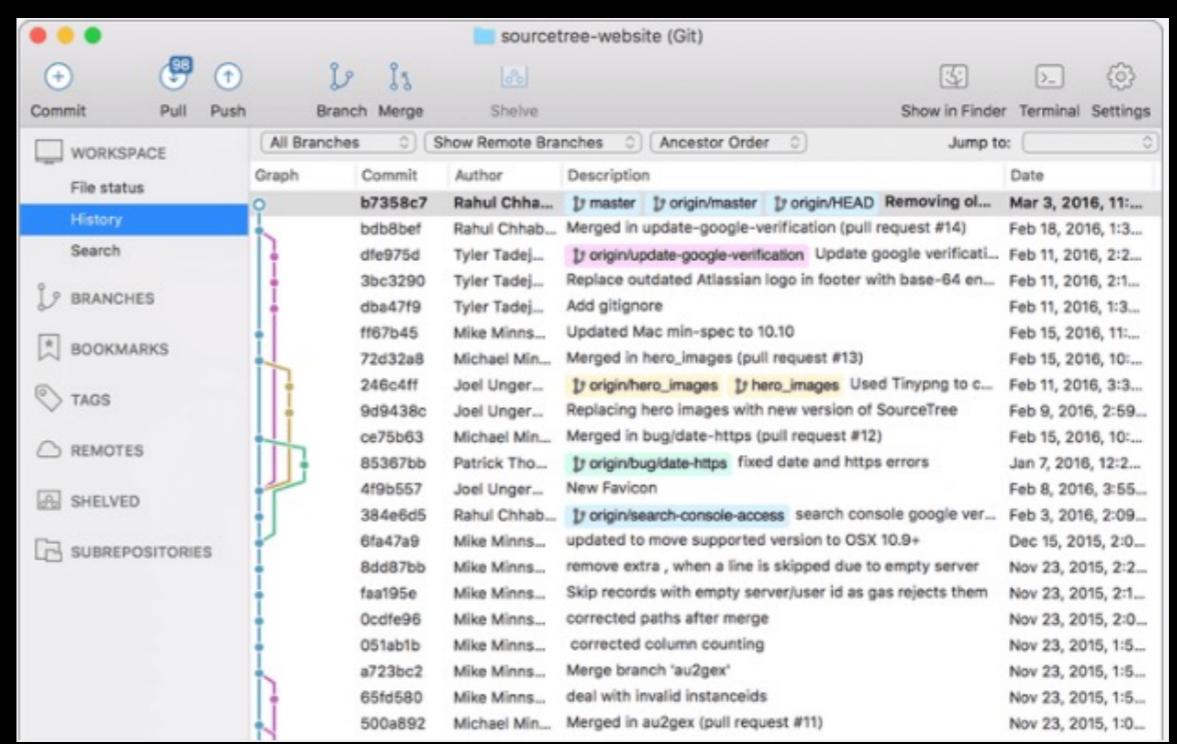
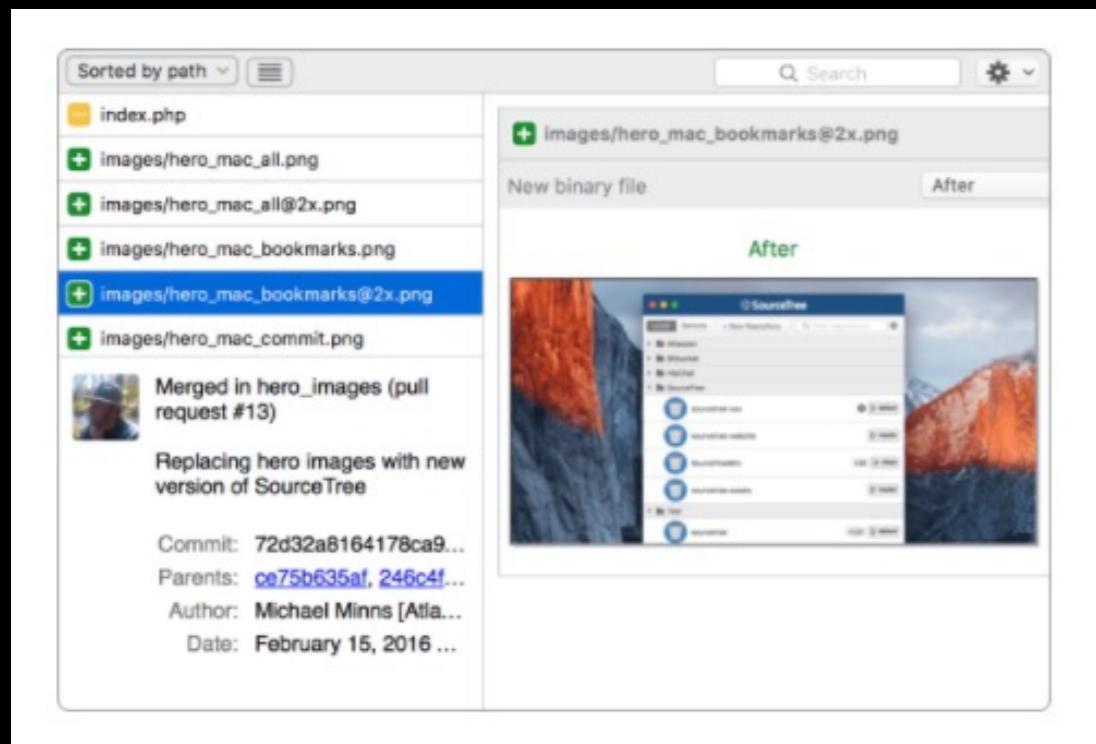
HOW DOES GIT WORK?



- Push
- Pull
- Commit
- Merge

WHAT IS SOURCE TREE?

- Sourcetree simplifies how you interact with your Git repositories so you can focus on coding.).
- A fully-featured GUI that offers an efficient, consistent development process right out of the box. Works with Git and Mercurial.



docs/themes/sourcetree/templates/base.html

Hunk 1 : Lines 1-7

```
1 1 <!DOCTYPE html>
2 2 <html>
3 3   <head>
4 4     <title>SourceTree Help</title>
5 5     <link rel="stylesheet" href="{{ SITEURL }}/{{ THEME_STATIC_DIR }}/css/{{
```

Hunk 2 : Lines 15-24

```
14 15   <footer>
15 16     <a href="http://blog.sourcetreeapp.com/">Blog</a> | <a href="http://w
16 17     - Copyright &copy; 2014
17 18     - <a href="http://www.atlassian.com/"><img href="{{ SITEURL }}/{{ THEME
18 19     + <div id="footer-right">
19 19       Copyright &copy; 2014
```

JIRA: USED FOR PROJECT MANAGEMENT

- JIRA Software is an agile project management tool that supports any agile methodology, be it scrum, kanban, or your own unique flavor.
- From agile boards to reports, you can plan, track, and manage all your agile software development projects from a single tool.
- JIRA Software supports any agile project management methodology for software development



JIRA Software

Teams in Space Scrum: Teams in Space

Version 6.3.3 UNRELEASED

Start: 10 Aug 2015 Release: 9 Oct 2015 Release notes

28 days left

12 Warnings 106 Issues in version 73 Issues done 4 Issues in progress 29 Issues to-do

1-10 of 106

P	T	Key	Summary	Assignee	Status	Development
↑	✓	TIS-111	The revolutionary Afterburner reporting capability	Jeff	DONE	UNDER REVIEW
↑	✗	TIS-110	Afterburner revision VI automation	Bryan	DONE	
↑	✓	TIS-109	Afterburner revision VI script	Sherri	DONE	MERGED
↑	✓	TIS-108	Afterburner revision VI demo	Brandon	DONE	MERGED
↓	✓	TIS-107	Afterburner revision VI prototype	Jay	DONE	
↑	✓	TIS-106	Add video chat interface	Kellie	DONE	1 commit
↑	✗	TIS-105	Create video of launch	Sara	DONE	
↑	✓	TIS-104	Write blog post for launch	Carlos	DONE	3 commits
↑	✓	TIS-103	Review pre-launch checklist	Kelly	DONE	
↓	✓	TIS-102	Afterburner revision VI redundant test	Karen	DONE	

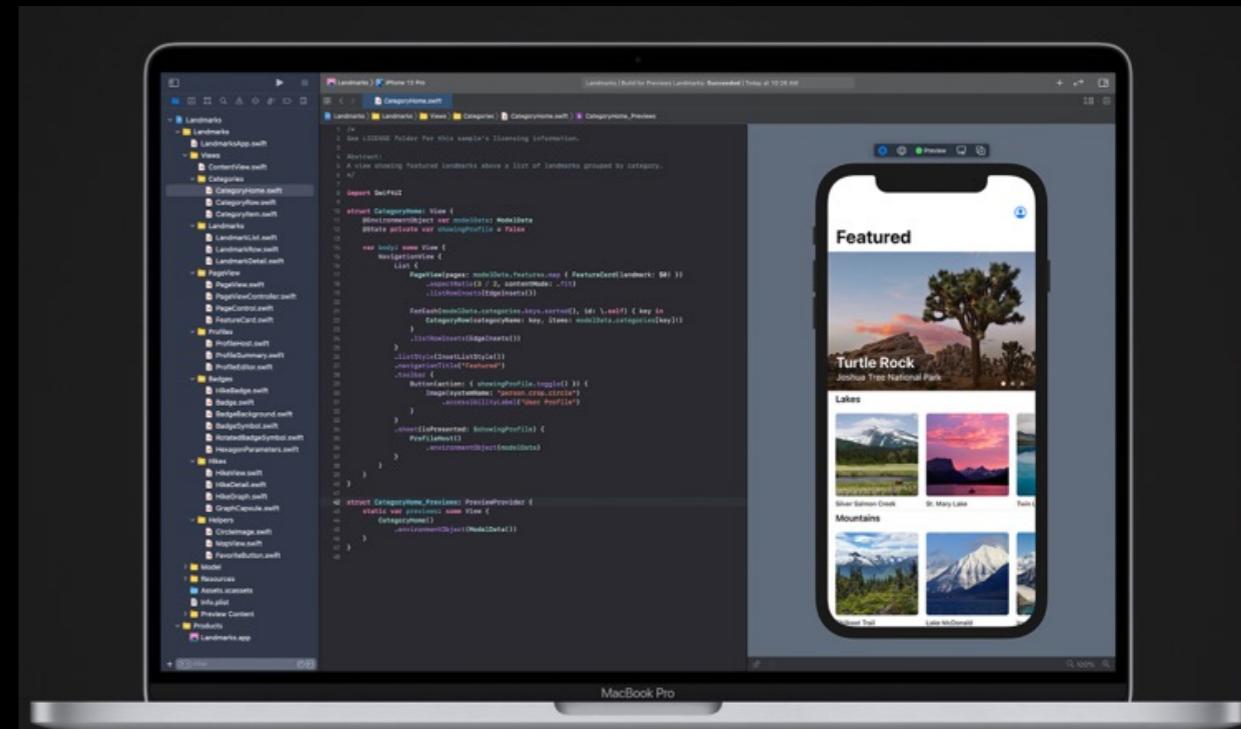
1-10 of 106

SWIFTUI OVERVIEW

- SwiftUI is a modern way to declare user interfaces for any Apple platform.

Create beautiful, dynamic apps faster than ever before.

- SwiftUI is an innovative, exceptionally simple way to build user interfaces across all Apple platforms with the power of Swift. Build user interfaces for any Apple device using just one set of tools and APIs.





TO LEARN SWIFTUI, YOU CAN FOLLOW HIM



A screenshot of a YouTube channel profile for "Abdul KARIM KHAN". The channel banner features a green and white abstract design with a large plus sign. The channel name "ABDUL KARIM KHAN" is displayed in large white letters, with "LEARN, CODE AND DEVELOP" in a smaller white box below it. Below the banner, the URL "linkedin.com/in/abdulkarimkhan/" is shown, along with social media icons for LinkedIn, Instagram, Medium, and GitHub, and the handle "@abdulkarimkhaan". The channel title "Abdul Karim Khan" is prominently displayed in large white text, with a smaller "Subscribed" button and a bell icon below it. The channel has 479 subscribers and 43 videos.

Abdul Karim Khan's YouTube Channel: <https://www.youtube.com/@abdulkarimkhaan>



ABDUL KARIM KHAN

LEARN, CODE AND DEVELOP

linkedin.com/in/abdulkarimkhan/ | @abdulkarimkhaan



SwiftUI Bootcamp For Beginner

Abdul Karim Khan

5 videos 37 views Last updated on 6 Nov 2023



▶ Play all

🔀 Shuffle

Welcome to my SwiftUI Tutorial for Beginners! In this video, we'll cover the basics of creating iOS applications with SwiftUI. Whether you're a designer, marketer, SEO specialist, or from any other professional background, you'll learn how to get started with iOS app development.



Overview and Brief Walkthrough of Xcode - SwiftUI Bootcamp #1

Abdul Karim Khan • 59 views • 3 months ago

Struct, Import and Modifiers - SwiftUI Bootcamp #2

Abdul Karim Khan • 35 views • 3 months ago

Shapes in SwiftUI - SwiftUI Bootcamp #3

Abdul Karim Khan • 29 views • 3 months ago

Frames and Alignment - SwiftUI Bootcamp #4

Abdul Karim Khan • 26 views • 3 months ago

VStack, HStack and ZStack in SwiftUI - SwiftUI Bootcamp #5

Abdul Karim Khan • 34 views • 3 months ago



BENEFITS OF SWIFTUI OVER STORYBOARD

- We no longer have to argue about programmatic or storyboard-based design, because SwiftUI gives us both at the same time.
- We no longer have to worry about creating source control problems when committing user interface work, because code is much easier to read and manage than storyboard XML.
- We no longer need to worry so much about stringly typed APIs – there are still some, but significantly fewer.
- We no longer need to worry about calling functions that don't exist, because our user interface gets checked by the Swift compiler.



PROGRAMMING IN SWIFT UI

The screenshot shows the Xcode interface with the following details:

- Project Structure:** SwiftUIBasics > SwiftUIBasics > ContentView.swift
- Code Editor:** The ContentView.swift file contains Swift code for aContentView. It includes imports for SwiftUI, defines a ContentView struct with a body containing an Image of a heart, and adds gesture recognizers for tap and long press events to toggle color and size changes.
- Preview View:** A preview of an iPhone X simulator showing a large red heart centered on the screen. The simulator's status bar shows the time as 09:41.
- Bottom Bar:** The Xcode bottom bar includes icons for preview, zoom, and other tools, along with a "Preview" button and a zoom percentage of -88%.

```
1 // ContentView.swift
2 // SwiftUIBasics
3 // Created by Simon Ng on 23/10/2019.
4 // Copyright © 2019 AppCoda. All rights reserved.
5
6 import SwiftUI
7
8 struct ContentView: View {
9     @State private var colorChange = false
10    @State private var sizeChange = false
11
12    var body: some View {
13        Image(systemName: "heart.fill")
14            .font(.system(size: 200))
15            .foregroundColor(colorChange ? .yellow : .red)
16            .scaleEffect(sizeChange ? 1.5 : 1)
17            .animation(.default)
18            .onTapGesture {
19                self.colorChange.toggle()
20            }
21            .onLongPressGesture {
22                self.sizeChange.toggle()
23            }
24    }
25
26 }
27
28 }
29
30 struct ContentView_Previews: PreviewProvider {
31     static var previews: some View {
32         ContentView()
33     }
34 }
```

Resource to Learn iOS Development

- <https://developer.apple.com>
- <https://www.udemy.com/course/swiftui-masterclass-course-ios-development-with-swift/>
- <https://www.udemy.com/https://store.raywenderlich.com/products/ios-apprentice>
- <https://store.raywenderlich.com/products/swift-apprentice>
- <https://developer.apple.com/tutorials/app-dev-training/>

YouTube Links, Articles and Mentors to Follow for iOS Development

- Let's Build that App:
<https://www.youtube.com/channel/UCuP2vJ6kRutQBfRmdcI92mA>
- Sean Allen:
<https://www.youtube.com/channel/UCbTw29mcP12YlTt1EpUaVJw>
- Paul Solt: <https://www.youtube.com/user/PaulSolt/videos>
- Swift Weekly: <https://swiftweekly.com/>
- Raywenderlich: <https://www.raywenderlich.com/>
- SWIFT by Sundell: <https://www.swiftbysundell.com/>
- LinkedIn Learning Courses by Steven Lipton:
<https://www.linkedin.com/learning/instructors/steven-lipton>



Swift Apprentice

\$59.99

4.9/5 ★★★★★ 10 reviews · Write a Review

PLATFORM iOS 13 LANGUAGE Swift 5.1 EDITOR Xcode 11

Beginning programming with Swift! This book takes you from beginner to advanced in Swift: Apple's modern programming language for iOS.

[BUY EPUB/PDF \\$59.99](#)

[Buy Paperback \(Amazon\)](#)

iOS Apprentice

\$59.99

4.3/5 ★★★★★ 14 reviews · Write a Review

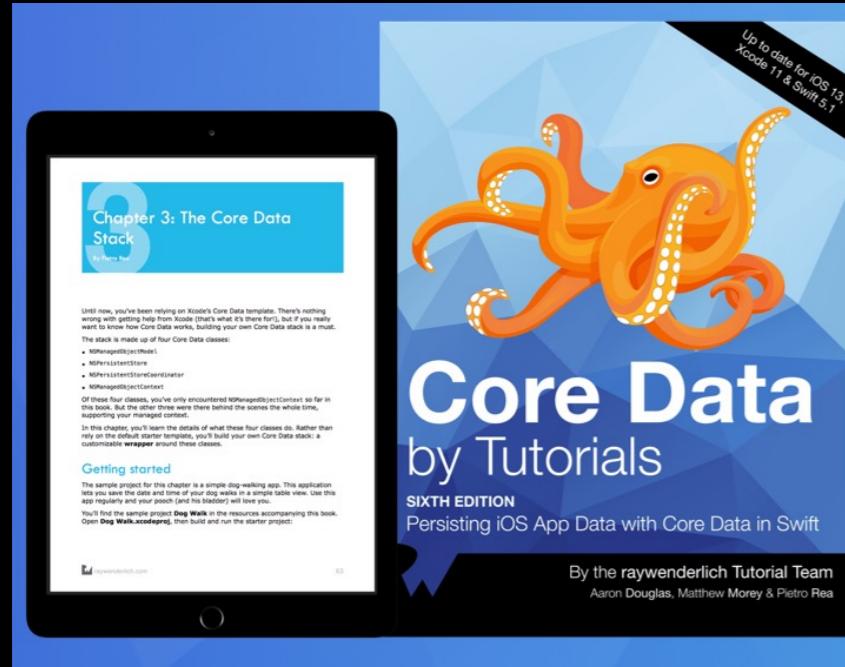
PLATFORM iOS 13 LANGUAGE Swift 5.1 EDITOR Xcode 11

In this iOS programming book for beginners, you'll learn how to build five complete iOS and Swift apps by following easy step-by-step tutorials.

Over 10,000 copies sold!

[BUY EPUB/PDF \\$59.99](#)

[Buy Paperback \(Amazon\)](#)



Core Data by Tutorials

\$59.99

4/5 ★★★★☆ 4 reviews · Write a Review

PLATFORM
iOS 13

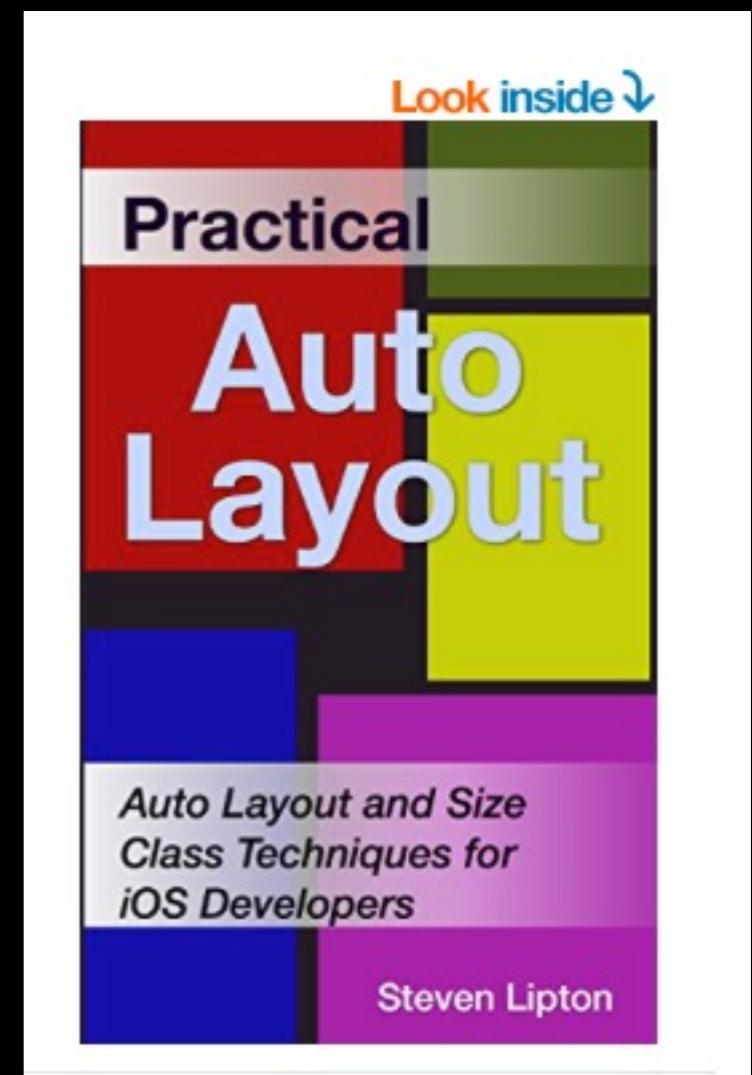
LANGUAGE
Swift 5.1

EDITOR
Xcode 11

In this book, you'll master Core Data in iOS using Swift. Comprehensive coverage of Core Data, from beginner to advanced topics. Covers setting up a Core Data Stack, data modeling, versioning and migration, unit testing, improving performance, and much more.

BUY EPUB/PDF \$59.99

Buy Paperback (Amazon)



The image shows the front cover of the book "ARKit by Tutorials, THIRD EDITION: Building Augmented Reality Apps in Swift". The cover features a blue background with a large illustration of a plesiosaur swimming. A black tablet on the left displays a screenshot of the book's content, specifically Chapter 14: Getting Started With Face-Based AR. The book is authored by the raywenderlich Tutorial Team, Chris Language, Namrata Bandekar, Antonio Bello & Tammy Coron.

ARKit by Tutorials

THIRD EDITION
Building Augmented Reality Apps in Swift

By the raywenderlich Tutorial Team
Chris Language, Namrata Bandekar, Antonio Bello & Tammy Coron

**Up to date for iOS 13,
Xcode 11 & Swift 5.1**

\$59.99

4.6/5 ★★★★★ 5 reviews · Write a Review

PLATFORM iOS 13 **LANGUAGE** Swift 5.1 **EDITOR** Xcode 11

Build immersive and engaging AR apps in ARKit!

Learn how to use Apple's augmented reality framework, ARKit, to bring the magic of AR into your apps as you work through some polished projects:

- Tabletop Poker Dice
- Immersive Sci-Fi Portal
- 3D Face Masking
- Location-Based Content
- Monster Truck Sim
- 2D and 3D detection and tracking app
- ...and more!

The image shows the front cover of the book "SwiftUI by Tutorials, FIRST EDITION: Declarative App Development on the Apple Ecosystem". The cover features a blue background with a large illustration of an oyster shell containing a pearl. A black tablet on the left displays a screenshot of the book's content, specifically Chapter 1: Getting Started with SwiftUI. The book is authored by the raywenderlich.com Tutorial Team, Antonio Bello, Phil Laszkowicz, Bill Morefield & Audrey Tam.

SwiftUI by Tutorials

FIRST EDITION
Declarative App Development on the Apple Ecosystem

By the raywenderlich.com Tutorial Team
Antonio Bello, Phil Laszkowicz, Bill Morefield & Audrey Tam

**Up to date for iOS 13,
Xcode 11 & Swift 5.1**

\$59.99

3.9/5 ★★★★★ 28 reviews · Write a Review

PLATFORM iOS 13 **LANGUAGE** Swift 5.1 **EDITOR** Xcode 11

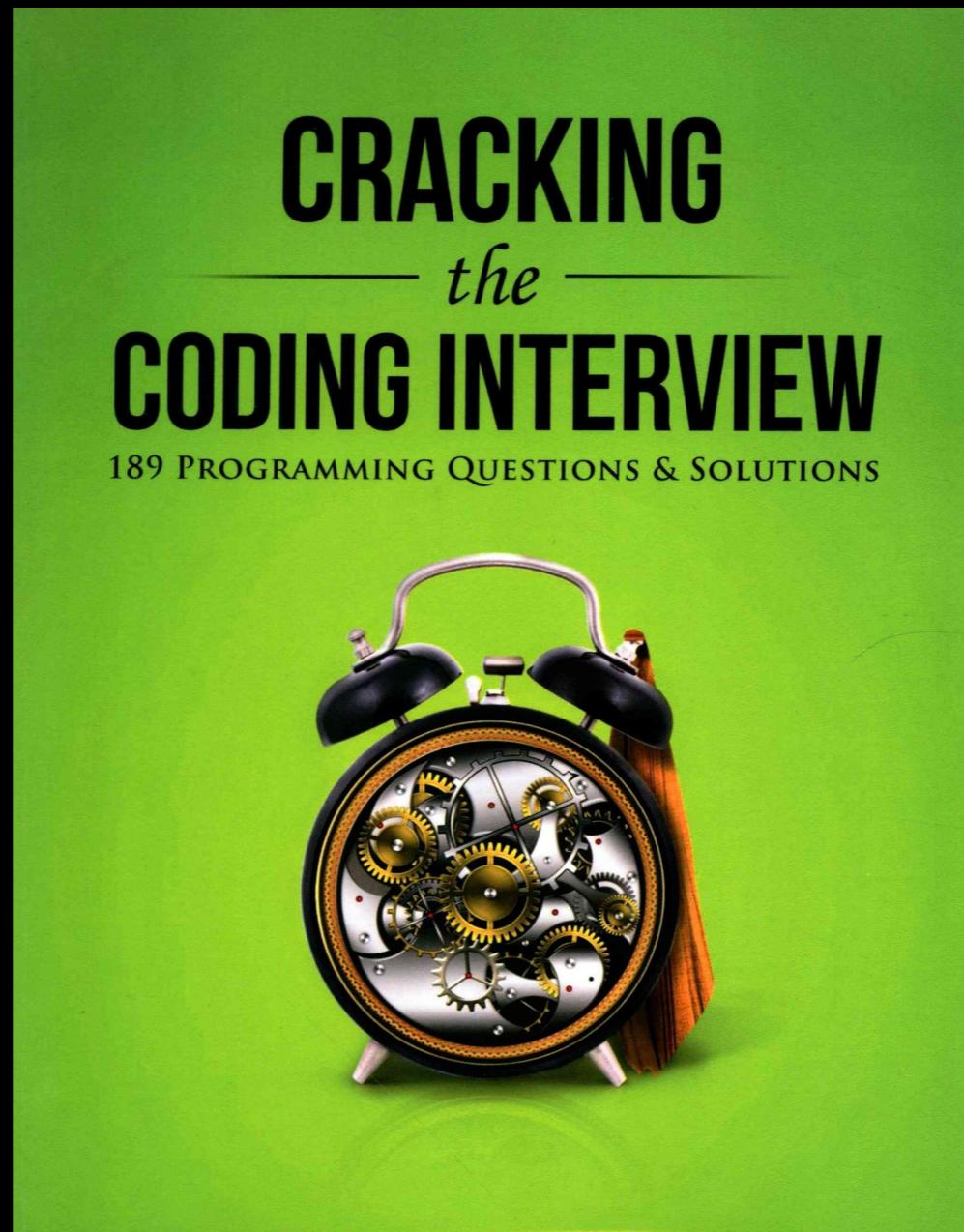
Build fluid and engaging declarative UI for your apps — using less code — with SwiftUI!

With SwiftUI, you can define what your app's UI *should* do with concise, declarative language, and say goodbye to tons of confusing UIKit code.

Build modern, responsive UI and animations that look great on iOS, iPadOS, watchOS, tvOS or even macOS!

BUY EPUB/PDF \$59.99

iOS Interview Preparation



GAYLE LAAKMANN McDOWELL | 6TH EDITION

Author of Cracking the PM Interview and Cracking the Tech Career

General Advice for an iOS Developer

- Be a part of **SWIFT PAKISTAN** to learn from Industry experts
- Read Apple Documentation and **Swift Weekly Daily**
- Watch Apple **WWDC Sessions**
- Make **Written, Audio, or Video Content** to share your knowledge with others
- Find **Good Mentors** and learn from their experiences and mistakes
- Do some **Small-Scale Projects** to build your portfolio and enhance your skill set
- Make **Good Friends** who are doing pursuing similar things
- Take **Calculated Risk**
- Be **grateful for what you have :)**

YOU CAN REACH ME OUT HERE



fahaj.ahmed@live.com



<https://www.linkedin.com/in/fahajahmed1>



<https://www.youtube.com/@FarhajAhmed>

THANK YOU FOR YOUR TIME 😊