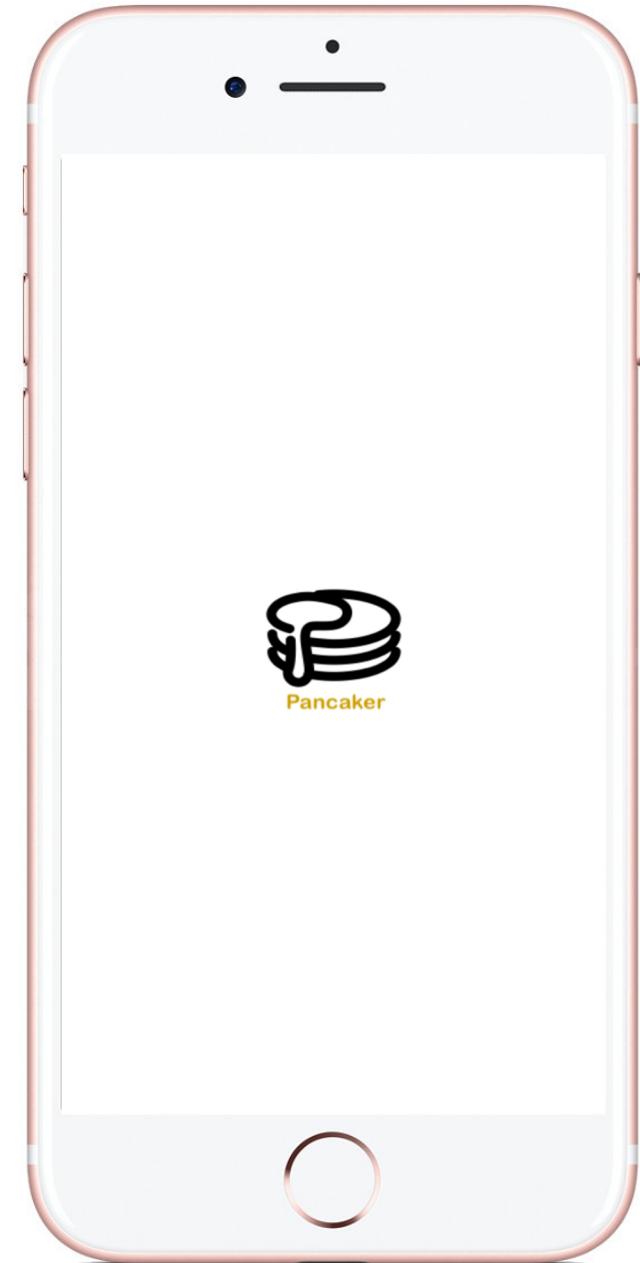
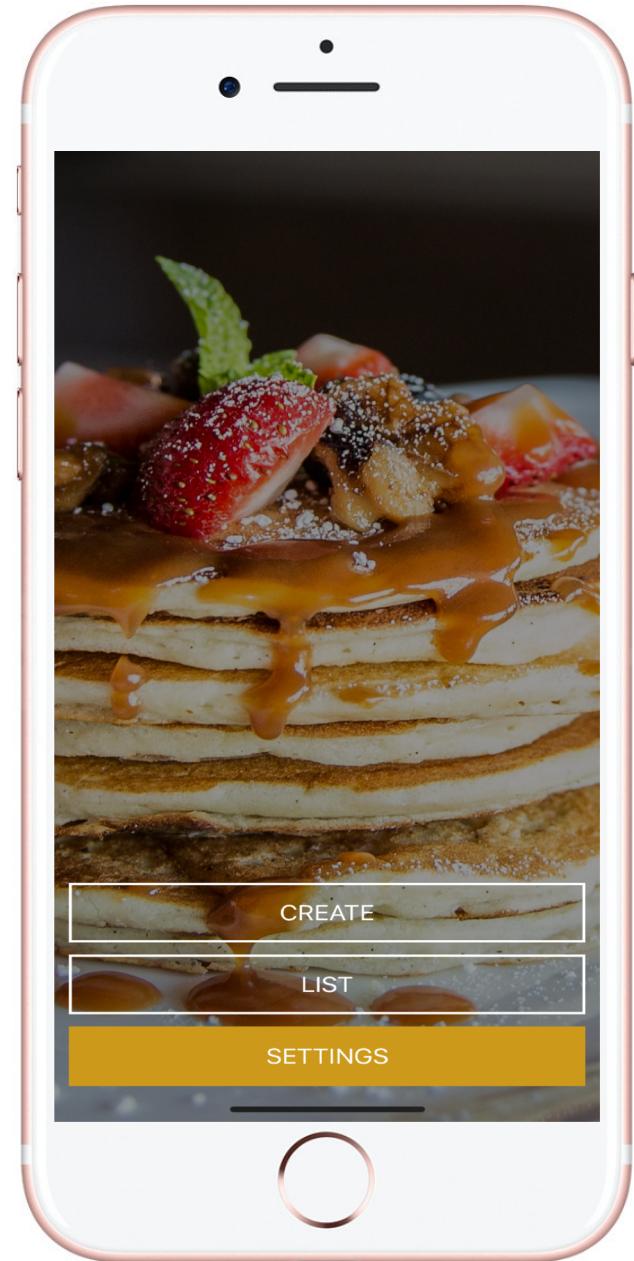


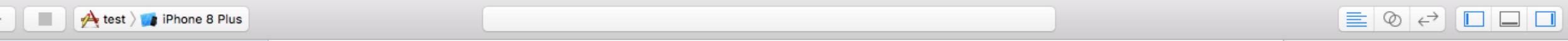
Building a Basic iOS Application



Section 1: Building the First View

- Introduction to Storyboard
- Code vs Storyboard
- UIButton with Layout constraints
- Practice





What is
Storyboard?

The Xcode storyboard editor displays the "Main.storyboard" file. The left sidebar shows the project structure with "Main.storyboard" selected. The storyboard preview area shows a single "View Controller" scene with a white background. The top right corner of the preview shows the time "9:41 AM" and a battery icon. The bottom right corner of the storyboard area has a large gray arrow pointing towards the main content area. The right sidebar is titled "Quick Help" and shows "No Quick Help" with a "Search Documentation" button.

View Controller Scene

- View Controller
- First Responder
- Exit
- Storyboard Entry Point

View Controller

9:41 AM

No Quick Help

Search Documentation

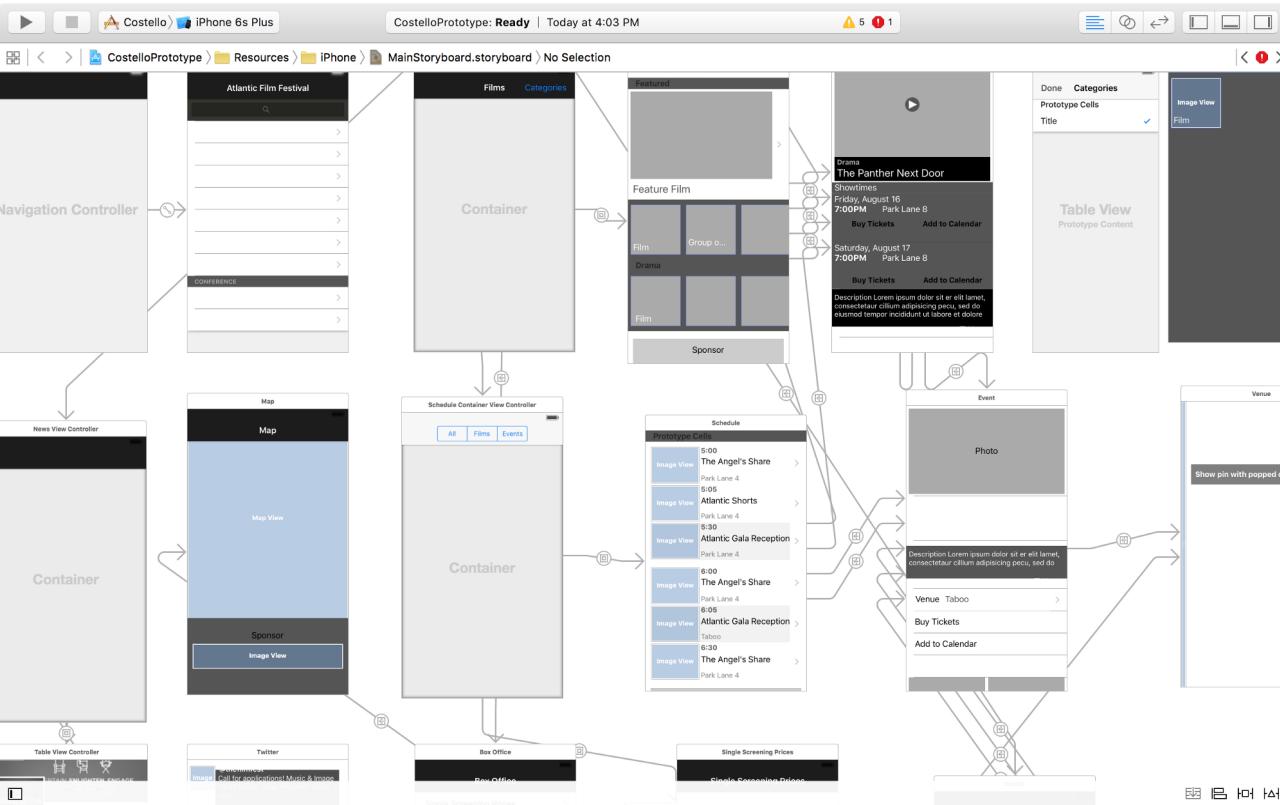
View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Navigation Controller - A

Folly of Storyboard

Code: An Alternative to Storyboard



```
1 +<?xml version="1.0" encoding="UTF-8"?>
2 +<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB" version="3.0"
3 toolsVersion="14109" targetRuntime="iOS.CocoaTouch" propertyAccessControl="none"
4 useAutolayout="YES" useTraitCollections="YES" useSafeAreas="YES" colorMatched="YES"
5 initialViewController="BYZ-38-t0r">
6 +    <device id="retina4_7" orientation="portrait">
7 +        <adaptation id="fullscreen"/>
8 +    </device>
9 +    <dependencies>
10 +        <deployment identifier="iOS"/>
11 +        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin"
12 +            version="14088"/>
13 +        <capability name="Safe area layout guides" minToolsVersion="9.0"/>
14 +        <capability name="documents saved in the Xcode 8 format"
15 +            minToolsVersion="8.0"/>
16 +    </dependencies>
17 +    <scenes>
18 +        <!-- View Controller -->
19 +        <scene sceneID="tne-QT-ifu">
20 +            <objects>
21 +                <viewController id="BYZ-38-t0r" customClass="ViewController"
22 +                    customModule="test" customModuleProvider="target" sceneMemberID="viewController">
23 +                    <view key="view" contentMode="scaleToFill" id="8bC-Xf-vdC">
24 +                        <rect key="frame" x="0.0" y="0.0" width="375" height="667"/>
25 +                        <autoresizingMask key="autoresizingMask" widthSizable="YES"
26 +                            heightSizable="YES"/>
27 +                    <subviews>
28 +                        <button opaque="NO" contentMode="scaleToFill"
29 +                            contentHorizontalAlignment="center" contentVerticalAlignment="center"
30 +                            buttonType="roundedRect" lineBreakMode="middleTruncation"
31 +                            translatesAutoresizingMaskIntoConstraints="NO" id="cX8-sf-NRa">
32 +                            <rect key="frame" x="5" y="632" width="365"
33 +                                height="30"/>
34 +                            <state key="normal" title="Second"/>
35 +                        </button>
36 +                    </subviews>
37 +                </viewController>
38 +            </objects>
39 +        </scene>
40 +    </scenes>
41 +</document>
```



Code Along

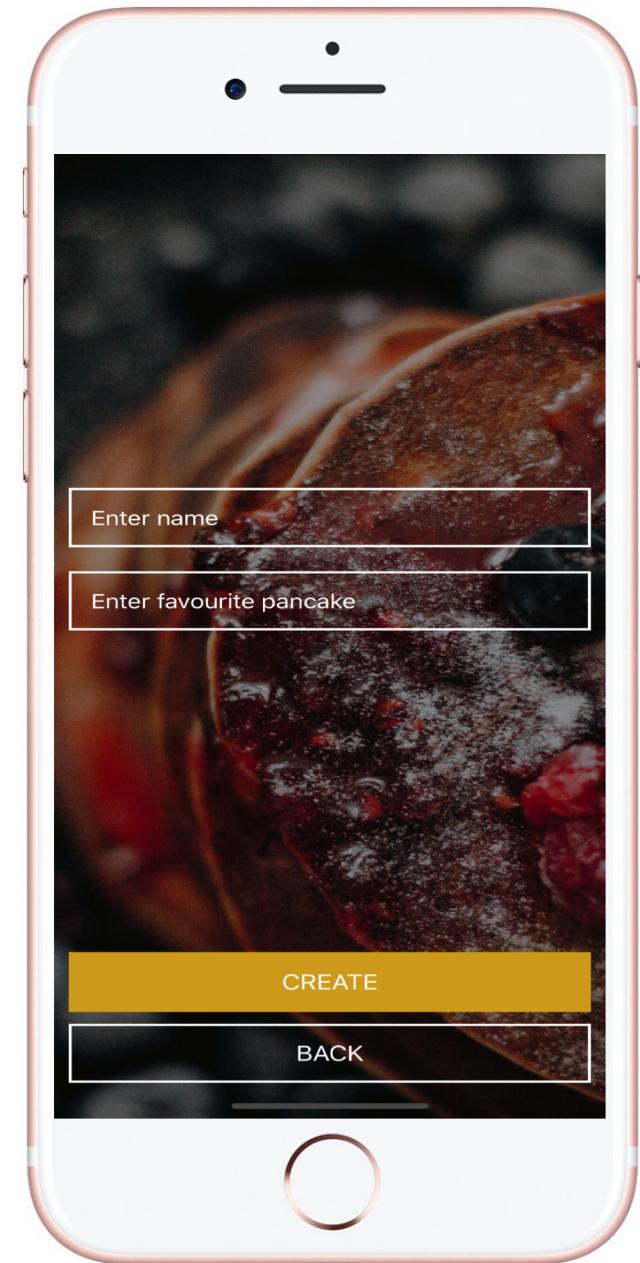


Your Turn

Add another button

Section 2: View Transition

- UITextField
- present() & dismiss() on button press
- protocol and delegate





Code Along

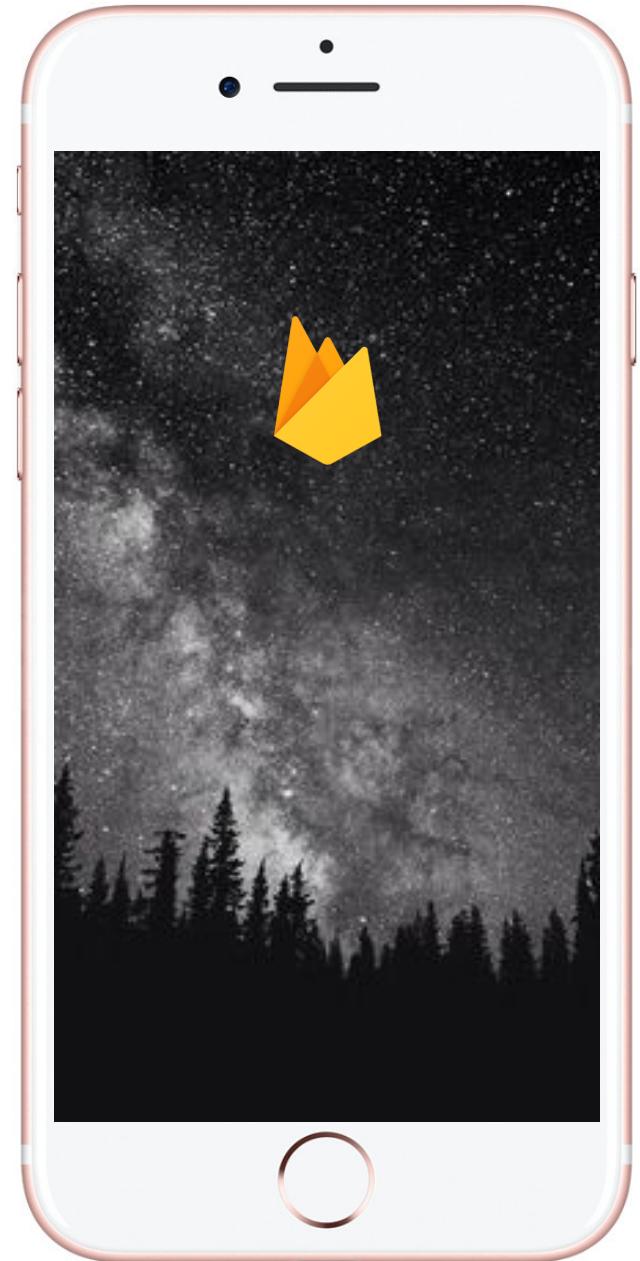


Your Turn

Add the 3rd View

Section 3A: Firebase Integration

- Introduction to Firebase
- Introduction to Cocoapods
- Adding Firebase to iOS project





What is **Firebase**?

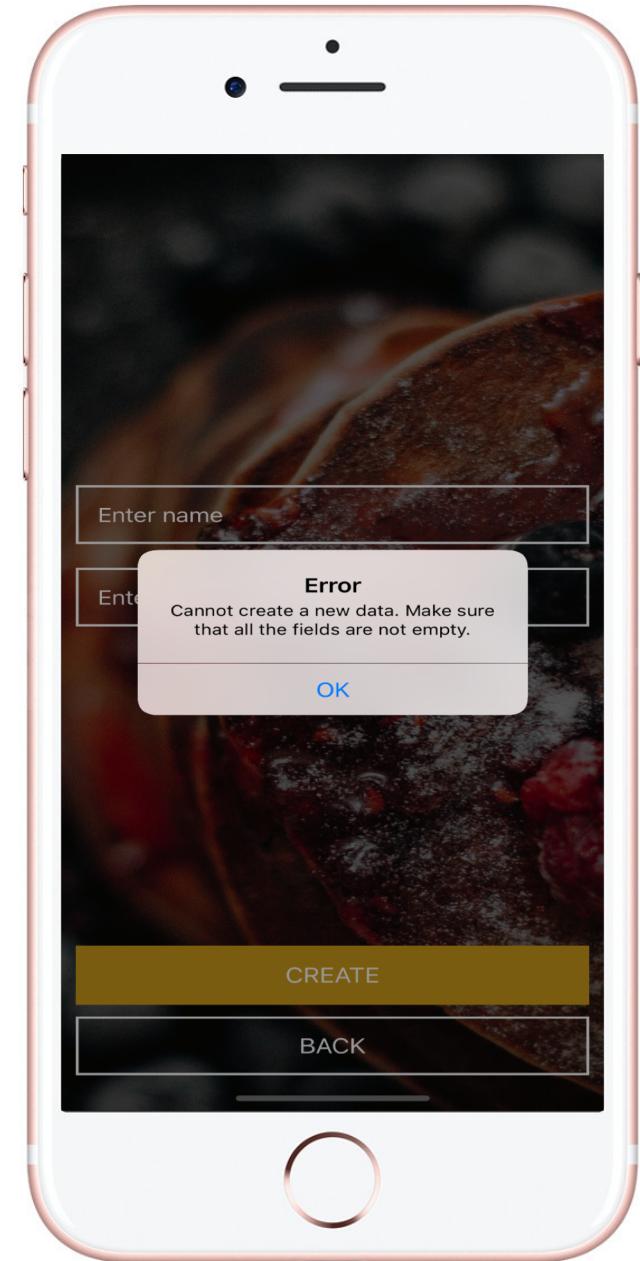


What is CocoaPods?



Section 3B: Storing Data to Firebase

- DatabaseReference, Key, Child, Value
- UIAlertController

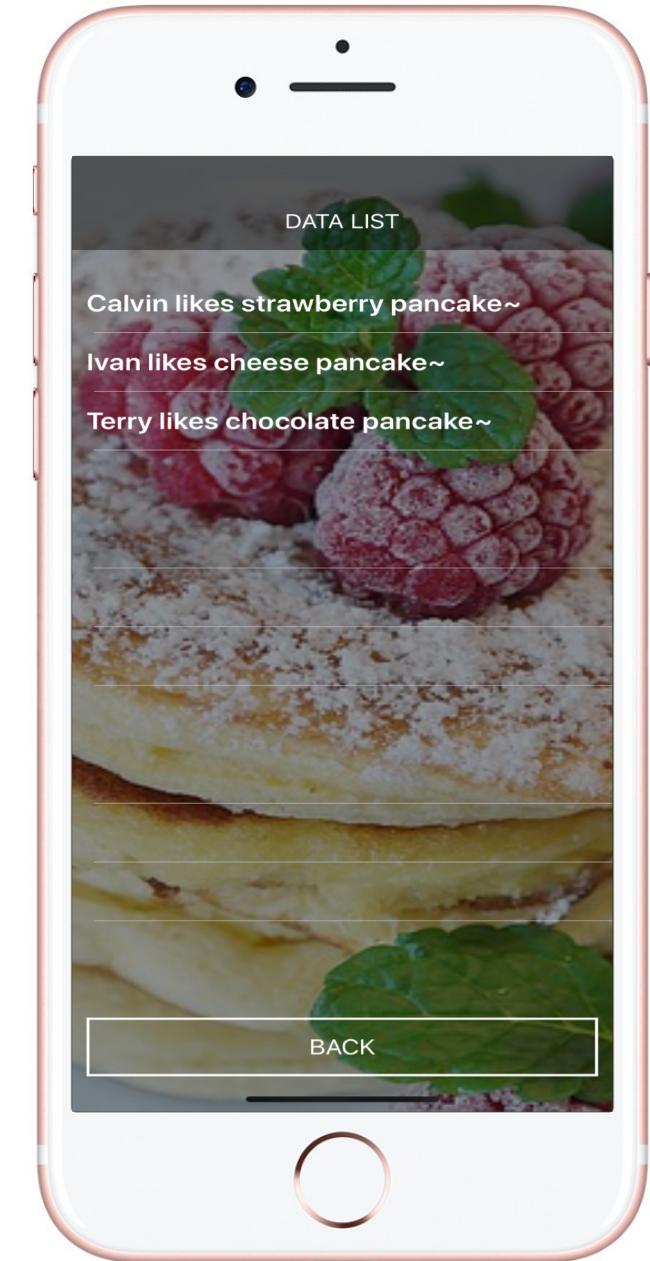




Code Along

Section 3C: Fetching Data from Firebase

- UITableView & UITableViewCell
- Delegate & Data Source
- Data Snapshot

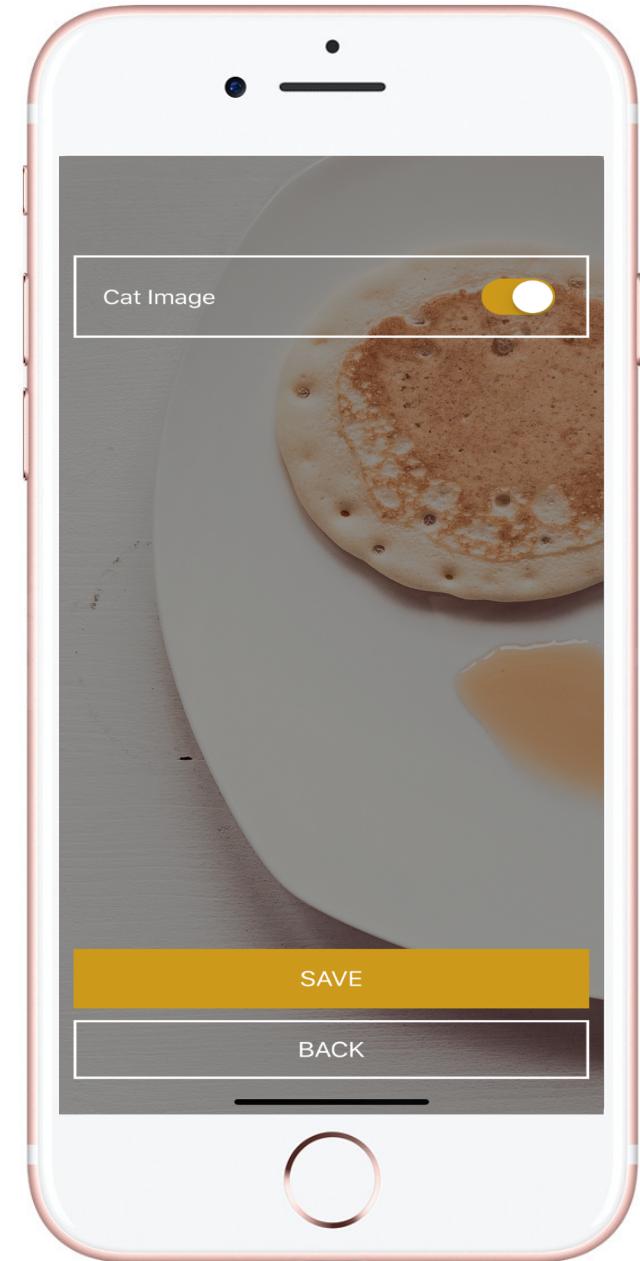




Code Along

Bonus: Storing Small Data Locally

- Storing small data (e.g. user settings) for as long as the application is installed
- Solution: UserDefaults



Things to Think About / Find Out

- So far we have seen View and Controller. Where is the Model?
- Can you see how MVC architectural pattern may potentially lead to Massive View Controller problem? If so, are there any better architectural patterns to implement?
- We have seen that adding UI components programmatically can produce a lot of boilerplate code, especially “constraints / anchor” code. How can we minimize this?
- Another way to store data locally is by using CoreData. What is the difference between UserDefaults and CoreData? When to use which?

Other Things to Explore

- Using Storyboard to build UI
- Firebase Authentication (Register + Sign In / Out)
- Storing pictures in Firebase
- UICollectionView
- UIView.animate
- UIGestureRecognizer
- Adding images / audio into iOS project

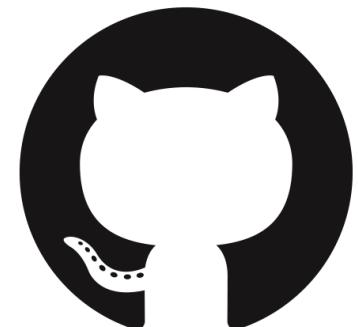


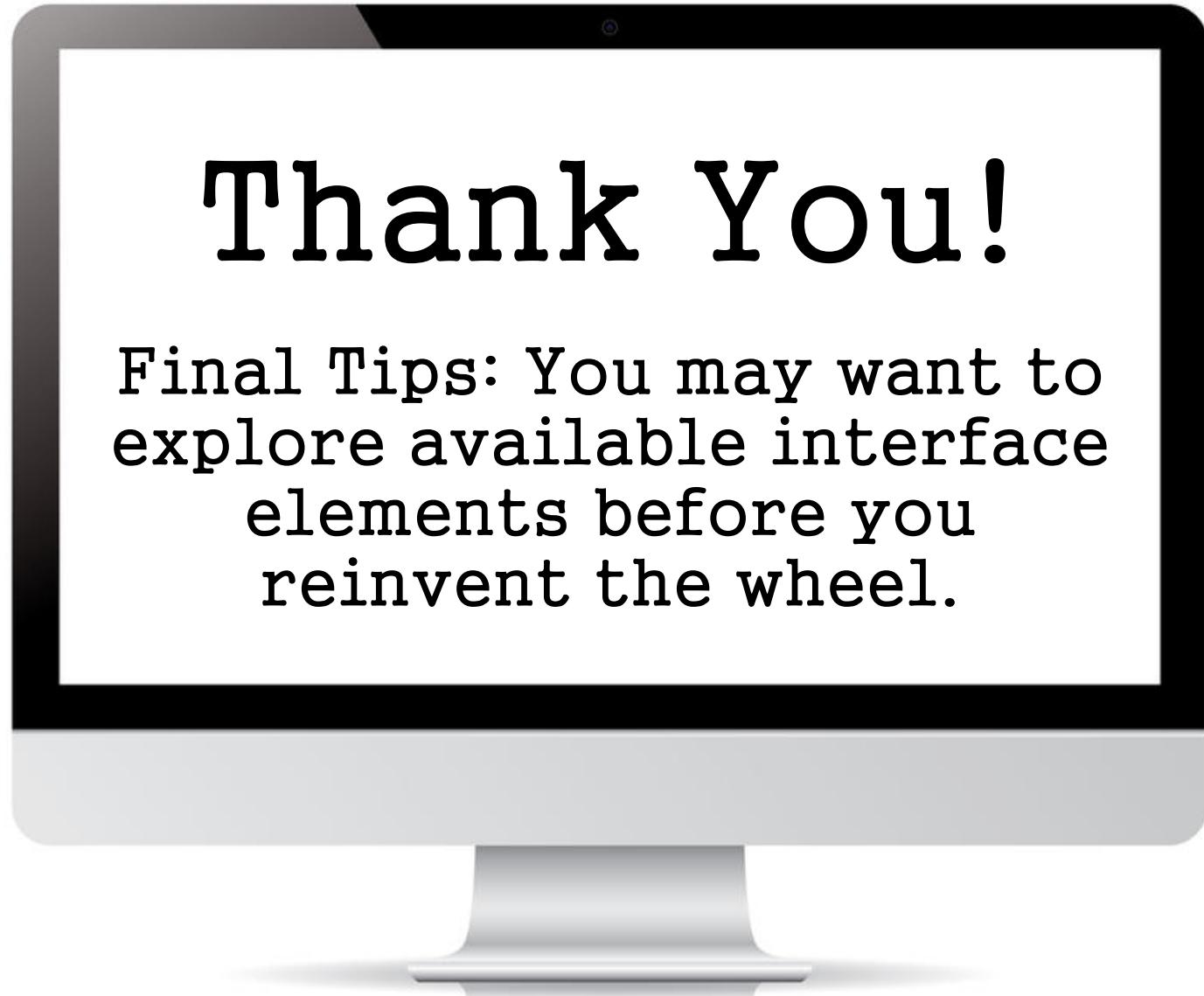
Checkout **Pancaker** Project on Github

Things not yet covered in the workshop:

- Launch Screen / Splash Screen
- UIImageView (background + cat images)
- UISwitch (user setting)
- Animations (button press animation + text field fly in / out animation)
- Model (with more than 1 attributes)
- UserDefaults implementation
- Global constants (e.g. colour palette)

<https://github.com/CT15/pancaker>





Switch

Slider

Segmented
Control

Stepper

Stack View

Date
Picker

Text View

Image
View