

IN4MATX 133: User Interface Software

Lecture 13:
Hybrid and Native Architectures

Professor Daniel A. Epstein
TA Lucas de Melo Silva
TA Jong Ho Lee

Announcements

- Midterm 2 (Tuesday) will be structured the same as Midterm 1
 - Mostly conceptual questions, with a little code interpretation
 - 45 minutes (will not take entire lecture time)

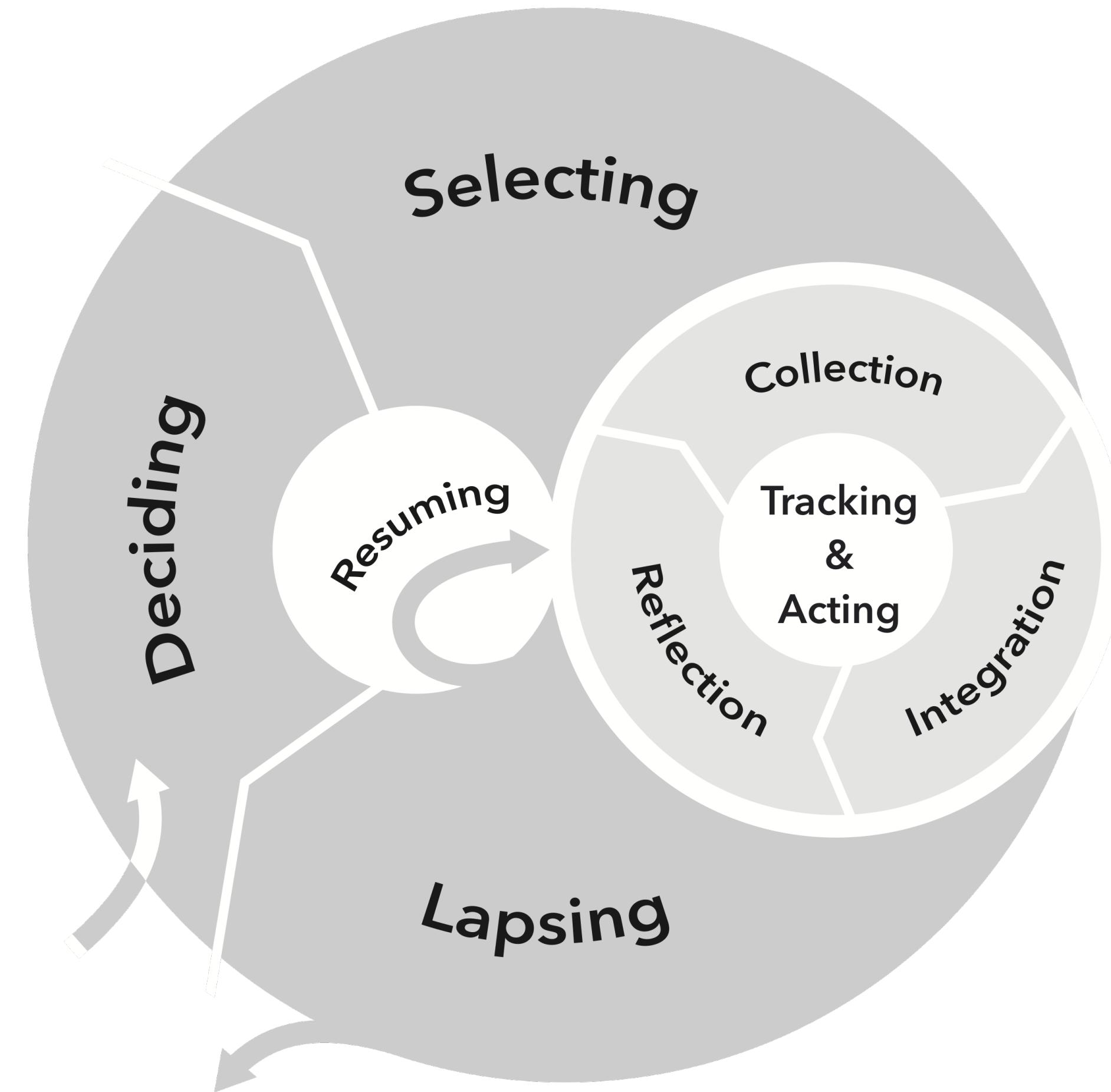
Where am I going?

Nov 3	Nov 4 Discussion 9:00-9:50 HH 254 10:00-10:50 HH 254 1:00-1:50 ALP 2500 2:00-2:50 ALP 2500	Nov 5 Mobile Design Principles & SASS 8:00-9:20 SSLH 100 Professor Epstein Office Hours 2:30-4:30 DBH 6093	Nov 6 Jongho Office Hours 1:00-3:00 ICS2 208	Nov 7 Hybrid & Native Architectures 8:00-9:20 SSLH 100 Professor Epstein Office Hours 2:30-4:30 DBH 6093	Nov 8 Lucas Office Hours 9:30-11:30 ICS2 208	Nov 9
Nov 10	Nov 11 Veteran's Day (Observed) Professor Epstein Away (CSCW)	Nov 12 Professor Epstein Away (CSCW) Midterm 2 Midterm in lecture 8:00-9:20 SSLH 100	Nov 13 Professor Epstein Away (CSCW) Jongho Office Hours 1:00-3:00 ICS2 208	Nov 14 Professor Epstein Away (CCC Workshop), Guest Tess Tanenbaum Mixed Reality Design and Play 8:00-9:20 SSLH 100	Nov 15 Professor Epstein Away (CCC Workshop) Lucas Office Hours 9:30-11:30 ICS2 208	Nov 16
Nov 17	Nov 18 A3 Due Spotify Browser in Angular Discussion 9:00-9:50 HH 254 10:00-10:50 HH 254 1:00-1:50 ALP 2500 2:00-2:50 ALP 2500	Nov 19 Ionic Components 8:00-9:20 SSLH 100 Professor Epstein Office Hours 2:30-4:30 DBH 6093	Nov 20 Jongho Office Hours 1:00-3:00 ICS2 208	Nov 21 Professor Epstein Away (NSF Panel), Guest Josh Garcia Smartphone Systems Security 8:00-9:20 SSLH 100	Nov 22 Professor Epstein Away (NSF Panel) Lucas Office Hours 9:30-11:30 ICS2 208	Nov 23
Nov 24	Nov 25 Discussion 9:00-9:50 HH 254 10:00-10:50 HH 254 1:00-1:50 ALP 2500 2:00-2:50 ALP 2500	Nov 26 Device Resources & Sensors 8:00-9:20 SSLH 100 Professor Epstein Office Hours 2:30-4:30 DBH 6093	Nov 27 Jongho Office Hours 1:00-3:00 ICS2 208	Nov 28 Thanksgiving	Nov 29 Thanksgiving (Observed)	Nov 30

**Being at a top research university
means you get to see the latest research,
and potentially even work on it**

My research broadly

- How we can design tracking technology (for physical activity, finances, mental health, etc.) to better fit into the realities of people's everyday lives
 - Acknowledge that people don't want to track forever



Where am I going?

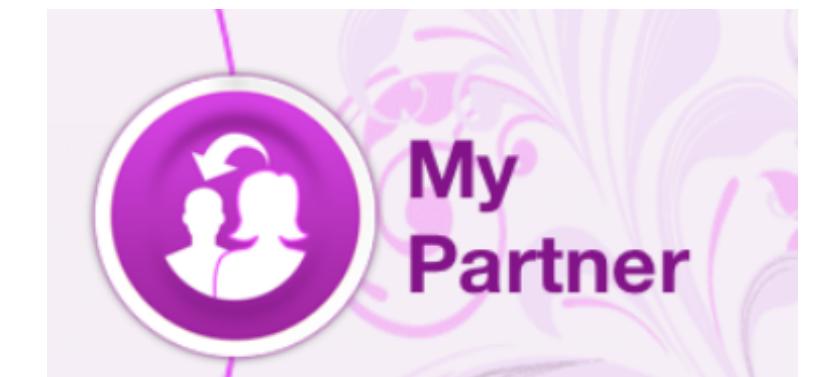
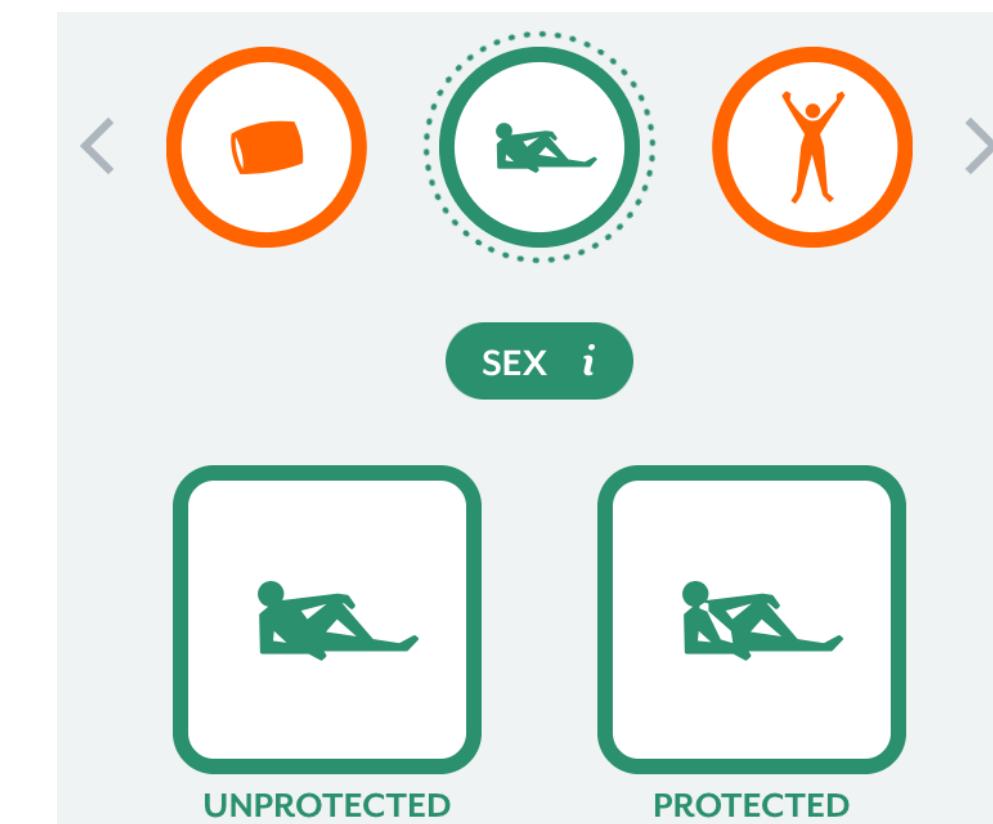
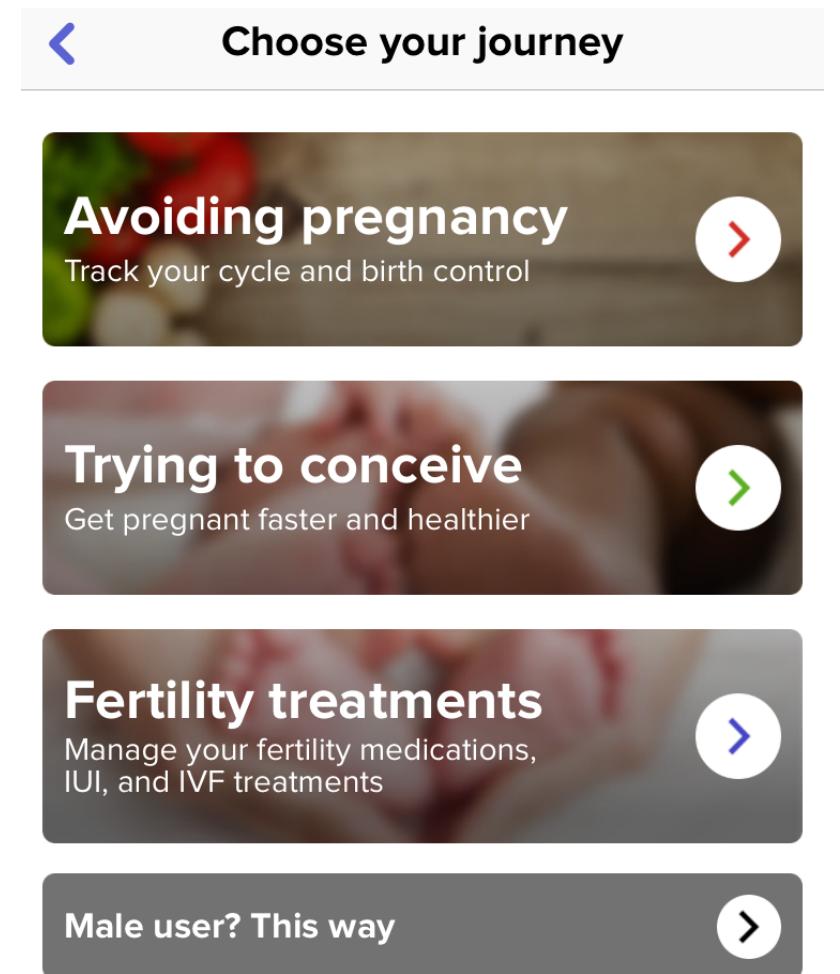
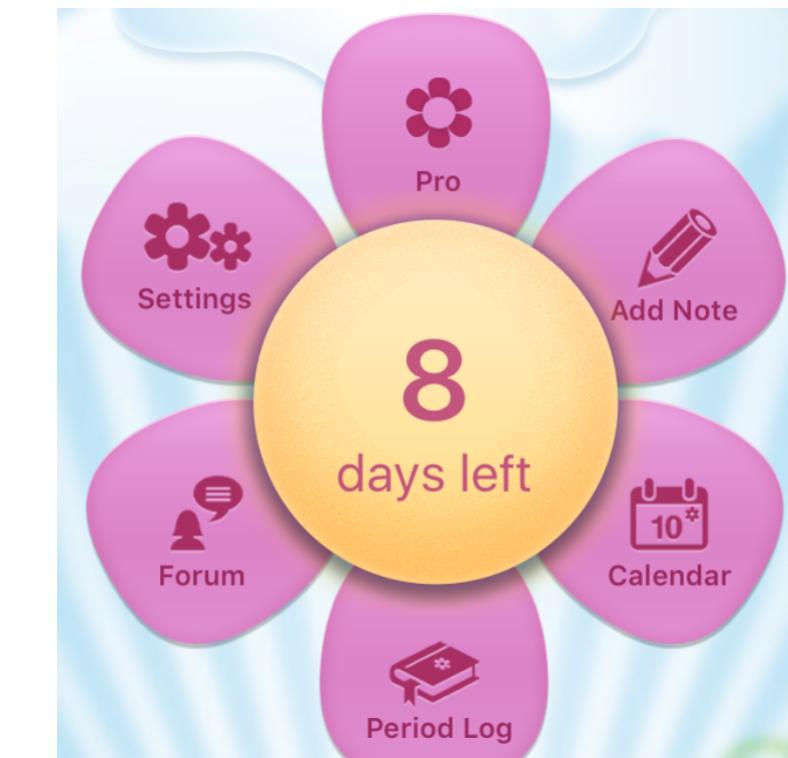


- “*The 22nd ACM Conference on Computer-Supported Cooperative Work and Social Computing*” in Austin, Texas
- Computer Supported Cooperative Work
 - Understanding and improving how people work together
 - Office settings, remote work, gig work, hospital work...
- Social Computing
 - Understanding how people use social media
 - Facebook, Twitter, Instagram, Snapchat, Twitch, Tinder, Grindr...

<http://cscw.acm.org/2019/>

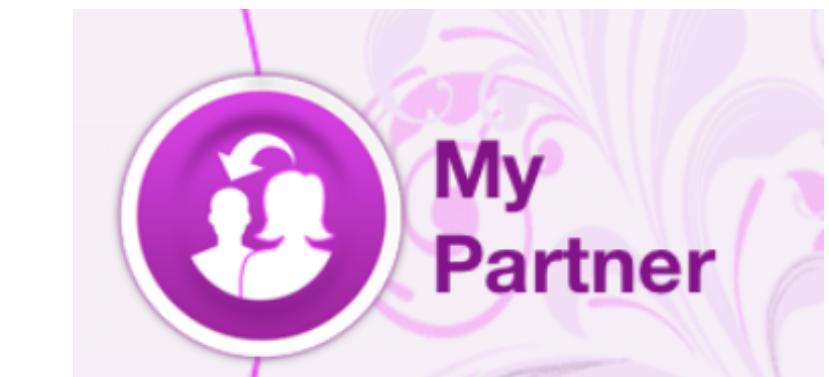
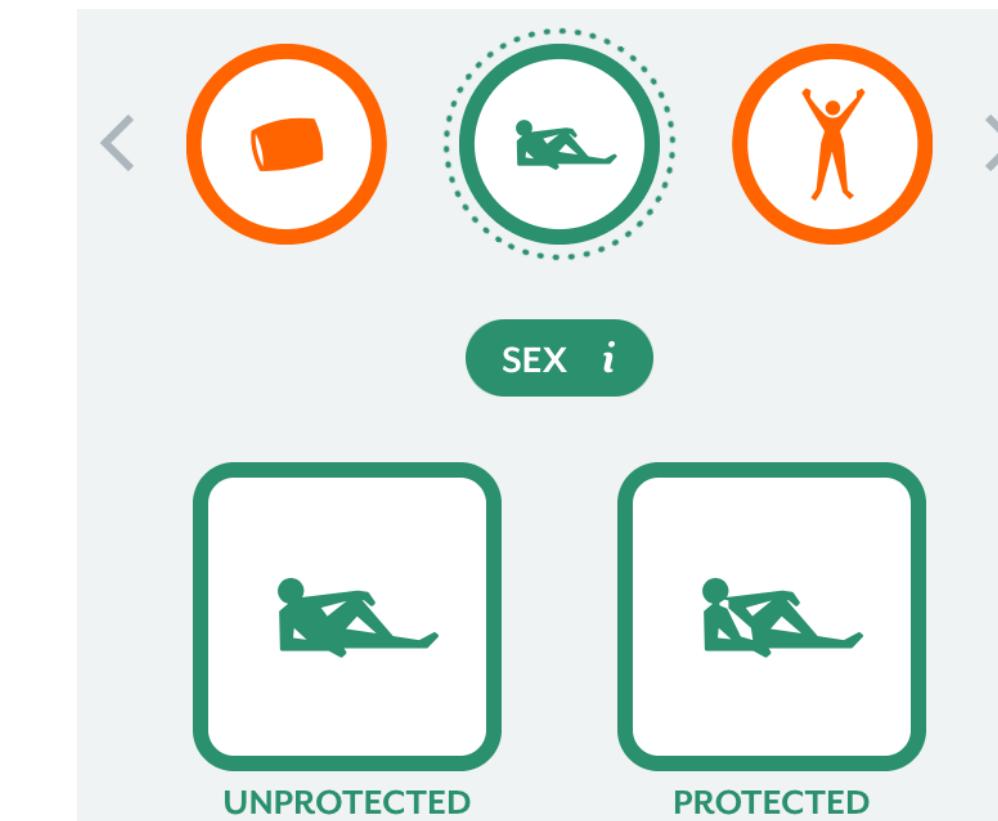
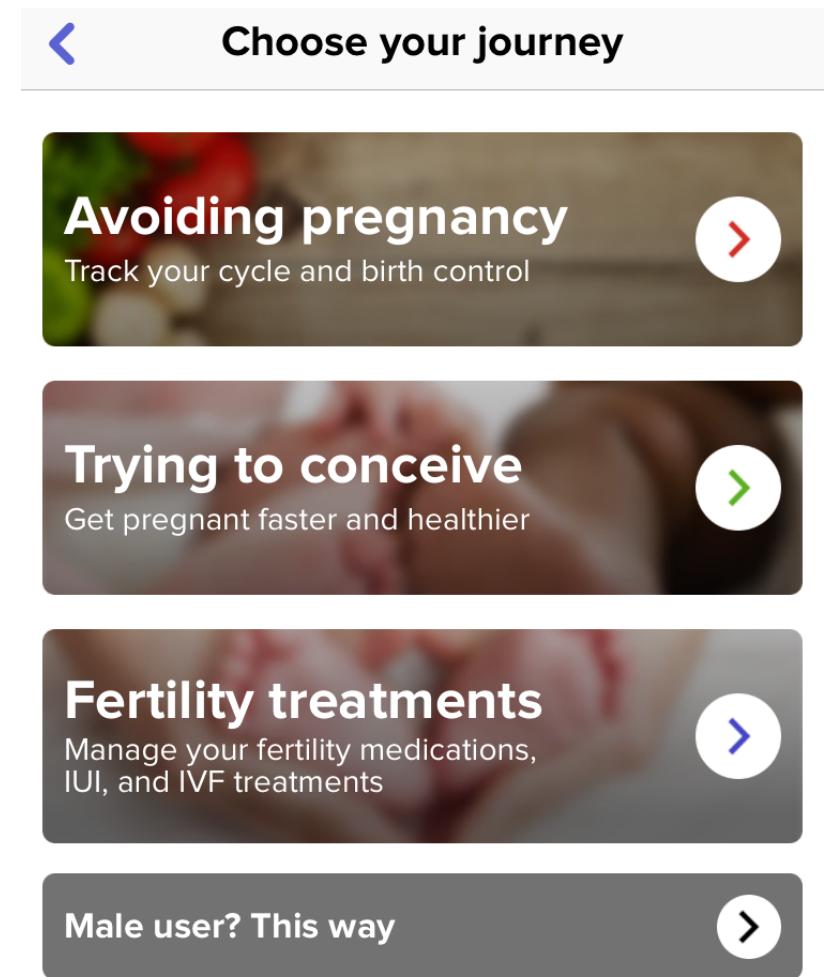
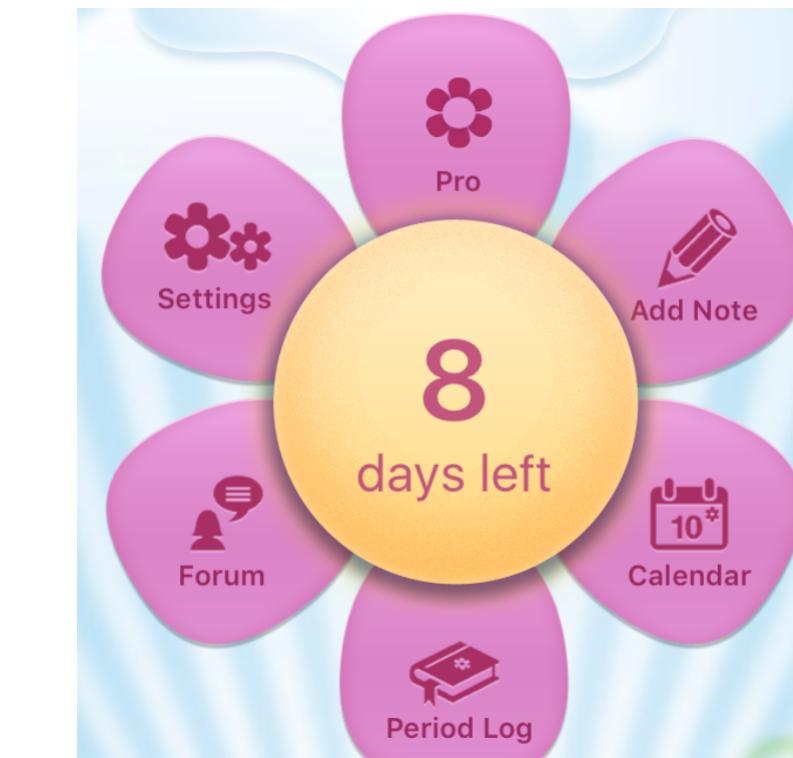
Women's Health, Wellbeing, and Empowerment

- Women's health technology assumes people's goals, identities, and preferences
 - “I used to be embarrassed when other people looked at my phone and saw a bright pink tracking app.”
 - “Had to find a period tracker that didn't misgender me. Only found two”



Women's Health, Wellbeing, and Empowerment

- These assumptions can make people feel unsupported, ignored, and excluded
 - “Sex options assume sex with a man... [it] reminds me I am not a ‘normal’ woman whenever I use the app.”
 - “[My app is] clearly trying to support my getting pregnant (which is not my intent) and not just agnostically for tracking”





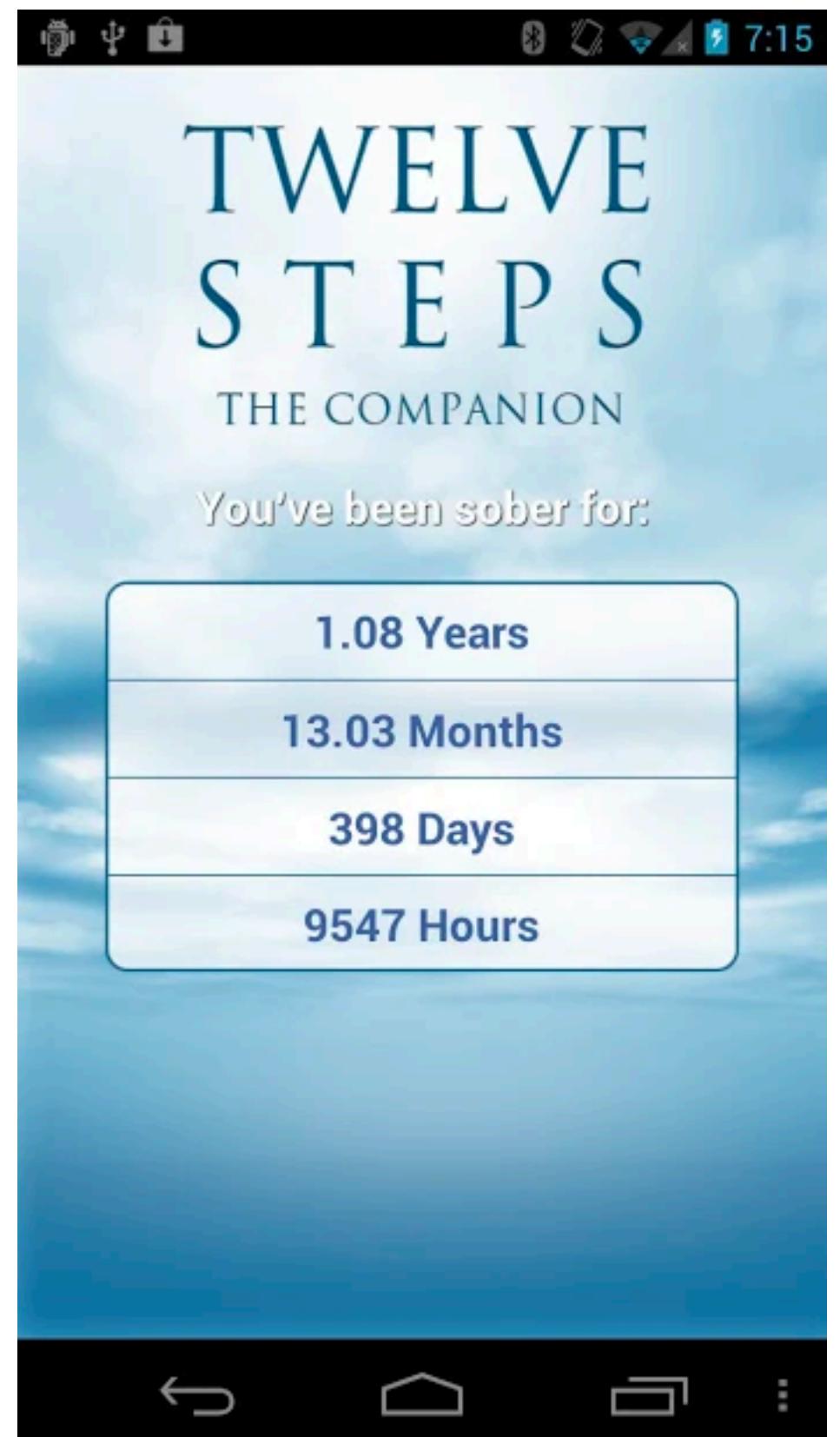
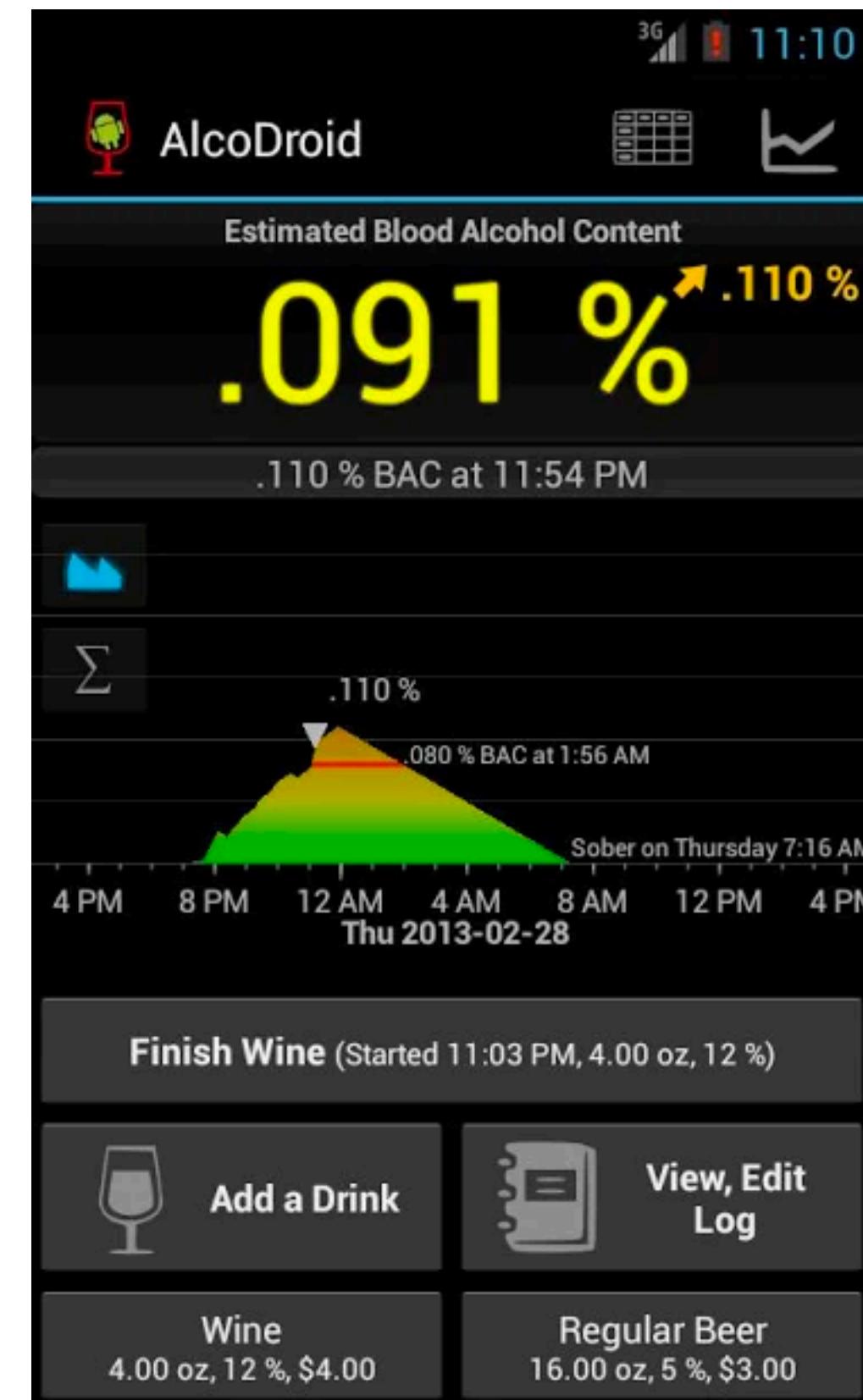
Where am I going?

- “*CCC workshop on Computational Support for Substance Use Disorder, Prevention, Detection, Treatment, and Recovery*” in Washington, DC
- CCC (Computing Community Consortium): a non-profit organization which aims to build community around important research topics
- They host frequent workshops on issues of national importance where computing plays a major role

<https://cra.org/ccc/events/mental-health-and-addiction/>

Substance Abuse

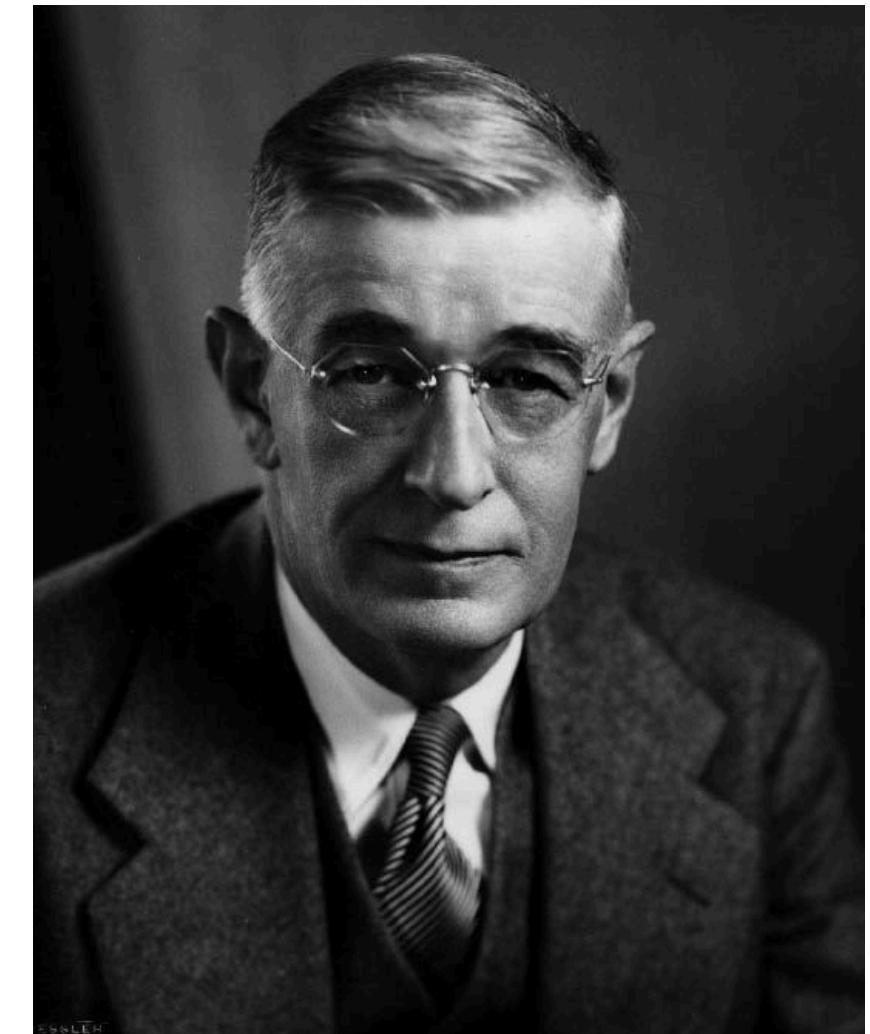
- Technology for monitoring substance use and managing recovery is becoming increasingly prevalent
 - Lapsing and abandonment are charged words in these domains, and need to be handled more carefully
 - How should technology account for abandonment of the technology versus abandonment of recovery?





Where am I going?

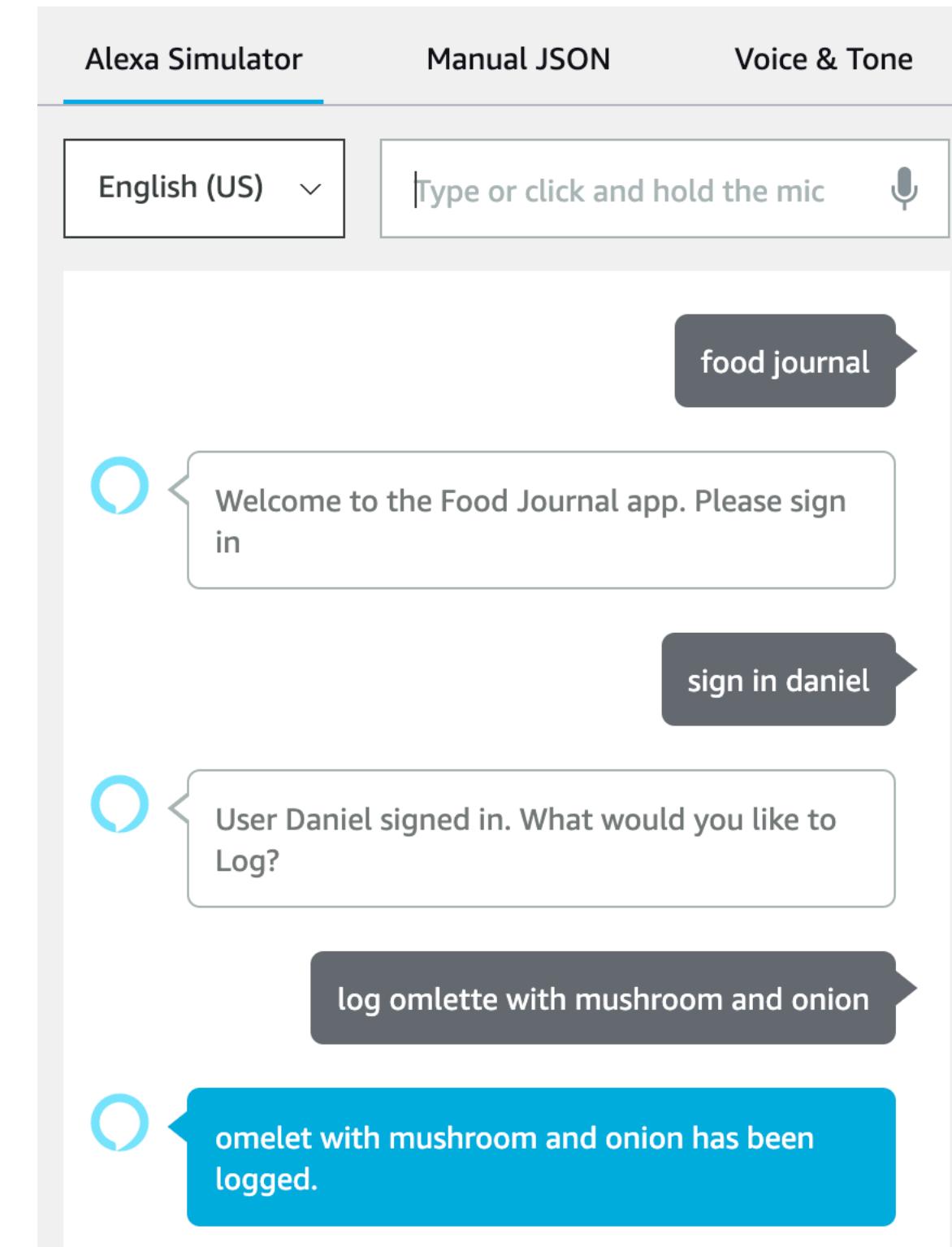
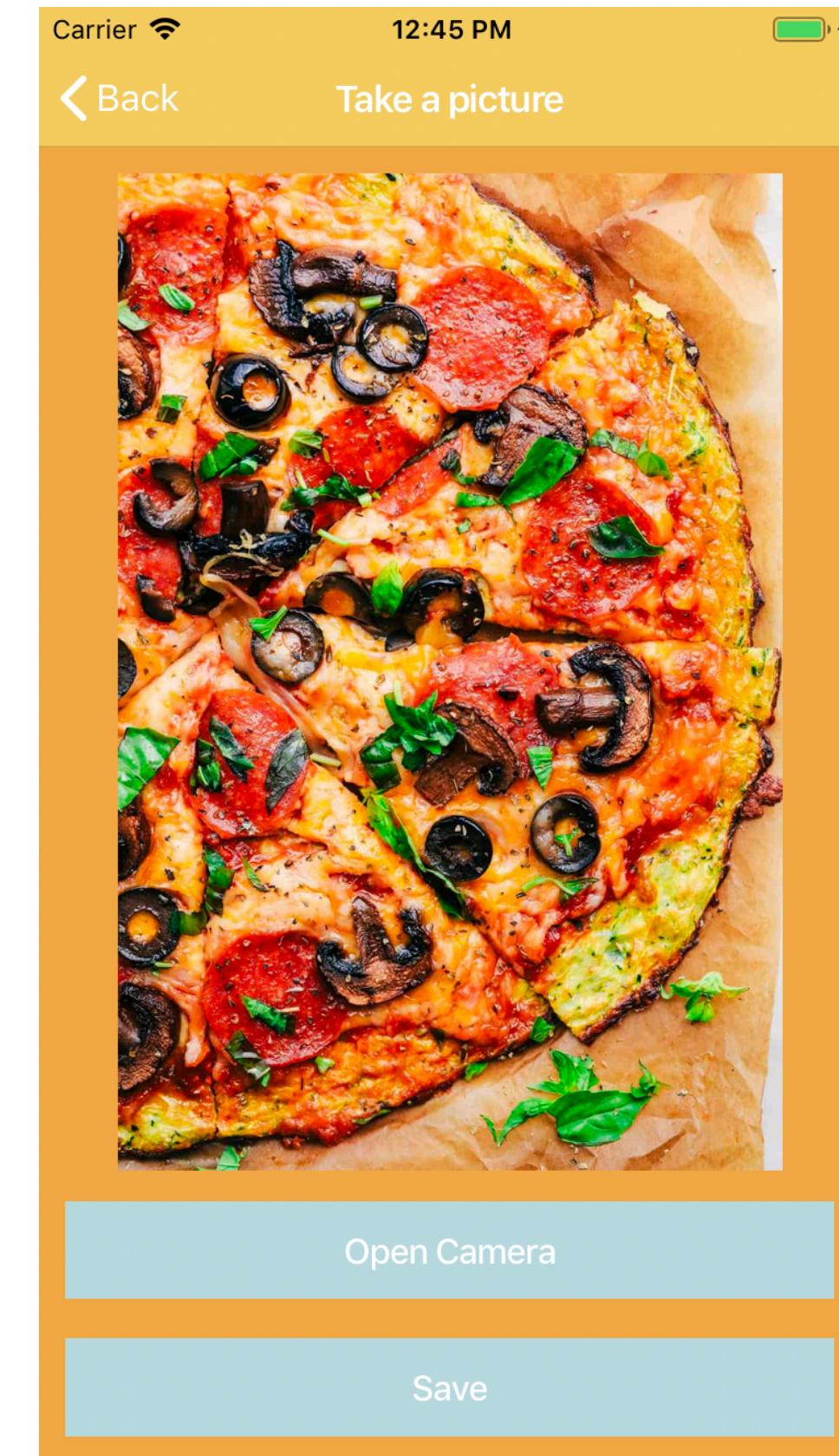
- National Science Foundation (NSF) review panel for Computing Research and Initiation Initiative (CRII) grants
- CRII: a research grant that recently-hired computing faculty can apply for
- I received one of these grants last year (\$175k), so I've been asked to “pay it forward” for the next batch of faculty
 - Reviewers are crucial for *anyone* to get funding from the government
 - Remember Vannevar Bush (memex)?



https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=504952

Multimodal Food Journaling

- Keeping track of what you eat is useful for diet and weight loss, but is challenging
 - What if technology supported food entry on a wide range of different modalities and devices?
 - Smartwatches, digital assistants, searches in food databases, photo uploads, recipes...
 - The goal is to be as complete as possible, not to automate capture



**If any of these topics interest you,
I'm always looking to involve
more students in research**

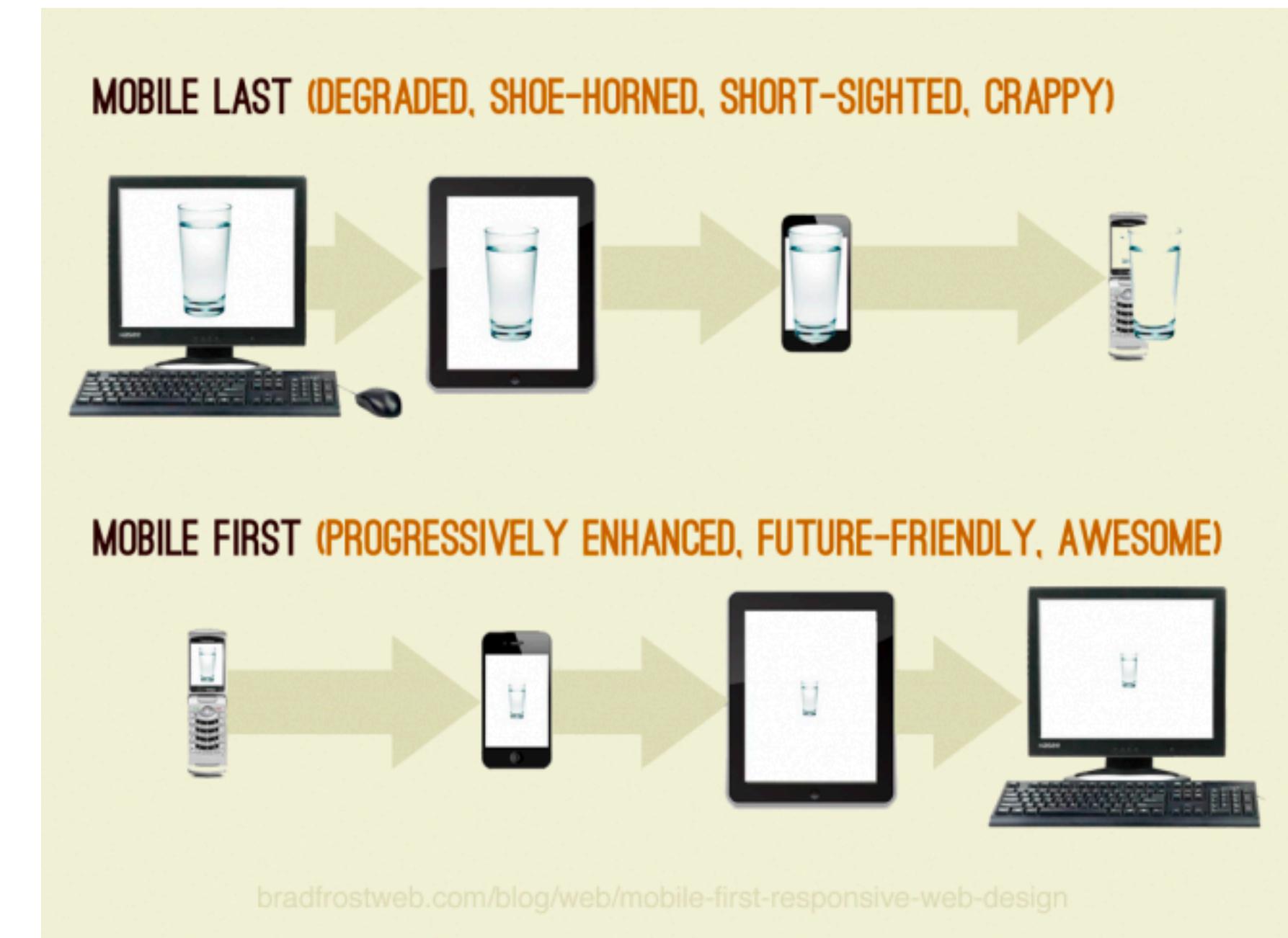
Today's goals

By the end of today, you should be able to...

- Differentiate approaches to developing mobile interfaces
- Describe advantages and disadvantages of developing native, hybrid, and web applications
- Explain which approach Ionic takes to app development

Mobile-first design

- Plan your design for mobile
- Then make your app *better* with more real estate
 - Add more features
 - Make existing features easier to navigate
- A lot of businesses make mobile-friendly websites before making dedicated apps



**Question: why might a business
want a mobile app
over a mobile website?**

**There are a variety of ways
to build mobile apps**

Mobile development methods

- Native
- WebView
- Hybrid
- Responsive

Native apps

- An app designed to work on a specific piece of hardware
- Usually built with tools created by the hardware or platform manufacturer
 - Android Studio for Android, in Java
 - Xcode for iOS, in Swift or Objective-C

Native apps

- As we think of them today, native apps started with the first iPhone
- Released a development platform alongside the hardware



Native apps

- iOS development languages:
 - Objective-C
 - Cocoa Touch
 - Swift
- These languages were either developed by or pretty much only used by Apple
 - Developer lock-in is a... Disadvantage? Advantage? Both?



Native apps

- iOS development tools:
 - Xcode
 - iOS Source Development Kit (SDK)
 - SDK provides access to phone's storage, camera, sensors, etc.



Native apps

- Android development languages:
 - Primarily Java
 - C and C++ via Android Native Development Kit (NDK)
- Align more closely with languages used in other contexts
 - Is this an advantage? A disadvantage?



Native apps

- Android development tools:
 - Android Studio
 - Android Source Development Kit (SDK)
 - Various IDEs like Eclipse or NetBeans



Native apps

- Platform-specific codebases
 - Android is in Java,
iOS is in Objective-C or Swift
 - Both use different libraries
to communicate with the hardware
- Usually require starting to code
from scratch



**What if we already made a website
for our app? Or have some other
existing codebase?**

What if we want to share code across phone platforms?

Solution: hybrid apps

Hybrid apps

- “Use a common code base to deploy native-like apps on a wide range of platforms”
- Two primary approaches:
 - WebView app
 - Compiled hybrid app

WebView app

- Run a webpage written in HTML/CSS/JavaScript, on the phone's internal browser
- Load that browser in a lightweight native app
- Ideally, expose some native APIs to the browser

WebView app

- Essentially, the app is just a website
- Allows the same or similar code to be used across an app and a website

WebView app frameworks

- Ionic
- jQuery mobile
- NativeScript
- These frameworks use web technologies (HTML, CSS, TypeScript, JavaScript) rather than platform-specific technologies



WebView app frameworks

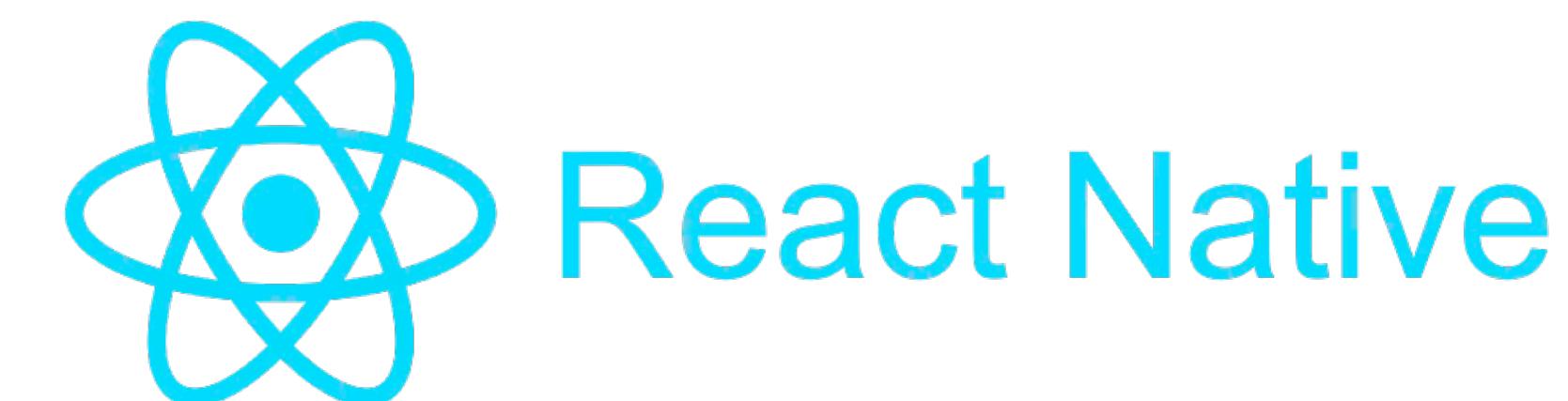
- WebView apps are just websites
- What do these frameworks provide?
 - Common mobile interface elements like sliders and buttons (more on that next week)
 - The native app for running the website
 - Some APIs for communicating with platform SDKs

Compiled hybrid apps

- “Write code in one language, such as C# or JavaScript, and compile it to native code supported by each platform”
- Result: a native app for each platform
- Challenge: less freedom in development

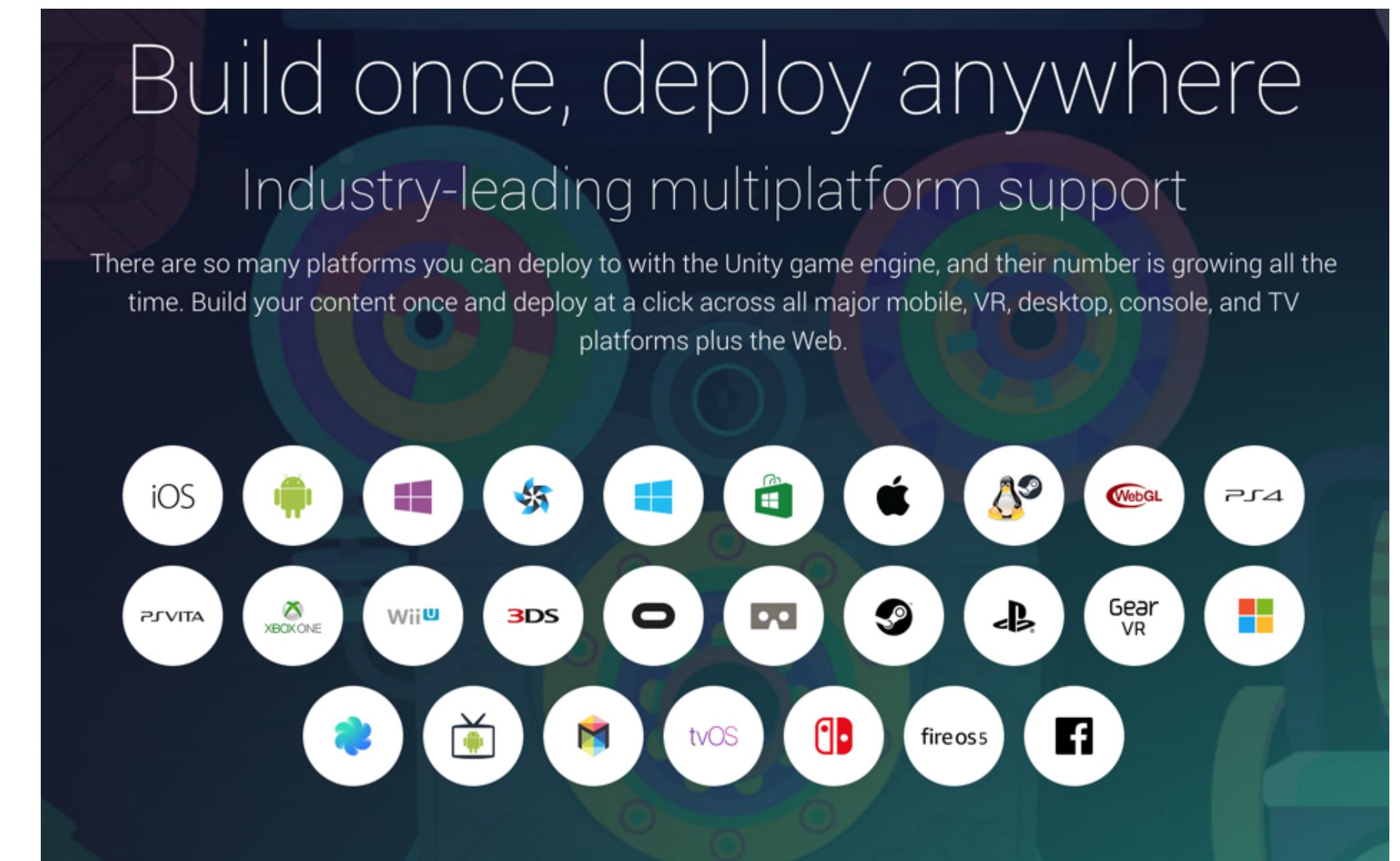
Compiled hybrid app frameworks

- Xamarin
 - C#
- Unity
 - C# or JavaScript
- React Native
 - JavaScript



Unity

- Leading game development platform
 - Supports consoles, web, and mobile
 - Will need to import or use platform-specific SDKs



React Native

- Uses React, a web framework similar to Angular
- Compiles a webpage to a native app

Who's using React Native?

Thousands of apps are using React Native, from established Fortune 500 companies to hot new startups. If you're curious to see what can be accomplished with React Native, check out these apps!



Question



**Which app will have the best and worst performance:
a Native, WebView, and Compiled hybrid app?
Performance broadly defined; page loading times, click latency, etc.**

- A Compiled hybrid will perform best, WebView will perform worst
- B WebView will perform best, Compiled hybrid will perform worst
- C WebView will perform best, Native will perform worst
- D Native will perform best, Compiled hybrid will perform worst
- E Native will perform best, WebView will perform worst

Question



**Which app will have the best and worst performance:
a Native, WebView, and Compiled hybrid app?
Performance broadly defined; page loading times, click latency, etc.**

- A Compiled hybrid will perform best, WebView will perform worst
- B WebView will perform best, Compiled hybrid will perform worst
- C WebView will perform best, Native will perform worst
- D Native will perform best, Compiled hybrid will perform worst
- E Native will perform best, WebView will perform worst

Performance is just one factor.

**How do we choose
a development approach?**

Business considerations

- Development time
- Development cost
- Maintenance concerns
- Available infrastructure

UX and design considerations

- Consistency with platform
- Device capabilities
- Interaction models supported
- Performance and usability

Technical considerations

- Programming languages
- Integration with device
- Performance
- Upkeep and maintenance
- Flexibility
- Compatibility

Pros and cons of each option

Strengths of hybrid apps

- Can share a codebase between web and mobile
- Can save time and effort (sometimes)
- Easily design for various form factors
- Access to some device capabilities

Weaknesses of hybrid apps

- Performance issues
- Inconsistency with platform
- Limited access to device capabilities

Strengths of native apps

- Consistent experience with platform
- Leverages full device capabilities
- Uses native UI elements

Weaknesses of native apps

- Need to support separate development for each platform
- Cost of app development and maintenance
- Need to learn/manage multiple programming languages
- Need to manage multiple sets of tools

Hybrid apps vs. native apps

- Hybrid apps are great when time or money is a concern and you need to deploy on multiple platforms
- Native apps are great when performance and consistency with the platform are major concerns

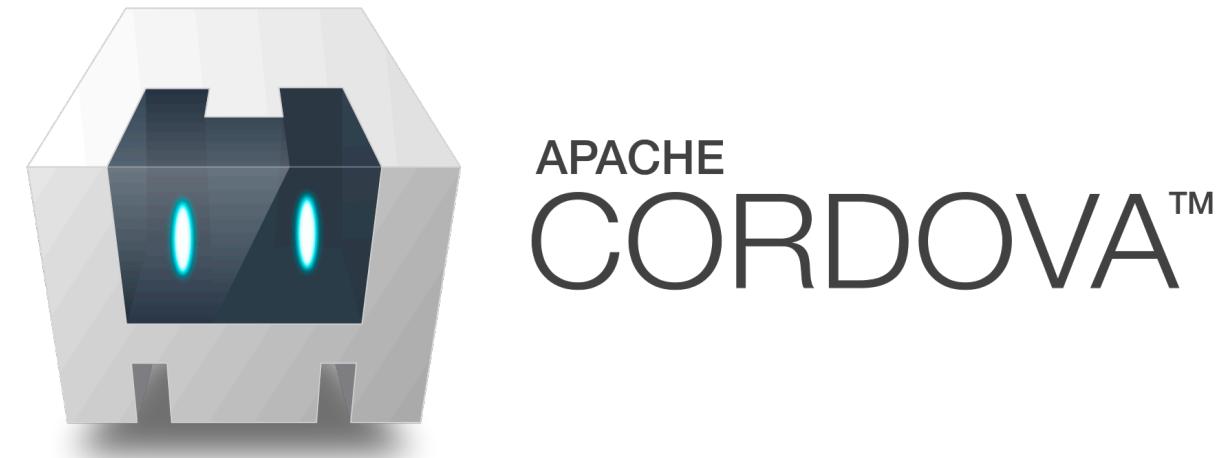
Hybrid apps vs. native apps

- Hybrid apps
 - News sites
 - Informational apps
 - Product showcase
 - Seasonal/one-off
- Native apps
 - Games
 - Content-heavy apps
 - Uses a lot of device resources
 - Needs specific OS capabilities

One Hybrid (WebView) framework: Ionic

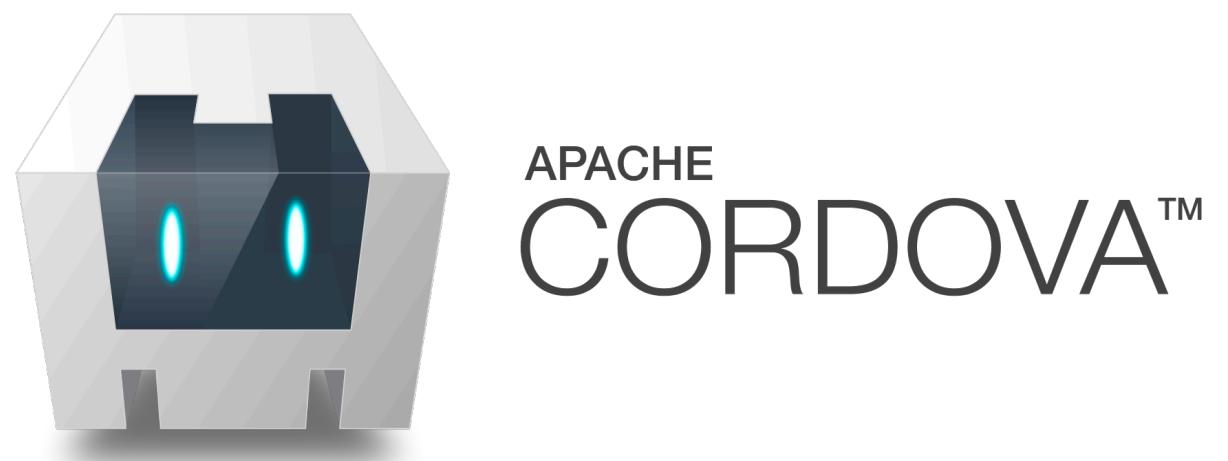
Ionic

- WebView app framework
- Launched in 2013
- Interface implemented in Angular
 - Recently added support for React
- Apache Cordova provides the native app which opens the WebView



Ionic Native

- Apache Cordova also provides libraries for connecting to device resources
- Ionic Native brings these libraries to Ionic as plugins
 - Ionic Native plugins are imported as services
 - Hundreds of plugins



<https://ionicframework.com/docs/native/>

Ionic Native

Some example plugins

- Geolocation
- Bluetooth
- Camera
- Health
- Gyroscope
- Pedometer

<https://ionicframework.com/docs/native/>

Ionic Native

Some example plugins

- Facebook
- LinkedIn
- WeChat
- Apple Pay
- Google Maps
- Youtube

<https://ionicframework.com/docs/native/>

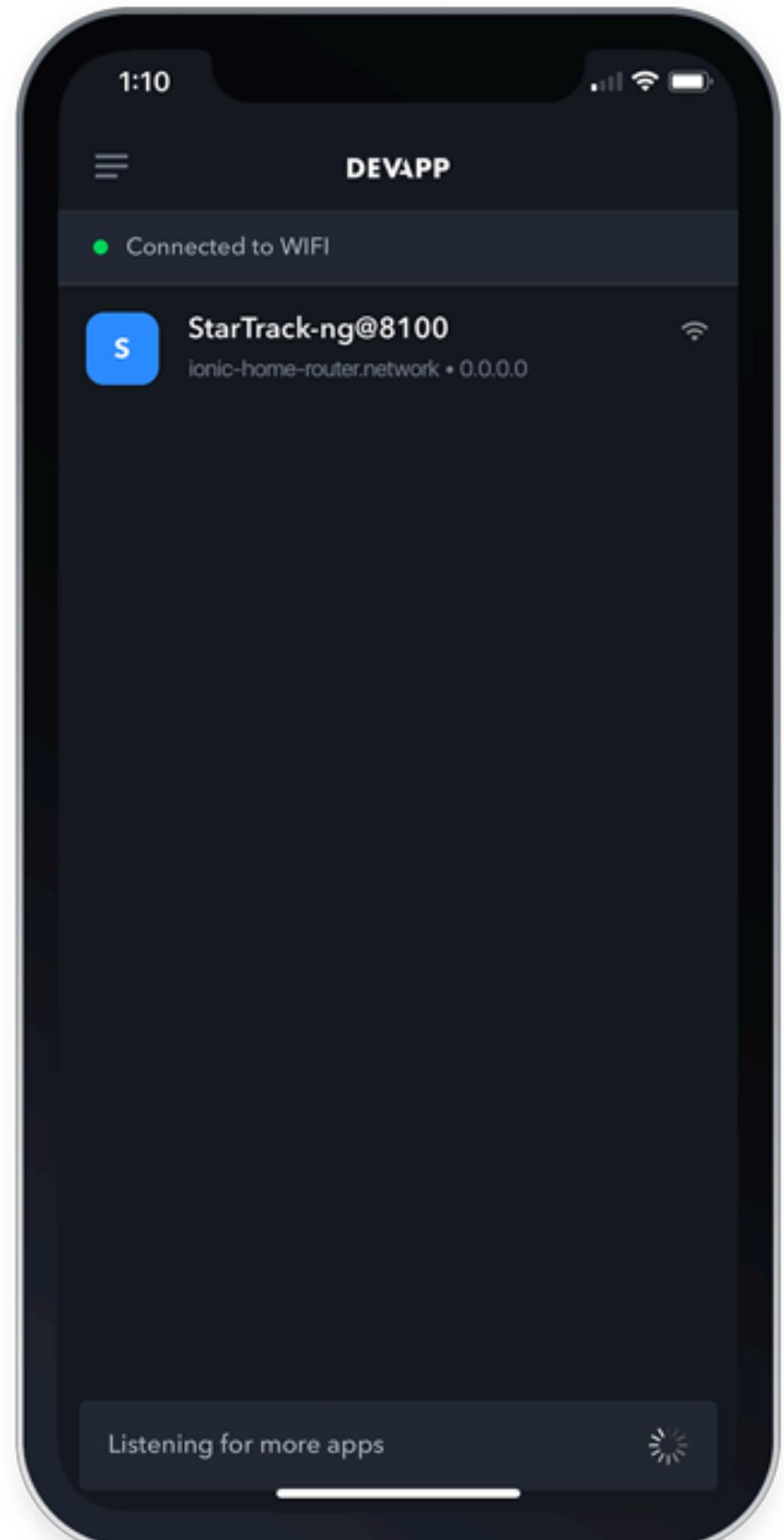
Deploying Ionic apps

- Involves packaging up an app and “signing” it as a developer
 - For Android, this requires installing Android Studio
 - For iOS, this requires installing Xcode and getting a developer account
- Can then “deploy” the app to the app store
 - The iOS app store includes features for “beta” deployment with a small group of developers
- This process is often a pain

<https://ionicframework.com/docs/building/ios> or <https://ionicframework.com/docs/building/android>

Ionic DevApp

- An app on Android and iOS made by Ionic
- Provides a WebView to open up Ionic apps
 - Lets you test your Ionic app on an actual device
 - Has some Ionic Native plugins, but not all

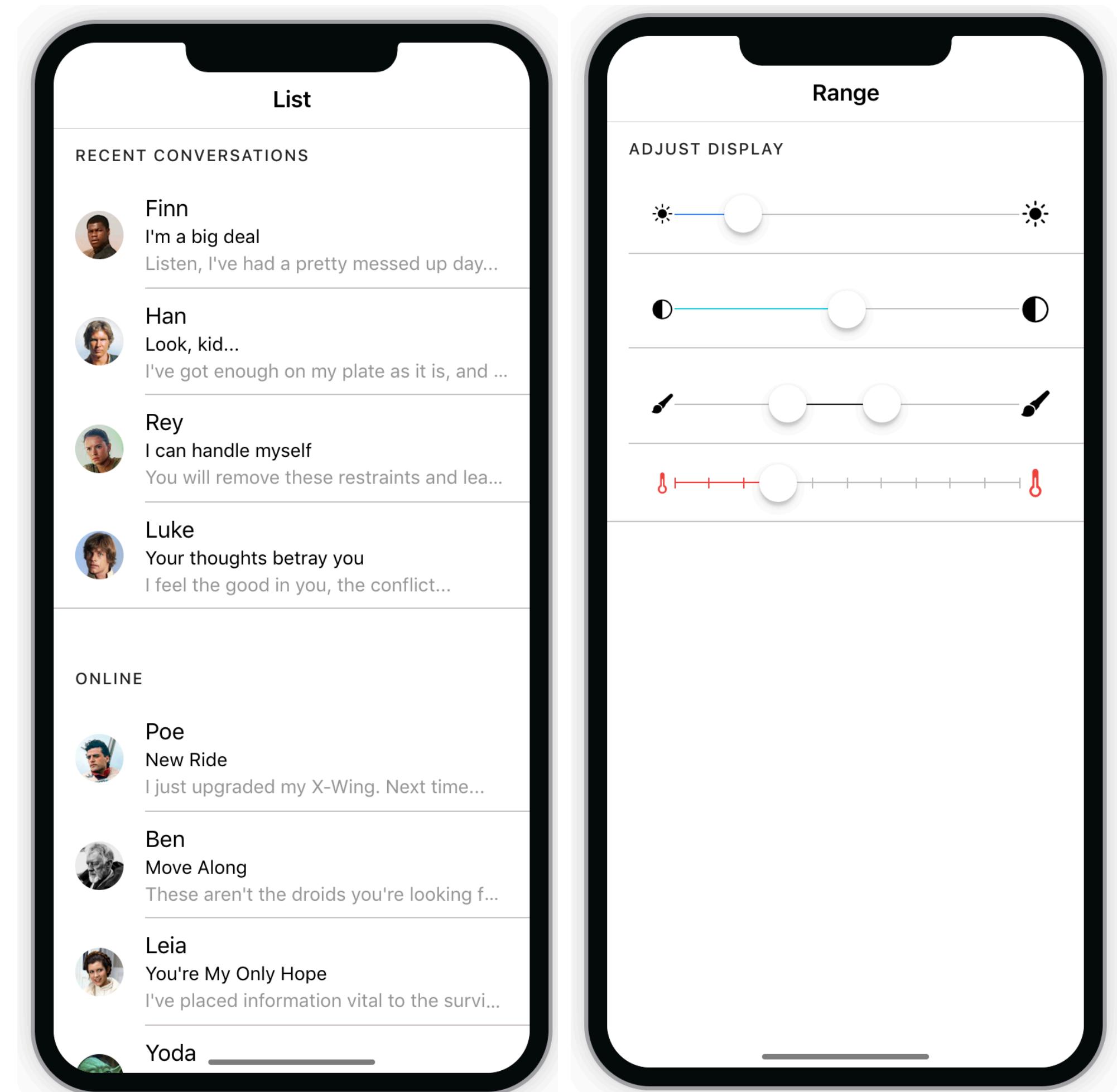


<https://ionicframework.com/docs/appflow/devapp/>

What does Ionic add over Angular?

Ionic components

- Ionic provides Angular-style components for a lot of interface elements common in mobile interfaces
 - Lists, buttons, sliders, tabs, modal dialogs, search bars, much more
 - These are the focus of next lecture



<https://ionicframework.com/docs/components/>

Today's goals

By the end of today, you should be able to...

- Differentiate approaches to developing mobile interfaces
- Describe advantages and disadvantages of developing native, hybrid, and web applications
- Explain which approach Ionic takes to app development

IN4MATX 133: User Interface Software

Lecture 13:
Hybrid and Native Architectures

Professor Daniel A. Epstein
TA Lucas de Melo Silva
TA Jong Ho Lee