Today: Course Project Details

• Warm-up: couple of mini-projects to learn basics of mobile development
• ½-semester (group) project: project of your choosing
• If help needed identifying team members, let me know!
• Teams are expected to “produce more” than individuals!
• Project can align with your doctoral, MS, honor’s thesis, or other projects (e.g., participation in coding competition), but this must be disclosed in proposal and approved by instructor!
• Collaboration among students/teams encouraged, but extend of collaboration must be disclosed in reports!
“Platforms”

- Which platform to develop on?
- Answer: all of them!

Native vs. Web vs. Hybrid
Native vs. Web vs. Hybrid

Mobile App Technology Stacks

Your App
- Native UI Views
- Native Hardware Access

Native Wrapper App
- Embedded Web Views
- Native U/I Views

Web Browser
- Operating System & Device

Your App
- Operating System & Device

"Hybrid" App
- Operating System & Device

Web App
Native vs. Web vs. Hybrid

Native
- Advanced UI interactions
- Focused performance
- App store distribution

Hybrid
- Web developer skills
- Access to native platform
- App store distribution

Native vs. Web vs. Hybrid

Native
- More device-specific features
- No code portability
- Advanced GUI experience
- High development cost
- Better performance
- Access to device hardware

Hybrid
- Less device-specific features
- High code portability
- Moderate GUI experience
- Moderate development cost
- Average performance
- Moderate access to device hardware
Options

React Native VS ionic VS Flutter VS Adobe PhoneGap

Options

[Bar chart showing popularity of different options]

© 2020
React Native

- Developed by Facebook, 2015
- Open-source JavaScript Programming Language
- Low Development Time
- Support for Third-Party Libraries
- Mobile Environments Performance
- NPM for Installation
- Responsive UI/UX
- Single Code Base
- React Native is UI-Focused
- Incorporate the functionality of other native apps

Ionic

- Open-source SDK for cross-platform mobile app development, 2013
- ‘Native Like’ Framework
- Strong Ecosystem
- Based on Apache Cordova
- Low learning curve
- Default UI That Is Easy to Customize
- Strong Community Support
PhoneGap

- Also known as ‘Apache Cordova, 2010
- CSS, HTML, and JavaScript
- Better Access to Native APIs
- Flexibility using Web Technologies
- Robust Backend Support
- UI Libraries improving the User Interface
- Strong Community Base

Flutter

- Google, 2017
- Complete SDK
- Dart – simple and effective language targeted at Java programmers
- High performance
- Hot reload function for instantaneous updates
- Ready-made and custom widgets for fast UI coding
- Internationalization and accessibility
Flutter Overview

- Flutter is a free and open-source mobile UI framework
- Created by Google and released in May 2017
- Native mobile app with only one codebase (no need to develop different apps for different platforms)
- High performance:
  - Dart compiles your code into native code
  - Flutter has its own widgets, avoids using OEM widgets (leads to reduced communication between app and platform, fewer compatibility issues, will work on future OS versions)
- Hot reload
- Rapidly growing community of developers
Flutter Overview

- Easy way to build elegant, native apps for Android and Apple iOS
- Dart is (relatively) easy to learn
- Plenty of examples for almost every situation on the web (Google search)
- Plenty of software tools and cloud tools available for free from Google
- Good documentation and plenty of material on the web
- You can see your code at work almost simultaneously

Flutter is Declarative

```dart
Widget build(BuildContext context) {
  return new Container(
    height: 400.0,
    padding: const EdgeInsets.all(8.0),
    child: todos.isEmpty
      ? new Text("empty, try to add something")
      : new ListView(
          children: todos.map((todo) => new Text(todo)).toList()));
}
```
Dart async/await

```dart
Future<Data> loadData() async {
  final json = await service.get();
  final data = await deserialize(json);
  return data;
}
```

Flutter Overview

![Flutter Framework Diagram]
Everything is a Widget

Everything is a Widget
Stateless Widget

- Immutable
- Once created, it doesn’t change

```dart
class HelloWorldScreen extends StatelessWidget {
  final String message = 'Hello world';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Text(message),
      ),
    );
  }
}
```

Stateful Widget

- Have a “State”
- State - set of data held by a widget that can change in its lifetime

```dart
class CounterScreenState extends State<CounterScreen> {
  int _counter = 0;

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Counter Screen'),
        ),
        body: Center(
          child: Text('Counter: $_counter'),
        ),
      ),
    );
  }
}
```
Anatomy of Flutter

Anatomy of Flutter
Anatomy of Flutter
Anatomy of Flutter
Anatomy of Flutter
My First Flutter App

- Install Flutter:
  - [https://flutter.dev/docs/get-started/install](https://flutter.dev/docs/get-started/install)
- Install your favorite editor:
  - [https://flutter.dev/docs/get-started/editor?tab=vscode](https://flutter.dev/docs/get-started/editor?tab=vscode)
- Test drive:
  - [https://flutter.dev/docs/get-started/test-drive?tab=vscode](https://flutter.dev/docs/get-started/test-drive?tab=vscode)
- My first app:
  - [https://flutter.dev/docs/get-started/codelab](https://flutter.dev/docs/get-started/codelab)
- Optional readings:
  - [https://flutter.dev/docs/get-started/learn-more](https://flutter.dev/docs/get-started/learn-more)

My First Flutter App

- Deliverable:
  - Submit brief video (< 30 seconds) of your app in action (emulator or device) via Sakai
  - Submit brief document (max. 1 page, pdf), stating your name, people you worked with (and how), and describe any aspects of this project that caused you unusual difficulties (if appropriate); also describe if the project is incomplete or faulty (provide specifics about why you weren't able to complete the assignment)
- Deadline: 2/17 11.59pm EST
Final Project

• Build upon your code from the mini projects to develop a complete mobile app that solves a problem of your choosing
• Project should have, at least, three “significant” features that go beyond what you developed for the mini projects

Potential Project Features

• Camera, microphone
• Accelerometer, gyroscope, barometer, magnetometer, …
• Social media
• Back-end integration
• Networking/communication features, NFC/RFID
• Push notifications
• Input/output features; control of objects; etc.
• Location-awareness
• Touch, swipe, etc., interface
• Account management, sign up, log in
• Mobile payment
• Build/re-design new hardware; 3D print
• Augmented reality, virtual reality
• Exploits various senses
• Third-party frameworks, libraries, services, features, …
• …
Project Structure

- Project proposal
- 2 written progress reports
  - Reports can also be used to adjust proposal if needed
- Final report, delivered with code
- Brief in-class demo/presentation of project

Project Samples

- The following slides show examples from previous years’ courses on Mobile Computing, Pervasive Computing, Smart Health, etc. (so not examples are purely software and none of them was built using Flutter!)
Biometrics Sensing

- Electrocardiography
- Electromyography
- Galvanic skin response
- Blood pressure
- Body position
- Air flow
- Temperature
- Blood sugar

HelpHub

| Category: Food Delivery | Description: Chloe Burns
| Will Cost: |
| Will Pay: |
| Location: |
| Start At: |
| Done By: |
| Details: |

| Category: Food Delivery | Description: Five Guy's
| Cost: $7
| Tip: $5
| Location: 1855 Vanessa street sou... |
| Start At: Dec 19 - 5:00 PM
| Done By: Dec 19 - 7:00 PM
| Details: Cheese Burger and Fre... |
BlitzPool

- Game engine
- Physics libraries
- Real-time features
- Game center

ND Sporter

- [https://www.youtube.com/watch?v=KxeTRTN0t10](https://www.youtube.com/watch?v=KxeTRTN0t10)
Camp-Arduino

ChargeButler
Pen Pal

- Anonymous communication anytime based on location and/or interest.

Scout
ClassShare

Bouncer
Hello App

UQueue

UQueue
- Broadcasting
- Likes/Dislikes
- Requests
Monster Hunter

The angel of death has come. Are you prepared?

UAV Tracker
iPhone Doctor

EasyEMR
Data Visualization

MedAlert for Doctors
MoodTracker

FaRCE
Portal

RideAlong

Solution: RideAlong

- Ride-sharing app specific to Notre Dame
- Can directly connect students who have cars with students who do not have cars
  - More specific than the Notre Dame Facebook groups
  - More likely to be used for shorter trips (e.g. to the store, rather than only to the airport)
- Students can both offer and request rides
De-Icing Sensors

Automatic Infrared Temperature Sensors
- Long Range IR Temp. Measurements (30ft)
- Low Cost
- Accurate
- Real Time Monitoring

Garden GROW
Other Projects

• Book Club
• Parking Space Finder
• Movie Finder
• Pick Up Sports Organizer
• LYNX Taxi Service
• Hearing Assessment App
• MySquirrel
• Occupational Therapy Exercises
• Stress Detector
• SnapnSpray
• Twext: Learning Betwixt Texts

That’s it for today…

• Next up: Mobile app development, tools, best practices