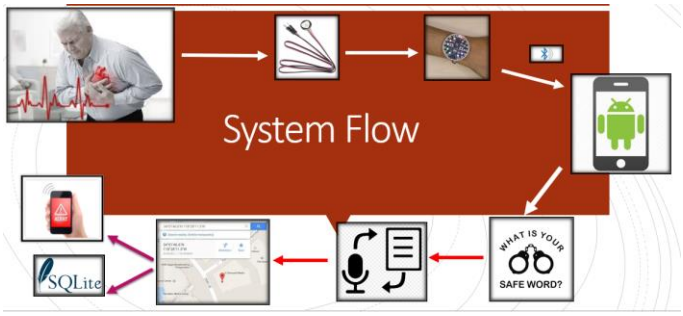
EE P 523: Mobile Applications for Sensing and Control

Additional resources Spring 2020

Example Final Projects from previous years

- ***SafetyBeats:*** automatic safety data collection in stressful situations



- ***SoundHero***: User attempts to match the kick drum by shaking the phone to the beat https://www.youtube.com/watch?v=-HPOUgXly5M

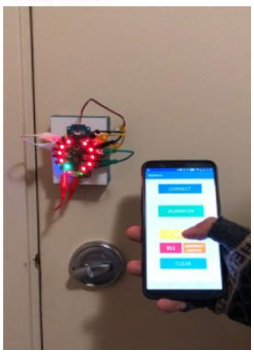
- ***Custom Dice roller*** (>10k downloads!) https://play.google.com/store/apps/details?id=com.fialasfiasco.customdiceroller

- ***CatLaser***: fun laser for pets controlled by a smartphone

- ***Treats dispenser***: smartphone-controlled pet treats dispenser https://www.youtube.com/watch?v=PCJbFCdNwso

- ***Travker = Travel + Tracker***: travelogue to record interesting moments during a trip https://www.youtube.com/watch?v=rvEOGNc1cqc

- ***myAlarm:*** remote control of home alarm, emergency calls, and door opening alert.



Kotlin vs Java

Official support for Kotlin for Android development was announced at Google I/O in 2017. Before that, there was an underground movement of Android developers using Kotlin even though it was not officially supported. Since 2017, Kotlin has become widely adopted, and it is most developer's preferred language for Android development. The tide has continued to turn toward Kotlin in a very big way.

The Android framework team has started adding *@nullable* annotations to legacy platform code. They have also released more and more Kotlin extensions for Android. And Google is in the process of adding Kotlin examples and support to the official Android documentation.

The Android framework was originally written in Java. This means most of the Android classes you interact with are Java. Luckily, Kotlin is interoperable with Java, so you should not run into any issues. You can see the Java API listings by browsing for the class you are interested in at developer.android.com/reference.

How to become a good Android developer *after* taking this course?

Good developers are each good in their own way, so you must find your own path after taking this course on. Where might you start? Here are some starting places:

- ***Write code***. Now. You will quickly forget what you have learned here if you do not apply it. Contribute to a project or write a simple application of your own. Whatever you do, waste no time: Write code.
- ***Learn.*** You have learned a little bit about a lot of things in this course. Did any of them spark your imagination? Write some code to play around with your favorite thing. Find and read more documentation about it – or an entire book, if there is one. Also, check out the Android Developers YouTube channel ***(youtube.com/user/androiddevelopers)*** and listen to the Android Developers Backstage podcast ***(androidbackstage.blogspot.com)*** for Android updates from Google.
- ***Meet people.*** Local meetups are a good place to meet like-minded developers. Lots of top-notch Android developers are active on Twitter
- ***Explore the open-source community***. Android development is exploding on github.com. When you find a cool library, see what other projects its contributors are committing to. Share your own code, too – you never know who will find it useful or interesting. The Android Weekly mailing list ***(androidweekly.net)*** is a great way to see what is happening in the Android community.