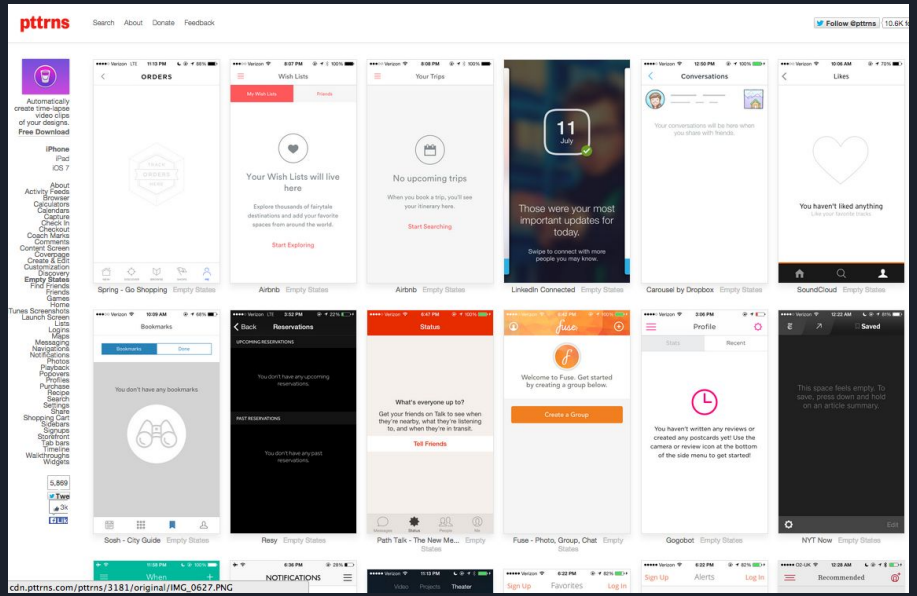
Kotlin Android Fragments Architecture

CS 402: Mobile Development

Ideas

[pttrns](#)



Topics

Planning

Setting up Main Activities

Design Patterns

Modularization

Ideas

Ideation is hard.

Keep a notebook for your ideas

Who is the audience?

Is it free, freemium, paid, ad, etc?

Failing to plan is planning to fail

Scope out the intent of the app

Make a list of features:

needs to haves

nice to haves

future stretch items

Fail Early, Fail Often

Design with sketches to rough out the idea

Wireframe with a tool

Proof of concept anything tricky

Use a design tool to mock out each screen

Prototype your idea

Budgeting - Time and Money

Can’t manage what you don’t understand

Can’t change what you can’t measure

List out each feature and mark it with a time estimate

3x those estimates

Hourly rate? ($ x 2,000 = yearly gross income)

Additional assets, software, services?

Commit Early, Commit Often

Source control your projects

Enable continuous integration such as Fastlane

Use CocoaPods or Carthage

User Testing

Will reveal a ton of what you took for granted.

Will expose issues your users have.

Will uncover thoughts someone new has.

Iterate

Release early, release often.

Release, release, release.

Approval ~2 hours, so almost no penalty in releasing again

Enable crash reporting and analytics

Setting up your Main Activity

A Main Activity is launched, based on the Launch Intent

It controls which layout is inflated and what other layouts or fragments will be displayed after that.

Singletons - Only one instance will ever exist

Factories - Objects whose sole purpose is to produce other objects. Ex. (JSON -> Swift object, or MazeFactory.createMaze())

Model-View-View-Model - Architect Android uses for development

Android Documentation on MVVM

Modularization

Wrap functionality into their own classes

Any kind functionality that needs to be accessed by several items is a candidate for a singleton

Activities or Fragments

Fragments provide a reusable user experience

Think MapView, WebView, your CustomView

Fragments control only their views and hold just enough info for their operation

Think markers, web pages, or info to draw your view)

Fragment transactions allow for "Back" or swapping out fragments within an activity

Activities or Fragments

Activities display multiple fragments

Activities can be launched from another activity or another app

Activities can use fragments (without a UI) as invisible workers

Activities can choose to display more fragments if it's a tablet

Activities or Fragments

Fragments should not care which activity they're in

Move all logic into fragment, keep activities simple

Activities or Fragments

Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.

You aren't gonna need it

XP practice of "do the simplest thing that could possibly work"

Any software engineering problem can be solved by adding a level of abstraction, except of course too much abstraction.

Always Have a Plan