### **Decomposition cheatsheet**

#### Choosing a layer



## < Shared

Code that is not specific to your application, code that serves as a foundation.

#### Self-check question

Can this code be used in a pizza shop app or an online bank?

Example: a <u>dropdown menu</u> can appear in a pizza shop app. a social meda post probably can't.

# **Entities**

Code that represents a real-life concept that your app is working with.

#### Self-check question

When describing your app, does this word appear as a subject or an object? Do your users/clients understand that word?

Example: <u>users</u> can write <u>posts</u>. Clients want to be able to add videos to their <u>posts</u>.

## Widgets

Code that combines the layers below to form meaningful blocks, interactive and complete with data.

#### Self-check question

When looking at your app's UI from a distance, does this stand out as a complete "block"?

Example: <u>A list of posts with pagination</u> and <u>the header</u> appear as standalone blocks.

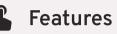
### Pages

Entire screens of your application, built mostly by combining the layers below. Similar to widgets, but on a larger scale.

#### Self-check question

Is this code ready to be plugged into the router and work for users as is?

Example: the <u>home page</u> of an online shop with login, fresh deals, categories, search, etc.



Interactions that provide real-life value to your app's users, the things people want to do with your entities.

#### Self-check question

When describing to a stranger what your app does, do you mention these actions?

Example: users can <u>write</u> and <u>edit</u> posts. Posts can be configured to <u>auto-delete</u> after 5 minutes.

# 🌣 Арр

Infrastructural code that makes your app actually work.

#### Self-check question

Is this something your framework or technical stack needs for your app to function?

Example: an <u>i18n provider</u> and a <u>router</u> make the app work and display sensible text to the user.