Octopus technical challenge

This document describes a technical challenge that forms part of the Octopus recruitment process for mobile engineers. Use the challenge to best show us how you develop great software and slick products.

Note: if you already have a small project that you're comfortable showing off and talking about, we're happy to take that in place of the technical test. We just want to get a good idea of your approach to development. If you do send a small project please read through the rest of the test and ensure it covers similar requirements to make sure we can properly assess your technical ability.

Background

Every website on the internet with cat gifs has mysteriously gone down. You smell your chance to create a cat gif app, make millions, and retire young to an island in the Caribbean…

Requirements

Deliverable

After the allotted time, a zipped file should be emailed back to your recruiter. This file should contain:

- The mobile project
- A .git folder with the commit history for the project
- A README that gives some context on your approach to the technical challenge and anything else you'd like to have added

The project should build successfully and run on a simulator/emulator.

Technical requirements

- The technical challenge should be built in Jetpack Compose or SwiftUI.

Application requirements

The application should use the The Cat API to display pictures and information about a user selected cat breed

At a minimum the app should contain the following:

- An interface to allow the user to select their desired cat breed
- A list displaying cat pictures of the chosen cat breed

Hints for the API

- Sign up to receive a auth token
- https://docs.thecatapi.com/example-by-breed

Stretch Goals

- Display information about the chosen cat breed
- Paginate the displayed list
- Network results are cached in memory
- UI changes are animated

Application notes

- The UI design of the app is up to you (pst, we love to see a slickly designed UI)
- The app should be architected in such a way that it's simple to add further selection criteria
- Testing is encouraged but not required, at a minimum the code should be architected in such a way that adding tests is simple
- Show off any architecture patterns you particularly like to use

Good Luck!