

Code Challenge



The Challenge

We would like you to write an application using **Kotlin** that utilises the Cat API (<https://thecatapi.com/>). You can implement the UI as you prefer. Feel free to use any libraries you feel are appropriate solutions to your technical needs. Also, write a short readme explaining the strategies decided during the development of the challenge.

Tip: The Cat API provides OpenAPI Spec and Postman docs.

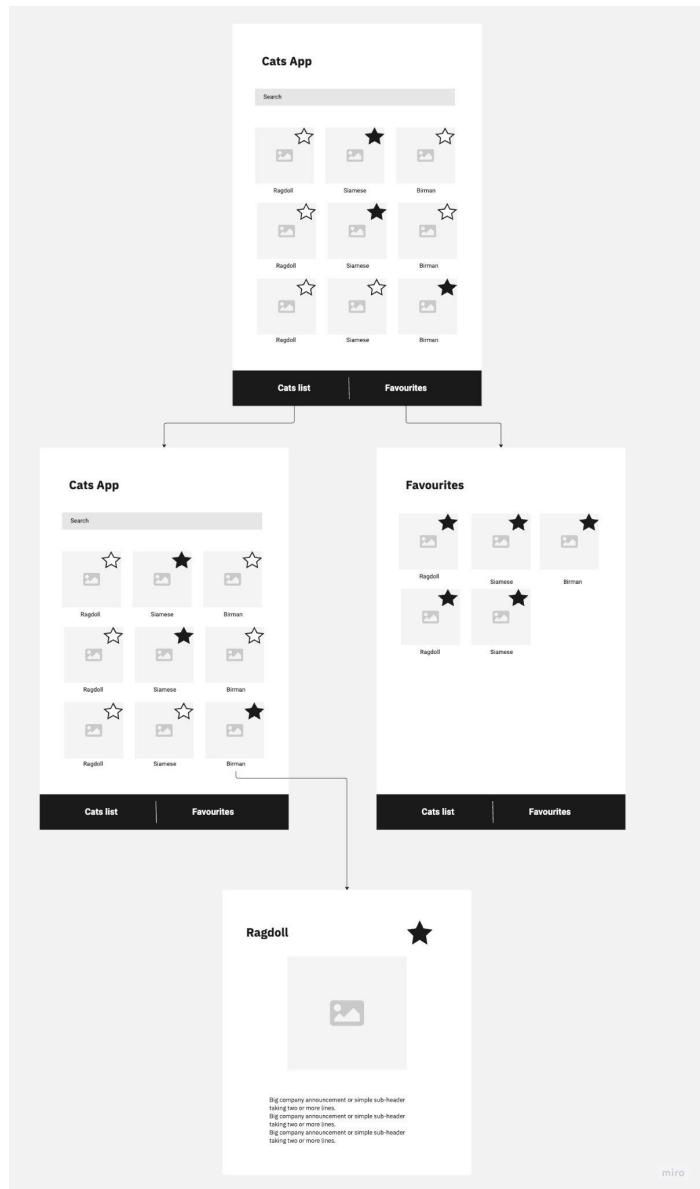
To deliver your code, we would like you to create a complete project in a **version controlled repository**, make it public, and share the link with us so that we can access it.

Preferably we would like you to perform **several commits** so that we can see how you organize your thought process during development so that we can see your progress and understand a bit about your way of thinking.

The Requirements

Your Application should have the following functionalities:

1. A screen with a list of cat breeds, with the following info:
 - a. Cat image;
 - b. Breed name.
2. The cat breeds screen should contain a search bar to filter the list by breed name.
3. The cat breeds screen should contain a button to mark the breed as favourite.
4. Implement a new screen to show the breeds marked as favourites.
 - a. Show the average lifespan of all the favourite breeds (you can use either the lower or the higher value in the range).
5. Implement a screen with a detailed view of a breed, with the following info:
 - a. Breed Name;
 - b. Origin;
 - c. Temperament;
 - d. Description;
 - e. A button to add/remove the breed from the favourites.
6. In order to navigate between the different screens use a Jetpack Navigation Component.
7. Pressing on one of the list elements (be it in whatever screen) it should open the detailed view of a breed.



Here is a low-fidelity wireframe to serve as inspiration:

Technical Requirements

- MVVM architecture
- Usage of Jetpack Compose for UI building
- Unit test coverage
- Offline functionality (i.e: consider using Room for data persistence)

Bonus Points:

- Error Handling
- Pagination for the list of cat breeds
- Modular design
- Integration and E2e tests

Some Considerations

Your submission will be evaluated on:

1. Implementation of the stated requirements
2. Application Architecture
3. The general quality of the code and it's resistance to crashing
4. Your use of Android coding conventions
5. Knowledge and usage of Android libraries and SDKs
6. UI and UX -- we don't want you to spend any time creating custom icons or other assets, the app should look clean and generally obey the Human Interface Guidelines

©2023 Sword Health Inc. All Rights Reserved

help@swordhealth.com | +1 385-308-8034

