CVSHealth

Android Developer Candidate Code Challenge

Congratulations on making it to the next phase of the interview process. We've developed this code challenge to help us evaluate where you are on your Android developer journey.

We'd like you to spend no more than 5 hours on this. We understand this is a serious time commitment, and we appreciate your willingness to help ensure you'll be a great fit with our Android development team.

We'd like for you to create a simple application from scratch. The goal of this exercise is to show us your skills. We'll ask you to show your app to a small group of Android developers (a code review). You'll demonstrate the working app, and then walk them through the code. They'll ask questions about your approach to solving the coding challenges faced while developing this application.

Good luck!

Get Started

Pre-Requisites - Install Android Studio

API Requirements - Your application must run on a recent API

Note Feel free to use outside resources. (e.g. Google, Stack Overflow, Ask us questions, Etc...)

What We're Looking For

During the code review, we'll be looking for certain things that might indicate you have the skills we're looking for. These things include:

- Excellent communication skills.
- Good coding practices. Industry-standard patterns and techniques.
- Well-formatted source code, a clean and consistent style, and proper naming.
- We'll ask you to show your app on simulator devices with different screen sizes.
- API calls and other routines that never block the main thread.
- Code without any memory leaks.
- Proper error handling.

CVSHealth

We expect you to fully understand every line of code in the app and be able to explain exactly how it works and why it's the best choice if asked.

The Rules

We expect **you** to write all of the code (excluding open-source libraries or auto-generated code) for this challenge. We'll know you wrote the code if you can clearly explain it during the code review.

We'll need you to write this app in Kotlin. Java experience is highly valued, but Kotlin language experience is required. React Native and other hybrid solutions are not acceptable.

You can use either Legacy XML or Jetpack Compose for the user interface.

Use any architectures, patterns, or techniques that you feel are appropriate for this task. Play to your skillset.

You may use the internet for help just as you normally would to find answers and code examples.

You can use any open-source libraries in your project.

You can ask for help or clarification at any time. Contact the person that sent you this code challenge and we'll do our best to respond quickly. But if you don't hear from us, keep coding so we have an example of your work to discuss during the review.

Task 1

Your challenge, should you choose to accept it, is to create, from scratch, an Android application that allows a user to search Flickr for images. Create a user interface with a search bar at the top and a list or grid below it to display thumbnail images. The user should be able to enter text into the search bar and see a list or grid of images whose tags match the search string. The search string can be a single word (such as "porcupine") or comma-separated words (such as "forest, bird"). As the user changes the text in the search bar, update the list or grid of images.

Fetch the list of images using this API from Flickr:

https://api.flickr.com/services/feeds/photos_public.gne?format=json&nojsoncallback=1&tags=porcupine

CVSHealth

This is a free public feed. No API key is required. You can learn more about this API here: The App Garden

Acceptance Criteria

- The search results should come from the API listed above.
- A search box and search button should be at the top screen.
- A list should be below the items in #2, and should be a grid with 2 columns.
- Each item displays the image and title.
- The title should be centered under the image.
- When performing the search, indicate progress while not blocking the UI.
- Make sure to handle state during orientation change
- The user interface must remain responsive to user interaction at all times.
- The user interface should be attractive and follow Android style and theming guidelines or Compose theming.

Task 2

As a user when I click on a photo from the list and see more information about it. When the user clicks on a photo they will navigate to a new screen for that photo.

Acceptance Criteria

- Clicking a photo from the list will take the user to a photo detail page

For any photo:

- The detail view shows the image, title, description, image width, image height, and author
- The image should span the entire screen horizontally.
- The details listed in #2 should be layout nicely below the image
- Make sure to handle state during orientation change

CVSHealth

Task 3

As a user, I would like to save my recent search terms so I can quickly search for them again. On the search screen from Task #1, use some local persistance to save recent searches. This will only save the search term that is entered, no response data is saved.

Acceptance Criteria

- When the user taps the search button, the search term is saved to local storage. You can use persistance of your choice (e.g. shared preferences, SQLite, etc...)
- When the search box is empty, the user will see a list of recent search terms. Include a "Recent Searches" header above the list.
- If there are no searches yet, the screen will show the user the following label: "You have no recent searches. Enter a search term above." with the header
- The recent searches list should have a max of 5 items, with the most recent on top.
- If a 6th search is done, remove the oldest term from persistance and save the new term.
- Once the user taps on a recent search item, the title and recent search list will disappear and the term will be searched for and show the results as before.
- Make sure to handle state during orientation change.

Task 4

Testing - Testing plays a huge part in the quality of an application

Acceptance Criteria

- Write a passing unit test which tests any method in your app
- Write a passing espresso test which tests any screen of your choice

Task 5

Accessibility - The Aetna Health app is used by a wide range of members with disabilities. For that reason increasing font size as well as talkback need to be implemented with those members in mind.

Acceptance Criteria

- Every screen in your app should gracefully handle itself when the font size is bumped up to the largest setting.
- When talkback is enabled, every screen should navigate through one item at a time in a logical order.