BS23 Android Task - 101

Top Android Repositories

Develop a simple Android application which shows the most starred github repositories by searching with the keyword "Android". You are open to develop this project in any architecture, pattern or packages. Although it is a simple project, we expect you will develop it as a large project. I mean, you should consider it a large scalable project which will add more and more features in future. We want to understand your approach and coding style for a large project. So try to create a maintainable, readable, reusable project with all best practices you know. Hope we will be impressed by your awesome codebase. Best of luck!

Requirements

1. Fetch repository list from GitHub API using "Android" as query keyword. (API doc).
2. The fetched data should be stored in a local database to permit the app to be used offline mode.
3. Fetching the repository list should be paginated by scrolling. Each time by scrolling, fetch 10 new items.
4. The required data can be refreshed from the API no more frequently than once every 30 minutes.
5. Show the list of repositories in the home page.
6. List can be sorted by either last updated date-time or star count (add a sorting button/icon)
7. Selected sorting option persists in further app sessions.
8. A repo details page, to which navigated by clicking on an item from the list.
9. Details page shows repo owner's name, photo, repository's description, last update date time in month-day-year hour:seconds format, each field in 2 digit numbers and any other fields you want
10. The repository list and repository details data which loaded once, should be saved for offline browsing.

UI Suggestion

There is no defined UI design for the assignment. You are free to design anything good. You can reflect your UI/UX concept here. Our basic requirement is (1) a list of repositories, (2) sorting icon, (3) repository details page. Now based on it, you can design any suitable UI. Also if you want, you can implement more features.

What we’re looking for

1. An app that meets all of the functional requirements above.
2. Your coding style - show us how you like to write your code (You know, coding is an art!).
3. Architecture - how you’re structured your code.
4. Principles - how you believe code should be written (OOP, SOLID, DI etc).
5. Quality - how you guarantee your code is functioning correctly.

Extra Credits

These requirements are not mandatory. But they will add huge extra points.

1. **Unit Testing coverage**.
2. UI Testing coverage.
3. Dagger Hilt, MVVM, Coroutines/Rx, Repository pattern with appropriate abstraction layer.
4. Central API call error handling.
5. App’s flavor (ex: DEV, QA, PROD etc).
6. Right now there is no need for adding auth tokens with API calls. But you can implement the way to handle it for the future. Also can add some retry mechanism when the API fails.

Special Instructions

1. Commit your project to a public **GitLab** repository from the initial setup (it’s Git Lab, not GitHub).
2. Make your repository public and share the URL with us so that we can check your progress.
3. **Provide a README file which will contain the overview of the project** (app screenshots, diagram of project architecture, sketch or anything you think important). If there are any unfinished tasks, please mention them in the README file. You can also add the future scope of the project to improve the code quality. The README file will showcase your work, so be concerned about it.
4. **Commit early and often**. Don’t commit everything after finishing the assignment. By reading your commit message, we will try to get an idea of how you approached the problem.
5. Make sure your project successfully runs on a real device.