TheD. Gmbh Android Task.

This task is meant to assess the technical expertise of
the candidate through a module that is very similar in its nature to a real module in a project.

The Points that are focused on in our assessment is:

- Knowledge of the Platform.
- **Class design and architecture.**
- **Code Quality and naming conventions.**
- Technical knowledge of used libraries.
- Conformance to the requirements.

Task Description.

The task required is a demo for a products listing and details. It should show the user a list of products clearly and can open the details of the product by selecting it.

The Listing should display all available items in a vertical List, each item should display:

- Product Image
- Product Name
- Price

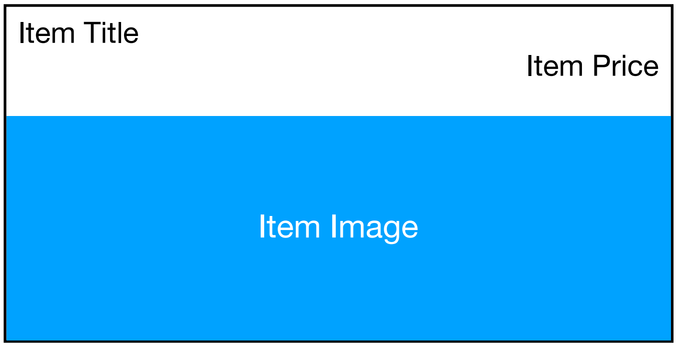
Upon opening the details of the item the product description should appear.

Technical Description.

- ### Application screens

The App should consist of 2 screens, the first is the listing of all the products, the second is the product details.
The API to retrieve the List of products is:
https://limitless-forest-98976.herokuapp.com

You are required to display the list where each item to appear in a similar manner.
Note: Make sure the content in the cell fits most screen sizes on the phones, no special support needed for the tablets for this task.



Dynamic Image Size

The items have a different image sizes, some are shorter than others, make sure the images in the cells fit properly without any additional space or being clipped vertically or horizontally (***Note***: The API provides the dimensions of the image)

Database Caching

When you relaunch the app show the data from the last session, and when its relaunched with no network connection show the data normally and allow the user to interact with it with no errors, we prefer using Room or Realm for that purpose. (***Note***: you can inform the user in a non-distracting way that there is no network connection)

Also make sure that the images are not downloaded every time.

Notes

- Use MVVM as your architecture patterns.
- Use Dagger 2 as your dependency injection framework.
- Use LiveData for your subscriptions.

Bonuses

- Write test cases for your code. (For all layers, JUnit preferred)

Code Requirements:

- Use minimum API version 16, Target API version Latest.
- Use **Kotlin** as the development language.
- **Pay attention to code quality, testable, readable and maintainable app architecture, and we recommend Google recommended app architecture.**
- 3rd party libraries can be used, provided that the project stays compilable as a standalone package.
- Document the classes and methods, no need to write a documentation line-by-line.
- No fancy design is required, but everybody loves a good looking UI.

Submission:

- You will need to send your task through a code repository (bitbucket, github, …).
- Duration of this task is 1 week from receiving it.
- Make sure the project is tested and is stable, its not exciting to see an assessment crashing frequently. :-)

The task will be reviewed by the Android Team. The review will take all aspects of your implementation into account.