First Orion Demo Project

As a part of the interview process for Android Developer position at First Orion, you are asked to complete the given Android project.

The goal of this project is to create a simple app with the following functionality and technical detail.

The app is going pull the `Posts` from a fake REST API and display them in a list view (see Page #1). When a `Post` item is clicked, the `User` detail screen is going to open. The `User` data is going to be pulled from the same REST API. `User` data can be also combined with the `Post` data on this page (see Page #2). The last page is going to allow the user to create a new `Post` and send it to the REST API (see Page #3).

In total, the app is going to have at least 3 screens. One Activity, and two Fragments are provided in the given project for this purpose. We keep the "Create New Post" page open for your creativity; you can use a new Fragment, a Dialog or any other Android components you'd like to use.

The `Posts` and `Users` data are going to be fetched from a fake REST API (https://jsonplaceholder.typicode.com/) and stored in the app's database. If the same data (either all the Posts or a User data) needs to be fetched again, that data should come from the database instead of downloading it from the REST API again. Once the application (Activity) is closed, the database should be cleared.

Please read the REST API documentation on their website before you start the project.

Requirements:

1. The data fetched from the REST API must be cached in the database until the app is closed.
2. There must be 3 pages in the app:
   a. List of all the Posts – the main screen
   b. User details (combined with the Post info) – when a Post is clicked
   c. Create a new Post – allows users to create and upload a new Post.
3. The provided interfaces and classes must be used and must not be removed. Modifications to function parameters and return types are allowed.
4. ViewModel classes must be used to load data from repository layer to the Fragments.
5. UI/UX design is up to you, you do not have to stick with the UI examples provided.
   The data fields that will be shown on the Post list items and User information page is also up to you. What we expect is a good looking yet simple UI UX design.
   Please use material design where possible.
6. Kotlin. (Tip: If you don't know Kotlin, you can write it in Java and convert it to Kotlin in Android Studio)

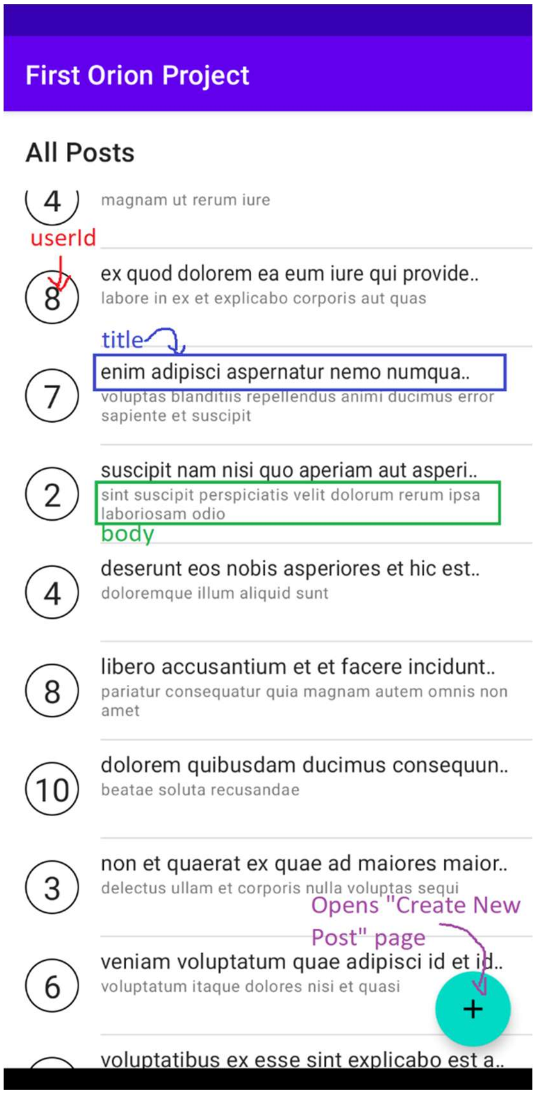
Optional but Preferred Libraries:

1. Retrofit – network layer
2. Room – database layer
3. LiveData or Kotlin Coroutines in ViewModel classes.

   You can use your favorite libraries as needed.

Sample UI

Page #1 - List of Posts can be designed like this. For more information and UI specifications please visit https://material.io/components/lists



First Orion Project

All Posts

4 magnam ut rerum iure

userId

8 ex quod dolorem ea eum iure qui provide..
labore in ex et explicabo corporis aut quas

title

7 enim adipisci aspernatur nemo numqua..
voluptas blanditiis repellendus animi ducimus error sapiente et suscipit

2 suscipit nam nisi quo aperiam aut asperi..
sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio

body

4 deserunt eos nobis asperiores et hic est..
doloremque illum aliquid sunt

8 libero accusantium et et facere incidunt..
pariatur consequatur quia magnam autem omnis non amet

10 dolorem quibusdam ducimus consequun..
beatae soluta recusandae

3 non et quaerat ex quae ad maiores maior..
delectus ullam et corporis nulla voluptas sequi

Opens "Create New Post" page

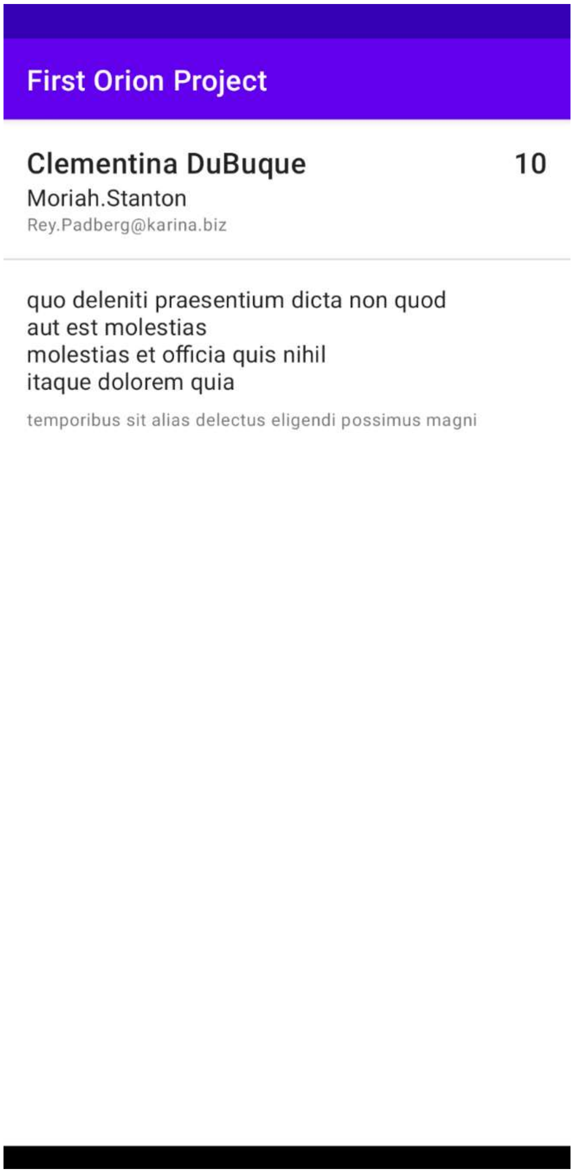
6 veniam voluptatum quae adipisci id et id..
voluptatum itaque dolores nisi et quasi

voluptatibus ex esse sint explicabo est a..

Page #2 - User details page will be opened in a new Fragment when a Post is clicked. The amount of data that is displayed on the screen is totally up to you. You do not have to show every data field in the User object you get from the REST API.

A sample transition to User details page can be like this. The same video can be seen under this link:

https://material.io/components/lists#behavior



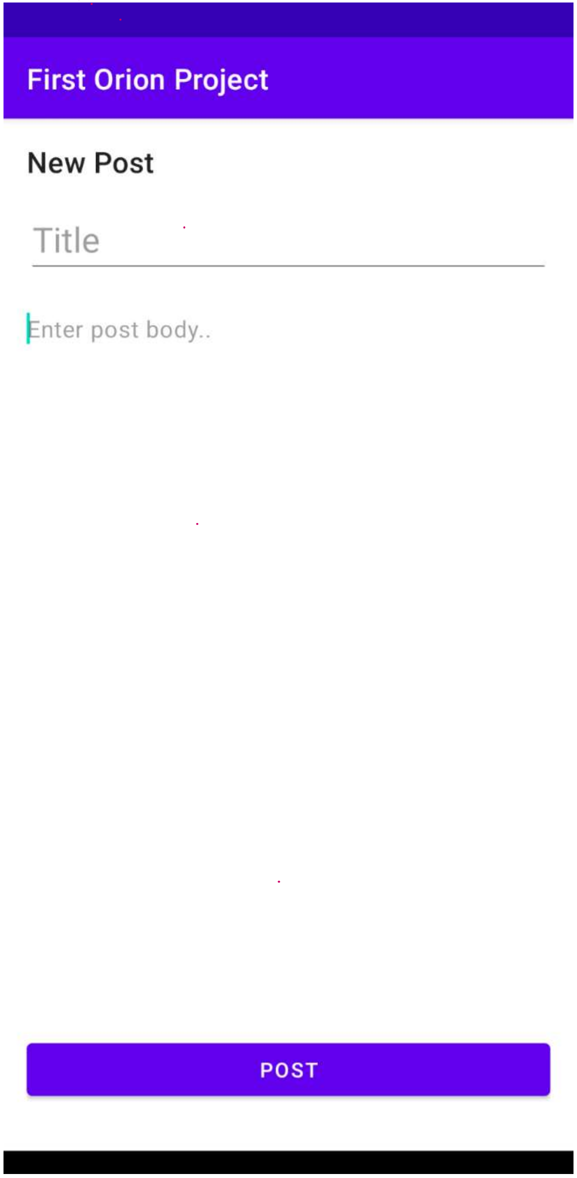
Clementina DuBuque 10
Moriah.Stanton
Rey.Padberg@karina.biz

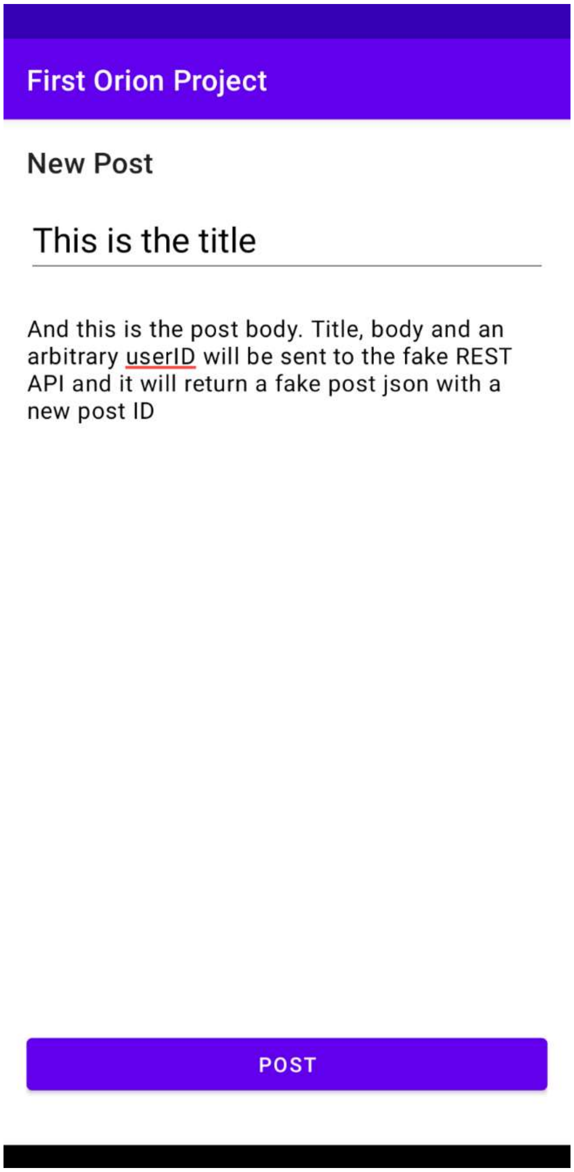
quo deleniti praesentium dicta non quod
aut est molestias
molestias et officia quis nihil
itaque dolorem quia

temporibus sit alias delectus eligendi possimus magni

Page #3 - "Create New Post" screen can be triggered with a Floating Action Button. Sample usages are depicted here: https://material.io/components/buttons-floating-action-button

This page should let the user to enter “Title” and “Body” and upload this post using the fake REST API.





This is the title

And this is the post body. Title, body and an arbitrary userID will be sent to the fake REST API and it will return a fake post json with a new post ID