Assignment 3: Network Project

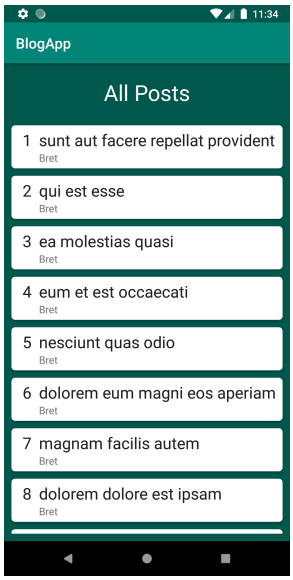
Pablo Guerra

I. Introduction

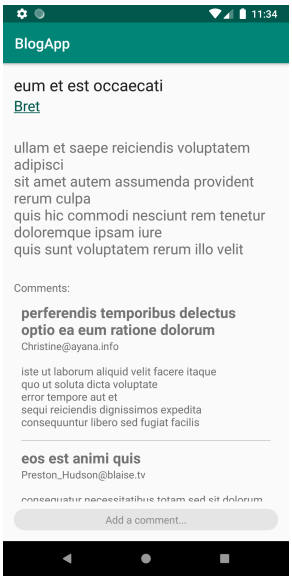
For our fourth assignment we were tasked with the job of developing an application that implements a REST Client interface to implement the front-end of a mock blog application. This application would request data from an online source to populate views of posts, users and comments. The data retrieved was in the form of JSON objects/arrays and were used to gather the necessary information for the required view. The user would also be able to post a comment on a post that would be sent to the server and displayed but would not persist.

II. Application Design

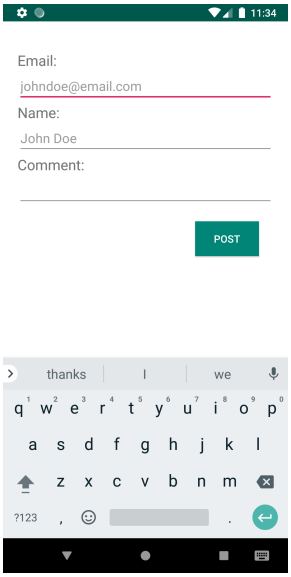
Unlike the previous assignments, the design process did not start in an external application like Adobe XD or Photoshop. Instead the design process was contained entirely in Android Studio. Below are some photos of the final application screens.



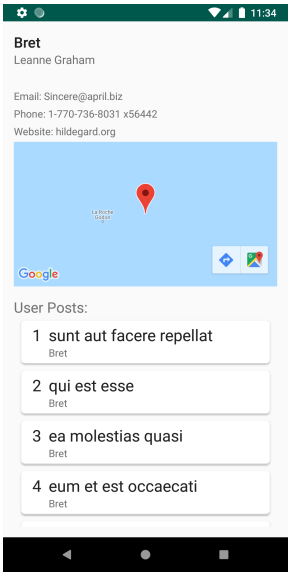
Posts List View



Single Post View



New Comment View



User View

A. *Screen Views*

As seen from the previous photos they are four main views that the user will see.

1. The Posts List View is the initial view that is shown when the user launches the application. This page shows a list of posts with the post ID and the user that posted it.

2. The second view is the single post view which shows the title, user and body of the post. This view also contains a list of comments for the post and a button for adding a new comment.

3. The third view is the new comment form. Here, the user is required to enter an email, name and comment. Once all fields are completed and the user presses the post button, the information is sent to the server and a response is displayed as well as the comment being added to the comments list.

4. The User View, which is the final view, can be launched from tapping on the username in the single post view. Here, the user information is displayed along with their location on the map and a list of their posts.

B. *App Functionality*

In order to make these views work properly and load the proper information, it is important that all the necessary information is obtained, parsed and displayed. Using the volley library, we can make request to the server to get the json arrays and objects. With this data, we can create user, comment and post objects than can be retrieved and used to populate the lists. We must also listen for user inputs to see which object the user is tapping to load the next view. For the new comment view, a popup window is displayed with a form that requests the necessary information to be completed. Once completed and the user taps the post button, we then send the data to the server which will respond with the accepted data if it is sent properly. The google map API was used in the user view to show the users location with a marker. The map was loaded as a lite version since the purpose is only to give a general view of the location and can be opened in the map application if the user wishes to see further details.

III. Results

After finishing the design and implementation, several cases were tested, including test cases provided in the grading rubric, to ensure that the application functioned as intended and that they were no obvious bugs discovered. The application successfully passed all these test cases and met all the functional requirements.

IV. Discussion

Overall the blog application works nicely and meets all the necessary requirements. The application is able to request data from an online server and use that data to dynamically create lists and views based on that data. The app can also send data to the server to be processed. This was a relatively simple assignments to work on but it still provided great insight on sending and requesting data from an online source.