<u>REPORT</u>

<u>Note:</u>

There are few assumptions which I made before developing the application

a) View Controller to be activities
b) Used only basic android components to establish network connectivity

<u>Overview:</u>

There are many efficient ways in establishing HTTP connection on android mobile application. In the recent times, many libraries and open source protocols are developed to perform this operation with a single line of code more efficiently for example Volley, Retrofit and okHTTP. But they lack representation of underlying implementation. Though I could have made use of it, I proceeded with using basic android framework components to avoid high level view abstraction and show underlying implementation.

<u>Design:</u>

The whole project is executed on AVD running on Android 8.1 OS, API 26. This project majorly consists of 3 packages and one Mainactivity (i.e. ButtonViewActivity). Each of these packages consists of helper classes. Following are the packages and respective class details: -

a) com.sbs.android.sbsassignment.categories:
   - This package holds POJO classes which can be used for serializing JSON data retrieved from Network operation.
   - In this case, "Data" is a POJO class we have used to serialize categories data.

b) com.sbs.android.sbsassignment.helper:
   - This package holds all the helper classes which can be used by main activity to perform any kind of non-UI related operations.
   - Definitions such as logic implementations, Network class definitions can be performed in the classes mentioned under this package.
   - In this case, "HTTPHelper" class is used as a helper class to define Network operations such as establishing HTTP connections, serializing retrieved data into POJO classes instances defined in categories package.

c) com.sbs.android.sbsassignment.services:
   - Different types of thread and service class definitions and implementations are part of this package.
   - In this case, "APIService" is an intent service that is used as a non-UI thread operation.
   - There are many android components available to perform thread operations such AsyncTask or AsyncTaskLoaders. But I have used Intent Service to perform this operation as intent service can resist device configuration changes and function properly in the background even when the activity is terminated. AsyncTask fails to do so.
   - Intentservice uses only one thread and perform all the operations sequentially upon it instead of generating multiple threads. Once all the tasks have been executed it automatically shut down the service which is not the case with other types of service classes.

- Declare services in AndroidManifest.xml file.

d) ButtonViewActivity: - It is a Main Activity (View Controller) used perform all kinds of logical functionalities on UI thread. In this case,
   - it is used to generate a button (“REQUEST”) dynamically and position it in the middle of the ViewGroup (Relative Layout).
   - Perform Broadcast receiving operation whenever any Broadcast intents are available in the applications.
   - Dynamically registering and unregistering broadcastreceivers.
   - Rendering XML layouts (activity_button_view.xml) as UI

Screenshots:

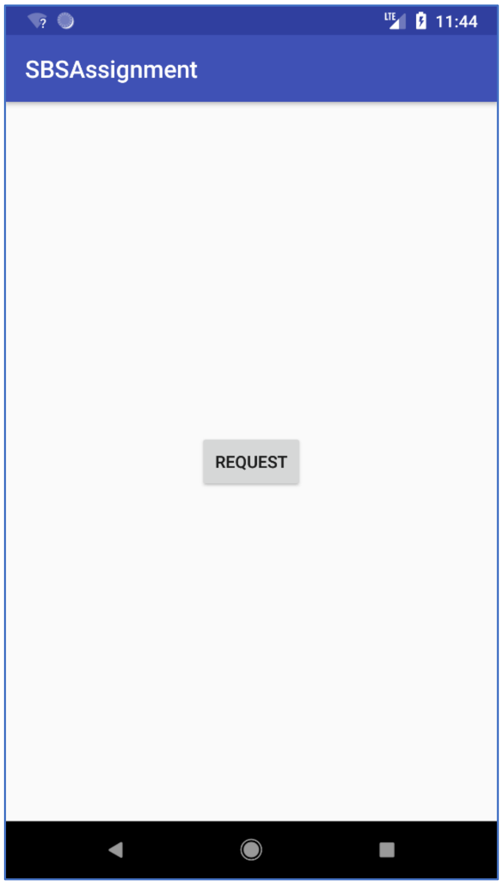
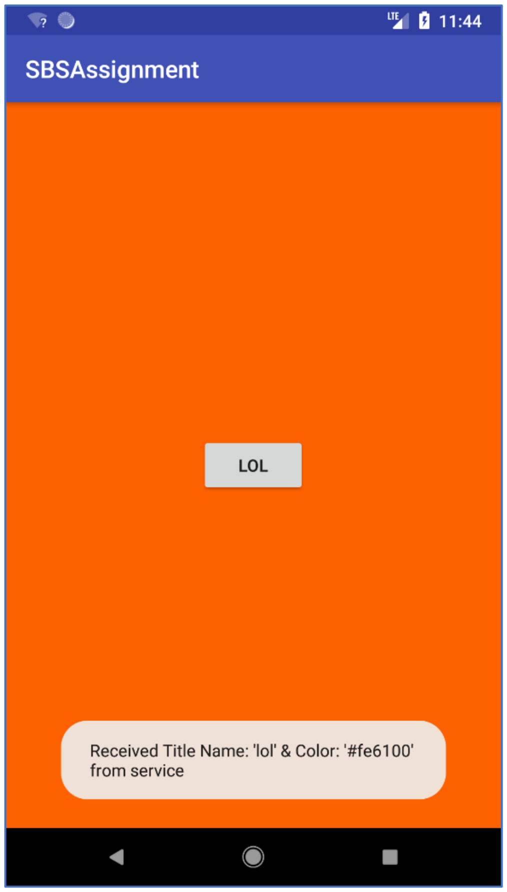


Fig.1 Screenshot Before Network call.



Received Title Name: 'lol' & Color: '#fe6100' from service

Fig.2 Screenshot after clicking Request button and performing Network connection to retrieve JSON Data.

a) Title of “Request” button can be changed multiple times by each click. Every time it is clicked HTTP connection is set up.