

Srikanth Gude – sgcnm@umsystem.edu

Git Hub Link - <https://github.com/gudesrikanth/webcourse/tree/main/Webpart/ICP10>

Gopi Nelluri – gng75@umsystem.edu

Git Hub Link -

https://github.com/gopinelluri9/demo_remote_repository/tree/main/WebPart/ICP10

ICP – 10(Fetching User Details from GitHub using RESTful Services)

Task Description:

In this assignment, we were asked to print the ID and User Name of each GitHub user from the website <https://api.github.com> using Android Studio, as well as any other information we learned.

When we execute the information of GitHub users will be printed on the Android phone. When you scroll down the page, you'll see the information for all of the GitHub users.

Process:

activity_main.xml

```
<androidx.core.widget.NestedScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:ignore = "MissingConstraints">

    <TextView
        android:id="@+id/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="25dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.core.widget.NestedScrollView>
```

In above code snippet we added NestedScrollView for nested scrolling from different positions of view change and will Handle scrolling response to up or down scrollings. And @+id/textView will displays 'Hello world' text.

MainActivity.java:

The Functionality of app is defined in this file.

```
textView = findViewById(R.id.textview);

Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.github.com")
    .addConverterFactory(GsonConverterFactory.create())
    .build();

ApiCollections apiCollections = retrofit.create(ApiCollections.class);

Call<List<User>> usersCall = apiCollections.getData();

usersCall.enqueue(new Callback<List<User>>() {
```

In above code findViewById method will find the view by given id. Constructed Retrofit object, and retrofit class is used to retrieve and upload data via REST Client from GitHub service interface and it will provide authorization to access data requested by user, with help of object we can call ApiCollection.class interface and it will retrieve data and stored in list structure.

onResponse:

```
public void onResponse(Call<List<User>> call, Response<List<User>> response) {
    if(response.isSuccessful()) {
        List<User> users = response.body();

        for(User user:users){
            String data = "";

            data += "\n\nID: " + user.getId() + "\n";
            data += "User Name: " + user.getUserName() + "\n\n";
            textView.append(data);
        }
    }
}
```

In above code onResponse function will display the details if response is successful, by using list interface we store retrieved data in structured format and for loop with condition of users will display in organized order.

onFailure:

```
@Override
public void onFailure(Call<List<User>> call, Throwable t) {
    Toast.makeText(context: MainActivity.this, text: "Data Failed", Toast.LENGTH_SHORT).show();
}
```

In above code onFailure function will throw an error message if response is not retrieved and Toast will display the message.

User.java:

```
public class User {
    private int id;

    @SerializedName("login")
    private String userName;

    public int getId() { return id; }
    public String getUserName() { return userName; }
}
```

In above screenshot user class will Serialize the name and parse from the format by

@SerializedName annotation and getId() will return id and getUserName() will return username.

ApiCollections.java:

```
public interface ApiCollections {  
    @GET("users")  
    Call<List<User>> getData();  
}
```

In above code snippet @GET annotation will request HTTP API to get userdata by response.

AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET"/>  
  
<application  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="ICP 10"  
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:supportsRtl="true"  
    android:theme="@style/Theme.ICP10">  
    <activity  
        android:name=".MainActivity"  
        android:exported="true">  
        <intent-filter>  
            <action android:name="android.intent.action.MAIN" />  
  
            <category android:name="android.intent.category.LAUNCHER" />  
        </intent-filter>  
    </activity>  
</application>
```

In above screenshot we need to specify internet permissions in the Androidmanifest.xml file in the uses-permission xml tag, because if we do not mention internet permissions, the response

data will not be coming from the url endpoint, and we will not be able to display User information.

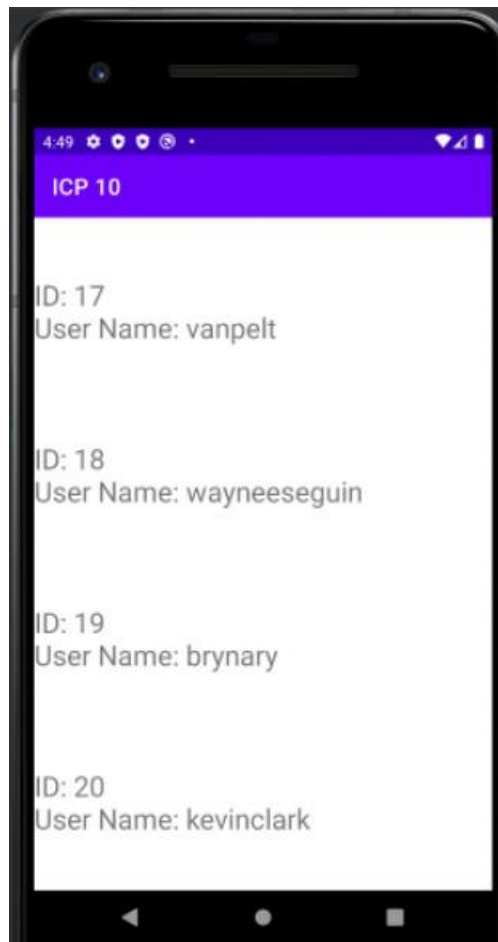
The Internet should be available in order to retrieve data from our url endpoint.

Output:

When we run the code, the display screen will show a list of github users from the endpoint, as well as information about each person such as their userid and username.



UI SCREEN



UI SCROLL DOWN SCREEN

Conclusion:

In this ICP we learned about how to develop a android application by using the Android studio, with RESTful services, Retrofit ListView and Adapter. And at last, we have designed and developed Android Application successfully. We didn't face any major issues while doing this ICP.