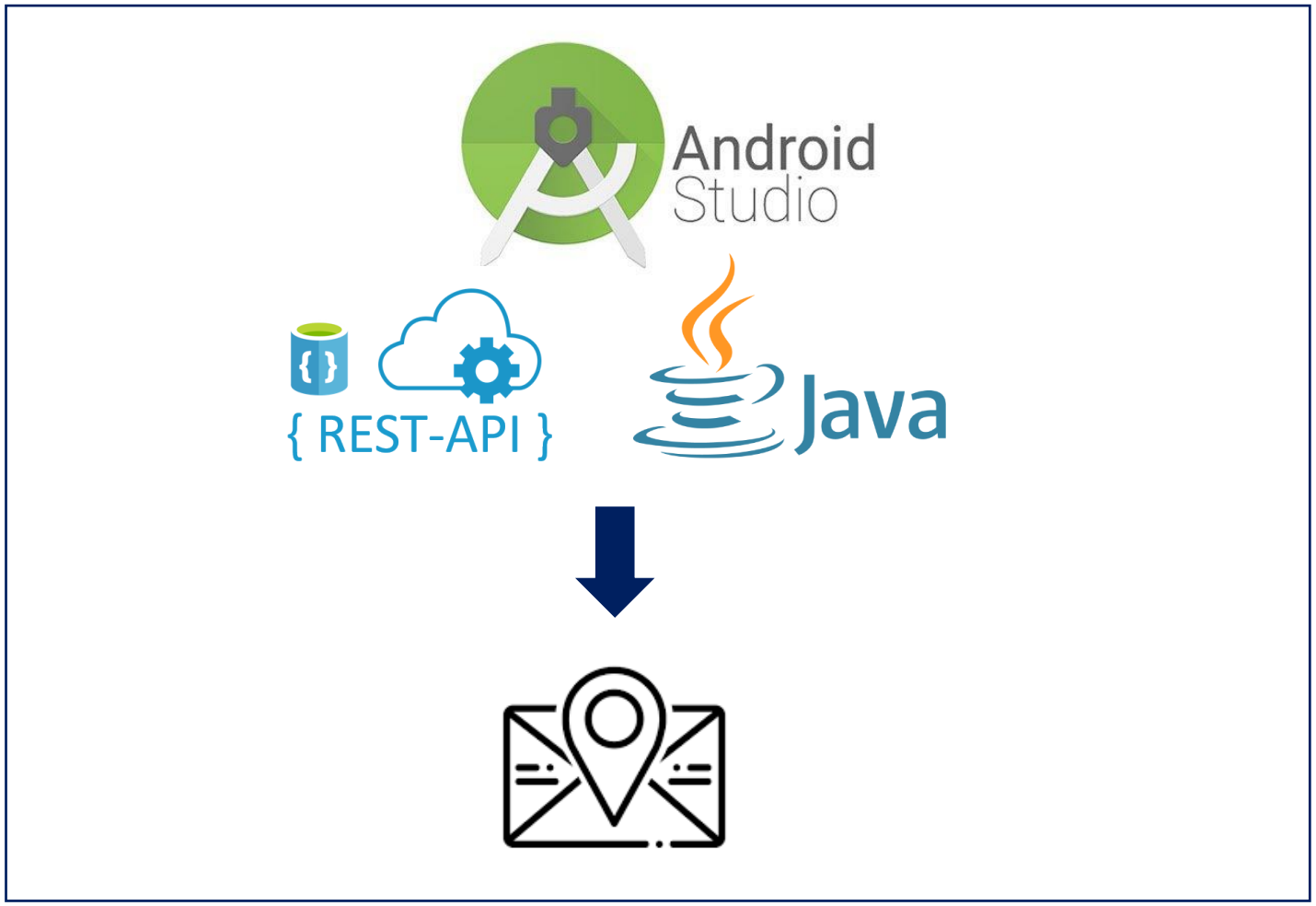


Mobile Programming: Third Year Project

"Developing Postal Information Rendering Application"

by

Khin Su Wai Phyo (23110171)

Agenda

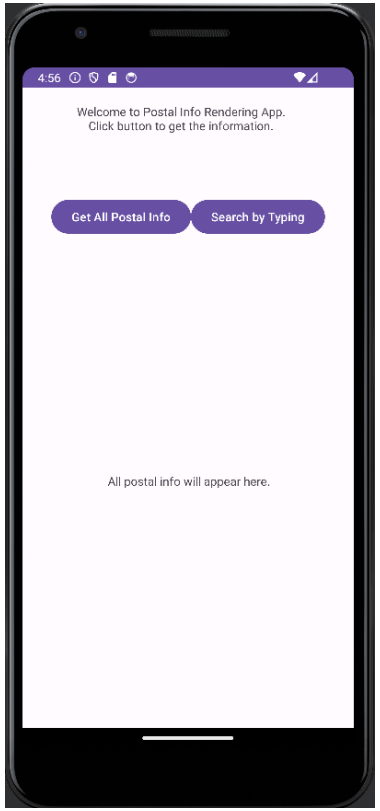
1. Introduction to the project
2. Applied API
3. Technical Workflow (Coding Process)
4. Result
5. Discussion (Challenges & Improvement Areas)
6. References

1 Introduction to the Project (1/2)

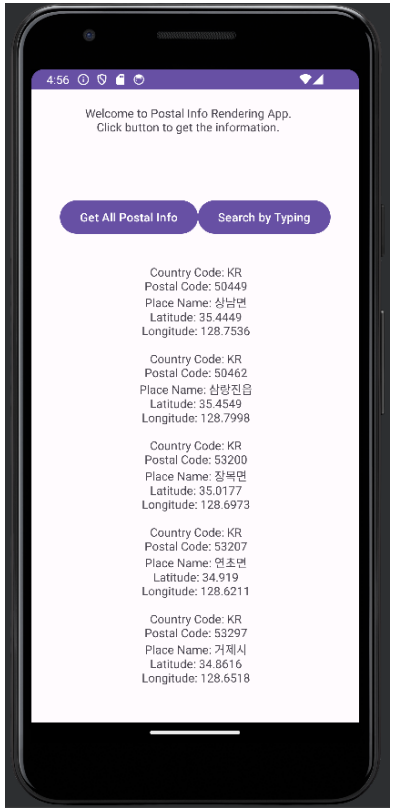
What is the Application Name?	“Postal Information Rendering”
What does this app provide?	Postal information about Postal Code, Place Name and Coordinates (Latitude and Longitude).
How is it provide?	It is provided by using REST API architecture which contain JSON object, array and string. API Link: https://data.opendatasoft.com/api/explore/v2.1/catalog/datasets/geonames-postal-code@public/records?limit=5&refine=country_code%3A%22KR%22
What is the Development Environment?	Android Studio and JAVA programming language

1 Introduction to the Project (1/2)

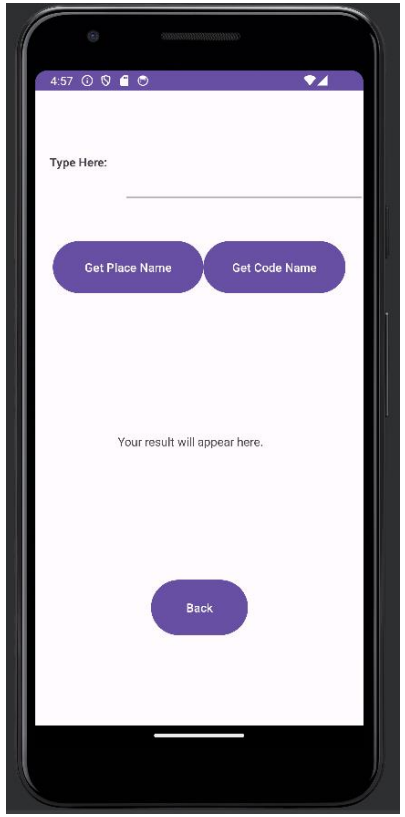
App Screen Display Series: (Screen to Screen = Changes after every click)



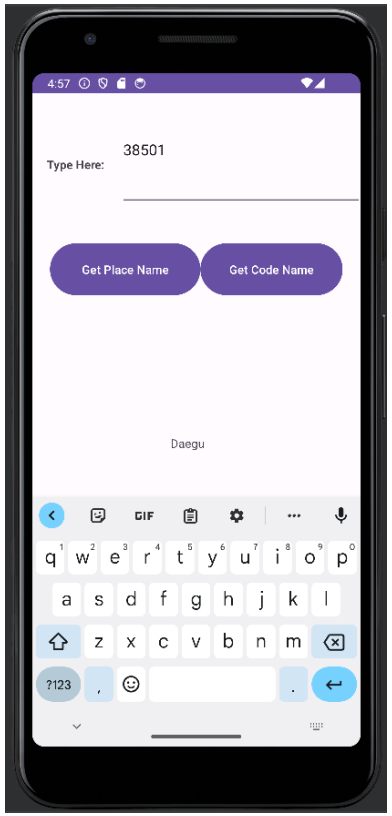
Welcome Screen



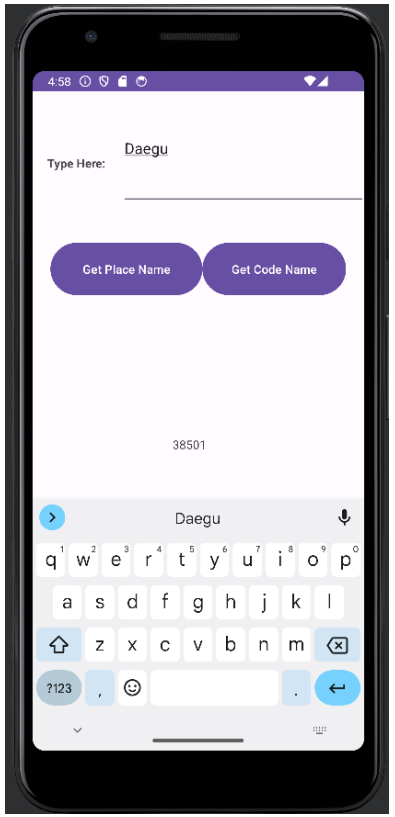
Get All
Postal Information



Get into
Search by Typing



Get Place Name by
Typing Code Name



Get Code Name by
Typing Place Name

2 Applied API

About API:

API is from Open Data Soft:
https://data.opendatasoft.com/explore/dataset/geonames-postal-code@public/api/

They provide all the postal code for all countries. In my project, I just used Korean postal code. API documentation and full API console are also available on that page.

The API link I used:

https://data.opendatasoft.com/api/explore/v2.1/catalog/datasets/geonames-postal-code@public/records?limit=20&refine=country_code%3A%22KR%22

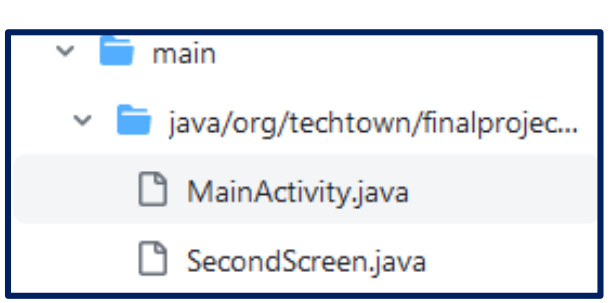
But I have limited the records from 20 to 5 which will be explained more in the next slides.

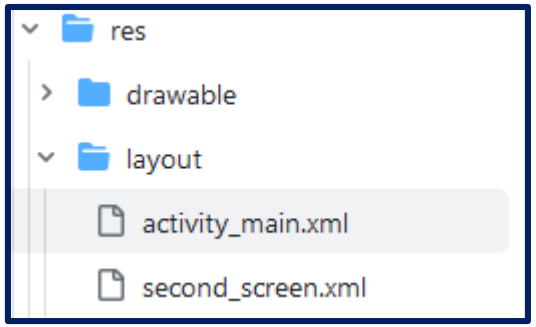
3 Technical Workflow (Coding Process) (1/6)

Overview structure:

```
android { this: BaseAppModuleExtension
    namespace = "org.techtown.finalprojectv2"
    compileSdk = 34

    defaultConfig { this: ApplicationDefaultConfig
        applicationId = "org.techtown.finalprojectv2"
        minSdk = 34
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }
```

Configuration information can be known here.





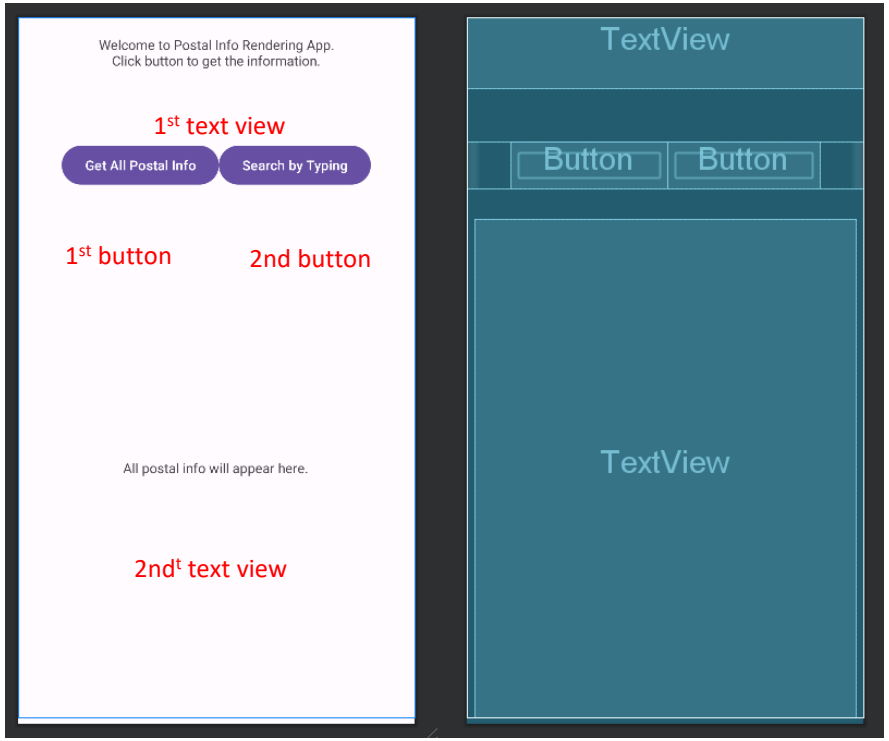
Mainly, this project is composed of two controller (java) files and two view (xml) files.

Source Code Link: https://github.com/Khinsuwaiphyo11/Postal_Info_Rendering_App

Technical Workflow (Coding Process) (2/6)

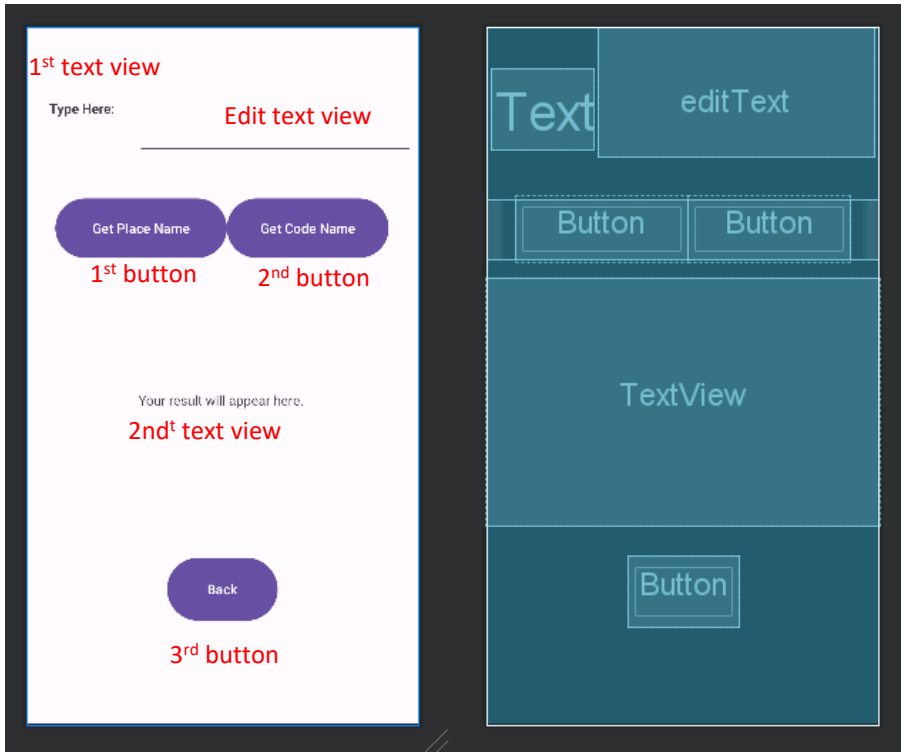
View (Layout) structure:

Layout has two xml files: one for the main screen where we can have all postal information and another one is where we can have the search by typing information.



activity_main.xml

It is composed of two buttons and two text views: First button will give the result shown in second text view. First text view is fixed. Second button will bring us to the next screen display.



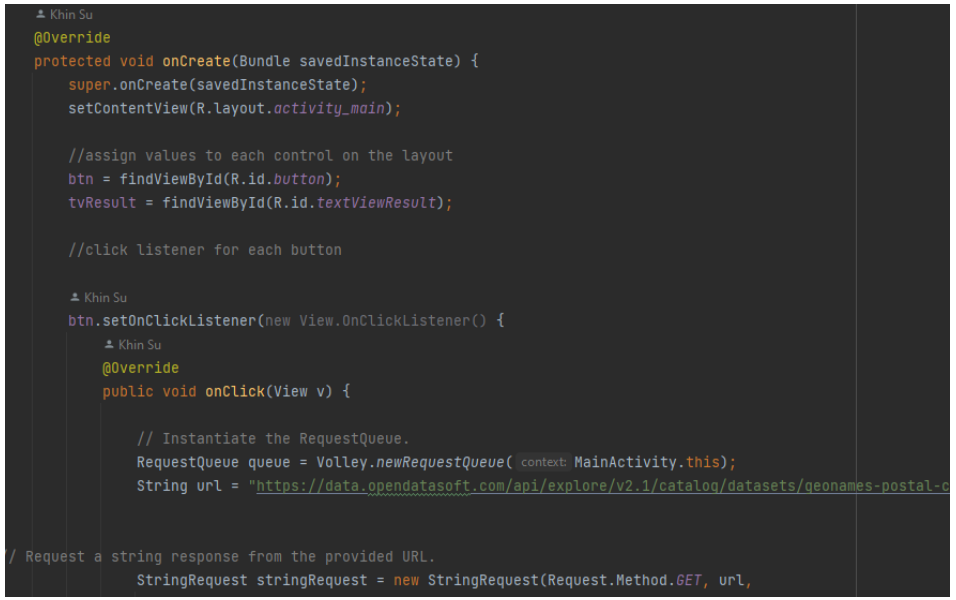
Second_screen.xml

It is composed of three buttons, two text views and one edit text view: When we type the input message in the Edit view and then click the first and second buttons, result will be shown in the second textview where first text view is fixed. Third button will bring us to the first screen display back.

3 Technical Workflow (Coding Process) (3/6)

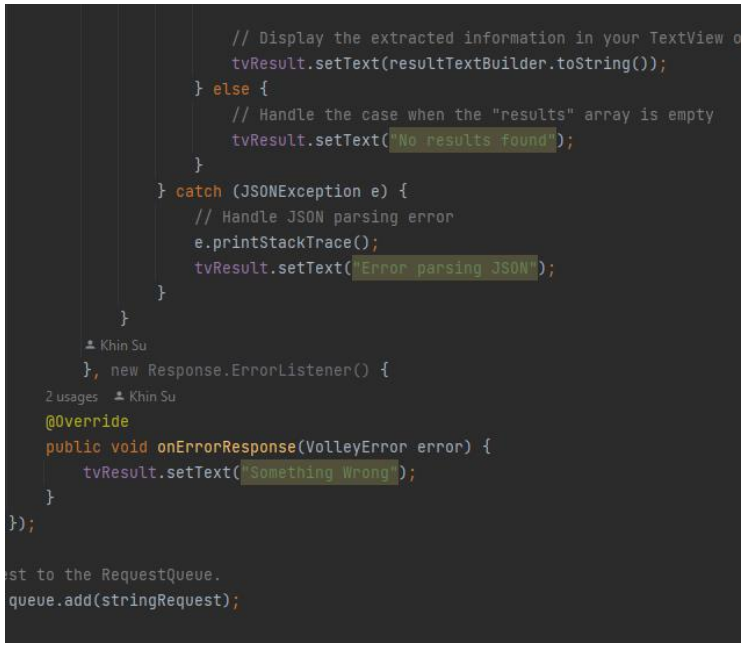
Main Controller (MainActiviyt.java) structure:

There are two controller files but here I will explain about main java file first. Two buttons are the main important features here.



```
Khin Su
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //assign values to each control on the layout
    btn = findViewById(R.id.button);
    tvResult = findViewById(R.id.textViewResult);

    //click listener for each button

    Khin Su
    btn.setOnClickListener(new View.OnClickListener() {
        Khin Su
        @Override
        public void onClick(View v) {

            // Instantiate the RequestQueue.
            RequestQueue queue = Volley.newRequestQueue( context: MainActivity.this);
            String url = "https://data.opendatasoft.com/api/explore/v2.1/catalog/datasets/geonames-postal-c

// Request a string response from the provided URL.
            StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
```

```
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        Khin Su
        new Response.Listener<String>() {
            Khin Su
            @Override
            public void onResponse(String response) {

                try {
                    // Parse the JSON response
                    JSONObject jsonResponse = new JSONObject(response);

                    // Get the "results" array
                    JSONArray resultsArray = jsonResponse.getJSONArray( name: "results");

                    // Check if the array is not empty
                    if (resultsArray.length() > 0) {
                        StringBuilder resultTextBuilder = new StringBuilder();

                        // Loop through each object in the array
                        for (int i = 0; i < resultsArray.length(); i++) {
                            // Get the current object
                            JSONObject resultObject = resultsArray.getJSONObject(i);

                            // Extract information from the object
                            String country_code = resultObject.getString( name: "country_code");
                            String postal_code = resultObject.getString( name: "postal_code");
                            String place_name = resultObject.getString( name: "place_name");
                            double latitude = resultObject.getDouble( name: "latitude");
                            double longitude = resultObject.getDouble( name: "longitude");
                            // Add more fields as needed...

                            // Append the information to the result text
                            resultTextBuilder.append("Country Code: ").append(country_code).append("\n")
                                    .append("Postal Code: ").append(postal_code).append("\n")
                                    .append("Place Name: ").append(place_name).append("\n")
                                    .append("Latitude: ").append(latitude).append("\n")
                                    .append("Longitude: ").append(longitude).append("\n")
                                    .append("\n"); // Add a line break between each record
                        }
```



```
                        // Display the extracted information in your TextView o
                        tvResult.setText(resultTextBuilder.toString());
                    } else {
                        // Handle the case when the "results" array is empty
                        tvResult.setText("No results found");
                    }
                } catch (JSONException e) {
                    // Handle JSON parsing error
                    e.printStackTrace();
                    tvResult.setText("Error parsing JSON");
                }
            }
            Khin Su
        }, new Response.ErrorListener() {
    2 usages  Khin Su
    @Override
    public void onErrorResponse(VolleyError error) {
        tvResult.setText("Something Wrong");
    }
});

st to the RequestQueue.
queue.add(stringRequest);
```

This button makes network request using the Volley library which is initiated to fetch and parse JSON data from a URL.
The result that I want to show is the string and double inside the JSON Array but that JSON Array is inside the JSON object. So, I have to change JSON object to JSON Array and then get the response from that Array.
There are three types to handle errors (/error listeners): when array is empty, when JSON has parsing errors and when Volley has error.

3

Technical Workflow (Coding Process) (4/6)

Main Controller (MainActiviyt.java) structure :

```
btn2 = findViewById(R.id.button2);
 Khin Su
btn2.setOnClickListener(new View.OnClickListener() {
     Khin Su
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), SecondScreen.class);
        startActivityForResult(intent, REQUEST_CODE_SScreen);
        intent.putExtra( name: "titleMsg",  value: "Search by Typing ");

    }
});
```

Here is simply the second button that will bring us to the next screen display.
Here I used REQUEST_CODE method which I have learned Multiple activity and Intent exercises.
It is important to add SecondScreen.class in the AndroidManifest.xml to make the control works.

3 Technical Workflow (Coding Process) (1/6)

Second Controller (SecondScreen.java) structure:

Here I will explain about second java file .

```
Khin Su
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.second_screen);

    typeText = findViewById(R.id.editText);
    resultView = findViewById(R.id.resultText);
    codeBtn = findViewById(R.id.codeButton);
    placeBtn = findViewById(R.id.placeButton);

    Khin Su
    codeBtn.setOnClickListener(new View.OnClickListener() {
        Khin Su
        @Override
        public void onClick(View v) {
            resultView.setText("38501");

        }
    });
```

Here, I wanted to show getting the query by typing code or place and getting the code or place names. But this is not actually the method that I should be using. **Database** method should be used actually to get the stored information and return the respective result of inquired query. But because of the time limitation and my lack of well understanding on that part, I skipped.

```
Khin Su
placeBtn.setOnClickListener(new View.OnClickListener() {
    Khin Su
    @Override
    public void onClick(View v) {
        resultView.setText("Daegu");

    }
});


backBtn = findViewById(R.id.backButton);
Khin Su
backBtn.setOnClickListener(new View.OnClickListener() {
    Khin Su
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(),MainActivity.class);
        startActivityForResult(intent, REQUEST_CODE_MAIN);
        intent.putExtra( name: "message",  value: "result message is OK!");
        setResult(Activity.RESULT_OK, intent);
        finish();
    }
});
```

By using intent method, back button will bring us back to the main screen. How this work is the same as of button 2 in previous main controller java file.

3 Technical Workflow (Coding Process) (4/6)

Necessary Add-ons:

In the AndroidManifest.xml file,

```
<uses-permission android:name="android.permission.INTERNET"/>
```

```
        </activity>
        <activity android:name=".SecondScreen"/>
    </application>

</manifest>
```

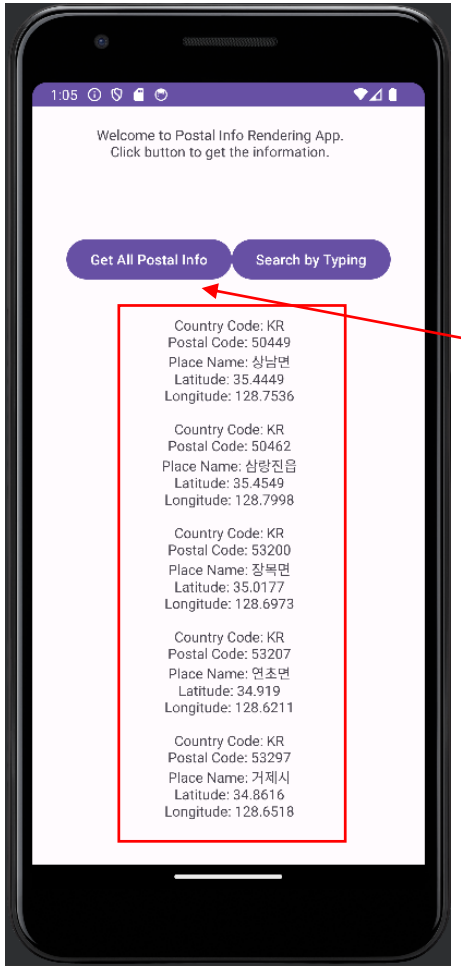
In build.gradle file, (need to be synced)

```
implementation ("com.android.volley:volley:1.2.0")
```

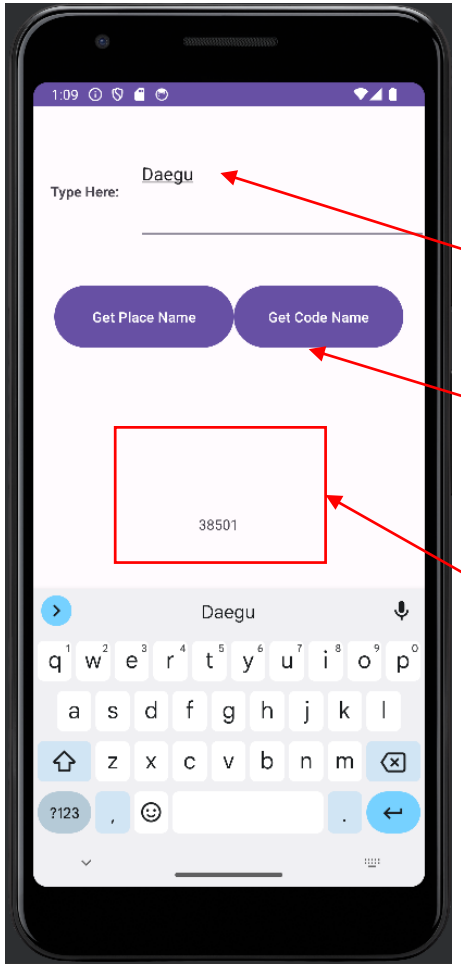
If we have Internet Security and anti-virus software, we may need to turn them off for a while to be able to start our newly developed application.

4 Result

Result show case:



This is what we obtained when we click the **Get All Postal Info** button. Here, I limited the result query to 5 and that is why it just show five blocks.
If we click **Search by Typing** button, it will bring us to the second screen.



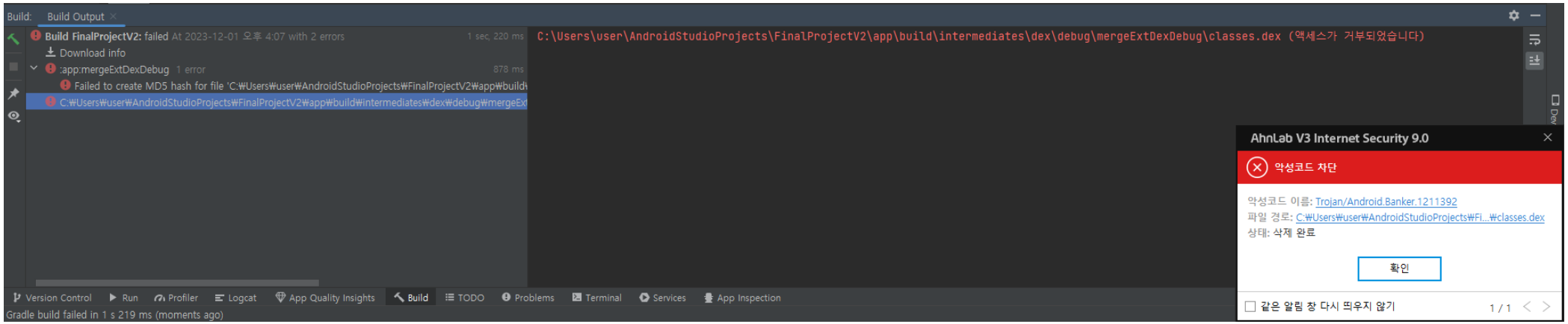
In the **Edit Text** view, we will type in the place name and when we click **Get Code Name** button, result (code) will pop up in the second **text view**. And vice versa with typing code number in and getting place name.

5

Discussion (Challenges and Improvement Areas)

Errors:

The first error I encountered with is the need to internet security permission or anti virus software issue stopping my app to install.



This is a very common error and can be overcame once we turn off the related software for a while. In my case, just one minute just enough to make my application run.

5

Discussion (Challenges and Improvement Areas)

Challenges:

- Manipulating the JSON object, array and string from the API dataset is difficult for me because I have limited knowledge about using JSON and java coding but honestly, Chat GPT helps me a lot.

- I have trouble with how I structured my layout design too. Using constraint layout is tricky if I cannot optimize its effectiveness to the fullest.

Improvement Areas:

- As I mentioned before, database technology is needed in my second screen “Search by Typing” part. If there is a database system for all postal information, we can optimized searching better.

Conclusion:

- Although there are limitations and challenges, I am satisfied with the simple prototype of the application I wanted to display regardless of having many improvement areas to fill in.
- What I like the most is finding out the alternative solutions to get the result what I want even I am stuck at one point.
- Not to mention, having practical knowledge is better than theoretical knowledge.

6

References

I have searched the application similar to my application but could not find one honestly.

I relied heavily on lecture slides, lab session material and Chat GPT.

Although, I have very different development, I first learn from this video if free code camp:
https://www.youtube.com/watch?v=xPi-z3nOcn8