**UNITY DEVELOPMENT INTERNSHIP**

**ASSIGNMENT 1*:***

**Task-1:** Sundatabase

**Code:**

Creating a complete and perfect Android app requires a comprehensive approach, including XML layouts, resources, and proper error handling. Below is an outline of the code you could use as a starting point:

**1.**Create a data model class for the API response:

public class ApiResponse {

private List<Client> clients;

private Map<String, ClientData> data;

// Getters and setters

}

**2.**Create data model classes for Client and ClientData:

public class Client {

private boolean isManager;

private int id;

private String label;

// Getters and setters

}

public class ClientData {

private String address;

private String name;

private int points;

// Getters and setters

}

**3.** Make the API request and parse the JSON:

import com.google.gson.Gson;

String apiResponse = makeApiRequest("API\_URL");

Gson gson = new Gson();

ApiResponse response = gson.fromJson(apiResponse, ApiResponse.class);

List<Client> clients = response.getClients();

4. Create the UI layout (XML files) for the list and popup:

<!-- activity\_main.xml -->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

... >

<!-- Filtering Dropdown -->

...

<!-- List View -->

<ListView

android:id="@+id/clientListView"

... />

</LinearLayout>

<!-- client\_list\_item.xml -->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

... >

<!-- Display label and points -->

...

</LinearLayout>

<!-- client\_popup.xml -->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

... >

<!-- Display name, points, and address -->

...

</LinearLayout>

**5.** Implement the ListAdapter for the list view:

public class ClientListAdapter extends BaseAdapter {

private List<Client> clients;

private LayoutInflater inflater;

// Constructor

public ClientListAdapter(Context context, List<Client> clients) {

this.clients = clients;

inflater = LayoutInflater.from(context);

}

// Override getView method

@Override

public View getView(int position, View convertView, ViewGroup parent) {

// Inflate and populate list item layout

...

// Set click listener to open popup

...

return convertView;

}

// Other required methods

...

}

**6.** Implement the popup functionality using DialogFragment:

public class ClientPopupDialog extends DialogFragment {

// Create popup dialog with name, points, and address

...

}

**7.** Add DOTween animations (you need to integrate the library):

Java:

implementation 'com.daimajia.androidanimations:library:2.4@aar'

1.import the necessary packages in your Java code:

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import androidx.annotation.Nullable;

import androidx.fragment.app.DialogFragment;

public class ClientPopupDialog extends DialogFragment {

@Nullable

@Override

public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.client\_popup, container, false);

// Apply animations to UI elements using ViewPropertyAnimator

animateUIElements(view);

return view;

}

private void animateUIElements(View view) {

// Animate nameTextView with a slide-in animation

view.findViewById(R.id.nameTextView).animate()

.translationXBy(-200) // Initial off-screen position

.translationX(0) // Final on-screen position

.setDuration(600)

.start();

// Apply similar animations to other UI elements

}

}