

**Lab 06**

CTE306 – Mobile Application Development

Date – 5th September 2022

Divyash Chhetri

02200174

BE 3 IT

Module Tutor: Mr. Pema Galey

Department of Information Technology

**College of Science and Technology**

**Aim**

Perform the task on following topics:

1. AsyncTask
2. AsyncTaskLoader
3. InternetConnection

**Theory**

The purpose of AsyncTask is to make it simple and appropriate to use the UI thread. The most frequent use case, however, is UI integration, which leads to Context leaks, missed callbacks, or crashes when settings are changed. Additionally, it behaves differently depending on the platform version, swallows exceptions from doInBackground, and offers no benefit over using Executors directly.

AsyncTask is not a general-purpose threading system; rather, it is intended to be a support class for Thread and Handler. AsyncTasks are best utilized for quick activities (a few seconds at the most.) It is strongly advised to use the various APIs offered by the java.util.concurrent package, such as Executor, ThreadPoolExecutor, and FutureTask, if threads are to be kept running for extended periods of time.

**Adaptive Layout**

**Program Code**

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="16dp"

tools:context=".MainActivity">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/instructions"

android:text="@string/instructions"

android:textAppearance="@style/TextAppearance.AppCompat.Title"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

<EditText

android:id="@+id/bookInput"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:hint="@string/input\_hint"

android:inputType="text"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/instructions" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/searchButton"

android:layout\_marginTop="8dp"

android:text="@string/button\_text"

android:onClick="searchBooks"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/bookInput"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/titleText"

android:layout\_marginTop="16dp"

android:textAppearance="@style/TextAppearance.AppCompat.Headline"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/searchButton"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/authorText"

android:layout\_marginTop="8dp"

android:textAppearance="@style/TextAppearance.AppCompat.Headline"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/titleText"/>

</androidx.constraintlayout.widget.ConstraintLayout>

*Strings.xml*

<resources>

<string name="app\_name">AsyncTask</string>

<string name="instructions">Enter a book name to find out who wrote the book. </string>

<string name="button\_text">Search Books</string>

<string name="input\_hint">Book Title</string>

<string name="no\_results">"No Results Found"</string>

<string name="loading">Loading...</string>

<string name="no\_search\_term">Please enter a search term</string>

<string name="no\_network">Please check your network connection and try again.</string>

</resources>

*MainActivity.java*

package com.example.asynctask;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.net.ConnectivityManager;

import android.net.NetworkInfo;

import android.os.Bundle;

import android.view.View;

import android.view.inputmethod.InputMethodManager;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

private EditText mBookInput;

private TextView mTitleText;

private TextView mAuthorText;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mBookInput = (EditText)findViewById(R.id.bookInput);

mTitleText = (TextView)findViewById(R.id.titleText);

mAuthorText = (TextView)findViewById(R.id.authorText);

}

public void searchBooks(View view) {

String queryString = mBookInput.getText().toString();

InputMethodManager inputManager = (InputMethodManager)

getSystemService(Context.INPUT\_METHOD\_SERVICE);

if (inputManager != null ) {

inputManager.hideSoftInputFromWindow(view.getWindowToken(),

InputMethodManager.HIDE\_NOT\_ALWAYS);

}

ConnectivityManager connMgr = (ConnectivityManager)

getSystemService(Context.CONNECTIVITY\_SERVICE);

NetworkInfo networkInfo = null;

if (connMgr != null) {

networkInfo = connMgr.getActiveNetworkInfo();

}

if (networkInfo != null && networkInfo.isConnected()

&& queryString.length() != 0) {

new FetchBook(mTitleText, mAuthorText).execute(queryString);

mAuthorText.setText("");

mTitleText.setText(R.string.loading);

} else {

if (queryString.length() == 0) {

mAuthorText.setText("");

mTitleText.setText(R.string.no\_search\_term);

} else {

mAuthorText.setText("");

mTitleText.setText(R.string.no\_network);

}

}

}

}

*FetchBook.java*

package com.example.asynctask;

import android.os.AsyncTask;

import android.widget.TextView;

import org.json.JSONArray;

import org.json.JSONObject;

import java.lang.ref.WeakReference;

public class FetchBook extends AsyncTask<String, Void, String> {

private WeakReference<TextView> mTitleText;

private WeakReference<TextView> mAuthorText;

public FetchBook(TextView titleText, TextView authorText) {

this.mTitleText = new WeakReference<>(titleText);

this.mAuthorText = new WeakReference<>(authorText);

}

@Override

protected String doInBackground(String... strings) {

return NetworkUtils.getBookInfo(strings[0]);

}

@Override

protected void onPostExecute(String s) {

super.onPostExecute(s);

try {

JSONObject jsonObject = new JSONObject(s);

JSONArray itemsArray = jsonObject.getJSONArray("items");

int i = 0;

String title = null;

String authors = null;

while (i < itemsArray.length() &&

(authors == null && title == null)) {

JSONObject book = itemsArray.getJSONObject(i);

JSONObject volumeInfo = book.getJSONObject("volumeInfo");

try {

title = volumeInfo.getString("title");

authors = volumeInfo.getString("authors");

} catch (Exception e) {

e.printStackTrace();

}

i++;

}

if (title != null && authors != null) {

mTitleText.get().setText(title);

mAuthorText.get().setText(authors);

} else {

mTitleText.get().setText(R.string.no\_results);

mAuthorText.get().setText("");

}

} catch (Exception e) {

mTitleText.get().setText(R.string.no\_results);

mAuthorText.get().setText("");

}

}

}

*NetworkUtils.java*

package com.example.asynctask;

import android.net.Uri;

import android.util.Log;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class NetworkUtils {

private static final String LOG\_TAG = NetworkUtils.class.getSimpleName();

private static final String BOOK\_BASE\_URL = "https://www.googleapis.com/books/v1/volumes?";

private static final String QUERY\_PARAM = "q";

private static final String MAX\_RESULTS = "maxResults";

private static final String PRINT\_TYPE = "printType";

static String getBookInfo(String queryString){

HttpURLConnection urlConnection = null;

BufferedReader reader = null;

String bookJSONString = null;

try {

Uri builtURI = Uri.parse(BOOK\_BASE\_URL).buildUpon()

.appendQueryParameter(QUERY\_PARAM, queryString)

.appendQueryParameter(MAX\_RESULTS, "10")

.appendQueryParameter(PRINT\_TYPE, "books")

.build();

URL requestURL = new URL(builtURI.toString());

urlConnection = (HttpURLConnection) requestURL.openConnection();

urlConnection.setRequestMethod("GET");

urlConnection.connect();

InputStream inputStream = urlConnection.getInputStream();

reader = new BufferedReader(new InputStreamReader(inputStream));

StringBuilder builder = new StringBuilder();

String line;

while ((line = reader.readLine()) != null) {

builder.append(line);

builder.append("\n");

}

if (builder.length() == 0) {

return null;

}

bookJSONString = builder.toString();

} catch (IOException e) {

e.printStackTrace();

} finally {

if (urlConnection != null) {

urlConnection.disconnect();

}

if (reader != null) {

try {

reader.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

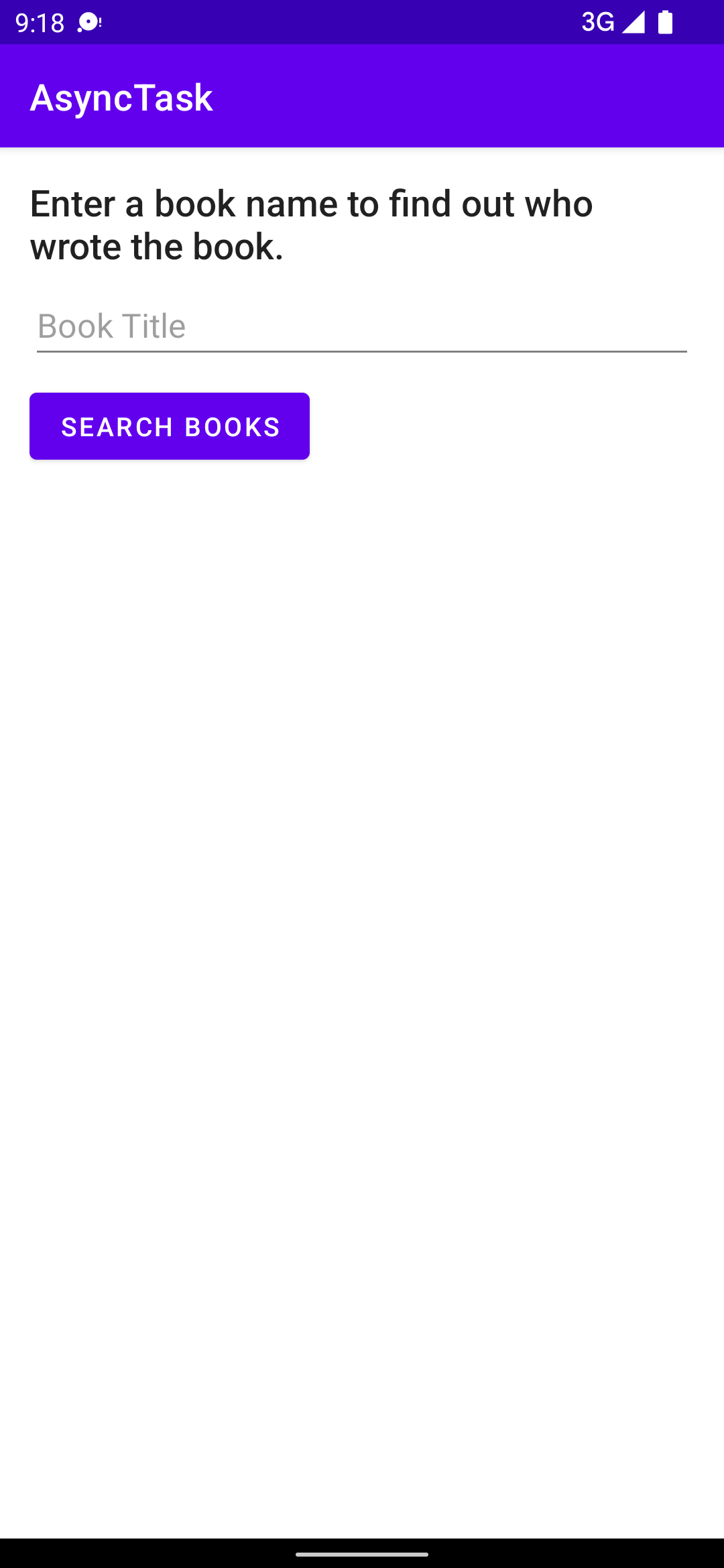
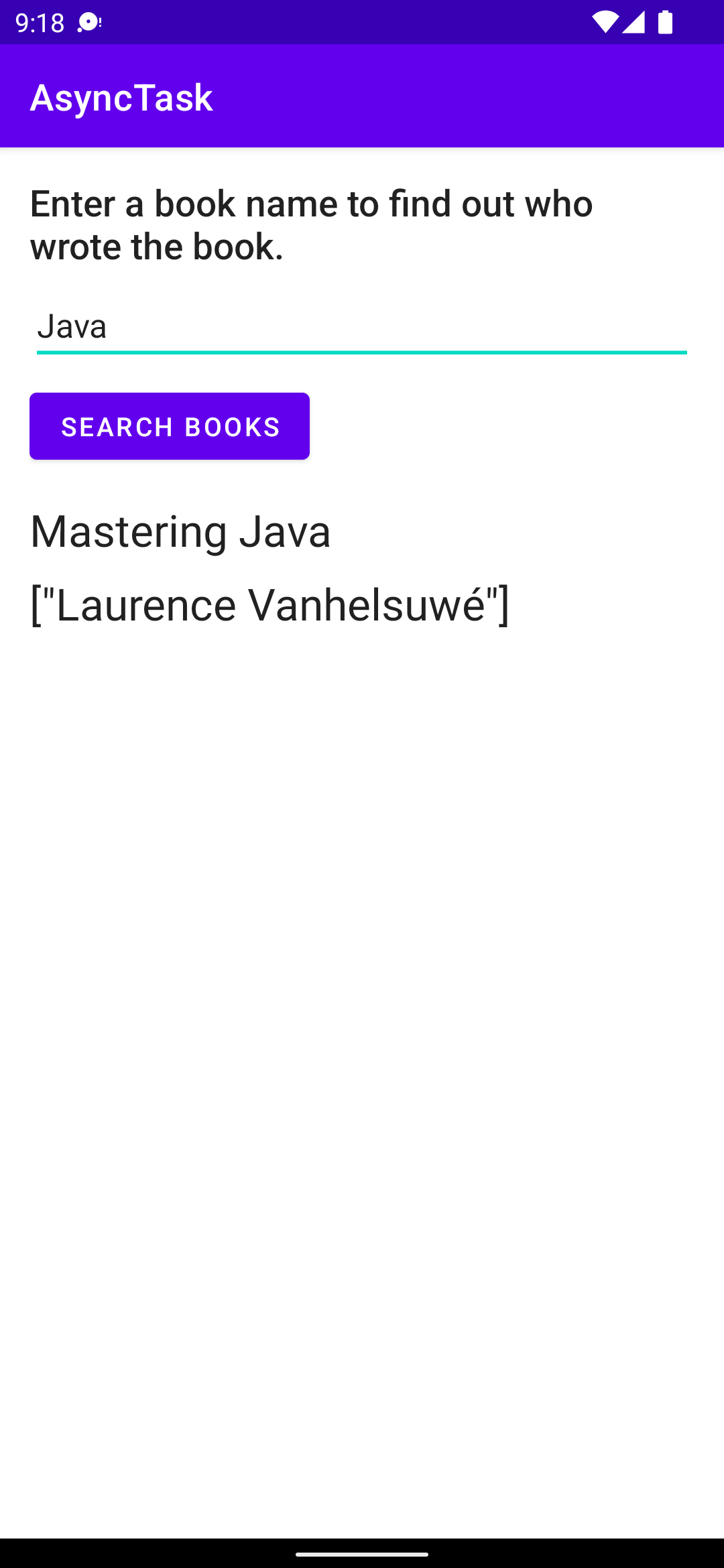
Log.d(LOG\_TAG, bookJSONString);

return bookJSONString;

}

}

**Output**

* *

**MockTest**

**Program Codes**

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

xmlns:app="http://schemas.android.com/apk/res-auto"

tools:context=".MainActivity"

android:background="#e8f4f8"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:textSize="40sp"

android:text="Weather App"

android:textColor="@color/black"

android:layout\_marginTop="20sp"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:layout\_marginTop="40sp">

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/etCity"

android:layout\_marginBottom="30dp"

android:textSize="20sp"

android:hint="Enter City Name....."

android:inputType="textPersonName"/>

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/btnGet"

android:layout\_marginBottom="10dp"

android:background="#0070c7"

android:textColor="@android:color/white"

android:onClick="getWeatherDetails"

android:text="Search"

android:textSize="16sp"

app:backgroundTint="@null" />

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="150dp">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/tvResult"

android:textSize="15sp"

android:textColor="@color/black"

android:layout\_marginTop="10sp"/>

</ScrollView>

</LinearLayout>

</LinearLayout>

*MainActivity.java*

package com.example.mocktest;

import android.graphics.Color;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.text.DecimalFormat;

public class MainActivity extends AppCompatActivity {

EditText etCity;

TextView tvResult;

private final String url = "https://api.openweathermap.org/data/2.5/weather";

private final String appid = "e53301e27efa0b66d05045d91b2742d3";

DecimalFormat df = new DecimalFormat("#.##");

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

getSupportActionBar().hide();

etCity = findViewById(R.id.etCity);

tvResult = findViewById(R.id.tvResult);

}

public void getWeatherDetails(View view) {

String tempUrl = "";

String city = etCity.getText().toString().trim();

if (city.equals("")) {

tvResult.setText("City field can not be empty!");

} else {

tempUrl = url + "?q=" + city + "&appid=" + appid;

StringRequest stringRequest = new StringRequest(Request.Method.POST, tempUrl, new Response.Listener<String>() {

@Override

public void onResponse(String response) {

String output = "";

try {

JSONObject jsonResponse = new JSONObject(response);

JSONArray jsonArray = jsonResponse.getJSONArray("weather");

JSONObject jsonObjectWeather = jsonArray.getJSONObject(0);

String description = jsonObjectWeather.getString("description");

JSONObject jsonObjectMain = jsonResponse.getJSONObject("main");

double temp = jsonObjectMain.getDouble("temp") - 273.15;

double feelsLike = jsonObjectMain.getDouble("feels\_like") - 273.15;

float pressure = jsonObjectMain.getInt("pressure");

int humidity = jsonObjectMain.getInt("humidity");

JSONObject jsonObjectWind = jsonResponse.getJSONObject("wind");

String wind = jsonObjectWind.getString("speed");

JSONObject jsonObjectClouds = jsonResponse.getJSONObject("clouds");

String clouds = jsonObjectClouds.getString("all");

JSONObject jsonObjectSys = jsonResponse.getJSONObject("sys");

String countryName = jsonObjectSys.getString("country");

String cityName = jsonResponse.getString("name");

tvResult.setTextColor(Color.rgb(68, 134, 199));

output += "Current weather of " + cityName + " (" + countryName + ")"

+ "\n Temp: " + df.format(temp) + " °C"

+ "\n Feels Like: " + df.format(feelsLike) + " °C"

+ "\n Humidity: " + humidity + "%"

+ "\n Description: " + description

+ "\n Wind Speed: " + wind + "m/s (meters per second)"

+ "\n Cloudiness: " + clouds + "%"

+ "\n Pressure: " + pressure + " hPa";

tvResult.setText(output);

} catch (JSONException e) {

e.printStackTrace();

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

Toast.makeText(getApplicationContext(), error.toString().trim(), Toast.LENGTH\_SHORT).show();

}

});

RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());

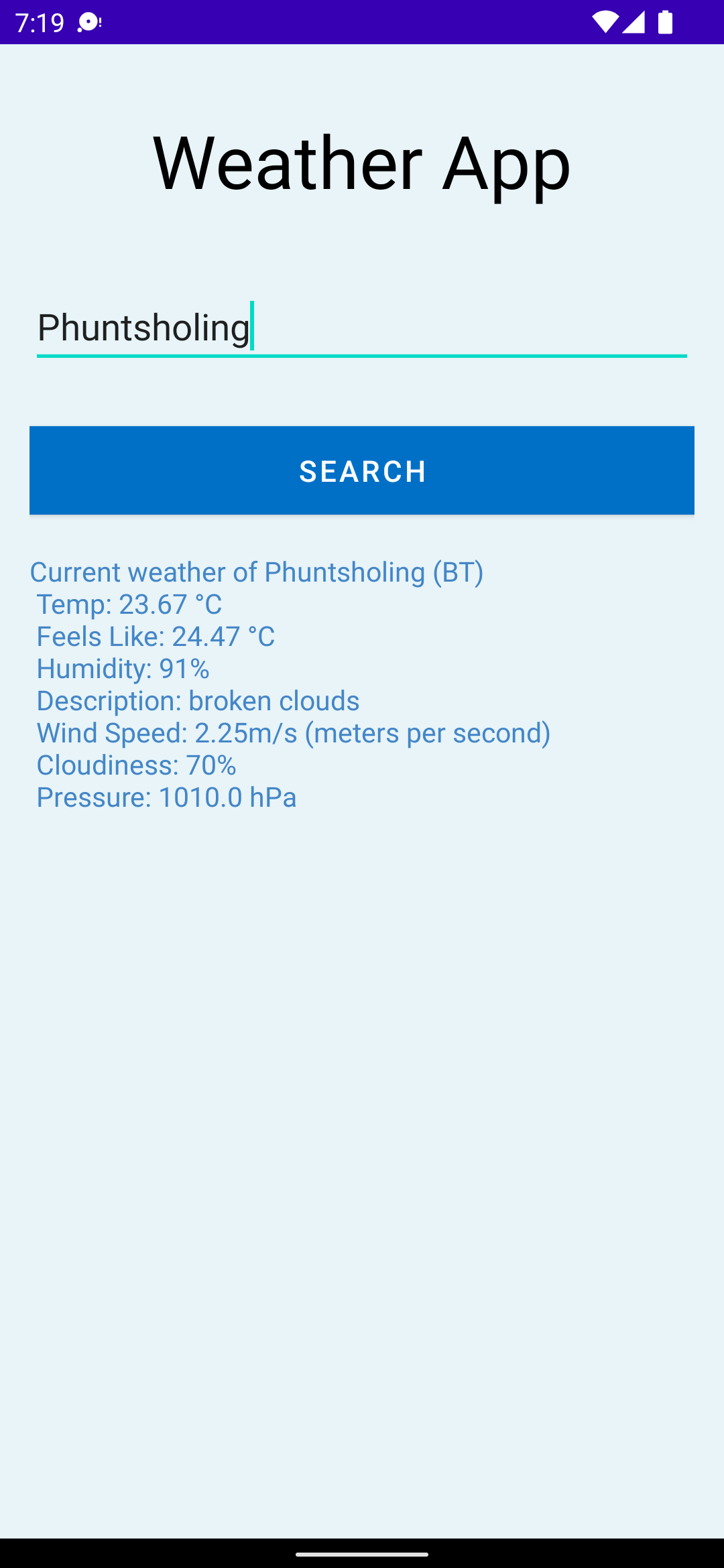
requestQueue.add(stringRequest);

}

}

}

**Output**

**

**Conclusion**

By completing this lab I learnt about AsyncTask, AsyncTask Loader, NetworkConnectivity and various other background operation concepts. Android codelabs were very helpful as steps to implement AsyncTasks were explained in detail and all steps were given to implement Who Wrote It? App.

# References

Android. (n.d.). *Android Studio Docs*. Retrieved 08 07, 2022, from Developers Android: https://developer.android.com/docs

Kakal, S. (2021, Dec 14). *A brief introduction to AsyncTask in Android with visualize examples*. Retrieved Sept 08, 2022, from DevGenius Blogs: https://blog.devgenius.io/a-brief-introduction-to-asynctask-in-android-with-visualize-examples-60a778de8a77?gi=299164c140ba

Googler. (2022, July 25). *Android fundamentals 07.2:AsyncTask and AsyncTaskLoader*. Retrieved Sept 08, 2022, from Android Codelabs: https://developer.android.com/codelabs/android-training-asynctask-asynctaskloader#0