

Grace Kopp

The search term in this app is used as a query to the `map.nd.edu/placemarks.json` URL. Without the query, this URL contains data on every single building in Notre Dame. With it, we can narrow our results to information on one particular building. For example, if the user types in “Fitzpatrick,” Fitzpatrick is used as the query. This changes the URL to `https://map.nd.edu/placemarks.json?q=fitzpatrick`. This is now one page of data all about Fitzpatrick which is parsed in the app and conveyed in a meaningful way. If no query matches the input, then no information is shown on the app.

MainActivity.java

```
package com.example.gracekopp.graceviewer;

import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;
import android.widget.TextView;

import com.example.gracekopp.graceviewer.utilities.NetworkUtils;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.net.URL;

public class MainActivity extends AppCompatActivity {

    private EditText mSearchBoxEditText;
    private TextView mDisplayTextView;
    private TextView mSearchResultsTextView;

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

        mSearchBoxEditText = (EditText) findViewById(R.id.et_search_box);
        mDisplayTextView = (TextView) findViewById(R.id.tv_url_display);
        mSearchResultsTextView = (TextView) findViewById(R.id.tv_search_results_json);
    } // end of onCreate

    private void makeSearchQuery(){
        String searchQuery = mSearchBoxEditText.getText().toString();
        mDisplayTextView.setText("Information about " + searchQuery + " :");
        mSearchResultsTextView.setText("");
        // actually perform search and get results
        new FetchNetworkData().execute(searchQuery);
    } // end of function make search

    //class to perform networking duties
    public class FetchNetworkData extends AsyncTask<String, Void, String>{

        @Override
        protected String doInBackground(String... params) {
            if (params.length == 0) return null;
            //build query url
```

```

String searchQuery = params[0];
URL url = NetworkUtils.buildUrl(searchQuery);
//get data from the query url
String responseString = null;
try {
    responseString = NetworkUtils.getResponseFromHttpUrl(url);
} catch (Exception e){
    e.printStackTrace();
}
return responseString;

} // end of function doInBackground

@Override
protected void onPostExecute(String responseData) {
    //super.onPostExecute(s);
    // process response data - which is a json formatted string
    String[] building_info = processJSON(responseData);
    // grab info from ND map json and append to search results
    for (String info : building_info){
        mSearchResultsTextView.append(info + "\n\n");
    }
} //end of function

public String[] processJSON(String responseJSONData){
    String[] building_facts = new String[4];
    try {
        // parse the JSON string to get the desired information

        // mapBuilding is a reference to outside array of 1 that contains info on the building
        JSONArray mapBuilding = new JSONArray(responseJSONData);

        // there is a dictionary inside of the mapBuilding array with the ID 0
        JSONObject facts = mapBuilding.getJSONObject("0");

        // once inside the dictionary, the key "name" corresponds to a string with the buildings full name
        String name = facts.getString("name");

        // another dictionary element contains a string with a description of the building
        String description = facts.getString("description");

        // visual_center is a dictionary within that dictionary that contains coordinate information
        JSONObject coordinates = facts.getJSONObject("visual_center");

        // once inside this sub dictionary, the keys "lat" and "lon" have strings with the buildings gps location
        String lat = coordinates.getString("lat");
        String lon = coordinates.getString("lon");

        // Concatenate all of these strings with labels and load into a return array
        building_facts[0] = "Name: " + name;
        building_facts[1] = "Description: " + description;
        building_facts[2] = "Latitude: " + lat;
        building_facts[3] = "Longitude: " + lon;

    } catch (JSONException e){
        e.printStackTrace();
    }
    return building_facts;
} // end function processJSON

} // end of clas FetchNetworkData

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //return super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
} // end function

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int itemThatWasClicked = item.getItemId();
    if (itemThatWasClicked == R.id.action_search){
        makeSearchQuery();
        return true;
    }
    return super.onOptionsItemSelected(item);
} // end function
}

```

NetworkUtils.java

```

package com.example.gracekopp.graceviewer.utilities;

import android.net.Uri;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Scanner;

/**
 * Created by gracekopp on 10/2/17.
 */

public class NetworkUtils {
    private static final String ND_URL = "https://map.nd.edu/placemarks";
    private static final String endFormat = ".json?q=";

    public static URL buildUrl(String userQuery){
        Uri builtUri = Uri.parse(ND_URL).buildUpon()
            .appendPath(endFormat)
            .appendPath(userQuery)
            .build();

        URL url = null;
        try {
            url = new URL(builtUri.toString());
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        return url;
    } // end of function buildUrl

    public static String getResponseFromHttpUrl(URL url) throws IOException {
        HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

        try {
            InputStream in = urlConnection.getInputStream();

```

```

Scanner scanner = new Scanner(in);
scanner.useDelimiter("\\A");

boolean hasInput = scanner.hasNext();
if (hasInput) return scanner.next();
else return null;
}finally {
    urlConnection.disconnect();
}
} // end of function
}

```

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"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.gracekopp.graceviewer.MainActivity">

    <EditText
        android:id="@+id/et_search_box"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter a query, then click Search"
        android:textSize="22sp" />

    <TextView
        android:id="@+id/tv_url_display"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textSize="22sp" />

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp">

        <TextView
            android:id="@+id/tv_search_results_json"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Make a search!"
            android:textSize="18sp" />

    </ScrollView>
</LinearLayout>

```

strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.gracekopp.graceviewer">

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

```

```

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>

```

STRINGS.XML

```

<resources>
    <string name="app_name">GraceViewer</string>
    <string name="search">Search</string>
</resources>

```

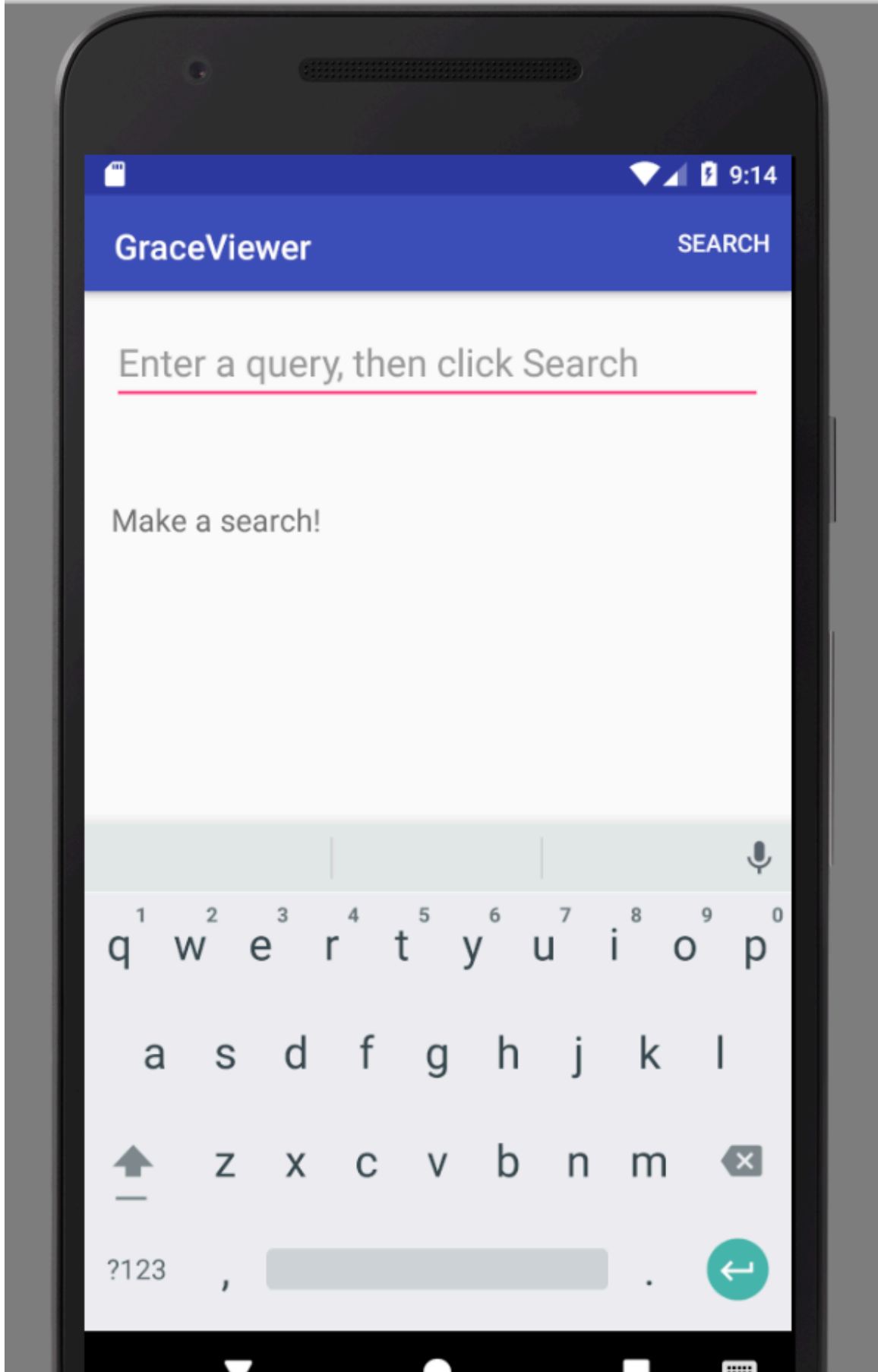
main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:title="@string/search"
        android:id="@+id/action_search"
        android:orderInCategory="1"
        app:showAsAction="ifRoom"
    />
</menu>

```

Android Emulator - Nexus_5X_API_...





9:16

GraceViewer

SEARCH

flaherty

Information about flaherty :

Name: Flaherty Hall

Description: <p>On the northeast side of campus, a new women's residence hall is under construction between Knott Hall and Pasquerilla East Hall and the just-begun East Campus Research Complex. Flaherty Hall will be a home to 225 students.</p>

Flaubert

flaherty

flayer to

1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p

a s d f g h j k l



z

x

c

v

b

n

m



?123

,

.



GraceViewer

SEARCH

fitzpatrick

Information about fitzpatrick :

Name: Fitzpatrick Hall of Engineering

Description: <p>Opened in 1979, it is the primary location for engineering teaching, research and computing.</p>

Latitude: 41.6994105

Longitude: -86.237174

Fitzpatrick | fitzpatrick | Fitzpatrick's

1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p

a s d f g h j k l

↑ z x c v b n m ✕

?123

,

.

