

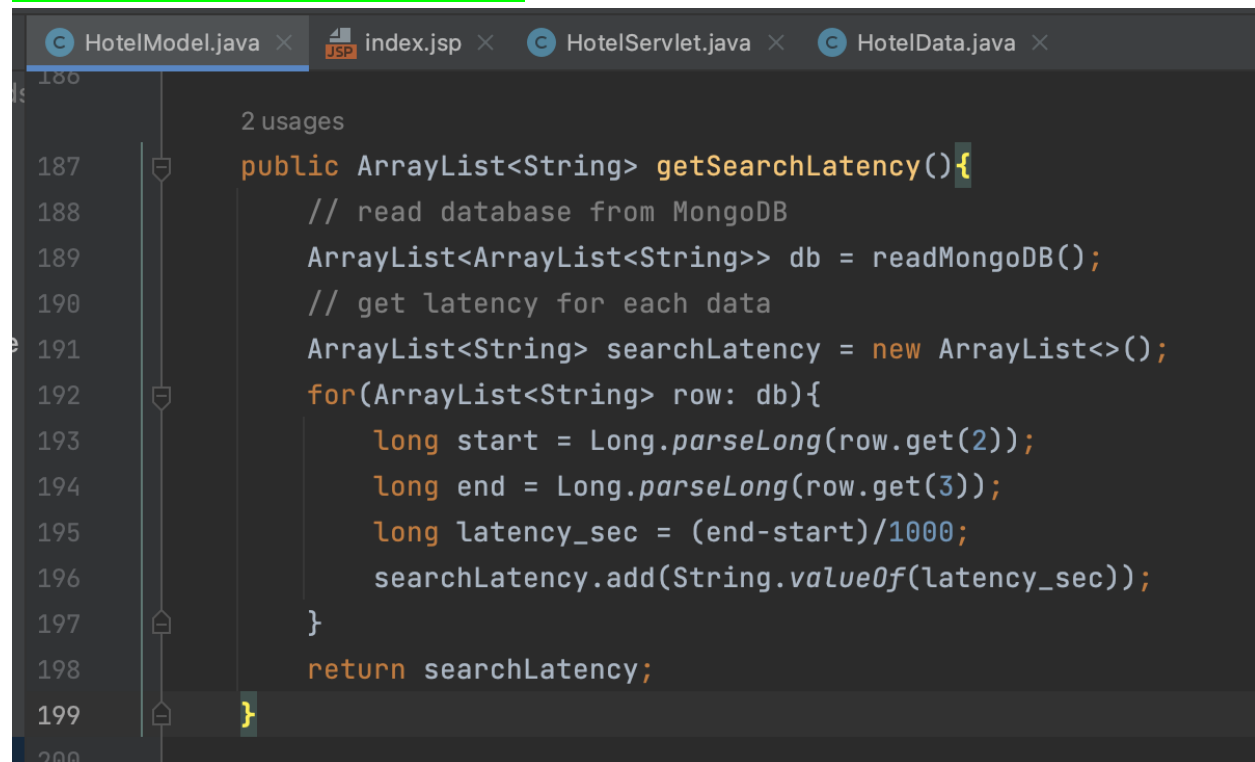
Project 4 Task 2

* Name: Yi Jyun (Eva) Chen, Yun Yung Wang

* AndrewID: yijyunc, yunyungw

* Email: yijyunc@andrew.cmu.edu, yunyungw@andrew.cmu.edu

Dashboard - Operations analytics*3



The screenshot shows an IDE with four tabs: HotelModel.java, index.jsp, HotelServlet.java, and HotelData.java. The HotelModel.java tab is active, showing the getSearchLatency method. The method is public and returns an ArrayList of Strings. It reads data from MongoDB, calculates the search latency for each row, and returns the results. The code is as follows:

```
186  
187 public ArrayList<String> getSearchLatency(){  
188     // read database from MongoDB  
189     ArrayList<ArrayList<String>> db = readMongoDB();  
190     // get latency for each data  
191     ArrayList<String> searchLatency = new ArrayList<>();  
192     for(ArrayList<String> row: db){  
193         long start = Long.parseLong(row.get(2));  
194         long end = Long.parseLong(row.get(3));  
195         long latency_sec = (end-start)/1000;  
196         searchLatency.add(String.valueOf(latency_sec));  
197     }  
198     return searchLatency;  
199 }  
200
```

```
HotelModel.java x index.jsp x HotelServlet.java x HotelData.java x
1 usage
201 public String getAverageLatency(){
202     ArrayList<String> searchLatency = getSearchLatency();
203     long total = 0;
204     for(String latency: searchLatency){
205         long latency_long = Long.parseLong(latency);
206         total += latency_long;
207     }
208     long avg = total/searchLatency.size();
209     return String.valueOf(avg);
210 }
211
```

```
HotelModel.java x index.jsp x HotelServlet.java x HotelData.java x
1 usage
213 public Map<String, Integer> getTopSearch(){
214     // read database from MongoDB
215     ArrayList<ArrayList<String>> db = readMongoDB();
216     Map<String, Integer> searchCityCount = new HashMap<>();
217     for(ArrayList<String> row: db){
218         String city = row.get(0);
219         // https://mkyong.com/java/java-how-to-update-the-value-of-a-key-in-hashmap/
220         searchCityCount.put(city, searchCityCount.containsKey(city) ? searchCityCount.get(city) + 1 : 1);
221     }
222
223     // Sorting reference to https://mkyong.com/java/how-to-sort-a-map-in-java/
224     // 1. Convert Map to List of Map
225     List<Map.Entry<String, Integer>> list =
226         new LinkedList<Map.Entry<String, Integer>>(searchCityCount.entrySet());
227
228     // 2. Sort list with Collections.sort(), provide a custom Comparator
229     Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
230         public int compare(Map.Entry<String, Integer> o1,
231             Map.Entry<String, Integer> o2) {
232             return (o2.getValue()).compareTo(o1.getValue());
233         }
234     });
235
236     // 3. Loop the sorted list and put it into a new insertion order Map LinkedHashMap
237     Map<String, Integer> TopCity = new LinkedHashMap<String, Integer>();
238     for (Map.Entry<String, Integer> entry : list) {
239         TopCity.put(entry.getKey(), entry.getValue());
240     }
241 }
```

Dashboard - Logs & Log useful information

[illegible]

[illegible][illegible][illegible]

```

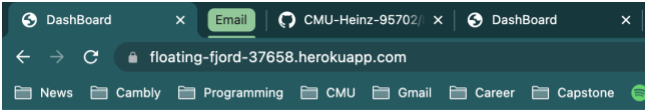
1  HotelModel.java ×  index.jsp ×  HotelServlet.java ×  HotelData.java ×
2
3  if(request.getRequestURI().contains("getHotel")){
4      String res = null;
5
6      // get the search parameter if it exists
7      String search = request.getParameter( name: "searchWord");
8      System.out.println(search);
9
10     res = hm.callAPI(search);
11     System.out.println(res);
12     //response.setContentType("text/html");
13
14     // Hello
15     PrintWriter out = response.getWriter();
16     response.setContentType("application/json");
17     response.setCharacterEncoding("UTF-8");
18     out.println(res);
19     out.flush();
20 }
21 // else get to the index page
22 else{
23     ArrayList<ArrayList<String>> readMongoDB = hm.readMongoDB();
24     ArrayList<String> searchLatency = hm.getSearchLatency();
25     String avg_latency = hm.getAverageLatency();
26     Map<String, Integer> topCity = hm.getTopSearch();
27
28     request.setAttribute( name: "database", readMongoDB);
29     request.setAttribute( name: "searchLatency", searchLatency);
30     request.setAttribute( name: "avg_latency", avg_latency);
31     request.setAttribute( name: "topCity", topCity);
32
33     nextView = "index.jsp";
34     // Transfer control over the correct "view"
35     RequestDispatcher view = request.getRequestDispatcher(nextView);
36     view.forward(request, response);
37 }

```


Store the log information in a database – MongoDB

```
HotelModel.java x index.jsp x HotelServlet.java x HotelData.java x
2 usages
110 public static void callMongoDB(String city, String fullLog, long apiCallTimeStamp, long apiResponseTimeStamp, boolean isSuccess){
111
112     ConnectionString connectionString = new ConnectionString("mongodb+srv://DSProject:DSProject123@cluster0.kbrdo.mongodb.net/?retryWrites=true&w=maig
113     MongoClientSettings settings = MongoClientSettings.builder()
114         .applyConnectionString(connectionString)
115         .serverApi(ServerApi.builder()
116             .version(ServerApiVersion.V1)
117             .build())
118         .build();
119     MongoClient mongoClient = MongoClient.create(settings);
120     MongoDB database = mongoClient.getDatabase( s: "DS");
121     MongoCollection<Document> collection = database.getCollection( s: "HotelData");
122
123     String content = String.format("city: %s, apiCallTimeStamp: %d, apiResponseTimeStamp: %d, isSuccess: %b", city,
124         apiCallTimeStamp, apiResponseTimeStamp, isSuccess);
125
126     // Taken from MongoDB Docs
127     // https://www.mongodb.com/docs/drivers/java/sync/v4.3/usage-examples/insertOne/
128     try {
129         System.out.println("Write to MongoDB: " + content);
130         InsertOneResult result = collection.insertOne(new Document()
131             .append("_id", new ObjectId())
132             .append("city", city)
133             .append("fullLog", fullLog)
134             .append("apiCallTimeStamp", apiCallTimeStamp)
135             .append("apiResponseTimeStamp", apiResponseTimeStamp)
136             .append("isSuccess", isSuccess));
137         System.out.println("Success! New Document ID: " + result.getInsertedId());
138     } catch (MongoException me) {
139         System.err.println("Unable to insert due to an error: " + me);
140     }
141 }
142
143 public ArrayList<ArrayList<String>> readMongoDB(){
144
145     ConnectionString connectionString = new ConnectionString("mongodb+srv://DSProject:DSProject123@cluster9.kbrdo.mongodb.net/?retryWrites=true&w=me
146     MongoClientSettings settings = MongoClientSettings.builder()
147         .applyConnectionString(connectionString)
148         .serverApi(ServerApi.builder()
149             .version(ServerApiVersion.V1)
150             .build())
151         .build();
152     MongoClient mongoClient = MongoClient.create(settings);
153     MongoDB database = mongoClient.getDatabase( s: "DS");
154
155     MongoCollection<Document> collection = database.getCollection( s: "HotelData");
156     MongoCursor<Document> docIterator = collection.find().iterator();
157
158     if (docIterator.hasNext()) {
159         System.out.println("Reading and printing all strings contained in the documents to the console:");
160         StringJoiner sj = new StringJoiner( delimiter: "\n");
161         ArrayList<ArrayList<String>> mongoDB_data = new ArrayList<ArrayList<String>>();
162
163
164
165         while(docIterator.hasNext()) {
166             Document doc = docIterator.next();
167             // for each row
168             ArrayList<String> mongo_DB_element = new ArrayList<>();
169             mongo_DB_element.add(doc.getString( key: "city"));
170             mongo_DB_element.add(doc.getString( key: "fullLog"));
171             mongo_DB_element.add(doc.getLong( key: "apiCallTimeStamp").toString());
172             mongo_DB_element.add(doc.getLong( key: "apiResponseTimeStamp").toString());
173             mongo_DB_element.add(String.valueOf(doc.getBoolean( key: "isSuccess")));
174             // append to the full database
175             mongoDB_data.add(mongo_DB_element);
176         }
177     }
178 }
```

Display operations analytics and full logs on a web-based dashboard



DashBoard

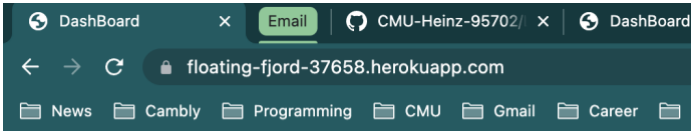
----- Analysis 1 -----

Top City	Search Count
Taipei	12
	8
East Elmhurst	6
Roissy-en-France	4
Las Vegas	2
Pittsburgh	2
Kathmandu	1
Dublin	1
Hayes	1
Taichung	1
Boston	1

----- Analysis 2 -----

Query # Search Time (in second)

1	1
2	1
3	1
4	14
5	1
6	1
7	1
8	0
9	0
10	2
11	1
12	1
13	1
14	0
15	0
16	0
17	1
18	1
19	1
20	1
21	1
22	2
23	1
24	1
25	1



26	1
27	0
28	0
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1

----- Analysis 3 -----

Average Latency of Search: 1

Deploy the web service to Heroku

Link to Heroku: <https://fast-shore-28017.herokuapp.com/>

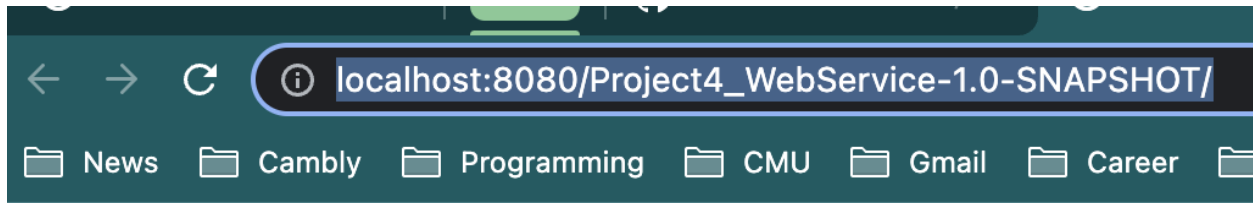
(fixed some minor errors, the link to Heroku screened shot above may be difference, Please use the updated link above as latest correct version)

Link on localhost: http://localhost:8080/Project4_WebService-1.0-SNAPSHOT/

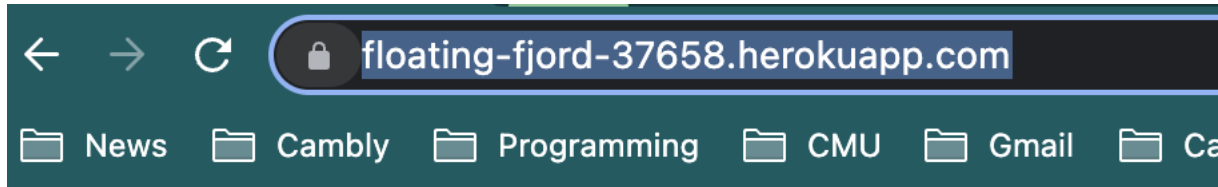
```
heroku -- -bash -- 97x31
Logging in... done
Logged in as yijyunc@andrew.cmu.edu
(base) EvadeMacBook-Pro:~ yjevachen$ cd /Users/yjevachen/Downloads/Project4\ 3/heroku
(base) EvadeMacBook-Pro:heroku yjevachen$ heroku container:login
Login Succeeded
(base) EvadeMacBook-Pro:heroku yjevachen$ export DOCKER_DEFAULT_PLATFORM=linux/amd64
(base) EvadeMacBook-Pro:heroku yjevachen$ heroku create
Creating app... done, ● fast-shore-28017
https://fast-shore-28017.herokuapp.com/ | https://git.heroku.com/fast-shore-28017.git
(base) EvadeMacBook-Pro:heroku yjevachen$ heroku container:push web -a fast-shore-28017
=== Building web (/Users/yjevachen/Downloads/Project4 3/heroku/Dockerfile)
[+] Building 0.9s (8/8) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 258B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/tomcat:10.1.0-M5-jdk16 0.5s
=> CACHED [1/3] FROM docker.io/library/tomcat:10.1.0-M5-jdk16-openjdk-slim 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 2.15MB 0.1s
=> [2/3] COPY ROOT.war /usr/local/tomcat/webapps/ 0.0s
=> [3/3] COPY tomcat_starter.sh /home/ 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:895e858f20d0c4f99a6035f6ef00e6dab17fcbc8c3203 0.0s
=> => naming to registry.heroku.com/fast-shore-28017/web 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix
hem
=== Pushing web (/Users/yjevachen/Downloads/Project4 3/heroku/Dockerfile)
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix t
hem
=== Pushing web (/Users/yjevachen/Downloads/Project4 3/heroku/Dockerfile)
Using default tag: latest
The push refers to repository [registry.heroku.com/fast-shore-28017/web]
a597723cc502: Pushed
f163e0dc4301: Pushed
a3573a3c7fb9: Mounted from floating-fjord-37658/web
43e230202e0b: Mounted from floating-fjord-37658/web
6ae56173d44a: Mounted from floating-fjord-37658/web
ee1d39244ef5: Mounted from floating-fjord-37658/web
f7e816b93eb6: Mounted from floating-fjord-37658/web
a548c9107c3a: Mounted from floating-fjord-37658/web
latest: digest: sha256:a4bf2fb3be274fe14722fb8d5dabef1af0c64c1e6eba33fcabc084c93f967095 size: 199
8
Your image has been successfully pushed. You can now release it with the 'container:release' comm
and.
(base) EvadeMacBook-Pro:heroku yjevachen$ heroku container:release web -a fast-shore-28017
Releasing images web to fast-shore-28017... done
(base) EvadeMacBook-Pro:heroku yjevachen$ heroku open -a fast-shore-28017
(base) EvadeMacBook-Pro:heroku yjevachen$
```


Start from localhost

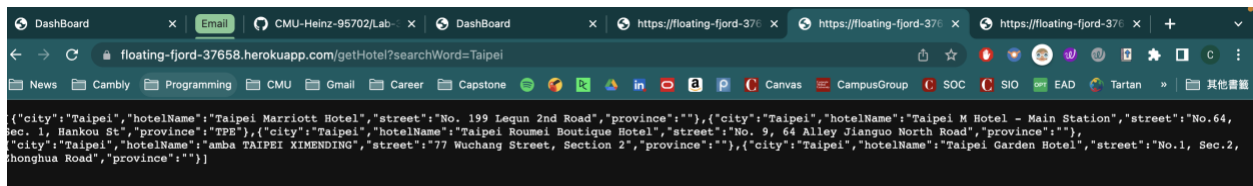


Deploy to Heroku

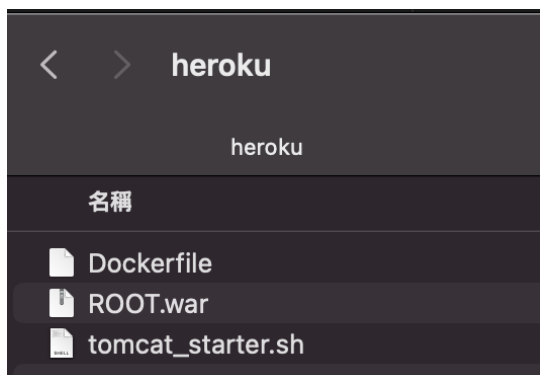
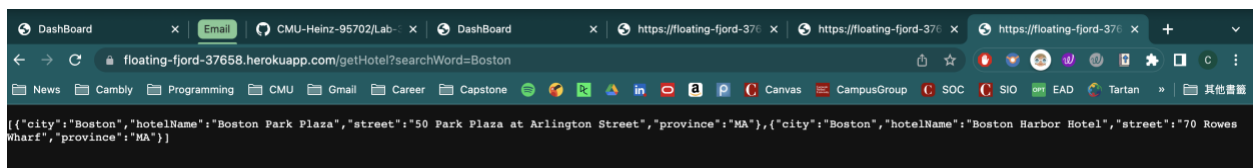


Some example search (of hotels in cities) with Heroku

Taipei



Boston



Implement a native Android application

HotelInfo.java (Main)

```
15
16 import java.util.ArrayList;
17
18 public class HotelInfo extends AppCompatActivity {
19     HotelInfo hotelInfoOrig = this;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25
26         /*
27          * The click listener will need a reference to this object, so that upon successfully finding a picture from Flickr, it
28          * can callback to this object with the resulting picture Bitmap. The "this" of the OnClickListener will be the OnClickListener, not
29          * this InterestingPicture.
30          */
31         final HotelInfo hotelInfo = this;
32
33         /*
34          * Find the "submit" button, and add a listener to it
35          */
36         Button submitButton = (Button)findViewById(R.id.submit);
37
38         // Add a listener to the send button
39         submitButton.setOnClickListener(new View.OnClickListener() {
40             public void onClick(View viewParam) {
41                 String searchTerm = ((EditText)findViewById(R.id.searchTerm)).getText().toString();
42                 System.out.println("searchTerm = " + searchTerm);
43                 GetHotel gh = new GetHotel();
44                 gh.search(searchTerm, hotelInfoOrig, hotelInfo); // Done asynchronously in another thread. It calls ip.pictureReady() in this thread whe
45             }
46         });
47     }
48
49     /*
50      * This is called by the GetHotel object when the result list is ready.
51      */
52     public void HotelListReady(ArrayList<String> hotelInfoList) {
53         TextView searchView = (EditText)findViewById(R.id.searchTerm);
54         TextView isFound = findViewById(R.id.isFoundResponse);
55         TextView hotelListView = (TextView)findViewById(R.id.hotelInfoList);
56         String searchTerm = searchView.getText().toString();
57         if (hotelInfoList != null) {
58             System.out.println("Hotel List");
59             isFound.setText("Here is a list of a hotel in " + searchTerm);
60             StringBuilder sb_list = new StringBuilder();
61             for (String hotelInfo: hotelInfoList) {
62                 sb_list.append(hotelInfo).append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
63             }
64             hotelListView.setText(sb_list);
65             hotelListView.setVisibility(View.VISIBLE);
66         } else {
67             System.out.println("No Hotel Result");
68             isFound.setText("Sorry, no hotel result found for a " + searchTerm);
69             hotelListView.setVisibility(View.INVISIBLE);
70         }
71         searchView.setText("");
72     }
73
74 }
```

GetHotel.java

```
HotelInfo.java × GetHotel.java × activity_main.xml × content_main.xml × AndroidManifest.xml ×
21
22 //Initialize variables
23 HotelInfo hotelInfo = null; // for callback
24 String searchTerm = null; // search hotel for this city
25 ArrayList<String> hotelInfoList;
26
27 // search()
28 // String searchTerm: the thing to search for on flickr
29 // Activity activity: the UI thread activity
30 public void search(String searchTerm, Activity activity, HotelInfo hotelInfo) {
31     this.hotelInfo = hotelInfo;
32     this.searchTerm = searchTerm;
33     new BackgroundTask(activity).execute();
34 }
35
36 private class BackgroundTask {
37
38     private Activity activity; // The UI thread
39
40     public BackgroundTask(Activity activity) { this.activity = activity; }
41
42     private void startBackground() {
43         new Thread(new Runnable() {
44             public void run() {
45
46                 try {
47                     doInBackground();
48                 } catch (JSONException e) {
49                     e.printStackTrace();
50                 } catch (IOException e) {
51                     e.printStackTrace();
52                 }
53
54                 // This is magic: activity should be set to MainActivity.this
55                 // then this method uses the UI thread
56                 activity.runOnUiThread(new Runnable() {
57                     public void run() { onPostExecute(); }
58                 });
59             }
60         }).start();
61     }
62
63     private void execute() {
64         // There could be more setup here, which is why
65         // startBackground is not called directly
66         startBackground();
67     }
68
69     // doInBackground( ) implements whatever you need to do on
70     // the background thread.
71     // Implement this method to suit your needs
72     private void doInBackground() throws JSONException, IOException {
73         hotelInfoList = search(searchTerm);
74     }
75
76     // onPostExecute( ) will run on the UI thread after the background
77     // thread completes.
78     // Implement this method to suit your needs
79     public void onPostExecute() { hotelInfo.HotelListReady(hotelInfoList); }
```

```

98  */
99  @ private ArrayList<String> search(String searchTerm) throws IOException, JSONException {
100      hotelInfoList = new ArrayList<>();
101      String jsonGet = "";
102      String hotelBaseUrl = "https://floating-fiord-37658.herokuapp.com/";
103      //http://localhost:8080/Project4_WebService-1.0-SNAPSHOT/getHotel?searchWord=pittsburgh
104      URL url = new URL( spec: hotelBaseUrl + "getHotel?searchWord=" + searchTerm.replaceAll( regex: " ", replacement: "%20"));
105      //Basic URL formatting
106      //reference https://www.tabnine.com/code/java/methods/java.net.HttpURLConnection.setRequestMethod
107      HttpURLConnection connect = (HttpURLConnection) url.openConnection();
108      connect.setRequestMethod("GET");
109      if (connect.getResponseCode() == 200) {
110          InputStreamReader reader = new InputStreamReader(connect.getInputStream());
111          BufferedReader br = new BufferedReader(reader);
112          String line = "";
113
114          while ((line = br.readLine()) != null) {
115              jsonGet = jsonGet + line;
116          }
117          // reference https://stackoverflow.com/questions/22461663/convert-inputstream-to-jsonobject
118          JSONArray jsonObject = new JSONArray(jsonGet);
119          // reference https://stackoverflow.com/questions/58488521/how-to-count-json-object-item
120          // looping through ALL Selections
121          int totalCityHotel = jsonObject.length();
122          System.out.println((totalCityHotel));
123
124          for(int i=0; i<totalCityHotel; i++){
125              String hotelName = jsonObject.getJSONObject(i).get("hotelName").toString();
126              String street = jsonObject.getJSONObject(i).get("street").toString();
127              String province = jsonObject.getJSONObject(i).get("province").toString();
128
129              StringBuilder sb = new StringBuilder();
130              sb.append("Hotel Name: ").append(hotelName).append(System.getProperty("line.separator")).append("Street: ").append(street).append(System
131              System.out.println(sb.toString());
132

```

Content_main.xml – using 3+ Views

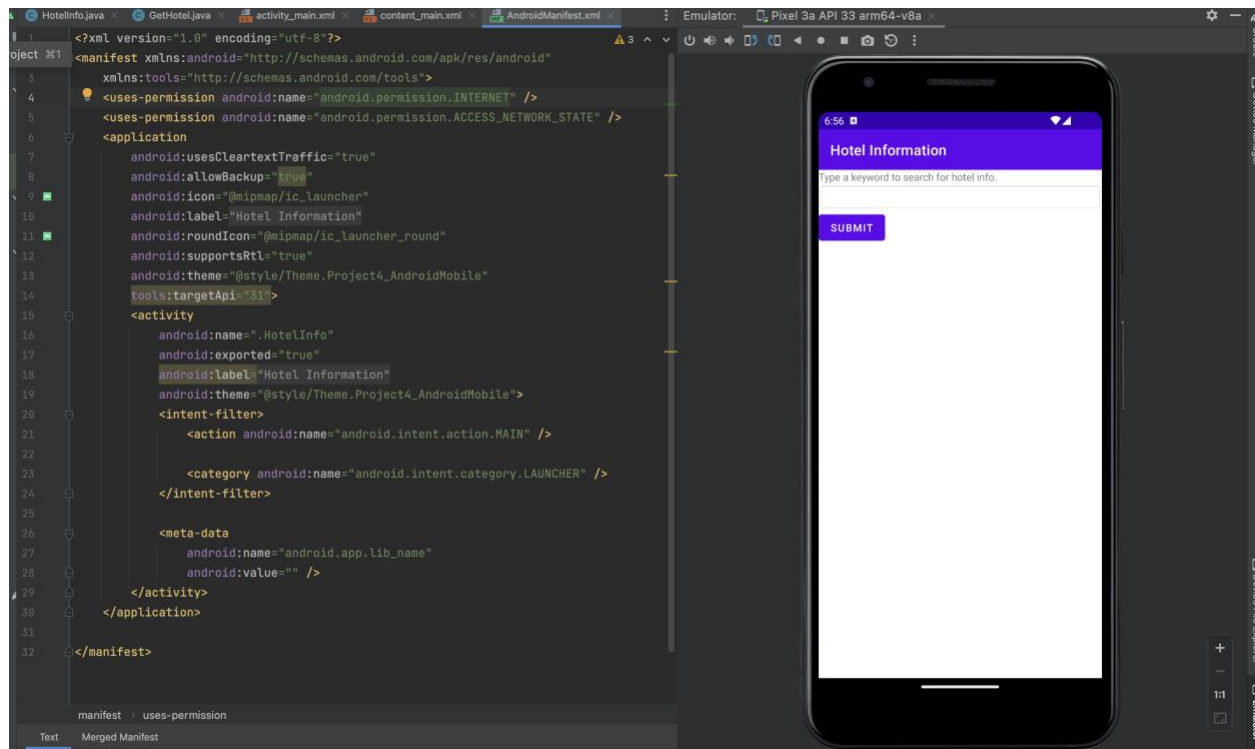
```

2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:orientation="vertical"
5      android:layout_width="fill_parent"
6      android:layout_height="fill_parent">
7
8      <TextView android:layout_width="fill_parent"
9          android:layout_height="wrap_content"
10         android:text="Type a keyword to search for hotel info." />
11
12
13     <EditText
14         android:id="@+id/searchTerm"
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:background="@android:drawable/editbox_background"
18         android:singleLine="true"
19         android:textColor="#0000" />
20
21     <Button
22         android:text="Submit"
23         android:id="@+id/submit"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         />
27
28     <TextView
29         android:id="@+id/isFoundResponse"
30         android:layout_width="match_parent"
31         android:layout_height="wrap_content" />
32
33     <TextView
34         android:id="@+id/hotelInfoList"
35         android:layout_width="wrap_content"
36         android:layout_height="wrap_content"

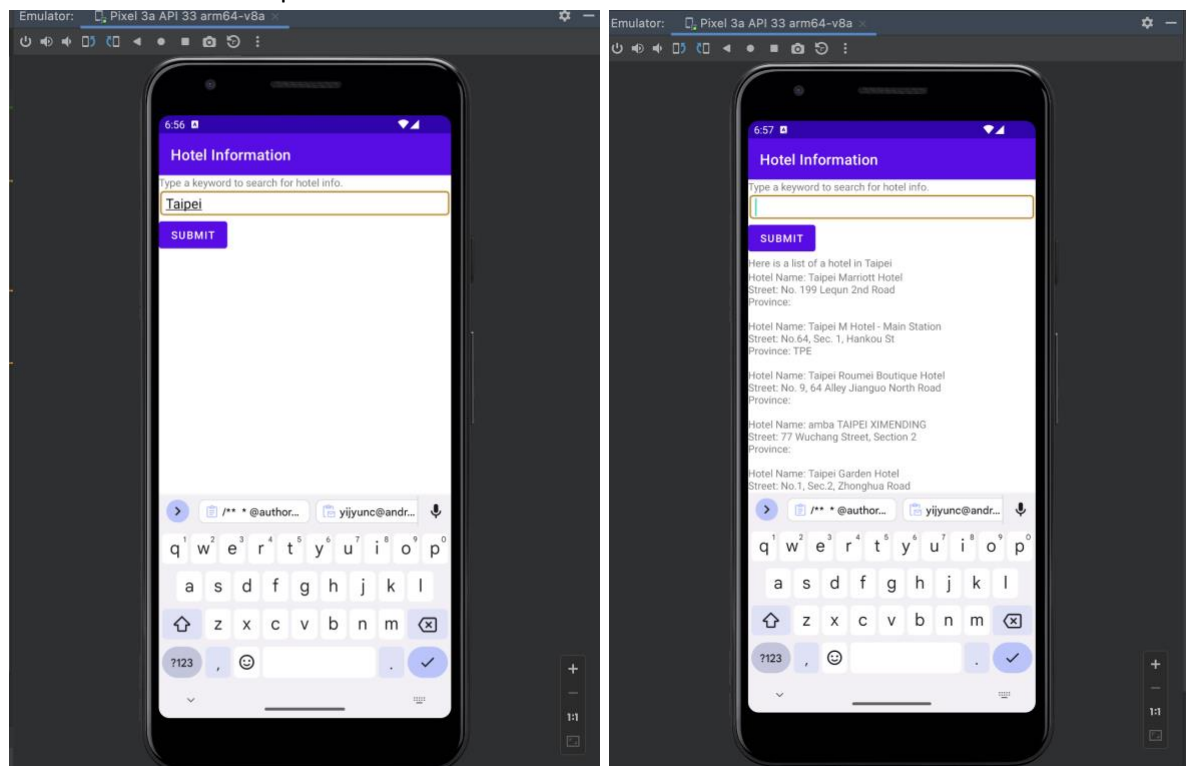
```

Demo of searching with App

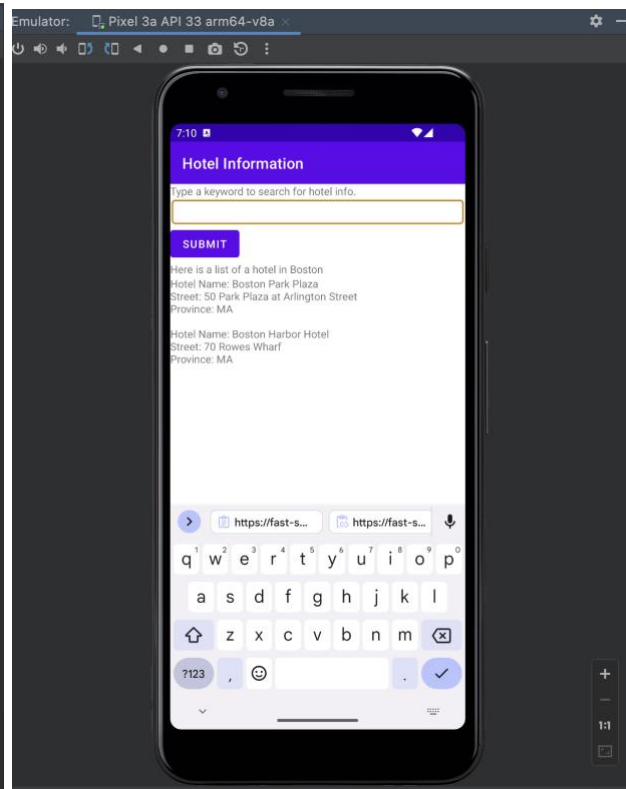
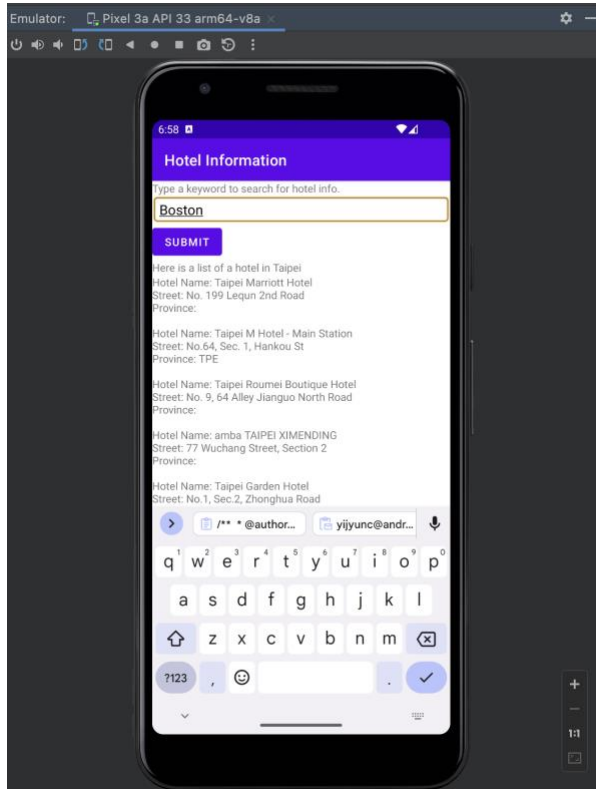
Initial View



Search for hotels in Taipei:



Support continuous search – searching for hotels in Boston after the result of Taipei hotels



No result found

