

**Term Project Report for CSCI 59000
On**

PotLocator



Advisor – Dr. Arjan Durrezi

Report by - Keyur Kirti Mehta

Indiana University Purdue University, Indianapolis

Contents

1) <i>Abstract</i>	4
2) <i>Introduction</i>	4
3) <i>Need & Survey</i>	5
1. Statistics:	5
2. Existing Research:	5
3. Using Sensors Android Services:	6
4) <i>Project Implementation</i>	7
1. Types of Users	7
2. Detecting GPS co-ordinates	7
3. Get location from name	8
4. Communication with Server code	8
5. Node.js	9
6. mySql	9
5) <i>Workflow</i>	10
6) <i>Results</i>	11
1. Login and Sign Up Screen	11
2. Locate Screen	11
3. List of user's detected potholes	12
4. List of all users' detected pothole	13
7) <i>Future Scope</i>	14
8) <i>Conclusion</i>	14
References	15

List of Figure

Figure 1: Application Workflow	10
Figure 2: Login Screen	11
Figure 3: Sign up activity	11
Figure 4: Locate potholes (current location & address)	12
Figure 5: List of marked pot holes	12
Figure 6: List of pot holes detected by all the users	13

1) Abstract

Pot Holes are major concern while travelling on roads. This android app will be used to detect the pot holes' GPS location. Any normal traveler can sign up for the app and start locating the pot holes observed by them and government agencies or the contractors will use this data to identify and repair the same. Google map API is used to detect the location co-ordinates and this data is stored on the cloud.

Keywords: GPS co-ordinates, Google map API, potholes, cloud

2) Introduction

Android is an open source application development platform. And android devices are widely used in people. So, any new content provided on this platform will surely reach to wide range of people. It also provides new features every day. Market for android is increasing since its inception and it will continue to grow in future. So, hot spot for new research and research based applications. Developing research based applications are also easy in android as it has many open source libraries and APIs which can be used easily to develop.

The 'PotLocator' is an android application which will be used to detect the potholes presents in the roads, highways. This application will use Google map API in order to trace the geo co-ordinates of a location. This located co-ordinates will be stored on the centrally located on cloud server. So, that single repository and widely accessible to the authorized users. The contractor or government employees can see the complete list of potholes traced by the users. So as they can use the same to repair it as soon as possible. The application is developed in android along with Microsoft Azure cloud service.

3) Need & Survey

1. Statistics:

Before developing the application, a general study was carried out to analyze the effect of pot holes on the roads. Many different sources were examined and research technique were studied. This study told the impact on vehicular driver/ vehicle of pot hole. Even what are the causes of pot hole has also been identified.

A Pot holes is failure in the structure, in this case of roads. These occurs due to the water and melted ice. The water generated from the melted ice seeps under the roads. This weaken the structure of the road. Also in case of low temperature, this water freeze out and convert into ice form which requires more space. This resulted in the pot holes. (1) Pot holes are occurred mainly because of poor drainage system.

Roads are the integral part of everyone daily life. People daily travel to various places like office, home, school. There are some statistics carried out by National Surface Transportation Policy and Revenue Study Commission of the U.S. Congress, (2) which tells us how widely we use road network.

- 24 million school children ride on 450,000 school buses 180 days per year.
- 50,000 ambulances make 60 million trips a year.
- Trucks carry 32 million tons of goods across the roads every day.
- 240 million registered vehicles travel 2.9 trillion miles annually

Also due to the heavy use of road network, chances of creating pot holes are also increased. Studies also proves that the pot holes are one of the major reason for the road accidents. One third of the total road accidents are due to the poor condition (2)of roads which also includes the pot holes. It has also been studied that the pot holes increases the average maintenance cost. Averagely the maintenance cost of vehicle is increase by \$377 per year. (2)

2. Existing Research:

Various research is carried out to effectively detect the pot holes are repair those to avoid any major consequences. Different sensors are used in order to detect the pot holes. In one of the study it was defined to detect the pot holes using RGB-D sensors. The detection is time consuming due to large road network. New Automated detection system is required to find and detect it. Generally systems use video cameras or laser sensors as the main Automated road analyzer (ARAN) equipped with area-scan cameras and strobe, or IR lighting and digital highway data vehicle (DHDV) equipped with a laser-based imaging system are the most widely used systems. New system uses Kinect devices. These captures the RGB image of the roads and along with that depth sensors will measures the depth variation. Based on the design algorithm, it detects the pot holes. These are efficient way to detect the pot hole. (3)

3. Using Sensors Android Services:

Another way to detect location is to use Google maps API, this can be integrated in Android application. API can detect the geo co-ordinates based on different widgets. This can be triggered based on event or android device sensors detection.

Google provide its Google map API which uses GPS to detect the locations. There are many different APIs provided by Google. (4) These APIs are free to use and embed in the application. Android API for maps is used to detect the GPS co-ordinates.

Accelerometer:

Most of the android devices come with this sensor. And it can be used in an android application. Android have abstract classes 'Services' which can be implemented with Sensors event listener. Services are the threads which can perform the operation without any user interaction. (5) I.e. playing a songs continuously one after another. Android-powered devices have built-in sensors which measures motion, orientation, and various other attributes based on the usage. It captures the raw data and convert it to as per the usage. Such as to monitor three-dimensional device movement or positioning. (6)

Android devices come with multiple sensors. This sensors can also be used as a service while developing the android application. An **accelerometer** is a sensor is used to measure the proper acceleration. Accelerometer uses the gravity component to calculate the acceleration. This sensor when is at rest uses earth's gravity ($g \sim 9.81 \text{ m/s}^2$) (7) But, when it is in free fall (falling toward the Earth) will measure zero. Because of force of gravity in opposite direction. It measures the force in all 3 direction in presence of gravity. Sensor measures the acceleration force in that is applied on a device in all three physical axes (x, y, and z). It will consider the force of gravity.

'onSensorChanged' method can be override (belongs to class SensorEventListener) to detect the changes in accelerometer. (8)The event will return the new force in all 3 directions. Then acceleration needs to be calculated and if changes it observed from previous observation then it can be marked to do some operation. (9) This can be done by detecting the change as follows:

```

if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
    SensorEvent event;
    float[] values = event.values;
    // Movement
    float x = values[0];
    float y = values[1];
    float z = values[2];

    float accelationSquareRoot = (x * x + y * y + z * z)
        / (SensorManager.GRAVITY_EARTH * SensorManager.GRAVITY_EARTH);

    if (accelationSquareRoot >= <threshold value>) {
        <perform desired action>
    }
}

```

4) Project Implementation

The project is developed on Android platform on client server architecture. Android Studio IDE is used to develop the application. Different screens were designed using activity and intent class. User can register to the application and register user can login to the application via login screen. There are 2 types of user login supported by 'PotLocator'.

1. Types of Users

Two types of user login is provided in the application.

a. Normal User:

These are the users who register to the application. And can trace the location of the pot hole while commuting. Users can see the list of pot holes detected by them on the list tab of the activity.

b. Employee User:

These are the special users, who can see all listed pot holes. They are government employees or the contractor who are responsible to maintain the roads. Registration for these user can only be done from back end. No normal user can be an employee users.

2. Detecting GPS co-ordinates

Google map API is used to detect the GPS co-ordinates of the location. First we need to generate the Google API key, which allows you to use Google map services. (10) Using the key GPS location can be detected. 'LocationManager' class are used to get the co-ordinates. (11) Here, method is used which acts as a listener and based on the given parameters, it reads the location.

```
locationManager.requestLocationUpdates("gps", MIN_TIME_BW_UPDATES,  
MIN_DISTANCE_CHANGE_FOR_UPDATES, gpsProviderLocationListener);
```

MIN_TIME_BW_UPDATES is the time interval after which read request is sent

MIN_DISTANCE_CHANGE_FOR_UPDATES is the minimum distance covered from previous requests to get the updated location.

gpsProviderLocationListener is the listener, which will be called based on the above parameter success.

Object of Location class will detect the latitude and longitude when the locationChange request is passed.

```
getLatitude();  
getLongitude();
```

And then pass this co-ordinates as object of LatLag class which later can be used to mark the location on map. (12)

```
LatLng getLatLng = new LatLng(latitude, longitude);
```

These co-ordinates can be marked on the map using Google map object and using the methods provided by its class. There are many different methods which can be used such as adding the marker on the location (13), moving the camera to desired location or zooming the camera (14) to certain level so that the accurate location can be detected.

```
mMap.addMarker(new MarkerOptions().position(getLatLng);  
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(getLatLng, zoomLevel));
```

3. Get location from name

GeoCoder class is used to get the location from the name or address.

```
geo_coder.getFromLocationName(<String>, <count>);
```

String can be any valid address. And <count> is no of location needs to be retrieved for the given address. This class uses the Google map address search and retrieve the corresponding GPS co-ordinates. (15)

4. Communication with Server code

Once the co-ordinates are located, this needs to be store on the server. For this HTTP request is used. The data is converted to JSON string using the JSON objects. And connection is established giving the specific link of the server. (16)

```
HttpURLConnection urlConnection = (HttpURLConnection)  
url.openConnection();  
urlConnection.connect();
```

The JSON object is then passed to the server using the connection which is created earlier. For this bufferstream input/ output is used as follows:

```
OutputStream outputStream = urlConnection.getOutputStream();
```


5. Node.js

Node.js is a JavaScript run-time environment for executing code on server-side. Node.js has an event-driven architecture capable of asynchronous I/O. (17) Based on the HTTP request received from client the node.js script is executed on server and perform the operation. It has different request types like get, post, put etc...

6. mySql

MySQL is used on the server side to store the user information and its marked pot holes coordinates in the database tables.

```
CREATE TABLE IF NOT EXISTS `user` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `first_name` varchar(32) NOT NULL,  
  `last_name` varchar(32) NOT NULL,  
  `email` varchar(64) NOT NULL,  
  `username` varchar(32) NOT NULL,  
  `password` varchar(32) NOT NULL,  
  `is_contractor` char(1) NOT NULL DEFAULT 'N',  
  PRIMARY KEY (`id`) )  
  
CREATE TABLE IF NOT EXISTS `potlocation` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `latitude` decimal(10,8) NOT NULL,  
  `longitude` decimal(11,8) NOT NULL,  
  `reported_by` int(10) unsigned NOT NULL,  
  `reported_on` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `is_repaired` char(1) NOT NULL DEFAULT 'N',  
  PRIMARY KEY (`id`),  
  KEY `FK_portlocation_user` (`reported_by`),  
  CONSTRAINT `FK_portlocation_user` FOREIGN KEY (`reported_by`) REFERENCES  
  `user` (`id`) )
```

5) Workflow

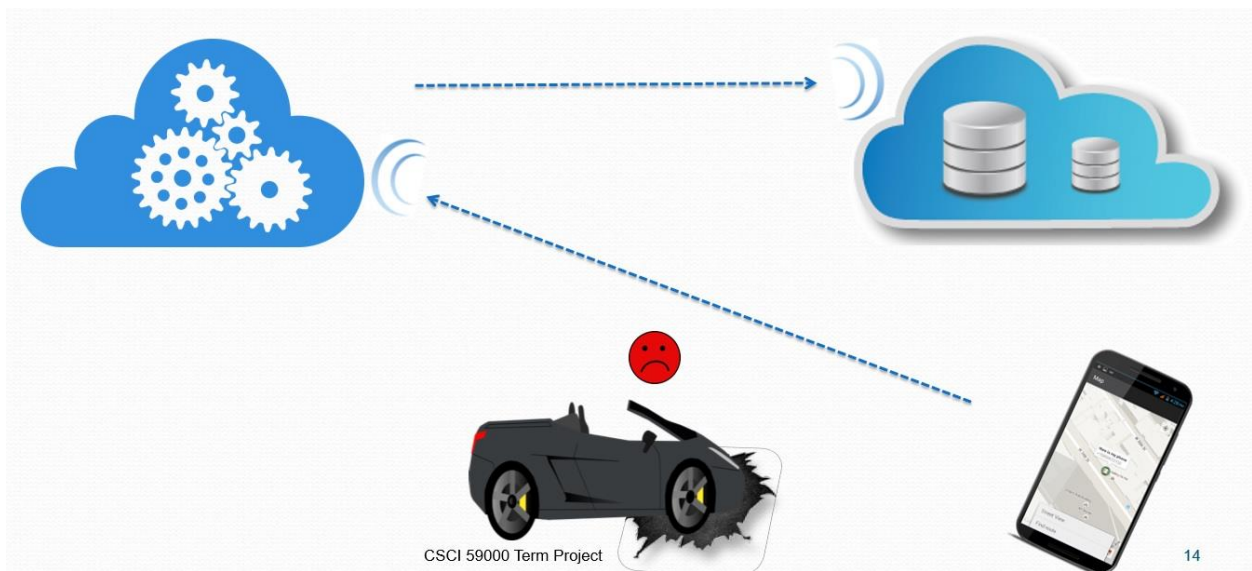


Figure 1: Application Workflow

1. While driving the car, if its car falls into one of the pot holes and user wants to report the same, then it will login to the application.
2. And click on 'Locate Pothole' to mark the co-ordinates of the same,
3. Once it is marked, the request will go to the node.js which is running on cloud server. The request get processed there.
4. And the result will be stored in the MySQL database which is also located on the same cloud.

6) Results

1. Login and Sign Up Screen

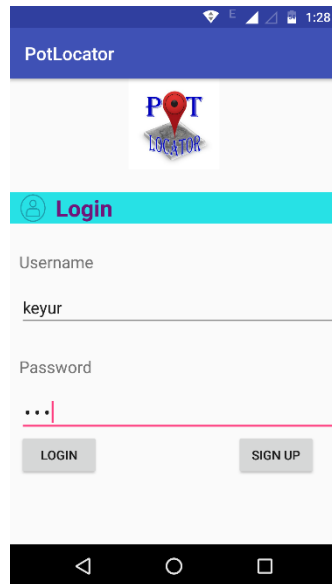


Figure 2: Login Screen

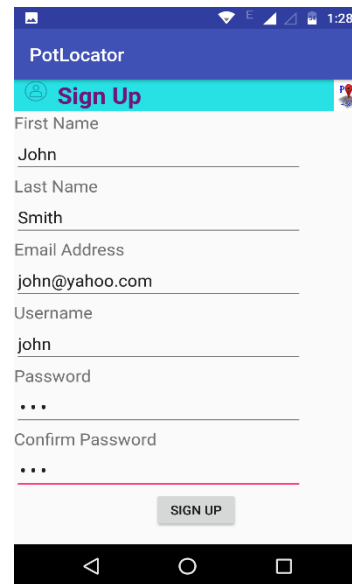


Figure 3: Sign up activity

User will see 'Login' activity on start of the application. Where new user have option to register for the application. And register users can login to the app by providing valid username and password.

2. Locate Screen

Once user is logged on to the app. He will see dashboard activity with 2 tabs. One is 'Locate'.

User have 2 option to detect the pot holes. One is by clicking on the 'Locate PotHole' button which will use current location co-ordinates. And other is typing the address of the location which will add marker to the location and then co-ordinates of that particular location is noted.

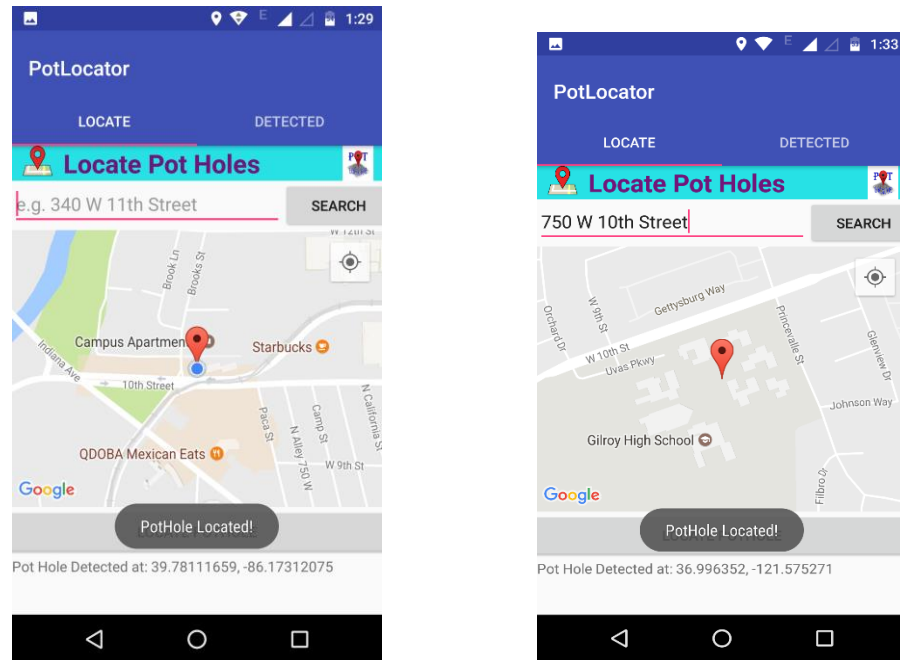


Figure 4: Locate potholes (current location & address)

3. List of user's detected potholes

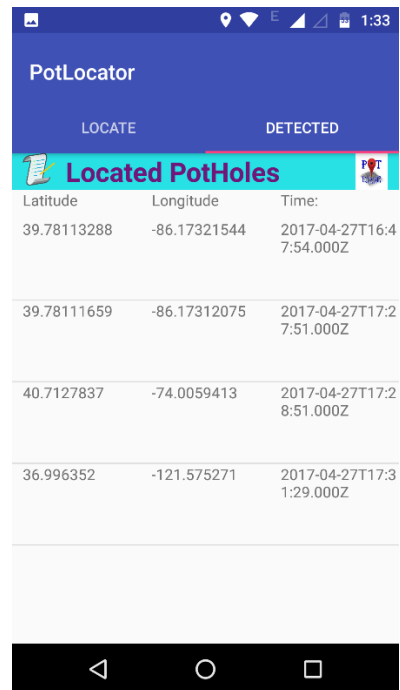


Figure 5: List of marked pot holes

Second tab of the normal user activity will display all the co-ordinates marked by the user.

4. List of all users' detected pothole



Latitude	Longitude	Time:
39.78113288	-86.17321544	2017-04-27T16:47:54.000Z
39.78111659	-86.17312075	2017-04-27T17:27:51.000Z
40.7127837	-74.0059413	2017-04-27T17:28:51.000Z
36.996352	-121.575271	2017-04-27T17:31:29.000Z
39.78113518	-86.17312863	2017-04-27T17:32:30.000Z
18.5204303	73.8567437	2017-04-27T17:33:31.000Z

Figure 6: List of pot holes detected by all the users

Employee user will get list of all the pot holes detected by users on login to the application.

7) Future Scope

1. Detects potholes based on Accelerometer sensors of android device that will automate the detection system and user involvement is not required.
2. Pothole repaired functionality will add feature to employee login. Once pot hole is repaired, it can set the flag for that pot hole as fixed and reported user will also see status of its pot holes.
3. Notification system: Add notification to users on detection of new pot holes and on status change.

8) Conclusion

Environmental conditions are not in our hands. Pot holes will always remain as part of the road networks. The application will make the repair process of the pot holes faster than the existing network. It will make road journey safer than the earlier. This will help to improve the quality of roads and ensure the safety of commuters on the roads. The project helped to learn the cloud services and its implementation.

References

1. [Online] <https://en.wikipedia.org/wiki/Pothole>.
2. [Online] <http://www.pothole.info/the-facts/>.
3. *Unsupervised Approach for Autonomous Pavement-Defect Detection and Quantification Using an Inexpensive Depth Sensor*. Mohammad R. Jahanshahi, Farrokh Jazizadeh, Sami F. Masri, Sami F. Masri, Burcin Becerik-Gerber. s.l. : American Society of Civil Engineers., 2013. 10.1061/(ASCE)CP.1943-5487.0000245..
4. [Online] <https://developers.google.com/maps/faq#whatis>.
5. [Online] <https://developer.android.com/reference/android/app/Service.html>.
6. [Online] https://developer.android.com/guide/topics/sensors/sensors_overview.html.
7. [Online] https://en.wikipedia.org/wiki/Gravity_of_Earth.
8. [Online]
https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER.
9. [Online] <https://developer.android.com/reference/android/hardware/SensorEvent.html#values>.
10. [Online] <https://developers.google.com/maps/documentation/javascript/get-api-key>.
11. [Online] <https://developer.android.com/reference/android/location/LocationManager.html>.
12. [Online]
<https://developers.google.com/android/reference/com/google/android/gms/maps/model/LatLng>.
13. [Online] <https://developers.google.com/maps/documentation/android-api/marker>.
14. [Online] <https://developers.google.com/maps/documentation/android-api/views>.
15. [Online] <https://developer.android.com/reference/android/location/Geocoder.html>.
16. [Online] <https://developer.android.com/reference/java/net/URLConnection.html>.
17. [Online] <https://en.wikipedia.org/wiki/Node.js>.