Gap Trap: A Pothole Detection and Reporting System Utilizing Mobile Devices

Steven Burgart
sburgart@cs.umd.edu
Advised by Dr. Nick Roussopoulos
nick@cs.umd.edu

August 3, 2014

Abstract

In this paper we present a new method and implementation for detecting and reporting roadway potholes. Potholes are a major problem that cost cities and motorists around the world billions of dollars per year. Gap Trap addresses the issue by automatically discovering and reporting potholes so that motorists may avoid them, and cities may quickly repair them. This paper compares Gap Trap with similarly designed systems, and goes on to discuss the implemented prototype system that demonstrates the usefulness of this idea.

1 Introduction

1.1 Background

A pothole is a type of failure in asphalt pavement caused by a combination of presence of water in the underlying soil structure and traffic passing over the affected area[1]. There has been research in attempting to prevent the formation of potholes, of which one idea is sealing roads in order to prevent moisture from penetrating the road surface and causing underlying damage[2]. However, this solution only delays the pothole formation, since the damage that created the crack needing to be sealed will also eventually damage the seal. Another possible solution is to apply an overlay layer of asphalt to the exiting asphalt. This major drawbacks to this are that it is a very costly solution, and it only delays pothole formation. Given this information we can conclude that pothole formation is an unavoidable nuisance that will continue to plague our motorways until a suitable alternative to asphalt becomes the new standard.

Since pothole formation is unavoidable, we can focus on solutions of how to deal with potholes once they are already formed. The issues that potholes present may be alleviated by addressing the following two issues: detecting and reporting potholes to city, and warning motorists of existing potholes so that they may be avoided. Gap Trap addresses both of these issues.

1.2 Motivation

Potholes are a universal inconvenience that affect all roadways. As of May 2014, there have already been 13,000 potholes reported in the Washington D.C. area for 2014[4]. The people who suffer most from the presence of potholes are those who drive on the roadways. They are affected in a number of ways: firstly from direct damage that results from driving over a pothole. This damage this causes will then need to be repaired. According to a study conducted by AAA, pothole damage cost drivers $6.4 billion in the US annually[3]. Secondly, potholes cause car accidents. Upon seeing an upcoming pothole, drivers may suddenly swerve to avoid them - an action that may lead to a wreck. This is supported by the fact that potholes rank 19th in causes of road accidents[5].

If potholes are detected and reported in a timely manner, then the amount of damage that they cause would be reduced. If the city offices in charge of repairing potholes was made aware of potholes locations in a timely manner then they could be more quickly filled, resulting in less motorists driving over them and causing further damage to the road and their car. Gap Trap addresses this issues by passively detecting the existence of potholes and reporting them to a publicly accessible database. A city that is monitoring this database could be made aware of a potholes existence in real time, significantly reducing the delay between pothole formation and fix.

Gap Trap also attempts to mitigate the issues of potholes causing road accidents. We predict that the accidents are caused by motorists swerving to avoid a pothole that they had just seen. If the driver had been made aware of the pothole well in advance, they would have had more time to maneuver around it. To provide this information to the driver we take advantage of the fact that there is a delta between when the system first detects a pothole and when it is fixed by the city. The Gap Trap is aware that a pothole exists, and it is also aware of the cars movement. With these two pieces of information together the system warns the driver of upcoming potholes in the road.

To summarize, the motivation for creating a system like Gap Trap comes from the monetary savings of less damage to vehicles and roadways, as well as the live-saving potential of preventing pothole-related accidents.

2 Related Work

There have been other efforts to create a system that is able to detect and report potholes. The main differences between these systems are their methods of detection, platforms they are implemented on, and where the data is reported.

2.1 Waze

Waze[6] is a free and popular GPS, mapping, traffic, and navigation app available for Android, Apple, and Windows phones. It includes a large number of features to support its navigation functionality, including a hazard reporting capability. This allows users who are currently using the app to report in real-time issues such as lane closures, road construction, debris in road, and also potholes. These reports generated by the user are then sent to the Waze central server for analysis. The report then becomes visible to other users who are also using the app. The end result is that users are warned of the approaching hazard and may confirm the hazard or report that it no longer exists.

There are some differences between Waze and Gap Trap that distinguish them. Firstly, road hazard reporting is only a small portion of the functionality offered by Waze while Gap Trap focuses entirely and specifically on pothole detection and reporting. Secondly, Waze depends on manual user intervention in order to detect and report potholes while Gap Trap utilizes sensors available on the mobile device to automatically detect and report potholes. Finally, the data reported by Waze is only accessible through the Waze app using proprietary communication channels. Gap Trap publishes detected potholes to a publicly accessible interface using standard JSON format.

2.2 Street Bump

Street Bump[7] is a project funded and commissioned by the city of Boston, Massachusetts. The Mayor's Office of New Urban Mechanics implemented this project in order to help deal with the large number potholes within the city that were going unreported. Unlike Waze, Street Bump automatically detects potholes using the accelerometer and GPS sensors built into the mobile device. The reports are then sent to a central server for analysis as well as uploaded to Open311[10], a collaborative model and open standard for civic issue tracking.

Although Street Bump is similar to Gap Trap, there is little information published about the algorithms used to perform automatic detection of potholes. There has been little development or news since Street Bump's release, and some of the information that used to available has since become inaccessible. Also, Street Bump is only available on Apple mobile devices - which hasn't received any updates since May 13, 2013.

2.3 Pothole Patrol

Pothole Patrol[8] is a project created by MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) in 2008. The goal of the project was to create a system for detecting and reporting the surface conditions of roads. This project was established in 2008, before smart phone technology was as ubiquitous as it is today. As such, the Eriksson, et. al utilized the inherent mobility of vehicles equipped with special sensors rather than commodity mobile devices. They equipped the vehicles with three accelerometers: one on the windshield, one on the dashboard, and one in an embedded PC. Using machine learning algorithms on the data from these accelerometers and a GPS sensor they were able to create a classifier that misidentifies road features as having potholes less than 0.2% of the time in controlled experiments.

The main difference between Pothole Patrol and Gap Trap is the devices used to gather data. Pothole Patrol relies on specifically selected hardware that must be installed into each vehicle that will be using the system, while the only hardware required of Gap Trap clients is any Android device equipped with a GPS, accelerometer, and internet connection. Also, the project appears to no longer be in development.

3 Gap Trap Implementation

A prototype implementation of the Gap Trap system has been designed and developed as a proof-of-concept in order to fully demonstrate the usefulness of an automated pothole detection system. The implementation utilizes a client-server architecture that will be described in the subsequent sections.

3.1 Client Device

When developing a system such as Gap Trap it is important to choose a client platform that provides all of the base functionality required of the project. The fact that the device needs to detect and report potholes implies a number of requirements. It will need to be mobile since it will be used within automobiles. It requires a GPS sensor in order to detect the location of the potholes. It will also need some sensors that can be utilized in order to detect the actual potholes. Finally, it will need internet access in order to report the detected potholes. The Android mobile device platform was chosen to satisfy all of these requirements. Specifically, an LG Nexus 5 device was used during the implementation and testing phases of the project. However, a Nexus 5 device is not required - any Android device equipped with a GPS sensor, accelerometer, and an internet connection will perform adequately as a client device.

3.1.1 Pothole Detection

Perhaps the most important aspect of the client device is how it actually detects potholes. A consistent and reliable method is required in order minimize the rate of false positives and missed potholes. The android device we used is equipped with an accelerometer sensor that continuously provides the force of acceleration felt on its three dimensions in the form of x, y, and z variables in meters per second squared.

We can infer that as the device is sensing, a pothole that has been driven over will register as a sudden jolt in acceleration. The direction of the force of acceleration is not as important as the total force, which can be calculated as in equation 1.

$$gforce = \frac{\sqrt{x^2 + y^2 + z^2}}{9.81} \quad (1)$$

You'll notice that this is simply the distance formula applied to three dimensions, and divided by the rate of acceleration of gravity on earth. The resulting value is a measure of gforce, where a gforce of 1 is equal to no acceleration relative to gravity (stationary).

The implemented method of detection uses the given accelerometer capabilities and the derived gforce. In sensing mode, the device will continuously poll the accelerometer and compute the gforce every 20 milliseconds. If at any point the total gforce felt is larger than some threshold then a pothole detection is triggered. Based on basic analysis and testing the current threshold is set to 3.25g.

3.1.2 Sensor & Radio Utilization

There are a number built-in sensors used by the client device for this project. As previously mentioned, the accelerometer is continuously polled to detect the gforce felt by the device. Once a pothole detection has been triggered, the GPS sensor is queried in order to determine the location of the pothole. These two pieces of information together with the current time stamp and a unique device identifier are compiled together into a report. Finally, the devices radio is used for internet access. This internet access is used to transmit the compiled report to a central server outlined in section 3.2.

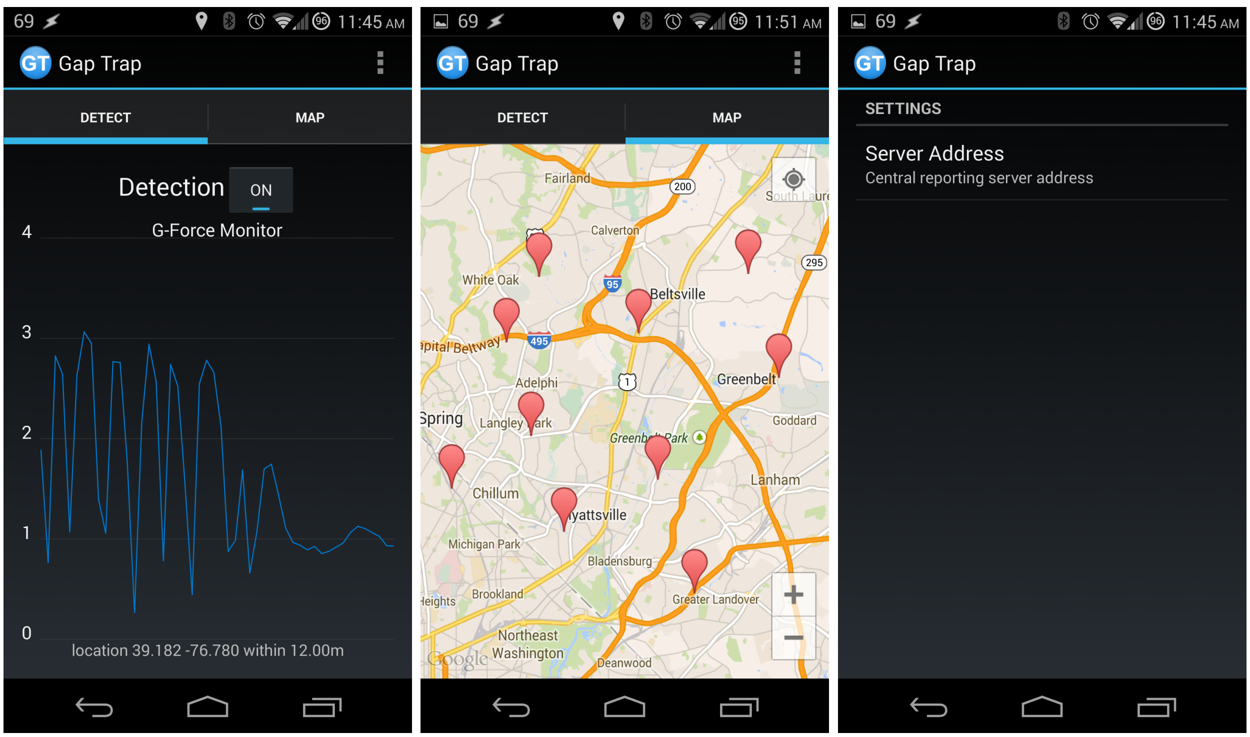
3.1.3 User Interface

The overall design of the user interface for the client has been completed. The Android application design guidelines published by Google[11] were followed very closely during the design phase of the user interface. The app utilizes fixed tabs as the main method of navigation within the app. There are two main UI views, or *fragments*: The *detect fragment* and the *map fragment* which are shown in figures 1a and 1b respectively.

The *detect fragment* is where pothole detection can be switched on or off using a toggle button. Once turned on the device is put into *sensing mode* and a time series graph displays the gforce detected by the accelerometer in real time. Pothole detection will continue even if the phone's screen has been switched off or the user navigates away from the detect fragment. The bottom of the detect fragment displays the device's current location, or *"acquiring location"* if the location has not yet been acquired. A pothole will not be reported unless the devices location is known.

While in sensing mode, and with the location acquired, the device will continuously poll the accelerometer until the gforce threshold described in section 3.1.1 is detected. Once detected, the user is prompted with a pop-up and notification sound to confirm that the jolt in acceleration detected was actually a pothole. Once confirmed, a report is compiled and sent to the central server.

Figure 1b shows the *map fragment*. This view uses the Google Maps API[12] to display a map of the area around the user. On this map are red indicators that mark the location of detected potholes. These markers are not only the potholes reported by the current user, but all potholes detected by all users that have been reported to the central server. A user may tap on a marker to view more information about the report it represents, such as the device that reported it, the time and date of detection, and the gforce that triggered the report.



(a) Detect UI (b) Map UI (c) Preferences

Figure 1: Client Device User Interface

3.1.4 Preferences

The client app contains a preferences activity shown in figure 1c. The preferences activity has one option and that is to configure the central reporting server address. By making this a configuration option a user can update the address of the central server if it moves or if the user chooses to send reports to a different server rather than the default. Without this option a user would need to re-install the app every time they wanted to make such a change.

3.2 Central Server

The Gap Trap system is centralized - meaning that in a complete system there may be an arbitrary number of clients, but there is only one central server. The client devices described in section 3.1 are responsible for automatically detecting and reporting potholes, while the central server has two main responsibilities.

The first responsibility of the server is the processing of reports sent from client devices. This entails listening for, collecting, processing, and storing the pothole reports. The second responsibility of the server is to display the collected pothole information in a meaningful way. How the server accomplishes these goals is described in the subsequent sections.

3.2.1 Implementation Details

The java programming language is ued to handle the handle the back-end logic of the web server. Leveraging JavaServer Pages technology, the entire server logic can be packaged into a single WAR file and deployed onto a tomcat instance, making it very portable. The user interface for the website is done using a combination of HTML, CSS, and JavaScript with the use of jQuery and Bootstrap libraries.

3.2.2 Report Processing

The server is always listening for report submissions from clients. The reports submitted by clients are sent using a standard HTTP POST request with the following parameters:

androidid A 64-bit string of hex characters that is unique to each android device

latitude Latitude of detection location using the signed degrees format

longitude Longitude of detection location using the signed degrees format

gforce The gforce felt by the device that triggered the automatic detection

Once received, the server will analyze the reports for any formatting errors and insert them into a database.

3.2.3 Retrieving Reports

Once inserted into the database the reports are then made available for retrieval. An interface exists between the database and web server that allows for the retrieval of reports in standard JSON format. This interface is used throughout the Gap Trap system. It is used by the client devices to display reports on the map fragment, and it is used by the web server user interface to display the data in a map format as well as in a data table format. An example result of a report query is shown below.

```json
[
  {
    "rid": 3365603,
    "androidid": "e2dfbf540a5bdd62",
    "timestamp": "2014-04-27 19:44:16",
    "latitude": 38.985483,
    "longitude": -77.01725,
    "gforce": 3.26
  },
  {
    "rid": 3365605,
    "androidid": "e2dfbf540a5bdd62",
    "timestamp": "2014-04-27 19:46:39",
    "latitude": 38.95452,
    "longitude": -76.991844,
    "gforce": 4.24
  },
  ...
]
```


3.2.4 Database

For data storage, a MySQL database is used. The schema of the database is illustrated in the form of an entity-relationship diagram in Figure 2. The database consists of two tables: *Device* and *Report*. *Device* stores a unique identifier for a client as well as a time stamp of the first report received from that device. The report table will hold records of pothole reports. Each report identifies the client, the location the pothole was reported from, the time of the report, and the gforce of the action on the phone that triggered the pothole detection.

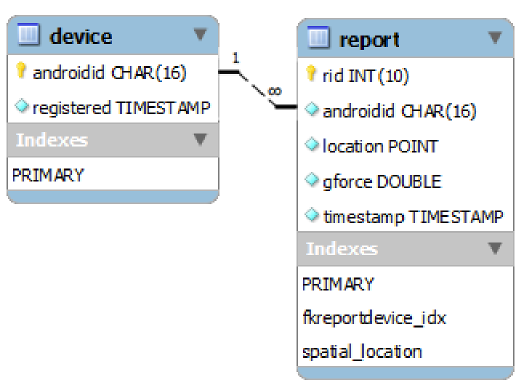


Figure 2: Database ER Diagram

The tables within a MySQL database must be configured to use a storage engine. Gap Trap uses the MyISAM storage engine for all of its tables. This is because MyISAM offers a number of advantages for geographic data. MyISAM fully supports spatial indices in the form of R-Trees. It uses a native Geometry data type known as a *Point* for locations, and a built-in function *st_distance* which is used for very fast determination of distances around a point. For example, locating all potholes within *x* miles of point *p* was computed in 0.259 seconds with 1,000,000 records. The query is shown in figure 3.

```
SELECT location FROM report WHERE st_distance(location, CENTER) <= DISTANCE;
```

Figure 3: Distance SQL Query

3.2.5 Web User Interface

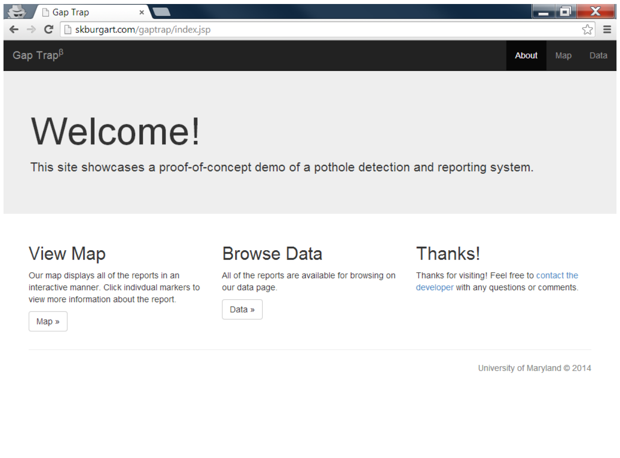
The server web user interface can be accessed from any web browser - a live demonstration of which is available at *http://skburgart.com/gaptrap*. The website consists of three main pages: the welcome page, the map page, and the data page which are shown in figures 4a, 4b, and 4c respectively.

The welcome page is self-explanatory - it is simply an entry point into the website that describes the project and the other pages that are available. The map page is very similar to the *map fragment* of the client user interface described in section 3.1.3. It displays markers on a map that represent automatically detected and reported potholes from every client device. Clicking on a marker will bring up a dialog providing more detailed information about the report - such as the device that reported it, the time and date of detection, and the gforce that triggered the report. Finally, the data page provides a breakdown of all pothole reports. The data is presented in a searchable and sortable table containing the device id, time stamp, latitude, longitude, and gforce of each report.

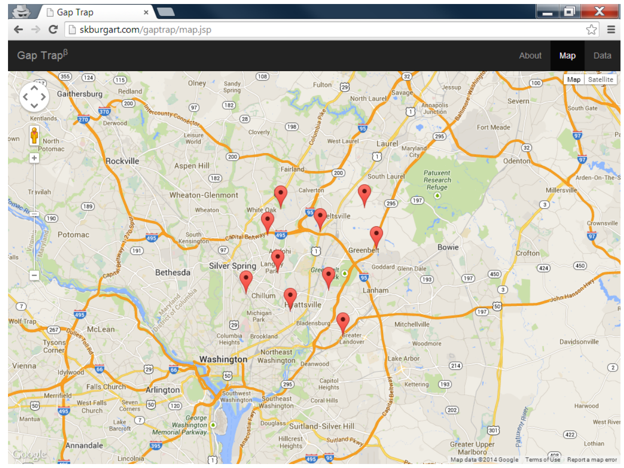
This web user interface would be particularly useful to city officials from around the country. It provides a complete and clear view of where the potholes are located within their city - allowing them to quickly dispatch repair teams to fill the potholes in a timely manner.

4 Future Work

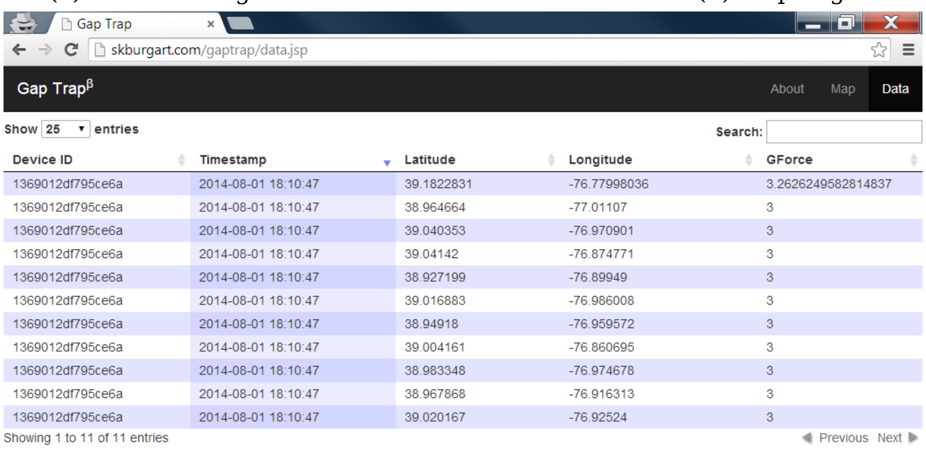
There are many areas where this system can be improved. For example, properly profiling potholes would be a large improvement. Right now the system only uses a g-force threshold to trigger pothole detection, but by testing the system with actual data gathered from vehicles driving over potholes we could properly profile the data using machine learning algorithms. This idea was explored by Mednis, et al, in



(a) Welcome Page



(b) Map Page



(c) Data Page

Figure 4: Web User Interface

their experiments with potholes detection using smartphone accelerometers[9]. After collecting the pothole data they applied different algorithms in attempt to properly classify potholes, such as Z-THRESH, STDEV(Z), and their own algorithm G-ZERO. Ultimately they were able to achieve positive detection rates as high as 90%.

To augment the client device, a navigation-like interface could be iplemented. With this feature a user would be warned when a pothole is approaching on the road on which they are travelling and they could take the necessary precautions to avoid it. Another area of improvement is refined search queries. Currently, the searching of the pothole data is not very intuitive or complex. It could be expanded to handle geographic queries and fine-tuned queries.

5 Conclusion

In this paper we have described a problem that currently exists with potholes and the issues they pose, as well as presented a solution in the forum of a comprehensive pothole detection named Gap Trap. We believe that this system could be a great asset to both motorists and city officials for quickly detecting and repairing potholes.

References

[1] Wright, Kendall C. "Patching The Pot Holes In Your Leadership Relationships" *Entelechy Training and Development, Inc.* Accessed June 13, 2014 http://www.entelechycan.com/pages/Articles/files/issues/issue_index53.pdf

[2] Johnson, A M. "Best Practices Handbook On Ashphalt Pavement Maintenance" *National Technical Information Service* 2009 http://trid.trb.org/view.aspx?id=648010

[3] Author Unlisted "AAA: pothole damage costs drivers $6.4 billion a year" *WUSA9* February 24, 2014 http://www.wusa9.com/story/news/nation/2014/02/24/potholes-damage-cost-us/5773501/

[4] Hasley, Ashley III "Potholes infest D.C. region area as relentless winter chews up the roadways" *The Washington Post* Februray 22, 2014 http://www.washingtonpost.com/local/trafficandcommuting/potholes-infest-dc-region-area-as-relentless-winter-chews-up-the-roadways/2014/02/22/0627257a-9a36-11e3-b931-0204122c514b_story.html

[5] Law Offices of Michael Pines, APC "Top 25 Causes of Car Accidents" *SeriousAccidents.com* Accessed June 15, 2014 http://seriousaccidents.com/legal-advice/top-causes-of-car-accidents/

[6] Author Unlisted "Waze: A GPS-based geographical navigation application program for smartphones", 2009-2014 *Waze Mobile* Accessed June 16, 2014 https://www.waze.com/about

[7] Author Unlisted "Street Bump" *Mayor's Office of New Urban Mechanics in Boston* Accessed June 16, 2014 http://www.streetbump.org/about

[8] Eriksson J, Girod L, Hull B, Newton R, Madden S, Balakrishnan H "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring" *MobiSys '08* http://robotics.usc.edu/ gaurav/CS546/readings/eriksson_mobisys2008.pdf

[9] Mednis A, Strazdins G, Zviedris R, Kanonirs G, Selavo L "Real Time Pothole Detection using Android Smartphones with Accelerometers" *2011 IEEE, DOI* http://strazdins.lv/papers/mednis2011androidpotholes.pdf

[10] Author Unlisted "A collaborative model and open standard for civic issue tracking" *Open311* Accessed July 2, 2014 http://www.open311.org/

[11] Author Unlisted, "Android Design" *Google* Accessed July 27, 2014 http://developer.android.com/design/index.html

[12] Author Unlisted "Google Maps Android API v2" *Google Developers* Accessed July 22, 2014 https://developers.google.com/maps/documentation/android/