University of Nairobi

School of Engineering

DEPARTMENT OF GEOSPATIAL AND SPACE TECHNOLOGY

DEVELOPMENT OF A MOBILE APPLICATION FOR RECORDING AND SHARING ROAD INCIDENTS DATA

CASE STUDY: NAIROBI COUNTY

MACHARIA, STEPHEN NGUNJIRI

F56/69157/2013

Project report submitted in partial fulfillment of the requirements for the degree of Master of Science in Geographic Information System, in the Department of Geospatial and Space Technology of the University of Nairobi.

JULY, 2017

DECLARATION

I, Stephen N Macharia hereby declare that this project is my original work. To the best of my knowledge, the work presented here has not been presented in any other institution of Higher Learning.

STEPHEN N MACHARIA ………………… ……………

Name of student Signature Date

This project has been submitted for review with my approval as university supervisor.

DR D.N. SIRIBA ………………… ……………

Name of Supervisor Signature Date

ACKNOWLEDGEMENT

First, I give thanks to the Almighty God for giving me the strength and good health to undertake this project. Secondly I also want to thank my supervisor Dr. D.N. Siriba for guiding me throughout the project and for his unwavering support. With his wisdom and contributions I was able to complete this research study. A lot of appreciation also goes to the road safety authorities in Kenya namely National Transport and Safety Authority (NTSA) and the Kenya Traffic Police Department for their support and expanding my knowledge on road safety subject. I also want to extend my thanks to the department at large for their support and their guidance. My thanks also go out to my colleagues and classmates who have walked with me throughout this period for their advice. Finally, I thank my family and friends for standing with me all the time.

Stephen N Macharia

University of Nairobi

July, 2017

ABSTRACT

A road incident could be an accident, carjacking, speeding, hit-and-run and among others. The frequency of these road incidents is on the rise in Kenya today with many of these incidents going unnoticed and the parties involved in these unreported cases often never being held accountable. One of the factors that contribute to increased frequency of road incidents is that road users lack a simple and proper medium which can be used to report such incidents and complains directly to the authorities. Most witnesses would rather leave the incident to chance rather than go to the relevant authorities because they lack a convenient mechanism to report such kind of incidents. A unified way of sharing road incidents data with different stakeholders including the general public is also absent. Use of Geographical Information Systems (GIS) to manage road incidents has since gained prominence around the world over time. Web mapping can be used as a quicker way to report road incidents as well as improve sharing of road incidents data.

In this study a road incidents recording and sharing system was developed. This system consists of a geodatabase of road incidents for the study area. A mobile application was also created and is used to report road incidents which are stored in the geodatabase. These incidents are displayed on a web map where these details can be viewed by any interested party. The Internet makes these web maps more accessible because no special software needs to be installed on the user's computer. Additionally authorized people can update the action progress and also flag off incidents which are resolved. This is important in ensuring that all incidents are attended to. Location information recording is supported in this application and depends entirely on a smartphone's inbuilt Global Positioning (GPS) module and Google Places Application Programming Interface (API). The development of this system followed the waterfall System Development Life Cycle (SDLC) and was supplemented by use of prototypes. A qualitative framework was developed to access the suitability and applicability of this medium for reporting and sharing road incidents data. The testing was conducted with a limited number of users in Nairobi area and was hugely successful with many agreeing that such a system would improve the reporting and response to road incidents. The users also found the application to be usable and easy to understand and can improve road incident reporting and contributing to road safety improvement.

TABLE OF CONTENTS

DECLARATION ... ii
ACKNOWLEDGEMENT ... iii
ABSTRACT ... iv
LIST OT TABLES ... vii
TABLE OF FIGURES ... vii
LIST OF ABBREVIATIONS AND ACRONYMS ... viii
Chapter 1 INTRODUCTION ... 1
1.1 Background ... 1
1.2 Problem Statement ... 2
1.3 Objectives ... 3
1.3.1 Main Objective ... 3
1.3.2 Specific Objectives ... 3
1.4 Justification of the Study ... 3
1.5 Scope ... 4
1.6 Limitations ... 4
Chapter 2 LITERATURE REVIEW ... 5
2.1 Road Incidents and Road Safety in Kenya ... 5
2.2 Trend in Road Traffic Incidents ... 6
2.3 Road Transportation in Nairobi County ... 7
2.4 Road Transport Agencies ... 8
2.5 GIS in Road Safety and Incident Reporting ... 8
2.5.1 Nduru Road Safety App ... 9
2.5.2 GIS-Alas (GIS-based Accident Location and Analysis System) ... 9
2.5.3 Ma3Route ... 9
2.6 Web Mapping ... 10
2.6.1 Web Mapping Application Programming Interfaces. ... 10
2.7 Volunteer Geographic Information ... 11
2.7.1 VGI and Base Mapping Coverage ... 12
2.7.2 VGI and Emergency Reporting ... 12
2.8 Systems Development Life Cycle (SDLC) ... 12

2.8 Summary ... 14
Chapter 3 METHODOLOGY AND MATERIALS ... 16
3.1 Software Development Life Cycle Methodology ... 16
3.2 Feasibility Study ... 16
3.3 Business Study ... 16
Research Design ... 16
3.4 System Design, Architecture and Implementation ... 18
3.4.1 Data Modelling ... 18
3.4.2 Application Implementation ... 19
3.4.3 System Evaluation and Testing ... 29
Chapter 4 ANALYSIS OF RESULTS ... 30
4.1 Results and Findings ... 30
4.2 Evaluation of Results ... 30
4.3 User Perception of the System ... 35
Chapter 5 CONCLUSIONS AND RECOMMENDATIONS ... 37
5.1 Conclusions ... 37
5.1.1 Challenges in Implementing the Application ... 37
5.2 Recommendations ... 37
5.3 Suggestions for Further Work ... 38
REFERENCES ... 39
APPENDIX: Sample source code ... 41

LIST OT TABLES

Table 3.1 Key System Entities 18
Table 4.1 System Evaluation Questionnaire 31

TABLE OF FIGURES

Figure 2.1 Comparison of fatalities financial year 2014/2015 7
Figure 2.2 Waterfall Methodology 14
Figure 3.1 System Entity Relationship Diagram 19
Figure 3.2 system Three-tier architecture 20
Figure 3.3 Overview of the system Architecture 21
Figure 3.4 System Use Case Model 22
Figure 3.5 Submit incident user form 25
Figure 3.6 Select image source user form 26
Figure 3.7 Distress call button interface 27
Figure 3.8 View incidents user interface 28
Figure 3.9 Track Reported Incidents user interface 29
Figure 4.1 Evaluation of impact of the incident recording application 32
Figure 4.2 Evaluation of impact of application to road safety 33
Figure 4.3 Evaluation of users willing to use the system 34
Figure 4.4 Evaluation of System friendliness and ease of use 35

LIST OF ABBREVIATIONS AND ACRONYMS

GPS Global Positioning System

PHP Hypertext Preprocessor

WHO World Health Organization

NTSA National Transport and Safety Authority

GRPS General Packet Radio Service

GIS Geographic Information System

OS Operating System

RTA Road Traffic Accident

RTI Road Traffic Incident

VGI Volunteer Geographic Information

KURA Kenya Urban Roads Authority

KRB Kenya Roads Board

OSGR Ordance Survey Grid Reference

SQL Structured Query Language

Chapter 1 INTRODUCTION

1.1 Background

A road incident refers to an untoward event which happens on the roads and which may lead to loss, damage, disaster or loss of lives. Such road incidents include road accidents, speeding, drunk driving, carjacking and hit-and-run. Road traffic incidents exert a huge burden on Kenya's economy and health care service. Most causes behind most road incidents in Kenya include lack of proper driving schools, unskilled drivers, poor road conditions, distracted driving, overloading and inadequate laws & legislations (Odero and Heda, 2003). According to the 2009 World Health Organization (WHO) global status report, Kenya recorded 3760 traffic deaths, the highest in the region. WHO, in the global status report on road safety 2015 report predicts global safety fatalities to rise to 2.4 million per year by 2030.In low and middle-income countries, data suggest that road traffic deaths and injuries are estimated to cause economic losses of up to 5% of GDP. These losses undoubtedly inhibit the economic and social development of developing countries (WHO, Global status report on road safety, 2015).

Road users often lack a simple medium where they can report road incidents and complaints directly to the authorities concerned. According to the National Transport and Safety Authority (2017) the frequency of road incidents that go unnoticed in Kenya is on the rise especially the hit and run incidents. Since there is no simple medium to report them, they go unaccounted for and hence the parties accountable are never held responsible. Road incidents recording and analysis plays a significant role in the strategy to reduce road incidents. This is true if supported by an efficient collection and analysis system. Traditional systems often consist of a database in a summary spreadsheet format using codes and mileposts to denote location, type, and severity of incidents. A geo-referenced incident database is location referenced. It incorporates a GIS graphical interface with the incident information to allow for query searches on various incident attributes. A geo-referenced incident database is easy to access, display, visualize, query, analyze and combine data from other databases by using GIS while at the same time providing flexible and visual report functions. This project aims to demonstrate how GIS and some of its related disciplines can assist in recording, processing and analysis of road incidents information and thus improving awareness and action using incidents data. It also provides some insight into the development of an Internet-based GIS incident reporting system. It addresses the deficiencies through efficiently, effectively and reliably reporting and disseminating data from road incidents in Kenya by developing a mobile computer application for reporting and sharing data regarding road incidents. The collected roads incident data benefits the public as well as other stakeholders such as the police workforce, the insurance industry, higher education institutions.

The rapid growth of technology has made life easier with increased use of internet and smartphones. According to Communications Authority of Kenya statistics, the number of mobile subscribers in Kenya has grown by 1% to reach 38.9 million by the end of quarter two of financial year 2016/2017. This has seen the mobile penetration increase to 88.2 percent. This advancement in technology has seen a large number of the population constantly using their mobile phones (Communications Authority, 2017).

The phenomenon of Volunteer Geographic Information (VGI) is a valuable source of information in the modern society. In this approach data sources are provided by communities of volunteers as opposed to government entities or for-profit businesses. This has been fuelled by the emergence of Web 2.0 technologies, miniaturization of Global Positioning Systems (GPS) devices and advancement of broad communication links. This has empowered more Internet users referred to as 'citizen' or 'sensor' to become increasingly in producing geographic information. In VGI, the citizen has been identified as an important and rapidly source of geographic data. It has also been identified that in future the citizen will play an increasing role both as a consumer and a producer of geographic data (Goodchild, 2007).

1.2 Problem Statement

The frequency of road incidents that go unnoticed are on the rise; for instance, hit-and-run cases. Such incidents often go unaccounted for and hence the parties responsible are not brought to account. Witnesses and victims of road incidents often lack a simple medium to report road incidents. The authorities on the other hand lack concrete evidence to solve road incidents since there is no proper road incident reporting system. Many policies have also been enacted to curb road carnage. However, many incidents still happen and do not reach the appropriate agencies in time. These indications, coupled with the fact that the total number of mobile subscribers in Kenya has increased therefore increasing the penetration of mobile telephony services, show that reporting road incident can be simplified using a simple medium and thus there is a great potential for a mobile application to report these incidents as they occur. There are several incidents, which happen and go unrecorded or even not reported because a simple medium for recording and reporting the road incidences to the appropriate authorities is not available.

Implementing a Geographical Information System (GIS)-based road incidents recording and sharing provides an opportunity for improved dissemination of information as well as improved visualization and analysis techniques. Use of GIS comes in handy in identifying the location of the incident reporter, mapping the incident to the location as well as maintaining an accurate database. The mobile application developed in this study runs on a mobile phone and helps reduce the overall time used in reporting incidents, as well as promoting sharing of this information.

1.3 Objectives

1.3.1 Main Objective

The main objective of this study is to develop a road incidents reporting system that contributes to road safety by improving reporting and response as well as improving the dissemination and analysis of information arising from these incidents.

1.3.2 Specific Objectives

I. Develop a road incident geodatabase for sharing data among the interested parties
II. Develop a mobile application through which people can report road incidents and a web application to supplement the mobile app functionality
III. To test the application with the case study area road users.

1.4 Justification of the Study

Much progress has been made in Kenya over the years in the different aspects of road safety by introducing new safety initiatives. These initiatives include improved education and training standards, improving road design, passing new legislations as well as enforcements and introducing new technology such as speed cameras and detectors. The primary objective of these road safety initiatives is to reduce road incidents by mitigating their occurrences. However, many of these have failed to keep pace. For instance, the requirement to install seat belts in all Public Service Vehicle (PSV) failed due to poor administration (Bachani *et al*., 2012).These initiatives have also included improved education and training standards, improving road design, passing new legislations as well as enforcements and introducing new technologies such as speed cameras and detectors and breathalyzers. Reducing road incidents requires efficient management of all the information related to such kind of incidents. The aim of this project is to develop a mobile application that can be used to report road incidents. With a proper road incident reporting system, road users can report any road incidents that pose a threat to road safety on Kenyan roads and thus also improve incident resolution by the authorities by assisting the authorities to respond, track and resolve reported incidents. It also provides a publicly accessible platform through which data can be shared and disseminated.

1.5 Scope

This project was limited to developing both a mobile application and a web application. The target users are all road users including pedestrians, motorists and other interested parties. The test case audience of this study is limited to the Nairobi urban population which was easy to reach and who provided evaluation and feedback of the system. The application was developed on Google's Android platform.

1.6 Limitations

I. The mobile app will be available for only Android devices and the target users were limited to the Nairobi urban population

II. VGI can sometimes have a lack of data quality assurance and also introduce issues surrounding liability and security.

Chapter 2 LITERATURE REVIEW

This section describes the current state of road safety in Kenya including the various Road Incident Systems that have been deployed and being used in the various parts of the world. It also reviews the concepts of web mapping and Volunteer Geographic Information (VGI).

2.1 Road Incidents and Road Safety in Kenya

Road incidents are a growing public health concern worldwide and are on the rise in the world today causing loss of lives, disabilities and loss of resources. Globally, road incidents are ranked as the ninth leading cause of death (World Health Organization, 2015). Road traffic injuries place a heavy burden on global and national economies and household finances. With more than 13 million deaths and 20 – 50 million injuries being directly linked to road traffic incidents in the world, the social and economic burden presents a compromising scenario for Kenya as a nation (Bachani, 2012). According to Manyara (2016), in Kenya alone, over 3000 people die through road incidents every year, most of them in their prime(15-45 years). Most of the road incidents lead to fatalities or serious disabilities which disenfranchise many families. Quite often the transport industry stakeholders blame the poor state of Kenya roads as the leading cause of incidents. According to the World Health Organization (2015), between 3000 and 13000 Kenyans lose their lives in road traffic crashes every year. WHO in its report 'The Global Status on Road Safety' further states that Kenya, Tanzania and Rwanda with respectively 29.1,32.9 and 32.1 deaths per 1000 people, are amongst the worst 10 performers, in terms of fatalities in Africa, and among the worst 20 worldwide. Uganda's rate is a little lower at 27.4. The majority of this is vulnerable road users including pedestrians, motorcyclists and cyclists. According to the Kenya Police data, on average, one in ten people who are involved in road related accidents die instantly with most cases involving the death of pedestrians (Odero *et al*., 2003). Road transport in Kenya is the most dominant form of transport accounting for 93% of all passenger and cargo traffic (Matheka *et al*., 2015). Matheka *et al*. (2015) further states that based on the Accident Cause Code Classification, the police reports indicate that 85% of road crashes are caused by poor driver behavior, of which driver error represents 44%, pedestrians and passengers 33% and pedal cyclists 7%.

The objective of establishing the National Transport and Safety Authority (NTSA) in Kenya was to ensure the provision of safe, reliable and efficient road transport service and harmonize the operations of key road transport departments and help in effectively managing the road transport subsector and minimizing loss of lifes. NTSA is also tasked with advising and making recommendations on matters relating to road transport and safety. NTSA also develops and implements road safety strategies. Using data from NTSA, of the 2000 road incidents reported between January and April 2017, motorcyclists and drivers, vehicle defects and pedestrians were identified as the main causes of accidents, while road defects and passengers did not significantly cause traffic accidents. NTSA further attributes the rise of road accidents in Kenya to bad road user behavior such as drink-driving, speeding and low levels of road safety awareness among users (National Transport and Safety Authority, 2015).

2.2 Trend in Road Traffic Incidents

Since independence road transport in Kenya continues to be the predominant mode of transport with much effort also being made on improving the road networks. The economic cost of road crashes is 5.6% of the Gross Domestic Product (GDP); three hundred billion Kenyan shillings. Further statistics for these Road Traffic Incidents (RTIs) conducted in 2013 indicate that 66% of incidents occurred during daytime while 34% occurred during nighttime. Also 60% of the reported RTIs occurred on rural roads and had a higher case fatality of 16%. 40% of the reported RTIs occurred in urban areas and a fatality rate of 11%.In overall human factors were responsible for 85% of all causes. Of all fatalities reported; pedestrians comprised 42%, passengers 38%, drivers 12% and cyclists 8 % (National Transport and Safety Authority, 2015). These road traffic crashes have enormous consequences to the Kenyan society including the health burden, disorienting family structure and affecting productive members of the society. Figure 2.1 shows a comparison of fatalities on Kenyan for the year 2014/2015 roads indicating that most fatalities involve pedestrians.

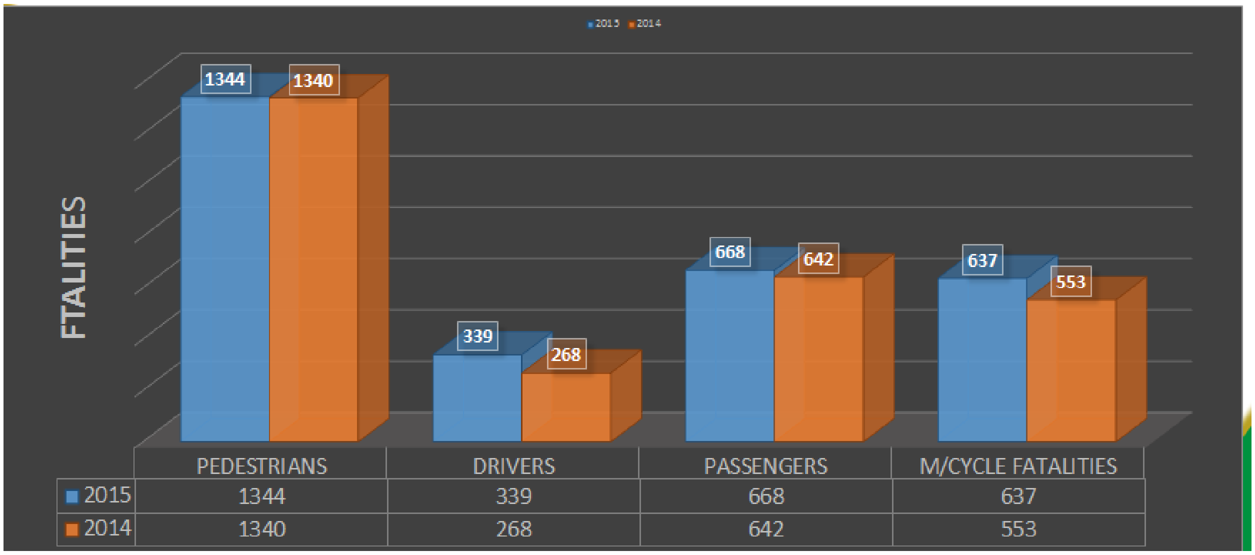

Figure 2.1 Comparison of fatalities financial year 2014/2015

(Source: National Transport and Safety Authority, 2016)

2.3 Road Transportation in Nairobi County

Nairobi, being the capital City of Kenya is an important hub for Eastern and Central Africa. It also hosts many government ministries, businesses and is a hub for many economic activities. The population of Nairobi city has experienced a significant growth since the year 1963 when it was at 350,000 (Mundia, 2017) to around 3.3 million in the year 2011. The City population continues to grow at the rate of 4.0% per year. Mundia (2017) further states that Nairobi City had 1, 214.5 kilometers of roads in 2006 of which 972 kilometers were paved while the remaining 178 were earth gravel roads. Continued urban sprawl has increased transportation demand in the city with a large population relying on public transport system, which has been marked with unstructured operations due to poor monitoring. Many organizations in Nairobi City structure their work hours between 8.00 a.m. and 5.00 p.m. This system has caused a scramble among the people trying to access their workstations or job opportunities within the set working hours. This has led to rapid growth in non-conventional means of public transport comprising of minibuses, taxis and more recently by commercial motorcycles. Most of these means are notorious for flouting traffic rules resulting to externalities such as accidents, congestion and corruption. Kenya Institute for Public Policy Research and Analysis (KIPPRA, 2017) estimates that in 2008 there were around 300,000 vehicles in Nairobi which has grown to around 2 million in 2013. A 2007 report by Ministry of Roads and Public works indicate that although only 15.3% of commuters in Nairobi use private cars. They account for 36% of vehicles on the city roads. Another 29% of commuters use public vehicles (matatus), which account for 27% of the vehicles on the roads (KIPPRA, 2017). The distance of the road network in Nairobi as in 2012 was about 58,000 kilometers (Kenya Roads Board, 2014), with approximately 1.2 million vehicles using these roads.

2.4 Road Transport Agencies

The Kenya Roads Act (2007) established three new agencies in charge of road infrastructure management, rehabilitation, maintenance and development. These are the Kenya National Highways Authority (KeNHA), Kenya Rural Roads Authority (KeRRA) and the Kenya Urban Roads Authority (KURA). The Kenya Roads Board (KRB) solicits and distributes funds for development of the road infrastructure while the Nairobi Traffic Department and City Inspectorate Department are responsible for enforcement of the traffic Act and County Government regulations. The National Transport and safety Authority (NTSA) whose vision is geared towards a sustainable, safe road transport system with zero crashes has been instrumental in the efforts to reduce road accidents and is responsible for registration, licensing and road safety. This informs us that the government appreciates the role of road transport and research geared towards addressing the influences of RTAs, which will foster a sound policy formulation framework.

2.5 GIS in Road Safety and Incident Reporting

GIS refers to a set of tools or database system or a set of functions that provide the capability of storing and maintaining large datasets of spatial and tabular information. Road incidents occur at a particular location and thus have a grid reference. This establishes the link between GIS and road incident data. GIS provides many analytical capabilities and features that can help improve upon the traditional methods of incident processing and analysis including database query and display, spatial and network analysis and interoperability of GIS with other systems. For example, many highway agencies have been using GIS to analyze road incidents data and identify the locations with high reported road incidents. Using such a system, a user can merge incident and roadway data, perform sliding and spot analysis and calculate frequency and rate of incidents.

2.5.1 Nduru Road Safety App

Nduru is Kenyan mobile application that is available for Android devices and which manages incidents related to road safety. Nduru is Swahili for ‘scream’. Through the application, users can flag situations that could potentially lead to an accident as well as report corrupt police officers, send complaints, view first aid guide among others. It gives road users the ability to take charge of their safety through the mobile phone. The application also has a speedometer to automatically detect over-speeding and report to the police if need be. It also has a video and audio tutorial that enables people to administer first aid and access crucial information such as emergency services and information on black spots. The application is available on the Nokia Platform, SMS version as well as Android version (Jackson, 2012).

The mobile application has no location or mapping capabilities that would allow users to visualize road incidents information. A person reporting the incident has no way of following up the resolution of incident.

2.5.2 GIS-Alas (GIS-based Accident Location and Analysis System)

It is a locationally referenced highway accident database. This was created by the Iowa Department of Transportation (DOT), law enforcement agencies, IOWA State University and several other entities. This system produces query and reporting functions in a graphical environment that facilitates advanced spatial query and display capabilities. It provides graphical data access enabling the user to view and select desired locations on the network. User query operations results can be shown on both tabular form and map, thereby creating reports that can be easily interpreted from query results and promoting the analysis of accident patterns and causal relationships (Souleyrette, Strauss, Pawlovich, & Estochen, 1998).

Some of limitations of the system include time and effort required for system conversion, required purchase and training for the GIS software, slower data access times and accounting for updates

2.5.3 Ma3Route

Ma3Route is a mobile/web platform that crowd-sources for transport data and provides users with information on traffic, matatu directions and driving reports. The major aim of Ma3Route is to make travelling easier in developing countries by democratizing timely transport information.

This informs city planning and transport through provision of transport data and trend analysis (ma3route,2016).

While users can report road incidents data using this application, the data provided in mostly traffic feed on different roads in Nairobi City. This data is also generally unstructured and sometimes lacks positional accuracy. Users also lack a clear way of tracking the resolution of a reported incident.

2.6 Web Mapping

Web mapping is the process of designing, implementing, generating and delivering maps on the World Wide Web. Websites are becoming increasingly popular especially with distant passage of information; one very effective way of making map information available to a group of non-technical end uses is through a webpage. According to Mitchell (2005), there are two broad kinds of web map applications; static and dynamic. Static maps are maps displayed as an image on a web page. On the other hand with dynamic mapping a user can interact with the map such as by querying of map layers and zooming. Such maps are much more complex and require programming skills. Web maps can deliver up to date information if they are generated from databases. PostGIS is a freely available open source-source spatial database extender for the popular PostgreSQL database management system and which can help to answer questions that could not be answered with a mere relational database. Its feature set equals or surpasses proprietary alternatives, allowing you to create locational aware queries and features with just a few lines of SQL code from PostgreSQL. PostgreSQL is a powerful, open source object-relational database system used with PostGIS to develop the spatial database.

2.6.1 Web Mapping Application Programming Interfaces.

There are many services that can be used to create web mapping applications, some are commercially licensed while others are Free and Open Source Systems. Typically, these services offer satellite imagery, street map and route planning and real time traffic conditions. Most of these services also offer an API to allow developers integrate their applications with these mapping services. For example one can embed a map on their website using minimal programming skills. Additionally most vendors have also introduced mobile friendly services that are intended to run on mobile devices. These mapping services that available include Leaflet, Modest Maps, Poly Maps and Google Maps. Google API offers a feature rich interface which can be used for web mapping. The maps can also be customized for a particular client. The maps are also continuously updated by Google and also thousands of volunteers.

2.7 Volunteer Geographic Information

The advancement of technologies nowadays enables participation by nonprofessionals, known as volunteers to participate in producing, sharing and consuming Geographic information. Volunteer Geographic Information (VGI) is a special case of the larger Web phenomenon known as user-generated content. It is the harnessing of tools to create, assemble and disseminate geographic data provided voluntarily by individuals. The growing interest in compiling and editing georeferenced data has manifested itself in the growth of volunteered geographic information systems such as OpenStreetMap and Ushahidi platform. VGI or locational crowdsourcing, a georeferenced type of citizen science is a growing area in the field of information science and which can be used to mobilize civic engagement. It is an assertive method of collecting geospatial information as opposed to the authoritative method employed by government agencies and private industry. The term was coined by geographer Michael F. Goodchild who, in exploring the world of user-generated content on the web, noted that "a remarkable phenomenon ... has become evident in recent months: the widespread engagement of large numbers of private citizens, often with little in the way of formal qualifications, in the creation of geographic information, a function that for centuries has been reserved to official agencies" (Goodchild, Citizens as sensors: the world of volunteered geography, 2007). Most of these people are mostly untrained and their actions are voluntary. In his keynote speech, "It's About Time: The Temporal Dimension in VGI," Professor Goodchild highlighted the importance of time in VGI to create a reliable and accurate representation of different occurrences and when in a particular situation. Traditionally maps have mostly emphasized the aspects of geography which are static. "You did not make maps of things that were changing," he said. "But people move, rivers change course, events occur, and emergencies develop. Geography is dynamic." Concerning quality, VGI can be said to be the most current data source. However, VGI compared to traditional authoritative data sources can be said to have poor positional accuracy and overall veracity. However, this assertion might not be entirely accurate and can be debated using Linus's law. This law states that the more people involved and watching over a project, the more likely errors can be spotted and fixed early (Howard and Leblanc, 2003). Another argument in support of VGI is the argument that near things are more related than distant things and therefore citizen contributions will be more accurate for places which these people understands. VGI has also played a key part in emergencies, for example, in the aftermath of Haiti earthquake in 2010; two rapid VGI projects helped coordinate disaster response. These included OpenStreetMap (OSM) and the Ushahidi project. OSM was able to create a digital street map of Port-au-Prince. This also included other places in Haiti very rapidly using fine-resolution imagery to trace vector maps of streets and other features while the Ushahidi project was able to post appeals for help.

2.7.1 VGI and Base Mapping Coverage

OSM is a global mapping application created by volunteers and is arguably one of the most successful applications created by volunteers. It is based on Wikipedia's peers' production model and provides free and downloadable coverage for locations and also geometries of specific topographic features. In some cases, data generated through VGI is more accurate and detailed and accurate than the data produced by authoritative sources.

2.7.2 VGI and Emergency Reporting

VGI is popular in what has been described as crisis mapping, emergency reporting and humanitarian aid. Within this humanitarian assistance, the power of the crowd can be clearly understood as a mass of people are converging to tackle a set of tasks. For instance, during the post-election violence that ensued in Kenya in 2007 after the presidential elections, local activists developed the Ushahidi platform to coordinate response to the crisis as well as enable citizens to report incidents of violence. This was through the online map as well as Short Messaging Service (SMS). From the initial development the platform has been made publicly available and meets different needs for crisis mapping (Crowe, 2012).

2.8 Systems Development Life Cycle (SDLC)

SDLC is a framework used by software industry to design, develop and test high quality softwares. It consists of detailed plan describing how to develop, maintain, replace and enhance software. A typical SDLC consists of the following stages;

I. Planning and Requirement Analysis. This part involves gathering requirements and reviewing the requirements. This stage also involves assigning tasks and also negotiating a timeline. There should be a strong idea of what deficiencies exist in the current structure and the goals for the new approach. A feasibility study is also done to determine if the goals of the plan can be accomplished.

II. Defining Requirements. This is the stage through which all approved requirements are documented. This stage defines user requirements, identifies needed features and other documentation. This is the buffer phase between gathering requirements and acting on them.

III. Designing and Development. This is the planning stage for all the technical aspects of a project. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture. An approved design is the catalyst for authorizing development for the new system. During this phase code is also written and construction and fine tuning of technical and physical configurations.

IV. Testing and deployment. During this stage users are brought in to test before deployment to identify areas of concern or improvements. The system is finally put to production environment to conduct business.

There are numerous benefits for deploying a system development life cycle that include the ability to pre-plan and analyze structured phases and goals. Developers understand what they should build and why it should be built. Everyone also understands the costs and the resources required. There are various SDLC models including waterfall method, agile method, iterative model and spiral model. Of all these models, the waterfall model is the oldest and the most straight-forward, it is simple to understand and use. It is also best suited for small projects where there are no uncertain requirements. With this method one phase is fully finished before the next phase can begin. The sequential phases in the waterfall method were;

1. **Requirement Gathering and analysis** − All the possible requirements of the system were captured in this phase and documented.
2. **System Design** − The requirement specifications from first phase were studied in this phase and the system design was prepared. This system design also specified hardware and system requirements and helps in defining the overall system architecture.
3. **Implementation** − With inputs from the system design, the system was first developed in small programs which were then integrated in the next phase. Each unit was developed and tested for its functionality.

4. **Integration and Testing** − All the units developed in the implementation phase were integrated into a system after testing. Post integration the entire system was tested for any faults and failures.
5. **Deployment of system** − Once the functional and non-functional testing is done; the product was deployed in the test environment.
6. **Maintenance** − There was the phase where issues which come up in the test environment were fixed. Figure 2.2 describes the phased approach of the water methodology outlining the various activities that happen at each phase.

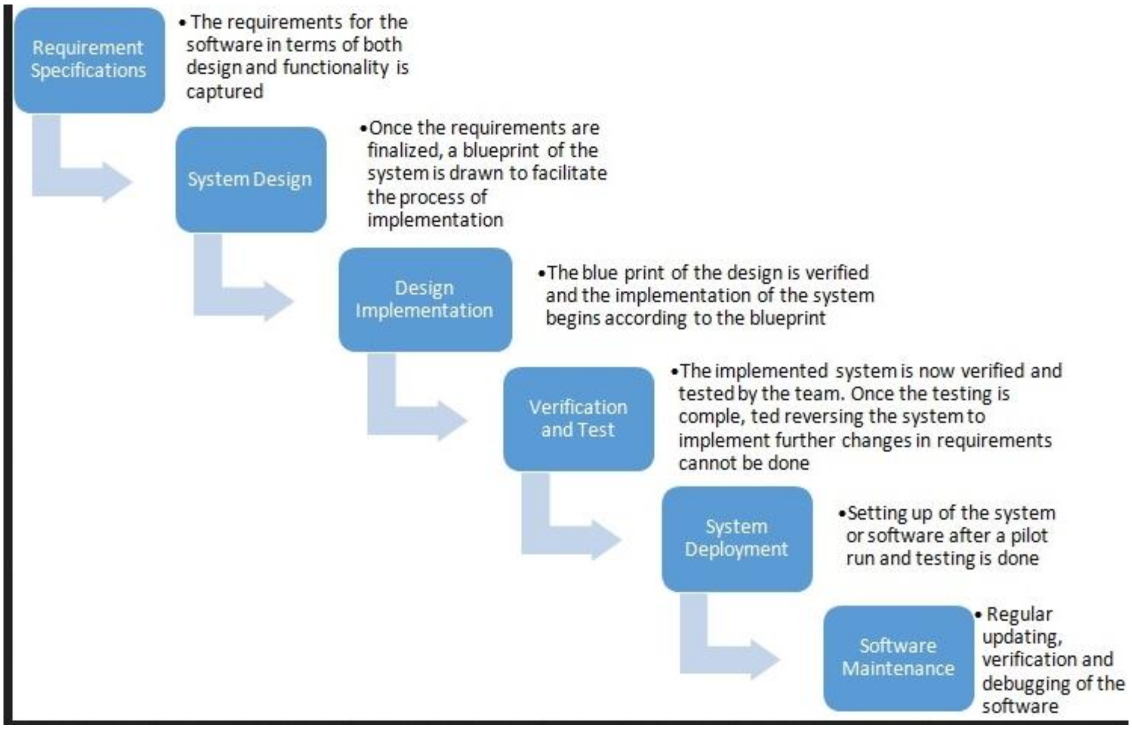

Figure 2.2 Waterfall Methodology

(Source:Waterfall Methodology in Project Management, 2014)

2.8 Summary

The literature review discussed in this chapter shows that road safety is a growing concern for many countries around the world. Though there have been improvements like enacting new policies and legislations, a lot still needs to be done. Effective road safety initiatives require efficient management of all data related to road incidents. Although there has been advancement of web technologies such as Web 2.0 and VGI which have also fuelled the growth of web mapping, there are still no proper and simple ways of reporting and sharing roads incidents data.

Most of the existing road incidents recording applications either have no way of tracking the resolution of a reported incident or the data for such incidents is not made available to the public.

Chapter 3 METHODOLOGY AND MATERIALS

This chapter discusses

I. The data, materials and the procedure adopted for the study
II. Methodology of the study
III. Feasibility studies conducted
IV. Implementation and evaluation criteria

3.1 Software Development Life Cycle Methodology

The software methodology used in designing and implementing this system is the waterfall method as it allows for departmentalization and control and development proceeded phase by phase.

3.2 Feasibility Study

A feasibility study was carried out to determine if the system is

I. Technically feasible
II. Is feasible within the estimated budget

This was carried out by extensive reviews of relevant literature on existing systems related to road incidents data reporting and analysis. The reviewed application as outlined in the literature review section included Nduru mobile application and m3route mobile application. Whereas these applications lack the requirements like checking the progress of a reported incident and viewing reported incidents nearby, they run on free and open platforms like the Googles Android Platform. The study was also carried out to analyze and evaluate the impact of the proposed system.

3.3 Business Study

Research Design

In this study a Descriptive research and Qualitative Research was done.

I. Descriptive Research. This was used to describe the characteristics of a population or phenomenon being studied. It depicts the participants in an accurate way. This provided some good understanding between roads incidents and the ability of the general public to report these incidents.

II. Qualitative Research. This was used to gain an understanding on views of the people in the context of the study area. This was so as to assess the user satisfaction in fulfilling the functional and nonfunctional requirements of the system.

a) Location of the Study

The study was done in the areas on the University of Nairobi Main Campus and environs within Nairobi County focusing on Waiyaki Way. This area was primarily chosen due to ease of accessibility from the University and also convenience for the researcher.

b) Target Population

The target population was from the general public within the location of the study area including motorists and pedestrians. The total number of respondents that was used in this study was fifty. This respondent were picked randomly based on their willingness to answer the interview questions and was majorly the pedestrians along Waiyaki Way in Nairobi.

c) Sampling strategy

Strategy was the plan set forth to ensure that the sample used in the research represented the population from which the sample is drawn. Purposive sampling also known as judgmental, selective, or subjective sampling was used in this study. This type of sampling was chosen as it can be very useful in situations when you need to reach a targeted sample quickly, and where sampling for proportionality is not the main concern.

d) Data Collection

The two main methods that were used in data collection were surveys by interviews and questionnaires. Data collection involved visits to the roads and streets and roads within the study areas and carrying out the interviews. The questionnaire was prepared in excel format and distributed through email and Google Docs. The data collected was aggregated by google in an excel format with the summarized responses. Interviews were chosen as they are quick and help keep focus and provide opportunities for follow-up. Questionnaires on the other hand were chosen as they allow for collection of large amounts of information and the results can be easily quantified.

3.4 System Design, Architecture and Implementation

This section discusses the system architecture of both the front-end and back-end sides of the application outlining the various requirements for the implementation. The architectural diagrams show a presentation of the various components of the system and their relationship in building the system. The system architecture of the road incident recording and analysis was based on the requirements gathered.

3.4.1 Data Modelling

This stage of the application development was required so as to define and analyze data requirements needed to support the processes of the application that was implemented. The top down approach proceeded from the conceptual modelling which was then translated to logical modelling. This was then translated to the physical data mode. Table 3.1 captures the key entities that were used for system data modelling. Figure 3.1 shows an Entity Relational Diagram that represents the entities involved and their logical structures.

Table 3.1 Key System Entities

Incident	This entity represents a unique incident that has been reported and stores all the incident attributes
Incident Types	The classifies the various incident types such as car-jackings, accidents, speeding etc.
Vehicle Types	Used to classify the vehicle that are involved in incidents
Users	This entity stores all the registered user details as well as different users access rights
Distresses	Used to store all distress calls that are reported

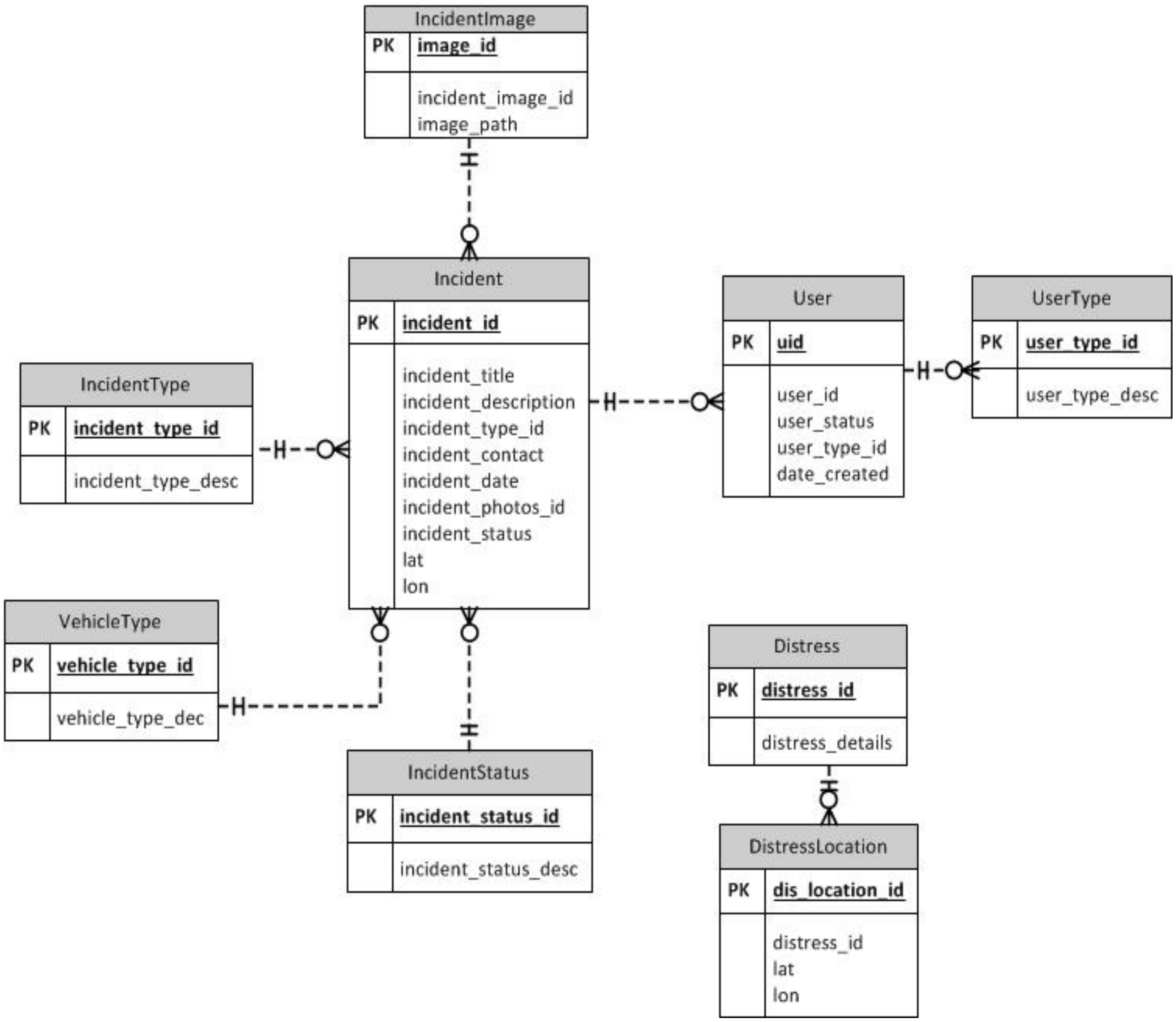

Figure 3.1 System Entity Relationship Diagram

3.4.2 Application Implementation

Three-tier Architecture software architecture was used in this project. The user interface, the functional logic and data access layer were developed and maintained separately as different modules. This makes it easy to continually change the application. The frontend represents the mobile application through which user interacts with the system. The business logic represents the application logic that runs on the webserver while the data is stored in a geo-database. Figure 3.2 shows how the Three-tier Architecture was used in this study.

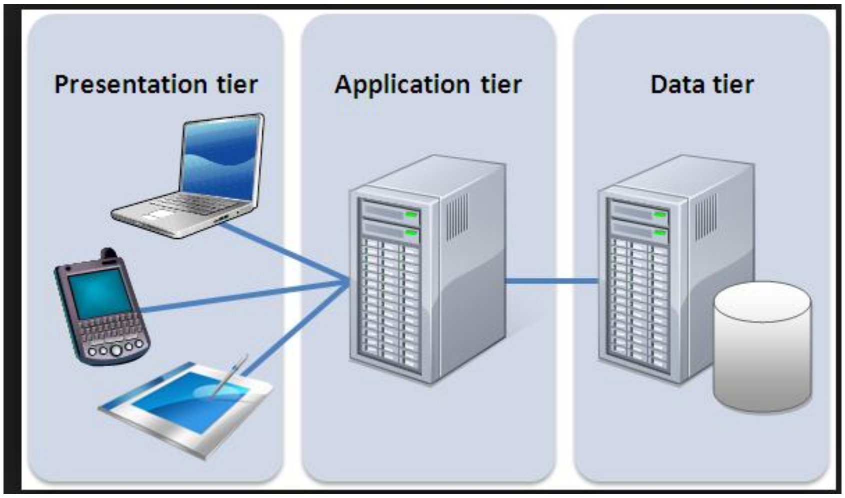

Figure 3.2 system Three-tier architecture

The different components of the incident reporting system are outlined below:

I. Mobile Application. This was developed and compiled for the Android platform. The mobile app was tested with an android device and an emulator. The application was tested on Android 4.4. The mobile application used JSON web services that were developed using PHP and the bridge between the interface and the incidents Geo-database. Android was chosen due to its flexibility and a wide user base. Sample source code for the mobile application is provided in Appendix A.

II. Web Application. The web application was developed using PHP and hosted online with an Apache webserver. PHP was chosen due to its popularity among both developers and hosting companies.

III. Geo-Database. The incidents database was designed using PostgreSQL and enable spatially using the PostGIS plugin. PostGIS was chosen due to its flexibility in handling geospatial data and growing popularity.

a) System Process Flow

The front end user is the road user in possession of a mobile phone with the mobile road incident reporting application installed. The incident report is submitted from within the application and contains details of the incident. The report is stored in the backend geodatabase server. This access could, for example, be shared with the police. Front-end users also have the option to view road incidents have been posted by other users. This acts as important information to them especially if there is an incident that may concern them while on the road such as a road accident. The backend user has a more refined view of the reported files. The user can analyze different reported incidents like over speeding public service vehicles, carjacking incidents and road accidents. These are all monitored on a map in the backend user's personal computer since the font user and backend user share the same source of data. Figure 3.3 shows an overview of the general architecture of the system, it shows the frontend application is used to capture the incidents while the back end module is used for display and analysis.

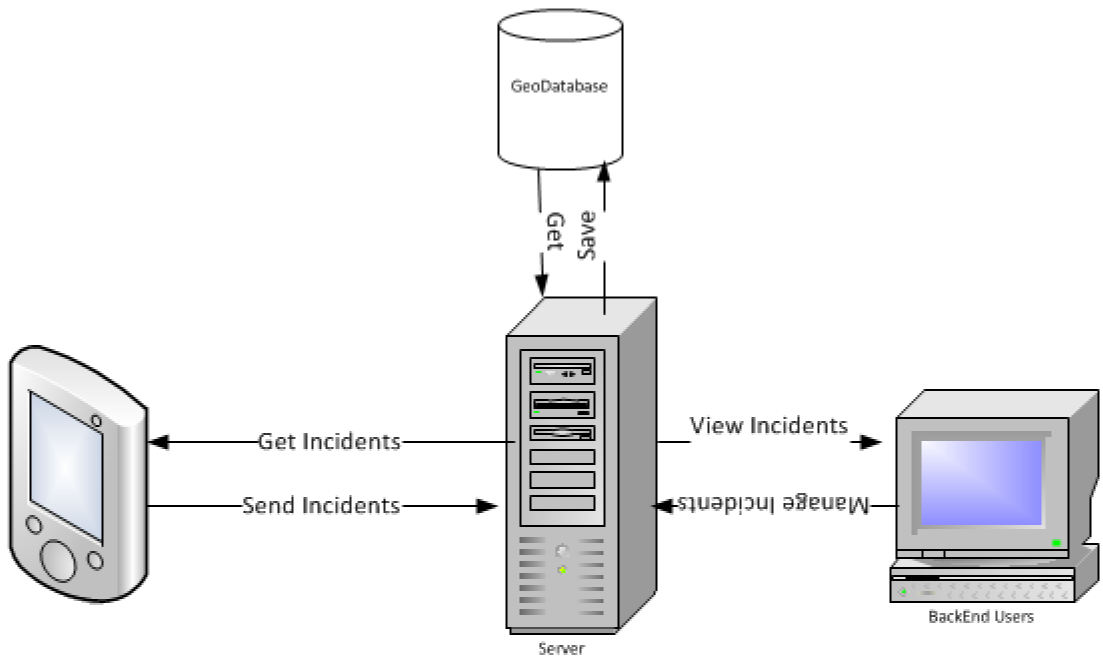

Figure 3.3 Overview of the system Architecture

b) Use Case Model

This was used to graphically capture the interactions among the elements that were used in this system as well as identify and organize system requirements. Each case provides some observable and valuable result to the stakeholders of the system. Figure 3.4 shows a use case model for the system. The figure illustrates the various scenarios when using the system. The frontend users can report incident, send a distress call as well view incidents that have been reported nearby at his current location. The backend users can track the progress of an incident as well as manage users and incidents.

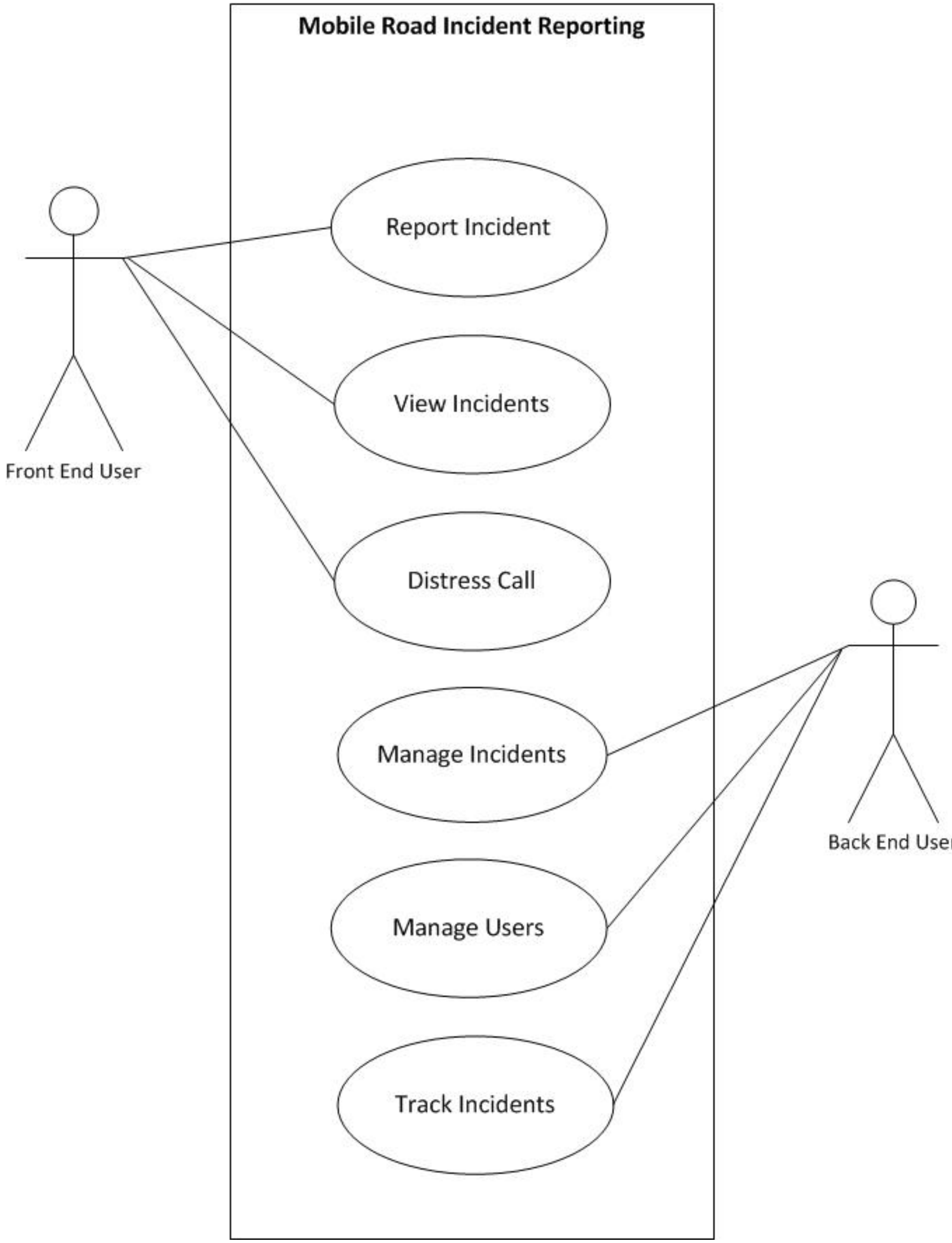

Figure 3.4 System Use Case Model

c) Functional Requirements

The functional requirements describe the core functionality of the system

I. **Registration**. A user after accessing the application and having successfully installed it submits details that will be used to create their user account. A user database is maintained and thus when one registers the user details are confirmed. In addition there are special users for admins who can register on the backend and manage the system.

II. **Login**. The user should be able to login to the application at different instances and should be provided with an option of whether to stay signed in to the application or login only when they need to access the application or data required.

III. **Send Road Incident Report.** After the user is logged in the system, he is now able to post road incidents via the file road incident page of the front-end application on the mobile phone. This functionality allows the user to add a GPS location to the posted report so as to track the location of the incident. The user is also free to take pictures of the incident i.e. if it is a road accident and finally post the report.

IV. **View Road Incidents.** The user can view road incidents that have been reported by other users. Some may include images that have been posted by other users.

V. **Distress Call.** This functionality allows the user to send bulk SMSs to important contacts in his phonebook at the touch of a button. It might be applicable for example in the case of a carjacking and no enough time for a call. This also sends the GPS location of the victim.

VI. **View and Track reported incidents.** This allows for the user to view different categories of reported incidents.

VII. **Clear Reports.** Once an incident has been resolved, an admin user can mark the incident as resolved so that it will no longer appear as an incident to other users.

d) Non-Functional Requirements

Nonfunctional requirements refer to the general behavior of the system and how it is expected to function.

I. **Security.** The system allows for authentication to ensure that incidents data is not altered by unauthorized users both on the frontend and the backend.

II. **Availability**. The system runs on different components which can be scaled to handle any failures. In addition the system has an availability of 80% to allow users to send and access the road incident information upon request.

III. **Accessibility**. The system is accessed via the internet and utilized the common HTTP protocol.

IV. **Maintainability**. The system is designed using the three-tier architecture which promotes loose coupling and maintainability.

V. **Scalability**. The system can handle a growing amount of work as different components can be scaled either horizontally or vertically.

e) User Interface Design

This part details the implementation of interfaces that are used capture the incidents. In addition the screenshots are captured to illustrate the various system functionalities.

I. Submit Road Incident Report

The user can either submit the road incident as a logged in user or as a guest user on the mobile phone. The functionality automatically attaches the GPS coordinates to the data so as to be able to tack the location the incident. The user can also upload an image of the incident either from the phone or take a new photo using the camera and finally submit the data. Figure 3.5 shows the screen that is filled by the user when he is reporting an incident. Figure 3.6 shows the options for the user to capture an image to be attached to a reported incident.

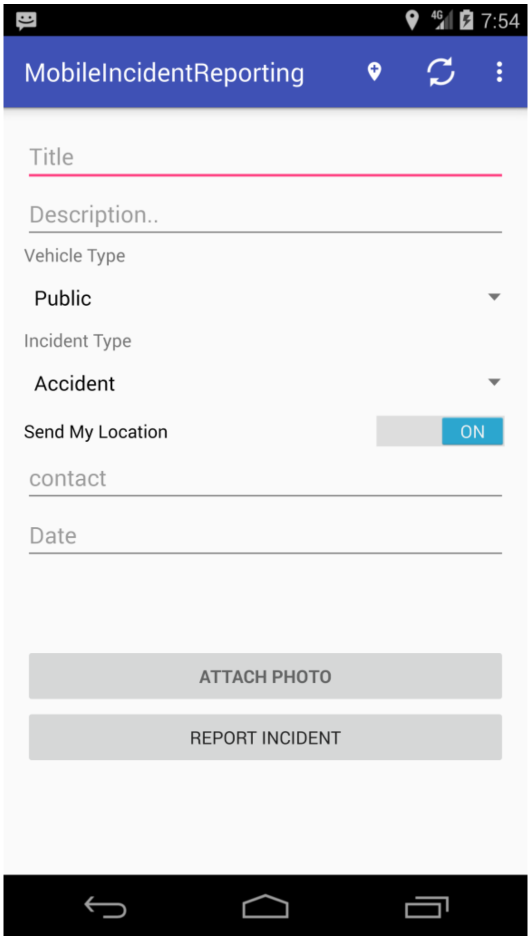

Figure 3.5 Submit incident user form

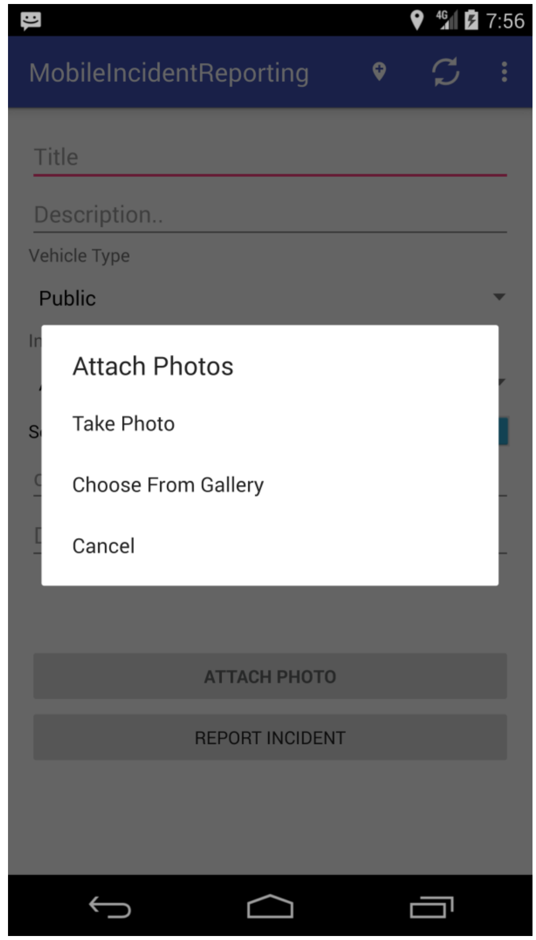

Figure 3.6 Select image source user form

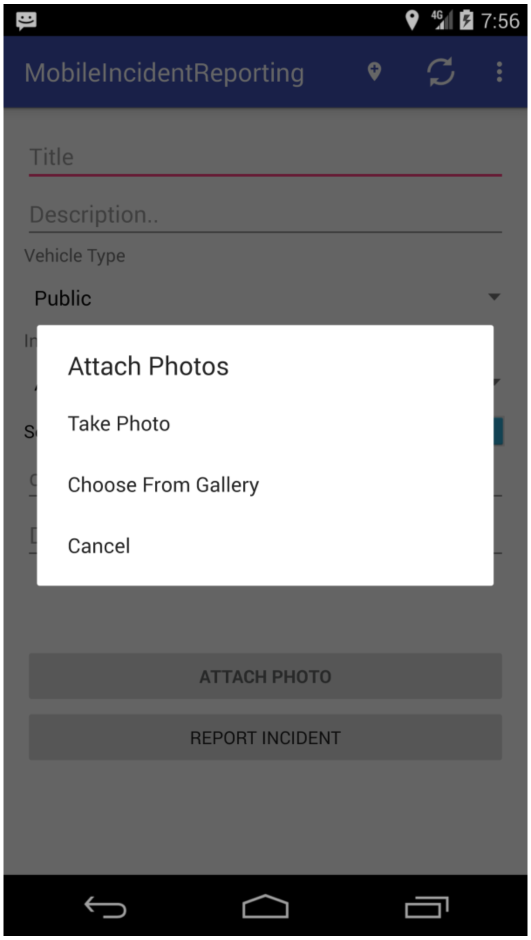

Figure 3.7 Select image source user form

II. Distress Call

This functionality allows the user to send an emergency alert at the touch of a button. For instance, in the case of a carjacking and there is not enough time to make a call. This functionality also captures your GPS location so that the location can be used for tracking purposes. Figure 3.7 shows the distress call button that the user can click to send an emergency SMS giving his location.

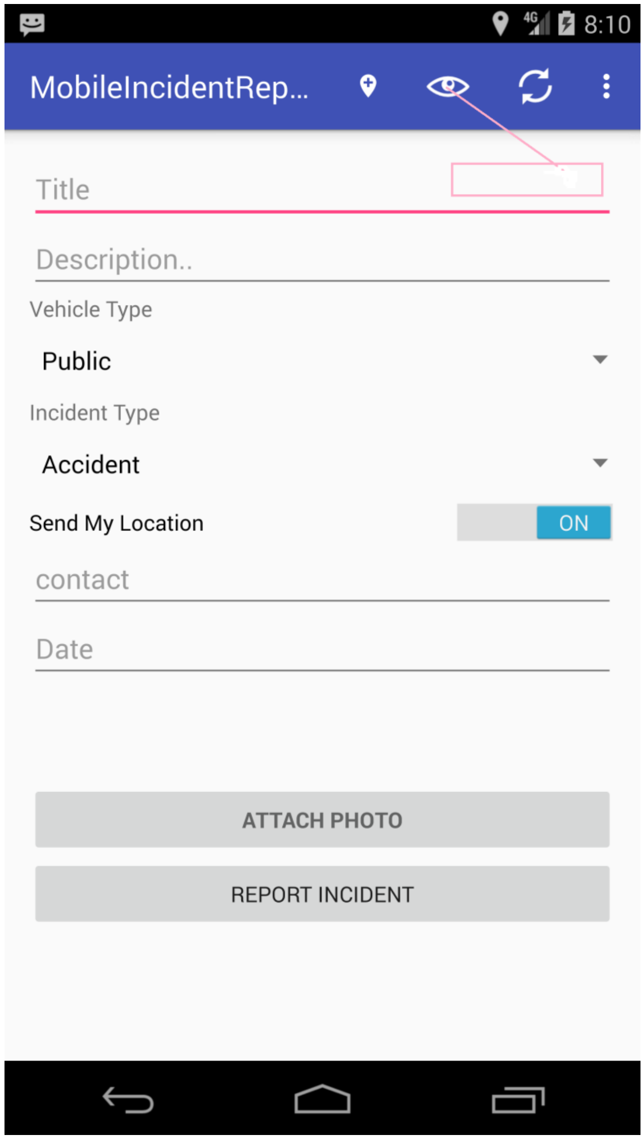

Figure 3.8 Distress call button interface

III. View reported incidents

This functionality is useful especially when travelling or using a particular route and automatically captures the user location and displays the nearby incidents that have been reported for that particular location. Figure 3.8 shows the reported incidents screen that shows the current location and all the reported incidents that have been reported near that location.

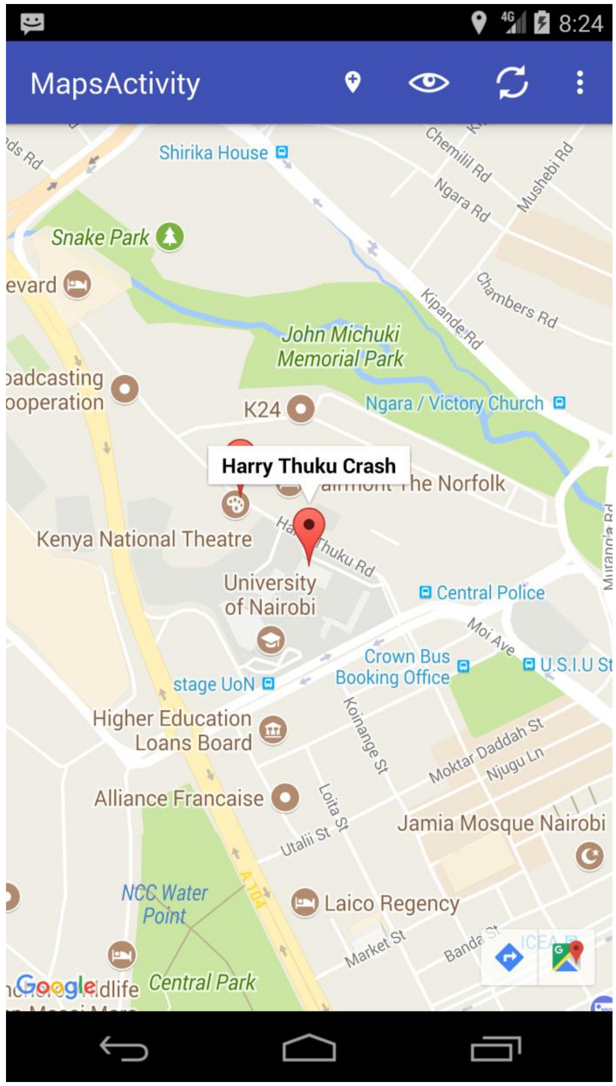

Figure 3.9 View incidents user interface

IV. View and Track Reported Incidents

This is the main screen that allows users to view reported road incidents on the fly. The data is represented on Google maps. All the reported incidents are reported on this map. The other can also specify different filter criteria so as to view incidents that belong to a particular category. Figure 3.9 shows the track incidents main page where the user can use various filters so as to focus on a particular area.

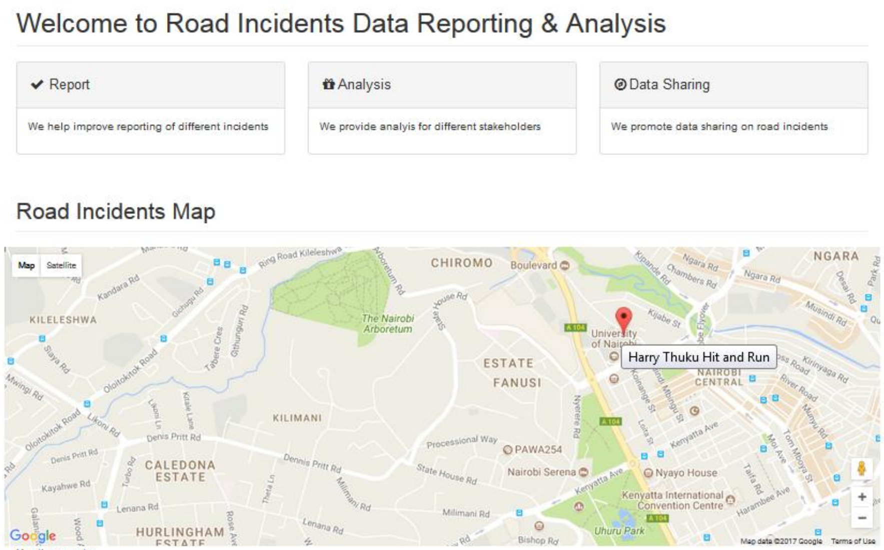

Figure 3.10 Track Reported Incidents user interface

3.4.3 System Evaluation and Testing

The prototype application underwent the following tests

1. User acceptance tests to measure user satisfaction
2. Functional tests to ensure the application meets the user requirements
3. Unit tests to test the different code modules.

System testing was done to rate different aspects of the system including functionality, the graphical user interface, user-friendliness and whether users easily understood the general concept of the system. The developed application was tested by the case study users. Majority of the users were students close to the University Way and who already had phones running on Android. The responses from these users formed the bases for evaluation of the system. The users were selected because they are a representation of the target users of the application. A two-step approach was designed in order to obtain a proper evaluation of the system. The first part consisted of a brief description of the system and functions of the system. This also contained the various tasks for the user to perform. The second part was a qualitative evaluation and consisted on both closed and open ended questions about the tasks that were performed by the system.

Chapter 4 ANALYSIS OF RESULTS

The results of user evaluation and assessment were analyzed and used as a basis to determine if application is usable. These results show that the target users are ready to use the application as it meets some unique needs for road users. The results also indicate that the application is easy to use and meets user needs.

4.1 Results and Findings

Different metrics were used during the testing of the system. These included

I. Easy use of use and system friendliness
II. Completeness of tasks
III. Appearance of the system.

Testing of the application also sought to identify opportunities for improvement and functionalities that might be added to the application to improve usability as well as meet new requirements. From the user evaluations during testing, it is clear that the application is useful and that the users are actually willing to use it. Based on the responses, the application is considered user friendly and easy to use. The system discussed here is primarily used for capturing road accidents. Most of the existing systems that were reviewed were either built for in-house or organizational use. They are also governed by various authorities and most of them the reporting module is often not open to the public.

4.2 Evaluation of Results

Out of the 20 questionnaires that were handed out to selected people, 17 were filled correctly and returned. This section discusses the results from this evaluation. The questionnaires had questions related to the impact of the application to road safety, users who were willing to use the application and system user friendliness and ease of use. Table 4.1 shows the questions that were prepared based on these parameters.

Table 4.1 System Evaluation Questionnaire

Parameter	Question	Answers Space
Impact of the System	Do you think the application would have an impact?	Yes,No
Impact of the System on Road Safety	Do you think the application would have an impact on Road Safety?	Yes,No
User willingness to use the system	Would you be willing to use such as application?	Yes,No
System Friendliness and ease of use	Was the system easy to use and achieve desired result	Excellent,Good,Fair,Bad
	How do you rate the graphical user interface of the system?	Excellent, Good,Fair,Bad
	How do you rate the system in terms of user friendliness?	Excellent, Good,Fair,Bad

1. Impact the application might have to authorities concerned

Figure 4.1 shows a pie chart detailing the respondents' perception on the impact of mobile road incident reporting system.

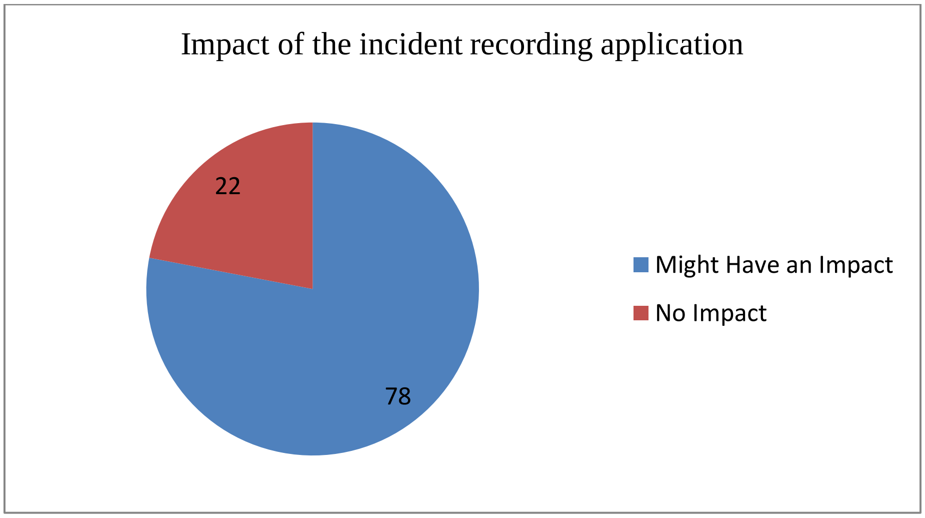

Figure 4.1 Evaluation of impact of the incident recording application

23% of the users indicated that the application will not have any impact as far as the road incident reporting is concerned. They stated the following reasons

I. There are already existing mediums to report such incidents like Facebook and twitter
II. Getting traction among users will be a big challenge
III. The system lacks a supporting policy framework

A majority of the users felt that the system will have some impact for the following reasons;

I. The application is specific enough and only focusses on road incident
II. Data can be mined from the central geo database
III. Images will serve as proof of the reports
IV. The application is simple and usable

2. Impact of system on Road Safety

Figure 4.2 shows a pie chart indicating the impact of application on road safety.

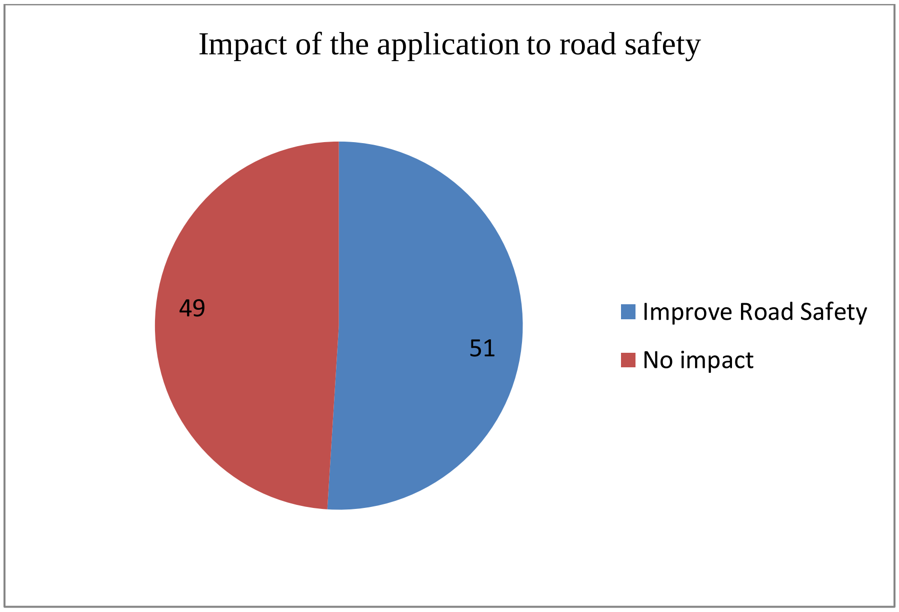

Figure 4.2 Evaluation of impact of application to road safety

51% indicated that the application will improve road safety. They had the following reasons

I. Incident response time would improve as the relevant authorities would act faster as they get information on real time.
II. Data collected can help prevent future occurrences of harmful incidents.

49% felt that it would not improve road safety. They cited the following reasons

I. Application runs only on smart phones which might not be available to some users
II. Users might not use the application as its not feature rich as compared to other platforms such as Facebook and Twitter.
III. The application might fail to function as expected because of various reasons.

3 Users willingness to use the System.

Figure 4.3 shows the percentage of respondents who would actually use the system.

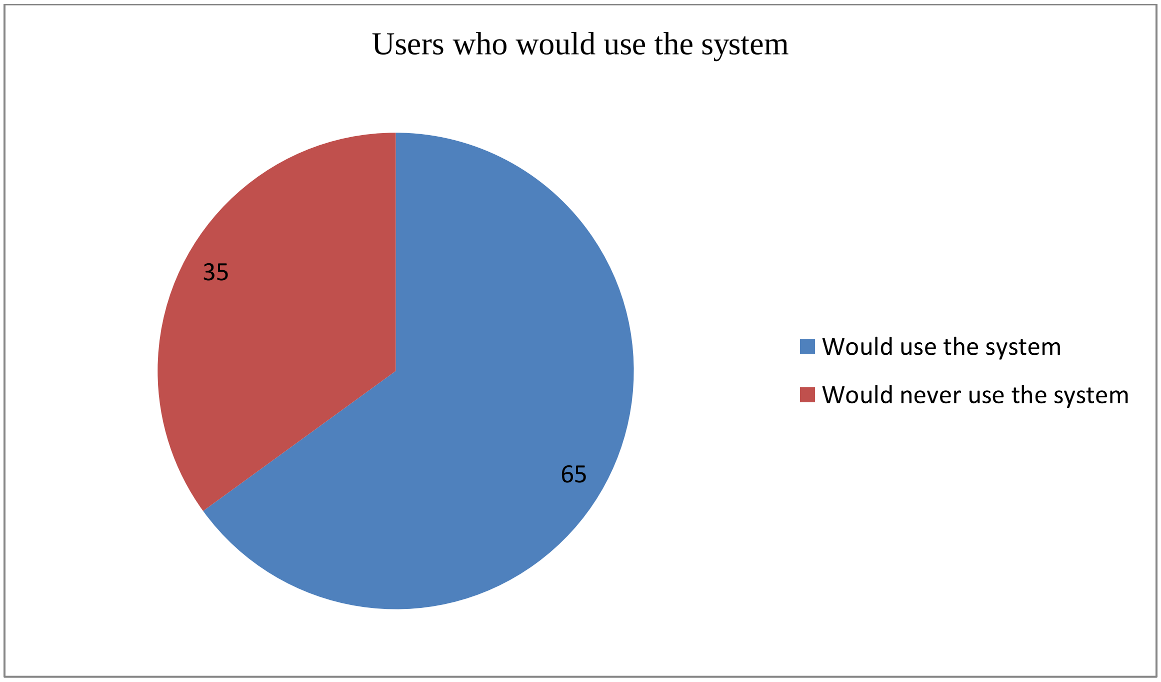

Figure 4.3 Evaluation of users willing to use the system

65% of users were positive that they would use the application but had some reservations

I. If it proves that the authorities can act faster
II. If user privacy is considered at all times. For example the reports data should not be divulged to third parties.
III. More features are added to the system. For instance services such as towing and insurance services.

35% of the users cited the following reasons as to why they would not use the system

I. Not incentives for reporting incidents
II. Availability of other reporting avenues and mediums.

4 Front End User Friendliness and Ease of Use.

Figure 4.4 shows a pie chart showing the ratings for user friendliness and ease of use

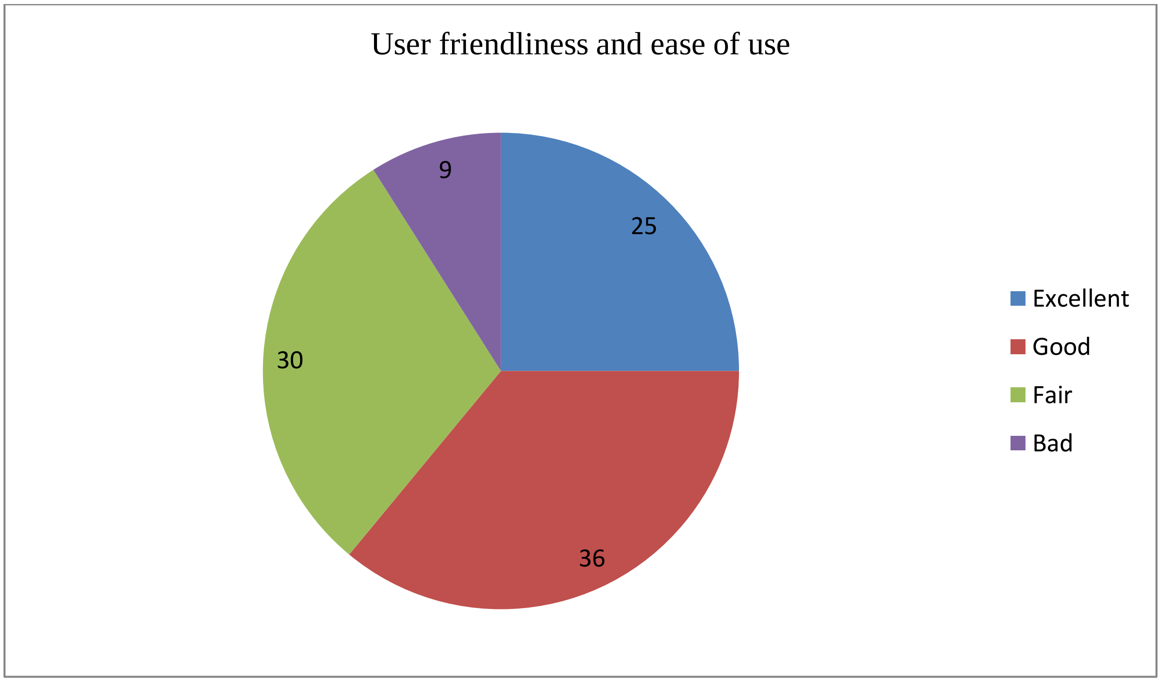

Figure 4.4 Evaluation of System friendliness and ease of use

Most of the respondents indicated that the ease of use was good and that theme should be improved and the icons improved

4.3 User Perception of the System

From the application testing, the following were the perceptions of the user towards mobile road incidents recording and sharing system. The people in support of the initiative had the following reasons

I. The application is specific enough by targeting only road incidents
II. It will help in road safety improvements initiatives

Most of those against the initiative also had various reasons

I. Other users expressed doubts over this system and argued that other mediums exists for reporting incidents such as Facebook and Twitter
II. The application would have to gain enough traction with users in order to start adding value.
III. User privacy might be compromised by disclosing user data to third parties.

4.4 Recommendations to the system by the respondents

The other additions that were made by the respondents improving the application included

I. Improvements on the look and feel of the application.
II. More features should be added so as to provide incentives to users.
III. Addition of feature to enable extraction of reports from the system.

Chapter 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The main aim of this project was to develop a mobile application that can be used to report road incidents and improve the resolution of these incidents and ultimately contribute to existing road safety initiatives. There exists an opportunity to tap into this area in Kenya today because of the increased penetration of smartphones and also the increased cased of road incidents. This challenge of incident reporting was reviewed, while there exists new ways of reporting these incidents such as Facebook and Twitter, road users still lack a simple and proper way of reporting these incidents. The gaps in this reporting process were investigated by reviewing related designs, architectures and models of incident reporting.

The proposed solution to improve this process was to design, develop and test an android based mobile application with the local road users in Nairobi. This application allows users to report any incident that they may witness and that happens on Kenyan roads. From the user testing and evaluations conducted, it can be said that the app fulfills its simplicity and usability requirement. Majority of the users also indicated that the application is generally user friendly and easy to understand.

5.1.1 Challenges in Implementing the Application

GPS was cause of excessive resource usage for some Android apps. This is because the power consumed by the device when using GPS is quite high. The application on the phones registers for fine-grained location updates which use GPS. This caused the application which requested for a GPS update frequently to often cause battery drain.

The reports are submitted by volunteers and witnesses. Since this data comes from the public, any user can decide to report inaccurate incidents. One of the ways to overcome this is to have ways of vetting this information that is submitted. This was a challenge in designing this system and was not explored.

5.2 Recommendations

With the increased number of mobile phones the relevant authorities involved in road safety initiatives should recognize and embrace the use of mobile phones in promoting road safety by improving the reporting and resolution process as well advancing the sharing and dissemination of these data. More initiatives should be also supported and funded to promote the integration of the various existing road systems among various departments and authorities.

At the moment most of the data is maintained in Silos by different stakeholders such as police and the NTSA. The registration of the users was also considered to be a bit consuming for reporting and it was considered that it can be made simpler by using email addresses for Single Sign On (SSO). The application can also be extended so as to run on different platforms and reach a wider audience base. Further research should also be done on how such kind of an application can be incorporated in the Kenyan authorities such as the traffic police.

5.3 Suggestions for Further Work

The application can be improved to cater for the changing needs of users. The existing functionality can be enhanced and updated to meet user needs and thus ensure retention. Further research can also be done on how verification and screening mechanisms can be introduced for these reported incidents to guarantee the accuracy and reliability of the data.

REFERENCES

Andrew, G. (2008). *Enabling a Simulation Capability in the Organisation.* Springer Science & Business Medi.

Bachani, A., Koradia, P., Herbert, H., Mogere, S., Akungah, D., Nyamari, J., et al. (2012). Road Traffic Injuries in Kenya: The Health Burden and Risk Factors in Two Districts. *Traffic Injury Prevention*, 24-30.

Communications Authority of kenya.*Third Quarter Sector Statistics Report For The Financial Year*. http://ca.go.ke: http://ca.go.ke/index.php/statistics,Retrieved June 12, 2017.

Crowe, A. (2012). *Disasters 2.0: The Application of Social Media Systems for Modern Emergency Management.* CRC Press.

GanttPro.Waterfall Methodology in Project Management. https://blog.ganttpro.com/en/waterfall-project-management-methodology-pros-and-cons/,Retrieved May 01, 2017.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 211–221.

Goodchild, M. F. (2009). Neogeography and the nature of geographic expertise. *Journal of Location Based Services,* 82–96.

Howard, M., & Leblanc, D. (2003). *Writing Secure Code.* New York: Microsoft Press.

Jackson, T. *Dangerous driving app launched in Kenya*. http://www.itwebafrica.com/mobile/309-kenya/229396-dangerous-driving-app-launched-in-kenya,Retrieved May 13, 2017.

Kenya Institute for Public Policy Research and Analysis.*KIPPRA Publications*.
http://kippra.or.ke/Resources/kippra-downloads.html,Retrieved May 21, 2017.

Kenya Roads Board.*Kenya Roads Board July 2014 Issue*.
http://krb.go.ke/our-downloads/krb%20newslettter%20issue%2012.pdf,Retrieved April 12, 2017.

ma3route.*About ma3route*.https://www.ma3route.com/,Retrieved June 08, 2017

Manyara, C. (2016). Combating Road Traffic Accidents in Kenya: A Challenge for an Emerging Economy. *Kenya After 50*, 101-122.

Matheka, D., Alkizim, F., Kipsaina, C., and Witte, J. (2015). Road traffic injuries in Kenya: a survey of commercial motorcycle drivers. *Pan African Medical Journal*, 21.

Mitchell, T. (2005). *Web Mapping Illustrated: Using Open Source GIS Toolkits.* O'Reilly Media, Inc.

Mundia, C. (2017). Urban Development in Asia and Africa The Urban Book Series. *Nairobi Metropolitan Area*, 293-317.

National Transport and Safety Authority.*Road Safety Status Report 2015*.

http://www.ntsa.go.ke/2016/Downloads/FINAL%202015%20ROAD%20SAFETY%20STATUS%20REPORT.pptx,Retrieved Jun 20, 2016.

National Transport and Safety Authority.*Final 2015 Road Safety Status Report*. http://www.ntsa.go.ke/2016/Downloads/FINAL%202015%20ROAD%20SAFETY%20STATUS%20REPORT.pptx, Retrieved February 19, 2016.

Odero, W., Khayesi, M., and Heda, P. M. (2003:10). Road traffic injuries in Kenya: magnitude, causes and status of intervention. *Injury Control and*, 53-61.

Souleyrette, R., Strauss, T., Pawlovich, M., and Estochen, B. (1998). GIS-ALAS: The Integration And Analysis Of Highway Crash Data In a GIS Environment. *Integrating the Transportation Business Using GIS. Proceedings of the 1998 Geographic Information Systems for Transportation (GIS-T) Symposium*, (pp. 411-415). Salt Lake City, Utah.

Tomaszewski, B. (2015). *Geographic information systems (GIS) for disaster management.* Boca Raton: CRS Press Taylor & Francis.

World Health Organization.*Global status report on road safety 2015*.http://apps.who.int/iris/bitstream/10665/189242/1/9789241565066_eng.pdf?ua=1,Retrieved June 10, 2016.

APPENDIX: Sample source code

1. Report Incident

```
package com.commigy.mobirep.mobileincidentreporting;

import android.app.DatePickerDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.content.FileProvider;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Base64;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
```

```java
import com.android.volley.toolbox.StringRequest;
import com.commigy.mobirep.mobileincidentreporting.utils.AppSingleton;
import com.commigy.mobirep.mobileincidentreporting.utils.GPSTracker;

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

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.net.URI;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import static android.icu.lang.UCharacter.GraphemeClusterBreak.T;
import static android.media.CamcorderProfile.get;
import static com.commigy.mobirep.mobileincidentreporting.R.id.imageView;

/**
 * A rport screen that offers incident capture.
 */
public class MainActivity extends AppCompatActivity {
    /**
     * Id to identity READ_CONTACTS permission request.
     */
    private static final int REQUEST_READ_CONTACTS = 0;
    /**
     * Request queue to hold all the requests
     */

    // UI references.
    private TextView titleEditText;
    private EditText descriptionEditText;
    private Spinner vehicleTypeSpinner;
    private Spinner incidentTypeSpinner;
    private Switch sendMyLocationSwitch;
    private EditText contactEditText;
    private EditText dateEditText;
    private View mProgressView;
    private View mLoginFormView;
```

```java
ProgressDialog progressDialog;
private Button mReportIncidentButton;
private Button mAttachPhotosButton;
private ImageView imageView;
private static final String ACTIVITY_TAG = "Main Activity";
private View showDialogView;
private TextView outputTextView;
private List<IncidentType> incidentTypes;
private List<VehicleType> vehicleTypes;
private ArrayList<String> stringIncidentTypes;
private ArrayList<String> stringVehicleTypes;
private GPSTracker gpsTracker;
private DatePickerDialog datePickerDialog;
private String currentPhotoPath;
private Bitmap bitmap;
private static Integer REQUEST_TAKE_PHOTO = 100;
private static Integer REQUEST_SELECT_IMAGE=101;
private String imageName;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Set up the incident report form.
    titleEditText = (AutoCompleteTextView) findViewById(R.id.titleEditText);
    descriptionEditText = (EditText) findViewById(R.id.descriptionEditText);
    vehicleTypeSpinner = (Spinner) findViewById(R.id.vehicleTypeSpinner);
    incidentTypeSpinner = (Spinner) findViewById(R.id.incidentTypeSpinner);
    sendMyLocationSwitch = (Switch) findViewById(R.id.sendMyLocationSwitch);
    contactEditText = (EditText) findViewById(R.id.contactEditText);
    dateEditText = (EditText) findViewById(R.id.dateEditText);

    mLoginFormView = findViewById(R.id.report_incident_form);
    mProgressView = findViewById(R.id.login_progress);

    progressDialog = new ProgressDialog(this);

    incidentTypes = new ArrayList<IncidentType>();
    stringIncidentTypes = new ArrayList<String>();
    vehicleTypes = new ArrayList<VehicleType>();
    stringVehicleTypes = new ArrayList<String>();

    gpsTracker = new GPSTracker(MainActivity.this);

    volleyGetVehicleTypes(AppConfig.GET_VEHICLE_TYPES_URL);
    volleyGetIncidentTypes(AppConfig.GET_INCIDENT_TYPES_URL);
```

```
        imageView = (ImageView) findViewById(R.id.imageView);

        mReportIncidentButton = (Button) findViewById(R.id.reportIncidentButton);
        mReportIncidentButton.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                reportIncident();
                //test();
            }
        });

        mAttachPhotosButton = (Button) findViewById(R.id.attachPhotosButton);
        mAttachPhotosButton.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                attachPhotos();
            }
        });

        dateEditText.setOnFocusChangeListener(new View.OnFocusChangeListener() {
            @Override
            public void onFocusChange(View v, boolean hasFocus) {
                if (hasFocus) {
                    Calendar c = Calendar.getInstance();
                    int tyear = c.get(Calendar.YEAR);
                    int tmonth = c.get(Calendar.MONTH);
                    int tday = c.get(Calendar.DAY_OF_MONTH);

                    datePickerDialog = new DatePickerDialog(MainActivity.this,
                            new DatePickerDialog.OnDateSetListener() {
                                @Override
                                public void onDateSet(DatePicker view, int year,
                                                      int monthOfYear, int dayOfMonth) {

                                    dateEditText.setText(" " + (monthOfYear + 1) + "/" + dayOfMonth +
"/" + year);
                                }
                            }, tyear, tmonth, tday);
                    datePickerDialog.getDatePicker().setMaxDate(System.currentTimeMillis());
                    datePickerDialog.show();
                }
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_mylocation:
            Intent intent = new Intent(this, MapsActivity.class);
            startActivity(intent);
            break;
        case R.id.action_refresh:
            Toast.makeText(this, "Refresh selected", Toast.LENGTH_LONG).show();
            break;
    }
    return true;
}

/**
 * Attempts to report incident.
 * If there are form errors (invalid email, missing fields, etc.), the
 * errors are presented and no actual report attempt is made.
 */
private void reportIncident() {

    // Reset errors.
    titleEditText.setError(null);
    descriptionEditText.setError(null);
    contactEditText.setError(null);
    dateEditText.setError(null);

    // Store values at the time of the report incident attempt.
    String title = titleEditText.getText().toString();
    String description = descriptionEditText.getText().toString();
    String contact = contactEditText.getText().toString();
    String date = dateEditText.getText().toString();

    boolean cancel = false;
    View focusView = null;

    // Check title is not empty
    if (TextUtils.isEmpty(title)) {
        titleEditText.setError(getString(R.string.error_empty_title));
        focusView = titleEditText;
        cancel = true;
    }
```

```
    // Check description is not empty
    if (TextUtils.isEmpty(description)) {
        descriptionEditText.setError(getString(R.string.error_field_required));
        focusView = descriptionEditText;
        cancel = true;
    }
    //Check contact is not empty
    if (TextUtils.isEmpty(contact)) {
        contactEditText.setError("This field is required");
        focusView = contactEditText;
        cancel = true;
    }
    //Check date is not empty
    if (TextUtils.isEmpty(date)) {
        dateEditText.setError("This field is required");
        focusView = dateEditText;
        cancel = true;
    }

    if (cancel) {
        // There was an error; don't attempt report incident and focus the view first
        // form field with an error.
        focusView.requestFocus();
    } else {
        // Submit the incident report,volley
        //Toast.makeText(this,"Incident has been reported",Toast.LENGTH_LONG).show();
        volleyReportIncident(AppConfig.REPORT_INCIDENT_MAIN_URL);
    }
}

public void volleyGetIncidentTypes(String url) {

    progressDialog.setMessage("....loading");

    String REQUEST_TAG = "com.androidtutorialpoint.volleyJsonArrayRequest";

    JsonObjectRequest jsonObjReq = new JsonObjectRequest(url,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    try {
                        progressDialog.hide();
                        JSONArray jsonArray = (JSONArray) response.get("incident_types");
                        //call method to iterarate the json array and put in the List
                        getIncidentTypes(jsonArray);
                    } catch (JSONException e) {
                        Log.e(ACTIVITY_TAG, "Message<<<<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
```

```
e.toString());
                    }
                }
            }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    });

    // Adding JsonObject request to request queue
    AppSingleton.getInstance(getApplicationContext()).addToRequestQueue(jsonObjReq,
REQUEST_TAG);
  }

  public void getIncidentTypes(JSONArray jsonArray) {
    for (int i = 0; i < jsonArray.length(); i++) {

        try {
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            IncidentType iType = new IncidentType(jsonObject.getInt("id"),
jsonObject.getString("name"));
            incidentTypes.add(iType);
            Log.d(ACTIVITY_TAG, "Message>>>>>>>>>>>>>>>>>>>>>>>>>>: " +
iType.getIncidentName());
        } catch (JSONException e) {
            Log.e(ACTIVITY_TAG, "Error<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
e.toString());
        }
    }
    //setting the array list with the labels
    for (int i = 0; i < incidentTypes.size(); i++) {
        stringIncidentTypes.add(incidentTypes.get(i).getIncidentName());
    }
    //setting the adapter to show the values in the dropdown
    incidentTypeSpinner.setAdapter(new ArrayAdapter<String>(MainActivity.this,
android.R.layout.simple_spinner_dropdown_item, stringIncidentTypes));
  }

  public void volleyGetVehicleTypes(String url) {

    progressDialog.setMessage("....loading");

    progressDialog.show();

    String REQUEST_TAG = "com.androidtutorialpoint.volleyJsonArrayRequest";
```

```
        JsonObjectRequest jsonObjReq = new JsonObjectRequest(url,
                new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {
                        try {
                            progressDialog.hide();
                            JSONArray jsonArray = (JSONArray) response.get("vehicle_types");
                            //call method to iterate the json array and put in the List
                            getVehicleTypes(jsonArray);
                            Log.e(ACTIVITY_TAG, "Message>>>>>>>>>>>>>>>>>>>>>>>>>>: " +
jsonArray.toString());
                        } catch (JSONException e) {
                            Log.e(ACTIVITY_TAG, "Error<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
e.toString());
                        }
                    }
                }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

                Log.e(ACTIVITY_TAG, "Message<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
error.toString());
            }
        });

        // Adding JsonObject request to request queue
        AppSingleton.getInstance(getApplicationContext()).addToRequestQueue(jsonObjReq,
REQUEST_TAG);
    }

    public void getVehicleTypes(JSONArray jsonArray) {
        for (int i = 0; i < jsonArray.length(); i++) {
            try {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                VehicleType vType = new VehicleType(jsonObject.getInt("id"),
jsonObject.getString("name"));
                vehicleTypes.add(vType);
                Log.d(ACTIVITY_TAG, "Message>>>>>>>>>>>>>>>>>>>>>>>>>>: " +
jsonObject.getString("name"));
            } catch (JSONException e) {
                Log.e(ACTIVITY_TAG, "Message<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
e.toString());
            }
        }
        //setting the array list with the labels
        for (int i = 0; i < vehicleTypes.size(); i++) {
            stringVehicleTypes.add(vehicleTypes.get(i).getVehicleTypeName());
```

```
        }
        //setting the adapter to show the values in the dropdown
        vehicleTypeSpinner.setAdapter(new ArrayAdapter<String>(MainActivity.this,
android.R.layout.simple_spinner_dropdown_item, stringVehicleTypes));
    }

    public void volleyReportIncident(String url) {

        //check that location settings are enabled before submitting an incident
        if (!gpsTracker.canGetLocation()) {
            gpsTracker.showSettingsAlert();
            return;
        }

        String REQUEST_TAG = "com.androidtutorialpoint.volleyJsonArrayRequest";

        final String lat = Double.toString(gpsTracker.getLatitude());
        final String lon = Double.toString(gpsTracker.getLongitude());

        progressDialog.setMessage("Sending data...");
        progressDialog.show();

        StringRequest stringReq = new StringRequest(Request.Method.POST, url,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            JSONObject jsonObject = new JSONObject(response);
                            Log.d(ACTIVITY_TAG, "Message>>>>>>>>>>>>>>>>>>>>>>>>>>: " +
jsonObject.getString("message"));
                            //Complete the process by showing the confirmation alert and clearing the
fields
                            LayoutInflater li = LayoutInflater.from(MainActivity.this);
                            showDialogView = li.inflate(R.layout.show_dialog, null);
                            outputTextView = (TextView)
showDialogView.findViewById(R.id.text_view_dialog);
                            AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(MainActivity.this);
                            alertDialogBuilder.setView(showDialogView);
                            alertDialogBuilder.setPositiveButton("OKAY", new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {

                                    titleEditText.setText("");
                                    descriptionEditText.setText("");
                                    contactEditText.setText("");
```

```
                dateEditText.setText("");
                titleEditText.requestFocus();
                imageView.setImageBitmap(null);
                imageView.setVisibility(View.INVISIBLE);
              }
            }).setCancelable(false)
                .create();
            outputTextView.setText(jsonObject.getString("message"));
            alertDialogBuilder.show();
            progressDialog.hide();
          } catch (Exception e) {
            Log.e(ACTIVITY_TAG, "Error<<<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
e.toString());
          }
        }
      }, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

      Log.e(ACTIVITY_TAG, "Message<<<<<<<<<<<<<<<<<<<<<<<<<<<<: " +
error.toString());
    }
  }) {
    @Override
    protected Map<String, String> getParams() {
      Map<String, String> params = new HashMap<String, String>();
      params.put("title", titleEditText.getText().toString());
      params.put("description", descriptionEditText.getText().toString());
      params.put("vehicle_type", vehicleTypeSpinner.getSelectedItem().toString());
      params.put("incident_type", incidentTypeSpinner.getSelectedItem().toString());
      params.put("incident_contact", contactEditText.getText().toString());
      params.put("incident_date", dateEditText.getText().toString());
      params.put("incident_lat", lat);
      params.put("incident_long", lon);
      params.put("incident_photo_id",imageName);
      params.put("incident_photo",getStringImage(bitmap));

      return params;
    }

    @Override
    public Map<String, String> getHeaders() throws AuthFailureError {
      Map<String, String> pars = new HashMap<String, String>();
      pars.put("Content-Type", "application/x-www-form-urlencoded");
      return pars;
    }
  };
```

```
    stringReq.setRetryPolicy(new
DefaultRetryPolicy(30000,DefaultRetryPolicy.DEFAULT_MAX_RETRIES,DefaultRetryPolic
y.DEFAULT_BACKOFF_MULT));

    // Adding JsonObject request to request queue
    AppSingleton.getInstance(getApplicationContext()).addToRequestQueue(stringReq,
REQUEST_TAG);
  }

  public void attachPhotos() {
    final CharSequence[] options = {"Take Photo", "Choose From Gallery", "Cancel"};

    final AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
    builder.setTitle("Attach Photos");
    builder.setItems(options, new DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialog, int which) {

        if (options[which].equals("Take Photo")) {
          Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
          //take the photo
          File photoFile = null;
          try {
            photoFile = createImageFile();
          } catch (IOException e) {
            Log.d(AppConfig.TAG, "Error creating file " + e.getMessage());
            e.printStackTrace();
          }
          //continue if the file was successfully created
          if (photoFile != null) {
            Log.d(AppConfig.TAG, ">>>>>>>>>>>>>>Photfile is not empty");
            intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(photoFile));
            startActivityForResult(intent, REQUEST_TAKE_PHOTO);
          }
        }
        else if(options[which].equals("Choose From Gallery"))
        {
          Intent intent=new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
          intent.setType("image/*");
          startActivityForResult(intent,REQUEST_SELECT_IMAGE);
        }
        else if(options[which].equals("Cancel"))
        {
          dialog.dismiss();
```

```
            }
        }
    });
    builder.show();
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_TAKE_PHOTO) {
        if (resultCode == RESULT_OK) {

            try {
                bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(),
Uri.parse(currentPhotoPath));
                imageView.setImageBitmap(bitmap);
                imageView.setVisibility(View.VISIBLE);
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        } else if (resultCode == RESULT_CANCELED) {
            Toast.makeText(getApplicationContext(), "User cancelled image capture",
Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(getApplicationContext(), "Sorry failed to capture image",
Toast.LENGTH_LONG).show();
        }
    }
    if(requestCode==REQUEST_SELECT_IMAGE)
    {
        if(resultCode==RESULT_OK)
        {
            Uri selectedImage = data.getData();
            String[] filePathColumn={MediaStore.Images.Media.DATA};
            Cursor cursor = getContentResolver().query(selectedImage, filePathColumn, null,
null, null);
            cursor.moveToFirst();
            int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
            String picturePath = cursor.getString(columnIndex);
            cursor.close();
            bitmap = BitmapFactory.decodeFile(picturePath);
            imageView.setImageBitmap(bitmap);
            imageView.setVisibility(View.VISIBLE);
            File f=new File(picturePath);
            imageName=f.getName();
```

```
        }
      }
    }

    private File createImageFile() throws IOException {
        // Create an image file name
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new
Date());
        String imageFileName = "IMG_" + timeStamp + "_";
        imageName="IMG_" + timeStamp + "_.jpg";
        File storageDir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
        File image = File.createTempFile(imageFileName, ".jpg", storageDir);
        // Save a file: path for use with ACTION_VIEW intents
        currentPhotoPath = "file:" + image.getAbsolutePath();
        return image;
    }

    private String getStringImage(Bitmap bitMap) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bitMap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        byte[] imageBytes = baos.toByteArray();
        String imageString = Base64.encodeToString(imageBytes, Base64.DEFAULT);
        return imageString;
    }
    private void test()
    {

Toast.makeText(getApplicationContext(),getStringImage(bitmap),Toast.LENGTH_LONG).sho
w();
    }
}
```

2. Reporting Incident- Backend Script

```
<?php

header("access-control-allow-origin: *");


include "db.php";


$dbh=pg_connect(DB_CONNECTION);


if(!$dbh)

{

	echo "The connection failed";

}


$title=$_POST["title"];

$description=$_POST["description"];

$vehicle_type=$_POST["vehicle_type"];

$incident_type=$_POST["incident_type"];

$incident_contact=$_POST["incident_contact"];

$incident_date=$_POST["incident_date"];
```

```
$incident_lat= doubleval($_LONG["lat"]);

$incident_long= doubleval($_LONG["long"]);

$incident_photo_id=$_POST["incident_photo_id"];

$incident_photo=$_POST["incident_photo"];


if((!empty($incident_photo_id) and !is_null($incident_photo_id)) and (!empty($incident_photo)
and (!is_null($incident_photo))))

{

    $imagepath="/mobirep/uploads/".$incident_photo_id;



    $serverUrl="http://commigy.com".$imagepath;



    file_put_contents($serverUrl,base64_decode($incident_photo));



}
```

```
$query="INSERT INTO
incident(incident_title,incident_description,vehicle_type,incident_type,incident_contact,incident
_photo_id,lat,long,incident_date,incident_photo_url,incident_photo) VALUES
('".$title."','".$description."','".$vehicle_type."','".$incident_type."','".$incident_contact."','".$incid
ent_photo_id."','".$incident_lat."','".$incident_long."','".$incident_date."','".$serverUrl."','".$incid
ent_photo."')";


$result=pg_query($dbh,$query);


if($result)
{
    $json['status']=0;
            $json['message']="The incident has been recorded";
}
else
{
   $json['status']=1;
   $json['message']="There was an error in recording the incident  ".pg_last_error($dbh);
}
```

```
echo json_encode($json);
```