THE UNIVERSITY OF ZAMBIA

SCHOOL OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER STUDIES

A Real Time, Dynamic and Extensible Information Map for Lusaka City

BY

MUNACHITOMBWE MICHELO

CST 4000 FINAL YEAR PROJECT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF ZAMBIA

SCHOOL OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER STUDIES

A Real Time, Dynamic and Extensible Information Map for Lusaka City

BY

MUNACHITOMBWE MICHELO

CST 4000 FINAL YEAR PROJECT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

DECLARATION

I the undersigned here declare that the Real Time, Dynamic and Extensible Information Map for Lusaka City is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references

Student Name	MunachitombweMichelo
Computer Number	2907362
Signature	
September 2013	

Supervisor Name	Andrea Theu
Signature	
September 2013	

ACKNOWLEDGEMENT

I would like to thank all the people who in one way or another helped me to do this project successfully.

In particular, I would like to thank my lecturer and supervisor Mr. A Theu for his guidance during the project. Furthermore I wish to extend my gratitude to all the lecturers in the department for all the help they offered.

I would also like to thank all my classmates for all the encouragement, information and the good environment that they offered.

My family was very supportive during this period. Mum and Dad who provided parental support. My siblings Peter, Noreen, Natuse, Rachael and Nyama who were always there for me doing the best that they could.

Last but not the least, I thank God for everything. Thank you all, God bless.

ABSTRACT

Modern technologies have made it possible for amateur news applications and related journalistic applications (blogs, social networks, collaborative publishing) to capture news faster and in larger amounts than any other news systems. These technologies (e.g. smart phones) can take pictures, record audios and videos and have access to the internet. However location of specific and desired news items from this large amount of news is very difficult. Poor organization of this information results in a limiting rather than an improvement of the news application. The project attempts to alleviate this problem by organizing the news around a map with the help of a mapping service. Therefore the application created is an integration of an amateur news application with a web mapping service. The geographical dimension added to the application by the mapping system will improve the capture of proximal and timely news and will improve searching. Also in order to aid news discovery, the application has a recommender system where users can get recommendations about news items which might be of interest to them.

LIST OF FIGURES

TABLE OF CONTENTS

1.0 CHAPTER 1 -- INTRODUCTION TO THE RESEARCH

1.1 Introduction

The project creates a public news system that allows Lusaka residents to obtain and publish public news easily and at a low cost. The system must organize the news such that it will be easy to obtain and publish the public news. The primary news organization strategy is to organize the news around geographical maps. The system contains a server to receive and distribute the news, mobile phones to publish and obtain the news and web browsers to publish and obtain the news.

1.2 Motivation and Significance

News consumption and production has not yet taken advantage of the full potential of the technologies available. Numerous technologies (e.g. smart phones) can take pictures, record audios and videos and have access to the internet. Let's say a newsworthy event occurs and Mr. A takes footage of it. Applications such as Facebook or Twitter only allow sharing information to the people Mr. A is connected to like friends on Facebook or followers on Twitter. If Mr. A desires to share that with the general public how can he do that effectively?

An application whose goal is to answer the previous question is needed. Every strategy employed to solve the problem will have its own strengths and weakness. Traditional news applications are good at sharing news that is relevant to a large section of the population with a high accuracy. They fail at disseminating news relevant to a few individuals and have been beaten in the time it takes to report the events by applications such Facebook and Twitter

The project's strategy is to organize news around a map with the help of a mapping service (i.e. Google maps).The strength of the strategy is that the geographical dimension added to the application by the mapping system will improve the capture and searching of proximal[1] (a geographical location relevant to the individual, for example the home location) and timely news.

1.3 Scope

Lusaka city is the geographical region that is focused on. In this area, local news is presented according to where it has happened or a relevant location. There is a facility for users to comment on the news items and therefore interact with one another concerning the news items. Users will have facilities to search for items according to their geographical context, time and other attributes of the items with the results presented on the map. Authorized users will be allowed to update information and put new items on the map.

There will be a facility that recommends news to the user. This is done to increase the applications ability to deliver the content that the each individual user is likely to be interested in. These recommendations are based on the user's history, geographical location and other relevant information [1]. Lastly users have a facility to obtain navigation directions to places and news locations that interest them.

1.4 Problem Statement

Modern technologies have made it possible for amateur news applications and related journalistic applications (blogs, social networks) to capture news faster and in larger amounts than any other news systems. However location of specific and desired news items from this large amount of news is very difficult. Poor organization of this information results in a limiting rather than an improvement of the news application.

1.4.1 Research Question

Can combining an amateur news application with a web mapping service produce a news system for Lusaka city that improves the capture of proximal and timely news and improves searching?

1.4.2 Hypothesis

Amateur news applications using modern technologies can capture timely news while presenting the news on a mapping service can improve searching and access to proximal news.

1.5 Aim and Objectives

1.5.1 Aim

To create and evaluate a combination of an amateur news application with a web mapping service.

1.5.2 Objectives

1. Review relevant applications.

Activities: Read literature on map based applications and news applications. Use websites for selected applications and also their applications on mobile phone (android devices).

Deliverables: Section of the report

2. Develop the Functional specification.

Activities: Interview potential system users. Analyze requirements. Develop specification

Deliverables: Functional specification

3. Design and implement the system.

Activities: Create a system design. Implement the design.

Deliverables: Use case diagrams, Class diagrams, Sequence diagrams, State diagrams, data model, and then the complete system (the web based application and the mobile application)

4. Test and evaluate the system

Activities: testing of application by users. Analyze test results .Review of project processes and product.

Deliverables: section of the report

5. Complete final report.

Activities: prepare final report

Deliverables: Final project report

1.6 Organization

This document is organized into 5 chapters. The current chapter makes the introductory chapter. It gives the necessary groundwork for the direction of this thesis.

Chapter 2 is the literature review and related works. Review on relevant technologies is done and 4 similar products to the one created are reviewed.

Chapter 3 is the Methodology. It contains the software artifacts that are created in the process of structured software engineering. These are the requirements, Design and the implementation

Chapter 4 is the Testing and Results. A description of the testing strategies employed and the results obtained is given.

Chapter 5 is the Discussion and Conclusion. The results obtained are analyzed and they are further related to the aims and objectives in this section.

The Appendix and References follows. These remain self-explanatory.

1.7 Summary

Therefore the project was conducted because of the problems and opportunities for improvement that have been discussed previously. An application that allows the general public to inform each user of what is happening at any location of interest would be greatly advantageous. These assumed benefits will be verified by achieving the aims and objectives and evaluating whether our assumptions are true.

2.0 CHAPTER 2 – LITERATURE REVIEW AND RELATED WORKS

2.1 Introduction

Citizen journalism is when public citizens play an active role in the process of collecting, reporting, analyzing, and disseminating news and information [2]. The potential of citizen journalism (i.e. amateur news) has been extensively researched and confirmed [3]. Social networking used together with cellular phones has already proven that this concept is very powerful [4][5]. Furthermore the concept of citizen journalism has been fully realized in South Korea where all OhMyNews [6] is the leading online local news site. Its motto is "every citizen is a reporter". Citizen journalism also follows the general web trend of the transition from information consumption to applications that support user participation in content creation [7]. That is consumers become prosumers (i.e. also produce) [8].

2.2 Literature Review

The amateur news systems discussed above can produce very large amounts of news. However organization and searching of this data becomes a difficult task. How does one find specific and desired news items from this large repository? Traditional media (BBC, ZNBC, the post) employ gatekeepers to as a solution to this problem [9]. Of the large amount of news gathered gatekeepers select what will be presented and what will not be presented and thus stream down the news to an appropriate size. This approach has its advantages and disadvantages [10]. The disadvantage that is focused on in the project is that few people decided what millions want to read or hear about.

When gatekeepers are absent individual consumers will need to browse through this large amount of content [11]. A range of approaches, algorithms and techniques have been suggested from simple ones to very complex ones [12] [13] [14]. This project proposes a relatively simple approach. The approach is to arrange news on a map by location using a modern geographical mapping system (Google maps). This will improve the capture and searching of timely and proximal[1] news.

Geographical mapping systems have been extensively used to organize and visualize data [15]. They have been used in "playful" systems to very critical applications [16][17]. Google maps is one such mapping system. In addition to maps and route planning it also offers a business locator

[14][15]. Third party applications use its services via its APIs [18]. There are over 2000 map mashups[2] that use Google maps [19]. The famous pioneer [20] of mashups is HousingMaps which advertises houses by location in the U.S.A.[21]. Zhappening[22] and Foursquare[23] are such types of applications but they focus on the "playful" check in and where you can find desired goods and services.

The proposed system can be divided into two parts. These are the web mapping service and the data to be overlaid on the maps. Many web mapping services exist. However few contain appreciable information about developing countries (i.e. Zambia). Open Street Maps is a good option as a mapping service since it is open source and has few restrictions. However it is not detailed for the Zambian region. Consequently, Google maps was chosen. Google maps now has its JavaScript Map API version 3 for the web (browser) [24]. It also has android map API version 2 for mobile devices running the android OS (phones and tablets)[25]. It places some restrictions on the free use of both APIs mentioned above.

The data to be overlaid on the maps comes from the users on PC browsers and mobile phones. All major browsers support the technology requirements of the system to be developed. They all support JavaScript and AJAX which is what Google maps API version 3 uses. The mobile operating system to be used is the Android OS[26]. Any version above 2.2 will suffice. Its advantages are that it is an open source OS. It is platform independent and many manufacturers support it[27]. It has the largest share of the smart phone market [28]. It is also very easy to distribute applications on it in comparison with other mobile operating systems. It also has a freely available Android Development kit.

Consequently the project builds the proposed system using the technologies which have been discussed above. It integrates together a citizen journalist application with Google maps creating a whole new experience for Lusaka city.

2.3 Related Works

2.3.1 Foursquare

Foursquare is a global location-based social networking website for mobile devices, such as smartphones. Users "check in" at venues using a mobile website, text messaging or a device-specific application by selecting from a list of venues the application locates nearby. Location is based on GPS hardware in the mobile device or network location provided by the application. Each check-in awards the user points and sometimes "badges". Using this information the application can make recommendations about which places a particular user should visit. The check in system allows the owner of a venue (e.g. shop owner) to make personal deals (e.g. discount on a price) with a particular user. Lastly the "check in", "badges" and "mayor ship" mechanisms give the application a "playful" side [16][17].

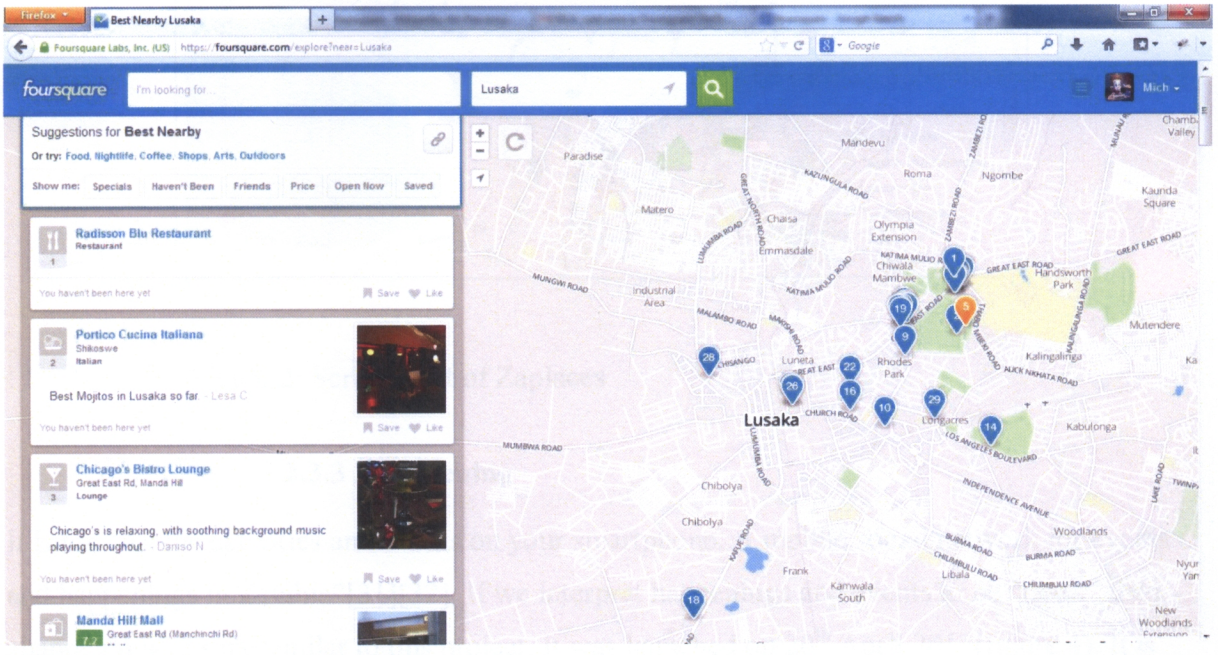

Figure 1 : Screen shot of foursquare

2.3.2 Zaplaces

Zaplaces is aims to improve the discovery of businesses. In their own words "At Zaplaces we seek to help people discover places as well as to provide businesses with a web presence."[29] It currently focuses on Lusaka in Zambia but it has officially indicated that the expansion is to include Africa as well [30]. It is run by Venivi limited that is a startup in Lusaka [29].

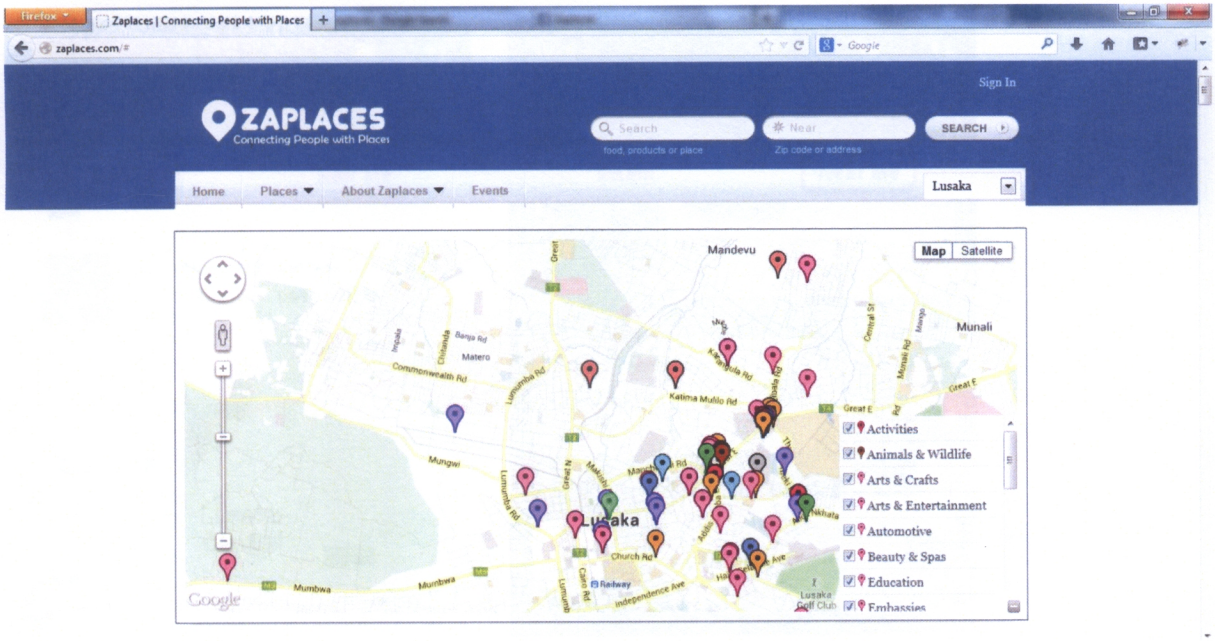

Figure 2: Screen shot of Zaplaces

2.3.3 Zhappening

It is for "All the activities and events on your smartphone. A mobile social network for events and happenings in Zambia"[31][32]. If we interpret happenings and events to be "news" then zhappenning is very similar to this project. It was launched on 20th April 2013(figure 4). It is about 6 months old while this project began about a year ago (July 2012). However differences remain. Zhappening has a general focus and currently combines ideas from several popular applications such as the "Check in", "Facebook wall", and blogging. It also has a global focus. It has an Android App and an I-phone App only. However this project has a specific focus on representing Lusaka's news on a map in order to meet the goals in the aim. The project has a recommendation component and navigation component. Zhappening does not have these at this

point in time. There are more differences. However, Zhappening is still in its early stages and is expected to change as it grows. Therefore the facts stated above may soon become invalid. It is run by Venivi limited that is a startup in Lusaka [29].

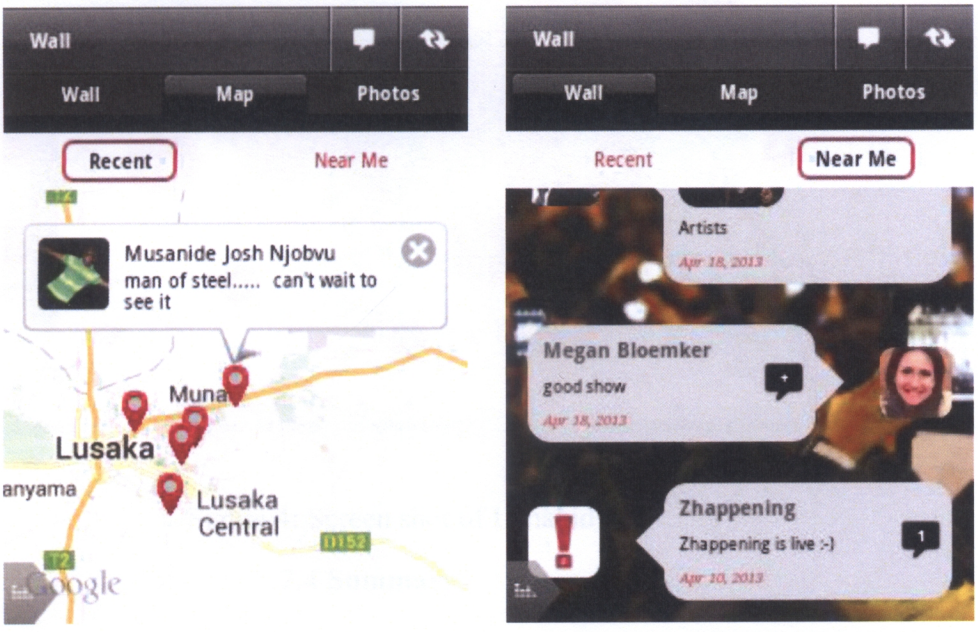

Figure 3: Screen shot of Zhappening

2.3.4 Ushahidi

Ushahidi is produced by the company named Ushahidi that specializes in developing free and open source software for information collection, visualization and interactive mapping. This system is as general as possible. It therefore touches upon all areas and ventures that concern web mapping, this project included. It was a website that was initially developed to map reports of violence in Kenya after the post-election fallout at the beginning of 2008. Therefore it is a proof of concept of the reporting power of "news on a map" even in the African region. However, since its initial creation it has focused on major crisis reporting all over the globe. Two facts summarize the situation. Firstly Ushahidi is involved in interactive maps of every sort. Secondly it has a global focus. These two facts inherently mean that it loses efficiency and effectiveness at a regional level such as Lusaka city. In fact, at the writing of this report there

was no user in the sample who was aware of its existence. This focus and relevance for the specific region of Lusaka city gives this project the only advantage (or difference) over the very versilte Ushahidi.

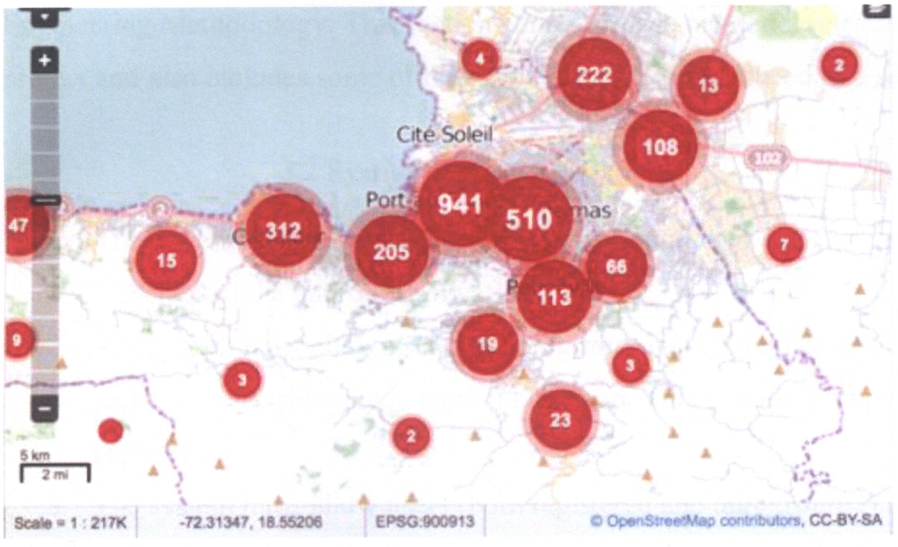

Figure 4: Screen shot of Ushahidi

2.4 Summary

With this review of literature, the scientific theoretical background of the work has been given. The current technologies related to the system have also been reviewed.

3.0 CHAPTER 3 – METHODOLOGY

3.1 Introduction

Good software engineering of complex projects requires the use of scientifically sound software Engineering Methodology. This section contains those methodologies that were employed in this project and also includes some of the artifacts that were produced as a result of those processes.

3.2 System Requirements

1. The system must allow registered users to input news items on the map on both the website and the android application
2. The system must allow users (both registered and unregistered) to view the news items that have been previously input on the map. This is both for the website and the android application
3. The system must allow users (both registered and unregistered) to search for specific news items or groups of related news items using the provided search functionality. The results should be displayed on the map. This is both for the website and the android application
4. The system must recommend news items to users (both registered and unregistered) .The results should be displayed on the map. The recommendations should be more personalized for registered users. This is both for the website and the android application
5. The system must allow users (both registered and unregistered) to get navigation directions to locations where news items have been posted .The resultant navigational directions should be displayed on the map. This is both for the website and the android application
6. The system must allow registered users to comment and interact over the items posted on the map. This is both for the website and the android application
7. The system must be easy to use and not require a steep learning curve

3.3 System Design

The System is designed in the Client-Server paradigm that is common for web applications. The Object-based architectural style is used for the general component design. We begin the System design with the use case diagram given below

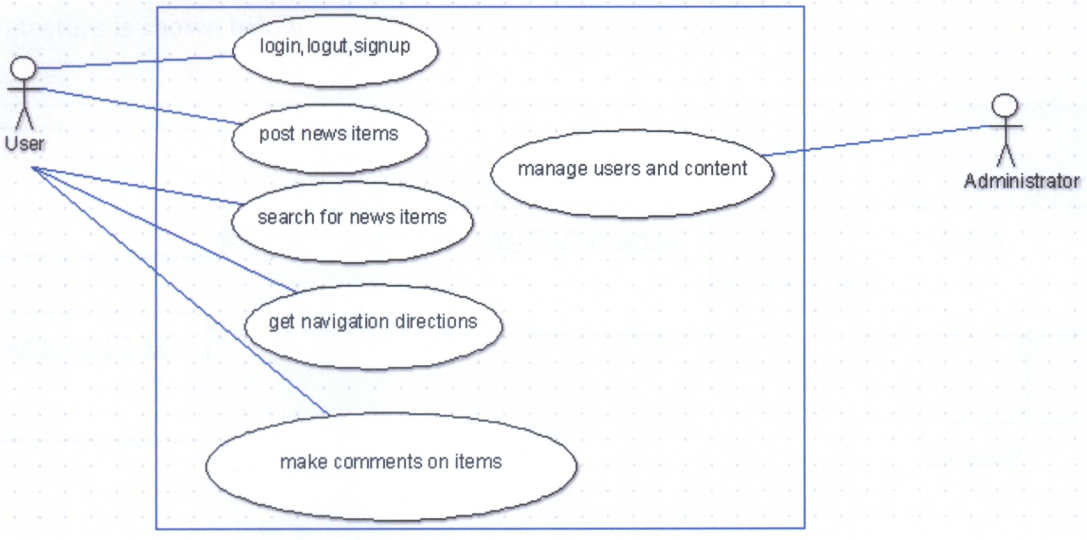

Figure 5: Use case diagram of the system

In figure 6, we have the use case diagram for the system. We here see the user who is representative of users of the system who wish to use the system for any of the functionality it provides. The user can be part of all the scenarios (use cases) that they are connected to. For example, we have the post news items scenario. The user puts information on the map using that scenario (use case). Other functionality available to the user shown in the diagram includes searching, navigation, login processes and commenting. Other functionality exists which has not been shown in the diagram as only the core functionality has been shown. Lastly we have the administrator role to the right. He must be able to manage the content of the system and also the users. This managing includes giving user rights and reviewing which content can be removed from the system in order to increase efficiency

3.3.1 Class Diagrams

Using the Object-Oriented Methodology, the classes were structured as shown below. Because of the larger number of classes that were produced in the final design, not all the classes are shown in the diagrams to follow below. If one is interested to see these, they should consult the softcopy which is contained in the CD that accompanies this project. The overview of the software architecture is shown below

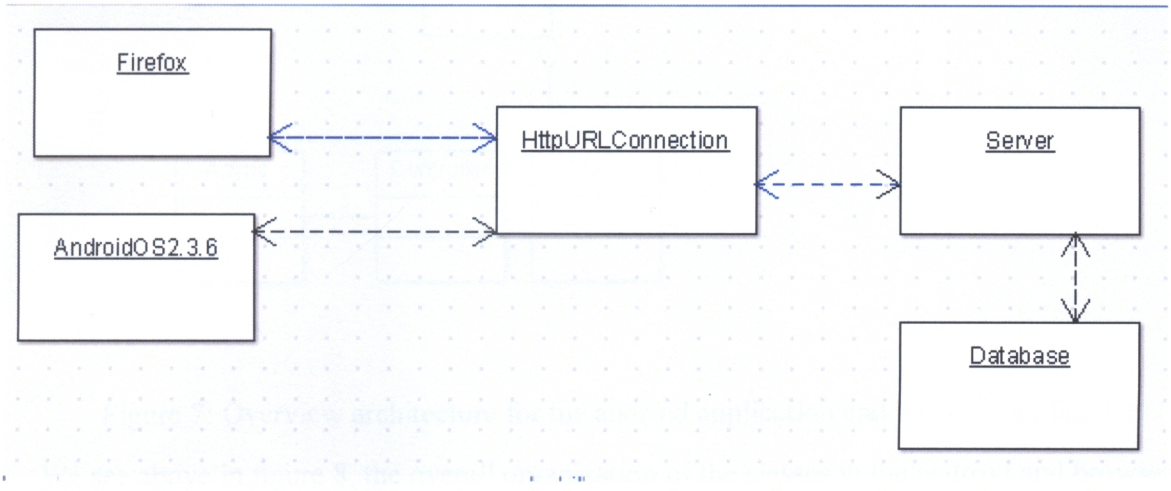

Figure 6: Overview of the software architecture

In Figure 7, at the top left we see Firefox. Firefox in this context stands for a standard browser. This represents all the parts of the software that will run in the browser (e.g. the website user interface). We have android OS 2.3.6. This represents all the software that will run on the android devices. There is a number is the minimum required android operating system is shown as version 2.3.6. Versions below this have no guarantees of being supported. Both the android and browser components communicate with the database and application servers using HTTP protocol connections. The application server and database server are flexible on the machines where they can be run.

Following the above overview, we expand on the software found on the android device and the browser. It should be noted that the system is also organized according to the model-view-controller architecture. This will be seen as an overriding general theme in the organization of the classes. The class diagrams are depicted in the diagrams below.

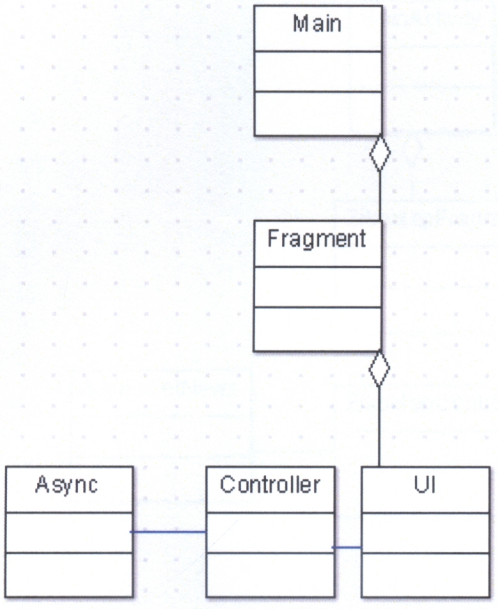

Figure 7: Overview architecture for the android application and browser application

We see above in figure 8, the overall organization of the classes in the android and browser application. The main class represents the root class of the application that loads where the application is started from. It also obtains all necessary resources from the system. The main application is further composed of Fragments (in the figure 8). The Fragment classes contain a given specific UI and all the other necessary components for that UI. The UI classes contain a given aspect of the complete user interface of the application. The controller handles all the events that the UI creates. The Async classes are used by the Controller to handle asynchronous loading of data from the Servers.

An example of the configuration described above is shown below for the android application

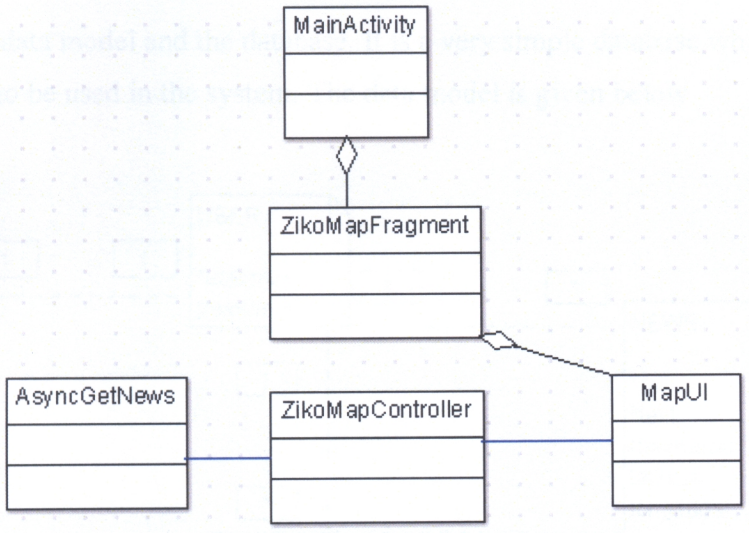

Figure 8: An Instance of the android application overview architecture

In figure 9 we see the MainActivity which is the root class of the android application that loads where the application is started from. It also obtains all necessary resources from the android system. ZikoMapFragment is the example Fragment. It contains everything to do with the map on the android application. An example of another Fragment is the LoginFragment which contains everything to do with login and logout. The MapUI implements does the actual drawing of the map and everything on it. The controller handles all the events that the MapUI creates. The Async classes are used by the ZikoMapController to handle asynchronous loading of data from the Servers onto the Map. The Website application is similarly organized as well.

3.3.2 Data Model and Database

Next we have the data model and the database. It is a very simple database which only captures the essential data to be used in the system. The data model is given below

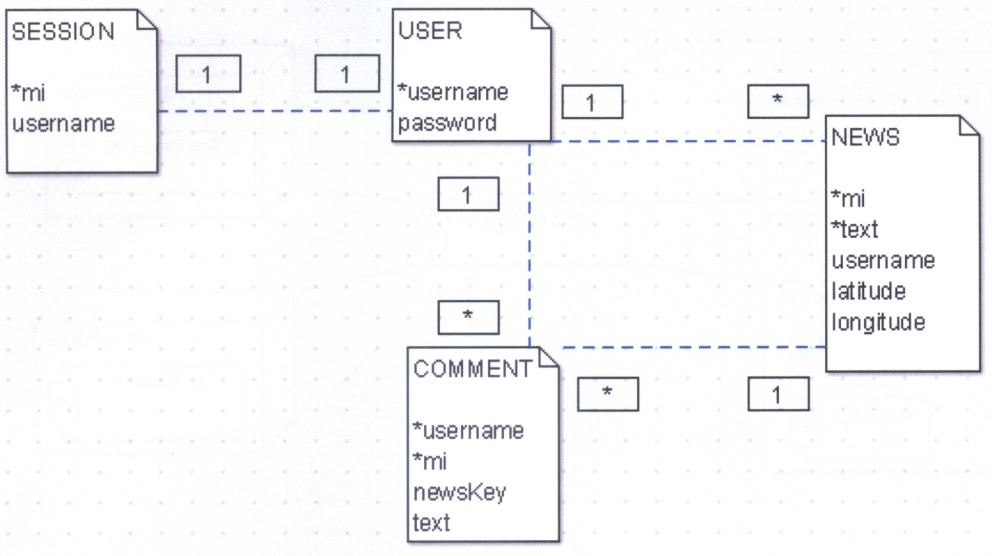

Figure 9: Data Model

So we have 4 data entities in Figure 10. The USER entity stores information about the users of the system. The SESSION entity stores information about the current users who are using the application. The COMMENT entity stores information about comments on a given NEWS entity. The NEWS entity stores information about the news item that was posted by the given USER. The small boxes beside the rectangles represent the multiplicities of relationships. The "*mi" implies that this should be unique key such that it can be a primary key or at least an alternate key. Explanation of the obscure data item is as follows

mi – a unique value for use in identification(preceded by *)

longitude – the longitudinal location of a news item on the map

latitude – the latitudinal location of a news item on the map.

3.3.3 Deployment Diagram

The hardware deployment scheme is shown below. This is the physical realization of the software architecture that was discussed above.

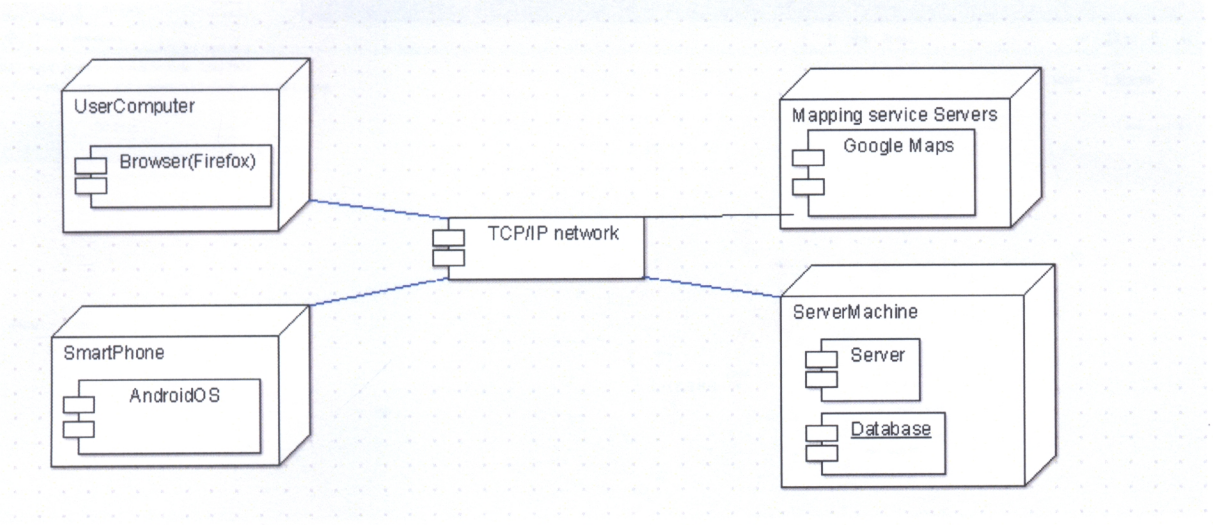

Figure 10: Deployment of the system

In Figure 11, we need at least 4 different machines for this configuration to work. The smart phone, user (client) machine all connect to the server machine over a TCP/IP network. The computer houses the web browser (e.g. Firefox). The smart phone houses the android application. The TCP/IP network is used for communication. The mapping servers are the Google servers that provide the mapping service. The Sever Machine houses the servers that contain the data model and database described above

3.4 System Implementation

Screen shots of the implemented system are given below. It is a technical fulfillment of the given system requirements, aims and objectives of this project.

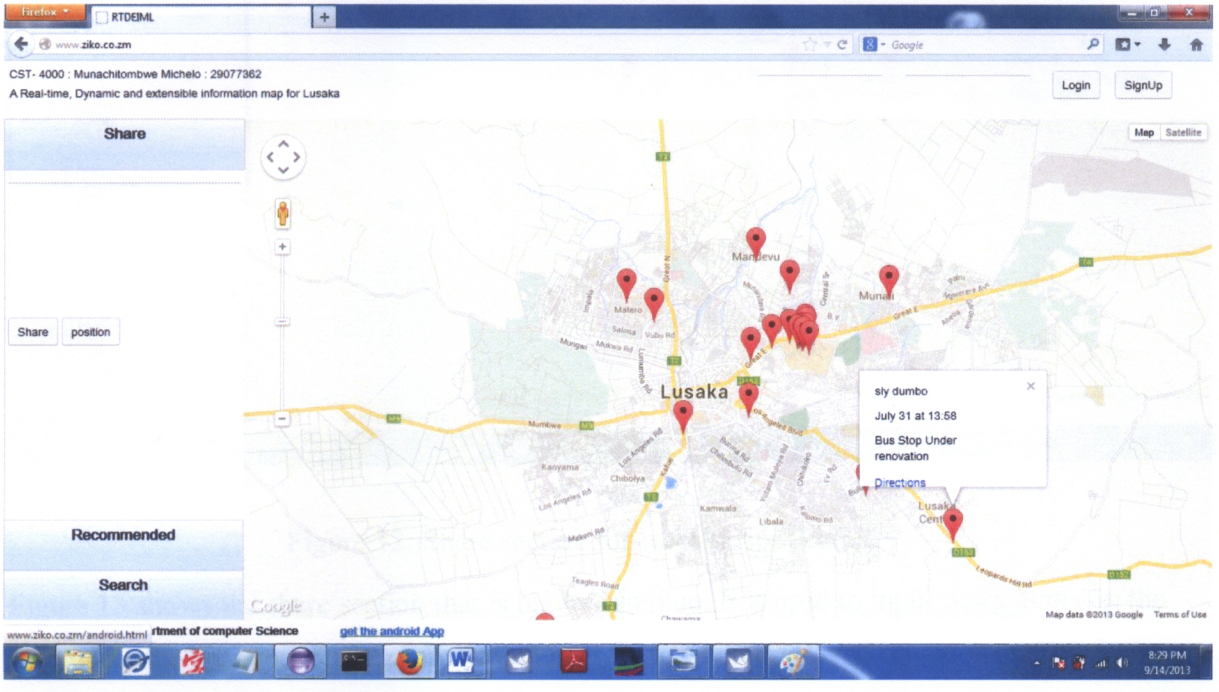

Figure 11 : The general website application screen shot

Figure 12 above shows the website component. Here we show an example news item that was posted by a "sly dumbo". We have the share section that is used to input news items. The recommended section is used to give recommendations. The search section is used to search for news items using the given text input.

Step by step we begin with the share section shown below

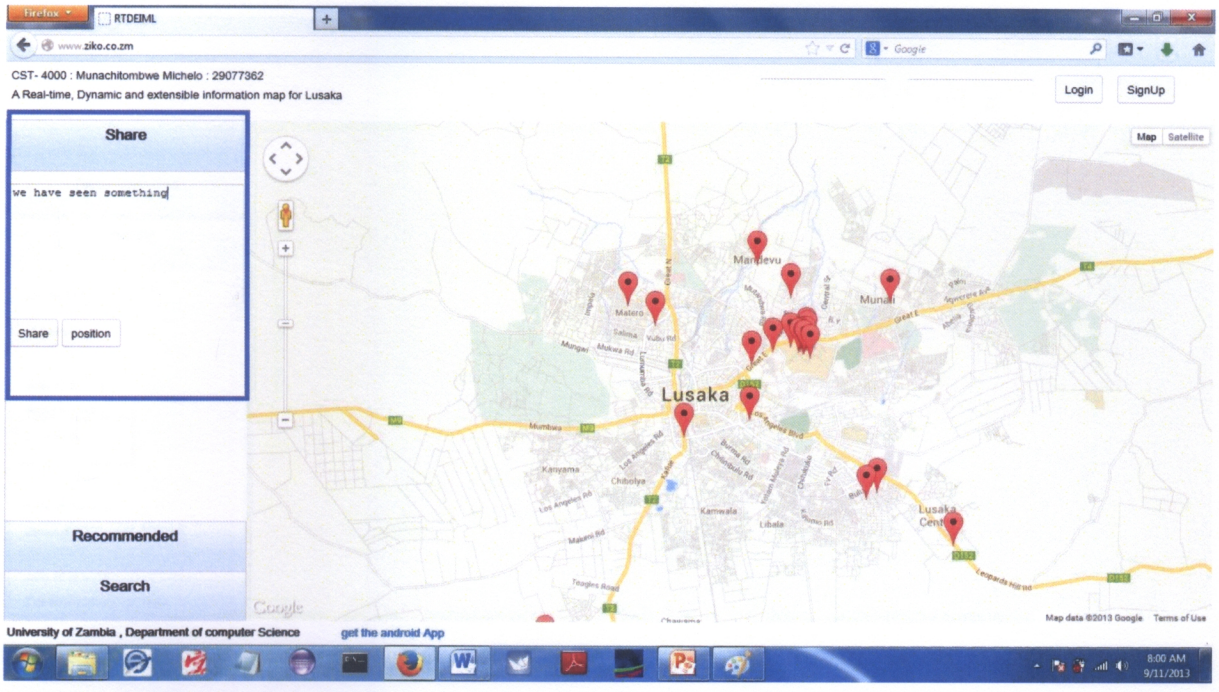

Figure 12: Share section for website application

Figure 13 shows the share section that is marked in blue. It is used to input news items on the map using the browser. The text area captures the text to be displayed on the map. This is the "we have seen something" in the diagram. The position button activates a marker that is used to select a location. The share button then posts the news item at the given location.

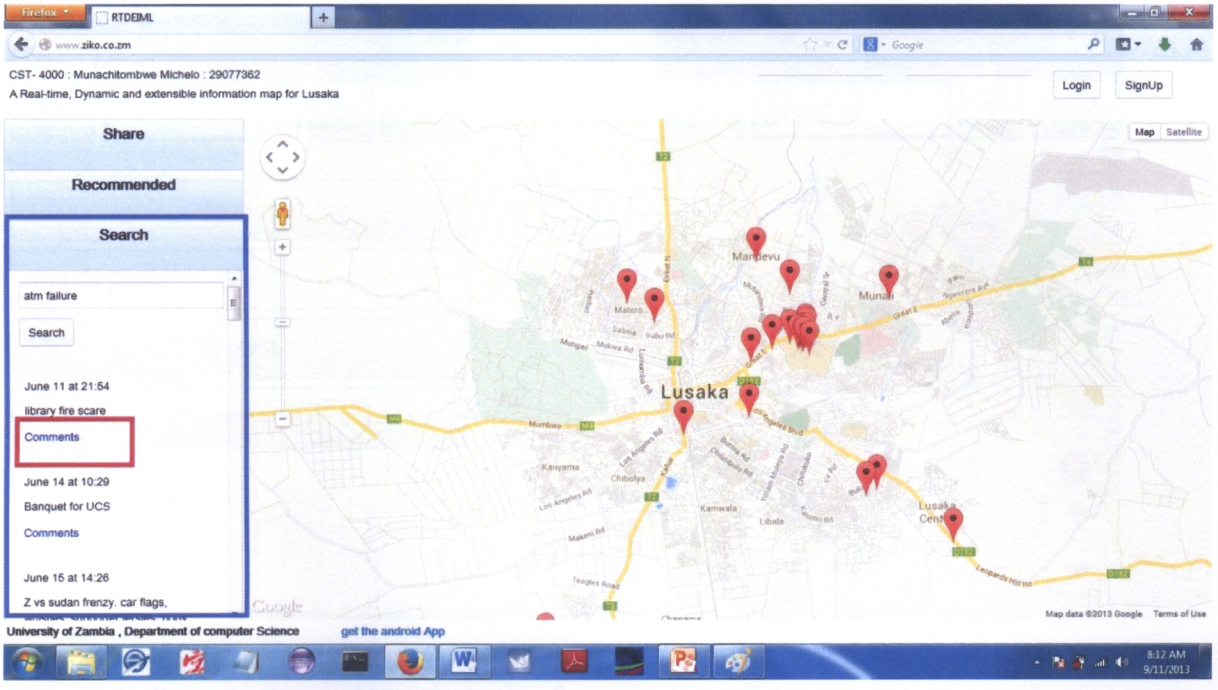

Figure 13: Search section for website application

Figure 14 shows the search section that is marked in blue. It is used to search for items on the map using a text search phrase. The text field in the screenshot contains the phrase "atm failure". We have a red square enclosing the comments link. This provides a facility for a user to comment on a given news item

The android screenshots are shown in Figure 15. To the left we see the share section that corresponds to the one that was discussed above. It also has the position and share button. They work the same as previously discussed above in the website component. The section used to search for news items is that text field with the blue outline. The button labeled "S" is the search button. The "Me" button takes the screen from the map view shown on the right to the share screen shown on the left.

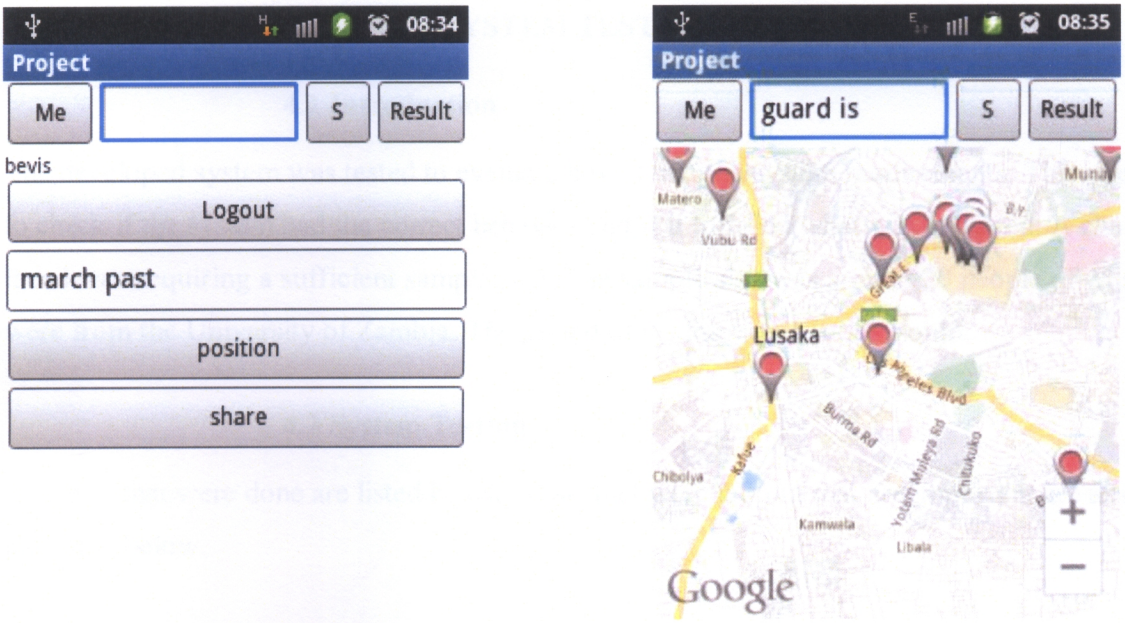

Figure 14: Share section and search section for android application

3.5 Summary

These are the methodologies that were employed and some of the artifacts that were produced as a product of those processes.

4.0 CHAPTER 4 – SYSTEM TESTING AND RESULTS

4.1 Introduction

The developed system was tested to evaluate how it met the system requirements. This was done to check if the system had the correct behavior and if it fulfilled what was expected. A challenge faced was acquiring a sufficient sample size. The sample size was around 20 people of which 18 were from the University of Zambia. The period of testing was 1 (one) month.

4.2 System Testing

The tests that were done are listed below. Tests that were left out that are important are also indicated below.

Graphical User Interface and Usability Testing: Users were asked to complete certain tasks (e.g. share a news item) on both the Web version and the android version. Crude measurements were taken to represent to effectiveness of the user Interface. This includes monitoring the difficulty with which a certain task was done including how long it took. The users were also asked to give their critique of the GUI. For usability testing, users were monitored as they performed the tasks checking for efficiency, accuracy, recall, emotional response and intuition. With regard to intuition, users were not given the steps of how to go about a certain task but instead allowed to perform the task using their experience from other systems.

Software performance and Compatibility Testing: Users were given tasks to use the system. While users performed these tasks the system was monitored for stability and responsiveness. These tests had a concurrency of 2 users at most. Therefore the load was very minimal. Compatibility tests were different for the website and the android application. The website compatibility tests involved evaluating how the system behaved under different web browsers (e.g. Firefox, Opera). The android application compatibility tests involved evaluating how the system behaved under different manufacturers (e.g. Samsung, Sony) and different Operating Systems (e.g. v4.0 –Ice cream Sandwich).

4.3 Results

Graphical User Interface and Usability: The difficulty of using the website GUI was measured to be average. After users were shown how to perform certain tasks, the amount of time that they required was minimal. Much of it went to typing text or thinking. The difficulty of using the android GUI was measured to be difficult. After users were shown how to perform certain tasks, the amount of time that they required was long compared to the same task on the website. It was noticed that the bigger the android device screen the easier the task was comparatively. Much of it went to manipulating the map (e.g. moving a marker). Several GUI improvements were requested by the users such as "double click" to close the "info window" on the map. All users said the website was easier to use than the android app. This is further explained below

Efficiency: The efficiency varied whether it was the android app or the website. The website had a particularly high efficiency. This was improved if the user was acquainted with Google maps previously. The android app was 5 times to 10 times slower than the website on tasks that required map manipulation.

Accuracy: The website was very accurate. Users could put items with the accuracy offered by Google maps. The android app however had a very high inaccuracy. Users settled for lesser accuracy due to the difficulty of manipulating the map.

Recall: All the users could recall the procedures once learnt for both the android and the website.

Emotional Response: Users who were unacquainted with Google maps on either the android or the website were mistakenly very impressed. This is because they thought the Google maps functionality was made by the application that was being tested. Therefore, they could not differentiate which extra functionality was under investigation. Users who were acquainted with Google maps were mildly "happy" with the application. No users in the sample had seen any other similar applications such as the ones in the literature review. So no information was obtained with regard to this.

Intuition: Users who were acquainted with Google maps could complete some tasks without been given all the detail. For example they could zoom out and move the marker without any help. Users who were unacquainted with Google maps required step by step help.

Software performance and Compatibility Testing: Under the minimalistic load under which the tests were done the performance of the system was good. The bottle neck was always the network connection. The website worked under all the major browsers. Google chrome had the most deviant UI. However this did not prevent or hinder any system functionality. The Android application looked different depending on the version of the Operating System only. Higher versions of the android OS gave better quality GUIs. Testing for different manufactures revealed problems with HTC. HTC androids (e.g. HTC Sense) had a problem with the GUI to the point of making the application unusable. This would require some specific code to rectify.

Miscellaneous Results: most users related the application to the social networking that occurs on Facebook. In fact users who were not given the background to the application shared/posted information that amounted to a "status" on Facebook.

4.4 Summary

These are the tests that were conducted and the results that were obtained. There are more helpful tests that could be carried out that were not done. These are security tests and testing the system under larger loads.

5.0 CHAPTER 5 -- DISCUSSION AND CONCLUSION

5.1 Introduction

The tests in the above section were analyzed and the discussion of them follows below. It shows that many areas need improvement and many opportunities are available. Also discovered are the challenges that await as the development of such a system progresses.

5.2 Discussion

We begin the discussion with a review on how well the objectives were met. The objective which was difficult to achieve was the testing of the System. Development of the system was challenging as it involved developing a website and a mobile application which have different mechanisms. The other objectives were well met.

The sample size was also of concern. A larger sample size would be more representative of the Lusaka city population. This being the case the conclusions obtained here are not as verified as they should be. Most results obtained from the testing described previously indicated that the system is workable and only requires having extra work done on it. This is especially the case of the GUI problems. Other problems such as scaling and security are generally difficult to accomplish for any type of application. The system only had the login mechanism as its security. No encryption was done to any of the information.

The android application was discovered to have considerably less usability than the website. This situation is not very surprising as the android platform Google maps API is not as advanced as the web API. Secondly the mobile devices do not have as much computing resources as the computers were the browsers were running. However the android application is of higher priority because ordinary citizens have mobile devices them with them practically all the time and hence will supply news items more frequently using the mobile devices. Therefore it is desirable to make the android application more usable as a higher priority than the website version

The testing revealed that users thought of the application relating it to the famous Facebook. Therefore some of them posted/shared what can be thought of as a Facebook "status update".

Consequently a way must be devised in order to educate users to the intent of the application. However this is not as clear cut as it sounds. This is because trivial news must be supported by the application. However it is difficult to differentiate between a trivial news item and a "status update". An example is "lungwani's wedding, I have never seen better". This contains a "news" item about the wedding and also sounds like a typical post on Facebook. This being true, it is expected that the other big applications (e.g. twitter) would also have an impact over the ways users perceive the use of the application.

Moderation is also required. Zambia is a land having a constitution. The offences against the constitution are not expected from the application but rather from the users of the application. Users could easily begin "dangerous" false information that runs like a wild fire and does much damage. There is a need to be able to moderate this in real time. In addition to this, the system must be in such a way that unscrupulous individuals may be deterred from abusing it. There remains work to be done in this regard.

5.3 Conclusion

In the context of the issues discussed above, the aim was achieved and the hypothesis was proven to be true. However, it is very clear that there is a lot of work to be done in order to have an operational and safe system. Other conclusions are that such a system is feasible using "freely" available resources. Long ago it was hindered by the expensive nature of a mapping service. A full functional and safe system would be very versatile to the public and could be a source of impartial news.

5.4 Future Works

The system could be improved upon in many areas. One of these is by providing a better GUI for the android application. We shall also focus on a recommending system. This recommending system can use location, time and other typical metrics that are used in typical recommending systems. Therefore as a user travels throughout the city he gets recommendation according to his history filtered also by his current location. Consider an item such as "a little boy is crying". If this news item is near enough to a user such that they can easily go to the location, then a user might be interested to find out why the boy is crying. This means that the item was "interesting"

to the user. However if the boy is so far away from the users current location, this will not interest the user at all. Therefore location awareness adds a dimension to the recommending system.

5.5 Summary

The core system functionality was created and many opportunities are available for improvement. However, these opportunities are not cheap and work must be done to achieve them safely.

APPENDIX

The code for the application is contained in a CD disc attached to this thesis. The amount of code was too much for this section to even adequately cover it. Therefore we only present some of the essential code here.

We show some code for the website application. It was created using Google Web Toolkit. Google Web Toolkit uses java to program and its compiler translates this to HTML and JavaScript. This explains why we will show java code for an application that should run on a web browser.

Below we show the class that handles the manipulation of the map on the browser

```
package zm.co.ziko.gwt.client.fragment;


import zm.co.ziko.gwt.client.NXInfoWindow;

import zm.co.ziko.gwt.client.datamodel.NewsGWTX;

import zm.co.ziko.gwt.client.ui.InfoWindowUI;

import zm.co.ziko.gwt.client.ui.InfoWindowUIImpl;

import zm.co.ziko.gwt.client.ui.NewsUI;

import zm.co.ziko.gwt.client.ui.NewsUIImpl;


import com.google.gwt.core.shared.GWT;

import com.google.gwt.dom.client.Node;

import com.google.gwt.user.client.Window;

import com.google.maps.gwt.client.GoogleMap;
```

```
import com.google.maps.gwt.client.GoogleMap.MouseMoveHandler;

import com.google.maps.gwt.client.GoogleMap.RightClickHandler;

import com.google.maps.gwt.client.DirectionsRenderer;

import com.google.maps.gwt.client.DirectionsRequest;

import com.google.maps.gwt.client.DirectionsResult;

import com.google.maps.gwt.client.DirectionsService;

import com.google.maps.gwt.client.DirectionsStatus;

import com.google.maps.gwt.client.InfoWindow;

import com.google.maps.gwt.client.InfoWindowOptions;

import com.google.maps.gwt.client.LatLng;

import com.google.maps.gwt.client.MapOptions;

import com.google.maps.gwt.client.MapTypeId;

import com.google.maps.gwt.client.Marker;

import com.google.maps.gwt.client.MarkerOptions;

import com.google.maps.gwt.client.MouseEvent;

import com.google.maps.gwt.client.TravelMode;


public class ZikoMapImpl implements ZikoMap {


      private GoogleMap googleMap;//the google map displayed

      private final GoogleMap mapNull = null;
```

```
private MapOptions mapOptions; //the settings on the google map

private Node mapNode;

private LatLng location;      //location on the map

private double latitude = -15.40;

private double longitude = 28.30;

private double zoom = 12;

private Marker shareMarker;//markers that are placed on the map

private Marker directionsMarker;//marker used for directions


InfoWindow[] infoWindowArray;

NXInfoWindow[] nxInfoWindows;


Marker[] markerArray;

//displays directions once the end points are known

private DirectionsRenderer directionsDisplay;


//this method clears all the markers on the map

@Override

public void clearMap() {


      for (int i = 0; i < markerArray.length; i++) {
```

```
            markerArray[i].setMap(mapNull);
            markerArray[i] = null;
      }

      markerArray = new Marker[0];
}

//used to load the google map from the google servers
@Override
public void addGoogleMap(Node mapNode) {
      try {
            this.mapNode = mapNode;
            googleMap = GoogleMap.create(this.mapNode, mapOptions);
      } catch (Exception e) {
            GWT.log("failed to create Map");
      }
}

public ZikoMapImpl() {
```

```
        initialiseMapOptions();


}
//this initializes the initial map that is drawn
private void initialiseMapOptions() {
        try {
                location = LatLng.create(latitude, longitude);
                mapOptions = MapOptions.create();
                mapOptions.setZoom(zoom);
                mapOptions.setCenter(location);
                mapOptions.setMapTypeId(MapTypeId.ROADMAP);
        } catch (Exception e) {
                GWT.log("failed to create map options");
        }
}


//this is used choosing a location for a news item
@Override
public void enableShareMarker() {
        shareMarker = Marker.create();
        shareMarker.setPosition(LatLng.create(-15.4, 28.3));
```

```
shareMarker.setMap(googleMap);

googleMap.addMouseMoveListener(new MouseMoveHandler() {

	@Override
	public void handle(MouseEvent event) {
		LatLng mouseLocation = event.getLatLng();
		// shareTArea.setText(mouseLocation
		// .toUrlValue(10.000000001));

		shareMarker.setPosition(mouseLocation);

	}
});

googleMap.addRightClickListenerOnce(new RightClickHandler() {

	@Override
	public void handle(MouseEvent event) {
		googleMap.clearMouseMoveListeners();
```

```
            }
        });

}

//this is used for getting the direction location
@Override
public void enableDirectionsMarker() {
        directionsMarker = Marker.create();
        directionsMarker.setPosition(LatLng.create(-15.4, 28.3));
        directionsMarker.setMap(googleMap);

        googleMap.addMouseMoveListener(new MouseMoveHandler() {
                LatLng mouseLocation;

                @Override
                public void handle(MouseEvent event) {
                        mouseLocation = event.getLatLng();
                        directionsMarker.setPosition(mouseLocation);

                }
```

```
        });

        googleMap.addRightClickListenerOnce(new RightClickHandler() {

                @Override
                public void handle(MouseEvent event) {
                        googleMap.clearMouseMoveListeners();
                }
        });

}

@Override
public LatLng getShareLocation() {
        if (shareMarker == null) {
                return null;
        }

        return shareMarker.getPosition();
}
```

```
@Override
public LatLng getDirectionsLocation() {
	if (directionsMarker == null) {
		return null;
	}
	return directionsMarker.getPosition();
}

@Override
public void disableShareMarker() {
	googleMap.clearMouseMoveListeners();

	googleMap.clearRightClickListeners();
	GoogleMap mapnull = null;
	shareMarker.setMap(mapnull);
	shareMarker = null;
}

@Override
public void disableDirectionsMarker() {
	googleMap.clearMouseMoveListeners();
```

```
        googleMap.clearRightClickListeners();
        GoogleMap mapnull = null;
        if (directionsMarker != null) {
              directionsMarker.setMap(mapnull);
        }
        directionsMarker = null;
    }
    @Override
    public void getDirections(LatLng origin , LatLng destination) {

        DirectionsService directionsService = DirectionsService.create();
        directionsDisplay = DirectionsRenderer.create();
        directionsDisplay.setMap(googleMap);
        DirectionsRequest request = DirectionsRequest.create();
        request.setOrigin(origin);
        request.setDestination(destination);
        request.setTravelMode(TravelMode.DRIVING);
        directionsService.route(request, new DirectionsService.Callback()
{
```

```
                @Override
                public void handle(DirectionsResult a, DirectionsStatus b)
{
                      if (b == DirectionsStatus.OK) {
                            directionsDisplay.setDirections(a);
                      } else {
                            Window.alert(b.getValue());
                      }
                }
          });

      }

// this is used to add the news markers on the map
      @Override
      public void addNews(NewsGWTX[] result) {
            nxInfoWindows = new NXInfoWindow[result.length];
            infoWindowArray = new InfoWindow[result.length];
            markerArray = new Marker[result.length];

            for (int i = 0; i < result.length; i++) {
```

```
                LatLng newsLocation = LatLng.create(result[i].latitude,
                                result[i].longitude);


                InfoWindowOptions iWOptions = InfoWindowOptions.create();
                iWOptions.setContent(i + " " + result[i].text);


                InfoWindowUI infoWindowUI = new
InfoWindowUIImpl(result[i]);


                nxInfoWindows[i] = NXInfoWindow.create((long) i);
                infoWindowUI.setNxInfoWindow(nxInfoWindows[i]);
                nxInfoWindows[i].setContent(infoWindowUI.getWidget());
                nxInfoWindows[i].setPosition(newsLocation);


                infoWindowArray[i] = InfoWindow.create(iWOptions);


                MarkerOptions mOptions = MarkerOptions.create();
                mOptions.setPosition(newsLocation);
                mOptions.setMap(googleMap);
                mOptions.setTitle(i + "");
```

```
                markerArray[i] = Marker.create(mOptions);

                markerArray[i].addClickListener(new ZikoCHandler(i,
nxInfoWindows,
                        markerArray));

        }

    }

    private void addNewsPlain(NewsGWTX[] result) {
        GWT.log("AllNewsCB " + result[0].text);

        infoWindowArray = new InfoWindow[result.length];
        markerArray = new Marker[result.length];

        for (int i = 0; i < result.length; i++) {
            LatLng newsLocation = LatLng.create(result[i].latitude,
                    result[i].longitude);

            InfoWindowOptions iWOptions = InfoWindowOptions.create();
```

```
                iWOptions.setContent(i + " " + result[i].text);

                infoWindowArray[i] = InfoWindow.create(iWOptions);

                MarkerOptions mOptions = MarkerOptions.create();
                mOptions.setPosition(newsLocation);
                mOptions.setMap(googleMap);
                mOptions.setTitle(i + "");

                markerArray[i] = Marker.create(mOptions);

                markerArray[i].addClickListener(new ZikoCHandler(i,
nxInfoWindows,
                        markerArray));

        }

    }

    //creates the info window once it is clicked
    class ZikoCHandler implements Marker.ClickHandler {
```

```
        int id;
        InfoWindow[] infoWindows;
        private Marker[] markers;
        NXInfoWindow[] nxInfoWindows;

        public ZikoCHandler(int i, InfoWindow[] infoWindows, Marker[]
markers) {

            id = i;
            this.infoWindows = infoWindows;
            this.markers = markers;

        }

        public ZikoCHandler(int i, NXInfoWindow[] nxInfoWindows,
                Marker[] markers) {

            id = i;
            this.nxInfoWindows = nxInfoWindows;
            this.markers = markers;
```

```
            }


            public void handle(MouseEvent event) {

                  nxInfoWindows[id].open(googleMap, markers[id]);

            }

      }


}
```

References

[1] Y.Manolopoulos, P. Symeonidis, A. Papadimitriou. (2011). *Geo-Social Recommendations.*

[2] S. Bowman, C. Willis. (2003).*We Media: How audiences are shaping the future of news and information.* The American Press Institute.

[3] Stuart Allan. (2007). *Citizen journalism and the Rise of "Mass Self-Communication": Reporting the London bombings.* Global Media Journal ,Australian Edition, Issue 1, Volume 1

[4]. D.J. Krajicek. (2012). *The Social Media Revolution In Breaking-News Journalism: Tips From the Front Line.*

[5] http://mashable.com/2012/04/18/social-media-and-the-news/

[6] http://www.ohmynews.com/ (English version is http://international.ohmynews.com/)

[7] T. O'Reilly. (2005) *What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software.*

[8] Florian Fisher.(2009). *Learning in Geocommunities an explorative view on geo-social network communities.* Learning with GeoInformation IV:12-21. Wichmann Verlag.

[9] Axel Bruns. (2009). *News blogs and Citizen journalis: New directions for e-Journalism.*Queens Land Institute of technology

[10] Chris Roberts.(2005). *Gatekeeping theory: an evolution*

[11] P. Shoemaker, P. Johnson. (2010). *Readers as gatekeepers of online news.*

[12] Earl Rennison ,(1994), *Galaxy* of *News* An *Approach* to *Visualizing* and *Understanding Expansive News Landscapes.* User Interface Software and Technology

[13] Hangzai Luo. (2007). *Analyzing Large-Scale News Video Databases to Support Knowledge Visualization and Intuitive Retrieval.* Visual Analytics Science and Technology, 2007. VAST 2007.

[14] Derya Ozkan, Pınar Duygulu .(2006). *Finding People Frequently Appearing in News.* Lecture Notes in Computer Science Volume 4071, 2006, pp 173-182

[15] R.E. Roth, K. S. Ross.(2009). *Extending the Google Maps API for Event Animation Mashups.* Cartographic perspectives.

[16] M. Heyman, D. Sheesley.(2008). *Maker! Mapping the world's data.* NACIS 2008. Missoula, MT.

[17] Mcconchie, A. L. (2008) *Mapping Mashups: Participation, collaboration, and critique on the World Wide Web.* PhD Dissertation, Geography. Vancouver, Canada, The University of British Columbia.

[18] MILLER, C. C. (2006) *A beast in the field: The Google Maps mashup as GIS/2.* Cartographica,41, 187-199

[19] E. Pietroniro, D. Fichter. (2007). *Map mashups and the rise of amateur cartographers and map makers.* ACMLA Bulletin No 127

[20] J. Wong, J. Hong. (2008). *Patterns in Mashups.*

[21] http://www.housingmaps.com

[22] http://www.zhappening.com

[23] https://foursquare.com/

[24] https://developers.google.com/maps/documentation/javascript/tutorial

[25] https://developers.google.com/maps/documentation/android/

[26] http://www.android.com/

[27] http://mashable.com/2012/11/14/android-72-percent/

[28] W. Lee. (2012). *Beginning Android 4 Development.* John Wiley & Sons, Inc.Svennerberg. (2010). *Beginning goggle maps API 3.* Apress

[29] http://zaplaces.com/about/

[30] https://www.facebook.com/pages/Zaplaces/131976673573516

[31] https://www.facebook.com/pages/Zhappening/229794963697268?fref=ts

[32] http://zhappening.com/