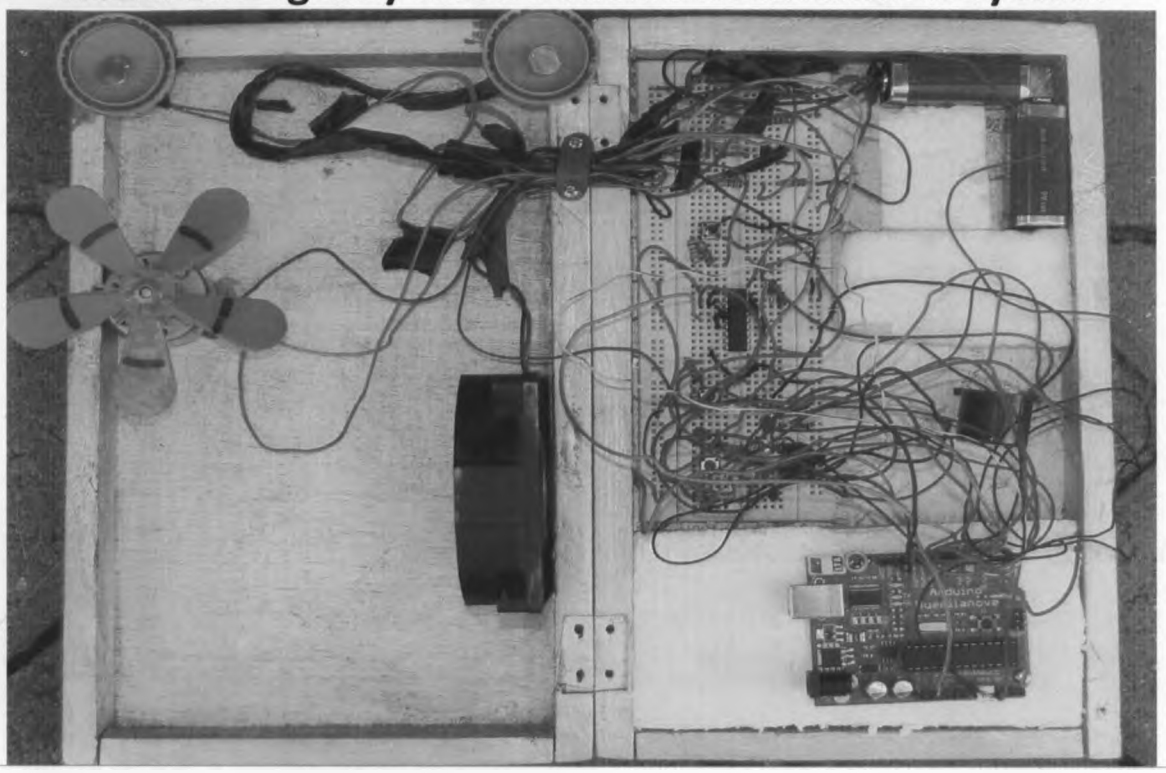
UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING & INFORMATISC
MSC INFORMATION SYSTEMS

TITLE: Emergency Information Dissemination System



By

James W. Kivuva
REG.NO. P56/P/7614/06

SUPERVISOR

DR. WAGACHA PETER W

February 2010

Submitted in Partial Fulfillment of the requirement of the Master of Science Information Systems

DECLARATION

This work is my original work and it has never been presented to any other institution for the award of any certificate whatsoever.

Signature_____ Date 08/09/10
James W. Kivuva
REG.NO. P56/P/7614/06

This project has been presented for examination with my approval as the students Supervisor.

Signed_____ Date 08/09/10
Dr Peter W. Wagacha

School of Computing & Informatics
University of NAIROBI
P. O Box 30197
NAIROBI

Acknowledgements

I would like to thank my supervisor Dr Peter. W. Wagacha for his quality guidance and encouragement through out the course of this project.

It is also in order to thank Ms E.M. Masinde for her support and encouragement especially at the project concept preparations and website design. I also would wish thank Professor Timothy M. Waema for the support he gave me.

My gratitude to my family for giving me the necessary support and the Kenya meteorological staff who assisted me in basic electronics training and logistic support .

Abstract

The problem under investigation is on the development of emergency information dissemination. Kenya is faced with many disasters and emergency. There is no clear boundary between an emergency and a disaster except in the magnitude. While countries in developed world and developing world like India have put in place such systems, it is worrying that most countries in Africa are ill prepared to deal with disasters.

The project has investigated the mechanisms of setting up an emergency information system by comparing with what is already developed by developed countries. A framework on the modules required for a successful system has been developed with consultations with the National Operational Center (NOC) the government organisation that is charged with coordinating rescue missions in Kenya.

While the project has demonstrated the opportunity that exist like the use of sensors to monitor events like fire or flood and relay that information when such event occurs in real time, there are many challenges. The Emergency Information Management System is huge and complex. From the literature on existing systems, its clear that to devlop such a system is not cheap and this could explain why poor countries are yet to develop an integrated solution

Table of Contents

List of Figures

Chapter 1. Introduction

1.0 Background

Kenya has experienced numerous disasters, which have resulted to loss of lives and property. The 2009 fires at the Nakumat supermarket and the Nakuru fire tragedy have showed that the country is poorly prepared to cope with arising disasters. Natural and man-made disasters, such as floods, droughts, plane crashes, earthquakes, high-rise building collapses, and fires, pose an ever-present challenge to public emergency services. In order to cope with such disasters, provision of information concerning the situation is an essential pre-requisite. Police, fire departments, public health, civil defence and other organizations have to react not only efficiently and individually, but also in a coordinated manner. Since coordination requires current information, and such information must be communicated upstream and downstream within and between organizations in real-time, the need arises for an integrated communication and information system for emergency management that provides efficient, reliable and secure exchange and processing of relevant information.

There are many technologies in existence such as emails and short messages (SMS) which can be used to warn people on arising hazards such as: a washed bridge or heavy rains that are likely to cause flooding. If such technologies are put into use, Kenya and its people will be more prepared to deal with any arising emergencies.

This project has demonstrated how to develop an intelligent fire fighting system. With the help of a SMS server, Web server and a microntoller with fire sensors, the project has demonstrated how fire fighting can combine the modern technologies where the fire alarm

can do more than just raise the alarm. In this study it has been demonstrated of a fire alarm that once it senses a fire it activate fire fighting mechanisms like opening the water nozzles and at the same time calling for more help by sending an SMS to the fire department.

1.1 Problem Statement

There are numerous events reported in Kenya including floods, droughts, accidents, fires and conflicts which creates a emergency situations. There has been an observed poor coordination when disaster strikes. (e.g. the 2009 fire tragedies of Nakummat Nairobi and oil tanker fire Nakuru, the Kiambu house collapse among others.) There is no Emergency Information dissemination System in place, which could be used to inform the public of an emergency situation or a an imminent disaster situation and give them a detailed information on how to deal with the situation. When a disaster strikes, information flow to the affected victims and the general public is crucial and can result in the reduction of loss of lives and property (Management, 2002)(Manitoba Health Disaster Management 2002).

1.2 Main Objective:

- To develop an emergency information dissemination system

1.2.1 Specific Objectives:

- To explore the technologies that can be used in emergency/disaster information dissemination systems.
- Develop a web based application that can send SMS, email for emergency information dissemination
- To explore how to integrate GIS with Emergency Information Dissemination System.

1.3 Justification

Kenya spends billions of shillings when disaster and emergencies strikes and is ill prepared to deal with them. There is no emergency information dissemination system in place. The use of such a system can assist to mitigate disasters and thus reduce the impact.

Information technology is coming to age in Kenya. There are tremendous benefits, which can be realised with the use of the ICT technologies in disaster management such as efficient information distribution, an integrated platform for sharing disaster information and cost effectiveness among others. While other countries, which are disaster prone like Kenya, have already seen the benefits and consequently utilized these technologies, Kenya has not yet tapped these resources.

1.4 Scope

The project is limited to addressing emergencies in Nairobi and has given more emphasis on fire emergencies. The project address the information dissemination module.

2.0 Literature Review

2.1 EMIS and emergencies

Emergencies are the routine events that local police, fire and emergency medical service personnel handle on a day to day basis. On the other hand, disasters are events that require more resources than a local jurisdiction has and will therefore require outside assistance. Emergency Managers might logically be called "Disaster Managers" as their role is to coordinate resources for larger events. An emergency is "a sudden, urgent, usually unexpected occurrence requiring immediate action". A disaster is "a calamitous (great misfortune) event, especially one occurring suddenly and causing great damage". Both emergencies and disasters can begin suddenly, but disasters usually result in a larger impact to the community when compared to emergencies. Since when an event occurs nobody can tell whether it is a disaster or not until some preliminary assessment of impact is done the information dissemination in both cases is not different.

There are many different type of Emergency Information systems (EMIS) (Wikipedia) and the main ones are the EMIS for Preparedness, the EMIS for Risk Mitigation, the EMIS for Response and the EMIS for Recovery. The EMIS for Preparedness is concerned with the preparation of contingency plans, stakeholder management and resource management. The EMIS for Risk Management is tasked with the determination of the possible risk areas and/or risk types and is often supported by a geographical information system (GIS). The EMIS for Response is concerned with the execution and tracking the contingency plans while the EMIS for Recovery does the cost calculation of the emergency and generation of reports (supported by tables, graphs, etc.) .An EMIS interacts with many other early alert systems and communicates with many legacy systems.

Other emergency information includes the Logistics support system (LSS) which is a tool created for the purpose of facilitating the communication among humanitarian agencies, non-governmental organizations, donors, media or countries and enhancing the coordination of humanitarian supply chain. Logistics support systems have the ability to provide decision-supporting reports that show which relief supplies were received or delivered, where they have been stored, which relief supplies have been requested and how useful they are at a specific moment. Consequently, the function of the logistics support system enhances the proactiveness of a humanitarian organization by improving problem traceability and performance monitoring and providing the advantage to timely change the plan of the supply chain without consequenses.

Two modules of logistics support systems have been created: a module that operates under Windows systems and a second module that operates under Internet Explorer. Logistics support systems have been used in many natural disasters, including Pakistan earthquake in October 2005 and Bolivian floods in February 2006.

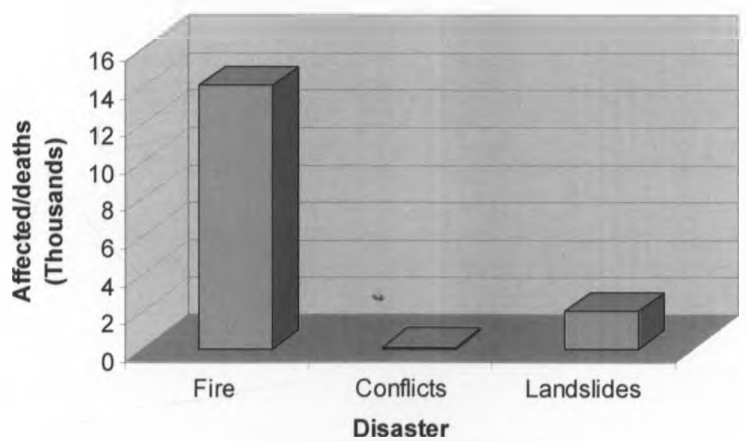


Figure 1: People affected by disaster in Kenya

Source: Disaster Risk Reduction Strategy for Kenya 2006-2016

As is shown in figure one above fire affects many people in Kenya. The same scenario is also found in Nairobi where most of the emergency incidents are caused by fires.

2.2 Institutional Framework

The National (Disaster) Operational Center (NOC) is charged with the overall responsibility of coordinating emergencies and disaster. It was established on 21st January 1998 following the devastating effects of the El Niño rains. It is manned on a 24-hour basis by officers drawn from various ministries and departments of the Government such as: the Department of Defence, the police and the Ministry of Health.

2.3 Information Dissemination systems.

Information plays a vital role. People are warned of imminent disaster by disaster managers. Disaster managers will therefore require communication platform than can reach people in need of advisory information during an emergency. Such information can reduce the level of anxiety and also alert others nearby on how to take precautionary measures and possibily avoid being affected altogether. Some of the technologies that we will review will include the electronic mail, the short message services (SMS) and the web services.

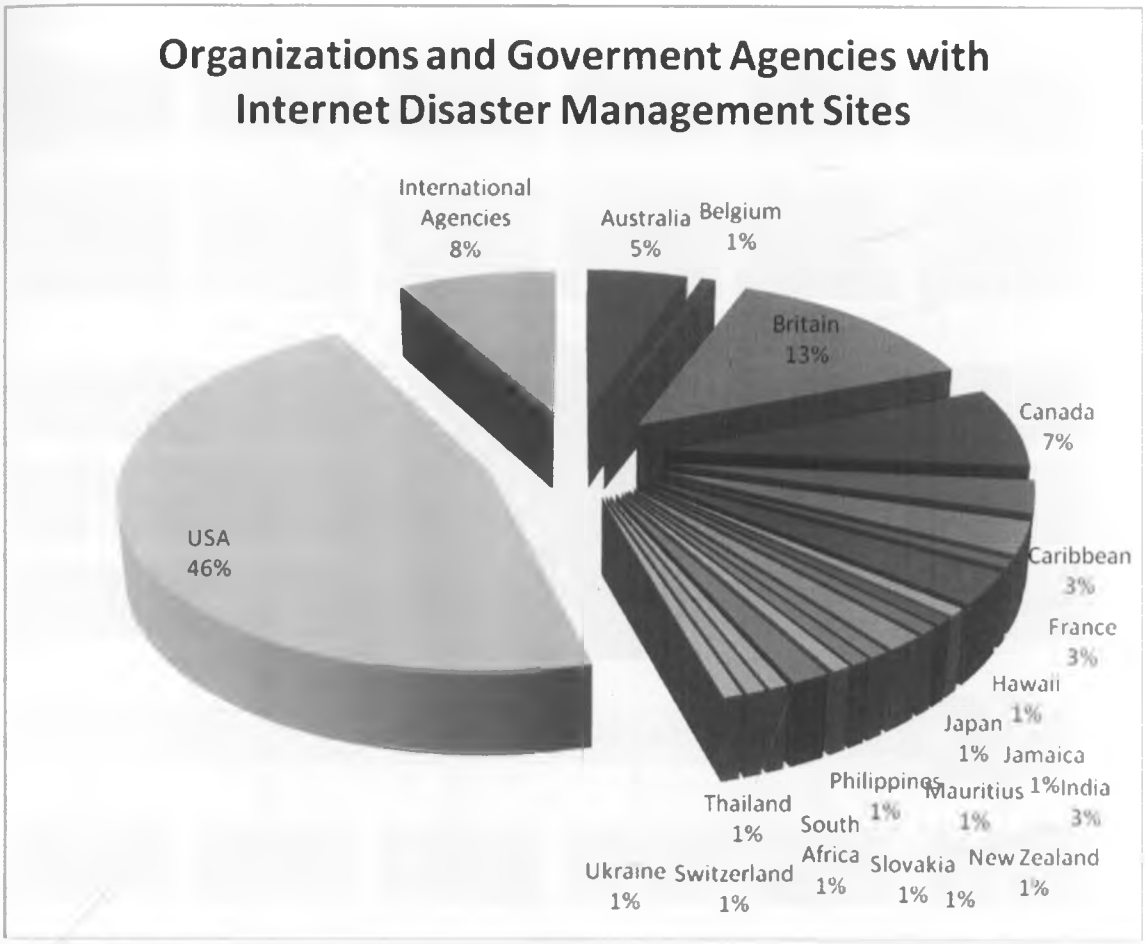


Figure 2: Global internet usage in disaster management

Source: KEELE University UK

As shown in figure 2 above internet usage in disaster management is done in developed countries. The USA is the country with the highest number of web based disaster information systems. Despite the fact that the USA is faced with numerous weather related disasters such as hurricanes it has used information technology to manage them. The internet usage in disaster/emergency management in Kenya and other African countries is not clear. South Africa is the only country in Africa with an internet based disaster management information system.

2.4 Information management and Communication

Integrated disaster risk management depends on access to reliable hazard and disaster risk information as well as effective information management and communication systems to enable the receipt, dissemination and exchange of information.

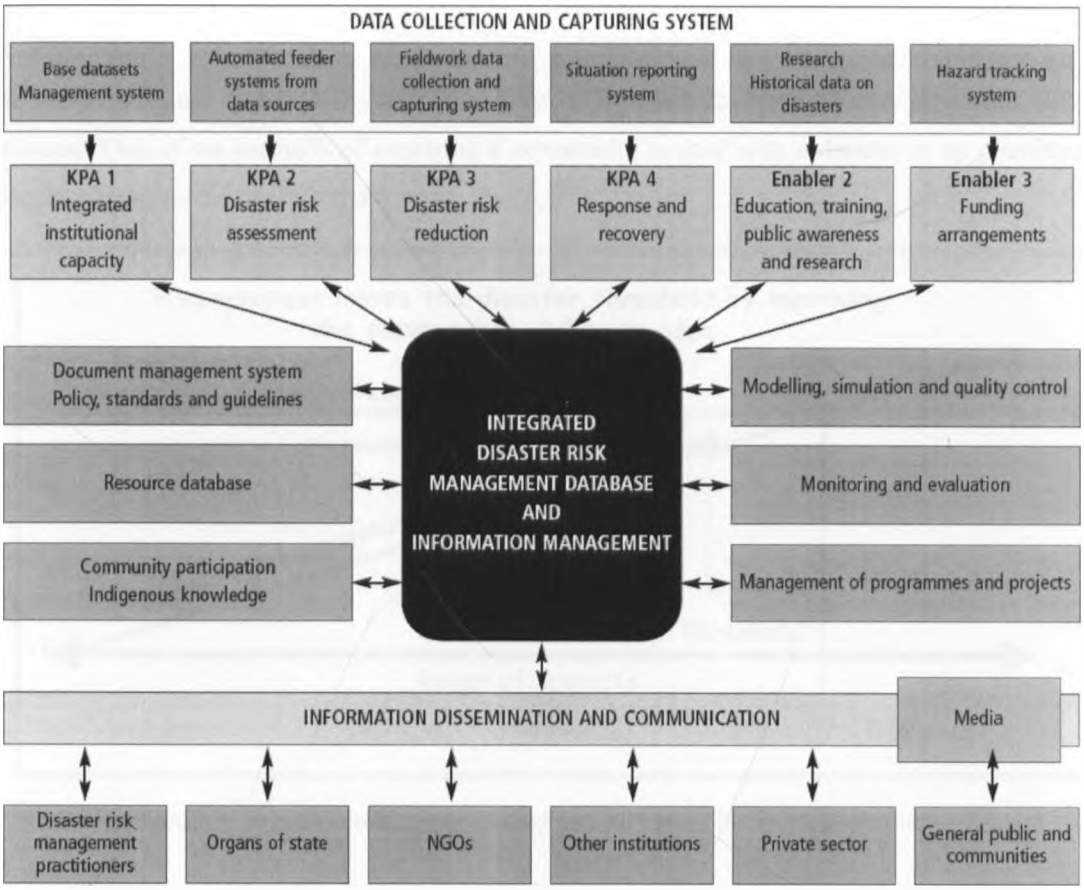


Figure 3 Information Dissemination system for Manitoba Heath Disaster management

Source: Manitoba Health Disaster Management 2002

Figure 4 below shows that if a community is better prepared that Disaster threshold will be high.

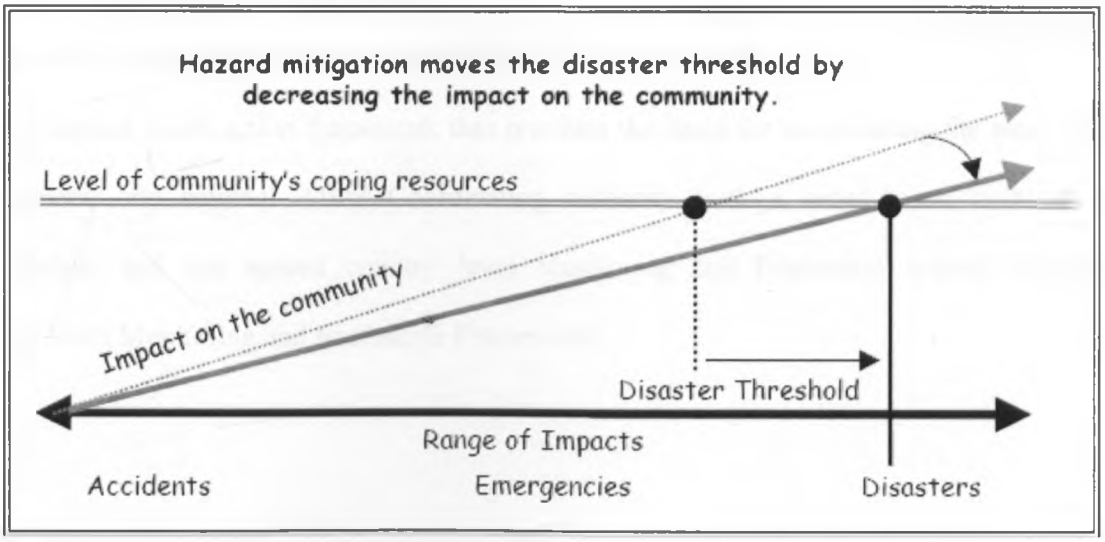


Figure 4 Relationship between community coping capacity and disaster

Source: Manitoba Health Disaster Management 2002

As show in figure 5 below, if the community is prepared, then the disaster threshold will also increase. One of the methods of preparing a community to deal with a disaster is by providing the community with relevant information and advice,

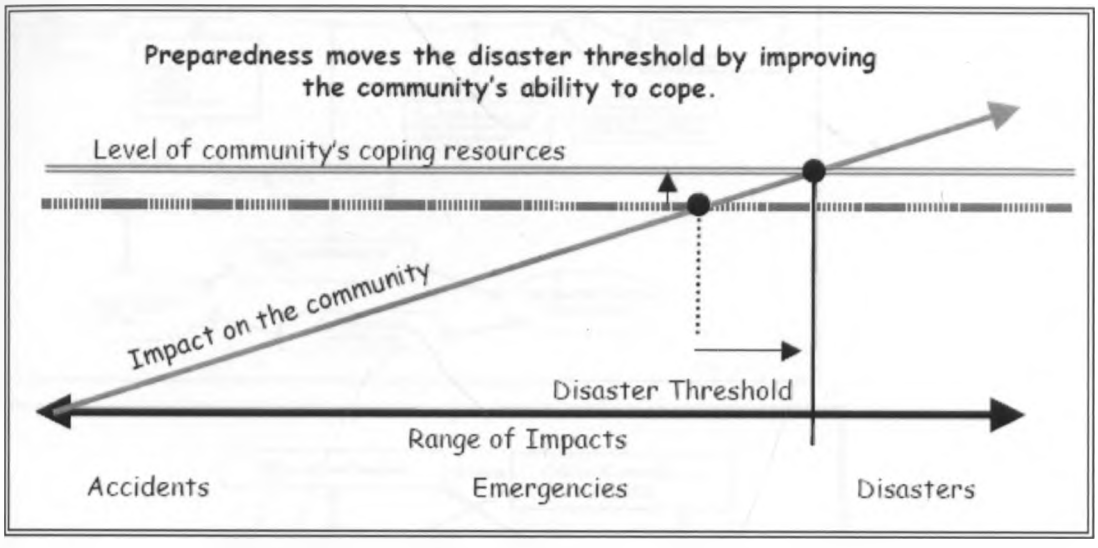


Figure 5 Relationship between community coping capacity and disaster preparedness

Source: Manitoba Health Disaster Management 2002

2.4.1 HIV/Aids Coordination

The Aids coordination is based on the *"Three Ones" Principles:*

One agreed AIDS action framework that provides the basis for coordinating the work of all partners, one national AIDS Coordinating authority with a broad-based multispectral mandate, and one agreed country level Monitoring and Evaluation system. (National HIV/Aids Monitoring and Evaluation Framework)

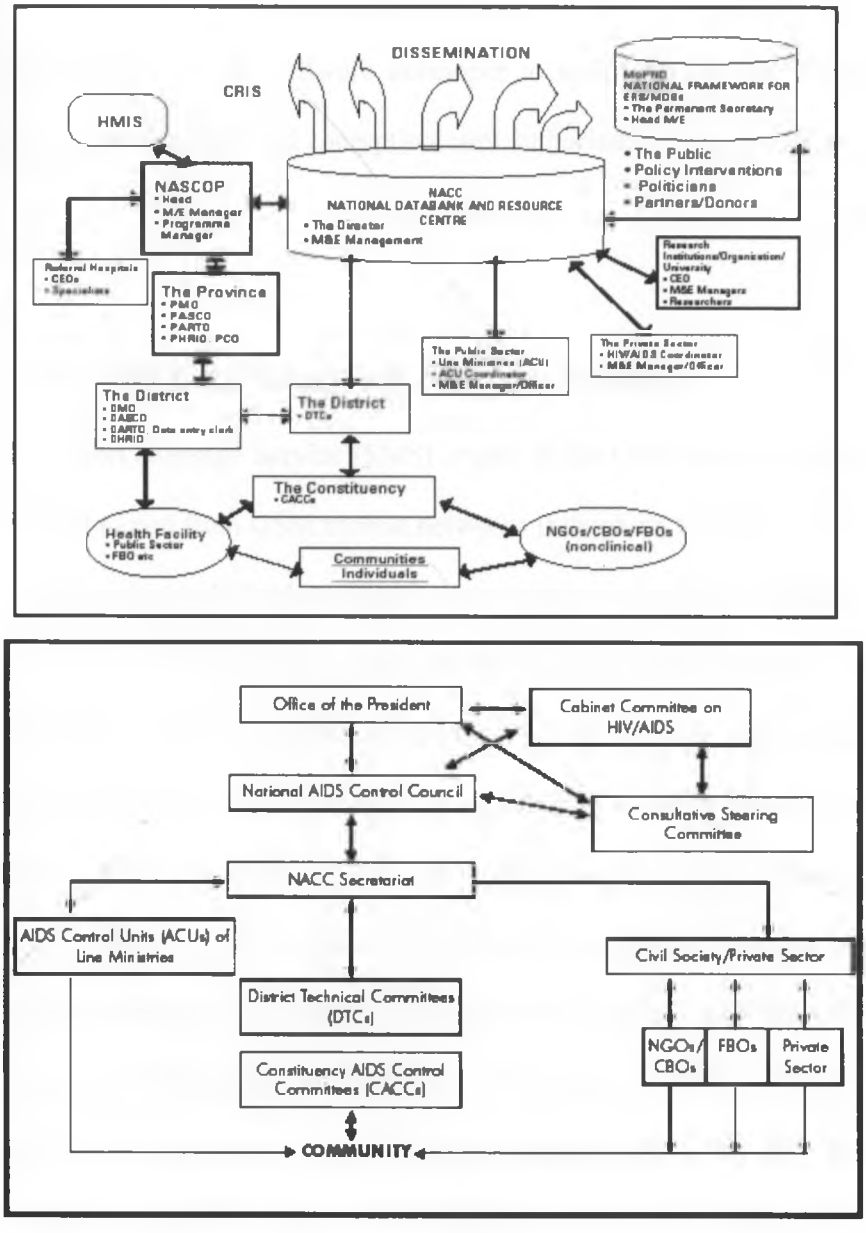


Figure 6: Aids Coordination organization Structure

Source National Hiv/Aids Monitoring And Evaluation Framework 2005

2.5 System Security

In order for information to be credible, there is need to ensure that the disseminated information is accurate and secure. This requires a secure information system that does not allow unauthorized users to log into the system and by so doing, corrupt or send false alarms.

PHP which is the main scripting language used is rich in security features. However it is the responsibility of the software developer to apply the security features. The main security feature are the password encryption, user authorisation, access control and the SQL Injection.

2.6 SMS Messaging Server Design

The Short Message Service (SMS) is part of the GSM specification and allows messages to be sent to and from GSM mobile networks throughout the world. A single short message can contain up to 160 characters and comprise of words, numbers or an alphanumeric combination. SMS also provides confirmation that a short message, has been delivered to its destination. SMS is a store and forward service where a short message is sent via a Short Message Service Center (SMSC). An advantage of this is that the destination mobile device does not have to be on the network at the time when the message is sent. If a destination mobile device is not available at the time the message is sent, the SMSC service centre will retry to deliver the message. SMS Messaging Server uses the GSM modem to send and receive SMS Messages. It supports a broad range of GSM modems that support the ETSI GSM 07.05 guidelines. The SMS Messaging consists of the SMS Messaging Server service which is a multi-threading SMS Messaging engine, a Configuration Database, a Message Database , Applications and Scripts, GSM Modem channels and VBScript Triggers to process incoming messages; Figure 7 below show the structure of the GSM gate way.

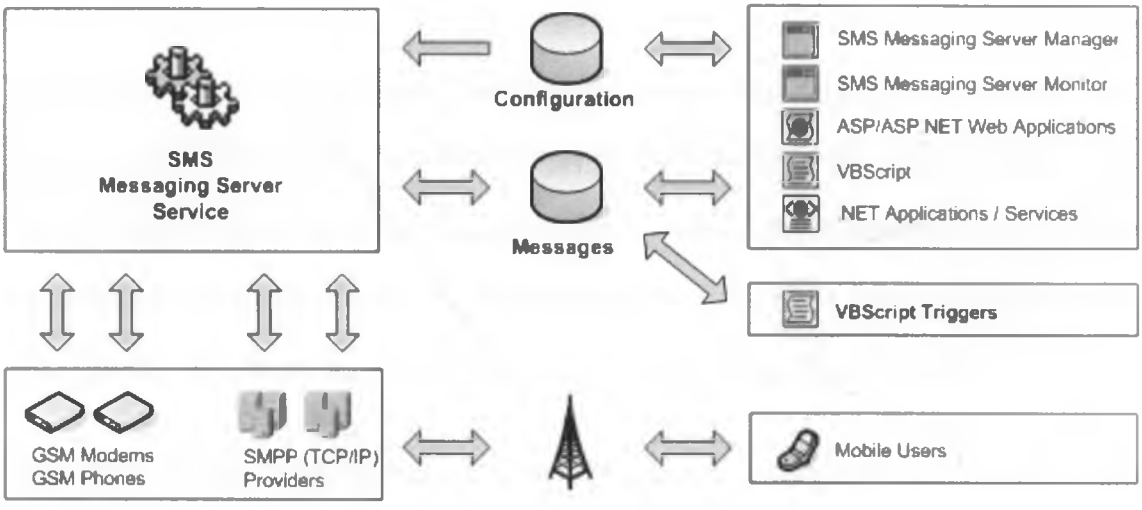


Figure 7: GSM Gateway

Source Active Experts

2.7 Related research

Indonesia has implemented a Disaster Management Information System and the following Achievements have been realised

1. Disaster management information system (including web-server at government) and network established among government agencies, donors, academics and NGOs and general public.
2. A website to access the information system completed with general organizational information, reports and data, coordination information, disaster reports, internal information section.
3. Four modules (web-based) developed: disaster prevention, volcano, earthquake and social conflict.

India, Afghanistan and Turkey have also established a disaster management information system.

Mr. Nathan Eagle of the Massachusetts Institute of Technology (MIT) and University of Nairobi researchers conducted a research on medical data collection using the mobile phones

to collect Malaria related data in Kilifi Kenya. The mobile based system developed is used by KEMRI and has replaced the paper-based system which was used by field workers. This system is more efficient and cost effective than the paper based system previously used. The mobile system is faster and allows the field workers to collect more data in the field and send the same information immediately to Nairobi from the field unlike the paper based system which used to take weeks before the information reached Nairobi from the field.

Kenya has witnessed a rapid growth of the mobile telephone industry from as low as less than 200,000 in the late ninety's to more than 7 million subscribers to date. The increase in demand of mobile phones services has brought about a corresponding increase in mobile phones related business. Recent research has shown that adding an additional ten mobile phones per 100 people boosts a typical developing country's GDP growth by 0.6 percent.

2.8 Alarm systems

Alarm system is composed of a central monitoring system which monitors heat and smoke sensors. When a sensor detects fire this information is relayed to the central processing microcontroller which activates an alarm. There are two main types of smoke detectors: ionization detectors and photoelectric detectors. A smoke alarm uses one or both methods, sometimes plus a heat detector, to warn of a fire. The devices may be powered by a 9-volt battery, lithium battery, or 240-volt Ac input. Ionization detectors have an ionization chamber and a source of ionizing radiation. The source of ionizing radiation is a minute quantity of americium-241 (perhaps 1/5000th of a gram), which is a source of alpha particles (helium nuclei). The ionization chamber consists of two plates separated by about a centimeter. The battery applies a voltage to the plates, charging one plate positive and the other plate negative. Alpha particles constantly released by the **americium** knock electrons off of the

atoms in the air, ionizing the oxygen and nitrogen atoms in the chamber. The positively-charged oxygen and nitrogen atoms are attracted to the negative plate and the electrons are attracted to the positive plate, generating a small, continuous electric current. When smoke enters the ionization chamber, the smoke particles attach to the ions and neutralize them, so they do not reach the plate. The drop in current between the plates triggers the alarm.

In one type of photoelectric device, smoke can block a light beam. In this case, the reduction in light reaching a photocell sets off the alarm. In the most common type of photoelectric unit, however, light is scattered by smoke particles onto a photocell, initiating an alarm. In this type of detector there is a T-shaped chamber with a light-emitting diode (LED) that shoots a beam of light across the horizontal bar of the T. A photocell, positioned at the bottom of the vertical base of the T, generates a current when it is exposed to light. Under smoke-free conditions, the light beam crosses the top of the T in an uninterrupted straight line, not striking the photocell positioned at a right angle below the beam. When smoke is present, the light is scattered by smoke particles, and some of the light is directed down the vertical part of the T to strike the photocell. When sufficient light hits the cell, the current triggers the alarm.

2.9 An Application Programming Interface (API)

An application-programming interface (API) is a set of programming instructions and standards for accessing a Web-based software application or Web tool. A software company releases its API to the public so that other software developers can design products that are powered by its service.

For example, Amazon.com released its API so that Web site developers could more easily access Amazon's product information. Using the Amazon API, a third party Web site can post direct links to Amazon products with updated prices and an option to "buy now."

An API is a software-to-software interface, not a user interface. With APIs, applications talk to each other without any user knowledge or intervention. When you buy movie tickets online and enter your credit card information, the movie ticket Web site uses an API to send your credit card information to a remote application that verifies whether your information is correct. Once payment is confirmed, the remote application sends a response back to the movie ticket Web site saying it's OK to issue the tickets.

2.10 Entity Relationship Model

An Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

Information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Sometimes, both of these phases are referred to as "physical design". There are a number of conventions for entity-relationship diagrams a summary is show in the diagram below.

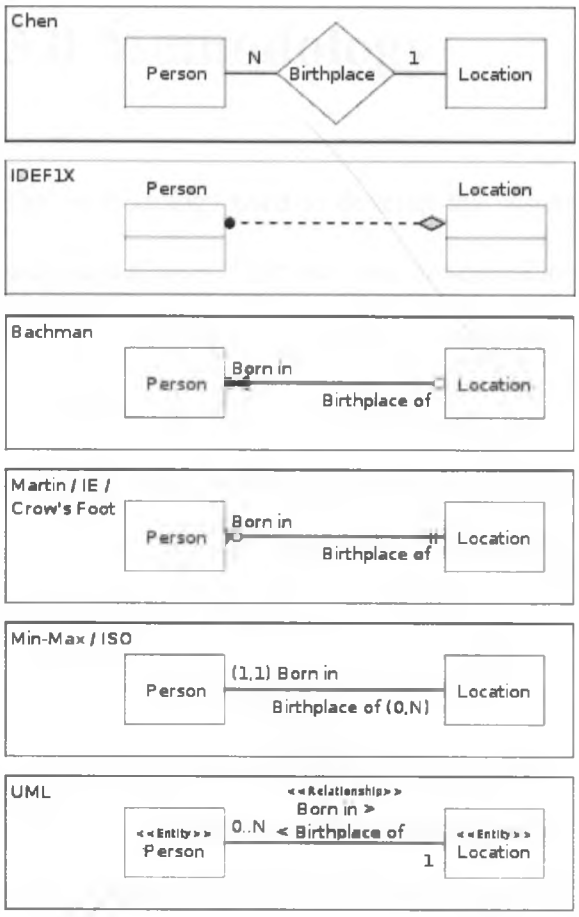


Figure 9: Entity Realtionship diagrams

Chen's notation for entity-relationship modeling uses rectangles to represent entities, and diamonds to represent relationships. Crow's Foot diagrams represent entities as boxes, and relationships as lines between the boxes.

3.0 Methodology

The methodology used to develop the system was prototyping. This method was used on incremental basis. First the basic dissemination modules were developed and tested and there after the modules were integrated to produce the final prototype. During the feasibility stage I was able to develop a conceptual frame work to address the required system. The prototype used was a throw away. In the first prototype, the basic requirements were simulated and tested. The basic requirements were the SMS module, the Email module and the Website Module.

A framework was developed and this resulted in the development of a second throw away prototype to address additional functionalities. The frame work addressed additional issues like how fire sensors can report emergencies/disasters. The use of smoke sensor made that a new module had to be developed to simulate how smoke sensors can be able to detect fire and send an SMS alert. In order to develop a good website interface a new user interface was developed with dropdown menus which could be more easily extended.

During the development of the SMS module, four SMS gateways were evaluated. This gateways included the Ozeki, the ActiveXperts, the Clickatell and the Kannel . The ActiveExperts was selected since it had a module for developing an SMS trigger. The SMS trigger is program that reads an incoming SMS and a program is written to process the incoming SMS message. Clickatell was the fastest method in sending an SMS since it uses the HTTP protocol. However dispite the fact that Clickatell is good it is not flexible since it is based abroad and requires electronic payments. The Kannel is very difficult to learn while the Ozeki open source version did not have all the modules. Advantages of using a local SMS

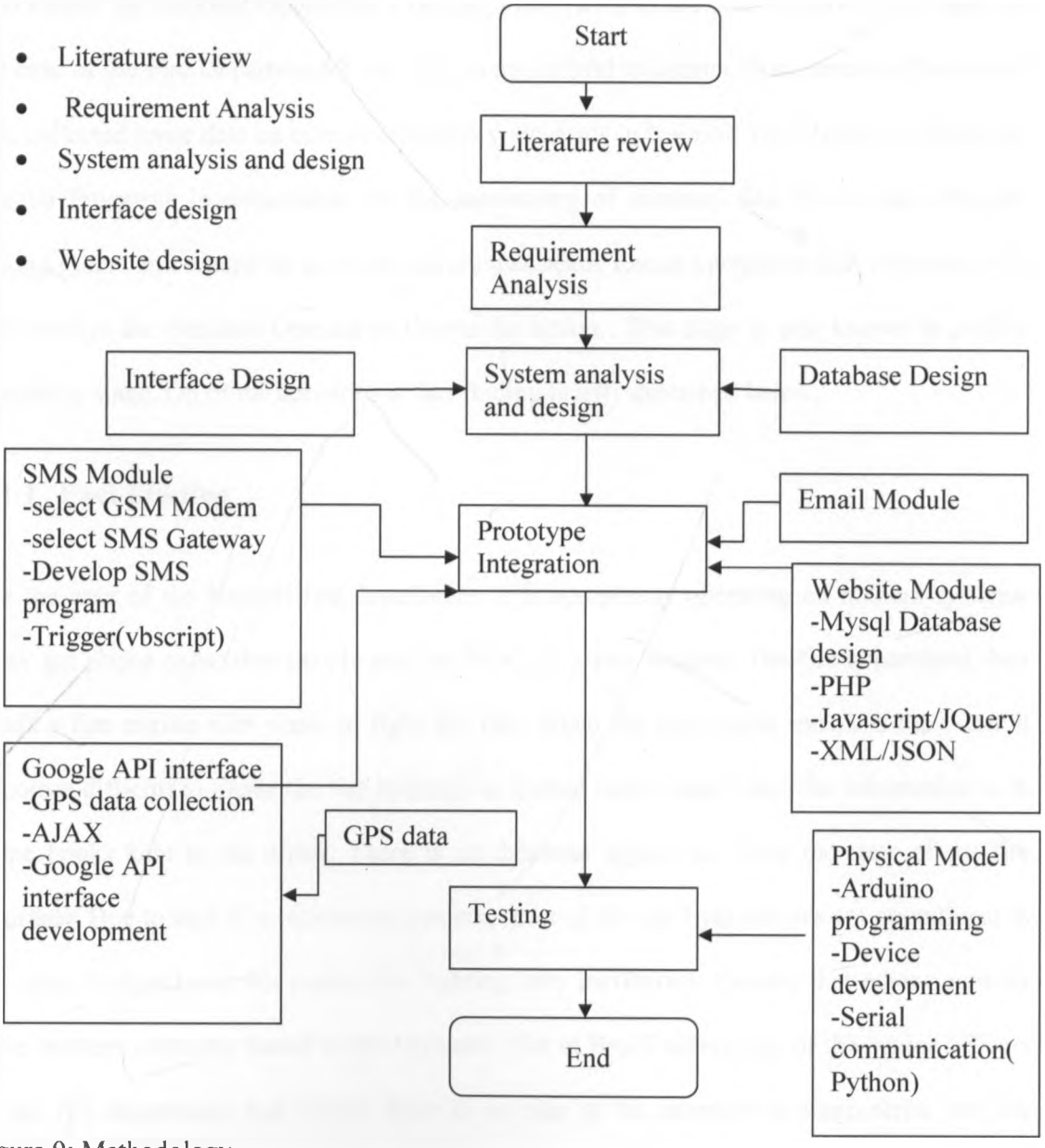
gateway is its flexibility for simulations. During initial test, the new mobile operator YU was offering free SMS facilities and this was an advantage to the research work

On the development of the physical alarm simulation which was used in the testing stage, I used the Atmega Microcontroller. I compared the Atmega microcontroller with the PIC family of microcontroller and found that writing programs for PIC requires using difficult assembler language while the Atmega has an open source language. The Atmega microcontroller is assembled into an Arduino device that can be programmed. However to simulate the working of an alarm, some basic electronic knowledge had to be acquired through internet tutorial and consultations with electricians on basic electronics like how the transistors work and the ground.

A brief schematic illustration of the methodology is illustrated in figure 9 below

The project involves the following:

- Literature review
- Requirement Analysis
- System analysis and design
- Interface design
- Website design



.Figure 9: Methodology

3.1 Requirements Analysis

During the requirement analysis I looked at three emergency systems, that of the National Operation Center, the Nairobi Fire Department and the Ministry of State for Special Programs. I also conducted a survey on the public through a questioner. During my visits to

both organizations I conducted interviews with the responsible officers. At the interviewed the Disater Manager Administration and Logic Mr Nelson N. Munyi on several occasions. I also visited the National Operations Control Office twice to see how operations are done. In the case of the Fire Department I was able to get helpful assistance from several officers and also collected some data on current emergency incidents in Nairobi. The Ministry of State for Special Programs is responsible for the monitoring of disasters like floods and droughts among others and whenever an event occurs that needs rescue operations that information is conveyed to the National Operations Center for action. This stage is also known as project feasibility stage. On of the activities is fact finding briefly described below.

3.1.1 Fact Finding

For the case of the Nairobi fire department, it is completely operating on manual systems. They get phone calls from people and the NOC on a fire incident. The fire department then sends a fire engine with water to fight the fire. When the fire engine exhausts the water it becomes difficult to locate the fire hydrants to collect more water since the information is in some books kept in the office. There is no database which can show the state of the fire hydrants. Due to lack of a monitoring system, most of the fire hydrants are not operational in the areas I visited and this makes fire fighting very ineffective. Despite that other countries have modern computer based control systems like in Brazil where one of the senior officers at the fire department had visited there is no plan at the moment to computerise the fire department. The National Operation Center (NOC) is responsible for coordinating rescue operations. However it is not fully computerized as I had expected. All its rescue operations are conducted through the telephone and Fax facilities. The NOC does not have a Geographical Information System (GIS) or even Maps on standby it relies on the Provincial Administration, the Police and the Army to do the actual rescue. Since it is poorly equipped,

its coordination capability is not well defined. The telephone directory is manual and the main telephone numbers are handwritten on a board. There is no way of communicating with the public except through the news media and there is a substantial delay in conveying such information to the affected population.

I also conducted a public survey on Nairobi Eastland where the majority of fire incidents take place. I found out that most of the people interviewed where willing to receive such information and that the most common communication equipment available was the mobile phone.

3.1.2 DATA COLLECTION

I was able to go round the city center with one officers from the Fire Department and collected fire hydrants GPS locations. This information was the put into a database for application with the Google Map API. Other data collected were emergency data records from the Nairobi Fire department and Masinde Muriro University. In the System analysis and Design stage I analyses the current system and designed a new one details will be shown in chapter 4 below.

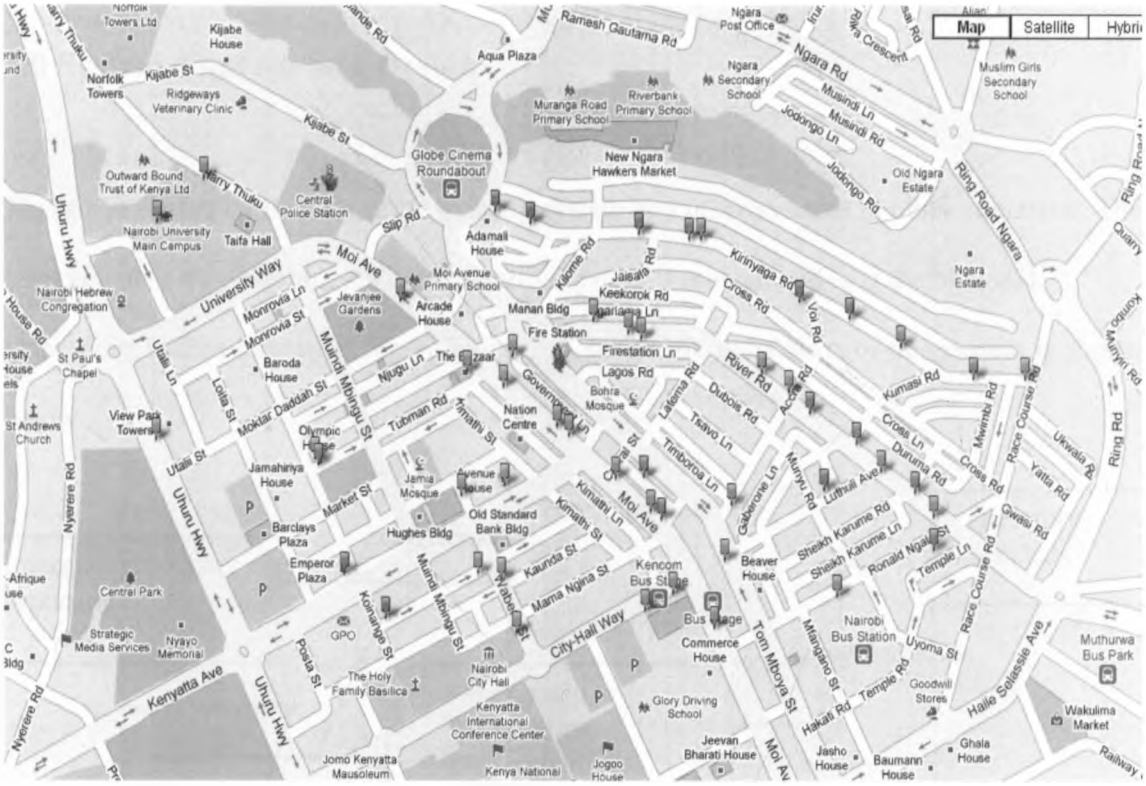


Figure 10: Fire hydrants Data collected from the City Center

4.0 System Analysis and Design

This stage started by looking at the existing system and other systems in other countries. The current emergency information dissemination system is manual and is at the National Operations Centre located at Nyayo House. A schematic illustration of this system is shown below.

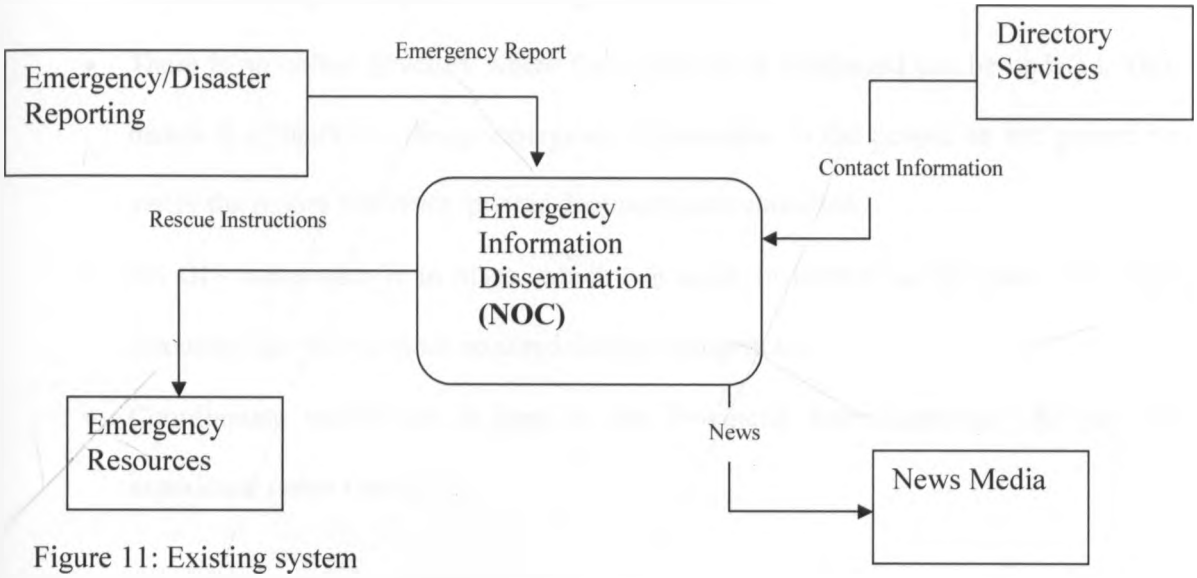


Figure 11: Existing system

The system works as follows; the system located at the National Operations Centre (NOC). It receives information about an emergency or a disaster from the public, the news media, the Ministry of State for Special Program, the Kenya Meteorological department, the Redcross and others. The NOC works with the Ministry of Special Programs to decide on the action to be done with the received information. Once the NOC decides to act on the information then it will pass the information to Rescue missions. The rescue missions are conducted by the Armed Forces in coordination with the Provincial Administration. Other organisations like the police, the Fire Fighting Departments and the Kenya Redcross are also involved in the

rescue mission. The NOC also sends the Information to the news media to convey information to the public.

4.1 Weakness of Current System

The following are the weakness in the existing systems:

- There is limited communication channels since it uses telephone and fax for real time operation. The email is not real time and there is no SMS module. The only means of communicating to the public is through news media.
- There is no online directory where the people to be conducted can be updated. This makes it difficult to convey emergency information to the people on the ground to verify the report. However, printed documents are classified.
- No GIS component is in place, which can assist in identifying hospitals and other resources like access roads required during emergencies.
- Coordination mechanism is done by the Provincial Administrations who rely on experience rather than ICT.

4.2 Proposed System Overview

In order to address the deficiencies of the existing system, an overview framework of the proposed emergency information dissemination system was developed. This was possible after comparison with other system. The framework that was able to address the requirements was the framework developed by Renato Iannella, Karen Robinson, Olli-Pekka Rinta-Koski all from NICTA. NICTA is the National Australia's Information and Communications Technology (ICT) Centre of Excellence.

Figure 12: CIMS Framework (Crisis Information Management System)

Source: NICTA

According to Renato Iannella and colleagues from NICTA, the **Incident Management** module supports the high-level recording of individual incidents while the **People Management** module supports the management of defined roles, teams, tasks and duties of individuals and organizations. The **Resource Management** module supports the management of resources during a crisis which involves all stages (discovery, commitment, deployment, return, extension, etc) for resources involved in the recovery and response phases of a crisis. The **Notification Management** module supports the management of the outgoing and the incoming information Messages and includes broadcast messages to large groups, even community wide, and routing of messages to the right people who need to be informed of the content. The **Situational Awareness Management** module supports the development of a "picture-of-operation" that encapsulates the current crisis, based on all the information currently available and would be aggregated situational reports or geo-spatial images with multiple layers showing current status of the incident and allowing planning operations. The **Document Management** module supports the effective categorization of the documents created and deposited into the CIMS and the **Report Management** module supports the creation of incident reports, based on the CIMS repository of information, such as status reports etc. The **Financial Management** module supports budgets, expenditures, and

reconciliation of financial transactions. The **Assessment Modeling** Management supports planning and the modelling of the incident, such as damage assessment, or storm-tide surge modeling. The **Authentication and Authorisation** Services support users to gain access and be authorised to perform secure functions in the CIMS. The **Directory Services** module supports a single view of users across the CIMS including federated identity services. The **Geospatial Services** support mapping of incident data to various map sources, such as road networks or satellite maps.

The above framework was demonstrated to the National Operations Center (NOC). I used interviews and observation to get an insight of the NOC operations. The NOC Disaster Manager Administration and Logistic was interviewed twice, at the initial stages and at the development of the model on all aspects of this framework. The CIMS model developed by NICTA was found to be suitable. According to the highly trained Disaster Manager Adminstration and Logistic this model was actually implementing the Incident Command System(ICS) this was after we compared it with the Incident Command System (ICS) . The only issue that was found to be fundamentally different was the Financial Module. There is already a government Financial Management system and according to the NOC experts this was able to address all financial matters. At the same time, different organisations in Kenya play different roles and are financially independent. The NOC has an elaborate Directory Services where different documents are classified and filed. The framework used to develop the system is shown below. The main difference between the CMIS and NOC ICS frame work is that the CMIS is an integrated framework while the NOC's ICS is a distributed framework. For the case of Australia the system is fully automated and highly developed and can be used by different users with different access level. The NOC model is highly manual with different components implemented by other organisations. For example the

authorization module is done by the Permanent Secretary Ministry of Special Programs. The People Management and Situation Management is done by the Provincial Administration. There is no Geosptial module but if the need arises then the Department of Surveys is called upon to produce the maps. As is visible the lack of a centralised computerised system implies that it is not possible to effectively deal with emergencies.

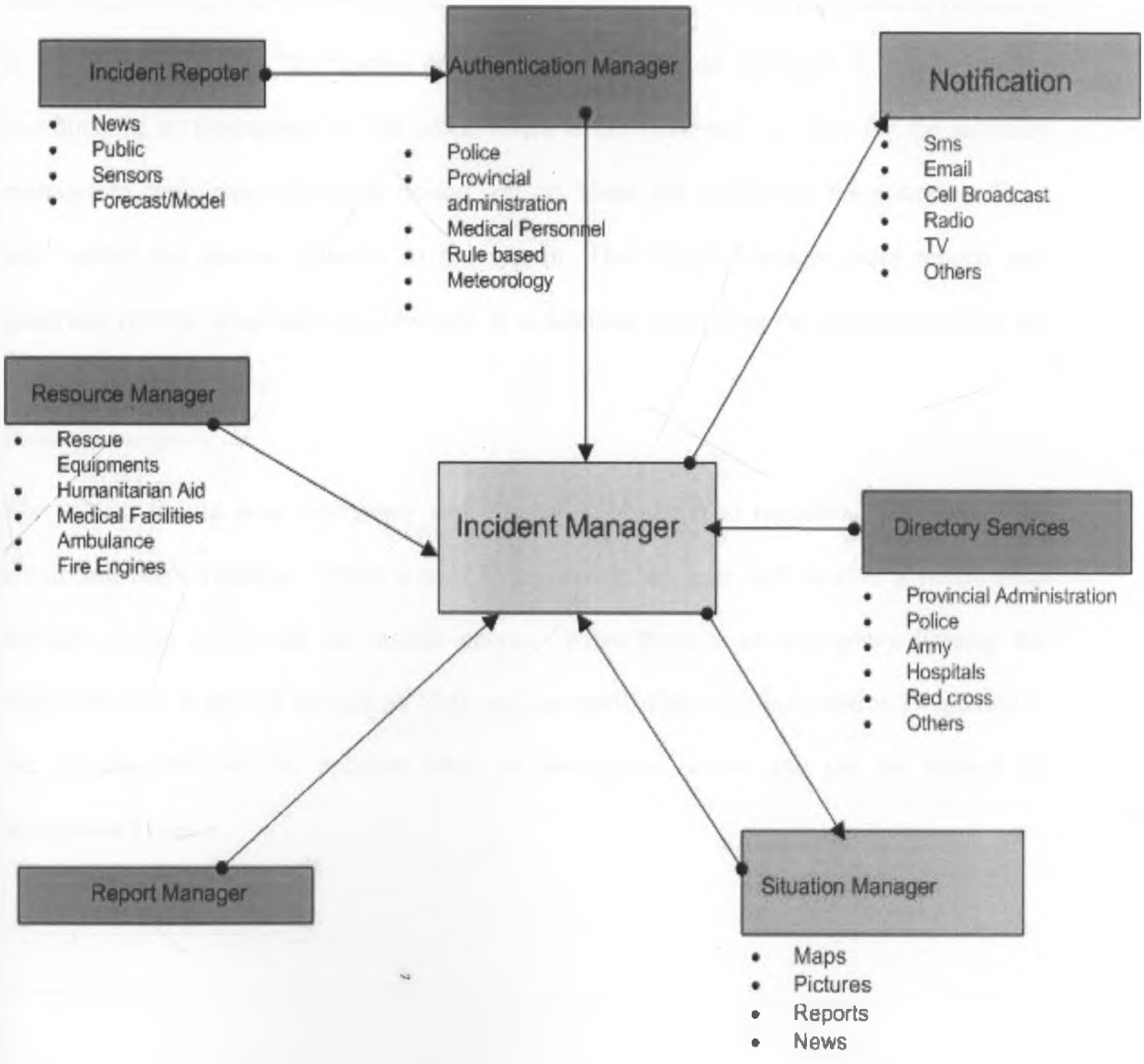


Figure 13: Incident Management Framework

4.2.1 A Brief Proposed System Description

The system should work as follows: the incident is reported to the incident Module by Sensors, the Publics news Media, Forecast like Kenya Meteorological department. Once the Incident Module receives the information it will route it to the Authentication Manager. The Authentication manager will validate the message through rule based system or by contacting other people in the area where the incident has been reported. Once the incident is verified it is the relayed by the Notification module. The situation manager is responsible to coordinating an Emergency on the place where it has occurred. In order for the situation manager to coordinate effectively he/she will get Maps and reports for the system and can also update the current situation to the system. The Report Manager store reports and generates reports. The Resource Manager is a database containing the resources which are need during the incident.

The system should send emergency and disaster information to registered users using the email and SMS systems. When a user is registered, the user will receive a notification through his/her email and the mobile phone. When there is an emergency/disaster, the registered user is alerted through an SMS and an email. The same information is updated to the website that will be updated when an emergency occurs and can be viewed by unregistered user.

4.3 System Design

Figure 13 below shows the context diagram of the Emergency Information Dissemination System. This system is a subset of the framework and will address information dissemination in both directions from the incident reporter and how such information can then be relayed to the relevant persons.

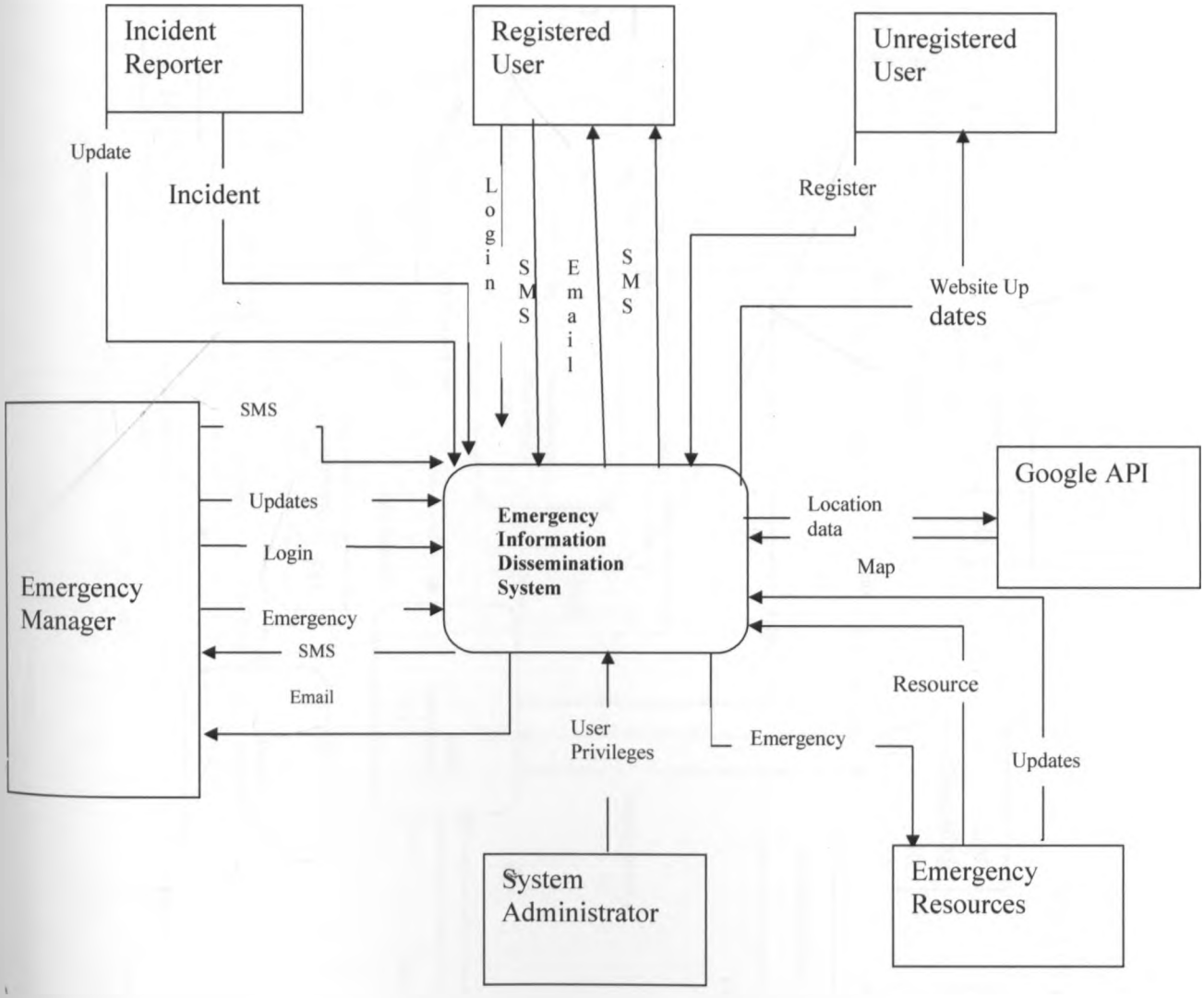


Figure 14 Context Diagram

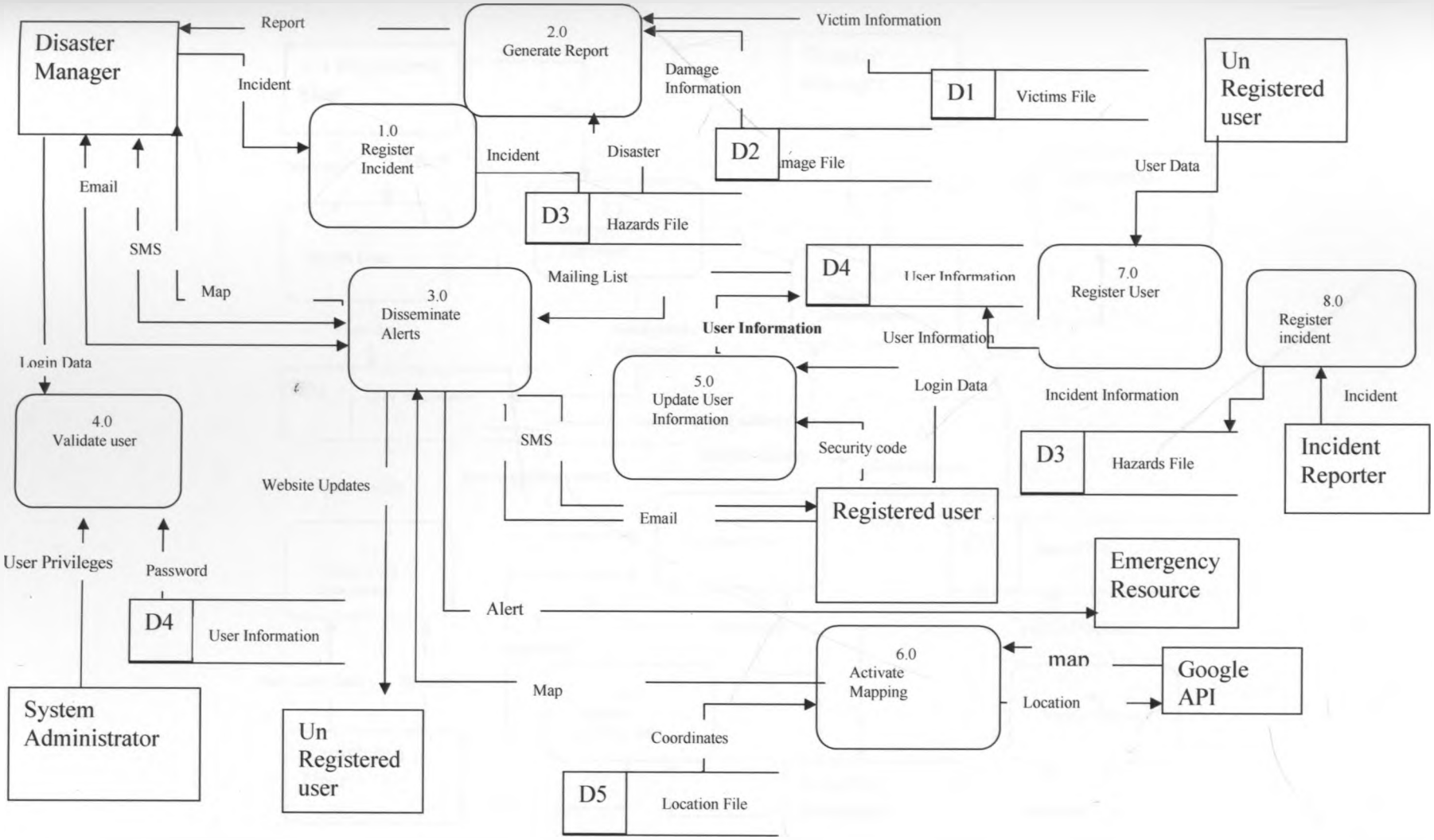


Figure 15 DFD LEVEL 0

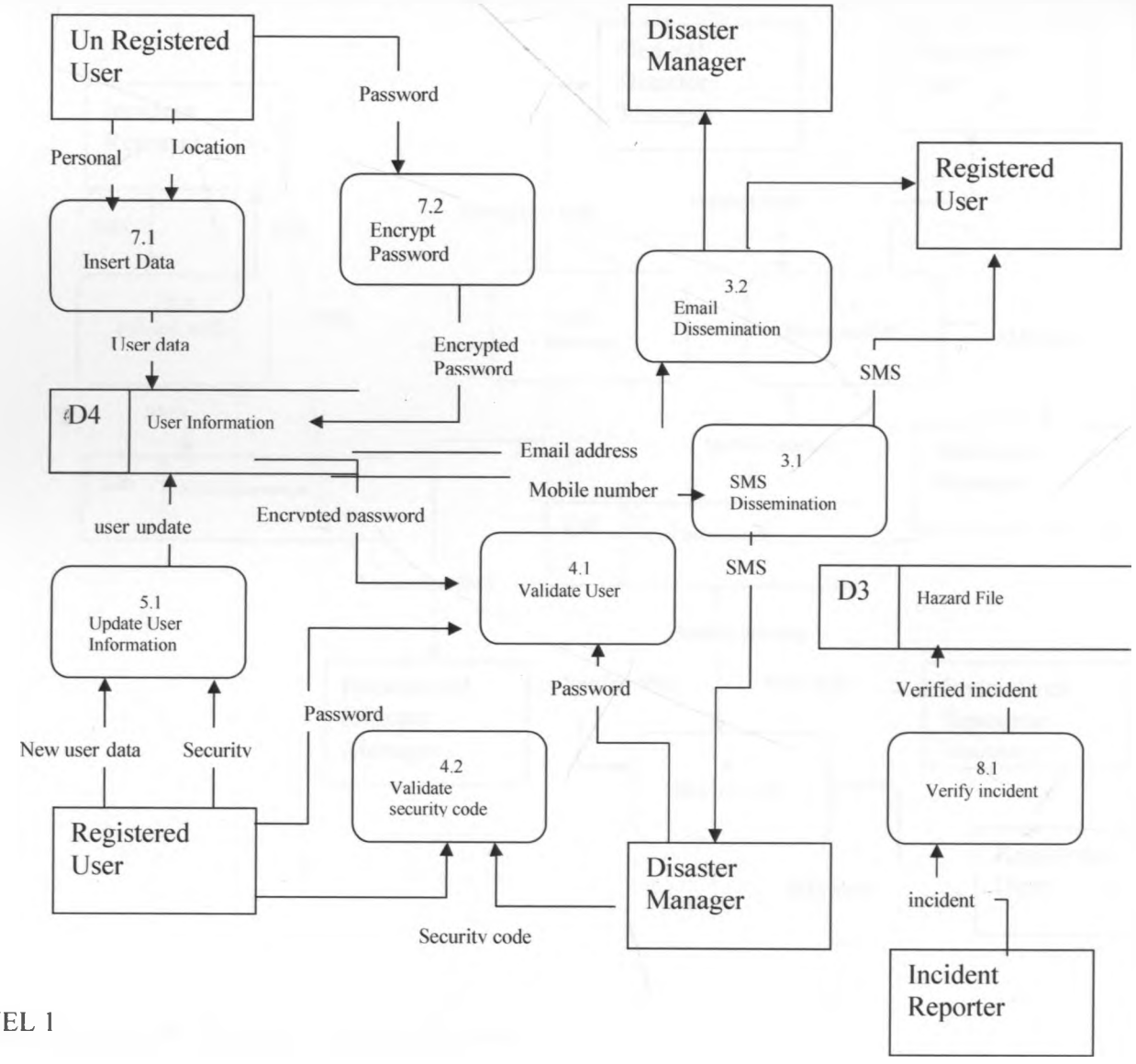


Figure 16 DFD LEVEL 1

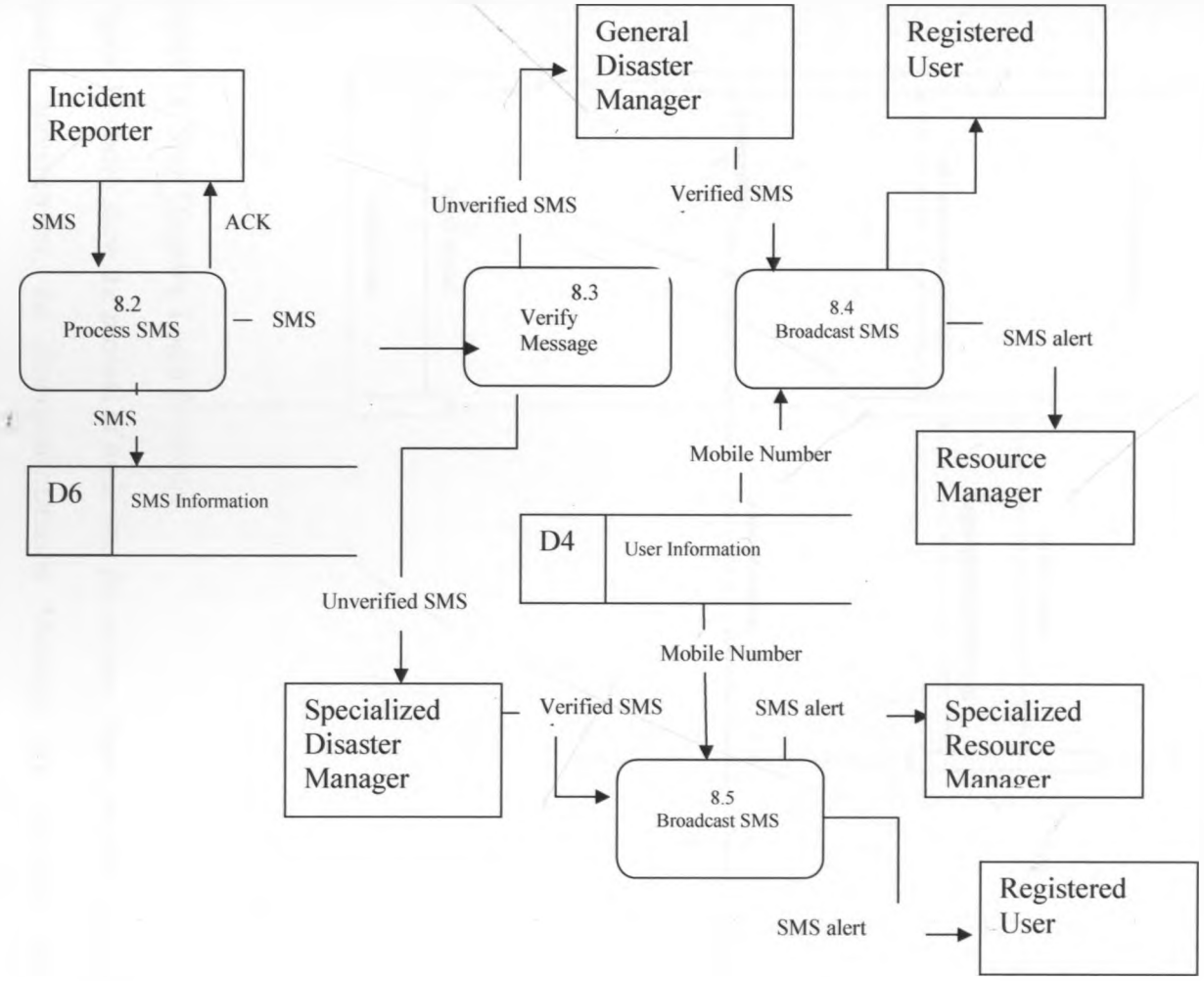


Figure 17: DFD Level 1 Showing the Working of the SMS module

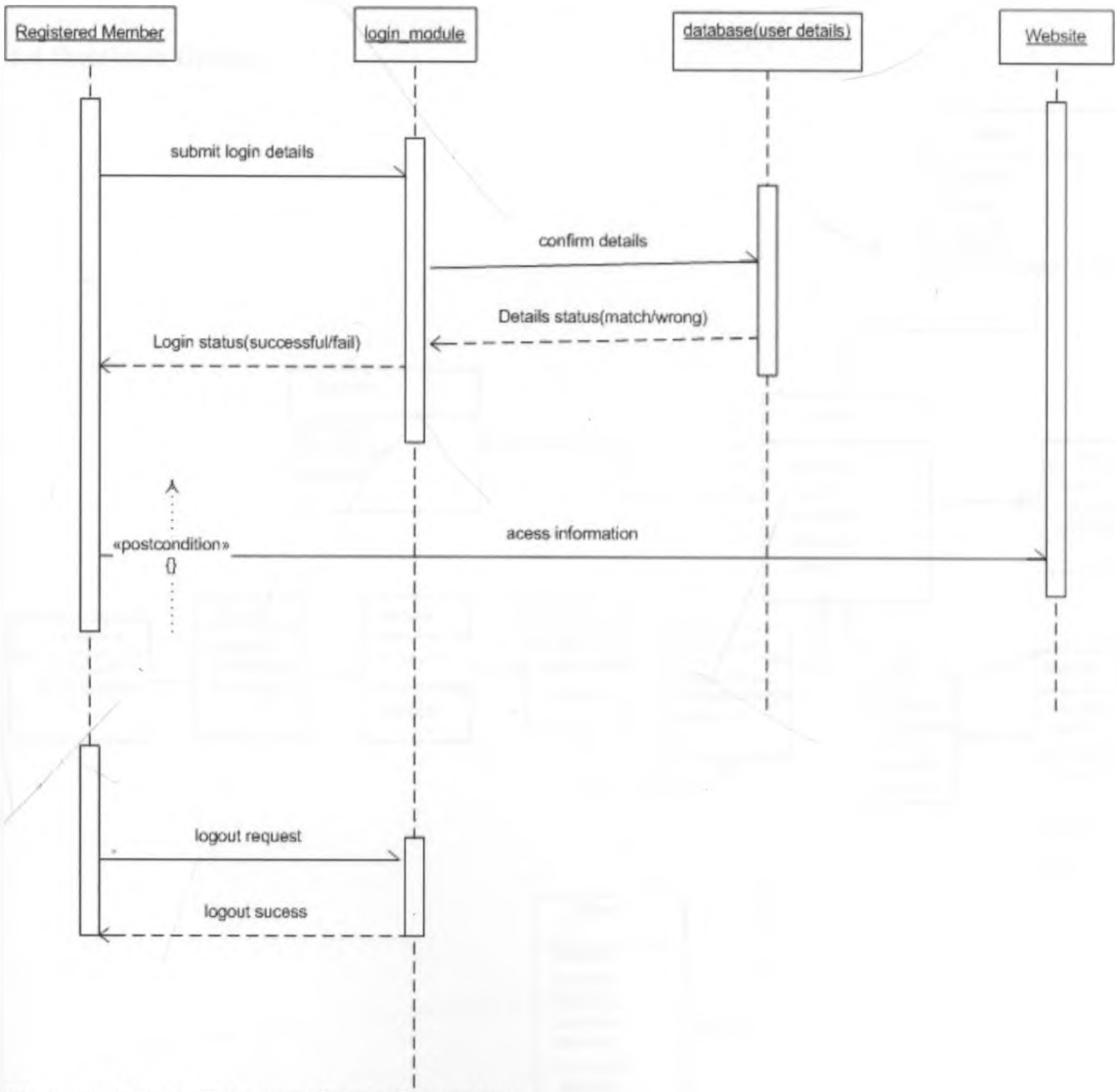


Figure 18: State Diagram: Login Procedure

Figure 18 above show the process of login into the system. There are four types of users: the system administrator, the Emergency/Disaster Manager, the registered user and the unregistered user. The unregistered user can only acess the website and view the website information while the registered user can login and change his/her profile and is able to receive and SMS and email. The registered user has limited access rights. The emergency manager on the other hand can access many functions of the system but can not change his

profile. The system administrator is the one who can change access rights of the users. The system uses password encryption and other security features like session management.

4.4 Database Design

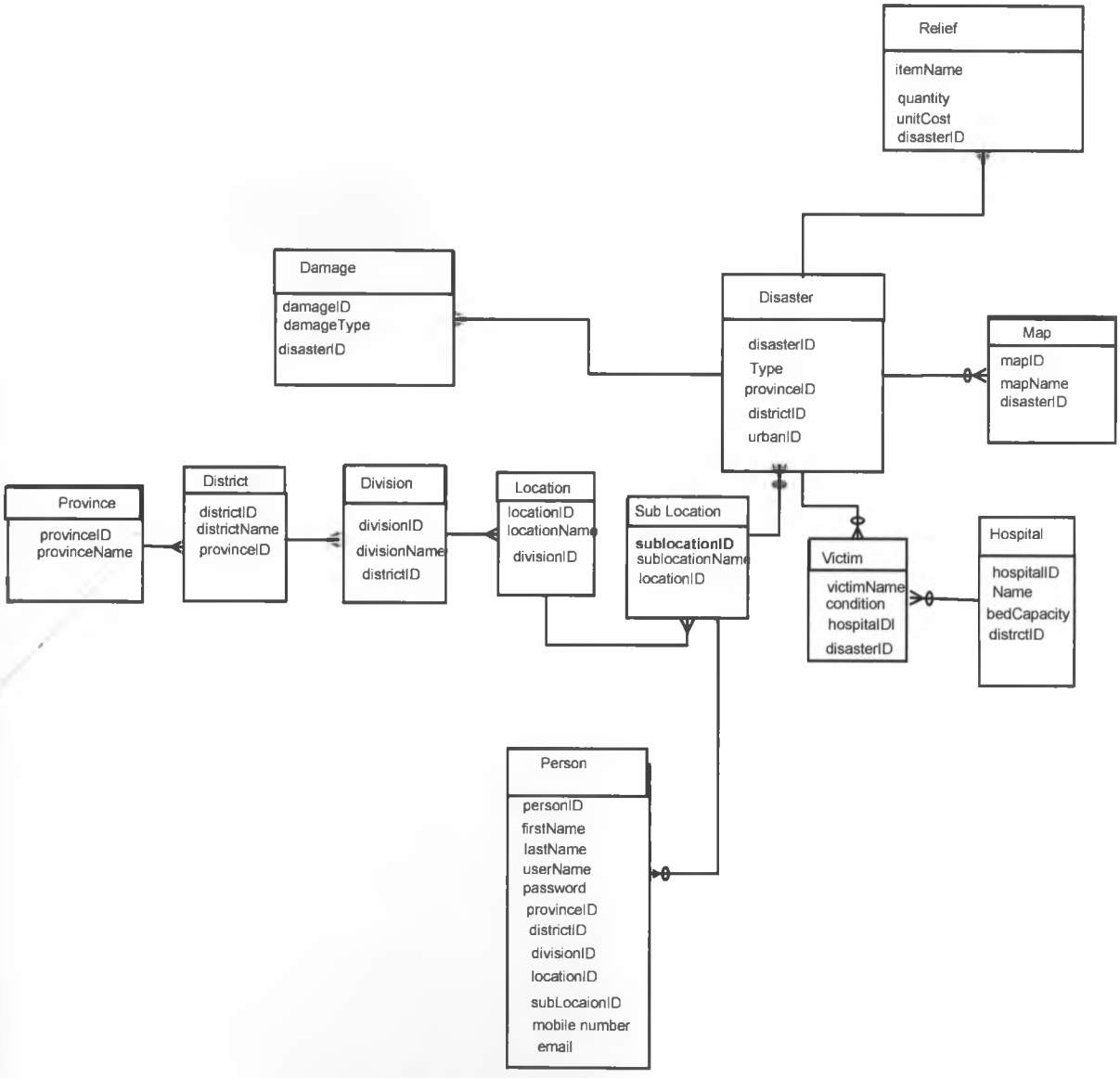


Figure 19: Database Design

Figure 19 above show the database design. There is also another database for the SMS module which has limited use in the project but is used to view the received and sent SMS. Other tables used for updates are not shown in the design.

4.4 User Interface Design

Figure20: Login Form

Please fill in your details to register to enable you to receive disaster alerts and updates through your mobile phone and your ema

Figure21: Registration Form

Update changes to your profile if there are new changes

Figure22: Personal Profile Update

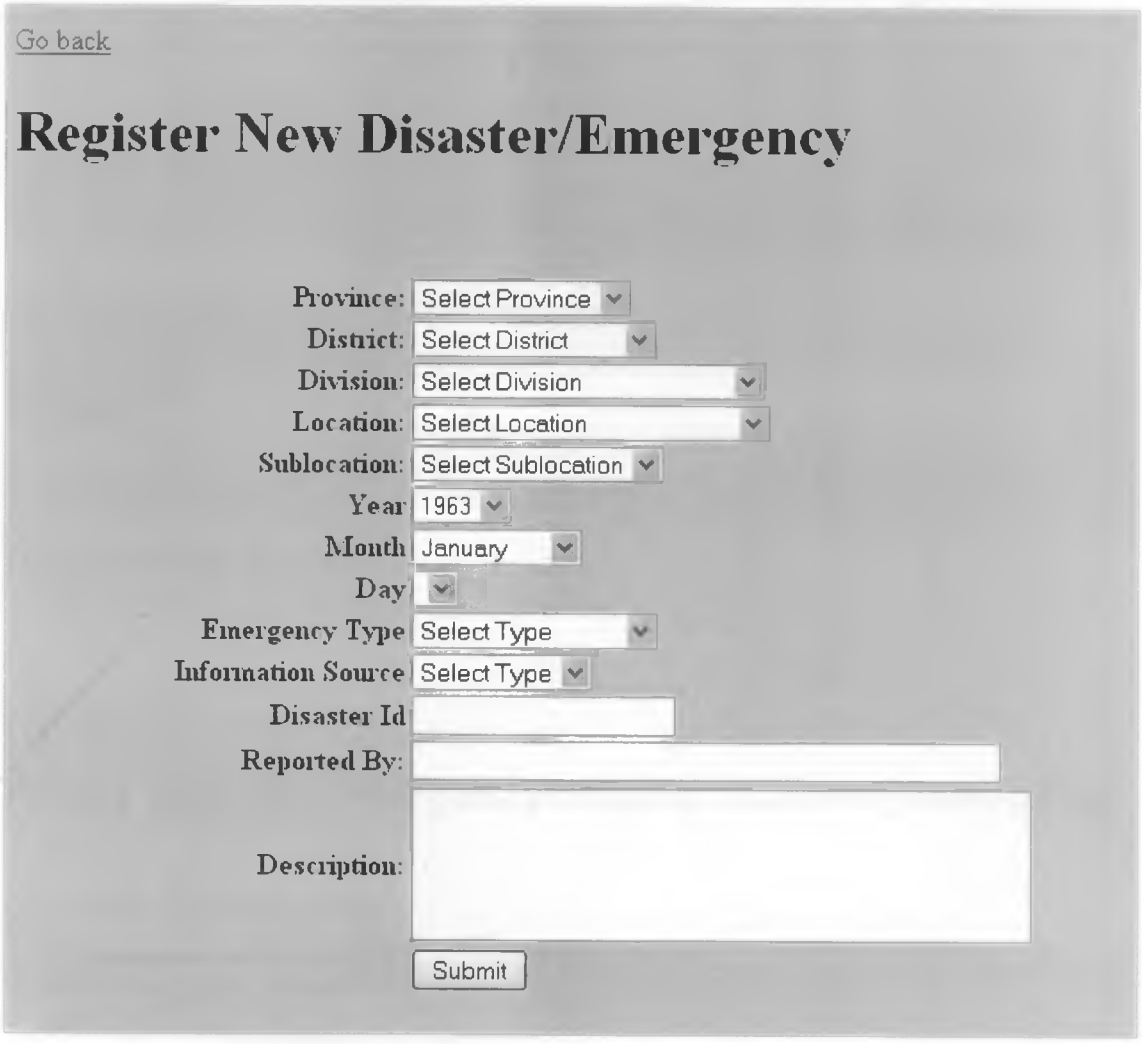


Figure23: User Registration

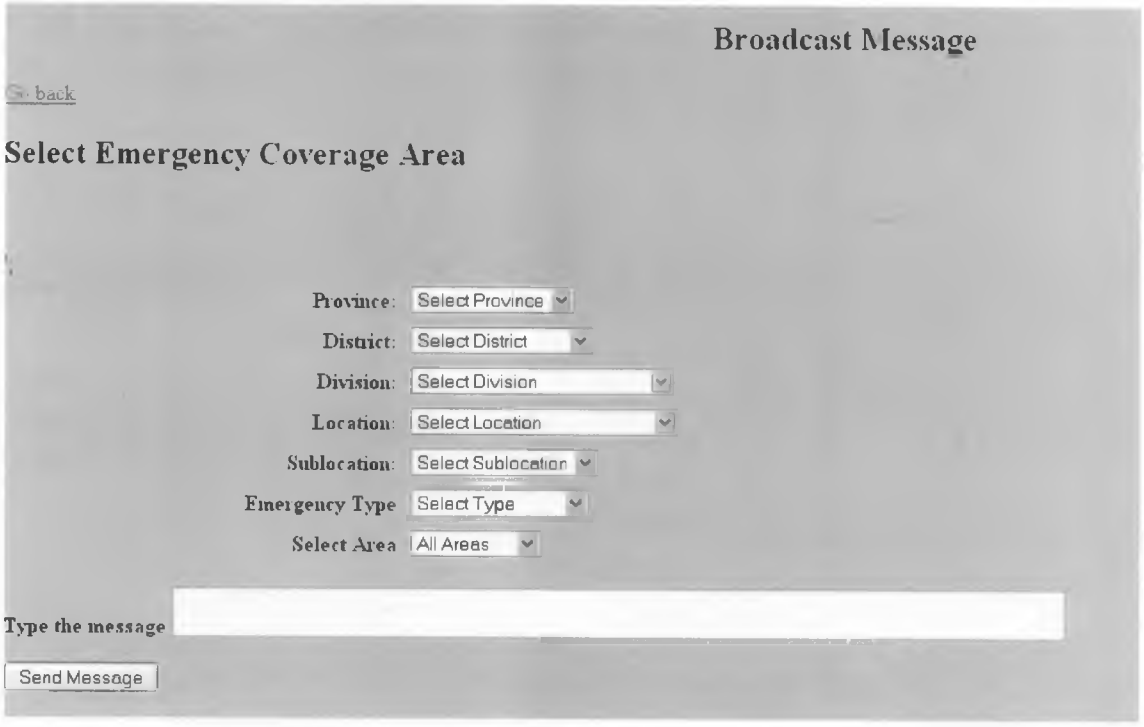


Figure24: Message Broadcast

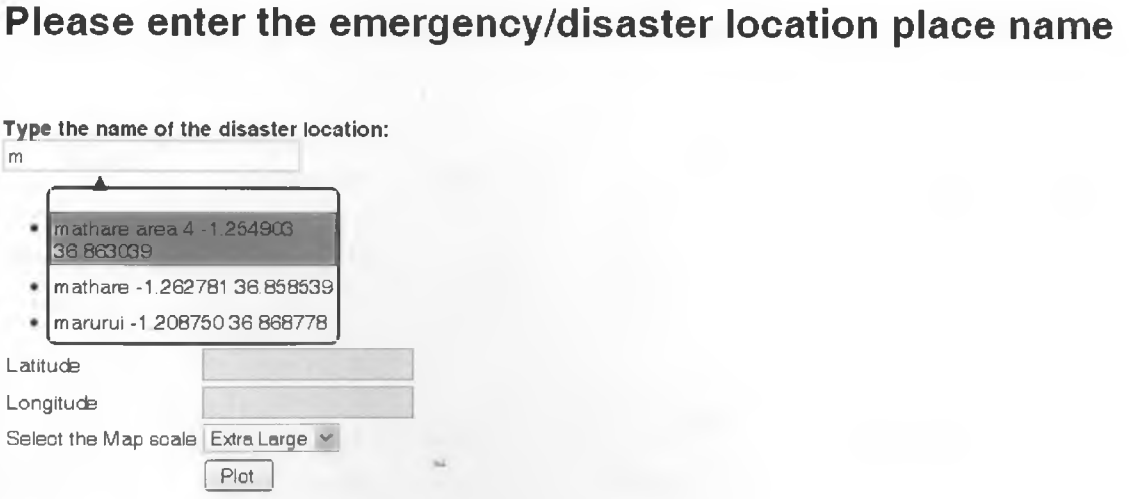


Figure25: Select a know location

CHAPTER 5: IMPLEMENTATION AND TESTING

Data Analysis from field survey

Frequency Table

Gender	Frequency	Percent
Male	37	50
Female	37	50
Total	74	100

Education	Frequency	Percent
Primary School	9	12.2
Secondary School	24	32.4
A-levels	5	6.8
Diploma	23	31.1
University	13	17.6
Total	74	100

work	Frequency	Percent
Public Sector	14	18.9
Private Sector	16	21.6
Self	9	12.2
Un Employed	35	47.3
Total	74	100.0

age	Frequency	Percent
Under 20	10	13.5
20-30	16	21.6
30 - 40	27	36.5
40-50	15	20.3
50-60	4	5.4
Over 60	2	2.7
Total	74	100.0

mobile	Frequency	Percent
Without	2	2.7
With	72	97.3
Total	74	100.0

radio	Frequency	Percent
Without	11	14.9
With	63	85.1
Total	74	100.0

TV	Frequency	Percent
Without	23	31.1
With	51	68.9
Total	74	100

Email	Frequency	Percent
Without	39	52.7
With	35	47.3
Total	74	100

Internet	Frequency	Percent
Without	42	56.8
With	32	43.2
Total	74	100

Fixed Telephone	Frequency	Percent
Without	69	93.2
With	5	6.8
Total	74	100

Language	Frequency	Percent
-	1	1.4
English	19	25.7
Swahili	21	28.4
Both	33	44.6
Total	74	100

Vernacular	Frequency	Percent
Non	35	47.3
Luo	10	13.5
Luyah	4	5.4
Kikuyu	8	10.8
Kamba	14	18.9
Kalenjin	3	4.1
Total	74	100

Switch off mobile	Frequency	Percent
False	40	54.1
True	34	45.9
Total	74	100

Mobile Off at Night	Frequency	Percent
False	51	68.9
True	23	31.1
Total	74	100.0

Mobile Off at day	Frequency	Percent
False	63	85.1
True	11	14.9
Total	74	100

Gender * language Crosstabulation

			language				
			Non	English	Swahili	Both	Total
Gender	Male	Count	0	9	12	16	37
		% within Gender	.0%	24.3%	32.4%	43.2%	100.0%
	Female	Count	1	10	9	17	37
		% within Gender	2.7%	27.0%	24.3%	45.9%	100.0%
Total		Count	1	19	21	33	74
		% within Gender	1.4%	25.7%	28.4%	44.6%	100.0%

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	1.512[a]	3	.680
Likelihood Ratio	1.899	3	.594
Linear-by-Linear Association	.073	1	.787
N of Valid Cases	74		

a. 2 cells (25.0%) have expected count less than 5. The minimum expected count is .50.

Gender * language * Education Crosstabulation

Education			language				
			0	English	Swahili	Both	Total
Primary	Male	Count		2	3	0	5
		% within Gender		40.0%	60.0%	.0%	100.0%
	Female	Count		0	2	2	4
		% within Gender		.0%	50.0%	50.0%	100.0%
	Total	Count		2	5	2	9
		% within Gender		22.2%	55.6%	22.2%	100.0%
Secondary	Male	Count	0	2	4	3	9
		% within Gender	.0%	22.2%	44.4%	33.3%	100.0%
	Female	Count	1	5	6	3	15
		% within Gender	6.7%	33.3%	40.0%	20.0%	100.0%
	Total	Count	1	7	10	6	24
		% within Gender	4.2%	29.2%	41.7%	25.0%	100.0%
A-Levels	Male	Count				2	2
		% within Gender				100.0%	100.0%
	Female	Count				3	3
		% within Gender				100.0%	100.0%
	Total	Count				5	5
		% within Gender				100.0%	100.0%
Diploma	Male	Count		4	4	4	12
		% within Gender		33.3%	33.3%	33.3%	100.0%
	Female	Count		3	1	7	11
		% within Gender		27.3%	9.1%	63.6%	100.0%
	Total	Count		7	5	11	23
		% within Gender		30.4%	21.7%	47.8%	100.0%
University	Male	Count		1	1	7	9
		% within Gender		11.1%	11.1%	77.8%	100.0%
	Female	Count		2	0	2	4
		% within Gender		50.0%	.0%	50.0%	100.0%
	Total	Count		3	1	9	13
		% within Gender		23.1%	7.7%	69.2%	100.0%

Chi-Square Tests

Education		Value	df	Asymp. Sig. (2-sided)
1	Pearson Chi-Square	4.140[a]	2	.126
	Likelihood Ratio	5.635	2	.060
	Linear-by-Linear Association	3.600	1	.058
	N of Valid Cases	9		
2	Pearson Chi-Square	1.265[b]	3	.738
	Likelihood Ratio	1.601	3	.659
	Linear-by-Linear Association	1.111	1	.292
	N of Valid Cases	24		
3	Pearson Chi-Square	[c]		
	N of Valid Cases	5		
4	Pearson Chi-Square	2.723[d]	2	.256
	Likelihood Ratio	2.856	2	.240
	Linear-by-Linear Association	.965	1	.326
	N of Valid Cases	23		
5	Pearson Chi-Square	2.568[e]	2	.277
	Likelihood Ratio	2.694	2	.260
	Linear-by-Linear Association	1.600	1	.206
	N of Valid Cases	13		

Gender * vernacular Crosstabulation

			vanacular						
			Non	Luo	Luyah	Kikuyu	Kamba	Kalenjin	Total
Gender	Male	Count	17	6	2	3	8	1	37
		%	45.9%	16.2%	5.4%	8.1%	21.6%	2.7%	100%
	Femal e	Count	18	4	2	5	6	2	37
		%	48.6%	10.8%	5.4%	13.5%	16.2%	5.4%	100%
Total		Count	35	10	4	8	14	3	74
		%	47.3%	13.5%	5.4%	10.8%	18.9%	4.1%	100%

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	1.548[a]	5	.908
Likelihood Ratio	1.563	5	.906
Linear-by-Linear Association	.004	1	.947
N of Valid Cases	74		

Gender * English Crosstabulation

			English		
			0	1	Total
Gender	Male	Count	12	25	37
		% within Gender	32.4%	67.6%	100.0%
	Female	Count	10	27	37
		% within Gender	27.0%	73.0%	100.0%
Total		Count	22	52	74
		% within Gender	29.7%	70.3%	100.0%

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	.259[a]	1	.611		
Continuity Correction[b]	.065	1	.799		
Likelihood Ratio	.259	1	.611		
Fisher's Exact Test				.800	.400
Linear-by-Linear Association	.255	1	.613		
N of Valid Cases	74				

Gender * language Crosstabulation

			language				
			0	English	Swahili	Both	Total
Gender	Male	Count	0	9	12	16	37
		% within Gender	.0%	24.3%	32.4%	43.2%	100.0%
	Female	Count	1	10	9	17	37
		% within Gender	2.7%	27.0%	24.3%	45.9%	100.0%
Total		Count	1	19	21	33	74
		% within Gender	1.4%	25.7%	28.4%	44.6%	100.0%

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	1.512[a]	3	.680
Likelihood Ratio	1.899	3	.594
Linear-by-Linear Association	.073	1	.787
N of Valid Cases	74		

Education * language Crosstabulation

			language				
			0	English	Swahili	Both	Total
Education	Primary	Count	0	2	5	2	9
		%	.0%	22.2%	55.6%	22.2%	100.0%
	Secondary	Count	1	7	10	6	24
		%	4.2%	29.2%	41.7%	25.0%	100.0%
	A-Level	Count	0	0	0	5	5
		%	.0%	.0%	.0%	100.0%	100.0%
	Diploma	Count	0	7	5	11	23
		%	.0%	30.4%	21.7%	47.8%	100.0%
	University	Count	0	3	1	9	13
		%	.0%	23.1%	7.7%	69.2%	100.0%
Total		Count	1	19	21	33	74
		%	1.4%	25.7%	28.4%	44.6%	100.0%

Chi-Square Tests

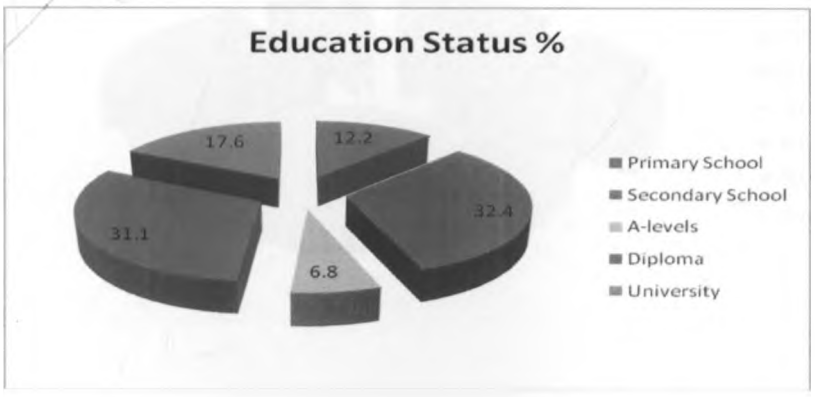
	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	19.674[a]	12	.074
Likelihood Ratio	22.123	12	.036
Linear-by-Linear Association	3.208	1	.073
N of Valid Cases	74		

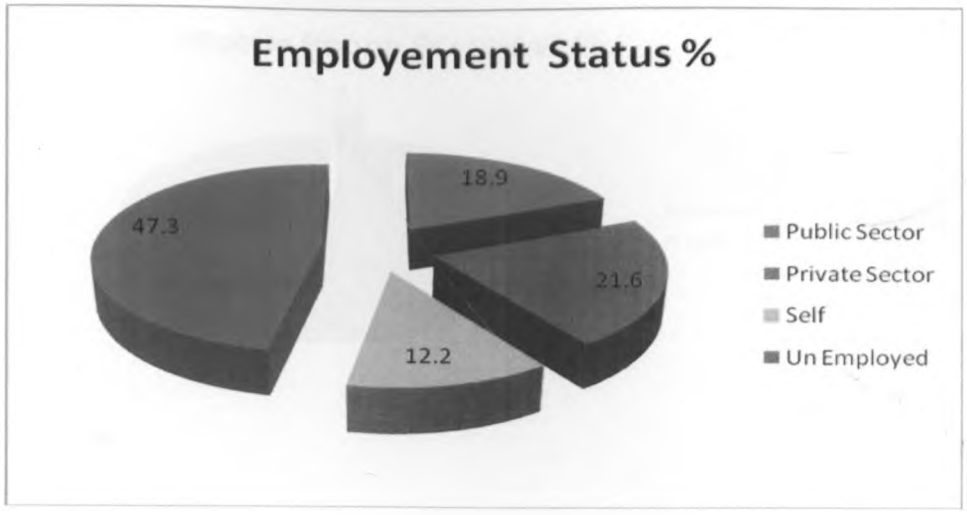
Gender *Switch Off Mobile Phones at Night Crosstabulation

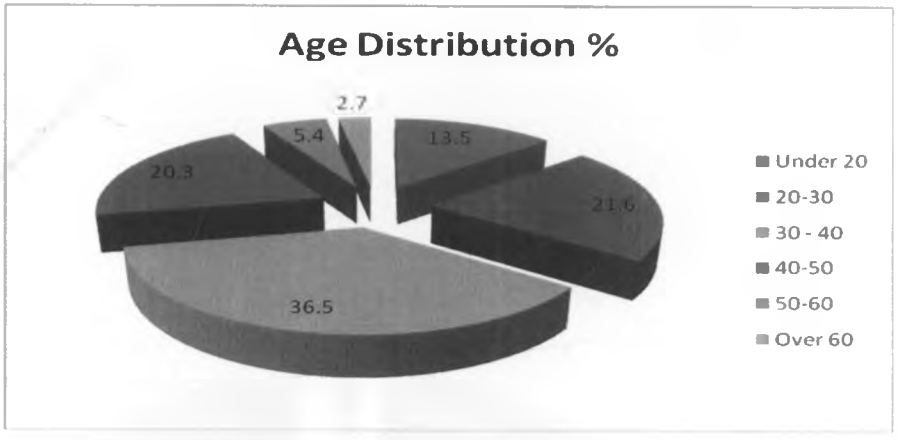
			Night		
			0	1	Total
Gender	Male		28	9	37
		% within Gender	75.7%	24.3%	100.0%
	Female		23	14	37
		% within Gender	62.2%	37.8%	100.0%
Total		Count	51	23	74
		% within Gender	68.9%	31.1%	100.0%

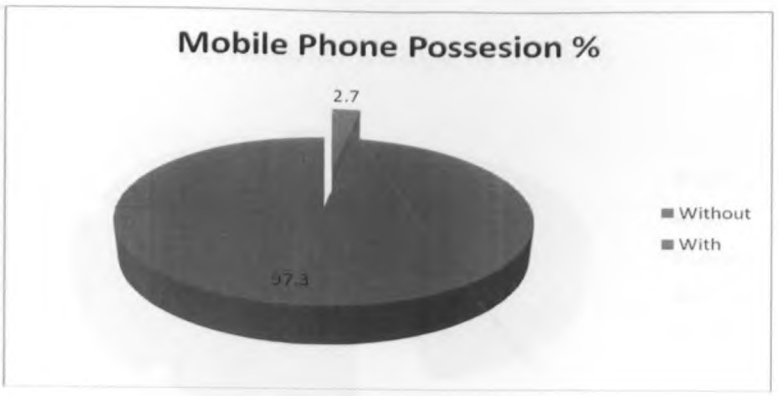
Chi-Square Tests

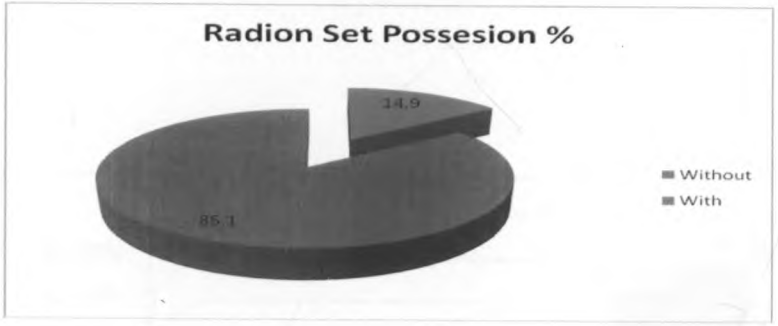
	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	1.577[a]	1	.209		
Continuity Correction[b]	1.009	1	.315		
Likelihood Ratio	1.587	1	.208		
Fisher's Exact Test				.315	.158
Linear-by-Linear Association	1.556	1	.212		
N of Valid Cases	74				

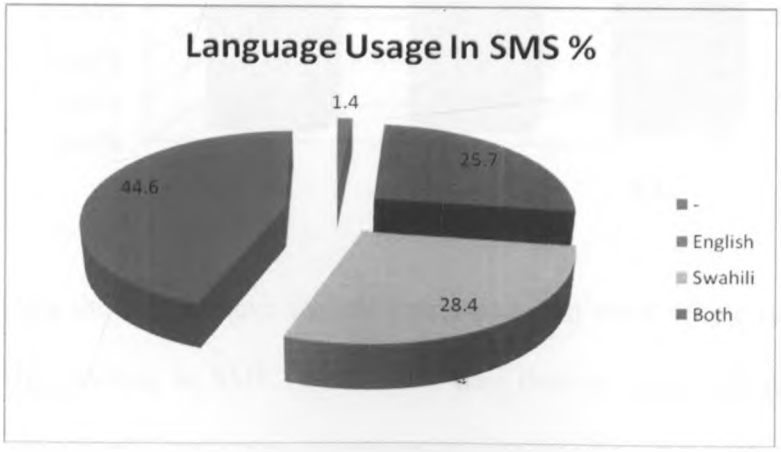


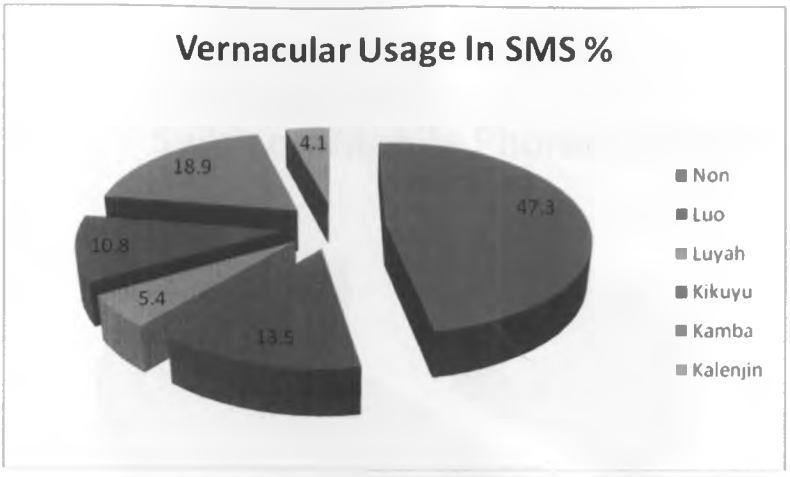

Employement Status %
18.9
47.3
21.6
12.2
Public Sector
Private Sector
Self
Un Employed

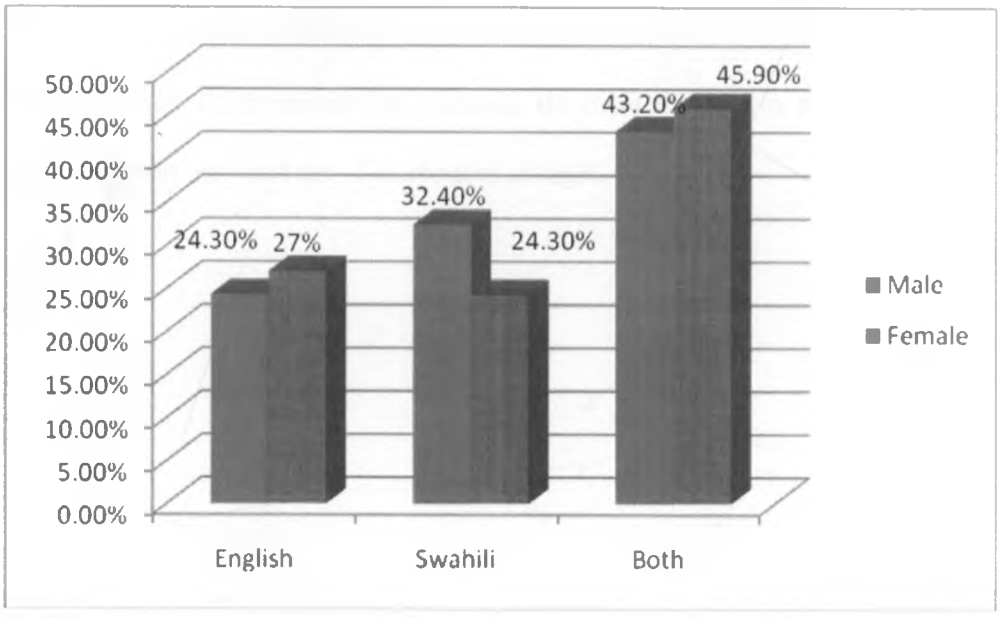

Age Distribution %
2.7
5.4
13.5
20.3
21.6
36.5
Under 20
20-30
30 - 40
40-50
50-60
Over 60


Mobile Phone Possesion %
2.7
97.3
Without
With

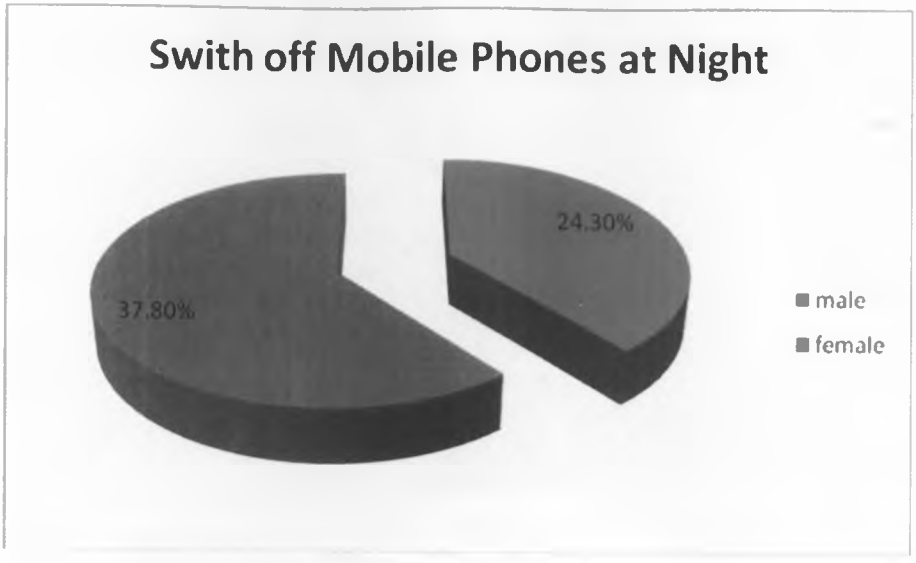

Radion Set Possesion %
14.9
85.1
Without
With


Language Usage In SMS %
1.4
25.7
44.6
28.4
-
English
Swahili
Both

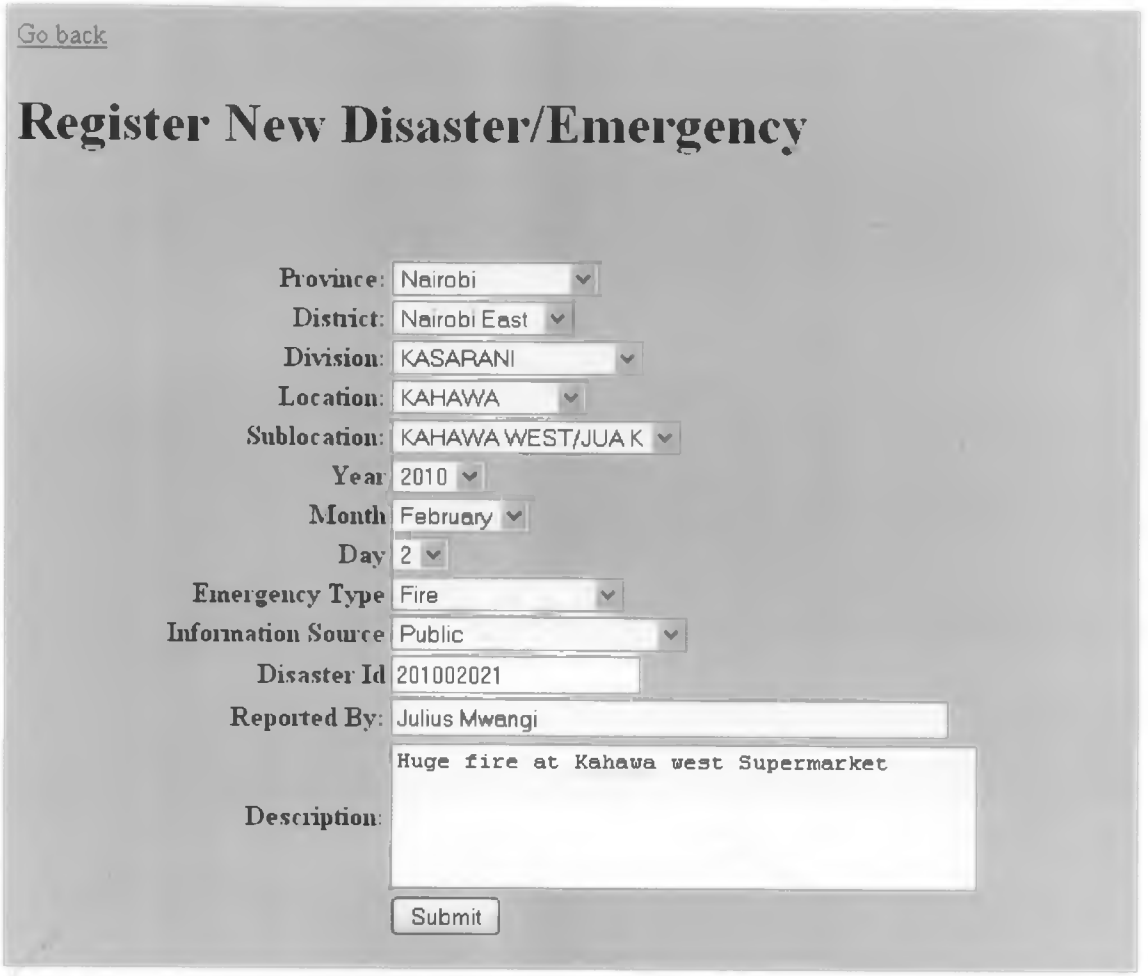


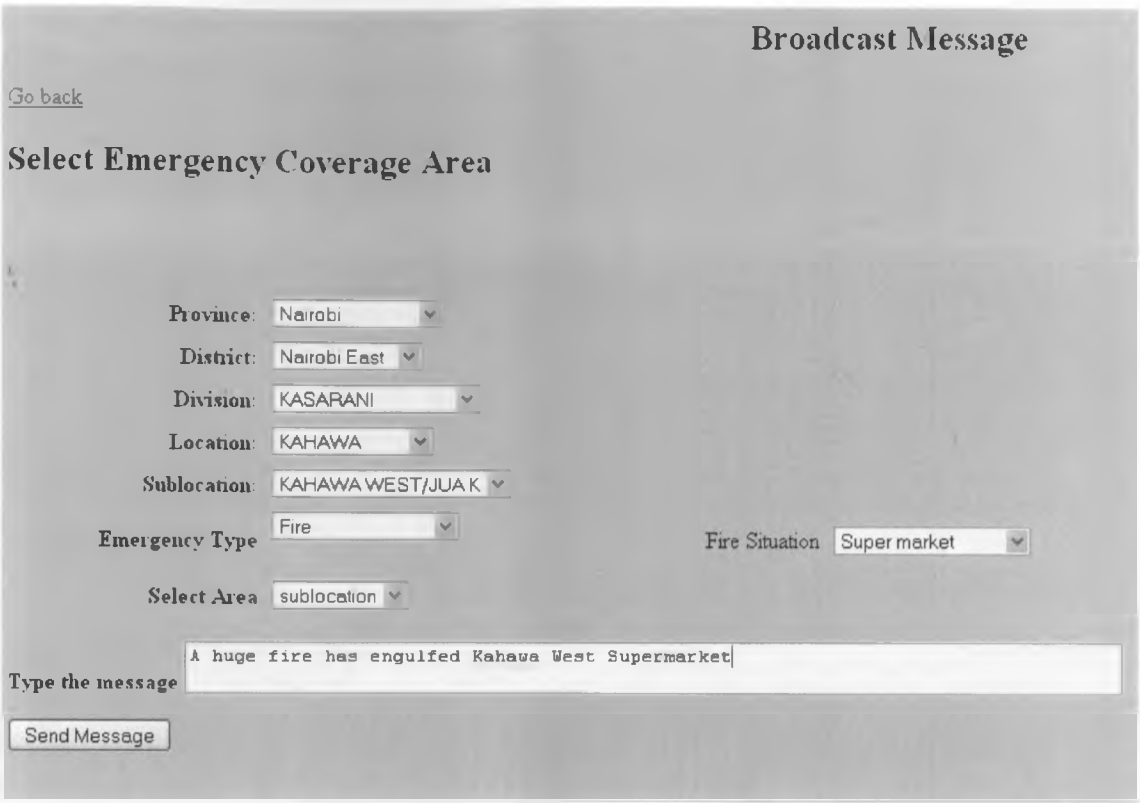


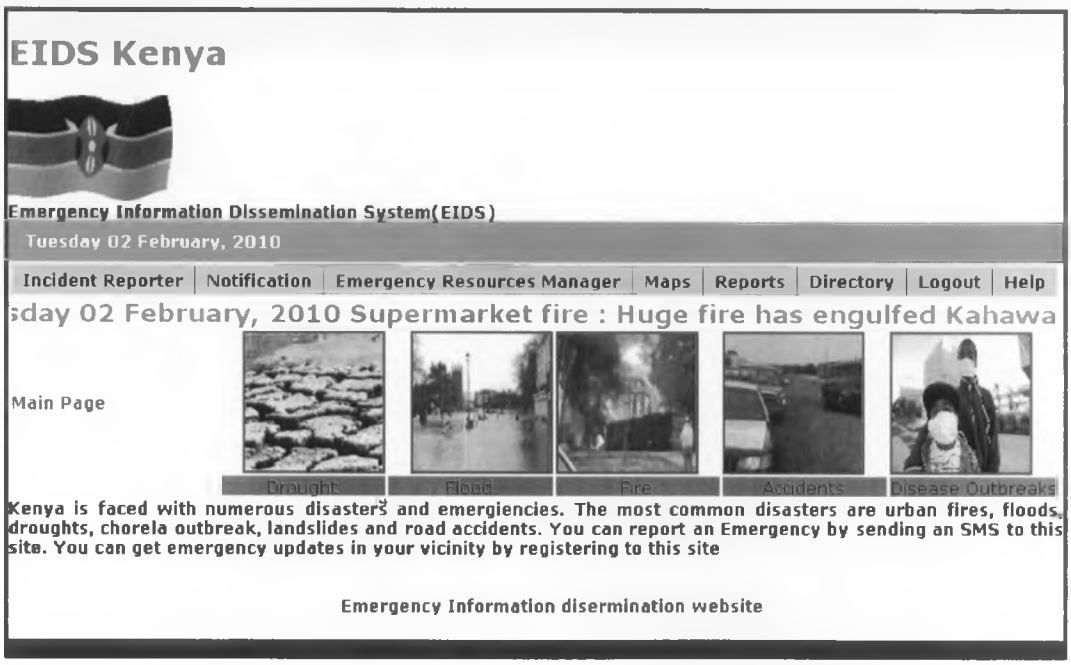
From the graph above the analysis shows that most people use both English and Swahili when writing an SMS. At the same time there is some indication that there are more men using Swahili and more women using English. This information was used in designing the SMS module. It is for this reason that the SMS program can understand English and Swahili key words FIRE, DISEASE, MOTO, UGONJWA. The SMS system can also send a mixed English and Swahili message.



Preliminary analysis show that more females are likely to switch off their mobile phones at night that men. Of the people interviewed, the main reason why people switch of the mobiles is that they do not want any disturbances at night.


Go back
Register New Disaster/Emergency
Province: Nairobi
District: Nairobi East
Division: KASARANI
Location: KAHAWA
Sublocation: KAHAWA WEST/JUA K
Year 2010
Month February
Day 2
Emergency Type Fire
Information Source Public
Disaster Id 201002021
Reported By: Julius Mwangi
Description: Huge fire at Kahawa west Supermarket
Submit


Broadcast Message
Go back
Select Emergency Coverage Area
Province: Nairobi
District: Nairobi East
Division: KASARANI
Location: KAHAWA
Sublocation: KAHAWA WEST/JUA K
Emergency Type: Fire
Fire Situation: Super market
Select Area: sublocation
Type the message: A huge fire has engulfed Kahawa West Supermarket
Send Message


EIDS Kenya
Emergency Information Dissemination System(EIDS)
Tuesday 02 February, 2010
Incident Reporter | Notification | Emergency Resources Manager | Maps | Reports | Directory | Logout | Help
sday 02 February, 2010 Supermarket fire : Huge fire has engulfed Kahawa
Main Page
Drought
Flood
Fire
Accidents
Disease Outbreaks
Kenya is faced with numerous disasters and emergiencies. The most common disasters are urban fires, floods, droughts, chorela outbreak, landslides and road accidents. You can report an Emergency by sending an SMS to this site. You can get emergency updates in your vicinity by registering to this site
Emergency Information disermination website

Type the name of the disaster location:

uhuru estate

Latitude -1.273833

Longitude 36.877950

Select the Map scale very small

Plot

CHAPTER 6: CONCLUSION & RECOMMENDATIONS

This project has demonstrated that there exist enormous opportunity in using Information systems in emergency and disaster information dissemination. While the traditional methods have been used by the news media to convey the information, the use of the SMS is an efficient media for communication where the public can use it to report an incident and can also use it to receive information on emergency situations. While the media has a huge coverage, it should not be the only link to the public. The preliminary survey has shown that not all people have radios and televisions and that the mobile phone is the most used.

The project has demonstrated that rather than the traditional fire alarms that only warn by raising the siren, fire alarms can be integrated with fire fighting equipments can activate the fire fighting equipments and also send an SMS to the owner and the other authorities for additional support.

RECOMMENDATIONS

The following are the recommendations

1. For the system to be operational then the use of other high capacity SMS gateways like the Clickatell is recommended since a mobile phone with a GSM is very slow and can not be able to serve a large population.
2. The use of the SMS and the its effectiveness in emergency information dissemination should be investigated further.
3. There is a huge opportunities in the use of the open source Arduino gadget in physical computing

4. The use of JQuerry and Json has powerful application capability in website application development

References

1. Gachet, A. (2004). Building Model-Driven Decision Support Systems with Dicodess. Zurich.
2. Gadomski A.M. et al. (1998). Integrated Parallel Bottom-up and Top-down Approach to the Development of Agent-based Intelligent DSSs for Emergency Management,TIEMS98, Washington, CiteSeerx - alfa:
3. Haag, Cummings, McCubbrey, Pinsonneault, Donovan (2000). Management Information Systems: For The Information Age. McGraw-Hill Ryerson Limited: 136-140. ISBN 0-072-81947-2
4. Haettenschwiler, P. (1999). Neues anwenderfreundliches Konzept der Entscheidungsunterstützung. Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft. Zurich, vdf Hochschulverlag AG: 189-208.
5. National Hiv/Aids Monitoring And Evaluation Framework 2005
6. Keen, P. G. W. (1978). Decision support systems: an organizational perspective. Reading, Mass., Addison-Wesley Pub. Co. ISBN 0-201-03667-3
7. Marakas, G. M. (1999). Decision support systems in the twenty-first century. Upper Saddle River, N.J., Prentice Hall.
8. Manitoba Finance. 1999. *Risk Management Policy Manual* Winnipeg: Government of Manitoba
9. Manitoba Health. 1997. *Community Health Needs Assessment Guidelines* Winnipeg: Government of Manitoba
10. *http://www.atmel.com/dyn/resources/prod_documents/doc8025.pdf 2/22010*
11. *http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove 2/2/2010*
12. *http://tldp.org/HOWTO/XML-RPC-HOWTO/xmlrpc-howto-competition.html 27/5/09*
13. *http://www.keele.ac.uk/depts/por/disaster.html 28/6/09*
14. *http://weblog.masukomi.org/writings/xml-rpc_vs_soap.htm 27/5/09*
15. *http://en.wikipedia.org/wiki/Software_prototyping 5/6/09*
16. http://www.ci.annapolis.md.us/faq.asp?page=1317
17. http://www.education.com/reference/article/Ref_Emergency_Disaster/
18. http://www.w3.org/TR/SOAP/
19. http://en.wikipedia.org/wiki/SOAP
20. **http://en.wikipedia.org/wiki/Emergency_Management_Information_System**

21. http://www.developershome.com/sms/GSMModemIntro.asp
22. http://www.developer.com/services/article.php/2195981
23. **http://chemistry.about.com/cs/howthingswork/a/aa071401a.htm** (28th dec 2009)
24. http://communication.howstuffworks.com/how-to-leverage-an-api-for-conferencing1.htm
25. Power, D. J. (1997). What is a DSS? The On-Line Executive Journal for Data-Intensive Decision Support 1(3).
26. Power, D.J. A Brief History of Decision Support Systems DSSResources.COM, World Wide Web, version 2.8, May 31, 2003.
27. Power, D. J. (2002). Decision support systems: concepts and resources for managers. Westport, Conn., Quorum Books.
28. Roy Fielding's Ph.D. dissertation "Architectural Styles and the Design of Network-based Software Architectures"
29. Sprague, R. H. and E. D. Carlson (1982). Building effective decision support systems. Englewood Cliffs, N.J., Prentice-Hall. ISBN 0-130-86215-0
30. Stanhope, P. (2002). Get in the Groove: building tools and peer-to-peer solutions with the Groove platform. New York, Hungry Minds
31. Stephens, W. and Middleton, T. (2002). Why has the uptake of Decision Support Systems been so poor? In: Crop-soil simulation models in developing countries. 129-148 (Eds R.B. Matthews and William Stephens). Wallingford:CABI.

APPENDIX A

Session security

Sessions and cookies are also two things where you have to watch out. Although they cannot breach your application's security they can be used to compromise user accounts through interception or stealing of the cookies. Cookie holds a value, a session identifier, which is associated with some sort of data on the server. If the user has a valid session ID then the data associated with the session will get into the `$_SESSION` super-global array. Sessions can also be transferred via the URL. Cross-site scripting (XSS) is the most common way through which cookies are stolen. If a valid session ID is stolen and that session is used for something like authentication then you will essentially be logged in as that user.

Issues with shared hosting

Most people host their website on what is called *shared hosting*. It is basically when there are multiple people having their websites hosted on a single server. On a server with a Linux operating system session data will by default be stored in the `/tmp` directory. It is a directory that stores temporary data and it will obviously have to be readable and writable by everyone. Therefore, if your session data is stored in there, which it is by default, then the other users can find it if they look hard enough. This poses the same security issues as with cookies being stolen using XSS.

Preventing session fixation

Since session ID can be stolen then there are ways to minimize the risk session fixation which includes:

Changing the session ID often. If we do that then the chance that the intercepted session ID will be valid will be greatly minimized if that ID changes often. We can use one of PHP' built-in functions called `session_regenerate_id()`. When we call this function the session

ID will be, no surprise, regenerated. The client will simply be informed that the ID has changed via an HTTP response header called `Set-Cookie`.

If you are using PHP 5.2+ then you can tell the browser that JavaScript should not be given access to the cookie using a flag called `http only`. You can set this flag using the `php.ini` directive called `session.cookie_httponly` or you can use the `session_set_cookie_params()` function.

Regarding the issue with the shared hosts, the fix is simple: store the data where only you have access. You can use the directive called `session.save_path` to set another path for storing them. You can also store them in a database, but then you will have to write your own handler using the function called `session_set_save_handler()`.

Encrypting Using PHP

PHP is highly equipped with encryption functions. A text or a password encryption can be done very easily using the functions md5() or sha1() or crypt() in php. It's a basic need to protect the password. The first step towards it is to encrypt it. Another security fixation is to create a security code image generator, or a CAPTCHA image, that is also helpful against spammers.

SQL Injection

One of PHP's greatest strengths is the ease with which it can communicate with databases, most notably MySQL. However, with that much power there are potentially huge security problems. The most common security hazard faced when interacting with a database is that of SQL Injection - when a user uses a security glitch to run SQL queries on your database.

Let's use a common example. Many login systems feature a line that looks a lot like this when checking the username and password entered into a form by a user against a database of

valid username and password combinations, for example to control access to an administration area:

```
$check = mysql_query("SELECT Username, Password, UserLevel FROM Users WHERE Username = '".$_POST['username']."' and Password = '".$_POST['password']."'");
```

Look familiar? It may well do. And on the face of it, the above does not look like it could do much damage. But let's say for a moment that I enter the following into the "username" input box in the form and submit it:

```
' OR 1=1 #
```

The query that is going to be executed will now look like this:

```
SELECT Username, Password FROM Users WHERE Username = '' OR 1=1 #' and Password = ''
```

The hash symbol (#) tells MySQL that everything following it is a comment and to ignore it. So it will actually only execute the SQL up to that point. As 1 always equals 1, the SQL will return all of the usernames and passwords from the database. And as the first username and password combination in most user login databases is the admin user, the person who simply entered a few symbols in a username box is now logged in as your website administrator, with the same powers they would have if they actually knew the username and password.

With a little creativity, the above can be exploited further, allowing a user to create their own login account, read credit card numbers or even wipe a database clean.

Fortunately, this type of vulnerability is easy enough to work around. By checking for apostrophes in the items we enter into the database, and removing or neutralising them, we can prevent anyone from running their own SQL code on our database. The function below would do the trick:

```
function make_safe($variable) {
$variable = mysql_real_escape_string(trim($variable));
return $variable;
```

```
}
```

Now, to modify our query. Instead of using _POST variables as in the query above, we now run all user data through the make_safe function, resulting in the following code:

```
$username = make_safe($_POST['username']);
$password = make_safe($_POST['password']);
$check = mysql_query("SELECT Username, Password, UserLevel FROM Users WHERE
Username = '".$username."' and Password = '".$password."'");
```

Now, if a user entered the malicious data above, the query will look like the following, which is perfectly harmless. The following query will select from a database where the username is equal to "\' OR 1=1 #".

```
SELECT Username, Password, UserLevel FROM Users WHERE Username = '\' OR 1=1
#' and Password = ''
```

Now, unless you happen to have a user with a very unusual username and a blank password, your malicious attacker will not be able to do any damage at all. It is important to check all data passed to your database like this, however secure you think it is. HTTP Headers sent from the user can be faked. Their referral address can be faked. Their browsers User Agent string can be faked. Do not trust a single piece of data sent by the user, though, and you will be fine.

How to secure MYSQL Database

By default, when you install **XAMPP** in your windows machine, the 'root' password for the MySQL is set to empty. But this is not recommended, as the MySQL database without a password will be accessible to everyone. To avoid this, a proper/secure password must be set to the user '**root**'. To do it in XAMPP, follow these steps

Resetting the 'root' user password in MySQL:

Start the Apache Server and MySQL instances from the XAMPP control panel.

After the server started, open any web browser and give http://localhost:8090/phpmyadmin/ (if you are running XAMPP on 8090 port). This will open the phpMyAdmin interface. Using this interface we can manager the MySQL server from the web browser.

In the phpMyAdmin window, select **SQL** tab from the right panel. This will open the SQL tab where we can run the SQL queries.

Now type the following query in the text area and click **Go**

```
UPDATE mysql.user SET Password=PASSWORD('password') WHERE User='root';
FLUSH PRIVILEGES;
```

Now you will see a message saying that the query has been executed successfully.

If you refresh the page, you will be getting a error message. This is because the phpMyAdmin configuration file is not aware of our newly set root password. To do this we have to modify the phpMyAdmin config file.

Modifying phpMyAdmin config file:

Open the file **[XAMPP Installation Path] / phpmyadmin / config.inc.php** in your favorite text editor.

Search for the string **$cfg['Servers'][$i]['password'] = '';** and change it to like this, **$cfg['Servers'][$i]['password'] = 'password';** Here the 'password' is what we set to the 'root' user using the SQL query.

Now all set to go. Save the config.inc.php file and restart the XAMPP server.

flock() is used to lock a file so that two or more people do not get access to it at the same time. This helps protect the file from being corrupted. flock() takes two arguments: a file handler and a lock type.

File Locking

Lock Type Explanation

LOCK_SH Reading lock. Others can read file.

Lock Type Explanation

LOCK_EX Exclusive lock. The file cannot be opened by others.

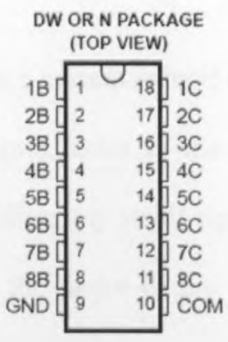
LOCK_UN Unlocks file.

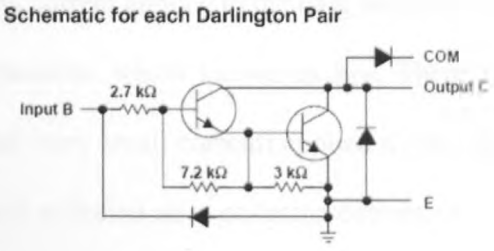
If a file is already locked by another user, flock() waits to get a lock. LOCK_NB LOCK_NB tells it not to wait

EXPRESS is a standard **data modeling language** for **product** data. EXPRESS is formalized in the ISO Standard for the Exchange of Product model **STEP (ISO 10303)**, and standardized as **ISO 10303**

The Darlington Transistor Array

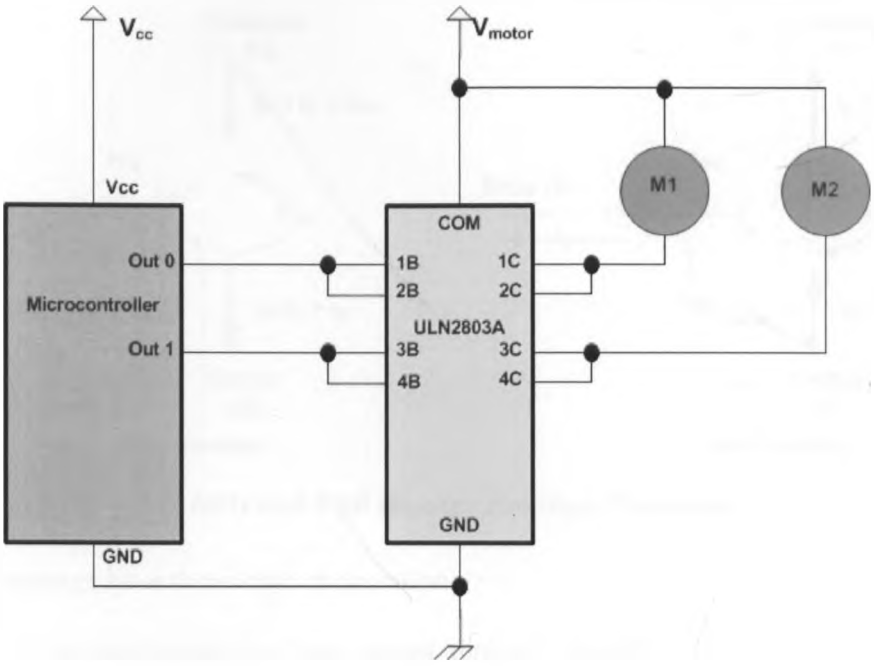
For more compact version of the Darlington pair transistor you could use the Texas Instrument ULN2803A which is contain 8 Darlington pair transistors with has build in 2K7 base resistor and clamp diode for each Darlington pair transistors. This makes this Darlington transistor array suitable for driving the relay or motor up to 500mA directly from the microcontroller output.





Texas Instrument ULN2803A Darlington Transistor

To increase the output current up to 1 A (2 x 500mA) you could simply use two Darlington transistor array connected in parallel, the following is the sample circuit for driving two DC motors using the ULN2803A Darlington transistor array:

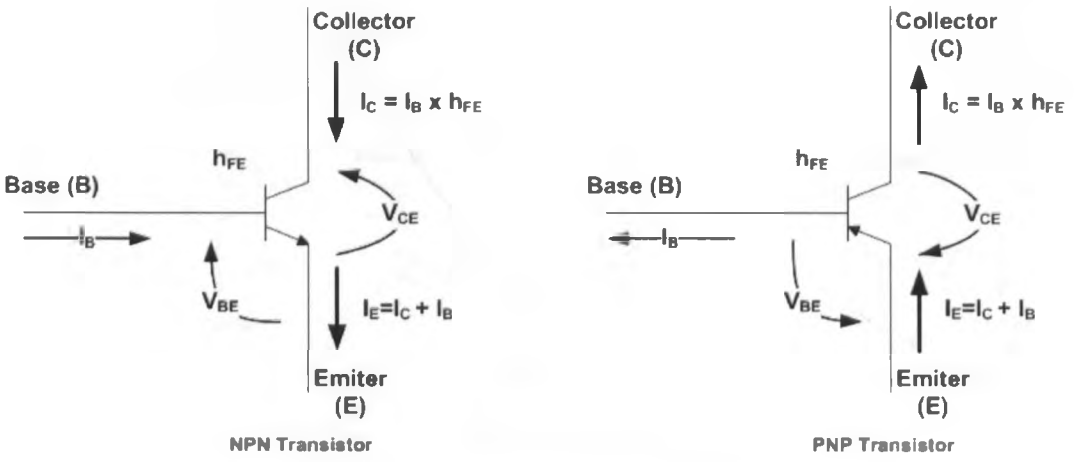


Driving two DC Motors using ULN2803A Darlington Transistor

Thanks to the build in internal 2K7 base resistor and the clamp diode, you don't need any external component when using ULN2803A to drive the DC motor from your microcontroller port. The Darlington transistor array ULN2803A could be used to drive up to 50 volt voltage load.

The Switch

The transistor actually works as a current gainer; any current applied to the base terminal will be multiplied by the current gain factor of the transistor which known as $\mathbf{h_{FE}}$. Therefore transistor can be used as amplifier; any small signal (very small current) applied to the base terminal will be amplified by the factor of h_{FE} and reflected as a collector current on the collector terminal side.



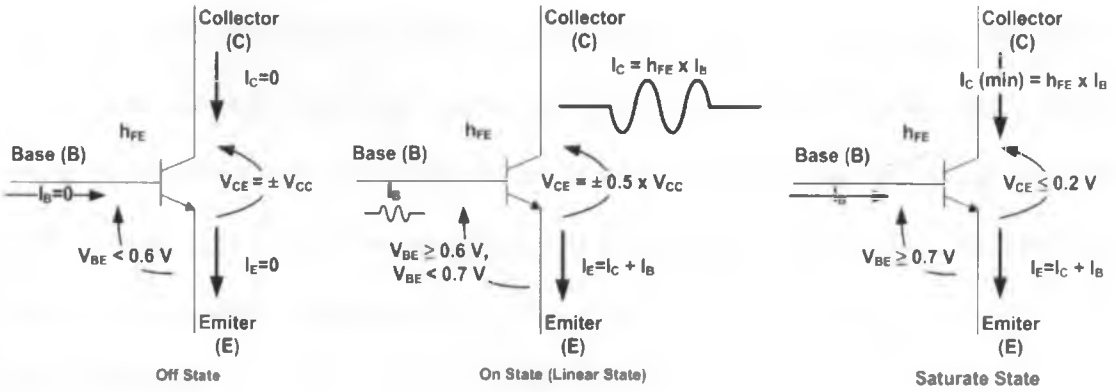
NPN and PNP Bipolar Junction Transistor

All the transistors have three state of operation:

Off state: in this state there is no base current applied or $\mathbf{I_B = 0}$.

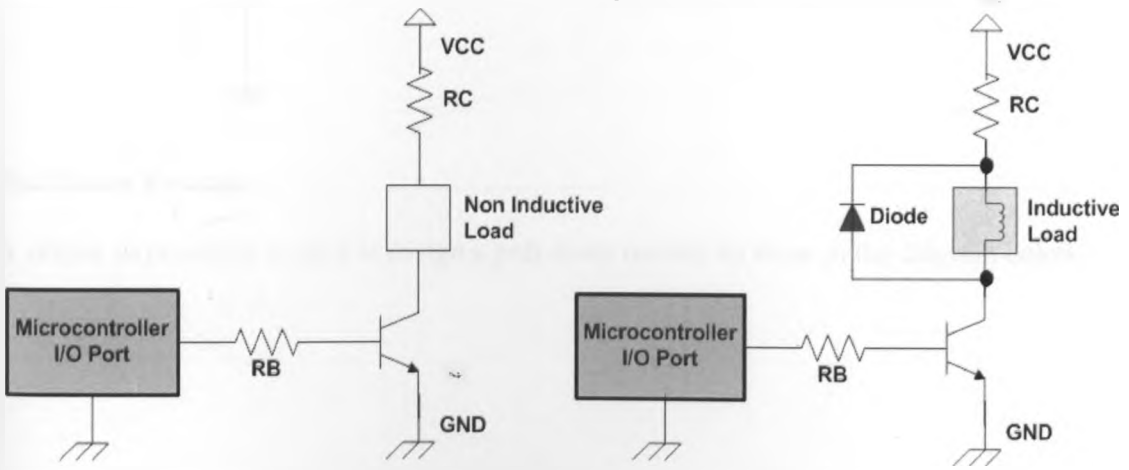
On active state: in this state any changes in I_B will cause changes in I_C as well or $\mathbf{I_C = I_B \times h_{FE}}$. This type of state is suitable when we use transistor as a signal amplifier because transistor is said is in the linear state. For example if we have a transistor with gain of 100 and we increase the I_B from 10uA to 100uA; this will cause the I_C to swing from 1000uA to 10000uA (1 mA to 10 mA).

On saturate state: in this state any changes in I_B will not cause changes in I_C anymore (not linear) or we could say I_C is nearly constant. This is the type of state that we are looking for on this tutorial.



The Transistor State

From the picture above we could see the voltage and current condition of transistor on each state; if you notice when transistor is in off state the voltage across collector and emitter terminal is equal to the supplied voltage, this is equivalent to the open circuit and when transistor is in saturate state the collector to emitter voltage is less then 0.2 Volt which is equivalent to the close circuit. Therefore to use transistor as a switch we have to make transistor **OFF** which equivalent to the logical "**0**" and **SATURATE** which is equivalent to the logical "**1**".



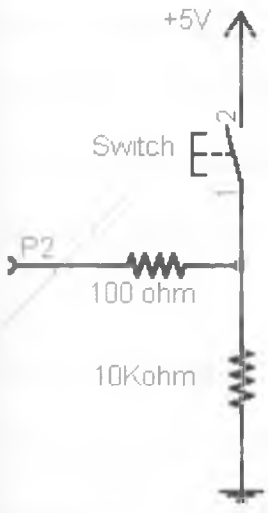
Typical Microcontroller interface Circuit

The diagram above showing typical microcontroller interface circuit using NPN transistor; the **RB** resistor is used to control the current on base terminal that make transistor **OFF** and

ON (saturate). The diode (also known as the clamp diode) in the inductive load circuit is needed to protect the transistor again the EMF (Electromotive Force) voltage generated by the inductor component when the transistor is switched on and off rapidly, this voltage is oppose the source voltage. The diode will act as a short circuit to the high voltage generated by the inductor component. The general purpose diode capable of handling minimum 1 A of current are the 1N4001, 1N4002, etc.

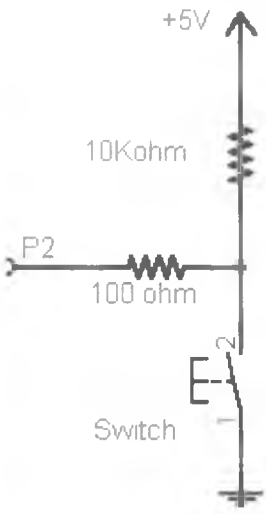
Pull-Up Resistor.

The diagram below show how a pull up resistor works. Initially P2 is at ground before the switch is closed. When the switch is closed then P2 will be at +5V.



Pull-Down Resistor

A similar explanation is used to design a pull down resistor as show in the diagram below.

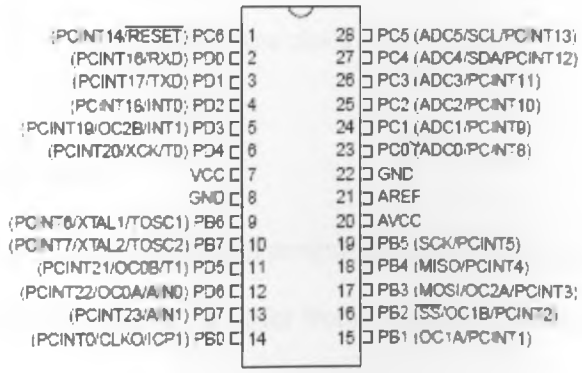


Microcontrollers

Microcontroller is designed to be a fully function computing device.. No other external components are needed for its application because all necessary peripherals are already built into it to save the time and space needed to construct devices.

The ATMEGA328

The following is a brief technical description of the atmega328 which was used in the construction of the Alarm.



Pin Descriptions

VCC : Digital supply voltage.

GND: Ground.

Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). ThePort B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is

un programmed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running.

Port D (PD7:0)

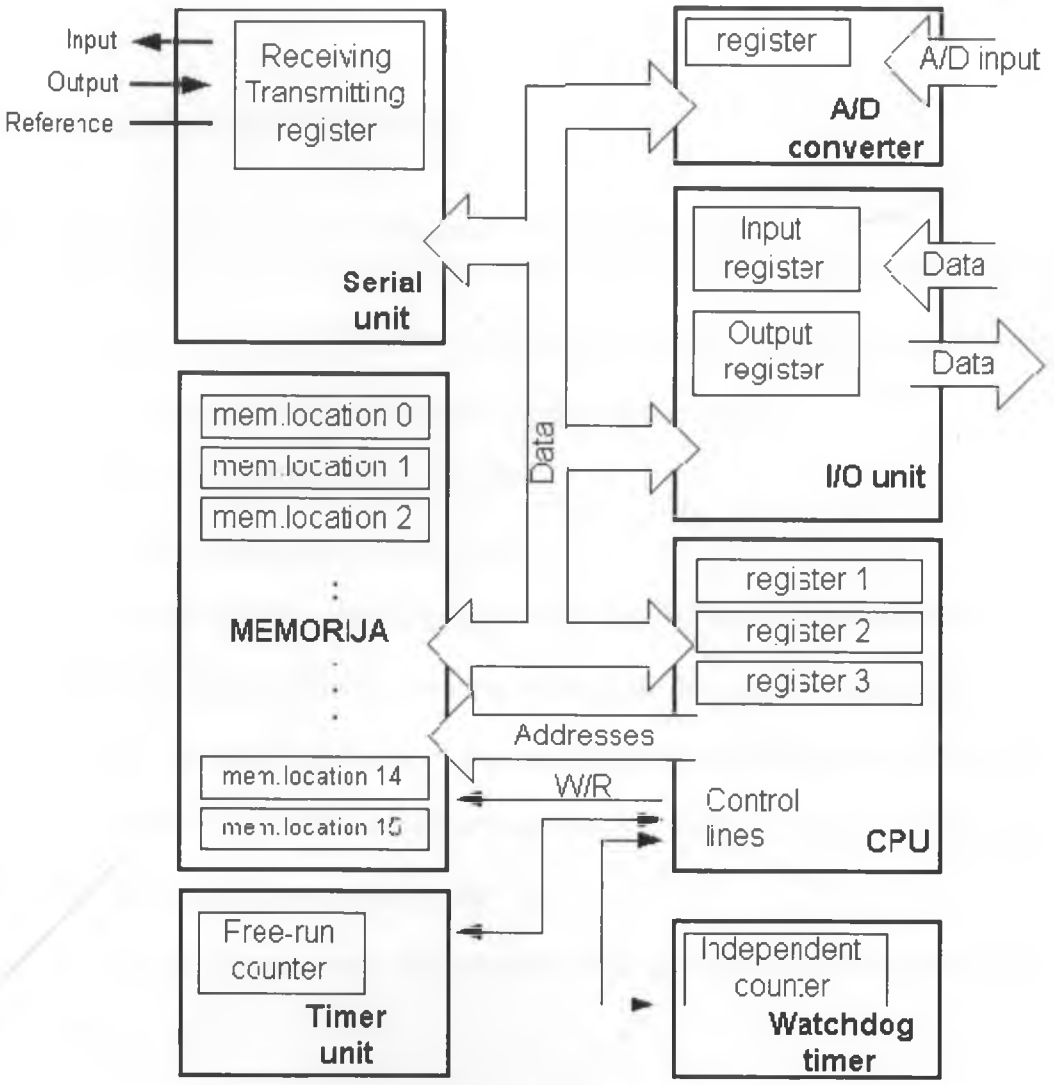
Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

ATMEGA 328 Uses the protocol we've just described is called in professional literature NRZ (Non-Return to Zero).

The diagram below shows how analog signals can be connected to a microcontroller using the Analog to Digital Converter

Block for converting an analogue to a digital form

Physical configuration of the interior of a microcontroller



Microcontroller outline with its basic elements and internal connections PIC

For a real application, a microcontroller alone is not enough. Beside a microcontroller, we need a program that would be executed, and a few more elements which make up a interface logic towards the elements of regulation (which will be discussed in later chapters).

APPENDIX B USER MANUAL

This system is made up of a website module and an SMS module.

1. Make sure you are running MYSQL server , and Apache on your computer. An ideal software will be XAMPP server that loads the services. XaMMP can be downloaded on http://www.apachefriends.org/en/xampp-windows.html
2. Load the program into the root server
3. COPY the database into MySql server
4. Once all the files have been copied the Login and fill in the registration form
5. Before you can sent Emails you have to configure the Apache Web server to your internet provider. If you are you are using the Safaricom Modem the you have to set it to SMTP Go to PHP.ini look for [mail.functio] set the following For Win32 only set SMTP=safaricom and smtp port = 25
6. You need to open a email account with google and insert your email setting into the PHP sms code.
7. To use the SMS program you need to install acticeXperts software into you system. Once you have installed the active expert create a trigger and copy the file into that director. You need to connect your computer to a phone with a GSM moderm or a GSM modem. Once everything in order you will the export the database to MySql database using the database utility. Follw the instructions. If your Mysql database has a password you should provide the correct password so that the SMS program can read and write to your database
8. Once the setup is completed, make sure that if you want to send email and SMS you have connected your Safaricom modem and is connected and you have also connected the GSM modem and is running.

APPENDIX C: SAMPLE PROGRAM CODE

```
' // ===================================================================

'// VB script Program to read SMS and Broadcast SMS Message by reading mobile numbers from msql database

' // ===================================================================

Option Explicit

CONST STR_DEBUGFILE    = "C:\Program Files\ActiveXperts\SMS Messaging Server\Sys\Tmp\dds.txt"

' Declaration of global objects

Dim g_objMessageDB, g_objDebugger, g_objConstants

' Creation of global objects

Set g_objConstants     = CreateObject( "AxSmsServer.Constants" )

Set g_objMessageDB     = CreateObject( "AxSmsServer.MessageDB" )

Set g_objDebugger      = CreateObject( "ActiveXperts.VbDebugger" )

' Set Debug file - for troubleshooting purposes

g_objDebugger.DebugFile = STR_DEBUGFILE

' // ===================================================================

' // Function: ProcessMessage

' // -------------------------------------------------------------------

' // ProcessMessage trigger function to process incoming messages

' // ===================================================================

Function ProcessMessage( numMessageID )

  Dim objMessageIn, objMessageOut, arrMessage

  Dim strMessageOutBody

  Dim numSongID, strCommand

  g_objDebugger.WriteLine ">> ProcessMessage"
```

```
' Open the Message Database
g_objMessageDB.Open
If( g_objMessageDB.LastError <> 0 ) Then
  g_objDebugger.WriteLine "<< ProcessMessage,  unable to open database"
  Exit Function
End If
' Retrieve the message that has just been received. If it fails then exit script
Set objMessageIn  = g_objMessageDB.FindFirstMessage( "ID = " & numMessageID )
If g_objMessageDB.LastError <> 0 Then
  g_objMessageDB.Close
  g_objDebugger.WriteLine "<< ProcessMessage,  FindFirstMessage failed, error: [" &
g_objMessageDB.LastError & "]"
  Exit Function
End If

' Set incoming SMS message status to: SUCCESS (previous state was: PENDING)
objMessageIn.Status = g_objConstants.MESSAGESTATUS_SUCCESS
g_objMessageDB.Save objMessageIn
g_objDebugger.WriteLine " Incoming message saved, result: [" & g_objMessageDB.LastError & "]"

' Optional: Modify any property of the incoming message, and save it
'objMessageIn.CustomField1 = 0   ' CustomField1 is an all-purpose numeric field
' objMessageIn.CustomField2 = ""   ' CustomField2 is an all-purpose string field
'g_objMessageDB.Save objMessageIn
```

```
  'g_objDebugger.WriteLine "Incoming message saved, result: [" & g_objMessageDB.LastError & "]"
    ReplyMessage( objMessageIn )
     Broadcast( objMessageIn )
  ' Close the Message Database
  g_objMessageDB.Close
  g_objDebugger.WriteLine "<< ProcessMessage"
End Function
' // ===========================================================================
' // ReplyMessage Function
' // ---------------------------------------------------------------------------
' // This function will send a reply to manager
' // ===========================================================================
Function ReplyMessage( objMessageIn)
  Dim  objMessageOut, arrMessage
   Dim strMessageOutBody
' Split received message body into pieces (separated by spaces)
   arrMessage = Split( UCase( objMessageIn.Body ), " " )
strMessageOutBody = "You have reported " + arrMessage(0)
  g_objDebugger.WriteLine ">> YourFunction"
     arrMessage = Split( UCase( objMessageIn.Body ), " " )
strMessageOutBody = "You have reported " + arrMessage(0)
Set objMessageOut = g_objMessageDB.Create
    If( g_objMessageDB.LastError = 0 ) Then
      objMessageOut.Direction = g_objConstants.MESSAGEDIRECTION_OUT
      objMessageOut.Type      = objMessageIn.Type
```

```
    objMessageOut.Status    = g_objConstants.MESSAGESTATUS_PENDING
    objMessageOut.Recipient = objMessageIn.Sender
    objMessageOut.ChannelID = 0  ' Any available SMS channel
    objMessageOut.Body      = "Your message has been received"
     'objmessageOut.Body = strMessageOutBody
    g_objMessageDB.Save objMessageOut
   End If
 g_objDebugger.WriteLine "<< YourFunction"
End Function
' // ===================================================================
' //broadcast  Function
' // -------------------------------------------------------------------
' // This function will query database and sent message to all
' // ===================================================================
Function Broadcast( objMessageIn)
 Dim objMessageOut
  Dim strMessageOutBody
  Dim arrMessage
    Dim oCn, ORs, ConnectionString, item
 g_objDebugger.WriteLine "<< YourFunction"
   Set oCn = CreateObject("ADODB.Connection")
Set oRs = CreateObject("ADODB.Recordset")
ConnectionString = "DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;" & _
                   "DATABASE=project;" & _
                   "USER=root;" & _
```

```
                "PASSWORD=mutheuwawewa;" & _
                "OPTION=3;"
'For a weird reason you can not put the DRIVER option on a seperate line.
'SERVER: should be the hostname of your mysql server (localhost is a common value)
'DATABASE: The name of the database you want to get information from
'USER&PASSWORD: err.. :)
'OPTION: See the mysql documentation on the odbc driver for information on options
'       It is a bitmask. Which means if you want option 1 & 2 you put 3.
'       If you would like to have options 1, 2 and 8 you put '11'.

oCn.open(ConnectionString)
'Open your connection
oRs.Open "Select * from person", oCn
'objRS.Open strQuery
If oRs.EOF Then
'Wscript.Echo "Record cannot be found."
Else
Do until oRs.EOF
'sent sms messages
Set objMessageOut = g_objMessageDB.Create
  If( g_objMessageDB.LastError = 0 ) Then
    objMessageOut.Direction = g_objConstants.MESSAGEDIRECTION_OUT
    objMessageOut.Type      = objMessageIn.Type
    objMessageOut.Status    = g_objConstants.MESSAGESTATUS_PENDING
    objMessageOut.Recipient = oRs("mobile")
```

```
    'objMessageOut.Recipient = objMessageIn.Sender
    objMessageOut.ChannelID = 0  ' Any available SMS channel
   ' objMessageOut.Body     = oRs("locationID")
     objmessageOut.Body = objMessageIn.Body
    g_objMessageDB.Save objMessageOut
   End If
oRs.movenext
  'wscript.echo oRs("login") & " " & objRS("password")
loop
End If
oRs.close()
oCn.close()
'Close all connections
End Function
```

ARDUINO USED TO RUN THE FIRE ALARM

```
/*
 * Switch and LED test program
 */

int ledPin = 12;            // LED is connected to pin 12
int led2Pin = 10;           // LED is connected to pin 12
int testPin = 4;
int switchPin = 2;          // switch is connected to pin 2
int sensor2   = 3;          // sensor 2 active by switch
int alarmPin = 11;
int offPin = 5;             // Press this to switch off alarm
int offlightPin = 7;        // lights when off switch is pressed
int val;                    // variable for reading the pin status
int val2;
int valoff;                 // variable for storing off pin status
int motor1 = 9;             // first motor low speed
int motor1H = 6;            // first motor high speed
int motor2 = 8;             // second motor
int stateSensor1;           // status of sensors and fire sensor1
int stateSensor2;           // status of sensors and fire sensor2
int status;                 // status of sensors and fire extingushers
int pin = 0; // analog pin
int pin2 =1; // analog pin for temperature sensor 2
double temp;  //Temperature
```

```
double temp2;  // temperature of sensor 2
int temppy;
int i, j;
int test;
int flagML = 0;   //flag to check if motor 1 low speed is on set to 1 when on
int flagMH = 0;   //flag to check if motor 1 highspeed is on set to 1 when on
double samples[10]; // variables to make a better precision
double samples2[10]; // variables to make a better precision
void setup() {
  pinMode(ledPin, OUTPUT);      // Set the LED pin as output
  pinMode(led2Pin, OUTPUT);      // Set the LED pin as output
  pinMode(testPin, OUTPUT);      // Set the LED pin as output
  pinMode(alarmPin, OUTPUT);      // Set the Alarm pin as output
  pinMode(switchPin, INPUT);    // Set the switch pin as input
  pinMode(sensor2 , INPUT);    // Set the sensor2  pin as input
  pinMode(offPin, INPUT);    // Set the off switch pin as input
  pinMode(offlightPin, OUTPUT);    // Set the off light as output
  pinMode(motor2, OUTPUT);    // Set the motor2 pin as output
   pinMode(motor1, OUTPUT);    // Set the motor1 lowspeed pin as output
    pinMode(motor1H, OUTPUT);    // Set the motor1 Hispeed  pin as output
  Serial.begin(9600); //get a serial output
}
```

```
void loop(){
  temp = 0;  // initialize temperature sensor on LM35
   temp2 = 0;
  for(i = 0; i <= 9; i++) { //reset
       samples[i] = 0;
       samples2[i] = 0;
     }
    for(i = 0; i <= 9; i++) { // gets 8 samples of temperature
    samples[i] = ( 5.0 * analogRead(pin) * 100.0) / 1024.0;
      temp = temp + samples[i];
    samples2[i] = ( 5.0 * analogRead(pin2) * 100.0) / 1024.0;
     temp2 = temp2 + samples2[i];
     delay(50);
    }
    temp = temp/10.0; // better precision
    temp2 = temp2/10.0;
  //  temp = ( 5.0 * analogRead(pin) * 100.0) / 1024.0;
  if(temp > 28) {
   if (flagML == 1)
      {
       flagML = 0;
       digitalWrite(motor1, LOW);   // turn motor1 low speed  off
       digitalWrite(motor1H, HIGH);  // turn motor1 high speed on
       flagMH = 1;
      }
```

```
  }
   if(temp < 28) {
    if (flagMH == 1 ) {
     flagMH =0;
     digitalWrite(motor1, HIGH);    // turn motor1 low speed  o
        digitalWrite(motor1H, LOW);  // turn motor1 high speed on
       flagMH = 0;
     flagML = 1;
     }
   }


if (Serial.available()) { //read from the python program
 stateSensor1 = digitalRead(switchPin);
 stateSensor2 = digitalRead(sensor2);
  int serData = Serial.read();
   switch (serData) { // use the data send to Arduino by Python program
 case '0':
   digitalWrite(testPin,HIGH);
   digitalWrite(alarmPin, HIGH);
   break;
 case '1':
   digitalWrite(testPin,LOW);
   digitalWrite(alarmPin, LOW);
   break;
 case '2':
```

```
   digitalWrite(testPin,HIGH);
   digitalWrite(alarmPin, HIGH);
  delay(1000);
   digitalWrite(motor1, HIGH);   // turn motor on
   flagML = 1;
  break;
 case '3':
   digitalWrite(motor1, LOW);   // turn motor off
   flagML = 0;
  break;
 case '4':
   digitalWrite(testPin,HIGH);
   digitalWrite(alarmPin, HIGH);
  delay(1000);
   digitalWrite(motor2, HIGH);   // turn motor on
  break;
 case '5':
   digitalWrite(motor2, LOW);   // turn motor off
  break;
case '6':
   digitalWrite(testPin,HIGH);
   digitalWrite(alarmPin, HIGH);
  delay(1000);
   digitalWrite(motor1, HIGH);   // turn motor on
   flagML = 1;
```

```
    digitalWrite(motor2, HIGH);   // turn motor on
    break;
 case '7':
    digitalWrite(testPin,LOW);
    digitalWrite(alarmPin, LOW);
    delay(1000);
    digitalWrite(motor2, LOW);   // turn motor on
    digitalWrite(motor1, LOW);   // turn motor on
    flagML = 0;
     break;
  case '8':
    Serial.println(temp);
   Serial.println(temp2);
   delay(100);
    break;
  default:
    Serial.println("Illegal operation ");
  }
 Serial.println(stateSensor1);
  Serial.println(stateSensor2);
 delay(1000);
}
// This is the code for running the Alarm when arduino is not communicating with the python
 val = digitalRead(switchPin);  // read input value and store it in val
 if (val == LOW) {            // check if the button is pressed
```

```
  digitalWrite(ledPin, HIGH);   // turn LED on
  digitalWrite(alarmPin, HIGH);
  digitalWrite(testPin, HIGH);
   delay(1000);
     digitalWrite(motor1, HIGH);
     flagML = 1;
 }
 if (val == HIGH) {            // check if the button is not pressed
  digitalWrite(ledPin, LOW);    // turn LED off
 }
val2 = digitalRead(sensor2);   // read input value and store it in val
 if (val2 == LOW) {            // check if the button is pressed
  digitalWrite(led2Pin, HIGH);   // turn LED on
  digitalWrite(alarmPin, HIGH);
  digitalWrite(testPin, HIGH);
   delay(1000);
     digitalWrite(motor2, HIGH);
 }
 if (val2 == HIGH) {           // check if the button is not pressed
  digitalWrite(led2Pin, LOW);    // turn LED off
 }
 valoff = digitalRead(offPin);   // read input value and store it in valoff
 if (valoff == LOW) {            // check if the button is pressed
  digitalWrite(offlightPin, HIGH);   // turn LED on
  digitalWrite(alarmPin, LOW);
```

```
  digitalWrite(testPin, LOW);
  digitalWrite(motor2, LOW);
  digitalWrite(motor1, LOW);
  flagML = 0;
 }
 if (valoff == HIGH) {          // check if the button is not pressed
  digitalWrite(offlightPin, LOW);   // turn LED off
 }

}
```

PYTHON CODE TO READ ARDUINO AND SENT SMS

```
import serial
import time
from decimal import *
def sentOwnerSmsSensor1(temp): # Function to send SMS to owner of house on fire
  port = 'COM8'
  try:
    ser1 = serial.Serial(port, 9600, timeout=3)
    ser1.write('AT+CMGF=1\r')
    ser1.write('AT+CMGS="0752762511"\r')
    #ser1.write(message)
    ser1.write('Your house may be on fire. smoke sensor on, room temperature is = ' + str(temp) + '
Centigrades  contact fire dept on 0752762512')
    ser1.write(chr(0x1A))
```

```
        ser1.close()

        print 'Your house may be on fire. smoke sensor on, room temperature is = ' + str(temp) + '
Centigrades  contact fire dstn 0752762512'

        print

    except:

        print "Failed to connect to GSM Modem SMS 'sensor 1 is on' not sent and the room temperature is
= " + str(temp)

#end of function

def sentSmsSensor1(temp):

    port = 'COM8'

    try:

        ser1 = serial.Serial(port, 9600, timeout=3)

        ser1.write('AT+CMGF=1\r')

        ser1.write('AT+CMGS="0752762512"\r')

        #ser1.write(message)

        ser1.write('Fire Smoke  sensor on, room temperature is = ' + str(temp) + ' Centigrades , location
uhuru estate 1.273833S, 36.877950E owner Mr Mutua 0751518135')

        ser1.write(chr(0x1A))

        ser1.close()

        print 'sms message sent: sensor 1 is on and the room temperature is = ' + str(temp) + ' degree
Centigrades located at uhuru estate 1.273833S, 36.877950E '

        print

    except:

        print "Failed to connect to GSM Modem SMS 'sensor 1 is on' not sent and the room temperature is
= " + str(temp)
```

```
#end of function


def sentSmsSensor2(temp):
   port = 'COM8'
   try:
      ser1 = serial.Serial(port, 9600, timeout=3)
      ser1.write('AT+CMGF=1\r')
      ser1.write('AT+CMGS="0752762512"\r')
      #ser1.write(message)
      ser1.write('Fire Smoke sensor on the room temperature  = ' + str(temp)+ ' Centigrades  and is located mathare north -1.115 , 36.786 owner Mwangi 0751505152')
      ser1.write(chr(0x1A))


      ser1.close()
      print 'sms message sent: sensor 2 is on and the room temperature is = ' + str(temp) + ' degree Centigrades at location -1.115 36.786 '
   except:
      print "Failed to connect to GSM Modem SMS 'sensor 2 is on and ' not sent and the room temperature is = " + str(temp)


#end of function
def sentSmstemp1():
   port = 'COM8'
   try:
      ser1 = serial.Serial(port, 9600, timeout=3)
      message = 'how are you'
```

```
        ser1.write('AT+CMGF=1\r')
        ser1.write('AT+CMGS="0752762512"\r')
        #ser1.write(message)
        ser1.write('House on fire temperature above critical level location  uhuru estate 1.273833S, 36.877950E owner Mr Mutua 0751518135')
        ser1.write(chr(0x1A))
        ser1.close()
        print 'sms message sent: Temperature room 1 rising'
        print
    except:
        print 'No connection to GSM Moderm SMS not sent'
def sentSmstemp2():
    port = 'COM8'
    try:
        ser1 = serial.Serial(port, 9600, timeout=3)
        ser1.write('AT+CMGF=1\r')
        ser1.write('AT+CMGS="0752762512"\r')
        #ser1.write(message)
        ser1.write('House on fire temperature above critical level located Mathare north -1.115 , 36.786 owner Mwangi 0751505152')
        ser1.write(chr(0x1A))
        ser1.close()
        print 'sms message sent:  Temperature in room 2 rising'
        print
    except:
        print 'No connection to GSM Moderm SMS "Temperature in room 2 rising" not sent'
```

```
    serialport = 'COM40' #the com port used for serial communication with Arduino
try:
    ser = serial.Serial(serialport, 9600, timeout=1)
    count = 0
    ser.setDTR(True)
    line = ser.readline()#how to get most recent line sent from device?
    line = line.rstrip("\r\n")
except:
    print "Failed to connect to fire alarm system"
time.sleep(1)
i = 0
flag1 = 0 #flag for sensor 1
flag2 = 0 #flag for sensor 2 to be used to sent sms if no yet
flag3 = 0 #Temperature sensor1
flag4 = 0 #Temperature sensor2
code = 8
if (code == 8):
    j = 20
else:
    j = 1
while (i < j):
    #code = 8
    if (code <= 8):
        try:
            ser.write(code)
```

```
time.sleep(1.8)
line = ser.readline()#how to get most recent line sent from device?
line = line.rstrip("\r\n")
time.sleep(.1)
line2 = ser.readline()#how to get most recent line sent from device?
line2 = line2.rstrip("\r\n")
if(code == 0):
   print " alarm on"
if(code == 2):
   print " Alarm and motor 1 on"
if(code == 4):
   print " Alarm and motor 2 on"
if(code == 6):
   print " Alarm and motor 1 and motor 2 on"
if(code == 8):
   print
   print "Temperature 1 = " + line + " degree centigrade"
   print "Temperature 2 = " + line2 + " degree centigrade"
   temp1 = Decimal(line)
   temp2 = Decimal(line2)
   if(temp1 > 28):
      if(flag3 == 0):
         sentSmstemp1()
         flag3 = 1
```

```
    if (temp2 > 28):
        if (flag4 == 0):
            sentSmstemp2()
            flag4 = 1
    difference =   temp1 - temp2
    print  "Temperature diference = " + str(difference)
sensor1 = ser.readline()
sensor1 = sensor1.rstrip("\r\n")

if (sensor1 == '1'):
    print "Sensor 1 off"
    flag1 = 0
if (sensor1 == '0'):
    print "Sensor 1 On"
    if (flag1 == 0):
        sentSmsSensor1(temp1)
        time.sleep(4)
        sentOwnerSmsSensor1(temp1)
        flag1 = 1
time.sleep(1)
sensor2 = ser.readline()#how to get most recent line sent from device?
sensor2 = sensor2.rstrip("\r\n")
if (sensor2 == '1'):
    print "Sensor 2 off"
if (sensor2 == '0'):
```

```
            print "Sensor 2 On"
            if (flag2 == 0):
                sentSmsSensor2(temp2)
                flag2 = 1
        time.sleep(1)
    except:
        print "Failed to connect to fire alarm system"

else:
    print "Invalid code"
    try:
        sensor1 = ser.readline()#how to get most recent line sent from device?
        sensor1 = sensor1.rstrip("\r\n")
        if (sensor1 == '1'):
            print "Sensor 1 off"
        if (sensor1 == '0'):
            print "Sensor 1 On"

        sensor2 = ser.readline()#how to get most recent line sent from device?
        sensor2 = sensor2.rstrip("\r\n")
    except:
        print "Failed to connect to fire alarm system"
    if (sensor2 == '1'):
        print "Sensor 2 off"
    if (sensor2 == '0'):
```

```
            print "Sensor 2 On"
        if (sensor2 == '1'):
            print "Sensor 2 off"
        if (sensor2 == '0'):
            print "Sensor 2 On"
        time.sleep(1)
    i = i + 1
try:
    ser.flush()
    ser.close()
except:
    print "Failed to connect to fire alarm system"
```

GOOGLE MAP

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Google Map using name of disaster location</title>
    <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAARrXu400eRVLMxsNCpIl
S6RQOIG0d60HdZ9IAVkGDv0H-oQ-DnBTIBDukdezT1IrMKNjIRb5AlwXOFg"
            type="text/javascript"></script>
<script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
    //<![CDATA[
// create icons to be used to map different items
    var iconBlue = new GIcon();
    iconBlue.image = 'images/mm_20_blue.png';
    iconBlue.shadow = 'images/mm_20_shadow.png';
    iconBlue.iconSize = new GSize(12, 20);
    iconBlue.shadowSize = new GSize(22, 20);
    iconBlue.iconAnchor = new GPoint(6, 20);
    iconBlue.infoWindowAnchor = new GPoint(5, 1);

        var iconGreen = new GIcon();
    iconGreen.image = 'images/iimm2-green.png';
    iconGreen.shadow = 'images/iimm2-shadow.png';
    iconGreen.iconSize = new GSize(12, 20);
    iconGreen.shadowSize = new GSize(22, 20);
    iconGreen.iconAnchor = new GPoint(6, 20);
    iconGreen.infoWindowAnchor = new GPoint(5, 1);


    var iconRed = new GIcon();
    iconRed.image = 'images//iimm2-red.png';
    iconRed.shadow = 'images/iimm2-shadow.png';
    iconRed.iconSize = new GSize(12, 20);
    iconRed.shadowSize = new GSize(22, 20);
    iconRed.iconAnchor = new GPoint(6, 20);
    iconRed.infoWindowAnchor = new GPoint(5, 1);

        var iconPurple = new GIcon();
    iconPurple.image = 'images/mm_20_purple.png';
    iconPurple.shadow = 'images/mm_20_shadow.png';
    iconPurple.iconSize = new GSize(12, 20);
    iconPurple.shadowSize = new GSize(22, 20);
    iconPurple.iconAnchor = new GPoint(6, 20);
    iconPurple.infoWindowAnchor = new GPoint(5, 1);

        var iconHosiptal = new GIcon();
    iconHosiptal.image = 'images/hospitals_002.png';
    iconHosiptal.shadow = 'images/hospitals.png';
    iconHosiptal.iconSize = new GSize(12, 20);
    iconHosiptal.shadowSize = new GSize(22, 20);
```

```
iconHosiptal.iconAnchor = new GPoint(6, 20);
iconHosiptal.infoWindowAnchor = new GPoint(5, 1);

    var iconFireStation = new GIcon();
iconFireStation.image = 'images/firedept.png';
iconFireStation.shadow = 'images/firedept_002.png';
iconFireStation.iconSize = new GSize(12, 20);
iconFireStation.shadowSize = new GSize(22, 20);
iconFireStation.iconAnchor = new GPoint(6, 20);
iconFireStation.infoWindowAnchor = new GPoint(5, 1);

    var iconPolice = new GIcon();
iconPolice.image = 'images/police.png';
iconPolice.shadow = 'images/police_002.png';
iconPolice.iconSize = new GSize(12, 20);
iconPolice.shadowSize = new GSize(22, 20);
iconPolice.iconAnchor = new GPoint(6, 20);
iconPolice.infoWindowAnchor = new GPoint(5, 1);


    var iconAmbulance = new GIcon();
iconAmbulance.image = 'images/truck.png';
iconAmbulance.shadow = 'images/truck_002.png';
iconAmbulance.iconSize = new GSize(12, 20);
iconAmbulance.shadowSize = new GSize(22, 20);
iconAmbulance.iconAnchor = new GPoint(6, 20);
iconAmbulance.infoWindowAnchor = new GPoint(5, 1);

    var iconDisaster = new GIcon();
iconDisaster.image = 'images/red-dot.png';
//iconDisaster.shadow = 'images/truck_002.png';
iconDisaster.iconSize = new GSize(30, 36);
// iconDisaster.shadowSize = new GSize(22, 20);
iconDisaster.iconAnchor = new GPoint(6, 20);
iconDisaster.infoWindowAnchor = new GPoint(5, 1);

var customIcons = [];
//customIcons["restaurant"] = iconBlue;
// customIcons["bar"] = iconRed;
    customIcons["hospital"] = iconHosiptal;
    customIcons["ambulance"] = iconAmbulance;
    customIcons["fire"] = iconFireStation;
    customIcons["police"] = iconPolice;
    customIcons["disaster"] = iconDisaster;
    customIcons["working"] = iconGreen;
    customIcons["notworking"] = iconRed;
// function load()
    function load(lat, logi, scale){
    //alert (' latitude = '+ lat + ' long = ' + logi);
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());

  //map.setCenter(new GLatLng(-1.28, 36.81), 13);
```

```
          //center the point to the disaster location

          map.setCenter(new GLatLng(lat, logi), scale);


           var disterpoint = new GLatLng(lat, logi);
        var marker = createMarker(disterpoint, "disaster", "", "disaster");
            map.addOverlay(marker);

         GDownloadUrl("phpsqlajax_genxml.php", function(data) {
           var xml = GXml.parse(data);
           var markers = xml.documentElement.getElementsByTagName("marker");
           for (var i = 0; i < markers.length; i++) {
             var name = markers[i].getAttribute("name");
             var address = markers[i].getAttribute("address");
             var type = markers[i].getAttribute("type");
             var point = new
GLatLng(parseFloat(markers[i].getAttribute("lat")),

parseFloat(markers[i].getAttribute("lng")));
             var marker = createMarker(point, name, address, type);
             map.addOverlay(marker);
           }
         });
       }
     }

     function createMarker(point, name, address, type) {
       var marker = new GMarker(point, customIcons[type]);
       var html = "<b>" + name + "</b> <br/>" + address;
       GEvent.addListener(marker, 'click', function() {
         marker.openInfoWindowHtml(html);
       });
       return marker;
     }

function ValidateForm()
{

   if(!IsNumeric(gps.lat.value ))
   {
      alert('You have not entered a numeric number')
 document.gps.lat.focus();
      return false;
   }


   if (!IsNumeric(gpsPosition.logitude.value))
   {
      alert('Please enter only numbers or decimal points in the account
field')
    gpsPosition.logitude.focus();
      return false;
      }
```

```
return true;

}
function testResults (form) {
    var latitude =parseFloat(form.lat.value);
        var longitude = parseFloat(form.logi.value);
        var scale = parseInt(form.scale.value);

        load(latitude, longitude, scale);


}
//onload="load()" onunload="GUnload()
    //]]>
  </script>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">

        function lookup(inputString) { //Jquery code to read database
                if(inputString.length == 0) {
                        // Hide the suggestion box.
                        $('#suggestions').hide();
                } else {
                        $.post("rpc.php", {queryString: ""+inputString+""},
function(data){
                                if(data.length >0) {
                                        $('#suggestions').show();
                                        $('#autoSuggestionsList').html(data);
                                        //var lat =
$('#autoSuggestionsList').html(data);
                                        //var lat = data.latitude;
                                //var longi = data.longitude;
                                //$('input.#lati').val(data);
                                //$("input.#longit").val(longi);
                                }
                        });
                }
        } // lookup
        function fill(thisValue) {
                $('#inputString').val(thisValue);
                //setTimeout("$('#suggestions').hide();", 200);
        //$('#inputString2').val("thisValue');
        }

        $(function() {
   $("#autoSuggestionsList").click(function() {
        var place = $("#inputString").val();
        var count = place.split("-"); //this function works for southern
hemispere (Nairobi)
         var location = count[0];
          var cordinates = "-"+count[1];
          var xy = cordinates.split(" ");
        //alert("name= "+location+" cordinates = "+cordinates+" split =
"+xy.length );
         var latitude = xy[0];
```

```
        var longitude = xy[1];
         $('input.#longit').val(longitude);
         $("#inputString").val(location)
 $('input.#lati').val(latitude);

    setTimeout("$('#suggestions').hide();", 200);
   });
 });

</script>

<style type="text/css">
        body {
                font-family: Helvetica;
                font-size: 14px;
                color: #000;
        }

        h3 {
                margin: 0px;
                padding: 0px;
        }

        .suggestionsBox {
                position: relative;
                left: 30px;
                margin: 10px 0px 0px 0px;
                width: 200px;
                background-color: #FFFFFF;
                //background-color: #212427;
                -moz-border-radius: 7px;
                -webkit-border-radius: 7px;
                border: 2px solid #000;
                //color: #fff;
                        //color: #000;
        }

        .suggestionList {
                margin: 0px;
                padding: 0px;
        }

        .suggestionList li {

                margin: 0px 0px 3px 0px;
                padding: 3px;
                cursor: pointer;
        }

        .suggestionList li:hover {
                background-color: #659CD8;
        }
</style>
  </head>
```

```
  <body>
  <h1>Please enter the emergency/disaster location place name</h1>
 <table>
<FORM NAME="myform" ACTION="" METHOD="GET">

<tr><td> Latitude</td><td> <input type='text' name="lat" id="lati"
value=''readonly /></td></tr><BR>
<tr><td>Longitude </td><td><input type='text' name="logi" id="longit"
value='' readonly/></td></tr><BR>
<tr><td>Select the Map scale</td><td><select name="scale">
<option value="10" >Extra Large</option>
<option value="12" >very Large</option>
<option value="13" >Large</option>
<option value="14" >Medium</option>
<option value="15" >small</option>
<option value="18" >very small</option>

</select>
</td></tr>

<tr><td></td><td><INPUT TYPE="button" NAME="button" Value="Plot "
onClick="testResults(this.form)">
</td></tr>
<div>
                        <b>     Type the name of the disaster location:
                                <br />
                                <input type="text" size="30" value=""
id="inputString" onkeyup="lookup(this.value);" onblur="fill();" />
                        </div>

                        <div class="suggestionsBox" id="suggestions"
style="display: none;">
                                <img src="upArrow.png" style="position:
relative; top: -12px; left: 30px;" alt="upArrow" />
                                <div class="suggestionList"
id="autoSuggestionsList">
                                         
                                        </b>
                                </div>
                        </div>
                        </FORM>
</table>


    <div id="map" style="width: 1200px; height: 600px"></div>

        <div>


        </div>


  </body>
</html>
```