MOBILE ANTI-THEFT SYSTEM

A PROJECT REPORT

Submitted by

AFEEZ OLAWALE ISHOLA

(300621220507)

in partial fulfillment for the award of the degree

of

B.Sc IT (Bachelor of Science in Information Technology)

BlueCrest College, Ghana
(Affliated with University of Education Winneba)

June– 2014

DECLARATION

I hereby declare that the project work entitled "MOBILE ANTI-THEFT SYSTEM"

submitted to the BlueCrest College, is a record of an original work done by me under the guidance of "Mr. Peter Dekpor, Supervisor", and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Information Technology. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

AFEEZ OLAWALE ISHOLA

(30062122057)

Dedication

Acknowledgement

Although many people assisted me with this Project, the first praise goes to the Almighty Allah for seeing me through the course and the project in particular. Mr. Peter Dekpor deserves the second thank you, for her wonderful supervising and guidance given in respect to the project development, assisted in putting the project documentation and design into meaningful international academic standard, keeping me on target, and his countless hours. Thank you, Mr. Peter.

To my Project Coordinator Mr. Charles Saah, he also contributed some advice on how to meet the project target deadline and the expected output of the project. To all the Academic board, supervisors, lecturers, Student Registrar and the entire Academic staffs thank all for making the Institute a fruitful one for me.

Without a doubt, the best benefit to being Bluescrest Student is participating in the Final Project it kind of wonderful. Also, thanks the effort from a colleague at work that assisted in proof reading the content of this project. My Regards to my Brothers, Mr. Kazeem, Rasheed and Kudus who supported on my Academic stands.

To my friends and roommates, who have seen little of me for too many months; and to my Mum, for her incredible patience.

Table of content

List of figures

List of abbreviations

Items	Abbreviations	Meaning
1	MATS	Mobile Anti-Theft System
2	SIM	Subscriber Identity Module
3	GPS	Global Positioning System
4	SMS	Short Message Service
5	GSM	Global System for Mobile Communication
6	GPRS	General Packet Radio Service
7	SD	Small Device
8	IDE	Integrated Development Environment
9	SQL	Structure Query Language
10	OS	Operating System
11	IMSI	International Mobile Subscriber Identity
12	IMEI	International Mobile Equipment Identity

Abstract

Mobile Anti-Theft system is an application based on Android used for tracing back stolen or lost mobile, once my application is installed onto a mobile phone and an alternate number is fed into the software (MATS).

The owners SIM card gets registered in the database, whenever phone is rebooted MATS is invoked in stealth mode and verifies whether if the SIM card present in mobile phone is of owner.

If the SIM belongs to owner (registered in database), the software doesn't do any activity, If SIM is been changed and that SIM is not registered in the database then, MATS sends a message to the alternative mobile number (friends / relatives number which is been saved while installing the application) in stealth mode and starts listening for incoming SMS messages.

Now if owner send a SMS request to MATS asking for GPS co-ordinates, MATS would do so, Since My application is based on GOOGLE Android operating system our system would send the complete address (postal address) as to a where the mobile is.

1.1 Introduction about the Project

Today, we live in a world of virtual reality. It makes sense to say that none of us can live without our computers and mobiles and other gadgets.

With mobile devices becoming more and more essential to survive with, people have started storing more and more confidential data and information on their mobile devices.

In most cases, it is the information that happens to be more important than the computer or mobile itself.

So, what happens if your mobile or laptop gets stolen or is lost? Wouldn't it be terrible to know that an unauthorized user has complete access to your information? Taking all of this in mind, Mobile Anti-Theft system is developed.

While some stolen mobile phones will wind up in pawn shops, sold on the street corners or in online classifieds, the vast majority of stolen devices are now being shipped internationally.

International markets (particularly countries in Africa, Latin America, and Asia pacific) pay a premium for Smartphone's, with the high demand pushing the prices way up made nearly $4 million in 8 months as middleman selling phones to Hong Kong, where they can fetch as much as $2000 per phone. (Arieanna Schwebber).

Even in Nigeria and Ghana most Mobile users are alarmed at the rate of GSM phone theft in the country, the experience pales into nothing when compared with happens in South Africa.

South Africa easily qualifies as the headquarters of telephone theft on the African continent.

In the average, Vodacom and MTN receives about 10,000 phone theft a month, no thanks to the activities of GSM phone thieves in down town Johannesburg, Lagos and Accra who make life difficult for cell phone subscribers. [1]

MATS will consist of a number of essential anti-theft features that many other devices do not contain.

It will use android built in feature to retrieve the information about the change of SIM on your mobile phone incase its get lost or stolen; this project will also help in tracking down your stolen mobile with GPS and GPRS technology built in most of the smart phones to identify the location of your mobile phone.

1.2 Problem Statement

In Africa as well as in many parts of world there are no mechanisms in place which can help the owner of the mobile recover his mobile phone.

Mobile Anti-Theft System is a project which helps us to track the location of the mobiles, when the mobile user loses his mobile phone.

Herein I propose to develop a system which is autonomous and intimates with the owner via SMS when it detects SIM change.

1.3 Background of Project

Currently, Africa now has more than 650 million mobile phone subscribers: that's more than either the United States or the European Union. And it's a market that has seen explosive growth.

Since 2000, the mobile phone market has grown 40-fold, from 16.5 million, according to the World Bank Ref link: [2].

It is now making sense to say that none of us can live without our computers, mobile phones and other gadgets.

In most Africa countries, mobile devices will become more and more essential to survive with, people have started storing more and more confidential data and information on their mobile devices.

In most cases, it is the information that happens to be more important than the computer or mobile phone itself.

So, what happens if your mobile or laptop gets stolen or is lost? Wouldn't it be terrible to know that an unauthorized user has complete access to your information? Taking all of this in mind, the Mobile Anti-Theft system is developed.

1.4 Project Scope

1. The Mobile Anti-Theft System will be installed on only mobile phones with Google android operating system.

2. Further on SIM change SMS can be received only if the thief's SIM has enough currency in it.

3. To get the location of the mobile phone it needs to have GPS built in it. To get the postal address the mobile should have GPS as well as GPRS and be connected to the internet.

1.4.1 Further improvement of the system can perform the following functionalities.

1. Functionality of the Mobile Anti-Theft system can be extended so that when requested it sends the recent call log through which the owner can contact the recently dialed numbers of thief and get to know more about the thief.

2. I proposed to add functionality later wherein the owner of the mobile can retrieve his data stored on the mobile via internet and can add or delete his personal data on the phone or even lock the phone.

3. Also, I will add functionality for taking the snapshot of the possessor and send it over to the owner.

4. In addition I proposed to make the application accessible in other mobile platforms like iPhone, Symbian, Blackberry and Windows Mobile phones.

1.5 Architecture

Below diagram depict the architecture design of the system, it consist of the mobile devices, Tower and the Internet. The communication is depicted with the arrow showing the path for which the first device communicate with the other devices using the Tower as a gateway to access the provider network and the internet for communicating with both devices.

The first Cell phone sends request through the Tower to the seconds cell phone and the second cell reply through the tower to the requestors, in case any of the cell phone need to communication or get information from the internet the ISP tower tunnel the request to the internet and service the request appropriately.

1.1 diagram below show the proposed system architecture

Cell phone

Internet

Cell phone
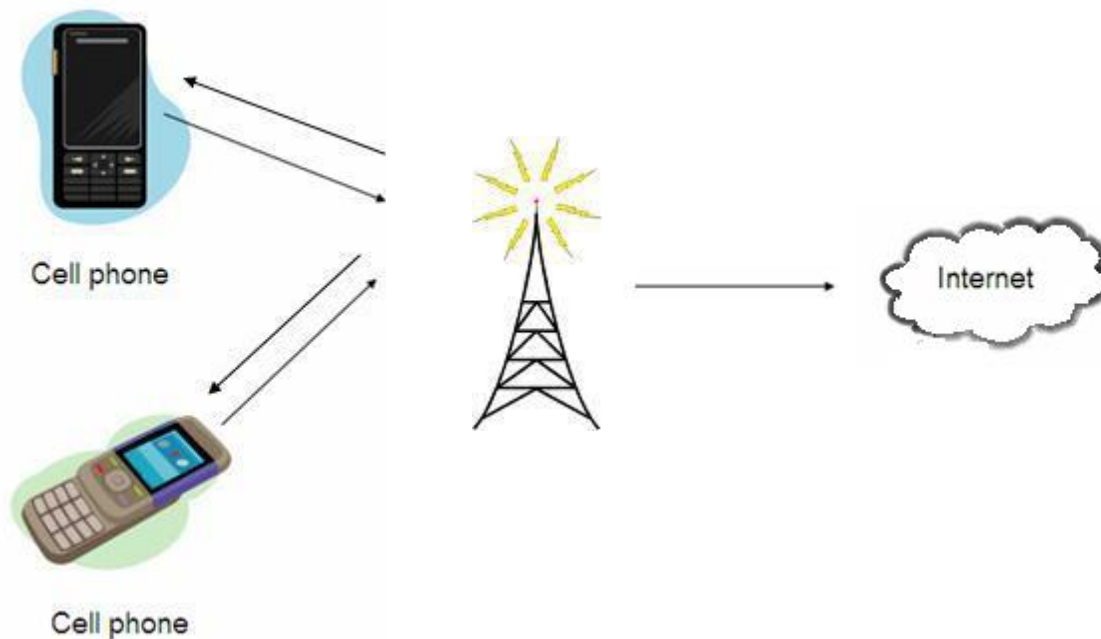
1.6 Significance

Almost every owner of a costly mobile handset fears the nightmare of losing his mobile phone.

I have come across many middle class people losing costly mobiles and unable to get back the same even with the help of police officers and in AFRICA there is no system in place to help such people.

Seeing their plight and the upcoming android platform which promises to be present on ever upcoming handset, this is why I proposed to design the application.

1.7 Chapter Organization

In this report I have briefly described about the introduction of our project, how it works, its limitations, future enhancements.

I have also explained about system problem statement and description in which I have specified the issue and the impact of the mobile theft.

I have also explained the project background that is the needs for implementing the project.

Also, the project scope, objective activities and deliverable were also been elaborated for how the project will work.

The system architecture was also been explained which shows the diagrammatic pictorial diagram of how the system will interact.

I have taken time to explain some literature review of the objective, activities and deliverables on this project.
In addition the requirement analysis and system implementation design has also been documented with accurate references.

Furthermore, The System code Implementation has been articulated and explained in detailed,

Lastly, I have documented and explained how the end user will interact with the system

2.1 Historical Background about Android Technology

Android is a new and very user friendly operating system for mobile devices which includes key applications, middleware and even uses Linux Kernel modified version. (Android, 2013)

Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

The first Android-powered phone was sold in October 2008 Ref link: [3] Android is open source and Google releases the code under the Apache License.

This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers.

Additionally, Android has a large community of developers writing applications ("apps") that extend the functionality of devices, written primarily in a customized version of the JAVA programming language.

In October 2012, there were approximately 700,000 apps available for Android, and the estimated number of applications downloaded from Google Play, Android's primary app store, was 25 billion.

A developer survey conducted in April–May 2013 found that Android is the most popular platform for developers, used by 71% of the mobile developer population. [4]

These factors have contributed towards making Android the world's most widely used smartphone platform,[5] overtaking Symbian in the fourth quarter of 2010,[6] and the software of choice for technology companies who require a low-cost, customizable, lightweight operating system for high tech devices without developing one from scratch.[7]

As a result, despite being primarily designed for phones and tablets, it has seen additional applications on televisions, games consoles, digital cameras and other electronics.

Android's open nature has further encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users [8] or bring Android to devices which were officially, released running other operating systems.

Android's share of the global smartphone market, led by Samsung products, was 64% in March 2013. [9] The operating system's success has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies. [10][11]

As of May 2013, 48 billion apps have been installed from the Google Play store, [12] [13] and as of September 3, 2013, 1 billion Android devices have been activated.

2.2 Major Android Technology Application Components

Application components are the essential building blocks of an Android application, each component is a different point through which the system can enter your application.

Not all components are actual entry points for the user and some depend on each other, but each one exists as its own entity and plays a specific role—each one is a unique building block that helps define your application's overall behavior.

There are four different types of application components, each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed.

Here are the five (5) types of application components:

2.2.1 Activities

An activity represents a single screen with a user interface, for example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails.

Although the activities work together to form a cohesive user experience in the email application, each one is independent of the others. As such, a different application can start any one of these activities (if the email application allows it).

For example, a camera application can start the activity in the email application that composes new mail, in order for the user to share a picture.

An activity is implemented as a subclass of Activity and you can learn more about it in the Activities developer guide in this project.

2.2.2 Content Provider:

A content provider manages a shared set of application data. You can store the data in the file system, a SQLite database, on the web, or any other persistent storage location which your application can access.

Through the content provider, other applications can query or even modify the data (if the content provider allows it).

For example, the Android system provides a content provider that manages the user's contact information.

As such, any application with the proper permissions can query part of the content provider (such as Contacts Contract. Data) to read and write information about a particular person.

Content providers are also useful for reading and writing data that is private to your application and not shared.
A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions. For more information, see the Content Providers developer guide on http://developer.google.com

2.2.3 Broadcast Receiver:

A broadcast receiver is a component that responds to system-wide broadcast announcements.

Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured.

Applications can also initiate broadcasts—for example, to let other applications know that some data has been downloaded to the device and is available for them to use.

Although broadcast receivers don't display a user interface, they may create a status bar notification to alert the user when a broadcast event occurs.

More commonly, though, a broadcast receiver is just a "gateway" to other components and is intended to do a very minimal amount of work.

For instance, it might initiate a service to perform some work based on the event.

A broadcast receiver is implemented as a subclass of Broadcast Receiver and each broadcast is delivered as an Intent object. For more information, see the Broadcast Receiver class on the Implementation chapter in this project.

2.2.4 Intent:

System messages which run inside the device, various applications notification such as hardware changes like SD card inserted, notifications of incoming data like SMS arrived and even application events are called as Intents.

It doesn't only allow you to respond to such intents but also to initiate other activities or let know when particular event occurs such as suggest WIFI availability when in range.

2.2.5 Services:

A service is a component that runs in the background to perform long-running operations or to perform work for remote processes.

A service does not provide a user interface. For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity.

Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it.

A service is implemented as a subclass of Service and you can learn more about it in the Services developer implementation on this project.

2.3 Services Available under Android Technology

2.3.1 Network

Android based devices are generally with Internet ready. I can take benefit of internet as I wish in any level from raw java sockets to built-in Web browser which is based on Webkit.

2.3.2 GPS

Most of the android devices have access to GPS which can tell where the device is exactly located on the earth using Google Maps.

GPS also helps in locating the desired location where I want to travel and even shows the places around us where I commonly go in everyday life which makes it easy to travel otherwise can be to locate the device and its movements in case the device is stolen

2.3.3 Phone Services

Android devices are similar to other phones which are typically used to make calls, send SMS and can be used for multimedia applications to download music anytime anywhere and games and everything else what I expect from a modern telephonic technology.

2.4 Why I Chose Android Platform and Technology over Others

1. *Zero start-up cost to begin development with*

The tool required to develop any android applications is free of cost and Google charge very small fee deal out application in the market. [15]

2. *Freedom to innovate*

Android OS is an open source platform which is based on Linux kernel and other open source libraries. Moreover are free to build applications which runs on android devices and even free to extend platform as well. (Grell, 2010)

3. *Freedom to collaborate*

Android developers are encouraged to share code with others and they don't even have to sign an NDA to do this. According to a survey conducted by Black Duck Software, the amount of open source libraries and mobile applications grew at a rate of 168% faster on Android compared to any other platform from year 2008-2009. (Grell, 2010)

4. *Multi-platform support*

Android OS are supported on several different hardware devices including various phones and tablets.

5. *Multi-carrier support*
Android powered smart phones are offered by most of the carrier services.

2.5 Tracking Record on the Black Market constitute in Mobile Theft

An insatiable appetite for smartphones has turned the black market into a global enterprise, efficiently sending ill-gotten gadgets wherever demand is greatest.

But no one has a complete picture of the size or scope of the black market. One can only catch it in glimpses.

In a report for NYPD, Timlin found stolen phones changing hand all over the city. "We saw bodegas, we saw local Laundromats, and we saw back-alley sales," he says. In March, the California attorney general announced the arrests of two individuals who allegedly paid homeless people to buy discounted phones on a two-year contract, and then shipped the devices in bulk to Hong Kong.

There, phones can sell for $2,000 each — 10 times as much as in the states. The accused allegedly took in almost $4 million in less than a year.

I hate the guys who do this type of stuff," says Marc Rogers of the online security firm Lookout. He is a hacker who frequents forums where information on the black market for cellphones is exchanged.

He says that in the global game of cat and mouse, the mouse is usually faster. For example, some European authorities created blacklists, where users could report stolen phones and block them from being used again on other networks.

But Rogers says criminals quickly realized that by shipping devices to foreign countries, they could sidestep the blacklists and probably sell for close to retail price.

Law enforcement tends to focus on thefts on the street and in subways. But Rogers believes police will only make progress when the black market itself is squeezed.

He says the security features in Apple's new operating system, like fingerprint ID and the requirement that you enter a password before resetting the phone, are a good start.

"Ultimately, it would be fantastic if I could get it set up so once a device is stolen, the only value there is from the parts," he says.

New York police will be on high alert when Apple's new iPhone goes on sale Friday. Since the first iPhone debuted six years ago, they've noticed that every new Apple product comes with a spike in street crime.[16]

3.1 Aims of the Project

Mobile Anti-Theft system is a project which helps us to track the location of the smart phones. It consists of Android client application which will automatically send SMS when SIM card is changed. Position tracker works on GPS (Global Positioning System) and GPRS.

3.2 OBJECTIVE

When requested MATS will fetch latitude and longitude satellites and send it as a SMS, moreover if mobile is connected to the internet it retrieves the postal address from Google maps.

To research on the issues faced with mobile theft, the growth rate of mobile phone usage as compare to PCs and Laptops, Physical and data Security issue with Mobile.

3.2.1   Activities on the theft Research:

1. Investigate the trend of World black market activities relating to mobile phone Document the number of stolen mobile phones that wind up in pawn shops, sold on the street corners or in online classifieds.

2. Investigate the number of stolen mobile devices that are shipped to international markets (particularly countries in Africa, Latin America, and Asia pacific).

3. Document the trend of Anti-theft mobile response activities in Africa.

4. Evaluate the use of mobile SMS as an integral part of mobile devices and mobile applications.

3.2.2   Deliverables on the theft Research:

1. Data showing the trend of mobile phone theft and re-sale on the world black market

2. Data map showing the number of stolen mobile devices that appear on international markets and obtain the Information on Anti-Theft Mobile responses in Africa

3. Documentation on the use of mobile SMS by mobile device users

3.3 OBJECTIVE

To build an Android client application which will automatically send SMS when SIM card is changed and also when user needs to know the location of the lost or stolen mobile phone.

3.3.1 Activities for automatic SMS Sender:

1. In this project I am going to examine how SMS technology via the APIs provided by Android can help us in track the location of the stolen or lost phone.

2. This project will get information from the user on installation; this information will be used to capture all the information about the phone; like the IMEI which is unique per phones.

3. Get the user relative phone number for receiving SMS in case the phone get lost.

4. SMS function which will send automatic SMS text message to an alternate phone number immediately when SIM is swapped on the registered for phone.

5. This app will take the relative phone number at first registration on the platform and store in a database.

6. When the user phone get lost or stolen, then the application will launch the SMS automatic process from the lost or stolen mobile phones with the number the user entered during the registration.

7. Finally it will send a "report of a stolen or missing phone" message to the alternate phone.

3.3.2 Deliverables for automatic SMS Sender:

1. Users registered with unique identification on the application.
2. SMS messaging services in the MATS for sending and receiving SMS in a stealthy mode incorporated into application.
3. The Application is able to sends Automatic SMS as soon as SIM swap happen the lost or stolen phone in a stealthy mode.

3.4 OBJECTIVE

To design an application by means of which I can track the location of the mobile phone of the desired user when it been stolen or lost.

3.4.1 Activities for locating the phone GPS / Postal Address:

1. To track the location of the smart phones.

2. The project will consist of Android client application which will send an SMS when user requested for the phone location.

3. I am going to use position tracker functionality which works on GPS (Global Positioning System) and GPRS on any smart phones. This will be used to fetch the latitude and longitude satellites of the location of the mobile

4. Send the location information and via SMS to the requestor phone from the stolen phone

5. Moreover if mobile is connected to the internet it will retrieves the postal address from Google maps. (ZHANG Hao, 2011) and send the retrieved location address to Relative number.

3.4.2 Deliverables for locating the Phone GPS / Postal Address:

1. Use goggle or yahoo position tracker to get the location of the stolen or lost phone

2. Get the location through Google map in case the stolen or lost phone has internet connections.

3. The application retrieves the location of the lost or stolen

4.1 Introduction on Requirement Analysis

The Requirement Analysis Documentation is nothing but a method of communicating what the whole project is about.

Objective, realistic and complete are the important factors to kept in mind while writing satisfactory documentation.

The whole process of implementation, operation of system and development is based on a proper documentation.

Therefore, the whole idea behind this project requirement analysis document is to represent the bounded physical representation of the body of information what I am trying to design will have the capacity to communicate the intended design of the project.

This document produces an artifact by collecting and representing information.

This document helps my supervisor to understand about the project in a lucid manner.

Its gives an overview of the existing system and how the proposed system will leverage on the existing system and also add its own functionality which is not available on the existing system.

In addition, it's contains the technology used and the hardware and software use in developing the application.

Finally, its justify the feasibility of designing the project as related to the economical , technology, behavioral aspect of the entire project, in conclusion it contain the requirements specification document which will be used in designing the prototype of the implementation of the project.

4.2 Existing System

There are lot of Mobile Anti-Theft software available in the market today, and all comes with their targeted functionality and platform, most effective mobile anti-Theft software has the basic functions and objective which is to check the position of a mobile phone, but majority of the software lack the main concept of the Anti-Theft system and the feasibility of detecting theft using the software.

Therefore, the proposed Anti-Theft Software will cover all the major flaws that those software was unable to capture. Example, the proposed application can be used has a mobile tracker.

4.3 Proposed System

MATS would be widely used as people are concerned about their mobile phones.

Moreover the software would be designed to use very less memory and the package can be downloaded over internet via Google play store.

This system can be used by concerned parents to track their children at any moment of time without their notice. Employers can track employees as well.

After owner of a mobile phone install the Mobile Anti-Theft System on his mobile phone, and the application is put to start by filling all the necessary information like his alternate number, his username and password which will be used to check the state of the phone and update his or her information and commit the information into the application database.

When he loses his mobile phone, he waits for SMS from the Mobile Anti-Theft Software.

When he / she receives that SMS, then he starts corresponding with the Mobile Anti-Theft System via requesting or postal address etc, relative of the owner can request for the location of the mobile phone either the mobile phone is lost or not, he acts as the master to the Mobile Anti-Theft System; supposing a scenario wherein the thief hasn't change the SIM, that allow the relative to request for the mobile location.

If in a normal scenario where by Thief changes the SIM of the Mobile phone and reboots the mobile phone on rebooting the mobile the Mobile Anti-Theft system will automatically triggered and intimation will be sent to the alternate phone number.

I belief the Mobile Anti-Theft System would be widely used as people are now concerned about their mobile phones.

Moreover I will make the software to be downloaded over the internet with less phone memory usage.

4.4 Feasibility

MATS would be widely used as people are concerned about their mobile phones, as I have stated on my investigation on the mobile theft issues faced all over the world, that all people care for is the data and not the phone itself.

Since the technology is an open source which is free of cost, therefore, I wouldn't need to invest money to develop the application, in addition, with my several years of experience with developing application under java platform the backbone for this development is mainly on java. Therefore, the software would be designed to use very less memory and the package can be downloaded over internet via google play store, and most importantly be intuitive enough to interact with.

4.5 Technology

ECLIPSE IDE

Eclipse is an Integrated Development Environment (IDE) and an extensible plug-in system which is primarily written in Java and Java applications can be created. The Eclipse IDE serves our development environment for designing and testing the application.

JAVA

The Primary activities like sensing the phone turn off and on, SIM changed, Available credit on the newly inserted SIM and send message to alternate phone is done with Java Technology under android SDK

XML

XML will be used for representing subjective data structures for communicating between the lost phone and the alternate phone by requesting and responding to services request.

GOOGLE MAPS API

offers services like street maps, navigation planner for travelling from source to destination by foot, car or any public transport and even helps in locating shops, restaurants , cinema and many other across the world this service will be used to locate the lost or stolen phone postal address.

SQLLITE DATABASE

SQLite

It will be used to capture information needed to recover the lost or stolen phone, please its act has the data store for request and respond to information require. And it directly embedded into the android phones.

4.5.1 Hardware

The hardware requirement for the MATS is an important part of the MATS standard.

The requirement is stated below

1. Any android mobile phones with ARM processor, GPS and GPRS support.

4.5.2 Software

Though the java cross-platform nature of java helps alleviate concerns over the wide range of mobile device operating systems. Even, so the Android specification lays some ground rules about what is expected of the operating system in android devices. Therefore, for are the major software specifications for the proposed system (MATS).

1. Minimum Version Google Android OS 1.5

2. A mechanism to read from and write to non-volatile(persistent)memory

3. A timing mechanism for establishing timers and adding time stamps to persistent data.

4. SQLlite database

These requirements, although are somewhat minimal, still provide a reasonably rich set of features that are available for use by MATS application.

5.1 Introduction to system design analysis

Design documentation is used to design the prototype of out the final document will look, therefore, in this chapter I have documented all the preliminary designs which comprises of all high level design of the system.

These are Database, Communication and Interface modules, also the low level core design which is of course the low level design of the system, in the low level design have took time to explain the all the functional and nonfunctional aspect of the system and how activity related to one another, and how the data flows between one another to another, the designs includes; Use Case diagram, Data flow diagram, and Entity Relational Diagram.

5.2   Preliminary Design

The Mobile Anti-Theft project contains 3 modules.

5.2.1 Database Module

This module is used to fetch the real data from user and store into database using SQLLite. In this module first user interface where user provides SIM IMSI number and alternate number then click submit button. Next it will store information in the database. The users can fetch the information from database to bind user interface control.

- User can enter name, alternate number and IMSI number.

- User can change the alternate number and IMSI number.

5.2.2 Communication Module

This application needs to be installed on mobile. The software helps us to obtain the current location of the phone using GPS facility and retrieve postal address with GOOGLE Maps using GPRS facility.

5.2.3 Core Module

We are using an Android OS service which will be acting as a communicator between device app and SQLLITE Server database in Phone.

This service will receive information as data and store in the database. This service starts automatically in stealth mode and check the SIM IMSI Number with the database.

If SIM IMSI Number does not match with the database, it sends a SMS to alternate number stating that SIM has changed else reboot normally. It always checks the SMS whether SMS contains query string that start with MATS or not?

If SMS contains query string then it will respond with the result in stealth mode (means thief does not know about the message).

5.3 Design Considerations

5.3.1 Input

- IMSI (International Mobile Subscriber Identity) number of the SIM.

- Alternate number of the Owner.

- Name of the Owner.

- Pre-defined text that will send to alternate number when SIM will change.

- SMS that contain query string.

5.3.2 Output

- Text message stating that SIM has changed.

- GPS Location of phone.

- Postal Address of the phone.

5.4 Use Case Diagrams

Use case diagram here shows the roles of all the people involved in this project. Here in our project we consider 3 cases they are owner, relative and thief.

5.4.1Below diagram show the Use case of the proposed system
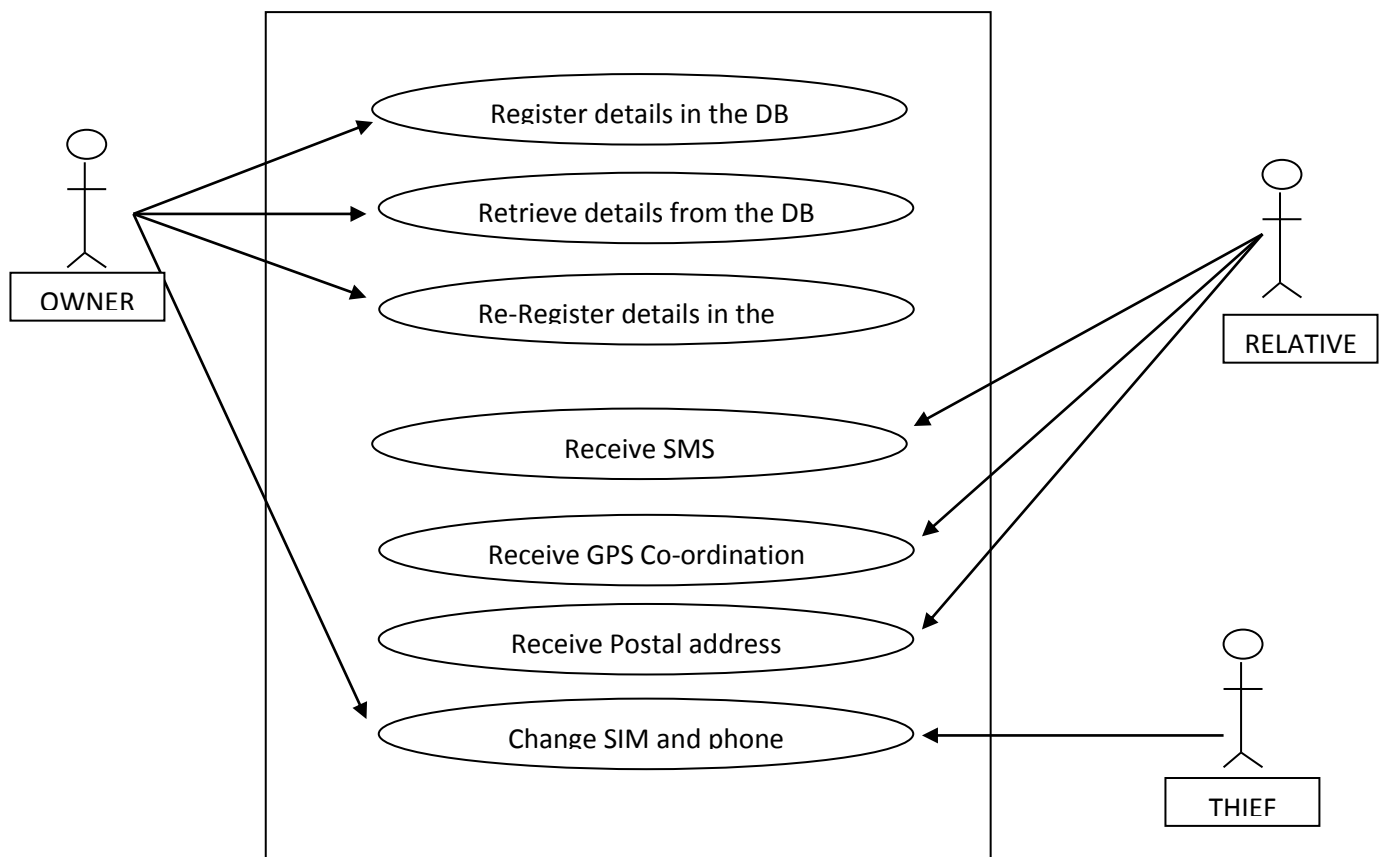
From the above use case diagram we know that the owner can register the details in the database when software is installed and at any point can retrieve the details and if necessary can change it accordingly and relative receives SMS, GPS co-ordinates and postal address whenever the change of SIM is done and thief doesn't play a role in this but he is related with changing SIM and rebooting as soon it's done, relative of the owner will get the details accordingly.

5.5 Data Flow Diagram

5.5.1 Foreground

The foreground includes the user interface of the application. Whenever the user opens the application it ask for login and if it's a new user then ask for user account creation. After the user has created his or her account, user may never open the application again, if require user can update his or her record this module is only accessible by the user. The application will all run at boot up without the knowledge of the user.

1.2 Diagram below is the foreground interface of the proposed system
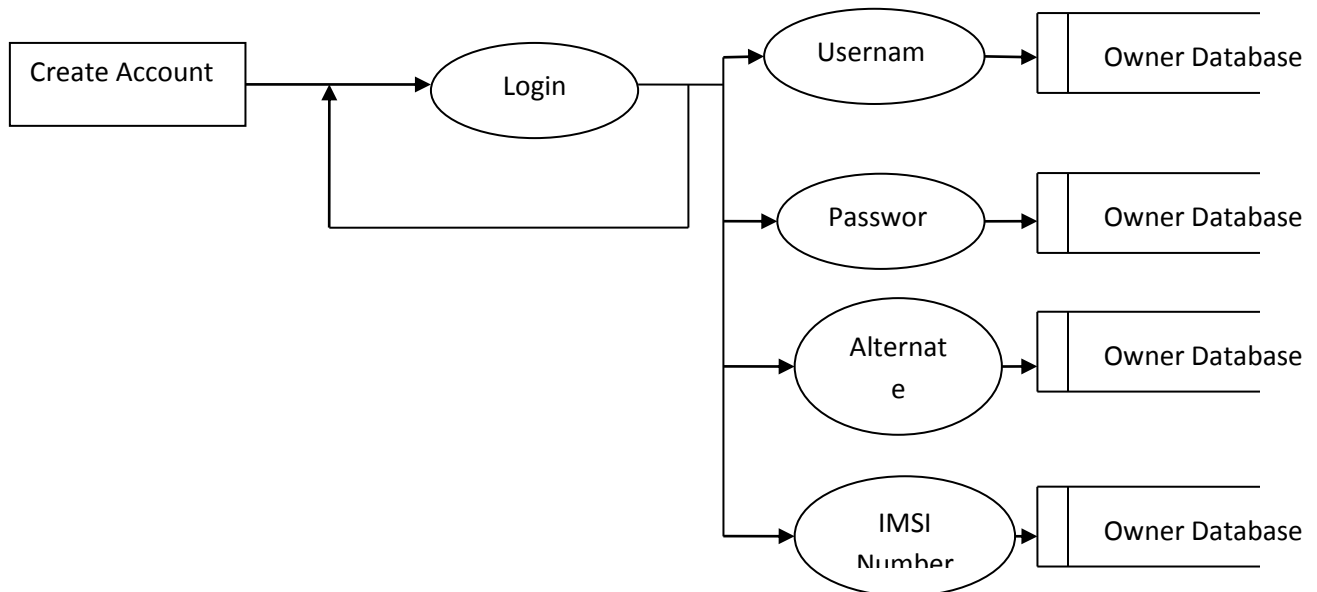
5.5.2 Background

The actual process running on the background is not known to the user. He only enters the required information when registering and saved to the application. Whenever the phone bootup the application will all check the mobile IMSI and if it's same as the stored data then it doesn't start any events. In a case if the IMSI stored is not same as current IMSI on the phone, the application will act and perform some events which will occur without the knowledge or without any hint, then sends SMS to the alternative phone number stored by the owner.

1.3 Diagram below is the background services of the proposed system
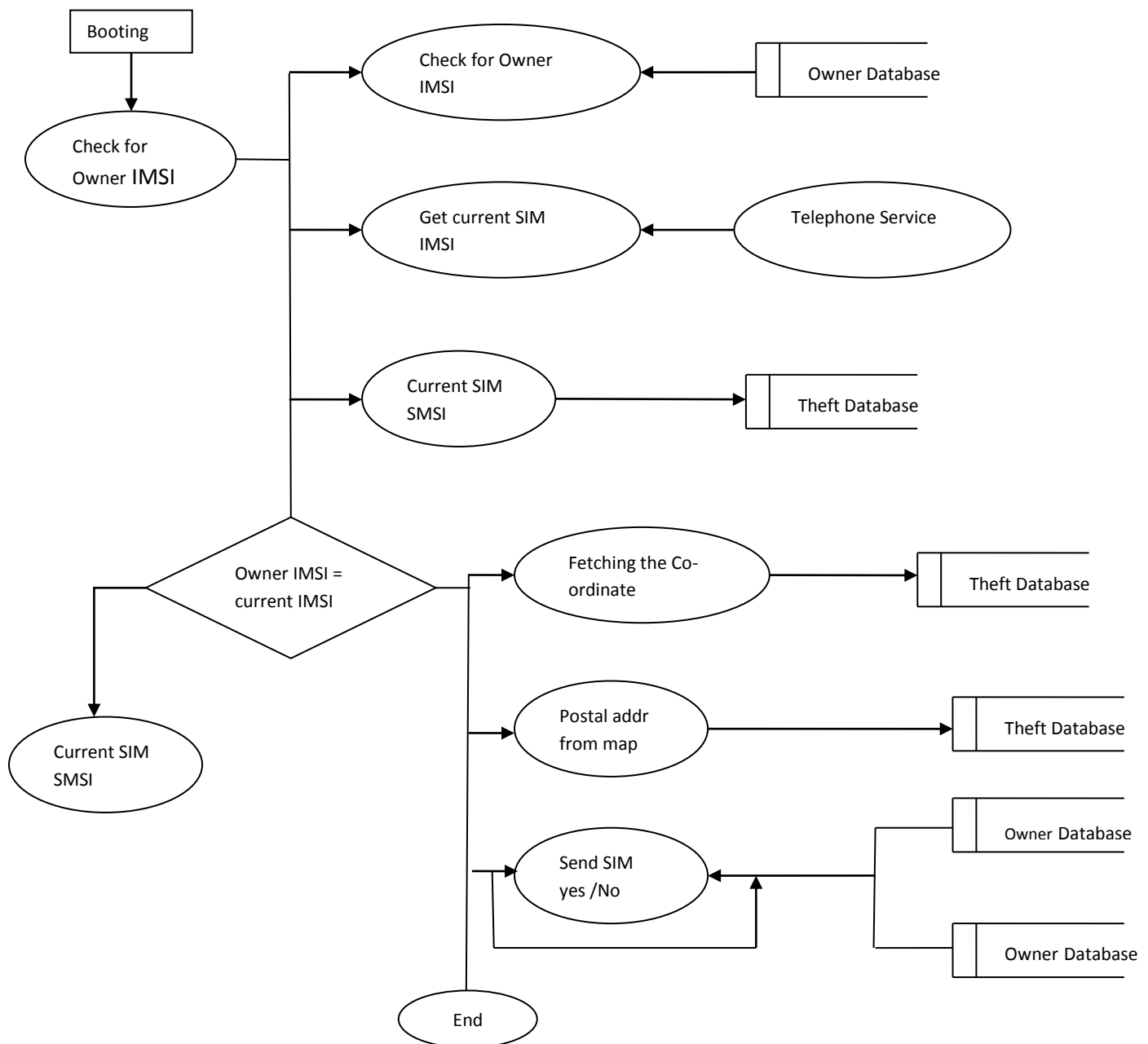
5.6 Entity Relationship Diagrams

Entity Relationship is used to show the relation among entities in a program, it also shows the strength of their relation, be it strong or wick relation. Below diagram show the entity relationship for the proposed application, Note that entity could exist from internal or external object.

1.4 Diagram below is the entity relational diagram of the proposed system
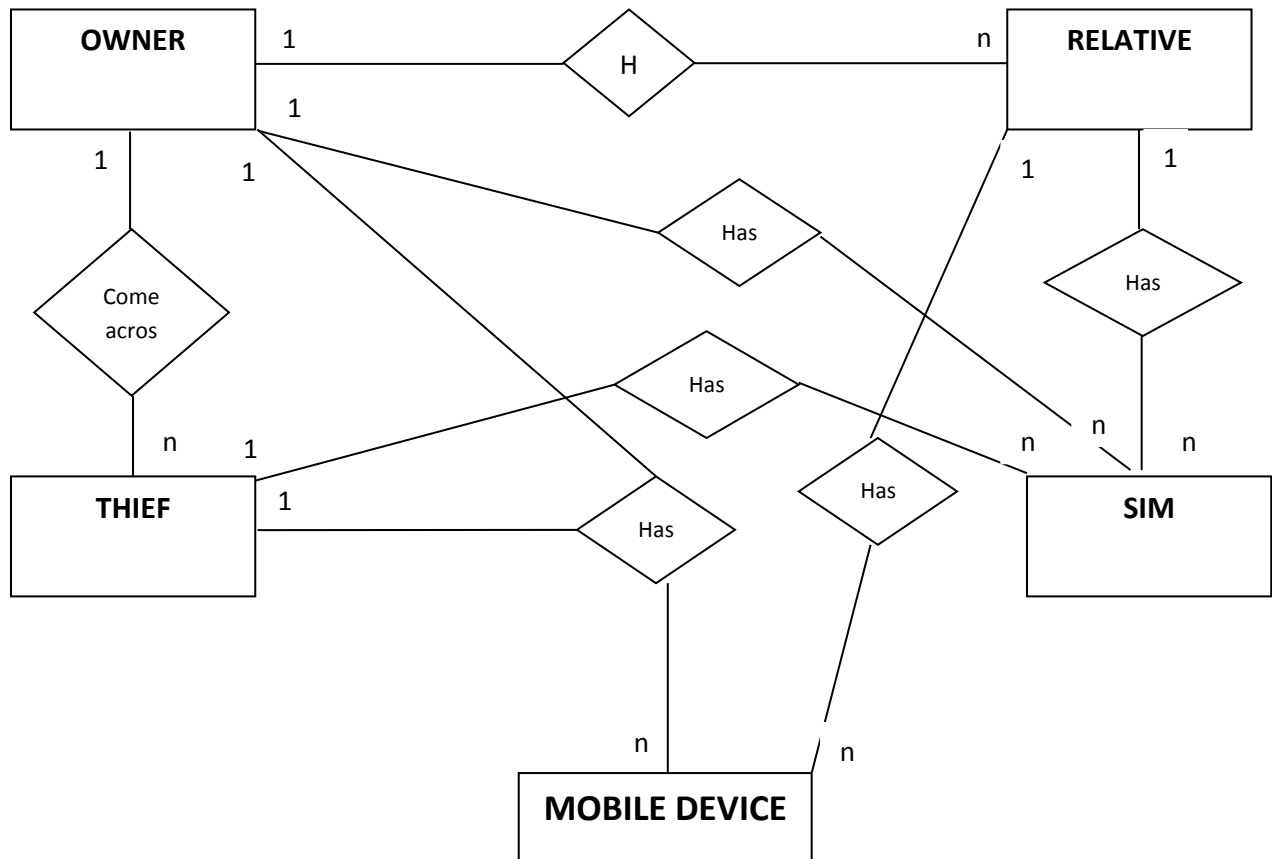
6.1 Introduction to Software code Implementation and Testing

Implementation means to carry out or to put into effect. The implementation phase of the system of any software is to obtain source code from the design specifications.

Implementation becomes necessary to obtain source code and related internal documentation required to understand the system and be demonstrated easily.

Documentation and code is written in a way so that testing, debugging and modification become easy. Post-implementation is nothing but an evaluation in which we can verify that objectives of the project is met completely and to check if actual cost of the project does not go beyond initial evaluation.

It can be even stated as review about the problems which need to be converted for the success of the project. Once implementation and conversion is done, review is conducted to check whether the system meets all the expectations and is changed if any improvements are required.

Post implementation also measures performance of the system against pre-determined necessities and also checks if any modification or re-design necessary.

6.2 Database module

Here in this module we develop a database using sqllite database which has one relation to store name, password, IMSI, relatives' phone number. We connect to sqllite database through java and commit values into database. Again when the phone is rebooted this database is opened and queried for IMSI number if it matches then we close it else we fetch relatives phone number and send an SMS to that number.

6.2.1 LoginDataBaseAdapter.java class

1.5 Diagram below is the source code for the login page

```java
 8  /**
 9   * @author AbdHafeez
10   *
11   */
12  public class LoginDataBaseAdapter
13  {
14          static final String DATABASE_NAME = "login.db";
15          static final int DATABASE_VERSION = 1;
16          public static final int NAME_COLUMN = 1;
17          static final String DATABASE_CREATE = "create table "+"LOGIN"+
18                          "( " +"ID"+" integer primary key autoincrement,"+ "USERNAME  text,PASSWORD text, "
19                                  + "PHONE text, MESSAGE text, IMEI text, IMSI text, HDID text,SERIALNO text, SERIALNO2 text, ANDROIDID text); ";
20          static final String CREATE_MSGTABLE = "create table "+"INCOMINGMSG"+
21                          "( " +"ID"+" integer primary key autoincrement,"+ "longitude text, latitude text); ";
22
23          public  SQLiteDatabase db;
24          private final Context context;
25          private DataBaseHelper dbHelper;
26
27          public  LoginDataBaseAdapter(Context _context)
28          {
29              context = _context;
30              dbHelper = new DataBaseHelper(context, DATABASE_NAME, null, DATABASE_VERSION);
31
32          }
33          public  LoginDataBaseAdapter open() throws SQLException
34          {
35              db = dbHelper.getWritableDatabase();
36              return this;
37          }
38          public void close()
39          {
40              db.close();
41          }
```

```java
75    public void insertEntry(String userName, String password, String phone, String msg, String imei, String imsi,
76            String hdID,String serialNo, String serialNo2, String androidId)
77    {
78        ContentValues newValues = new ContentValues();
79        // Assign values for each row.
80        newValues.put("USERNAME", userName);
81        newValues.put("PASSWORD",password);
82        newValues.put("PHONE", phone);
83        newValues.put("MESSAGE", msg);
84        newValues.put("IMEI", imei);
85        newValues.put("IMSI", imsi);
86        newValues.put("HDID", hdID);
87        newValues.put("SERIALNO", serialNo);
88        newValues.put("SERIALNO2", serialNo2);
89        newValues.put("ANDROIDID", androidId);
90
91        // Insert the row into your table
92        db.insert("LOGIN", null, newValues);
93        ///Toast.makeText(context, "Reminder Is Successfully Saved", Toast.LENGTH_LONG).show();
94    }
95
96    public String getPhoneNum() {
97        String Table_Name="LOGIN";
98
99        String selectQuery = "SELECT  * FROM " + Table_Name;
00        Cursor cursor = db.rawQuery(selectQuery, null);
01        if(cursor.getCount()<1) // UserName Not Exist
02        {
03            cursor.close();
04            return "NOT EXIST";
05        }
06        cursor.moveToFirst();
07        String data = cursor.getString(cursor.getColumnIndex("PHONE"));
08        cursor.close();
09        return data;
10    }
```

The above code is used to create a database, connect and disconnect from the MATS database, the declaration on line 12 is used to declare the name of the class for the MATS database as Login Database Adapter.

The line 14 is used to declare the name of the Database as 'login.db', line 15 declare the sqllite database version, the 16 declare the column name, also, line 17 is used to create the database table as LOGIN, note that this method is the same as creating table in normal sql query.

The line 27 is used create a constructor for the database class, the purpose is load all the property such as the application context, the database name, and the version of the database. of the database as when called.

The method in line 33 is used to establish connection to the database using open() method, inside this method we have the getwritabledatabase() method is called in it, which is used to open the database for read and write permission. The method also throws any known exception in case the DB failed to connect.

Line 37 is used to disconnect or close the connectivity to the database after use, this is necessary because of security issue.

The insertEntry() method in 75 is used to insert the obtained records such, Phone IMEI, SIM IMSI, username, Phone serial number e.t.c, and get stored in the LOGIN table. Please note that some information are such as username, password, relative phone number are obtained directly from the user while, the IMSI and IMEI are auto generated using System inbuilt functions.

The method getPhoneNum() in line 96 to 110 is used to query the database table and retrieve the relative phone number.

6.3 Core Module Implementation

This module is related to user interface. Getting user input, validating it and then connect to database module to commit the same if already registered user has to enter valid password to gain access to interface.

6.3.1 MainActivity.java Class

1.6 Diagram below is the source for the home screen activity

```
46 /**
47  * @author AbdHafeez
48  *
49  */
50 public class MainActivity extends FragmentActivity implements
51         LocationListener,
52         GooglePlayServicesClient.ConnectionCallbacks,
53         GooglePlayServicesClient.OnConnectionFailedListener {
54
```

The code above is used to define the MATS Home Screen, its extends Fragment Activity for designing the interface, and implements LocationListener, GooglePlayServicesClient.ConnectionCallbacks, GooglePlayServicesClient.OnConnectionFailedListener Interface API for getting the location from google play.

6.3.2 The onCreate(Bundle saveInstance) method

1.7 The Screenshot below is the source code for onCreate(Bundle saveInstance) method

```
85    @Override
86    protected void onCreate(Bundle savedInstanceState) {
87        super.onCreate(savedInstanceState);
88        setContentView(R.layout.activity_main);
```

As shown above the onCreate() method is called from line 85 to 170 is used to initialize all the activity component when the application is trying to open, some of the components includes the TextView, Button, Location Manager and main fragment activity.

6.3.3 add_Phone(View v)

1.8 The Screenshot below is the source code for add_Phone(View v) method

```
171    public void add_phone(View v){
172
173        final Dialog dialog = new Dialog(MainActivity.this);
174        dialog.setContentView(R.layout.addphone);
175        dialog.setTitle("Add Phone Number");
176        final  EditText addnum=(EditText)dialog.findViewById(R.id.phoneNumberText);
```

The add_Phone(View v) method is a submit button implemented as part of the component of the main activity for adding extra Relative contact to the application.

6.3.4 Onstop() Method

1.9 The Screenshot below is the source code for Onstop() method

```
199    @Override
200    public void onStop() {
201
202        // If the client is connected
203        if (mLocationClient.isConnected()) {
204            stopPeriodicUpdates();
205        }
206
207        // After disconnect() is called, the client is considered "dead".
208        mLocationClient.disconnect();
209
210        super.onStop();
211    }
```

This method is an inbuilt Activity API method used for overriding the Activity lifecycle, this particular method is used for stopping any runtime component before stopping the main activity, it is useful for handling linkages on the application. Here I used it to disconnect the location connection component for conserving phone battery.

6.3.5 onPause() Method

1.10 The Screenshot below is the source code for OnPause() method

```
213    @Override
214    public void onPause() {
215
216        // Save the current setting for updates
217        mEditor.putBoolean(LocationUtils.KEY_UPDATES_REQUESTED, mUpdatesRequested);
218        mEditor.commit();
219
220        super.onPause();
221    }
```

This method is also part the activity method state lifecycle, it is relevant for the current state of the activity component before it get deployed, here I used it to store the connect GPS state in a content Receiver for future use.

6.3.6 onStart() Method

1.11 The Screenshot below is the source code for OnStart() method

```
226    @Override
227    public void onStart() {
228
229        super.onStart();
230        mLocationClient.connect();
231
232    }
```

This method is used to automatically connect to Location Manager for searching for the current location, even if the GPS device is disabled. It is part of the Activity lifecycle which is called when phone is starting up.

6.3.6 OnResume() Method

```
252⊝     @Override
253      public void onResume() {
254          super.onResume();
255
256          // If the app already has a setting for getting location updates, get it
257          if (mPrefs.contains(LocationUtils.KEY_UPDATES_REQUESTED)) {
258              mUpdatesRequested = mPrefs.getBoolean(LocationUtils.KEY_UPDATES_REQUESTED, false);
259
260          // Otherwise, turn off location updates until requested
261          } else {
262              mEditor.putBoolean(LocationUtils.KEY_UPDATES_REQUESTED, false);
263              mEditor.commit();
264          }
265
266      }
```

The onResumed() method is also part of the category of the Activity lifecycle which is called as soon as the activity leave the onPause state. Here, I used the method to get the last location which is stored on the Content Receiver in onPause() method, and continue to make the activity location component to get the lastest location update.

6.3.7 getLocation() method

```
368⊝     public String getLocation() {
369
370          // If Google Play Services is available
371          if (servicesConnected()) {
372
373              // Get the current location
374              Location currentLocation = mLocationClient.getLastLocation();
375
376              // Display the current location in the UI
377              mLatLng.setText(LocationUtils.getLatLng(getBaseContext(), currentLocation));
378              gps = LocationUtils.getLatLng(getBaseContext(), currentLocation);
379          }
380          return gps;
381      }
```
The getLocation() method is a user define location for checking whether the google play location service is currently connected and retrieve the current GPS location from it.

6.3.8 getAddress() method

```
391      public void getAddress() {
392
393          // In Gingerbread and later, use Geocoder.isPresent() to see if a geocoder is available.
394          if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD && !Geocoder.isPresent()) {
395              // No geocoder is present. Issue an error message
396              Toast.makeText(this, R.string.no_geocoder_available, Toast.LENGTH_LONG).show();
397              return;
398          }
399
400          if (servicesConnected()) {
401
402              // Get the current location
403              Location currentLocation = mLocationClient.getLastLocation();
404
405              // Turn the indefinite activity indicator on
406              mActivityIndicator.setVisibility(View.VISIBLE);
407
408              // Start the background task
409              (new MainActivity.GetAddressTask(this)).execute(currentLocation);
410          }
411
412      }
```
The getAddress() method is also a user define method, its user the current android system installed Google play service to retrieve the current address from Google MAP over the Internet.

6.3.9 UserData Class

The userdata class is used to capture data required for interacting with the MATS from the user, its captured userid, password, relative contact and message from the user and automatically generate the phone IMEI and SIM IMSI from the phone operating system saving the user time from searching for this information on the phone input it accordingly. Then, get it committed into the SQLlite database table created.

Below are the methods and field used in gathering the information.

```
24          loginDataBaseAdapter=new LoginDataBaseAdapter(this);
25          loginDataBaseAdapter=loginDataBaseAdapter.open();
26
27          final EditText userIDText = (EditText) findViewById(R.id.userNameText);
28          final EditText phoneText = (EditText) findViewById(R.id.phoneText);
29          final EditText passwordFieldText = (EditText) findViewById(R.id.passwordText);
30          final EditText msgFieldText = (EditText) findViewById(R.id.msgText);
31          final Button submit = (Button) findViewById(R.id.buttonAccount);
```

The code in line 24 and 25 are used to instantiate the database class and open the database for read and write permission.

The code in lines 27 to 30 are used to create four (4) fields namely; userIDText, phoneText, passwordFieldText and msgFieldText respectively. While the code in line 30 is used to create a submit button called submit for committing all the values into the MATS database table.

Validate and saving the information onto the Database

```
46      submit.setOnClickListener(new View.OnClickListener() {
47
48          @Override
49          public void onClick(View v) {
50              String userID2 = userIDText.getText().toString();
51              String phone2 = phoneText.getText().toString();
52              String password2 = passwordFieldText.getText().toString();
53              String msg2 = msgFieldText.getText().toString();
54              if(userID2.equals("")|| phone2.equals("")|| password2.equals("")||msg2.equals(""))
55              {
56                      Toast.makeText(getApplicationContext(), "Field Vaccant", Toast.LENGTH_LONG).show();
57                      return;
58              }
59              else
60              {
61                  final SharedPreferences settings = getSharedPreferences(PREV_NAME, 0);
62
63                  String password = settings.getString("password", "1234");
64                  String imsi = settings.getString("imsi", "imsiString");
65                  String hdID = settings.getString("hwID", "hwID");
66                  String serialNo3 = settings.getString("serialNo2", "serialNo2");
67                  String androidid = settings.getString("androidId", "androidId");
68                  String msg = settings.getString("msg", "msg");
69                  Log.i("message in UserData", msg);
70                  loginDataBaseAdapter.insertEntry(userID2, password, phone2, msg, password2, imsi, hdID, msg2, serialNo3, androidid);
71                  Intent myintent = new Intent(getApplicationContext(), MainActivity.class);
72                  startActivity(myintent);
73                  Toast.makeText(getApplicationContext(), "Account Successfully Created ", Toast.LENGTH_LONG).show();
74
75              }
76          }
77      });
```

The above are used to check and validate the inputted value whether they conform with the exact way we want the details to be filled before saving them onto the database, the conditional statement in line 54 does the validation.

The code in line 70 calleds the insertEntry() method we've create in the Database class, and pass all the require parameters to the method and the loginDatabaseAdapter object used it to save the data onto the database table.

6.4 Communication Module Implementation

This module deals with communication. It intimates the relative on SIM change, responds to incoming SMS, gets the GPS co-ordinates and sends it to the relative, connects to internet sends the GPS co-ordinates to Google maps and gets the postal address from it and forwards it to the relatives phone number. The communication here is with satellites, with the other mobile phone and on internet.

6.4.1 The GetAddress AsyncTask Class

This class is used to work on the background and query for the GPS coordinate and the Postal address then handed it over to the required class for further processing

6.4.2 doInBackground() method

This method does exactly as its name suggest does all the checking in a silent mode without interference with the current activity to retrieve the GPS coordinate and Postal address with reverse geo-conferencing mechanism.

6.4.3 Receiver Class

The Receiver is a class which extends the broadcast Receiver API class for receiving any notification such incoming SMS, Email, battery drainage, SIM change and so on.

Therefore, I implement this class to handle all the incoming SMS alert and respond if the SMS body conform to the predefine query.

onReceive() Method

```
718    @Override
719    public void onReceive(Context context, Intent intent) {
720        getAddress();
721        String gps = getLocation();
722        Log.v("Lat/Long", gps);
723        Toast.makeText(getBaseContext(), gps, Toast.LENGTH_LONG).show();
724        Toast.makeText(getBaseContext(), streetName, Toast.LENGTH_LONG).show();
725        Bundle extras = intent.getExtras();
```

The onReceive() method foes all the operation stated above, please note that this method only execute at intervals for which the operation needed to execute.

6.4.4 MyReceiver Class

This is another Broadcast Receiver class which extends the BroadcastReceiver API class and Implement the onReceive() method for check the SIM state, and act according to the given query. In case the SIM is swapped its automatically send ad SIM to the relative phone number.

6.4.5 MyService Class

This is a service class it extends the Intentservice API class on android to implement steady state service for checking the phone state at bootup and notify the broadcast receiver to check the current SIM against the registered before the phone rebooted, this operation is done before the phone completed its boot sequence in a stealthy mode.

6.4.6 The Manifest File

This file contains all the components of the application attributes. All Android application will have AndroidManifest.xml file in its root directory. All the information about the application should be presented to android which is done by this manifest file, so that the application can run successfully on android. What the manifest file does is explained below;

1. It creates java package name for application which acts as a unique identifier.

2. It describes the components such as services, receivers, activities, broadcast and content providers which are collected in application and also the names of classes which employ all components and publish their capability such as which Intent can they handle.

3. It also determines which process host application components.

4. It declares permissions in order to access protected space of API and also with which applications it can interact.

5. And also declares permission in order for other applications to interact with its components.

6. The minimum level of API that the android applications require.

7. Lists linked libraries of that particular application.

<application> Tag
This is a container tag which defines the application components viz services, activities, receivers, content providers and permissions.

<service> Tag
Service is declared as application components. Services doesn't have visual user interface like activities. These services run in background or as a rich API which can be called by other applications. In manifest file all the services should be represented the one which are not tagged will not run at all.

<activity> Tag
Activity is a part of the visual user interface which implements parts of the application. Same as service it should be represented in manifest file or else will not be run.

<Receiver> Tag
The Receiver class is declass as application components, which receives all broadcasted notification on the application and act on it less than 10 seconds, its has no visual display.

<Use-Permission> Tag

It is a security component of the application, which the android OS check before installing the application, in order to notify the users what and what the application will access on the phone, in the application I have numerous permission setup.

7.1 Installing the MATS Application on Android Phone

Step 1: Install File Manager

Android does not natively come with any method of browsing the data on your SD card, so you will need to install a file manager from a market. There are a large variety of file manager available on the Android, but ASTRO File Manager will do.

Step 2: Copy MATS.apk file to SD card

Once you have ASTRO File manager installed connect your android device to your PC using your USB cable. Mount the SD card and copy over the MATS.apk file to the SD card for installation.

The MATS.apk file on ASTRO File Manager
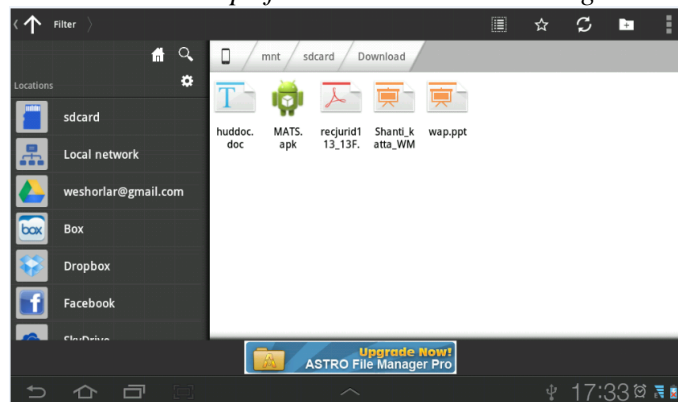
Step 3: Install MATS.apk on your android

On your android device, navigate to the MATS.apk file using the downloaded ASTRO file Manager and select it.

This will open a dialog box allowing you to install the MATS app select "Open App Manager".

This will display an agreement dialog box for the installation
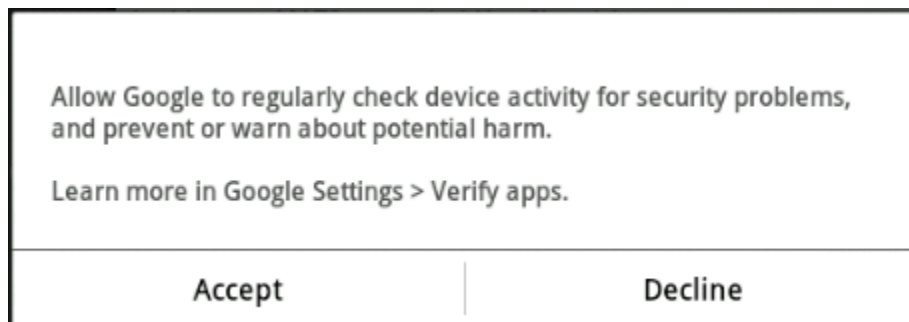
On the next two pages select "Install" again to install the MATS.apk
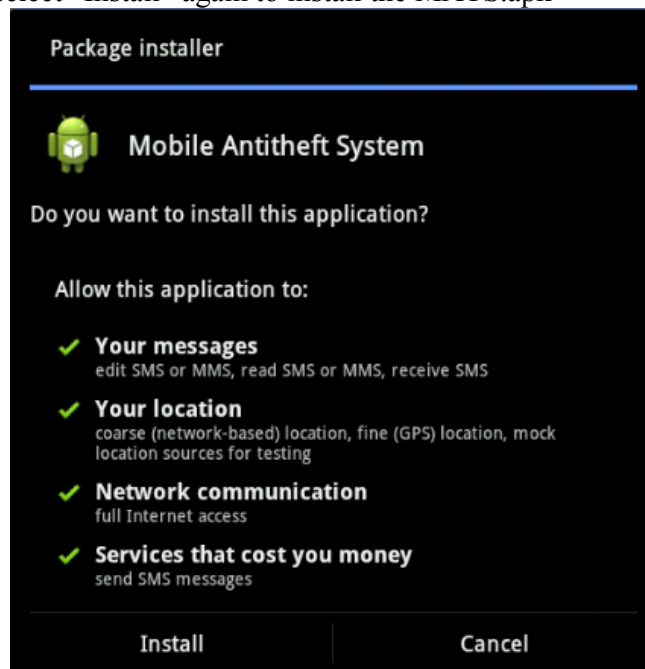

7.2 End user Documentation Guide:

Below is the end user oriented documentation for using Mobile Anti-Theft System.

After the Application is download and stalled on any android device from Google Play or online, the first interfaces you see is the Main Interface which consists of User Login and Signup Button, below screenshot depict the Main Interface.

The login button is on the left while the Sign up Button is on Right; this design is imitated from the Twitter Main Interface Design.

7.3 Main Interface

1.12 Splash Screen Interface for the MATS
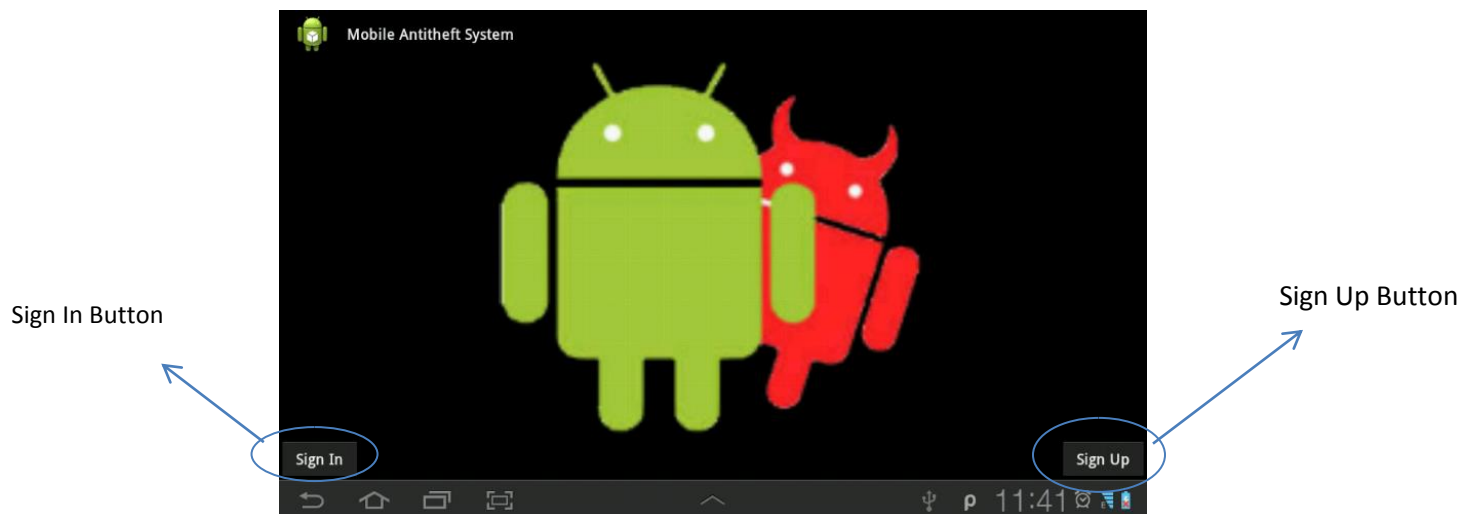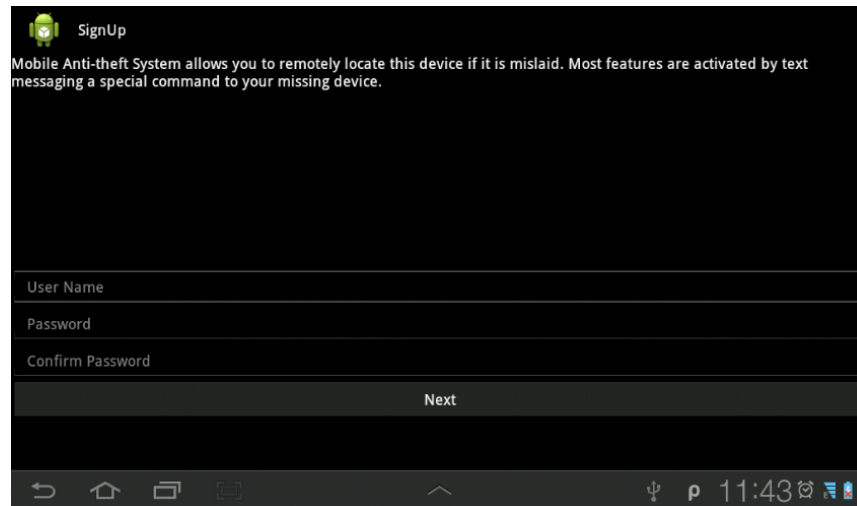
7.3.1 Signing Up to MATS

Step 1: select signup Button

Select sign up button located at the right down corner on the interface, this will display the sign up form interface, as shown below.
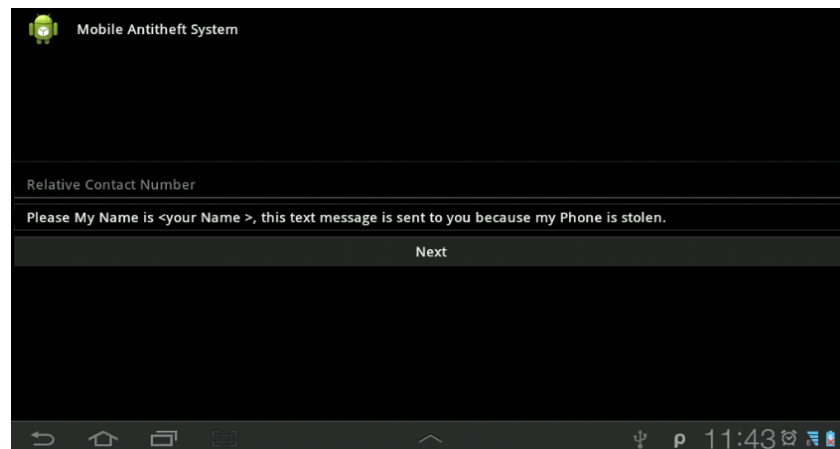
1.12 Signup Screen Interface for the MATS

Step 2: Enter the Username, Password and Confirm Password

On this interface user is required to type his / her user name, password and retype the password to confirm the password conformance.

Step 3: Click Next Button

Click on the Next button to go to the next interface, where user will enter the Relative contact and the contact message, the next interface is show below.

7.4 Adding Relative Contact Interface

Step 4: Enter Relative Number

User is required to enter the relative contact number on the field and move to the next field.
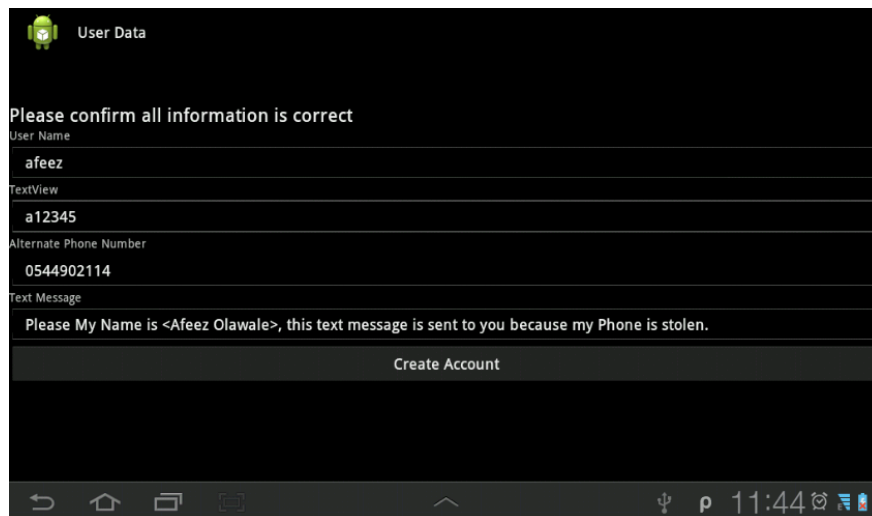
Step 5: Edit the <your name> content

The second field allow user to Edit the sentences and put his / her name of the <your name> field in order to communicate effectively to the recipient/buddy.

Step 6: Select Next Button

Then user should click on Next button to review all the data entered for confirmation below committing to the database.

7.5 The confirmation Interface is shown below.

Step 7: Review Entered Information and select Create Account

After all information entered confirms to be okay, then user clicks on Create Account Button to finish the signup process. At this stage system will switch to monitoring mode and start monitoring any Theft on the Device.

7.5 Logging onto the MATS

Step 1: Select Login Button on the Home Interface

Select the Sign in Button on the down left area of the screen, this will display a popup as shown below

1.13 login Screen Interface for the MATS
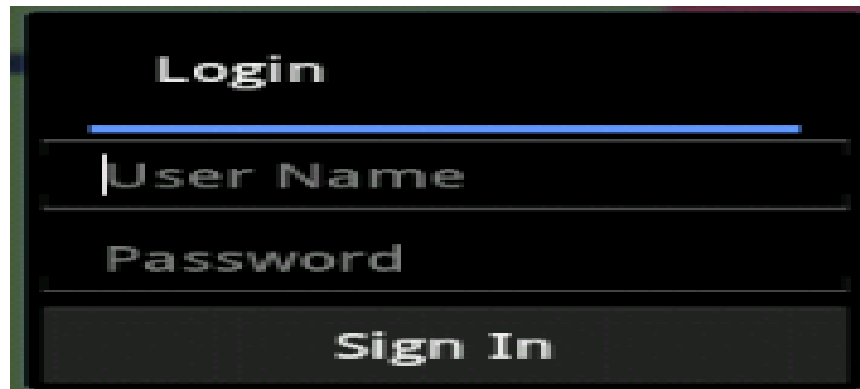
Step 2: On the Login Interface Enter User name, password and select Sing In Button

On the popup login Dialog user needs to input the user Name and Password created while signing up on the system, then click on Sign In to logon to the User monitoring console.

Note; if the User Inter Invalid data the System will flag an error message as shown below;
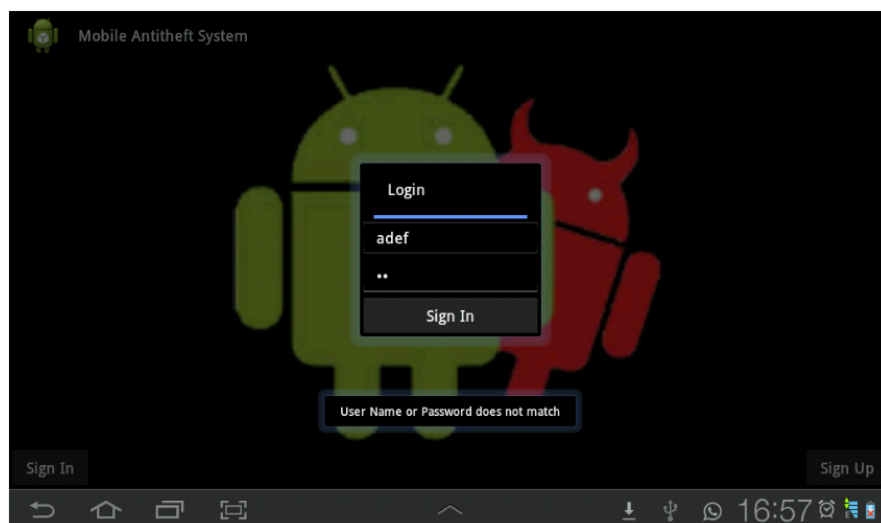
And will allow user to sign in user the correct credential the User Monitoring Home Interface will be displayed, Once successfully logon, user will automatically navigated to the User monitoring console as shown below.

*1.14 User Monitoring Home Screen Interface for the MATS*
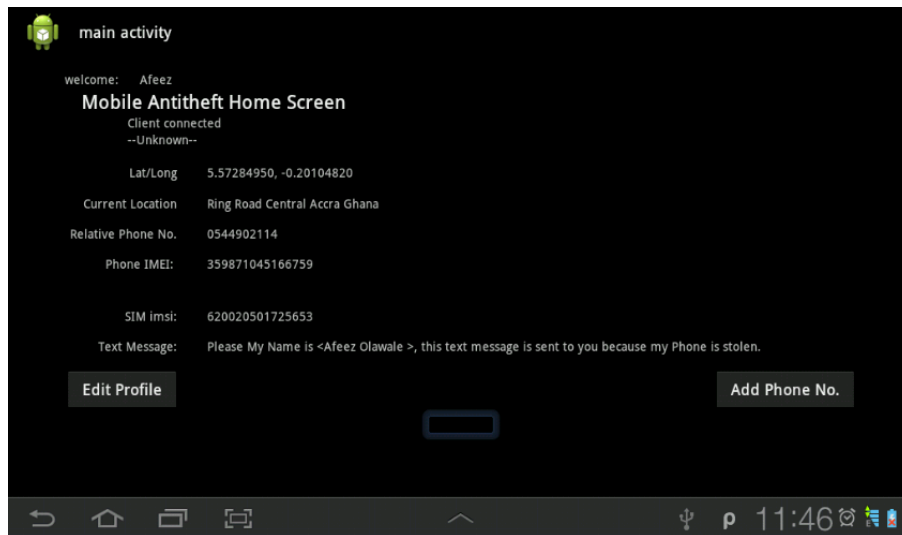
User monitoring Home Screen is shown above, user can edit is profile and add more relative phone number on this interface, below guide is used to Edit and Add more Phone Numbers.

7.6.1 Edit Profile

Step 1: Select the Edit Profile Button

To Edit Profile select edit profile button, this will take you to the user profile interface, which contain the current user information, below screenshot show the edit profile interface. Therefore, user can edit the username, password, Relative contact and Text message respectfully.

1.15 Edit profile Screen Interface for the MATS
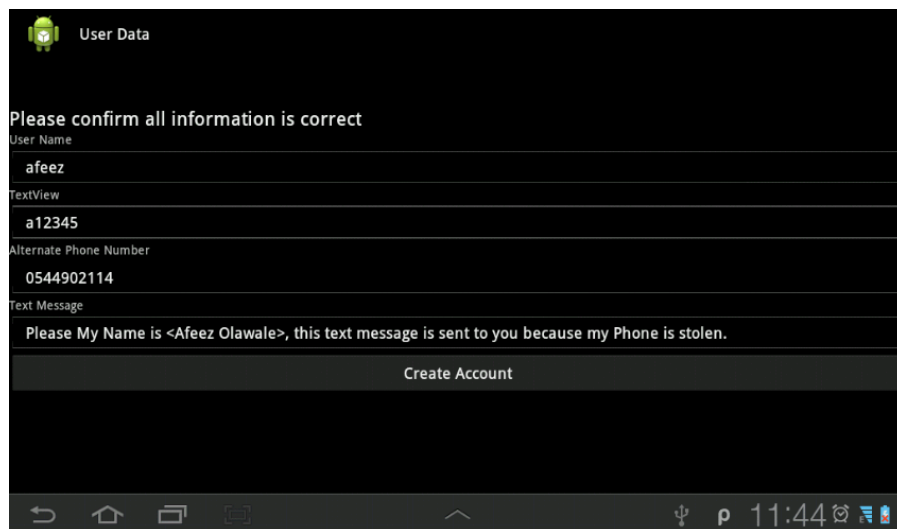
Step 2: Edit the require field and select Create Button

After user is done with editing the profile, then click on create account button, the information get updated on the database and on the User Home Monitoring screen.

7.6.2 Add Phone Number

Step 1: select Add Button the Home Screen

User can as well add more contact information some that more people could receive SMS from MATS and interact with it accordingly. Below outline the process of the adding more contact

1.16 Add phone number Screen Interface for the MATS
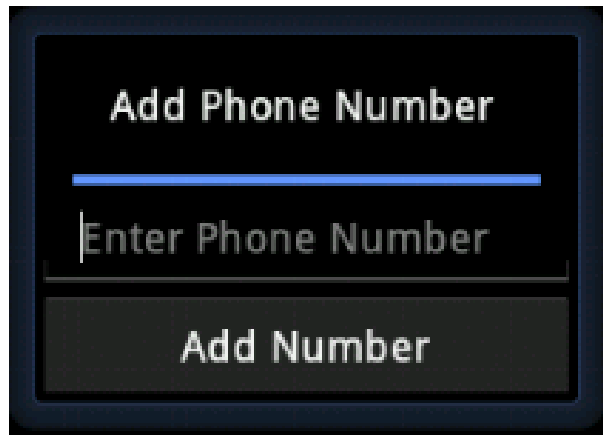
Step 2: Enter the new Phone Number

On the Add Phone Number Dialog box, enter the new phone number on the text field

Step 3: Select the Add Number button

Click on the Add Number Button to add extra phone number.

Conclusion

On theft detection my Application system would send an SMS to the owner alerting him of the phone number of the thief without the knowledge of thief in stealth mode.

Then our system would retrieve GPS co-ordinates from satellites and then send a SMS within a second.

Then MATS (Mobile Anti-Theft system) would connect to internet and send those GPS co-ordinates to GOOGLE MAPs and then using Google maps API, it would retrieve the postal address of the stolen phone and send it as an SMS to the relatively stored number, this functionality is not found on any of the existing anti-theft system.

Now the application is working for Device application, It's has localized system functionality and also accessing the details anywhere in the world making it an internationalize application.

Later I can implement some telecom concepts, so that the user will get the information about the activities of the company in a mobile device. As technology and requirements are changing day by day, I can add more functionality and implement the system with new requirements. The system is designed in such a way that it is flexible to change any further requirements Prescribed by the user.

References

References

(1)

Godfrey. (2013, November 12). *20031013292.html*. Retrieved November 12, 2013, from allafrica:
http://allafrica.com/stories/20031013292.html

(2)

Bank, W. (1991, August 14). *EXTINFORMATIONANDCOMMUNICATIONANDTECHNOLOGIES/Resources/282822-
1346223280837/MainReport.pdf*. Retrieved September 28, 2013, from www.worldbank.org:
http://siteresources.worldbank.org/EXTINFORMATIONANDCOMMUNICATIONANDTECHNOLOGIES/Resour
ces/282822-1346223280837/MainReport.pdf

(3)

Google. (2007, October 10). *android_overview.html*. Retrieved September 23, 2013, from
www.openhandsetalliance.com: http://www.openhandsetalliance.com/android_overview.html

(4)

LLC, M. O. (2010, May 28). *reports/q3-2013/*. Retrieved September 16, 2013, from www.developereconomics.com:
http://www.developereconomics.com/reports/q3-2013/

(5)

SPA, R. (1998, August 02). *newsroom/google%E2%80%99s-android-becomes-world*. Retrieved October 14, 2013,
from www.canalys.com: http://www.canalys.com/newsroom/google%E2%80%99s-android-becomes-
world%E2%80%99s-leading-smart-phone-platform

(6)

news/Android-steals-Symbians-Top-Smartphone-OS-crown_id16332 . (2001, November 21). Retrieved November
01, 2013, from www.phonearena.com: http://www.phonearena.com/news/Android-steals-Symbians-
Top-Smartphone-OS-crown_id16332

(7)

gadgets/2012/11/on-androids-5th-birthday-5-things-we-love-about-android/. (1998, December 30). Retrieved
November 09, 2013, from http://arstechnica.com/: http://arstechnica.com/gadgets/2012/11/on-
androids-5th-birthday-5-things-we-love-about-android/

(8)

/custom-roms-for-android-explained-and-why-you-want-them. (n.d.). Retrieved September 23, 2013, from http://www.androidpolice.com: http://www.androidpolice.com/2010/05/01/custom-roms-for-android-explained-and-why-you-want-them/

(9)

2013/07/01/android-led-by-samsung-continues-to-storm-the-smartphone-market-pushing-a-global-70-market-share/?ncid=tcdaily. (2005, June 10). Retrieved September 9, 2013, from http://techcrunch.com/: http://techcrunch.com/2013/07/01/android-led-by-samsung-continues-to-storm-the-smartphone-market-pushing-a-global-70-market-share/?ncid=tcdaily

(10)

8301-30686_3-20092399-266/google-just-bought-itself-patent-protection. (1994, March 05). Retrieved September 25, 2013, from http://news.cnet.com: http://news.cnet.com/8301-30686_3-20092399-266/google-just-bought-itself-patent-protection

(11)

us/google-android-installations-app-downloads,news-11861.html. (2000, August 03). Retrieved October 16, 2013, from http://www.tomsguide.com: http://www.tomsguide.com/us/google-android-installations-app-downloads,news-11861.html

(12)

watch?v=1CVbQttKUIk. (2005, February 15). Retrieved October 02, 2013, from www.youtube.com: http://www.youtube.com/watch?v=1CVbQttKUIk

(13)

news/technology-22542725. (n.d.). Retrieved November 23, 2013, from http://www.bbc.co.uk/: http://www.bbc.co.uk/news/technology-22542725

(14)

(Murphy). (2008). *ANdroid Cookbook for dummy.* Apress.

(15)

2010/05/01/custom-roms-for-android-explained-and-why-you-want-them/. (2009, December 07). Retrieved October 01, 2013, from www.androidpolice.com: http://www.androidpolice.com/2010/05/01/custom-roms-for-android-explained-and-why-you-want-them/

(16)

MARRITZ, I. (n.d.). *dear-apple-good-luck-against-the-smartphone-black-market.* Retrieved November 01, 2013, from www.npr.org: http://www.npr.org/blogs/alltechconsidered/2013/09/16/222125010/dear-apple-good-luck-against-the-smartphone-black-market

(17) developer.android.com: http://developer.android.com/guide/components/fundamentals.html

news/technology-22542725. (n.d.). Retrieved November 23, 2013, from http://www.bbc.co.uk/: http://www.bbc.co.uk/news/technology-22542725