

Implementation of Middleware for Internet of Things in Asset Tracking Applications: In-lining Approach

Masters Dissertation

by

Admire Mhlaba

Supervised by

Dr. Muthoni Masinde

This dissertation was conducted and submitted in fulfilment of the requirements of degree

Magister Technologiae: Information Technology,

at the

Central University of Technology, Free State, South Africa (CUT)



Declaration

I, Admire Mhlaba, student number [REDACTED], declare that the work in this dissertation is a presentation of my original research work and has been submitted for the award of Magister Technologiae: Information Technology at the Central University of Technology, Free State. Wherever contributions of other people are involved, every effort has been made to indicate this clearly, with due reference to the literature and acknowledgement. Further, this work was done under the guidance of Dr Muthoni Masinde, Department of Information Technology, at the Central University of Technology, Free State.

Admire Mhlaba

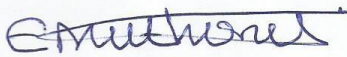
Date: 25 June 2015

Signature: 

In my capacity as supervisor of this dissertation, I certify that the above statements are true to the best of my knowledge.

Dr Muthoni Masinde

Date: 25 June 2015

Signature: 

Abstract

Internet of Things (IoT) is a concept that involves giving objects a digital identity and limited artificial intelligence, which helps the objects to be interactive, process data, make decisions, communicate and react to events virtually with minimum human intervention. IoT is intensified by advancements in hardware and software engineering and promises to close the gap that exists between the physical and digital worlds. IoT is paving ways to address complex phenomena, through designing and implementation of intelligent systems that can monitor phenomena, perform real-time data interpretation, react to events, and swiftly communicate observations. The primary goal of IoT is ubiquitous computing using wireless sensors and communication protocols such as Bluetooth, Wireless Fidelity (Wi-Fi), ZigBee and General Packet Radio Service (GPRS).

Insecurity, of assets and lives, is a problem around the world. One application area of IoT is tracking and monitoring; it could therefore be used to solve asset insecurity. A preliminary investigation revealed that security systems in place at Central University of Technology, Free State (CUT) are disjointed; they do not instantaneously and intelligently conscientize security personnel about security breaches using real time messages. As a result, many assets have been stolen, particularly laptops.

The main objective of this research was to prove that a real-life application built over a generic IoT architecture that innovatively and intelligently integrates: (1) wireless sensors; (2) radio frequency identification (RFID) tags and readers; (3) fingerprint readers; and (4) mobile phones, can be used to dispel laptop theft. To achieve this, the researcher developed a system, using the heterogeneous devices mentioned above and a middleware that harnessed their unique capabilities to bring out the full potential of IoT in intelligently curbing laptop theft.

The resulting system has the ability to: (1) monitor the presence of a laptop using RFID reader that pro-actively interrogates a passive tag attached to the laptop; (2) detect unauthorized removal of a laptop under monitoring; (3) instantly communicate security violations via cell phones; and (4) use Windows location sensors to track the position of a laptop using Google-maps. The system also manages administrative tasks such as laptop registration, assignment and

withdrawal which used to be handled manually. Experiments conducted using the resulting system prototype proved the hypothesis outlined for this research.

Acknowledgements

Firstly, I thank GOD for blessing me with the strength, wisdom and courage to undertake this arduous task, since a great deal of effort and determination was needed to complete this dissertation. It has unquestionably been by far one of the most challenging and exciting learning experiences in my academic career so far. It would have been hard to complete this research without the help and support of my family, friends, girlfriend and other kind-hearted people around me, who have in many ways contributed in different ways in the preparation and completion of this Master's dissertation.

Secondly, I would like to extend warm words of gratitude to my supervisor, Dr. Muthoni Masinde. Throughout the research process, she has given motherly support, guidance, insightful comments and help through her extensive knowledge from both her rich academic background and experience from working with other students. I am grateful for her huge personal and professional sacrifices, availability whenever I needed her to discuss or ask questions. I am far more grateful for her perpetual counselling, whenever I was down in spirit and feeling discouraged.

Moreover, I would like to express special thanks to my mentor, Professor Thandwa Mthembu, for opening doors for me throughout my academic life and for his dedication to lending a helping hand whenever it was needed most. Furthermore, I am supremely grateful for the financial support I received from CUT and all the people who openly met with me for interviews, guidance and contributions regarding the work in this dissertation. They all have undoubtedly contributed significantly towards the completion of this dissertation.

Final thanks go to my mother, aunt, grandmother and late great grandmother who raised me, kept me in their prayers, supported me and most importantly, believed in me, GOD bless you all. I also humbly ask GOD to shower his divine blessings upon these gentle and kind-hearted souls, Botle Malebo, Nomasonto Hlewayo, Victor Litabe, Zolisa Thungatha, Adedayo Adedeji and Boiki Mphore for standing by me through thick and thin and for their brotherly and sacrificial type of love towards me.

Table of Contents

Declaration	i
Abstract	ii
Acknowledgements	iv
Table of Contents	v
List of Figures	ix
List of Tables	xi
List of Appendices	xii
List of Abbreviations	xiii
1. Chapter 1: Introduction.....	1
1.1. Background	1
1.2. IoT Technologies.....	3
1.2.1. Wireless Sensors Networks.....	3
1.2.2. Radio Frequency Identifiers.....	4
1.2.3. Security Technology	4
1.2.4. IoT Communication and Controlling Technologies	5
1.2.5. IoT Middleware Technologies	6
1.3. Problem Statement	6
1.4. Research Questions	8
1.5. Objectives and Justification	8
1.6. Scope and Contribution.....	9
1.7. Research Methodology.....	10
1.8. Structure of Dissertation.....	11
2. Chapter 2: Literature Review	13
2.1. Introduction	13
2.2. An Overview of Internet of Things (IoT).....	14
2.3. IoT Implementation Challenges	15
2.4. IoT Hardware	17
2.4.1. Wireless Sensor Networks (WSNs)	17

2.4.2.	Radio Frequency Identification (RFID)	20
2.4.3.	Biometric Scanners	24
2.4.4.	Mobile Phones	27
2.5.	Middleware.....	29
2.5.1.	Middleware for Internet of Things.....	30
2.5.2.	Classification of Middleware Approaches for WSN	30
2.5.3.	Ambient Intelligent and Cyber-Physical Systems	33
2.5.4.	Cornell Cougar Sensor Database System	35
2.5.5.	Middleware Linking Applications and Networks (MiLAN)	36
2.5.6.	Semantic Middleware for IoT	37
2.5.7.	IoT Middleware Implementation Challenges	39
2.6.	Related IoT Applications	40
2.7.	Summary of IoT Technologies.....	42
3.	Chapter 3: Methodology	47
3.1.	Introduction	47
3.2.	Research Methods and Design	47
3.3.	Research Methodology Used	48
3.4.	CUT Case Study.....	50
3.4.1.	CUT Asset Management Problem Factors	51
3.4.2.	CUT Investigation Results	53
4.	Chapter 4 System Design and Development	58
4.1.	Framework Development.....	58
4.1.1.	Proposed Middleware Architecture	58
4.1.2.	Proposed Middleware Layers and Sub-Components.....	61
4.2.	System Analysis and Design	66
4.2.1.	Software Development Approach Justification	66
4.2.2.	System Development Approach	66
4.2.3.	System Specification Requirements	68
4.2.4.	System Use Case.....	69
4.2.5.	Database Design.....	74
4.3.	System Implementation.....	75

4.3.1.	Overall System Database Implementation.....	77
4.3.2.	Middleware Interaction with Database	79
4.3.3.	System Architecture and Middleware Implementation	80
4.3.4.	Microcontroller Pre-programming.....	83
4.3.5.	Data Capture	85
4.3.6.	Monitoring	87
4.3.7.	Alerts Communication and Dissemination	89
4.3.8.	Recovery	90
5.	Chapter 5: System Testing and Results	93
5.1.	Experiment Case 1	93
5.2.	Experiment Case 2	94
5.3.	Experiment Case 3	96
5.4.	Experiment Case 4	97
5.5.	Experiment Case 5	97
5.6.	Experiment Case 6	100
5.7.	Experiment Case 7	102
5.8.	Experiment Case 8	103
5.9.	Experiment Case 9	105
5.10.	Experiment Case 10.....	106
6.	Chapter 6: Conclusion and Further Work.....	109
6.1.	Discussion	109
6.1.1.	Hardware-based Tracking System Merits.....	115
6.1.2.	Theoretical Research Contributions.....	116
6.2.	Implementation Challenges and Solutions	118
6.3.	Further Research Work	120
6.3.1.	Future Activity 1	120
6.3.2.	Future Activity 2	121
6.3.3.	Future Activity 3	121
6.3.4.	Future Activity 4	122
	References.....	123
	Appendices.....	137

List of Figures

Figure 1.1: Proposed LMTS overview.....	9
Figure 1.2: LMTS architecture	10
Figure 1.3: Constructive research structure	11
Figure 2.1: An illustration of interconnection of IoT	15
Figure 2.2: Generic node architecture.....	19
Figure 2.3: Fingerprint structure	25
Figure 2.4: Fingerprint recognition process.....	26
Figure 2.5: Mobile cellular subscriptions	29
Figure 2.6: Taxonomy of WSNs middleware	32
Figure 2.7: WSNs middleware architecture.....	33
Figure 2.8: Query processing architecture	36
Figure 2.9: High-level diagram of a system that uses MiLAN.....	37
Figure 2.10: Semantic middleware architecture within WSN	38
Figure 3.1: Asset Ownership.....	54
Figure 3.2: Asset Replacement Affordability	54
Figure 3.3: Theft hotspots	55
Figure 3.4: Targeted offices	55
Figure 4.1: Proposed middleware architecture	62
Figure 4.2: Overview of system architecture and functions	63
Figure 4.3: XP process.....	68
Figure 4.4: Overall system use cases	71
Figure 4.5: Initial database entity identification	75
Figure 4.6: XP development principles	76
Figure 4.7: Excerpt of prototype database design.....	78
Figure 4.8: Database utilities	79
Figure 4.9: System components integration and interaction.....	82
Figure 4.10: Arduino microcontroller programming	83
Figure 4.11: Waspnote microcontroller programming extract	85
Figure 4.12: Data capturing interface	86

Figure 4.13: Monitoring sequence diagram	87
Figure 4.14: Laptop monitoring interface	88
Figure 4.15: Authentication interface	89
Figure 4.16: Communicated messages	90
Figure 4.17: Laptop recovery.....	91
Figure 4.18: Recovery sequence diagram.....	92
Figure 5.1: Database load test.....	94
Figure 5.2: Breach communication time.....	95
Figure 5.3: Location services	96
Figure 5.4: Identity mismatch error	97
Figure 5.5: Location finder query	98
Figure 5.6: Laptop tracking using query SMS.....	99
Figure 5.7: Database connection interface.....	101
Figure 5.8: Connection feedback	101
Figure 5.9: Database server connections	101
Figure 5.10: Error handling.....	102
Figure 5.11: Activity log data	103
Figure 5.12: Asset assignment	104
Figure 5.13: Asset registration.....	106
Figure 5.14: Password request	107
Figure 5.15: Password changing.....	107
Figure 5.16: Power control commands	108
Figure 6.1: Account identity	110
Figure 6.2: Mobile phone GPS access controller.....	110
Figure 6.3: Laptop GPS access controller.....	110
Figure 6.4:Find my phone service	111
Figure 6.5: Device's location access	111
Figure 6.6: Hardware based laptop tracking model.....	112

List of Tables

Table 2.1: RFID example applications	21
Table 2.2: RFID tag categories	23
Table 2.3: Summary of IoT supporting technologies	43
Table 3.1: Security system awareness	54
Table 3.2: Degree of importance for laptops and data.....	55
Table 3.3: System function usage	56
Table 3.4: Level of students' vigilance towards laptops.....	56
Table 3.5: Tag acceptance level.....	57
Table 3.6: Software acceptance level.....	57
Table 4.1: Prototype system requirements.....	69

List of Appendices

Appendix 1 : Preliminary investigation questions (A)	137
Appendix 2: Preliminary investigation questions (B).....	138
Appendix 3: Create record code	139
Appendix 4: Update record code	140
Appendix 5: Geographical position code.....	141
Appendix 6: Waspnote gateway code.....	142
Appendix 7: SQL broker code	143
Appendix 8: Location finder spike code	144
Appendix 9: Write off records	145
Questionnaire page 1.....	146
Questionnaire page 2.....	147
Questionnaire page 3.....	148
Questionnaire page 4.....	149
Questionnaire page 5.....	150
Questionnaire page 6.....	151

List of Abbreviations

3G	Third Generation
4G	Fourth Generation
ADC	Analog to Digital Convertor
ADT	Abstract Data Type
AmI	Ambient Intelligence
AP	Access Point
API	Application Programming Interface
BS	Base Station
CCTV	Closed Circuit Television
CeNSE	Central Nervous System for the Earth
CPS	Cyber-Physical Systems
CPU	Central Processing Unit
CRA	Constructive Research Approach
CUT	Central University of Technology, Free State
DLL	Dynamic Link Library
DsWare	Data Service Middleware
EPC	Electronic Product Code
FTT-MA	Flexible Time-Triggered Middleware Architecture
GHz	Gigahertz
GPRS	General Packet Radio Service
GPS	Global Positioning System

GSM	Global System for Mobile Communication
HCI	Human Computer Interface
HF	High Frequency
IC	Integrated Circuit
ICE	Internet Communications Engine
ICT	Information Communication Technology
IDE	Integrated Development Environment
IERC	European Research Cluster on the Internet of Things
IDS	Intrusion Detection System
IoT	Internet of Things
IPX	Internetwork Packet Exchange
ISDR	International Strategy for Disaster Reduction
LF	Low Frequency
LMTS	Laptop Monitoring and Tracking System
LTE	Long-term Evolution
MCU	Microcontroller Unit
MHz	Megahertz
MiLAN	Middleware Linking Applications and Networks
MWOS	Microsoft Windows 7/8 Operating System
OMG	Object Management Group
OaC	Over-the-Air Computing
P2P	Peer to Peer
POS	Personal Operating Space

QoS	Quality of Service
RAM	Random Access Memory
RD	Research Design
RDBMS	Relational Database Management System
RF	Radio Frequency
RFID	Radio Frequency Identification
SD card	Secure Digital Memory Card
SDK	Software Development Kit
SIM	Subscriber Identity Module
SINA	System Information Networking Architecture
SMS	Short Message Service
SoA	Service-oriented Architecture
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
UHF	Ultra High Frequency
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSNs	Wireless Sensor Networks
XP	Extreme Programming

1. Chapter 1: Introduction

1.1. Background

Advancements in computing technology have seen the internet being transformed, changing from a communication medium, primarily used for browsing the Web, sending and receiving emails, accessing multimedia content and services, playing games, and evolving into an interconnection of addressable things/objects known as Internet of Things (IoT) (Vermesan and Friess, 2014). IoT has become a prime research and development area because of (1) broad applicability in our daily lives, (2) ability to link digital and physical worlds, (3) compatibility with traditional internet standards in supporting services such as data transmission, analytics, applications, and communications (Gubbi, Buyya and Marusi, et al., 2013), and (4) ability to deliver the goals of, ubiquitous computing of networked physical objects, enabling ‘anytime, anyplace’ connectivity ‘for anything and by anyone’ (Kosmatos, 2011).

Internet of Things (IoT) is both an evolutionary and revolutionary paradigm which since its inception in the late 2008 (Ashton, 2009), has received significant attention from both scientists and researchers globally. Internet has been transformed from a medium that transports bytes of data from computer to computer, to an internet of connected devices (Mattern and Floerkemeier, 2010). The most common definition of IoT is: “IoT is a model in which objects or things are given unique and addressable identities and network connectivity; that enables these objects to communicate, monitor phenomena and be controlled using existing network infrastructures such as Bluetooth, Xbee, Wireless fidelity (Wi-Fi), 3G and internet” (Atzori, Iera and Morabito, 2010). The most well-known vision of IoT is Ubiquity, which is centred on interconnecting “anything, anywhere, anytime, and by anyone”.

Implementation of wireless sensor networks (WSNs) in Africa is hindered by economic, political and technical influences (Kyobe, 2011). The cost of wireless sensor kits is still beyond people’s reach: wireless sensor kits cost between 200 to 1000 Euros (Cooking-hacks.com, 2014). Wireless sensors’ communication protocols allow coverage of only between 500 to 10 000 metres (Libelium.com, 2014), and Global System for Mobile Communications (GSM) in most African countries (especially rural areas) is not reliable enough to support interruption-free

communication. In order to implement an efficient, effective, relevant, cost-effective and sustainable IoT security system for regions such as the Free State in South Africa, there is therefore a need to integrate the cheaper and more mature Radio Frequency Identifier (RFID) technology and the more prevalent mobile phones with WSNs.

Insecurity, both of our assets and lives, is a problem of great concern in South Africa and elsewhere in the world. Recently published crime statistics along with a theft assessment report by FreightWatch corroborate this. The report by FreightWatch (2013) outlined the widespread cargo truck hijacks and facility robberies that remained a huge cause for insecurity concerns in South Africa, despite a 17.8% decrease in the number of truck hijackings between April 2011 and March 2012. These perpetrators orchestrated and committed 821 hijacks over the 12-month period, compared to 999 during the same period a year earlier.

After conducting a preliminary investigation; it was discovered that a number of students and personnel at CUT frequently lost laptops if left unattended for just a few minutes. Despite the swift response by the University's security department, these laptops are never recovered.

Statistics from other parts of Africa are not any different; for instance, hundreds of innocent lives have been lost over the years among the nomadic communities in Ethiopia, Kenya, Somalia and Nigeria due to cattle rustling (Okoli and Okpaleke, 2014). Wireless sensor networks (WSNs), if innovatively integrated with the more mature Radio Frequency Identifier (RFID) technology and the relatively prevalent mobile phones, are able to deliver an effective and affordable tracking solution that can address the many insecurities facing our communities today.

Applications built around the IoT paradigm are propelled by three components which expedite pervasive computing: (1) hardware such as wireless sensors, actuators, cell phones and Radio Frequency Identifiers (RFIDs); (2) middleware to aid with mediation and data analysis; and (3) applications in form of prototypes (Gubbi et al., 2013). This study delivers an IoT integration architecture implementation that is tested using a laptop monitoring and tracking system. As such, the system is limited to the following elements under each of these components: Hardware – wireless sensors, RFIDs, mobile phones and fingerprint scanners; Middleware – a connection between hardware and application; Prototype – a functioning application that ties all the different hardware technologies together.

1.2. IoT Technologies

Applications built around the IoT paradigm are propelled by three components which expedite pervasive computing: (1) hardware such as wireless sensors, actuators, cell phones and Radio Frequency Identifiers (RFIDs); (2) middleware to aid with mediation and data analysis; and (3) applications in form of prototypes (Gubbi, Buyya and Marusi, et al., 2013). In IoT, miniature wireless sensors glean data, relay it to a base station for processing, processed data is transformed into meaningful information which can be analysed and used to make predictions about the future occurrence of a particular phenomenon. Most this information evolves into knowledge which is used to create novel computing algorithms and solutions to more multifarious problems through design and implementation of smart systems. Over time this newly acquired knowledge transcends into intellectual wisdom used to fuel up artificial intelligence (Rowley, 2007).

1.2.1. Wireless Sensors Networks

Wireless sensor networks comprise of an array of sensor motes, each embedded with one or more sensors to measure diverse phenomena, communication modules (for example Xbee, Bluetooth, Wi-Fi and GSM), storage medium such as secure Digital (SD) memory card and a micro-controller for data processing, and a battery to power the sensor. The sensors in a mote are programmed to (1) monitor different phenomena such as temperature, toxicity, seismic, water leakage, bendability, humidity, soil moisture, atmospheric pressure and acceleration to mention but a few, (2) process and analyse raw sensor data readings and (3) communicate the data to a base station or trigger preconfigured events to mitigate a peculiar manifestation (McGrath and Scanail, 2014).

The new generation of wireless sensor network nodes have paved way for the design and implementation of smart cities, smart agriculture, smart transportation and smart security systems. These wireless sensor motes come in different form factors, equipped with sophisticated on-board computational algorithms and wireless communication and energy saving techniques. Wireless sensor technologies are still far from reaching their maturity level, and therefore show great capacity and potential in developing solutions that would have otherwise been deemed

impossible to solve, and also extends to domains never before imagined (Vermesan and Friess, 2013).

1.2.2. **Radio Frequency Identifiers**

Radio frequency identification (RFID) technology is an instinctive and wireless identification process of unique objects using electromagnetic frequencies. RFID can be thought as an extension to the bar code technology, although RFID has augmented data carrying, communication and processing capacity and is a non-contact technology (Roberts, 2006).

RFID systems are made up of tags (the tag's integrated circuit is electronically programmed with unique identification data) which are glued on objects or assets to be identified, and readers (a reader has an antenna to transmit electromagnetic signals and a transceiver used to decipher the transmitted tag data) which interrogate the tags to acquire the tag identification data. The functioning of RFID systems incorporates tags and readers interacting with tagged objects (assets) and database systems. RFID technology has been adopted and deployed in applications ranging from building access control, supply chain tracking, toll collection, and vehicle parking access control, tracking library books, theft prevention, and vehicle immobiliser systems (Roberts, 2006).

1.2.3. **Security Technology**

Security is pervasive and can be classified as an imperative basic human need in both digital and physical space. The need for a robust, dependable, acceptable, accurate and inimitable identification and authentication system to combat both digital and physical delinquencies such as of fraud, cyber-crimes and identity theft perpetuated by use of outmoded security systems, which relied on passwords, personal identification number (PIN), passport, drivers licence and identification cards, ensued the culmination of biometric identification (Lourde and Khosla, 2010).

Biometric identification is a fool proof technology that uses individual's distinctive data that is conveniently accessible, and in a non-impudent manner and that is non-destructible over time such as fingerprints, palm-prints and irises. This study is limited to fingerprint scanning, which involves processes such as (1) fingerprint enrolment which is the capturing of a subject's

fingerprints; (2) storage of the data extracted from the subject; and (3) authentication which is the comparison of fingerprint patterns to the one stored on the database. If the fingerprint patterns match, then the subject is given the green light to use the system or access some resources (Jain and Kumar, 2010).

Developed countries use biometric technology in Visa application systems, and border control systems (Jacobs and Van Ranst, 2008). Fingerprint technology has also been implemented on mobile phones such as Samsung Galaxy S5, S6 and Apple iPhone 6 as way to curb theft, unauthorized use of these phones and also as a means to control access to sensitive data and mobile e-commerce and banking applications on phones (Pocovnicu, 2009).

1.2.4. **IoT Communication and Controlling Technologies**

Communication is imperative within IoT because it is the technology that makes IoT ubiquitous and wireless connectivity can be considered the back-bone of IoT. Most prevailing sensor network applications use short to long range technologies to achieve communication over time and space. Implementing appropriate communication channels (broadcast or multicast) along with routing algorithms has a direct impact on the scalability and durability of WSNs. Sensor nodes need to communicate among themselves to transmit sensor readings in single or multi-hop to a base station, and the base station might also communicate some directives back to deployed sensor nodes (Gubbi, Buyya and Marusi, et al., 2013).

There are different wireless communication protocols used in IoT that provide the flexibility and mobility of systems and these include: (1) Bluetooth (IEEE 802.15.1) – is a cheap and short range wireless radio communication system; (2) ZigBee or Xbee (IEEE 802.15.4) – is a wireless personal area network (WPAN) that provides self-organized, multi-hop, and reliable mesh networking with long battery lifetime, and operates in the personal operating space (POS) of 10m; (3) Wi-Fi (IEEE 802.11 a/b/g) – is a wireless local area network (WLAN) that provides access to internet at broadband speeds when connected to an access point (AP) (Lee, Su and Shen, 2007); and (4) General Packet Radio Service (GPRS) – is the new carrier service for GSM that amplifies and streamlines wireless access to packet data networks, such as the Internet due to its high speed data transmission rates and shorter connection time to packet data networks (Bettstetter, Vogel and Eberspacher, 1999).

Mobile phones have become a critical ubiquitous tool in the operation of IoT, evolving from devices primarily used to receive calls and text message into smart portable computing devices capable of generating, processing, storing and sharing data through the cloud or using diverse communication technologies. Equipped with computer like features, mobiles phones have become a new hub of IoT applications and diverse sensors used to monitor, visualise, manage and control IoT systems (Lane, Miluzzo, Lu, et al., 2010). Mobile phones are revolutionizing business processes and services by closing the dislocation of time and place. This simply means information gets to you without the need to depart from a remote place (Traxler, 2011).

1.2.5. **IoT Middleware Technologies**

Middleware technology for IoT is a light software layer that acts as digital glue used to piece together diverse spheres of applications and hardware (wireless sensors, RFID, biometric scanners) interchanging data over heterogeneous communication protocols and channels. The overall objective of a middleware is interoperability through implementation of a plug and play environment that provides a common interface and communication standard among the heterogeneous devices belonging to diverse domains in IoT. Technically, middleware provides the application communication interface between applications and hardware by masking their heterogeneity (Bandyopadhyay, Sengupta, and Maiti, et al., 2011).

Middleware are categorised based on their functionalities such as adaptability, context awareness, quality of service, security, efficient resource management, fault tolerance and resilience, interoperability, portability, scalability and target application to mention but a few. There exist a number of middleware solutions such as Ubiware, Hydra, Sirena, Cougar and MiLAN to name but a few (Bandyopadhyay, Sengupta, and Maiti, et al., 2011).

1.3. **Problem Statement**

Laptop theft has been skyrocketing in recent years. Some of the reasons fuelling this increase are: (1) laptops are portable and easy to conceal and pocket away; (2) laptops fetch a good second-hand price on the informal market; (3) availability of easy online disposal platforms such as Bidorbuy, OLX and Gumtree, where some of the stolen goods are sold cheaply and anonymously without revealing one's identity to the buyer.

The prime reason for this increase, however, emanates from the difficulty to track and trace the whereabouts of these stolen laptops. Many solutions have been developed in an attempt to annihilate this growing calamity, but their cost has left many preferring to do without one. Despite advances in hardware engineering and the availability of technologies to combat this menace, laptop manufacturers are not doing much to help, partly because they are not integrating tracking hardware such as GPS and GPRS in new laptop models.

Although incipient, the capability of IoT, notably WSNs, in implementing smart and resilient security systems is illimitable. On the other hand, RFID technology has been utilized since the 1940s to create effective and reliable security systems. Perpetual breakthroughs and advances in portable mobile phone engineering have not only led to powerful and more intelligent mobile devices but versatile and inexpensive mobile handsets. At virtually 123%, South Africa has one of the most remarkable mobile phone infiltration levels in Africa (GSMA and Deloitte, 2012). Innovative utilization of these phones could be acclimated to develop systems to alleviate the insecurity menace. An array of portable computing devices such as: tablets, laptops, smart phones, smart cards, and biometric scanners could be added to the list of equipment that could be exploited to deliver cost-effective and robust solutions to secure people and assets.

Africa as a continent is technologically inept; for example, most noticeably innovative IoT projects are pioneered by conglomerates such as IBM's Smarter Planet Project and HP Labs' Central Nervous System for the Earth (CeNSE) project and research consortiums such as the European Research Cluster on the Internet of Things (IERC). The second observation is that none of these initiatives are directly based in or involving Africa, let alone African researchers-driven. According to Masinde (2012), the continent is waiting for the bigger brothers (Europe, North America and Asia) to innovate, then she adopts. This study supports the fact that Africa does not have to follow this old trend that has always led to solutions that do not fit the realities in the continent, as supported by the famous "transferring of Northern designs to Southern realities" ideology (Heeks, 2002).

Many problems arise when dealing with several devices from different vendors because they use different communication protocols and standards. Integrating these heterogeneous devices might be a tedious and resource-intensive task. In order to overcome this technical quagmire, this study designed a generic middleware architecture for use in a laptop monitoring and tracking system

(LMTS), which was utilized to dwindle the complexities presented by the heterogeneity of hardware and improve interoperability.

1.4. Research Questions

Question 1:

- How can the integration of WSNs, RFIDs, biometrics and mobile phones into a monitoring and tracking tool, prevent and curb laptop loss?

Question 1.1:

- To what extent does the adoption of the aforementioned IoT minions ensure they are capable of creating a generic, intelligent, robust, resilient, reliable and scalable laptop anti-theft system?

1.5. Objectives and Justification

The primary goal of this study was to develop a cost-effective, intelligent and generic tracking and monitoring prototype for use in domains of the Central University of Technology, Free State. In Patton (1983) and Ratcliff (1988) prototyping is defined as “the process of developing a trial version of a system known as a prototype or its components in order to clarify the requirements of the system or to reveal critical design considerations”. In achieving this goal, the following sub-objectives were derived:

- To investigate the magnitude of laptop theft and the security systems at the institution in question.
- To understand what perceptions the victims of asset theft have of a theft-detering software.
- To propose and utilize a generic middleware architecture in creating an IoT-based system made up of at least four components: RFIDs, WSNs, mobile phones and biometric readers.
- To evaluate the middleware using a laptop monitoring and tracking system (LMTS).

1.6. Scope and Contribution

In this study, the researcher worked towards developing an integration middleware architecture for use in a laptop monitoring and tracking system (LMTS). This laptop monitoring and tracking middleware was built, based on the work by Hwang and Yoe (2011). The middleware was intended to process data efficiently, consume diverse data spawned by various interconnected devices, provide event-driven services based on data generated, and allow a proficient and flexible interface to interact with heterogeneous IoT hardware. The middleware is shown in Figure 1.1 below; its role was to provide a standard platform to facilitate the manner in which the LMTS interacts with different hardware by masking their heterogeneity.

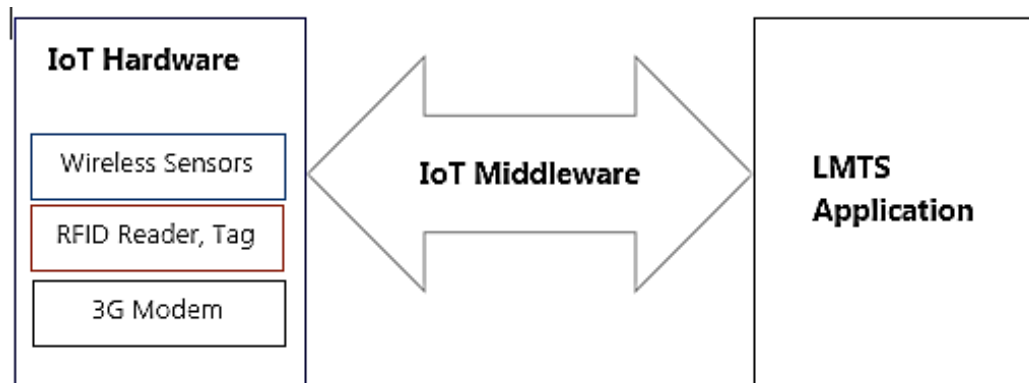


Figure 1.1: Proposed LMTS overview

Effective early warning systems consist of four components: (1) gathering the risk knowledge; (2) monitoring and predicting the situation; (3) communicating the warning messages; (4) responding to the warning (International Strategy for Disaster Reduction (ISDR), 2006). This laptop monitoring and tracking system (LMTS) adopted the aforementioned modules and adapted their names to: Data Gathering, Monitoring, Communication and Recovery as depicted in Figure 1.2. To evaluate the conceptualized middleware, a laptop tracking and monitoring system that integrates various forms of IoT hardware was implemented. The system prototype middleware was evaluated using diverse experimental cases to: (1) determine the competency to detect and communicate security violations; (2) measure the success of the middleware in processing many diverse database requests; (3) understand the responsiveness of the middleware in handling various over-the-air-computing (OaC) commands issued from mobile phones; and

(4) lastly observe the accuracy of GPS coordinates generated to track the geographical position of a laptop.

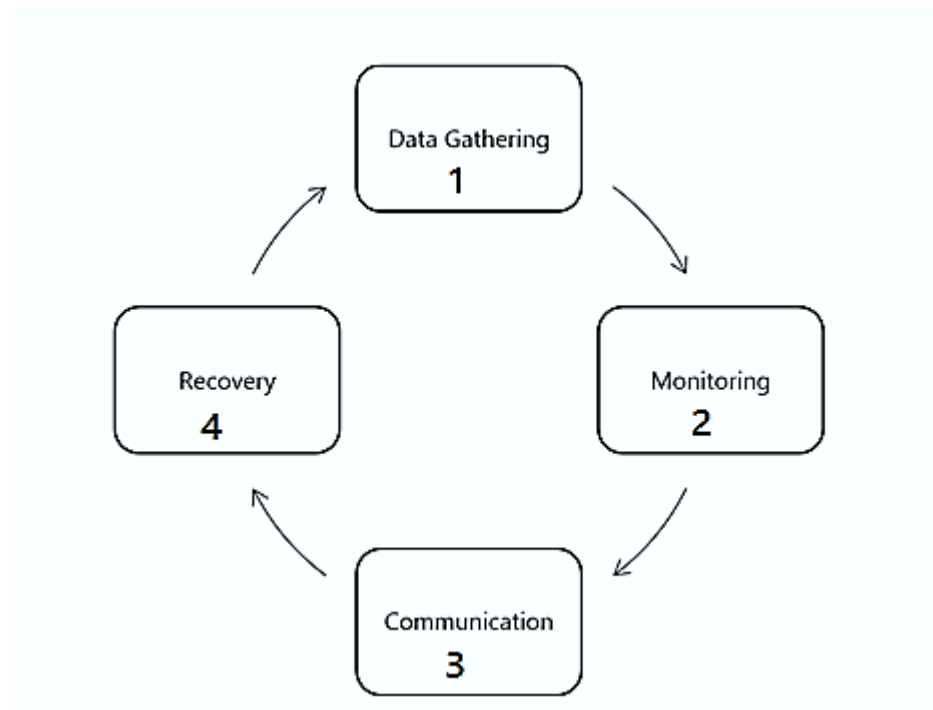


Figure 1.2: LMTS architecture (adapted from ISDR, 2006)

1.7. Research Methodology

This study was conducted following a seven step constructive research process suggested by (Kasanen, Lukka and Siitonen, 1993). Figure 1.3 below provides a graphic view of these steps; according to Kasanen et al. (1993), the initial step entails problem identification, followed by a deep analysis of the target domain and knowledge needed to propose a solution. These two steps are linked to activities carried out in Chapters 2 and 3.

The fourth step entails the transformation of system requirements into artefacts such as use-cases, story-boards, database designs, and architecture designs; this step is succeeded by the fifth step - implementation of the designs into a prototype; these activities have been presented in chapter four. The sixth step encompasses the evaluation of the newly developed construct and Chapter 5 contains the discussion of this step. The last step concludes the study and gives a brief insight into outstanding future activities; Chapter 6 was used to address this.

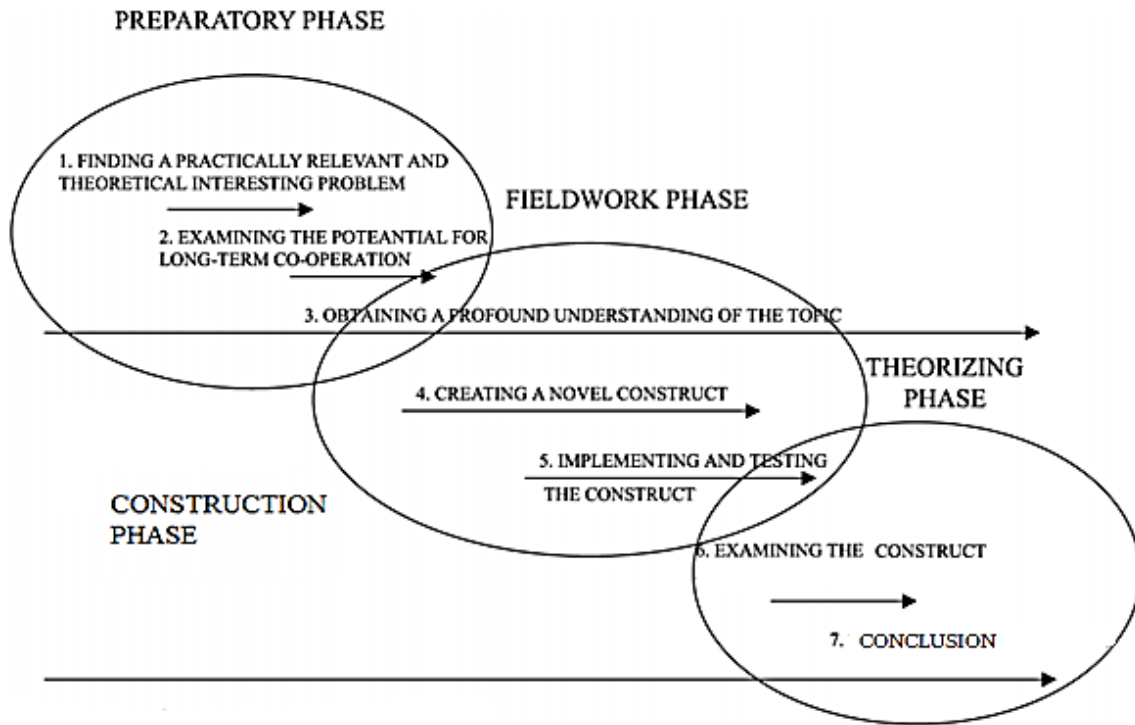


Figure 1.3: Constructive research structure (*adapted from Labro and Tuomela, 2003*)

1.8. Structure of Dissertation

The rest of the dissertation is structured as follows:

Chapter 2 presents review of relevant literature, in terms of: (1) IoT contextual overview; (2) IoT implementation challenges; (3) IoT hardware; (4) IoT integration approaches; (5) IoT tracking applications; and (6) IoT middleware implementation challenges.

Chapter 3 presents (1) research methods used to conduct the study; (2) CUT case study; and (3) CUT case study investigation results.

Chapter 4 presents (1) framework development; (2) system analysis and design; and (3) prototype implementation. The purpose of this chapter is to erect the prototype using technologies discussed in chapter two, and using the methods in chapter three to bind everything together using the envisioned IoT middleware architecture.

Chapter 5 is dedicated to discussing the results of the prototype derived from experimental tests conducted using the developed prototype.

Chapter 6 concludes the research study and also delves into discussing future directions the study intended to take.

2. Chapter 2: Literature Review

2.1. Introduction

Internet of Things was developed to act as a linking platform between digital and physical worlds, with the intention to make our daily activities more manageable and affluent (Gershenfeld, Krikorian and Cohen, 2004). For this physical and digital linkage to materialize, there was a need to provide a digital identity to real-world objects. This opened up new computing avenues to be explored by researchers who are now working around the clock to deliver applications that actively participate in diverse domains with minimum human intervention, to solve real life problems. According to research commissioned by Cisco, there will be 50 billion devices connected to the internet by 2020 (Zaslavsky, Perera and Georgakopoulos, 2013). This explosion of connected tiny computing devices is worrisome because there are no common standards to support interoperability, privacy and integration of these heterogeneous devices.

However, the realization of this relationship between digital and real world and the development of applications is a great challenge for industry experts and individual researchers. The challenges emanate from the need to conceal the underlying complexity of the environment by shielding applications from explicit management of: (1) incompatible network protocols; (2) miniscule heterogeneous devices, that are sometimes battery-powered and with limited computational capabilities; (3) parallelism, data replication and network faults (Han, Yoon, and Youn, et al., 2004). If poorly incorporated, it will result in applications that suffer from integration, interoperability, scalability, security and synchronized data management issues (Hadim and Mohamed 2006).

Middleware services provide a novel approach that support the implementation, maintenance, and operation of IoT-based applications through provision of a platform that: (1) shields hardware heterogeneity in sensor networks; (2) coordinates and distributes activities to sensor nodes; (3) performs data filtering, aggregation and storage; and (4) significantly enhances the development of diverse applications (Römer, Kasten and Mattern, 2002). This chapter presents related literature that was used in acquiring an understanding of basic and advanced concepts

regarding IoT-enabling technologies, such as RFID, wireless sensors, biometric scanners and mobile phones. The last section discusses middleware architectures in detail.

2.2. An Overview of Internet of Things (IoT)

Internet of Things (IoT) is both an evolutionary and revolutionary paradigm which, since its birth in late 1998 as a concept (Ashton, 2009), has received considerable attention especially from researchers around the globe and is projected to interconnect billions of devices that can sense, communicate, compute and potentially actuate (Zaslavsky, Perera and Georgakopoulos, 2013). Internet over the years has transformed, changing from a communication medium, primarily responsible for conveying bits of data from computer to computer; and evolved into an interconnection of addressable things/objects; this has come to be known as Internet of Things (IoT) (Du and Roussos, 2013). Ashton (2009) crafted the term “*Internet of Things*”, though it had a meaning and used in a context different from what it is known today. “*IoT is an interconnection of addressable things/objects using diverse technologies such as wireless network and internet*”.

IoT has become a prime research and development area because of: (1) broad applicability in our daily lives; (2) ability to link digital and physical worlds; (3) communication protocols that are compatible with traditional internet standards, to support services such as data transmission, analytics, applications, and communications (Gubbi et al., 2013); and (4) support of ubiquitous computing through implementing the ‘4As’ vision of the IoT paradigm (Interconnection of Anything, Anytime, Anywhere by Anyone) for convenient delivery of services (ITU, 2008).

IoT has become a universal breeding ground for many researchers and commercial entities currently working around the clock to deliver IoT-driven solutions that utilize the IoT paradigm to solve both research and business problems. Diverse applications used in different domains such as aviation, military, medical, environmental and agriculture have been developed utilizing the IoT paradigm (Priyadarshini, 2013). Some of the most promising and important IoT applications are in the security and health domain. A leap into the future from a technological crystal ball helped CISCO to predict the future IoT trend; about 50 billion things/objects will be interconnected by 2020 (Evans, 2011). This massive and remarkable growth will be driven by continuous advancements in electrical and software engineering.

This interconnection of ‘things’ is not limited to digital computing objects such as radio frequency identifiers (RFID), mobile phones, tablets and computers, but incorporates ‘things’ such as human beings, plants, domestic/wild animals, virtually ‘anything’ (see Figure 2.1).

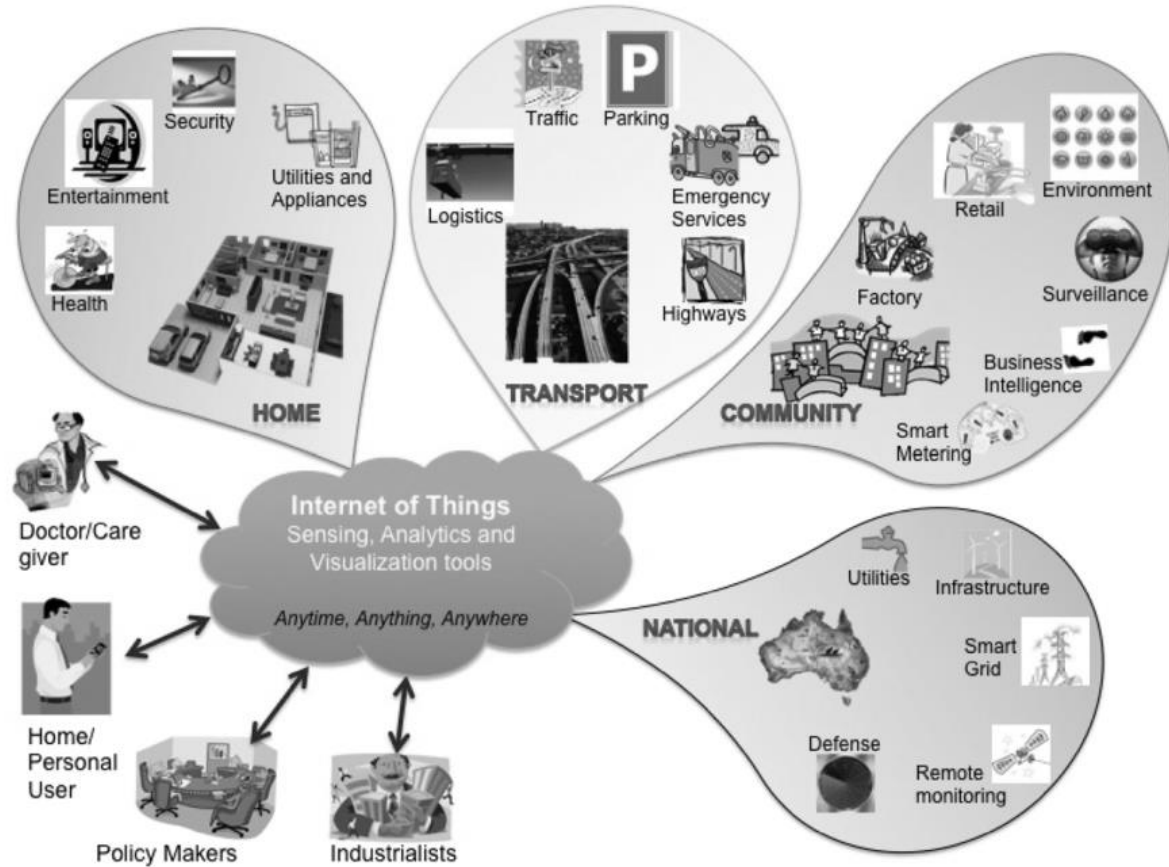


Figure 2.1: An illustration of interconnection of IoT (Gubbi et al., 2013)

2.3. IoT Implementation Challenges

The manifestation of global device interconnection using the IoT paradigm involves integrating (in most cases): (1) dissimilar devices and data networks; (2) numerous communication standards and protocols; and (3) a wide variety of applications. This endeavour often involves merging legacy networks protocols (internetwork packet exchange (IPX)), standards and applications. Dealing with this kind of diversity is one of the major IoT ‘showstoppers’; below are some well-documented factors researchers and industry experts are working diligently to address.

Heterogeneity of devices in the IoT paradigm is a major factor that need to be accounted for since IoT devices use different computational algorithms (MiLAN or Cougar) and different communication protocols (GSM, GPRS, TCP, IP). Therefore, architectures need to have a high management level of such heterogeneity (Miorandi, Sicari, De Pellegrini and Chlamtac, 2012).

Low energy consumption circuits are critical to the fruition of IoT. This entails development of innovative techniques to commendably manage energy consumption at different levels of the network design, from network routing down to the architecture of individual devices (Vermesan and Friess, 2013).

Communication in IoT is achieved using different standards such as frequency bands supported by varying hardware; this means different communication protocols such as transmission control protocol/internet protocol TCP/IP version 4 or version 6 may be used. IoT-oriented communication protocols are needed, they should be able to support multi-frequency bands, manage complexities of different standards of TCP/IP, and improve quality of service through provision of high transmission rates. Without clear and accepted standards, the adoption of ubiquitous computing may soon spiral out of control and ultimately be unattainable. Viable communication standards centred on energy efficiency, security and privacy and compatible protocols at different frequencies are therefore needed to support interconnection of objects on a global scale (Bandyopadhyay and Sen, 2011).

Global implementation of IoT might be hindered by **device interoperability**. Hardware from different vendors might not be interoperable, even if they operate using the same standards (Chaqfeh and Mohamed, 2012). This entails the creation of hybrid standards that integrate heterogeneous communication standards and protocols that operate at diverse frequencies, compatible with distributed or centralized architectures, and merge with other communication networks (Kominers, 2012).

Cyber-criminal activities have proved over the years that no system or network is immune to hacking. This is evident because **security breaches** have been on the rise due vulnerable security measures in place to protect communication networks. IoT is not an exception because it is being built on network standards known to have been breached and that is a huge threat to the realization of universal connectivity of things (Weber, 2010). Sound technical solutions are needed to guarantee privacy and security of data and the consumers of that data. A potential

solution could be found in hybrid security mechanisms that combine hardware-level security with evolving data encryption algorithms, key diversification to deliver supreme security measures that are more resistant and resilient to attacks (Mayer, 2009).

Techniques that engineers should use to attain global interconnection using IoT include: (1) dynamic discovery of devices and objects; (2) on-the-fly hardware reconfiguration; (3) hybrid communication standards and protocols (Sheng, Yang and Yu, et al., 2013); and (4) hybrid hardware-defined security measures.

2.4. IoT Hardware

The success of IoT is driven by interconnection of a myriad devices that have on-board sensing, communication and processing capabilities. The following section covers some of these devices that are relevant to this research in detail.

2.4.1. Wireless Sensor Networks (WSNs)

Wireless sensor networks (WSNs) are an invaluable component of realizing IoT; they form the ‘digital skin’ through which to ‘sense’ and collect the context of the surroundings and information of the physical environment. WSNs are especially instrumental in introducing intelligence to IoT because of their ability to cooperate and collaborate in carrying out tasks. A wireless sensor network (WSN) is a digital sensory system made up of a collection of millimetre-scale, self-contained, micro-electro-mechanical devices that contain sensors, computational processing ability, wireless receiver and transmitter technology and power supply (Yoneki and Bacon, 2005).

There exist a range of sensors capable of measuring physical, chemical and biological properties of objects and their environment. Prominent features of WSNs that qualify them as main candidates for ‘pervasive computing’ are smart integration with existing networks, multifunctional, context-awareness, self-configurability, self-sufficiency, easy of deployment (Blasi, Cacace and Casone, et al., 2007) and indeed their support for 4As vision of the IoT paradigm (ITU, 2008). Once interconnected together, sensors form a WSN. These sensors have

found use in a wide range of domains including defence, science, transportation, civil engineering and security (Priyadarshini, 2013).

WSNs applications fall under 3 categories: detection, tracking and monitoring (ITU, 2008). Examples of applications include: (1) disaster relief operations (Li, Huynh, Das and Du, 2008); (2) biodiversity mapping (Martonosi, 2004); (3) intelligent/smart buildings (Yeh, Wang and Tseng, 2009); (4) precision agriculture (Shinghal, Noor, Srivastava and Singh, 2010); and (5) drought monitoring and prediction (Masinde, Bagula and Muthama, 2012). WSNs also provide rich contextual information and alerting mechanisms against peculiar circumstances with continuous monitoring (Culler-Mayeno, 2006). For example, they reduce the need for caregivers and help the chronically ill and elderly to live an independent life (Alemdar and Ersoy, 2010).

Different nodes (nodes) are configured to have different maximum broadcast ranges, and this range is approximately 30 metres according to Culler-Mayeno (2006). These nodes create links with each other in different configurations to maximize performance. All the nodes are linked to a dedicated sensor node (the sink node) that sends data to a command station used to decode, analyse and process sensed data. In some configurations, the command station might communicate some directives to the dedicated node, which in turn broadcasts the message to all other child nodes in the network topology (Estrin, 2005). As shown in Figure 2.2 below, a sensor node is made up of four main components.

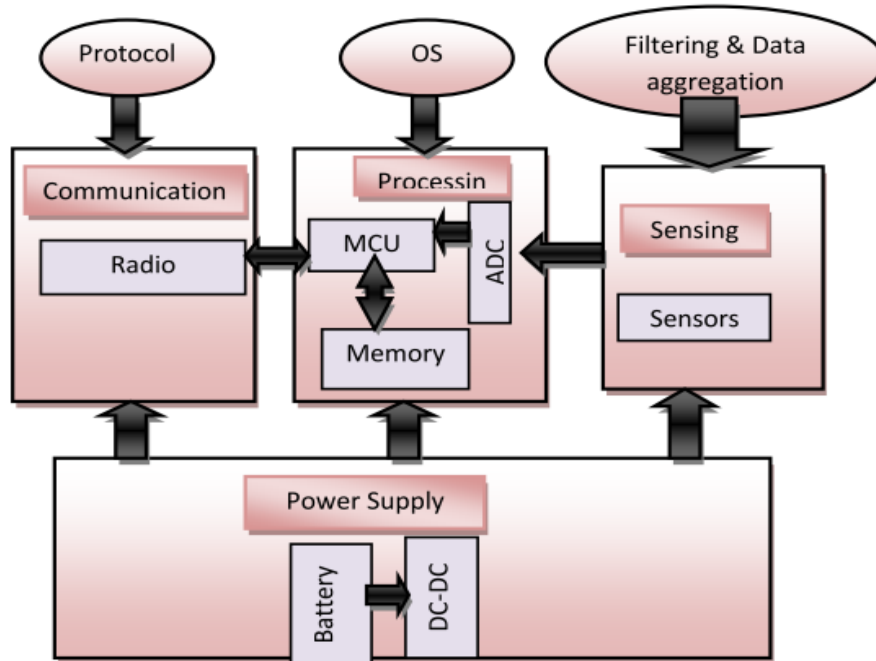


Figure 2.2: Generic node architecture (*adopted from Maraiya, Kant, and Gupta, 2011*)

Communication Unit: The radio wirelessly facilitates data interchange among sensor nodes in a given wireless network topology. The radio consists of a transmitter and receiver operating at unlicensed bands of near 2.4 GHz (global). It operates in four different modes in order to be energy efficient: transmit, receive, idle and standby modes. Thus, when it is not operational, that is transmitting or receiving, it is vital to temporarily put the radio in sleep mode to conserve precious energy (Maraiya et al., 2011).

Processing Unit: The microcontroller unit (MCU) provides decision making intelligence to the sensor node. The processing unit has an analog to digital convertor (ADC) used to digitize data from sensors and memory to store collected data and instructions needed to perform action on the data. The processing unit also performs other functions such as formatting data into packets for greatest transferring efficiency and optimizes power consumption to prolong the life of sensor nodes (Maraiya et al., 2011).

Sensing Unit: Sensors are designed to intelligently perform data assemblage once deployed and ultimately sense environmental or physical phenomena, passed to ADC responsible for transforming electrical signals into digital data through a process known as digitization. Many types of sensor categories exist such as, – (1) physical, (2) biological, (3) chemical – that

measure ecological parameters such as temperature, humidity, light intensity, vibrations, magnetic fields, and acoustics. Some sensor nodes are, for example, designed to have more sensors added to give them additional sensing capabilities (Maraiya et al., 2011).

Power Unit: The battery's prime job is to give life in the form of electrical energy to run the sensors, microprocessors and radio. It is important to design network topologies which are energy efficient to conserve this much needed resource that is required to keep the sensor node up and running. If a sensor node runs out of power, it dies and this may disrupt the normal operations of a network (Maraiya et al., 2011).

2.4.1.1. Operational Principle of Wireless Sensor Networks

WSNs collect data about what is happening, analyse it and perform preconfigured action. Action taken varies from WSNs implementation and depends on the calibration, configuration and purpose of WSN. Setting off an alarm or sending a message to cell phone could be the triggered action (Fitzek, Pedersen and Perrucci, et al., 2014). Each sensor node collects and transfers data using four steps: (1) Data Collection – periodical extraction of raw sensor readings; (2) Data Processing – including data encryption, routing decisions, data decryption, data encapsulation and aggregation of sensed data readings; (3) Data Packaging – data is formatted into packets capable of propagating the network efficiently; and (4) Communication – data is routed and transmitted to base station for processing.

A base station (BS) is a localized computer running special software, used mainly to receive, process, analyse and interpret sensor readings gathered by sensor nodes, save data to a connected database and send commands to sensor nodes through respective gateways (Kohanbash, Valada and Kantor, 2011). Configuration of nodes in the WSN may take different topologies (the structure or layout of a network) ranging from peer-to-peer (P2P), star, and mesh, flat-based, cluster-based, tree-based and chain-based topologies (Mamun, 2012). This study was limited to peer-to-peer topology, which is the one that was selected in configuring the nodes.

2.4.2. Radio Frequency Identification (RFID)

RFID is a contactless smart technology used to distantly retrieve data from or write (store) data to memory chip embedded within the integrated circuit of tags (Finkenzeller, 2003). RFID is capable of remotely locating and identifying tagged objects spontaneously using radio waves.

RFIDs are microelectronic devices that comprises of a microchip and an antenna; characteristically, the microchip has data-carrying capacity of 2 kilobytes or less. RFIDs have become a common replacement of barcode technology (Chen, 2010). In this case, RFID devices serve the same purpose as a barcode or a magnetic strip found at the back of debit or credit card by providing distinctive identification for tagged objects.

Advancements of RFID technology instigated diverse ways to integrate the technology in a myriad applications such as healthcare, library and tracking systems. Examples of such applications are presented in Table 2.1:

Table 2.1: RFID example applications

a) E-Passport – The American government in 2004 explored the use of an electronic passport (e-passport) with a RFID micro-chip affixed that operates at 13.56 MHz and which also complies with ISO 14443A and 14443B standards.
b) Toll Collection – Use of active RFIDs tags has been employed in America for toll collection for vehicles travelling at highway speeds in places like Maryland, New Jersey and New York. EZ-Pass is an example of this toll collection. The system operates at any frequency and is far more convenient because it is linked to the users' credit card to handle payments.
c) Payment Systems – Near Field Communication devices which operate using high frequency RFIDs with improved data speeds.
d) Animal Tracking – Low-frequency RFIDs have long been used for animal tracking. A well-known implementation of an animal tracking system is Zebra-Net Project conducted in Kenya Africa, for tracking zebras (Martonosi, 2004).

According to Finkenzeller (2010); three important components of a Radio-Frequency Identification system are:

RFID Transponder: Also known as a tag, this makes up the core of the RFID system; it performs data-carrying services. Transponders have unique identification information and are glued to objects to be located or identified. Tags are designed in many different shapes and sizes (form factor), and the main differentiating parameter is frequency range. They are available in “read only” versions, which can only be read-out and “read/write” versions with dual functionality to perform both read and write the transponder. They have unique, universal identification numbers which make it possible to track or monitor object movement within a given location. These tagged objects have their information linked to a system database. The transponders/tags are divided in three categories, depending on their operation mode:

- a) **Active RFID Tags** – have their own power source and signal transmitters for data transfer; these tags are not reliant on reflected waves and therefore may communicate over long distances ranging from 100 to 225 metres. They may have memory of about 128 kilobytes (Finkenzeller, 2003). Their lifespan is battery-dependent and they stop working once out of power. The advantage of these tags is that the reader can be positioned much farther away and still receive emitted signals.
- b) **Semi-passive RFID Tags** – are very similar to passive tags except that they are embedded with a small battery. The battery constantly powers the tag’s integrated circuit (IC) used for processing, data storage, modulating and demodulating radio-frequency (RF) signals. They are thus faster in response and are used to increase reading range; they are most beneficial in unfriendly and interference-free environments (Ahson and Ilyas, 2008).
- c) **Passive RFID Tags** – on the other hand, they do not have their own internal power source. As such, they can be much smaller (inexpensive and of lighter designs) and have a virtually unlimited life-span compared to other two categories. However, their reading range is limited due to absence of power to amplify RF signals; it is determined by the amount of power obtained from radio signals emitted by the reader and may cover ranges of about 10 centimetres (10cm) to 6 metres (Clampitt, 2014). They are mostly used for object identification and asset tracking and are maintenance-free, with data storage capacity as little as 10 kilobytes (10kb). Their operation is initiated when radio waves from the reader reach the chip’s antenna and are then converted into electricity enabling the tag to send back information stored on the chip.

Electronic Product Code (EPC): This is, for instance, a storage area used to store user-programmed instructions in ultra-high frequency (UHF) tags. The transmission range for UHF passive tags ranges from 3-6 metres and beyond 30 metres for active tags. Communication is achieved by a technique called backscatter where readers reflect back the reader's radio waves. The reflected signal transmits data encoded on the tag back to the reader (Finkenzeller, 2010). Three common operational frequency ranges for RFID tags are summarized in Table 2.2 below:

Table 2.2: RFID tag categories

RFID Tags Type	Tag Description	Frequency Range
Low Frequency (LF)	Have slow data transfer rate, shorter reading range which extends from 30 KHz – 300 KHz.	500 KHz.
High Frequency (HF)	Have higher data transmissions rates and are pricier than LF tags. Both LF and HF tags base their coil designs on inductive coupling, which uses the magnetic near field of antennas for communication.	15MHz
Ultra-High Frequency (UHF)	Have the greatest transmission range and with data transmission rates faster than LF or HF tags.	850MHz, 950MHz, 2.42.5GHz up to 5.8GHz

RFID Antenna: This is responsible for the emission of electromagnetic waves created by the reader, and the reception of radio frequency signals reflected back by the transponder. An antenna is made up of a coil with one or more windings. Antennas take different forms based on

their application as well as their ‘read and write’ range. Its electromagnetic field can be constantly produced or sensor-activated depending on the system design. The operating frequency discussed above is another parameter that determines how tags interact with each other on a physical level (Ahson and Ilyas, 2008).

RFID Reader: This is an electronic device used to wirelessly and intelligently retrieve or write data to tags through a process known as interrogation. Readers and tags exchange data using RF signals. The reader’s antenna has: (1) a transmitter responsible for emission of carrier radio waves; (2) a modulator to impact data commands upon this carrier signal; and (3) an amplifier to amplify the signal to activate the tag (passive), and thus giving the tag energy enough to initiate data communication.

The receiver has a demodulator responsible for decoding received signal and extracting data from tags. Some models are also rooted with another amplifier to intensify the signal for data reception and processing. The reader has a control unit integrated with a microprocessor that acts upon this data using an operating system and memory for data filtering and storage. After all the data have been processed, it is ready to be sent to a host system for further analysis. RFID readers come in many form factors, some are handheld or mobile and some are fixed (Finkenzeller, 2003).

There are several factors that can compromise the distance at which a tag can be read (the read range) or written to. The transponder antenna, the antenna gain, polarization of the antenna and the orientation and frequency used for identification or writing task. Another factor to consider is the placement of the tag on the object to be identified; this has a direct impact on the RFID system’s interrogation abilities (Finkenzeller, 2003).

2.4.3. Biometric Scanners

The term ‘biometrics’ originates from the Greek and is a derivative from the words ‘bio’ meaning life and ‘metric’ meaning standard of measurement. Biometrics therefore refers to a construct of science and technology for measuring inimitable life phenomena (Bhargava, Bhargava and Mathuria, et al., 2012). Biometrics in security context is a mechanism that uniquely identifies people by comparing distinctive physical characteristics; it compares distinctive physical characteristics (Jain and Kumar, 2010). Predominant biometric systems in

place today encompass facial, iris, voice, and fingerprint recognition systems. This technology is reliable but expensive; it is one of the most applied choices of identification due to its dependability, non-intrusive interfaces, and cost-effectiveness.

Fingerprint scanners scan fingerprints; the rough patterns visible on the front surface of a human finger. A fingerprint has an array of properties which encompass: (1) ridges which cover copious properties, comprising common features such as ridge-count pattern, delta point, geometric framework, and core point; and (2) minutiae features such as delta point, end point, break point, core point and fork, shown in Figure 2.3. The minutiae features are never the same (Zhang, Liu, et al., 2011).



Figure 2.3: Fingerprint structure (*adapted from Zaeri, 2011*)

Fingerprint identification is extensively achieved using a technique called ‘minutiae matching’ (Zaeri, 2011). It involves the examination of some unique fingerprint points determined by the local ridge features and their correlation. Conspicuous ridge characteristics are: (1) ridge cessation, which defines a point where a ridge ends snappishly; and (2) ridge branching which defines points where a ridge forks or diverges into branch ridges (Bhargava et al., 2012). A delta point is the tri-radial point with three ridges radiating from it, and a core point is the top most point on the inner most ridge (Zaeri, 2011).

The identification process revolves around comparison of papillae and dermal ridges of fingertips found on the objects touched or the finger itself. Biometric systems capture data and use a special set of algorithms to compute and compare this data with a stored reference data pattern (Finkenzeller, 2010). Fingerprint scanners are a trusted biometric and are widely used because of the convenience of acquiring fingerprints, uniqueness and permanence. The distinctiveness, universality and availability of inexpensive fingerprint sensors and its fool proof nature make the technology stand out above many identification systems in place.

According to Panda, Giri and Kumar, et al. (2012), there are four phases in the fingerprint recognition process, which are: (1) fingerprint image acquisition; (2) image pre-processing; (3) minutiae extraction; and (4) minutiae matching and identification. The fingerprint images require pre-processing in order to thoroughly and accurately identify the minutiae points for recognition. Fingerprint image pre-processing is further divided into image enhancement, binarization and thinning (Panda et al., 2012). Figure 2.4 below captures these steps diagrammatically.

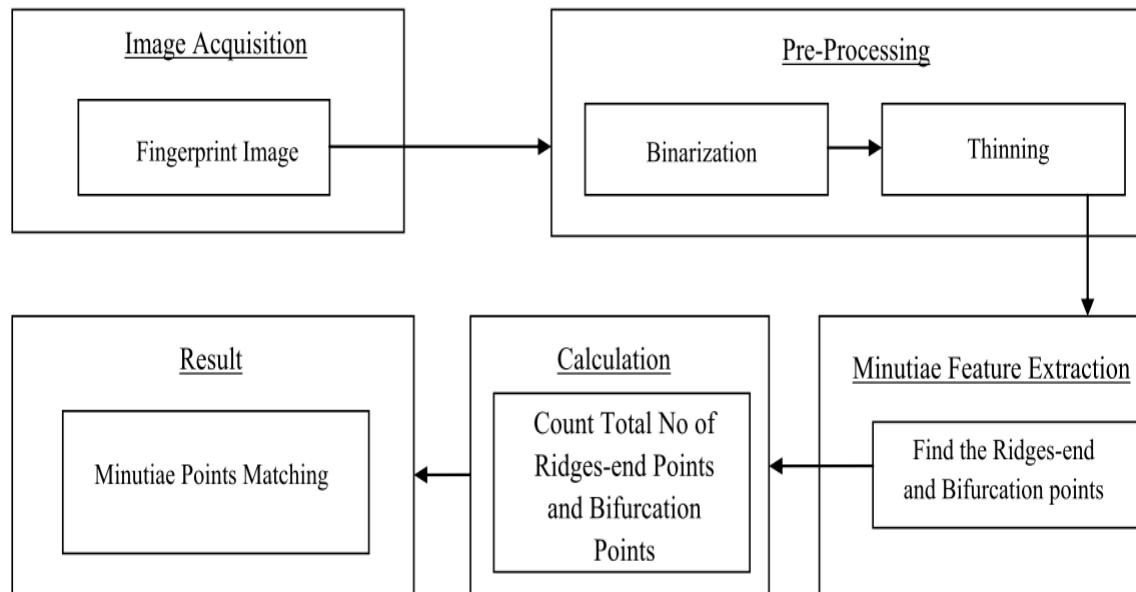


Figure 2.4: Fingerprint recognition process (adopted from Bhargava et al., 2012)

Image Acquisition: refers to the acquisition of fingerprint image using optical sensors.

Image Pre-processing: to reliably facilitate the extraction process, eliminate noises in the image, and transform it to a clear point with observable edges, so as to extract defined features. It has a direct impact on the identification outcome (Fernandez, Fierrez, Ortega-Garcia, et al., 2007).

Image Enhancement: is the improvement of the fingerprint image in order to augment the anatomical variance of ridges and valleys. Common fingerprint image enhancement algorithms are Gabor-based filtering, and Fourier-based filtering (Aguilar, Sánchez, Toscano, et al., 2008).

Binarization: this is meant to coalesce the grey scales, so that the image could be converted to binary representation (Bhargava et al., 2012).

Thinning: to eliminate the pixels on ridge edges and decrease their thickness, to keep ridges within a pixel of one width (Ezhilarasan, Suresh Kumar, Santhanakrishnan, et al., 2010).

Feature Extraction: extracting features directly from the grey scale image by means of tracking the grey scale crests, identifying the positions of features and defining the nature of features based on the tracking result.

Matching and Identification: is the use of extracted minutiae features to decide whether two fingerprints are identical and from the same finger. It involves comparing and matching the minutiae features of input fingerprint with fingerprint template stored in a database, and determining the most identical fingerprint (Fernandez et al., 2007). This process may be influenced by factors such as skin distortion cuts, dirt and humidity.

2.4.4. Mobile Phones

Advancements in communication technology have revolutionized the way humans and machines communicate. The evolution of wireless communication technology opened a plethora of opportunities and the most noticeable was the birth of mobile phones, also known as cellular phones (Kumar, Liu, Sengupta and Divya, 2010). Early mobile phones were pricey, bulky, and had poor network reception as a result they were difficult to use (Agar, 2013). Mobile phones are shifting from communication to service-oriented tools used around the globe by many people to: (1) conduct commercial services such as banking and shopping; (2) surf the internet; (3) access both business and personal emails; (4) schedule meetings, book flights or hotel accommodation; and (5) conveniently and cheaply obtain information on a variety of economic, political, social and environmental topics (Aker and Mbiti, 2010).

The perpetual growing demand for faster network speeds to meet advancements in mobile communication systems led to fourth generation (4G), also known as LTE (Long-term

Evolution) (Arshad, Farooq and Shah, 2010). 4G was developed to improve services offered by 3G and promised to deliver, wireless services at any time, to anyone and anywhere, communication speeds higher than 3G, amplified performance in streaming multimedia based content. 4G was intended to act as a panacea to limited bandwidth in 3G (Azira and Omar, 2013).

Mobile phones, equipped with various computer-like functions and services have become an integral part of communication and computing mainly because of their resourcefulness. Diverse functionalities of a camera, accelerometer, barometer, gyroscope, global position system (GPS), fingerprint sensors, glucometer, heart rate sensors, message service, can be used to build a myriad of customized applications and services to allay a specific problem (Lane, Miluzzo, et al., 2010). Research organizations and individual researchers are exploring many ways to leverage on these diverse collection of sensors embedded in mobile phones to deliver applications across diverse domains for social impact, environmental protection and monitoring, homecare, healthcare, safety, e-commerce and transportation, and to leverage just-in-time information to make our movements and actions more environment friendly (Khan, Xiang, Aalsalem, et al., 2013).

Advancements in mobile technology can be seen as a gateway to manifestation of opportunities that may in different ways uplift the economy and social wellbeing of people. This is evident in the proliferation of mobile services that are gaining attention such as: (1) **mHealth** is a platform for providing health-care services through mobile programs that can monitor and report peculiar medical conditions. Medical practitioners can monitor, assess and help patients improve their health in real time, and this has the potential to reduce the cost of healthcare (Kumar, Nilsen, Pavel, et al., 2013). (2) **mEducation** is the use of mobile phones to deliver educational content, to facilitate informal learning in out-of-school environments so as to complement formal schooling (Kumar, Tewari, Shroff, et al., 2010). (3) **mMarketing** refers to the use of mobile phones as an innovative channel for transmitting advertising or marketing messages to consumers (Liu, Sinkovics, Pezderka, et al., 2012).

Mobile phones are perhaps the most influential Information Communication Technologies (ICTs) tool when it comes to monitoring and tracking applications in the developing countries of Africa. One pointer to this fact is that Africa has achieved an average mobile phone penetration

level of 69% (ITU, 2014), which is considerably above that of computers. Figure 2.5 below highlights these statistics. Mobile penetration has also increased rapidly in Sub-Saharan region of Africa, from just 1% in 2000 to 54% in 2012 (GMS and Deloitte, 2012). Much of this growth has been attributed to increased affordability of mobile services over the past years. The latter is partly triggered by an array of factors, including increased competition in a number of markets, equipment price cuts in terms of handset prices and mobile networks as well as the growing scale of mobile network operators. Mobile network operators are working at full throttle to boost investments in infrastructure to augment overall network coverage as well as introduce new service offerings that entices average to low-income earners (Sub-Saharan Africa Mobile Economy, 2013).

African Countries	Mobile-cellular telephone subscriptions per 100 inhabitants													
	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Botswana	12.66	18.63	18.37	24.28	28.19	30.06	43.41	60.14	76.84	96.02	120.01	145.98	153.79	160.64
Congo (Rep.)	2.24	4.68	6.75	9.81	11.12	15.76	25.16	34.26	46.61	73.80	90.44	91.94	98.76	104.77
Côte d'Ivoire	2.93	4.44	6.16	7.57	9.77	13.51	23.02	41.61	57.22	70.88	82.20	89.45	91.23	95.45
Egypt	2.06	4.16	6.58	8.35	10.83	18.99	24.66	40.54	54.69	72.10	90.50	105.08	119.92	121.51
Ghana	0.69	1.26	1.95	3.92	8.14	13.44	23.73	33.76	50.07	63.77	71.87	85.27	100.99	108.19
Kenya	0.41	1.87	3.60	4.69	7.31	12.89	19.97	30.06	42.05	48.62	61.03	66.81	71.17	71.76
Morocco	8.16	16.44	21.15	24.88	31.27	41.14	52.66	65.31	73.71	80.93	101.07	114.02	119.97	128.53
Namibia	4.32	5.52	7.66	11.29	14.28	22.14	29.66	38.46	49.84	76.12	89.50	98.96	95.02	118.43
Nigeria	0.02	0.21	1.21	2.38	6.73	13.32	22.55	27.45	41.66	47.96	54.66	57.96	66.80	73.29
South Africa	18.59	23.70	29.67	35.97	43.82	70.40	81.08	85.28	89.52	91.25	97.90	123.20	130.56	145.64
Zimbabwe	2.13	2.49	2.68	2.87	3.35	5.09	6.67	9.62	12.94	30.96	58.88	68.87	91.91	96.35

Figure 2.5: Mobile cellular subscriptions (ITU, 2014)

2.5. Middleware

The feasible operation of an integrated IoT architecture that facilitates interoperability and communication between heterogeneous or homogeneous IoT objects can be realized by a lightweight software layer or a set of sub-layers interposed between the technological and the application levels known as middleware (Atzori, Iera and Morabito, 2010). Hwang and Yoe (2011) defined middleware as a software that “*supports flexible integration of hardware and application and provides services such as distributed computing environments, remote procedure calls, messaging to users, regardless of the hardware, operating system and network*

used”. Atzori et al. (2010) pointed out that middleware has been gaining traction due to its ability to simplify development of new services and the integration of legacy technologies into new ones, while exempting programmers from knowing diverse technologies implemented at lower layers.

2.5.1. Middleware for Internet of Things

Being a fairly new paradigm, the definition of IoT is still elusive; it has even acquired different names such as “*the internet of everything*” (Cisco, 2014) and “*the Semantic Web of Things*” (Brock and Schuster, 2006). The common departure, however, is the concept of ‘things’ which are the entities people are concerned about; more importantly, however, these things should be identified as part of the internet. Both solution designers and researchers acknowledge the fact that the nightmare brought by the sheer number (could potentially be in millions) and the heterogeneity (with respect to the nature of the components, standards, data formats, protocols among others) of these ‘things’ is better addressed by middleware architectures.

These architectures should be able to support self-management of the systems, that is, support: self-configuration, self-optimization, self-protection, and self-healing (Kephart and Chess, 2003). This ensures that irrespective of the application domain, the generic salient characteristics of IoT are maintained; some of these, as explained by Rodríguez-Molina, Martínez, Castillejo, et al. (2013) are: (1) omnipresence which is realized via ubiquity (and explained in the 4As vision of IoT (ITU,2008)); (2) calmness – that is, not disturbing the equilibrium/environment as perceived by users; in other words, the interaction with the users should be as natural as possible and unconscious.; (3) reliability – ensuring no significant interruptions, continuity and self-healing; (4) security – ensuring confidentiality similar to other conventional systems; and (5) Ambient Intelligence (AmI) that enables the system to be ‘aware’ and also ‘recognize’ situations.

2.5.2. Classification of Middleware Approaches for WSN

So far, it is the WSNs and their associated technologies that play the greatest role in realizing IoT characteristics above. It is not accidental therefore that most middleware architectures in place so far are designed around WSNs. From this perspective, the middleware acts as an insulator that hides the internal workings of the system by providing homogenous and abstract environment to

the highest layers (either application consumers or application developers) (Chatzigiannakis, Mylonas and Nikolettseas, 2007). Hadim and Mohamed (2006), on the other hand, view middleware as the glue that sticks together the networking hardware, operating systems, network stacks, and applications. According to the duo, successful middleware must address a whole lot of challenges posed by both the WSNs (and their salient characteristics) as well as the applications running on them (WSNs). These include: (1) managing limited power and resources; (2) scalability, mobility, and dynamic network topology; (3) heterogeneity; (4) dynamic network organization; (5) real-world integration; (6) application knowledge; (7) data aggregation; (8) quality of service; and (9) security.

Traditionally, middleware was introduced in distributed applications development to act as a broker between the operating system and the application. Given the similarities between WSNs and the traditional distributed systems, it would seem that the already established middleware solutions such as CORBA, ICE, SOGI, OMG and web services could be used; however, they have been found to be inadequate in meeting IoT and WSNs requirements (Noguero, Calvo, Perez, et al., 2013). Consequently, tens of middleware solutions specific to WSNs have been developed; these can be categorized into three: (1) in-network middleware that is uploaded on the nodes; (2) server-side middleware that runs on a server; and (3) a hybrid that combines both 1 and 2 (Hwang and Yoe, 2011). Another categorization that is based on programming models that the architecture uses produces two broad classes: *programming support* and *programming abstractions* (Hadim and Mohamed, 2006). Further sub-categorization of these two is illustrated in Figure 2.6.

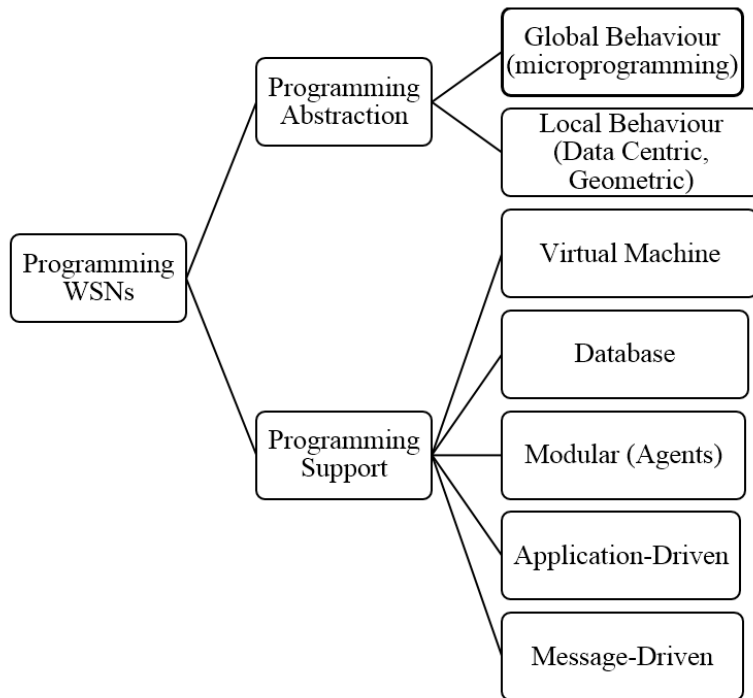


Figure 2.6: Taxonomy of WSNs middleware (Hadim and Mohamed, 2006)

In the database approach, the entire system appears as a virtual database to which the users can issue SQL-like queries to extract data of interest. It is mostly implemented in the form of a relational database; this is in fact its main weakness because it fails to support spatio-temporal relationships among data elements. Examples include; Cougar (Cornell Database Group, 2014), TinyDB (Bonnet, Gehrke and Mayr, et al., 2001), *SINA* (System Information Networking Architecture) (Srisathapornphat, Jaikao and Shen, 2000) and *DsWare* (Data Service Middleware) (Li, Son and Stankovic, 2003).

The second category that is relevant to this research is the Application-driven class of middleware which allows the programmers direct manipulation of the WSNs based on the specific requirements of the applications. This leads to the drawback that the middleware and the application are tightly coupled (cannot be generalized easily). However, it offers some advantages such as assured quality of service. A good example here is the *Milan's* (Middleware Linking Applications and Networks) (Heinzelman, Murphy and Carvalho et al., 2004).

In Heinzelman, et al. (2004), the following (Figure 2.7) classification of the relevant (to WSNs) middleware architecture is presented:

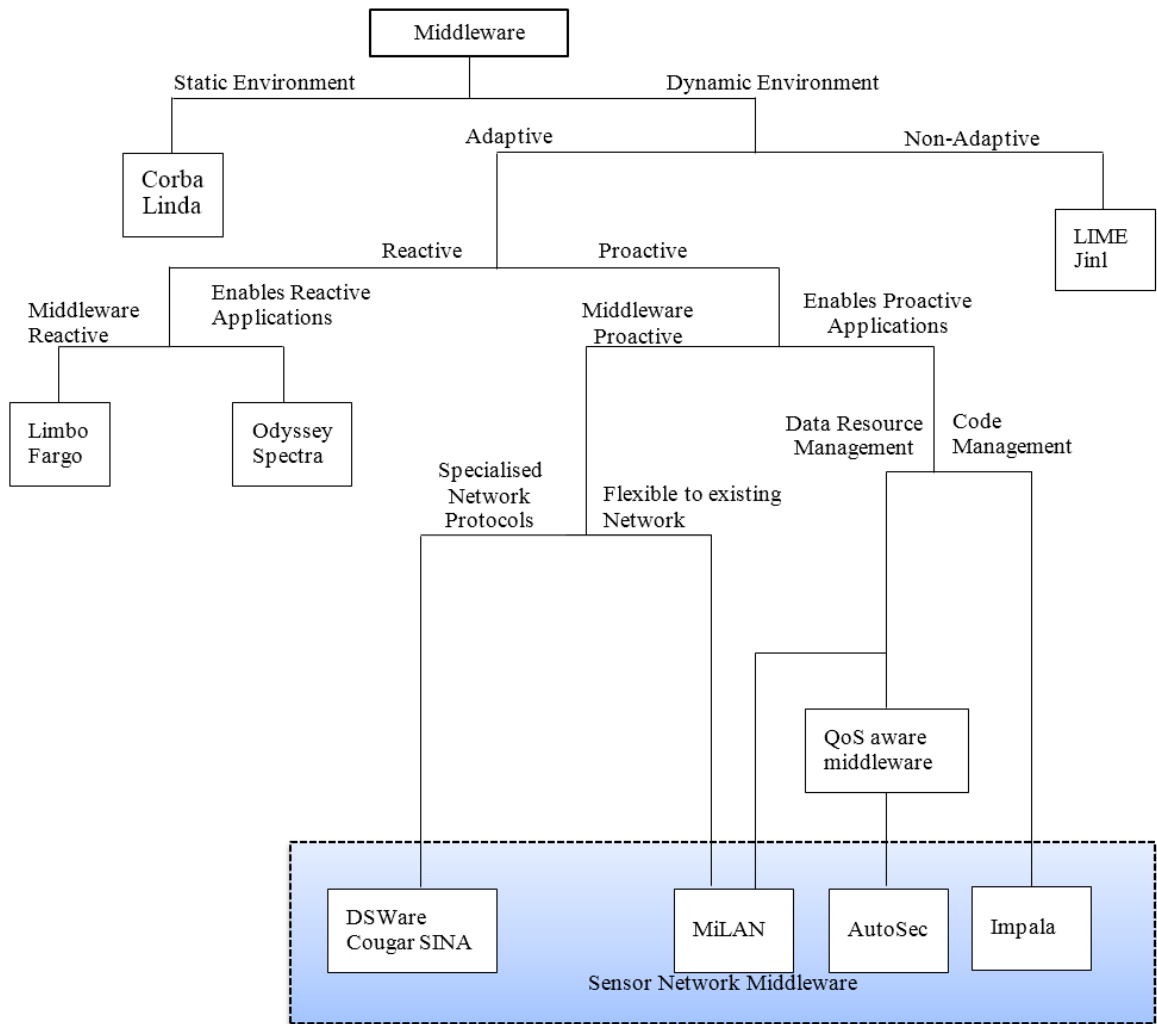


Figure 2.7: WSNs middleware architecture (Heinzelman, et al., 2004)

2.5.3. Ambient Intelligent and Cyber-Physical Systems

One of the features expected in location-based service IoT applications is Ambient Intelligence (AmI); this is the ability to be sensitive, adaptive and responsive to changes in the physical environment (Coronato, 2012). AmI systems are now required to be time-sensitive and resource-aware. We increasingly witness AmI systems that combine sensors and actuators with interactive and sensing multi-media traffic; these are commonly known as Cyber-Physical Systems (CPS). The main operations of AmI applications are: to receive the state of the environment, execute

reasoning algorithms and then act upon the environment in an adaptive and ubiquitous way. This is analogous to the four components of an early warning system presented earlier in Figure 1.2; except that the last two components are merged into one. To add on to the complexity of AmI, CPS must be able to synchronize actions of several distributed devices and resource management (Noguero et al., 2013). Given their need for tighter timing, safety, security and robustness, asset tracking and monitoring systems (the focus of this research) qualify to be classified under AmI.

There are no known (to authors) database-application-based IoT/WSNs middleware solutions targeted to this domain. Conversely, due to its criticality (deals directly with human life), the healthcare application area has witnessed a proliferation of AmI applications. This has been matched by nearly equal number middleware solutions. Although not as conspicuous as in healthcare, smart-home applications domain has also received substantial attention. Noticeable also is the fact that successful middleware architecture tend to be service-oriented and recently, semantic-oriented. Further, the solutions tend to incorporate a database in one form or another. Some of these middleware solutions (considered relevant to this research) are explained below:

Uranus (Coronato, 2012) is an elaborate service-oriented middleware architecture meant for integrating different kind of biomedical sensors; it supports the development of ambient assisted living and vital signs monitoring applications. It has five main services for handling environmental sensing, communication, human computer interface (HCI), context and correctness. The middleware has been tested in two real-life applications: one for monitoring of the oxygen on chronically ill (not hospitalized) patients and the second one is used in a hospital's nuclear medicine department to monitor patients who have been injected with radioactive substance.

FTT-MA (Flexible Time-Triggered Middleware Architecture) (Noguero et al., 2013), on the other hand, is a high-level middleware architecture and that supports (through a methodology and design tool) design, development and operation of intelligent CPS by ensuring that all (both functional and non-functional) system requirements are met.

2.5.4. Cornell Cougar Sensor Database System

A Cougar middleware is based on the database middleware approach in which sensor data is considered to be a virtual relational database. Cougar uses SQL-like language and implements WSN management operations in terms of queries (Bonnet, Gehrke and Seshadri, 2001).

There are two approaches for processing sensor queries: (1) warehouse approach, and (2) distributed approach. In warehouse approach, processing of sensor queries and access to the sensor network are separated. First there is data extraction from sensor networks, followed by transmission of data to a central database server where data is accumulated and stored for querying and examination. This approach has two major drawbacks: it lacks dynamism and flexibility since users are restricted from calibrating system performance. Secondly, transmission of huge amounts of data from sensors is risky due database communication failures (Yao and Gehrke, 2002).

In distributed approach sensors in the sensor network run a dedicated software known TinyDB or Cougar. TinyDB has many of the features of a traditional query processor (e.g., the ability to select, join, project, and aggregate data) (Gehrke and Madden, 2004). In Cougar, signal processing functions are represented as Abstract Data Type (ADT). Virtual joins are an effective way of executing ADT functions that do not return a value in a timely fashion because they are often asynchronous, because they incur high latency or because they return multiple values over time (Bonnet et al., 2001). Figure 2.8 depicts the query processing architecture for Cougar based wireless sensor networks. In this query architecture, programs are highly distributed and the operating system must cautiously manage energy consumption and radio bandwidth while sharing information and processing.

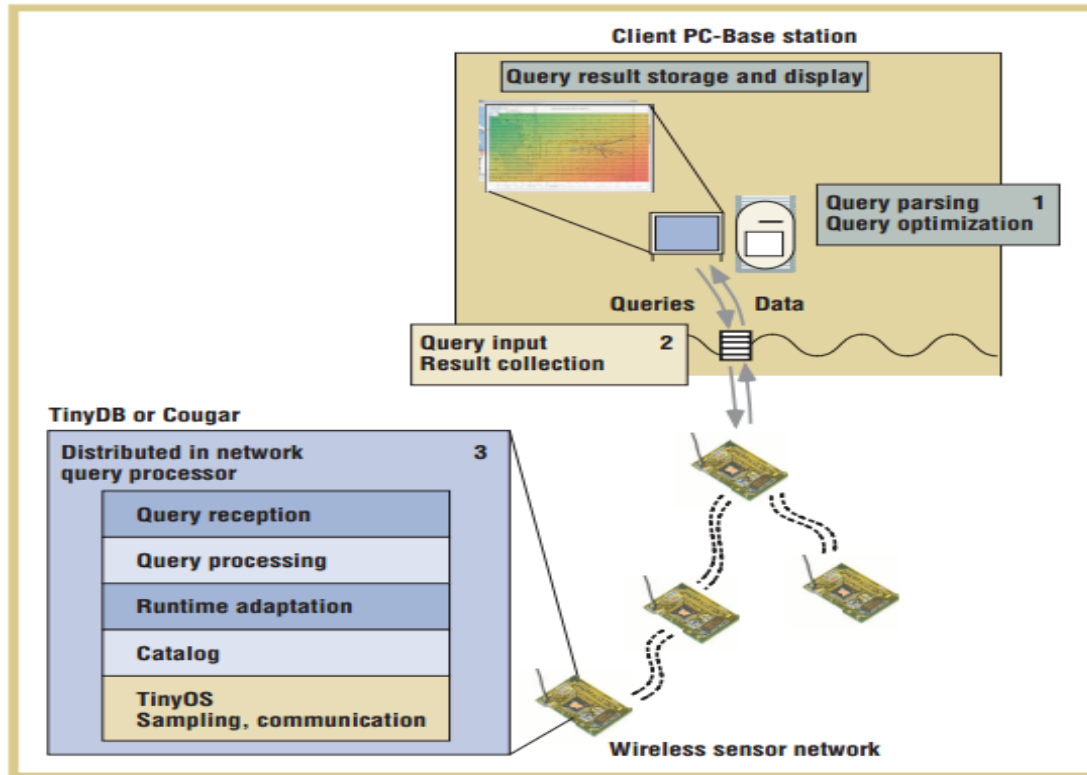


Figure 2.8: Query processing architecture (Gehrke and Madden, 2004)

In Cougar middleware, sensors have a query optimizer that defines the data-querying strategy, in-network query processing and this includes decisions about the data that should be extracted from sensors (Bonnet et al., 1999). This technique is efficient and preserves computing resources on sensor node which ultimately prolongs the lifetime of the sensor network (Yao and Gehrke, 2002).

2.5.5. Middleware Linking Applications and Networks (MiLAN)

MiLAN also regarded as an adaptive middleware implements an application driven approach which brings forth a new aspect in the middleware design by appending an architecture that glues application execution with network protocol stack (Nikolić, Penca and Segedinac, et al., 2011) and tackles very well the challenges of QoS requirements.

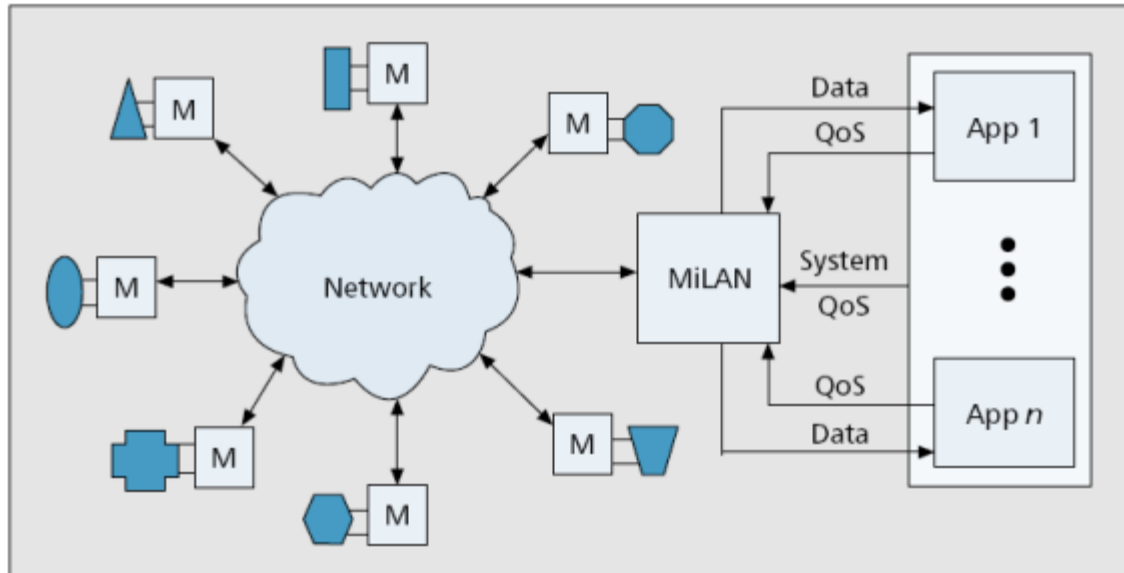


Figure 2.9: High-level diagram of a system that uses MiLAN (Heinzelman et al., 2004)

Figure 2.9 depicts the overall employment of MiLAN. The letter ‘M’ inside squares represent the MiLAN running on a sensor. Attached to sensor are other kinds of sensors such as pollution, magnetic field or humidity. The right side of the diagram shows applications specifying their QoS needs and the middleware is responsible for the delivery of the requested data.

MiLAN middleware permits applications to submit a query and stipulate their sensing and QoS needs. MiLAN generates an execution plan, which: (1) adjusts the network parameters to lengthen application lifetime while sustaining provision of quality needs; (2) fine tunes and manages the network load through dictating data source nodes and routing nodes with the most energy efficient paths, that fulfils the QoS obligations while maximizing energy efficiency (Faghih and Moghaddam, 2011). However, MiLAN’s architecture suffers from hardware heterogeneity in the sense that it supports communication with hardware that tolerates its network protocol stack.

2.5.6. Semantic Middleware for IoT

According to Kephart and Chess (2003), complex environments need computing systems capable of running themselves with minimal human management. A new generation middleware platform (UBIWARE) allows: (1) creation of self-managed; (2) self-configuration; (3) self-optimization; (4) self-protection; and (5) self-healing in complex industrial systems consisting of

distributed, heterogeneous, shared and reusable components of different nature, e.g. smart machines and devices, sensors, actuators, RFIDs, web-services, software components and applications, and humans (Katasonov, Kaykova and Khriyenko, et al., 2008).

Semantic Middleware Architecture provides several significant contributions not available to other types of middleware that have been postulated by other researchers. According to Rodríguez-Molina, Martínez and Castillejo, et al. (2013), there are several modules that make up the semantic architecture which include: (1) Ontology Module: integrates any new vocabulary related to new services that may spring up as the result of adding new devices (smart meters, home loads, etc.); (2) Repository Module: involves semantic storage of data that will be required for different tasks; (3) Services Module: deals with the services that are present in the middleware architecture and is scalable enough to support addition of new ones; (4) Resources Module: flexibly manages all the heterogeneous hardware infrastructures required to harvest data from the context; and (5) Inference Engine Module: provides the semantics required to treat the information flowing through the middleware layer and works closely with the ontology module.

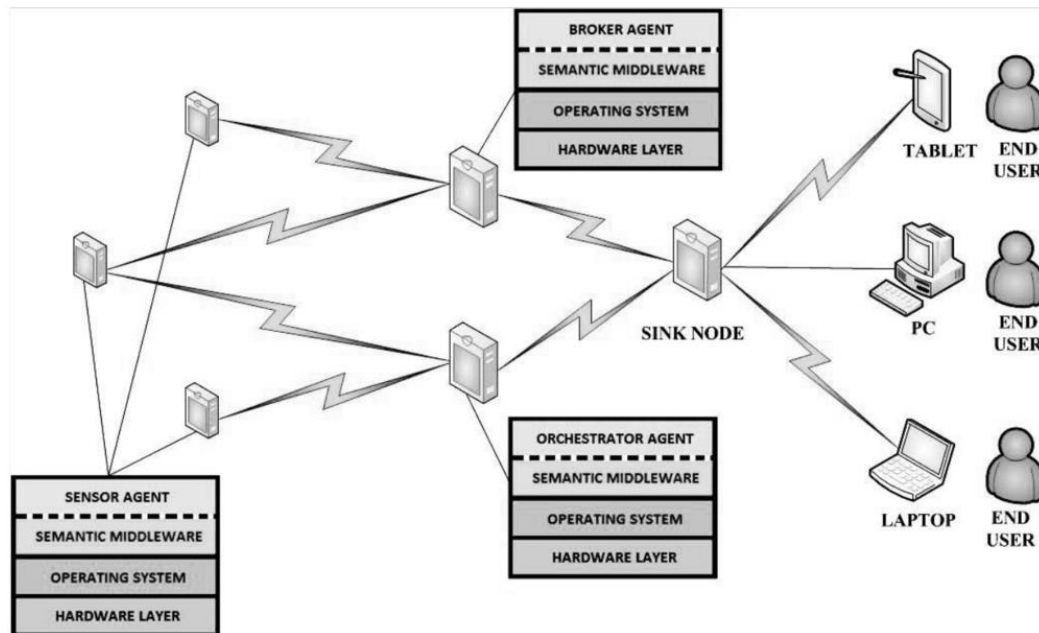


Figure 2.10: Semantic middleware architecture within WSN (Rodríguez-Molina et al., 2013)

The nodes presented in Figure 2.10 are configured and deployed with tiny software agents that enable them to perform designated tasks. These software agents uploaded on the sensor, determine activities confined to a sensor node; however it is possible for one node to have

multiple functions; because several software agents can be uploaded on a single node, switching between agents, depending on the context (Rodríguez-Molina et al., 2013).

However, according to Chaqfeh and Mohamed (2012), UBIWARE will remain a research prototype without an implementation potential, until an adequate security infrastructure is embedded into it.

2.5.7. IoT Middleware Implementation Challenges

None of the integration approaches, including common approaches such as service-oriented architecture (SOA) and emerging approaches such as the ‘*HOMEROS*’ middleware architecture, which adopts a hybrid-network model to efficiently manage enormous resources, context, location, and various services (Han, Yoon and Youn, et al., 2004), are a simple and seemingly magical solution to complications associated with integration, interoperability, compatibility and parallelism in IoT applications.

A **dependable and responsive** IoT middleware that is capable of handling and servicing numerous devices, process volumes of data without compromising responsiveness is one challenge that many middleware solutions are failing to address. Applications built around the IoT paradigm need to **scale up** from time to time; the success of a middleware is measured by its ability to effectively handle various events from dynamic connected objects and timely process raw data much faster in order for action to be taken. A reliable IoT middleware must handle small-scale to large-scale scalability without compromising the performance and responsiveness of the middleware (Paridel, Bainomugisha and Vanrompay, et al., 2010).

Integration is characterized by differences in: (1) application coding language; (2) data formats; (3) deployment models; (4) computing device architecture; (5) operating systems; and (6) the nature of activities involved; these differences present complexity in integration that does not yet have a vibrant solution. Diverse data from heterogeneous devices need to be examined by the middleware layer, such that preconfigured actions such as triggering alerts or invoking corrective responses will be taken based on the interpretation of that data (Hadim and Mohamed, 2006). Due to integration problems, there is no assurance that these actions will be set in motion before things turn into a chaotic event.

Performance in flexible middleware systems depend on capture and indirection mechanisms (the ability to reference something using a name, reference, or container instead of the value itself), which induce performance degradation. This delinquent may be alleviated by numerous optimization mechanisms, which aim at eradicating the needless overheads by such techniques as **middleware in-lining**, which is injecting the middleware code directly into the application (Krakowiak, 2009).

Data management is an integral part of IoT middleware. There are billions of devices connected to the internet and these IoT objects generate mammoth amounts of data such as identification data, positional data, environmental data, historical data and descriptive data. Managing these voluminous data is a prime responsibility of IoT middleware. This results in floods of data traversing through networks, being collected, analysed and processed. It is therefore imperative that novel and subtle approaches to find, fetch, decode and transfer data efficiently be developed, to properly iron out challenges with data encryption, data querying, indexing, process modelling and transaction handling (Bandyopadhyay and Sen, 2011).

2.6. Related IoT Applications

Object traceability is a paramount aspect that IoT is currently addressing. The proliferation of affordable wireless sensors and RFID technology triggered a great deal of interest in object tracking algorithms. Numerous approaches for object tracking and monitoring have been proposed and implemented, but most have been out-performed by the capabilities of WSNs and RFID discussed in 2.4.1 and 2.4.3 above. Traditionally, barcode scanning was the prominent technology used in collecting data in an automatic and contactless manner (Arendarenko, 2009). Barcode scanning devices are used to perform data reading from barcodes but for this reading to work, the barcode scanner must be positioned precisely near the barcode. Compared to manual tracking processes, barcode was seen as a big break-through and a preferred faster approach to tracking objects because it was economical to install and read data from.

Despite heavy adoption worldwide, barcode technology is failing to meet up with huge business demands. Barcode technology is gradually getting out-dated because of: (1) limited storable amount of information, which is no longer adequate to support many complex business applications; (2) read errors attributed to damages suffered by barcodes; and (3) the inability of

barcodes to support multiple out of sight reading. RFID technology has become the leading object identification and tracking technology due to advantages such as contact-less, multi-object recognition, non-line-of-sight, long distance, large store memory, programmability and penetrability (Zhang, Ouyang and He, 2008). RFID has been used not only in supply chain and logistic business but in aviation industry as well. Zhang et al. (2008) developed an RFID-based system to support baggage handling and baggage tracking. They suggested that baggage at the airport, may not only be assembled, as well as checked more precisely, but may also be traced across the entire world.

Moreover, DHL has successfully deployed a system called DHL thermonet, which basically is an RFID-based airfreight service that allows customers to track temperatures of sensitive pharmaceuticals or biomedical products throughout the shipping process. According to DHL, if a temperature discrepancy is detected, the technology will help DHL identify the problem faster, and thus address that issue before other goods are damaged or, at least, in time to save the goods inside the carton whose tag has measured excessively warm temperatures. In addition, such an action not only saves the cost of that product, DHL explains, but also saves the customer from a potential loss in sales if goods were unable to be delivered in a satisfactory condition at the expected time (Swedberg, 2014).

The success of object tracking and identification technology facilitated by WSN and RFID is improving security in business and industrial environments through improved efficiency, fast and accurate data capturing, access and processing, real time status updates, and all this data helps to secure mobile assets and saves companies from asset loss. This fact is supported by a success implementation and deployment of a Mobile Asset Tracking System (Kim, Jo and Lee, et al., 2012). Kim et al. (2012) developed tracking system that supports both legacy WSN services and management of mobile assets that can be tracked simultaneously. They proposed a network architecture that was divided into three tiers: (1) Mobility Tier – supports bi-directional location awareness and asynchronous message delivery; (2) Sensor Network Tier – provides a reliable network implementation for stationary nodes; and (3) Backbone Network Tier – uses coordinating gateways that converts WSN messages received from stationary nodes into TCP/IP messages for the server side middleware and vice versa. A middleware was used to manage real time locations of mobile assets, facilitate routing of messages and asynchronous delivery of

messages. This application was tied to a database stored on a server that stored assets location data. A smart phone was used to access services, identify asset location and transmit service specific messages used to control or trigger predefined actions (Kim et al., 2012).

Another tracking application of interest is a Child Tracking System designed by Chen (2010), which accurately locates the geographical position of children playing in a park, with the intention of easily helping parents find a lost child. This system integrated three components which include: (1) RFID technology – children carried with them passive tags which were powered by radio waves coming from readers positioned around the park to reveal identifying data; (2) WSN – was used to gather data in each RFID reader and transmit it to a remote base station; and (3) Motions detection sensors – were used as energy conservation mechanism in the sense that they were commanding RFID readers to sleep when no human activity is detected or activate if movement is detected (Chen, 2010).

The resulting system was capable of locating children within the RFID reader's read range in real time, perform remote data collection with the help of wireless sensors and limit power consumption, therefore extending the lifetime of the system (Chen, 2010).

2.7. Summary of IoT Technologies

Table 2.3 below summaries the literature review discussed throughout this chapter. The table presented below reveals where and how the versatile middleware proposed in this study inherited its features, which were used to develop the laptop monitoring and tracking tool. The Table is structured as a 4-column table with the first column identifying the technology along with its supporting paper reference(s); the second column explains the benefits of the technology; the third column details known disadvantages of the technology; and the fourth column briefly discusses how the technology used in other papers is being adopted and adapted to address the envisioned requirements of the laptop monitoring and tracking system.

The table below summarises the details of the hardware that supports the development of the envisioned system and also the middleware approaches whose revered features were be used to create a hybrid versatile middleware that was used to mask the complexities of heterogeneous hardware.

Table 2.3: Summary of IoT supporting technologies

Technology	Strength	Limitations	Closeness to the Research
Wireless Sensor Nodes (Culler-Mayeno, 2006)	Wireless sensors can access the internet and transfer gleaned data to the base station. The base station may also wirelessly give directives to nodes over the internet to control or update their behavior.	The threat of hackers is a serious problem because the operating system for the sensor motes is an “open-source” system, which allows relatively easy access to and manipulation of code.	The bi-direction communication between the base station and wireless motes is an important aspect in security based systems. This allows easy control of sensor motes in real time.
Radio Frequency Identification (Finkenzeller, 2003)	RFID systems are non-contact and do not require line-of-sight to work and can thus be used in visually and environmentally challenging conditions.	Tags with high read range are bulky and battery powered. Passive Tags are not battery powered and have short read range. The placement of tags on the objects to be identified has a direct impact on the RFID system’s interrogation abilities.	The non-contact communication between the tags and readers can be used to facilitate identification of stolen laptops and to initiate automatic monitoring of tagged laptops once they get into the scanning zone.
Biometric Scanners (Fingerprint) (Jain and Kumar,	Biometrics in security context is a mechanism that	The known problem with functionality of Fingerprint	This fool proof identification process could be used to

2010) and (Bhargava, Bhargava and Mathuria, et al., 2012)	uniquely identifies people by comparing distinctive physical characteristics.	Verification System is: it depends on the quality of image. The quality of fingerprint image is affected by both physical damage, scanner's-surface and comparison algorithm	control various functions of the envisioned LMTS functions, particularly turning off monitoring of the laptop, by the owner to stop triggering security breaches.
Mobile Phones (Khan, Xiang, Aalsalem, et al., 2013) and (Lane, Miluzzo, Lu, et al., 2010)	Mobile phones are equipped with various computer-like functions and services, which can be harnessed to deliver diverse applications which can have direct and positive impact on the socio-economic growth of African countries.	Mobile phones are energy dependent and expensive to operate. They are difficult to operate in areas with limited network reception. High theft rates are another cause for concern although plans are in the pipeline to combat this using "Kill Switches", according to Apple.	Mobile phones can be useful in this research in delivering real time security breaches to laptop owners and security guards and may also be used to control various LMTS functions remotely using SMS.
Cougar Middleware (Bonnet, Gehrke and Seshadri, 2001), (Bonnet et al., 1999)	Enhances application's flexibility due to elimination of a centralized system for collecting sensor data. In Cougar	Cougar is concerned with power conservation and providing query processing strategies that aim to conserve resources. A key	The approach allows users to issue queries in a declarative SQL-like language. This feature can be harnessed to handle queries from users

	middleware, sensors have a query optimizer that defines the data-querying strategy, in-network query processing and this includes decisions about the data that should be extracted from sensors.	limitation of Cougar based systems is the assumption that sensor nodes are largely homogeneous and therefore they are not suitable for heterogeneous sensor networks.	issued particularly from the mobile phones. Stored procedures can be written to periodically extract laptop's geo-position and relay this data to a database.
MiLAN Middleware (Nikolić, Penca and Segedinac, et al., 2011), (Faghih and Moghaddam, 2011) and (Rodríguez-Molina et al., 2013)	Applications submit a query and specify their sensing and QoS requirements to the middleware in terms of graphs describing sensor quality of service and state-based variable requirements. In response to a query, MiLAN creates an execution plan, which specifies the source nodes and the routing tree, such that satisfies the QoS requirement while maximizing energy	Under MiLAN, networks must be configured in a very accurate way, due to the fact that the group of nodes is chosen by making use of its extended architecture. MiLAN presents a mostly application layer-focused middleware architecture.	The features of MiLAN can be used to identify, execute and manage the various middleware services such as location, data management and SMS communication. LMTS users can issue various query commands from either the system or mobile phones. The execution decisions can be managed by adopting decision making features of MiLAN and the

	efficiency.		responses to those queries.
Ambient Intelligence (Coronato, 2012) and (Cook, Augusto and Jakkula, 2009)	This is the ability to be sensitive, adaptive and responsive to changes in the physical environment.	Ambient Intelligence lead to invasion of privacy due to invisible and full wireless surveillance network, sharing an unparalleled amount of sensitive data about our public and private lives.	Context (location and activity) awareness is paramount to building Ambient Intelligence based and interactive security applications. With this feature GPS and GPRS hardware can be utilized to accurately generate and communicate laptop's locus data.

3. Chapter 3: Methodology

This chapter presents the research approach, the empirical techniques that were applied as well as the logical norms sustaining this research. This chapter explains in detail the scope and boundaries of research design, data collection and analysis methods, explaining the stages and processes involved in the development of the middleware and system prototype. The researcher also introduces a case study and discusses the results of investigations carried out. Lastly, the details of software development method used, and how this development was conducted, followed by a justification of the chosen method, are provided.

3.1. Introduction

The term ‘Research methodology’ is a concatenation of the word ‘research’ meaning a scientific endeavour to study a unique idea undertaken to probe a phenomenon which helps in understanding it better and finding applicable tools and techniques to find a panacea to the plight under investigation (Plano Clark, 2005). Research compels researchers to go beyond personal knowledge, experience, feelings and opinions, which ultimately helps researchers to conduct investigations, observe, analyse and comprehend findings and draw up conclusions. The word ‘methodology’ means logical approach to be used and followed to arrive at a solution to allay the identified research problem. According to Kothari (1988), research comprises *“defining and redefining problems, formulating hypotheses or suggested solutions; collecting, organizing and evaluating data; making deductions and reaching conclusions; and finally, carefully testing the conclusions to determine whether they fit the formulated hypotheses”*.

3.2. Research Methods and Design

According to Plano Clark (2005), there are two predominant types of research paradigms, namely: (1) quantitative and (2) qualitative paradigms. Quantitative model is based on the measurement of quantity of some phenomena that can be expressed in terms of quantity; this measurement should be objective rather than subjective and statistically valid according to Kothari (2009). The qualitative model is based on gathering, scrutinizing, and interpreting data by observing some phenomena, while the quantitative model dwells on amounts and

measurement of things, the qualitative model dwells on a thorough understanding of definitions, concepts, characteristics, metaphors, symbols, and descriptions of things (Berg, 2007).

Constructive research approach (CRA) is another model on the horizon in the fields of engineering and information system, which slowly gained momentum in the research and design spectrum over the years (Gregor, 2006; Jones and Gregor, 2007). The main focus of CRA is generation of new knowledge that can be applied in solving real-world problems and using that newly acquired understanding of a phenomenon to patch up missing links in pre-existing knowledge (Crnkovic, 2010). CRA can be seen as a way of reducing the gap between science and practice (Labro and Tuomela, 2003). CRA could be considered as virtually a hybrid research paradigm since it tries to collate theoretical, technical and practical interests of knowledge in a single research study.

The constructive research model inherits most principles from well-understood research roots; that is to say, it often employs both qualitative and quantitative methods in conducting empirical investigations, prior commencing artefact construction. This is done to acquire broad knowledge and understanding of the phenomena being studied (Crnkovic, 2010).

A research design (RD) constitutes the blueprint for the accumulation, measurement and thorough scrutiny of collected data; this influences the researcher in making informed decisions with the allocation of scarce resources. Firstly, RD aids in minimizing the expenditure of time, monetary resources by stipulating well in advance the direction the research is to take. Secondly, having a well-articulated blueprint or map helps to smooth up the flow of research operations, thereby aiding the researcher not to plan to fail, because having no plan, is a good recipe for disaster. Thirdly, a RD provides an overview of all the research processes, which can also help to elicit insightful and well thought out perceptions of experts from that research arena and ultimately aids in achieving the objectives for the study without any hassles or overlooking other critical research processes.

3.3. Research Methodology Used

This research revolved around the CUT's laptop theft-deterring system, which is a real-life problem; CRA was deemed best. In this case, a system prototype for tracking assets at CUT was

developed and used to test the functionalities of the design. A novel middleware integration architecture was designed and implemented with the intention to intelligently and automatically monitor and track CUT's laptops in a flexible, secure, self-organizing and user friendly manner. In order to increase the prospect of being practically feasible, the researcher ensured that the innovated construction was relevant, simple and easy to use (Kasanen, Lukka and Siitonen, 1991).

Experimental design was applied in testing the practicality of a prototype system. The reasoning behind experimental design is its ability to: (1) produce tangible results that minimize bias; (2) prove functionality and reliability of the developed construct; (3) provide practical learning experience through scrutiny of the prototype; and (4) draw unambiguous conclusions that convincingly answer the research questions posed in the dissertation.

Constructive research implies a very close involvement and co-operation between the researcher and practitioners in a team-like manner (Lukka, 2000). Here, experiential learning is expected to take place. The data-gathering methods used are: (1) document analysis of data pertaining to asset audit (an extract of the documents analysed can be found in Appendix 9); (2) interviews with the Head of Security and victims of laptop theft; this enabled the researcher to collect information on the key security aspects that needed urgent attention, as well the main weaknesses of the current security systems deployed by CUT.

The victims of asset theft were identified from analysing documents used by the security department to record asset theft cases. An excerpt of interview questions can be found in Appendix (Preliminary investigation questions 1-2); (3) questionnaires were issued randomly, – each member of the population having an equal chance of being selected (Teddlie and Yu, 2007) – to potential users of the laptop monitoring and tracking system during a case study investigation. This was done with the intention of understanding the magnitude of laptop, tablet and cell phone theft and later to probe users' perceptions about the system prototype. The questionnaire used can be found in Appendix (Questionnaire page 1-6); and (4) experiments were conducted during the testing and evaluation phase of the resulting system prototype.

3.4. CUT Case Study

The Central University of Technology (CUT) is not spared from the laptop theft catastrophe; from the investigations conducted, it was discovered that CUT was insecure and that it was still in a vulnerable state. This is partly because of the non-existence of an integrated intelligent security system to dissipate and mitigate loss of students' and institutions' laptops and indeed other assets. A preliminary investigation uncovered that CUT delegated asset monitoring to security guards positioned at entry and exit points. According to unstructured interviews conducted with some of the security personnel, it was uncovered that these security personnel are not mindful of activities that take place within buildings until a formal complaint has been reported. For example, by the time they become conscious of a stolen asset, the asset is long gone and out of CUT premises, making recovery nearly impossible.

According to statistics from asset audit documents and records made available to the researcher by the Asset Management Department at CUT. The institution spends a great deal of money (approximately above ZAR150, 000 a year) to replace stolen laptops and data projectors. This total monetary value of lost assets is also corroborated by ZAR152, 419.00 worth of laptops, cell phones and tablets stolen from both students and CUT personnel; the latter monetary figure is according to data gleaned from the questionnaires.

This money could have been channelled elsewhere and helped to develop the institution. Not only money is lost but valuable company and student data that may compromise the integrity of the institution, if it gets into wrong hands. Several incidents of this nature have been reported but little has been done to alleviate or mitigate this escalating problem. Rogue students and staff members take advantage of unavailability of security surveillance cameras or any monitoring mechanism and sneak into unattended offices, steal valuable institutional assets and get away with them. The question is, how many more assets are being stolen that the Asset Management and Security Department at CUT is not aware of?

CUT has a number of security measures in place which include the following: (1) Physical security personnel that are employed to ensure proper execution of access control management policies and procedures. (2) Surveillance cameras; CUT deployed closed-circuit television (CCTV) surveillance systems in some of the institution's buildings. (3) Intrusion detection

systems (IDSs) for detecting unauthorized entry into a restricted and protected regions and alerting of responsible security personnel. (4) Biometrically controlled doors intended to keep unauthorized individuals out of lecture halls and laboratories with valuable computing resources such as computers and data projectors. (5) Library RFID security system that compels students at CUT not to take items from the library without checking out these items; the system triggers an alarm if a student tries to sneak out library material.

However, security systems in place at CUT present the following problems:

- 1) Security measures currently rely heavily on security guards who may make mistakes or can be compromised. Although CCTV could be used to bridge this gap, the technology does not work in all circumstances as they may provide blurry images or footage and many rudimentary CCTVs produce poor footage at night since some do not have night vision capabilities. Some perpetrators may conceal their identity from being captured by the cameras, making the footage useless. Again, they are quite costly to install and maintain not forgetting the cost of replacing stolen CCTVs.
- 2) Poor maintenance of security systems has left most sections of the institution insecure and as a result, people take advantage of this and continue stealing assets.
- 3) Security measures in place are mainly targeted at controlling access to buildings and have little to do with asset monitoring and tracking.
- 4) The security systems currently in place are disjointed; they do not talk to each other, nor do they instantaneously and intelligently send real-time security breach messages to security personnel.

According to Aristotle, *“The whole is greater than the sum of its parts”*. As described above, CUT currently runs an array of isolated security technologies and procedures; the University could achieve much more if these were integrated using IoT paradigm. These technologies complement each other in such a powerful way that integrating them, yields solutions (to problems) better than their individual sum would not otherwise solve.

3.4.1. CUT Asset Management Problem Factors

This section highlights some problems observed during interviews with Asset Management personnel which are later transformed into the main requirements of the system prototype:

Asset Tagging – For inventory purposes all newly purchased assets (laptops, tablets, projectors and smart phones) at CUT are tagged using barcodes. These barcodes have been replaced with the more mature RFID technology.

Asset Registration – After an asset has been tagged with a barcode, the subsequent step is to capture the details of each asset such as asset name, purchase value, serial number, and date of purchase, barcode number and insurance number (if the asset has been insured). These details are stored into Microsoft Excel document. The barcodes are often scanned using a barcode reader, but sometimes, this process is currently done manually. Asset records are sometimes kept in a book; this is a problem because asset records become difficult to store, update and retrieve in time of need.

Asset Assignment – This entails the act of leasing an institution's asset to employees. This process is done manually as well and the recipient of the asset is given a paper that is signed by both parties; this then serves as proof that this asset is now temporarily in the employee's possession for a specified period of time. Employees are mandated to carry this temporary proof of ownership with them in case the security personnel request it during search processes often conducted at exit points. There is a twofold problem here: (1) this paper proof can easily be lost or (2) duplicated and given to someone else to deceive security personnel and sneak out with institutional asset(s).

Asset withdrawal – This involves the act of returning a leased institution's asset by employees to a representative of Asset Management who in turn acknowledges that the asset has been returned and requests the employee to sign a given document. Like most of the other processes, this is also done manually making asset management a burdensome and time-consuming process. Managing assets in this manner is inefficient, given the voluminous CUT assets. This makes the whole process susceptible to errors and records may lack integrity. Critical paper records can be torn out of asset record books, making it difficult to trace or look up a particular record.

Asset Tracking and Monitoring – This was an aspect of significant interest to the researcher from the data gleaned from interview sessions with the Head of Security. The source of the theft problem was identified to be that most security systems in place have little or nothing to do with securing assets but buildings. As mentioned above, the researcher could not identify a system

deployed to monitor or track the whereabouts of assets and as a result many assets have been reported stolen.

The constructive research approach compels researchers to conduct a holistic and realistic investigation and analysis of the problem. In this research, this was achieved through understanding of insecurity problems at CUT. The investigations conducted identified problems discussed above and were considered relevant in this study because they disclosed to the researcher that there were loop holes in the manner in which assets were being managed; there was no security system in place to combat and relieve CUT of asset theft predicament. This is how the researcher was able to identify a pertinent real world problem that needed an innovative approach to iron out. Problem identification was the initial step in this study as indicated in Figure 1.3; this was followed by an investigation of the problem. In the rest of this dissertation, the researcher shows how these problems were transformed into potential prototype functions that aims at combating the problems CUT is currently grappling with; the implementation of the proposed system prototype is also described.

3.4.2. CUT Investigation Results

To better understand the severity of laptop theft and indeed of other assets such as tablets and cell phones around CUT, a questionnaire was designed to establish the magnitude of this menace. The researcher was not only interested in knowing the extent of theft severity, but wanted to investigate if the target population would uphold the use of a monitoring and tracking system. This section of the dissertation is intended to present findings from data that was collected randomly from a target population made up of students and staff members.

The sample size for this study was 50; herein quantitative results are presented in form of tables, and graphs and will show mainly in percentage terms the level of users' opinions or attitudes towards the asset monitoring and tracking system. In other cases, the responses of users were computed to get an average value that summarizes their views. A copy of the questionnaire used in this study can be found under Appendices (Questionnaire page 1-6).

The gender distribution of the population that participated in this study was 68% men and 32% women; 92% of the entire sample comprised of students and the other 8% was staff members such as lecturers and these were the only categories that the researcher targeted. Non-academic

staff were not included in the interview process mainly because the knowledge acquired from analysing asset theft documents was indicating that this category was least affected. However, the results obtained from these two categories can be safely generalised across the entire institution because none is immune to theft and as long as the institution's buildings and offices remain under secured, asset theft will perpetuate.

Table 3.1: Security system awareness

Technology	Aware	Unaware	No Response
Fingerprinting system	72%	26%	1%
Surveillance security cameras	68%	30%	2%
Library book anti-theft system	70%	28%	2%
Electromagnetic door locks	56%	40%	4%
RFID Tag-controlled door locks:	38%	56%	6%
Average	60.8%	36%	3%

Table 3.1 above summarises the sample population's security systems awareness, substantiating the fact that there is no system dedicated to monitor or track assets at CUT.

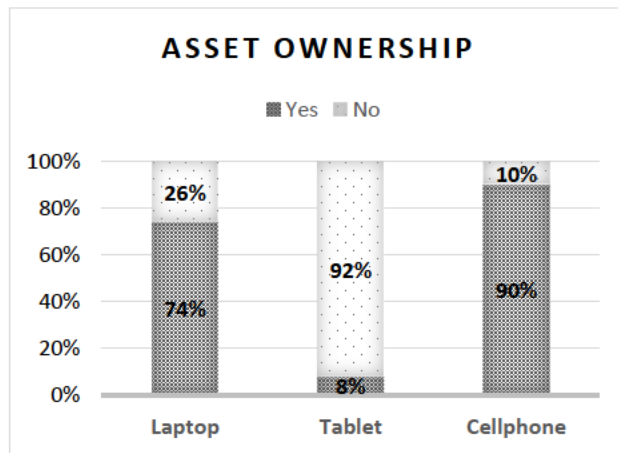


Figure 3.1: Asset Ownership

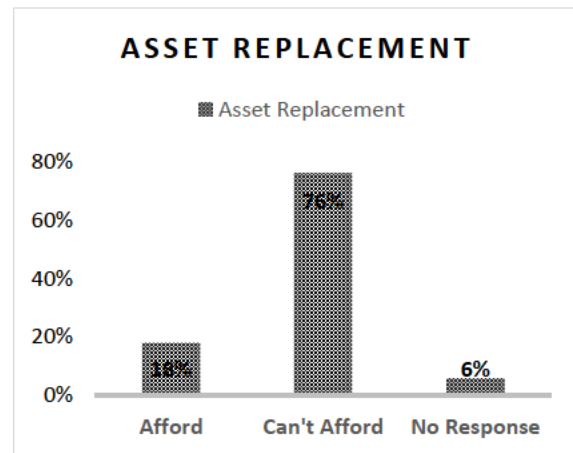


Figure 3.2: Asset Replacement Affordability

Figure 3.1 summarises asset (laptop, tablet and cell phone) distribution within the population group. Figure 3.2 shows statistics (based on question 6.1 of the questionnaire) of the sample

population's ability to afford replacement of lost assets. The researcher probed the extent of how important a laptop and mostly the data on it is to participants; Table 3.2 below, summarises their response.

Table 3.2: Degree of importance for laptops and data

Item	Not Important	Fairly Important	Neutral	Important	Very Important
Laptop	0%	2%	2%	4%	92%
Data	0%	0%	0%	2%	98%
Average Laptop Importance: 4.9 out of 5				Average Data Importance: 5 out of 5	

The data also indicated that 56% of participants have lost either a laptop or tablet or cell phone in the past five years while 44% did not experience any loss. Asset loss by category was distributed as follows: 21% was laptops, 39% cell phones, 29% of participants lost both a laptop and cell phone, and 11% represents lost assets other than laptop and cell phone.

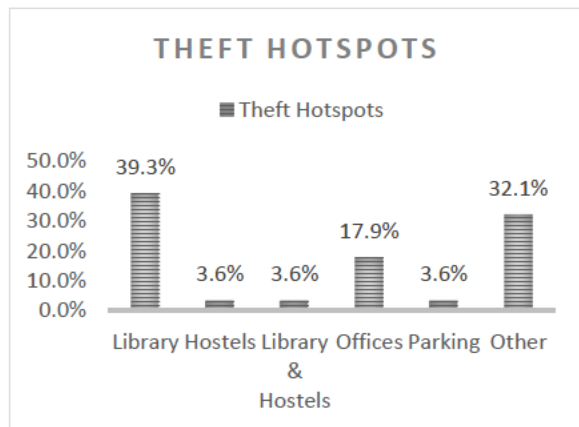


Figure 3.3: Theft hotspots

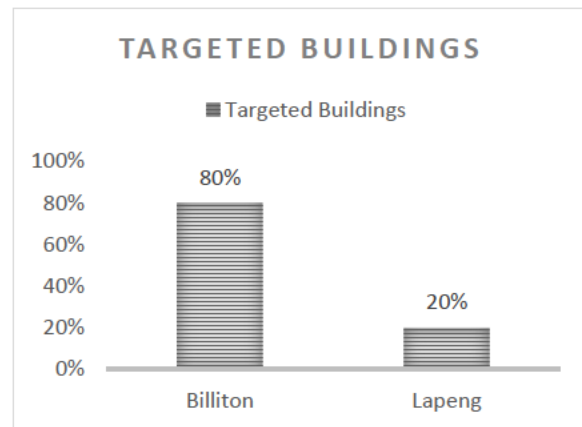


Figure 3.4: Targeted offices

The results depicted in Figure 3.3 above synopsis the quantities in percentage terms of assets lost along with the locations in which the assets were stolen from, while Figure 3.4 explains the 17.9% of thefts that occurred in offices (Figure 3.3) and shows the buildings they are situated in. The unavailability of asset monitoring and tracking systems has been confirmed as an appalling

phenomenon by participants in this study, because 93% of lost assets were never recovered and only 7% was recovered. This is the gap this study has attempted to fill.

As part of the investigation, the questionnaire asked the sample population questions that would gather data about whether or not they would support the use of a monitoring and tracking system. Table 3.3 summarises the data from participants who are interested in using the system prototype's functions such as laptop monitoring, SMS notifications and SMS to control the prototype and laptop tracking.

The average acceptance rate for each functionality is presented as well. These average figures are weighed against the value of 5. In addition to the questionnaire, the LMTS was also presented and demonstrated before students during a work shop and students were asked to vote for the system that they thought would solve a real life challenge. The system presented in this study was voted the best and the researcher was given an award for the work that was being done.

Table 3.3: System function usage

	Not Important	Fairly Important	Neutral	Important	Very Important	No Response
Monitoring	0%	2%	16%	10%	70%	2%
SMS Notification	0%	0%	14%	16%	68%	2%
Tracking	0%	4%	14%	10%	72%	0%
Average Monitoring: 4.5 out of 5		SMS Notification: 4.6 out of 5		Average Tracking: 4.5 out of 5		

Further, 88% of participants expressed that they find designated study areas unsafe for their assets, while 12% find study spots as being safe to leave anything unattended. Participants were asked to express how frequently they leave their assets (particularly laptops) unattended while in locations where assets are prone to theft; Table 3.4 summarises their opinions.

Table 3.4: Level of students' vigilance towards laptops

Always	Often	Sometimes	Rarely	Never	No
---------------	--------------	------------------	---------------	--------------	-----------

					Response
10%	12%	16%	16%	42%	4%
Average Vigilance Rate: 3.7 out of 5					

Participants were asked if they minded having an RFID tag attached to their laptops (if it helps improve security); Table 3.5 summarizes their responses, while Table 3.6 shows the responses from participants who were comfortable leaving their laptops in the care of a monitoring and tracking software.

Table 3.5: Tag acceptance level

Yes	Maybe	No	No Response
22%	20%	56%	2%
Average Acceptance 2.3 out of 3			

Yes	Maybe	No	No Response
50%	36%	14%	0%
Average Acceptance 2.4 out of 3			

Table 3.6: Software acceptance level

Moreover, 80% of participants rated laptop monitoring and tracking software as very important, 16% rated this as important and 4% were neutral in their responses. Lastly, 78% of the surveyed group was keen on using SMS commands to remotely control the prototype, 14% were indecisive and 8% did not welcome the idea. Moreover, 68% of the survey participants were comfortable with receiving SMS security breaches on their mobile phones, 24% did not embrace the idea and 12% was indecisive in this regard.

The findings presented above have indicated the magnitude of laptop theft and other assets. This was quantified in monetary value; theft hotspots were also made known. The average endorsement rate of software usage in monitoring and tracking laptops stood at a positive average of 2.4 out of 3, which signifies that the majority of the sample welcomed this idea.

4. Chapter 4 System Design and Development

4.1. Framework Development

In this chapter, the details of the proposed middleware, along with details of its structure and components that make up its architecture are presented.

4.1.1. Proposed Middleware Architecture

Systems developed around IoT paradigm are composed of a collection of various objects interconnected by different communication technologies, where each device functions through local and/or remote interaction with the real world or other devices and systems (Krakowiak, 2009). Management of IoT-centred systems involves a plethora of tasks such as object monitoring, capturing usage patterns, object tracking, triggering alerts via alarms, SMS, etc. Execution of these tasks involves remote access to systems and devices, data collection, analysis, interpretation, and reaction to peculiar events. A middleware provides an all in one solution to the management of the tasks outlined above through separation of functions and well-defined interfaces, provision of infrastructure that provides an environment for heterogeneous components, as well as a set of common services, such as transaction management and security (Krakowiak, 2009).

Before explaining the middleware architecture in great detail, it is noteworthy to mention that the middleware for this research study was implemented using a unique technique called ‘middleware in-lining’, which was postulated by Krakowiak (2009). Middleware in-lining entails writing middleware code in modules that are injected directly into the application. This technique mitigates some of the middleware design challenges discussed in section 2.5.7, earlier on in this dissertation.

The benefits of the proposed middleware are:

- Embedding the middleware code in the application was considered an ideal approach by the researcher for two reasons: (1) the middleware was intended to run on laptops with Microsoft Windows 7/8 Operating System (MWOS). The latter provides a rich platform

to manage diverse computing resources such as software services/agents, virtual windows location sensors, fingerprint scanners, universal serial bus (USB) ports, random access memory (RAM), central processing unit (CPU) and storage space. The researcher designed the middleware to harvest the power of services provided by MWOS such as network manager, location manager and windows start-up service responsible for automating execution of predefined windows services and applications at windows start-up; (2) developing a middleware that runs on MWOP, allowed the researcher to focus on application-specific requirements, through the provision of adequate services and resources without having to worry about limited resources on wireless sensors.

- The LMTS middleware adapted some features from Cougar middleware; the middleware implemented a simple database interface that supports: (1) exchange of varying volumes of data with laptops sharing the centralised relational database; and (2) use of structured query language (SQL) query commands implanted in the middleware to extract data from windows location sensors, and propagate this newly acquired location data to a remote database through a TCP/IP connection. The centralized database was configured and optimized to handle multiple simultaneous service requests such as data retrieval, data updates and data storage from many laptops connected to and sharing the database server. (3) The designed middleware was fault tolerant because it was calibrated to intelligently search and automatically connect to a backup database server, in case it could not access the main database server.
- The LMTS middleware was designed to support quality of service (QoS) as in MiLAN, although the delivery of QoS was implemented differently from the context it is used in MiLAN. Here, mobile phones are viewed as an integral part of the system, because the middleware supports over-the-air computing (OaC) (bi-directional communication with mobile phones). This means that the middleware can transmit messages to mobile phones and mobile phones also requests services from the middleware via SMS commands. This relationship required reliable delivery of acknowledgements; that is to say, the middleware was supposed to reply to service requests as a confirmation of request reception. The middleware's ability to instantaneously communicate security breaches was considered as the aspect that matches it to delivery of quality service.

- The Windows operating system improves manageability of services and supports interoperability by defining standard interfaces; the provision of standard interfaces helped the researcher not to worry about hardware heterogeneity during middleware design because this was the task left out to the operating system to address. The same middleware architecture may be reused on windows based systems to support connection of dissimilar hardware using a common interface such as USB ports.
- Middleware modularity entails the separation of the application and middleware layers such as database management layer. Modularity prevents other middleware layers from breaking up because of an error that occurred in another layer. It also makes it easy for the middleware to be scalable because it can be easily upgraded to cater for other functions that may be needed in future.
- The LMTS middleware was event driven; this was seen as a significant aspect because it is the element that delivers the intelligence needed in IoT applications. Here, the prototype was implemented using event driven approach. This was so because event handling offer developers the opportunity to tweak applications to unparalleled levels needed to meet the requirements of the system. To fully take advantage of multiple (persistent, periodic and temporal) events generated by the application, hardware and database layers, the researcher implemented middleware services (SMS, location and database) that execute on independent threads (a thread is a logical sequence of instructions, executed one after another within a process (Buyya, Vecchiola and Selvi, 2013)). This way, the middleware architecture supports service execution parallelism, which involves modularising programs into individual components that execute independently on separate threads (Silberschatz, Galvin and Gagne, 2009); this annihilated the need to queue events before they can be examined and acted upon. This resulted in a highly responsive and effective middleware that instantaneously perform preconfigured actions depending on the event generated.

In a nutshell, the proposed middleware has the following characteristics: (1) uses windows operating system to hide the heterogeneity of the various hardware components, and communication protocols that are used by different parts of LMTS; (2) provides a uniform, standard, high-level interfaces to the application developers and integrators, this way the applications can easily interoperate and be refactored, ported, and recompiled; and (3) supplies a

set of common services to perform various general purpose functions, in order to avoid duplicating efforts and to facilitate collaboration between top and low level-layers.

4.1.2. Proposed Middleware Layers and Sub-Components

Figure 4.1 below provides a diagrammatical overview of the proposed middleware architecture and it is composed of three layers, namely application, middleware and hardware. Each layers implements and utilizes various components and services.

Application layer found at the top layer of the middleware architecture presented in Figure 4.1 is not considered part of the middleware, but delivers services to end users through utilization of all functionalities exposed by layers below it (Atzori et al., 2010).

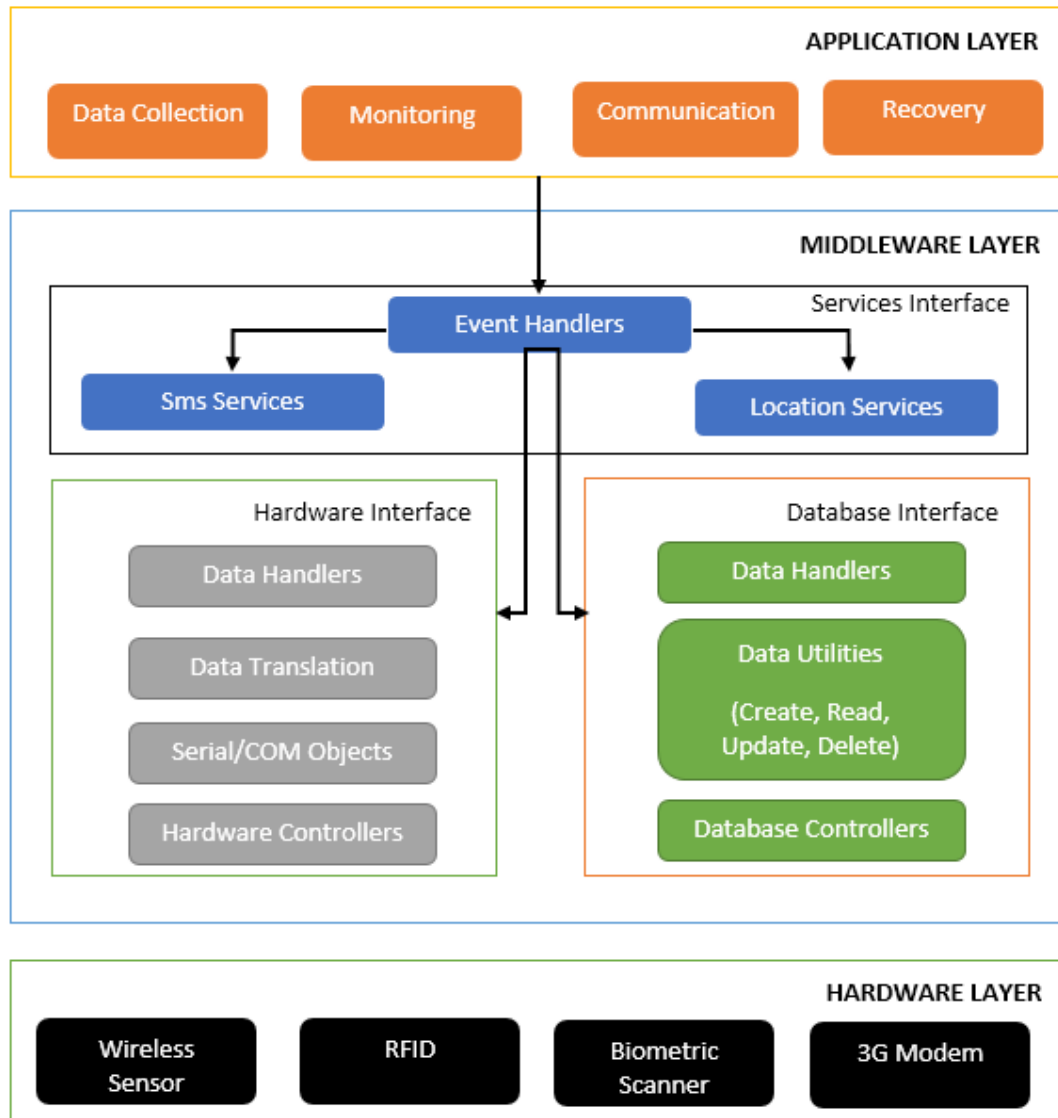


Figure 4.1: Proposed middleware architecture (adopted from Hwang and Yoe, 2011)

The application layer of the middleware architecture presented in Figure 4.1 contains the early warning system functions which were first introduced in Figure 1.2 of Chapter 1. These four core functions have been extended to show their interaction with system database along with other system components as displayed in Figure 4.2 below.

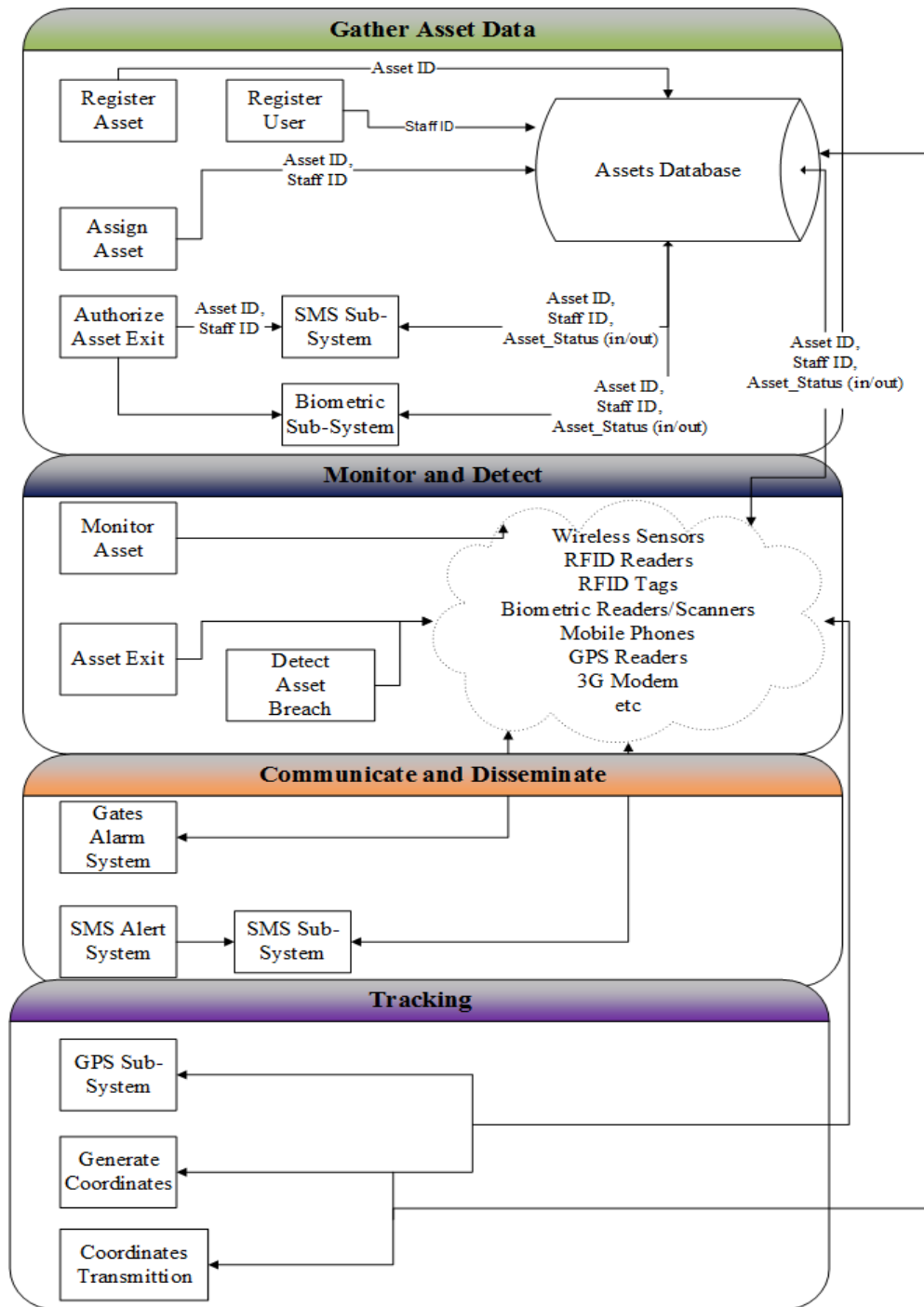


Figure 4.2: Overview of system architecture and functions

In this study, the application layer integrates four early warning modules highlighted in Figure 4.2 and whose main functions are:

- a) **Gather Assets Data** – The system maintains up-to-date records of assets, with an ability to register laptops and store the details in a database, assign, re-assign, transfer or withdraw laptops to or from staff members. An example would be a member of staff is assigned a tagged laptop and an assignment record is created and saved to the system database.
- b) **Monitoring and Detection** – This entails autonomous and intelligent ways to automatically monitor and detect laptop security violations. It should in particular detect the unauthorized exit of a laptop. Here, a combination of RFID readers, RFID tags, wireless sensors, mobile phones, Global Positioning System (GPS) devices, Geofencing, alarm bells and biometric readers/scanners are used. At this point, biometric scanners are used to facilitate turning on/off monitoring which commands the middleware to temporary start/stop reading the tag attached to the laptop, without triggering a security alarm.
- c) **Dissemination and Communication** – This module is responsible for sending appropriate alerts/information relating to laptop security violations to all stakeholders. It could include setting off alarms at the institution's main gates, sending SMS notification to a staff member assigned the particular laptop, security personnel along with asset management personnel in charge of assets at the institution.
- d) **Recovery Capability** – This module handles some of the activities needed to track and recover laptops. Though most of such activities may not be automated, this module incorporates Geofencing, virtual GPS sensors to enable tracking laptop location and Google maps to show laptop geographical position.

The middleware layer supports the following services and interfaces:

- 1) **SMS Services** – defines services responsible for communicating with an SMS gateway which is a mechanism that equips computer systems with the ability to transmit or receive Short Message Service (SMS) to or from telecommunications network. SMS platform achieves connections to the mobile phone networks through internet protocols or wireless links, but messages are in the end routed into the mobile network. This middleware service also implements event handlers that are triggered upon receiving a message. There are also functions to decipher the message and process it accordingly, which give

users the ability and liberty to control the application from their phones through SMS commands. This SMS service is primarily responsible for notification purposes and utilizes the hardware layer particularly a 3G modem to offer its services.

- 2) **Location Services** – defines utilization of windows location data providers that can either be a software service or hardware device with the ability to produce geographical coordinates for applications. The geographic position of a computer or device can be determined in a number of ways, including any of the following: (1) Wireless Fidelity (Wi-Fi) triangulation; (2) Global Position System (GPS); (3) Cell phone tower triangulation; (4) IP address resolution (Doty, Mulligan and Wilde, 2010). Location aware applications need services that periodically extracts objects' positional data in form of latitude and longitude and transmit this data to a remote storage facility, such as a centralized database server. Location services could be handy if incorporated with Geo-Fencing which allows triggers to be set up whenever a device enters or exits the pre-configured boundaries (Sanquetti, 2004).
- 3) **Hardware Interface** – defines the mechanism used to interconnect the hardware and the computer. Commonly used hardware interfaces for attaching external peripherals to computers are universal serial bus (USB) and serial port. Hardware controllers in computing context refers to a software program that manages or directs the flow of data bits between devices and the computer, these are also known as device drivers (Rubini and Corbet, 2001). Data translation aids in the conversion of bits of data or data model to another representation that is understood by the upper-level consumers of generated data. Devices often exchange data with systems in form of bits which are in binary form and these bytes of data need to be converted to a format that the target application or service understands. Data handlers are object instances that store converted data for use by services awaiting consumption of that data.
- 4) **Database Interface** – several applications need access to their respective data repository. This means the data need to be captured, stored, retrieved and manipulated. Microsoft ADO.NET provides consistent access to data sources such as Microsoft SQL Server (is a relational database management system (RDBMS), as well as data sources exposed through OLE DB and XML. Data controllers are classes integrated into .Net library used for connecting to a database, executing commands, and retrieving results (Jorgensen,

2012). Data utilities include the most basic level of procedures to create, read, update, and delete database records. Each of these operations is performed through SQL code and are accessible through ADO.NET. Data handlers are objects instances that hold data used to create new records, data read from the database that is manipulated or acted upon and saved back to the database.

Hardware Layer – individual application domains implement and integrate different types of physical devices. Each hardware device provides unique services and may have its own communication protocols and standards, required in order to interact and exchange data with the application. The hardware layer is often characterized by homogeneous devices interconnected locally or remotely through technologies such TCP/IP, Bluetooth and Wi-Fi. These devices are considered data generators, because their prime responsibility is interacting with the physical world, through sensing and digitizing environmental parameters. Data gleaned are acted upon and consumed by upper layers of the middleware stack (Hwang and Yoe, 2011).

4.2. System Analysis and Design

4.2.1. Software Development Approach Justification

The evolution of software development has seen many software development methods/models being suggested. Some development models have shown remarkable success in constant environments. In an ideal dynamic technologically driven world where frequent changes in technology, requirements, and staff are on the rise, traditional software development models are making software development a more burdensome process (Bhalerao, Puntambekar and Ingle, 2009). Traditional software development methods have proven to be less fruitful in dynamic environments because of the constant changes in requirements and technology. According to Cohn and Ford (2003), traditional software development methods have delivered a software success rate of less than 40% in such environments.

4.2.2. System Development Approach

The LMTS inherited developmental characteristics from agile method which include: (1) iterative and incremental development process, meaning a code artefacts had to be revised and

rewritten until a more satisfactory one that outperforms the previous version has been produced. The development process was incremental in the sense that code artefacts were constructed piece by piece in an iterative manner; (2) the researcher avoided biting off more chunks of work than that which the researcher could handle; so a modular-driven approach was adopted, meaning the researcher broke the project into smaller and more manageable tasks guided by the system Use Case. Each Use Case (task) was implemented separately, with considerable refactoring practices to eliminate duplication of functions. This modular approach placed priority on code cohesion (degree of interaction (Myers, 1978)) and low coupling (degree of mutual interdependence (Schach, 2008)). (3) This approach perpetually produced system releases, which are small planned versions of the system that make business sense. These system release were subject to thorough cycles of unit and integration testing to identify bugs (logical errors), syntax errors and test functionality of newly integrated system features. Figure 4.3 below shows adopted extreme programming (XP) processes diagrammatically, which aided the researcher (developer) to build the system prototype without complications and deliver a fully functional system that alleviates laptop theft.

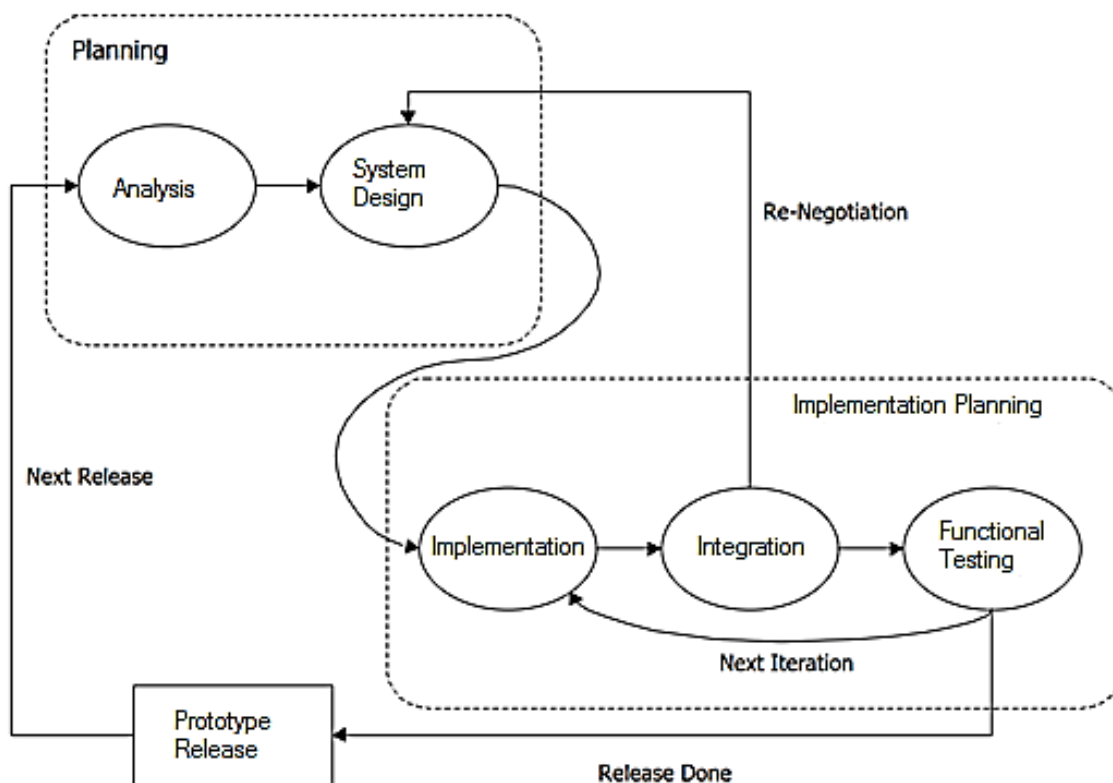


Figure 4.3: XP process (adopted from Dudziak, 2000)

It is noteworthy that the adoption of extreme programming paradigm was influenced by constructive research approach steps described in Chapter 3. Figure 4.3 depicted above, does not conflict with the one presented in Figure 1.3. These two Figures (4.3 and 1.3) complement each other, but Figure 4.3 stresses strong emphasis on activities that took place from step three through to step six of Figure 1.3, and how iterative these steps are.

4.2.3. System Specification Requirements

As part of the planning phase depicted in Figure 4.3 above, the researcher conducted interviews with Asset Management Department personnel and students to gain insights about the requirements for the new system. This initial step helped the researcher clearly understand that the objective of this study was to develop a laptop monitoring tool, with tracking capabilities that will curb laptop theft within the territories of CUT. During the case study discussed in Chapter 3 along with interviews conducted, the features or functions the system needed in order to become such a viable security system, were discovered. Later on during the planning and system analysis phase, the tools needed to accomplish the task of developing this systems were also identified. Table 4.1 below summarizes the functional requirements as well as hardware and software needs.

Table 4.1: Prototype system requirements

Functional Requirements	
1. Laptop Monitoring	2. Assignment Withdrawal
3. Laptop Tracking	4. Activity Logs
5. Early Breach Notification	6. User Registration
7. Laptop Registration	8. System Authentication Mechanism
9. Laptop Assignment	
Hardware Requirements	
1. Arduino Mega Microcontroller	2. Arduino RC522 RFID Reader
3. Passive RFID Tags	4. FlexiForce weight/pressure sensor
5. Libelium Waspote Sensor Board	6. Libelium Event Board
7. 3G Modem	8. Subscriber Identity Module (SIM) Card
9. Futronic FS88 Fingerprint Scanner	10. GPS sensor
11. Camera Sensors	
Software Requirements	
1. Microsoft SQL Server 2012®	2. Ozeki Sms Gateway®
3. Arduino IDE®	4. Waspote IDE®
5. Visual Studio 2013®	6. Fingerprint SDK®
7. ModelRight V4®	8. SQLQueryStress Version 0.9.7.0®

4.2.4. System Use Case

After a successful identification of system requirements summarised in Table 4.1, the researcher carried out a thorough analysis of these functional requirements with the intention to draw out an overall blueprint for the development of the new system. This analysis phase was iterative and

incremental in nature. This was so because in order to reduce the uncertainty and ambiguities in obtaining system requirements, there was the need to conduct repeated consultation sessions with stakeholders. These consultations took place in the form of unstructured interviews and meetings and brainstorming was the order of business.

In the final phase of planning and analysis, there was the need to transform these functional requirements into technical models. This was achieved through employment of Use Cases, Story Boards and other technical system representations such as flow charts. Guided by XP developmental principles suggested by Dudziak (2000), the researcher approached this arduous task with the goal to produce, natural designs, simple enough to get the job done effortlessly. Figure 4.4 below shows the transformation of system requirements into a Use Case diagram.

The following section provides a full explanation of the overall prototype Use Case and how different actors interact with the proposed system.

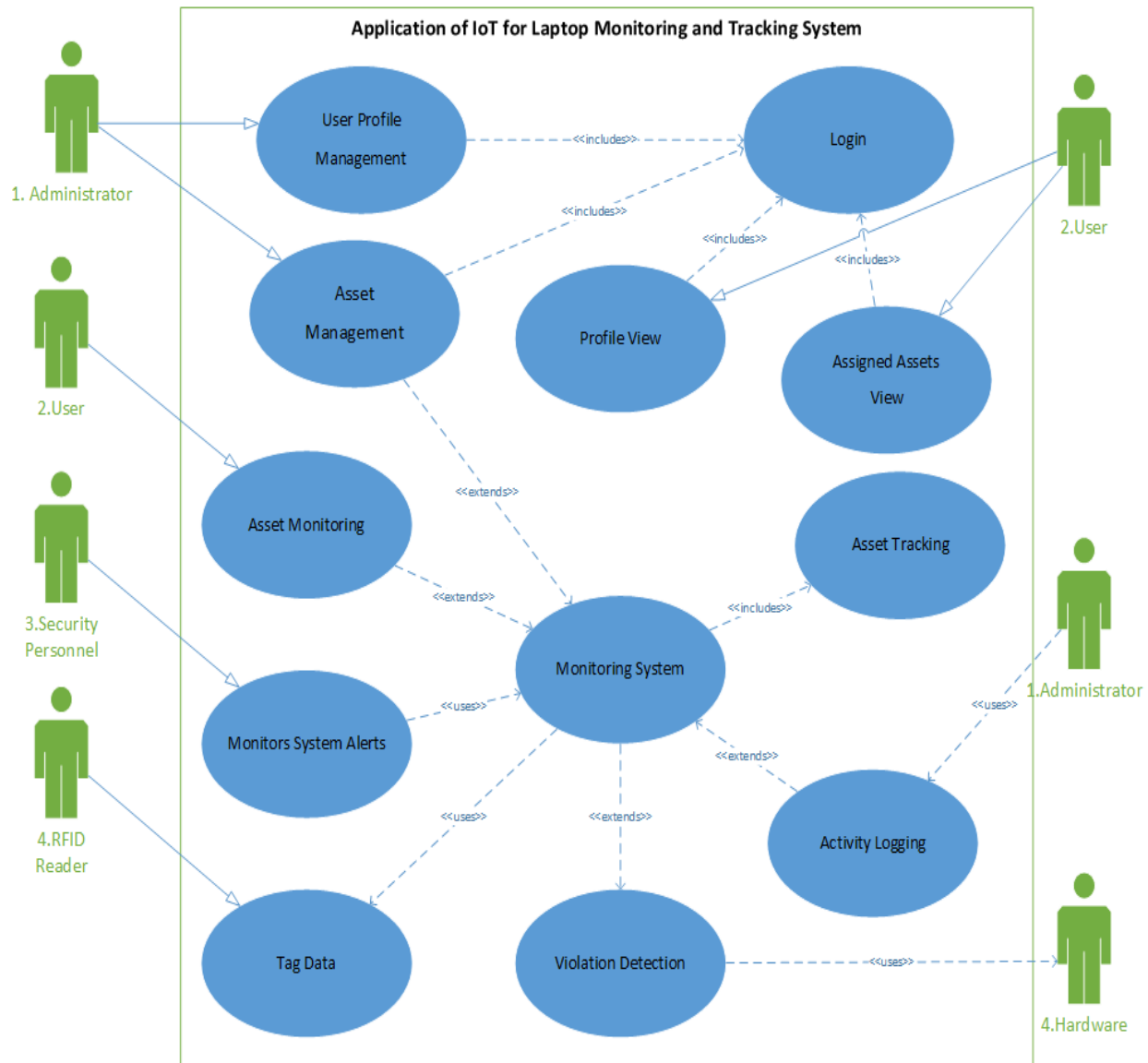


Figure 4.4: Overall system use cases

User Profile Management: system administrators use biometric fingerprint scanners to log in into the monitoring system and are primarily responsible for registering new system users through creation of user accounts/profiles. An administrator can create new user accounts and gathers users information; this involves enrolment of user fingerprints, used to authenticate users to the system and save the data to a database. An administrator can delete user accounts (that are no longer needed) from the database. An administrator is also responsible for updating user

account data, removing or adding new information. This could entail enrolling of new fingerprints, changing contact details.

Asset Management: administrators login into the system using biometric fingerprint scanner and go a step further to manage assets through registration of laptops on the system. This entails capturing laptop details and creating a profile out of captured data. Laptop serial number, tag number, laptop name and laptop cost are some of the data used to register a new laptop on the system. An administrator is responsible for the assignment of all registered laptops to their respective users. Date of assignment, laptop identity number, and user identity number are some of the data used in assignment process. An administrator can also delete laptop records and is also tasked with updating data records, removing or adding new information. Moreover, laptops which have been assigned to users can be withdrawn from registered users through the process of asset withdrawal. Changes can be made to assignment records, these records can be updated through extending or reducing lease time.

Profile View: users have the ability to view their user information or profile, and they are mandated to authenticate themselves to the system using fingerprint scanner; registered system users have been disallowed from changing any of their information for purposes of information and security integrity. Should any registered user wish to change their personal information, they are compelled to do so via system administrators.

Assigned Asset(s) View: here, users are allowed to view a list of assets assigned to them, registered users simply scan their finger on a scanner and if the user is identified, the system retrieves records from database and displays them for the user to see. Unique asset details are shown to the user, including how much time is left before asset lease elapses.

Asset Monitoring: registered users have the ability to start surveillance on their assigned laptop. A user simply provides authentication to the system through (a) fingerprint for local or (b) password for remote-system access. Remote asset surveillance can be initiated from a mobile phone. Surveillance is conducted by repeated interrogation of RFID tag attached to the asset or by repeatedly monitoring asset's pressure exerted on the pressure sensor. If the user identity has been validated and accepted by the system, then asset monitoring commences. The system also continuously extracts GPS coordinates and saves them to system database. These coordinates are used for identifying the geographical location for the laptop.

Violation Detection: this is the most critical system function. Once a registered system user initiates asset monitoring whether remotely by cell phone or locally through authentication by fingerprint scanner, the system starts listening for system breach events. The Asset monitoring use case above gives details on how monitoring is conducted. Assume a scenario such as this: the system is monitoring a laptop and someone comes and attempts to snatch the laptop away, either by breaking the communication between RFID reader and tag or lifting it off from the pressure/weight sensor. Such system operation anomalies are treated as security breaches and the system instantaneously sends a breach detection alert via SMS to a security personnel within the building. This prototype can be extended to include additional functions such as remote controlling of building doors. For example, if a security breach has been detected, the system may be configured to initiate building lockdown. This way all security measures in place can be integrated through IoT and this can deter asset theft by a significant margin.

Asset Tracking: this study adopted the use of GPS to track laptop location using embedded windows location sensors and envisioned to implement Geo-Fencing technology to generate alerts whenever a stolen laptop enters the perimeters of CUT. This function is entirely managed by the middleware, particularly the location services. A detailed description of location service was presented above under section 4.1.2. Location service is a locus-aware agent of the middleware that provides the geographical position of a laptop and transmits this data to a remote database for storage and later retrieval. It is significant to mention that, this locus aware agent is able to determine the geographical point of individual laptops that have location sensors and have a reliable connection to internet. This is so because the location service uses Wi-Fi triangulation or IP address data to compute the coordinates and a connection to transmit these coordinates to a remote database server.

Activity Logging: this is an imperative aspect of most security systems and logs are a good reporting and diagnostic mechanism to track application errors and events. Log files make available relevant information that is required to understand the happenings of critical system elements. Logs are handy in times of catastrophe; they remedy events that have taken place and that were unforeseen and most importantly, help to avoid future occurrence of similar events. Furthermore, logs provide real time monitoring and notification, if events happen that need to come to the security researcher's attention. Event log available in Windows operating system is a

good example of how logs are used to notify security administrators of triggered events. The security prototype follows the logging trend, by keeping track of system events and save them to a log file.

A system administrator manages these activity logs through performing various actions such as log archiving, log deletion, log analysis, and log retrieval. Log file contents are saved in human readable format and have time stamps associated with a particular event. Administrators along with security personnel may retrieve these activity logs when investigating peculiar cases; they can serve as good source of evidence if something incomprehensible takes place (Forte, 2009). This helps to clean the database and avoid keeping unwanted and unnecessary data, which might consume a lot of space and once a database is full of out-dated data, it might degrade its performance, resulting in slowness and long system response times.

4.2.5. Database Design

A database is a shared, integrated computer structure that stores a collection of raw data and metadata (data about data) according to Rob, Coronely and Crockett (2008). Based on the case study and system use cases identified (Figure 4.4) during system requirement gathering and analysis, it was determined that a relational database with tables highlighted in Figure 4.5, will be necessary to meet the requirements of the security system prototype. A relational database invented by Codd at IBM (1970) is a group of data items structured as a set of logical and related tables, from which data can be stored, retrieved or reassembled in diverse ways without having to reorganize the database tables (Codd, 1970). This database model served as an initial phase for the design of the security system prototype database. Figure 4.5 shows a portion of the initial unrefined database design structure along with relationships among different entities that help in storage, retrieval, updating and logging of different system activities.

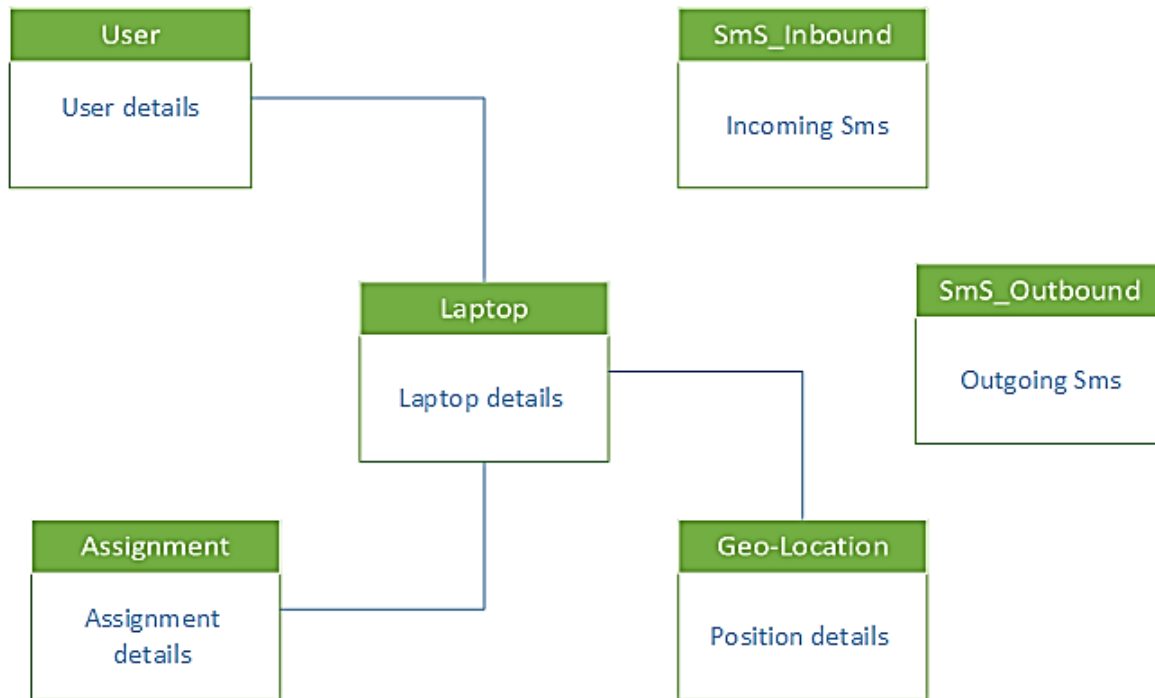


Figure 4.5: Initial database entity identification

4.3. System Implementation

Adoption of XP benefited this study in a number of ways including the following: (1) this project was driven to success due to careful planning, which is a critical aspect as it helped provide an overall direction the development process needed to take and make known all the resources needed to tackle the problem at hand (Dudziak, 2000). (2) The use of Use Cases, Database ERD, Story Boards and Flow Charts helped to define the project in both technical and non-technical manner, hence elevating thorough understanding of the project. Such diagrammatic representation of the system, served as small releases that helped determine the time lines of the project, along with the core functions of the system, including hardware and software requirements.

(3) These small system releases were presented to people involved in the development process to review the system, provide new insights and direction. Feedback from involved parties helped the system developer not to deviate from what the users of the system really wanted. (4) Adherence to XP approach, provided opportunities to improve and fine-tune work, because it's iterative by nature, meaning planning process, system design, artefact refactoring, and unit

testing as well as continuous system integration were all iterated throughout the development cycle. (5) Furthermore, XP helped in adapting to changes in requirements and system functions due to hardware and time limitations. (6) Lastly, XP exposed the researcher to the following core development principles that served as developmental guidelines.

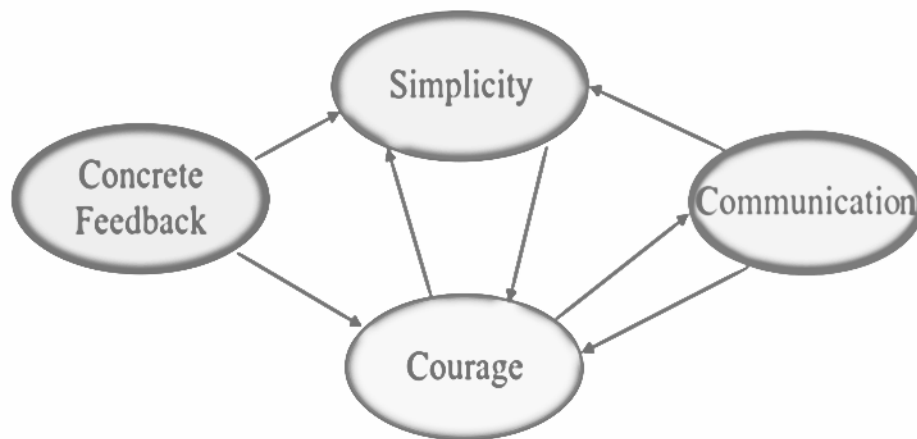


Figure 4.6: XP development principles (*adopted from Dudziak, 2000*)

Simplicity – equipped with this principle, the researcher learned to write code artefacts that proved not difficult to read and understand, thus making the developed prototype system less burdensome to maintain or apply changes to, when the need to arise. Adherence to simple designs is significant because having a very complicated system with lots of unused or unwanted functions leads to reduced productivity, because the system hinders users from completing business tasks. It is significant to mention that, it is by far more costly to maintain or add some new functionality to a complex system in the real world. Simplicity in XP means striving to “Doing The Simplest Thing, That Could Possibly Work” and “If You Are Not Going To Need It, Do Not Do It” according to Schach (2008).

Communication – engaging with different stakeholders was a crucial and critical aspect in prototype development process. Communication benefits were evident during and after interviews because, effective communication drove the development process to success, through acquisition of developmental insights and knowledge exchange. Poor communication practices

are a contributing factor towards software development failure. Communication in this study gave all stakeholders a shared mutual view of the system, which minimized conflict of interests with intended users of the system (Konovalov and Misslinger, 2006). Communication from all interested parties helped to minimize changes to the system, thus cutting short the development time.

Concrete Feedback – The researcher often received insightful criticism from the stakeholders about the system. These critics helped in circumventing, implementation of functions that deviated from what the intended system users needed. Developmental feedback about the system, from other developers also helped to map the system against the stipulated requirements and brainstorm simple algorithms to allay complicated issues.

Courage – the development process involved different stakeholders as part and parcel of the development team, this unity in development gave the researcher (developer) the much-needed courage to cope and feel comfortable with code refactoring, whenever it was deemed necessary. This was important because the researcher was often expected to deliver functions that the researcher lacked knowledge of.

In this study, it was discovered that XP supports use of spike solution(s) (Konovalov and Misslinger, 2006). A spike solution is a simple program to explore potential panaceas that addresses the problem at hand. Spikes are meant to generate insightful knowledge and brainstorming of ideas good enough to solve technical glitches and are discarded after use, they are rarely integrated into the system (Konovalov and Misslinger, 2006). The researcher also got courage to conduct code refactoring; which is the practice of restructuring software source code to make it easy to read and comprehend, without affecting its outward performance. This involved reviewing and revising existing system code and modifying it so that future changes can be implemented more easily (Schach, 2008).

4.3.1. Overall System Database Implementation

The database for this system was designed and fine-tuned using SQL server 2012; the database modelling was created from a software called ModelRight version 4.0 (Modelright.com, 2014). The system database was well normalized for several reasons which include: (1) to rid data redundancy; (2) increase response time, during periods of high load stress; (3) to insulate, too

much entity dependence which might cause a lot of integrity errors; and (4) normalization helped to shape the overall structural design of the database; meaning it is now possible to modify one record, without the need to effect a similar change, in many other related records because changes are instantaneously cascaded across related records (Rob et al., 2008).

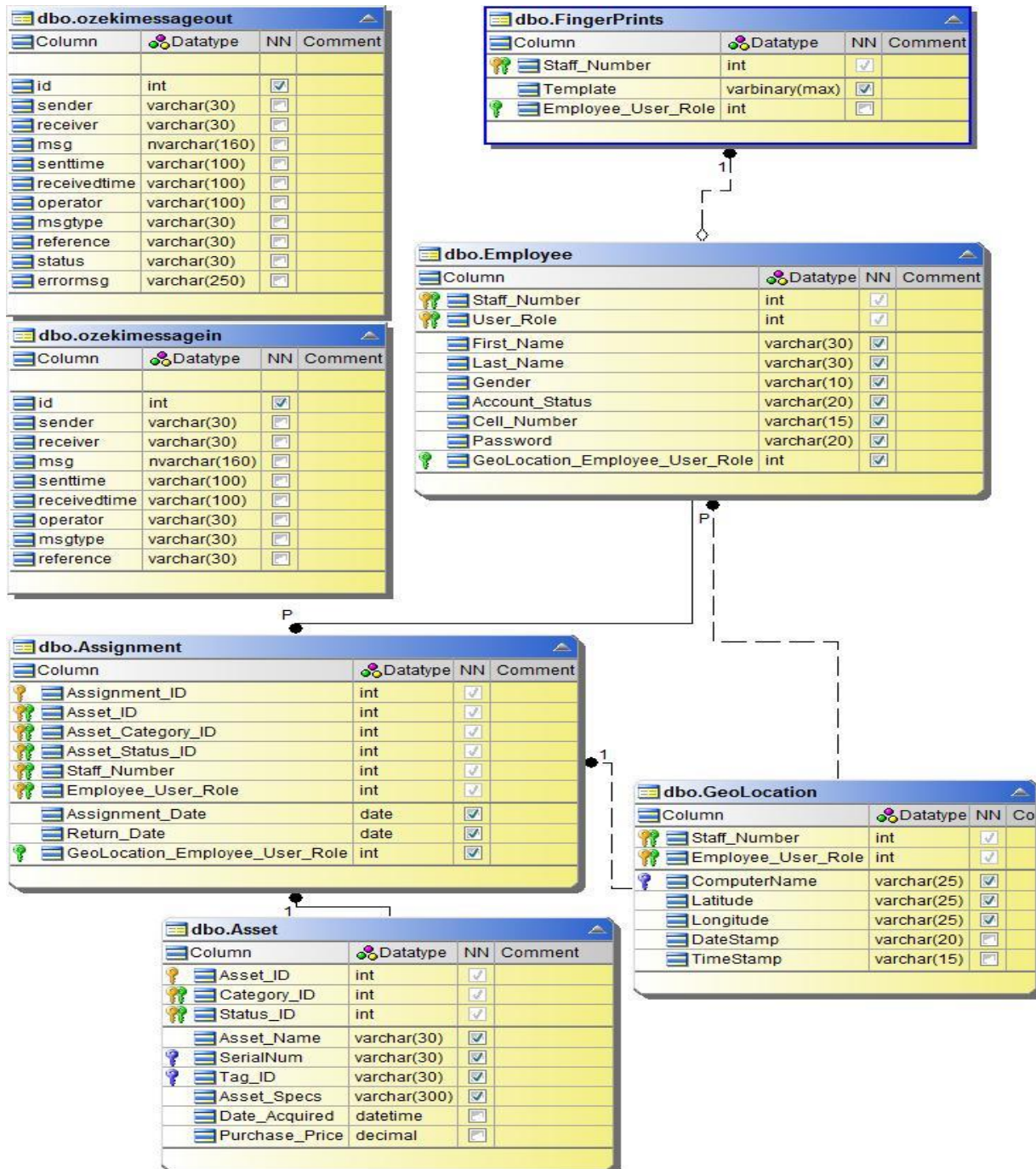


Figure 4.7: Excerpt of prototype database design

Through iterative refinement processes and consultations, the database structure was redesigned and optimized to cater for future expansion needs and further prototype augmentations. The choice of using Microsoft SQL server 2012 was influenced by many factors such as: (1) ease of integration with development tools such as Microsoft Visual Studio 2013, which was used to develop the system prototype; (2) Compatibility with standard application programming interfaces (APIs); (3) The ability to handle multiple requests without affecting the quality of service and its capability to support small to large databases; and (4) it provides a robust platform for building, deploying, and managing solutions that span on-premises and cloud (Jorgensen, 2012). The database structure (Figure 4.7) along with the use cases (Figure 4.4) presented above, provided guidelines in the development of business, presentation and data access layer for the system.

4.3.2. Middleware Interaction with Database

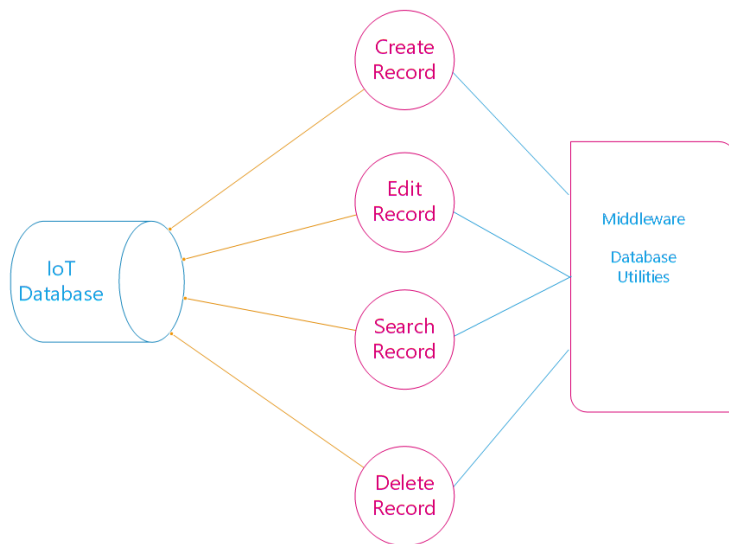


Figure 4.8: Database utilities

Figure 4.8 shows how the system middleware implemented utilities that interact with the prototype database to perform various system functions. These database utilities were implemented in a generic manner to accommodate future addition of new database entities without rewriting the middleware code. The system designs revealed the need to create one parameterised function for each of the utilities and the application layer had to provide two

parameters (Table name and an array of data) to use the functions. The sample code for these functions is depicted in Appendix 3 (Create record code) and Appendix 4 (Update record code).

Create Record – the system prototype provides a user-friendly interface to capture data needed to facilitate and support various system operations highlighted in Figure 4.8 above. All captured data are saved to the database by a simple click of a button that in turn triggers an event that instructs a middleware service to initiate the data-saving process.

Edit Record – records saved in the database can be altered by system administrators. These administrators simply retrieve the record that needs to be edited by querying the database through middleware service functions. If the record is found, it is displayed on the system interface, giving the system user the ability to apply necessary changes to the record and save these changes by a simple click of a button. Such an action fires an event handler that validates the data and ultimately triggers the middleware's data saving procedure. This process helps the system to always maintain updated records, with accurate values to avoid system malfunction.

Search Record – before system administrators can either create a record or edit a record, the middleware provides procedures that iterate through saved records to check for duplicates. There are certain parameters whose values must be unique (for example, employee number and mobile phone number) and should never be repeated. So should the administrator intend to create a new record, the system has automated functions that checks for record duplication. The system raises an alert in case of a violation of this condition and requests the user to rectify the problem. This helps system administrators to maintain data referential integrity within the database environment.

Delete Record – at times, the database needs cleaning by removing unnecessary and unwanted records. Keeping obsolete records compromises the database performance, which ultimately results in long response times. These database cleaning services are made available to users through middleware delete function.

4.3.3. System Architecture and Middleware Implementation

An early warning system approach mentioned in 1.4 of Chapter 1, was adopted in this research to develop the tracking and monitoring system prototype. This way, the resulting system is generic

(can be ported across several domains). The four components of early warning systems have been adapted into fundamental system functions and the details of these have been provided in section 4.1.2 and presented diagrammatically in Figure 4.2. Shown in Figure 4.9 below is a conceptualized system integration and design architecture that illustrates how the interaction of the application, middleware and different hardware was accomplished in the system prototype.

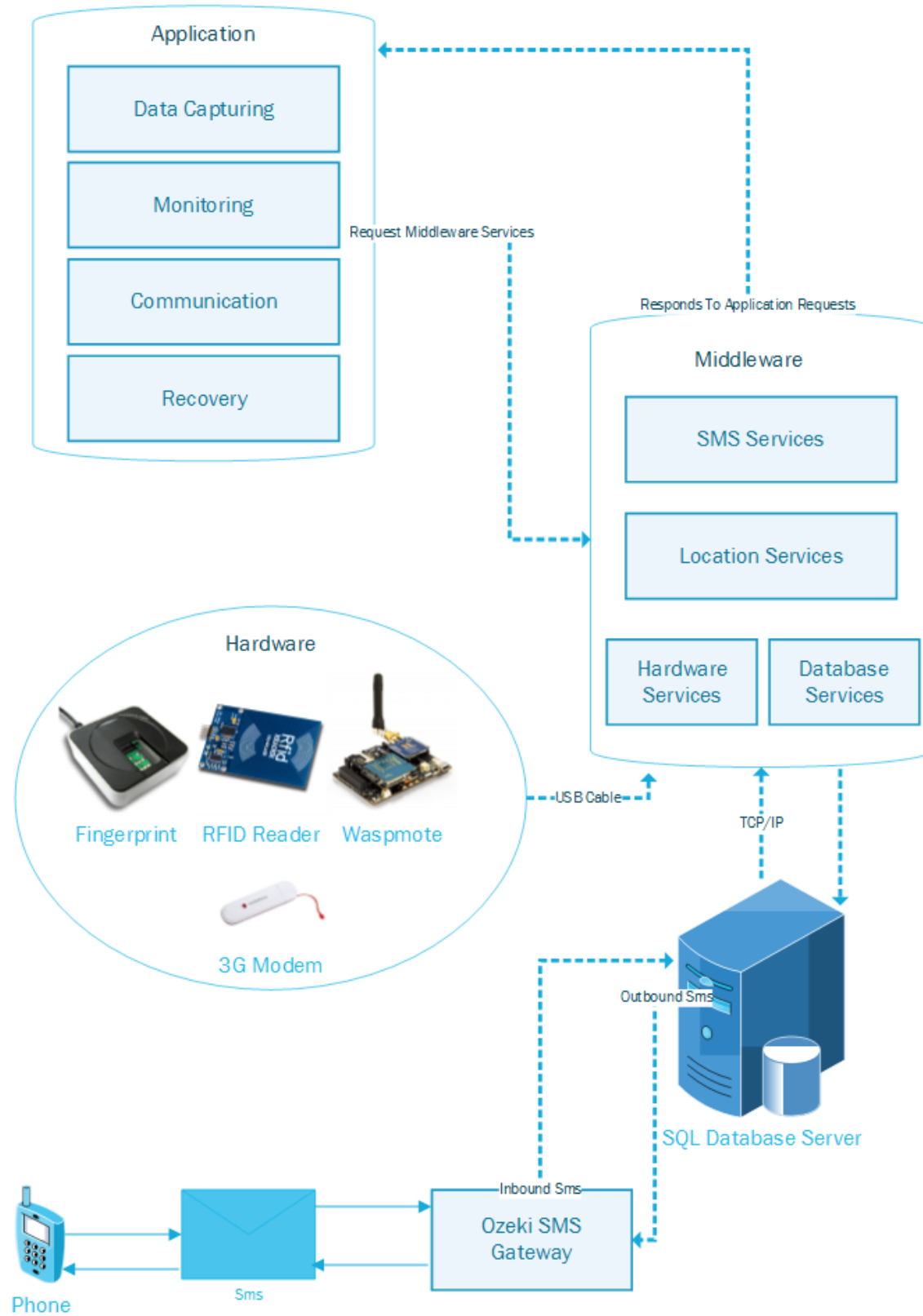


Figure 4.9: System components integration and interaction

4.3.4. Microcontroller Pre-programming

<pre> /* * Read a card using a mfrc522 r * Pin layout should be as follc * MOSI: * MISO: * SCK: * SS(SAD/SDA: * RST: Pin 9(Usually Not Connec * The sketch will display the inf eg "RFID TOKEN FOUND TOKEN NO: Dec: 114, 101, 99, 181, 193 Hex: 72, 65, 63, B5, C1" */ #include <SPI.h> #include <RFID.h> #define SS_PIN 53 #define RST_PIN 9 RFID rfid(SS_PIN, RST_PIN); // Setup variables: int serNum[5]; int val; boolean statusCheck; void setup() { Serial.begin(9600); SPI.begin(); rfid.init(); pinMode(13, OUTPUT); statusCheck = true; } </pre>	<pre> void loop() { if (rfid.isCard()) { if (rfid.readCardSerial()) { for(int x = 0; x < 5; x++) { Serial.print(rfid.serNum[x], DEC); } Serial.println(""); val = 3; statusCheck = true; } } while(val < 2) { if(statusCheck == false) { digitalWrite(13, HIGH); delay(1000); return; } else { Serial.println("Removed"); digitalWrite(13, HIGH); statusCheck = false; delay(1000); return; } } digitalWrite(13, LOW); val = val--; delay(1000); //change the speed at which th } </pre>
--	---

Figure 4.10: Arduino microcontroller programming

Figure 4.10 above shows the C++ code that was programmed into an Arduino microcontroller to interrogate a tag that is within the RFID reader's read range. The tag attached to the laptop responds by reflecting back its unique code to the reader and the reader passes that data to the

middleware for further processing. As long as the reader can communicate with the tag, the middleware does not raise violation alerts. Figure 4.11 below shows an excerpt of another approach of how laptop monitoring was achieved. Here the C++ code works as follows; the laptop is placed on a pressure/weight sensor, and the middleware records the weight and starts watching for any sharp drop or increase in weight. If the middleware detects a difference margin that is too high from the initial weight captured then the middleware reacts by sending a breach message to the middleware. The middleware then initiates the communication of the breach with laptop holder and security personnel.

```

void setup()
{
    // put your setup code here, to run once:
    SensorEventv20.ON();
    USB.ON();
}

void loop()
{
    // put your main code here, to run repeatedly:
    value = round(SensorEventv20.readValue(SENS_SOCKET1, SENS_RESISTIVE));
    USB.print(F("Current Weight Reading: "));
    USB.println(value);

    if(USB.available() > 0)
    {
        command = USB.read();

        if(command == 0x31)
        {
            if(tempValue == 0)
            {
                tempValue = value = SensorEventv20.readValue(SENS_SOCKET1, SENS_RESISTIVE);
                USB.println(F("Weight To Monitor Set."));
            }
        }
        else if(command == 0x32)
        {
            tempValue = 0;
            USB.println(F("Asset Monitoring Stopped"));
        }
    }

    if(tempValue != 0)
    {
        if(((value-tempValue) > 1000) || ((tempValue - value) > 1000))
        {
            USB.println(F("Security Breach Detected"));
            tempValue = 0;
        }
    }

    delay(1000);
}

```

Figure 4.11: Wasp mote microcontroller programming extract

4.3.5. Data Capture

The processes involved here have been emphasized in the Use Case shown in section 4.3.4 and explained in detail under a) User Profile Management and b) Asset Management of Figure 4.4. A

synopsis of these processes is presented here. The system maintains up-to-date records of laptops and with an ability to register new laptops and store the details in a database, assign, re-assign, transfer or withdraw laptops to or from staff members. Here, a member of staff is assigned tagged laptop(s) and a record is created and saved to the system database. The system also captures user fingerprints using a biometric scanner and a unique tag number using an RFID reader.

The screenshot shows a 'Profile Management' window with a dark theme. It contains three panels:

- Personal Details:**
 - Staff Number: [text input]
 - First Name: [text input]
 - Last Name: [text input]
 - Gender: [--Select Gender--] (dropdown)
 - Position: [--Select Position--] (dropdown)
 - Department: [--Select Department--] (dropdown)
- Account Details:**
 - Account Status: [Active] [Inactive] (radio buttons)
 - User Role: [--Select User Role--] (dropdown)
 - Cell Phone: [text input]
 - Account Password: [text input]
 - Confirm Password: [text input]
 - Enroll Fingerprints: [Click To Enroll] (button)
- Options:**
 - Save Record (button with disk icon)
 - Update Record (button with refresh icon)
 - Delete Record (button with trash icon)
 - Search Record (button with magnifying glass icon)
 - Cancel Record (button with X icon)

Figure 4.12: Data capturing interface

An example of data capturing interface has been presented in Figure 4.12 above, showing what data are needed to create a user profile. A correct user record requires data such as staff number, first name, last name, gender, employee position, employee department, account status, user role on the system, cell phone number (to communicate system events with the user), account password (that is used to authenticate user to the system when operating the system from a cell phone) and lastly fingerprints (used to give user access to the system).

Once the user different from the administrator has supplied the required data, the system administrator attempts to save the record by clicking button 'Save Record'. Once this button has been clicked, the supplied data is validated by the system; validation is done in the form of checking that all the fields have been supplied with data. If any field is empty (without data), an error is raised and the system highlights the fields that require attention. Another type of validation that takes place is duplication check; here, a middleware service uses a supplied data

parameter (for example, employee number) and compares it with data stored in the database. If the middleware service encounters similar data values or duplicates, the system also informs the administrator about it and requests values to be changed prior saving record to database. The interface also shows that administrators can also edit, delete and search a specific record through middleware services presented in Figure 4.8.

4.3.6. Monitoring

This application service entails the process of monitoring to detecting breaches and deterring of laptop theft. As explained earlier, this activity does not work alone but is highly dependent on several hardware components and middleware services. A sequence diagram presented in 4.13 below shows steps involved to achieve laptop monitoring.

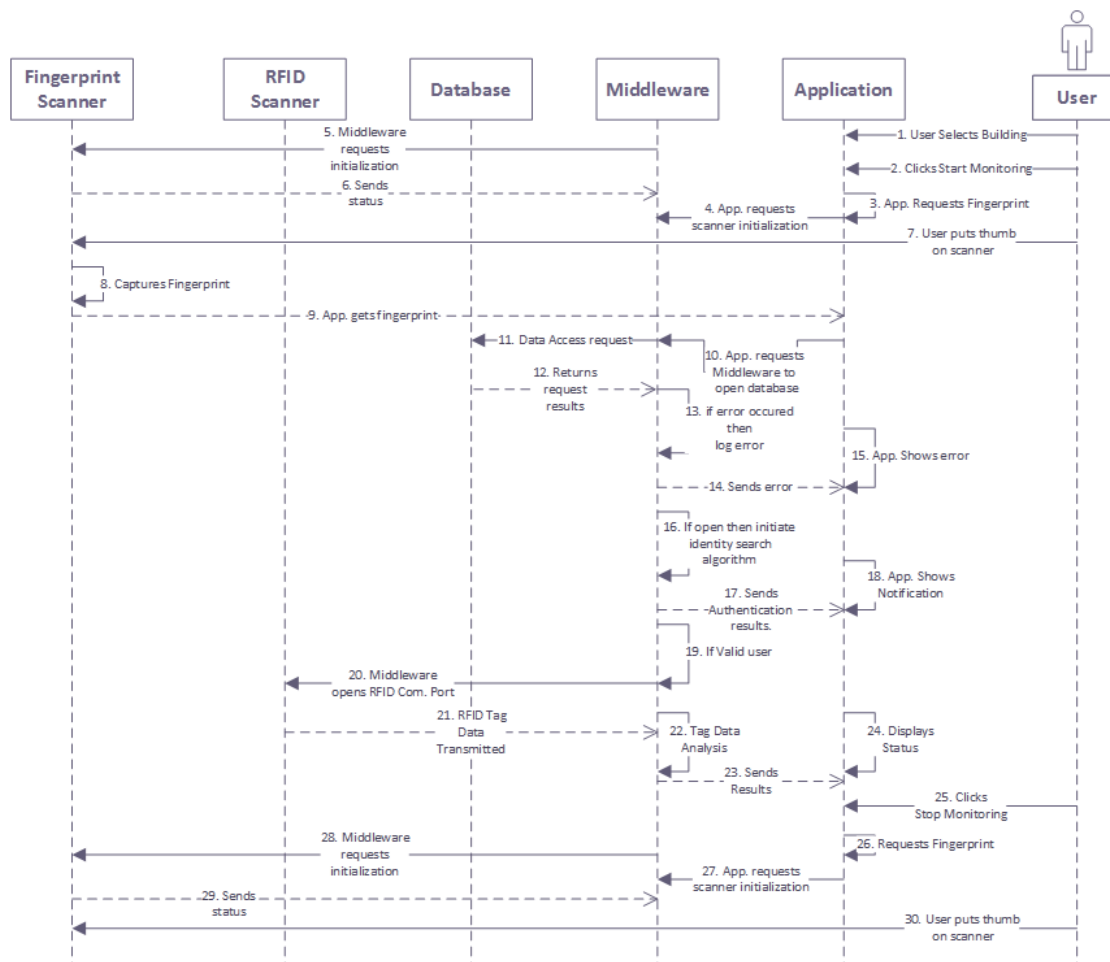


Figure 4.13: Monitoring sequence diagram

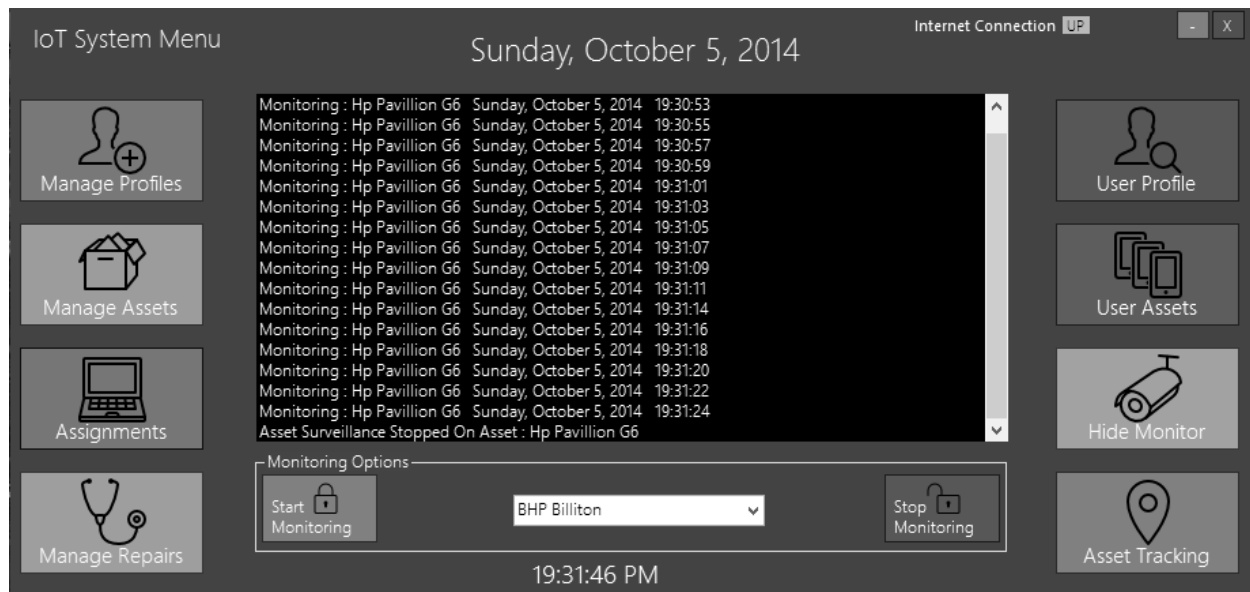


Figure 4.14: Laptop monitoring interface

Here, the user simply connects the USB cable of either the RFID reader or pressure/weight sensor and runs the LMTS software. From the interface shown in Figure 3.14, the user selects the building he/she is in and clicks button ‘Start Monitoring’ from the interface. The system requests the user to place his/her finger on fingerprint scanner for authentication. The captured fingerprint template is then compared to the one stored in database through middleware database services. If the system finds a match, then, another middleware service procedure validates the tag number to check if it corresponds to the identity of the person requesting to monitor the laptop. RFID tag validation is skipped if the user decides to monitor the laptop using a pressure/weight sensor. Thereafter, monitoring commences through continuous interrogation of the tag or comparison of pressure/weight exerted on the sensor by the laptop.

Should the laptop holder wish to stop monitoring, then he/she simply clicks ‘Stop Monitoring’ button shown in Figure 4.14 above and the system responds by requesting a fingerprint used to initiate the monitoring process. This biometric authentication mechanism is briefly discussed below.



Figure 4.15: Authentication interface

Figure 4.15 exhibits the authentication interface users use to request access to system functions. Button ‘Enrol’ is disabled and unusable to users; only system administrators can use this button to create new fingerprints during user registration. Users simply click button ‘Identify’ to start the verification and validation process. If user fingerprint template stored in the database matches the base template on the biometric scanner, the system gives the user access to use available system functions.

If the system detects that templates mismatch, the system denies the user access to system functions and displays a message to inform the user why the system denied granting access. If someone other than the laptop owner tries to steal the laptop by breaking the communication between the tag reader or pressure/weight sensor and middleware then, the middleware reacts by triggering alerts.

There was also an attempt to integrate a video sensor module to the wireless sensor used to detect weight discrepancies; the reason for doing this was to capture a picture of the perpetrator who is attempting to steal the laptop and transit this picture as bytes of data to the middleware. The middleware would then facilitate the communication of this security breach.

4.3.7. Alerts Communication and Dissemination

This module utilizes Ozeki SMS gateway (Ozekisms.com, 2014), 3G Modem through middleware services and cell phones and is responsible for sending appropriate alerts/information relating to laptop security violations. The system was also designed to support alarms at the institution’s main gates, SMS to a staff member assigned the particular laptop, security personnel along with asset management personnel in charge of assets at the institution. This process is mirrored in Violation Detection in section 4.3.4. Figure 4.16 below shows a sample of an SMS alert message sent out, to conscientize users about the events that have taken place.

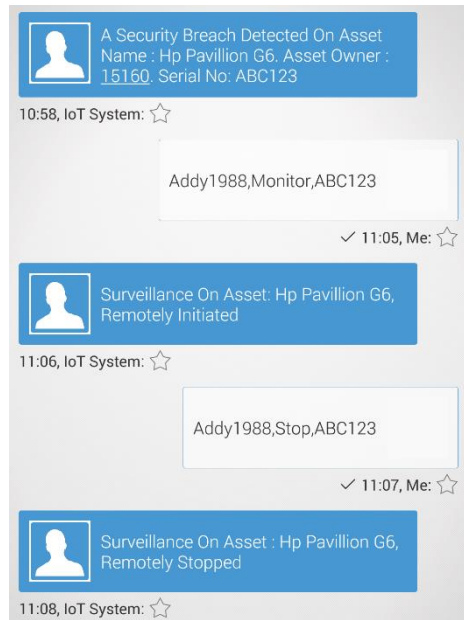


Figure 4.16: Communicated messages

Figure 4.16 depicts this process in action, along with its outcome. In this case value ‘Addy1988’ represents the user account password, ‘Monitor’ denotes the command and ‘ABC123’ represents laptop serial number. The user’s mobile phone number is the unique identity used by the system. If parameters are okay, the system responds by sending feedback to the user as depicted in SMS extract above (Figure 4.16).

4.3.8. Recovery

This module handles some of the activities needed to track and recover a stolen laptop. Tracking was made possible by utilisation of the built-in Windows Location Sensor available in Windows 7 and 8 which has the ability to supply location data to applications, based on Wi-Fi triangulation and IP address data. Microsoft Windows SDK exposes location service interface, through a dynamic link library (DLL). Any application can use this interface to interact with location service and query location data from the native code layer as it becomes available, or manipulate level of accuracy required by the application (Msdn.microsoft.com, 2014a). The sample code used to extract geographical position data is shown in Appendix 5 (Geographical position code).



Figure 4.17: Laptop recovery

Google maps showing the Laptop's location based on GPS newly generated coordinates stored on the database.

This simplifies programming because it became possible to integrate location services into the prototype through extraction of latitude and longitude data and save these to system database for each computer running the monitoring and tracking system. The middleware has a service that query location data after a given interval and save the data to the database. This was done such that current and most recent location data can be used to track the location of the laptop. The system prototype has commands to automatically start running in the background once someone signs into the computer, and starts broadcasting location data as discussed under section 4.1.1. This means the prototype system needs internet access to be able to access the remote database server. Figure 4.17 shows the results of using the middleware location services to determine the laptop's locus point, here Google maps are used in conjunction with location data (29°07'19.6\"S, 26°12'53.6\"E) that is stored in the database server.

The sequence diagram that explains how recovery is achieved is shown below.

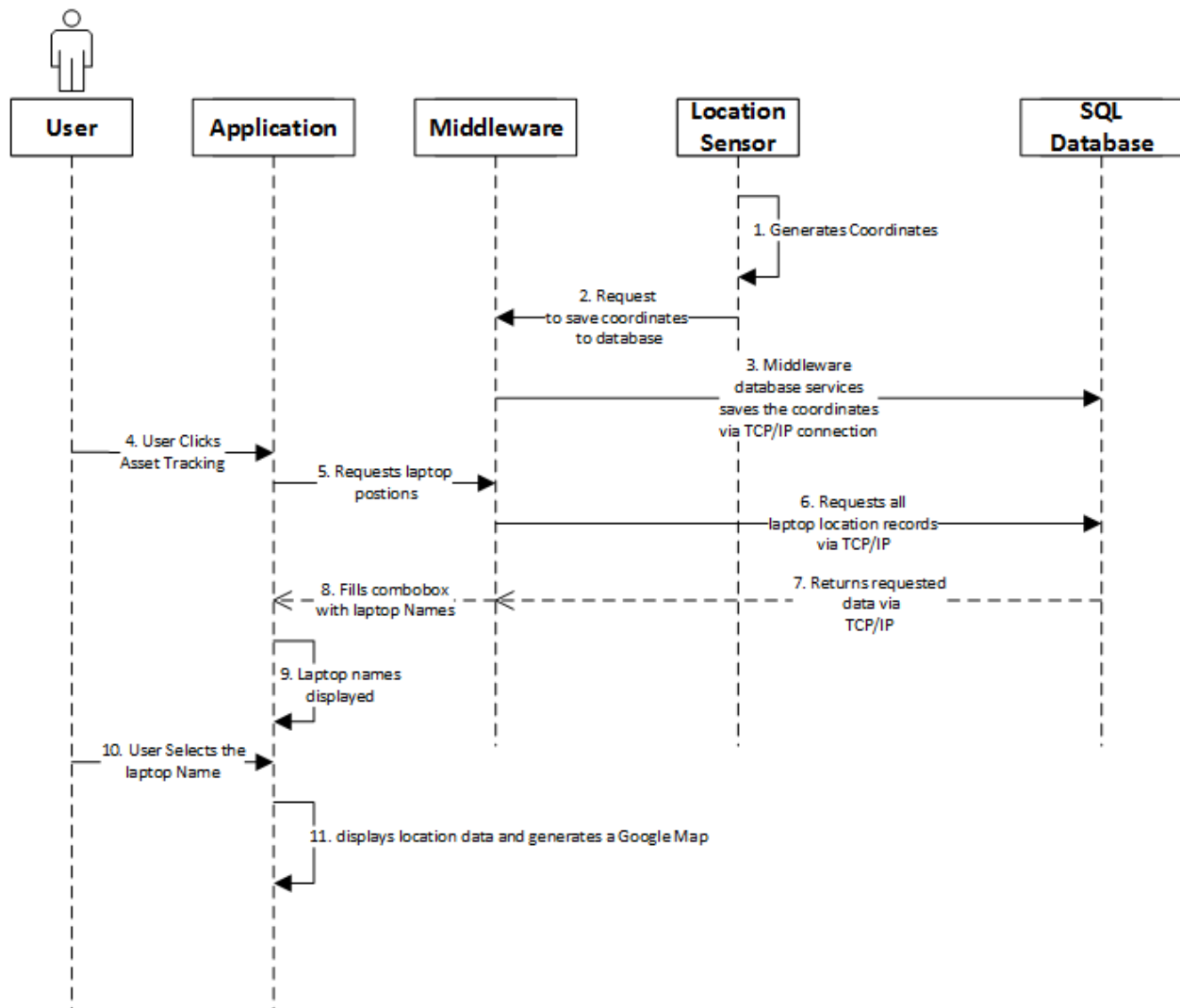


Figure 4.18: Recovery sequence diagram

5. Chapter 5: System Testing and Results

System evaluation was a perpetual process throughout the development life cycle of the prototype. The Agile developmental approach (particularly extreme programming) provided this study with unlimited system performance testing opportunities, beginning at requirements gathering stage. This early stage performance analysis offered great developmental directions in identifying overall system functions and how they would all be linked together.

The goal behind this early stage evaluation was to glean material and information about the project as a whole including, system functions, anticipated user actions, identification of middleware architecture and services, overall prototype architecture, and any other details that were deemed useful in directing the development process. Throughout the development process, system testing was conducted in the form of unit testing, regression testing and integration testing (Schach, 2008). The final test was conducted in the form of experimental cases, in which the researcher meticulously observed and documented the behaviour of the system prototype. The documented results are shown and discussed in qualitative terms. These tests aided the researcher in meeting the objectives of the research project.

5.1. Experiment Case 1

In this study an experiment was piloted to evaluate database load stress performance and to better understand how the middleware database services were handling user requests. This test was conducted using a software tool called ‘SQLQueryStress Version 0.9.7.0’ (Datamanipulation.net, 2014), which was configured to connect to the remote SQL database server and simulate the load stress and how the remote database would handle multiple requests from different users.

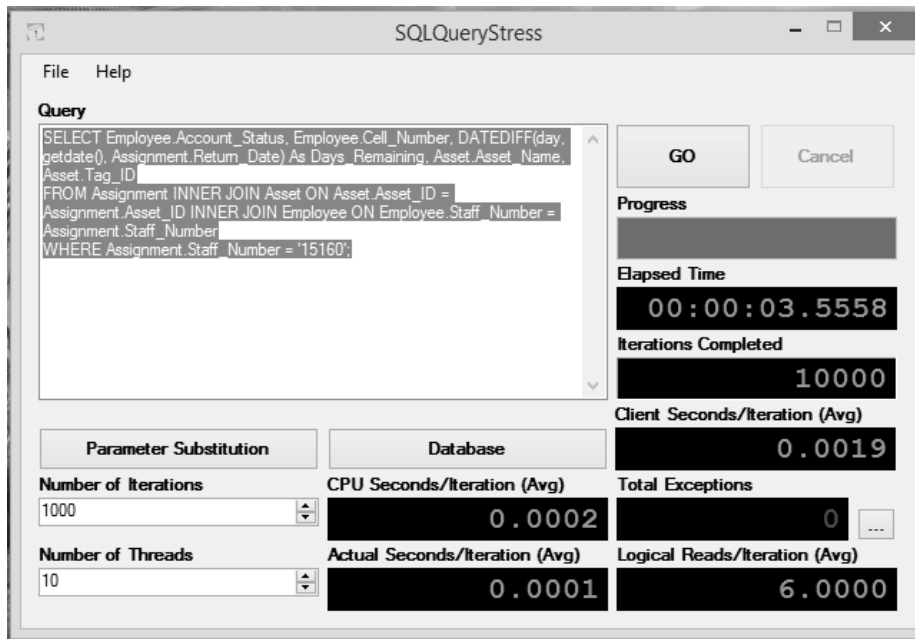


Figure 5.1: Database load test

Figure 5.1 above summarizes the results of a sample query that was run against the remote database server. The number of iterations was set at 1000 and the number of threads was set at 10; it took an average of 0.0002 seconds for the central processing unit (CPU) to complete the processing cycle. The total number of iterations was 1000 processes multiplied by 10 (threads) users which equals 10 000 processes, processed in 03.5558 milliseconds. This was a good response based on the researcher's observation, which confirms that the database design was well normalized; this would elevate the middleware's performance in handling of database functions to create, read, update and delete records without performance degradation.

5.2. Experiment Case 2

A second experiment was conducted to determine the overall response time the LMTS would take to communicate a security breach. This was done to find out if a perpetrator could be swiftly apprehended, before disappearing off the crime scene.

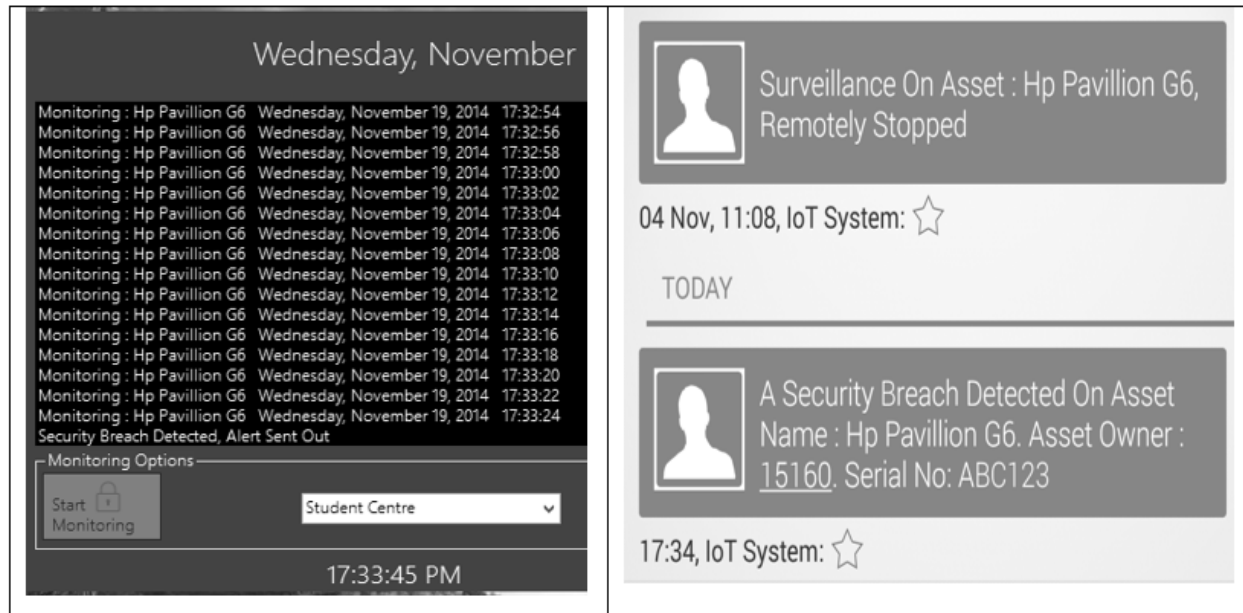


Figure 5.2: Breach communication time

Figure 5.2 above shows the results of the experiment conducted and how the middleware is effective in processing and communicating breach events. As seen above, LMTS detected a security breach at 17:33:24, the message was passed down to the middleware SMS services which saved the breach message to a database outbound Table. The new Table record was then picked up by the Ozeki SMS gateway with the help of background processes and conducted message transmission to its intended recipient(s). It is significant to mention that Ozeki SMS gateway was installed and configured on the remote server hosting the database. The message was delivered approximately within 36 seconds because the SMS excerpt is showing that at 17:34 the recipient was made aware of the breach.

The response time was considered reasonable especially after taking into account the fact that the LMTS is a prototype and the SMS gateway is using a Cell C SIM card. Cell C is the third biggest mobile network service provider in South Africa and the speed and efficiency of its services are no match to that of other bigger players (Vodacom and MTN). Although the researcher is satisfied with this observation, more work need to be done to further decrease the response time to approximately 10 to 15 seconds in future.

5.3. Experiment Case 3

A third test case was conducted to investigate the middleware location service's ability to accurately show a laptop's location using Google maps. In this experiment, one subject was asked to go into any building of his choice within the campus, but without revealing the building name or building location. Here the subject was supposed to connect his/her laptop to the internet using Wi-Fi or Ethernet cable, but was not mandated to run the LMTS, because it was configured to auto-start along with other Windows services at system start-up.

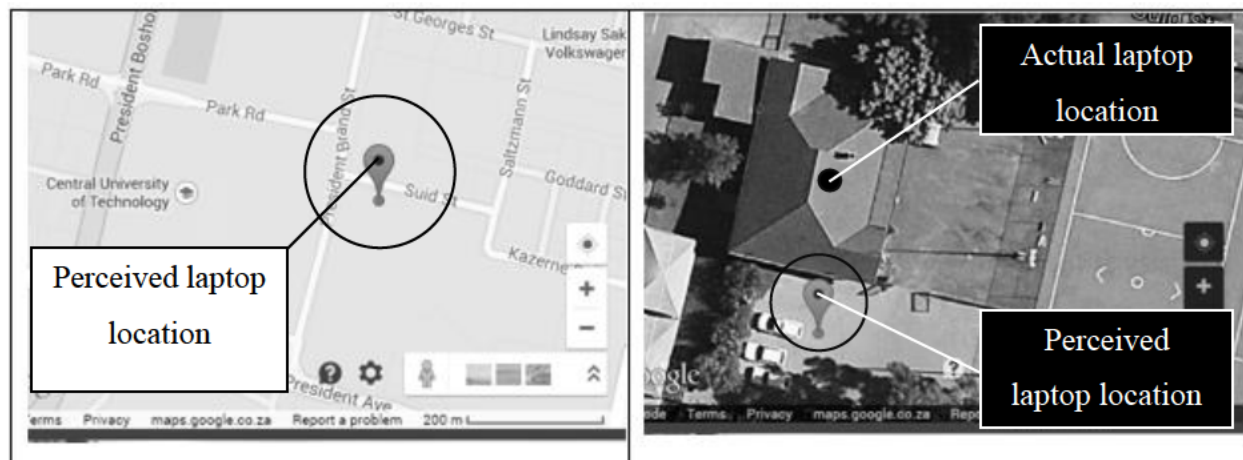


Figure 5.3: Location services

The results of the experiment are depicted in Figure 5.3 and the LMTS was beyond doubt able to show us the building into which the subject had gone. The building name was identified as Student Exchange house, located along Suid Street. The subject was called back, and the screen shots from Google Maps were presented to the subject for confirmation. The subject agreed but the LMTS had a minor error in coordinate computation because the subject was not using the laptop outside the building as shown by the screen shot (perceived location), but was actually inside the building (actual location).

The reason behind this inaccurate computation of coordinates was attributed to be the algorithm used by Windows location sensors to triangulate an object's location using data coming from IP address. It is imperative to point out the fact that laptops do not have GPS devices unlike smart mobile devices. Perhaps the results would have been different had hardware manufacturers integrated this location-aware sensor into laptops.

5.4. Experiment Case 4

A fourth experiment was conducted with the intention to observe the reaction the prototype would have if a registered user attempted to stop monitoring of a laptop that was initiated by another user. In this experiment, two subjects (A, B) were chosen to participate, while the researcher observed the events as they unravelled. These two subjects (A, B) were registered on the system and had laptops assigned to them (L1, L2) prior the experiment. Both participants were told to run the laptop monitoring software that was installed on their laptops (L1, L2). Now subject A was asked to attempt to stop monitoring of laptop L2 which was assigned to subject B.

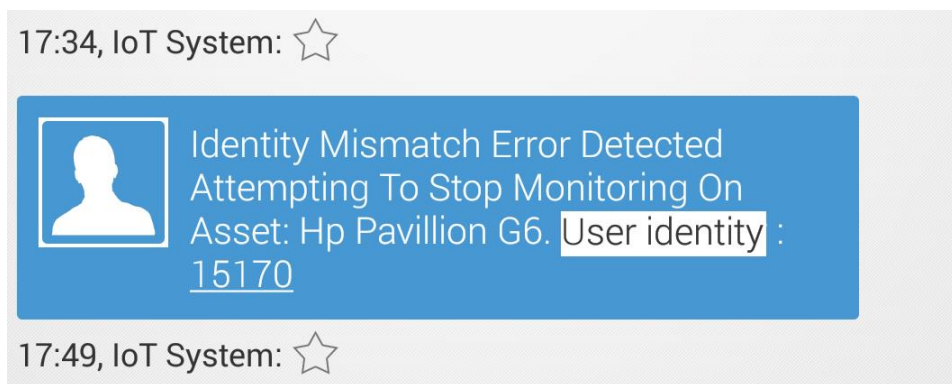


Figure 5.4: Identity mismatch error

The results of the experiment are shown in Figure 5.4, the system was intelligent enough to detect that subject A with identity 15170 had committed an identity violation on laptop L2 by sending an instant message via SMS to subject B. From this experiment the researcher was convinced that appropriate reaction was taken by the system and that the prototype was able to distinguish laptop owners based on their fingerprints that were used to initiate and stop the laptop monitoring process.

5.5. Experiment Case 5

The fifth experiment was intended to test the interaction between the middleware SMS service and Location Service. The logic behind this experiment was to establish if the system was able to deliver the location of the laptop to a user, via SMS. The user simply sends a query to the system via SMS, using a specific command. The middleware then decodes the SMS query and instructs its services to trigger the respective events, which handles the user request.

Figure 5.5 below shows the query the user used to request the location of the laptop. The SMS query sent to the system comprised three parameters: (1) the user's mobile phone number was used as the user identification element on the system; (2) the password used to authenticate to the system; and (3) the command that instructs the application middleware was kind of service is being requested by the user. The middleware then triggers the events to handle this service request.

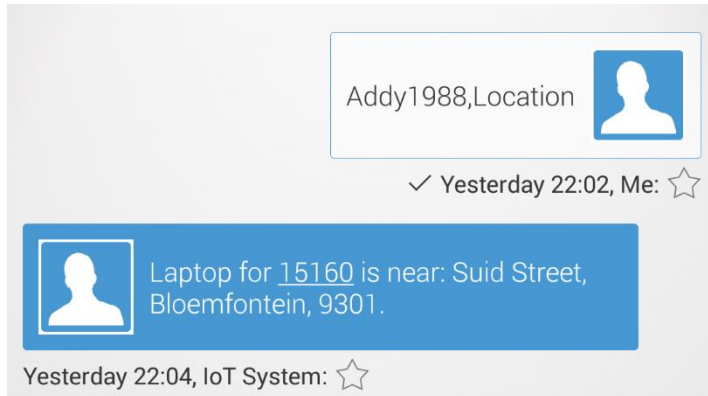


Figure 5.5: Location finder query

As illustrated in Figure 5.5, the middleware was smart enough to translate the GPS coordinates saved on the database into a physical address. The response from the middleware comprised the following: (1) the **street name** (Suid Street); (2) **city** (Bloemfontein) in which the laptop is; and (3) the **zip code** (9301) associated with the city. This aspect of the system, proves that if hardware manufacturers can include GPRS and GPS hardware modules in laptops, laptop owners can simply query the location of the laptop from a cell phone; hence making laptop tracking easy. This technique is ideal because it eliminates the need to have a GPS module turned on all the time which might significantly waste the laptop's battery energy. The spike solution implemented to determine the feasibility to transmute GPS coordinates into the above location details can be found in Appendix 8 (Location finder code).

Figure 5.6 below illustrates the sequence of activities that took place to achieve the goals of the experiment: (1a) user sends the SMS command shown in Figure 5.5, (2a) GPRS network provides network connectivity to transmit message; (3a) SMS relayed to destination; (4a) SMS is delivered and Ozeki background services, detects the inbound SMS; (5a) Ozeki services triggers events that saves the inbound SMS to database entity called 'ozekimessagein' shown in Figure

4.7; (6a) SQL service broker detects new database record stored in 'ozekimessagein' and broadcast it to all laptops connected to the database via TCP/IP connection; (7a) the middleware of each laptop triggers an event handler upon receiving a copy of the message; the event handler does decode the SMS, to determines the laptop intended to process the message; this is achieved by another middleware service that fetches the record from the database using cell phone number as the identification parameter.

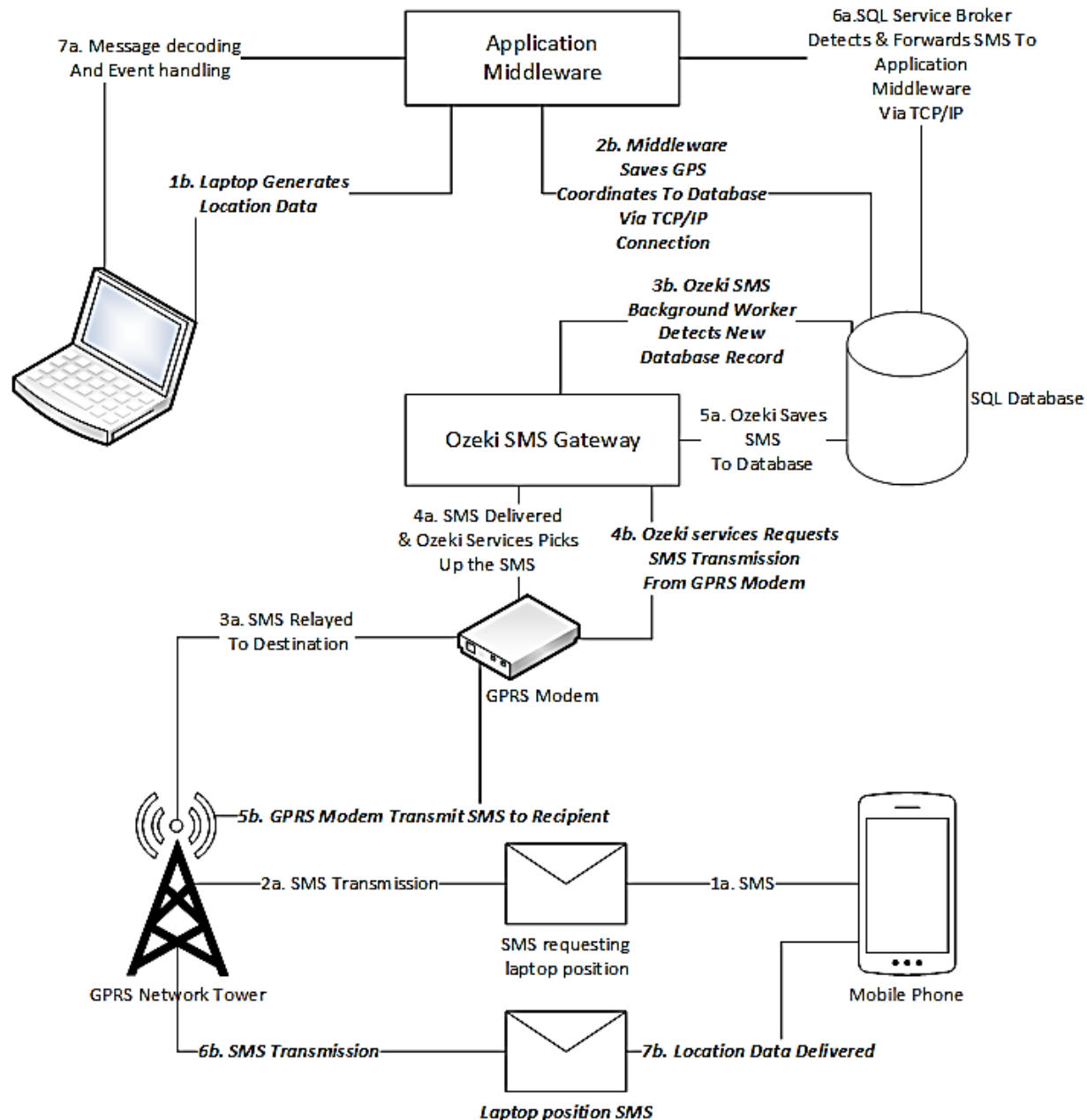


Figure 5.6: Laptop tracking using query SMS

The retrieved record has the '*laptopName*', which the middleware uses to compare to the '*MachineName*' parameter requested by the middleware from the operating system. If the '*laptopName*' (from the database) matches '*MachineName*' (from the operating system), then the laptop processes the message and the other laptops drop the message.

Activity (1b) signifies the action taken in response to the request made by the user; (2b) middleware saves the location information to the database entity called '*ozekimessageout*' as depicted in Figure 4.7; (3b) Ozeki detects the outbound message; (4b) Ozeki requests message transmission from GPRS; (5b) GPRS initiates message transmission; (6b) the GPRS network tower is propagating the message; (7b) laptop location is delivered to user via SMS.

5.6. Experiment Case 6

The application's middleware was again assessed to determine its behaviour, if the system was changed from utilising a centralised database, to use a distributed database setup. This goal for this experiment was to investigate fault tolerance and flexibility of the system. To achieve this, the researcher spread the system's database across three different and independent computers that had Microsoft SQL Server installed on them. For this to work the researcher added an interface to allow users to configure database connection parameters such as: (1) database server IP address; (2) database user name and user password; and (3) database name.

The middleware was also configured to allow database connections to be changed at runtime, without the need to rewrite the code for database services. Any database connection that was added was saved to a file called '*ConnectionString*'; it is this configuration file that facilitated the execution of this experiment. The Figure 5.7 below shows the interface that was used, to add multiple database connections and Figure 5.8 depicts the response users get upon clicking 'Save' button in Figure 5.7.

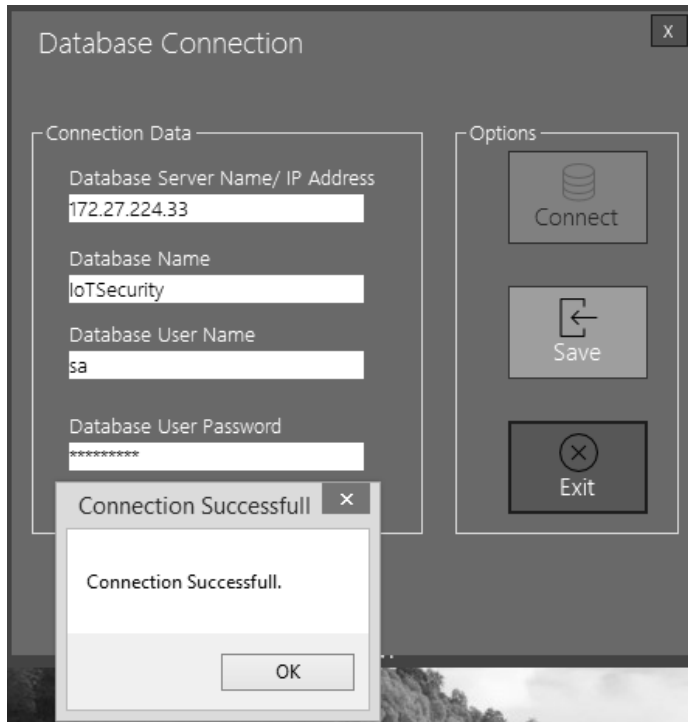


Figure 5.7: Database connection interface

This was the setup needed for the experiment to take place. After all the connections had been saved to a file, the experiment commenced. The goal was to test if the database could switch between connections if it failed to establish a connection to one database server. The three database servers were randomly taken offline, while the system was executing. It is unfortunate that it was impossible to capture this behaviour in action, because the execution of this took place behind the scenes.

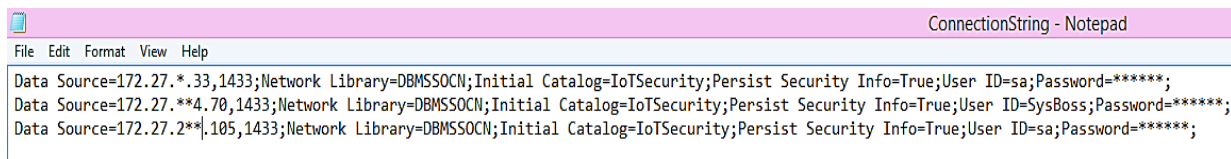


Figure 5.9: Database server connections

However, the system middleware was able to automatically switch connections, looking for an active database server that was reachable. The middleware loaded connections from the file 'ConnectionString' (Figure 5.9), and saved them in a temporary array structure; whenever the system could not establish a connection to the database, the middleware would trigger an event that loops through connections in the array structure looking for one that would return a

successful database connection. The middleware would continue to use the active connection until it fails. The middleware passed the experiment and showed exceptional behaviour, which qualifies it as fault-tolerant and flexible.

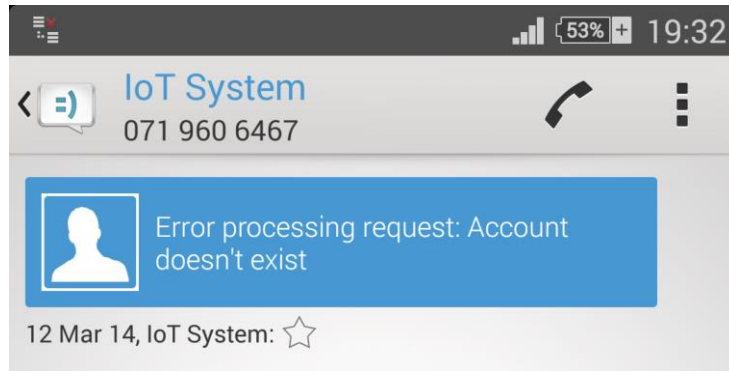


Figure 5.10: Error handling

To corroborate the fault tolerance of the middleware, another experiment was conducted. In this case, the researcher wanted to observe what would happen if a user who is not registered, thus unknown by the system attempted issuing an SMS command. The results are displayed above in Figure 5.10. The system uses the cell phone number to distinguish between registered and unregistered users. In this case, it can be deduced that this cell phone number is unknown.

5.7. Experiment Case 7

The seventh system test is a continuation of the experiment case 6 above. Here the researcher wanted to find out the behaviour of the system if all three database servers were taken offline. The system did not encounter a fatal error; instead, the database service of the middleware has an agent that kept trying to establish a connection to either of the database servers known to it. To better understand what happened, activity logs generated by the system were examined. Activity logs were discussed in detail in section 4.2.3 of Chapter 4.

```
A socket operation was attempted to an unreachable network.)
A socket operation was attempted to an unreachable network.)
A socket operation was attempted to an unreachable network.)
A connection attempt failed because the connected party did not properly respond after a period of time,
A connection attempt failed because the connected party did not properly respond after a period of time,
A connection attempt failed because the connected party did not properly respond after a period of time,
A connection attempt failed because the connected party did not properly respond after a period of time,
A connection attempt failed because the connected party did not properly respond after a period of time,
A connection attempt failed because the connected party did not properly respond after a period of time,
```

Figure 5.11: Activity log data

The importance of log files should never be overlooked or underestimated; the middleware has agents that observe activities and report any peculiar behaviour detected. Figure 5.11 serves as proof of the paramount importance of these log files. From the depiction, we can observe that the system indeed attempted to connect to either of the three database server but none of the servers responded to the connection requests sent by the system.

5.8. Experiment Case 8

The prototype attempted to solve many of the problems the researcher learned about during the case study discussed in section 3.4 of Chapter 3. Here the researcher endeavoured to automate most of the services that were being conducted manually. One service that was conducted manually and inefficiently is Asset Assignment. In this experiment, the researcher wanted to qualitatively communicate the experience of accomplishing this task using a software tool.

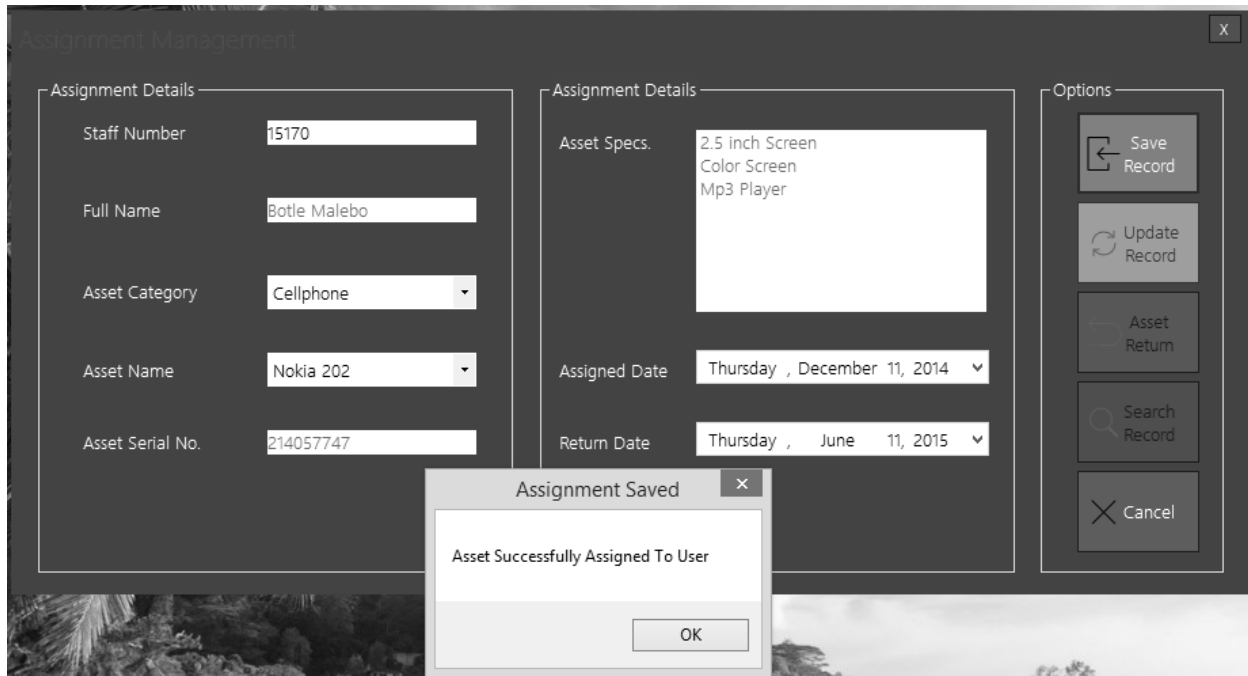


Figure 5.12: Asset assignment

Figure 5.12 shows the system interface that was used during this experiment. According to the researcher's experience, this was by far the easiest way any person with limited computer skills could use to assign assets to users. The researcher completed this task using five mouse clicks. First click was used to select the task the researcher wanted to accomplish, here the researcher clicked 'Create Record', which was automatically renamed to 'Save Record' after clicking the button. Second step was supplying the user employee number, which was '1570'; after punching in the last digit, the system automatically retrieved the employee name from the database. The third click was used to select the asset category and the researcher clicked on 'Cell phone'; the system then retrieved a list of cell phones that were available for leasing. The fourth click was used to select the model of the cell phone from the list. Here, the system also automatically fetched and displayed the specifications of the cell phone the system user had chosen. The system was also intelligent enough to automatically fill in the date of asset assignment and return date. By default, the policy of CUT allows leasing assets for a minimum period of six months and this explains the return date of 11June 2015. However, the system user, has a choice to override this behaviour by either shortening or lengthening the lease period.

The fifth click was used to click button 'Save Record', using the mouse pointer and *voila!* Employee 15170 had the cell phone successfully assigned to his/her possession. The overall

process took less than a minute to complete. The advantages of using a system to manage asset assignment are: (1) the system always maintains up to date records, which can easily be retrieved, updated and archived; (2) there is no room for manipulating the assignment process, because employees must produce their identity information to the system, which is again validated for authenticity; (3) the system automatically generates and communicates notifications, reminding employees when the asset lease period is going to expire via SMS; (4) The process is easy, efficient, and not time consuming; if a user-friendly system is used. Lastly, the system also handles asset withdrawal as depicted in Figure 5.12, 'Asset Return' button is associated with accomplishment of this task.

5.9. Experiment Case 9

In this experiment, the researcher carried on with testing more functions that were done manually as revealed by the case study investigation and discussed in section 3.4.1 of Chapter 3. The researcher went ahead to test Asset Registration. This entails capturing of data about new or old assets for auditing and leasing purposes, this feature was explained in detail in section 3.4.1 in Chapter 3. The goal behind this experiment is to also qualitatively communicate the experience of undertaking such a task using a software tool.

This task was slightly different from the one presented in experiment case 8; the major difference was limited automation, the system user was required to punch in most of the data required to create a comprehend asset record. The user had to provide asset details such as asset name, asset category (for example, laptop, tablet, cell phone), asset status (for example, available meaning asset is ready for assignment, damaged meaning asset needs repair), asset serial number, asset tag number (which was scanned from the tag attached to the asset), asset specifications, date acquired and the value of the asset in monetary terms. Figure 5.13 depicts the interface used to capture the asset details.

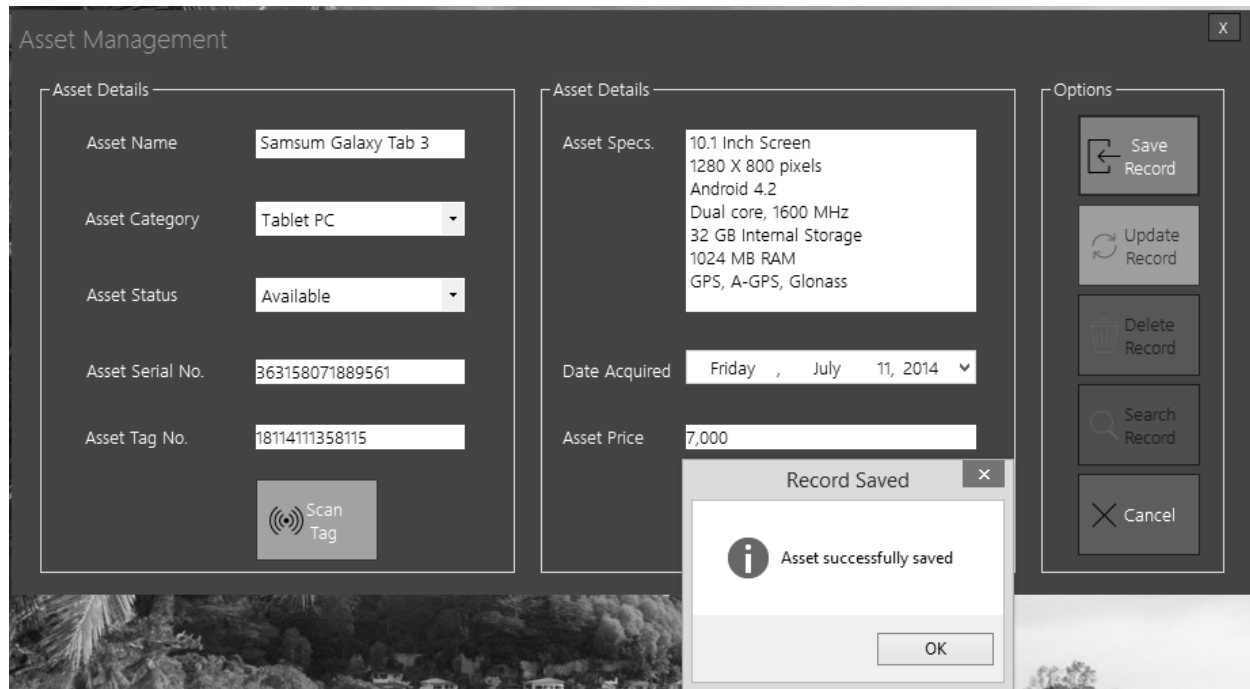


Figure 5.13: Asset registration

Once all the details of the asset have been supplied in their respective interface fields, the user simply clicks the button ‘Save Record’, which instructs the middleware database services to fetch this data and save it to a remote database. The system also communicates to the user the state of the record-saving process. As depicted in Figure 5.13, there is message informing the system user about the success of the task. The advantages of using this system, far outweighs the manual system and again the middleware database utilities provides a way to easily and instantly retrieve and update records.

5.10. Experiment Case 10

This final experiment was conducted to observe the convenience of using over-the-air computing (OaC) commands remotely to control and perform system tasks from a mobile phone. Here, the subject that volunteered to participate in this experiment was given a mobile phone whose SIM card number was registered on the system. The subject was then instructed to send three different commands: (1) to request a new remote system access password using an SMS command formatted in this manner (*‘current password’, ‘new password’*); (2) to set a new remote system access password of this choice using an SMS command formatted in this way (*‘current*

password, *change password*, *user defined password*); and (3) to remotely control the laptop's power options by sending an over-the-air computing (OaC) command in this fashion (*current password*, *power option*). Power option commands that the system was able to process include *shutdown*, *hibernate* and *logoff*. The results of the experiment are depicted below.

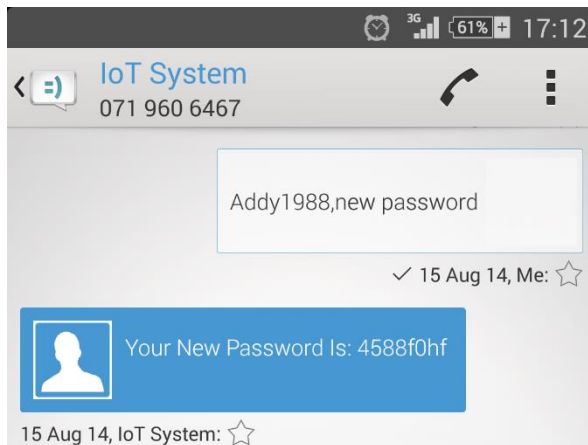


Figure 5.14: Password request

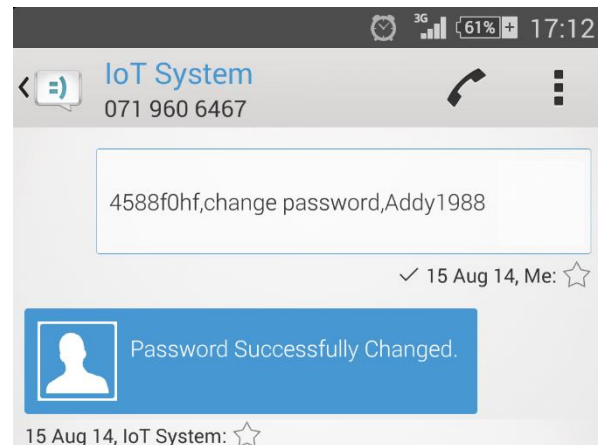


Figure 5.15: Password changing

Figure 5.14 and Figure 5.15 show that the system successfully received and processed these different over-the-air computing (OaC) commands sent from the subject's mobile phone. This serves as proof that, mobile phones are powerful resources in IoT because they can be used to generate information, control other systems using SMS commands or special software such as 'TeamViewer' and display the state of those systems. This study has demonstrated the convenient use of simple SMS commands sent remotely, to conduct different system operational activities that would have required the user to do them locally. Another activity that the subject was able to accomplish remotely using over-the-air computing (OaC) commands was controlling the laptop's power options. The figure below provides evidence to confirm this finding.

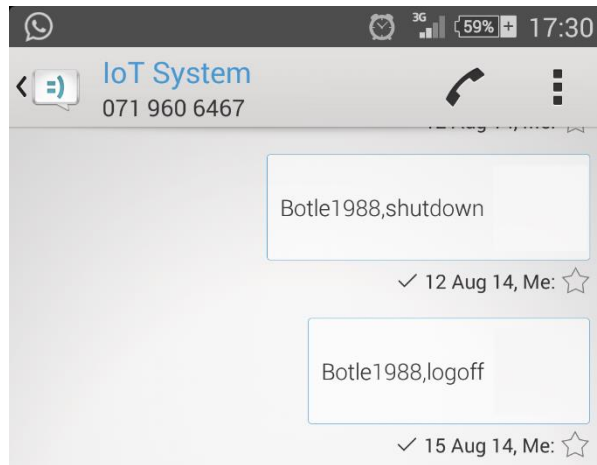


Figure 5.16: Power control commands

Figure 5.16 shows the commands that the subject sent to the LMTS prototype. The system's middleware was programmed to manipulate Windows operating system in order to accomplish these tasks. However, at the time of writing, the prototype was unable to give users feedback via SMS because the laptop would be turned off; meaning the middleware services responsible for communication would be inactive since the laptop is off.

Based on the discussion of this chapter and evaluation of the system, it is not far-fetched or biased to conclude that there is considerable potential to solve these real-world problems using the prototype presented in this study. Although in its current state, it may not achieve much, but with further refinements, it may eventually result in a system that eradicates most of the problems identified at the case study.

6. Chapter 6: Conclusion and Further Work

6.1. Discussion

In this dissertation the researcher presented both the laptop monitoring and tracking middleware and a system prototype that implements the proposed middleware architecture. The middleware supports an array of services such as: (1) extraction of locus data from windows location sensor; (2) laptop monitoring using RFID reader and passive tags; (3) bi-direction SMS communication; and (4) utilisation of database services to facilitate data management through SQL commands. The LMTS middleware can be considered as a potential hybrid middleware solution, because of its versatility and adoption of characteristics found in commonly used middleware solutions such as MiLAN and Cougar. The versatility of this middleware was in its ability to support parallel event handling, without compromising the delivery of services and its capacity to take advantage of resources and services offered by windows operating system environment and management of diverse hardware such as fingerprint scanner, Arduino RFID scanner, and a modem.

The prototype presented in this research was tested for conformance to the requirements set during a case study conducted and presented in section 3.4 of Chapter 3. The results of this system have been thoroughly discussed in Chapter 5. The prototype was developed to serve as proof of concept; the idea behind development of the prototype is twofold:

Firstly, the researcher wanted to prove that a novel IoT-based application can be used to track laptops; despite maturity of technology in asset tracking, laptops still remain vulnerable to theft because, there are no hardware-based solutions known to the researcher that can remotely track laptops. Most laptop tracking software are expensive due to the yearly subscription model that many cannot afford; the researcher considered them less helpful because, if a perpetrator formats (erases) the hard-drive of a stolen laptop, the laptop tracking software is permanently wiped off. This leaves us with no concrete solution to allay laptop tracking.

To fully comprehend the weaknesses of using software on either a mobile phone or laptop for tracking purposes, the researcher learned that these tracking software rely mainly on two things: (1) an identity which normally is in form of an email address as depicted in Figure 6.1 below and (2) access to the GPS sensor on the device as shown in Figures 6.2 and 6.3 underneath.

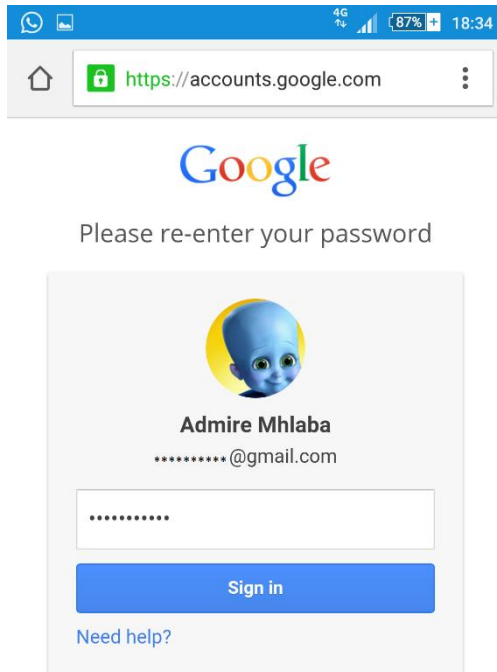


Figure 6.1: Account identity

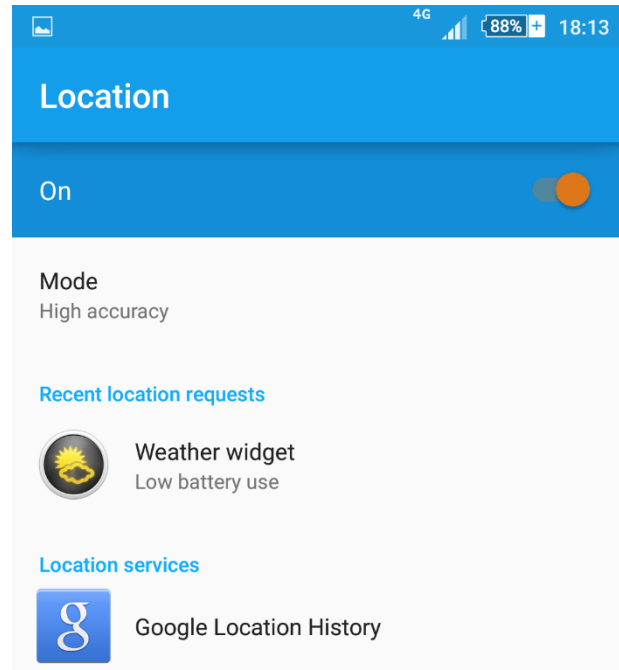


Figure 6.2: Mobile phone GPS access controller

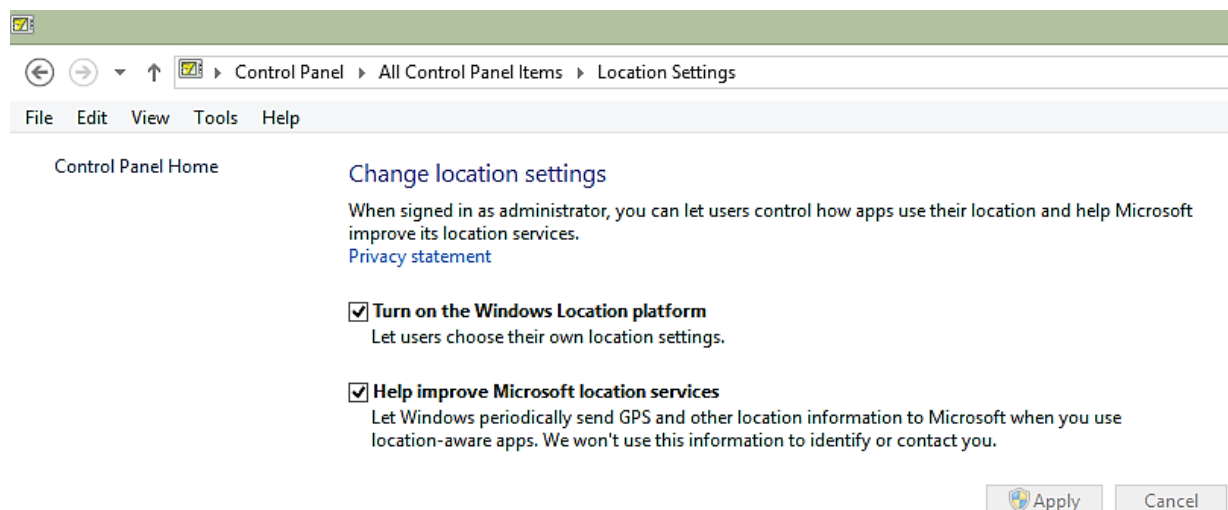


Figure 6.3: Laptop GPS access controller

As shown in both Figures 6.2 and 6.3, if the GPS sensor is turned on, the tracking software extracts locus data and relays it to a corresponding server (computer) for storage. In the case of mobile phones, these GPS coordinates are automatically synchronised with Google servers using the email address as the user account identity. This is one of the reasons why people who own

mobile phones running Android operating system are compelled to have a Gmail account prior using the mobile phone.

If a person loses his laptop or mobile phone, they can easily track it by using the email address and password to access a web-based service that queries the corresponding data servers for the most recent locus data. The geographical location of a device is then shown on a map using either Google or Bing map services as portrayed in Figure 6.4 using a service called *find my android phone* from Google.

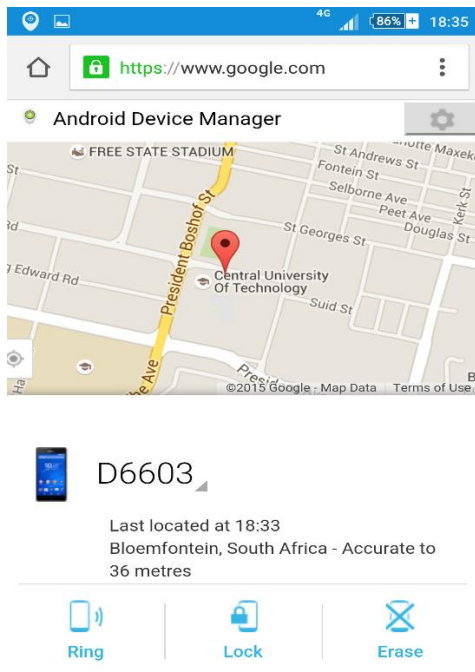


Figure 6.4: Find my phone service

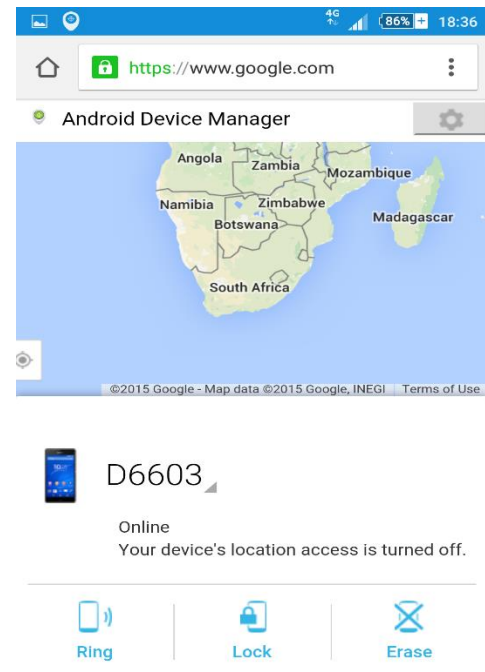


Figure 6.5: Device's location access

The drawback of this approach is evident when the laptop's or mobile phone's GPS sensor is turned off. The tracking software cannot extract GPS data used to determine the device's location as shown in Figure 6.5. Furthermore, if the culprit either uninstalls the tracking software from the laptop or resets the mobile phone to its default-state prior use, the account associated with the tracking software will also be removed. These two conditions diminish the tracking software's ability to recover let alone track stolen or lost devices. However, the tracking model presented in Chapter 5 Figure 5.6 was a software-based solution that mirrors the potential hardware-based tracking concept depicted in Figure 6.6 below that promises unparalleled capability to conquer device tracking and recoverability.

The Figure below highlights a potential concrete and permanent hardware based laptop tracking model.

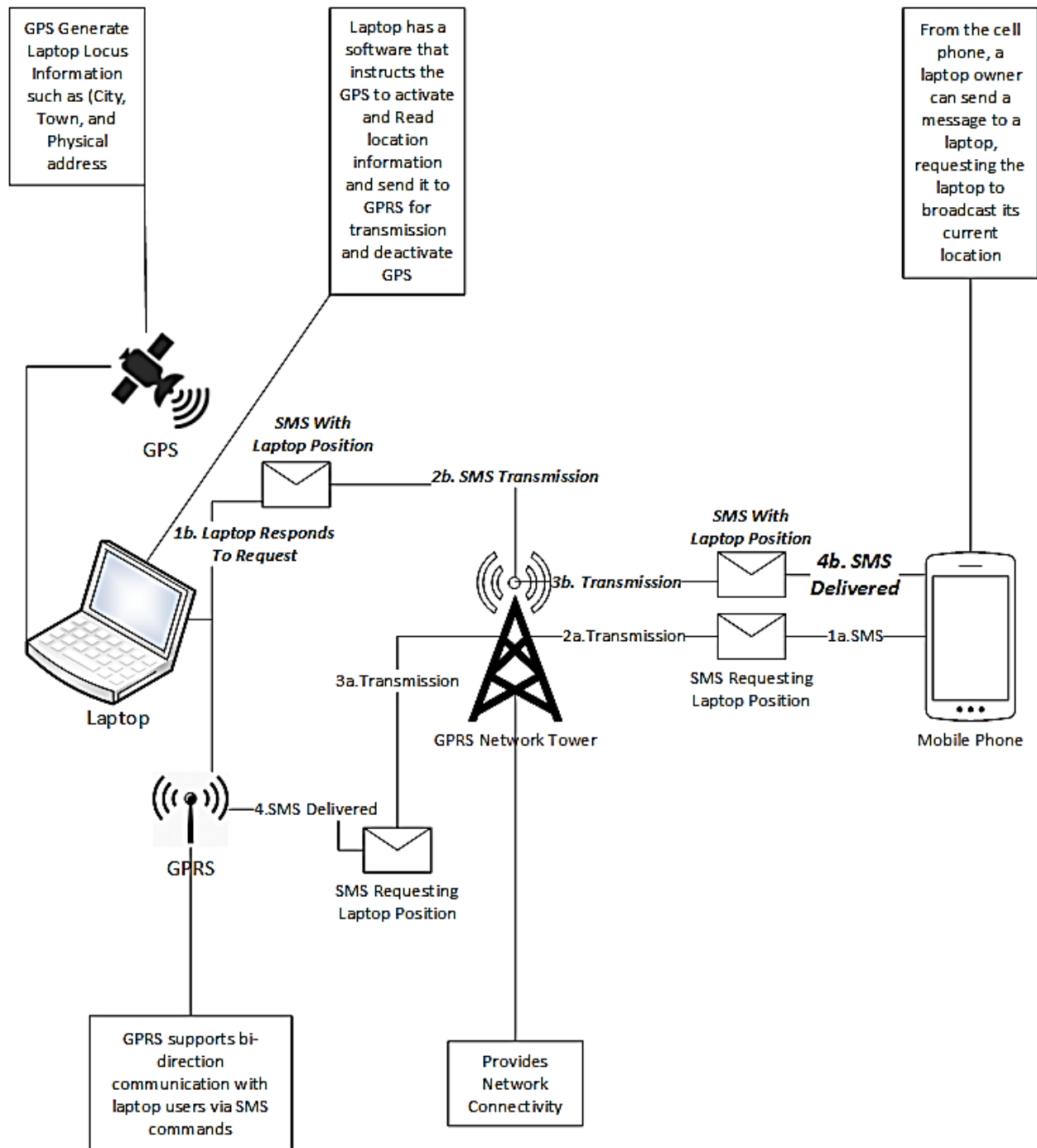


Figure 6.6: Hardware based laptop tracking model

This solution works as follows: (1) the laptop must be integrated with common GPS and GPRS hardware, but the GPRS module must have an irremovable SIM card. (2) The laptop serial

number should not be attached on the laptop using a label, but must be printed out on a slip given to the purchasing client as a secret code known only by the buyer, along with the contact number for the embedded SIM card; and (3) if anyone steals the laptop; then the victim of the laptop theft simply sends the serial number of the laptop to the contact number given to them; (4) upon receiving the serial number, the laptop's GPRS instructs the GPS module to generate locus data and send it back to the GPRS module for transmission (this eliminates the need to have an always-on GPS, which might rapidly drain the battery of the laptop); and (5) the laptop's GPRS communicates the location of the laptop with the victim, as depicted in Figure 5.5. This way we can totally proliferate recoverability of stolen laptops and circumvent theft of assets valuable to us.

The second reason was to prove that the problems identified during the case study had potential to be solved using a software tool. The results of the experiments conducted substantiate this potentiality.

In Chapter 3, the researcher presented the constructive methodology and experimental research design and tied the research to both qualitative and quantitative approaches of data collection and analysis. This chapter also discussed the case study and highlighted the problems being experienced by students, lecturers and the Asset Management Department and Security Department. The researcher investigated the case study domain through questionnaires, interviews and document analysis. A detailed report of the findings has been presented; which in so many ways shaped and influenced the manner in which the succeeding steps were directed. The case study results also aided the researcher in the achievement some of the objectives outlined in Chapter 1.

In Chapter 4, the researcher presented the laptop monitoring and tracking middleware architecture; the chapter outlined how the proposed middleware was used in integrating heterogeneous hardware used to answer the research question posed in Chapter 1. The middleware was fine-tuned, to help in addressing problems investigated during CUT case study discussed in Chapter 3. The resulting middleware was modularised to support various services mentioned above and functions, such as: (1) parallelism in event handling; (2) exploitation of Windows operating system services; and (3) fault tolerant. The chapter also presented the overall system design architecture and a database that provided the intelligence needed to facilitate the

interaction and delivery of diverse system services. The prototype was developed to utilise the middleware services to deliver of four major functions namely: (1) data capturing; (2) phenomena monitoring; (3) communication of events; and (4) recovery of objects; these functions were first presented in Figure 1.2 of Chapter 1.

In Chapter 5, the researcher evaluated the prototype that was designed through rigorous experimental cases conducted; this was done to conform to the research design presented in Chapter 3. The aspects of the prototype that were subjected to experiments are those that help to prove or refute the success of the proposed integration architecture and middleware design in answering the research questions that instigated this study. The elements of the system that were tested include:

- a) Database stress load intended to understand the ability of the system to handle scalability and its level of responsiveness to queries during operation peak times.
- b) Promptness of the prototype in detecting security breach and communication of those breaches via SMS.
- c) The ability to achieve laptop tracking using Google Maps and the accuracy level of windows location sensor in providing location data acquired through Wi-Fi triangulation or scrutiny of IP address data.
- d) Prototype vulnerability, in terms of the detection of deceptive actions that compromise the operation of the prototype.
- e) SMS query intelligence in delivery of geographical information.
- f) Fault tolerance and system flexibility.
- g) Analysis of activity logs to determine prototype malfunctions.
- h) The capability of the prototype in automating functions such as Asset Registration and Asset Assignment, which are conducted manually at the problem domain.

6.1.1. Hardware-based Tracking System Merits

The good news about the proposed hardware-based tracking system is that, the model depicted in Figure 6.6 is generic enough to be integrated into other valuable electronic assets (smart televisions, cell phones, tablets and cars) as well, because of the minuscule nature of GPRS and GPS hardware. The choice to employ GPRS and GPS hardware was influenced by their maturity and reliability levels, which emanates from years of research and tweaking. Additionally the tracking model utilises the generic middleware in-lining approach proposed in this dissertation, particularly the SMS and Location services to decipher and process over the air commands received via SMS and glean locus data from the implanted GPS hardware module.

The hardware-based tracking model presented in Figure 6.6 was discovered to be beneficial in the following ways:

- 1) Its implementation is cost effective because it uses infrastructure such as wireless network towers that have already been deployed by mobile network service providers (MTN, Vodacom and Cell C) and reliable hardware such as GPRS that has proven to be effective and efficient in managing bi-directional communication and GPS modules for generation of accurate geographical coordinates.
- 2) It harnesses the availability of strong network coverage of installed wireless towers and this expedites reachability of the most remote areas thus making asset recoverability more convenient.
- 3) It has the ability to take advantage of network roaming services which means a stolen asset can be tracked anywhere around the globe, anytime by anyone.
- 4) This model is not money gobbling unlike traditional subscription-based models, meaning once implemented, and this model has high potential to be embraced by the majority and deliver unparalleled asset security and recovery levels.
- 5) Implanting hardware was discovered as the ultimate resolution because any attempts to remove these tracking devices (GPRS and GPS) from the circuit board results in damaging the asset, unlike retro fits which can be safely detached from the device without causing any damage to the asset.

In conclusion, what makes this model outstanding is the fact that it uses middleware in-lining approach; equipped with this technique, it was discovered that instructions can be programmed into the operating system of mobile phones and laptops making it possible to achieve bi-direction machine (mobile phone) to machine (laptop) communication via SMS query commands and this can be accomplished even if the laptop or mobile phone has been formatted. The GPRS and GPS on the laptop get their power from the battery and laptops can also be redesigned to incorporate a back-up and irremovable small battery to power-up the GPRS and GPS hardware to cater for emergency and unforeseen scenarios. This hardware-based tracking model can be considered the only missing aspect in laptop designing because new laptop models such as Dell Latitude E7240 already have irremovable batteries, which guarantees perpetual energy supply to the tracking devices.

6.1.2. Theoretical Research Contributions

The adoption of constructive research approach entails the endeavour to strike a balance between both theoretical and practical contributions to the board of knowledge which can be used in future to solve similar or related problems by extending the given theories to come up with better constructs that can annihilate problems at hand. The main theoretical contributions this research has brought forth are: (1) the research firstly adopted and endorsed middleware in-lining concept and further enhanced the middleware architecture by extending its internal layers. It was discovered during the research process that the middleware architecture could be more versatile, resilient and responsive if a modular approach is adopted as explained in section 4.1.1 of this dissertation.

The merits of this modular approach were also explained in section 4.1.1 and evidence to corroborate these merits was presented in Chapter 5. (2) The experienced obtained from this work has resulted in coming up with a new definition of IoT; IoT can be thought as a concept of giving objects a digital identity and limited artificial intelligence which helps the objects to be interactive, process data, make decisions, communicate and react to events virtually with minimum human intervention.

Given this new definition, the researcher has provided adequate evidence to support the fact that not only do objects alert us of peculiar phenomenon and take corrective measures but through an

intelligent middleware, it is possible to interact with these objects. In this dissertation, this was accomplished through SMS communication. The reason why GPRS communication was used is because it was discovered that most African countries are not well connected to the internet but there has been a high mobile penetration level. So this study presented solutions that perfectly fit the realities of the continent's problems through harnessing already existing infrastructure.

This seamless and remote interaction between objects, systems and humans opens a plethora of opportunities. For instance, if a smart fridge is able to order food automatically, why then should we restrict this to one way communication? Will it be wrong to ask the fridge specific questions like "What to cook from the fridge? What food stuff have gone bad or expired? When does it expect to be cleaned?" On a different note, the security of cloud based storage services such as iCloud, Google Drive and One Drive to mention but a few, was tainted due to hacking (illegal access) into people's private accounts and publication of sensitive material. People now have little faith in these services despite numerous efforts to spruce this up. Now imagine a scenario like this: you forget an important presentation on your computer at home and it is not stored on any cloud storage service because it contains sensitive company information. Will not it be nice to send a simple SMS to your laptop requesting it to search for your presentation and email it to you? You can even narrow down the searching time by specifying the location of the presentation.

The study also revealed that middleware in-lining approach permits customisation of middleware layers and services, which means that functions from revered middleware such as SOA, MiLAN, and Cougar to mention but a few can be borrowed and integrated into one hybrid middleware solution. This qualifies the proposed middleware architecture to be flexible, reliable, resilient, proactive, reactive and generic enough for use in logistic applications such as fleet management system used to monitor and track movement of cargo. The presented middleware can be utilised as well in diverse environments (academic, military, medical, and manufacturing) that require real time asset monitoring, tracking and processing of voluminous data gathered from heterogeneous hardware.

This envisioned object, system and human interaction concept was the salient theoretical contribution though it was used in a security context, this can also be applied in different scenarios as explained above. The research has proven that it is possible to virtually query any

electronic device to share or broadcast its geographical position in the event of being lost or stolen making asset tracking and recoverability a convenient process. This is conceivable through adoption and standardisation of the hardware-based tracking model presented in 6.1.1 and the enhanced middleware architecture in section 4.1.1.

The hardware-based asset tracking model is a better alternative to the infamous “Kill Switch” (*is a safety mechanism used to shut off a device in an emergency situation in which it cannot be shut down in the usual manner*) approach postulated by Apple which has the following features: (a) *completely and as quickly as possible abort the operation, even if this damages equipment;* (b) *be operable in a manner that is quick, simple (so that even a panicking operator with impaired executive function can activate it), and, usually, and (c) be obvious even to an untrained operator or a bystander* (Wikipedia, 2015). The “Kill Switch” approach does not assist in asset recovery but makes devices un-usable or inoperable, whereas with the hardware-based tracking model proposed in this study, device recoverability is guaranteed, which saves people money to replace the lost or stolen device and there is no damage whatsoever done to the device being recovered.

Lastly, the researcher presented adequate evidence to support the fact that the research questions had been answered and the objectives of this study have also been met.

6.2. Implementation Challenges and Solutions

During the system implementation phase there was the need to create an SMS Gateway using a Wasmote sensor integrated with GPRS SIM 900 module. The gateway was created successfully and was able to send and receive messages with issues unbeknown to the study investigator. However, the gateway had performance issues emanating from both hardware and software code written into Wasmote sensor as shown in Appendix 6 (Wasmote gateway code). The code was written in such a way that the sensor listens for incoming messages for 60 seconds and listens for outgoing messages for 30 seconds. The problem was, if the sensor was busy listening for incoming messages and an outgoing message arrives, the sensor was unable to break the incoming message listening cycle before 60 seconds have elapsed. Meaning the outgoing message would only be sent after the sensor is done with incoming cycle.

This created a significant delay, especially when a critical event that needed immediate attention has occurred. On average the sensor would take ± 90 seconds to communicate a security breach and this was a very huge delay. However, several attempts to improve the situation were conducted, but it did not yield any satisfactory results. The researcher also noted that the sensor was not so reliable because, if the prototype sends at most 25 messages, the sensor would jam, because the space in the subscriber identity module (SIM) card would be full and needed to be freed by deletion of old messages. An algorithm to address this was embedded within the code, but still the results were not satisfactory, performance wise. The researcher's study leader then proposed and procured Ozeki SMS Gateway software which ironed out this technical problem.

Moreover, Ozeki solved two problems: (1) the response time was significantly reduced from approximately 90 to 40 seconds; and (2) the SIM card issue was taken out of the equation because Ozeki saves both incoming and outgoing messages to respective database tables with infinite space.

However, the researcher's joy was short-lived because, along the way another logical problem cropped up. The problem was, the prototype was using a remote database and once Ozeki receives an incoming message it saves it to a database Table. The question was, how to alert all systems connected to this shared database that there was a new message awaiting consumption or processing. The researcher attempted querying the incoming Table every second for new records, but that had performance issues and this solution was ditched.

After reading a lot of online articles, the researcher stumbled across an article on Microsoft developer network (MSDN) about SQL Service Broker an extract of this is highlighted in Appendix 7 (SQL broker code). *“Application developers who use Service Broker can distribute data workloads across several databases without programming complex communication and messaging internals. This reduces development and test work because Service Broker handles the communication paths in the context of a conversation. It also improves performance. For example, front-end databases supporting Web sites can record information and send process intensive tasks to queue in back-end databases. Service Broker ensures that all tasks are managed in the context of transactions to assure reliability and technical consistency”* (Msdn.microsoft.com, 2014b). This is what solved the problem, but then another issue was noticed.

Since the application is able to receive and process commands from users via cell phone, how then does the system know which computer is supposed to consume a particular message. This was solved by associating each user's mobile phone number with a unique computer name also referred to as '*MachineName*' in Visual Studio Environment.

Moreover, another technical problem that the researcher wasn't able to resolve was the need to extend RFID read range using a wireless Waspote sensor. This is yet another area worth investigating and investing research time into. Lastly, the researcher wanted to find a more intuitive way to track a laptop's geographical position without having a lot of wires connected or attached to the laptop. Most computers do not come integrated with GPS hardware and people tend to devote to plug and play universal serial bus (USB) peripherals to add the needed features. This issue was solved by using Microsoft Location Services, which uses virtual sensors to acquire location knowledge through the use of Wi-Fi triangulation and IP address data. This was explained in detail in section Chapter 3 under Asset Tracking.

Despite this success, there are tons of more things to be done in order to make the system more robust, resilient and efficient. The subsequent future system recommendations to be considered are discussed below.

6.3. Further Research Work

6.3.1. Future Activity 1

During the case study conducted and presented in Chapter 3 section 3.4, the researcher alluded to the disjointment of diverse security systems in place at CUT as the major reason behind the sky-rocking menace of laptop theft. The solution presented in this study did not endeavour to provide a solution to this existing problem. This entails that the system prototype along with the middleware architecture needs reviewing and perhaps redesigning in order to cater for this aspect of paramount importance. The idea behind this is, for example, the RFID readers and tags are used in detecting security breaches and trigger alerting events, the system prototype should not only send SMS notifications to the victim and security personal, but should have the ability to control other objects such as locking of doors and windows of the office of laboratory where the crime has taken place. This also involves triggering alarms and activating the wireless video

camera to record the activities as they unravel in real time. This envisioned asset monitoring and tracking system could therefore be used to dispel the increasing theft of government sponsored laptops and tablets in high schools around South Africa.

6.3.2. Future Activity 2

Throughout the study, from the early warning model presented in Chapter 1 Figure 1.2, to the middleware architecture presented in section 4.1.2 and Figure 4.2 in Chapter 4; the study introduced the need to communicate system events via SMS; this was further implemented and tested in Chapter 5. The researcher noticed the efficiency and reliability of this service and now contemplates extending its capabilities in the prototype. As demonstrated in Chapter 5, the middleware supports bi-directional communication with users using cell phones; this means the system can be controlled from cell phones by just issuing simple commands.

The researcher would like to add an over-the-air computing (OaC) functionality that allow users to extend the asset lease date further when it is about to expire. Currently in its state, the prototype allows this, but to achieve this, an asset holder must consult Asset Management personnel to do this on behalf of the asset holder. To add a bit of flexibility, this function will be done also from a cell phone, using a simple SMS command formatted as follows: ‘*Password*’, ‘*Extend*’, ‘*Days*’; the password is used to authenticate the asset holder to the system along with the cell phone number as explained in section 4.3.7 of Chapter 4; the term ‘*Extend*’ denotes the command which can easily be translated to a request to add more days to the asset lease period; here ‘*Days*’ represents the amount of time the asset holder wants this lease period extended by.

6.3.3. Future Activity 3

In this study, the researcher introduced a technique called Geo-Fencing; in the study the researcher explained that laptops were being monitored using a passive tag and RFID readers designed using Arduino microcontroller, as explained in section 4.3.6 of Chapter 4. The researcher plans to elevate the prototype’s ability to monitor mobile assets. To achieve this, there is a need to integrate Geo-Fencing function to allow the system prototype to report the entry or exist of a stolen asset into CUT premises. This was hard to achieve because the passive tags used in the current implementation can be read by Arduino RFID reader, from a very close range of

about 3 centimetres. This entails deployment of WSN devices with RFID tag detection, to conduct monitoring and reporting services; learning power optimization routing algorithms and data transmission protocols. However, as hardware technology continues to become more advanced, it will soon be possible to have tiny active RFID tags that can be embedded into laptops while supporting longer read ranges.

6.3.4. Future Activity 4

Lastly, the researcher wishes to engage laptop hardware manufactures and investigate the possibility to embed GPS and GPRS devices, which laptop users can control remotely. This could assist in recoverability of stolen laptops. For example, if a laptop could be embedded with a GPRS module that has an irremovable SIM card, then laptop owners who were given this unique SIM card number at the time of purchase, can send an SMS to this number requesting the laptop to broadcast GPS coordinates. If this can become a standard feature in laptop design, then we could see a significant reduction in laptop theft. This was thoroughly discussed in the beginning of this Conclusion chapter.

References

- Agar, J. (2013). *Constant touch: A global history of the mobile phone*. Cambridge: Icon Books.
- Aguilar, G., Sánchez, G., Toscano, K., Miyatake, M.N. & Meana, H.P. (2008). Automatic Fingerprint Recognition System Using Fast Fourier Transform and Gabor Filters. *The proceeding of Scientifica*. Vol. 12, no. 1, pp. 9-16.
- Ahson, S. & Ilyas, M. (2008). *RFID handbook*. Boca Raton: CRC Press; pp. 3-10.
- Aker, J. & Mbiti, I. (2010). Mobile Phones and Economic Development in Africa. *Journal of Economic Perspectives*, 24(3), pp.207-232.
- Alemdar, H. & Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15), pp.2688-2710.
- Arendarenko, E. (2009). A study of comparing RFID and 2D barcode tag technologies for pervasive mobile applications. Master's Thesis. University of Joensuu, Department of Computer Science and Statistics.
- Aristotle, Greek Philosopher (384 BC–322 BC).
- Arshad, J., Farooq, A. & Shah, A. (2010). Evolution and Development towards 4th generation (4G) mobile communication systems. *Journal of American Science*, 6(12) (1545-1003), pp.63-68.
- Ashton, K. (2009). That 'internet of things' thing. *RFiD Journal*, 22(7), 97-114.
- Atzori, L., Iera A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks* 54 (2010) 2787–2805.
- Azira, E., & Omar, S. (2013). An Acceptance of 4G (Fourth Generation) Mobile Network in Malaysia. *International Journal of Information and Communication Technology Research*, 3(8) (2223-4985), pp.232-235.

- Bandyopadhyay, D. & Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1), 49-69.
- Berg, B.L. (2007). *Qualitative Research Methods for the Social Sciences* (4th Ed.). Boston, MA: Allyn & Bacon.
- Bettstetter, C., Vögel, H. J., & Eberspächer, J. (1999). GSM phase 2+ general packet radio service GPRS: Architecture, protocols, and air interface. *Communications Surveys, IEEE*, 2(3), 2-14.
- Bhalerao, S., Puntambekar, D. & Ingle, M. (2009). Generalizing agile software development life cycle. *International Journal on Computer Science and Engineering*, 1(3), 222–226.
- Bhargava, N., Bhargava, R., Mathuria, M. & Cotia, M. (2012). Fingerprint Matching using Ridge-End and Bifurcation Points. *IJCA Proceedings on International Conference on Recent Trends in Information Technology and Computer Science 2012 ICRTITCS* (6):12-1.
- Blasi, D., Cacace, V., Casone, L., Rizzello, M., Rotolo, S. & Bononi, L. (2007). Ad hoc wireless sensor networking: Challenges and issues. *ST Journal of Research*, [online] Volume 4 – Number 1 – Wireless Sensor Networks, pp.19-25. Available at:
<http://www.cs.unibo.it/bononi/Publications/110930152-ST-Journal-of-Research-4-1-Wireless-Sensor-Networks.pdf> [Accessed 10 Apr. 2014].
- Bonnet, P., Gehrke, J., Mayr, T., & Seshadri, P. (1999). Query processing in a device database system. Technical Report. Ithaca, NY, USA: Cornell University.
- Bonnet, P. Gehrke, J. & Seshadri, P. (2001). Towards Sensor Database Systems. In 2nd International Conference on Mobile Data Management (MDM01), pp 314-321.
- Brock, D. & Schuster, E. (2006). On the semantic web of things. In Semantic Days 2006, Stavanger, Norway, April 26-27.
- Buyya, R., Vecchiola, C. & Selvi, S. (2013). *Mastering cloud computing*. Waltham, MA: Morgan Kaufmann.

- Chaqfeh, M. & Mohamed, N. (2012). Challenges in middleware solutions for the internet of things. In Collaboration Technologies and Systems (CTS), 2012 International Conference on (pp. 21-26). IEEE.
- Chatzigiannakis, I., Mylonas, G. & Nikolettseas, S. (2007). 50 ways to build your application: A survey of middleware and systems for wireless sensor networks. In: Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on (pp. 466-473). IEEE.
- Chen, C. (2010). Design of a Child Localization System on RFID and Wireless Sensor Networks. *Journal of Sensors*, Vol. 2010, 01-07.
- Cisco: SDN, Collaboration, Internet of Things to Drive Future Trends. 2014. Cisco: SDN, Collaboration, Internet of Things to Drive Future Trends. [ONLINE] Available at: <http://www.eweek.com/networking/slideshows/cisco-sdn-collaboration-internet-of-things-to-drive-future-trends.html>. [Accessed 10 April 2014].
- Clampitt, H. (2014). The RFID HandBook 7th Edition: RFID for Dummies. [online] Rfidhandbook.blogspot.com. Available at: <http://rfidhandbook.blogspot.com/2004/11/rfid-for-dummies.html> [Accessed 22 May. 2014].
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387.
- Cohn, M., & Ford, D. (2003) "Introducing an Agile Process to an Organization", In IEEE Computer Society, pp.74-78.
- Cook, D. J., Augusto, J. C. & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4), 277-298.
- Cooking-hacks.com (2014). Waspnote – Shop. Retrieved 11 December 2014, from <http://www.cooking-hacks.com/shop/waspnote>.
- Cornell Database Group – Cougar. 2014. Cornell Database Group – Cougar. [ONLINE] Available at: <http://www.cs.cornell.edu/bigreddata/cougar/index.php>. [Accessed 22 November 2014].

- Coronato, A. (2012). Uranus: A middleware architecture for dependable AAL and vital signs monitoring applications. *Sensors* 2012, 12, 3145–3161.
- Crnkovic, G. D. (2010). Constructive research and info-computational knowledge generation. In *Model-Based Reasoning in Science and Technology* (pp. 359-380). Berlin Heidelberg: Springer.
- Culler-Mayeno, E. (2006). A Technical Report: Wireless Sensor Networks and How They Work. Prepared for Ann Holms, *University of California Santa Barbara*. Vol. 1, pp. 01 - 08.
- Datamanipulation.net (2014). SQLQueryStress – SQL Server query performance testing tool. [online] Available at: <http://www.datamanipulation.net/sqlquerystress/> [Accessed 11 Oct. 2014].
- Doty, N., Mulligan, D. K. & Wilde, E. (2010). Privacy issues of the W3C Geolocation API. arXiv preprint arXiv:1003.1775.
- Du, P. & Roussos, G. (2013). Adaptive communication techniques for the Internet of Things. *JSAN*, 2(1), pp.122-155.
- Dudziak, T. (2000). eXtreme Programming An Overview. 1st ed. [ebook] *Methoden und Werkzeuge der Software Produktion WS 1999/2000*, pp.5-13. Available at: http://csis.pace.edu/~marchese/CS616/Agile/XP/XP_Overview.pdf [Accessed 10 May 2014].
- Estrin, D. (2005). Reliability and storage in sensor networks. *Center for Embedded Network Sensing*.
- Ezhilarasan, M., Suresh Kumar, D., Santhanakrishnan, S., Dhanabalan, S. and Vinod, A. (2010). Person Identification Using Fingerprint by Hybridizing Core Point and Minutiae Features. *International Journal on Computer Science and Engineering*, Vol. 02, No. 09, 3075-3078.
- Faghih, M. & Moghaddam, M. (2011). SOMM: A New Service-Oriented Middleware for Generic Wireless Multimedia Sensor Networks Based on Code Mobility. *Sensors*, 11(12), pp.10343-10371.

- Fernandez, F.A., Fierrez, J., Ortega-Garcia, J., Gonzalez-Rodriguez, J., Fronthaler, H., Kollreider, K. & Bigun, J. (2007). A comparative study of fingerprint image-quality estimation methods, *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 4, pg.734-743.
- Finkenzeller, K. (2003). *RFID handbook*. Chichester, England: Wiley.
- Finkenzeller, K. (2010). *Rfid Handbook. Fundamentals And Applications In Contactless Smart Cards, Radio Frequency Identification And Near-Field Communication*, Third Edition. 3rd ed. Munich, Germany: John Wiley and Sons, Ltd.
- Fitzek, F., Pedersen, M., Perrucci, G., Rein, S & Gühmann, C. (2014). *Convergence of mobile devices and wireless sensor networks*. 1st ed. [ebook] Springer, pp.1-4. Available at: <http://kom.aau.dk/~ff/documents/WWRF17WG3FITZEK.pdf> [Accessed 14 Jun. 2013].
- Forte, D. (2009). Do encrypted disks spell the end of forensics? *Computer Fraud and Security*. 2009(2). pp. 18-20.
- FreightWatch. (2013). [online] Available at: http://www.freightwatchintl.com/sites/default/files/attachments/FreightWatch%202013%20Global%20Cargo%20Theft%20Threat%20Assesment%20Full_0.pdf [Accessed 20 Nov. 2014].
- Gehrke, J. & Madden, S. (2004). Query processing in sensor networks. *IEEE Pervasive Computing*, 3(1), 46-55.
- Gershenfeld, N., Krikorian, R. & Cohen, D. (2004). The Internet of Things. *Scientific American*, 291(4), 46–51.
- Gregor, S. (2006). The nature of theory in information systems. *Mis Quarterly*, 611-642.
- GSMA & Deloitte. (2012). *Sub-Saharan Africa Mobile Observatory*. [online] GSMA, pp.7-11. Available at: http://www.gsma.com/publicpolicy/wp-content/uploads/2013/01/gsma_ssamo_full_web_11_12-1.pdf [Accessed 9 Jun. 2013].

- Gubbi, J., Buyya, R., Marusi, S. & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 1645–1660, 16.
- Hadim, S. & Mohamed, N. (2006). Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE distributed systems online*, 7(3), 1.
- Han, S. W., Yoon, Y. B., Youn, H. Y. & Cho, W. D. (2004, May). A new middleware architecture for ubiquitous computing environment. In *Software Technologies for Future Embedded and Ubiquitous Systems*, 2004. Proceedings. Second IEEE Workshop on (pp. 117-121). IEEE.
- Heeks, R. (2002). i-development not e-development: Special issue on ICTs and development. *Journal of International Development*, 14(1), 1-11.
- Heinzelman, W. B., Murphy, A. L., Carvalho, H. S. & Perillo, M. A. (2004). Middleware to support sensor network applications. *Network, IEEE*, 18(1), 6-14.
- Hwang, J. & Yoe, H. (2011). Study on the context-aware middleware for ubiquitous greenhouses using wireless sensor networks. *Sensors*, 11(12), pp.4539-4561.
- International Strategy for Disaster Reduction (ISDR). (2006). Choice Reviews Online, 44(01), pp.44-0045a-44-0045a.
- ITU (2008). Ubiquitous Sensor Networks (USN). ITU-T technology watch briefing report series, No. 4. ITU, pp.1-7. Available at: http://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000040001PDFE.pdf
- ITU (2014). Statistics. Retrieved 11 April 2014, from <http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>.
- Jacobs, J. & Van Ranst, M. (2008). Biometric Fingerprinting for Visa Application: Device and Procedure Are Risk Factors for Infection Transmission. *Journal of Travel Medicine*, 15(5), 335-343. doi:10.1111/j.1708-8305.2008.00232.
- Jain, A. K. & Kumar, A. (2010). Biometrics of next generation: An overview. *Second Generation Biometrics*, 12(1), 2-3.

- Jones, D. & Gregor, S. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, 8(5), 1.
- Jorgensen, A. (2012). *Microsoft SQL Server 2012 bible*. Hoboken, N.J.: Wiley.
- Kasanen, E., Lukka, K. & Siitonen, A. (1991). “Konstruktiiivinen tutkimusote liiketaloustieteessä”, *Liiketaloudellinen aikakauskirja*, Vol. 40, pp.3021-7.
- Kasanen, E., Lukka, K. & Siitonen, A. (1993), “The constructive approach in management accounting research”, *Journal of Management Accounting Research*, Vol. 5, pp.243-264.
- Katasonov, A., Kaykova, O., Khriyenko, O., Nikitin, S. & Terziyan, V. (2008). *Smart Semantic Middleware for The Internet of Things*. 1st ed. [ebook] International Conference on Informatics in Control, Automation and Robotics (ICINCO), pp.1-8. Available at: <http://www.cs.jyu.fi/ai/papers/ICINCO-2008.pdf> [Accessed 22 Nov. 2014].
- Kephart, J. O. & Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–50.
- Khan, W., Xiang, Y., Aalsalem, M. & Arshad, Q. (2013). Mobile Phone Sensing Systems: A Survey. *IEEE Commun. Surv. Tutorials*, 15(1), pp.402-427.
- Kim, T. H., Jo, H. G., Lee, J. S. & Kang, S. J. (2012). A mobile asset tracking system architecture under mobile-stationary co-existing WSNs. *Sensors*, 12(12), 17446-17462.
- Kohanbash, D., Valada, A. & Kantor, G. (2011). *Base Station Design and Architecture for Wireless Sensor Networks*, pp.1-6.
- Kominers, P. (2012). *Interoperability Case Study: Internet of Things (IoT)*. Berkman Center Research Publication No. 2012-10. Available at SSRN: <http://ssrn.com/abstract=2046984>
- Konovalov, S. & Misslinger, S. (2006). *Extreme Programming*. 1st ed. [ebook] pp.3-7. Available at: [http://www14.in.tum.de/konferenzen/Jass06/courses/3/presentations/Extreme Programming.pf](http://www14.in.tum.de/konferenzen/Jass06/courses/3/presentations/ExtremeProgramming.pf) [Accessed 16 Jul. 2014].
- Kosmatos, E. (2011). Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture. *AIT*, 01(01), pp.5-12.

Kothari C.R. (2009) Research methodology methods and techniques, Second edition, New Delhi, New Age International (P) Ltd publishers.

Kothari, R.C. (1988), Research Methodology, New Delhi: Wiley Eastern Ltd.

Kraskowiak, S. (2009). Middleware Architecture with Patterns and Frameworks. 1st ed. [ebook] pp.14-17. Available at: <http://proton.inrialpes.fr/~kraskowia/MW-Book/main-onebib.pdf> [Accessed 11 Mar. 2014].

Kumar, A., Liu Y., Sengupta, J. & Divya, (2010). Evolution of Mobile Wireless Communication Networks: 1G to 4G. International Journal of electronics & communication technology. 1 (1), pp.68-72.

Kumar, A., Tewari, A., Shroff, G., Chittamuru, D., Kam, M. & Canny, J. (2010, April). An exploratory study of unsupervised mobile learning in rural India. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp.743-752). ACM.

Kumar, S., Nilsen, W., Pavel, M. & Srivastava, M. (2013). Mobile Health: Revolutionizing Healthcare Through Transdisciplinary Research. Computer, 46(1), pp.28-35.

Kyobe, M. (2011). Investigating the key factors influencing ICT adoption in South Africa. Journal of Systems and Information Technology, 13(3), pp.255-267.

Labro, E. & Tuomela, T. S. (2003). On bringing more action into management accounting research: process considerations based on two constructive case studies. European Accounting Review, 12(3), 409-442.

Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T. & Campbell, A. (2010). A survey of mobile phone sensing. IEEE Communications Magazine, 48(9), pp.140-150.

Lee, J. S., Su, Y. W. & Shen, C. C. (2007, November). A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE* (pp. 46-51). IEEE.

Li S., Son S. & Stankovic J., "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," Proc. 2nd Int'l Workshop Information Processing in Sensor Networks (IPSN 03), LNCS 2634, Springer, 2003, pp.502-517.

Li Y., Huynh, D. T., Das, S. K. & Du, D. Z. (2008). Wireless Algorithms, Systems, and Applications. In: Third International Conference, WASA 2008. Dallas: Springer.

Libelium.com (2014). Waspote Wireless Interfaces Overview – Complete List of Modules | Libelium. Retrieved 11 December 2014, from <http://www.libelium.com/products/waspote/interfaces/>

Liu, C., Sinkovics, R., Pezderka, N. & Haghirian, P. (2012). Determinants of Consumer Perceptions toward Mobile Advertising – A Comparison between Japan and Austria. *Journal of Interactive Marketing*, 26(1), pp.21-32.

Lourde, R. & Khosla, D. (2010). Fingerprint Identification in Biometric Security Systems. *IJCEE*, 852-855. doi:10.7763/ijcee.2010.v2.239.

Lukka, K. (2000). The key issues of applying the constructive approach to field research. Reponen, T. (ed.), pp. 113-28.

Mamun, Q. (2012). A Qualitative Comparison of Different Logical Topologies for Wireless Sensor Networks. *Sensors*, 12(12), pp.14887-14913.

Maraiya, K., Kant, K. & Gupta, N. (2011). Application based Study on Wireless Sensor Network. *International Journal of Computer Applications*, 21(8), pp.9-15.

Martonosi, M. (2004). The Princeton ZebraNet Project: Sensor Networks for Wildlife Tracking. Princeton University, pp.2-7.

Masinde, E. (2012). Bridge between African Indigenous Knowledge and Modern Science on Drought Prediction. Ph.D. University of Cape Town.

Masinde, M., Bagula, A. & Muthama, N. J. (2012). The role of ICTs in downscaling and up-scaling integrated weather forecasts for farmers in sub-Saharan Africa. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*. pp. 122-129. ACM.

Mattern, F. & Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. In *From active data management to event-based systems and more* (pp. 242-259). Berlin Heidelberg: Springer.

- Mayer, C. P. (2009). Security and privacy challenges in the internet of things. *Electronic Communications of the EASST*, Vol. 17 (ISSN 1863-2122), pp.2-10.
- McGrath, M. & Scanail, C. (2014). *Sensor technologies*. [New York]: ApressOpen.
- Miorandi, D., Sicari, S., De Pellegrini, F. & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7), pp.1497-1516.
- Modelright.com (2014). ModelRight Download full version trials. Retrieved 11 December 2014, from <http://www.modelright.com/downloads.asp>.
- Msdn.microsoft.com (2014a). Introduction to the Sensor and Location Platform in Windows (Windows). Retrieved 11 December 2014, from [http://msdn.microsoft.com/en-us/library/windows/desktop/dd318936\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd318936(v=vs.85).aspx)
- Msdn.microsoft.com (2014b). SQL Server Service Broker. Retrieved 13 December 2014, from <http://msdn.microsoft.com/en-us/library/bb522893.aspx>
- Myers, G.J. 1978. *Composite/Structured Design*. Van Nostrand Reinhold, New York.
- Nikolić, S., Penca, V., Segedinac, M. & Konjović, Z. (2011, May). Semantic web-based architecture for managing hardware heterogeneity in wireless sensor network. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics* (pp. 42). ACM.
- Noguero, A., Calvo, I., Perez, F. & Almeida, L. (2013). "FTT-MA: A Flexible Time-Triggered Middleware Architecture for Time Sensitive, Resource-Aware AmI Systems", *Sensors*, Vol. 13, pp.6229-6253, 2013.
- Okoli, A. and Okpaleke, F. (2014). Cattle rustling and dialectics of security in Northern Nigeria. *International Journal of Liberal Arts and Social Science*, [online] Vol. 2 No.3 (2307-924X), pp.109-115. Available at: <http://www.ijlass.org> [Accessed 22 Nov. 2014].
- Ozekisms.com (2014). SMS Gateway for Windows with SMPP,UCP,CIMD2 and GSM modem support. Retrieved 13 December 2014, from <http://www.ozekisms.com/>
- Panda, I., Giri, S. R., Kumar, P. & Mohaptra, A. (2012). A New Approach to Fingerprint Recognition. *International Journal on Computer Science and Engineering (IJCSE)* (0975-3397) Vol, 4.

- Paridel, K., Bainomugisha, E., Vanrompay, Y., Berbers, Y. & De Meuter, W. (2010). Middleware for the internet of things, design goals and challenges. *Electronic Communications of the EASST*, 28.
- Patton, B. (1983). Prototyping - a nomenclature problem. *ACM SIGSOFT Software Engineering Notes*, 8 (2), pp. 14-16.
- Plano Clark, V. L. (2005). Cross-disciplinary analysis of the use of mixed methods in physics education research, counseling psychology, and primary care.
- Pocovnicu, A. (2009). Biometric Security for Cell Phones. *Informatica Economica*, 13(1/2009), 57-63.
- Princeton.edu (2014). Thread (computer science). Retrieved 13 December 2014, from [https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Thread_\(computer_science\).html](https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Thread_(computer_science).html)
- Priyadarshini, A. (2013). Internet of Things: Applications and Challenges In Technology And Standardization, National Institute Of Science & Technology Palur Hills, Berhampur, Orissa – 761008, India.
- Ratcliff, B., (1988). Early and not-so-early prototyping – rationale and tool support, Computer Software and Applications Conference, 1988. COMPSAC 88. Proceedings. Twelfth International, 5-7 October 1988, IEEE Xplore, pp.127-134.
- Rob, P., Coronely, C. & Crockett, K. (2008). Data bases systems: design, implementation and management. CengageLearning EMEA.
- Roberts, C. (2006). Radio frequency identification (RFID). *Computers & Security*, 25(1), 18-26. doi:10.1016/j.cose.2005.12.003.
- Rodríguez-Molina, J., Martínez, J. F., Castillejo, P. & de Diego, R. (2013). SMArc: a proposal for a smart, semantic middleware architecture focused on Smart City energy management. *International Journal of Distributed Sensor Networks*, 2013, pp.1-17.
- Rodríguez-Molina, J., Martínez, J. F., Castillejo, P. & López, L. (2013). Combining wireless sensor networks and semantic middleware for an Internet of Things-based sportsman/woman monitoring application. *Sensors*, 13(2), 1787-1835.

- Römer, K., Kasten, O. & Mattern, F. (2002). Middleware challenges for wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(4), 59-61.
- Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal Of Information Science*, 33(2), 163-180.
- Rubini, A. & Corbet, J. (2001). *Linux device drivers*. Sebastopol: O'Reilly & Associates.
- Sanqunetti, D. R. (2004). U.S. Patent No. 6,721,652. Washington, DC: U.S. Patent and Trademark Office.
- Schach, S. (2008). *Object-oriented software engineering*. Boston, Mass.: McGraw-Hill; pp.54-181.
- Sheng, Z., Yang, S., Yu, Y., Vasilakos, A. V., McCann, J. A. & Leung, K. K. (2013). A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *Wireless Communications, IEEE*, 20(6), 91-98.
- Shinghal, K., Noor, A., Srivastava, N. & Singh, R. (2010). Wireless Sensor Networks In Agriculture: For Potato Farming. *International Journal of Engineering Science and Technology*, Vol. 2(8) ;(3955-3963), pp.1-3.
- Silberschatz, A., Galvin, P. & Gagne, G. (2009). *Operating system concepts*. Hoboken, NJ: J. Wiley & Sons.
- Bandyopadhyay, S., Sengupta, M., Maiti, S. & Dutta, S. (2011). Role of middleware for internet of things: A study. *International Journal of Computer Science & Engineering Survey (IJCSSES)*, 2(3), 94-105.
- Srisathapornphat, C., Jaikaeo, C. & Shen, C. C. (2000). Sensor information networking architecture. In *Parallel Processing, 2000. Proceedings. 2000 International Workshops on*, pp.23-30. IEEE.
- Sub-Saharan Africa Mobile Economy. (2013). *Mobile Economy Sub-Saharan Africa*. [Online] Available: http://gsma.com/newsroom/wp-content/uploads/2013/12/GSMA_ME_Sub_Saharan_Africa_ExecSummary_2013.pdf. [Accessed 22 July 2014].

- Swedberg, C. (2014). DHL Thermonet Tracks Drugs and Life-Sciences Goods With RFID Temperature Tag - RFID Journal. [online] Rfidjournal.com. Available at: <http://www.rfidjournal.com/articles/view?10777> [Accessed 22 August. 2014].
- Teddlie, C. & Yu, F. (2007). Mixed methods sampling: A Typology With Examples. *Journal of Mixed Methods Research*, 1(1), pp.77-100.
- Traxler, J. (2011). The Mobile and Mobility: Information, Organisations and Systems. In *Information Systems Development* (pp. 25-34). New York: Springer
- Vermesan, O. & Friess P. (2014). Internet of things applications – from research and innovation to market deployment. The River Publishers Series in Communications, Alborg, Denmark, 2014. ISBN: 9788793102941. pp.287-313.
- Vermesan, O. & Friess, P. (2013). Internet of things. Aalborg: River Publishers.
- Weber, R. (2010). Internet of Things – New security and privacy challenges. *Computer Law & Security Review*, 26(1), pp.23-30.
- Wikipedia (2015). *Kill switch*. Retrieved 27 May 2015, from http://en.wikipedia.org/wiki/Kill_switch.
- Yao, Y. & Gehrke, J. (2002). The Cougar approach to in-network query processing in sensor networks. *ACM Sigmod Record*, 31(3), pp.9-18.
- Yeh, L. W., Wang, Y. C. & Tseng, Y. C. (2009). iPower: an energy conservation system for intelligent buildings by wireless sensor networks. *International Journal of Sensor Networks*, 5(1), pp.1-10.
- Yoneki E. & Bacon J. (2005). "A survey of wireless sensor network technologies: Research trends and middleware's role". Technical Report UCAM-CL-TR646, University of Cambridge.
- Zaeri, N. (2011). Minutiae-based Fingerprint Extraction and Recognition. INTECH Open Access Publisher, pp.55-69.
- Zaslavsky, A., Perera, C. & Georgakopoulos, D. (2013). Sensing as a service and big data. *arXiv preprint arXiv:1301.0159*.

Zhang, D., Liu, F., Zhao, Q., Lu, G. & Luo, N. (2011). Selecting a reference high resolution for fingerprint recognition using minutiae and pores. *Instrumentation and Measurement, IEEE Transactions on*, 60(3), 863-871.

Zhang, T., Ouyang, Y. & He, Y. (2008). Traceable Air baggage handling system based on RFID tags in the airport. *Applied Electronic Commerce Research (JTAER)*, 3(1), 106-115.

Appendices

C. Asset Loss prevention (security System)

Question 1.
Does your company have an asset anti-theft security system?
No
Question 2.
What security system(s) is in place to guard asset from theft?
Security Guards
Question 3.
How (who, when) are asset losses reported?
Stolen Assets are report to Assert Management and Security department, this happens as soon as the incident is noticed, a report is created by person responsible for the asset.
Question 4.
What asset loss investigation channels/options available?
There is an in-house investigation office and if the problem is too big then police is involved to address the situation at hand.
Question 5.
Do you have a CCTV system monitoring your facility?
Yes in certain buildings; yes they are helpful through provision of tangible evidence
Question 6.
Are CCTV images stored for at least 90 days?
No...but kept for 3 weeks
Question 7.
Are all entry and exit points alarmed?
No....guarded by security personnel

Appendix 1 : Preliminary investigation questions (A)

D. Asset Loss Statistics

Question 1.
Do you have records of how many laptops have been reported lost thus far?
Yes.
Question 2.
Do you have records of how many printers have been reported lost thus far?
Yes.
Question 3.
Do you have records of how many computers have been reported lost thus far?
Yes.
Question 4.
What is the overall total monthly asset loss in monetary terms?
????
Question 5.
What is the overall total annual asset loss in monetary terms?
???
Question 6.
How often are information assets (laptops, printers and computers) stolen?
???

Appendix 2: Preliminary investigation questions (B)

```

20 public static ActionResult CreateDB_Record(string table_Name, object[] properties)
21 {
22     result = new ActionResult();
23     try
24     {
25         result = DBSUtilities.LoadDBS(table_Name);
26
27         if(result.ReturnCode.Equals(0))
28         {
29             DBSUtilities.dataRow = DBSUtilities.dataSet.Tables[table_Name].Rows.Find(properties[0]); // Find (ID)
30
31             if (DBSUtilities.dataRow == null)
32             {
33                 DBSUtilities.dataRow = DBSUtilities.dataSet.Tables[table_Name].NewRow(); //Creating NewRow
34
35                 for(int x = 0; x < DBSUtilities.dataRow.Table.Columns.Count; ++x)
36                 {
37                     DBSUtilities.dataRow[x] = properties[x];
38                 }
39
40                 DBSUtilities.dataSet.Tables[table_Name].Rows.Add(DBSUtilities.dataRow);
41
42                 DBSUtilities.dataAdapter.Update(DBSUtilities.dataSet, table_Name);
43
44                 result.ReturnCode = 0; //setting action result success code
45                 //setting action success message
46                 result.ReturnMessage = "Record Successfully Saved";
47             }
48             else
49             {
50                 UpdateDB_Record(table_Name, properties);
51             }
52         }
53         else
54         {
55             result.ReturnCode = - 1;
56             result.ReturnMessage = "Error Source: CreateDB_Record()/n/nDatabase Loading Failed";
57         }
58     }
59     catch (Exception e)
60     {
61         result.ReturnCode = -1;
62         result.ReturnMessage = "Error Source: CreateDB_Record()/n/n" + e.Message;
63     }
64
65     if (DBSUtilities.connection != null)
66     {
67         DBSUtilities.connection.Close(); //closing database connection
68     }
69     return result;
70 }

```

Appendix 3: Create record code

```

112 public static ActionResult UpdateDB_Record(string table_Name, object[] properties)
113 {
114     result = new ActionResult();
115
116     try
117     {
118         result = DBSUtilities.LoadDBS(table_Name);
119
120         if (result.ReturnCode.Equals(0))
121         {
122             DBSUtilities.dataRow = DBSUtilities.dataSet.Tables[table_Name].Rows.Find(properties[0]); // Find (ID)
123
124             if (DBSUtilities.dataRow != null)
125             {
126                 for (int x = 1; x < DBSUtilities.dataRow.Table.Columns.Count; ++x)
127                 {
128                     DBSUtilities.dataRow[x] = properties[x];
129                 }
130
131                 DBSUtilities.dataAdapter.Update(DBSUtilities.dataSet, table_Name);
132
133                 result.ReturnCode = 0; //setting action result success code
134                 //setting action success message
135                 result.ReturnMessage = "Record Successfully Updated";
136             }
137         }
138         else
139         {
140             result.ReturnCode = -1;
141             result.ReturnMessage = "Error Source: UpdateDB_Record()/n/nDatabase Loading Failed";
142         }
143     }
144     catch (Exception e)
145     {
146         result.ReturnCode = -1;
147         result.ReturnMessage = "Error Source: UpdateDB_Record()/n/n" + e.Message;
148     }
149
150     if (DBSUtilities.connection != null)
151     {
152         DBSUtilities.connection.Close(); //closing database connection
153     }
154     return result;
155 }

```

5 references

Appendix 4: Update record code

```

21 public GeoLocation(string computerName)
22 {
23     ComputerName = computerName;
24
25     GetGeoLocationEvent geoLocEvent = new GetGeoLocationEvent(GetLocationEvent);
26
27     geoLocEvent.Invoke();
28 }
29
30 1 reference
31 private void GetLocationEvent()
32 {
33     try
34     {
35         watcher = new GeoCoordinateWatcher(GeoPositionAccuracy.High);
36
37         watcher.PositionChanged += new EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>(watcher_PositionChanged);
38
39         bool started = watcher.TryStart(false, TimeSpan.FromMilliseconds(5000));
40
41         if (!started)
42         {
43             MessageBox.Show("GeoCoordinateWatcher timed out on start.", "GeoLocation Error");
44         }
45     }
46     catch (Exception err)
47     {
48         new FileHandler("Log File.txt").ActivityLogging("Error Source: GeoLocation GetLocationEvent() - " + err.InnerException.Message);
49     }
50 }
51 1 reference
52 public static void GeoLocationStop()...
53
54 1 reference
55 private void watcher_PositionChanged(object sender, GeoPositionChangedEventArgs<GeoCoordinate> e)
56 {
57     SaveGeoLocationEvent geoEvent = new SaveGeoLocationEvent(SavePosition);
58
59     geoEvent.Invoke(e.Position.Location.Latitude, e.Position.Location.Longitude);
60 }
61
62 1 reference
63 private void SavePosition(double Latitude, double Longitude)
64 {
65     try
66     {
67         GeoPostion.UpdateGeoPosition(ComputerName, Latitude, Longitude);
68     }
69     catch (Exception err)
70     {
71         new FileHandler("Log File.txt").ActivityLogging("Error Source: GeoLocation SavePosition() - " + err.InnerException.Message);
72     }
73 }
74
75 }

```

Appendix 5: Geographical position code

```

void loop()
{
    // put your main code here, to run repeatedly:

    //Sending an SMS
    while(true)
    {
        if (USB.available() > 0) //check if there is data in buffer
        {
            val = USB.read(); // read data in buffer

            if (val == 0x31) //send message if "1" is received
            {
                val = 0;
                x = 0;
                y = 0;

                do
                {
                    if (USB.available() > 0)
                    {
                        input = USB.read(); // read data in
                        ++val;
                        if (3 <= val && val <= 12)
                        {
                            contact_no[x] = input;
                            ++x;
                        }
                        else if (15 <= val && val <= 159)
                        {
                            message[y] = input;
                            ++y;
                        }
                    }
                }
                while (USB.available() != 0);

                contact_no[10] = '\0';
                message[y] = '\0';

                answer = _3G.sendSMS(message, contact_no);

                if (answer == 1)
                {
                    USB.println(F("0")); //Sms Sent
                }
                else if (answer == 0)
                {
                    USB.println(F("-2")); //Sms Sending Error
                }

                memset(message, 0, 160);
                memset(contact_no, 0, 11);
            }
        }
    }
}

else
{
    time = millis();
    answer = 0;
    // waits for receive a SMS
    while ((answer != 2) && (millis() - time) < 60000)
    {
        answer = _3G.manageIncomingData();
        // Condition to avoid an overflow (DO NOT REMOVE
        if (millis() < time)
        {
            time = millis();
        }
    }

    // then shows the text of the message
    if (answer == 2)
    {
        USB.print(_3G.tiffNumber);
        USB.print(F(", "));
        USB.println(_3G.buffer_3G);
    }
    else
    {
        count = _3G.getTotalSMS();

        if (count > 20)
        {
            for (int8_t index = 0; x < count; ++index)
            {
                _3G.deleteSMS(index);
            }
        }
    }
}
}

```

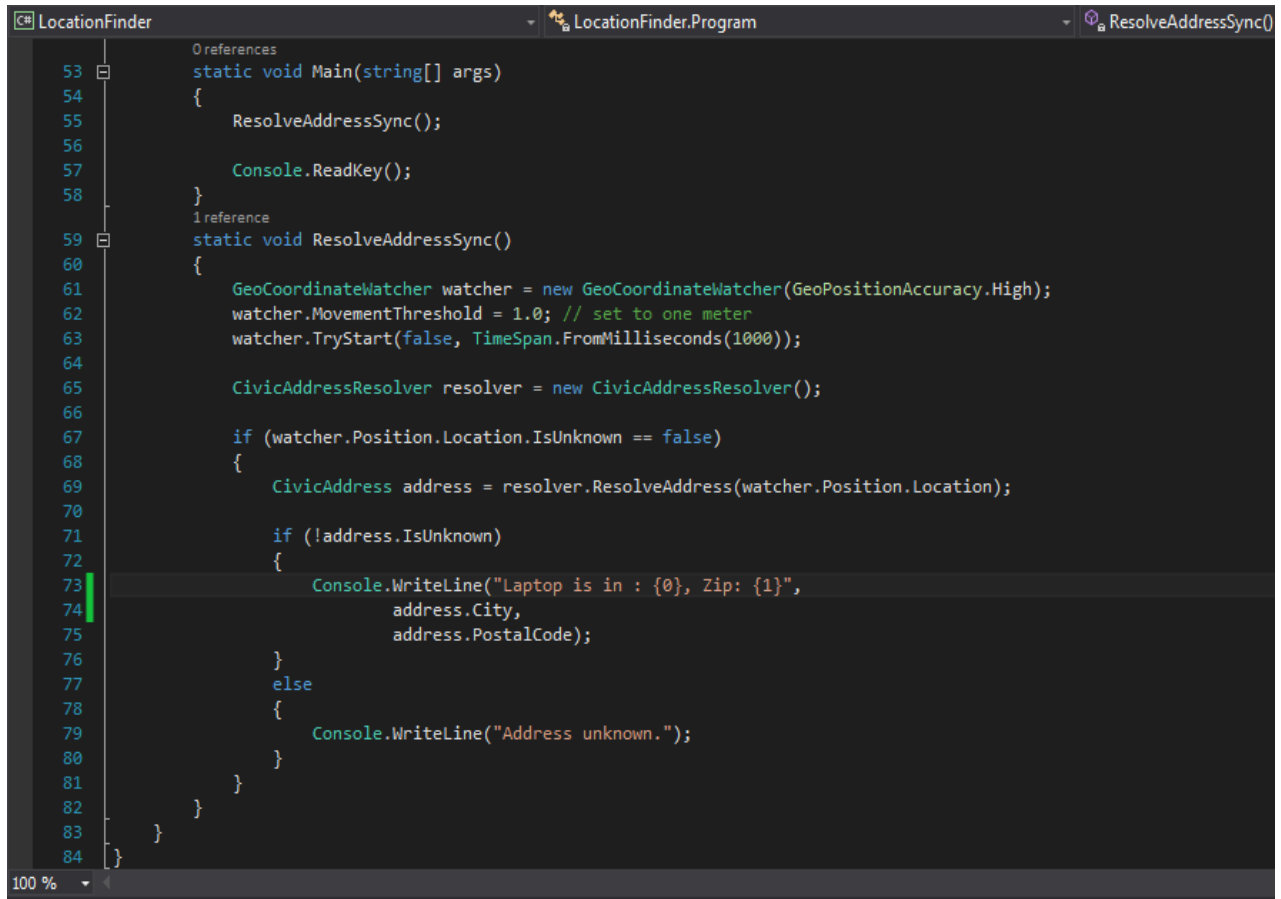
Appendix 6: Wasmote gateway code

```

135 // You must stop the dependency before starting a new one.
136 // You must start the dependency when creating a new one.
137 SqlDependency.Stop(DBSUtilities.ConnectionString);
138
139 SqlDependency.Start(DBSUtilities.ConnectionString);
140
141 using (SqlConnection cn = new SqlConnection(DBSUtilities.ConnectionString))
142 {
143     using (SqlCommand cmd = cn.CreateCommand())
144     {
145         cmd.CommandType = System.Data.CommandType.Text;
146
147         cmd.CommandText = "SELECT id, sender, msg FROM dbo.[ozekimessagein]";
148
149         cmd.Notification = null;
150
151         // creates a new dependency for the SqlCommand
152         dataDependency = new SqlDependency(cmd);
153         // creates an event handler for the notification of data
154         // changes in the database.
155         // NOTE: the following code uses the normal .Net capitalization methods, though
156         // the forum software seems to change it to lowercase letters
157         dataDependency.OnChange += new OnChangeEventHandler(Dependency_OnChange);
158
159         cn.Open();
160
161         using (SqlDataReader dr = cmd.ExecuteReader())
162         {
163             if (dr != null && dr.HasRows)
164             {
165                 while (dr.Read())
166                 {
167                     recordID = dr.GetValue(0).ToString();
168                     newSmsData[0] = dr.GetValue(1).ToString();
169                     newSmsData[1] = dr.GetValue(2).ToString();
170                 }
171             }
172             else
173                 newSmsData = null;
174         }
175
176         if (newSmsData != null)
177         {
178             new GetComputerNameEvent(GetComputerName).Invoke("0" + newSmsData[0].Substring(3).Trim());
179
180             if (Utilities.IsStringEmpty(ComputerName).Equals(false))
181             {
182                 if (ComputerName.ToUpper().Equals(System.Environment.MachineName.ToUpper()))
183                 {
184                     SmsInboundEvent newSms = new SmsInboundEvent(SmsInbound);
185                     newSms.Invoke(newSmsData);
186                     //
187                     DBSUtilities.GeneralQuery("DELETE FROM ozekimessagein WHERE id = '" + recordID + "'");

```

Appendix 7: SQL broker code



```
53  static void Main(string[] args)
54  {
55      ResolveAddressSync();
56
57      Console.ReadKey();
58  }
59  static void ResolveAddressSync()
60  {
61      GeoCoordinateWatcher watcher = new GeoCoordinateWatcher(GeoPositionAccuracy.High);
62      watcher.MovementThreshold = 1.0; // set to one meter
63      watcher.TryStart(false, TimeSpan.FromMilliseconds(1000));
64
65      CivicAddressResolver resolver = new CivicAddressResolver();
66
67      if (watcher.Position.Location.IsUnknown == false)
68      {
69          CivicAddress address = resolver.ResolveAddress(watcher.Position.Location);
70
71          if (!address.IsUnknown)
72          {
73              Console.WriteLine("Laptop is in : {0}, Zip: {1}",
74                              address.City,
75                              address.PostalCode);
76          }
77          else
78          {
79              Console.WriteLine("Address unknown.");
80          }
81      }
82  }
83 }
84 }
```

Appendix 8: Location finder spike code

Q116										
	M	N	O	P	Q	R	S	T	U	V
	Depreciation Cost Center	Depr Meth	Useful Lifetime	Acquire Date	Purchase Price SUM	Inventory Value SUM	Asset Description	Residual Value SUM	Calculation1	Replacement Value SUM
1										
2										
3	NULL	NULL	NULL	18/Nov/1985	1825.60	1.00	GENERATOR	0	ZAR 1,825.60	2008.16
4	NULL	NULL	NULL	26/Nov/1985	550.76	1.00	FAN EXTRATION	0	ZAR 550.76	605.84
5	8830	S	60	04/Dec/1985	35000.00	1.00	LAB VOLT EQUIPMENT	1	ZAR 35,000.00	38500.00
6	8240	NULL	NULL	04/Dec/1985	144.14	1.00	NULL	0	ZAR 144.14	158.55
7	8240	NULL	NULL	04/Dec/1985	101.92	1.00	NULL	0	ZAR 101.92	112.11
8	8240	NULL	NULL	04/Dec/1985	112.50	1.00	NULL	0	ZAR 112.50	123.75
9	NULL	NULL	NULL	05/Dec/1985	476.00	1.00	BOYOCOUS BATH	0	ZAR 476.00	523.60
10	NULL	NULL	NULL	05/Dec/1985	104.16	1.00	DINING TABLE	0	ZAR 104.16	114.58
11	NULL	NULL	NULL	05/Dec/1985	218.40	1.00	DROP HAMMER	0	ZAR 218.40	240.24
12	8240	NULL	NULL	05/Dec/1985	84.00	1.00	NULL	0	ZAR 84.00	92.40
13	NULL	NULL	NULL	31/Dec/1985	523.20	1.00	AIR CONDITIONER HITACHI	0	ZAR 523.20	523.20
14	NULL	NULL	NULL	31/Dec/1985	523.20	1.00	AIR CONDITIONER HITACHI	0	ZAR 523.20	523.20
15	NULL	NULL	NULL	31/Dec/1985	266.82	1.00	BOOKCASE	0	ZAR 266.82	266.82
16	NULL	NULL	NULL	31/Dec/1985	266.82	1.00	BOOKCASE	0	ZAR 266.82	266.82
17	NULL	NULL	NULL	31/Dec/1985	599.20	1.00	BOYLE LAW APP.	0	ZAR 599.20	659.12
18	NULL	NULL	NULL	31/Dec/1985	380.80	1.00	ELECTRO MAGNET	0	ZAR 380.80	418.88
19	NULL	NULL	NULL	31/Dec/1985	380.80	1.00	ELECTROMAGNETIC KIT	0	ZAR 380.80	418.88
20	NULL	NULL	NULL	31/Dec/1985	380.80	1.00	ELETROMAGNETIC KIT	0	ZAR 380.80	418.88
97	8240	S	60	25-Jan-1997	13224.00	6898.41	MUTSUBISHI DATA VIDEO PROJECTOR : AV CC	1	ZAR 13,224.00	13224.00
98	NULL	NULL	NULL	24-Dec-2005	930.00	1.00	L-SHAPED WOODEN DESK	0	ZAR 930.00	930.00
99	4070	S	36	11-May-2006	6218.89	1.00	INTEL PENTIUM COMPUTER : MECER	1	ZAR 6,218.89	6218.89
100	4000	S	36	22-Jul-2006	1290.24	69.22	UPS MECER 2000 VA BLACK	1	ZAR 1,290.24	1290.24
101	8240	S	168	24-Jan-2007	17100.00	9453.65	PROJECTOR VIDEO/DATA ASK C180	1	ZAR 17,100.00	17100.00
102	8241	S	36	01-Sep-2007	12842.10	10887.11	LAPTOP LATITUDE E4310	0	ZAR 12,842.10	12842.10
103	8241	S	1	29-Jan-2008	4865.90	1.00	MECER INTEL	1	ZAR 4,865.90	4865.90
104	NULL	S	36	01-Mar-2008	1425.00	1425.00	17" PANEL BLACK MONITOR : MUSTEK	1	ZAR 1,425.00	1425.00
105	8240	S	120	08-Oct-2008	1799.00	1618.32	GPS XLTT5 V3	0	ZAR 1,799.00	1799.00
106	8240	S	180	26-Feb-2009	14330.25	10366.20	PROJECTOR VIDEO/DATA MITSUBISHI	1	ZAR 14,330.25	14330.25
107	0616	S	36	12-Mar-2009	12711.00	2656.63	D630 NOTEBOOK : DELL	1	ZAR 12,711.00	12711.00
108	8240	S	36	08-May-2009	18896.88	1.00	DELL D620	1	ZAR 18,896.88	18896.88
109	8241	S	36	01-Mar-2010	18883.36	7781.16	LATITUDE INTEL CORE E6500 NOTEBOOK : DE	1	ZAR 18,883.36	18883.36
110	8240	S	173	27-Oct-2010	9999.00	6244.58	PROJECTOR VIDEO/DATA PANASONIC	1	ZAR 9,999.00	9999.00
111									ZAR 134,515.62	

Appendix 9: Write off records

Research Project Towards:
**An Integrated Internet of Things Based System for Tracking and
Monitoring Assets**

Data Collection Tool – System Implementation Survey [October – 2014]

1. Introduction

Research has revealed that insecurity of assets and lives is a problem around the world. Further, implementation of security systems in many organisations tend to take a disjointed approach; that is, they do not instantaneously and intelligently send real time security breach messages to security personnel. Such systems are mainly targeted at controlling access to buildings and have little to do with asset monitoring. This leads to constant loss of assets (especially moveable ones) such as phones, laptops, data projects, tablets and so on.

With advancements in computing technology, paradigms such as big data, cloud computing, Internet of things and Internet of everything have emerged. Internet of Things (IoT) refers to interconnection of things/objects that are uniquely addressable. Traditionally 'things' could be digital items such as radio frequency identification (RFID), smart phones, smart vehicles and computers. The latter can be used to develop an intelligent asset tracking system and minimize assets' loss. This precisely was the aim of this project; an asset tracking system was developed and tested within the CUT environment. Mr. Admire Mhlaba as part of his M.Tech, IT Degree, carried out the project.

The aim of this questionnaire is to prepare the grounds for the implementation of this system. Before implementing this system, Mr. Mhlaba, is seeking the views on asset security/insecurity from CUT staff and students. The valuable input is purely for academic purposes and will be used to assess and improve the system prototype.

2. Consent Form

I have read the Information Sheet and have had the details of the study explained to me. My questions have been answered to my satisfaction, and I may ask further questions at any time.

I understand I have the right to withdraw from the study at any time and decline to answer any particular questions.

I agree to provide information to the researcher on the understanding that my name will not be used without my permission.

I agree/do not agree to the interview being recorded electronically.

I understand that I have the right to ask for the tape to be turned off at any time during the interview

I agree to participate in this study under the conditions set out in the information sheet

Surname:

Name

Date:

Admire Mhlaba – yaddy@gmail.com - Department of Information Technology, CUT

1

3. Interviewee Background Information

(a) Name (optional):

(b) Gender

Male ☐ Female ☐

(c) Category

Staff ☐ Student ☐

If student, please provide:

(i) Year of Study:

1st year ☐ 2nd year ☐ 3rd year ☐ 4th year ☐

(ii) Course Enrolled for: (e.g. NDip, IT)

4. Knowledge of CUT's Asset Security System

Are you aware of the following security systems installed at CUT?

	Security System	Yes	No
1	Fingerprinting system for rooms such as lecture theatres:	<input type="checkbox"/>	<input type="checkbox"/>
2	Surveillance security cameras:	<input type="checkbox"/>	<input type="checkbox"/>
3	Library book anti-theft security system:	<input type="checkbox"/>	<input type="checkbox"/>
4	Electro Magnetic door locks for staff offices:	<input type="checkbox"/>	<input type="checkbox"/>
5	RFID Tag controlled door locks:	<input type="checkbox"/>	<input type="checkbox"/>

5. Asset Ownership

Do you own any of the following assets? Tick (X) where appropriate:

	Assets	Yes	No
	Laptop?	<input type="checkbox"/>	<input type="checkbox"/>
	Cell phone?	<input type="checkbox"/>	<input type="checkbox"/>
	Tablet?	<input type="checkbox"/>	<input type="checkbox"/>
	Other (Specify)? <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

6. Asset Loss/Theft

6.1	Can you or your parents afford replacing a lost laptop?	Yes <input type="checkbox"/>	No <input type="checkbox"/>
-----	---	------------------------------	-----------------------------

6.2 Using a 5-point scale where 1 means Not Important, 2 means Fairly Important, 3 means Neutral, 4 means Important and 5 means Very Important. How important is your laptop/Tablet and data on it?

Laptop	Data
1 <input type="checkbox"/>	1 <input type="checkbox"/>
2 <input type="checkbox"/>	2 <input type="checkbox"/>
3 <input type="checkbox"/>	3 <input type="checkbox"/>
4 <input type="checkbox"/>	4 <input type="checkbox"/>
5 <input type="checkbox"/>	5 <input type="checkbox"/>

6.3	Have you lost an asset in the last 5 years?	Yes	No
-----	---	-----	----

If yes, please answer questions (a) to (h) below:

a) What type of asset did you lose?

Laptop	
Cellphone	
Tablet	
Other	

b) From what location was the asset stolen?

Office	Hostel
Library	Parking
Lab.	Other :

c) If office, please specify building name:

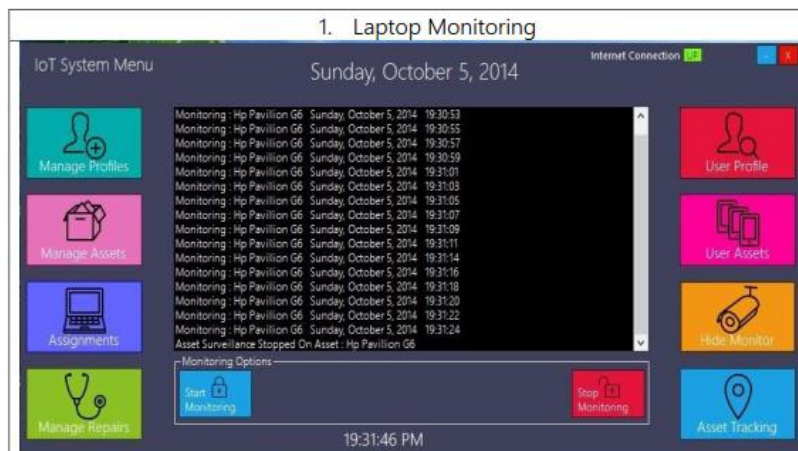
d) If lab, please specify laboratory name:

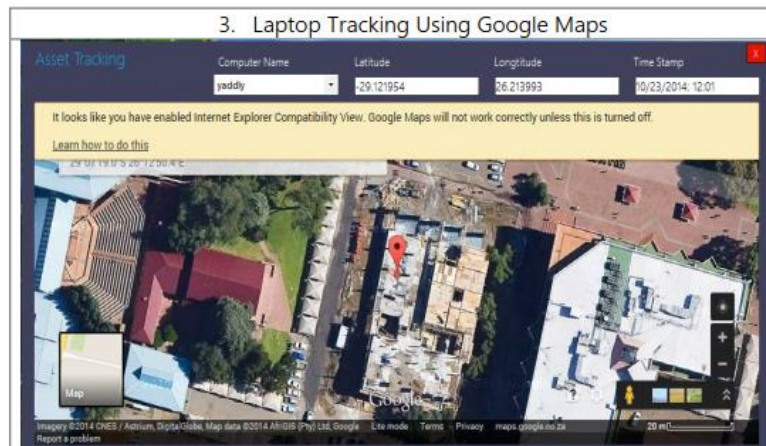
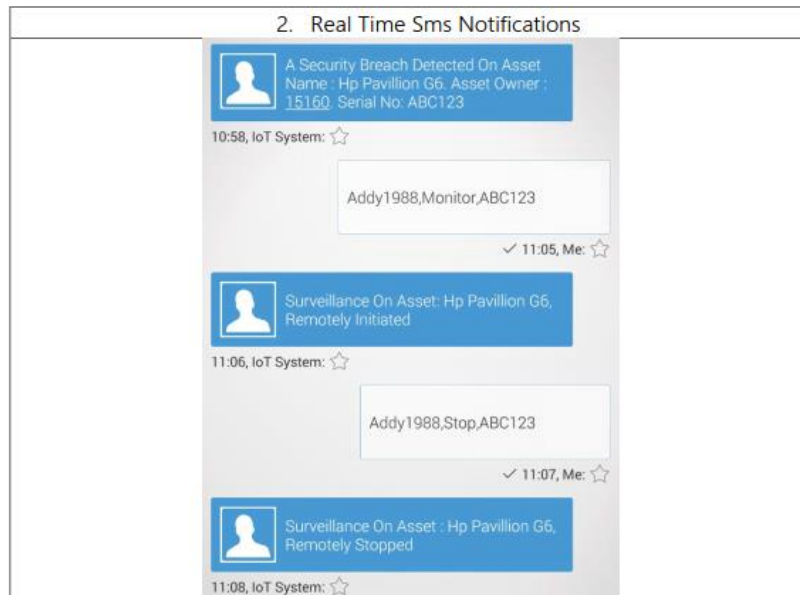
e)	Was the asset ever recovered?	Yes	No
f)	If yes, specify how it was recovered:		
g)	What extra measures did you put to avoid the loss in future?		
Never leave any of my assets in the library, lab or elsewhere in CUT			
h)	Kindly estimate the value of the lost asset along with data		

7. Integrated Asset Management System

The implemented system has the following main functions:

- 1) On your request (via SMS or laptop) tracks and monitors your asset.
- 2) Informs you when there is a breach.
- 3) Uses Google maps to indicate the location of your lost/stolen asset.





- a) Using a 5-point scale where 1 means Not at All and 5 means Most likely, give a rating on the likelihood that you would use each of the following aspects of the system. Assume very critical and valuable assets such your laptop and tablet are at risk?

Monitoring		Sms Alerts		Tracking	
1		1		1	
2		2		2	
3		3		3	
4		4		4	
5		5		5	

- b) How safe is your laptop within designated study areas or offices around the campus?

Safe	
Unsafe	

- c) Using a 5-point scale where 1 means Always and 5 means Never. How often do you leave your laptop unattended at campus?

1	
2	
3	
4	
5	

- d) Using a 3-point scale where 3-No, 2-Maybe and 1-Yes. Do you mind having a monitoring tag attached to your laptop if it improves security?

1	
2	
3	

- e) Using a 3-point scale where 1-No, 2-Maybe and 3-Yes. Would you leave your laptop under the care of a laptop monitoring software?

1	
2	
3	

- f) Using a 5-point scale where 1 means Less Important and 5 means Highly Important. How important do you think a laptop monitoring and tracking software is?

1	
2	
3	
4	
5	

- g) Using a 3-point scale where 1-No, 2-Maybe and 3-Yes. Would you feel comfortable controlling a security application from your mobile phone?

1	
2	
3	

- h) Using a 3-point scale where 3-No, 2-Maybe and 1-Yes. Would you mind receiving security breach notifications on your mobile phone?

1	
2	
3	

8. Request for Contacts for Future Involvement

- 8.1 During the implementation of this project, you may be called upon to test and evaluate the system. Are you interested in participating?

Yes	No	
-----	----	--

- 8.2 If yes, please give us the following information about yourself:

Full Names:

--

Student/Staff Number:

--	--	--	--	--	--	--	--	--

Phone Number:

--	--	--	--	--	--	--	--	--