A MOBILE APPLICATION FOR SUPPLY CHAIN COORDINATION OF ARTEMISININ-BASED COMBINATION THERAPY DRUGS

by

Tshililo Ramovha

submitted in accordance with the requirements for

the degree of

MASTER OF INFORMATION TECHNOLOGY

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof Marcia Mkansi

CO-SUPERVISOR: Prof Ernest Mnkandla

July 2021

DECLARATION

Name: Tshililo Ramovha

Student number: 3878 5374

Degree: MTECH: INF TECHNOLOGY (98802)

Exact wording of the title of the dissertation as appearing on the electronic copy submitted for examination:

A mobile application for supply chain coordination of Artemisinin-based combination therapy drugs

---

I declare that the above dissertation is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the dissertation to originality checking software and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

(The dissertation will not be examined unless this statement has been submitted.)

27 September 2021

ACKNOWLEDGEMENTS

This research was supported by the University of South Africa.

I gratefully acknowledge the comments and suggestions received from Charles Mawadri, Okelle Ambrose, Dr Olutta Nagitta, and my co-supervisor, Prof Ernest Mnkandla.

A special thanks to my supervisor, Prof Marcia Mkansi for your comments and guidance.

ABSTRACT

The purpose of this study was to develop a mobile application for the supply chain coordination of Artemisinin-based Combination Therapy (ACT) drugs that addresses overstock and understock issues in a multi-embedded healthcare supply chain. A literature review explored the supply chain coordination problems, and identified the needs of the various stakeholders at the micro, market and macro environment levels as related to the supply and distribution of ACT drugs. Apart from revealing the gaps at the micro, market and macro environment levels, the study also identified the problems being experienced in African countries due to the centralisation of IT systems, which makes it difficult to control the inventory and to determine when to order drugs.

To address the problem, a quantitative computer programming approach was used to incorporate the Internet of Things (IoT) model and barcode technology. Barcode was identified as a technology that will help to identify, track and monitor trends in terms of the use of ACT drugs. As this technology works best on a smartphone, the study adopted the Internet of Things (IoT) model. The mobile application was developed using Xamarin technology and the Microsoft dot net (.NET) framework. The study used an experimental research strategy to test the system for any anomalies, and the findings revealed a functional mobile application that responds to the needs of the multi-embedded supply chain coordination problem related to ACTs. The study was limited to the computer laboratory, and excluded impact assessments. The developed application will benefit businesses, non-government organizations (NGOs), government agencies and social entrepreneurs to leverage digital technologies to generate positive social impactful outcomes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

The following abbreviations are used throughout the study.

ACT	Artemisinin-based Combination Therapy
API	Application Programming Interface
CMS	Central Medical Store
DLS	district level stores
HMIS	Health Management and Information Systems
IoT	Internet of Things
ICT	Information and Communication Technology
IT	Information Technology
MASA	mesh app and service architecture
MoH	Minister of Health
NHP	National Hospital Policy
NMS	National Medical Store
RMS	Regional Medical Stores
SCM	Supply chain management
SDLC	Software Development Life Cycle

CHAPTER 1:
INTRODUCTION

1.1 INTRODUCTION

This chapter introduces the aim of the study, followed by a discussion of the background to the problem that this research study intended to resolve. In Section 1.4 the problem statement is outlined. To explain clearly what the study tried to achieve, the primary and secondary research objectives are outlined in detail in Section 1.5. Section 1.6 presents the research questions to identify the sub-dimensions that can be converted into technology, and explains how the principles and technologies related to the Internet of things can be used to resolve the identified problems. Section 1.7 discusses the importance and benefits of this study, followed by a discussion of the limitations in Section 1.8.

1.2 BRIEF OVERVIEW

Based on the foundation laid by Nagitta and Mkansi (2019), this study aimed to develop a mobile application (also known as mobile app) for the distribution and supply chain coordination of Artemisinin-based Combination Therapy (ACT) drugs. In their study, Nagitta and Mkansi (2019) addressed the theoretical problem of the multi-embedded supply chain coordination of ACTs. Their findings reveal several dimensions related to the micro, market and macro dimensions that are deemed critical to the availability of ACTs. The study by Nagitta and Mkansi (2019) also revealed critical logistics dimensions that needed to be addressed.

According to several scholars (Nagitta & Mkansi, 2019; Stanley, Cynthia, Chad & Gregory, 2009), lack of the relevant Information and Communication Technology (ICT) systems, such as an appropriate decision support system, amplified the problem of the multi-embedded supply chain coordination of ACTs, which will require a huge investment to solve. Although mobile applications are available in some areas, they do not cover the end-to-end processes of the supply chain coordination of ACT drugs (Mpimbaza *et al.,* 2015). For example, the applications are not integrated, and some of the processes are done manually (Khurana, Chhillar, Kumar & Gautam, 2013),

which makes it difficult to respond to an emergency and leads to the inability to develop forecasts.

Due to the lack of an integrated centralised Information Technology (IT) system, it is difficult to share information, and as a result, the distribution centre is unable to determine the stock levels at each hospital (Umlauf & Park, 2018). This is also the reason why top management is unable to make an informed decision that will help to improve the availability of ACT drugs (Singh, 2011).

This study extended the finding of previous studies by developing a mobile application that complements the theoretical findings of the supply chain coordination framework that was proposed by Nagitta and Mkansi (2019) (to be briefly discussed in the next section). The study applied the Internet of Things (IoT) model (Gelogo, Hwang & Kim, 2015) and its principles, as suggested by Nastic, Sehic, Le, Truong and Dustdar (2014).

Some of the underlying technologies required for the IoT model are a QR code reader and barcode scanner technology, as used in distribution centres, and industries such as the pharmaceutical, warehouse, retailing and logistics industries. The QR code reader, or barcode scanner technology, helps to automatically identify, track and monitor the usage trends of ACT drug packages labelled with a QR code or barcode (Da Xu, He & Li, 2014).

1.3 BACKGROUND TO THE PROBLEM

The public health systems of most countries in Africa are decentralised, which makes it difficult to control the inventory at a local level, and makes it almost impossible to determine when to order stock (Vledder, Friedman, Sjöblom, Brown & Yadav, 2015:4). According to Naggita and Mkansi (2019), the decentralisation of functions is exacerbated by the various levels of coordination within the country. Similarly, Dowling (2011) observed that the supply chain and logistics management of the functions related to the supply of ACTs may operate at different levels. Furthermore, several supply chains are operating within a country with many points of intersection and different kinds of stakeholders.

The lack of an integrated IT system appears to be the underlying cause of the coordination problem. Matthews (2014:3) denoted that the provision and dispensing

of drugs at the state-owned enterprises mainly rely on a centralised distribution model, where the Central Medical Store (CMS) operates as the import hub. The CMSs then export drugs to the Regional Medical Stores (RMS). Thereafter, the RMSs transport the drugs to the district level stores (DLS), or directly to the health facilities.

Williams, Roh, Tokar and Swink (2013:543) indicate that the inability to share supply and demand information between different stakeholders disrupts the ACT ordering process and makes it difficult to act proactively when there is an emergency. It is evident that the decentralisation of the health system causes the supply chain to malfunction (Naggita & Mkansi, 2019).

The distribution centre has no oversight concerning the consumption of drugs, and the delivery of ACTs is based on estimations (Kumar, Singh & Shankar, 2015). Furthermore, the Health Management and Information Systems (HMIS) are unable to provide real-time data for monitoring and reporting purposes, as the data is only processed once a month into the national data bank system and both systems are not integrated (Nanyunja *et al.,* 2011). According to the Budget Monitoring and Accountability Unit (BMAU) (2015:1-4), the continuous stock-outs of medical supplies is caused by the lack of a data management solution that can collect historical and present data, analyse the statistics based on raw information, and deliver insights for making better future decisions concerning prioritisation, forecasting, planning and monitoring (Nakyanzi, Kitutu, Oria & Kamba, 2010:154).

The next section discusses the peculiarities of coordination requirements at the different levels of the ACT supply chain (micro, market, macro and logistics) (Nagitta & Mkansi, 2019).

1.3.1 Micro environment

The term ‘micro environment’ refers to factors that are internal to organisations (Singh, 2011:633). The supply chain coordination dimensions that are related to the micro environment include top management, mutual understanding, relationship management, information sharing, organisational dimensions and responsiveness, as explained below (Singh, 2011; Nagitta & Mkansi, 2019):

- Top management is responsible for making decisions on what should happen when the health facility runs out of stock, they also support redistribution (of resources) and support Continuous Medical Education (CME).
- Mutual understanding is associated with the support for organisational strategic goals and the building of trust with employees, as supported by Boswell (2006:1504).
- Information sharing helps the organisation to be proactive concerning the support and supply of ACTs (Chen, Kazman & Haziyev, 2016; Pinna, Carrus & Marras, 2015).
- Responsiveness refers to a capable system that responds swiftly when the need arises.
- Demand management helps supply chain employees to be responsive (Turkyilmaz, Bulak & Zaim, 2015:1-2).
- The organisational dimensions ensure that state-owned entities, like general hospitals, have systems in place that will enable top management to deal with demand management, with little or no participation from the operations (Han & Hong, 2016).

1.3.2 Market environment

Watsierah and Ouma (2014) assert that at the level of the market environment, stakeholders, including retailers, have to integrate their system so that they can share critical information and are able to make ACTs available to the patients. Unfortunately, there is no integration of systems between the internal and external stakeholders to facilitate partnerships and collaborations, and to enable decision-making and develop joint research and educational plans. The critical supply chain dimension necessary for the distribution of ACTs from the market environment include supply chain interdependence and information sharing with donors and the Ugandan Ministry of Health (MoH), in the case of the current study.

1.3.3 Macro environment

The macro environment refers to the factors that affect the delivery of ACTs, such as political, economic, technological and legal factors, as discussed below.

- **Political:** Politicians exercise their influence to support the continuous supply of ACTs to the hospitals and clinics. They mostly address the community negatively concerning service delivery to their benefit, according to Park, Chang and Jung (2017).
- **Economic**: According to Lalvani, Yadav, Curtis and Bernstein (2010), a developing country, such as Uganda, depends on donors to ensure the supply of ACTs to their hospitals. The level of poverty among the community members, especially in Uganda, contributes to the levels of ACT stock-outs because the people cannot afford to pay for the medicine and depend on the hospitals to supply it (Bate, Hess & Mooney, 2010).
- **Technology**: The lack of an analytics tool to track the consumption of ACTs, stock status and ordering plan creates a problem concerning the availability of ACTs (Cocosila & Archer, 2010).
- **Social-cultural**: Even though the government can make an effort to ensure that ACTs are available at the hospitals, community members believe in self-medication, hence, when they hear that the medicine is available they go to the hospital even though they are not sick (Granovetter, 2005).
- **Legal**: The regulatory framework and policies that guide the use of ACTs are available but are not properly enforced (Iqbal, Geer & Dar, 2017). Following a confirmatory factors analysis and multi-decision criteria, the study by Nagitta and Mkansi (2019) revealed that the critical supply chain coordination sub-dimension factors from the macro environment which call for technological attention are the legal, social-culture and technology factors. Legal factors include the enforcement of policies and procedures that form part of the business rules required for system control (Iqbal *et al.*, 2017). For example, implementing a first-in first-out, or last-in last-out policy during the dispensing of ACTs.

1.3.4 Logistics

The term ‘logistics’ refers to the critical activities that influence the availability of ACTs. The supply chain in a typical hospital has five basic functions, namely, forecasting, procurement, quantification, storage, and dispensing (Iqbal *et al.*, 2017).

Forecasting is the primary activity within the healthcare logistics cycle, followed by quantification. Kasapoglu (2016) states that the challenge with drug forecasting is that

forecasts have to rely on previous months' dispensation from the health facilities which was processed manually and was based on individual experience and the historical knowledge base.

According to Khurana *et al.* (2013:9), there is also the challenge of the terminologies given to malaria therapies. At some hospitals, the administrator has challenges in monitoring the stock levels due to the different terminologies being used.

In Uganda, the procurement of drugs is planned based on the available budget and not on the demand, and general hospitals are not autonomous entities (Hakiza & Basheka, 2012).

1.4 PROBLEM STATEMENT

A study by Bruno, Nyanchoka, Ondieki and Nyabayo (2015) reports on the multi-embedded supply chain coordination problem that has an impact on the availability and distribution of ACTs in Uganda. The problem is amplified by the lack of technology that can integrate different stakeholders across the micro, market and macro environment (Kraiselburd & Yadav, 2012). Furthermore, there is the high use of manual and legacy systems which are not integrated, making it hard to coordinate the logistics activities necessary for the availability and distribution of ACTs, such as procurement, dispensing, forecasting, quantification and store management (BMAU, 2015).

According to Nagitta and Mkansi (2019), the current coordination of ACTs involves different stand-alone systems, such as M-track, the Health Management Information System (HMIS), logistics management information system (LMIS), m-Health, Rx-solution (the electronic drug management system), and the short messaging service (SMS), all of which are not integrated. The systems do not address the multi-embedded supply chain problem that integrates stakeholders at the micro, market and macro levels, and their functionalities do not support the monitoring and management of ACTs at the different levels of the supply chain.

1.5 RESEARCH OBJECTIVES

This study aims to develop a mobile application (mobile app) for the supply chain coordination of Artemisinin-based combination therapy (ACT) drugs, and the objectives that were formulated for the study are presented below.

1.5.1 Primary research objective

The primary research objective of the study was to develop a mobile application using the Internet of Things (IoT) model outlined in more detail in Section 2.2.2.3 and the system development life cycle outlined in detail in Section 2.3.

1.5.2 Secondary research objectives

The following secondary research objectives were formulated for the present study:

1. To identify the critical logistics, micro, market and macro dimensions, and the associated sub-dimensions of ACTs that can be converted into technology features. System requirements gathering and analysis was used to achieve this.
2. To map the features of ACT software for stakeholders across the micro, market and macro environments. System design was used to achieve this.
3. To apply the principles of the IoT and technologies to develop the software, and to link stakeholders across the micro, market and macro environment. System implementation was used to achieve this.

1.6 RESEARCH QUESTIONS

The following research questions were formulated to help to identify the sub-dimensions that can be converted into technology features and the IoT principles and technologies that can be used to develop the app.

1. What are the critical logistics, micro, market and macro dimensions, and associated sub-dimensions of ACTs that can be converted into technology features?
2. What are the features of ACT software that are associated with the micro, market and macro stakeholders?

3. How can IoT principles and technologies be applied to the development of software and to connect stakeholders at the micro, market, and macro levels?

1.7 SIGNIFICANCE OF THE STUDY

This study will improve the processes of coordinating the supply and distribution of ACTs and address the issue of the non-availability of ACTs to patients who are in need, by developing an integrated mobile application solution that provides real-time monitoring of the stock delivery, stock level and consumption level at each health facility. The integrated mobile application solution will be patient-oriented to enable a seamless health service experience across the health system.

1.8 SCOPE AND LIMITATION OF THE STUDY

This study and the development of the mobile application are limited to the scope of data obtained from a three-year research project conducted by Nagitta and Mkansi (2019) to advance the theoretical supply chain coordination of ACT drugs. The mobile application is intended for use by general hospitals in Uganda, and excludes the private sector.

1.9 CHAPTER SUMMARY

This chapter introduced the aim of the study, and also revealed that lack of Information and Communication Technology amplified the problem. It further explained the challenges of applications that are not integrated and how this problem could be resolved. The chapter also introduced the requirements, according to the following dimensions, namely, micro, micro, market and logistics. This chapter further presented the research objectives, the research questions, the importance of this research and discussed the limitations of the study. The next chapter deals with the main concepts and the models that will be used to address the problem.

CHAPTER 2:
LITERATURE REVIEW

2.1 INTRODUCTION

Chapter 1 gave a preview of the supply chain problem as related to the supply of ACTs, and how the researcher intended to solve the problem by developing a highly specialised app based on the principles of the IoT. Such a solution will be compatible with low-performance mobile devices, using aspects of the technological features derived from the theoretical framework developed by Nagitta and Mkansi (2019). In this chapter, the researcher provides an in-depth review of the theory behind the technology use and the supply chain problem. The theory discussion is a hybrid of the supply chain coordination theory, which covers the problems under study, and the IoT model, which covers the solution that was used in the current study to solve the problems.

2.2 THEORETICAL BACKGROUND

This section discusses the theory related to how to improve the distribution of ACT drugs by firstly, discussing the supply chain coordination problems. Thereafter, the IoT model and how to integrate IoT with the micro, macro and market dimensions are discussed.

2.2.1 Supply chain coordination theory

Coordination theory posits that greater effectiveness and efficiency can be achieved in an organisation by the continuous identification and assignment of tasks, and their respective interdependencies (coordination dimensions) (Malone & Crowston, 1994). The theory has been advanced by scholars in several business sectors, ranging from plant manufacturing and retailing, to supply chain management (Labiad, Beidouri & Bouksou, 2014). A common underlying principle arising from the latter scholars is that every organisation needs to continually identify responsibilities, together with their interdependencies, referred to as coordination dimensions or mechanisms, among the different actors in the workplace to ensure that all activities are ‘glued together’ (coordinated) to achieve better performance. Labiad *et al.* (2014) further emphasise that such elements are ‘glued together’ by information technology (IT), a dimension

(or mechanism) which is missing in the distribution of ACTs in Uganda. Organisations that fail to coordinate their supply chains face the challenges of irresponsive order fulfilment, excess inventory, low capacity utilisation, high total cost, and low customer satisfaction, among others (Simatupang, Wright & Sridharan, 2002).

This theory not only provides the background to the principle of coordination but also offers an approach that can be implemented to coordinate the supply chain problem using coordination theory. The study by Nagitta and Mkansi (2019) offers an extension of the coordination theory by including the elements and stakeholders involved in a multi-embedded supply chain, with specific focus to ACTs. The latter scholars offer the platform for the mobile application development, as shown in the next sections.

The present study used coordination theory as the lens that investigated how the micro, market, macro and logistical activities of stakeholders can be connected through the use of IT to enable the efficient and effective distribution of ACTs. The framework or findings that arose from Nagitta and Mkansi's (2019) coordination framework provided the various dimensions that are critical for the coordination of ACTs across the different stakeholders at the micro, market and macro level, and the relevant logistical activities.

The lack of an integrated technological platform has been cited as one of the key challenges in managing the distribution, and overstock and understock of ACTs. Scholars refer to IT as the critical thread that connects the various stakeholders and the different aspects of coordination theory (Matthews, 2014; Williams *et al.*, 2013).

As technology is the backbone that holds together coordination, this study bridges into the IoT as the theoretical lens that will be able to connect mobiles across the various stakeholders (micro, market and macro) and products (herein, ACTs), as discussed in the next section.

2.2.2 Integration model for supply chain coordination framework

According to Nagitta and Mkansi (2019), the challenges that arise due to the decentralisation of functions in Uganda are exacerbated by the levels of coordination within the country. Similarly, Dowling (2011) observed that the supply chain and logistics management of the functions related to the supply of ACTs may operate at different levels. The applications that are available for supply chain and logistics

management are not integrated causing difficulties in the responsiveness to an emergency, and an inability to develop forecasts (Mpimbaza *et al.*, 2015). Due to the lack of an integrated centralised IT system it is difficult to share information, and as a result, the distribution centre is unable to determine the stock levels at each hospital (Umlauf & Park, 2018).

Lapalme *et al.* (2016) mentioned that modern applications and modern system architectures are inherently heterogeneous, distributed and diverse. Keiningham *et al.* (2020) further mentioned that at the same time, business goals, such as an optimised customer experience, process efficiency, data quality and digital business agility, require application integration architects to connect and compose the elements of those systems in timely, efficient, reliable, secure and cost-effective ways.

According to Kähkönen (2017), most organisations have existing integration middleware that cannot be easily retired, and that offer more connectors to legacy systems. The result is a diverse landscape of integration models to choose from. This section compares and explores the strengths and weaknesses (SWOT) of the three most common integration models, namely, the Enterprise Service Bus (ESB), Integration-Platform-as-a-Service (iPaaS), and the IoT model.

The current research study limited the scope to just those integration models that provide an application-centric approach to mediation. These three integration models are analysed in the next sub-sections and Table 2.1 (Li & Chen, 2012; Sutherland & Chetty, 2016; Giret, Garcia & Botti, 2016; Popa & Vaida, 2019).

2.2.2.1 Enterprise Service Bus

Giret *et al.* (2016) define the Enterprise Service Bus (ESB) as a model that defines a centralised middleware platform able to integrate multiple applications and act as a data repository. The major strength of ESB is that it defines the standard platform for integration with a prebuild adapter to connect both applications and data repositories. However, its major weakness is that ESB is not agile and flexible because of its monolithic architecture, Furthermore, ESB solutions do not support multitenancy which is discordant with the project under study.

2.2.2.2 Integration-Platform-as-a-Service

Integration-Platform-as-a-Service (iPaaS) defines how applications and a data repository integrate in a cloud model, making it easier to share data across applications (Sutherland & Chetty, 2016; Toman, Furey & Curran, 2020). The major strength of IPaaS is that it standardises the integration of cloud-based applications. In addition, it supplies data sources with on-premises systems using an Application Programming Interface (API), and lastly, IPaaS is agile. However, the flexibility is limited because of vendor dependency or locked which also make it unsuitable for this project under study.

2.2.2.3 Internet of Things model

The IoT model defines the interconnection of IoT devices, such as mobile devices, with either the cloud or an on premise computing model (Caivano, Cassano, Lanzilotti & Piccinno 2018). The strength of IoT model is that each component is independent, making it more flexible and agile. In addition, the integration is API-centric, and most of the IoT applications and data repositories can be deployed either on the cloud or on premise.

However, smart devices depend on good network coverage, which is a weakness, especially in a country, or place where that is lacking. The objective of the study is to develop a mobile application that can be used at any place and that can be able to share data in real time. This weakness will be addressed by ensuring that all the identified areas where the ACT drugs need to be administered or dispensed must have full network coverage as a prerequisite. The next section justifies why the IoT model was relevant for the project under study.

Table 2.1 summarises the main strengths and weakness associated with each integrated model.

Table 2-1: Summary of Integrated Model

Model	Strengths	Weakness	Reference
Enterprise Service Bus (ESB)	▪ Provide standard platform for application integration. ▪ Consistent way to integrate services into the architecture. ▪ Prebuilt components and adapters into various protocols.	▪ Single point of failure. ▪ Still evolving in support for cloud application integration. ▪ Encourages monolithic architecture. ▪ ESB solutions do not support multitenancy.	Giret *et al.* (2016), Thönes (2015), Khan, Sengupta & Sarkar (2015)
Integration-Platform-as-a-Service (iPaaS)	▪ iPaaS offers deep support for cloud service integration, from both integration and operational perspectives. ▪ Integrated applications can operate with a variety of protocols and data formats. ▪ Strong support for an API-centric approach. ▪ Enables organisations to streamline their processes by creating seamless connectivity.	▪ Limited support for Commercial off-the-shelf (COTS) application connectors and legacy protocols and formats. ▪ Integration metadata and operational data are stored and processed in the provider's cloud platform. ▪ Data governance cannot be automatically enforced. ▪ Geographic coverage.	Sutherland & Chetty (2016), Toman *et al.* (2020)
Internet of Things	▪ Cloud deployment and integration. ▪ Application and data repository integrated using API-centric. ▪ Application enablement.	▪ No international standard of compatibility for the tagging and monitoring devices. ▪ Bandwidth coverage.	Li (2013), Li & Chen (2012), Caivano *et al.* (2018)

Source: Researcher's own compilation

2.2.3 Justification for IoT as the model to integrate supply chain coordination framework

The term ‘Internet of Things’, also known as IoT, refers to "a network of everyday devices, appliances, and other objects equipped with computer chips and sensors that can collect and transmit data through the Internet" (King, 2018, para. 1). One family of technology that fuels the IoT is Auto-ID. According to Ashton (2009), Auto-ID is identification technology that is used in the industry to automate the reading or capturing of product information to increase efficiency and reduce errors.

According to Statler (2016), this includes technologies such as the Barcode reader, Quick Response (QR) reader, Near-field Communication (NFC), and Radio Frequency Identification System (RFID). The collected data is transformed into actionable data to gain an accurate and deep understanding of business, which can enhance business operations, efficiency, or even foretell future events, like stock running out, without any direct human intervention (Statler, 2016). According to Li and Chen (2012), the IoT model comprises three layers: the sensing layer, network layer and application layer that have been acknowledged from diverse industries. The three layers are briefly discussed below.

2.2.3.1 Sensing layer

According to Puccinelli and Haenggi (2005), the sensing layer mostly deals with small electronic devices that have limited memory, and are powered by a battery with an on-board actuator and sensor. These devices are able to operate independently as sensing devices, or are built as part of the system for sensing and control. A typical IoT device has the following three main capabilities, namely, to sense or record data, to perform light computing, and to share data using the network. There are many types of sensors attached to objects or devices to collect information, such as RFIDs, QR codes and barcodes (Lotlikar, Kankapurkar, Parekar & Mohite, 2013).

Lotlikar *et al.* (2013) did a comparison of the following types of scanners, namely, the barcode, QR code and RFID, as below.

- The barcode is used most often in retailers. It is accurate and easy to use, however, barcodes require a huge investment.
- QR codes are used most often in manufacturing, and are easy to use. They provide quick access to mobile customers, and QR codes can be created without any costs.

- RFID tags are used in a wide variety of industries, for example, they are used in theft prevention in retail. However, RFID tags require high start-up costs. RFID has now emerged to printed RFID labels to address nanotechnologies (Singh, Singh & Nalwa, 2017).

The current study made use of sensing devices on ACT products as a key potential link to the network and application layers. The following section provides an understanding of the network layer and its relevance to this study.

2.2.3.2 Network layer

The different IoT devices found on the sensing layer need to be connected to the mobile application via network technologies (Sun, Song, Jara & Bie, 2016). The network technologies transmit to the application layer. These network technologies are critical components of the IoT ecosystem (Hammi, Khatoun, Zeadally, Fayad & Khoukhi, 2017). A study by Aloi *et al.* (2017) indicated that typical IoT technologies are equipped with communication capabilities that enable them to communicate with the IoT devices on one end, using the Internet Protocol (Internet) to connect to the other side.

The current study explored the cloud, mobile applications and web technologies that are crucial for integration with the application layer within the context of the Ugandan healthcare case study. Furthermore, the study considered the use of unstructured supplementary service data (USSD) in cases where patients have limited access to the internet, as a possible method to link patients with healthcare centres. Below is a review of how the network layer connects to the application layer.

2.2.3.3 Application layer

The applications of IoT differ, depending on the service that the organisation provides. The majority of the research has been done on RFID (Yu & Jiang, 2017), which is extensively utilised in the manufacturing and retail industry (Nayak, Singh, Padhye & Wang, 2015). For example, when the products are scanned for dispatch, top management can monitor the delivery status and they will be able to estimate the delivery date (Paul, Chatterjee & Guha, 2019). Many retailers, such as Walmart in the United States of America (USA), use a dashboard that has the capabilities to automatically analyse the movement of stock without human intervention, and this

helps them to become more proactive (Kay, High, Atchley & Walmart Stores Inc., 2018).

According to Alicke, Rexhausen and Seyfert (2017), when consumption increases, management will be able to see the real-time stock status on the dashboard, which will allow them to respond to the relevant area. Real-time data empowers management to plan improvements in decision-making, and depending on the forecasting output, management will be able to limit further losses for the company (Sousa, Pesqueira, Lemos, Sousa & Rocha, 2019). Apart from these types of applications offering real-time help to organisations that allow them to gain an accurate and deep understanding of forecasting errors, they also enable organisations to react rapidly to mitigate the effects of an operational problem.

Gao and Su (2016) indicate that most retailers and manufacturers have introduced online purchases that require some type of technology. For example, when a consumer completes an order at a Walmart shop, the consumer receives a notification that will enable them to track the order with a mobile app (Rajas & Pund, 2017). According to Wheelock, Weiss, Sweet, Allred and Becker (2016), when the order arrives at the destination store, the consumer receives an SMS notification.

The use of QR codes is increasing, even though RFID still dominates in packaging departments (Chen, Du, Cheng & Po, 2016). Retailers and the manufacturing industry have developed a relationship in the form of a contractual agreement with their suppliers that are using systems built with different system interfaces (Chkanikova, 2016). They have real-time access to the reporting (dashboard) functionality of the system, for example, when the demand increases, they receive an alert (Sousa *et al.*, 2019).

The findings from the various studies discussed above and the practice of supplier-retailer integration as related to the distribution and management of products (Sousa *et al.*, 2019; Wheelock *et al.*, 2016; Li, 2013) provided a guideline for the current study on how the micro, market, macro and logistical activities, and patients can be integrated at the application layer of the IoT principle. The study further explored the use of auto-ID (Barcode reader, QR, NFC, RFID) to scan ACT products. Below is an in-depth discussion of the hospital structure to be connected by the IoT in Uganda.

2.2.3.4 Hospital structures and general supply chain of ACT drugs in Uganda

Hospitals that form an integral part of the health system in Uganda provide primary, secondary and tertiary health services. Hospital service delivery in Uganda is guided by the National Hospital Policy (NHP) whose main goal is to ensure equity of access to hospital services through effective management resource mobilisation, and guaranteeing that hospitals provide quality and affordable services. Below is an illustration of the supply chain flow of drugs from the National Medical Store (NMS) to the health centres.

As seen from Figure 2.1 below, all public hospitals, including national, regional referral and district hospitals, are served drugs direct from the NMS on a pull system. However, the hospital may also receive drugs directly from donors or procure them from recommended private pharmacies.

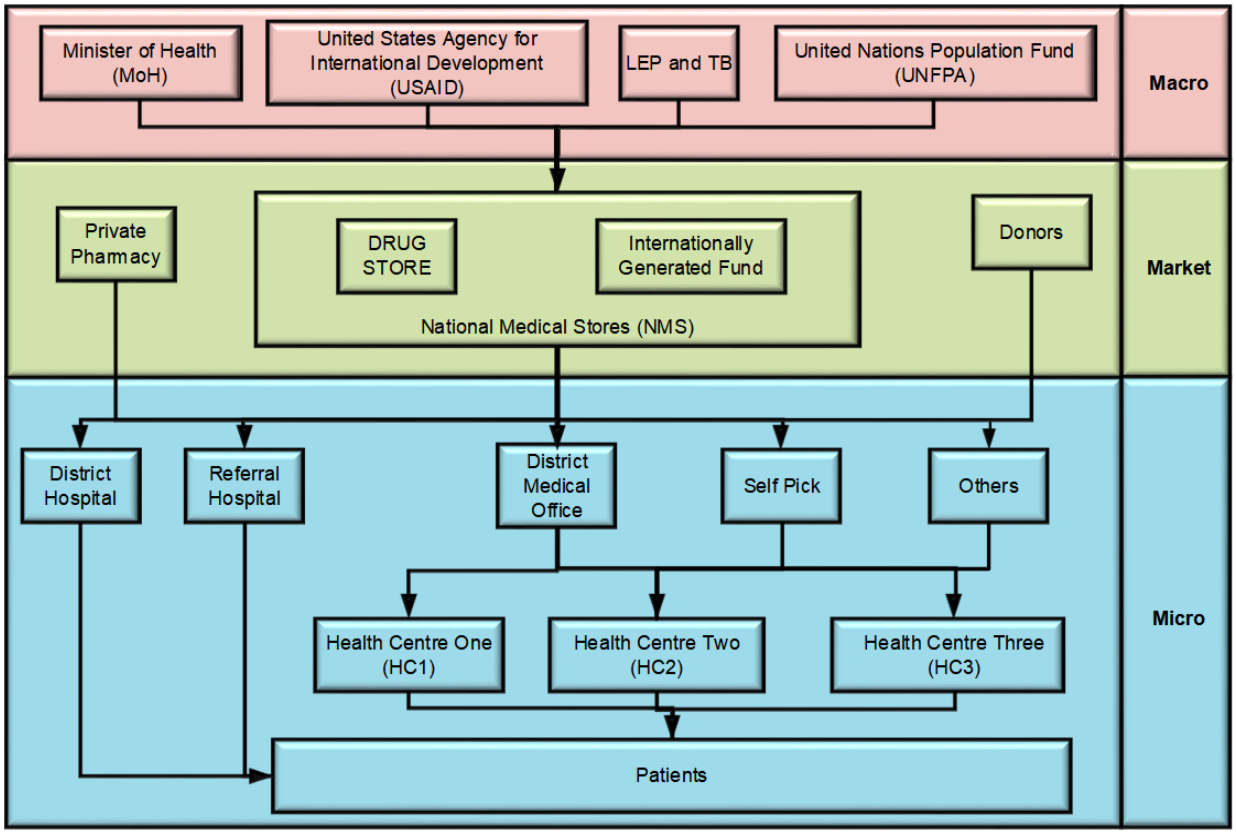


Figure 2.1: General supply of ACT drugs in Uganda and stakeholders

Source: Nagitta & Mkansi (2019)

2.2.4 Approach to coordination problem using IoT

The current study drew on the framework developed by Nagitta and Mkansi (2019) in gaining an understanding of how the coordination of supply chain interdependencies

at the micro, market and macro levels impact the availability of ACTs in general hospitals in Uganda. The different coordination dimensions, as discussed in Section 1.3.1 to 1.3.4, were converted into theoretical technological features which allowed the researcher in the current study to develop a solution that would cover all the identified stakeholders. The theoretical technological features were mapped based on the relevant functionalities as related to the various stakeholders.

Figure 2.1 above shows the complex structure of the medicine supply chain and key players at the micro, macro and market environments. Typically, the hospital supply chain management (SCM) includes the internal chain that consists of aspects such as the hospital warehouse, patient care units and patients, while the external chain consists of manufacturers, distributors, and vendors (Rivard-Royer *et al.,* cited in Laundry & Beaulieu, 2013:468).

A three-year investigation by Nagitta and Mkansi (2019) revealed the interactions of the micro, market and macro environments, and the associated critical coordination dimension in developing countries with regards to the supply and distribution of ACTs.

Figure 2.2 below provides the conceptual technological interpretation of the critical supply chain coordination across the micro, market, macro and logistical activities that was implemented in the current study using the IoT model.

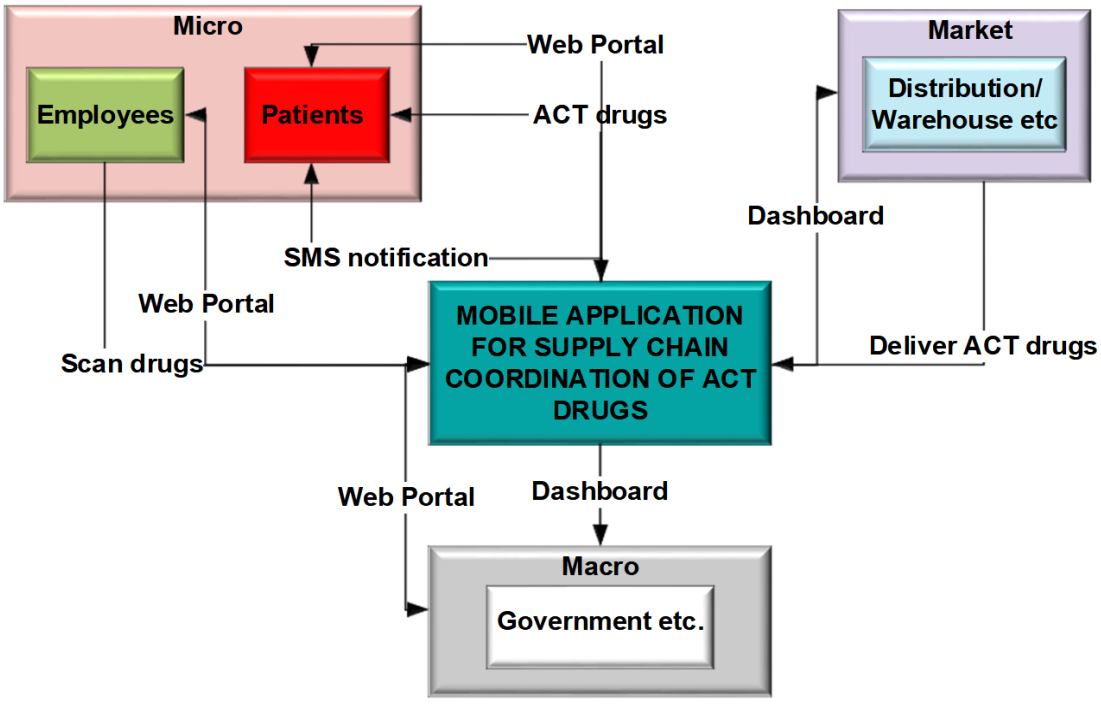


Figure 2.2: Main functionality and stakeholders

Source: Researcher's own compilation

Table 2.2 below outlines the preliminary mapping of the IoT model explained above in Section 2.2.2. The table summarises the high-level technological features of the proposed solution.

Table 2-2: Mapping of IoT model with high level technological features

IoT layer	Environment	Features
Application	Micro	Dashboard, web portal to track and trace ACT drugs, dispensing of ACT drugs to patients, SMS notification.
	Macro	Dashboard to view and analyse stock status, view the demand of drugs.
	Market	Dashboard, view the demand of drugs.
Network	Network infrastructure	
Sensor	Micro	Interface to scan ACT drugs when dispensing to patient and web portal for administration.
	Market	Interface to scan ACT drugs when distributing to district office, hospital, and so on.

Source: Researcher's own compilation

The market and macro environments are part of the major management environment, as discussed in detail by previous studies (Nieman & Bennett, 2002; Fahey & Narayanan, 1986). The main feature of relevance to the current study is the dashboard that allows the user to view and analyse the stock status and to respond to demand.

At the micro-environment level, employees from the various health care centres, namely, the district hospitals, referral hospitals, district medical offices and health centres, receive ACT drugs from either private pharmacies, national medical stores or donors, as illustrated in Figure 2.1 above, which they then dispense to patients. The ACT drugs are scanned using the sensor devices that are in use at the distribution centre before the drugs are dispatched to the hospitals and health centres. It will automatically update the system and reflect on the dashboard. A similar principle will apply when an employee dispenses ACT drugs to patients, it will automatically inform patients by SMS.

The IoT model is illustrated in Figure 2.3.

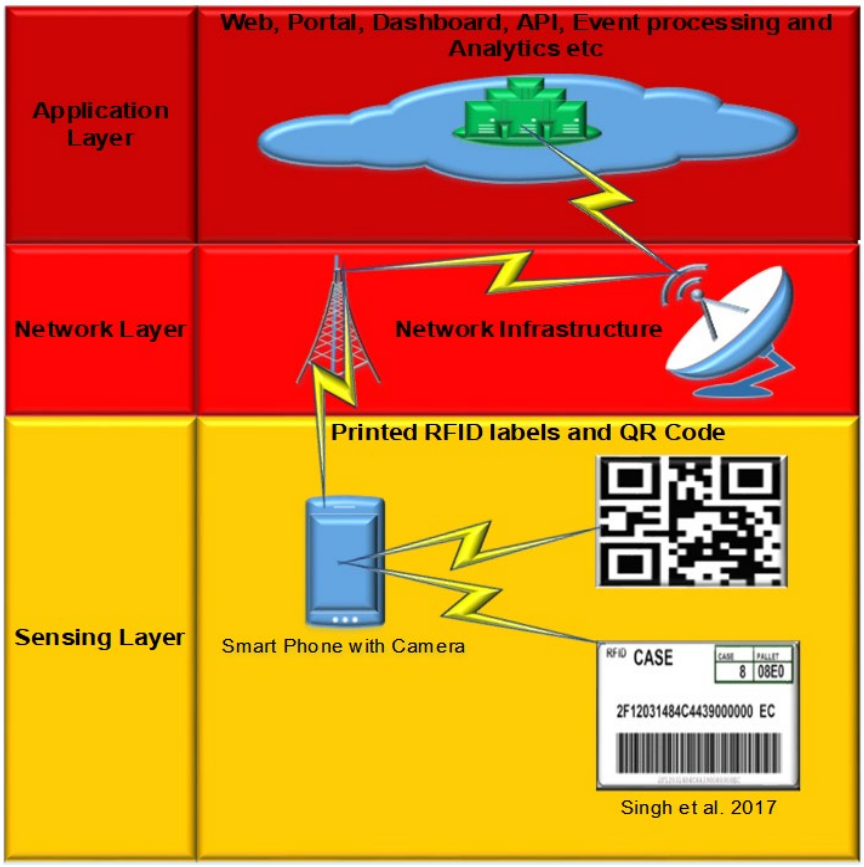


Figure 2.3: IoT Model

Source: Li (2013)

Whilst the IoT computer models provide the model for digitalising a supply chain coordination framework, Section 2.3 reviews the system development methodologies that are crucial in bringing the digital system to life.

2.3 SOFTWARE DEVELOPMENT METHODOLOGIES

According to Bassil (2012), the Software Development Life Cycle (SDLC) is often used in industries such as finances, government, engineering and computational sciences. Bassil (2012) also stated that the following are the most common, popular and successful SDLC models: the waterfall model, iterative model, spiral model, v-shaped model and agile model. These models will be discussed in this section.

The SDLC models mentioned above share similar fundamental features. They are made up of phases or steps to accomplish the results and produce a final product or solution.

2.3.1 Waterfall Model

The waterfall model is a structured SDLC, which means that you need to finish one stage before moving on to the next stage (Kramer, 2018; Afif *et al.*, 2020; Almeida & Simões, 2019). Afif *et al.* (2020) added that each stage relies on output from the previous stage and has its project plan. The model is suitable for a project with predefined complete requirements that cannot be changed at a later stage (Almeida & Simões, 2019). The waterfall is easy to understand and simple to manage.

The waterfall model was selected for the ACT project because of its requirements that did not need to be changed within a limited time frame. The resources could be prescheduled as per the project plan and stage of the project.

2.3.2 V-Shaped Model

Saravanan, Jha, Sabharwal and Narayan (2020) mentioned that the V-shaped model grew out of the waterfall and is characterised by a corresponding testing phase for each development stage. Like the waterfall, each stage begins only after the previous one has ended. This model is useful when there are no unknown requirements, as it is still difficult to go back and make changes. The V-shaped model is suitable for small projects, and where all the users and technical or development stakeholders are available for each stage; therefore, it was not suitable for this project.

2.3.3 Iterative Model

Sharma and Bala (2019) mentioned that the iterative model gradually develops the features or functions, which it immediately releases to production and does not wait for other features to be completed. Okesola (2020) added that instead of starting with fully known requirements, a set of software requirements is implemented, then tested and evaluated, after which further requirements are pinpointed. Sharma and Bala (2019) indicated that with each cycle, a new version of the software is produced. This model gives a working version early in the process. However, the model demands users to be always available, which would not have worked for the current project.

2.3.4 Spiral Model

Miraz and Ali (2020) highlighted that the spiral model is one of the most flexible SDLC methodologies that takes a cue from the Iterative model and its repetitions. The project passes through four phases over and over in a ‘spiral’ until completed, allowing for

multiple rounds of refinement (Dhir, Kumar & Singh, 2019). The spiral model is suitable for a project with components or segments that need to be completed and delivered immediately to the operating environment. As the completed system had to be delivered at the end of the project, the spiral model was not suitable for the current study.

2.3.5 Agile Model

By breaking the requirements into cycles, the agile model quickly delivers a component of a system (Saravanan *et al.*, 2020). Okesola *et al.* (2019) further state that the model produces ongoing releases, each with small, incremental changes from the previous release. At each iteration, the component is tested. Tam, Da Costa Moura, Oliveira and Varajão (2020) added that this model emphasises interaction, as the users, developers and testers work in collaboration throughout the project. Jha, Sabharwal and Narayan (2020) argue that since this model depends heavily on users' interaction, the project can go wrong if the user is not exactly clear what he or she requires of the project.

It was decided that this model would not be suitable for the development of the mobile app in the present study because the app needed to be developed based on the pre-approved scope, and no changes could be made as the continuous feedback from users could create scope creep. Similar to the iterative model, the agile model will not work for the project under study because users always needed to be available to provide feedback.

Table 2.3 summarises advantages and disadvantages associated with each SDLC model.

Table 2-3: Advantages and disadvantages of the SDLC Model

Model	Advantages	Disadvantages	Reference
Waterfall	▪ More detailed, robust scope and design structure due to upfront planning and documentation. ▪ Forces structured organisation. ▪ Allows for early design changes. ▪ Suited for milestone and date-focused project. ▪ Simple to use and understand. ▪ Each phase has a defined result which minimises the risk. ▪ Easy to classify and prioritise tasks.	▪ The system is ready only after the last phase is complete. ▪ The late discovery of flows can lead to a devastating realisation regarding the legitimacy of the entire system. ▪ Late client feedback. ▪ Delayed testing period.	Kramer (2018); Afif *et al.* (2020); Almeida & Simões (2019)
Iterative	▪ Some functions can be quickly developed at the beginning of the development lifecycle. ▪ Flexibility and readiness to the changes in the requirements. ▪ Ensures that newer iteration is an incrementally enhanced version of previous iteration's version. ▪ Turnaround time is smaller. ▪ Easy adaptability.	▪ Iterative model requires more resources than the waterfall model. ▪ Constant management of change. ▪ Increased pressure on user engagement. ▪ Scope creep.	Okesola (2020); Sharma & Bala (2019)
Spiral	▪ Lifecycle is divided into small parts, and if the risk concentration is higher, the phase can be finished earlier to address the threats. ▪ The development process is precisely documented yet scalable to the changes. ▪ The scalability allows for changes and adding new functionalities, even at relatively late stages. ▪ The earlier the working prototype is done, the sooner users can point out flaws.	▪ The risk control demands involvement of highly skilled professionals. ▪ Big number of intermediate stages requires excessive documentation.	Dhir *et al.* (2019); Gaur & Aggarwal (2019); Miraz & Ali (2020)

Model	Advantages	Disadvantages	Reference
V-shaped	▪ Every stage of V-shaped model has strict results, so it is easy to control. ▪ Testing and verification take place in the early stages. ▪ Good for projects where requirements are static and clear.	▪ Lack of the flexibility. ▪ Relatively big risks.	Saravanan *et al.* (2020)
Agile	▪ Focus on modern techniques. ▪ Highly adaptive. ▪ Collaboration with user assists with quick feedback. ▪ Allows for iterative development.	▪ Not easy to estimate the time and resources required. ▪ System not properly documented. ▪ Users are not always available, which may delay the release. ▪ Scope creep.	Okesola *et al.* (2019); Saravanan *et al.* (2020); Tam *et al.* (2020)

2.4 JUSTIFICATION FOR SELECTED SOFTWARE DEVELOPMENT METHODOLOGY

The waterfall model (Figure 2.4) which is one of the most popular SDLC models, according to Balaji and Murugaiyan (2012), contains five sequential stages, and they are as follows: analysis, design, implementation, testing and maintenance.

Mahalakshmi and Sundararajan (2013) explained that due to the fact that the Waterfall model has existed for many years and is the most popular of the traditional software development models, most software development companies and industries have used it as their chosen development methodology to develop, support and maintain their software.

The scholars, Ragunath, Velmourougan, Davachelvan, Kayalvizhi and Ravimohan (2010) outline that the waterfall model has sequential stages that have to be performed one after the other. The study can proceed to the next stage only when its earlier stage is fully completed and signed off by the project sponsor. The waterfall model can be continuously iterated at each stage until the objectives have been met.

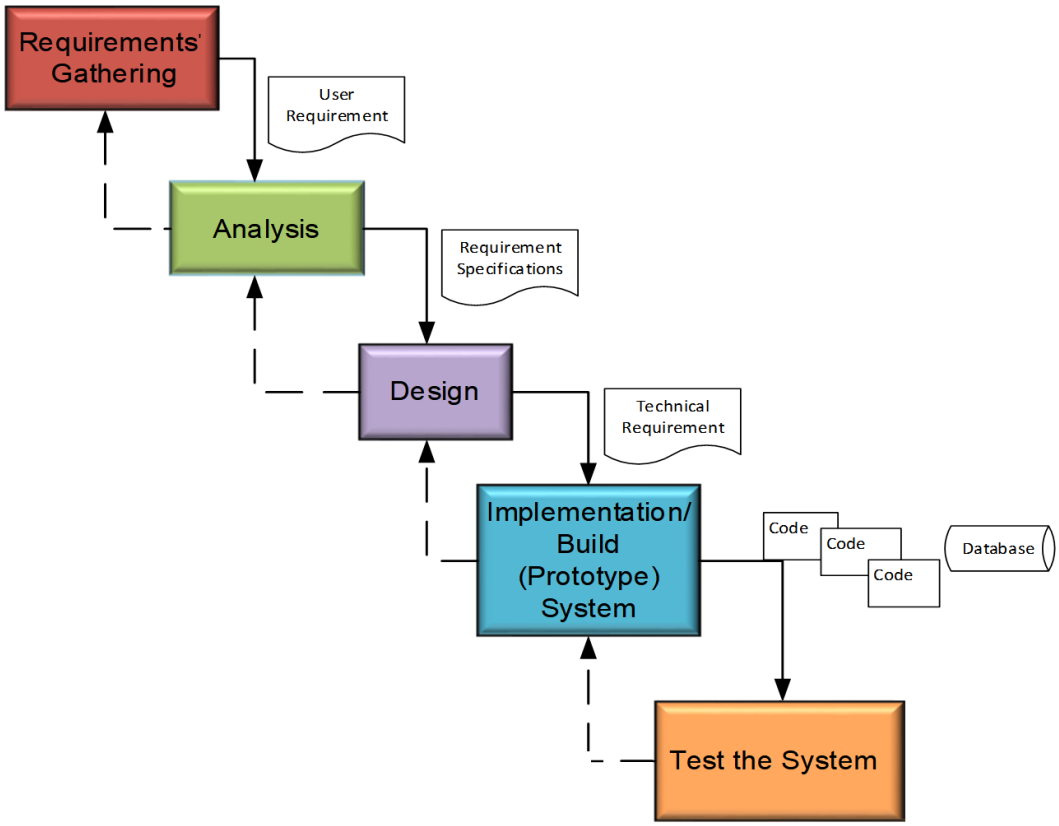


Figure 2.4: Waterfall Model

Source: Adapted from Ragunath *et al.* (2010)

2.4.1 Requirements' gathering stage

During the requirements' gathering stage, data is gathered to describe how the software will function and be developed. The requirements' gathering stage is outlined in detail in Chapter 4.

2.4.2 Analysis stage

The analysis stage categorises the user requirements according to the functional and non-functional requirements, and then converts the user requirements into a model. The current study outlined in detail both the functional and non-functional requirements of the study.

Normally, functional requirements are represented using use cases, as discussed briefly in the next section. The use case illustrates the users' interactions with the system or software. The flowchart showing the information flows is discussed briefly in Section 2.4.2.2.

The term 'non-functional requirements' refers to the different measures, constraints, limitations and requirements required in the design and operation of the software or solution.

2.4.2.1 Use cases

According to Booch, Rumbaugh and Jacobson (2010), a use case can be defined as a graphical representation of the identified functionalities. It also represents the application's workflow by showing the entry point to the process up to the exit point. A use case diagram usually portrays the relationship between the actor and the system. Lee (2012) recommends the use of use case diagrams to enhance the understanding of the solution and they can also be used to describe the system.

Figure 2.5 below shows a high-level use case diagram of a mobile app for ACT drugs.

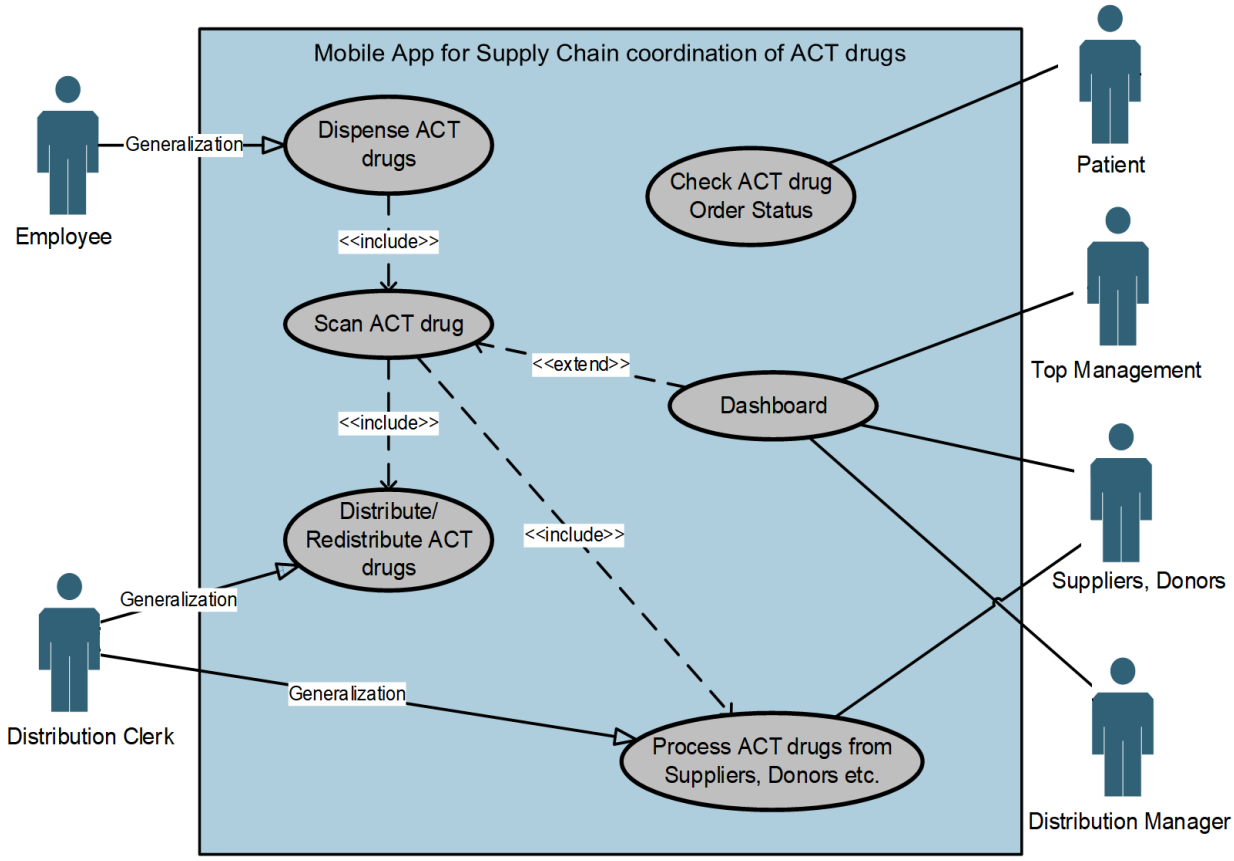


Figure 2.5: High-level use case diagram for ACT drugs' mobile app

2.4.2.2 Flowcharts

According to Khan and Khan (2011), a flowchart is a graphical depiction of a process or system that describes the sequencing of actions needed to generate output. A flowchart is frequently used to document a system. Flowcharts may cover diverse levels of detail as required, from a high-level overview of a whole application system to a detailed diagram of one process within a bigger system.

Recker (2010) mentions that flowcharts are a vital instrument for the development of processes. By presenting a graphical illustration, they assist in the identification of the different components of a process and explain the interrelationships among the various processes, as shown in Figure 2.6 below.

Though flowcharts are old design instruments, they are an important tool, especially in information systems' functioning, as related to the system's analysis and design.

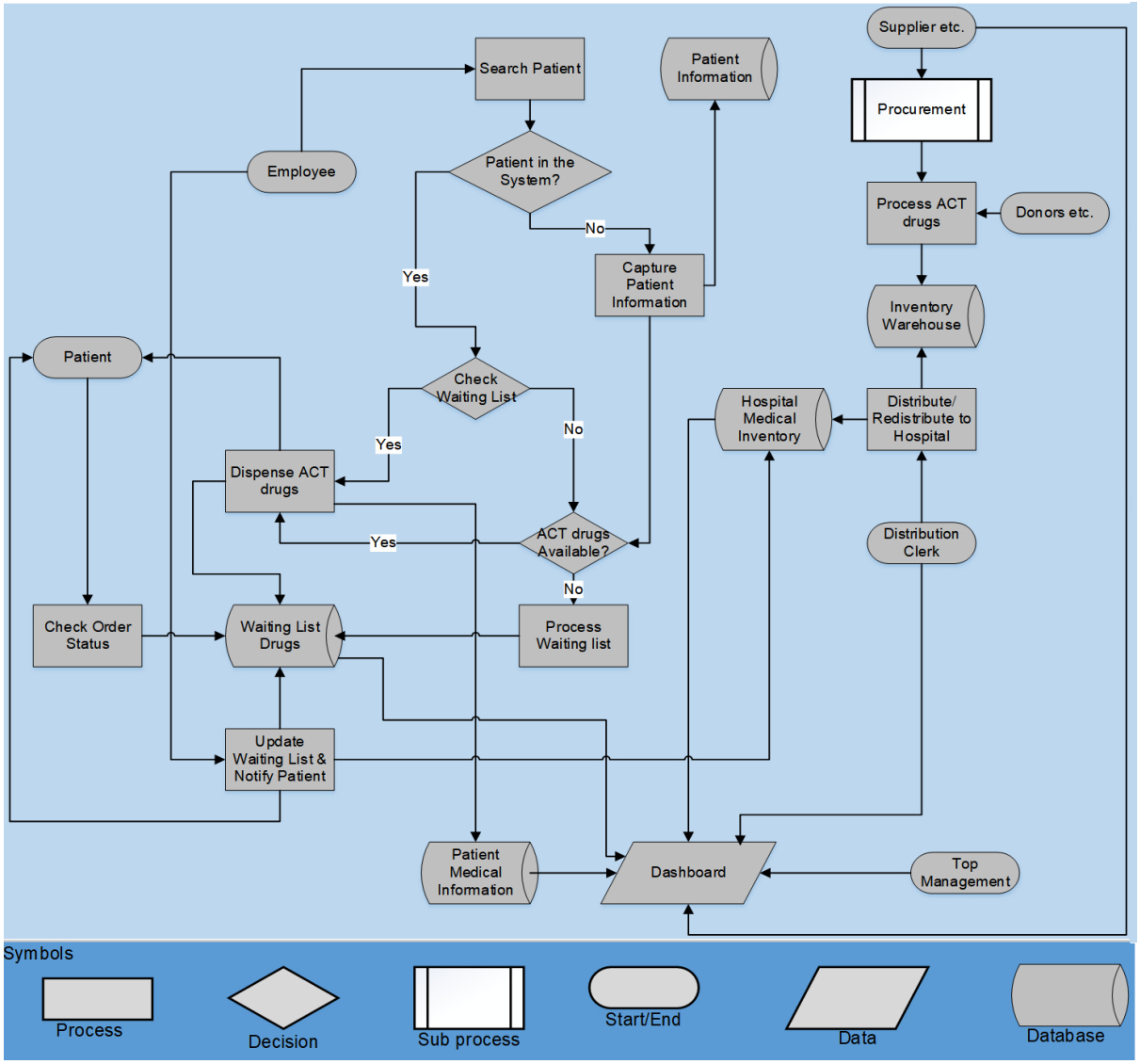


Figure 2.6: High-level flowchart of a mobile app of ACT drugs

2.4.3 Design stage

The design stage describes in detail the required specifications or features and operations that will meet the functional and non-functional requirements of the proposed system or solution (Scroggins, 2014; Rani, 2017). It is at this stage that the developer starts with the development of algorithms, system architecture, database modelling and design, and the graphical user interface design. The design stage, as relevant to this study consider both structured and object-oriented design discuss in the subsequent sections 2.4.3.1 and 2.4.3.2.

2.4.3.1 Structured design

Booch *et al.* (2008) defined structured design as a method to convert requirement specifications, including the Data Flow Diagram (DFD) into a system flowchart that can be implemented using a computer programming language like C or Pascal. Polak (2009) argued that the central theme of structured design is that the structure of the data determines the program structure. This was supported by Lakhoua (2012), who further explained that precise and accurate identification of the data structures lead to a well-structured program. Booch *et al.* (2008) explained that the system flowchart shows the relationships among various programs, sub-systems and their files or databases. Lakhoua argued that structured design is best utilised in applications that are well defined, and have a hierarchical structure of information. According to Herrmann (2001), the structured design method starts with the system specification that identifies the inputs, the desired outputs and a description of the functional aspects of the systems that transform the data.

Booch *et al.* (2008) indicated that the DFD can be mapped into the design structure by two means, namely, transform analysis and transaction analysis, as discussed below: Transform analysis is applied when the data flow in the input-output stream has clear boundaries (Abdelrahman, 2015; Booch *et al*., 2008). The DFD is mapped into a structure that allocates control to three basic modules: input, process and output. Transaction analysis is applied when a single information item causes flow to branch along one of many paths. Diatchki and Jones (2006) mentioned that the structure design uses programming like C, Pascal, ALGOL and Modula-, which are not suitable for this study.

2.4.3.2 Object-oriented design

According to Booch *et al.* (2008), object-oriented design (OOD) provides a mechanism that encompasses three important concepts in software design, namely, modularity, abstraction, and encapsulation. Mukherjee (2016) explained that OOD is basically an approach that models the problem in terms of its objects and the operations performed on them. Herrmann (2001) elaborated further by saying the conversion of requirements into a program starts at the top level by identifying the objects and classes, their relationships to other classes, and their major attributes. Their inheritance relationships then derive a class hierarchy from them. Booch *et al.* (2008)

concur by arguing that OOD starts by identifying the objects, and once the objects have been identified, the set of operations that act on the objects are examined.

Mukherjee (2016) further explained that there are basically three types of operations: those that manipulate data, those that perform computations, and those that monitor an object. Diatchki and Jones (2006) differentiated OOD from structured design with programming languages like C++, JAVA and C# (C sharp), which are suitable for this study.

2.4.4 Implementation/Build (Prototype) system stage

This stage refers to the fulfilment of the system requirements and design specifications into the actual running software component (Scroggins, 2014; Rani, 2017). At this stage, the code is written and compiled into an operational application, and the actual database is created. The implementation of a system is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project.

2.4.5 Test the system

Once the system is built, the researcher tests the logic, error handling, flow of the data, and ensures that all the defined functionalities are working as expected using the test cases. The test results are interpreted and evaluated based on the requirements of the system defined at the earlier stages.

2.4.5.1 Black box testing

With black box testing, the tester does not have any information about the internal working of the software system (Khan & Khan, 2012). Nidhra and Dondeti (2012) define black box testing as an approach where the software is tested without any knowledge of the internal structure of the program or application. Black box testing is also known as functional testing. Nidhra and Dondeti's (2012) study further explained that black box testing is done by end-users who are not concerned with the code errors or uncovering incorrect programming, they only test if the software meets the defined requirements or specifications.

Black box testing is a high-level testing that focuses on the behaviour of the software (Khan, 2011). The testing is based on external expectations, and the internal behaviour of the application is unknown. In other words, it involves testing from an

external or end-user perspective. This type of testing is ideal for users testing like Functional testing, Usability testing, Acceptance testing, and Performance testing (Khan & Khan, 2012), which is excluded in this development based study.

2.4.5.2 White box testing

White box testing uses coding experience as part of the testing procedure (Khan & Khan, 2012). Beydeda, Gruh and Stachorski (2001) mentioned that white box testing is also known as structural testing. The purpose of a white box test is to detect logical errors in the program code of a mobile application by providing appropriate input, and verifying the output against the output defined at the functional requirements' stage of the development. It is a testing approach in which the internal structure is known to the tester. This type of testing is best suited to a lower level of testing like Unit Testing, Integration testing, Memory leakage testing, Interrupt testing, and Installation testing, which is suitable for this development based study. Acharya and Pandya (2012) further explain that when a product fails, testers go deep into the code to find the root cause of the error.

2.5 CHAPTER SUMMARY

This chapter presented the main concepts that constitute the parameters of the study, from supply chain coordination theory, and integration models for digitalising supply chain coordination theory. Further, this chapter presented the justification for the theory, model and development approach that were selected as most suitable for the study. In the next chapter, the research design and methodology applied in the current study are discussed in detail.

CHAPTER 3:
RESEARCH DESIGN AND METHODOLOGY

3.1 INTRODUCTION

Chapter 2 discussed in-depth review of supply chain coordination and IoT framework integration, which is crucial for the beginning of the research process. In this chapter, the methodology that underpins the study is discussed.

The term ’research methodology’ entails more than merely the method the researcher used to gather data, or a method to discover the outcome of a particular problem related to a specific subject or topic (Alaeddini & Salekfard, 2013; Schwartz-Shea & Yanow, 2013).

In the realm of research methodology, the researcher utilises diverse principles for resolving or examining a certain research problem. Alaeddini and Salekfard (2013) describe a research method as a route to execute research. A research method presents a complete set of methods focused on the research topic (Schwartz-Shea & Yanow, 2013).

Nogueira, Romero, Espadas and Molina (2013) recommend that the research method should be selected according to the following criteria: the focus area, the field of study and the research questions. There are a whole range of research methods with diverse attributes that can be selected in terms of the research objectives. The research methodology involves the research paradigms and philosophies, and research approach, as discussed in the next section.

3.2 RESEARCH PARADIGMS AND PHILOSOPHIES

A research paradigm is a common beliefs and agreement shared between researchers (Khaldi, 2017). Kivunja and Kuyini (2017) added that the research paradigm reflects the researcher’s knowledge and understanding of the research and their intentions pertaining to the subject or topic. Makombe (2017) further states that the research paradigm constitutes the abstract assumption and principles that guide how the researcher perceives the research, and how the researcher translates and tackles the research to create knowledge. The assumption is understood as an introductory

utterance of reasoning which is based on the philosophising researcher's understanding of the research topic.

According to Maarouf (2019), a paradigm comprises four elements, namely, epistemology, ontology, methodology and axiology, as defined in the next sub-section.

Varpio and MacLeod (2020) explain that ontology enables the researcher to examine the underlying beliefs and philosophical assumptions about the information's existence and reality; epistemology establishes the truth in gathered information; axiology refers to the ethical issues that need to be considered when embarking on research.

Maarouf (2019) added that methodology refers to the research design, methods, approaches and procedures used in an investigation that is well planned to find out something. Killion and Fisher (2018) indicated that the contribution of the paradigm elements lies in the attempts to establish knowledge and understanding of the supply chain coordination framework of ACT drugs.

Philosophical stance of the current study

The aim of the research was to develop a mobile application for the supply chain coordination of ACT drugs. It was put forward that a lack of ICT, such as decision support systems, amplified the problems experienced in the multi-embedded supply chain coordination of ACTs (Nagitta & Mkansi, 2019; Stanley *et al.*, 2009). Therefore, an interpretive approach was needed to understand the social context of the required information system, namely, the supply chain coordination through which the mobile app was developed and construed by people, and through which it influenced, and was influenced by its social setting.

The supply chain coordination framework for ACTs can be viewed as a scientific body of knowledge, ascertained as truth, and which was proved by Nagitta and Mkansi (2019) and Stanley *et al.* (2009). However, the researcher in the current study adopted an ontological perspective in his views on the supply chain coordination framework for ACTs, as the human conceptualisation of the phenomenon in general hospitals in Uganda.

From this ontological perspective, therefore, the focus of this research is not the subject of the supply chain coordination framework for ACTs itself, but to understand

the existence and realities of critical logistics, micro, market, and macro dimensions, and the associated sub-dimensions of ACTs that can be converted into technological features.

The researcher's ontological perspective of the ACTs' supply chain coordination framework as a human construction, supports the epistemological stance of understanding the supply chain coordination framework through subjective and interpretative sense-making. Interpretative sense-making stems from the interpretivism worldview, which posits that there is no single version of the truth (Oates, 2005). Byrne (2002) further mentions that researchers can put forward more than one explanation and choose the option that seems more likely because of the evidence presented. This view therefore has an impact both on the way the researcher decides to obtain data pertaining to the supply chain coordination framework for ACTs and the way in which the data will be analysed in terms of how the knowledge related to the supply chain coordination framework for ACTs is brought about and how knowledge from the research is achieved.

3.3 RESEARCH APPROACHES

The researchers, Hamad, Savundranayagam, Holmes, Kinsella and Johnson (2016), mentioned that the process of research involves emerging questions and procedures, data typically collected in the participant's setting, data analysis inductively building from particulars to general themes, and the researcher making interpretations of the meaning of the data. Cvetkoska (2017) further states that the research approach includes the plan and procedures to broaden the philosophical assumptions related to detailed methods of data collection, analysis and interpretation. The selection of a research approach is based on the type of research problem or topic (Johnson & Onwuegbuzie, 2004). McCusker and Gunaydin (2015) indicated that the research approach assists in answering the research questions and how the research objectives can be achieved. There are three common research approaches, namely, qualitative, quantitative, and mixed methods research that are reviewed in the next sub-section (McCusker & Gunaydin, 2015; Hamad *et al.*, 2016; Khaldi, 2017).

3.3.1 Qualitative research

According to Khaldi (2017), qualitative research is a research approach that collects and examines non-numerical data to achieve an understanding of the problem. McCusker and Gunaydin (2015) further explain that non-numerical data can be obtained by the researcher from either open-ended questionnaires or questions, recordings, documents, artefacts, and so on. The major strength of the qualitative research approach is that it explains something which numbers alone are unable to reveal and is more flexible. Qualitative research is used when the researcher has no idea what to expect. As it is used to define the problem or develop the problem, it is outside the purview of the development phase (computer-based programming) of this project.

3.3.2 Quantitative research

Almalki (2016) defines quantitative research as an approach that tests objective theories by examining the relationship among variables. Maxwell (2019) further states that these variables, in turn, can be measured, typically on instruments, so that numbered data can be analysed using statistical procedures. The major advantage of quantitative research methods is the objectivity of the studies. Analysing strictly numerical data can allow for far less interpretation than would be present in qualitative research. This generally leads to more conclusive and objective answers to the research questions of the study.

According to Bryman (2017), a quantitative research approach deals with quantifying and analysing variables to obtain the results. The data is generated mainly by experimental and/or survey methods, although it can be generated by other research methods. The data analysis intends to look for patterns in the data that will allow the researcher to draw conclusions.

The current study selected a quantitative approach, as it involves the use of experimental methods which are elaborated on further in the next sub-section.

3.3.3 Mixed methods research

Johnson and Onwuegbuzie (2004) stated that the mixed methods research approach involved collecting both quantitative and qualitative data through inquiry, integrating the two forms of data, and using distinct designs that may involve philosophical

assumptions and theoretical frameworks. Denscombe (2008) further states that the core assumption of this form of inquiry is that the combination of qualitative and quantitative approaches provides a more complete understanding of a research problem than either approach alone could attain. The major strength of the mixed method approach is the use of qualitative data to augment the quantitative outcomes of the study, and also to clarify any contradictions between the quantitative results and qualitative findings.

The current study focused on the quantitative phase of system development which is limited to a computer laboratory.

3.4 RESEARCH STRATEGIES ASSOCIATED WITH QUANTITATIVE RESEARCH

Scandura and Williams (2000) mentioned that there are two common research strategies associated with the quantitative research approach, namely, survey and experimental techniques.

3.4.1 Surveys

A survey is the most basic instrument for quantitative research. Surveys probe pre-defined questions using different types of methods such as online surveys, questionnaires, and so forth. In computing, a survey is commonly use in the user evaluation of software systems. The survey strategy is mostly associated with the philosophical paradigm of positivism, since it looks for patterns and generalisations, but it can also be used with interpretive and critical research (Oates, 2005; Mittal, 2016).

3.4.2 Experimental techniques

Huebscher and McCann (2008) define the experimental technique as a scientific approach where one or more independent variables are manipulated and applied to one or more dependent variables to measure their effect on the latter. Experimental techniques are often associated with research in sciences and are used for testing hypotheses (Oates, 2005; Huebscher & McCann, 2008; Mittal, 2016). Feitelson (2006) referred to experimentation in computing as the practice of scientific experimental methods for the evaluation of computer systems. Basili (1996) described

experimentation in computing as part of a feedback loop for the development of aspects such as models, systems, and other elements of computer science.

For the purpose of the current study, the researcher employed computer lab experiments to test and evaluate the requirements of the system defined at the gathering stage in Section 4.1, using some predefined sets of variables to determine how well the mobile application meets its specifications or how well it performs.

3.4.3 Data-collection techniques considered in this study

There are a number of different ways to collect data, though these generally fall into four broad categories: interviews, questionnaires, literature sources and observations (Kaplan & Maxwell, 2005). Interviews and questionnaires are utilised to collate data from individuals regarding their opinions, experience and knowledge. However, the results can be subjective and unclear, and it can often be difficult to analyse the results in a quantitative manner. Literature sources involve the collection of data from already published texts available in the public domain (Choy, 2014). Mittal (2016) explained that observation is the most common technique that allows for the collection of immediately objectifiable data that can be statistically analysed.

For the purpose of the current study, following the literature review, the relevant requirements were verified with the conceptualisers of the multi-embedded coordination framework (Nagitta & Mkansi, 2019), and key stakeholders from the healthcare supply chain. Thereafter, observations were used to collect computer-based data during the testing of the mobile application system.

3.5 DATA RELIABILITY AND VALIDATION

Reliability and validity are concepts used to evaluate the quality of research. Kukul, Gökçearslan and Günbatar (2017) defined validity as the extent to which a concept is accurately measured in a quantitative study. Thatcher (2010) argues that in quantitative research, validity is the extent to which any type of testing that was performed actually tests what it was intended to test. Reliability relates to the consistency of obtaining the same results each time the instrument is used (Splett *et al.*, 2020). Korkmaz (2017) confirmed that the concepts of validity and reliability are interrelated and demonstrate different properties of the measuring instrument. The next sub-section outlines validity and reliability pertaining to software development.

3.5.1 Validity

Lee and Rine (2004) define software validation as the process of measuring system software to ensure that it meets defined functional requirements, as well as the end-users' expectations. There are three major types of validity in quantitative research namely, content, construct and criterion validity (Heale & Twycross, 2015). Schijven and Jakimowicz (2003) argue that the type of validity referred to must relate to the purpose of the concept of interest.

Content validity measures the instrument to determine if it sufficiently tested all the content that needed to be covered (Wiley, 2002). Content validity relates to this study because it measured whether the developed system covers all the pre-defined functional requirements to ensure that the developed system meets all the pre-defined users' requirements, as well as the end-users' expectations.

Construct validity ensures that the method constructed to measure matches that which needs to be measured (Wiley, 2002; Middleton, 2019). In software development, the developer constructs algorithms to produce results that meet the output of pre-defined functional requirements. In this study, the developed functionalities were tested using test cases to ensure that the output defined in Section 4.2 is the same as that generated by the system during white box testing.

Criterion validity uses two or more different instruments to measures the same variable (Wiley, 2002; Middleton, 2019). Pickard, Kitchenham and Jones (1998) explained that, in software engineering, criterion validity works well when the same system functionality is tested by different users to validate whether the functionality produces the same results. As the current study focused on the development of a mobile application, the system was tested by the developer rather than by obtaining data from participants or end-users, hence criterion validity was not suitable for this study.

3.5.2 Reliability

In computing, the word reliability means the failure-free operation of the software system in a pre-specified time and environment (Kapur, Pham, Gupta & Jha, 2017). According to the above-mentioned authors, reliability in terms of a software system is measured in two ways. Similarly, Sahu (2019) mentioned that when measuring

reliability, the first step is software testing, followed by end-user feedback which can be collected after the delivery of the software.

Furthermore, reliability can be measured by implementing three different approaches, namely, Test-Retest, Parallel Forms and Internal consistency (English & Keeley, 2014) as elaborated on below.

Test-retest reliability measures the consistency of the test results when the same test is conducted on the same sample tested at a different point in time (Clayson, 2021; Middleton, 2019). In software development, the testing can be conducted in two different environments, namely, the development environment and test environment (Sirard *et al.*, 2011; Polit, 2014; Mohajan, 2017). In this study, the first test was conducted by the developer in the development environment during the white box testing. White box testing uses coding experience as part of the testing procedure (Khan & Khan, 2012).

Parallel forms reliability measures the correlation between two equivalent versions of a test (Middleton, 2019). In software development, parallel forms reliability tests the same functionality using different parameters or attributes (Polit, 2014; Mohajan, 2017). In this study, the developer used different inputs to test the same functionality (discussed in Chapter 5) to ensure the correlation of the results.

Internal **consistency** assesses several related items that are intended to measure the same construct (Tang & Babenko, 2014). In software development, end-users measure the entire system, this includes performance, usability and so on (Master, Cheryan & Meltzoff, 2016). The end-user feedback then analyses the data to determine system reliability. For the purpose of study, the internal consistency of the system was tested to ensure the error-free operation of the entire system.

3.6 ETHICAL CONSIDERATIONS

Resnik (2015) defines research ethics as the moral basis that controls or governs how researchers should execute research work. Adolph, Hall and Kruchten (2011) explain that moral concepts guide researchers to perform and report research work without deception or intention to harm the participants of the study or members of the society as a whole, whether knowingly or unknowingly.

Research ethics is involved with the following areas, namely, integrity, openness, respect for intellectual property, confidentiality and so on (Resnik, 2015; Hammersley, 2015; Aydemir & Dalpiaz, 2018). Research ethics promotes the aims of research, such as expanding knowledge. This study was computer-laboratory based, and did not involve participants post the development of the application. However, four people (the two authors of the supply chain coordination framework, the Chief Pharmacist and District Medicines Management Supervisor) were consulted to validate the requirement's analysis following the first phase of the waterfall model.

In line with the University of South Africa's ethical policy, consent was obtained from all the stakeholders who validated functional requirements in this study. The relevant ethical clearance that was required to guarantee the ethical principles of this research was obtained from the UNISA-CAES Health Research Ethics Committee of the University of South Africa (see Appendix A).

3.7 LIMITATION AND RESTRICTIONS

Even though ICT has the potential to enhance the healthcare system to a large extent, emerging countries are far from harvesting these benefits because of certain limitations (Keengwe, Onchwari & Wachira, 2008). The implementation of IoT is much more complex because of some major common restrictions and limitations, such as network infrastructures, and so on. The current study is limited to computer-based observation, and did not solicit the views of potential users due to the limited time allocated to a master's project.

3.8 CHAPTER SUMMARY

This chapter discussed the research design and methodology adopted by the current study. The chapter first discussed the research paradigms and philosophies, followed by a discussion of the three common research approaches, namely, quantitative, qualitative and mixed methods, and then discussed the research approach that was adopted by the current study. Lastly, the ethical considerations of the study were discussed. In the next chapter, system design and implementation are discussed in detail.

CHAPTER 4:
SYSTEM DESIGN AND IMPLEMENTATION

This chapter builds on the methods articulated in chapter 3 by introducing the system design and implementation process followed in the development of the ACT mobile app project. The system design and implementation are grounded on the waterfall development model depicted in Section 2.3.1 that was adapted into Figure 4.1 (on the next page).

Waterfall system development has five stages, namely, requirements' gathering, analysis, design, implementation, testing and evaluation, as briefly explained below:

- According to Aysolmaz and Demirors (2014), the first stage, user requirements' gathering, entails a description of what the finished product will look like in the eyes of the user.
- In the second stage, the analysis stage, the user requirements are analysed and functionalities are converted into the defined system functions that the researcher intends to develop (Balaji & Murugaiyan, 2012).
- According to Ragunath *et al.* (2010), the third stage, the design stage, provides a detailed description of the necessary specifications, features and operations that will satisfy the functional requirements of the proposed system.
- Mahalakshmi and Sundararajan (2013) mentioned that in the fourth stage, the implementation stage, the system architecture and database diagrams are designed. The development includes pseudo-code, screen layouts, and the database management system.
- In the testing stage, all the pieces of code are integrated and deployed in the testing environment.

Figure 4.1 provides a graphical illustration of the waterfall model and the outputs of each stage.

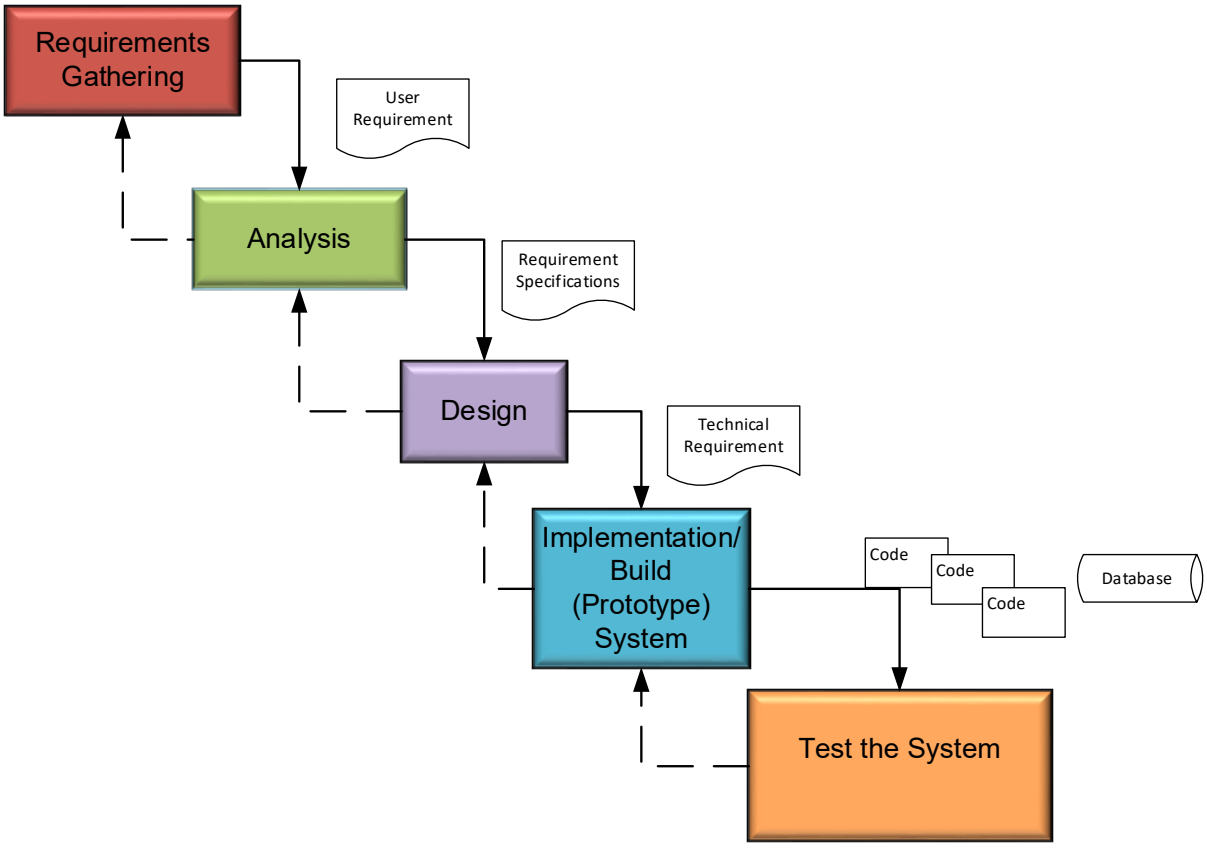


Figure 4.1: Waterfall model with output

Source: Adapted from Ragunath *et al.* (2010)

Each of the stages of waterfall development as it pertains to this study, is discussed in the sections below.

4.1 REQUIREMENTS' GATHERING STAGE

The first stage of the waterfall model shown in Figure 4.1 above is the requirements' gathering stage. In gathering the requirements, the study followed Bormane and Bērziša's (2017) *Business Analysis Book of Knowledge* (BABOK) that offers a road map that can be used for stakeholder engagement towards ensuring that stakeholders agree on the requirements, system architecture and design for satisfactory system implementation. Tătaru and Fleacă (2019) further explain that researchers should elicit requirements from key stakeholders, and then transforms their requirements into functional tasks.

The requirements gathering for this study started with a literature review of the multi-embedded coordination framework (Nagitta & Mkansi, 2019), after which the requirements were verified with key stakeholders from the healthcare supply chain as part of the elicitation process.

According to Mathiesen, Bandara, Delavari, Harmon and Brennan (2011), stakeholder identification involves identifying the stakeholders (who will be directly or indirectly impacted by the change) and their characteristics, as well as analysing the information, once collected. The first stakeholders for this study were Nagitta and Mkansi (2019), the authors of the supply chain coordination framework for malaria drugs who detailed the requirements for the system, including their interpretation of the key stakeholders across the macro, market and micro environments. Following the consultation with Nagitta and Mkansi (2019), the study approached the Chief Pharmacist and District Medicines Management Supervisor involved in the healthcare supply chain of ACT products in Uganda. Figure 4.2 illustrates the requirements' gathering stage, as implemented in the current study.

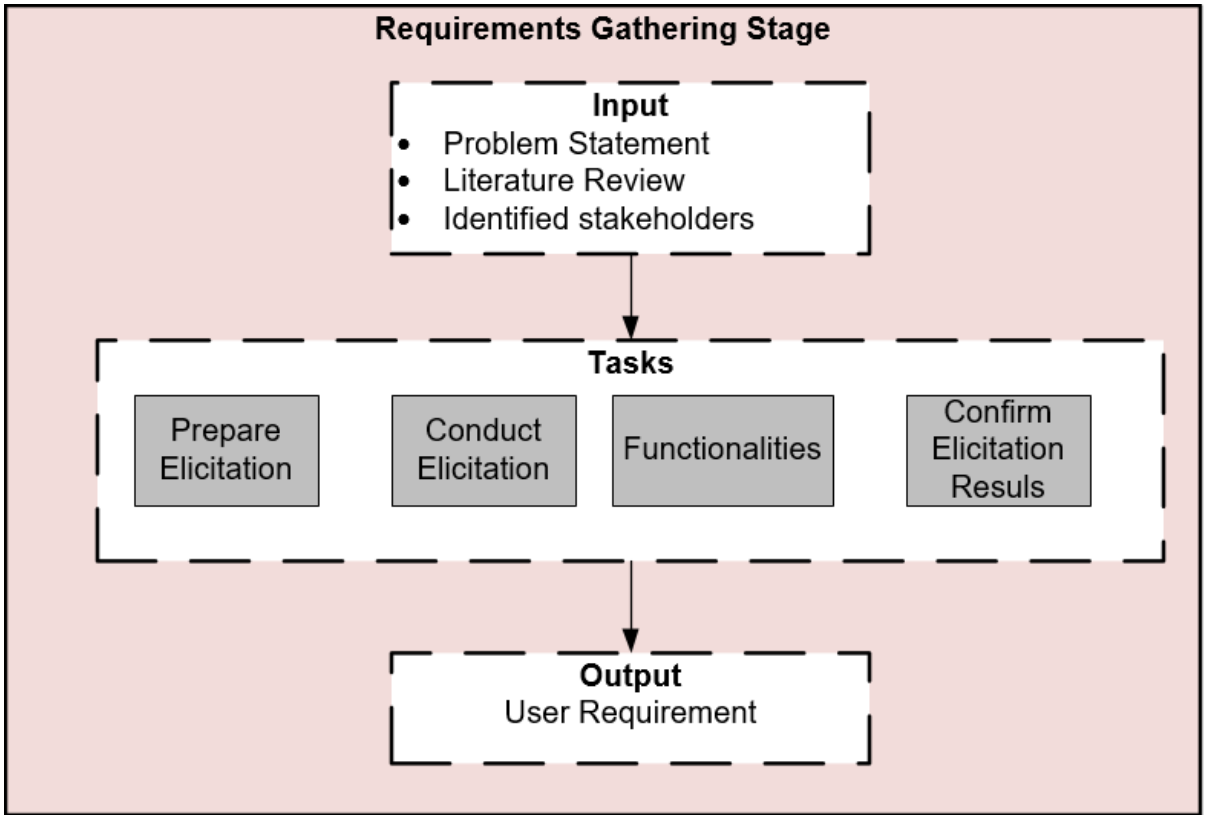


Figure 4.2: Requirements' gathering stage

Source: Researcher's own compilation

The elicitation process is detailed in the next section, followed by how the study translated the requirements into functional tasks.

4.1.1 Elicitation

Stakeholder engagement followed the elicitation process described by Ferraris and Fernandez-Gago (2020). According to Ferraris and Fernandez-Gago (2020), elicitation is the generation or getting of information from the identified stakeholders or other sources. The information can be collected by speaking with stakeholders directly through interviews, researching topics, observing or experimenting, or simply being handed information (Saad & Dawson, 2018).

According to Jonasson (2016), elicitation is composed of the following tasks: prepare for elicitation, conduct elicitation, document functionalities and confirm functionalities (Naeem *et al.*, 2017). Below is how the study followed the different aspects of elicitation.

4.1.1.1 Prepare for elicitation

Preparing for elicitations involves ensuring that the stakeholders have the information they need to provide, and that they understand the nature of the activities they are going to perform (Jonasson, 2016). For the purpose of the current study, preparing for elicitation required of the researcher to arrange virtual meetings with stakeholders and to send the related documents before the meeting. The researcher in the current study sent an invitation to all the identified stakeholders for a virtual meeting.

4.1.1.2 Conduct elicitation

Jonasson (2016) explained that the conduct elicitation step describes the work performed to understand stakeholder needs and to identify potential solutions that may meet those needs. The data was compiled from the literature review and populated into a template. The stakeholders were taken through the template and allowed to review and update where necessary. Following the document review and interpretation, the study translated each requirement into functional tasks as discussed below.

1. Functionality

Kurtanović and Maalej (2017) define functionality as that which the system should do, in other words, a functionality will describe a particular behaviour or function of the

system when certain conditions are met. Sharma (2017) explains that documenting the functionality map involves significant stakeholder conversations, combined with the rigorous principles of that which constitutes a functionality.

For the purpose of the current study, input from stakeholder representatives was critical to ensure clear descriptions of the micro, macro, and market dimensions. In addition, it was equally critical to specify the organisational elements, roles, technologies, and such, to achieve the outcome of the supply chain of ACT drugs.

In this study, the process of converting requirements into functional tasks started by capturing and documenting all the functionalities that represent the full scope of what the micro, macro and market dimension did at that stage (irrespective of how well it did it) or what it desired to be able to do in the future. All the identified functionalities were documented using the same table format with the following principles, namely, trigger, purpose, input, process, output, and business rules mentioned above. The micro, macro, logistics and market dimensions as related to the requirements' gathering stage of the current study are discussed below.

A. Micro

Following a confirmatory factor analysis and multi-decision criteria, the study by Nagitta and Mkansi (2019) revealed the critical supply chain coordination sub-dimension factors emanating from the micro environment that required technological attention.

The researcher in the current study, in consultation with the relevant stakeholders, confirmed the micro sub-dimensions that would be addressed and achieved by technology, such as ensuring feedback on stock status (see Table 4.1, column 4).

The last column of Table 4.1 highlights the conceptual functionality of such a sub-dimension. The second step of converting requirements into functional tasks involved organising the information logically. Hence, the next section offers the interpretation of Table 4.1 following the trigger, purpose, input, process, output, and business rules principle.

Table 4-1: Micro dimensions

Micro			
#	**CFA Results**	**Sub-dimensions**	**Functionalities**
1.1	Top management	▪ Frequent feedback on stock status ▪ Redistribution ▪ Provision of transport	1.1.1 Demand Report 1.1.2 Stock Status Report 1.1.3 Active Orders Report
1.2	Mutual understanding	▪ Communicating of policy change to patients ▪ Clear instruction to use ACTs	1.2.1 Instructions on use of ACTs 1.2.2. SMS notifications to patients
1.3	Information Sharing	▪ Scheduled issuance by stores ▪ Internal redistribution	1.3.1 Information on stock status Report (Same as 1.1.2.) 1.3.2 Notice boards (E-mail, SMS)
1.4	Relationship management	▪ Joint decision-making ▪ Feedback loop	1.4.1 Feedback loop with other health facilities including suppliers
1.5	Responsiveness	▪ Internal redistribution ▪ Efficient delivery by pharmacy	1.5.1 Scheduled issuance 1.5.2 Supplier schedule 1.5.3 Internal transfers of ACT 1.5.4 Placement of emergency orders 1.5.5 Internal redistribution
1.6	Organisation dimensions	▪ Accountability for ACTs ▪ Issuance of local guidelines	1.6.1 Centralised system

Source: Researcher's own compilation

The functionalities, as listed in Table 4.1 are discussed below.

Demand Report Functionality

The purpose of this function is to assist top management in making provision for, or supporting the redistribution of ACT drugs.

Table 4.2 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-2: Demand Report functionality

Dimension: Micro			
CFA Results	1.1 Top Management	**Trigger**	3.2.1. Consumption Report
Functionality	1.1.1 Demand Report	**Users**	Top Management
Input/Action	**Process**		**Output/Response**
▪ Number of patients need ACT ▪ Malaria seasons ▪ Peak times	Generate demand report		Demand Report
Business Rules	Monthly, weekly and quarterly sharing of reports		
Security	Top management		

Stock Status Report Functionality

The purpose of the Stock Status Report functionality is to inform top management and other related stakeholders which health facilities are at risk of stock-out of ACTs.

Table 4.3 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-3: Stock Status Report functionality

Dimension: Micro			
CFA Results	1.1 Top Management	**Trigger**	3.5.7 Dispense ACTs
Functionality	1.1.2 Stock Status Report	**Users**	Top Management, Political, etc.
Input/Action	**Process**		**Output/Response**
▪ Number of ACTs dispensed	Generate stock status report		Daily, weekly, monthly, year, and season stock status reports
Business rules	Monthly, weekly and quarterly sharing of reports		
Security	Top management		

Active Orders Report Functionality

The purpose of the Active Orders Report functionality is to support and prioritise emergency orders based on the demand report. Table 4.4 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-4: Active Orders Report functionality

Dimension: Micro			
CFA Results	1.1 Top Management	**Trigger**	1.1.1 Demand Report
Functionality	1.2.2 Active Orders Report	**Users**	Top Management, Political, etc.
Input/Action	**Process**		**Output/Response**
▪ Demand Results	Process Order		Order report
Business Rules	Monthly, weekly and quarterly sharing of reports		
Security	Top management		

Instructions on use of ACTs

The purpose of Instructions on use of ACTs functionality is to ensure that patients are given the right treatment. Table 4.5 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-5: Instruction for use of ACT functionality

Dimension: Micro			
CFA Results	1.2 Mutual understanding	**Trigger**	1.1.1 Health work
Functionality	1.2.2 Instructions on use of ACTs	**Users**	Top Management, Political, etc.
Input/Action	**Process**		**Output/Response**
▪ Dosage information	Display the instruction for use		Direction or Instruction for use
Business Rules	Clear information of ACT's uses		
Security	Top management		

SMS/Email notifications to patients' functionality

The purpose of SMS/Email notifications to patient's functionality is to remind patients on how to use the medication.

Table 4.6 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-6: SMS notifications to patients' functionality

Dimension: Micro			
CFA Results	1.2 Mutual understanding	**Trigger**	3.5.7 Dispense ACTs
Functionality	1.2.2 SMS/Email notifications to patients	**Users**	Patients, etc.
Input/Action	**Process**		**Output/Response**
▪ Patient Cell phone Number	Send SMS with the instruction for use to patient		SMS with Instruction for use
Business Rules	Appropriate information needs to be sent to clients		
Security	Pharmacist, Dispenser and health worker		

Notice boards (E-mail, SMS) functionality

The purpose of the Notice boards (E-mail, SMS) functionality is to ensure that patients are given the right treatment. Table 4.7 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-7: Notice boards (E-mail, SMS) functionality

Dimension: Micro			
CFA Results	1.3 Information Sharing	**Trigger**	Running out of stock or increase of demand
Functionality	1.2.2 Notice boards (E-mail, SMS)	**Users**	Employees, Management etc.
Input/Action	**Process**		**Output/Response**
▪ Stock status	Send stock status notice to stakeholders		Stock status notice via E-mail or SMS
Business Rules	Sharing of information on the stock status in real time		
Security	Top management		

Feedback loop functionality

The purpose of the Feedback loop functionality is to conduct surveys with regard to the delivery of the redistribution of ACTs.

Table 4.8 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-8: Feedback loop functionality

Dimension: Micro			
CFA Results	1.4 Joint decision making & relationships	**Trigger**	Acknowledgement of delivery of ACT
Functionality	1.4.1 Feedback loop with other health facilities, including suppliers	**Users**	Employees
Input/Action	**Process**		**Output/Response**
▪ Questionnaire	Rate the service delivery or redistribution of ACT		Delivery or redistribution scoring
Business Rules	Upon receiving the stock, health facility must provide feedback		
Security	Top management		

Scheduled issuance functionality

The purpose of the Scheduled issuance functionality is to schedule drugs in preparation for delivery. Table 4.9 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-9: Scheduled issuance functionality

Dimension: Micro			
CFA Results	1.5 Responsiveness	**Trigger**	Supplier or donors confirmed the delivery
Functionality	1.5.1 Scheduled issuance	**Users**	Supplier, Employees, Donors
Input/Action	**Process**		**Output/Response**
▪ Schedule plan	To display schedule plan in the various departments		Schedule plan displayed
Business Rules	Schedules need to be shared promptly		
Security	Supplier and Top management		

Supplier schedule functionality

The purpose of the Supplier schedule functionality is to have a visible record of the schedule from the supplier and to help Health facilities to align their orders with the supplier's schedule. Table 4.10 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-10: Supplier schedule functionality

Dimension: Micro			
CFA Results	1.5 Responsiveness	**Trigger**	Placing of order to the supplier, Donors
Functionality	1.5.3 Supplier schedule	**Users**	Ordering Team
Purpose	To have visibility of schedule from the supplier and to help health facilities to align their orders with the supplier's schedule		

Input/Action	Process	Output/Response
▪ Dates	Generate supplier schedule	Supplier schedule report
Business Rules	Follows the supplier's schedule when placing orders	
Security	Authorised Personnel (Pharmacist, Supplier, Dispenser)	

Internal transfers of ACT functionality

The purpose of the Internal transfers of ACT functionality is to have medicines redistributed from facilities with overstock to health facilities that have low stock. Table 4.11 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-11: Internal transfers of ACTs

Dimension: Micro			
CFA Results	1.5 Responsiveness	**Trigger**	Redistribution of medicines
Functionality	1.5.4 Internal transfers of ACT	**Users**	Employees, Top management

Input/Action	Process	Output/Response
▪ Increases demand for ACTs	Process redistribution document	Report on ACTs redistributed
Business Rules	Following stock redistribution guideline	
Security	Authorised personnel	

Internal redistribution between Health Facilities functionality

The purpose of the Internal redistribution between Health Facilities functionality is to transfer the stock from health facilities with stock to another health facility with low stock, or when demand has increased. Table 4.12 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-12: Internal redistribution between Health Facilities

Dimension: Micro			
CFA Results	1.5 Responsiveness	**Trigger**	Redistribution
Functionality	1.5.5 Internal redistribution between Health Facilities	**Users**	Top management, Employees

Input/Action	Process	Output/Response
▪ Health facility with stock ▪ Health facility without stock ▪ Number of required ACTs	Process redistribution	The number of ACTs transferred from one Health facility to another
Business Rules	Following stock redistribution guidelines	
Security	Authorised personnel	

The macro dimensions, as related to the requirements' gathering stage of the current study are discussed below.

B. Macro

Following a confirmatory factor analysis and multi-decision criteria, the study by Nagitta and Mkansi (2019) revealed the critical supply chain coordination sub-dimension factors from the macro environment that required technological attention.

In consultation with the relevant stakeholders, the researcher in the current study confirmed the sub-dimensions that needed to be addressed and achieved by technology. Hence, the section below offers the interpretation of Table 4.13, following the trigger, purpose, input, process, output, and business rules principle.

Table 4-13: Macro dimensions

Macro			
#	**CFA Results**	**Sub-dimensions**	**Functionalities**
2.1	Legal	▪ Clinical guidelines ▪ Testing policy	1.1.1 Compliance with dosage (Business rule on the functionality)
2.2	Social – Cultural	▪ Compliance with dosage ▪ Social clicks	2.2.1 Dosage information (SMS/email)
2.3	Economic	▪ Cost sharing ▪ Donor funds	2.3.1 None
2.4	Technological	▪ Use of Rapid Diagnostic Tests (RDT)s ▪ M-track	2.4.1 Integrated mobile application (part of the non-functional requirement)

The functionalities, as listed in Table 4.13, are discussed below.

Dosage Information (SMS/Email) functionalities

Following the translation of each macro requirement into a functional task, the researcher sought further details about each macro functionality identified above in Table 4.13 using the input, trigger, process, output, and business principles. Table 4.14 describes the functional task of the Dosage information (SMS/email) functionality.

The purpose of the Dosage information (SMS/Email) functionality is to send an SMS/email informing the patient on how to use the medicine. Table 4.14 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-14: Dosage information (SMS/email)

Dimension: Macro			
CFA Results	3.1 Legal	**Trigger**	Dispense ACT
Functionality	2.1.1 Dosage information (SMS/email)	**Users**	Patients
Input/Action	**Process**		**Output/Response**
▪ Patient information	Send SMS or email after dispensing medicine		SMS or email to patient
Business Rules	Strengthening health education talk on medicines use		
Security	Authorised staff to dispense		

C. Logistics

Following a confirmatory factor analysis and multi-decision criteria, the study by Nagitta and Mkansi (2019) revealed the critical supply chain coordination sub-dimension factors emanating from the logistics environment (see Table 4.15 below) that called for technological attention. In consultation with the relevant stakeholders, the researcher in the current study confirmed the sub-dimensions that would be addressed and achieved by technology, namely: Quantification, procurement and ordering, forecasting, and storage management.

Table 4-15: Logistics functionalities

Logistics			
#	**CFA Results**	**Sub-dimensions**	**Functionalities**
3.1	Quantification	▪ Consideration of malaria seasons ▪ Information from dispensing logs ▪ Maximum-minimum levels and monthly consumption	3.1.1 Monthly consumption 3.1.2 Maximum-minimum stock levels 3.1.3 Information from the dispensing logs 3.1.4 Malaria seasons based on History Report 3.1.5 Peak times based on History Report
3.2	Forecasting	▪ Stock cards ▪ Disease patterns ▪ Estimating monthly consumption	3.2.1 Consumption Report 3.2.2 Estimating the average monthly consumption
3.3	Store management	▪ Labelling ▪ Verification of expiry dates ▪ Updated stock cards ▪ Monitoring temperatures ▪ Medicine registers	3.3.1 Receive order or donation 3.3.2 Register drug
3.4	Procurement	▪ Needs identification ▪ Developing procurement plans ▪ Approved budgets ▪ Adherence to delivery schedules and; ▪ Requisitioning per plan	3.4.1 Order drug
3.5	Dispensing	▪ Prior testing of blood	3.5.1 Dispense ACT

Following the translation of each Logistics requirements into a functional task, the researcher sought further details of each macro functionality identified above in Table 4.15 using the input, trigger, process, output, and business principles as presented in the sub-sections below.

Maximum–minimum stock levels functionality

The purpose of the Maximum–minimum stock levels functionality is to report on the maximum and minimum stock levels from each health facility, including the warehouse, and to create visibility to ease the process of redistribution. Table 4.16 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-16: Maximum–minimum stock levels

Dimension: Logistics			
CFA Results	3.1 Quantification	**Trigger**	Consumption report
Functionality	3.1.2 Maximum–minimum stock levels	**Users**	Top management, store assistants
Input/Action	**Process**		**Output/Response**
▪ Stock status report	Process real time maximum, minimum stock levels for all health facilities		Dashboard of stock level for all health facilities
Business Rules	To work within the redistribution guidelines		
Security	MS, Top management, Pharmacist, DMMS		

Information from the dispensing logs functionality

The purpose of the Information from the dispensing logs functionality is to observe and measure the consumptions.

Table 4.17 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-17: Information from the dispensing logs

Dimension: Logistics			
CFA Results	3.1 Quantification	**Trigger**	Procurement planning
Functionality	3.1.3 Information from the dispensing logs	**Users**	Logistics team
Input/Action	**Process**		**Output/Response**
▪ Demand Report	Generate monthly consumption data from the dispensing log		Consumption Report
Business Rules	Use of consumption data, morbidity data and disease surveillance reports		
Security	Top management		

Malaria seasons based on History Report functionality

The purpose of the Malaria seasons based on History Report functionality is to assist in planning and to be proactive for the next season. Table 4.18 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-18: Malaria seasons based on History Report

Dimension: Logistics			
CFA Results	3.1 Quantification	**Trigger**	Planning and ordering
Functionality	3.1.4 Malaria seasons based on History Report	**Users**	Top management and logistics team
Input/Action	**Process**		**Output/Response**
▪ Select season	Generate Malaria season Report		Malaria seasonal report
Business Rules	Use of consumption data, morbidity data and disease surveillance reports		
Security	Top management		

Peak times based on History functionality

The purpose of the Peak times based on History Report functionality is assist in planning for the peak times within the season. Table 4.19 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-19: Peak times based on History Report

Dimension: Logistics			
CFA Results	3.1 Quantification	**Trigger**	Ordering more ACTs
Functionality	3.1.5 Peak times based on History Report	**Users**	Logistic Team
Input/Action	**Process**		**Output/Response**
▪ Select season	Generate the Peak time Report		Peak time Report
Business Rules	Orders should done base on consumption data and disease prevalence		
Security	Top management, logistic Team and DHO		

Receive drugs (Labelling and barcode functionality)

The Labelling and barcode functionality is optional in cases where the ACTs do not have a barcode that can be read by a smartphone. The system must be able to generate a barcode or QR code that is compatible with any smartphone device. The label will identify the destination of the product. Table 4.20 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-20: Labelling and barcodes

Dimension: Logistics			
CFA Results	3.3 Store management	**Trigger**	Labelling and registration of medicines
Functionality	3.3.1 Receive order or donation	**Users**	Store assistants
Input/Action	**Process**		**Output/Response**
▪ Identification number ▪ Barcode or QR code	Generate barcode or QR code Generate the label to identify destination		Barcode or QR code Health facility to receive the ACTs.
Business Rules	Enforcing logistics management policy All drugs that come to the NMS must conform to the standard and guidelines of the National Department of Health and World Health Organization		
Security	Top management (MS, Pharmacist, DMMS and Dispenser)		

Register drug functionality

The purpose of the Register drug functionality is to capture the medicine into the Health database and to assign it to the Health facility. Table 4.21 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-21: Register drug

Dimension: Logistics			
CFA Results	3.3 Store management	**Trigger**	Labelling and registration of medicines
Functionality	3.3.3 Register drug	**Users**	Store assistants, Pharmacist, Dispenser
Input/Action	**Process**		**Output/Response**
▪ Barcode or QR code	Assign the code Allocate medicine to the Health facility		Medicine code Medicine assigned to the Health facility
Business Rules	All the drugs that go to Health Facilities must have been registered into the system, either by store manager at the NMS or by the Supplier.		
Security	Authorised distribution by specific personnel		

Order drug functionality

The purpose of the Order drug functionality is to order drugs either from the distribution centre or supplier. Table 4.22 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-22: Order drug

Dimension: Logistics			
CFA Results	4.1 Procurement	**Trigger**	Ordering of medicines
Functionality	3.3.1 Order drug	**Users**	Top management and logistics team
Input/Action	**Process**		**Output/Response**
▪ Medicine code ▪ Quantity	Process the order Process quantity needed		Order Number Number supplied
Business Rules	Order Medicine Ordering based on approved budget Delivery lead-times No order must be sent directly to Health Facility from the Supplier		
Security	Top management		

Dispense ACT functionality

The purpose of the Dispense ACT functionality is to hand over medicine to the patient. Table 4.23 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-23: Dispense ACT

Dimension: Logistics			
CFA Results	3.5 Dispensing	**Trigger**	Appropriate dispensing of medication according to clinical needs
Functionality	3.5.1 Dispense ACT	**Users**	Patients
Purpose	To hand over medicine to the patient		
Input/Action	**Process**		**Output/Response**
▪ Patient Number	Process Dispense ACT		Patient received ACT
Business Rules	The drug must be dispensed to only patients in the system.		
Security	Only Pharmacist must have access to the Dispense functionality		

The market dimensions, as related to the requirements' gathering stage of the current study, are discussed below.

D. Market

Following a confirmatory factor analysis and multi-decision criteria, the study by Nagitta and Mkansi (2019) revealed the critical supply chain coordination sub-dimension factors from the market environment that called for technological attention.

In consultation with the relevant stakeholders, the researcher in the current study confirmed the sub-dimensions that would be addressed and achieved by technology, namely: Visibility of information to all regions and Online sharing of information. (See Table 4.24 below).

Table 4-24: Market dimensions

Market			
#	**CFA Results**	**Sub-dimensions**	**Functionalities/Technology**
4.1	Supply chain interdependence (SCI)	▪ Regular regional monitoring ▪ Use of personal phones	4.1.1 Visibility of information to all regions
4.2	Information sharing with donors & MoH (ISS)	▪ Quarterly meetings ▪ Sharing of weekly or quarterly reports	4.2.1 Online sharing of information

Following the translation of each market requirement into a functional task, the researcher sought further details of each macro functionality identified above in Table 4.24 using the input, trigger, process, output, and business principles as presented in the sub-sections below. The following section provides more detail of each market functionality identified above in Table 4.24.

Visibility of information to all regions functionality

The purpose of the Visibility of information to all regions functionality is to view the real time consumption of ACTs. Table 4.25 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-25: Visibility information to all regions

Dimension: Market			
CFA Results	4.1 Supply chain interdependence (SCI)	**Trigger**	Sharing of information on stock status
Functionality	4.4.1 Visibility of information to all regions	**Users**	Top management, Political leaders, Logistics team
Input/Action	**Process**		**Output/Response**
▪ Select stock status	Activate real-time dashboard		Dashboard for ACT consumptions
Business Rules	Authorised personnel to access the Dashboard for ACT consumptions		
Security	Top management MS, Pharmacist,		

Online sharing of information functionality

The purpose of the Online sharing of information functionality is to disseminate information regarding ACTs across health facilities to MoH, donors, and so on. Table 4.26 below outlines the functionality in detail and also maps it with the CFA results.

Table 4-26: Online sharing of information

Dimension: Market			
CFA Results	4.2 Information sharing with donors & MoH (ISS)	**Trigger**	Timely planning,
Functionality	4.2.1 Online sharing of information	**Users**	MoH, donors, DHTs and Facility in-charges
Input/Action	**Process**		**Output/Response**
▪ Demand report	Generating reports on ACT stock status, consumptions		Availability of reports
Business Rules	Timely sharing of information with the key stakeholders		
Security	Top management		

The last stage of the requirements' gathering process was to confirm the functionalities with all the stakeholders. Jonasson (2016) explained that the confirmation of functionalities step involves ensuring that stakeholders have a shared understanding of the outcomes of elicitation, that the requirements were appropriately interpreted and recorded, and that the information that was received was compared with other information to look for inconsistencies or gaps.

The requirements that were gathered across the different levels of the coordination framework from the micro, macro, logistics, and market environment were confirmed by all stakeholders to be appropriately devolved into functional tasks. Sharma (2017) further added that this task also involves comparing the information received with other information to look for inconsistencies or gaps.

Most importantly, the functional tasks produced from the requirements' gathering stage served as inputs into the analysis stage of the waterfall development method. It was necessary to model the requirements and associated functional tasks into flowcharts and use cases as discussed in the next section.

4.2 ANALYSIS STAGE

The second stage of the waterfall model (Figure 4.1) is the analysis stage. Hailes (2014) defines requirements analysis as the task that structures and organises the requirements collected during the elicitation activities in the form of models. According to Mylopoulos, Chung and Yu (1999), the model is a descriptive and visual way to convey information to a specific audience to support the analysis, communication and understanding, for example, through the use of a flowchart. In addition, use case models may also be used to confirm knowledge, to identify gaps that the researcher may have, and to identify duplicate information.

For the purpose of the current study, the analysis stage produced flowcharts and use case models. Flowcharts explain how functionalities are integrated and interconnected to each other, whereas use cases are used to group functionalities to create a sub-system.

Figure 4.3 below depicts how the user requirements inputs from the requirements' gathering stage were analysed into models such as flowcharts and use cases. The output becomes a technical specification that can be understood by a technical person, such as a developer.

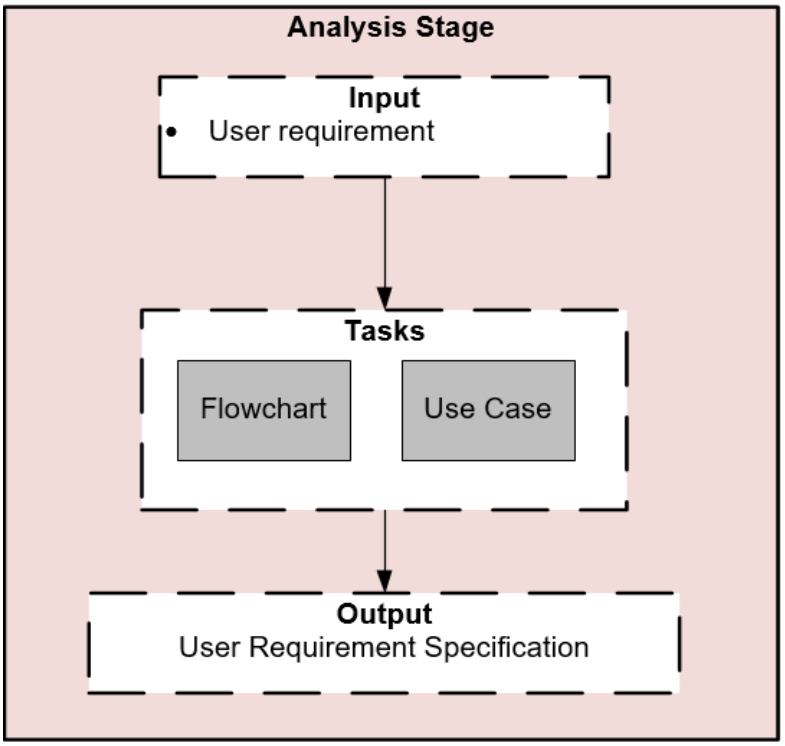


Figure 4.3: Analysis stage

Source: Researcher's own compilation

Flowcharts and use cases were utilised to demonstrate a better understanding and communicate the analysis results, as illustrated in Figures 4.4 to 4.8 below.

4.2.1 Flowchart

According to Khan and Khan (2011), a flowchart is a graphical depiction of a process or system that describes the sequencing of actions needed to generate output.

Notes: In Figure 4.4 below, number 3.3.1 ‘Receive Order or Donation’ (Labelling and Barcode) and 3.3.2 ‘Register drug’ correspond with the numbers in Table 4.15. Number 1.5.1 ‘Schedule issuance’ corresponds with the numbers in Table 4.1. In addition, number 1.5.1 ‘Schedule issuance’ is divided into two steps, namely, Step 3A and 3B to address a normal order and an emergency order, respectively.

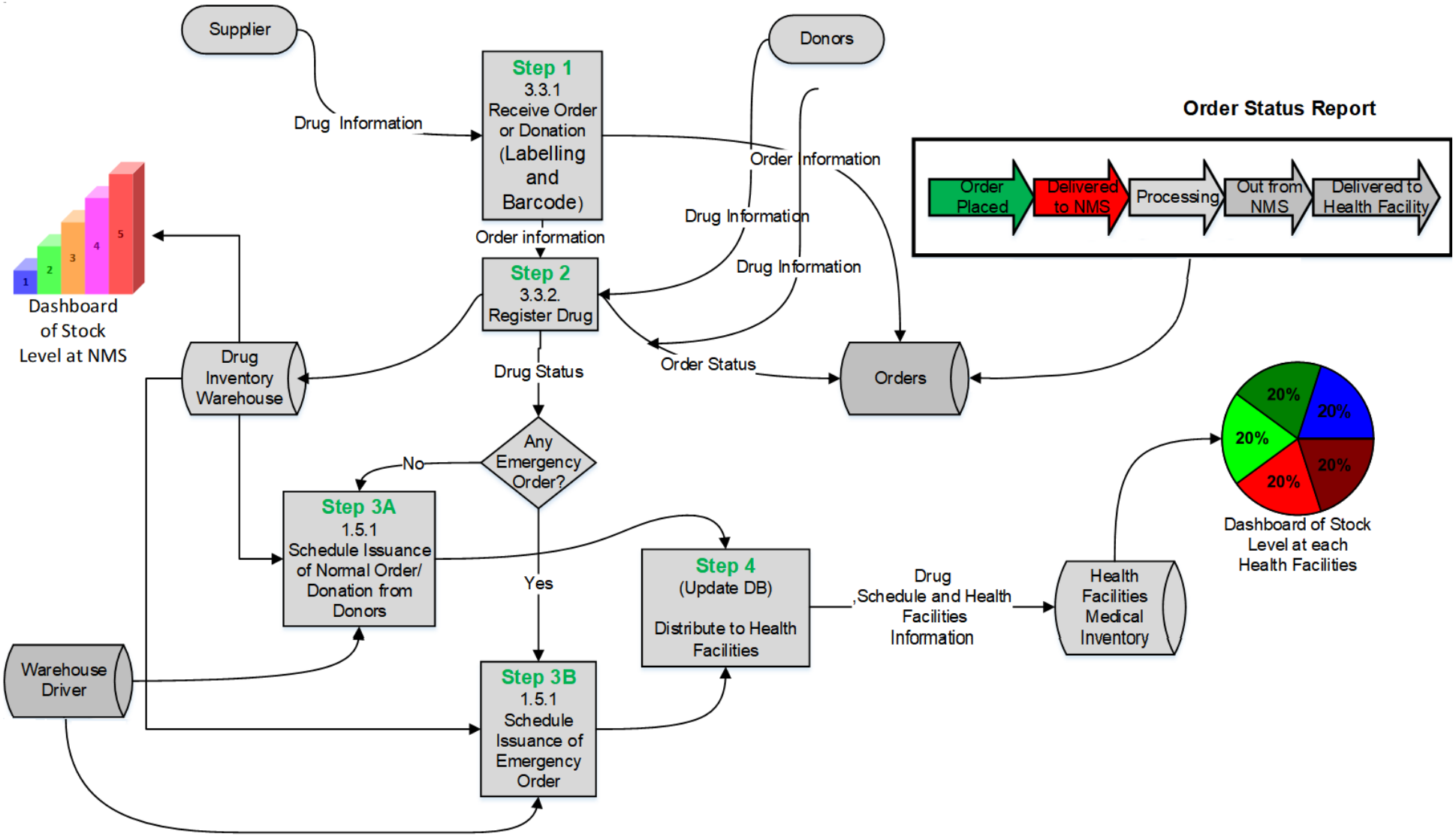


Figure 4.4: Receiving and dispatch process flow

Figure 4.4 represents a process flow of the micro-functional tasks that deal with receiving and dispatch gathered in the first stage of the waterfall development methodology and it corresponds with the numbers in dimension Table 4.1.

According to the Business Rule listed in Table 4.20 ‘Receive drugs’ (Labelling and barcode functionality), all drugs that come to the NMS must conform to the standard and guidelines of the National Department of Health and World Health Organization, therefore, when the NMS receives the order or donation it must first be checked before it can be registered into the system at Step 1 in Figure 4.4.

As mentioned in the logistics section above, procurement refers to the activities related to the acquisition of products from supplies. According to the Business Rule listed in Table 4.22 ‘Order Drug’ functionality, no order can be sent directly to the health facility from the supplier or donor, prior to verification and registration at the NMS. With reference to Figure 4.4 above, Step 2 can only be processed at the NMS.

Step 2 ‘Register Drug’: According to the Business Rule in Table 4.21 ‘Register drug’ functionality, all the drugs must first be registered at the NMS before being distributed to health facilities. Once a drug has been registered in the system, it will reflect on the NMS dashboard and the order status will be automatically updated.

Step 3 ‘Schedule Issuance’: The National Store Manager will allocate an order or donation to the driver according to priority request from the health facility.

Step 4 ‘Distribute to Health Facilities’: the driver will deliver drugs to health facilities and update delivery status in the system.

Notes on Figure 4.5 below: Number 3.5.1 was broken down into five processes, namely, Step 1A) search patient, Step 1B) capture patient information, Step 2) update waiting list, Step 3) dispense ACT drugs, and Step 4) add patient to waiting list. These steps came from the analysis of the results of 3.5.1 ‘Dispense ACT’ as shown in Table 4.15. Number 1.4.1 ‘Feedback of service and delivery’ came from the analysis of the 1.4.1 Feedback loop with other health facilities, including suppliers in Table 4.1.

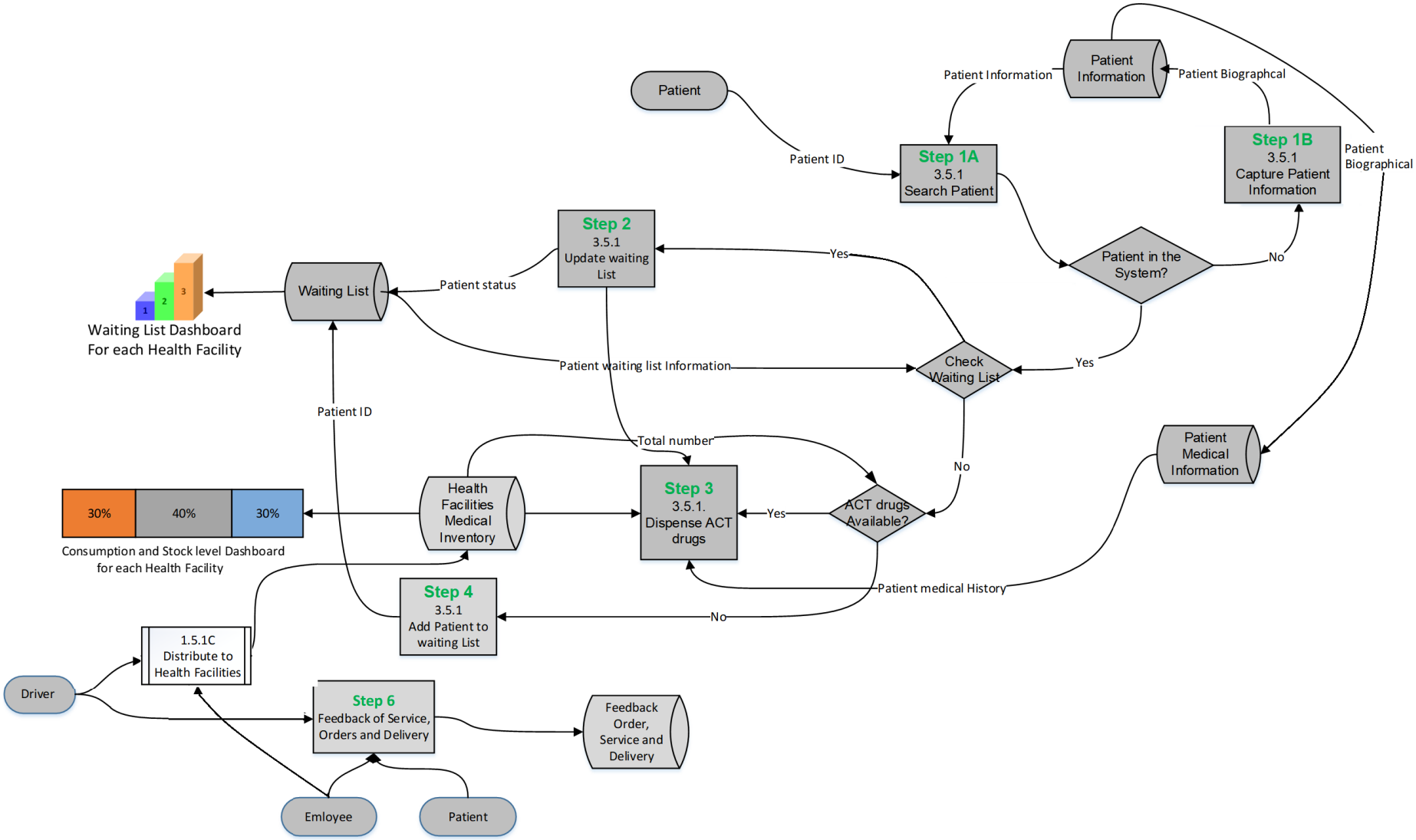


Figure 4.5: Dispense process flow

Figure 4.5 represents a process flow of micro-functional tasks that deal with drug dispensing as gathered in the first stage of the waterfall development methodology, and it corresponds with the numbers in dimension Table 4.15.

From the micro-level, the patient must be in the system before receiving the drug, according to the Business Rule in Table 4.27. Step 1 has two sub-steps, namely: Step 1A and Step1B to ensure that patient information is accurate. The Pharmacist must first search for the patient in Step 1A ‘Search patient’. If the patient detail is not found, the patient must be added to the system in Step 1B ‘Capture Patient Information’. Once the patient has been added into the system, the Pharmacist will first check if the stock is still available, and if not, the patient can be added to the waiting list.

To be able to monitor the live dispatch of drugs to patients, it is necessary to monitor stock levels and to know when the stock reaches re-order points. Only patients in the system can receive the drug in Step 3 (Figure 4.5) above. This will also help the following functionalities to meet their intended purpose, namely, the Demand report functionality; Stock status report functionality; Active orders report functionality; Maximum-minimum stock levels functionality; Information from the functionality of the dispensing log; Malaria seasons based on History Report functionality; and Peak times based on History Report functionality.

When Step 3 is executed, the system automatically updates the consumption and stock level dashboard and the above-mentioned functionalities are also updated.

At the market level, suppliers and distributors will be able to monitor the live consumption and dispatch of the drugs for predictive production plans and production order quantities. Using the above functionalities, the system will provide them information crucial for production and resources planning such as raw materials, labour, and machinery to offer just-in-time service to hospitals and clinics. The system will provide top management with the intelligence of the when, who, where, and what necessary to combat theft and the sudden disappearance of stock.

Notes: As seen in Figure 4.6 below, number 3.4.1 ‘Process order’ (Step 4) and 3.4.1 ‘Check delivery supplier schedule’ (Step 2C) were broken down after the analysis of number 3.4.1 ‘Order drug functionality’, as in Table 4.15.

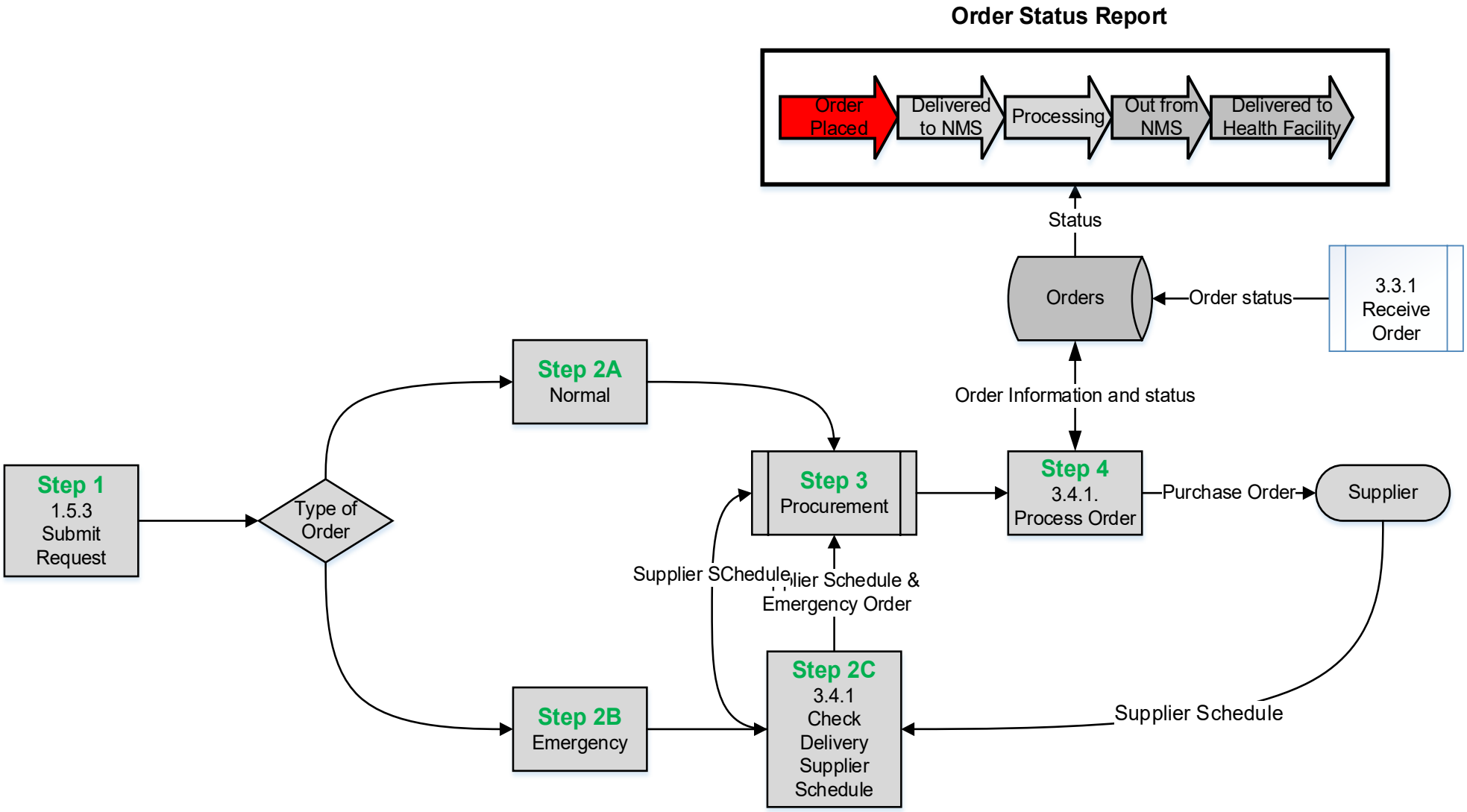


Figure 4.6: Order process flow

Note: In Figure, 4.7, number 3.4.1 ‘Stock level warning’ or demand increase came from the ‘Analysis of order drug’ functionality in Table 4.15.

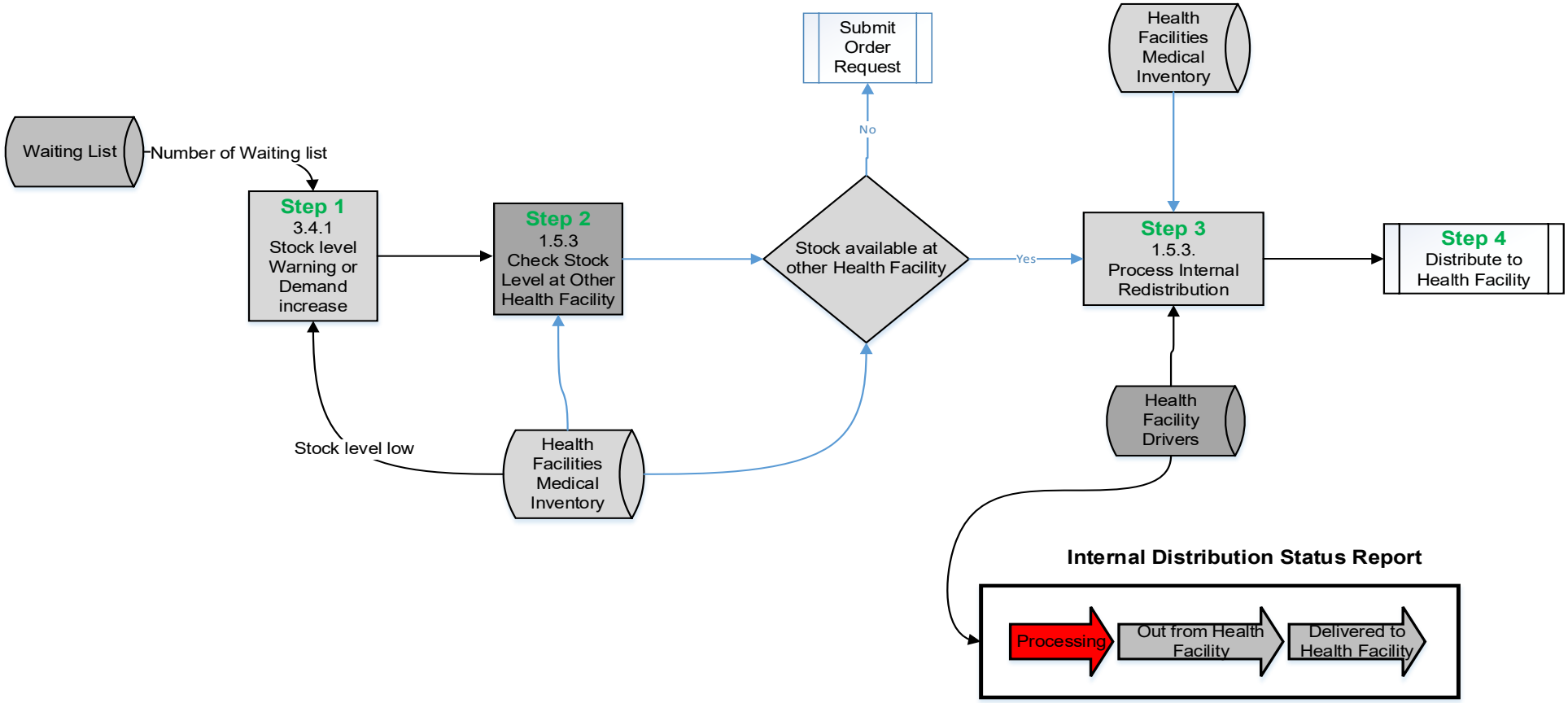


Figure 4.7: Internal redistribution process flow

Figure 4.7 represents the process flow of the micro-functional tasks that deal with internal redistribution. The different process flows were used to produce the use cases in Figure 4.8 below, which is discussed in detail in Section 4.2.2.

As indicated in Figure 4.7 above, when the stock reaches re-order points, the economic order quantity feature alerts the health facility to replenish the stock by way of sending a request to the NMS. The request can be sent as either normal or emergency, hence, Step 2A for normal status and Step 2B for emergency status. This status will be used to determine the urgency of the order. If the status is emergency, the NMS must be able to check which supplier will respond within a specified time to the emergency (Step 2C). The supplier must share their stock levels. The procurement process (Step 3) is done outside the scope of the system, but provides feedback to the process with the permission to order drugs from the supplier.

In Step 1 ‘Stock level Warning’ or ‘Demand increase’ (Figure 4.7), the system will warn if the stock level is reaching a certain level or if the demand is increasing. Step 2 ‘Check Stock Level at Other Health Facility’ will help the Pharmacist to check the closest health facility before sending a request to the NMS. The system will alert the health facility and the NMS about areas with low and high consumption. This data is crucial for redistribution, to respond to sudden demand or a humanitarian crisis, and helps to avoid overstocks.

After analysing the user requirements and flowcharts in Section 4.2, the researcher was able to understand what the critical logistics, micro, market, and macro dimensions and the associated sub-dimensions of ACTs are that can be converted into technology features. The following core features were identified:

A. Market
    1. Receiving and Dispatch
    2. Order
    3. Supplier
B. Micro
    1. Dispense
    2. Internal Redistribution
C. Macro
    1. Dashboard

The core features were used to group related use case to create sub-systems.

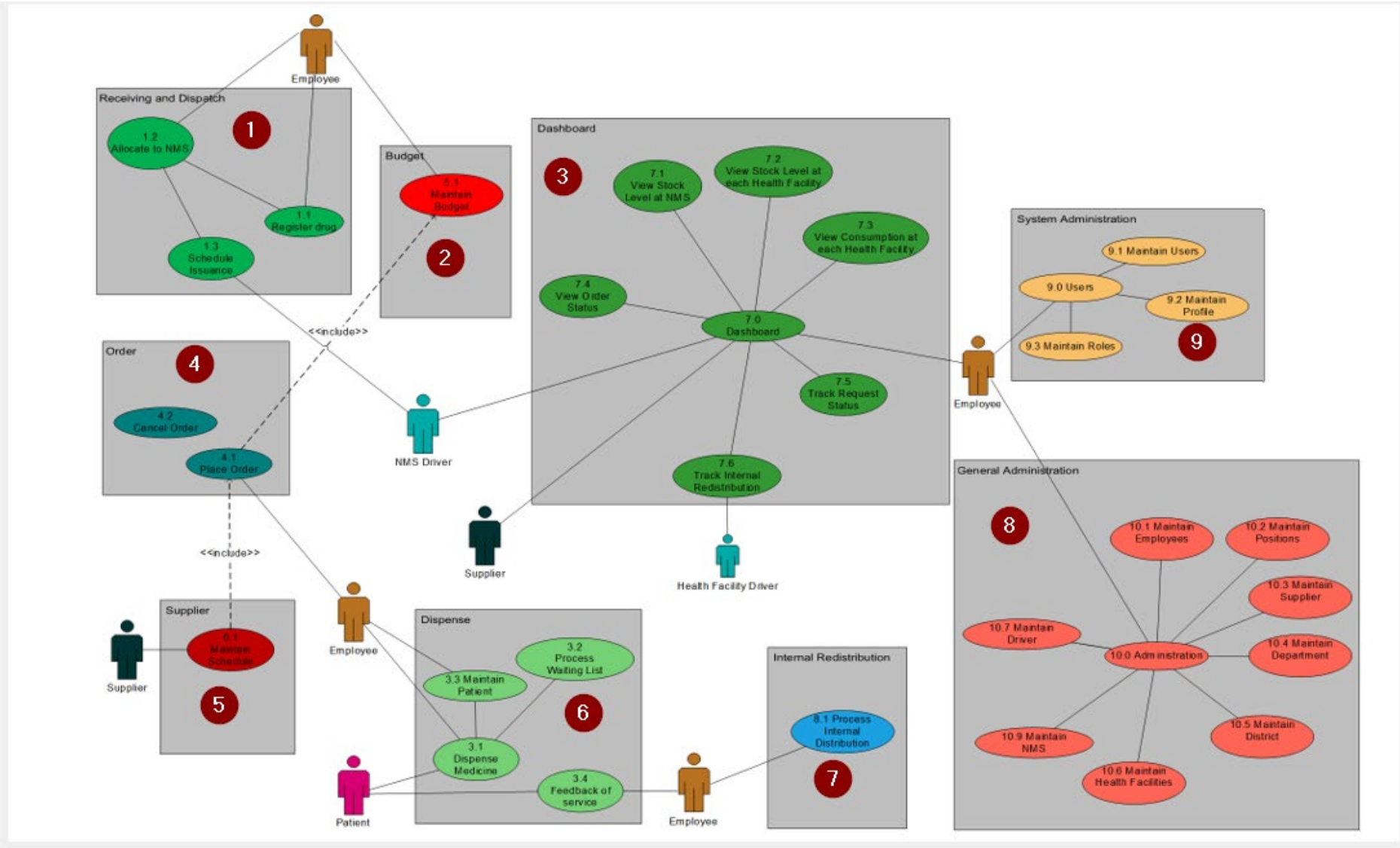


Figure 4.8: Use case context diagram

4.2.2 Use cases

According to Wautelet *et al.* (2016), a use case is made up of a set of scenarios. Each scenario is a sequence of steps that encompass the interaction between a user and a system. A use case represents a functionality in the form of graphics. Singh, Sharma and Saxena (2016) explain that an actor in a use case diagram refers to the role that a user plays in the system. Singh *et al.* (2016) further explain that use cases that are grouped together to accomplish a specific goal are called a sub-system, hence, Figure 4.8 above grouped the use cases accordingly.

The use case diagram in Figure 4.8 is the first that the developer or programmer will work with when starting the coding process. Khan and Khan (2011) indicated that use case diagrams are applied to visualise the behaviour of a system, sub-system, or class, to allow users to comprehend how to use that element, and so that developers can implement the use case element.

Figure 4.8 above shows grouped use cases, and the following names of the groups, namely, Receiving and Dispatch, Budget, Dashboard, Order, Supplier, Dispense, Internal Redistribution, System Administration and General Administration Those group names becomes the names of each sub-system.

According to Wang *et al.* (2016), each use case requires a vital amount of text to define it. This text is usually formatted as shown in Table 4.27.

Table 4-27: Use case detail information framework

Use case section	Description
Name	A suitable name for the use case, such as functionality.
Brief description	A narrative of the use case's role and purpose.
Flow of events	A description of what the system does with regard to the use case.
Special requirements	A description that collects all the requirements, such as non-functional requirements on the use case, that are not considered in the use case model, but that need to be taken care of during design or implementation.
Preconditions	Any constraints on the system at the time the use case may start.
Post conditions	Any constraints on the system at the time the use case will terminate.

4.2.2.1 Receiving and dispatch sub-system

The receiving and dispatch sub-system registers drugs from either supplier or donors, and allocate the drug to the NMS for accountability and proper reporting. It also schedules the drugs for delivery, including allocating the task to the driver. The sub-system includes the following use cases, namely, register drug, schedule issuance, and allocate to National Medical Store.

Register drug use case

The purpose of the register drug use case is to capture the drugs from either suppliers or donors into the NMS database.

Table 4-28: Register use case

Use case section	Description
Name	1.1 Register
Brief description	All the drugs must be registered in the system before distribution to the health facilities.
Flow of events	Search the barcode from the database. If in the system, check the expiry date. If not in the system, all the required information must be captured.
Special requirements	All drugs must have a barcode.
Preconditions	None
Post conditions	None

Allocate to NMS use case

Before the drugs can be distributed to the health facility, it must first be recorded under the name of the receiving NMS for the purpose of reporting.

Table 4-29: Allocate to NMS use case

Use case section	Description
Name	1.2 Allocate to NMS
Brief description	All the drugs must first be allocated to NMS before can be distributed to Health Facility.
Flow of events	Add the barcode and quantity to the NMS.
Special requirements	A user must only add drugs to the linked NMS
Preconditions	Drugs must have registered in the system.
Post conditions	None.

Schedule Issuance use case

The drug can only be distributed to the health facility after being registered and assigned to the NMS. The purpose of this use case is to prepare for the delivery, based on the priority and the drugs being assigned to the driver.

Table 4-30: Schedule Issuance use case

Use case section	Description
Name	1.3 Schedule Issuance
Brief description	Allocate the drugs to the driver for delivery.
Flow of events	Add the request number, then order number, and then update the tracking report.
Special requirements	Health facility must have requested the drugs.
Preconditions	None.
Post conditions	None.

4.2.2.2 Budget sub-system

The purpose of the budget sub-system is to maintain the budget.

Maintain Budget use case

No health facility can procure the drugs without having sufficient budget, unless it is obtained from donors. The system will not allow the procurement of drugs without the required budget, therefore, the purpose of this use case is to view and update the budget.

Table 4-31: Maintain Budget use case

Use case section	Description
Name	5.1 Maintain Budget
Brief description	To ensure that health facilities are able to manage their budget.
Flow of events	First check if the budget is available before making the request for drugs.
Special requirements	None.
Preconditions	None.
Post conditions	None.

4.2.2.3 Order sub-system

The purpose of the order sub-system is to order the drugs from the supplier. According to the Business Rule, only the NMS can order the drugs on behalf of health facilities.

Place Order use case

The purpose of this use case is to place and cancel an order only with registered suppliers. The placing of the order will depend on the priority and delivery time.

Table 4-32: Place Order use case

Use case section	Description
Name	4.1 Place Order
Brief description	To place the order with the supplier.
Flow of events	Receive the request from the health facility and then place the order with the supplier.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Table 4-33: Cancel Order use case

Use case section	Description
Name	4.2 Cancel Order
Brief description	To cancel the order with the supplier.
Flow of events	Cancel the order, then update the request status.
Special requirements	Order must only be placed by the NMS.
Preconditions	None.
Post conditions	None.

4.2.2.4 Supplier sub-system

All the suppliers must be registered into the database before they can receive or supply drugs. The purpose of this sub-system is to register suppliers into the database.

Maintain Schedule use case

When the NMS places the order with the supplier, they must first check which suppliers have stock and the delivery time. The purpose of this use case is to allow the supplier to update the available stock and delivery time on a regular basis.

Table 4-34: Maintain Schedule use case

Use case section	Description
Name	6.1 Maintain Schedule
Brief description	The NMS must be able to see the expected delivery schedule and availability of stock from the supplier so that they can respond fast to orders that have high priorities.
Flow of events	Supplier updates the stock level and status or expected delivery date.
Special requirements	None.
Preconditions	None.
Post conditions	None.

4.2.2.5 Dispense sub-system

The purpose of the Dispensing sub-system is to register the patient. All patients who receive drugs must have been registered into the system before they can receive drugs. If there are no drugs available, a patient can be put on a waiting list. The

dispense sub-system has the following use case, namely, Dispense drug, Process waiting list, and Maintain patient, as presented below.

Dispense drug use case

The purpose of the Dispense drug use case is to dispense drugs to the patient and to create a medical record.

Table 4-35: Dispense Medicine use case

Use case section	Description
Name	3.1 Dispense Medicine
Brief description	Give drugs to patient.
Flow of events	Check the availability of stock, search patient, and then give the drug to the patient.
Special requirements	Drugs cannot be given to patients who are not in the system.
Preconditions	Stock must be available and patient must be in the system.
Post conditions	None.

Maintain Patient use case

The purpose of the Maintain patient use case is to add, update and view patient information.

Table 4-36: Maintain Patient use case

Use case section	Description
Name	3.3 Maintain Patient
Brief description	Add, update and view patient on the system.
Flow of events	First search for patient on the system, and if there are no results, then add the patient.
Special requirements	Patient information.
Preconditions	None.
Post conditions	None.

Process Waiting List use case

The purpose of the Process waiting list use case is to put the patient in the waiting list if no drugs are available.

Table 4-37: Process Waiting List use case

Use case section	Description
Name	3.2 Process Waiting List
Brief description	Add patient to the waiting list.
Flow of events	If the health facility has run out of stock, a patient can be put on a waiting list.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Feedback of service use case

The purpose of the Feedback of service use case is to rate the service of both the pharmacy and the delivery driver.

Table 4-38: Feedback of service use case

Use case section	Description
Name	3.3 Feedback of service
Brief description	Provide feedback of service rendered.
Flow of events	Rate the service by selecting the status and also put a comment.
Special requirements	None.
Preconditions	None.
Post conditions	None.

4.2.2.6 Internal Distribution sub-system

The purpose of the Internal distribution sub-system is to request drugs from the nearest health facility if the health facility is running out of stock, or urgently needs to assist patients who are on waiting list. The sub-system has only one use case, namely, Process internal distribution use case.

Process Internal Distribution use case

The purpose of the Process internal distribution use case is to request drugs from the nearest health facility.

Table 4-39: Process Internal Distribution use case

Use case section	Description
Name	8.1 Process Internal Distribution
Brief description	To request drugs from the closest health facility.
Flow of events	Check the available stock and patients waiting for drugs. If the availability is less than the patients waiting for drugs, send the request to the nearest health facility.
Special requirements	None.
Preconditions	None.
Post conditions	None.

4.2.2.7 Dashboard sub-system

The purpose of the Dashboard sub-system is to report on the activities that are currently happening in the entire ACT system. This starts from when the NMS receives stock, the delivery of drugs, and lastly, the dispensing of drugs to the patient. The sub-system has the following use cases, namely, View stock levels at the NMS, View stock levels at each health facility, View consumption at each health facility, Track request status, Track internal redistribution, and View order status.

View Stock Level at NMS use case

The purpose of this report or graph is to view the stock levels at all active NMSs.

Table 4-40: View Stock Level at NMS use case

Use case section	Description
Name	7.1 View Stock Level at NMS
Brief description	Live status of stock level at each NMS.
Flow of events	View graph of live stock level of NMS.
Special requirements	None.
Preconditions	None.
Post conditions	None.

View Stock Level at each Health Facility use case

The purpose of the View stock level at each health facility use case is to view the report or graph of stock levels at each health facility.

Table 4-41: View Stock Level at each health facility use case

Use case section	Description
Name	7.2 View Stock Level at each health facility
Brief description	Live status of stock levels at each health facility.
Flow of events	View the graph live of stock levels each health facility.
Special requirements	None.
Preconditions	None.
Post conditions	None.

View Consumption at each Health Facility use case

The purpose of the View consumption at each health facility use case is to view the report or graph of the consumption level at each health facility.

Table 4-42: View Consumption at each health facility use case

Use case section	Description
Name	7.3 View Consumption at each health facility
Brief description	View the live consumption graph at each health facility.
Flow of events	View the live consumption graph at each health facility.
Special requirements	None.
Preconditions	None.
Post conditions	None.

View Order Status use case

The purpose of the View order status use case is to view the progress of the requested order.

Table 4-43: View Order Status use case

Use case section	Description
Name	7.4 View Order Status
Brief description	To check the status of an order.
Flow of events	Search the order using the order number to view the status.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Track Request Status use case

The purpose of the Track request status use case is to view the status, or to see what has been done with regard to your request and when you will receive your request.

Table 4-44: Track Request Status use case

Use case section	Description
Name	7.5 Track Request Status
Brief description	To track the progress with regard to request.
Flow of events	Search the tracking using the request number to view the status.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Track Internal Redistribution use case

The purpose of the Track internal redistribution use case is to view the status, or to find out what has been done about your request and when you will receive your request.

Table 4-45: Track Internal Redistribution use case

Use case section	Description
Name	7.6 Track Internal Redistribution
Brief description	To track the progress with regard to internal redistribution request.
Flow of events	Search the tracking using the request number to view the status.
Special requirements	None.
Preconditions	None.
Post conditions	None.

4.2.2.8 General administration sub-system

The purpose of the General administration sub-system is to manage and maintain the data that support and ensure that the system achieves its objectives. The sub-system has the following use cases, namely, Maintain employees, Maintain position, Maintain supplier, Maintain department, Maintain district, Maintain health facilities, Maintain NMS, and Maintain drivers.

Maintain Employees use case

The purpose of the Maintain employees use case is to add, update and view employees.

Table 4-46: Maintain Employees use case

Use case section	Description
Name	10.1 Maintain Employees
Brief description	To add new employee and update employee information.
Flow of events	New appointment or change of employee information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain Positions use case

The purpose of the Maintain position use case is to add, update and view positions.

Table 4-47: Maintain Positions use case

Use case section	Description
Name	10.2 Maintain Positions
Brief description	To add new position and update position information.
Flow of events	New position or change of position information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain Supplier use case

The purpose of the Maintain supplier use case is to add, update and view suppliers.

Table 4-48: Maintain Supplier use case

Use case section	Description
Name	10.3 Maintain Supplier
Brief description	To add new suppliers and update supplier information.
Flow of events	New supplier or change of supplier information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain Department

The purpose of Maintain department use case is to add, update and view departments.

Table 4-49: Maintain Department use case

Use case section	Description
Name	10.4 Maintain Department
Brief description	To add new department and update department information.
Flow of events	New department or change of department information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain District use case

The purpose of the Maintain district use case is to add, update and view district.

Table 4-50: Maintain District use case

Use case section	Description
Name	10.5 Maintain District
Brief description	To add new district and update district information.
Flow of events	New district or change of district information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain Health Facilities

The purpose of the Maintain health facility use case is to add, update and view health facilities.

Table 4-51: Maintain health facilities use case

Use case section	Description
Name	10.6 Maintain Health Facilities
Brief description	To add new health facilities and update health facilities' information.
Flow of events	New health facility or change of health facility's information
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain National Medical Store (NMS) use case

The purpose of the Maintain NMS use case is to add, update and view the NMS.

Table 4-52: Maintain National Medical Store (NMS) use case

Use case section	Description
Name	10.9 Maintain National Medical Store (NMS)
Brief description	To add new NMS and update NMS information.
Flow of events	New NMS or change of NMS information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain Driver use case

The purpose of the Maintain driver use case is to add, update and view drivers.

Table 4-53: Maintain Driver use case

Use case section	Description
Name	10.7 Maintain Driver
Brief description	To add new driver and update driver information.
Flow of events	New driver or change of driver information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

4.2.2.9 System Administration sub-system

The purpose of the System administration sub-system is to create users so that they can log in into the system, and assign a role to their profile according to their job description. The sub-system has the following use cases, namely, Maintain users, Maintain roles and Maintain a profile.

Maintain Users use case

The purpose of the Maintain user use case is to add, update and view users.

Table 4-54: Maintain Users use case

Use case section	Description
Name	9.1 Maintain Users
Brief description	To add new user and update user information.
Flow of events	New user or change of user information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain User Profile use case

The purpose of the Maintain user profile use case is to add, update and view a user profile.

Table 4-55: Maintain User Profile use case

Use case section	Description
Name	9.2 Maintain User Profile
Brief description	To add new user profile and update user profile information.
Flow of events	New user profile or change of user profile information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Maintain Roles

The purpose of the Maintain roles use case is to add, update and view roles.

Table 4-56: Maintain Roles use case

Use case section	Description
Name	9.3 Maintain Roles
Brief description	To add new role and update role information.
Flow of events	New role or change of role information.
Special requirements	None.
Preconditions	None.
Post conditions	None.

Login use case

The purpose of the Login use case is to allow an authorised user to login into the system, and to view the sub-system according to assigned roles.

Table 4-57: Login use case

Use case section	Description
Name	9.4 Login
Brief description	To control access to the system. The user will only have access according to their role and responsibilities.
Flow of events	The user can be either employee or supplier or driver. The user will enter username and password. Once login is successful, the user will only see the functionality that a user has access to.
Special requirements	None.
Preconditions	None.
Post conditions	None.

The user requirements analysis outcome and/or specifications (flowcharts and use cases) are used to design the system architecture. The system architecture covers the logical and physical architecture of the mobile application, as discussed in the next section.

4.3 DESIGN STAGE

Mademann (2018) defined system design as a conceptual model that outlines the structure, behaviour and views of a system. Booch et al. (2008), mentions two common software design methods, the first being: structured design, defined as a method to convert requirements specification such as Data Flow Diagrams (DFD) into a system flowchart that can be implemented using a programming language; and the second being object-oriented design, defined as a mechanism that encompasses modularity, abstraction, and encapsulation as three important concepts in software design.

This section therefore starts by justifying the design method selection, and then proceeds to discuss the design in detail.

4.3.1 Justification for the study's design method: Object-oriented design

According to Booch *et al.* (2008), OOD uses UML notations, such as use cases and class diagrams presented in Section 4.2. The use cases assist in identifying the objects, which in this study include mobile phone, barcode, QR codes, and the scanner. The amalgamation of sensing objects (such as mobile phones, scanners, barcode, and the QR code), application layer (such as Azure cloud, APIs, web, portal, Dashboard, database), and the network layer such as the internet are integral in the IoT framework and supports the client-server architecture model.

Object-oriented design has a natural modular design structure that easily allows changes to be made to it without affecting the other modules. In this study, the modules that are integral to the object–oriented programming are APIs (using C# language), and mobile application (using Xamarin platform, dot net, and C#). Although most scholar and programmers rarely use OOD within a waterfall model (Han, 2006; Brindha & Vijayakumar, 2015), the integration of both structured waterfall principles and OOD programming was critical and possible in this study. The OOD programming is crucial for flexible adjustment of the sensing and application layers which had several objects, hence the waterfall model was necessary for the supply chain coordination flow of the micro, macro and market stakeholders. This study provides a context and evidence of how OOD can complement structured principles of the waterfall model in the IoT context. Therefore, the mobile application developed in the current study used the Application Programming Interface (API) to create independence between the mobile application and the database. The mobile application was developed using C# and other related object-oriented programming languages that are outlined in Section 4.3.3.1 (below).

According to Balasubramanian, Aloqaily, Zaman and Jararweh (2018), the objectives of both logical and physical design are to outline and document the logical and physical components of a system, to provide clarity around how those component elements relate to one another.

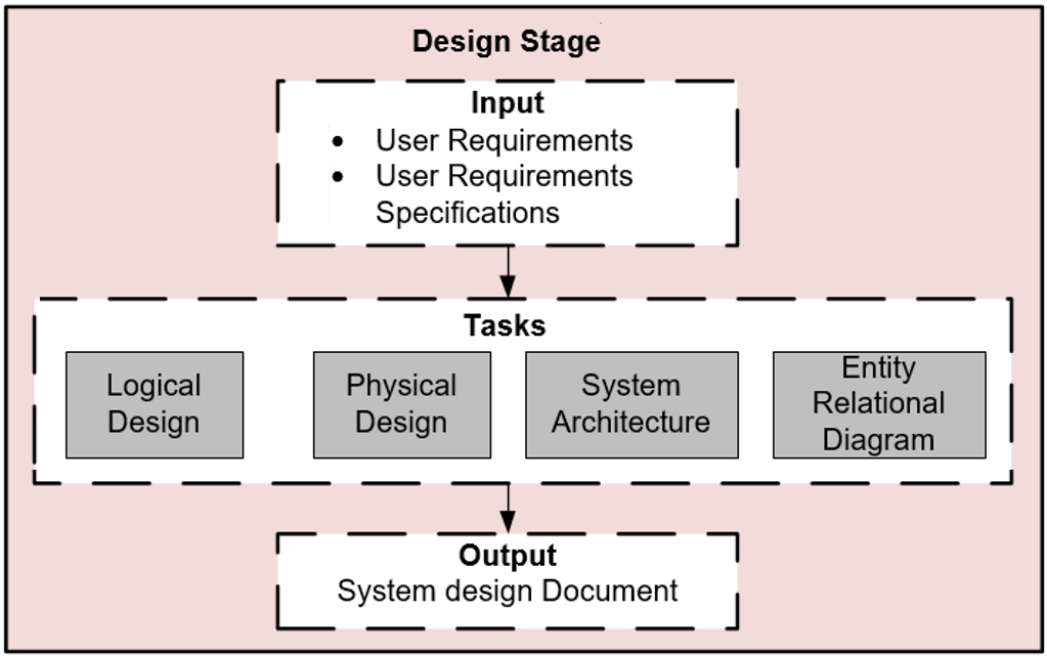


Figure 4.9: Design stage

4.3.1.1 Logical design

The logical design outlines the layers and technologies that need to be in place, and how they are logically related or connected to each other. For the purpose of the current study, the researcher applied the IoT principles to create logical layers, as illustrated in Figure 4.10 below. This will make it easy for cloud deployment. It also shows how the system integrates.

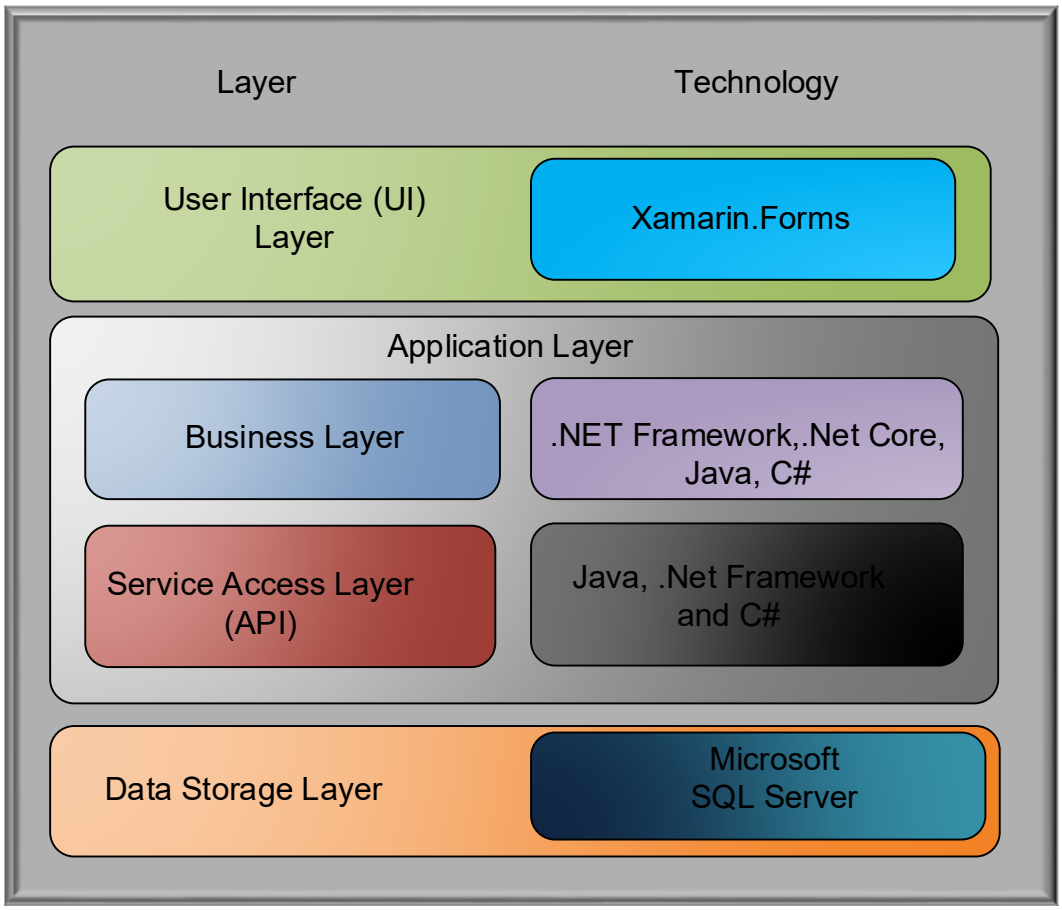


Figure 4.10: Logical design

According to Doebelin and Kleeberg (2015), the user interface is usually graphical and accessible, or can be viewed through a monitor which displays content and information that is useful to the end-user. It controls the application's core functionality by performing detailed processing, and is usually coded in a programming language, such as Python, Java, C++ and .NET.

According to Wieringa, Blanken, Fokkinga and Grefen (2003), the application layer is mostly divided into two layers depending on the application requirements, such as, for example, the business layer that provides the logic that contains the business rules and also contains the code to interface with the service layer, and the service access layer that is used to access database services from the on-premise database server or the cloud: from complex web Application Programing Interface (API) services, such as WCF, JSON, or REST, to the simple retrieval of data from remote servers. It encapsulates the networking behaviour and provides a simple API to be consumed by the application and UI layers.

Data in this tier is kept independent of application servers or business logic, and is managed and accessed with programs, such as MongoDB, Oracle, SQL lite, MySQL, and Microsoft SQL Server (Wieringa *et al.*, 2003).

4.3.1.2 Physical design

The physical design specifies the actual devices that need to be in place and how they are physically related or connected to each other.

According to Savu (2011), cloud deployment models show how the cloud services are made available to users. Savu (2011) also mentions the four typical deployment models associated with cloud computing are as follows:

- **Public cloud** is a platform that uses the standard cloud computing model to make resources, such as hardware (OS, CPU, memory, storage) or software (application server, database) on a subscription basis, available to users remotely. The public cloud platform is commonly used for application development, testing, a non-mission-critical system such as file-sharing, e-mail service, and so on.
- **Private cloud** is a platform that is only accessible to users who have been given access. The platform can be housed on-premise or off-premise. Such a platform may be managed by the organisation itself to support users, or by a service provider that takes care of it either on-premise or off-premise.
- In a **hybrid cloud**, an organisation makes use of both private and public cloud platforms.
- **Community cloud** is a deployment model that manages two or more institutions or organisations that are sharing resources such as virtual machines, applications or storage, and that are a member of a community.

There are significant variations on these four deployment models, depending on other factors that will be outlined in the next section, but they serve to address the broad questions as to how one can deploy cloud resources.

The NIST Cloud Model makes the following two important points (Bohn *et al.*, 2011):

- According to Jamsa (2012), a customer or tenant can have greater security control over more resources as one moves from Cloud SaaS (Software-as-a-Service) to PaaS (Platform-as-a-Service) and again from PaaS to the IaaS (Infrastructure-as-a-Service) service model. (This is discussed in more detail below.)

- Jamsa (2012) further explained that a customer or tenant could achieve greater security control over more resources when moving from a public cloud to a community cloud, and again from a community cloud to a private cloud.

Figure 4.11 below is an adaption of the NIST Cloud Computing Model (NIST, 2009) that has been annotated to reflect the discussion in this section on customer and tenant control. The next section examines the issue of control in detail.

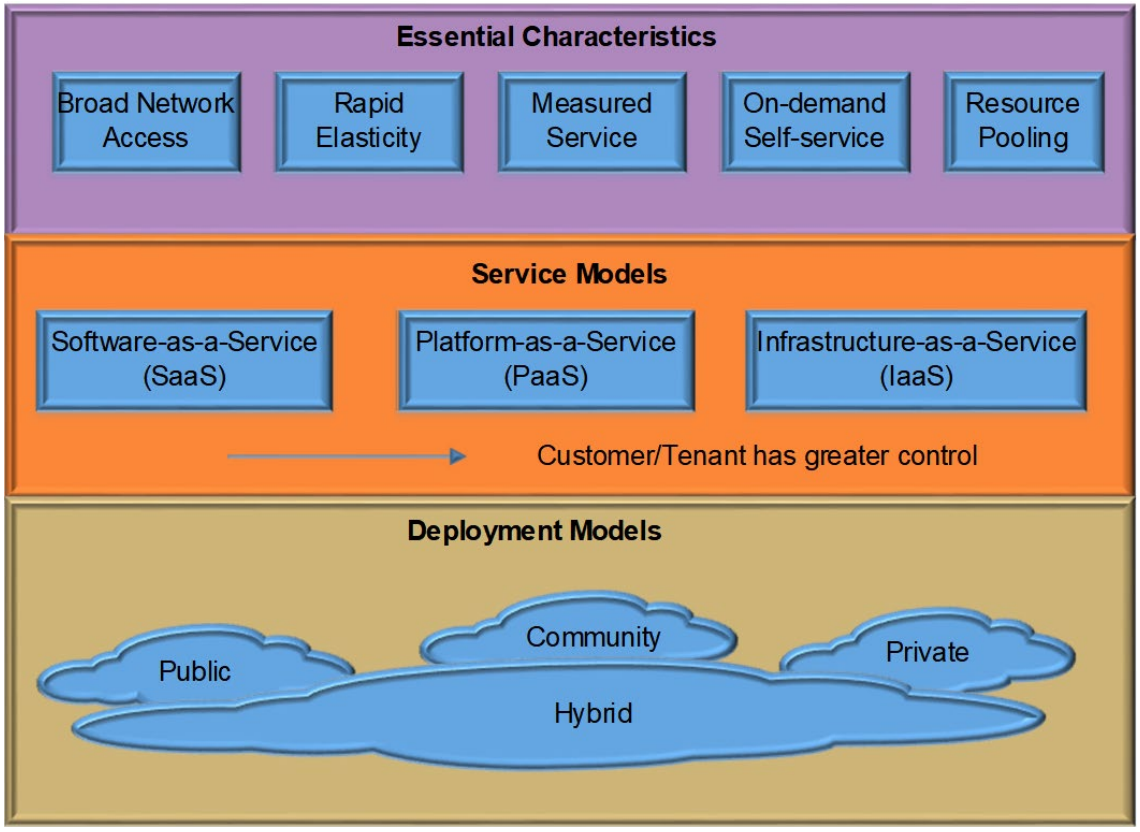


Figure 4.11: The annotated NIST cloud model

According to NIST (2009), they are three service models for cloud computing, as shown in Figure 4.11 above. Cloud Software-as-a-Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a Web browser (for example, Web-based e-mail). The consumer does not manage or control the underlying cloud infrastructure that includes the network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Cloud Infrastructure-as-a-Service (IaaS)

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems; storage, deployed applications, and possibly limited control of select networking components (for example, host firewalls).

Deploying the ACT mobile app to the cloud, for example, the Azure platform, requires a computer image that behaves like an actual computer, known as a virtual machine. In addition, the storage of the data files, including the Database Management System (DBMS), as illustrated in Figure 4.12, requires the minimum virtual machine and storage specifications, as listed in Tables 4.58 and 4.59, or an Azure App Service Environment (ASE) that provides a fully isolated and dedicated environment to securely run app service applications at a high scale.

Table 4-58: Virtual machine minimum specifications

Virtual machine			
Operating system	**Size**	**vCPUs**	**RAM**
Windows 2019-Datacenter	Standard B2s	4	8 GiB

Table 4-59: Disk storage minimum specifications

Disk storage				
Storage type	**Size (GiB)**	**Max IOPS**	**Max throughput (MBps)**	**Encryption**
Premium SSD	500	500	100	SSE with PMK

4.3.2 System architecture

This section describes the system architecture for the mobile application of ACT. The mobile application will be packaged into the Android Package Kit, and then deployed to Google cloud. The API and database will both be deployed into Azure. Figure 4.12 shows the various components of the system architecture.

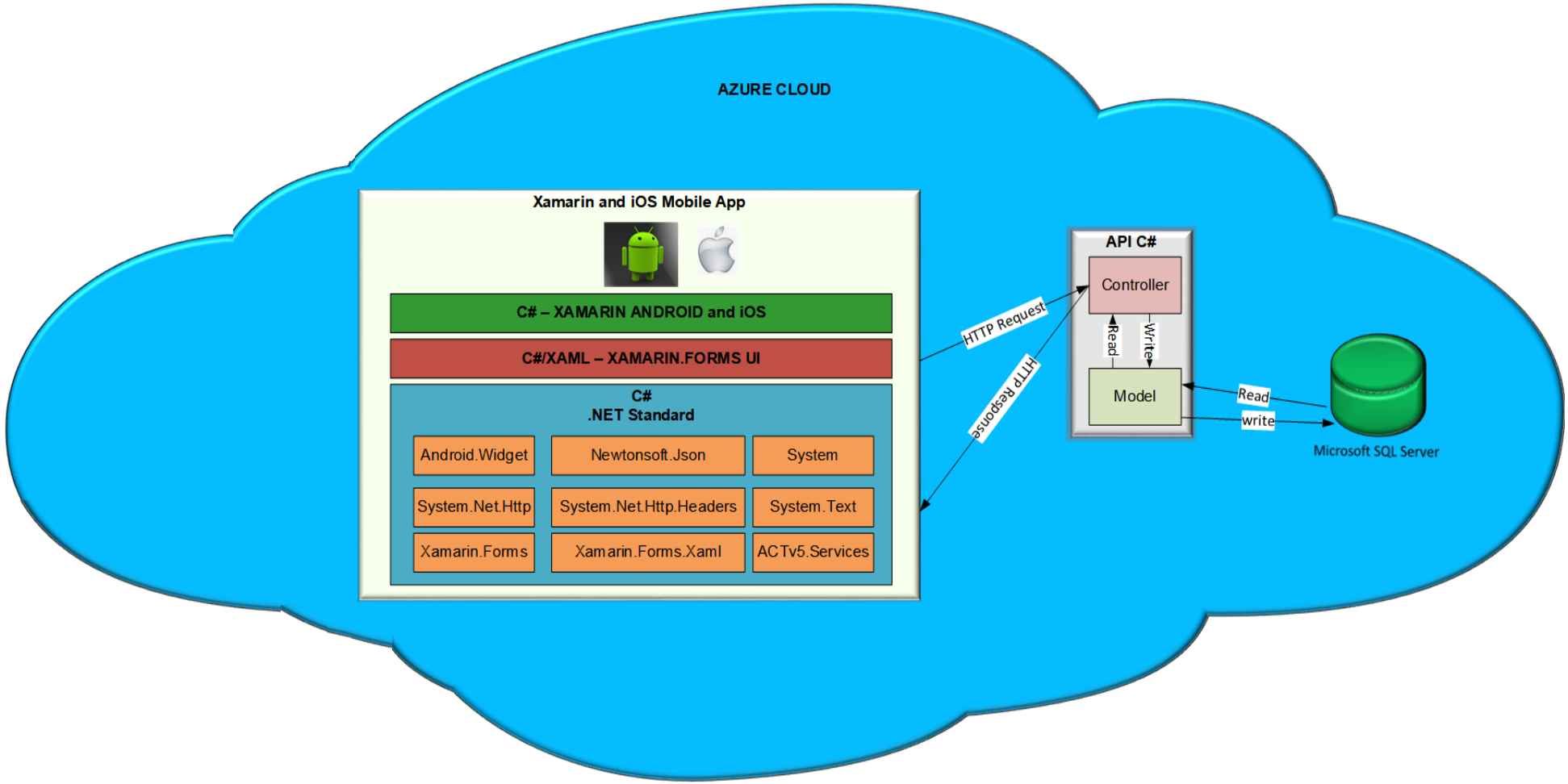


Figure 4.12: System architecture of ACT mobile app

4.3.3 Entity Relational Diagram

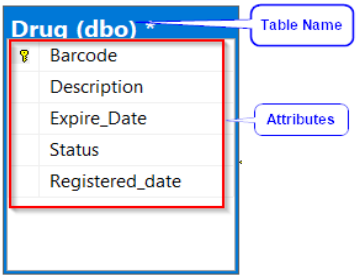
The Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design, as explained by Brady and Loonam (2010). An ERD contains different symbols and connectors that visualise the following two important pieces of information: the most common entities within the system scope, and the inter-relationships among these entities (Rossi, 2014).

Entities in ERD very often refer to business objects such as people/roles (for example, Student), tangible business objects (for example, Product), intangible business objects (for example, Log), and so on. 'Relationship' is about how these entities relate to each other within the system.

All the primary keys start with the character to identify the entity, for example, Employee: EMP00001, National Medical Store: NMS01, Health Facility: HF01, and so on.

Figures 4.13 to 4.22 provide the ERDs for each element relevant to the current mobile app development. The figures show how the database is structured, including the entities and associated relationships among the entities. Table 4.60 shows the symbols and their meaning, as used in the figures.

Table 4-60: ERD symbols

Symbol	Name
	Entity
	Relationship

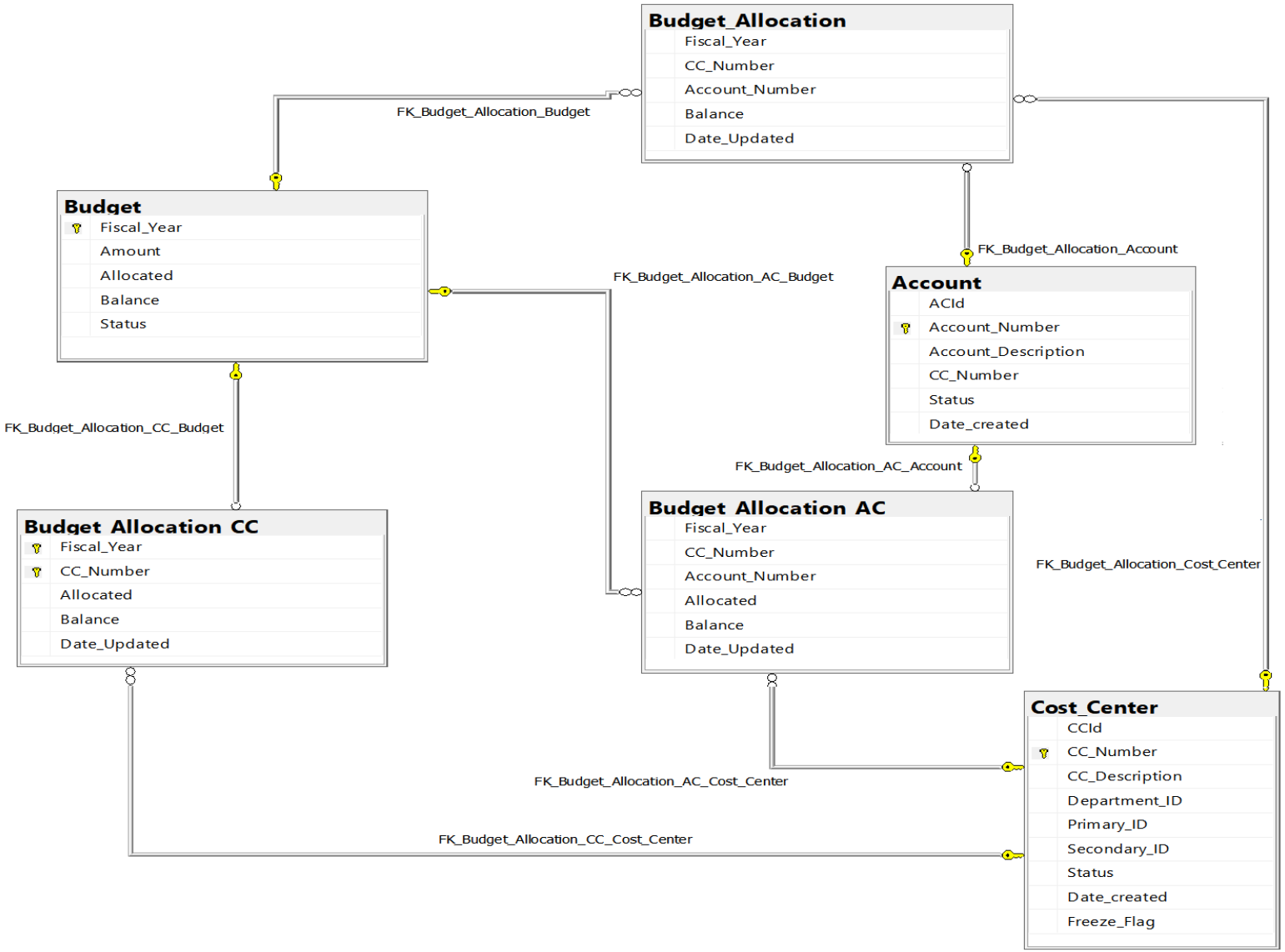


Figure 4.13: ERD 1 of 10

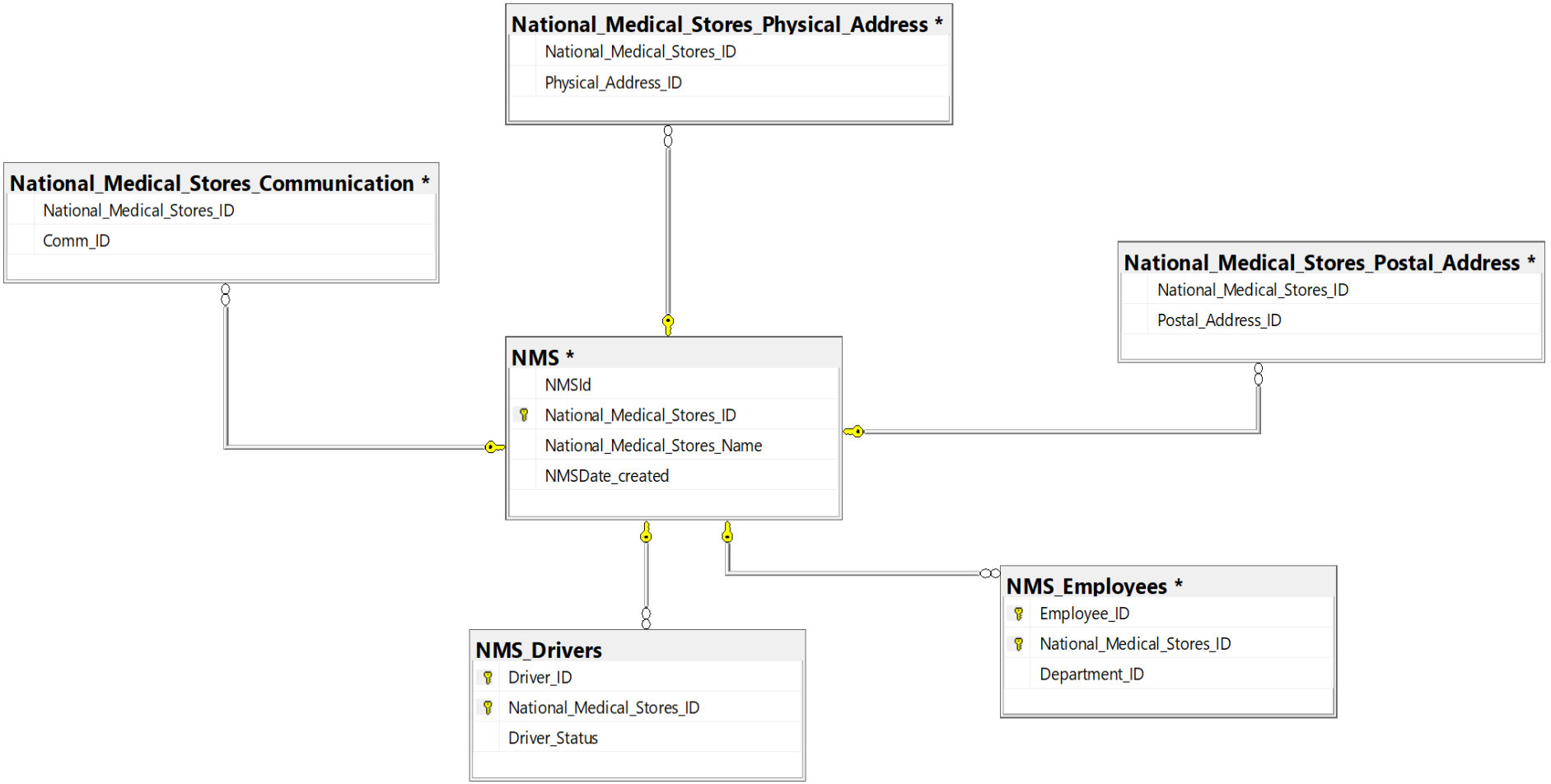


Figure 4.14: ERD 2 of 10

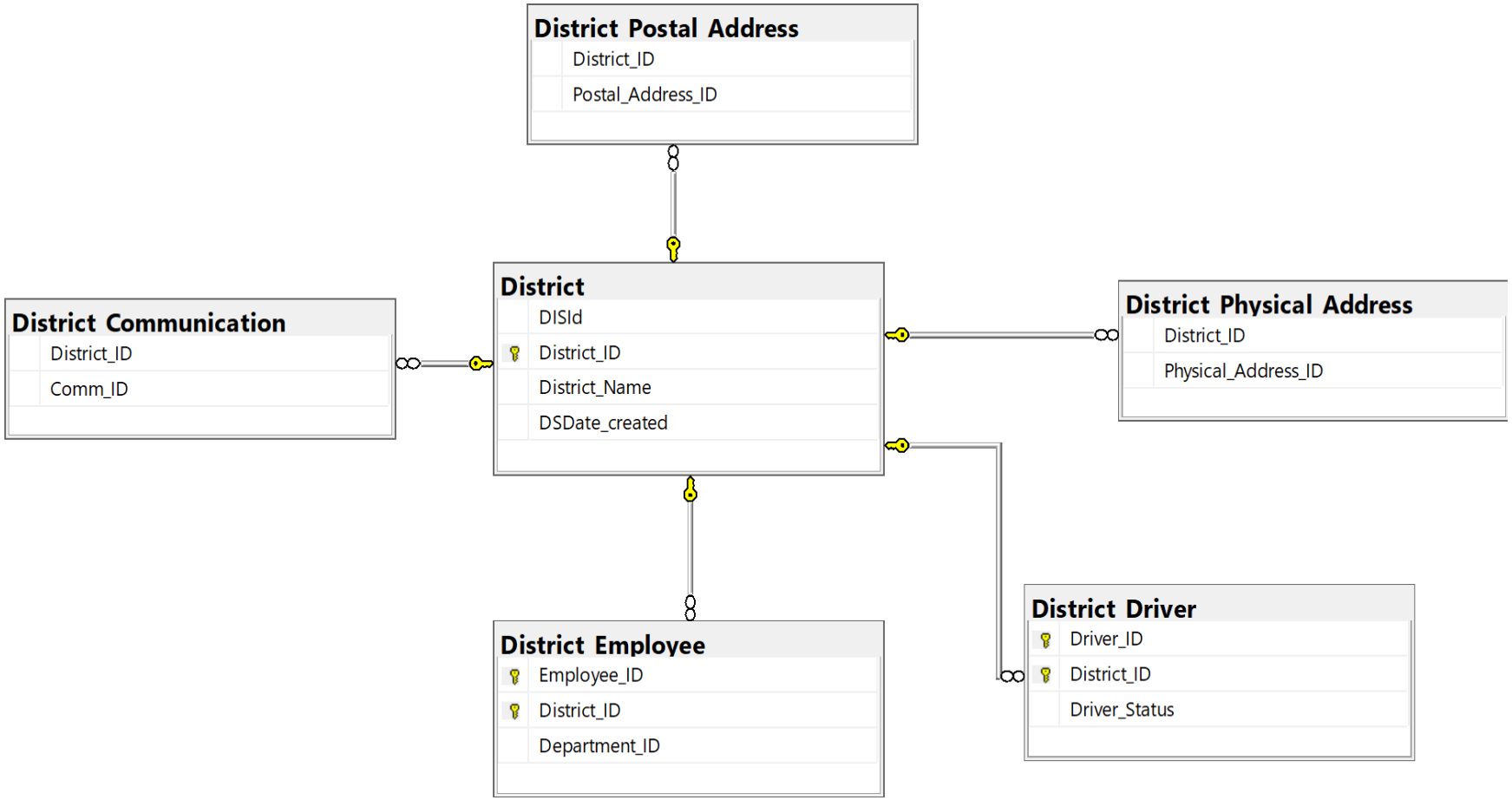


Figure 4.15: ERD 3 of 10

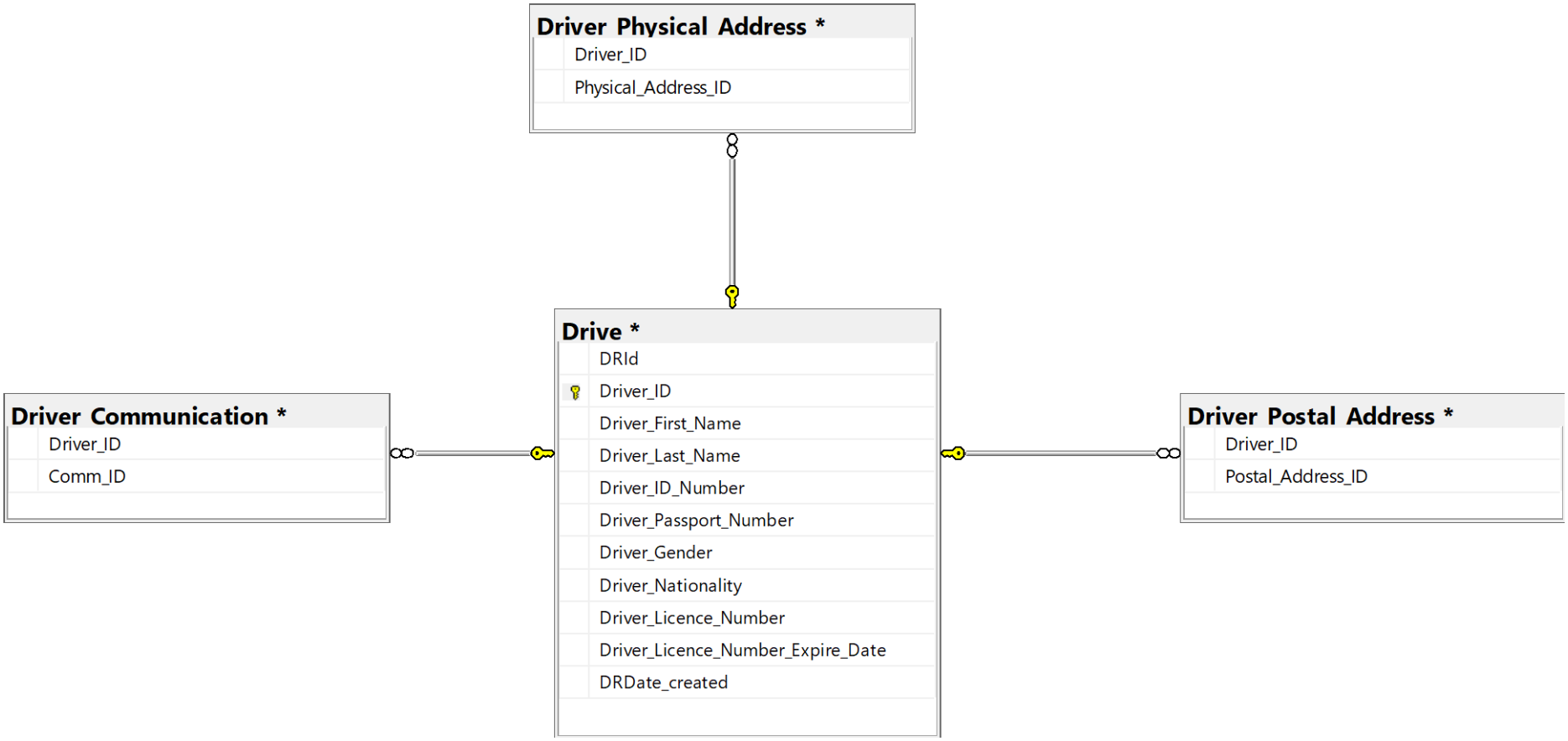


Figure 4.16: ERD 4 of 10

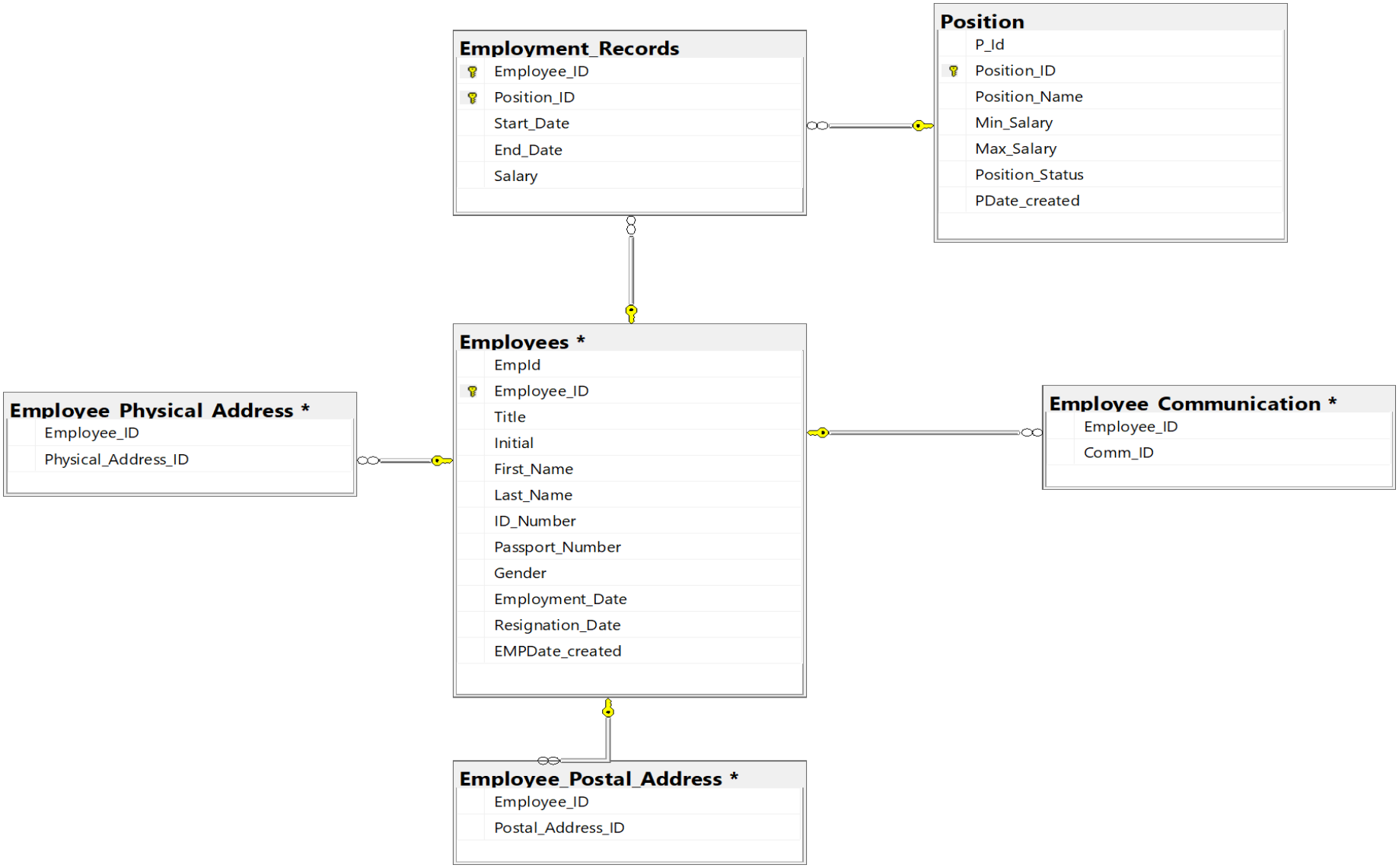


Figure 4.17: ERD 5 of 10

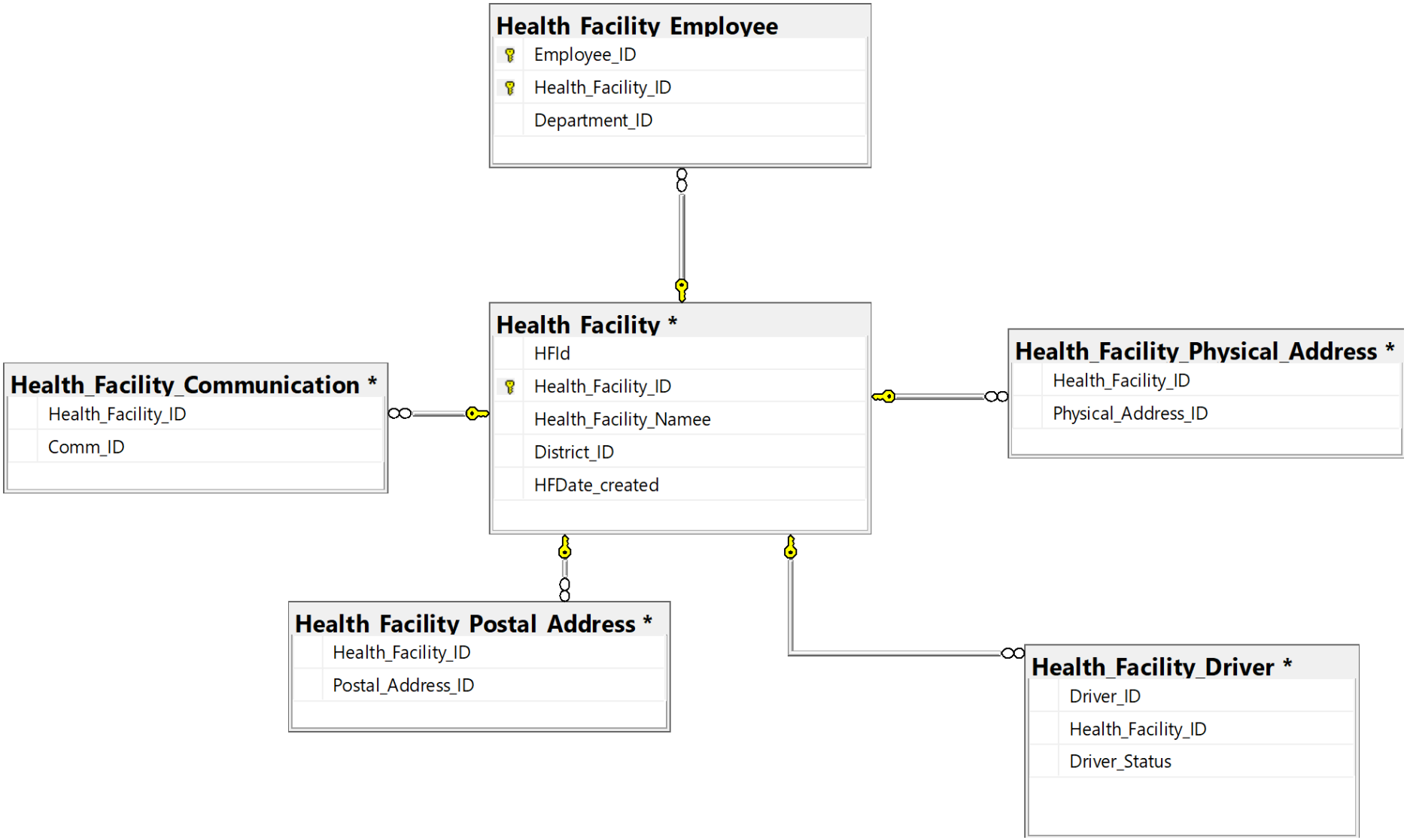


Figure 4.18: ERD 6 of 10

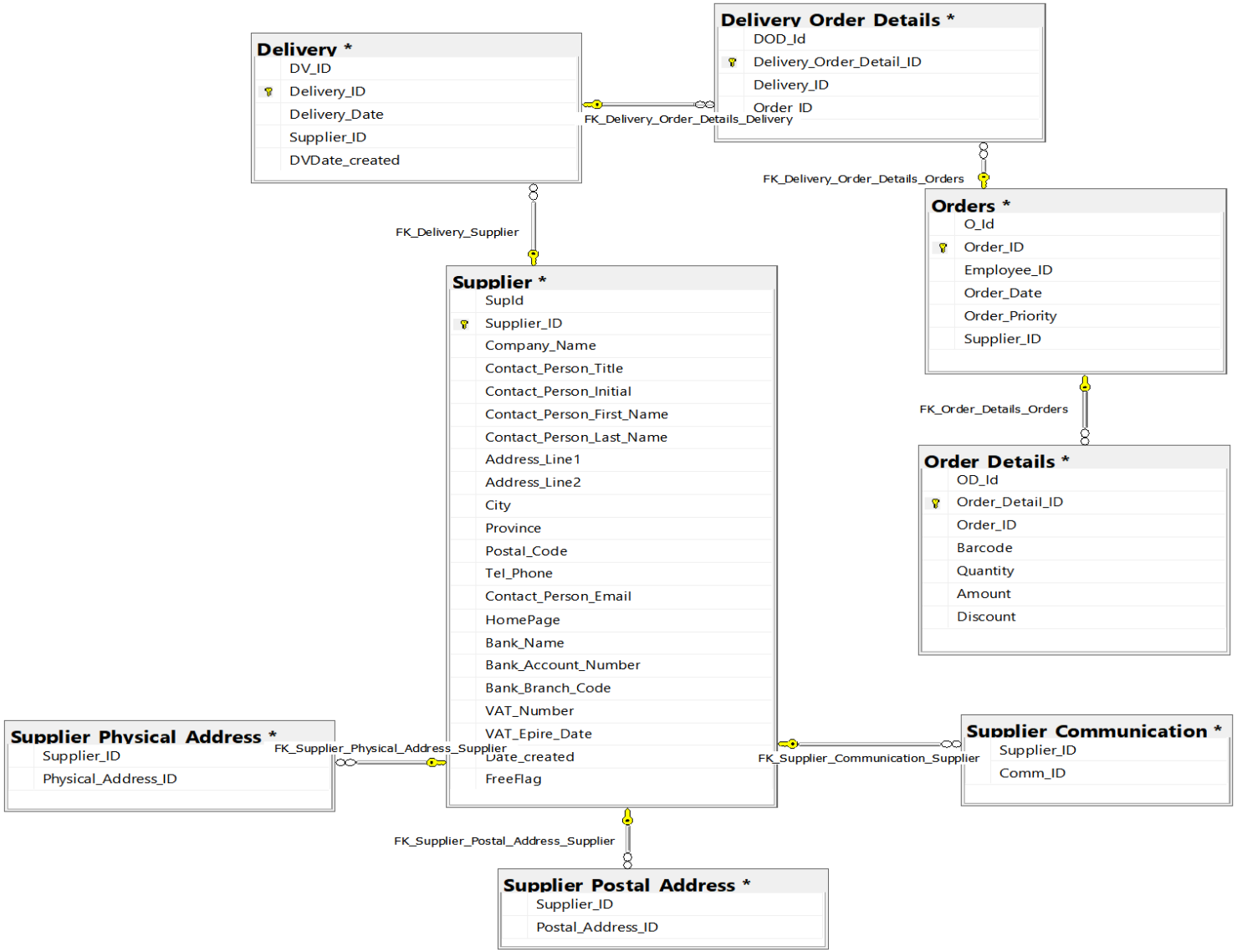


Figure 4.19: ERD 7 of 10

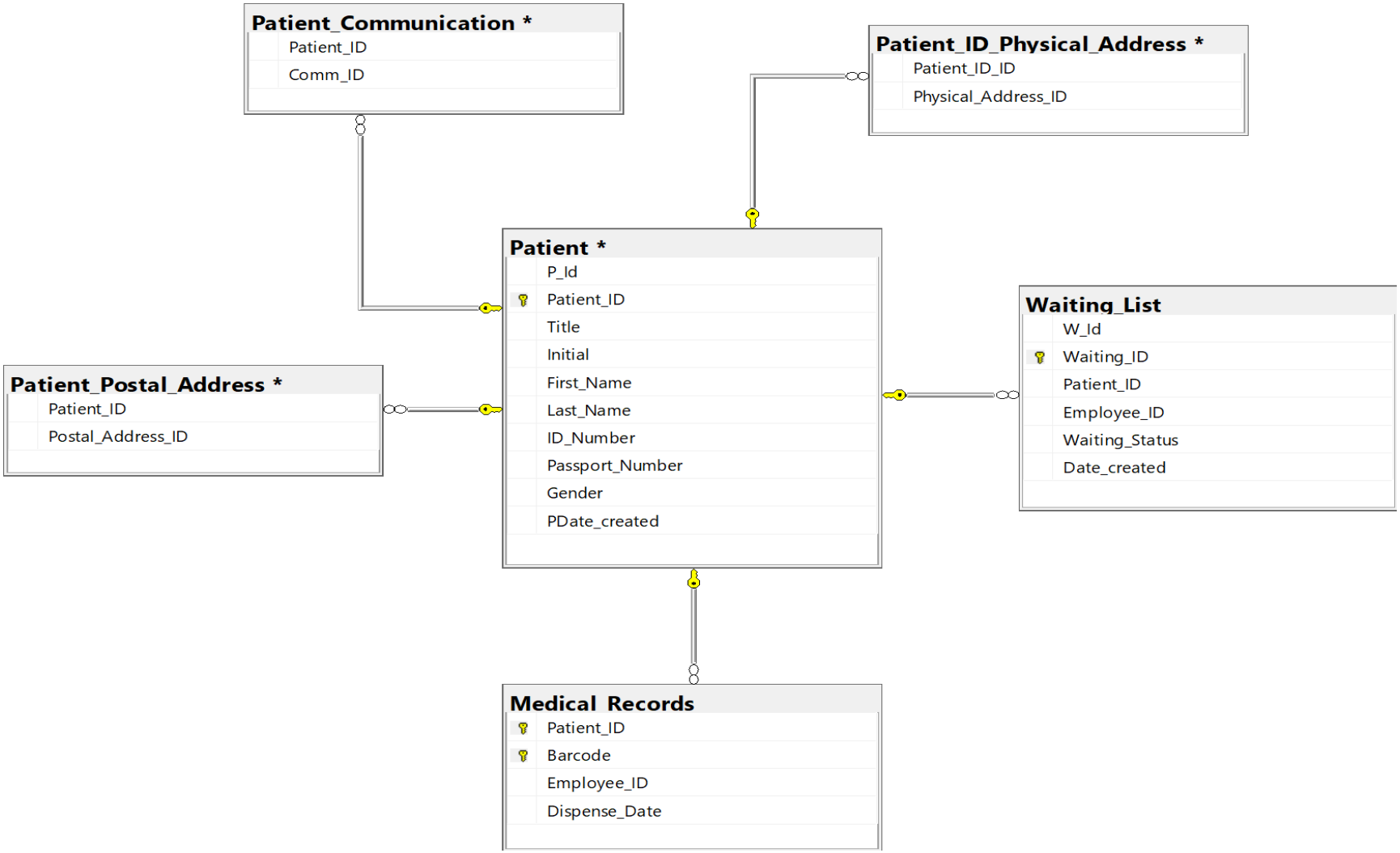


Figure 4.20: ERD 8 of 10

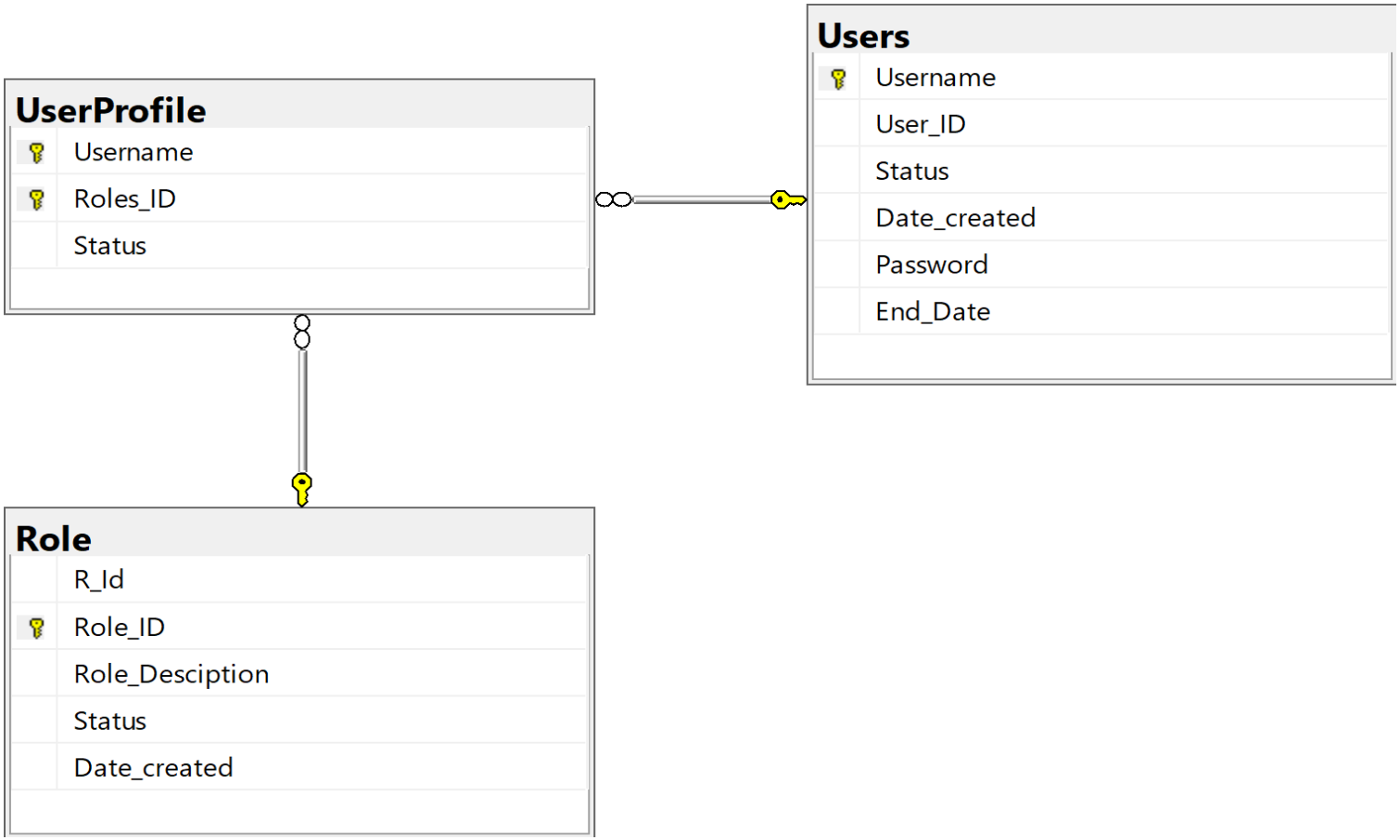


Figure 4.21: ERD 9 of 10

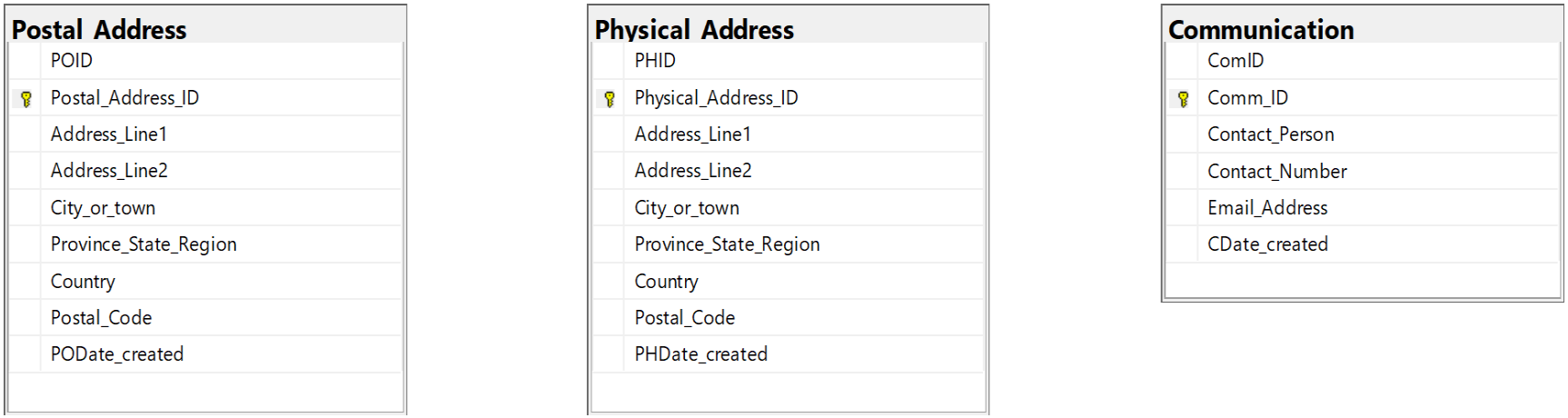


Figure 4.22: ERD 10 of 10

The output of the design stage is the technical requirements that will be used in the implementation stage to develop the functionalities, which are integrated, developed and tested. At the integration and testing stage, all the functionalities developed in the implementation phase are integrated into a system after each functionality has been tested, as discussed in the next section.

4.4 IMPLEMENTATION STAGE

The fourth stage of the waterfall model (Figure 4.1) is the implementation stage. At this stage, the output from the requirements, analysis and design stages are used as input to develop the unified user interface, mobile application screens, Database Management System (DBMS) and coding of the app. The technical specifications are also used to build a cloud environment where the application will be running. The outputs of this stage are the different sub-systems and the DBMS, as shown in Figure 4.23 below.

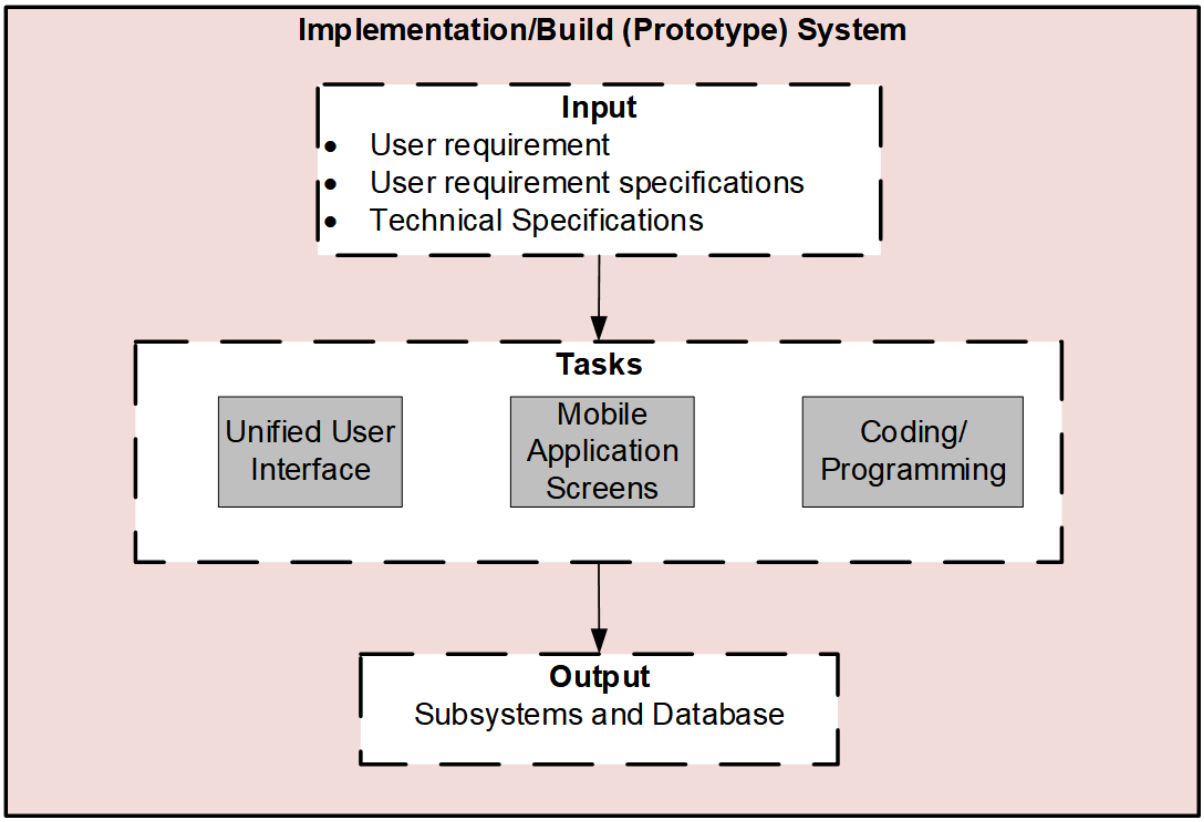


Figure 4.23: Implementation stage

4.4.1 Unified User Interface

According to Wulf and Blohm (2020), the unified User Interface (UI) uses responsive web design principles to provide an optimal viewing and interaction experience for any screen size, device, or orientation. Thov and Quatius Ltd (2016) further elaborate that a unified UI brings all the rich experiences to any device that you are using. Whether you are using a browser, tablet, or phone, you will be able to have a similar experience. The UI re-uses as much as possible of an app without having to redesign or re-code what has already been coded. To achieve this, the app is encapsulated inside the unified UI and is made to look as if it is part of one system (Carter *et al.*, 2016). This is accomplished by hosting the ACT mobile app in a Xamarin shell flyout. Figure 4.24 below shows the core features of the mobile app as explained in the analysis stage. The details of each screenshot are explained in the next section.

Figure 4.24: ACT mobile app high level

4.4.2 Mobile app screens

The Administrator of the system, for example, the Pharmacist, and all the identified stakeholders will use the Login Screen, illustrated in Figure 4.25, to log in to the mobile app. All users must first be verified by the system administrator before being allowed on to the system. According to Park *et al.* (2017), politicians have an interest in the supply of ACTs, therefore, unverified users might compromise the integrity of the data.

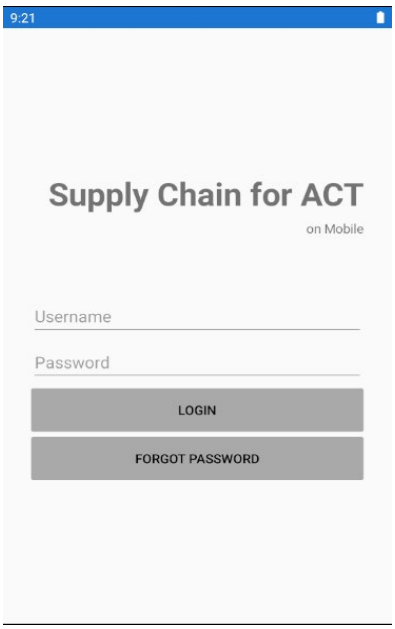


Figure 4.25: Login screen

4.4.2.1 Main menu screen

The main menu is illustrated in Figure 4.26 and the elements discussed below.

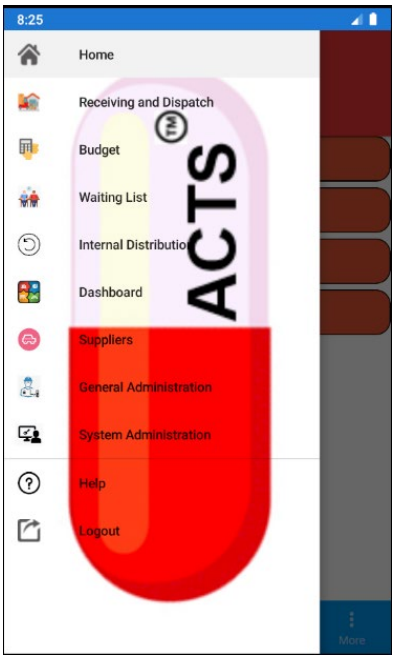


Figure 4.26: Main menu

The functionalities on the main menu have been grouped as follows:

- **Receiving and Dispatch:** The menu will process received drugs from either supplier or donor by registering them into the system. Once registered, the drugs can be distributed to the health facilities that have placed requests.

- **Budget:** The menu will manage and control the budget. According to Nagitta and Mkansi (2019), health facilities cannot request ACT drugs if they do not have the budget.
- **Waiting List:** If the health facility has run out of ACT drugs, a patient can be put on hold.
- **Internal Distribution:** Each health facility has access to the stock level of the other health facilities. It makes it easy for the health facility to check the nearest health facility from which they can request the ACT drugs to address the demand.
- **Dashboard:** The dashboard displays aspects such as key performance indicators (KPI); metrics and key data points to monitor the consumption of drugs; health facilities running out of stock; and delivery to health facilities that have requested ACT drugs.
- **Suppliers:** An Administrator can add and update supplier information. The suppliers do have limited access to the system, for example, they have access to the functionality of registering drugs.
- **General Administration:** An Administrator maintains the following functionalities under General Administration, namely: employees, positions, suppliers, department health facility, drivers and NMS information.
- **System Administration:** An Administrator maintains users, user profiles and the role of the users using the system.

4.4.2.2 Receiving and Dispatch sub-menu screen

The Receiving and Dispatch sub-menu, illustrated in Figure 4.27 below, has four functionalities, namely, 1) Order, 2) Register medicine, 3) Schedule issuance, and 4) Process request by NMS.

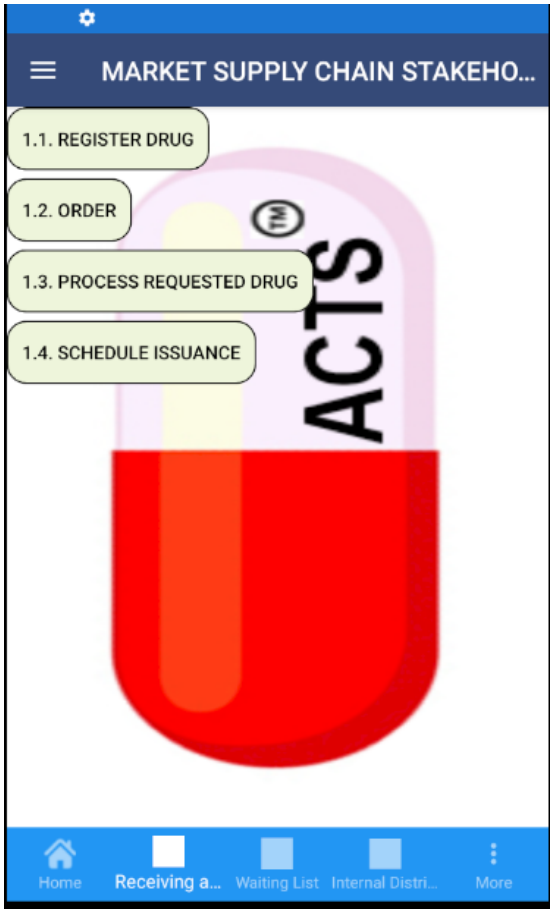


Figure 4.27: Receiving and Dispatch menu

1) The **Order menu,** shown as Point 1.1 in Figure 4.27 above, also has more sub-menus, namely, Receive, Create a new order, and Update existing order.

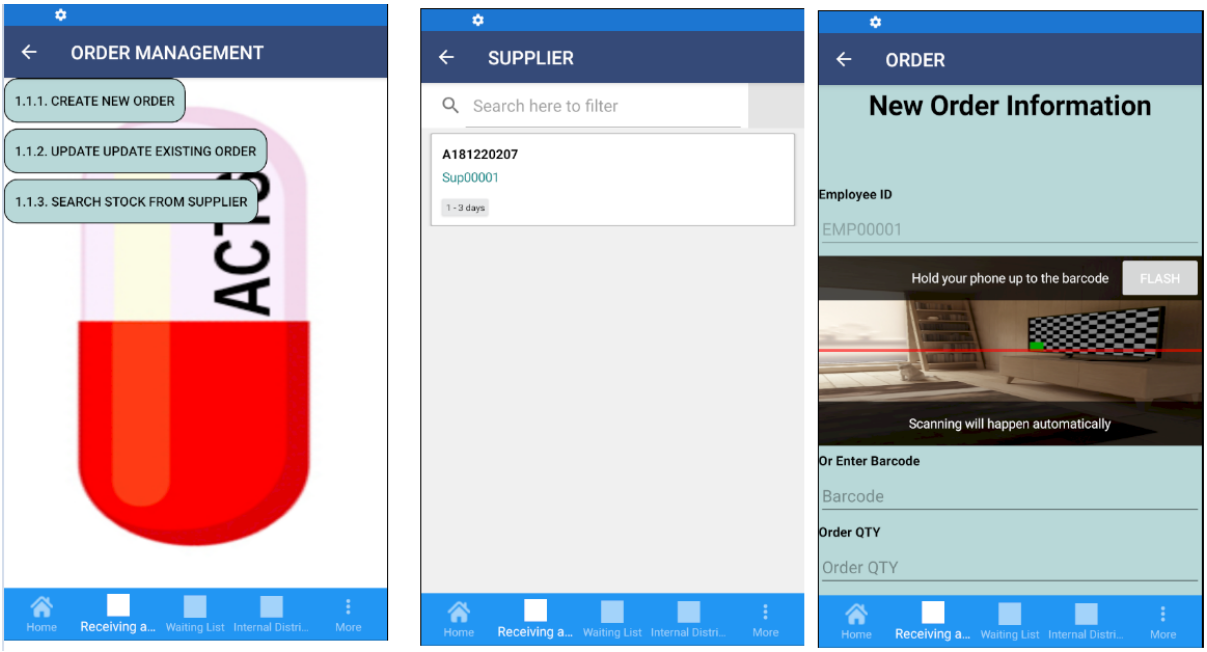


Figure 4.28: Order menu

The **Register Medicine menu,** shown as Point 1.2 in Figure 4.27, has two menus, namely, Register drug, and Update drug (Figure 4.29).

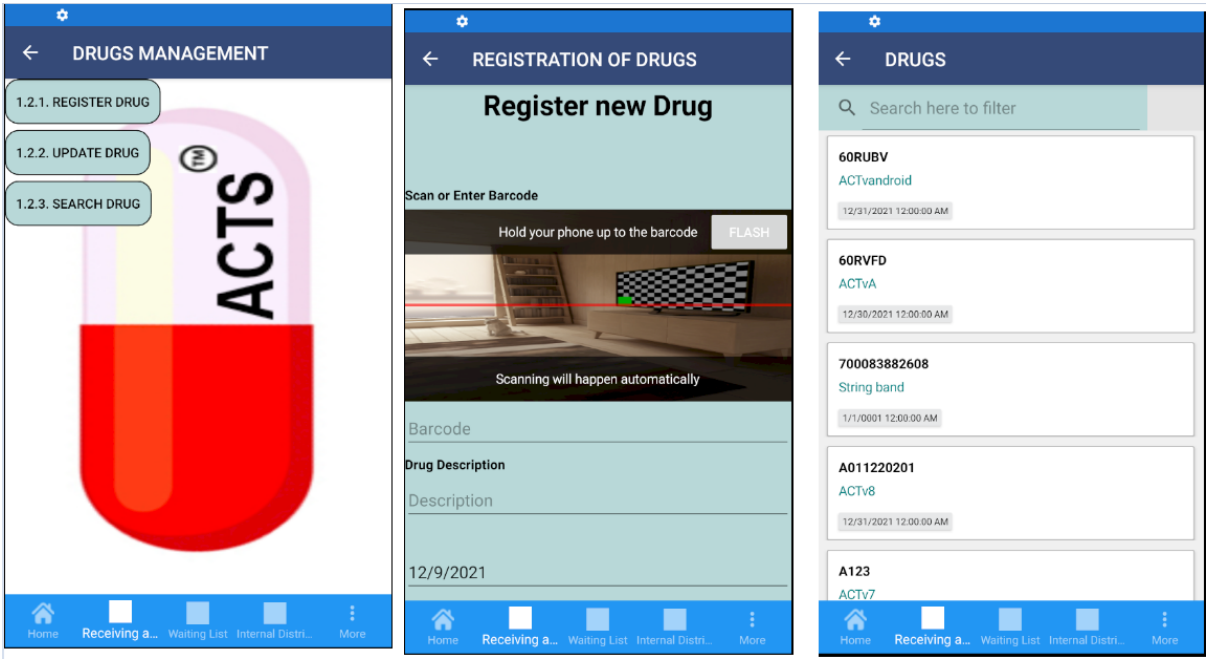


Figure 4.29: Registration of medicine

2) The **Schedule issuance menu,** shown as Point 1.3 in Figure 4.27, has two menus, namely, Create new schedule, and Update existing schedule (Figure 4.30). Once the schedule has been created, users can start to track the request.

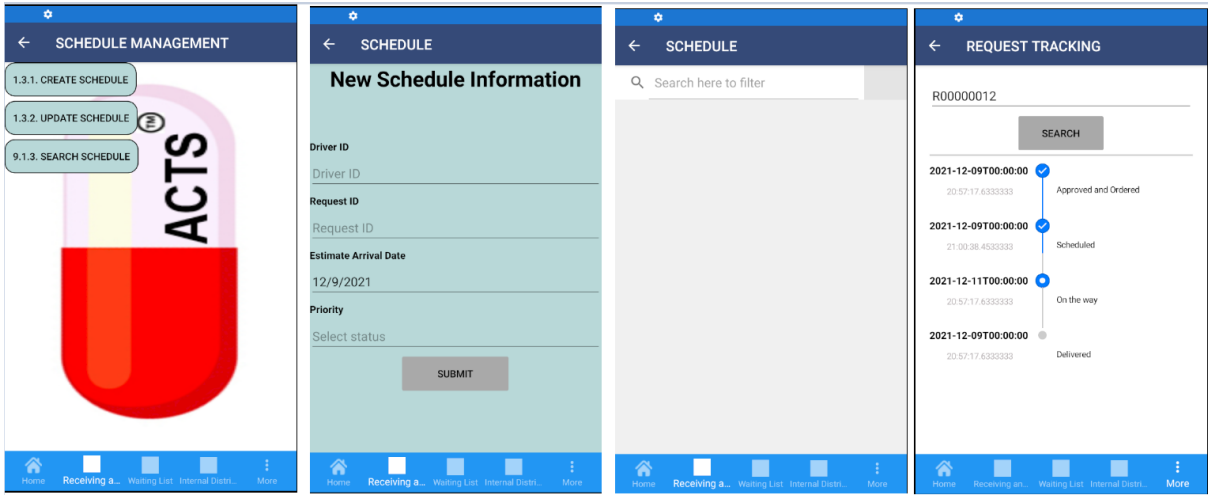


Figure 4.30: Scheduling and tracking of medicine

Once the stock has arrived, the Pharmacist can use the below menu (Figure 4.31) to start dispensing drugs to the patient. The system will only allow the Pharmacist to dispense drugs to patients that are in the system.

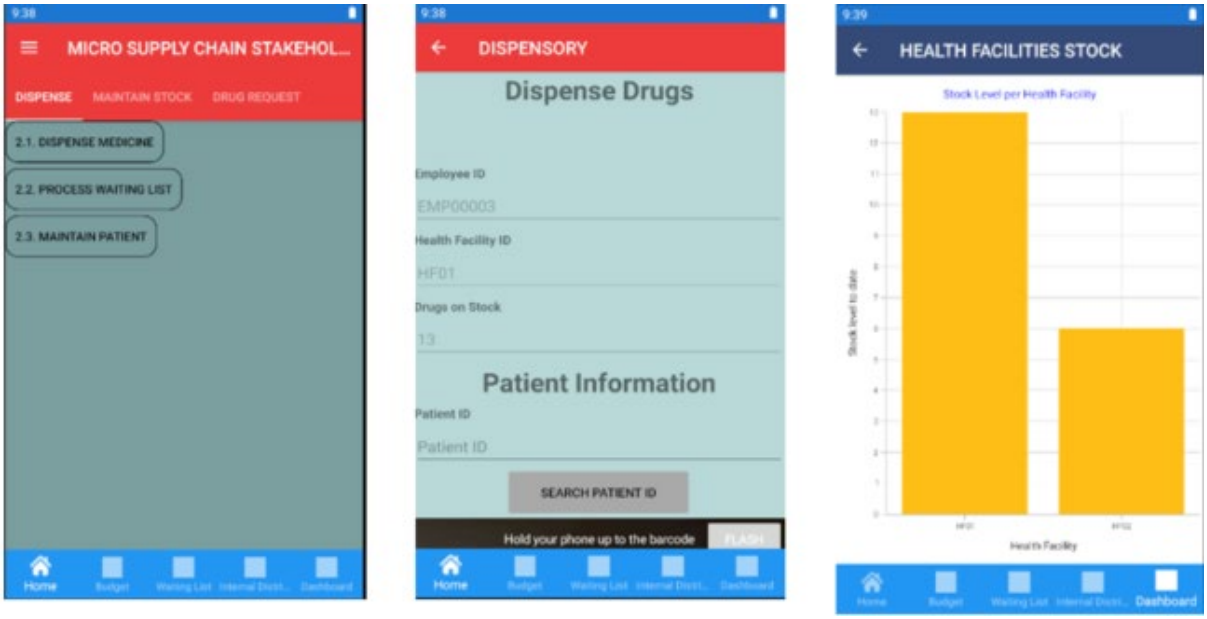


Figure 4.31: Dispensing of drugs

For each health facility, when the graph shown on the screenshot below in Figure 4.32 reaches an order point, the system will send a notification to the Pharmacist to start with the process of requesting more drugs.

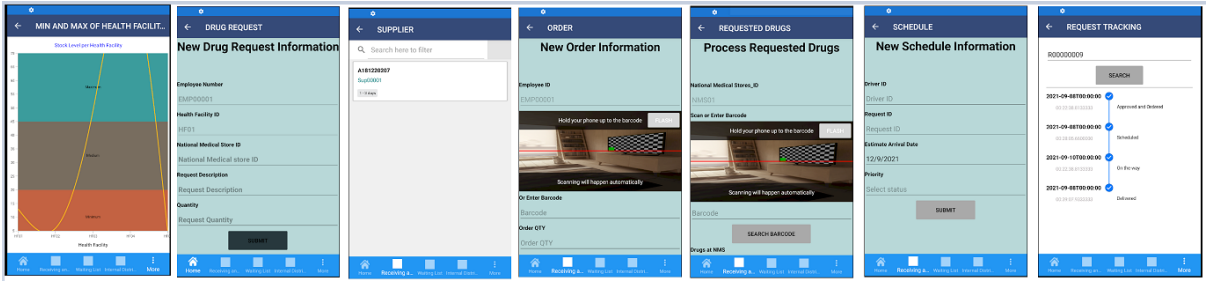


Figure 4.32: Ordering point and requesting new drugs

4.4.3 Programming

According to the TechBeacon (2020), the top tier mainstream programming languages are Java, JavaScript, Python, Ruby, PHP, C#, C++, and Objective-C. The mobile app developed in the current study was written using C#, from back-end code, such as business logic and data access, to native API access because the developer has experience and skills in using C#.

According to Snider (2019), .NET is a developer platform made up of tools, programming languages, and libraries suitable for building many different types of applications. The platform includes the C# programming language and its compilers,

base libraries for working with strings, dates, files/IO, editors and tools for Windows, Linux, macOS, and Docker.

Xamarin is an open-source platform for building modern and performance applications for iOS, Android and Windows with .NET. Xamarin is an abstraction layer that manages the communication of shared code with the underlying platform code. Xamarin runs in a managed environment that provides conveniences such as memory allocation and garbage collection. In addition, Xamarin.Forms is a feature of Xamarin that extends the .NET developer platform with tools and libraries for building mobile apps.

In terms of the current study, the pages, layouts, and controls of the mobile app were built using Xamarin.Forms and the syncfusion library that support Xamarin. To view the code, refer to the app.

Post-implementation and integration, the entire system was tested for any faults and failures, prior to being deployed to the cloud environment or released for users to start using the app, as discussed in depth in Chapter 5.

4.5 CHAPTER SUMMARY

Data was collected using the predictive approach were a formal document was circulated to all identified stakeholders. The document helped to refine the functionalities' names and descriptions and supported the subsequent analysis of gaps, duplications, and redundancies. A template was selected that covered all the characteristics of the functionalities. The functionalities were captured and documented to the micro, macro and market dimensions.

Flowcharts and use cases were used to confirm the requirements, and identify gaps and duplicate information. The researcher was able to understand what the critical logistics, micro, market and macro dimensions were that were associated with the sub-dimensions of ACTs that could be converted into technology features. The following core features were identified: Receiving and Dispatch, Dispense, Order, Internal Redistribution, and Dashboard.

Flowcharts detailed the steps in how the supply of ACTs should flow from the system point of view, while use cases were used to convert all the identified functionalities to

allow the programmer or developer to understand and apply them for the purpose of a test case.

The architecture of the mobile app was driven by digital mesh because it supports the IoT. The logical deployment of a mobile app is divided into three layers, namely, the User Interface, Application, and Data Storage layer. The technology used on the Interface layer was Xamarin Forms, on the Application layer the .Net framework, and Java and C# were used, and the Data Storage layer used the Microsoft SQL Server. The solution will be deployed in the cloud environment.

The User Interface was implemented using unified User Interface (UI) because it uses responsive web design principles to provide an optimal viewing and interaction experience for any screen size, device, or orientation. The programming language that was used to implement the app is C#. The next chapter discusses the integration and testing of the system.

CHAPTER 5:
SYSTEM TESTING

This chapter delves into the testing of the system which design, and development was discussed in chapter 4. There are two fundamental methods for testing software behaviour and performance, namely, white box testing and black box testing (Nidhra & Dondeti ,2012).. Acharya and Pandya (2012) define black box testing as a method that focuses on testing the requirements or specifications of the software entity under test. Khan and Khan (2012) explained that white box testing uses coding experience as part of the testing procedure.

This study used white box testing to test whether the system had been programmed according to the functional requirements, whether all the functionalities were covered, and whether the programs were handling input errors appropriately.

The researcher used a test case to achieve the testing demands of the white box method. According to Kim *et al.* (2016), a test case is a specification of the inputs, execution conditions, testing procedures, and expected results that define a single test to be executed to achieve a particular software's testing objectives.

For software testing that is comprised of many related tasks, each with its own artefacts and outcomes, the development of test cases is a vital step (Arif & Ali, 2019). According to Mendoza and Gu (2018), test cases are a key factor in testing processes because they identify and communicate the conditions that will be implemented in the test, and are necessary to verify the successful and acceptable implementation of the software requirements. Test cases are used to make sure that the software satisfies the requirements of the system. Yoo and Harman (2012) explain that developers create test cases as soon as use cases are available, before coding.

Table 5.1 summarises the testing spectrum of who should perform what type test and the scope of the tests (Khan & Khan, 2012).

Table 5-1: Testing spectrum

Testing type	Methods	Specification	Who will do this testing?	General scope
Unit	White box testing	Low-level design actual code structure	Developer	Component or unit
Integration	White and black box testing	Low and high-level design	Developer	All components or units
Functional	Black box testing	High-level design	Users	Entire application
System	Black box testing	Requirements' specifications	Users	Entire application
Acceptance	Black box testing	Requirements' specifications	Users	Entire application
Usability	Black box testing	Requirements' specifications	Users	Entire application
Performance	Black and white box testing	High-level design	Developer and users	This can be for any of the above

5.1 WHITE BOX TESTING OF MOBILE APPLICATION

The researcher conducted testing at the computer laboratory to test the logic, error handling, flow of the data, and to ensure that all the defined functionalities were working as expected using the test cases. The test cases had predefined inputs and expected results, as outlined in Section 4.1.

5.1.1 Receiving and dispatch testing

As explained in Section 4.2, the registration of drugs is only done at the NMS, therefore, the functionality of 'Register drug' is only available to NMS employees. Drugs are registered in the system before being allocated to the NMS and health facility database.

When the drugs arrive from either the supplier or donor, employee at the NMS check the drugs for compliance before they are registered and allocated to health facilities. The NMS will then dispatch the drugs to health facilities.

The tests discussed below tested the registration of the drug, the allocation to the NMS for reporting, and the dispatch to the health facility.

According to Darce and Quinlan (2011), a barcode, known as code 32, is often used in the health sector. The types of barcode that were used for testing are as follows:

- Code 32, also known as Italian PHARMACODE, is used to identify pharmaceutical products in Italy. It encodes numeric data in a compressed format by using the Code 39 - Regular character set. This symbology is the only type of barcode in common use that does not require a checksum. The PHARMACODE is typically used for in-house inventory control and security verification, ensuring that packaging materials and contents match and are correct. It is not used for point-of-sale purchases, which is ideal for ACT drugs (Grell, 2014).
- The QR-Code is an efficient, two-dimensional (2D) barcode symbology that allows easy encoding of MECARD data, including phone numbers and web URLs.

The number assigned to both barcode is the test date, for example, 01 December 2020, translated to 01122020 (refer to Table 5.2 below). Both Code 32 and the QR code have the same identification number to test the readability of both barcodes.

Table 5-2: Code 32 and QR barcode used for testing

#	Data	Code 32	QR Code
1	A181220207		
2	A151220201		
3	A131220206		
4	A271220206		
5	A211220203		
6	A011220201		

5.1.2 Register drug test case

The objective of the ‘Register drug’ test case is to test if the functionality can capture and save drugs from either suppliers or donors into the National Medical Store (NMS) database.

The test case is demonstrated in Table 5.3 (on the next page).

5.1.3 Allocate to National Medical Store (NMS) test case

The objective of the allocate drugs to NMS test case is to test if the system can add drugs to the NMS.

The ‘Allocate to National Medical Store (NMS)’ test case is demonstrated in Table 5.4.

5.1.4 Request drug test case

The objectives of the request drug test case are to test if the employee number and health facility ID will be automatically displayed on the screen, and if the user will be able to create a request.

The ‘Request drug’ test case is demonstrated in Table 5.5.

5.1.5 Process drug request test case

The objective of process drug request test case is to test if the drug will be deducted from NMS and drug request tracking initiated.

The ‘Track request status’ test case is demonstrated in Table 5.6.

5.1.6 Schedule issuance test case

The objective of the schedule issuance test is to test if the system can assign the request and order to a driver, including the priority.

The ‘Schedule issuance’ test case is demonstrated in Table 5.7.

Table 5-3: Register drug test case

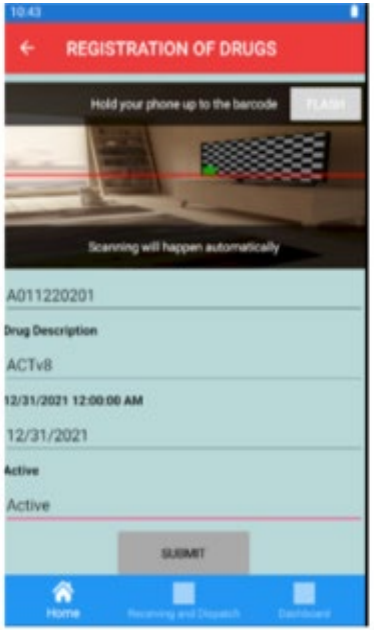
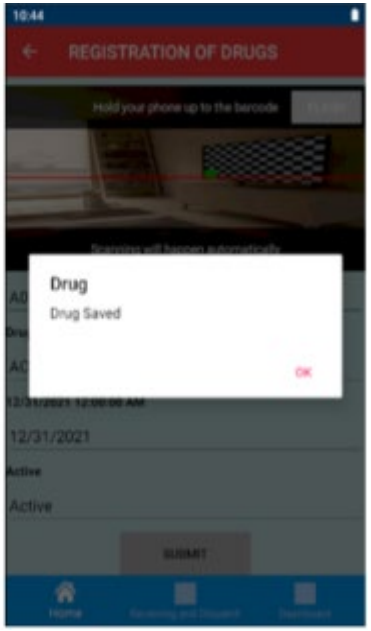
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC1	1.1 Register drug	Registration of drug	A011220201	Registered/Saved drug	Pass
		Registration of drug with missing information	ACT v8		
			Expiry date: 13/12/2021		
			Status: Active		
Screenshots	  				
Dependency	User must have been created, linked to a profile R001 in the system.				
Comments	Barcode can either be scanned or manual type.				

Table 5-4: Allocate to National Medical Store (NMS) test case

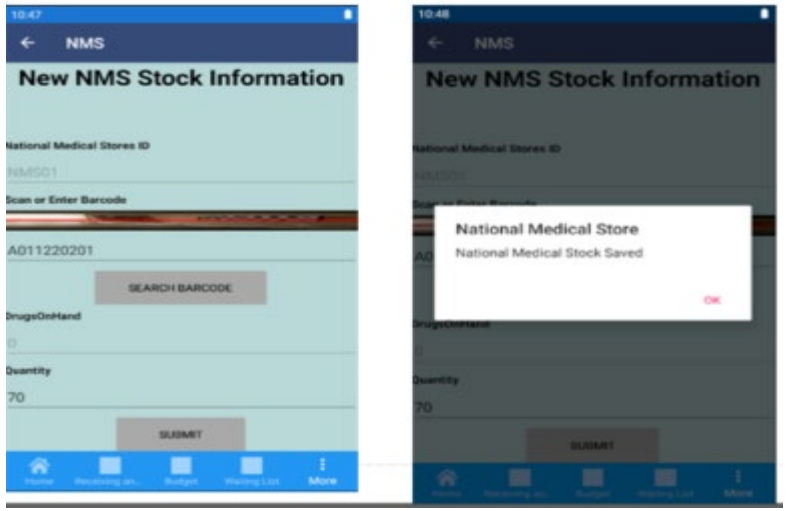
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC2	1.2 Allocate to National Medical Store (NMS)	Add the drug to the NMS		Added/Saved stock	Pass
		Add unregistered drug to the NMS	NMS automatically added	Not added/Saved	Pass
			Drugs on hand automatically added after searching barcode. Drugs on hand: 0. The system checks the barcode linked to automatically display the NMS ID.		
			Quantity: 70		
			a (to test unregistered drug)		
Screenshots					
Comments	An employee cannot add the stock if not linked to the NMS. User cannot enter drugs on hand.				

Table 5-5: Request drug test case

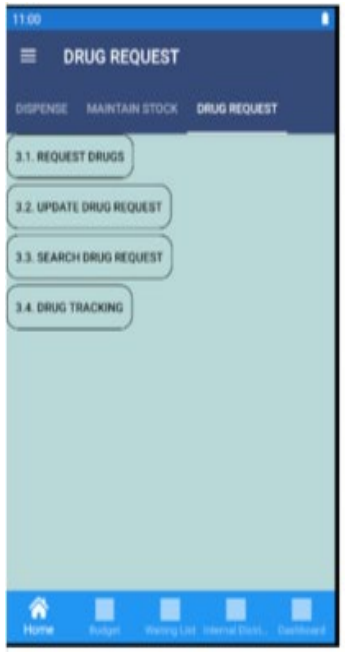
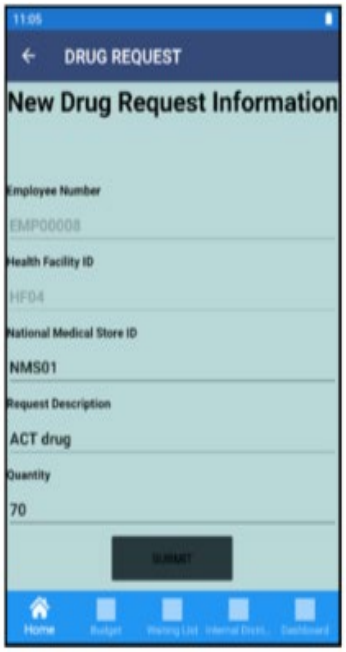
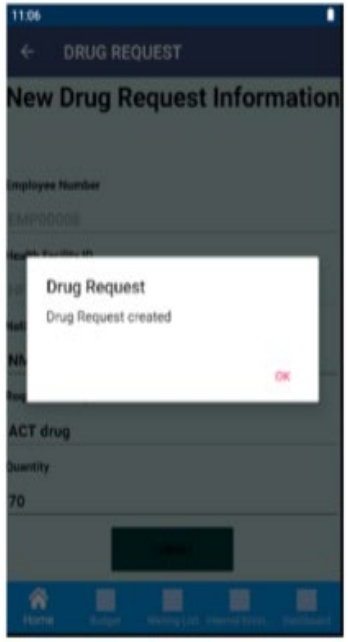
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC3	2.1 Request drug	Create a request	Employee Number: EMP00008 and Health Facility ID: HF01 linked with the employee automatically added Request (ACT drug) Quantity: 70	Request Created/Saved	Pass R00000009
Screenshots	   				
Comments	The order can only be placed if NMS does not have stock				

Table 5-6 Process Drug Request

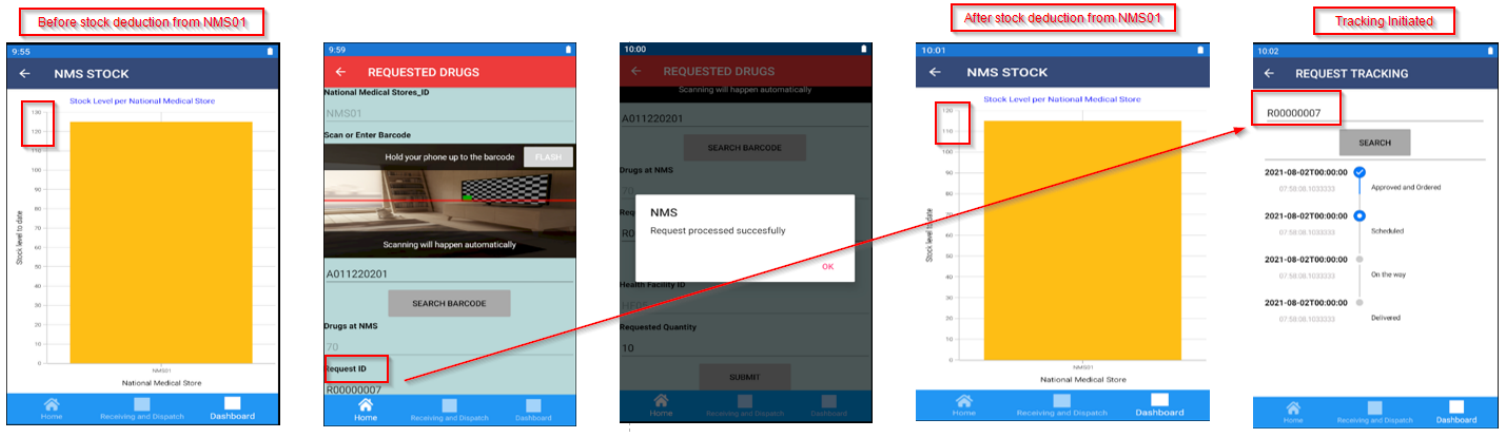
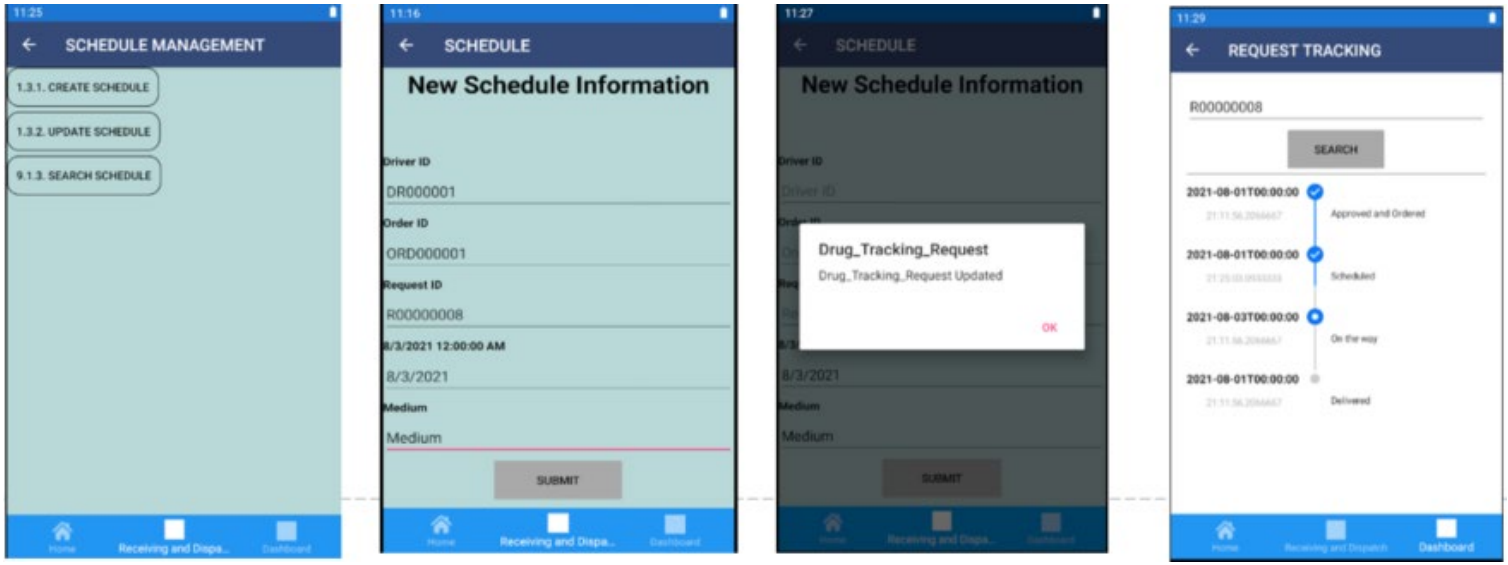
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC4	3.1 Process drug request	Process requested drug	National Medical Store: NMS01 and Health Facility ID: HF05	Stock deducted from NMS and Request drug tracking initiated	Pass
			Quantity: 10		
			Request ID: R0000007		
Screenshots					
Comments	The order can only be placed if NMS does not have stock. Request tracking created automatically with approved request status				

Table 5-7: Schedule issuance test case

Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail

TC4	1.3 Schedule Issuance	Schedule Issuance	Driver ID: DR000001 Order Number: ORD000001 Request ID: R00000008 Estimate delivery date: 08/03/2021 Priority: Medium	Schedule created/Saved	Pass
Screenshots					
Comments	When schedule created, drug tracking updated automatically.				

5.2 DISPENSE TESTING

Dispense testing covers the process of receiving the stock from the NMS and checking the stock levels.

5.2.1 Maintain health facilities test case

The objective of the maintain health facility test case is to process the drugs coming from the NMS by adding them into the health facility's database.

The 'Maintain health facilities' test case is demonstrated in Table 5.8 (on the next page).

5.2.2 Dispense drug test case

The objective of the dispense test case is to test if the employee's and health facility's IDs automatically appear on the screen before dispensing drugs to the patient, and also to test the dispensing of drugs to the patient. Once the drug has been dispensed to the patient, the user views if the graph reflects the updated stock levels.

The 'Dispense drug' test case is demonstrated in Table 5.9.

Table 5-8: Maintain health facilities test case

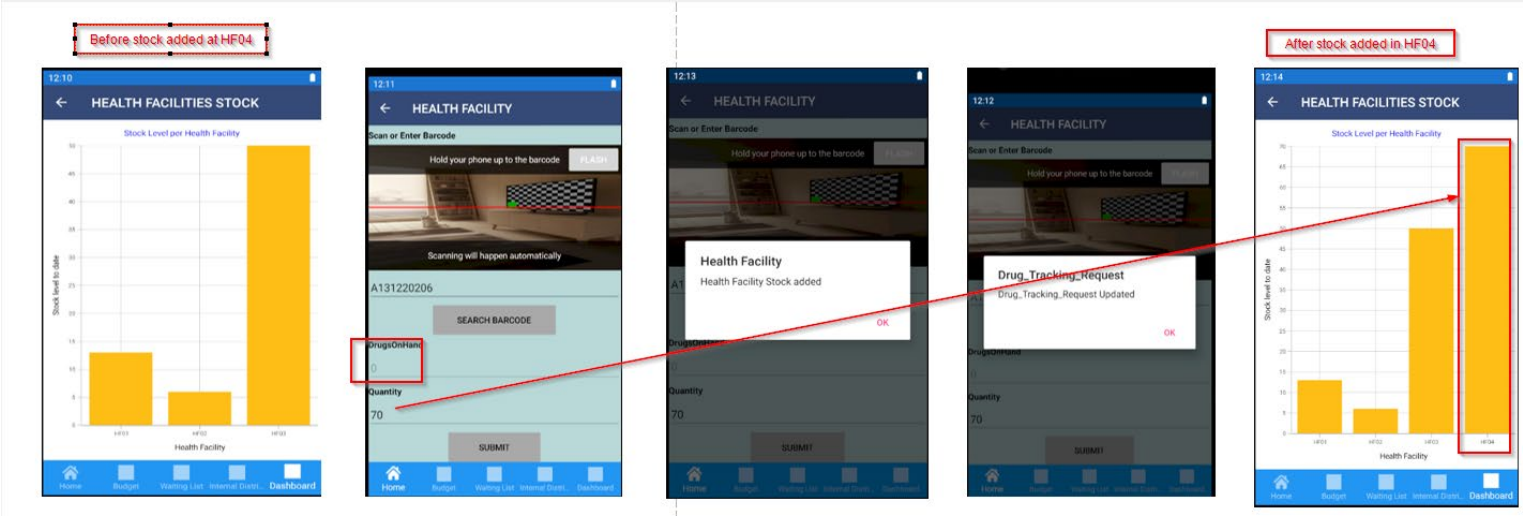
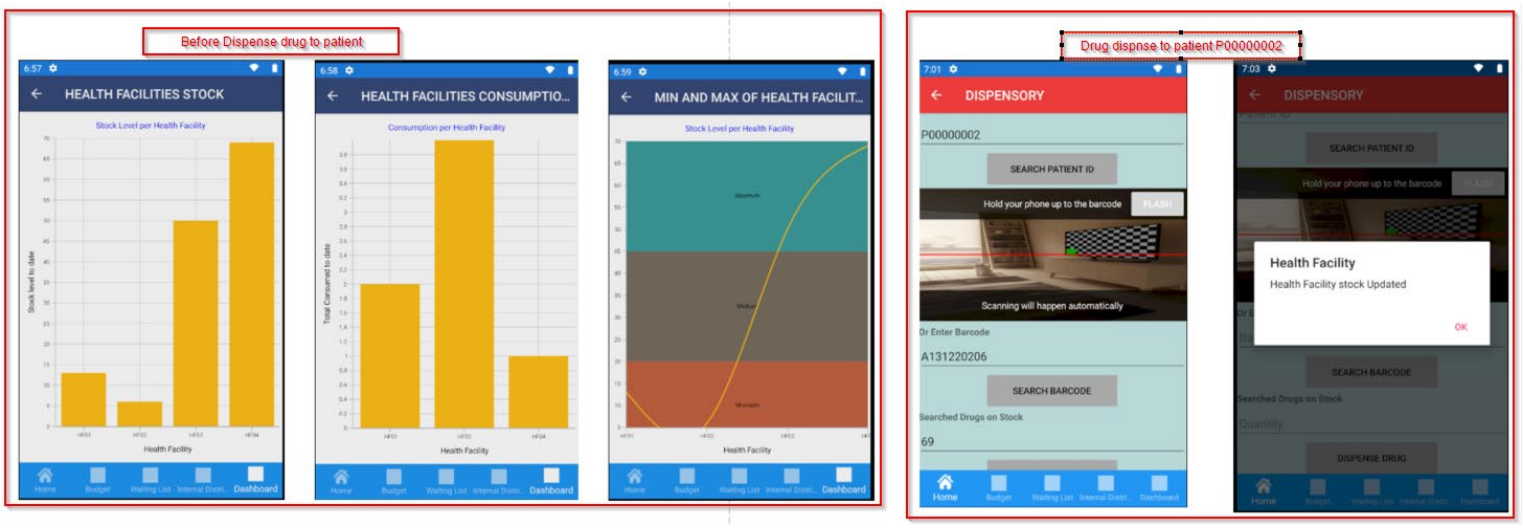
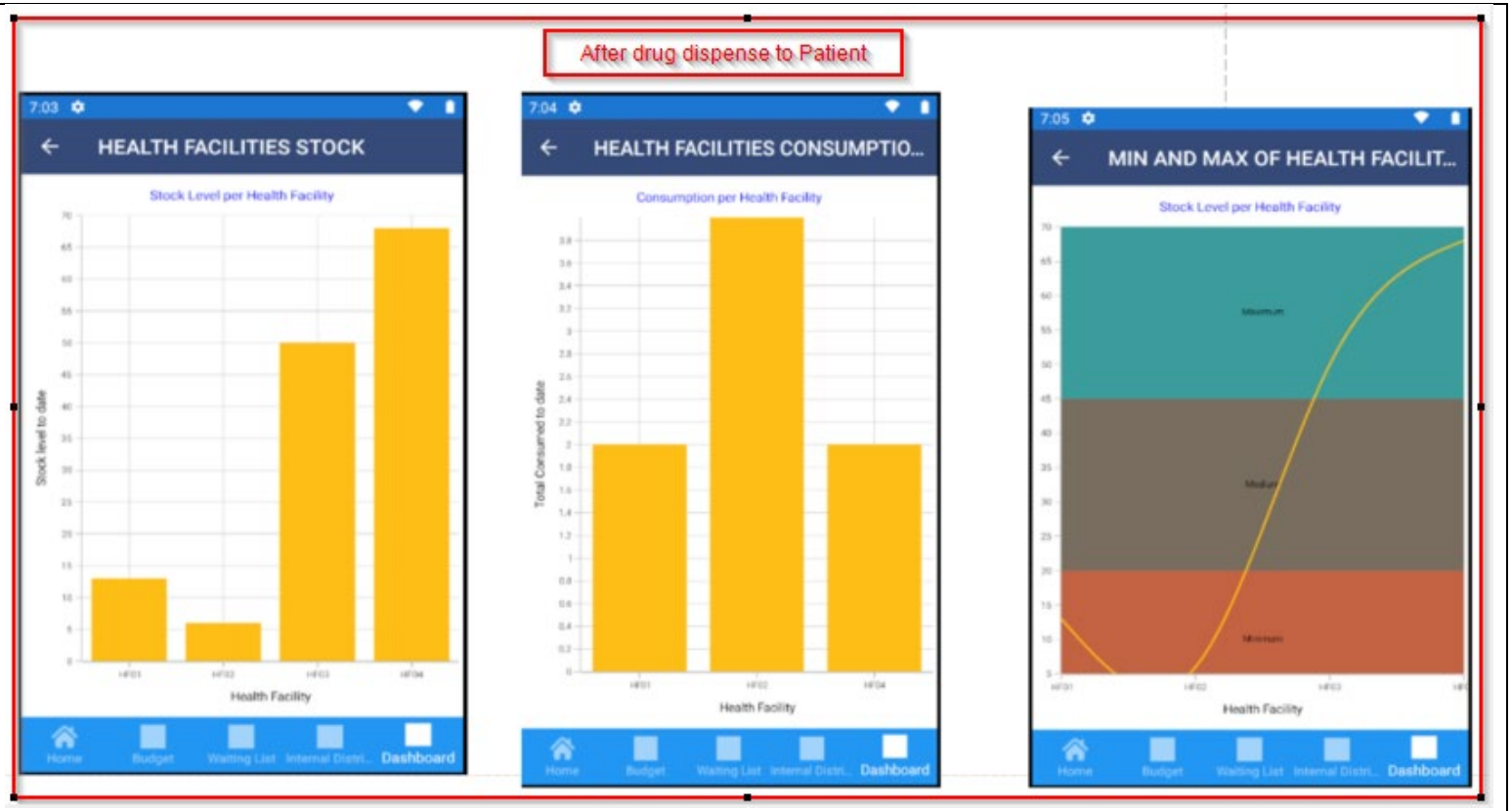
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC6	10.6 Maintain Health Facilities	Received the requested drug and add to health facility	Health Facility ID: HF04 (Automatically inserted)	Stock added/Saved	Pass
			A131220206		
			Quantity: 70		
Screenshots					
Comments	The statuses are automatically updated at four stages, namely: when the order is approved, when the schedule is created, when the driver picks up the request, and when the health facility receives the drug. The graph shows that HF01 already has stock, hence the 70 added increased the graph.				

Table 5-9: Maintain health facilities test case

Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC7	3.1 Dispense Medicine	Dispense drug to Patient	Employee ID: EMP00007 (Automatically inserted on user login)		
			Health Facility ID:HF04 (Automatically inserted)	Stock level added reduced by 1, and consumption increased by 1	Pass
					
			Patient ID: P00000002		

Screenshots	

	
Comments	HF04 has 69 drugs available on starting the process. When searching the barcode, the system automatically displays the total number of drugs for that particular barcode, which currently is at 69.

5.3 SECURITY TESTING

The security test case tests when a user login can only be viewed as per the user profile. This has been archived by using roles. System roles control what sections in the system are available to different types of users. The ACT mobile app has five different types of roles, namely, Administrator, Driver, Micro Supply Chain Stakeholders, Market Supply Chain Stakeholders and Macro Supply Chain Stakeholders.

The **Administrator** has access to all functionalities in the system. The **Driver** has access to the tracking and dashboard functionality. The **Micro Supply Chain Stakeholders** have access to reporting (Dashboard) and are able to request drugs on behalf of the health facility. The **Market Supply Chain Stakeholders** have access to register and distribute drugs, including ordering. The **Macro Supply Chain Stakeholders** have access to issue drugs to the patients and are able to maintain stock levels.

The following role IDs and role descriptions were created for the case test.

Role_ID	Role_Description
R001	Administrator
R002	Driver
R003	Micro Supply Chain Stakeholders
R004	Market Supply Chain Stakeholders
R005	Macro Supply Chain Stakeholders

The following user profiles, linked to the role IDs, were created for the case test.

Username	Password	Role_ID
Jane	test	R001
Reymond	test	R002
Judith	test	R003
Tshililo	test	R004
Sandra	test	R005

5.3.1 Login test case

The objective of a login test case is to test if the user can access only that sub-system to which the user is provided permission. The ‘Login’ test cases for Administrator, Driver, Micro Supply Chain Stakeholders, Market Supply Chain Stakeholders and Macro Supply Chain Stakeholders are demonstrated in Table 5.10 to 5.14 (on the next page).

5.3.1.1 Administrator test case

The Administrator manages and administrates the entire system, including providing support. The objective was to test and view if the Administrator has access to all the functionalities of the system.

The ‘Login’ test case is demonstrated in Table 5.10 (on the next page).

5.3.1.2 Driver test case

The Driver only tracks the delivery of the drugs. The objective of this test case was to test if the driver has view functionality to only the track drug request.

The ‘Driver’ test case is demonstrated in Table 5.11.

5.3.1.3 Micro Supply Chain Stakeholders test case

Micro supply chain stakeholders are the Hospital, Health centre and clinic. The core functionality is dispensing drug to patients and maintaining stock levels. The objectives of this test case was to test if the micro supply chain stakeholders have access to only the functionalities for dispensing drugs, requesting drugs and maintaining stock levels.

The Micro Supply Chain Stakeholders test case is demonstrated in Table 5.12.

5.3.1.4 Market Supply Chain Stakeholders test case

Market supply chain stakeholders are the Suppliers, Manufacturers, Distribution Centre, Pharmaceutical and National Medical Store for Uganda. The market core functionalities are the registration and distribution of drugs, including the ordering functionality. The objective of this test case was to test if the market stakeholders have access only to the functionalities for the registration and distribution of drugs, including ordering and processing the new request.

The Market Supply Chain Stakeholders test case is demonstrated in Table 5.13.

Table 5-10: Administrator test case

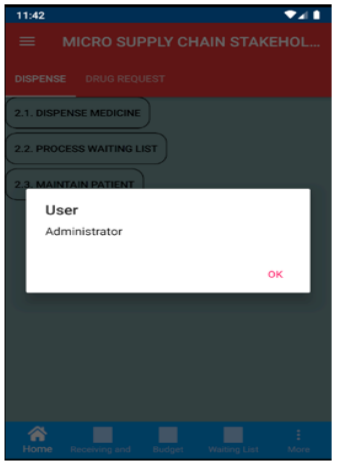
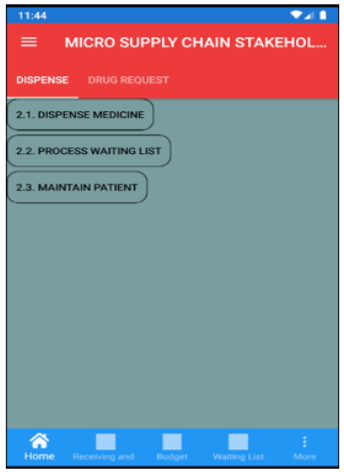
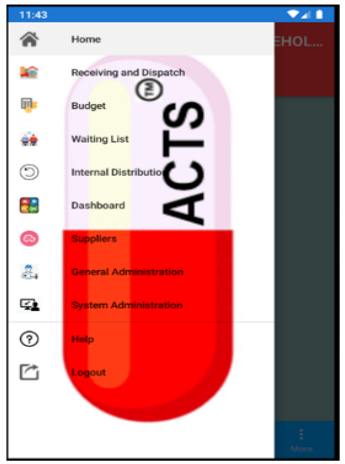
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC8	9.4.1 Login as an Administrator	Users can view according their role assigned on the users profile	Username and roles Jane :R001	View all functionalities assigned to Administrator	Pass
Screenshots	   			
Comments	Administrator has access to the whole system. Username and password are case sensitive.				

Table 5-11: Driver test case

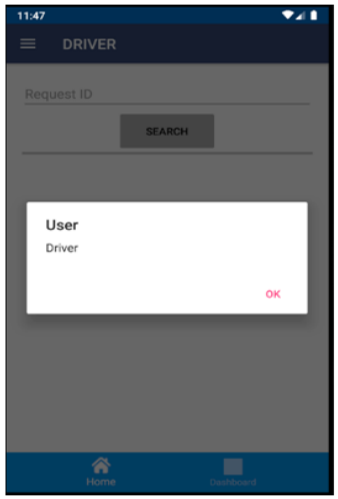
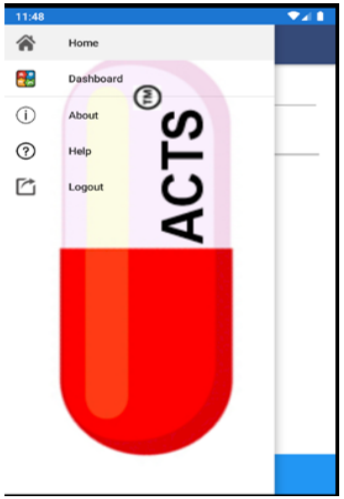
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC9	9.4.2 Login as a Driver	Users can view according their role assigned on the users profile	Username and roles Reymond: R002	View only functionalities assigned to Driver	Pass
Screenshots	   				
Comments					

Table 5-12: Micro supply chain stakeholder test case

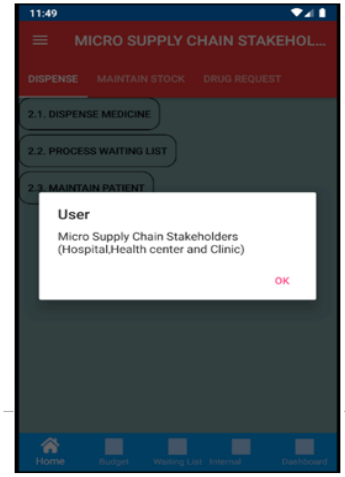
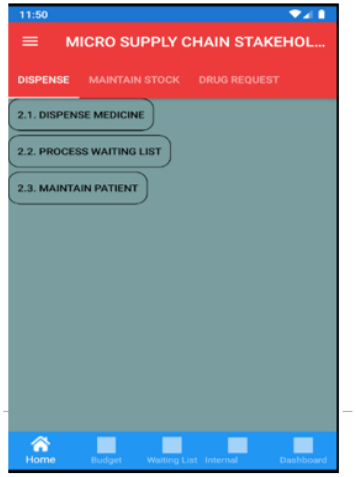
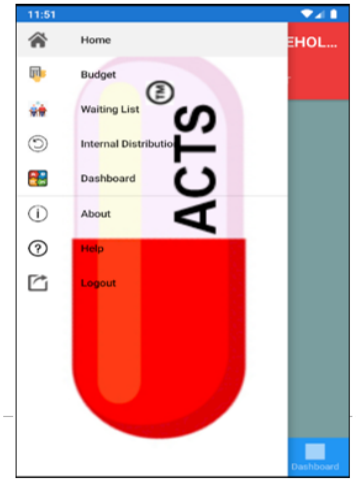
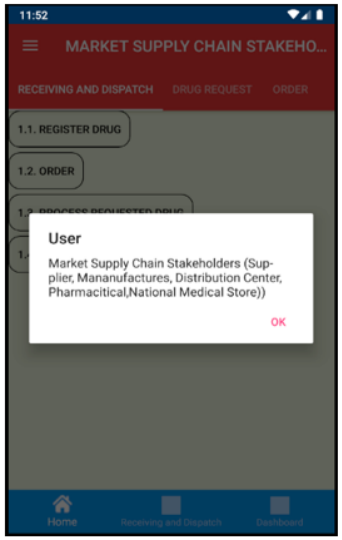
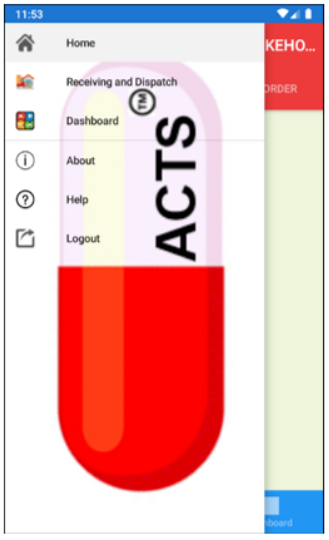
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC10	9.4.3 Login as Micro Supply chain stakeholder	Users can view according their role assigned on the users profile	Username and roles Judith: R003	View only functionalities assigned to micro stakeholders	Pass
Screenshots	   				
Comments					

Table 5-13: Market supply chain stakeholder test case

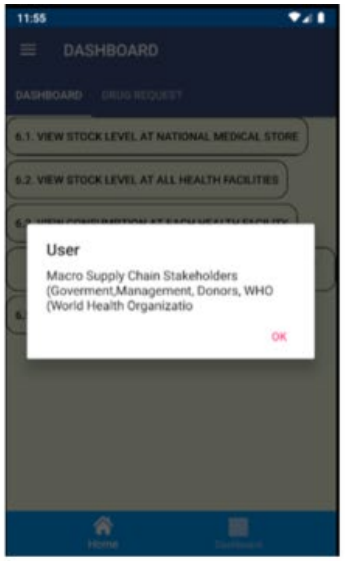
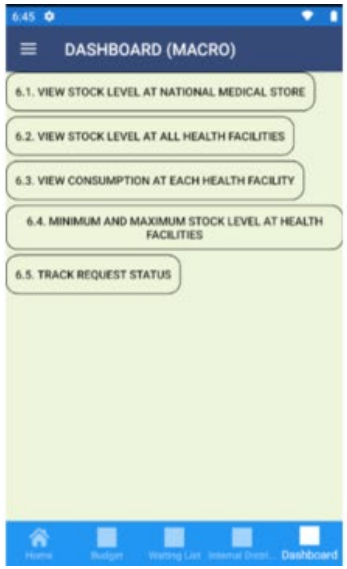
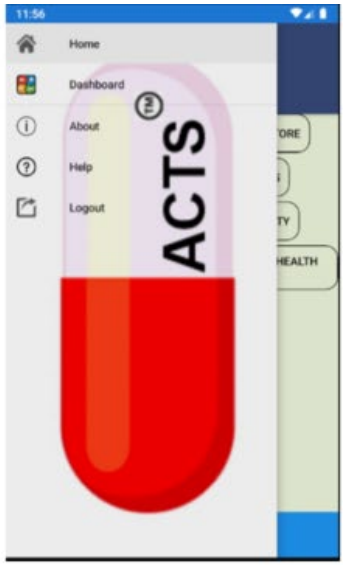
Test Case ID	Use case	Scenario/Condition	Data	Expected results	Pass/Fail
TC11	9.4.4 Login as a Market Stakeholder	Users can view according their role assigned on the users profile	Username and roles Tshililo: R004	View only functionalities assigned to Market stakeholders	Pass
Screenshots	   				
Comments					

5.3.1.5 Macro Supply Chain Stakeholders Test case

Macro supply chain stakeholders are the Government, Management, Donors and the WHO. The core functionalities are reporting and requesting drugs on behalf of the health facilities. The objective of this test case was to test if the macro stakeholders have access to only the functionalities for viewing reports or the dashboard and drug requests.

The Macro Supply Chain Stakeholders test case is demonstrated in Table 5.14.

Table 5-14: Macro supply chain stakeholders test case

Test Case ID	Use Case	Scenario/Condition	Data	Expected Results	Pass/Fail
TC12	9.4.5 Login as Macro Supply chain stakeholder	Users can view according their role assigned on the users profile	Username and roles Sandra :R005	View only functionality assigned to only macro supply chain stakeholder	Pass
Screenshots	   				
Comments					

5.4 CHAPTER SUMMARY

The use case methodology was used to develop test cases. The mobile app that was developed for the current study was tested to ensure that all the functionalities were integrated and working together.

The testing covered receiving and dispatching, dispensing and reports. The security functionality was also tested to ensure that roles and responsibilities work as defined.

The outcomes of the test confirmed that the mobile app was integrated and all tested functionalities were working as expected.

CHAPTER 6:
FINDINGS, RECOMMENDATIONS AND CONCLUSION

This chapter summarises the findings from chapter 5 testing in accordance with the key parameters of the literature, and discusses how the research questions assisted the researcher in the current study to achieve the research objectives. This study extends the theoretical findings related to the supply chain coordination framework proposed by Nagitta and Mkansi (2019) by developing a mobile application. The mobile application was developed using the IoT (IoT) model and the system development life cycle methodology. The next section presents a summary of the findings related to each objective, followed by the recommendations, limitations and conclusion.

6.1 RESEARCH OBJECTIVES AND QUESTIONS

The main objective of the research study was to develop a mobile application (mobile app) for the supply chain coordination of ACT drugs. This main objective was further supported by several secondary objectives.

6.1.1 Research objectives

In order to achieve the main objective, the following secondary research objectives were formulated:

1. To identify the critical logistics, micro, market and macro dimensions, and the associated sub-dimensions of ACTs that can be converted into technology features.
2. To map the features of ACT software for stakeholders across the micro, market and macro environments.
3. To apply the principles of the IoT and technologies to develop the software, and to link stakeholders across the micro, market and macro environments.

6.1.2 Research questions

1. What are the critical logistics, micro, market, and macro dimensions, and the associated sub-dimensions of ACTs that can be converted into technology features?

2. What are the features of ACT software that are associated with the micro, market, and macro stakeholders?

6.2 SUMMARY OF FINDINGS

This section presents a summary of the finding obtained from the literature review, and describes how the objectives of the research study were met.

6.2.1 Summary of literature review findings

The lack of integrated Information Technology systems has been cited as a challenge in the supply of ACT drugs, causing the overstock and understock of ACTs. The framework developed by Nagitta and Mkansi (2019) provided the different dimensions that are critical for the coordination of ACTs across the various stakeholders in the micro, market, macro and logistical environments. The above-mentioned researchers recommended the development of a mobile application that will provide real-time monitoring of stock levels, and improve the end-to-end distribution of ACT drugs.

The identified stakeholders from the micro, macro, and market environment include top management, suppliers, and so on. The current study adopted the IoT principles and technologies to develop a mobile solution that will encompass all the stakeholders. The IoT technology will be able to provide real-time monitoring of activities such as the delivery of stock, stock levels, and consumption levels. The current study extended the findings of Nagitta and Mkansi (2019) by developing a mobile application, which was the main research objective of the study. The findings related to the objectives of the study are summarised in the next section.

6.2.2 Summary of findings related to research objectives

The three research objectives were all met and produced satisfactory performance during testing and deployment. Below is a summary of the findings related to each objective:

Objective 1: To identify the critical logistics, micro, market and macro dimensions, and the associated sub-dimensions of ACTs that can be converted into technology features.

In the requirement stage of the waterfall model (Figure 4.1), the framework of Nagitta and Mkansi (2019) was used to gain an understanding of how the coordination of

supply chain interdependencies at the micro, market, logistics and macro levels impact the availability of ACTs in general hospitals in Uganda. Confirmatory factor analysis and multi-decision criteria assisted to reveal the critical supply chain coordination sub-dimension factors from the micro, macro, logistics and market environment, which called for technological attention.

Some of the functionalities that were identified did not require a technological solution. Below is a summary of the identified functionalities categorised according to micro, macro, logistics and market, that were elicited by the key stakeholders and confirmed following the conversion of requirements into functional tasks:

- **Micro**

  a) Demand Report
  b) Stock Status Report
  c) Active Orders Report
  d) Instructions on use of ACTs
  e) SMS notifications to patients
  f) Information on Stock Status Report
  g) Notice boards (Email, SMS)
  h) Internal Redistribution
  i) Feedback loop with other health facilities, including suppliers
  j) Scheduled issuance
  k) Supplier schedule
  l) Internal transfers of ACTs
  m) Placement of emergency orders
  n) Centralised System

- **Macro**

  a) Compliance with dosage (Business Rule on the functionality)
  b) Dosage Information (SMS/Email)
  c) Integrated mobile application (part of the non-functional requirement)

- **Logistics**

  a) Monthly consumption
  b) Maximum–minimum stock levels

c) Information from the dispensing logs
d) Malaria seasons, based on History Report
e) Peak times, based on History Report
f) Consumption Report
g) Estimating the average monthly consumption
h) Receive order or donation
i) Register drugs
j) Order drugs

- **Market**

a) Visibility of information to all regions
b) Online sharing of information

Objective 2: To map the features of ACT software for stakeholders across the micro, market and macro environments.

Mapping of the features was achieved through the activities of the second (Analysis) and third (Design) stages of the waterfall development method. At the Analysis stage, flowcharts were used to describe the sequencing of the actions needed to generate the output, and flowcharts were also used to identify the gaps and any duplication of functionalities. The flowcharts assisted to group and map related use cases to create the relevant sub-systems.

The use case methodology was used to show the sequence of steps that encompass the interaction between a user and a system. The researcher was able to develop a test case using the use cases.

At the Design stage, the IoT principles were applied and assisted in identifying the three logical layers, namely, the User Interface Layer, Application Layer, and Data storage layer, where the mobile application would be deployed. The aim of this was to ensure that each layer was completely independent, which makes it easy for cloud deployment and provides the flexibility of a hybrid cloud solution. The mobile application would physically be deployed on the cloud in response to the IoT model and principles.

Objective 3: Apply the IoT principles and technologies to develop the software, and link stakeholders across the micro, market and macro environments.

The last objective of the study was achieved through the fourth (Implementation), fifth and sixth (Integration and Testing) stages of the waterfall development methodology. The Xamarin technology was used to develop the user interface, together with the unified User Interface (UI) principles that provide an optimal viewing and interaction experience, suitable for any screen size, device, or orientation.

In the last two stages of integration and testing, the developed sub-systems were integrated to allow seamless testing. All the sub-systems that were tested passed the tests, but the size of the application is big, especially since it is going to be deployed to a mobile device.

6.3 LIMITATIONS OF THE STUDY

Not all smartphones have the capability to scan barcodes using the camera. In addition, another limitation of the mobile applications is that most countries do not have a public cloud region or a major cloud provider within their country or geopolitical borders. Most cloud providers are for-profit entities, therefore, they will not go through the effort of building out an entire zone if there will be no net revenue gain over time.

In practicality, this means that these metro area clouds will pop up in the major population and commerce centres that are already bustling with potential workloads and customers to capture, which will create a limitation in the access to cloud storage in a remote area, especially in African countries.

6.4 RECOMMENDATION FOR FUTURE WORK

This study was mainly computer laboratory-based, and excluded a post-development study to assess the impact. The study recommends that future studies should conduct surveys to assess the user understanding of the functionalities that were developed, and to get an indication of the technology acceptance level and impact on the health care supply chain. The testing of the mobile app with selected users in a real working environment will reveal the impact of the system in averting overstock and understock of ACT drugs.

6.5 CONCLUSION

The study aimed to develop a mobile application (mobile app) for the supply chain coordination of ACT drugs. The functionalities were identified using the framework developed by Nagitta and Mkansi (2019). Through interaction with the identified stakeholders, the study was able to identify the critical supply chain coordination sub-dimension factors from the micro, macro, logistics and market environments that called for technological attention.

The functionalities were analysed at the analysis stage of the study. At this stage, flowchart diagrams were used to remove any duplication of functionalities, and to help the users to understand how the information will flow. The use case methodology was used to create sub-systems.

At the design stage, the IoT principles were used to design the logical architecture. The physical architecture was designed based on the logical architecture and the cloud model.

The system was developed using the following Microsoft stack, namely, Xamarin for the user interface, C# for logic, ASP.net for the Application Programming Interface (API), and SQL Server for the Database Management System (DBMS).

The initial test was conducted by the developer to ensure that all sub-systems are integrated and responsive to one another.

REFERENCES

Abdelrahman, O.H. 2015. Analysis of the University of Khartoum Library System Using the Structured Systems Analysis and Design Methodology (SSADM). *Cybrarians Journal*, 38.

Acharya, S. & Pandya, V. 2012. Bridge between Black Box and White Box–Gray Box Testing Technique. *International Journal of Electronics and Computer Science Engineering*, 2(1): 175-185.

Adolph, S., Hall, W. & Kruchten, P. 2011. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4): 487-513.

Afif, A.N., Noviyanto, F., Sunardi, S., Akbar, S.A. & Aribowo, E. 2020. Integrated application for automatic schedule-based distribution and monitoring of irrigation by applying the waterfall model process. *Bulletin of Electrical Engineering and Informatics*, 9(1): 420-426.

Alaeddini, M. & Salekfard, S. 2013. Investigating the role of an enterprise architecture project in the business-IT alignment in Iran. *Information Systems Frontiers*, 15(1): 67-88.

Alicke, K., Rexhausen, D. & Seyfert, A. 2017. *Supply Chain 4.0 in consumer goods*. McKinsey & Company.

Aloi, G., Caliciuri, G., Fortino, G., Gravina, R., Pace, P., Russo, W. & Savaglio, C. 2017. Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. *Journal of Network and Computer Applications*, 81: 74-84.

Almalki, S. 2016. Integrating Quantitative and Qualitative Data in Mixed Methods Research--Challenges and Benefits. *Journal of education and learning*, 5(3): 288-296.

Almeida, F. & Simões, J. 2019. Moving from waterfall to agile: Perspectives from IT Portuguese companies". *International Journal of Service Science, Management, Engineering, and Technology*, 10(1): 30-43.

Arif, K.S. & Ali, U. 2019. "Mobile Application testing tools and their challenges: A comparative study. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-6). IEEE. Pakistan, January 30-31, 2019.

Ashton, K. 2009. That 'internet of things' thing. *RFID Journal*, 22(7): 97-114.

Aysolmaz, B. & Demirors, O. 2014. "Modeling business processes to generate artifacts for software development: a methodology" In *Proceedings of the 6th International Workshop on Modeling in Software Engineering* (pp. 7-12). Hyderabad, India, June 2 - 3, 2014.

Aydemir, F.B. & Dalpiaz, F. 2018. “A roadmap for ethics-aware software engineering”. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare) (*pp. 15-21). IEEE. Sweden, May 29, 2018.

Balaji, S. & Murugaiyan, M.S. 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1): 26-30.

Balasubramanian, V., Aloqaily, M., Zaman, F. & Jararweh, Y. 2018. “Exploring computing at the edge: A multi-interface system architecture enabled mobile device cloud”. *In 2018 IEEE 7th International Conference on Cloud Networking (CloudNet)* (pp. 1-4). IEEE. Tokyo, Japan, 22-24 October, 2018.

Basili, V.R. 1996. “The role of experimentation in software engineering: past, current, and future”. In *Proceedings of IEEE 18th International Conference on Software Engineering* (pp. 442-449). IEEE. Montreal, QC, Canada, 6-10 October 2003.

Bassil, Y. 2012. A simulation model for the waterfall software development life cycle. *arXiv preprint arXiv:1205.6904*.

Bate, R., Hess, K. & Mooney, L. 2010. Antimalarial medicine diversion: stock-outs and other public health problems. *Research and Reports in Tropical Medicine*, 010:1. Dovepress [Online]: Available at DOI: 10.2147/RRTM.S13242. [Accessed on 15 October 2019].

Beydeda, S., Gruh, V. & Stachorski, M. 2001. “A graphical class representation for integrated black-and white-box testing”. In *Proceedings IEEE International Conference on Software Maintenance. ICSM* 2001 (pp. 706-715). IEEE. Florence, Italy, 7-9 November 2001.

Booch, G., Rumbaugh, J. & Jacobson, I. 2010. *The Unified Modeling Language User Guide, 1999*. Menlo Park, CA: Addison-Wesley Longman Inc. (p. 285).

Booch, G., Maksimchuk, R.A., Engle, M.W., Young, B.J., Connallen, J. & Houston, K.A. 2008. Object-oriented analysis and design with applications. *ACM SIGSOFT software engineering notes,* 33(5): 29-29.

Bormane, L. & Bērziša, S. 2017. “Role of ‘bridge person’ in software development projects”. In *International Conference on Information and Software Technologies (pp. 3-14).* Springer, Cham. Druskininkai, Lithuania, October 12–14, 2017.

Boswell, W. 2006. Aligning employees with the organization’s strategic objectives: Out of “line of sight”, out of mind. *International Journal of Human Resources Management*, 17: 1489–1511.

Brady, M. & Loonam, J. 2010. Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry. *Qualitative Research in Organizations and Management: An International Journal,* 5(3): 224-237.

Brindha, J. and Vijayakumar, V., 2015. Analytical comparison of waterfall model and object-oriented methodology in software engineering. Advances in Natural and Applied Sciences, 9(12), pp.7-11.

Bruno, O., Nyanchoka, O.A. Ondieki, M.C. & Nyabayo, M.J. 2015. Availability of essential 257 medicines and supplies during the dual pull-push system of drugs acquisition in Kaliro District, Uganda. *Journal of Pharmaceutical Care & Health Systems,* S2-006. doi:10.4172/jpchs.S2-006.

Bryman, A. 2017. *Quantitative and qualitative research: further reflections on their integration.* In J. Brannen (Ed.). *Mixing methods: Qualitative and quantitative research* (pp. 57-78). London, UK: Routledge.

BMAU (Budget Monitoring and Accountability Unit). 2015. Briefing Paper. Continuous stock-outs of medical supplies in Uganda: What are the root causes? [Online]: Available at: http://www.finance.go.ug/dmdocuments/BMAU%2015. [Accessed on 14 October 2019].

Byrne, D. 2002. *Interpreting quantitative data*. Newcastle upon Tyne, UK: Sage.

Caivano, D., Cassano, F., Lanzilotti, R. & Piccinno, A. 2018. “Towards an IoT model for the assessment of smart devices”. In Proceedings of the 2018 international conference on advanced visual interfaces (pp. 1-3). May 29 - June 01, 2018, Castiglione della Pescaia, Italy.

Carter, P., Mulliner, C., Lindorfer, M., Robertson, W. & Kirda, E. 2016. „Curious Droid: automated user interface interaction for android application analysis sandboxes”. In *International Conference on Financial Cryptography and Data Security* (pp. 231-249). Springer, Berlin, Heidelberg.

Center of Global Development. N.d. Background Paper. Center of Global Development, Massachusetts, Washington, DC, pp. 1-52. Available [online] at: www.cgdev.org. (Accessed 08 October 2019).

Chen, D. 2013. Enhancing hospital supply chain performance: A relational view and empirical Test. *Journal of Operations Management*, 31(6): 391 – 408.

Chen, P.J., Du, Y.C., Cheng, K.A. & Po, C.Y. 2016, June. Development of a management system with RFID and QR code for matching and breeding in Taiwan pig farm. In 2016 *13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 1-5). IEEE. 28 June - 01 July, 2016, Chiang Mai in Chiang Mai, Thailand.

Chen, H.M., Kazman, R. & Haziyev, S. 2016. Strategic prototyping for developing big data systems. *IEEE Software*, 33(2): 36-43.

Chkanikova, O. 2016. Sustainable purchasing in food retailing: interorganizational relationship management to green product supply. *Business Strategy and the Environment*, 25(7): 478-494.

Choy, L.T. 2014. The strengths and weaknesses of research methodology: Comparison and complimentary between qualitative and quantitative approaches. *IOSR Journal of Humanities and Social Science*, 19(4): 99-104.

Clayson, P.E., Carbine, K.A., Baldwin, S.A., Olsen, J.A. & Larson, M.J. 2021. Using generalizability theory and the ERP Reliability Analysis (ERA) Toolbox for assessing test-retest reliability of ERP scores Part 1: Algorithms, framework, and implementation. *International Journal of Psychophysiology*, 150: 50-72.

Cocosila, M. & Archer, N. 2010. Adoption of mobile ICT for health promotion: an empirical investigation. *Electronic Markets*, 20(3): 241-250.

Cvetkoska, V. & Savić, G. 2017. Efficiency of bank branches: empirical evidence from a two-phase research approach. *Economic research-Ekonomska istraživanja*, 30(1): 318-333.

Da Xu, L., He, W. & Li, S. 2014. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4): 2233-2243.

Denscombe, M. 2008. Communities of practice: A research paradigm for the mixed methods approach. *Journal of mixed methods research*, 2(3): 270-283.

Dhir, S., Kumar, D. & Singh, V.B. 2019. Success and failure factors that impact on project implementation using agile software development methodology. In *Software engineering* (pp. 647-654). Singapore: Springer. doi.org/10.1007/978-981-10-8848-3_62

Diatchki, I.S. & Jones, M.P. 2006. Strongly typed memory areas programming systems-level data structures in a functional language. In *Proceedings of the 2006 ACM SIGPLAN workshop on Haskell* (pp. 72-83). Portland, Oregon, USA, 17 September 2006.

Doebelin, N. & Kleeberg, R. 2015. Profex: a graphical user interface for the Rietveld refinement program BGMN. *Journal of applied crystallography*, 48(5): 1573-1580.

Dowling, P. 2011. *Healthcare Supply Chains in Developing Countries: Situational Analysis*. Arlington, VA.: USAID DELIVER Project, Task Order 4.

English, T. & Keeley, J.W. 2014. Internal Consistency Approach to Test Construction. *The Encyclopedia of Clinical Psychology*: 1-3.

Fahey, L. & Narayanan, V.K. 1986. *Macro environmental analysis for strategic management* (The West Series in Strategic Management). St. Paul, Minnesota: West Publishing Company.

Feitelson, D.G. 2006. Experimental computer science: The need for a cultural change. [Online]. Available from http://www. cs. huji. ac. il/~ feit/papers/exp05. pdf (pp.1-37).

Ferraris, D. & Fernandez-Gago, C. 2020. TrUStAPIS: a trust requirements elicitation method for IoT. *International Journal of Information Security*, 19(1): 111-127.

Gao, F. & Su, X. 2016. Omnichannel retail operations with buy-online-and-pick-up-in-store. *Management Science*, 63(8): 2478-2492.

Gaur, D. & Aggarwal, S. 2019. Selection of software development model using TOPSIS methodology. In *Data and Communication Networks* (pp. 123-133). Singapore: Springer.

Gelogo, Y.E., Hwang, H.J. & Kim, H.K. 2015. Internet of things (IoT) framework for u-healthcare system. *International Journal of Smart Home,* 9(11): 323-330.

Giret, A., Garcia, E. & Botti, V. 2016. An engineering framework for service-oriented intelligent manufacturing systems. *Computers in Industry*, 81: 116-127.

Granovetter, M. 2005. The impact of social structure on economic outcomes. *Journal of Economic Perspectives,* 19(1): 33–50.

Grell, L. 2014. Pharma Code 2013: Guidelines-Law-Order.

Hakiza, J.B.R. & Basheka, B.C. 2012. Procurement planning and organisational conflicts in Kampala capital city authority-Uganda: implications for service delivery using a case of central division. *International Journal of Procurement Management*, 5(6): 797-819.

Hailes, J. 2014. *Business Analysis Based on BABOK® Guide Version 2–A Pocket Guide*. VW's-Hertogenbosch, Netherlands: Van Haren.

Hamad, E.O., Savundranayagam, M.Y., Holmes, J.D., Kinsella, E.A. & Johnson, A.M. 2016. Toward a mixed-methods research approach to content analysis in the digital age: the combined content-analysis model and its applications to health care Twitter feeds. *Journal of medical Internet research*, 18(3): e60.

Hammi, B., Khatoun, R., Zeadally, S., Fayad, A. & Khoukhi, L. 2017. IoT technologies for smart cities. *IET Networks*, 7(1): 1-13.

Hammersley, M. 2015. On ethical principles for social research. *International Journal of Social Research Methodology,* 18(4): 433-449.

Han, Y. & Hong, S. 2016.*The impact of accountability on organisational performance in the U.S. Federal Government: the moderating role of autonomy*. Review of Public Personnel Administration. 0734371X16682816.

Han, X.I.A.O., 2006. Reuse of Waterfall Model for Object-oriented Software. Journal of Xihua University (Natural Science Edition), p.02

Herrmann, P. 2001. Information flow analysis of component-structured applications. In *Seventeenth Annual Computer Security Applications Conference* (pp. 45-54). IEEE. 10 - 14 December 2001, New Orleans, LA, USA.

Huebscher, M.C. & McCann, J.A. 2008. A survey of autonomic computing—degrees, models, and applications. *ACM Computing Surveys (CSUR)*, 40(3): 1-28.

Iqbal, M.J., Geer, M.I. & Dar, P.A. 2017. Medicines management in hospitals: a supply chain perspective. *Systemic Reviews in Pharmacy*, 8(1): 80-85.

Jamsa, K. 2012. *Cloud computing: SaaS, PaaS, IaaS, virtualization, business models, mobile, security and more.* Burlington, MA: Jones & Bartlett Publishers.

Jonasson, H. 2016. *Determining project requirements: mastering the BABOK and the CBAP exam*. Boca Raton, FL: CRC Press.

Johnson, R.B. & Onwuegbuzie, A.J. 2004. Mixed methods research: A research paradigm whose time has come. Educational researcher, 33(7): 14-26.

Kähkönen, T. 2017. Understanding and managing enterprise systems integration.

Kaplan, B. & Maxwell, J.A. 2005. Qualitative research methods for evaluating computer information systems. In Evaluating the organizational impact of healthcare information systems (pp. 30-55). Springer, New York, NY.

Kapur, P.K., Pham, H., Gupta, A. & Jha, P.C. 2011. Software reliability assessment with OR applications (p. 364). London: Springer.

Kasapoglu, O.A. 2016. Selection of the forecasting model in health care. *Journal of Hospital Medical Management,* 2(2): 1-5.

Kay, K., High, D.R. & Atchley, M.D. & Wal-Mart Stores Inc. 2018. *Shopping facility assistance systems, devices, and methods to identify security and safety anomalies*. U.S. Patent 9,875,502.

Keengwe, J., Onchwari, G. & Wachira, P. 2008. Computer technology integration and student learning: Barriers and promise. *Journal of Science Education and Technology*, 17(6): 560-565

Keiningham, T., Aksoy, L., Bruce, H.L., Cadet, F., Clennell, N., Hodgkinson, I.R. & Kearney, T. 2020. Customer experience driven business model innovation. Journal of Business Research, 116: 431-440.

Killion, L. & Fisher, R. 2018. 1. Ontology, Epistemology: Paradigms and Parameters for Qualitative Approaches to Tourism Research. In W.J. Hillman & K. Radel (Eds.). *Qualitative methods in tourism research* (pp. 1-28). Bristol, UK: Channel View Publications.

Kivunja, C. & Kuyini, A.B. 2017. Understanding and applying research paradigms in educational contexts. *International Journal of Higher Education*, 6(5): 26-41.

Khaldi, K. 2017. Quantitative, Qualitative or Mixed Research: Which Research Paradigm to Use? *Journal of Educational and Social Research*, 7(2): 15.

Khan, G., Sengupta, S. & Sarkar, A. 2015, March. Modeling of services and their collaboration in enterprise cloud bus (ECB) using UML 2.0. In 2015 *International Conference on Advances in Computer Engineering and Applications* (pp. 207-213). IEEE.

Khan, M. & Khan, S.S. 2011. Data and information visualization methods, and interactive mechanisms: A survey. *International Journal of Computer Applications*, 34(1): 1-14.

Khan, M.E. & Khan, F. 2012. A comparative study of white box, black box and grey box testing techniques. *International Journal* of *Advanced Computer Science* and *Applications*, 3(6).

Khurana, S., Chhillar, N., Kumar, V. & Gautam, S. 2013. Inventory control techniques in medical stores of a tertiary care neuropsychiatry hospital in Delhi. *Health,* 5(1): 8-13.

Kim, H.K., Yeo, H., Hwang, H.J., Ramos, C. & Marreiros, G. 2016. *Effective mobile applications testing strategies*.

Kimmerly, R.S. 2010. *Using attributes to identify and filter pluggable functionality*. U.S. Patent 7,831,956. Microsoft Corp.

King, D.L. 2018. The Internet of Things (IoT) and Libraries. [Online]. Available from: https://davidleeking.com/the-internet-of-things-iot-and-libraries/ [Accessed 25 April 2021].

Korkmaz, Ö., Çakir, R. & Özden, M.Y. 2017. A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior,* 72: 558-569.

Kraiselburd, S. & Yadav, P. 2012. Supply chains and global health: an imperative for bringing operations management scholarship into action. *Production and Operations Management Society*, 0(0): 1–5.

Kramer, M. 2018. Best practices in systems development lifecycle: An analyses based on the waterfall model. *Review of Business & Finance Studies*, 9(1): 77-84.

Kukul, V., Gökçearslan, Ş. & Günbatar, M.S. 2017. Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(1): 158-179.

Kumar, R., Singh, R.K. & Shankar, R. 2015. Critical success factors for implementation of supply chain management in Indian small and medium enterprises and their impact on performance. *IIMB Management Review*, 27: 92-104. [Online]: Available at: www.sciencedirect.com

Kurtanović, Z. & Maalej, W. 2017, September. Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 490-495). Lisbon, Portugal 4 - 8 September 2017.

Labiad, N., Beidouri, Z. & Bouksou, O. 2014. A review on supply chain coordination: A classification of coordination mechanisms and problems. *Asian Journal of Management Research*, 4(3): 617-630.

Lakhoua, M.N. 2012. Using structured analysis for the control of real-time systems. *Journal of Engineering and Technology Research*, 4(5): 82-88.

Lalvani, P., Yadav, P., Curtis, K. & Bernstein, M. 2010. *Increasing patient access to antiretroviral: Recommended actions for a more efficient global supply chain*. Center for Global Development. [Online]. Available from: https://www.cgdev.org/sites/default/files/archive/doc/HIVAIDSMonitor/ARV_Background-FINAL1.pdf

Laundry, S. & Beaulieu, M. 2013. “The challenges of hospital supply chain management from central stores to nursing units”. In B.T. Denton (Ed.). *The handbook of healthcare operations management: methods and application*. London: Springer.

Lapalme, J., Gerber, A., Van der Merwe, A., Zachman, J., De Vries, M. & Hinkelmann, K. 2016. Exploring the future of enterprise architecture: A Zachman perspective. *Computers in Industry,* 79: 103-113.

Lee, S. 2012. Unified Modeling Language (UML) for Database Systems and Computer Applications. *International Journal of Database Theory and Application*, 5(1): 157-164.

Lee, S.W. & Rine, D.C. 2004. Case Study Methodology Designed Research in Software Engineering Methodology Validation. In SEKE (pp. 117-122).

Li, T. 2013. Principle, Framework and Application of Internet of Things. In *Applied Mechanics and Materials,* 303: 2144-2148. Trans Tech Publications.

Li, T. & Chen, L. 2012. “Internet of things: Principle, framework and application”. In Y. Zhang (Ed.). *Future Wireless Networks and Information Systems* (pp. 477-482). Berlin, Heidelberg: Springer.

Lotlikar, T., Kankapurkar, R., Parekar, A. & Mohite, A. 2013. Comparative study of Barcode, QR-code and RFID System. *International Journal of Computing Technological Applications*, 4(5): 817-821.

Lu, Y. & Cecil, J. 2016. An Internet of Things (IoT)-based collaborative framework for advanced manufacturing. *The International Journal of Advanced Manufacturing Technology*, 84(5-8): 1141-1152.

Maarouf, H. 2019. Pragmatism as a supportive paradigm for the mixed research approach: Conceptualizing the ontological, epistemological, and axiological stances of pragmatism. *International Business Research*, 12(9): 1-12.

Mademann, F. 2018. The 5G system architecture. *Journal of ICT Standardization*, 6(3): 77-86.

Mahalakshmi, M. & Sundararajan, M. 2013. Traditional SDLC Vs Scrum Methodology–A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6): 192-196.

Makombe, G. 2017. An expose of the relationship between paradigm, method and design in research. *The qualitative report*, 22(12): 3363.

Malone, T.W. & Crowston, K. 1994. The interdisciplinary study of coordination. *Computing Surveys*, 26(1): 87-119.

Master, A., Cheryan, S. & Meltzoff, A.N. 2016. Computing whether she belongs: Stereotypes undermine girls’ interest and sense of belonging in computer science. *Journal of educational psychology,* 108(3): 424.

Mathiesen, P., Bandara, W., Delavari, H., Harmon, P. & Brennan, K. 2011. A comparative analysis of business analysis (BA) and business process management (BPM) capabilities. *ECIS 2011 Proceedings*.

Matthews, I. 2014. *Challenges to Distribution and Marketing of Health Products in Emerging Markets. Life drops*. [Online] Available at: http://liveleaf.com/wp-content/uploads/2016/11/LIT-185-A- A-Distribution-and-Marketing-of-Health-Products-in-Emerging-Markets.pdf. [Accessed on 06 October 2019].

Maxwell, J.A. 2019. Distinguishing between quantitative and qualitative research: A response to Morgan. *Journal of mixed methods research*, 13(2): 132-137.

McCusker, K. & Gunaydin, S. 2015. Research using qualitative, quantitative or mixed methods and choice based on the research. *Perfusion*, 30(7): 537-542.

Mendoza, A. & Gu, G. 2018, May. Mobile application web api reconnaissance: Web-to-mobile inconsistencies & vulnerabilities. In *2018 IEEE Symposium on Security and Privacy (SP)* (pp. 756-769). IEEE.

Middleton, F. 2019. Research methods: Definitions, Types, Examples. [Online]. Available: https://www.scribbr.com/methodology/ [Accessed 20 March 2021].

Miraz, M.H. & Ali, M. 2020. Blockchain enabled smart contract based applications: Deficiencies with the software development life cycle models. arXiv preprint arXiv:2001.10589.

Mittal, S. 2016. A survey of techniques for approximate computing. *ACM Computing Surveys (CSUR)*, 48(4): 1-33.

Mpimbaza, A., Miles, M., Sserwanga, A., Kigozi, R., Wanzira, H., Rubahika, D., Nasr, S., Kapella, B.K., Yoon, S.S., Chang, M. & Yeka, A. 2015. Comparison of routine health management information system versus enhanced inpatient malaria surveillance for estimating the burden of malaria among children admitted to four hospitals in Uganda. *The American Journal of Tropical Medicine and Hygiene*, 92(1): 18-21.

Mohajan, H.K. 2017. Two criteria for good measurements in research: Validity and reliability. Annals of Spiru Haret University. *Economic Series*, 17(4): 59-82.

Mukherjee, M. 2016. Object-Oriented Analysis and Design. *International Journal of Advanced Engineering and Management*, 1(1): 1-11.

Mylopoulos, J., Chung, L. & Yu, E. 1999. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1): 31-37.

Naeem, M., Ashraf, R., Ali, N., Ahmad, M. & Habib, M.A. 2017. “Bottom up approach for better requirements elicitation”. In *Proceedings of the International Conference on Future Networks and Distributed Systems* (pp. 1-4). Cambridge United Kingdom, July 19 - 20, 2017.

Nagitta, O.P. & Mkansi, M. 2019. Exploring the supply chain coordination dimensions for Artemisinin-based combination therapies in Uganda.

Nakyanzi, J.K., Kitutu, F.E. Oria, H. & Kamba, P.F. 2010. Expiry of medicines in supply outlets in Uganda. *Bulletin of the World Health Organization 2010*; 88: 154–158. doi:10.2471/BLT.08.057471. [Accessed on 11 October 2019].

Nanyunja, M., Orem, J.N., Kato, F., Kaggwa M., Katureebe, C. & Saweka, J. 2011. “Malaria Treatment policy change and implementation: the case of Uganda”. *SAGE-Hindawi Access to Research Malaria Research and Treatment*. Volume 2011, Article ID 683167. [Online]: Available from: http://www.hindawi.com/journals/mrt/2011/683167/. [Accessed on 06 October 2019].

Nastic, S., Sehic, S., Le, D.H., Truong, H.L. & Dustdar, S. 2014, August. Provisioning software-defined IoT cloud systems. In *2014 international conference on future internet of things and cloud* (pp. 288-295). IEEE. Barcelona, Spain, 27–29 August, 2014.

Nayak, R., Singh, A., Padhye, R. & Wang, L. 2015. RFID in textile and clothing manufacturing: technology and challenges. *Fashion and Textiles*, 2(1): 1-16.

Nidhra, S. & Dondeti, J. 2012. Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications*, 2(2): 29-50.

Nieman, G. & Bennett, A. 2002. *Business management a value chain approach*. Pretoria: Van Schaik Publishers.

NIST. 2009. *Definition of cloud computing v15, NIST, Editor.* National Institute of Standards and Technology: Gaithersburg, MD (2009).

Nogueira, J.M., Romero, D., Espadas, J. & Molina, A. 2013. Leveraging the Zachman framework implementation using action–research methodology – case study: aligning the enterprise architecture and the business goals. *Enterprise Information Systems*, 7(1): 100-132.

Okesola, O.J., Adebiyi, A.A., Owoade, A.A., Adeaga, O., Adeyemi, O. & Odun-Ayo, I. 2020. “Software Requirement in Iterative SDLC Model”. In *Computer Science On-line Conference* (pp. 26-34). Springer, Cham. Virtual, 6-9 July 2020.

Oates, B.J. 2005. Researching information systems and computing. Newcastle upon Tyne, UK: Sage.

Park, K. O., Chang, H. & Jung, D.H. 2017. How Do Power Type and Partnership Quality Affect Supply Chain Management Performance? *Sustainability*, 9: 127. doi:10.3390/su9010127 www.mdpi.com/journal/sustainability.

Paul, S., Chatterjee, A. & Guha, D. 2019. Study of Smart Inventory Management System Based on The Internet of Things (IoT). *International Journal on Recent Trends in Business and Tourism,* 3(3): 27-34.

Pickard, L.M., Kitchenham, B.A. & Jones, P.W. 1998. Combining empirical results in software engineering. *Information and software technology*, 40(14): 811-821.

Polak, P. 2009. Systems Analysis and Design in Polish Universities Curricula: Structured or Object-Oriented. In Systems Analysis and Design for Advanced Modeling Methods: Best Practices (pp. 88-95). IGI Global.

Polit, D.F. 2014. Getting serious about test–retest reliability: a critique of retest research and some recommendations. *Quality of Life Research*, 23(6): 1713-1720.

Popa, S. & Vaida, M.F. 2015. ENTERPRISE BUS AS A SERVICE-AN IDEAL CLOUD CONNECTIVITY MODEL. *Acta Technica Napocensis*, 56(1): 29.

Pinna, R., Carrus, P.P. & Marras, P. 2015. Emerging trends in healthcare supply chain management — an Italian experience. [Online]: Available at: http://dx.doi.org/10.5772/59748. [Accessed on 13 October 2019].

Puccinelli, D. & Haenggi, M. 2005. Wireless sensor networks: applications and challenges of ubiquitous sensing. *IEEE Circuits and systems magazine*, 5(3): 19-31.

Rajas, M. & Pund, M.A. 2017. The Internet of Things–Improving Customer Shopping Experience at Stores. *Development*, 248: 256.

Ragunath, P.K., Velmourougan, S., Davachelvan, P., Kayalvizhi, S. & Ravimohan, R. 2010. Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC). *International Journal of Computer Science and Network Security*, 10(1): 112-119.

Rani, S.B.A.S.U. 2017. A detailed study of Software Development Life Cycle (SDLC) models. *International Journal of Engineering and Computer Science*, 6(7).

Recker, J. 2010. Opportunities and constraints: the current struggle with BPMN. *Business Process Management Journal*, 16(1): 181-201.

Resnik, D.B. 2015, December. What is ethics in research & why is it important. In ideas.

Rossi, B. 2014. Entity relationship *diagram*.

Saad, A. & Dawson, C. 2018. Requirement elicitation techniques for an improved case based lesson planning system. *Journal of Systems and Information Technology*.

Sahu, K. & Srivastava, R.K. 2019. Revisiting software reliability. *Data Management, Analytics and Innovation:* 221-235.

Savu, L. 2011. Cloud computing: Deployment models, delivery models, risks and research challenges. In *2011 International Conference on Computer and Management (CAMAN)* (pp. 1-4). IEEE. Wuhan, China, 19-21 May 2011.

Saravanan, T., Jha, S., Sabharwal, G. & Narayan, S. 2020, December. Comparative Analysis of Software Life Cycle Models. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (pp. 906-909). IEEE. Galgotias College of Engineering and Technology, Greater Noida, India, December 18-19, 2020.

Scandura, T.A. & Williams, E.A. 2000. Research methodology in management: Current practices, trends, and implications for future research. *Academy of Management Journal,* 43(6): 1248-1264.

Schijven, M. & Jakimowicz, J. 2003. Construct validity. *Surgical Endoscopy and Other Interventional Techniques*, 17(5): 803-810.

Schwartz-Shea, P. & Yanow, D. 2013. *Interpretive research design: Concepts and processes*. London, UK: Routledge.

Scroggins, R. 2014. SDLC and development methodologies. *Global Journal of Computer Science and Technology*, 14(7).

Sharma, M.K. 2017. A study of SDLC to develop well engineered software. *International Journal of Advanced Research in Computer Science*, 8(3).

Sharma, K. & Bala, M. 2019. New Failure Rate Model for Iterative Software Development Life Cycle Process. arXiv preprint arXiv:1910.00903.

Simatupang, T.M., Wright, A.C. & Sridharan, A. 2002. The knowledge of coordination for supply chain integration. *Business Process Management Journal*. 8(3): 289-308.

Singh, M., Sharma, A.K. & Saxena, R. 2016. Formal transformation of UML diagram: Use case, class, sequence diagram with Z notation for representing the static and dynamic perspectives of system. In *Proceedings of International Conference on ICT for Sustainable Development* (pp. 25-38). Springer, Singapore.

Singh, R.K. 2011. Developing the framework for coordination in supply chain of SMEs. *Business Process Management Journal*, 17(4): 619-638.

Singh, R., Singh, E. & Nalwa, H.S. 2017. Inkjet printer nanomaterial based flexible radio frequency identification (RFID) tag sensors for the internet of nano things. *RSC Advances*, 7(77): 48597-48630.

Sirard, J.R., Forsyth, A., Oakes, J.M. & Schmitz, K.H. 2011. Accelerometer test-retest reliability by data processing algorithms: results from the Twin Cities Walking Study. *Journal of Physical Activity and Health*, 8(5): 668-674.

Snider, E. 2019. *Mastering Xamarin. Forms: App architecture techniques for building multi-platform, native mobile apps with Xamarin*. Forms 4. Packt Publishing Ltd.

Sousa, M.J., Pesqueira, A.M., Lemos, C., Sousa, M. & Rocha, Á. 2019. Decision-Making based on Big Data Analytics for People Management in Healthcare Organizations. *Journal of medical systems*, 43(9): 290.

Splett, J.W., Perales, K., Al-Khatib, A.M., Raborn, A. & Weist, M.D. 2020. Preliminary development and validation of the Interconnected Systems Framework-Implementation Inventory (ISF-II). *School Psychology*, 35(4): 255.

Stanley, E.F., Cynthia, W., Chad, A. & Gregory, M. 2009. Supply chain information sharing: Benchmarking a proven path. *Benchmarking: An International Journal*, 16(2): 222-246.

Statler, S. 2016. “Barcodes, QR Codes, NFC, and RFID”. In S. Statler (Ed.). *Beacon Technologies,* pp. 317-331. Berkeley, CA: Apress.

Sun, Y., Song, H., Jara, A.J. & Bie, R. 2016. Internet of things and big data analytics for smart and connected communities. *IEEE access*, 4: 766-773.

Sutherland, S. & Chetty, G. 2016. Investigation into Interoperability in Cloud Computing: An Architectural Model. *Journal of Computers*, 11(2): 159-168.

Tătaru, I.M. & Fleacă, E. 2019. Technologies for Modeling Business Processes. *FAIMA Business & Management Journal,* 7(2): 31-41.

Tang, W., Cui, Y. & Babenko, O. 2014. Internal consistency: Do we really know what it is and how to assess it. *Journal of Psychology and Behavioral Science*, 2(2): 205-220.

Tam, C., da Costa Moura, E.J., Oliveira, T. & Varajão, J. 2020. The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, 38(3): 165-176.

Thatcher, R. 2010. Validity and Reliability of Quantitative Electroencephalography. *Journal of Neurotherapy,* 14: 122-152.

Thönes, J. 2015. Microservices. *IEEE software*, 32(1): 116-116.

Thov, B. & Quatius Ltd. 2016. Display screen with graphical user interface. U.S. Patent Application 29/480,557.

Toman, M., Furey, E. & Curran, K. 2020. Real-World Efficacy of an Infrastructure-Free Geomagnetic Indoor Positioning System. In *2020 31st Irish Signals and Systems Conference (ISSC)* (pp. 1-8). IEEE. Letterkenny, Ireland, 11-12 June 2020.

Turkyilmaz, A., Bulak, M.E & Zaim, S. 2015. Assessment of TQM practices as a part of supply chain management in healthcare institutions. *International Journal of Supply Chain Management,* 4(4): 1-9.

Wheelock, P., Weiss, H.J., Sweet, N., Allred, C. & Becker, M. , PROGRESSIVE COMPUTER SERVICES Inc and PROGRESSIVE COMPUTER SERVICES 2016. *Method and apparatus for acquiring detailed delivery tracking*. U.S. Patent Application 14/835,318. Progressive Computer Services Inc.

Wieringa, R.J., Blanken, H.M., Fokkinga, M.M. & Grefen, P.W. 2003. Aligning application architecture to the business context. In *International Conference on Advanced Information Systems Engineering* (pp. 209-225). Springer, Berlin, Heidelberg.

Umlauf, R. & Park, S.J. 2018. Stock-outs! Improvisations and processes of infrastructuring in Uganda's HIV/Aids and malaria programmes. *Global public health*, 13(3): 325-338.

Varpio, L. & MacLeod, A. 2020. Philosophy of science series: Harnessing the multidisciplinary edge effect by exploring paradigms, ontologies, epistemologies, axiologies, and methodologies. *Academic Medicine*, 95(5): 686-689.

Vledder, M., Friedman, J. Sjöblom, M., Brown, T. & Yadav, P. 2015. Optimal supply chain structure for distributing essential drugs in low-income countries: Results from a randomized experiment. Ross School of Business Working Paper No. 1269, March 2015. University of Michigan. Social Sciences Research Network Electronic Paper Collection: Available from: http://ssrn.com/abstract=258571. [Accessed on 06 October 2019].

Watsierah, C.A. & Ouma, C. 2014. Access to artemisinin-based combination therapy (ACT) and quinine in malaria holoendemic regions of western Kenya. *Malaria Journal*, 13: 290.

Wautelet, Y., Heng, S., Hintea, D., Kolp, M. & Poelmans, S. 2016. Bridging user story sets with the use case model. In *International Conference on Conceptual Modeling* (pp. 127-138). Springer, Cham.

Williams, B.D., Roh, J. Tokar, T. & Swink, M. 2013. Leveraging supply chain visibility for Responsiveness: the moderating role of internal integration. *Journal of Operations Management*, 31(7-8): 543–554.

Wiley, D.E. 2002. Validity of constructs versus construct validity of scores. The role of constructs in psychological and educational measurement: 207-227.

Wulf, J. & Blohm, I. 2020. Fostering Value Creation with Digital Platforms: A Unified Theory of the Application Programming Interface Design. *Journal of Management Information Systems*, 37(1): 251-281.

Yu, W. & Jiang, Y. 2017. Mobile RFID mutual authentication protocol based on hash function. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 358-361). IEEE.

APPENDIX A:
ETHICAL CLEARANCE CERTIFICATE

UNISA university of south africa

UNISA-CAES HEALTH RESEARCH ETHICS COMMITTEE

Date: 05/10/2020

NHREC Registration # : REC-170616-051
REC Reference # : 2020/CAES_HREC/131
Name : Prof M Mkansi
Staff #: 90215028

Dear Prof Mkansi

Decision: Ethics Approval from 01/10/2020 to 30/09/2023

Researcher(s): Prof M Mkansi
mkansm@unisa.ac.za; 076-833-3274

Dr OP Nagitta
poluka@umi.ac.ug

Working title of research:

A mobile application supply chain coordination software for artemisinin-based combination therapies drugs (ACTs)

Qualification: Non-degree purposes

Thank you for the application for research ethics clearance by the Unisa-CAES Health Research Ethics Committee for the above mentioned research. Ethics approval is granted for three years, **subject to submission of yearly progress reports. Failure to submit the progress report will lead to withdrawal of the ethics clearance until the report has been submitted.**

The researcher is cautioned to adhere to the Unisa protocols for research during Covid-19.

Due date for progress report: 30 September 2021

*The **low risk application** was **reviewed** by the UNISA-CAES Health Research Ethics Committee on 01 October 2020 in compliance with the Unisa Policy on Research Ethics and the Standard Operating Procedure on Research Ethics Risk Assessment.*

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za