University of Fort Hare

Together in Excellence

Department of Computer Science

A comparative analysis of mobile application development frameworks: A case study of mobile application development for water usage management in Alice and Fort Beaufort communities.

By

Kwabena Manu

A project submitted for the fulfilment of the requirements for a Master of Science Degree in Computer Science.

Faculty of Science and Agriculture

Alice

South Africa

Supervisor: Mr. S. Ngwenya

Co-Supervisor: Prof K. Sibanda

DECLARATION

I, Kwabena Manu declare that this research, which I hereby submit for the Master of Computer Science at University of Fort Hare, is my work and that it has never been submitted for any other degree at this or any other institution of higher learning.

Signature:

Date: 07 February 2019

University of Fort Hare
Together in Excellence

ACKNOWLEDGEMENTS

I would like to thank the Almighty God for his love and grace towards me and for the strength to successfully complete this project. My sincere appreciation goes to my supervisors Mr. S. Ngwenya and Prof. K. Sibanda for their guidance and support throughout the research process. I benefited a lot from you, God bless you. I would like to thank Risk and Vulnerability Science Centre (RAVAC) headed by Dr. L. Zhou at University of Fort Hare for their financial support without which my dream of pursuing master's degree would have not been achieved. Gratitude goes to my family (Mr & Mrs Amoah, Juliet Amoakoah Danso, Gracie Amoakoah Manu, Mr. Mojeed Adedoyin Agoro, Sis. Hannah, Sis. Sarah and Bro. TT) for their motivation and encouragement. Finally, I would like to thank my lab mates and all those who assisted in the completion of my research.

ABSTRACT

Mobile phones have become an integral part of standard of living. Users and customers expect sensible and very useful applications in less time. In this competitive market, it is an enormous challenge to develop high performance mobile applications that might meet the expectations of end users. Despite the fact that development of new applications for each mobile operating system in short time is fairly an issue, mobile operating systems vendors are giving their best available resources for creating applications in additional convenient ways. These days, there is a common tendency to look for less complex and quicker solutions that could be used in the process of software development. Developers of a new mobile application have to undertake variety of selections and decisions, including the target platform as well as the development technology to utilize. Mobile application development frameworks contribute in solving this problem. Several frameworks have emerged, which we classify and evaluate their appropriateness. In order to compare existing development frameworks solutions in this research, we compiled a set of criteria to assess mobile application development approaches. In view on these criteria, we evaluated five frameworks that is, PhoneGap, Xamarin, App Inventor, Sencha Touch and DragonRad. For proof of concepts, the chosen framework from the five evaluated mobile development frameworks was used to develop application for water usage management. The research will equip mobile application developers to gain more insight into mobile development frameworks available, therefore helping them to choose the most appropriate framework for their project.

Keywords: Mobile Phones, Frameworks, Mobile Applications, Operating System

ABBREVIATIONS

1G	-	First Generation
2G	-	Second Generation
3D	-	Three-Dimensional
3G	-	Third Generation
4G	-	Fourth Generation
API	-	Application Programming Interface
ATM	-	Automated Teller Machine
BCL	-	Base Class Library
BES	-	BlackBerry Enterprise Server
CLI	-	Command Line Interface
CMD	-	Command Prompt
CMDA	-	Code-Division Multiple Access
EDGE	-	Enhanced Data Rate For GSM Evolution
FCM	-	Firebase Cloud Messaging
FLOW	-	Field Level Operations Watch
GeSI	-	Global e-Sustainability Initiative
GIS	-	Geographical Information System
GPRS	-	General Packet Radio Service
GPS	-	Global Positioning System
GSM	-	Global System for Mobile Communication
GSMA	-	Global System for Mobile Communication Association
GUI	-	Graphical User Interface
HTTP	-	Hyper Text Transfer Protocol
iCOMMS	-	Information for Community Service
ICTs	-	Information and Communication Technologies
IDE	-	Integrated Development Environment
iOS	-	iPhone Operating System
IoT	-	Internet of Things
ITU	-	International Telecommunication Union

JME	-	Java Platform Micro Edition
LINQ	-	Language Integrated Query
LTE	-	Long-Term Evolution
M4D	-	Mobile Phone For Development
MDGs	-	Millennium Development Goals
MIT	-	Massachusetts Institute of Technology
MMS	-	Multimedia Messaging
MP	-	Mobile Phone
MVC	-	Model View Control
NFC	-	Near-field Communication
NGO	-	Non-Governmental Organization
OS	-	Operating System
PC	-	Personal Computer
RF	-	Radio Frequency
RFID	-	Radio-frequency Identification
RIM	-	Research In Motion
SCADA	-	Supervisory Control and Data Acquisition
SDGs	-	Sustainable Development Goals
SDK	-	Software Development Kit
SIM	-	Subscriber Identity Module
SMS	-	Short Message Service
UN	-	United Nations
URL	-	Uniform Resource Locator
USAID	-	United States Agency for International Development
USB	-	Universal Serial Bus
USSD	-	Unstructured Supplementary Service Data
VM	-	Virtual Machine
WaGs	-	Water Action Groups
WAP	-	Wireless Application Protocol
WQR	-	Water Quality Reporter
WRC	-	Water Research Council

WRT	-	Web Runtime
WSB	-	Water Services Boards
WSPs	-	Water Service Providers
WWF	-	World Wide Fund
WYSIWYG	-	What You See Is What You Get

LIST OF FIGURES

LIST OF TABLES

Table of Contents

CHAPTER ONE

INTRODUCTION TO RESEARCH PROBLEM AND ITS SETTING

1.1Introduction

Mobile applications are well known among users of smartphones and tablets. Mobile users interconnect through applications over the mobile device's interfaces such as touch-screen, sound, and visual inputs and outputs. Applications can incorporate information from imbedded or outside sensors, web, or any of the capacities local to devices and have solid computation control. Because of their diminutive size and light energy utilization, mobile devices are portable instruments and are finding uses out of the office (Kepley, 2014). Mobile applications are valuable tools for several purposes. In the water sector, mobile applications are very valuable in enabling the sharing of knowledge and information on water resources for households and for industrial use. The applications can be used to monitor water distribution system, monitor consumers' consumption rate, test water qualities, lessen physical data collection faults, and to increase transparency and accountability. To develop a new Mobile Application, a couple of decisions need to be made. The primary one is the platform on which the application will run. Examples of platforms include iOS platforms and Android platforms. After selecting the platform, the next vital question to consider would be what frameworks ought to be utilized to develop the given mobile application? Hence, the developer or technical leader has the responsibility for arriving at this choice. There are a number of frameworks for creating mobile applications. The most known examples of such frameworks include PhoneGap, Xamarin, App Inventor, Sencha Touch and DragonRad. Each framework has diverse qualities and shortcomings. Recently, a considerable number of mobile operating systems required more advancement and to realize that, open source mobile frameworks came up to allow development of applications for various operating systems.

The essential question that this research attempts to answer is: What is the appropriate technology to use for development and improvement of a mobile application for a given situation and requirements? This research also seeks to propose the use of relevant mobile development framework to develop mobile application for water usage management issues such as undetected leakages due to faulty water regulatory mechanisms in the Alice and Fort Beaufort communities. In view of common prerequisites of applications and on a broad arrangement of data assets, we deduce subjective conditions. These criteria along with other comparing procedures are used to assess five mobile frameworks. Literature data, application requirements and most essentially, own experiences are establishment for making a decision as to what degree a framework satisfies a specific model (Heitkötter et al, 2013). In light of these criteria, we assess PhoneGap, App Inventor, Sencha Touch, Xamarin, DragonRad, therefore, evaluating their appropriateness for specific circumstances. These frameworks will be evaluated and discussed in subsequent segments.

1.2Why mobile phone?

Mobile phones (MPs) are but one form of the pervasive Information and Communication Technologies (ICT). Devices such as personal computers, laptops and the Internet are all used to improve developments in both developed and developing countries, so why focus on mobile phones? The most obvious answer is the scale of adoption.

The most widespread information technology across the world today including developing countries is the mobile phone. As a key component of information technology, mobile phone has the potential to support education, knowledge development and trade which are important issues in developing countries (Wade, 2004). The majority of people in the least developed countries still live in rural areas and their livelihood depends on the primary industries. A research by Hellstrom (2010) showed that almost one in five Africans owns a phone. The widespread use of mobile phones should add to more use of voice and Short Message Service (SMS) solutions as they offer easy accessibility.

In addition to voice communication, MPs allow the transfer of data, which can be used in the context of applications for the purposes of health, education, commerce or governance. Mobile Phones can serve as the backbone for early warning systems to mitigate agricultural risks and safeguard agricultural incomes (George et al, 2011). In Turkey, local weather forecasts transmitted through SMS provided very timely warnings of impending frosts or conditions that favored pests (Hellstrom, 2010).

The less privilege in Africa tend to use public access facilities and to share phones. Research shows that in typical rural districts of Africa, up to 80% of households make regular use of phones (Tulchin, 2011). One of the key features driving growth in mobiles is that they are readily available, and inherently suited to remote areas with poor infrastructure. In addition, the prepaid system of low denomination, scratch cards are perfectly matched to the economic situation of many Africans, It is recognized that mobiles offer potentially cheap means of communicating, especially through the use of SMS. It is important to consider constraints facing women in access to and use of mobile phones, but preliminary evidence indicates that mobile phone appears to be a gender neutral tool (Kefela, 2011).

Mobile phones differ in two important ways from other ICTs. Firstly, they are easier for the rural poor to access than many other technologies, which tend to be more expensive and require infrastructure. Landlines have existed for decades, but only 3% percent of Africans had access to one (Cranston, 2010). In contrast, in the space of five years, mobile telephone subscriptions on the continent shot up from 12% to 45% (Cranston, 2010). Secondly, mobile technology moves users into the area of interaction. Moraa et al (2013) reported that the number of mobile subscriptions in India is twice the number of piped water connections per person, whilst in Africa, Global System for Mobile (GSM) signal have surpassed the number of individuals with a better-quality of water supply.

Studies also tell us that using mobile phones is considered as important means of providing timely information for rural dwellers. Based on this evidence, it can be concluded that mobile phones are devices that can have positive impact in the water and agriculture sectors, and can change rural people’s life in terms of providing them with the necessary information they require.

1.3Related work

There exist a number of studies on distinctive technologies and the most ideal approach for creating mobile applications. Palmieri et al. (2012), analysed four common cross-platform frameworks namely MoSync, PhoneGap, Rhodes, and DragonRad. The authors selected Rhodes framework for the reason that it supports web-based services and the MVC framework. On the other hand, the authors recognised some distinct advantages for the other three frameworks. They emphasized that PhoneGap has a wrapper that supports quite a lot of IDEs. MoSync offers supports for highest number of programming languages and APIs across JavaScript and C/C++ and DragonRad comes with its own IDE with broad features, including drag and drop GUI for developers to design, develop and install mobile applications.

Heitkotter et al, (2013) outlined criteria to evaluate cross-platform development frameworks, and used the criteria to compare Web applications, applications created with Titanium Mobile and PhoneGap, and natively developed applications. They concluded that PhoneGap is ideal; however Titanium makes it less demanding to make user interfaces which look like native applications.

These studies provide useful insights into the workings of the studied frameworks. However, while these publications raise a better understanding of the frameworks, they do not actually compare the different approaches. Most studies are centred on frameworks such us PhoneGap, Appcelerator Titanium and jQuery neglecting the other frameworks. There is also missing research on a framework for comparative assessment of mobile frameworks that is adaptable over time and to the particular subjective needs of different people and organizations (Majchrzak et al., 2015).

1.4Problem statements

It is common knowledge nowadays that Mobile applications have become a very important part of our lives. Several organizations are using mobile devices for service delivery and customer liaison and this enhances productivity and also raises profit margins. The challenge arises when it comes to how these mobile applications are developed. The main factors that determine or influence these challenges include the framework of development. Hence this work seeks to explore various mobile development frameworks that could be used to build such systems. A web-based interface may seem to be a straightforward answer, however, it is not usually useful since it limits access to certain mobile functionalities and would solely be accessible once internet connection exists. It is anticipated that the results of the comparative analysis among the chosen frameworks will determine the framework of choice and on this framework, a mobile application for water management in the Alice and Fort Beaufort communities will be developed. We employed a mixed methods approach in conducting this research in which general literature review and evolutionary prototyping were used on various stages of project development.

1.5 The aim

The aim of the research is to provide an overview of the currently relevant technologies in the mobile environment to support decision making with regard to mobile application development frameworks. This overview contains the current major development frameworks and tools for mobile application development. We also produce a comparison of the currently available frameworks for developing mobile application for multiple platforms and providing guidelines for choosing the most suitable development framework for a given use case. For proof of concept, the chosen development framework will be used to develop mobile application for water usage management.

1.6Research questions

This research seeks to address the following questions in order to highlight the frameworks options available for mobile application developers:

1. What influences the rate of growth of usage of mobile applications?
2. What types of mobile development frameworks are there or are being used?
3. What is the most appropriate framework to be used?
4. How can a novel mobile application be designed to address water loss problems?

1.7Research objectives

The main objective of this work is to evaluate mobile development frameworks. To achieve this, the following specific objectives are set;

1. To determine the factors that influences the rate of increase of usage of mobile applications.
2. To determine the current mobile development frameworks in use.
3. To identify the most suitable mobile development framework for development of mobile applications.
4. To develop mobile application for water usage management for Alice and Fort Beaufort communities.

1.8Significance of the research

This research will equip mobile application developers to gain more insight into mobile development frameworks available, thus helping developers to choose the most appropriate framework for their project. The research is significant in a sense that, by giving knowledge into reported mobile development frameworks, it makes an undertaking effective and feasible, and how achievement can be estimated with regards to mobile application development. The significance of open-source as a figure for making decision on a framework must not be neglected because it might result in technical debt if not accounted for since one is using libraries maintained by third parties.

1.9Outline of the research

The rest of the research is structured as follows; Chapter 2 reviews literature and discusses mobile technologies and platforms. It gives a brief discussion on various mobile operating systems. It includes a brief overview of mobile application development frameworks. A case study of mobile applications in the water sector has also been discussed in this chapter.

Chapter 3 presents the research methodology that is used in this research. It describes the research design, data collection instruments, data analysis and ethical clearance consideration of the study.

Chapter 4 discusses mobile application development frameworks. It focused on the evaluation of five mobile application development frameworks that were selected for research. The frameworks that are discussed here are PhoneGap, Xamarin, App Inventor, Sencha Touch and DragonRad

Chapter 5 discusses system design, implementation and testing. It includes a brief discussion of tools used in the development of the application. This also includes explanation of source code and user interfaces of the system. The chapter explains some of the errors that were generated during testing. It also discusses results that were obtained after the usability test. The chapter also presents the feedback received from the participants that tested the application.

Chapter 6 is the discussions and Conclusions. This chapter gives the overall discussion of results obtained in this study. The chapter concludes the whole dissertation and proposed future work to improve the research.

1.10 Conclusion

This chapter has given the introduction and background of the research including the problem statements, aim, objectives, research questions and the significance of the study. The next chapter present literature review of this research

CHAPTER TWO

GENERAL LITERATURE REVIEW ON MOBILE TECHNOLOGY

2.1Introduction

The section gives an overview of mobile technologies and platforms. It discusses various mobile technologies from the first generation to fourth generation. It also includes discussions of some mobile operating systems currently in use. It is essential to discuss these technologies so as to know what features or functions individuals want on their mobile phones when going to the mobile market. Mobile Phones are the world's fastest developing innovative technology in terms of scope and adoption. This trend makes them an enthralling ICT platform for tending to the striking challenges at all levels (Mongi et al., 2015).

2.2Mobile technologies/platforms

The first generation (1G) of wireless telephone technology and mobile telecommunications emerged around the 1980′s. The pioneering 1G networks at that time used only analog signals and voice signaling. In 1G network, voice calls were simply regulated to a higher frequency, typically to 150mhz and up (Sharon, 2008, Mir and Kumar 2015). The first cellular network using 1G standard was introduced by NTT (Nippon Telegraph and Telephone) in 1979 in Japan (Fendelman, 2018).

The second generation or 2G was launched in 1991 by Radiolinja for the GSM standard. This technology was based on binary codes like a series of zeros and ones (Mir and Kumar 2015). 2G allowed for enhanced data services and also introduced short messaging service (SMS). This allowed for greater privacy, efficient data transfer and also less expensive equipment. Two revisions or additions to this generation are sometimes referred to 2.5G and 2.75G (Sharon, 2008). The combined introduction of General Packet Radio Services (GPRS) and the usage of Code-Division Multiple Access (CDMA) one networks collectively came to be known as 2.5G (Fendelman, 2018). GPRS provided data transfer rates from 56-115kbit/s. So, services like WAP (Wireless Application Protocol) and MMS (Multimedia Messaging) were introduced, along with Internet services. 2.75G was the name given to the evolution of EDGE

(Enhanced Data rates for GSM Evolution) or Enhanced GPRS (EGPRS) (Sharon, 2008). This was due to the introduction of 8PSK encoding, which facilitated higher data transfer rates of up to 236.8kbits/s, almost triple of the previous rates (Fendelman, 2018)

Third generation (3G) was introduced by NTT DoCoMo in Japan, in 2001. 3G used different radio frequencies, so it required different equipment to achieve the new high data transfer rates (Fendelman, 2018). 3G data transfer rates are 384kbits/s to 2Mbits/s, it allows for services like video calls, video conferencing, online conference call, mobile TV, online gaming that were not available to the previous generations. These speeds are broadband equivalent, so the applications and capabilities are enhanced greatly. 3G also provides greater security and privacy. As with 2G, minor evolution of the standards resulted in 3.5G and 3.75G. Again, these standards allowed for higher data transfer rates, exceeding 2Mbits/s, reaching about 14Mbits/s (Taylor et al, 2018).

4G is the fourth generation of cell phone mobile communications standards. It is a successor of the third generation (3G) standards (Mir and Kumar, 2015). A 4G system provides mobile ultra-broadband Internet access, for example to laptops with USB wireless modems, to smartphones, and to other mobile devices (Cassavoy, 2018). Conceivable applications include amended mobile web access, IP telephony, gaming services, high-definition mobile TV, video conferencing and 3D television. Recently, Android and Windows enabled cellular devices also falls in the 4G category. One base advantage of 4G is that it can, at any point of travelling time, provide an internet data transfer rate higher than any existing cellular services excluding broadband and Wi-Fi connections (Cassavoy, 2018).

Long Term Evolution (LTE) is a wireless broadband technology designed to support roaming internet access by mobile phones and other handheld gadgets. Since LTE offers critical advancements over older cellular communication standards, some refer to it as a 4G technology, along with WiMAX (Worldwide Interoperability for Microwave Access). With its architecture based on Internet Protocol, unlike many other cellular internet protocols, LTE is a high-speed connection that supports browsing websites, VoIP, and other IP-based services. LTE can theoretically support downloads at 300 megabits per second or more (Bradley, 2019)

Mobile phone remains one of the most prevalent information technologies both in developed and developing countries. Its uses continue to expand. Apart from using mobile phones for making calls, there are other features which are gaining popularity nowadays like Camera, Music, Global precipitation system (GPS) etc. These built-in features are provided by all the major available mobile Operating Systems such as Android, iphone, Blackberry, Symbian, webos, JME and Windows phone (Palmieri et al, 2012). All the mobile operating systems mentioned above are popular in the market due to their uniqueness. For instance, Android which is based on Java is freely available, meaning that it is free to any cell phone carrier. IPhone is popular due to its customizable operating system which you can be used to download applications made by Apple like games, GPS, and other tools. Blackberry is a smartphone with a multimedia player and third-party software installation (Mir and Kumar 2015). Although these operating systems are so rich in libraries and built-in features, there is still increasing market pressure to introduce new operating systems to match users' high expectations. The basic architecture and programming language support of these operating systems are different from each other. One built application is not compatible for all other operating system, which forces the application developers to build the same application again for other operating system. The next sections give brief overview of the various mobile operating systems mentioned above.

2.3Mobile operating systems

Android: Android is a Linux-based operating system purposely designed for touchscreen mobile devices such as Smartphones and tablet computers, developed by Google in conjunction with the Open Handset Alliance (Singh, 2014). Android has a large community of developers writing applications that improve and extend the functionality of devices. Applications are primarily done in a customized version of Java, and these applications can be downloaded from online stores such as Google Play (formerly Android Market), the application store run by Google, or third-party sites (Khan and Haris, 2017).

iOS (iPhone OS): iOS is a mobile operating system developed and distributed by Apple Inc. It runs on devices such as iPhone, iPod touch, and iPad. The operating system directs or controls the use of device hardware and provides the technologies needed to implement native applications (Götz et al, 2017). The iOS Software Development Kit (SDK) contains the tools and interfaces needed to develop, install, run, and test applications. Applications are built using the iOS system frameworks and Objective-C based on C programming language and run directly on iOS. Apple does not license its operating system (iOS) for installation on non-Apple products. The user interface of iOS is on the general idea of direct manipulation, using multi-touch gestures. Elements that are used to control the interface consist of sliders, switches, and buttons (Andahi, 2018).

Blackberry OS: The Blackberry Operating System is a software platform developed by its manufacturer Research in Motion (RIM) (Okediran et al, 2014). Its OS provides multi-tasking that maximizes and supports use of the devices with specialized input devices including: trackball, trackpad and touchscreen. Blackberry device is equipped with the RIM implementation software of proprietary wireless-oriented protocols. The Blackberry device is always on and participating in some form of wireless push technology. The current version of the Blackberry OS has numerous capabilities and features. These features include over the air activation to synchronize contracts and appointments with Microsoft outlook, a password keeper program to store sensitive information and ability to customize your Blackberry data. Third-party developers can develop software using the existing Blackberry Application Programming Interface (API) classes, though applications that make use of certain functionality must be digitally signed (Bala et al, 2015).

Symbian: Symbian is a mobile operating system (OS) and computing platform designed for Smartphones and currently maintained by Accenture (Edwards and Barker, 2004). The Symbian platform is a real time, multi-tasking operating system specifically designed to run well on resource-constrained systems, improving performance and battery life whilst minimizing memory usage. Symbian platform is a replacement to Symbian OS and Nokia Series 60. Unlike Symbian OS, which needed an additional user interface system, Symbian includes a user interface component based on S60 5th Edition (Maji et al, 2010). The language for Symbian operating system is C++ using Symbian software's coding standards. Research shows that languages like Java ME, Python, Ruby, .NET, Flash Lite, and Web Runtime (WRT) Widgets can also be used to program for Symbian devices (Maji et al, 2010).

webOS: webOS, also known as Open webOS, formerly called HP webOS is a mobile operating system running on Linux kernel (Kochanska, 2014). It was initially developed by Palm but has been acquired by Hewlett-Packard (HP). HP provides resources for webOS developers and instructions for taking part in the HP Palm Developer Program. With webos SDK, developers are able to develop applications for webos devices such as the HP Veer and the HP Touchpad. Included in the SDK is also the PDK, which gives developers the privilege to have access to compilers, utilities, coding libraries, scripts, and documents that help developing in C/C++ (Allen, 2009, Kochanska, 2014). Applications can also be written in languages like JavaScript, HTML5, and CSS using either the Mojo or Enyo framework. Navigations on webos devices use multi-touch gestures on the touchscreen. The interfaces use “cards” to control or direct multitasking and represent applications (Allen, 2009).

Windows Mobile: Windows Mobile is a mobile operating system for smartphones and Pocket PCs (Kamboj and Gupta, 2012). In February 2010, Microsoft announced Windows Phone to replace Windows Mobile, with the new operating system incompatible with Windows Mobile devices and software (Renner, 2014). As a result, Windows Mobile has been discontinued. Windows mobile is a variant of Windows CE for mobile phones. Windows CE was originally developed for palmtop computers and Pocket PC pdas with stylus-touch screens, and later adapted for use with keyboard-equipped smartphones (Kamboj and Gupta, 2012). Phones have become the largest installed base for CE, though market share has fallen since the introduction of Android and iPhone. Windows Phone 7 is a substantial redesign that provides a richer user interface. Windows phones are programmed in C++ (Renner, 2014).

2.4Overview of mobile application development frameworks

There have been various advances in mobile computing technology over the years. Presently, mobile devices exist in a variety of form factors, for example watches, smartphones, tablets, and cars. These gadgets are furnished with powerful processors with ample capacity and different collection of sensors capabilities (Mbayen and Edgard, 2013). Coupled with advances in development frameworks, operating systems and middleware for mobile devices, programmers would now be able to benefit rich programming APIs to develop applications that leverage these advances in hardware. Current mobile application stores contain countless applications, and the diversity of applications available to end-users has additionally added to the popularity of mobile devices. These advances in hardware and software have made mobile devices feasible swap for PCs. At the same time, we are additionally seeing a move in the act of programming advancement due to a great extent to the elements of mobile application development frameworks. Until a few years ago, the task of developing applications was mostly confined to teams of software engineers, either in the open-source community or at IT organizations. Conversely, it is common nowadays for individual or small groups to develop and distribute application via mobile application stores (Boushehrinejadmoradi et al, 2015).

There are different categories of mobile development frameworks available today. The first category, commonly known as native framework. When creating native applications, developers implement an application for one specific operating system using its software development kit (SDK) and frameworks. The application only runs on that specific OS environment. For instance, applications for android are commonly modified in Java, access its platform functionalities through Android's frameworks, and render its user interface by using platform-provided components. Also, applications for iOS use Objective-C programming language and Apple's frameworks. If multiple operating systems are to be supported by native applications, they need to be developed one by one for every OS. Users will install native applications from the platform's application store or other platform-provided installation means. They receive an application that has the look and feel of the platform (Charland and Leroux, 2011, Firtman, 2010). Examples of native frameworks include Android Studio, App Inventor, and Apple Xcode. These frameworks mostly support one target platform. Developers can also use a cross-platform mobile development framework to develop native applications for different mobile operating systems from the same code-base. Applications developed this way are very similar to native applications, but cannot be considered true native as they have some key differences to applications developed using the platform-appropriate SDK and development methods (Boushehrinejadmoradi et al, 2015). The Xamarin framework allows developers to automatically build Android and iOS apps using this code base. Likewise, Apportable is iOS development framework; however, developers build applications using Objective-C and the iOS SDK, and leverage Apportable to produce Android applications from this code-base. The native development approach for mobile platforms leads to multiple versions of the application's code-base, which are difficult to maintain and evolve over time. Therefore, developers are increasingly adopting cross-platform mobile application development frameworks. These frameworks allow developers to program the application's logic once in a high-level language, and provide tool-support to allow the application to execute on a number of mobile platforms.

Another category of mobile development approach is Web-based framework. Mobile web applications are mostly implemented with HTML, CSS, and JavaScript. It uses browser as its runtime environment and in this way, capitalizes on the good browser support of mobile platforms. When using this approach, developers implement their application as one web site optimized for mobile devices, which the web browser then interprets. The optimization has to account for the different screen size and usage philosophy of mobile devices (Charland and Leroux, 2011). Due to the standardized technologies, the web site is often accessed in a very similar means by mobile browsers on all platforms. However, mobile web applications at times cannot use devices specific hardware features, for example, camera or GPS sensor. They typically cannot be installed on the mobile device however are retrieved through Uniform Resource Locator (URL). Usually, web applications will at least look and behave like common web pages, varying in appearance and behaviour from the native user interface elements provided by the platform (Charland and Leroux, 2011, Firtman, 2010). Examples of web application development frameworks include PhoneGap, Sencha touch and IBM MobileFirst.

Hybrid applications combine web and native technologies in an attempt to resolve the issue with web applications' restricted access to hardware, when using common web technologies. Usually, the application is implemented as a website using standard HTML technologies, however rather than displaying it in a web browser; it is wrapped as a native application using a specialized engine (Ottka, 2015). The page is displayed in a regular WebView or similar feature inside the application, and the engine provides frameworks for accessing the device's native functionalities, usually through JavaScript (Heitkötter et al, 2012). Hybrid applications are installed on the device the same way as native applications. Hybrid application caches its data locally on the device on installation, removing the requirement for constant data connection. This also increases its speed and responsiveness over a pure web application (Hartmann et al, 2013). Like web applications, hybrid applications can use web technologies to render a web-like GUI or imitate native GUIs (Hasan and Haque, 2016). There are five mobile application development frameworks which have been selected from the categories of frameworks elaborated above. The selected frameworks have been evaluated and discussed in chapter 4 of this dissertation. The next section provides a general idea of how mobile applications are used in the water sector.

2.5Overview of mobile applications: A case research in the water sector

This section attempts to unload and comprehend what role mobile applications may play in the water sector. When mobile phones were first introduced in the market only the wealthy could afford the technology. Nowadays, mobile phones have become not only accessible to every household but also important tool for receiving and sharing information across the world. Ordinary citizens including rural farmers, fishermen and livestock keepers can afford to own mobile phones (Mongi et al., 2015). As at 2013, six billion individuals out of the worlds' seven billion population own mobile phones (UN News Centre, 2013). Mobile phone can give comprehensive information gathering capability utilizing voice-based strategy or Wireless Internet Gateway and this makes it important tool in water resources management. However, the use of mobile phones has not been put to its full potential in the water sector (Champanis et al, 2013). Otuke (2016) argues that the use of ICTs including mobile phones in the water sector has not reached its full potential.

Mobile services can assist the water sector in mitigating governance challenges by increasing transparency, accountability and participation by various stakeholders (Hellstrom and Jacobson, 2014). ICTs in the form of mobile phones can aid in making data transfer more proficient, lessen physical data collection faults, and promote effective monitoring at an effective cost (Ndaw and Niyungeko, 2015).

For instance, Lifelink, an application developed by Denmark-based Grundfos AS, employs submersible borehole pumps powered by using solar or wind power to offer water to upraised capability tank. Gravity forces the water to stream into a faucet unit in a small building, which also serves as a payment facility. Customers purchase water credits that have been saved on a smart card making use of mobile phone banking (Mollins, 2012). They insert the smart card into an opening at the faucet unit to buy water. Some percentage of the money goes to aid and renovation. Lifelink group of workers are capable of monitoring water temperature, strain, and the sum of water tapped from any place within the world through the internet. The company, which provides this technology in urban and rural communities, says it has implemented drinking water supply frameworks for 100,000 individuals in Kenya and is extending into other African nations and Asia (Mollins, 2012). Figure 2-1 is example of Grunfos Lifelink 'Water ATM'.

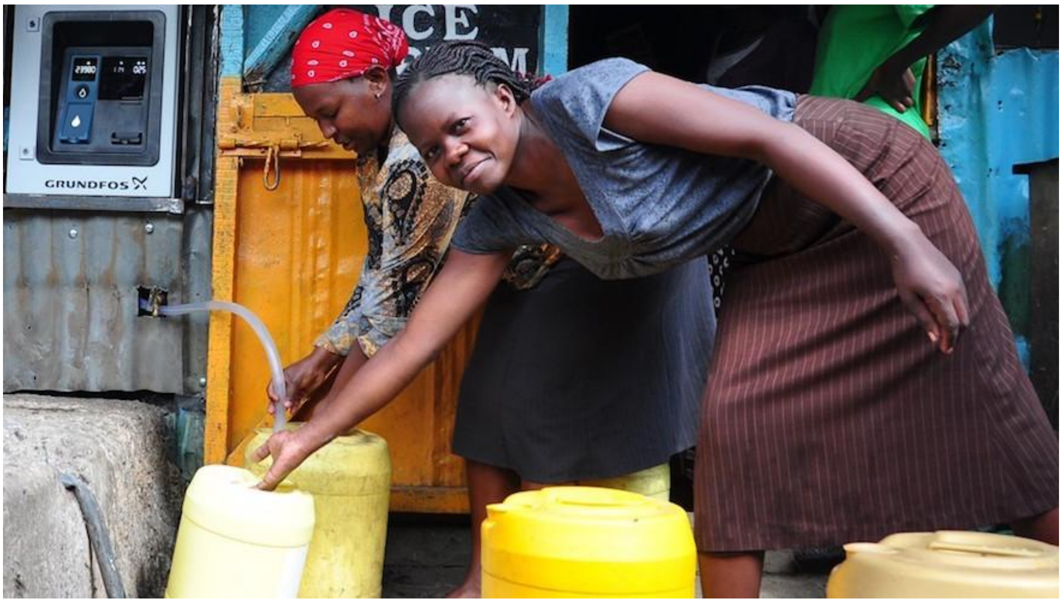


Figure 2-1: Grunfos Lifelink 'Water ATM' example.

The mWater platform is a service oriented mobile-to-web monitoring system. The platform was established by a mobile solutions company called Manobi in 2007. Presently, mWater is used to monitor about 35,000 water points in four countries namely; Senegal, Mali, Benin and Niger. mWater pilot projects were conducted in these four countries to determine the viability of the use of ICT tools to improve the monitoring and management of rural and small town piped water projects (Foster et al, 2012). As at 2013, mWater was engaged in around 252 water frameworks in Benin, Niger, Mali and Senegal (Annerose and mWater, 2018). In all about 246900 individuals are served in Niger and 415900 individuals assisted in Mali. The mWater encompasses an ICT design structure whereby mobile applications and web platforms are produced to back the life cycle of water supply frameworks, from development to operation which includes information gathering, technical and monetary administration. In Benin for instance, the use of the mWater platform brought about new drawings of resources and water systems for 51 rural water-point. Also, in Senegal, mWater managed to sustain the mapping of 70% of existing water systems that is about 28000 water centres (Ndaw and Niyungeko, 2015).

To test water quality, mWater in collaboration with United States Agency for International Development (USAID), developed a mobile enabled application that makes it easy for communities to enter data from supported water sample and then provides test results for a particular water source. Users receive instant feedback regarding their water sample tested, while the data is uploaded to the cloud for mapping and sharing with others (Foster et al, 2012).

Water Quality Reporter (WQR) is a low priced water testing application that can be used in low-resource locations. Water quality testing in rural regions is a challenge due to lack of appropriate equipment and infrastructure for water quality examination, hence the development of this low-cost application to aid low-resource and emerging countries (Rivett et al., 2013). Water Quality Reporter is available in South Africa, Vietnam and Mozambique. The Water Quality Correspondent mobile phone application is based on an open-source JavaRosa platform and uses data exchange by general packet radio service (GPRS), which is less costly than SMS and enables the sending of finished forms with water quality information. The project furthermore utilizes a SMS-based gathering methodology which utilizes Rapid SMS to supervise incoming messages from basic phones that do not support the Java platform (Hutchings et al, 2012). Water Care unit administrator from Amathole District Municipality, which covers rural zones of South Africa's Eastern Cape Province, lauded the application and said it is helpful to utilize as you will be able to click on the application at any time and see water quality outcomes without requiring to be at the office. The application is likewise being utilized within the Chris Hani Municipality around Queenstown, according to Francois Nel, Assistant Director of Municipal Health Services and Environmental Management, who said it was simple to apply and required small training (Labuschagne, 2012).

A mobile application called WaterQuality, is the coordinated effort between Northern Kentucky University and the Foundation for Ohio River Education, which collaborated to enhance the effectiveness of water quality observing, while providing a stage for learning. WaterQuality enables clients to make a profile for an observing site in which they can enter chemical and bacterial monitoring information acquired from standard monitoring equipment. The application at that point gives charts and outlines that characterize and give healthy ranges to different water quality measurable (Redling, 2013).

NFC RFID-Tracked Drinking Water is Near-field communication (NFC) enabled phones with radio-frequency identification (RFID) used by Haitian water supervisors and specialists to track chlorine levels (Arnold and Colford, 2007). Haiti had trouble providing clean drinking water for about 9.7 million people; a circumstance which was aggravated by earthquakes and a cholera epidemic resulting in the loss of thousands of lives (Ndaw and

Mwangi, 2015). The mountainous territory combined with poor transportation and communications systems made distribution of clean drinkable water problematic (Greenemeier, 2011). With the introduction of NFC RFID-Tracked Drinking Water over 35000 families in Haiti were able to have access to clean water and frequencies of cholera among users were reduced to about 50% (Greenemeier, 2011).

SweetSense is IoT (Internet of Things) service platform, sensors were deployed on hand pumps, electric pumps, and a variation of other water, energy, and infrastructure assets to measure rates of utilization and functionality. This service platform is available in Rwanda and other developing countries. Over the next five years, SweetSense has used its cellular-based, remote observing sensor IoT platform to inform project partners when a water point fails to operate so it can be repaired instantly, rather than weeks or months afterwards. End users include local technicians, NGO program supervisors, province government specialists, and local entrepreneurs who review easy-to-understand sensor information from a web dashboard. Sensors are used not only to alert project partners of water-point uptime and performance, but moreover to supply basic market research data that can underpin local business models enabled by the sensors and enhance water accessibility (GSMA, 2016 and Aeris, 2017).

MajiVoice was developed with the aim of improving efficiency, minimizing response time indistribution of urban water service providers through provision of a well-organized and clear system to coordinate and solve consumer grievances raised to Water Action Groups (WaGs) (Welle et al, 2016). MajiVoice is a mobile to web framework based on open-source computer program that serves as a stage for consumers to log their complaints to the water service provider. Complaint logging is done through registered short code service (USSD), structured SMS, or on the web employing a computer or internet empowered phone. Complaints can still be received by phone calls and afterward physically logged into the Majivoice framework by the Water Service Provide. Majivoice is a valuable instrument for decision-making and some advantages is acknowledged from its utilization including, enhanced work process, enhanced reaction time to client protestations and in this manner improved client trust in the utility (Ndaw and Mwangi, 2015). Figure 2-2 shows a decreased resolution time and number of complaints processed through Majivoice in 2014.

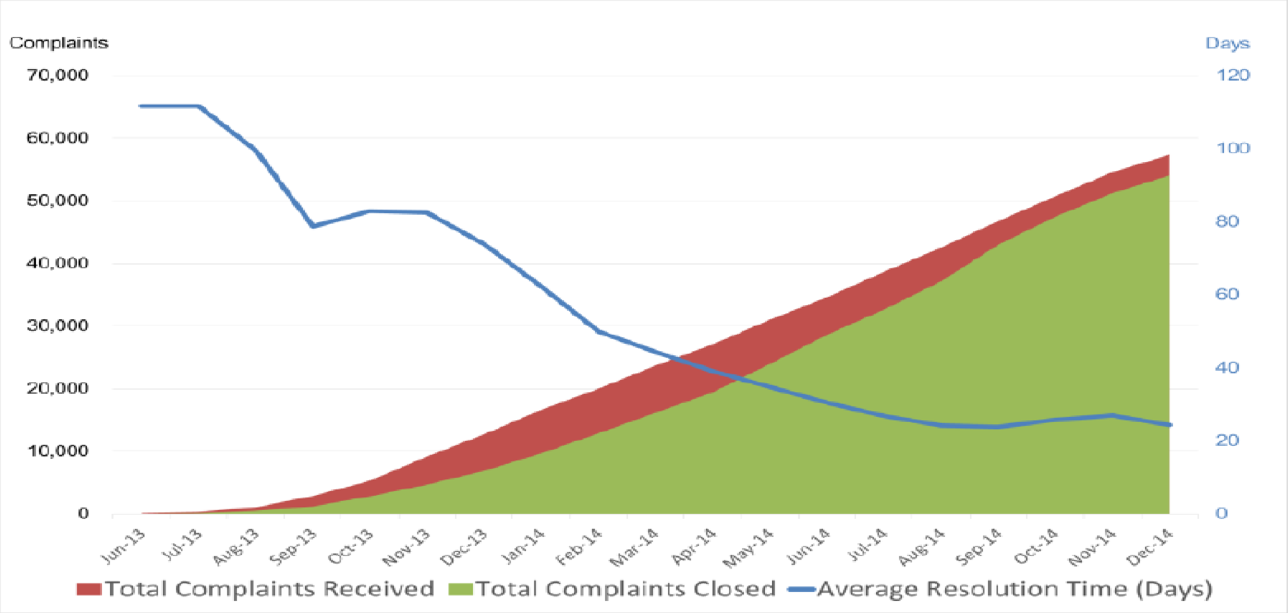


Figure 2-2: Decreased resolution time and number of complaints processed through Majivoice (Ndaw and Mwangi, 2015).

Smart Hand Pumps are mobile phone based low-cost networks available in Kenya and Zambia. It is an automated monitoring technology; SIM cards are set inside hand pump handles to give real-time monitoring of hand pump performance. It Monitors the movement of their handles and calculates the amount of water withdrawn from the pump. It updates the district and water managers through text messages as to water usage. Water Managers can immediately know when and where a pump needs fixing (Welle et al, 2016). Figure 2-3 is picture of smart hand pump in Kenya.

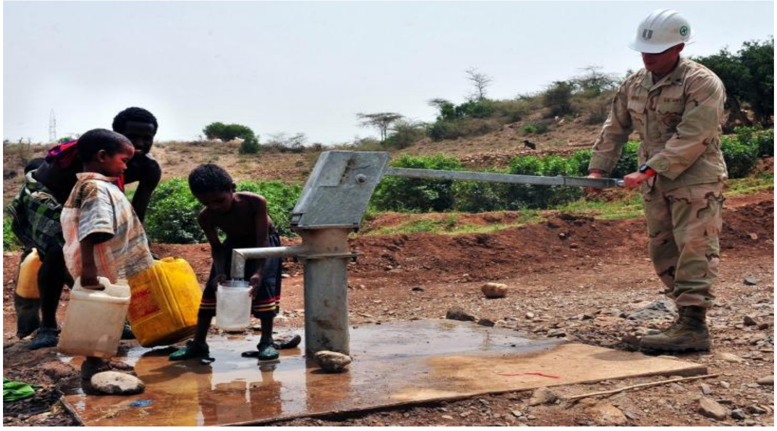


Figure 2-3: Smart Hand Pump in Kenya

Human Sensor Web is a project intended for community-driven services for focused and geo-referenced monitoring of water distribution and sanitation coverage in Zanzibar (Tanzania). The individuals use mobile phones to report if a water point is dry, faulty, or if the water is not clean for consumption (Welle et al, 2016).

Field Level Operations Watch (FLOW) is a GPS and Mobile Phone technology which is open source data collection tool that permits users to create surveys that include texts, photos, videos and GPS coordinates. Data about the location and functionality of water points as well as access levels of water and sanitation at household and public institution level are collected using Android phones (Tsega et al, 2013).

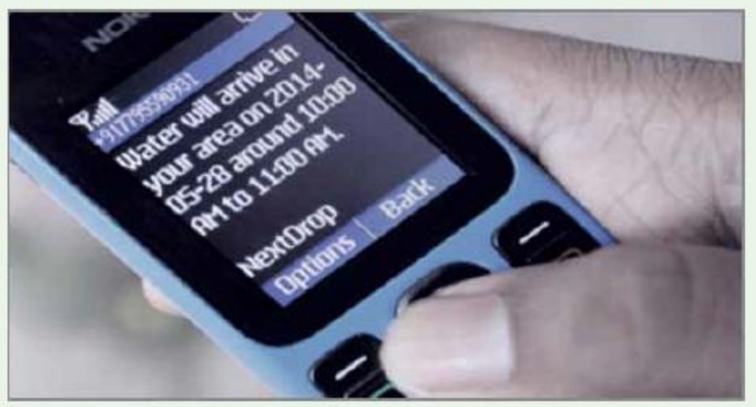
There is an initiative to improve capability of estimating floods through early cell phone and flag alerts is being tried in Bangladesh to relieve the effects caused by disasters and primary floods,. This activity has guided a community-based cell phone early caution framework in five Union Parishads within the major river banks of Bangladesh (Nguyen and Shaw, 2011). The early caution system is activated whenever there is a regional flood caution. During the regular flood season, about 1411 family units within the five pilot ranges get daily early cell phone and flag cautions. This initiative ensures that the disaster areas get alerts of possible flooding in time, supporting them to leave their homes and look for shelter in allocated locations hence preventing the devastating effects of natural disasters (Shaw, 2011). Since the dispatch of the cell phone early warning system, disaster-related fatalities and misfortunes of resources have diminished altogether within the five pilot regions.

Jisomee meter which means "Read your Meter" is a mobile invention based solution that has allowed the urban poor to conveniently access water services at the lowest rates. This metering was implemented in Soweto South Africa. It has empowered the urban poor in informal settlements to read their meters and make payments, submit meter reading via SMS through their mobile phones (Heland et al, 2015).

Mobile phone payments or mobile money might be a solution to increasing income collection from consumers (Foster et al., 2012). Mobile money appeared to be effective in East Africa, especially Kenya, because it permits funds to be rapidly, effectively, and cost effectively exchanged between parties that were formerly "unbanked". For water Service providers, mobile water payments offer a cost effective approach to bills collection and alleviating the cycle of poor financial and operational execution in two important ways. To begin with, if transaction costs incurred by clients when paying bills are lessened, payment becomes more convenient and volume of payment is likely to increase thus improving income collection. Secondly, mobile water payments reduces work load, rental and cash security costs thereby enhancing operational efficiencies of water service providers (Champanis et al., 2013). Moreover, water service providers can offer more adaptable payment options with small extra regulatory burden, meaning that serving the poor and extending the client base, might end up being commercially rewarding (Hope et al., 2011).

DropDrop is another type of mobile application used in the water sector. The application runs on Android phones and was developed by iCOMMS in Cape Town (South Africa) for users to track their water consumption. For users to monitor and access their daily water usage and estimated bill, they need to enter the water meter readings into the application on regular basis. Information on methods for water conservation, municipal contacts details is also provided in the application (Rein et al, 2013).

NextDrop is a mobile phones platform that provides families and individuals with precise and convenient data about community piped water delivery, including when delay occurs in supplying water or pipe damage that could influence water transportation. The application enables water authorities to get instant information about the position of their distribution system (Kumpel et al, 2012). The application is able to deliver weekly messages to families updating them about the status of water supply and the reasons for non-supply. The original intention was to ensure that individuals living in remote regions will not have to spend between twenty to forty hours out of each month holding up for water. NextDrop now serves about 75000 community members within the twin cities of Hubli-Dharwad, Punjab, India (Kote and Olmos, 2011, Ndaw and Niyungeko, 2015). Figure 2-4 is example of SMS NextDrop sent to a user.



+917795549931

Water will arrive in your area on 2014-05-28 around 10:00 AM to 11:00 AM.
NextDrop

Figure 2-4: SMS from NextDrop to a user (Ndaw and Niyungeko, 2015)

Tambua is an application used for identification of employees by customers at their residences. Customers have often cited insecurity as a reason for denying Water Meter Readers access to their homes. This application has been helpful in curbing customer insecurities as it links them directly to the company database where they are able to identify staff by using employee's number. As a result of this application there has been an increase in meter reading and more staff are being allowed entry into customers' yards for meter reading. Revenue levels have been reported to have also increased (Hope et al., 2012).

Although there are numerous water management tools available, these systems are not usually integrated and so there is no communication between one another. This scenario prevents Integrated Water Resources Management to be fully implemented. Adoption of mobile applications water tools has been slow in some developing countries because of lack of resources and infrastructure. Although mobile phones are available, the consumers need to be educated on how to use the tools for paying bills, monitoring water use and reporting water leaks (Brandes et al., 2005, Ndaw and Niyungeko, 2015).

2.6Reflections on literature review

Mobile phone technology, structure and functionality have continued to witness tremendous improvements since its inception. The evolution of mobile phones is by far one of the fastest advancements that mankind has ever witnessed. The growing necessity and expectations of users assisted the evolution of smartphone from a basic mobile. Mobile phones have witnessed enormous headways in connectivity, multimedia, security, productivity, and gaming ever since their introduction. Mobile application development is driven by advancements in technology which requires that businesses do not limit their visions for next few years. The rapid rate of technological change and progression within the mobile market and the fact that the market share of smartphones is changing quickly between different systems make it challenging for developers to strategically plan a development project solely from a technical point of view.

As far as connectivity standards are concerned, the literature review indicates that there has been a fast shift ranging from the basic 1G, which was the primary generation of wireless telecommunication technology to the present day's 4G LTE network that can support a peak data transfer rate of 1GBit/second. Previously, the handset supported either voice calls or connectivity to the Internet but not both simultaneously. However, with the growing bandwidth support of mobile technology, voice over LTE has started emerging to support voice calls without interrupting constant internet connectivity. At the beginning of the smartphone revolution, the iPhone iOS dominated the mobile market, however Google Android has now surpassed iPhone regarding market share, due partly to the ability of the Google brand and partly to the platform's openness. Other mobile operating systems include the BlackBerry and Windows 10 Mobile. The question arises: which platform or framework are you looking to develop your mobile application on? Native (Android/iOS) or Web (MobileFirst/PhoneGap)? For now, web applications are not yet able to emulate native applications perfectly. Native applications dialogue directly to the operating system, while web applications exchange to the browser, which dialogues to the OS. This further layer that web applications need to go through makes them slightly slower and coarser than native applications. However, the fast stride of mobile web browser development is allowing more native device features to be accessed by web applications, slowly reducing the gap between web and native.

The analysis of various current and past mobile applications in the water division has demonstrated that mobile phone advancement and usage is complex, Therefore, to apply mobile phones appropriately and successfully in the water sector requires proper understanding. There is clear indication that mobile applications can aid management functions of the water sector and the literature is optimistic about the value of mobile application regarding development.

When considering a mobile approach, one of the significant decisions for many developers is to decide between varied mobile application developments frameworks. Presently, there are mobile applications for every facet of life and undoubtedly, these applications have made and are making life extremely easy for people on a daily basis. The challenge remains how to optimise the satisfaction and experience each of the applications have been designed to give its users. There is no reasonable doubt that the mobile application development industry will remain on the flourishing side especially with daily emergence of superior add developers and development companies that are willing and ready to provide the required satisfaction and needs of mobile applications users. Hence, application developers continuously are working to keep up to date with the demands. Therefore, there is a requirement of analysis into available mobile application development frameworks in order for developers to make informed decision when deciding on frameworks to develop applications that meet satisfaction and needs of its users. This analysis can contribute to concepts relating to selecting one development framework over the other.

2.7Conclusion

The chapter has given an overview of mobile technologies and platforms. It has discussed various technologies from the first generation to fourth generation mobile technologies. It also included discussions of some mobile operating systems currently in use. The analysis of the numerous mobile applications in the water division both past and present, highlights the complex and evolving nature of mobile technologies and the need for proper understanding of their fluidity in order to apply mobile phones appropriately in the water sector.   The next chapter present the research methodology of this research.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1Introduction

The research methodology usually explains how research is conducted. The best approach to efficiently tackle a research problem is by sensibly embracing different steps involved in the research process (Pérez, 2009). Hence, this chapter provides the general description of the research methodology that was used to conduct this research.

3.2Research Design

Research design permits the researcher to plan, structure and carryout the investigation that will be utilized to assemble proof of tests or experiments undertaken (Creswell, 2017). This research uses mixed methods approach that includes literature review and evolutional prototyping. Mixed methods research is a methodology for conducting research that involves a combination of various research methods. The rationale for choosing mixed methods design approach is that, it is suitable for answering research questions that neither quantitative nor qualitative methods could answer alone (Tashakkori and Creswell, 2007). Mixed methods can be used to gain a better understanding of connections or contradictions between qualitative and quantitative data; they can facilitate diverse avenues of investigation that improve the evidence and enable questions to be answered more profoundly (Wisdom and Creswell, 2013). This work makes use of a combination of literature review and evolutional prototyping.

Literature review was used mainly to conduct preliminary study on various mobile development frameworks so as to ascertain which one is recommended for building mobile applications. Literature survey was also used to understand the background of the study in general. This involved the history of mobile development frameworks and their attributes and experimental methods.

Through literature review five mobile frameworks were identified and assessed based on various comparative techniques as presented in chapter four. The frameworks were evaluated on seven assessment criteria and thereafter, one framework out of the five evaluated, was chosen for the development of mobile application for water usage management as proof of concept. However, to develop application for a particular group of people there is a need to gather requirements from intended users. In this case, a qualitative method was employed as a means to conduct interviews and carry outfield observations in both Alice and Fort Beaufort communities, two of the study sites. The aim was to obtain user requirements for the application. This, therefore, suggest that an embedded design of mixed methods approach was adopted. The Embedded Design is a mixed method design in which one data set provides a supportive, secondary role in a study based primarily on the other data type (Baran, 2016, Bian, 2015). The mobile application developed is called *SaveAmanzi* and is developed through evolutionary prototyping which is presented in the next section.

3.2.1 Evolutionary prototyping

There is a trend in the software engineering community to build systems using prototyping as the model on which to base development. Increasing costs of software and the lower rates of systems addressing users' needs are cultivating an enthusiasm for prototyping as the primary methods of creating software. Evolutionary prototyping is a form of software system creation in which developers gather the most complete, best-understood requirements possible; design and implement a prototype; and have the customer evaluate the prototype (Vennapoosa, 2012). The emphasis is on accomplishing functionality for demonstrating a portion of the system to the eventual user for feedback and further system growth. Evolutionary prototype life cycle begins first by delivering an initial fielded prototype then subsequent modifications and enhancements result in delivery of further more mature prototypes. In other words, the prototype is first designed and then it is coded and tested. The developers initially do the testing and the results are used to derive the specifications. This process continues until the user accepts the eventual product. This approach is generally used wherever specifications cannot be developed upfront (Antón et al, 2001). Figure 3-1 shows evolutionary prototype model.

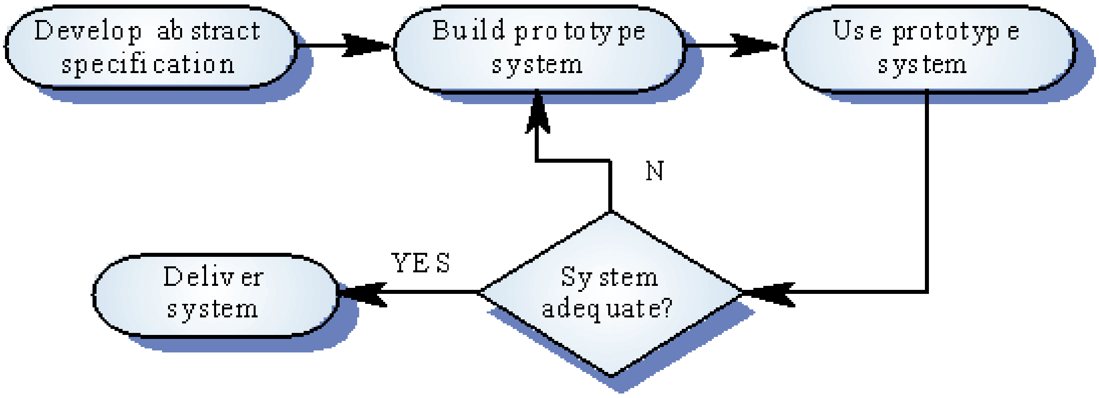


Figure 3-1: Evolutionary Prototype Model

The following steps explain how each phase was executed:

1. Development of abstract specification: In this phase, we identified basic requirements and planned for the creation of a prototype for the *SaveAmanzi Application*. We used questionnaires to gather requirements from both Alice and Fort Beaufort communities. The data analysis of the questionnaire is found in Appendix 3.

2. Build prototype application: In this phase, we created initial prototype and used it to define the requirements. Quick prototype is implemented when requirements were known. It includes the important aspects like input and output format of the application and we focused on those features which are visible to the user rather than the detailed plan.

3. Use prototype application: This was done through experimentation by the developer and also using participants taken from the potential users from Alice and Fort Beaufort communities. The intended users are the best people to get feedback regarding whether the application is good or not.

4. System Adequate: At this point we review our prototype and add or redefine requirements. We make further changes according to criticism from potential users. The process of refining the prototype is continual until all the requirements of users are met.

5. Deliver the application: When the users are satisfied with the developed prototype then the application is developed on the basis of final prototype.

Evolutionary prototyping is one practical approach to create systems where it is difficult or impossible to establish a detailed system specification document. The key to success in the evolutionary prototyping approach is to use techniques, which allow for rapid system iterations (Despa, 2014). Suggested changes may be incorporated and validated as quickly as possible. This may mean using high level programming language for software development. Special-purpose environments and integrated software tools may be used to support and accelerate the development process (Larman and Basili, 2003). An important difference between evolutionary prototyping and a specification-based approach to development is in verification and validation. Verification is only meaningful when a program is compared to its specification. If there is no specification, verification is impossible. The validation process should demonstrate that the program is suitable for its intended purpose rather than its conformance to a specification (Rai and Dhir, 2014).

By the iterative nature of the prototype's evolution, the customer gets the opportunity to accept, reject, or change the prototype, and in turn, accept, reject, or change the project's requirements (Larman and Basili, 2003). The most needed functionality will be produced first, thereby solving the user's basic needs. This prevents the users from waiting through the entire software process to get a functional system and having to wait for another period of time if there are any discrepancies in the user's requirements (Mahapatra and Goswami, 2015). This model establishes user engagement with the system. This increases the chances of producing a system that meets the users' requirements and also more likely to have users commit to the use of the system. However, this procedure requires frequent feedback from the user in order to validate and judge the success of the system (Despa, 2014). Another benefit associated with evolutionary prototyping is that of discovering a more complete set of requirements for the software project. Once a prototype baseline is established, users can often find additional functions that the prototype must provide. It allows systems to be developed and delivered rapidly and system development costs are reduced (Alshamrani and Bahattab, 2015).

3.3Data collection instruments

Data collection instruments refer to techniques or ways used to gather information, for example, review of literature, questionnaire, tests, and organised interviews (Khomba 2011).

The data collection instruments measure the responses to the main research questions posed in the research (Vosloo, 2014).

In this research, literature review and questionnaire were used to obtain data relevant to the research's objectives and research questions. Background information on mobile developments frameworks is researched through a literature review. Online search engines and databases, including Google Scholar and IEEE Xplore, provide general information on the subject with keywords such as "mobile application frameworks", "mobile application development frameworks". More information on individual frameworks is gained by searching with the framework's name and visiting the official websites of the framework. To better understand the problem at hand and come up with a suitable mobile application, it is always essential in software development process to understand the behaviours and views of intended users of a particular system. The questionnaire was used to identify requirements for the development of the case study mobile application. The post-system test questionnaire also helped to acquire data relevant to the testing of the case mobile application.

3.4Data analysis

Content analysis was used to analyse the information gathered in this research. Content analysis is a technique that might be utilized with either qualitative or quantitative information. Content analysis is a research technique used to make replicable and legitimate inferences by interpreting and coding literary or textual materials (Elo and Kyngäs, 2008). By efficiently assessing texts (e.g., documents, oral communication, and graphics), qualitative information can be transformed into quantitative data (Randolph, 2012). One of the main advantages of content analysis is that it helps to reduce and improve information that has been gathered, while in the meantime delivering outcomes that may then inform the the use of quantitative techniques (Fass and Turner, 2015).

Mobile development frameworks' evaluation results are designated separately for each framework in subheadings. Data collected in this research is first coded and entered into the tables for simple analysis. Descriptive perspectives of the findings are displayed with the assistance of tables, graphs and charts alongside other descriptive statistical indicators.

While coded information allows reduction of errors in data analysis, the use of a Likert scale ensured effectiveness of quantitative information in this research. Likert scale is a psychometric response scale essentially used in a questionnaire to acquire participant’s preferences or level of agreement with a statement or set of proclamations (Bertram, 2013).

3.5Ethical consideration

Ethics is very important in research because it defines what is and is not permissible to do when carrying out research. According to (Creswell, 2017) ethics are the standard of principles that must be followed and adhered to by the researcher when conducting a research. For this study, permission was obtained from the University of Fort Hare Ethics Committee before proceeding with the research. The ethical clearance dealt with the issues of respect for participants, obtaining consent of participants and protecting the identity of informants and the information they provide from external or unauthorised parties and ensuring that participants are not exposed to any risk.

3.5.1 Respect

In this research concerted effort was made to show respect to participants at all levels during data collection process showing sense of appreciation and care for the feelings and well-being of the respondents.

3.5.2 Informed consent

Respondents were properly informed that participating in the research project was voluntary and they were free to discontinue with their participation if they so wish. Participants were provided clear and sufficient information to enable them understand what they were being asked to do.

3.5.3 Confidentiality and anonymity

According to Kaiser (2009) and Mohajan (2017) confidentiality refers to the researcher's responsibility to protect all gathered information from being divulged to any other person. Participants in this research were assured that the information they provide will be used purely for research purpose and will not be accessed by any other person. The issue of anonymity was addressed by ensuring that identity and privacy of participants were protected by not disclosing or linking their names to their responses. All the questionnaires in this research were self- administered and easy to complete. The researcher sought and obtained permission from the appropriate authorities or structures to interact with the community so that the target respondents can complete the questionnaire seamlessly.

3.5.3 Anticipated risk and precaution

The participants in this survey experienced no physical danger or pain. There was no psychological stress or emotional effects experienced beyond the normal levels to be expected in daily life. To minimize risks, the survey was voluntary; the participant can choose to stop taking part in the survey at any point in time during the course of the survey.

3.6Conclusion

This chapter has described the broader research methodologies and the specific research method that was used in this study. The chapter suggests that in general, literature survey allows a better understanding of the background of the study. It was significant to explain the research design, data collection instruments and data analysis procedures used for this study, including the ethical consideration. The chapter has shown that evolutionary prototyping model was used for development of the *SaveAmanzi* application. Due to the iterative nature of evolutionary prototype model, the user gets the chance to accept, reject, or change the project's requirements. The next chapter presents the evaluation of mobile development frameworks in order to gain knowledge about frameworks available when deciding on choosing appropriate mobile development framework for mobile application development.

CHAPTER FOUR

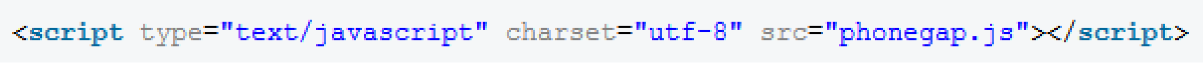
EVALUATION OF MOBILE DEVELOPMENT FRAMEWORKS

4.1Introduction

There exists extensive variety of frameworks developers can exploit to quickly deploy mobile applications. This section looks at mobile framework technique to application development and presents five frameworks that will be evaluated. Evaluation of knowledge on reported mobile development frameworks is essential in accessing the feasibility and effectiveness of any undertaking to develop new mobile applications. The five frameworks that are analysed in this chapter include PhoneGap, Xamarin, App Inventor, Sencha Touch and DragonRad

4.2Phonegap

PhoneGap is an open-source framework for developing native and cross-platform mobile applications (Pazirandeh and Vorobyeva, 2015). PhoneGap was given to the Apache Software Foundation (ASF) under the name Apache Cordova and thus is an open source distribution of Apache Cordova. Targeting several mobile operating systems demand knowledge of distinctive programming languages and frameworks. PhoneGap responds to this challenge by enabling a developer to create applications for various mobile platforms utilizing the existing recognizable web technologies (Amatya and Kurti, 2014). PhoneGap offers a set of JavaScript APIs that enables programmers to access the devices' functions such as camera, accelerometer, and contacts utilizing JavaScript. For instance, a code line below shows JavaScript code that includes the phonegap.js file which lets you call native API's of android (Suterwala, 2010).



```
<script type="text/javascript" charset="utf-8" src="phonegap.js"></script>
```

PhoneGap enables developers and organizations to develop commercial and open-source applications, and also gives opportunity to use licenses combination. With PhoneGap, mobile applications can be developed for Android, Bada, webOS, BlackBerry, iOS, Symbian, and Windows Phone operating systems. PhoneGap is a valuable tool for creating mobile applications utilizing cutting edge and current languages, for example HTML, HTML5, CSS, CSS3 and JavaScript, rather than using lesser known programming languages like Objective-C or other languages (Ribeiro and da Silva, 2012). Since the applications are created with no native code and using the normal web technology stack which are constant over various mobile platforms, the fundamental codebase continues as the same and ought to be compact to various mobile platforms with marginal to no modifications (Ribeiro and da Silva, 2012). The applications are packaged as native applications for each targeted platform utilizing a particular platform's SDK and can be made accessible for installation from each device's application store. User interface framework, for example, jQuery Mobile or Dojo Mobile can also be utilized and combined within the application development. It has the advantage of bringing numerous points of interest to skilled programmers and entices particularly web developers (Palmieri et al, 2012). PhoneGap architecture comprises 3 layers, namely Web Application, PhoneGap, and OS and local API's.

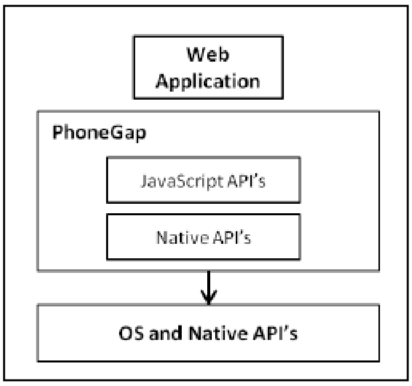


Figure 4-1:PhoneGap Architecture

In Figure 4-1, the top layer characterises the application source code. The centre layer comprises JavaScript and native API's. Basically, it is liable for interfacing between web application and PhoneGap layers (Ciman et al, 2014). The usefulness of centre layer is to preserve the relationship between JavaScript API's and local API's of respective mobile operating system. PhoneGap offers JavaScript API's to programmers which then enables them to have access to advanced device functionalities, like Barcode, Accelerometer, Bluetooth, Calendar, Camera, Contacts, File, GPS, Menu, and among others (Palmieri et al, 2012).

With PhoneGap, a developer has the choice of utilizing PhoneGap Build which goes about as a bridge between the various mobile platforms. Without PhoneGap Build, the developer would need to compile the application for every platform separately. PhoneGap Build is a web service where the developers transfer the source code to, in a compress form (zip file), and receives either by scanning QR-code with a smartphone or downloading package for the various operating systems (Bönström, 2014). Figure 4-2 illustrates PhoneGap Build.

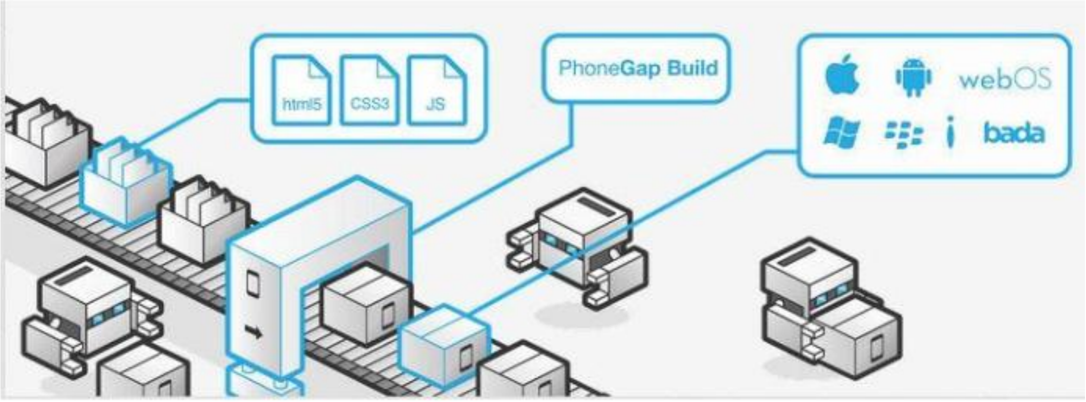


Figure 4-2: PhoneGap Build (Bönström, 2014).

When a developer is compiling the application, he has the choice to turn on a function called hydrating. This enables users to update their applications to most recent version whenever a new compilation has been made instead of downloading the entire application once more (Ciman et al, 2014, Bönström, 2014).

The quality of the user interface is the major concern of PhoneGap because of the feature of the platform’s rendering engine. Web views on other platforms have some limits, and are usually why the quality of PhoneGap user interface is significantly lower than applications with a native user interface (Pazirandeh and Vorobyeva, 2015). Also, PhoneGap cannot be extended with native user interface. Because of the iOS limitations, the iOS compass can't be accessed with PhoneGap, which makes improvement of some projects and creating of some applications difficult.

4.3Xamarin

Xamarin provides commercial software development tools which enable a developer to create applications for Android, iOS and Windows (Gridin, 2015). It allows developers familiar with C# and .NET to write applications for iOS and Android platforms (Pazirandeh and Vorobyeva, 2015). Using Xamarin, developers build their application by creating a single code base written in the C# language. Developers can have access to all C# language features like ‘Generics’ and ‘Async’, and at the same time have the capacity to access any mobile specific features and build a native user interface. The Xamarin library gives numerous common controls and layouts that are mapped straight to the native ones at run time, so that applications look fully native (Radi, 2016, Peppers et al, 2016).

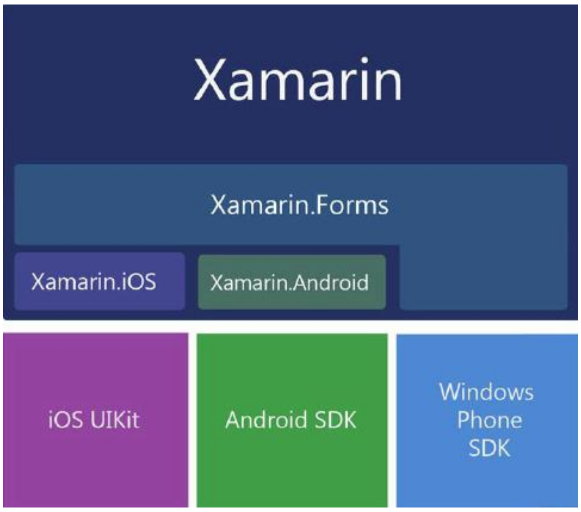


Figure 4-3: Xamarin forms mapped to native operating system libraries (Radi, 2016)

Xamarin depends on the free and open-source Mono .NET framework, which permits Microsoft .NET applications to run on multiple platforms. It provides two mobile development products, Xamarin.iOS and Xamarin.Android. It consists of class libraries, virtual machine, and a C# compiler. Applications are made either in Visual Studio or Xamarin's IDE called Xamarin Studio (Petzold, 2015). Applications created with Xamarin are developed in such a way that all the logic is written in C# and .NET, and this code is shared among other platforms (Pazirandeh and Vorobyeva, 2015). Xamarin IDE emanates as MonoDevelop for android applications and as Visual Studio plug-in component for windows and Mac operating system, and as Monotouch for Mac OSX only (Webinos, 2018). Xamarin takes after a native compiling cross platform framework. Figure 4-4 shows Xamarin Forms shared among different platforms.

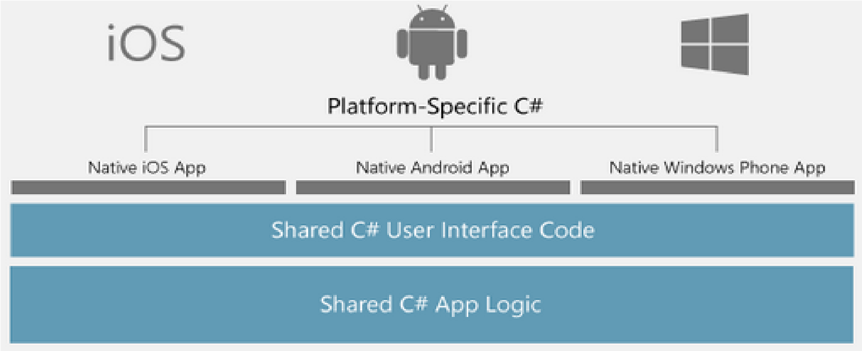


Figure 4-4: Xamarin Forms shared among different platforms

Xamarin uses the same language, APIs and data structures for more than 75% of application code over all mobile development platforms (Redda, 2012). Xamarin makes use of .NET Base Class Library (BCL) which is a group of Microsofts.NET framework libraries and it utilizes Language Integrated Query (LINQ), a feature of .NET framework that elongates the control of C# to handle data source querying (Redda, 2012).

Xamarin forms offer layout and template that comprises two components required for any application development. The "Application" component initialises the application while the "Page" component represents a single screen within the application. Xamarin forms give diverse layouts and templates for "pages" to accommodate the developer's needs. Each template possesses its own particular visualization and behaviour. Figure 4-5 shows examples of different Xamarin "page" templates.

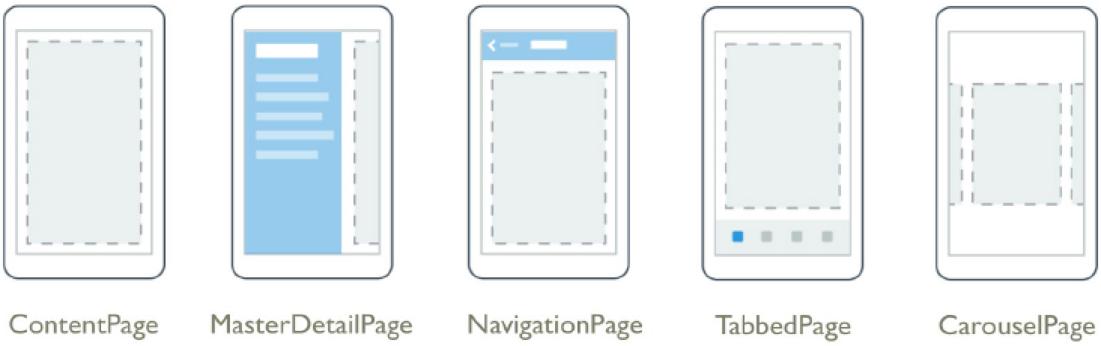
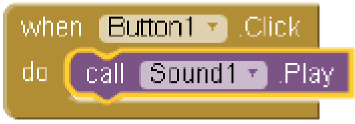


Figure 4-5: Different Xamarin Page Templates

Moreover, Xamarin provides View classes which give most of the standard visual for example, Entry, Frame, and Box View. Normally, these can be modified visually through their properties like background colour, and text colour. Additionally, it enables developers to control their behaviours, and utilize events, for example, textchanged event (Radi, 2016).

4.4App Inventor

App Inventor is an open-source, web-based framework that empowers creation of mobile applications for Android devices (Amasha and Al-Omary, 2017). Google and Massachusetts Institute of Technology (MIT) created App inventor in 2012. It is based on a kind of programming referred to as a blocks-based programming language (Meehan and Sabin, 2013). The commands are defined either as event-driven blocks, for instance, *when button.clicked do…,* or set blocks to set the values of a component or a variable. In Figure 4-6 below, when button1 clicked it plays a sound (sound1).



```
when Button1 .Click
do  call Sound1 .Play
```

Figure 4-6: Example of Block Editor in App Inventor

The user interface design in App inventor is straightforward; it is done by dragging a component onto a design screen. The style of designing the user interface is known as "what you see is what you get" (WYSIWYG) (Amasha and Al-Omary, 2017). App Inventor gives two primary components to simplify database activities, one is TinyDB, to store data directly on the Android gadget and the other one, is TinyWebDB, to store information in a web database. Other high-level components comprise GPS location, orientation sensors, text-to speech recognition, scanning barcodes and among other (Perdikuri, 2014). Figure 4-7 shows app inventor's architecture:

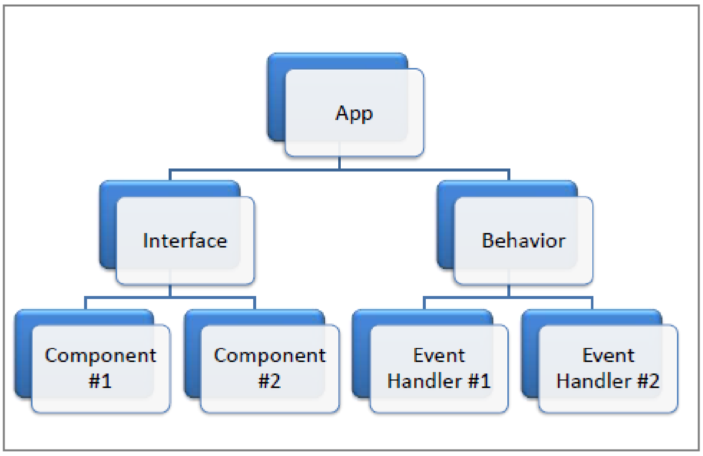


Figure 4-7: App Inventor Architecture (Trivedi, 2012)

In Figure 4-7, the user interface comprises Components; the components are same as tools in windows forms or Web forms. One approach to define an application's internals is to break it down into two sections, that is, components and behaviours (Trivedi, 2012). There are two primary kinds of components in an app inventor, these are visible and non-visible. The visible components are the ones you can see when the application is launched; examples are buttons, text boxes, and labels. Non-visible components are those that cannot be seen, rather they give access to the built-in functionality, for instance, the Texting component sends and processes SMS texts, the Location Sensor component controls the device's location, and the TextToSpeech component talks (Wolber et al, 2011).

Application's behaviour on the other hand, is ideally challenging and most times, complex. The App inventor behaviour characterizes how the application should react to events, both user initiated (example button click) and external (example an SMS text arriving to the phone). The difficulty of specifying such communicating behaviour is why writing computer programs is so challenging (Trivedi, 2012, Wolber et al, 2011).

Creating application occurs in three windows as illustrated in Figure 4-8 below. The primary window which is on a browser, allows the developer to choose and set up components on the application screen. The second window is a Java window where the programmer can connect blocks with the purpose of controlling the application features. The third window permits real-time testing and debugging of application, which can be done on emulator or a real device (Pokress and Veiga, 2013). App inventor is broadly used because it uses an integrated programming editor that comprises several tools to assist developers in designing and creating their applications (Hsu and Ching, 2013). App inventor enables programmers to test their applications while programming with an emulator or a device rather than waiting to compile (Gestwicki and Ahmad, 2011).

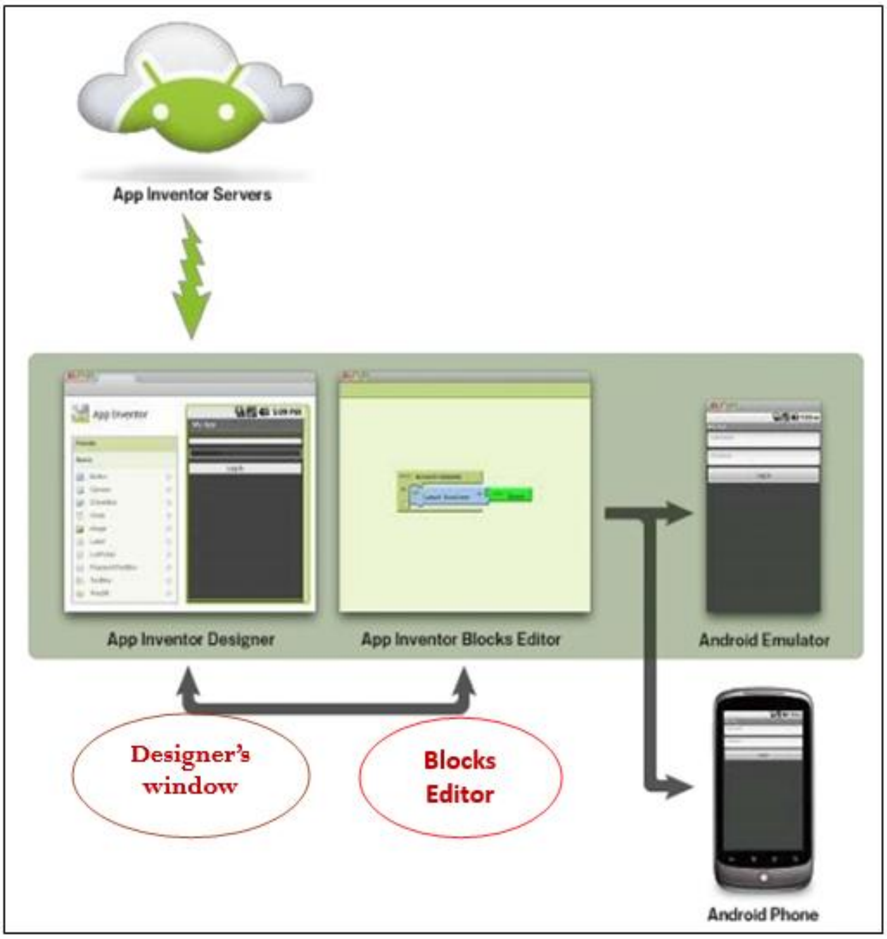


Figure 4-8: Three Windows of App Inventor

Applications created with App inventor can in any case be put on Play Store; also, a gallery for App inventor programs exists where individuals can download the source code and additionally the real application. This is a completely unique gallery and may emerge as the first marketplace where an application user could buy an application and after that change the source code to improve it to meet their needs (Kepley, 2014). App inventor can be used online via browsers like Chrome or Firefox. The MIT App Inventor project looks to democratize software development by engaging all individuals, particularly the youth, to move from technology consumption to technology creation.

4.5Sencha Touch

Sencha Touch is a framework for developing user interface for mobile applications. It enables developers to make mobile application using basic HTML, CSS, and JavaScript that supports different mobile devices, example android, iOS, Windows, and BlackBerry (Heitkötter et al, 2013). It depends on Model View Controller (MVC) architecture. It is possible to create different applications that is compatible with different sizes of devices, for instance, phone and tablet (Sánchez Blanco, 2016). In a recent update, the MVC framework support was integrated, which is important when creating enterprise applications, as the code will be more structured (Pawar, 2014). Sencha Touch has a flexible layout manager that assist with how data and content are organized and displayed over various mobile devices with different operating system (Stolzle, 2018). The framework incorporates a strong data package that can consume data from any backend data source. It also provides a responsive touch features, thus, the user can effortlessly navigate while using the mobile application (Tengroth, 2016).

Sencha Touch is optimized for creating application that works over various platforms. To make application development simple, Sencha Touch provides simple yet effective application architecture that leverages the MVC design (Tengroth, 2016). This approach keeps code clean, testable, and simple to maintain, and offers many benefits with regards to creating applications (Clark and Johnson, 2013). Application is a gathering of Models, Views, Controllers, Stores, and Profiles, in addition to extra metadata for application related elements, for example application icons and launch screen images (Sencha, 2018). Figure 4-9 shows Sencha Touch architecture.

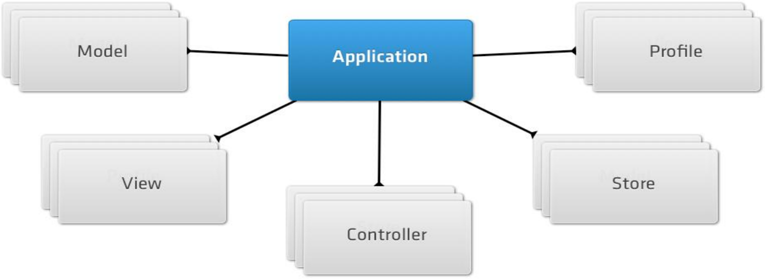


Figure 4-9: Sencha Touch Architecture

- **Models**: characterise a sort of data object in application - for instance an e-commerce application may have models for User, Product, and Order
- **Views**: are in charge of displaying information to application users and for leveraging the in-built components in Sencha Touch
- **Controllers**: handles interaction with your application, by listening for user interactions, for example, taps and swipes, and taking action accordingly.

- **Stores**: are in charge of loading data into the application and for controlling segments, for example, Lists and DataViews
- **Profiles**: It empowers to effectively modify your application's user interface for phones or tablets, while sharing as much code as possible

Sencha Touch Application Programming Interface (API) does not have the following competence (Steczko, 2016);

- The application does not have access to the device's camera, contacts, and accelerometer.
- It does not have push notification facility.
- With regards to license policy, it is free for open source applications but paid for commercial applications.
- It is not good for hardcode graphics and animation applications, example gaming applications.

4.6Dragonrad

DragonRAD is a mobile application development framework by Seregon Solutions and it is under a commercial license. DragonRad is a tool for creating mobile applications for operating systems like, Android, BlackBerry and Windows Mobile. DragonRad offers developers with drag-and-drop programming environment which then helps to save programming time and to develop logics and focusing database driven applications (Ribeiro and da Silva, 2012). DragonRad allows features provided by the operating system such us camera, maps, calendar, contacts and others to integrate and synchronize with a database system (KC, 2014). DragonRad come with their own IDE, which can be configured for platforms such as iOS, Android, BlackBerry, and Windows Mobile (Ribeiro and da Silva, 2012). DragonRad consist of 3 major components within its architecture: Designer, Host and Client which are described below. Figure 4-10 shows DragonRad architecture.

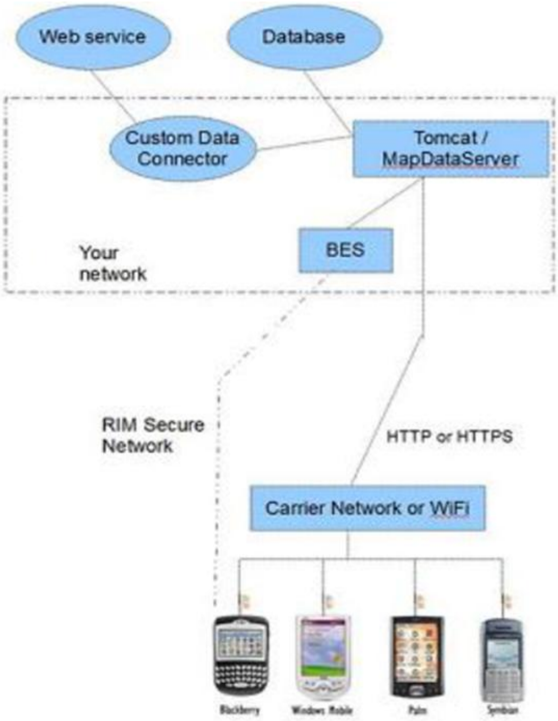


Figure 4-10: DragonRad Architecture (KC, 2014)

The Designer provide the developer in designing the application and creating them utilizing a simplified drag-and-drop environment. The DragonRAD Designer is a WYSIWYG intuitive instrument that can run on a Windows desktop environment. It enables the user to build the application in a simple way, lessening the effort of coding and making simpler the maintenance of the code. However, scripting can be utilized when more advancement is required (KC, 2014).

DragonRad Host functions as mediator between enterprise and application running on Linux or Windows. It provides support for querying communications when the network is unavailable and establishes synchronous connection with database accessibility. Some functions include taking information query from the device, running data query on particular targets sending information back to the device dependent on demand, controlling data updates from the devices and updating databases, as well as compressing data and checking data packets (KC, 2014). The host can be run on Linux or Windows Server, and sets up connection with a mobile device for information correspondence (KC, 2014).

The DragonRad Client component offers customized functionalities such as application name, DragonRad host installation link, change icon and DragonRad project and offers the feel of native applications to run on devices. This component performs like a native application on device which runs and interprets code of the built application by designer. Additionally, it has an emulator which can run and troubleshoot or debug applications, with an online compilation tool. The region containing the network gives all backend upkeep like a custom data connector that could fit for any web service available, whereas Tomcat/MapDataServer is to assist the database. This full network is connected to mobile phone with various operating systems with the assistance of the Wi-Fi. The other sub-part BlackBerry Enterprise Server (BES) is particularly for BlackBerry products like PlayBook (Ribeiro and da Silva, 2012). This tool permits the making of application in exceptionally simple way with only three steps, which are connect, build and deploy.

4.7Criteria for comparison

This section gives explanation for the choice of frameworks and outlines criteria used for comparing the development frameworks. The choice of frameworks was done by taking into account frameworks that can create applications on at least some of the main mobile operating systems, for example, android, iOS, windows phone. These days, there are numerous frameworks and tools available, yet few can create applications for the defined mobile operating system. To maintain a logical stand on the comparison, this research has adapted criteria from Majchrzak et al (2015), Sommer and Krusche (2013), Pazirandeh and Vorobyeva (2015), and Palmieri et al (2012), that might be accommodating to comprehend which framework may be fitting for development of mobile application.

The adapted criteria that have been taken into consideration in this research include the following: Each criterion has a name and is denoted by a letter for the perspective and a running digit. Each criterion has a brief description as indicated in table 4-1.

CRITERION NAME	DESCRIPTION
C1	**Mobile operating systems platform support** It is to evaluate the types of mobile operating system that these frameworks support. When it supports more platforms and later version of platforms, it may decrease a developer's effort and cost. Moreover, it can expand productivity and help to understand the potential impacts on particular business model (Nitze and Schmietendorf, 2013, Majchrzak et al, 2015).
C2	**Framework licences and cost** To assess the terms and conditions of use. This standard examines whether a framework is either a free tool or an open-source, and the license under which it is distributed.

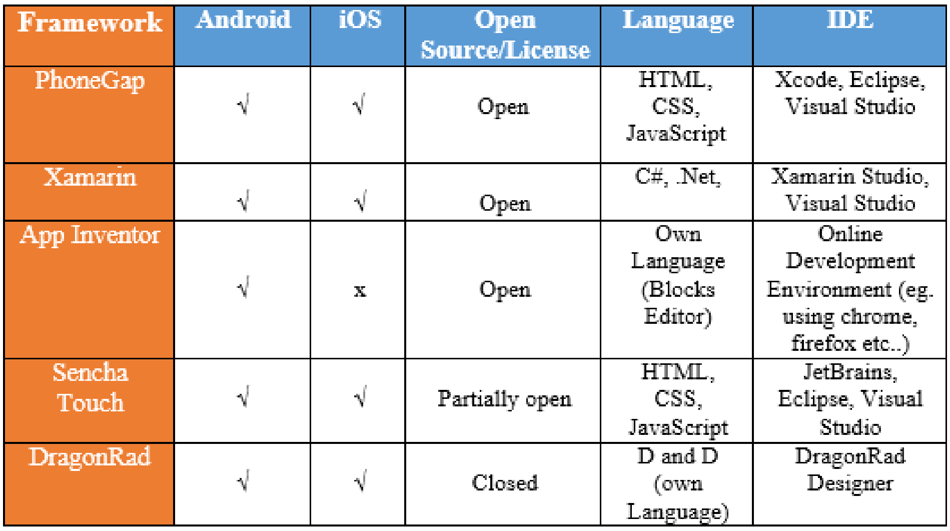
C3	**Development Environment** Assesses development and highlights the development environment related to a particular framework, especially the tool support (IDE, debugger, and emulator) and functionality such as automated testing. This criterion involves the method of creating the graphical user interface (GUI). Likelihood to create and test the user interface without having to frequently deploy it to a device or an emulator is viewed as advantageous (Majchrzak and Grønli, 2017).
C4	**Learning curve** Time and effort required to know and understand a framework impact and its suitability. Instinctive models, perhaps looking somewhat like the already known paradigms, can be learnt quickly. Developers undergoing training to be equipped with a particular language include possibly delays in development process and (Palmieri et al, 2012).
C5	**Packaging and Distribution** After the development of application is finished, the next step is to distribute it. This criterion evaluates how simple it is to distribute applications developed with a particular framework to users or organization. One option is to use the app stores of mobile platform-specific, since clients frequently wants to utilize this distribution channel. On the other hand, be dependent on app stores only has disadvantages, a framework offering different distribution channels also has merits(Palmieri et al, 2012).
C6	**Look and Feel** Despite the fact the appearance of an application can be changed amid development, it does matter whether a framework essentially supports a native look and feel or whether its user interface looks and behaves like a Web site. Most users look for applications that bear a resemblance to native applications. As this is frequently cited requirement of applications, this criterion measures whether these frameworks offer support for a native look and feel (Sommer and Krusche 2013).
C7	**Runtime Performance** The performance at runtime advises the general impression of an application. It is to assess if it functions appropriately at whatever point and wherever a client needs. It tries to compare the application's responsiveness on user-interaction at start-up and runtime. To attain satisfactory execution it is important to handle blunder in application immediately and enable a user to utilize a terminal effectively (Xanthopoulos and Xinogalos, 2013).

Table 4-1: Criteria description

4.8Evaluation

This section describes and evaluates the mobile development frameworks; we assess the five mobile frameworks in accordance with the criteria stipulated above. Results are designated separately for each framework in subheadings.

Publicly accessible information like framework documentation, community resources, and reviews are useful in gaining knowledge about the quality of a framework. For data to evaluate, each framework's project page and the official homepages were referred to. Each of the 7 assessment criteria are scored for five different frameworks on a 5-point scale from '1' for '*very poor*', '2' for '*poor*', '3' for '*fair*', '4' for '*good*' to '5' for '*very good*'. This section also introduces the comparison made among the frameworks in view of some of comparison criteria explained in the previous section. Table 4-2 demonstrates the various features of the five mobile development frameworks.



Framework	Android	iOS	Open Source/License	Language	IDE
PhoneGap	√	√	Open	HTML, CSS, JavaScript	Xcode, Eclipse, Visual Studio
Xamarin	√	√	Open	C#, .Net,	Xamarin Studio, Visual Studio
App Inventor	√	x	Open	Own Language (Blocks Editor)	Online Development Environment (eg. using chrome, firefox etc..)
Sencha Touch	√	√	Partially open	HTML, CSS, JavaScript	JetBrains, Eclipse, Visual Studio
DragonRad	√	√	Closed	D and D (own Language)	DragonRad Designer

Table 4-2 Features of mobile development frameworks

4.8.1 Evaluation of phonegap

- **Mobile operating system platform support.**

PhoneGap offers support for all the major mobile operating systems. It supports Android, iOS, Windows Mobile, Blackberry, Samsung Bada and web OS (Smutný, 2012, KC, 2014).

- **Framework license and cost**

PhoneGap was given to the Apache Software Foundation (ASF) under the name Apache Cordova and thus is an open source distribution of Apache Cordova. The PhoneGap development framework offers open source capability to the developers to create application only for once. Thereafter, in the event that you need to develop application more than once, PhoneGap will charge you on monthly basis (Kim et al, 2015).

- **Development environment**

PhoneGap has diverse methods among frameworks, examples includes the use of native IDE’s such us XCode for iOS, Eclipse for Android, and Visual Studio for Microsoft. There are different paid and open source JavaScript as well as HTML5 structures which will encourage the development. PhoneGap application developers can't create an iOS application without downloading iOS SDKs, and this is not likely without a Mac PC or operating system (Ciman et al, 2014, Ribeiro and da Silva, 2012).

- **Learning curve**

It is critical to have a beginner competency range of skillset in no less than one mobile native operating system programming. Also, developers ought to have a significant range of abilities in HTML and JavaScript. There is no reason to learn other complex programming languages to develop applications, all a developer should know is JavaScript, HTML and CSS. These languages are common for web developers (Ciman et al, 2014, Kim et al, 2015).

- **Packaging and distribution**

There are two choices accessible for building and packaging applications. The developer can begin with building and packaging locally using the PhoneGap Command-line Interface (CLI) and then utilizing PhoneGap Build cloud service to simplify the build and application packaging process (Smutný, 2012). The PhoneGap Build service enables uploading of web-code to the Adobe cloud service to be compiled into intended mobile operating system. Apple uses Distribution Certificates to identify a developer or development team. To distribute application to Apple App store, you will need a distribution certificate (Smutný, 2012, Kim et al, 2015).

- **Look and feel**

Look and feel of the applications will be non-native as the graphical user interface is created from HTML, CSS and JavaScript. The most ideal standard in terms of designing and programming a rich user interface is not mentioned. Hence, programming the application of great interface has turned out to be tedious for developers (Bönström, 2014).

- **Runtime performance**

Performance of the hybrid application created utilizing PhoneGap will be lower than the applications created natively. Performance in some cases causing an issue if your application requires lot of animation or with plenty of graphic elements (Kim et al, 2015, Bönström, 2014).

4.8.2 Evaluation of Xamarin

- **Mobile operating system platform support.**

Xamarin supports three major mobile operating systems namely, android, iOS and windows. Moreover, it also supports application to be ported on Mac.

- **Framework license and cost**

Xamarin has limited access to open source libraries. Xamarin goes under paid licence and a free trial option, so if the application created with Xamarin is purposed for commercial use, there is a probability to buy a commercial license and avoid any complications (Radi, 2016).

- **Development environment**

Applications are created either in Visual Studio or Xamarin's IDE called Xamarin Studio. Xamarin forms offer a plugin that can be installed in visual studio for mobile development. Even though the use of visual studio plugin seems more expensive since it involves a commercial licence, it is more beneficial to use it than the Xamarin studio because you can take advantage of many of its plugin that would simplify and accelerate development (Radi, 2016).

- **Learning curve**

Xamarin uses C# as a programming language. Therefore, to develop application with Xamarin, developers will have significant learning curve, because they need to learn the C# programming language to get some grips of basic knowledge with the framework.

- **Packaging and distribution**

Xamarin studio comes with a respective native builds which can be transferred onto the platform stores. The Xamarin Component Store is a catalogue of free and paid components that include stunning user interface controls. The Component Store is integrated into Xamarin Studio and Visual Studio, so developers can find and manage components from their IDE of choice.

- **Look and feel**

Xamarin pulls native look and feel which is the primary differentiator in the comparison of cross-platform competitors. Xamarin is natively compiled, which makes it a better option for developing high performance applications with native look and feel.

- **Runtime performance**

Applications created using Xamarin will show the same performance as that of native mobile applications.

4.8.3 Evaluation of App Inventor

- **Mobile operating platform support**

Currently, App Inventor lets you develop applications for only Android devices with Google announcing in December 2017 that they will be implementing App Inventor for iOS support since most developers have requested for it over the years.

- **Framework license and cost**

App Inventor is an open source, web-based system that allows developing a mobile application for Android devices. Android application can be developed, installed or distributed at no cost.

- **Development environment**

App Inventor is a web-based visual programming development environment with simplified drag-and-drop interface which objective is to assist design and implementation of mobile applications. App Inventor uses internet browser and either a connected phone or emulator. The two main elements for development are the Component Designer and the Blocks Editor.

- **Learning curve**

App Inventor uses blocks programming to enable developers to focus on design and programming logic instead of language syntax. The blocks language reduces the need to remember and type code. The decrease of typing rate broadly lessens chances of syntax errors for beginners, the blocks provides visual hints which makes development task easier. The App inventor intuitive online development environment coupled with the blocks programming makes it easy to learn the framework.

- **Packaging and distribution**

With App inventor developers can share application in an executable form (.apk) which can be installed on Android device, or in source code form (.aia) that can be loaded into App Inventor and recreated into apk executable file. Developers can also distribute their applications on the Google Play Store. Developers are required to simply go to Google Play Publishing Home and follow the steps to publish application to Google Play.

- **Look and feel**

The component Designer is where developers lay out the look and feel of an application, and specify the functionalities it must have. You choose layouts for the user interface, for example, Buttons, Images, and Text boxes, thus developers are able to create native application for android devices that leverage native look and feel.

- **Runtime performance**

The runtime performance of the native application created using App inventor will be higher than the mobile hybrid or web application. App inventor has some issue with performance when a particular application has more than 10 screens or pages.

4.8.4 Evaluation of Sencha Touch

- **Mobile operating system platform support**

Sencha Touch is a framework that enables developers to create applications that support different mobile devices, examples are android, iOS, Windows, and BlackBerry (Litayem et al, 2015).

- **Framework license and cost**

Sencha Touch is a partially open source framework for mobile applications. Sencha Touch is licensed either under GPLv3 or a free-of-charge commercial license that allows closed source distribution. More enhanced support and specialised tools can also be obtained at additional cost (Tengroth, 2016).

- **Development environment**

Sencha Touch developers use IDE like JetBrain IDEs, Visual Studio, or some light editors such as Sublime Text or Notepadd++. But there are some limitations, and as such there are paid commercial plugin to integrate into some types of IDEs to give more features and functionalities. Sencha Touch involves a lot of effort in creating small applications because of its structural overhead with concepts like MVC (Litayem et al, 2015, Tengroth, 2016).

- **Learning curve**

When learning Sencha Touch, one needs to deal with JavaScript and the framework's API. Developers have to acquaint themselves with the component structure and the manner by which components are instantiated and interrelated to each other. It depends on a heavyweight framework, hence requires complex learning which to a certain extent, can be inconvenient (Litayem et al, 2015).

- **Packaging and distribution**

Sencha Cmd is capable to change a Sencha Touch application into mobile application that has access to device functionality and can be distributed on various App Stores. Native packaging is a feature in Sencha Touch 2 and with a single command one can package application for iOS and for Android. Sencha Touch native packaging also supports Apache Cordova APIs and Packaging (Litayem et al, 2015).

- **Look and feel**

Sencha Touch is known for its capacity to offer a native application experience through its native-like themes and gadgets. Sencha Touch offers developers with over 50 high performance user interface widgets created particularly for mobile platforms, as well as themes for platform like iOS, Android and Windows to give created applications the most native look possible.

- **Runtime performance**

Sencha Touch is a high performance JavaScript framework for creating HTML5 mobile applications which can be compiled into native applications utilizing PhoneGapor Sencha's command line tool. Therefore, runtime performance of a Sencha Touch application is somehow close to native performance.

4.8.5 Evaluation of DragonRad

- **Mobile operating system platform support**

The DragonRAD SDKs enable developers to build, design, manage and deploy mobile applications in a single development environment and use it through iOS, Android, Windows Mobile and BlackBerry platforms (Kim et al, 2015).

- **Framework licence and cost**

A primary weakness of DragonRad is that it has license price and is under non-open source redistribution.

- **Development environment**

DragonRad has its own IDE, which can be configured for various platforms such us iOS, Android, BlackBerry and Windows. It has drag-and-drop visual environment or graphical user interface for developers to create and install mobile applications. Features of drag-and-drop are not only assisting developers to design applications, in addition it reduces the efforts for maintenance and coding (KC, 2014, Ribeiro and da Silva, 2012).

- **Learning curve**

DragonRad has drag-and-drop features, which requires lessened programming ability to create applications. It enables developers to design, manage and deploy mobile applications with ease. Therefore, learning curve will be less as compared to frameworks that requires complex languages (Kim et al, 2015).

- **Packaging and distribution**

DragonRad also has an emulator which can run and debug applications, with an online compilation tool. Developed applications can be installed and distributed via DragonRad designer which comes under commercial license fee.

- **Look and feel**

DragonRad host offers the feel of native applications to run on devices. The component works like a native application on device which supports run and interpret code of the developed application by designer.

- **Runtime performance**

Because DragonRad leverages the native look and feel of application therefore its runtime performance will be higher as compared to mobile web applications.

4.8.6 Results of assessments of frameworks

In the sections above comparison and evaluation between PhoneGap, Xamarin, App Inventor, Sencha Touch and DragonRad were made using different criteria with detailed explanation. For each criterion, the frameworks were assessed on the score of 1 to 5. A score of 1 implies the framework performs very poor on that particular criterion whereas 5 implies a very good score. It should be taken into consideration that scores depend primarily on the information taken from the official documentation of the five frameworks and several other online resources. Some sources have also been taken from couple of researches from the reviews and other research publications. Therefore, the total objectivity of the scores cannot be claimed, nonetheless the focal reason here is to give a review about a possible evaluation matrix for the mobile development frameworks. Based on the resources sought and own individual judgement, the frameworks scores are tabled as indicated in Table 4-3 below.

Criterion Symbol	PhoneGap	Xamarin	App Inventor	Sencha Touch	DragonRad
C1	5	5	1	5	5
C2	4	3	5	2	1
C3	3	3	5	4	4
C4	5	3	4	3	4
C5	4	4	5	4	4
C6	3	4	4	3	4
C7	3	4	4	4	4

Table 4-3: Summary of Assessment

4.9Conclusion

This section presented a comparison of different frameworks for mobile development. The evaluation considered PhoneGap, Xamarin, App Inventor, Sencha Touch and DragonRad. The assessment results as presented in Table 4.3 compare the evaluation gained by each framework against all the chosen criteria. The assessment summary in table 4.3 has been discussed in chapter 6 of this research. The features and characteristics elaborated within this chapter might be essential in selecting one framework rather than other. From these features and characteristics, there is a possibility to recommend the more suited target group or market of each studied framework and to emphasize the principle qualities and shortcomings of each one. The next chapter uses of one of the evaluated frameworks as proof of concept to develop mobile application for water usage management.

CHAPTER FIVE

SYSTEM DESIGN, IMPLEMENTATION AND TESTING

5.1Introduction

This chapter presents system design, implementation and the testing of the components to determine if they perform as required. It starts by presenting a case study scenario that guides us in the development process through the framework identified in chapter 4. This acts as a prelude to all sections that follow which includes system design, architecture, implementation and testing. Requirement elicitation is also presented as a way of proving the background on which the system is built.

5.2Case study

In the previous chapter we presented App Inventor as the ideal mobile development framework to develop mobile applications for android devices. This chapter uses this framework to develop a mobile application to demonstrate some of the selling features of this framework. The design of the application is based on the current challenges faced by many municipalities in managing water. We have chosen Alice and Fort Beaufort Towns under Raymond Mhlaba Municipality as the research sites. This is due to the proximity of the two sites to University of Fort Hare where the research is undertaken. It is anticipated that the application will provide residents with the ability to communicate with water authorities concerning all water issues. Hence the ideal application should be:

- Available on Android.
- Access to the network.
- Notification Alert
- Low costs development.
- Access to Smartphone GPS.

Furthermore, App Inventor is an impressive framework, and programming method provided by the framework clearly differs from conventional languages and attracts a beginner. The development environment of App Inventor is an advanced cloud type. The fact that the layout of components and the description of processing can be done with separate windows reduces complexity and helps us to design applications. App inventor also has a time constraint that means the development process of applications is much shorter as compared to other frameworks. The application being developed is named *SaveAmanzi* because of its anticipated usage for water management. App inventor has support for the requirements outlined above for *SaveAmanzi* application. Since it does not require advance programming experience to use it, App Inventor has produced a huge number of new makers in the mobile spectrum. More than a million individuals have attempted it, together developing 4.7 million applications, with more than 30,000 published on Google Play store (Wolber et al, 2015). Biggers et al (2008) emphasized that this kind of power is engaging an extremely general gathering of youngsters and has implications for broadening participation in computer science. Moreover, App inventor can have a much more extensive societal effect and give individuals an alternate understanding and point of view of the technology they use each day (Abelson, 2011).

5.3Requirement gathering

Requirement gathering is the procedure for finding out or generating requirements and prerequisites for an intended system by communicating with end users, system users and others who have a stake in the system development (Kavitha and Thomas, 2011). The main aim of requirement gathering is to develop and maintain refined and descriptive system requirements specification document (Gupta and Gupta, 2018). To better understand the problem at hand, it is always essential in software development process to understand the behaviours and views of intended users of a particular system. Appendix 5 presented requirement analysis and discussion, and it revealed some issues associated with water in the two research sites. The identified challenges have been highlighted below.

The survey (*interviews, observation, and questionnaire*) results from the two research areas, showed that there are a lot of water leakages in Alice and Fort Beaufort towns especially in Alice town. It can be deduced that water is being lost on daily basis in these two research areas. There are no appropriate and effective measures on the part of water authorities and consumers to address this issue which could further worsen the insufficient water supply in these areas.

Interviews conducted with water authorities in these two towns (Alice and Fort Beaufort Water Works) revealed that people in these two communities hardly report water leakages to authorities. The reason being that many people do not know about the existence of water authorities and those who are aware of them, do not have means of reaching them.

It was learnt from the survey results that, there is lack of communication between consumers of water and delivers of water services. Consumers do not have a clue of their water consumption rate, for instance how much they are been charged per kilolitre (*1000 litre*) of water consumed. A question raised in this research is, how a novel mobile application can be designed to address water management. Mobile phones can be used to address water that is lost on daily basis in Alice and fort Beaufort and can be used to create awareness on water conservation. Based on this scenario, a mobile application was developed as a tool to assist in the management of water so as to reduce water loss in Alice and fort Beaufort.

5.4 System design

System design is the way towards structuring of elements of a system such as the architecture, modules and components, the distinctive interfaces of those components and the data that goes through that system dependent on the predetermined requirements (Buede and Miller, 2016). The design process involves six stages. Figure 5.1 shows the software development process and model that was used to provide framework for technical and non-technical activities in order to deliver a quality system that meets user requirements. The six phases have been briefly explained below.

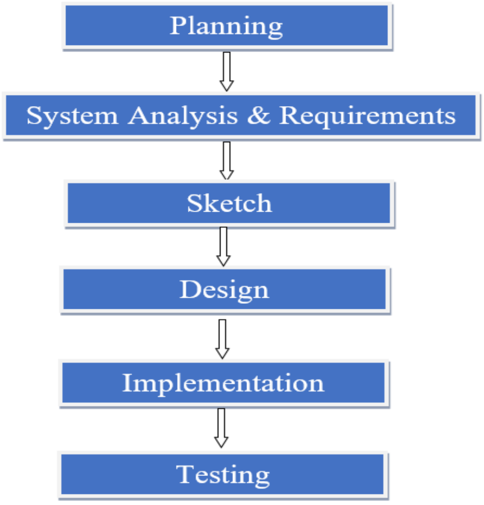


Figure 5-1: System Design process

Planning: The purpose of planning was to find out the scope of the problem and determine solutions. We identify whether or not there is the need for a new system to achieve the stipulated objective.

System analysis and requirements: In this second phase, end user requirements are analysed and project goals converted into the defined system functions that we intend to develop. Requirement gathering is the most crucial part at this stage of the software design process. Requirements are set of functionalities that the system needs to meet in order to be successful. A survey was conducted in both Alice and Fort Beaufort to determine system functionalities.

Sketch: Here, we combine our ideas with information obtained from the requirement gathering to suggest several possible design solutions. We sketched several possibilities on paper. This stage involves user interface definition of required feature. We generated ideas and work on basic sketches.

Design: In this phase, we prepare the system and software design from the requirement specifications which were studied in the second phase. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

Implementation: Here, we implement back-end functionality and front interface of the system. On receiving system design documents, the work is then divided in units and the actual coding started. We first build prototypes and the final product is developed from it.

Testing: After the code is developed, we test against the requirements to make sure that the application is actually solving the needs addressed and gathered during the requirements phase. During this phase, we used two levels of testing, unit testing and usability testing.

5.5System architecture

System architecture is the fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviours (Faisandier, 2013). The *SaveAmanzi* mobile application will need to communicate to the GPS application within the mobile device, see figure 5.1. The application will need both internet and GPS connection to fetch and display address of a particular place. In a situation whereby the user of the application does not know the address of their current location, by using the GPS connection, users can automatically get the address of their current location.

Water leakage report information needs to be stored somewhere for response by the water authorities and for that, a database will be used. The mobile application and the web-portal will communicate with database. The mobile application will use the database to add and get data while web-portal will also add and modify data in the database. All of the database communication is made possible through internet connection. Figure 5-2 below shows *SaveAmanzi* mobile application Architecture.

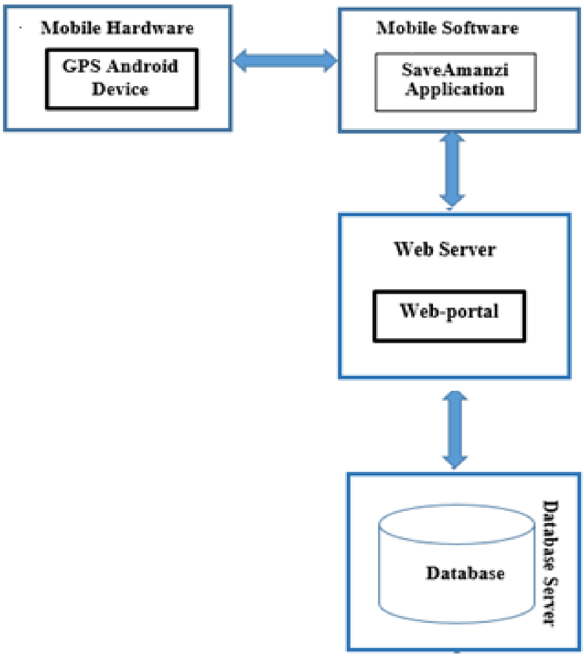


Figure 5-2: SaveAmanzi Application Architecture

5.6User case scenarios

Use case diagrams are referred to as behaviour diagrams used to portray a set of activities or actions that some system or systems subject should or can perform in collaboration with one or more external users of the system (actors) (Klimek and Szwed, 2010). The motive of a user case diagram is to demonstrate the distinctive ways that a user may interact with a system. An accurate, comprehensive insight into how a system is used by its users in practice is essential for designing systems that meet user expectations (Lennox-Chhugani, 2018). The mobile application consists of two parts: one part of the application is used by community dwellers and the other part for water authorities. There are 3 types of users that interact with the application: community members, water authorities and the system administrator. These three users have different use of the system so each of them has their own requirements.

A community member user can use the application to: report water leakages, receive message from water authorities, get water tariff information, get contact details of local qualified plumbers and get information on tips of saving water. The community members can also use the application to get information on safety tools and emergency guidelines to minimize water loss. The community members can also receive random notifications on water conservation.

The authorities responsible for water service delivery in these communities will be able to login as a technician and receive reports on water leakages from community dwellers. They will manage the information about the report, for example upon receiving a leakage report the will go to the site and fix the leakage. They can also send message to all the community members that have the application installed on their phones.

The administrator is a user who interacts with the web portal. They are responsible for managing the overall system so there is no incorrect information within it. The administrator can manage the information for community member and water authority. The administrator is responsible for administering user accounts, reset water authority's user passwords, activate/deactivate user accounts, and monitor database and server security. The administrator is also responsible for sending random notification on water conservation. Figure 5-3 below illustrates the different tasks that can be performed by users on the *SaveAmanzi* Mobile Application.

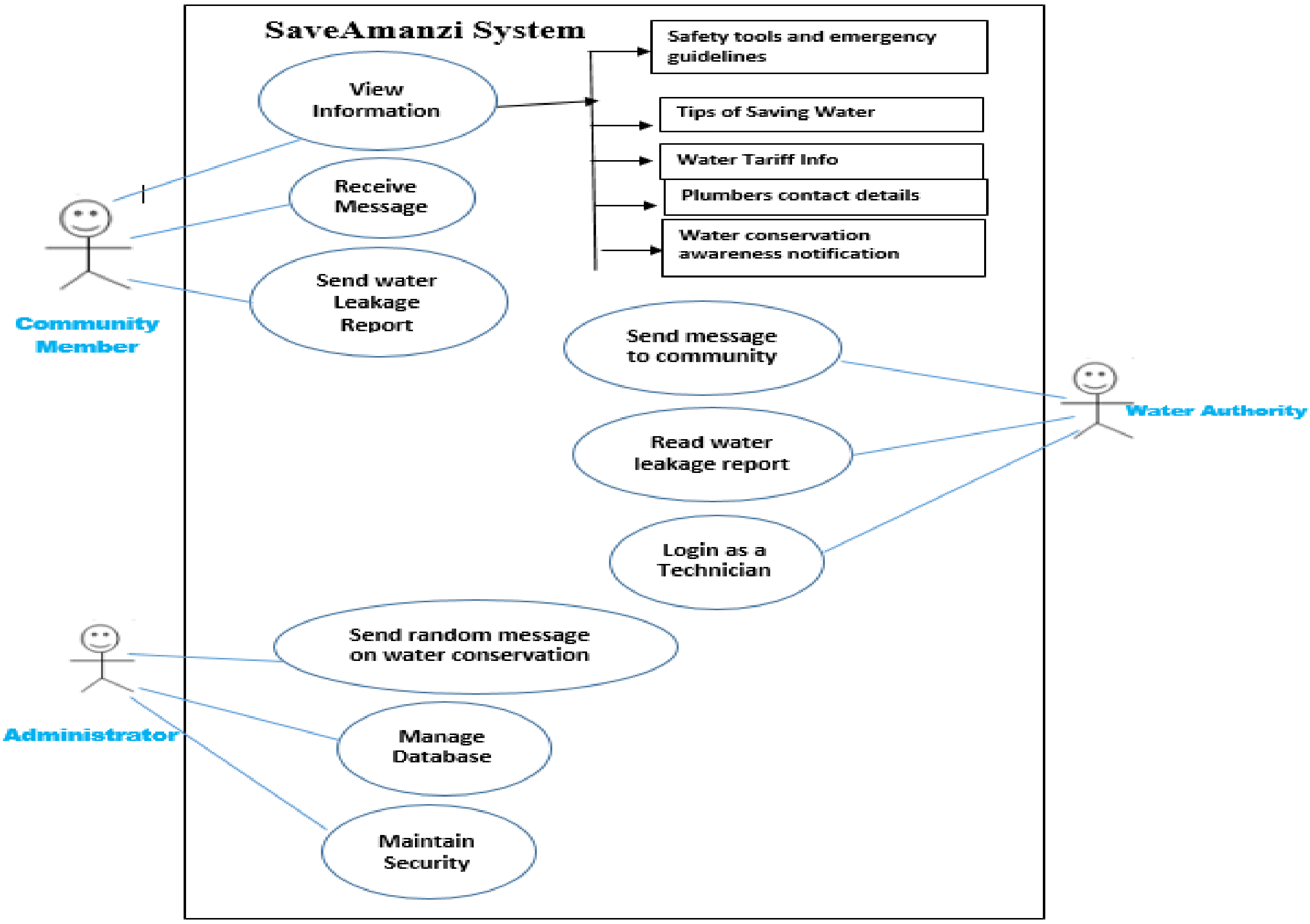


Figure 5-3: User case diagram

5.7Functional requirements

Functional requirements deal with what the system should do or provide for its intended users (Van Lamsweerde, 2009). The following functional requirements were identified for the *SaveAmanzi* application:

1. **Water leakage Reporting**
   - ✓ User should be able to report water leakages
   - ✓ Water authorities should be able to read water leakage reports
2. **Medium of communication between water authorities and community**
   - ✓ Users must be able to receive and read messages from water authorities
   - ✓ Water authorities must be able to send messages to users
3. **Safety tools and emergency guidelines to minimize water loss at home**
   - ✓ Users must be able to read safety tools and emergency guidelines information
4. **Tips of saving water**

- ✓ Users must be able to read tips of saving water information provided in the application

5. **Awareness notification on Water conservation**
   - ✓ Users should be able to receive notification on water conservation
6. **Water Tariff information**
   - ✓ Users should be able to read information on water tariffs
7. **Contact details of Local qualified plumbers**
   - ✓ Users should able to get contact details of local plumbers provided on the application.

5.8Class diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML) (Boustedt, 2010). This section presents the two important classes of the system. The technician class and user class were identified as the two most important classes of the system. Technician class has two properties and two operational functions namely, SendMessage() and ViewLeakage(). The user class has six operational functions as illustrated in Figure 5-4. The two classes have one-to-one relationship, this is because a technician depends on user for view leakage, on the other hand, a user depend on technician to receive message about water delivery.

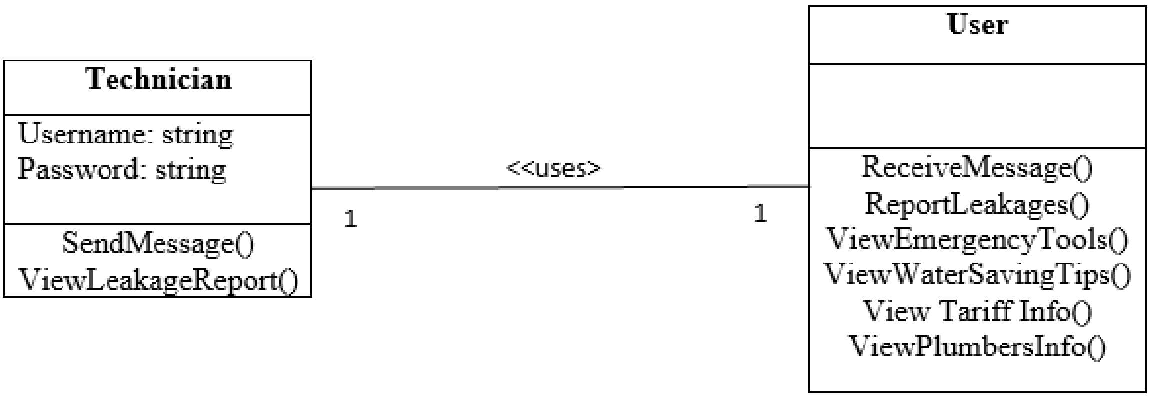


Figure 5-4: Class Diagram

5.9Technical requirements

The Technical requirements describe the specific technical capabilities and features necessary for successful implementation of the user requirements. The *SaveAmanzi* application was developed using App Inventor Framework. App inventor uses graphical interfaces which allow developers to drag and drop visual objects to create an application that can run on Android devices (Soares, 2014). App Inventor is a free and open server software system that runs within the Massachusetts Institute of Technology (MIT) App Inventor's cloud. To develop the *SaveAmanzi* application, we require a Google account to access the App Inventor's development service. To create an application's user interface and choose the structure of its building components, we ought to utilize a browser, in which they run the Component Architect tool. To execute the *SaveAmanzi* application's behaviour and logic of the component methods, we used the Blocks Editor, a Java web application that is downloaded on demand from the Component Designer on the developer's client computer.

The App Inventor language has a rich library of component classes. Component classes represent basic user interface facilities like textbox, button, canvas, arrangement, etc., sprites (touch-sensitive objects), and a wealth of built-in, smartphone specific functional features such as texting, phone calling, contact search, sensors, camera, sound player, video player, and so on. Applications developed with App Inventor is compiled and packaged into an application package file (APK) to be distributed and installed on Android mobile devices. The packaging command, available in the Component Designer, generates the APK file and downloads it to the user's computer, a USB-connected phone, or a phone connected over Wi-Fi. The downloading over Wi-Fi option requires the MIT App Inventor Companion application, which is downloadable from play store establishes a connection between the Android mobile device and client computer over the MIT RendezVous Server. Other technologies used along with App Inventor are as follows: Firebase cloud messaging, oneSignal messaging account, Google App Engine, Python, and TinyWebDB. These technologies have been defined below.

The *SaveAmanzi* application will be on an Android mobile device. The application's functionality will be dependent on standard Android operating system version 2.3 and above operating system features that runs properly. Android is a software cluster for mobile devices that includes an operating system, key applications and middleware. The Android Software

Development Kit (SDK) provides the tools and Application Programming Interfaces (APIs) required to begin developing applications on the Android platform using the Java programming language (Holla and Katti, 2012). About the design, Kernel of Android is based on Linux kernel and further furnished by Google (Zilpe, 2012). Figure 5-5 shows the major components of the Android operating system. The components are described in more detail below.

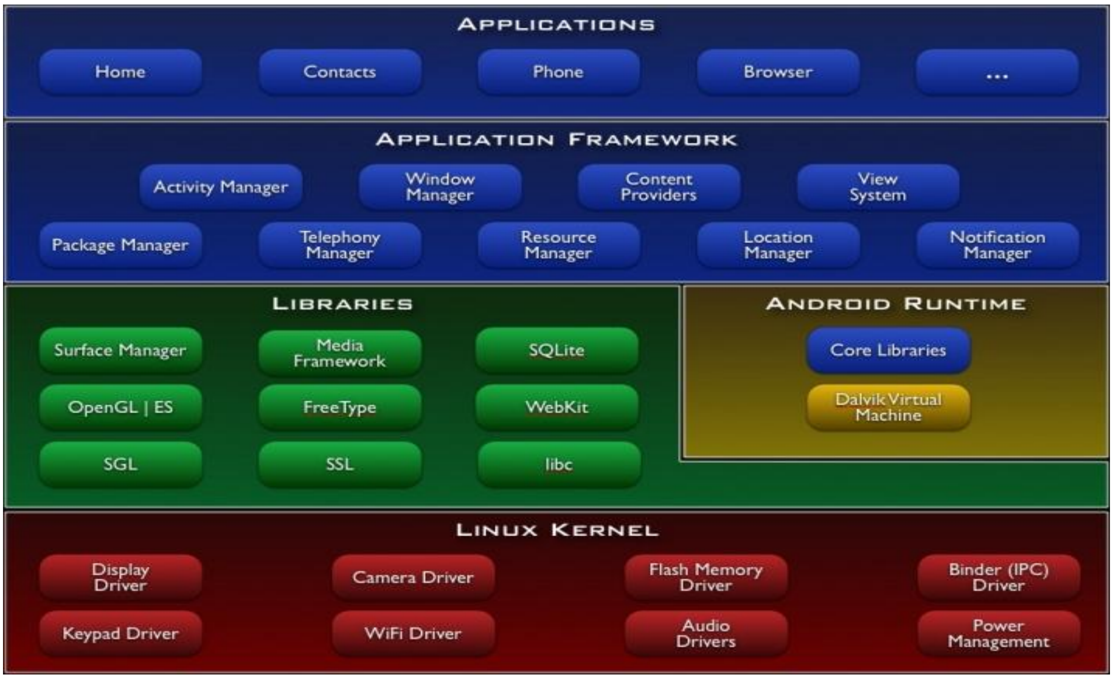


Figure 5-5: Android Architecture

Applications: Android includes a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications werewritten using the Java programming language (Dalmasso et al., 2013).

Application Framework: By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more (Tarkoma, 2009). Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (Butler, 2011). This same mechanism allows components to be replaced by the user.

Application Component:

Libraries: Android includes a set of C/C++ libraries used by various components of the Android system.

Android Runtime: Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple Virtual Machines (VMs) efficiently (Hu, 2008). The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool (Hu, 2008). The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Linux Kernel: Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack (Coskun et al, 2013).

5.9.1 Definitions of other technologies used with App inventor

Firebase Cloud Messaging (FCM) provides connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost (Moroney, 2017).

TinyWebDB is an App Inventor component that allows you to store data persistently in a database on the web (Wolber et al, 2011).

Google App Engine is a web framework and cloud computing platform for developing and hosting web applications in Google-managed data centres (Maharana et al, 2015).

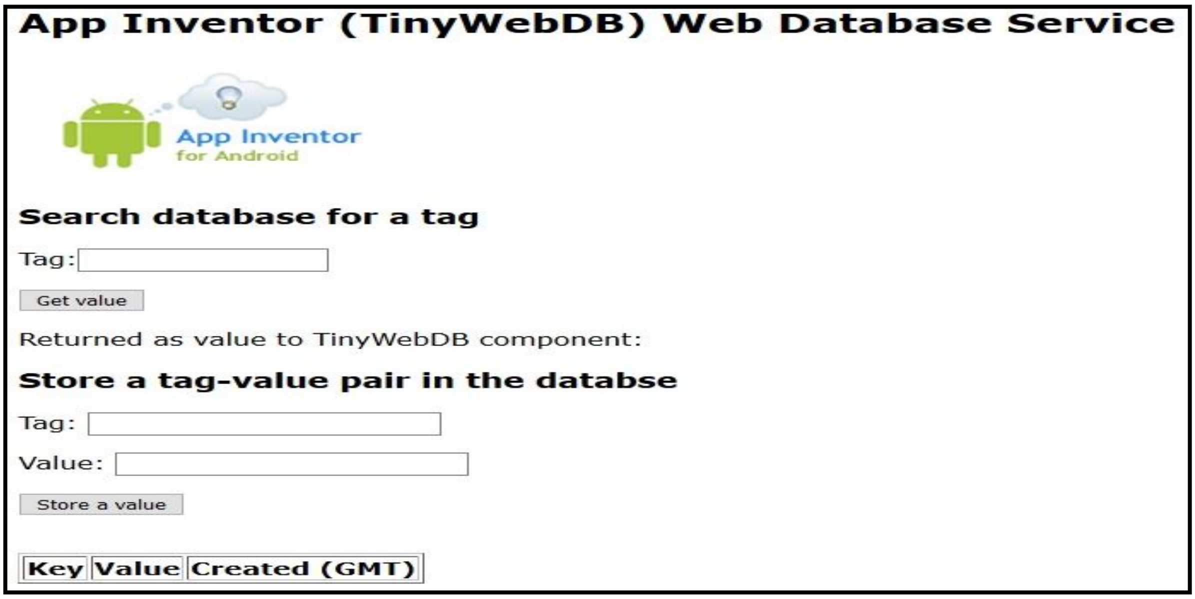
OneSignal is a simple and smart service that sends push notifications across many platforms (OutSystems, 2018). OneSignal features reliable delivery of millions of notifications, segmentation and targeting, automated delivery, localization and support for all major app development tools (Khemry, 2016).

Python is an interpreted high-level programming language for general-purpose programming (Severance et al, 2016).

5.10 Database description

Earlier on within the development process we determined that the use of a data store was important to store water leakage report. The data store additionally enabled us to experiment with different branches of the code. App Inventor library has a database component class called TinyWebDB. This class enables communication between the application and a specialized web-based database of (tag, value) pairs, hosted through the Google App Engine. TinyWebDB element has StoreValue ( ) and GetValue( ) methods, by which tagged values are placed into and retrieved from the database. Once a GetValue( ) server request succeeds, a GotValue event is generated with (tag, value) information that is made available to the application's logic.

Databases are most useful when it comes to storing information that fits into logical categories and also storing data persistently. TinyWebDB is a database on the cloud. This database is much bigger than the one found on mobile phones and can also be accessed by any user of the app. TinyWebDB has web interface (http://appinvtinywebdb.appspot.com/) that allows developers to browse and edit the database. The site also provides a human-readable web interface that a database administrator can use to examine the data stored there. With TinyWebDB you can write apps that store data on the Web, using a StoreValue/GetValue protocol. Data is always stored as a tag-value pair, with the tag identifying the data for later retrieval. Figure 5-6 illustrates web interface of TinyWebDB database and Table 5-1 below shows event handlers of TinyWebDB and its description.



App Inventor (TinyWebDB) Web Database Service

Search database for a tag

Tag:

Returned as value to TinyWebDB component:

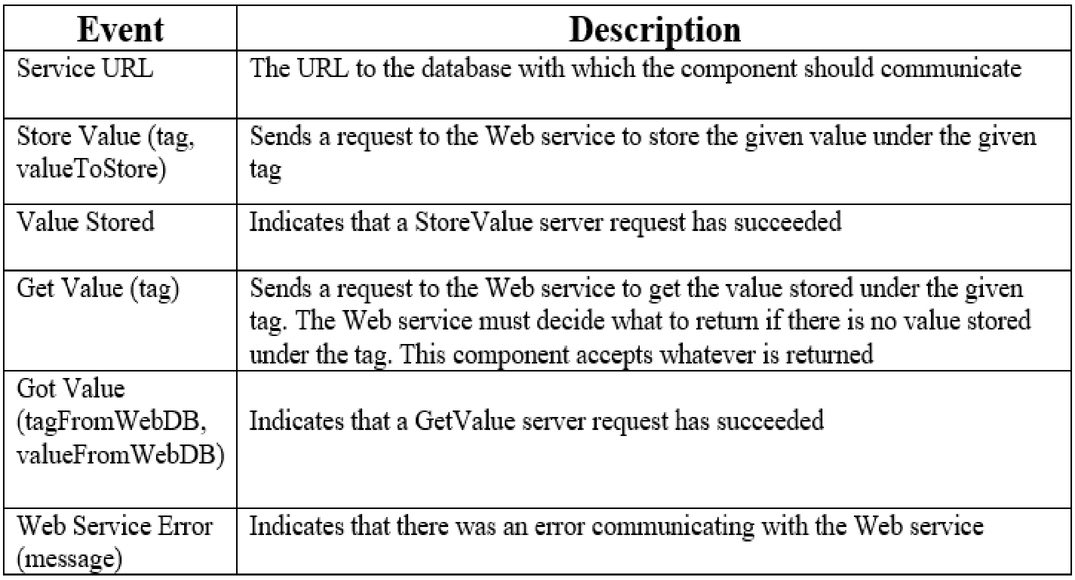
Store a tag-value pair in the databse

Tag:

Value:

Key	Value	Created (GMT)

Figure 5-6: TinyWebDB Database Interface

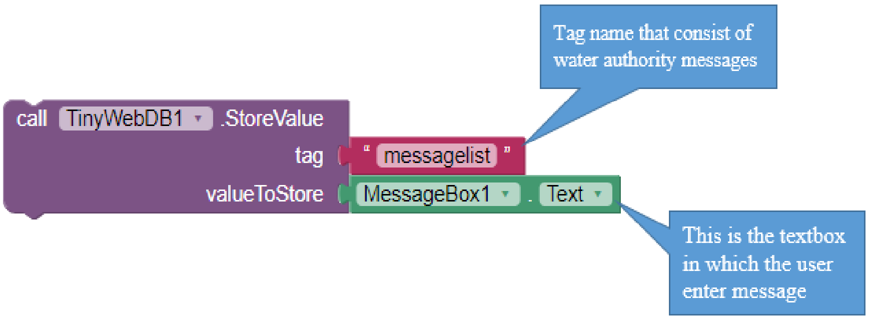


Event	Description
Service URL	The URL to the database with which the component should communicate
Store Value (tag, valueToStore)	Sends a request to the Web service to store the given value under the given tag
Value Stored	Indicates that a StoreValue server request has succeeded
Get Value (tag)	Sends a request to the Web service to get the value stored under the given tag. The Web service must decide what to return if there is no value stored under the tag. This component accepts whatever is returned
Got Value (tagFromWebDB, valueFromWebDB)	Indicates that a GetValue server request has succeeded
Web Service Error (message)	Indicates that there was an error communicating with the Web service

Table 5-1: TinyWebDB Events

5.11 Storing and Requesting Data with TINYWEBDB

The database for this project consists of two tags, namely messagelist and userreport. The usersreport tag contains all water leakages report sent by the community members. The messagelist tag consists of all messages send by the water authorities to the community members. To store data into the messagelist tag, a request is made to the web service to store the given value under the given tag. To store data, the following (Figure 5-7) TinyWebDB event block is called.

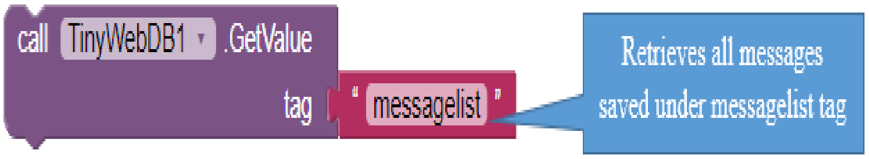


Tag name that consist of water authority messages

This is the textbox in which the user enter message

Figure 5-7: StoreValue Event Block

To retrieve all the messages in the database sent by the water authority, the following Figure 5-8 event block is called.



Retrieves all messages saved under messagelist tag

Figure 5-8: GetValue Event Block

5.12 Constraints and limitations of the system

System constraints are limitations or restrictions that emerge from within a system (Dettmer, 2006). The internet requirement is constraint for the application, since the application gets information from the database over the internet, it is important that a user should be connected to the internet for the application to function properly. Thus without internet connection user may not be able to access database information.

Another constraint is that, it is assumed that GPS components in all phones are not the same, probably the response time might not be the same for each one of them. Additionally, there might be a distinction between what navigation features every one of them gives.

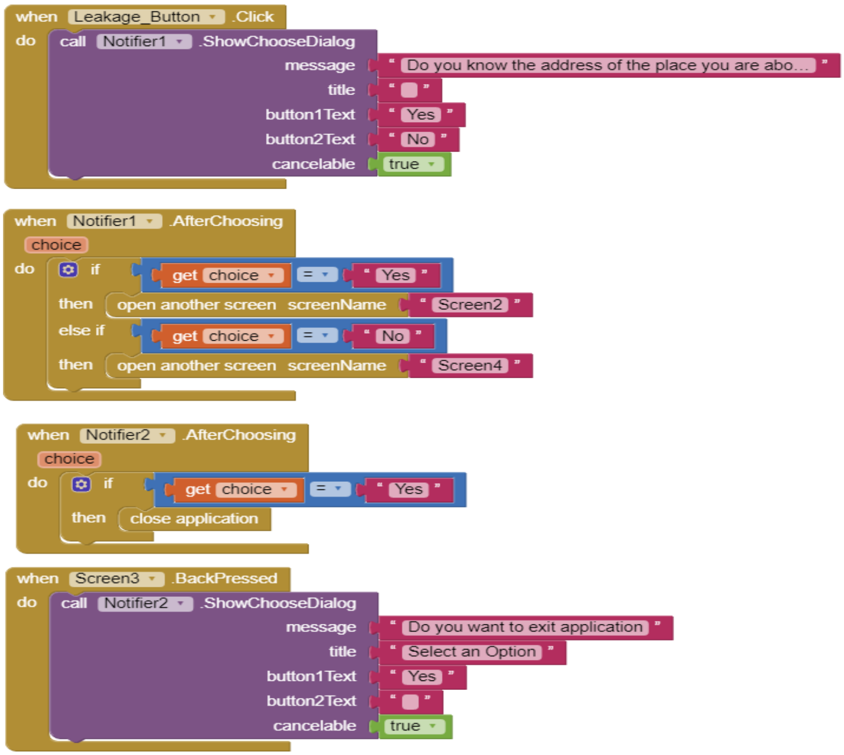
The application is restricted only to Android phones and is dependent on standard android operating system version 2.3 and above. The Android platform is chosen because majority of the respondents (57.5% in Alice and 47.5% in Fort Beaufort) indicated they own Android phones.

5.13 Implementation

To achieve a particular goal using the application, the application should provide the necessary information at the right time. This section discusses the coding blocks and user interfaces of the *SaveAmanzi* application. A partial list of coding of proposed application, namely: (i) main screen (ii) Technician Screen, (iii) Leakage Report Screen, (iv) Water Conservation Notification and (v) Safety Tools and Emergency Guidelines Screen. See Appendix 5 for complete coding blocks.

5.13.1 Coding of main interface

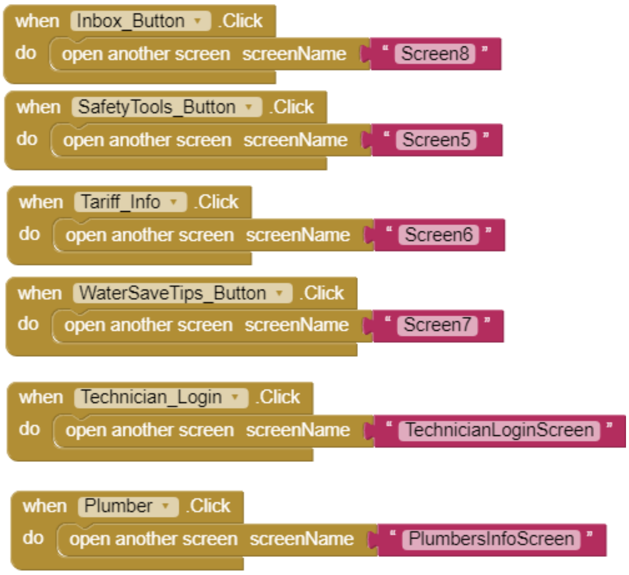
In Figure 5-9, there is one button event namely *Leakage_Button*, two notifier events and one screen event. In this button event, there is method (*Notifier1.ShowChooseDialog*) that shows a dialog box with two buttons, from which the user can choose. The *cancelable* is set true so there will be an additional cancel button. Pressing the *Leakage_Button* will raise the *Notifier1.AfterChoosing* event. The "choice" parameter to *Notifier1.AfterChoosing* will be the text on the button that was pressed, or "Cancel" if the cancel button was pressed. In the *Notifier1.AfterChoosing* IF THEN and ELSE IF THEN statements are used, and the user can select any choice from the list. When *Screen3.BackPressed* event is called, *Notifier2. AfterChoosing* is raised



```
when Leakage_Button .Click
do  call Notifier1 .ShowChooseDialog
                 message  " Do you know the address of the place you are abo... "
                   title  "  "
             button1Text  " Yes "
             button2Text  " No "
              cancelable  true

when Notifier1 .AfterChoosing
 choice
do  if      get choice = " Yes "
    then    open another screen  screenName  " Screen2 "
    else if get choice = " No "
    then    open another screen  screenName  " Screen4 "

when Notifier2 .AfterChoosing
 choice
do  if      get choice = " Yes "
    then    close application

when Screen3 .BackPressed
do  call Notifier2 .ShowChooseDialog
                 message  " Do you want to exit application "
                   title  " Select an Option "
             button1Text  " Yes "
             button2Text  "  "
              cancelable  true
```

Figure 5-9: Main Screen Code Blocks (a)

In Figure 5-10, there are six button events namely (i) *Inbox_Button*, (ii) *SafetyTools_Button*, (iii) *Tariff_Info*, (iv) *WaterSaveTips_Button*, (v) *Technician_Login*, and (vi) *Plumber*. In each button event, there is *open another screen* with the screen name in front; *open another screen* is a block code in App Inventor that is used to navigate to another screen. After selecting the desired button, the screen will be opened.

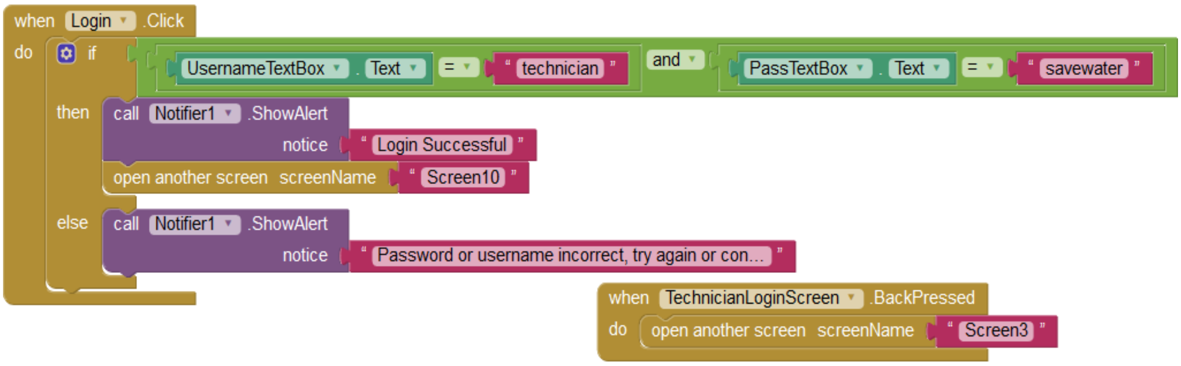


```
when Inbox_Button .Click
do   open another screen  screenName  " Screen8 "

when SafetyTools_Button .Click
do   open another screen  screenName  " Screen5 "

when Tariff_Info .Click
do   open another screen  screenName  " Screen6 "

when WaterSaveTips_Button .Click
do   open another screen  screenName  " Screen7 "

when Technician_Login .Click
do   open another screen  screenName  " TechnicianLoginScreen "

when Plumber .Click
do   open another screen  screenName  " PlumbersInfoScreen "
```

Figure 5-10: Main Screen Code Blocks (b)

5.13.2 Coding of technician interface

In Figure 5-11, there is one button event with name *Login.* In the button IF THEN and ELSE statement is used, it checks if the login credentials are correct or not. If the Login details are correct it calls the *Notifier* method to show alert "Login Successful" and then navigate to another screen or else, it gives error message. When *TechnicianLoginScreen.BackPressed* it goes back to the main screen (screen3).

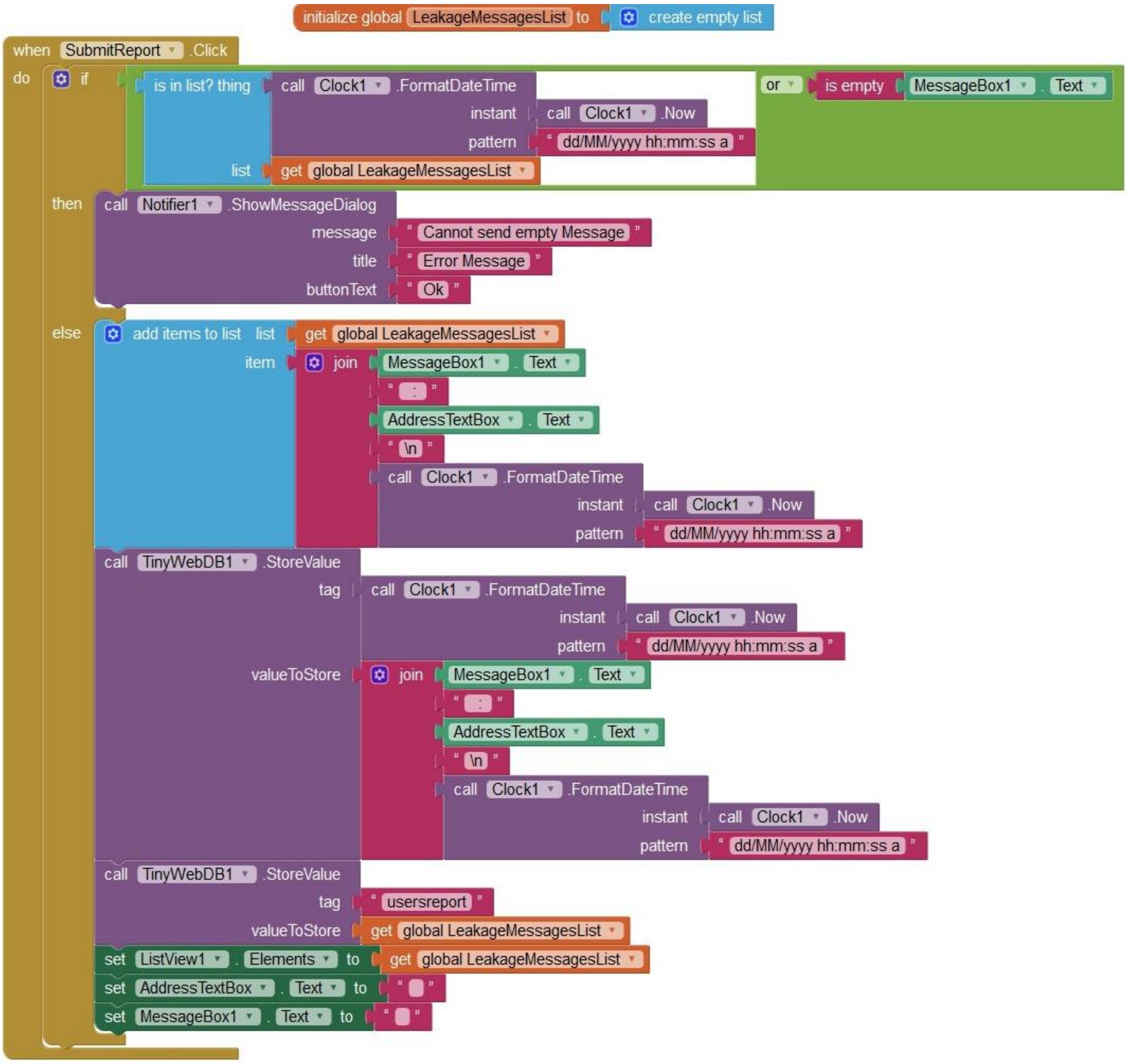


```
when Login .Click
do   if     UsernameTextBox . Text  =  " technician "  and  PassTextBox . Text  =  " savewater "
     then   call Notifier1 .ShowAlert
                              notice  " Login Successful "
            open another screen  screenName  " Screen10 "
     else   call Notifier1 .ShowAlert
                              notice  " Password or username incorrect, try again or con... "

when TechnicianLoginScreen .BackPressed
do   open another screen  screenName  " Screen3 "
```

Figure 5-11: Code Blocks of Technician Login Screen

5.13.3 Coding for water leakage report interface

In Figure 5-12, there is one button event namely *SubmitReport* and one global variable (*LeakageMessagesList*) which is initialized as empty list. In the button event IF THEN and ELSE statement is used: it first checks if a user input empty data in the textbox (*MessageBox1*) or not. If the user inputs nothing in the textbox and pressed *SubmitReport* button it shows error message or else, it adds the user inputted data to the list and then stores it in the TinyWebDB database under the tag name *usersreport*. Then it sets all the list elements onto a list view namely *ListView1*. The *AddressTextBox and MessageBox1* are then set to empty.

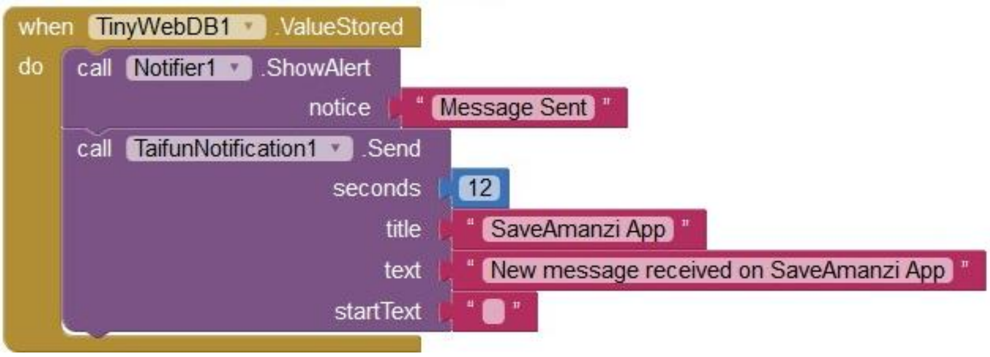


```
initialize global LeakageMessagesList to  create empty list

when SubmitReport .Click
do  if    is in list? thing  call Clock1 .FormatDateTime
                                         instant  call Clock1 .Now
                                         pattern  " dd/MM/yyyy hh:mm:ss a "
                       list  get global LeakageMessagesList
          or  is empty  MessageBox1 . Text
    then  call Notifier1 .ShowMessageDialog
                            message  " Cannot send empty Message "
                              title  " Error Message "
                         buttonText  " Ok "
    else  add items to list  list  get global LeakageMessagesList
                             item  join  MessageBox1 . Text
                                         " : "
                                         AddressTextBox . Text
                                         " \n "
                                         call Clock1 .FormatDateTime
                                                      instant  call Clock1 .Now
                                                      pattern  " dd/MM/yyyy hh:mm:ss a "
          call TinyWebDB1 .StoreValue
                           tag  call Clock1 .FormatDateTime
                                             instant  call Clock1 .Now
                                             pattern  " dd/MM/yyyy hh:mm:ss a "
                  valueToStore  join  MessageBox1 . Text
                                      " : "
                                      AddressTextBox . Text
                                      " \n "
                                      call Clock1 .FormatDateTime
                                                   instant  call Clock1 .Now
                                                   pattern  " dd/MM/yyyy hh:mm:ss a "
          call TinyWebDB1 .StoreValue
                           tag  " usersreport "
                  valueToStore  get global LeakageMessagesList
          set ListView1 . Elements to  get global LeakageMessagesList
          set AddressTextBox . Text to  " "
          set MessageBox1 . Text to  " "
```

Figure 5-12: Code Blocks for Water Leakage Report

5.13.4 Coding for update notifications

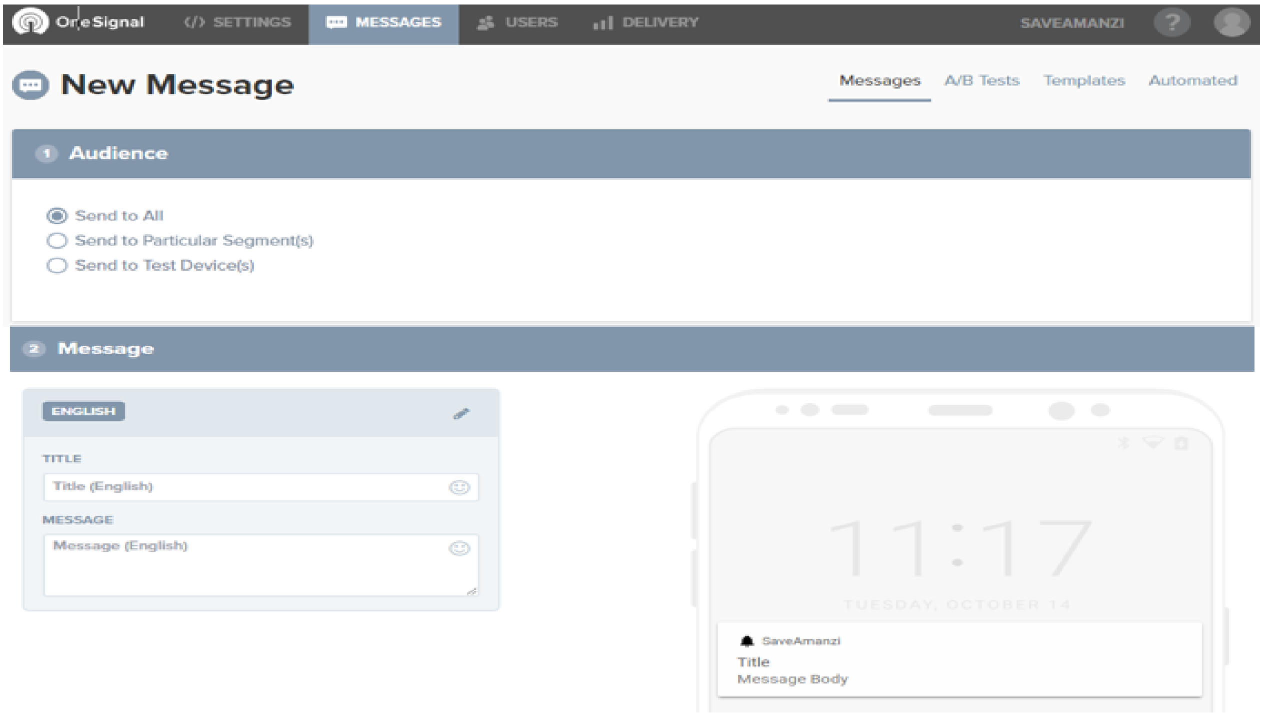
In figure 5-13, there is TinyWebDB one event; when a message is been added to the database, a user receives a notification of new message on the application after 12 seconds.



```
when TinyWebDB1 .ValueStored
do  call Notifier1 .ShowAlert
                    notice  " Message Sent "
    call TaifunNotification1 .Send
                    seconds  12
                      title  " SaveAmanzi App "
                       text  " New message received on SaveAmanzi App "
                  startText  " "
```

Figure 5.11 Figure 5-13: Code Blocks for Notifications

Figure 5-14 shows OneSignal push notification technology. With this web interface water conservation notification can be sent to all users of the *SaveAmanzi* application whether the application is in use or not, as long as they have the application installed on their Android phones they will be able to receive the notification. A message can also be sent to a particular segment or testing devices as illustrated in figure 5-14 below.



OneSignal | SETTINGS | MESSAGES | USERS | DELIVERY | SAVEAMANZI

New Message

Messages | A/B Tests | Templates | Automated

1 Audience

● Send to All
○ Send to Particular Segment(s)
○ Send to Test Device(s)

2 Message

ENGLISH

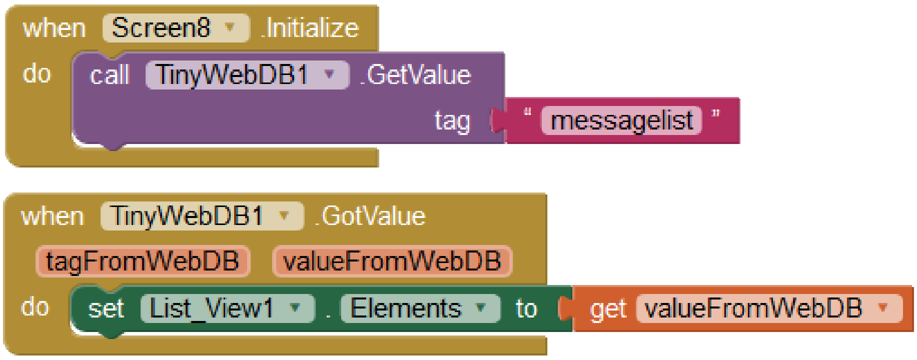
TITLE
Title (English)

MESSAGE
Message (English)

Figure 5-14: OneSignal Web Interface

5.13.5 Coding for retrieving messages from database

In figure 5-15, there is one screen event and one database event namely *Screen8.Initialize* and *TinyWebDB1.GetValue*. When screen8 is initialized it tries to get all messages in the database under the tag name *messagelist*. If it succeeds in getting any data the *TinyWebDB1.GetValue* event is raised, and then set all the retrieved messages on a list view.



```
when Screen8 .Initialize
do  call TinyWebDB1 .GetValue
                         tag " messagelist "

when TinyWebDB1 .GotValue
  tagFromWebDB   valueFromWebDB
do  set List_View1 . Elements to get valueFromWebDB
```

Figure 5-15: Code Blocks for Retrieving Messages from Database

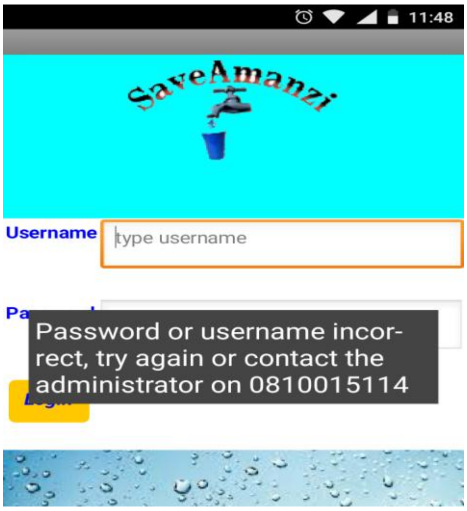
5.14 System testing

This section describes the testing of the *SaveAmanzi* application. The purpose of this test was to view and observe how usable the application will be for its users. The two levels of the testing that were used are unit testing and usability testing.

5.14.1 Unit testing

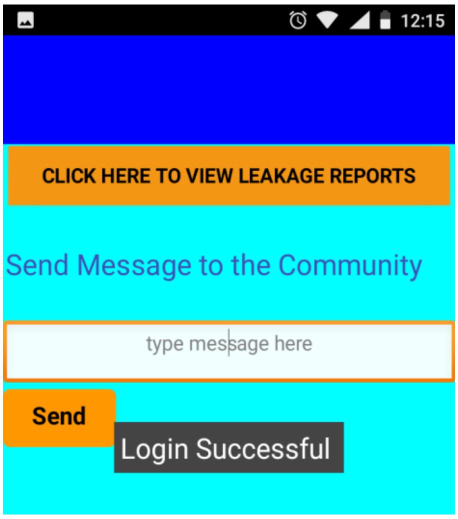
The unit testing refers to the individual testing of separate units of a software System (Kölling, 2006). In this project unit testing allows us to check whether a unit behaves the way it was intended and whether a unit corresponds to the design specifications. Unit testing provides the ability of testing each of the *SaveAmanzi* application unit independently (Sawant et al, 2012). Before any system could be integrated there should be the need to make sure that all its units or components are in good running state with no errors (Beizer, 2003). Input Design is the procedure of the converting user oriented inputs to a computer based format (Brighenti, 2009). Wrong inputted information has the most common sense of error in data processing. Any ambiguity conceiving at input leads to a total fault in output. The objective of designing the input data is to make the data entry easy, logical and error free as possible

(Rosenblatt, 2013). Inputs are made using forms and data validation is done at the time of inputting data. The *SaveAmanzi* application should be easy to use and provide feedback information for its user at any point in time. Figure 5-16 shows feedback information to user when wrong password or username is inputted in the login form and Figure 5-17 exhibit login success screen.



Password or username incorrect, try again or contact the administrator on 0810015114

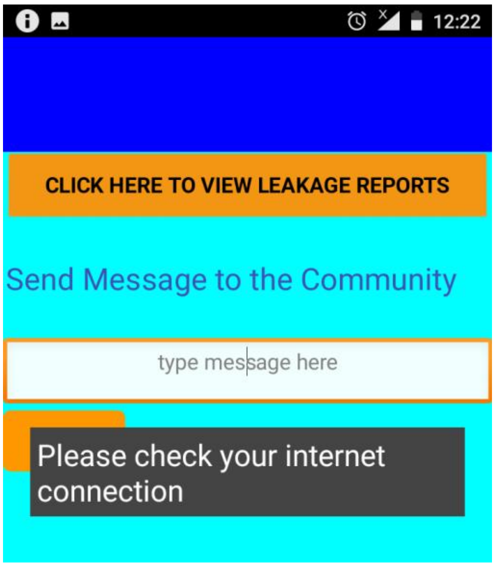
Figure 5-16: Login Error Screen


12:15
CLICK HERE TO VIEW LEAKAGE REPORTS
Send Message to the Community
type message here
Send

Login Successful

Figure 5-17: Login Success Screen

Figure 5-18 indicates there is no internet when a user tried to send message to the community members. Figures 5-19 shows an empty message cannot be sent when a user tried to report water leakage.

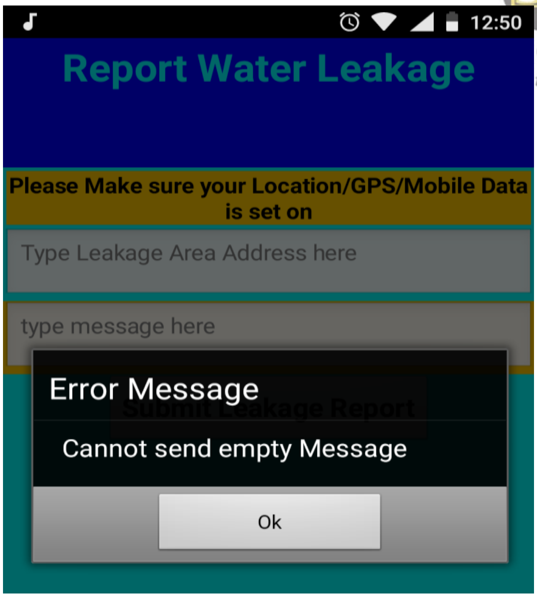


Send Message to the Community

type message here

Please check your internet connection

Figure 5-18: Internet Connection Error Screen



Report Water Leakage

Please Make sure your Location/GPS/Mobile Data is set on

Type Leakage Area Address here

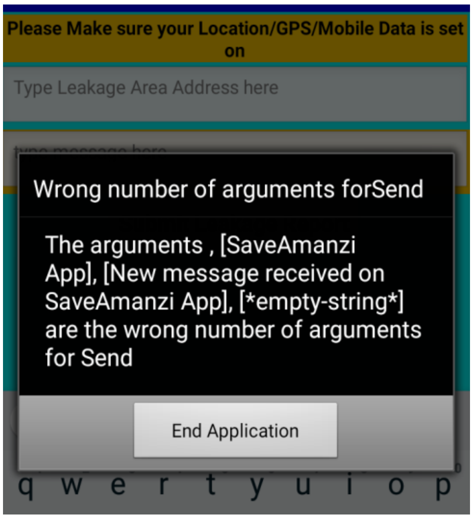
type message here

Error Message

Cannot send empty Message

Figure 5-19: Empty Message Error Screen

The following are some of the errors that were generated during the development of the unit testing;



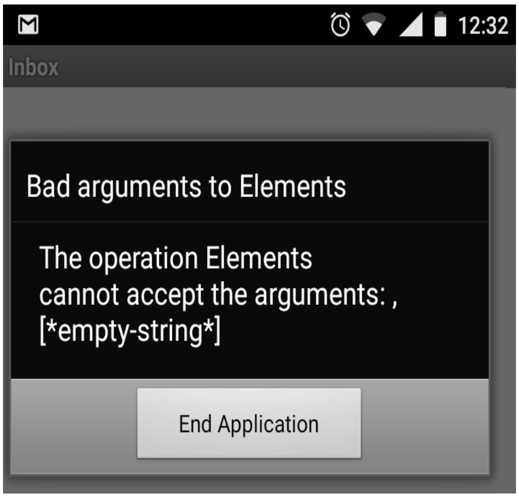
Please Make sure your Location/GPS/Mobile Data is set on

Type Leakage Area Address here

Wrong number of arguments forSend

The arguments , [SaveAmanzi App], [New message received on SaveAmanzi App], [*empty-string*] are the wrong number of arguments for Send

Figure 5-20: Wrong Arguments



Bad arguments to Elements

The operation Elements cannot accept the arguments: , [*empty-string*]

Figure 5-21: Bad Arguments

Figure 5-20 was generated when the user tried to send water leakage report. This error prevented the user's details to be captured into the database. The error also prevented users from receiving notification alert for new message since report was not able to be sent. Meaning that if this error is not addressed users will not be able to send leakage report. Figure 5-21 above was displayed when user wanted to navigate to their inbox and view message sent by water authorities. This error prevented the user from receiving message from local water authorities. Therefore, unit testing was meant to address these errors and correct them so that the system can perform their functions as they are intended to do.

5.14.2 Usability testing

Usability testing is a technique used to evaluate application by testing it on selected users (Moreno-Ger et al, 2012). It is a crucial method for examining how application users understand and use the application to execute a particular task. In this project usability testing was considered to be the best way to understand how users experienced using the application. Usability testing focuses on determining if the application is easy to learn, satisfying to use and contains the functionality that the users desire (Barnum et al, 2004).

5.14.2.1 Testing method

This section discusses the method used to conduct the test. To understand how users feel after using the application is vital when designing usability test scenarios that will help in data collection. Usability test scenarios were designed to test the effectiveness of screen layouts and on-screen features of the *SaveAmanzi* application. Usability test scenarios required users to complete actual tasks that are likely to encounter during actual testing of application (Rubin and Chisnell, 2008). Some of these tasks include:

1. Report water leakage
2. Login as technician
3. Water conservation awareness notification (by Administrator)

The next sections explain each task of the testing scenarios and why the task was chosen.

1. Report water leakage

In this task participants are supposed to navigate to the report water leakage button. No information about how to navigate to that command from the home screen was provided.

This task was specifically designed to evaluate the labels or commands of the home screen. It was important to investigate if participants knew the correct buttons to use at any particular instance. Reporting water leakages is a very important component of *SaveAmanzi* application. Thus it is necessary to test the effectiveness of this component.

2. **Login as technician**

In this task scenario participants were asked to go to the technician login screen. No information was provided for the participants on how to navigate to this destination. This task is created with the intention of knowing whether the application is understood and whether the commands or labels were effective guiding users. Since leakage reports can only be viewed through this command it is therefore important to know its effectiveness. In addition, water authorities can also send messages to the community members only by login successfully as technicians; hence it became necessary to test the effectiveness of this component as well.

3. **Water conservation awareness notification**

In this task participants were asked to send notification to all phones that have *SaveAmanzi* application installed. No information was provided about how to navigate to the text box where message can be entered and be posted afterwards. This task was designed to find whether notification can be sent from the web-portal. It was also designed to find out whether all phones that have the application can receive the notification send by the participants (administrator).

5.14.3 Selection of participants for testing

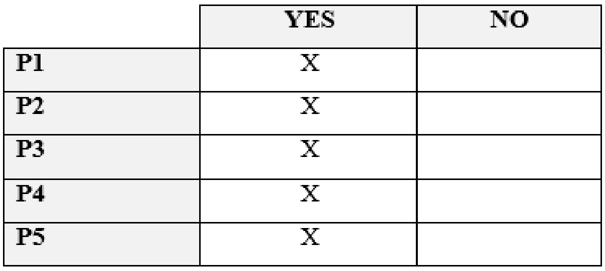
Five participants were involved in the testing process of the application, of which two were Fort Beaufort community members, two Alice community members and one Computers Science Honours Student who has a broad knowledge of Human Computer Interaction (HCI). Although the number of participants may seem rather small, research by Nielsen (2012), Lewis (2001), and others pointed out that about 80% of the usability problems associated with a product are detected by testing only four to five users with well-defined test scenarios.

To get the feedback of the testing session, questionnaire was designed for participants to provide their impressions after evaluating the application. These provided insight into the participants' thoughts and feelings about their experience in using the *SaveAmanzi* application, and their suggestions based on the testing experience. The questionnaire has been discussed as shown in appendix 4. Testing results has been discussed in the next chapter.

5.15 Testing results

This section comprises an important part of this research that describes the findings obtained after participants have tested the application. It is also important to know how participants were labeled. Participants are labelled as P1, P2, P3, P4, and P5 when cited in a table or when quoted. Results were collected in accordance with each question asked during the testing session.

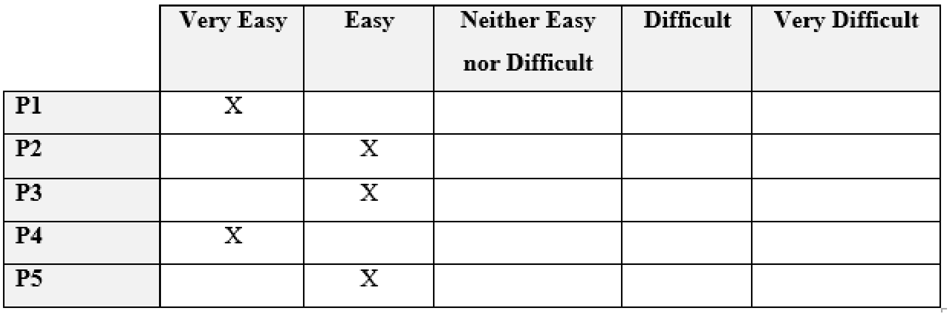
Table 5-2 lists each participant involved in the testing session and shows the results of Question 1.



	YES	NO
P1	X	
P2	X	
P3	X	
P4	X	
P5	X	

Table 5-2: Results of Question 1

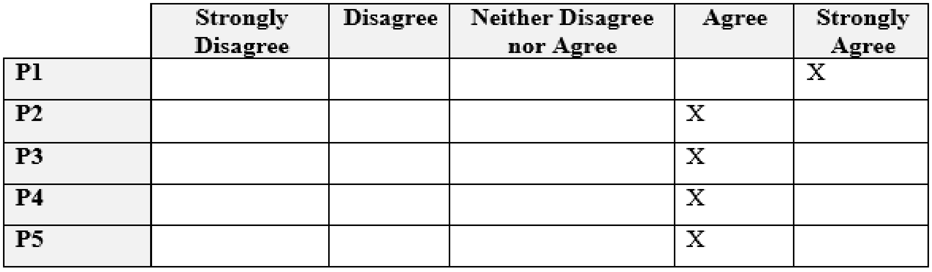
Table 5-3 below lists each participant involved in the testing session and shows the results of Question 2.



	Very Easy	Easy	Neither Easy nor Difficult	Difficult	Very Difficult
P1	X				
P2		X			
P3		X			
P4	X				
P5		X			

Table 5-3: Result of Question 2

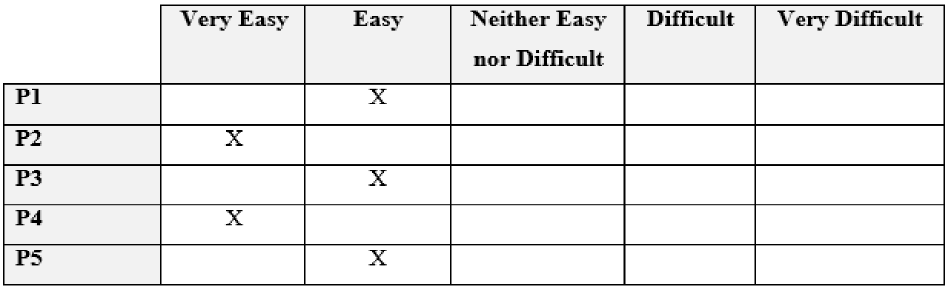
Table 5-4 lists each participant involved in the testing session and shows the results of Question 3.



	Strongly Disagree	Disagree	Neither Disagree nor Agree	Agree	Strongly Agree
P1					X
P2				X	
P3				X	
P4				X	
P5				X	

Table 5-4: Result of Question 3

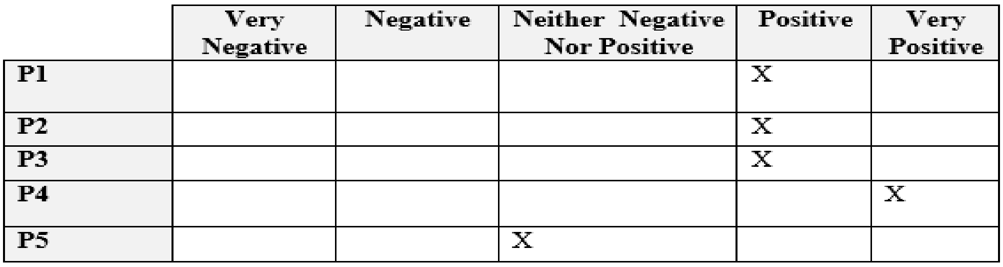
Table 5-5 lists each participant involved in the testing session and shows the results of Question 4.



	Very Easy	Easy	Neither Easy nor Difficult	Difficult	Very Difficult
P1		X			
P2	X				
P3		X			
P4	X				
P5		X			

Table 5-5: Result of Question 4

Table 5-6 lists each participant involved in the testing session, it shows the results of Question 5.



	Very Negative	Negative	Neither Negative Nor Positive	Positive	Very Positive
P1				X	
P2				X	
P3				X	
P4					X
P5			X		

Table 5-6: Result of Question 5

5.16 Discussion of test results

The Tables 5-2 to 5-6 have been discussed here, Table 5-2 shows that all the participants were able to complete the entire task. This result implies that the participants found the application usable in terms of completing task.

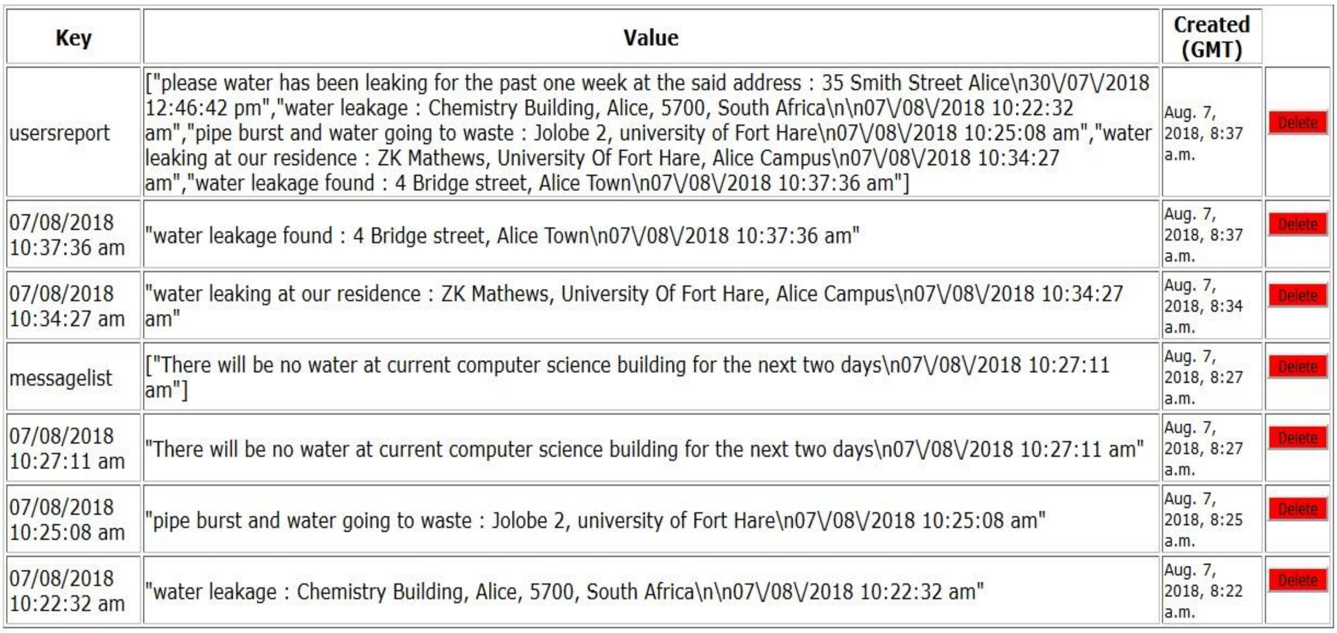
Participants were asked of their experience of using the *SaveAmanzi* application comparing with other application they have used before. Results in Table 5-3 shows that participants **P1** and **P3** rated the application as ‘Very Easy’ while participants **P2, P3** and **P5** rated the application as ‘Easy’. None of the participants found the application to be difficult or very difficult to use as compared with other application they have used. The usability of the application ranked high because most of the participants were able to complete their task.

Table 5-4, shows that participants were really satisfied with how the labels and buttons were organized. Four participants (**P2, P3, P4 and P5**) agreed that the buttons were well organized and easy to find while one participant (**P1**) strongly agreed that the labels were well organized.

Looking at the results in Table 5-5, majority of participants gave positive ratings. Three **(P1, P2** and **P3**) out of five rated the ease of navigation as Easy while participants (**P2** and **P4**) ease of screen navigation as Very Easy. It can be said that buttons and labels were well organized; hence they liked the way feedback information was presented.

Participants were asked to rate their overall impression and satisfaction with the application. Table 5-6 shows that almost all of the participants were satisfied with the application. Four out of five rated their overall impressions of the application as Positive (**P1, P2,** and **P3**) to Very Positive (**P4**). Only one participant was neutral about the application.

The usability test results (Table 5-6) shows that the application is usable and user-friendly. Out of five participants that tested the application, four gave positive feedback over their overall impression of the application. The functionalities of the application include: Report water leakages, tips of saving water, safety tools and emergency guidelines to minimize water loss, water tariff information, medium of communication between water authorities and community members and contact details of qualified local plumbers can be found on the application. Table 5-7 below shows data that was captured in the database during testing session.

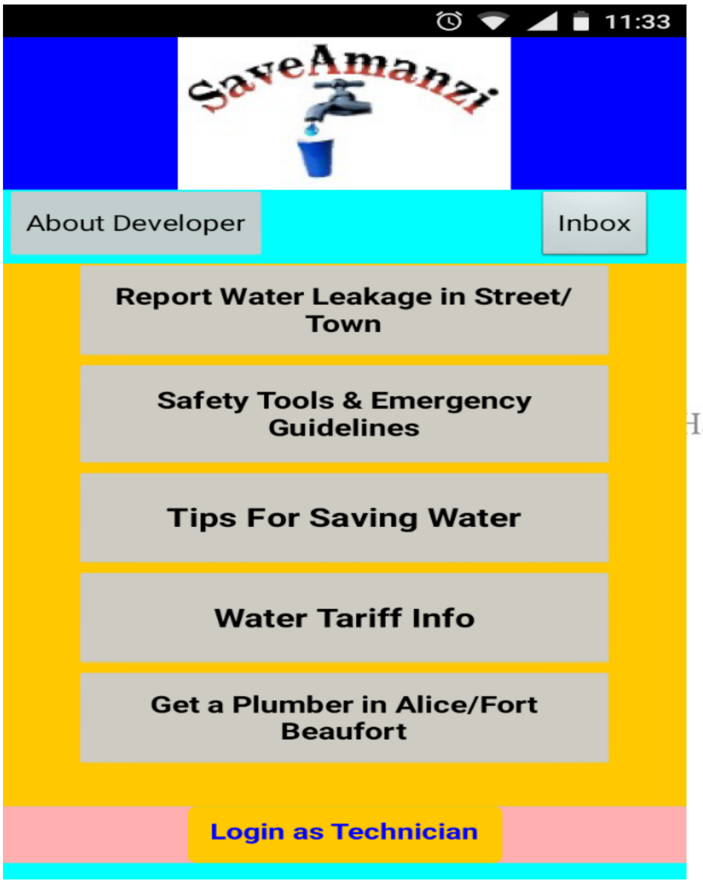


Key	Value	Created (GMT)	
usersreport	["please water has been leaking for the past one week at the said address : 35 Smith Street Alice\n30\/07\/2018 12:46:42 pm","water leakage : Chemistry Building, Alice, 5700, South Africa\n\n07\/08\/2018 10:22:32 am","pipe burst and water going to waste : Jolobe 2, university of Fort Hare\n07\/08\/2018 10:25:08 am","water leaking at our residence : ZK Mathews, University Of Fort Hare, Alice Campus\n07\/08\/2018 10:34:27 am","water leakage found : 4 Bridge street, Alice Town\n07\/08\/2018 10:37:36 am"]	Aug. 7, 2018, 8:37 a.m.	Delete
07/08/2018 10:37:36 am	"water leakage found : 4 Bridge street, Alice Town\n07\/08\/2018 10:37:36 am"	Aug. 7, 2018, 8:37 a.m.	Delete
07/08/2018 10:34:27 am	"water leaking at our residence : ZK Mathews, University Of Fort Hare, Alice Campus\n07\/08\/2018 10:34:27 am"	Aug. 7, 2018, 8:34 a.m.	Delete
messagelist	["There will be no water at current computer science building for the next two days\n07\/08\/2018 10:27:11 am"]	Aug. 7, 2018, 8:27 a.m.	Delete
07/08/2018 10:27:11 am	"There will be no water at current computer science building for the next two days\n07\/08\/2018 10:27:11 am"	Aug. 7, 2018, 8:27 a.m.	Delete
07/08/2018 10:25:08 am	"pipe burst and water going to waste : Jolobe 2, university of Fort Hare\n07\/08\/2018 10:25:08 am"	Aug. 7, 2018, 8:25 a.m.	Delete
07/08/2018 10:22:32 am	"water leakage : Chemistry Building, Alice, 5700, South Africa\n\n07\/08\/2018 10:22:32 am"	Aug. 7, 2018, 8:22 a.m.	Delete

Table 5-7: Data Captured in Database

We executed the *SaveAmanzi* application using Android based platform. Figure 5.22 to 5.24 below shows the output screens of the main application.

The main screen shows almost all the features of *SaveAmanzi* application. This screen shows right after the splash screen. Figure 5-22 is an illustration of the main screen. From this screen users can view information such as, safety tools and emergency guidelines, tips of saving water, water tariff and get contact details of qualified plumbers. Users can also report leakages from this screen and as well as check their inbox *(message from water authority*). Authorized water authorities can login as a technician from this screen.


11:33
SaveAmanzi
About Developer
Inbox
Report Water Leakage in Street/ Town
Safety Tools & Emergency Guidelines
Tips For Saving Water
Water Tariff Info
Get a Plumber in Alice/Fort Beaufort
Login as Technician

Hare

Figure 5-22: Main Screen

Figure 5-23 below shows login screen for water authorities. This screen allows water authorities to login as a technician in order to view the technician main screen. When the technician has logged on successfully, they can view water leakage report sent by the community members

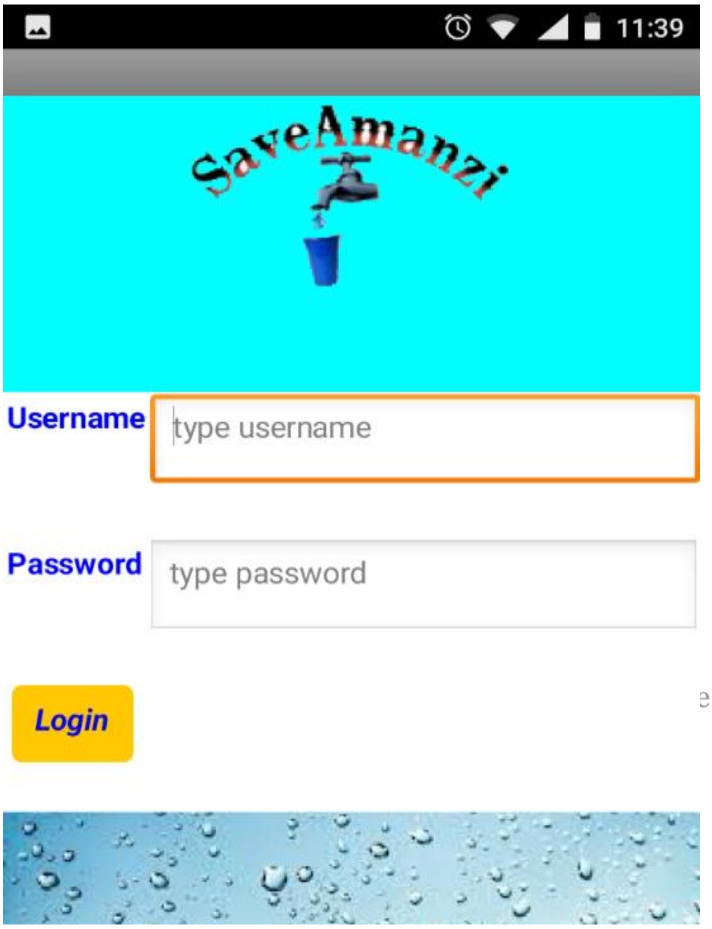


Figure 5-23: Technician Login Screen

In figure 5-24 below, when the technician has logged on successfully, they can view water leakage report sent by the community members. They can also send message to all community members who have the application installed on their phones.

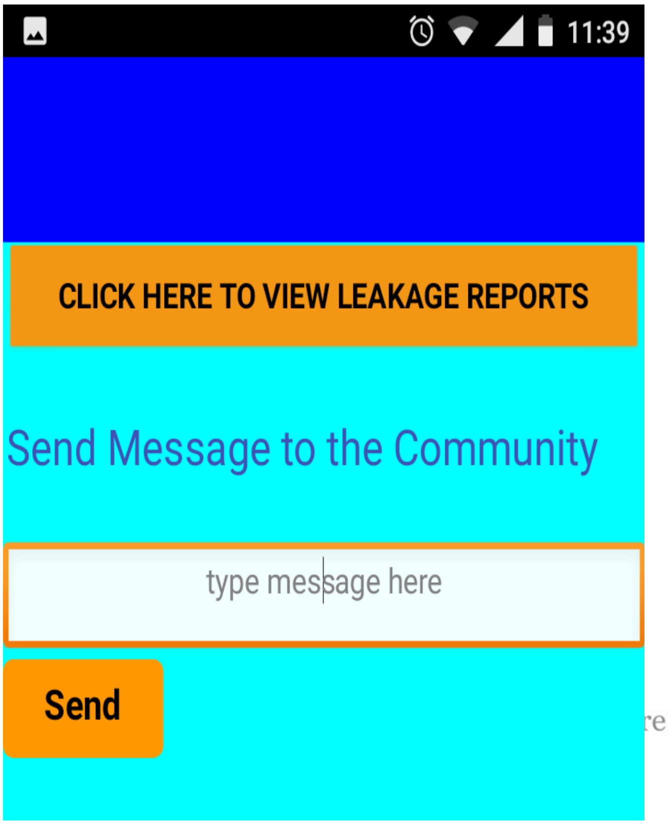

11:39
CLICK HERE TO VIEW LEAKAGE REPORTS
Send Message to the Community
type message here
Send
re

Figure 5-24: Technician Main Screen

5.17 Non-functional requirement

Non-functional requirements cover the rest of the requirements which are not covered by the functional requirements (Bochmann, 2010). The non-functional requirement elaborates a performance characteristic of the system (Glinz, 2007). Non-functional requirements of the *SaveAmanzi* have been explained below.

5.17.1 Security

Security is the condition of being protected against danger or loss. Therefore, every application has to be secured against hackers or unsigned access of it. The access permissions for *SaveAmanzi* application data may only be changed by the system's administrator. Passwords shall never be viewable at the point of entry or at any other time.

Any technician or water authority personnel who is to be allowed to access the application is given username and password and given his/her own access rights so that only authorized and authenticated users can access the leakage reports and also send message to community members. Also the password is not directly stored in the database to prevent hacking.

5.17.2 Availability

Unless the application is non- operational, the system shall present all users with notification informing them that the system is unavailable.

5.17.3 Maintainability

A development programmer who has at least one year of experience supporting this application shall be able to add a new product feature, including source code modifications and testing. The system shall not be shut down for maintenance more than once in a 24- hour period.

5.17.4 Scalability

The *SaveAmanzi* application shall be scalable to support unlimited growth in the number of users.

5.18 Conclusion

This chapter has discussed how the *SaveAmanzi a*pplication was designed and implemented. It has also shown user interfaces of the various functionalities of the application. It included the discussion of the application architecture, database design, testing, functional and non-functional requirements. Post-test results of the *SaveAmanzi* application have been analyzed in next chapter. The next chapter presents the overall key findings and discussion, and conclusions of this research.

CHAPTER SIX

RESEARCH FINDINGS, RECOMMENDATIONS, FUTURE WORK AND CONCLUSIONS

6.1Introduction

This chapter discusses key results, future work and finally, overall conclusion. The dissertation produced a number of findings in relation to the project objectives.

6.2Findings and discussion

6.2.1 Addressing research objectives

Objective 1: To determine the factors that influences the rate of increase of usage of mobile applications.

Mobile Phones have experienced phenomenal growth over the years. Applications in the sectors of agriculture, health, education and governance have illustrated the benefits that mobile phone applications can provide in collecting information and improving work processes. The first objective of the study was to take stock of factors that influence the growth of mobile applications usage. This objective was accomplished through a review of literature. Since mobile phone technology is on the rise, mobile applications are surely not the ones to be left behind. The 3G and 4G technology was introduced by network providers enabling people with 3G and 4G capable handsets to browse the net via their mobile units, place a video call, download streaming, send multi-media messages and many more. Mobile internet access provides common internet features like alerts, weather data, e-mail, search engines, instant messages, and game and music downloading. In chapter 2, we learnt that, mobile applications have become an integral part of people's daily routines. Applications have been changing the marketing landscape very rapidly in recent years. First, it was the ability to share content limitlessly, and now it has moved over to providing business owners with more creative ways to reach new customers. The convenience of having a mobile phone at hand has increased the tendency of most consumers shopping from home. This has led marketers to turn to mobile applications in a bid to reach these consumers and try to convert them into customers. In some developing countries, with little telecommunications infrastructure, this is the technology that gave poor people access to health and legal services. It helped that service providers offer a wide range of model handsets the public can choose from. This research provided an overview of existing mobile phone applications to help future change agents in the global water sector in chapter 2. Key lessons from this research are meant to help guide careful consideration of social factors, technical options, and program management strategies that can impact the success of a mobile phone solution. Mobile phone applications are powerful mechanisms for supporting the achievement of development goals. Through this research we have exhibited a survey of numerous manners by which mobile phone for development (M4D) applications and information interventions could be designed and used to serve better in good governance relating to water security. The research indicates that mobile phones have the potential to make data transfer more proficient, decrease physical data collection deficiencies, and promote effective monitoring at an effective cost.

Objective 2: To determine the current mobile development frameworks in use.

The features and characteristics elaborated within this dissertation might be essential to the selecting of one framework rather than other. From these features and characteristics, there is a possibility to recommend the more suited target group or market of each studied framework and to emphasize the principle qualities and shortcomings of each one.

It can be deduced from the assessment summary that PhoneGap, Xamarin, Sencha Touch and DragonRad are the frameworks that offer more compatibility for development of applications on various mobile operating systems. This is an ideal benefit for developers and business proprietors since one developed application can be built in other platforms for other business models. Hence these frameworks scored higher in terms of mobile OS support. App Inventor is the only framework that scored very poor among the frameworks evaluated, since it only support development of mobile application on android device at time of writing this research.

The license of tools is a useful parameter for comparison of frameworks. As shown in Table 4.2 PhoneGap, Xamarin and App inventor licenses are free and in addition, they offer an open-source support. This criterion is very crucial for developers that would create applications and need to give help to the development platform without having commercial restriction. Then again, commercial licenses could be valuable for organizations that need to get support directly from the manufacturer. Sencha Touch is authorised either under commercial license that allows closed source distribution. Enhanced support and a specialised advancement instrument can also be obtained at additional charge. Xamarin plugin to the Visual studio comes under paid licence and a free trial option and developer studio licence needs to be renewed every year. DragonRad is solely framework that presents differences in terms of its non-open source redistribution. The significance of open-source as a figure for making decision on a framework must not be neglected because it might result in technical debt if not accounted for since one is using libraries maintained by third parties.

Concerning framework IDE's, PhoneGap has three valuable approaches to create mobile applications, examples include XCode for applications on iOS, Eclipse for Android, and Visual Studio for Microsoft applications. This type of model is fine however restricts the freedom for developers to utilize their preferred IDE. Also, the method does not enable developers to have a centralized development environment; hence the needed skillset to compile the source code and create the final product (application) is high. Xamarin Studio may be great IDE, but supports creating of iOS and Android applications only in Mac operating system environment. On windows, Xamarin Studio gives the developers opportunity to create applications only for Android. Graphical user interface (GUI) components may still have to be coded numerous times to achieve a platform-specific look and feel and might not have access to all native functionality (Bell, 2017). App inventor like DragonRad is a simplified visual drag-and-drop programming tool, which depends on a web-based graphical user interface (GUI) builder (Papadakis et al, 2014). As a replacement for using written code, App inventor uses blocks that snap together like puzzle pieces. This takes out the requirement for developers to memorize a certain programming language or code and concentrate on how the application will interrelate with its users. Sencha Touch developers use IDE like JetBrain IDEs, Visual Studio, or some light editors such as Sublime Text or Notepadd++. But there are some limitations; hence Sencha provides paid commercial plugin which is Sencha IDE plugins to integrate into some types of IDEs to give more features.

In terms of learning curve, PhoneGap and Sencha Touch are the frameworks that have supports for web programming languages like HTML, CSS and JavaScript. Applications are designed in HTML, CSS and JavaScript what lowers the barrier of adoption for web developers. These languages are standard language for web developers hence it learning curve will be less as compared to lesser known languages. With regard to PhoneGap and Sencha Touch, it is critical to have a beginner competency range of abilities in at least one mobile native operating system programming. HTML, CSS, and JavaScript in alignment with web standards are highly suitable for creating cross-platform applications since they are standardized, well known, simple but capable and well-supported. App Inventor and DragonRad have support for its own language, drag-and-drop Block editor. The drag-and-drop environment makes development easy than learning and typing codes. On the other hand, Xamarin uses C# as a programming language, therefore to be able to develop a native application, a developer will have major learning curve, as he or she is expected to learn the C# language and get some essential knowledge of logical application development.

PhoneGap can be complied locally using the Command-line Interface (CLI) and also using PhoneGap Build cloud service to simplifier the build and application packaging process. Xamarin studio comes with a respective native builds which can be transferred onto the platform stores. Native packaging is a feature in Sencha Touch and with a single command, one can package application for iOS and for Android (Clark and Johnson, 2013). With regard to application packaging and distribution, all the five frameworks have established tools for packaging, installing and distribute application but the one that provides more than two distribution channel and scored high on this criterion is App inventor.

Web applications or applications created with PhoneGap tend to appear slightly different from native applications and more like web sites. Xamarin Compiles into a true native application or APK, uses native user interface tools and has cross platform benefits. Xamarin is a good match to native mobile development, and can help make development easy and more efficient. Perhaps developers undervalued it due to lack of knowledge and patience with regards to its learning curve (Gridin, 2015). The look and feel of a Sencha Touch applications look like that of iOS applications. Sass is an extension to CSS which can be used for customization. Despite the fact that Sass enables powerful and easy design modification, a native Look and Feel for Sencha application would require high effort. Sencha Touch is ideal for large and complex applications. App inventor and DragonRad use its own native IDEs and therefore leverage the native look and feel of mobile applications.

The performance of PhoneGap applications has frequently been criticized particularly in the case graphics and animations. Runtime performance of a Sencha Touch application is close to native performance. Frameworks that can deploy native applications instead of hybrid and web were scored higher as shown in table 4.3 of chapter 4. This was because applications with native user interface generally have higher performance.

Objective 3: To identify the most suitable mobile development framework for development of mobile applications.

Lots of frameworks are accessible online today and it has been the key challenge to understand the best one to attain the goals of a specific user or organisation. There is no one definitive framework for applications however decisions should be tailored to needs and expectations, carefully weighted strength and weakness. There will never be a winner once it involves programming languages and frameworks. The winner is the choice you think are the best for you and your product currently and into the longer term (Majchrzak et al, 2015). It might be that the best option is to write Swift for iOS and Java for Android, simply because you're amazing at those. One important factor which is of high interest to consider is the development time and complications while developing. In concluding, the selection of approach and framework for application development completely depends on requirements of user and developer. This investigation presented a model and guidelines for supporting the choice of the right mobile development framework for the development of a new mobile application from given a requirement. Summarizing from the evaluation from chapter 4, App Inventor is impressive framework, and programming method provided by the framework clearly differs from conventional languages and attracts a beginner. Looking at all the frameworks evaluated in this research, we chose App Inventor for the development of SaveAmanzi application. App Inventor seems to be the appropriate option for this research. App inventors have support for the requirements that was outlined for *SaveAmanzi* application.

Objective 4: To develop mobile application for water usage management for Alice and Fort Beaufort communities.

The goal of this objective was to develop *SaveAmanzi* application, an application for Android mobile devices using the App Inventor language and MIT App Inventor development and packaging service. The process of application development is not so different from the traditional software development process. However, if we develop Android applications more quickly with a project team having developers who deficient in Android SDK and Java programming experience, it will greatly contribute to improving applications development productivity. The App Inventor is used by many departments relating computer science and engineering in college and many students interested in smart applications in school. Those who are deficient to programming knowledge can create Android applications by using App

Inventor. App Inventor has the advantage of component visual programming, where one can perform drag and drop of visual components, and then give programmable behaviours to logic blocks to develop mobile applications easily. Therefore, using App Inventor for application prototyping is highly efficient because it is easy to use and intuitive and it connects well with other components in the development environment.

To demonstrate the selling features of App Inventor, *SaveAmanzi* application was developed to address some of the water related challenges found in the two research sites. To develop this application, there was the need to identify common challenges experienced by people in Alice and Fort Beaufort. Survey conducted at the two research sites led to the identification of common challenges. This gave us the requirements necessary for the development of the *SaveAmanzi* application. The application was implemented based on the objectives of this project.

6.2.2 Addressing research questions

Question 1: What influences the rate of growth of usage of mobile applications?

There are key factors that drove the enormous growth of mobile applications. There is a worldwide increase in smartphone usage, as they turn out to be more economical and can be used in all aspects of our regular day to day activities. Technically, a rise in smartphone usage means there are additional opportunities for mobile applications developers, as businesses progressively incorporate applications into their business and new businesses operate totally off an application. Businesses and technologists alike understand the significance of having their application in a smartphone, whether it is the Apple iPhone or Android phones. Also, users loved the convenience of applications they could download and use on their phones. Mobile application users love the convenience and ease of mobile applications. Since most mobile phone owners kept their phones with them, they could conduct a quick search for the application they needed. Once found, downloading and installing the application is easy. Regardless, social media applications have essentially affected how we interact with each other. Through social media, we are able to easily interact with people from all over the world. Mobile applications have gone far beyond social media, games, and other personal apps. Employers are now using mobile applications for various work-related reasons, with most applications cantered around productivity, operations, and management. A mobile application might be used to dispatch a maintenance team on the shop floor or to track the location of a driver while making deliveries. The growth of mobile application is also due to advancements in mobile technologies which require businesses to have a vision for next few years. Together, these factors drove the expansion of mobile applications in ways desktop application developers couldn't envision. There is no doubt that mobile applications will be around for quite some time to come. While technologies may change and the way applications themselves are built, we will still need to use them to connect with each other, to shop, and to pay bills.

Question 2: What types of mobile development frameworks are there or are being used?

Mobile Development Frameworks offer developers a broad foundation of functionality that can be modified for mobile application specific software. Frameworks can be in three classifications: native frameworks for platform-specific development, mobile web application frameworks, and hybrid applications, which combine the features of both native and mobile web application frameworks. The following are some of the mobile development frameworks that are currently used in the categories mentioned above:

Native application frameworks: *Android Studio, Xcode, App Inventor, DragonRad, etc.*

Web application frameworks: *Ionic, jQuery Mobile, Sencha Touch, Reactive Native, etc.*

Hybrid application frameworks: *Xamarin, PhoneGap, Framework7, Native Scripts, etc.*

Question 3: What is the most appropriate framework to be used?

Mobile application development is presently at its peak, with a lot of developers jumping on the trend to make the most on this lucrative field. However, the most important drawback that developers face across streams is the different development frameworks, programming languages and operating systems. Also when we come to evaluate the comparison between such topics there is no best ever, it is your purpose that helps you to find a right fit for you and your project. Identifying need and requirements is what required figuring out the best for you. Therefore, for the purpose of this research, we chose App Inventor for the case application not because it is the best but it fit well for the requirements gathered for *SaveAmanzi* application.

Question 4: How can a novel mobile application be designed to address water loss problems?

The report of various current and past mobile applications in the water division in Chapter 2, section 2.5, has demonstrated that mobile applications can aid management functions of the water sector and the literature is optimistic about the value of mobile application regarding development in the sector. *SaveAmanzi* application was developed to address some of the water related challenges found in the two research sites. The application handles following water issues that were found during requirement gathering:

1. **Water leakage Reporting**
   - User should be able to report water leakages
   - Water authorities should be able to read water leakage report
2. **Medium of communication between water authorities and community**
   - Users must be able to receive and read messages from water authorities
   - Water authorities must be able to send messages to users
3. **Safety tools and emergency guidelines to minimize water loss at home**
   - Users must be able to read safety tools and emergency guidelines information
4. **Tips of saving water**
   - Users must be able to read tips of saving water information provided in the application
5. **Awareness notification on Water conservation**
   - Users should be able to receive notification on water conservation
6. **Water Tariff information**
   - Users should be able to read information on water tariffs
7. **Contact details of Local qualified plumbers**
   - Users should able to get contact details of local plumbers provided on the application.

6.3Future work

There are as yet numerous issues to handle, for example, the human side in the form of end-user experience. Future work will incorporate a longitudinal research to explore framework maturity, and in addition the perspectives on user interface design and interaction.

To make the application accessible by many and more effective, the following has been proposed as part of future work. Currently, the application only interacts with users in English. In future the application should be able to communicate with users in other languages for example, Xhosa. At the moment, the *SaveAmanzi* only runs on the Android platform, other platform like iOS will be considered so that application can reach more community members, hence enhancing scalability.

Further research should be carried to establish the extent water authorities in these two sites have adopted mobile applications in water resources management and climate change adaptation.

6.4Overall conclusion

The aim of this research was to study different mobile development frameworks for developing mobile applications for diverse mobile platforms and the best development framework for different use cases. Five main development frameworks were identified and evaluated. In this research, we presented a set of criteria for evaluating mobile development frameworks for mobile applications. The summary results of the evaluation have been compiled in Table 4-3, which can be used as references. The evaluation was done based on a list of criteria, which included 7 items namely, mobile operating system platform support, framework license and cost, development environment, learning curve, packaging and distribution, look and feel, and runtime performance. Each framework was rated by each criterion on a scale of 1(very poor) to 5(very good). These rate values represented how important each criterion is for a given framework. The resultant analysis of the framework according to the criteria has shown that App Inventor is preferred by the majority of programmer as it ticked all boxes of the measurement variables. However, we still maintain the idea by Majchrzak et al (2015) that, there will never be a winner once it involves programming languages and frameworks. The winner is the choice you think are the best for you and your product currently being worked on.

App Inventor turned out to be a very good choice for the case application (SaveAmanzi), especially when working with very limited time, budget and resources. App Inventor is native mobile development framework, it must be remembered that native applications provide better performance and provide more fluid and responsive user experience than web applications. App Inventor is a great alternative for native development when the application being developed is relatively simple. Setting up App Inventor is simple and once the setup is completed, App Inventor development is very efficient and straightforward. Development using App inventor's blocks editor programming which is easy to learn, makes it easy for any developer to start experimenting with App Inventor.

This development process began with information gathering and user study, in order to understand the principles for *SaveAmanzi* application. The survey along with academic literature and discussions formed the basis for the requirements and the technical implementation of *SaveAmanzi* application. Afterwards, *SaveAmanzi* application was designed and developed as a functional prototype for Android mobile phones. Testing and evaluation was also conducted during and at the end of thesis project by both the experts and participants that were selected.

Overall the project can be considered a success. Although there were some difficulties with deployment of *SaveAmanzi* application on real Android phone, the problems were quickly solved. The application was able to meet all requirements and the testers were pleased with the result as indicated in Table 5-6. Despite the increasing literature track of mobile application development frameworks, our research still handles a unique subject.

REFERENCES

Abawi, K., (2013). Data collection instruments (Questionnaire & Interview). Training in Sexual and Reproductive Health Research Geneva.

Aeris, (2017). SweetSense uses IoT to improve access to safe water in developing countries.

Allen, M., (2009). Palm webOS: The Insider's Guide to Developing Applications in JavaScript using the Palm Mojo™ Framework. "O'Reilly Media, Inc.".

Alshamrani, A. and Bahattab, A., (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. International Journal of Computer Science Issues (IJCSI), 12(1), p.106.

Amasha, M.A. and Al-Omary, S., (2017). Quizrevision: A Mobile Application using the Google MIT App Inventor Language Compared with LMS. learning, 8(6).

Amatya, S. and Kurti, A., (2014). Cross-platform mobile development: challenges and opportunities. In ICT Innovations 2013 (pp. 219-229). Springer, Heidelberg.

Andahi, A., (2018). iOS Application Development.

Annerose, D and mWater, (2018). e-Business development service to improve rural & Small town water service delivery. URL: https://www.wsp.org/sites/wsp.org/files/publications/Presentation-Presenter-Daniel-Annerose.pdf (accessed 07 May 2018).

Antón, A.I., Carter, R.A., Earp, J.B. and Williams, L.A., (2001). EPRAM: Evolutionary Prototyping Risk Analysis & Mitigation (e-Commerce Software Development Process Document). North Carolina State University at Raleigh, Raleigh, NC.

Arnold, B.F. and Colford Jr, J.M., (2007).Treating water with chlorine at point-of-use to improve water quality and reduce child diarrhea in developing countries: a systematic review and meta-analysis. The American journal of tropical medicine and hygiene, 76(2), pp.354-364.

Bala, K., Sharma, S. and Kaur, G., (2015). A research on smartphone based operating system. International Journal of Computer Applications, 121(1).

Baran, M.L. ed., (2016). Mixed methods research for improved scientific study. IGI Global.

Barnum, C., Henderson, E., Hood, A. and Jordan, R., (2004). Index versus full-text search: a usability research of user preference and performance. Technical Communication, 51(2), pp.185-206.

Bertram, D., (2013). Likert Scale is the meaning of life. Yugoslavia: University of Belgrade. URL: http://poincare. matf. bg. ac. rs/~ kristina/topic-dane-likert. pdf.

Bian, H., (2015). Mixed methods research.

Biggers, M., Brauer, A. and Yilmaz, T., (2008). Student perceptions of computer science: a retention research comparing graduating seniors with cs leavers. In ACM SIGCSE Bulletin (Vol. 40, No. 1, pp. 402-406). ACM.

Bochmann, G.V., (2010). Non-Functional Requirements.

Bönström, D., (2014). Smartphone application in PhoneGap: M2C's electric vehicle smart charger.

Boushehrinejadmoradi, N., Ganapathy, V., Nagarakatte, S. and Iftode, L., (2015). Testing cross-platform mobile app development frameworks (t). In 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 441-451). IEEE.

Boustedt, J., 2010. Ways to Understand Class Diagrams.

Bradley Mitchel, 2019. LTE (Long Term Evolution) Definition. URL: https://www.lifewire.com/definition-of-lte-broadband-817465 (accessed 10/03/2019).

Brandes, O. M., Ferguson, K., M'Gonigle, M., Sandborn, C., (2005). At a Watershed: Ecological Governance and Sustainable Water Management in Canada, Victoria, POLIS Project on Ecological Governance, University of Victoria.

Butler, M., (2011). Android: Changing the mobile landscape. IEEE Pervasive Computing, 10(1), pp.4-7.

Campbell, D.T. and Stanley, J.C., (2015). Experimental and quasi-experimental designs for research. Ravenio Books.

Cassavoy, L., (2018). What is 4G wireless? URL: https://www.lifewire.com/what-is-4g-wireless-577577 (accessed 06 September 2018).

Champanis, M., Rivett, U., Gool, S. and Nyemba-Mudenda, M., (2013). ICTs in the water sector–where do we stand?. Water Research Commission Report to be published 2013.

Charland, A. and Leroux, B., (2011). Mobile application development: web vs. native. Queue, 9(4), p.20.

Ciman, M., Gaggi, O. and Gonzo, N., (2014). Cross-platform mobile development: a research on apps with animations. In Proceedings of the 29th Annual ACM Symposium on Applied Computing (pp. 757-759). ACM.

City of Cape Town, (2017). Water and sanitation customer perception and satisfaction survey project-final report. URL: https://resource.capetown.gov.za/documentcentre/documents/city%20research%20reports%20and%20review/final_report_coct%20wssp_29112014cc.pdf (accessed 05 July 2018)

Clark, J.E. and Johnson, B.P., (2013). Sencha touch 2 mobile javascript framework. Packt Publishing Ltd.

Coskun, V., Ok, K. and Ozdenizci, B., (2013). Professional NFC application development for android. John Wiley & Sons.

Creswell, J. W., (2014). Research Designs 4th edition. Washington DC: SAGE Publications.

Creswell, J.W. and Clark, V.L.P., (2017). Designing and conducting mixed methods research. Sage publications.

Dalmasso, I., Datta, S.K., Bonnet, C. and Nikaein, N., (2013). Survey, comparison and evaluation of cross platform mobile application development tools. In Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International (pp. 323-328). IEEE.

Deloitte, G. S. M. A. (2012). Sub-Saharan Africa Mobile Observatory; 2012. GSM, London Google Scholar.

Despa, M.L., (2014). Comparative study on software development methodologies. Database Systems Journal, 5(3), pp.37-56.

Dettmer, H.W., (2006). Systems and constraints: the concept of leverage. Port Angeles, WA: Goal Systems International. Retrieved January, 21, p.2009.

Do, D.Q., Comparison between Tools Used for Cross Platform Mobile Application, URL: https://www.researchgate.net/profile/Dat_Do4/publication/287736318_Comparison_between_Tools_Used_for_Cross-Platform_Mobile_Applications/links/5679088508aebcdda0ed40d5/Comparison-between-Tools-Used-for-Cross-Platform-Mobile-Applications.pdf (accessed 06/10/2018)

Dolnicar, S. and Hurlimann, A., (2010). Australians' water conservation behaviours and attitudes. Australasian Journal of Water Resources, 14(1), pp.43-53.

Drew T., (2011). Selecting Mobile ICT Devices For Agriculture Services & Applications In Sub-Saharan Africa: Supported by USAID's Fostering Agriculture Competitiveness Employing Information Communication Technologies (FACET) project under the Financial Integration, Economic Leveraging, Broad-Based Dissemination and Support Leaders with Associates award (FIELD-Support LWA).

Edwards, L. and Barker, R., (2004). Developing series 60 applications: a guide for symbian OS C++ developers. Pearson Higher Education.

Elo, S. and Kyngäs, H., (2008). The qualitative content analysis process. Journal of advanced nursing, 62(1), pp.107-115.

Faisandier, A., (2013). Systems architecture and design. Sinergy'Com.

Fass, S. and Turner, K., (2015). The quantitative and qualitative content analysis of marketing literature for innovative information systems: the Aldrich Archive. arXiv preprint arXiv:1505.04401.

Fendelman, A., (2018). 1G, 2G, 3G, 4G, & 5G Explained. URL: https://www.lifewire.com/1g-vs-2g-vs-2-5g-vs-3g-vs-4g-578681 (accessed 02 October 2018).

Firtman, M., (2010). Programming the mobile web. "O'Reilly Media, Inc.".

Foster, T., Hope R., Thomas, M. (2012). Impacts and implications of mobile water payments.

Gale, H.F., 1997. Is there a rural-urban technology gap?. Total quality management (TQM), 45, pp.1-5.

George, T., Bagazonzya, H., Ballantyne, P., Belden, C., Birner, R., Del Castello, R. and Treinen, S., (2011). ICT in agriculture: connecting smallholders to knowledge, networks, and institutions. Washington, DC: World Bank.

Gestwicki, P. and Ahmad, K., (2011). App inventor for Android with studio-based learning. Journal of Computing Sciences in Colleges, 27(1), pp.55-63.

Glinz, M., (2007), October. On non-functional requirements. In Requirements Engineering Conference, 2007. RE'07. 15th IEEE International (pp. 21-26). IEEE.

Götz, F.M., Stieger, S. and Reips, U.D., (2017). Users of the main smartphone operating systems (iOS, Android) differ only little in personality. PloS one, 12(5), p.e0176921.

Greenemeier, L., (2011). Aqua plan: Could cell phones help aid workers ensure Haiti's supply of clean drinking water. Scientific American.

Gridin, O., (2015). Xamarin as a tool for mobile development.

Gupta, A. and Gupta, C., (2018). A Collaborative Situational Method Engineering Approach for Requirement Gathering: A Re-Defined View. International Journal of Information Retrieval Research (IJIRR), 8(1), pp.1-19.

Hartmann, G., Stead, G. and DeGani, A., (2013). Cross-platform mobile development. Tribal, Lincoln House. The Paddocks, Tech. Rep. Disponível em:< https://wss. apan. org/1539/JKO/mole/Shared% 20Docum ents/Cross-Platform% 20Mobile% 20Development. pdf>. Acesso em 16 out.

Hasan, M. and Haque, M., (2016). Mobile Application Development Approaches: Recommendation for E-commerce Enterprises.

Heitkötter, H., Majchrzak, T.A., Ruland, B. and Weber, T., (2013). Evaluating Frameworks for Creating Mobile Web Apps. In WEBIST (pp. 209-221).

Heland F, Bondesson A, Nyberg M, Westerberg P., (2015). The Citizen Field Engineer, Crowd sourced Maintenance of Connected Water Infrastructure. In: Johannsen, V. et al. (ed).

Hellstrom, J. (2010). The Innovative Use of Mobile Applications in East Africa. SIDA Review 2010:12, SIDA, Stockholm, Sweden.

Hellström, J. and Jacobson, M., (2014). ‘You Can’t Cheat the Community Anymore’–Using Mobiles to Improve Water Governance. In Proc. 4th Int. Conf. on M4D, Dakar (pp. 48-59).

Holla, S. and Katti, M.M., (2012). Android based mobile application development and its security. International Journal of Computer Trends and Technology, 3(3), pp.486-490.

Hope, R., Foster, T., Money, A. and Rouse, M., (2012). Harnessing mobile communications innovations for water security, Global Policy, 3(4), 2012, pp.433-442.

Hsu, Y.C. and Ching, Y.H., (2013). Mobile app design for teaching and learning: Educators’ experiences in an online graduate course. The International Review of Research in Open and Distributed Learning, 14(4).

Hu, W.C. ed., (2008). Internet-Enabled Handheld Devices, Computing, and Programming: Mobile Commerce and Personal Data Applications: Mobile Commerce and Personal Data Applications. IGI Global.

Hutchings, M.T., Dev, A., Palaniappan, M., Srinivasan, V., Ramanathan, N., Taylor, J., Ross, N. and Luu, P., (2012). mWASH: mobile phone applications for the water, sanitation, and hygiene sector. Report for Nexleaf Analytics & the Pacific Institute, pp.1-115.

Intelligence, G.S.M.A., (2016). Global mobile trends. GSMA. Retrieved September, 18, p.2017.

Jason Bell, (2017). Cross-Platform Mobile Development with Xamarin, URL: https://www.wintellect.com/wp-content/uploads/2017/03/Xamarin_Webinar_for-attendees.pdf (accessed 02 October, 2018).

Kaiser, K., (2009). Protecting respondent confidentiality in qualitative research. Qualitative health research, 19(11), pp.1632-1641.

Kamboj, V. and Gupta, H., (2012). Mobile operating systems. International Journal of Engineering Innovation & Research, 1(2), pp.115-120.

Kang, H. and Cho, J., 2015. Case research on efficient Android programming education using multi Android development tools. Indian Journal of Science and Technology, 8(19).

Kavitha, C.R. and Thomas, S.M., (2011). Requirement gathering for small projects using agile methods. IJCA Special Issue on Computational Science-New Dimensions & Perspectives, NCCSE.

KC, S., (2014). Platform Independent Connections to Internet of Things.

Kefela, G.T., (2011). The impact of mobile phone and economic growth in developing countries. African Journal of Business Management, 5(2), p.269.

Kepley, S., (2014). Rapid development of mobile apps using App Inventor and AGCO API (Doctoral dissertation, Kansas State University).

Khan, F.H. and Haris, M., (2017). Evolution of Android Operating System: A Review.

Khemry Khourn, (2016). Unlimited Free Push Notifications with OneSignal and Cordova. Url: https://medium.com/the-web-tub/unlimited-free-push-notifications-with-onesignal-and-cordova-1a415fcc6e1b (accessed 09/08/2018).

Khomba, J.K., (2011). Redesigning the Balanced Scorecard model: an African perspective. A PhD thesis. University of Pretoria, Pretoria.

Kim, H.J., Karunaratne, S., Regenbrecht, H., Warren, I. and Wunsche, B.C., (2015). Evaluation of cross-platform development tools for patient self-reporting on mobile devices. In Proc. 8th Australasian Workshop Health Inform. Knowl. Manag., Sydney, Australia (pp. 55-61).

Klimek, R. and Szwed, P., (2010). Formal analysis of use case diagrams. Computer Science, 11, pp.115-131.

Kochanska, I., (2014). OS for embedded systems-OpenEmbedded.

Kote, T., and Olmos, A. (2011). NextDrop Final Report.

Kumpel, E., Sridharan, A., Kote, T., Olmos, A. and Parikh, T.S., (2012). NextDrop: Using Human Observations to Track Water Distribution," In NSDR.

Labuschagne, L., (2012). Water monitoring 'easier' with free mobile phone app. URL: https://www.scidev.net/global/r-d/news/water-monitoring-easier-with-free-mobile-phone-app.html (accessed 04 May 2018).

Langos, S., (2014). Athens as an international tourism destination: An empirical investigation to the city's imagery and the role of local DMO's. MSc in Marketing Management, University of Derby.

Larman, C. and Basili, V.R., (2003). Iterative and incremental developments. a brief history. Computer, 36(6), pp.47-56.

Lennox-Chhugani, N., (2018). A User-Centred Design Approach to Integrated Information Systems–A Perspective. International Journal of Integrated Care, 18.

Lewis, J.R., (2001). Evaluation of procedures for adjusting problem-discovery rates estimated from small samples. International Journal of Human-Computer Interaction, 13(4), pp.445-479.

Litayem, N., Dhupia, B. and Rubab, S., (2015). Review of cross-platforms for mobile learning application development. International Journal of Advanced Computer Science and Applications, 6(1).

Mahapatra, H.B. and Goswami, B., (2015). Selection of software development methodology (SDM): a comparative approach. International Journal of Advanced Research in Computer Science and Software Engineering, 5(3).

Majchrzak, T. and Grønli, T.M., (2017). Comprehensive analysis of innovative cross-platform app development frameworks. In Proceedings of the 50th Hawaii International Conference on System Sciences.

Majchrzak, T.A., Wolf, S. and Abbassi, P., (2015). Comparing the capabilities of mobile platforms for business app development. In EuroSymposium on Systems Analysis and Design (pp. 70-88). Springer, Cham.

Maji, A.K., Hao, K., Sultana, S. and Bagchi, S., (2010). Characterizing failures in mobile oses: A case research with android and symbian. In Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on (pp. 249-258). IEEE.

Masi, E., Cantone, G., Mastrofini, M., Calavaro, G. and Subiaco, P., (2012). Mobile apps development: A framework for technology decision making. In International Conference on Mobile Computing, Applications, and Services (pp. 64-79). Springer, Berlin, Heidelberg.

Mbayen, M. and Edgard, G., (2013). A Mobile Application Development Strategy-Finding Model.

Mccabe, G.P. and Moore, D.S., (2005). Introduction to the Practice of Statistics Ise.

Meehan, D. and Sabin, M., (2013). QuizPower: a mobile app with app inventor and XAMPP service integration. In Proceedings of the 14th annual ACM SIGITE conference on Information technology education (pp. 103-108). ACM.

Mir, M.M.U.I. and Kumar, S., (2015). Evolution of Mobile Wireless Technology from OG to 5G. International Journal of Computer Science and Information Technologies, 6(3), pp.2545-2551.

Mohajan, H., (2017). Research Methodology.

Mollins, J., (2012). Mobile technology boosts access to clean water for the poor. URL: https://www.csmonitor.com/World/Making-a-difference/Change-Agent/2012/1015/Mobile-technology-boosts-access-to-clean-water-for-the-poor (Accessed 20 April 2018).

Mongi, H.J., Mvuma, A.N., Kucel, S., Tenge, A.J. and Gabriel, M., (2015). Accessibility and utilization of mobile phones for governance of water resources in the Lake Victoria Basin: Constraints and opportunities in Tanzania. African Journal of Environmental Science and Technology, 9(5), pp.438-450.

Moraa, H., Salim, A. and Otieno, A. (2013). M-Governance: Use of Technology in Water, Technology Development and Platform Enhancements for Successful Global E-Government Design, p.247.

Moreno-Ger, P., Torrente, J., Hsieh, Y.G. and Lester, W.T., (2012). Usability testing for serious games: Making informed design decisions with user data. Advances in Human-Computer Interaction, 2012, p.4.

Moroney, L., (2017). Firebase Cloud Messaging. In The Definitive Guide to Firebase (pp. 163-188). Apress, Berkeley, CA.

Ndaw, M.F. and Niyungeko, D., (2015). Unlocking the potential of information communications technology to improve water and sanitation services Liberia case research, World Bank, Water and sanitation program.

Ndaw, M.F., Mwangi, P.N., (2015). Unlocking the Potential of Information Communications Technology to Improve Water and Sanitation Services. Kenya case research.

Nguyen, H. and Shaw, R., (2011). Chapter 3 Adaptation to Droughts in Cambodia. In Droughts in Asian Monsoon Region (pp. 49-66). Emerald Group Publishing Limited.

Nielsen, J., (2012). How many test users in a usability research. Nielsen Norman Group, 4(06).

Nitze, A. and Schmietendorf, A., (2013). Cross-Platform Mobile Application Development. In User Conference for Software Quality, Test and Innovation.

Okediran, O.O., Arulogun, O.T., Ganiyu, R.A. and Oyeleye, C.A., (2014). Mobile operating systems and application development platforms: A survey. International Journal of Advanced Networking and Applications, 6(1), p.2195.

Onyenankeya, K., Caldwell, M. and Okoh, A., (2015). Water conservation and culture of indifference among college students: The nexus of descriptive norms. Journal of Human Ecology, 52(1-2), pp.15-25.

Ottka, S., (2015). Comparison of mobile application development tools for multi-platform industrial applications.

Otuke, J.O., (2016). Role of information communication technologies in water management (doctoral dissertation, University of Nairobi).

OutSystems, (2018). How to Use Push Notifications with OneSignal. Url: https://success.outsystems.com/Documentation/Development_FAQs/How_to_Use_Push_Notifications_with_OneSignal (accessed 09/08/2018).

Palmieri, M., Singh, I. and Cicchetti, A., (2012). Comparison of cross-platform mobile development tools. In Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on (pp. 179-186). IEEE.

Pawar, A.P., Jagtap, V.S. and Bhamare, M.S., (2014). Survey on Techniques for Cross Platform Mobile Application Development. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 3(10), p.3554.

Pazirandeh, A. and Vorobyeva, E., (2015). Evaluation of cross-platform tools for mobile development.

Peppers, J., Taskos, G. and Bilgin, C., (2016). Xamarin: Cross-Platform Mobile Application Development. Packt Publishing Ltd.

Perdikuri, K., (2014). Students' Experiences from the use of MIT App Inventor in classroom. In Proceedings of the 18th Panhellenic conference on informatics (pp. 1-6). ACM.

Pérez, N., (2009). Research methodology: An example in a real project. Laboratory of Optics and Experimental Mechanics, Instituto de Engenharia Mecânica eGestão Industrial.

Pete Cranston (2010). The potential of mobile applications for positive social and economic change in rural communities.

Petzold, C., (2015). Creating Mobile Apps with Xamarin. Forms Preview Edition 2. Microsoft Press.

Pohl, K., (2010). Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated.

Pokress, S.C. and Veiga, J.J.D., (2013). MIT App Inventor: Enabling personal mobile computing. arXiv preprint arXiv:1310.2830.

Priyanga, P., (2013). Methods of Data Collection. URL: https://www.slideshare.net/priyansakthi/methods-of-data-collection-16037781 (accessed 30 May 2018).

Radi, A.A., (2016). Evaluation of Xamarin Forms for MultiPlatform Mobile Application Development.

Rai, P. and Dhir, S., (2014). Impact of different methodologies in software development process. International Journal of Computer Science and Information Technologies, 5(2), pp.1112-1116.

Randolph, J.J., Gaiek, L.S., White, T.A., Slappey, L.A., Chastain, A., Prejean-Harris, R. and Hansard, C., (2012). A Quantitative Content Analysis of Mercer University Theses. Georgia Educational Researcher, 9(1), p.81.

Redda, Y.A., (2012). Cross platform Mobile Applications Development: Mobile Apps Mobility (Master's thesis, Institutt for datateknikk og informasjonsvitenskap).

Redling, A., (2013). WaterQuality app makes tracking data easy and educational. URL: http://www.fondriest.com/news/waterquality-app-makes-tracking-data-easy-and-educational.htm (accessed 04 May 2018).

Rein, P., Champanis, M. and Rivett, U., (2013). December. Drop drop: prototyping a mobile application educating on the water system through private meter readings. In Proceedings of the Sixth International Conference on Information and Communications Technologies and Development: Notes-Volume 2, pp. 124-127.

Renner, T., (2014). Mobile OS-Features, Concepts and Challenges for Enterprise Environments. SNET Project Technische Universitat Berlin.

Ribeiro, A. and da Silva, A.R., (2012). Survey on cross-platforms and languages for mobile apps. In Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the (pp. 255-260). Ieee.

Rivett, U., Champanis, M. and Wilson-Jones, T., (2013). Monitoring drinking water quality in South Africa: Designing information systems for local needs, Water SA, 39(3), pp.409-414.

Rubin, J. and Chisnell, D., (2008). Handbook of usability testing: how to plan, design and conduct effective tests. John Wiley & Sons.

Sánchez Blanco, A., (2016). Development of hybrid mobile apps: Using Ionic Framework.

Sencha, (2018). Intro to Applications with Sencha Touch, URL: https://docs.sencha.com/touch/2.4/core_concepts/about_applications.html (accessed 01/10/2018).

Severance, C.R., Blumenberg, S. and Hauser, E., (2016). Python for Everybody: Exploring Data in Python 3. CreateSpace Independent Publishing Platform.

Sharon, M., (2008). An introduction to mobile technologies and services. Socialight. URL http://uberthings. com/mobile/intro_to_mobile. Pdf (accessed 12 July 2017).

Shaw, R., (2011). Information and Communications Technologies (ICT) and Water in the Face of Climate Change: Issues and Research Priorities in Asia.

Singh, R., (2014). An Overview of Android Operating System and Its Security. Int. Journal of Engineering Research and Applications, 4(2), pp.519-521.

Smutný, P., (2012). Mobile development tools and cross-platform solutions. In Carpathian Control Conference (ICCC), 2012 13th International (pp. 653-656). IEEE.

Soares, A., (2014). Reflections on teaching app inventor for non-beginner programmers: issues, challenges and opportunities. Information Systems Education Journal, 12(4), p.56.

Sommer, A. and Krusche, S., (2013). Evaluation of Cross-Platform Frameworks for Mobile Applications. In Software Engineering (Workshops) (pp. 363-376).

Steczko, J., (2016). Analysis of companies' experience with cross-platform development compared to native development for mobile devices.

Stolzle S., (2018). Build Tizen HTML5 Apps w/ Sencha Architect & Sencha Touch, URL: https://download.tizen.org/misc/media/tds2013/slides/Learn-how-to-Build-Tizen-HTML5-Apps-with-Sencha-Architect-&-Sencha-Touch.pdf (accessed 02/10/2018).

Suterwala, A., (2010). Creating an Android "Hello World" Application With PhoneGap, URL: https://code.tutsplus.com/tutorials/creating-an-android-hello-world-application-with-phonegap--mobile-2532 (Accessed 12 October 2018)

Tarkoma, S., (2009). Mobile middleware: supporting applications and services. John Wiley & Sons.

Tashakkori, A. and Creswell, J.W., (2007). Exploring the nature of research questions in mixed methods research.

Taylor, K.H., Takeuchi, L. and Stevens, R., (2018). Mapping the daily media round: novel methods for understanding families' mobile technology use. Learning, Media and Technology, 43(1), pp.70-84.

Tengroth, W., (2016). Cross-Platform Development: Bringing a desktop web application to the mobile platform.

Trivedi, K., R., (2012). Mobile Application Development using App Inventor for Android Devices, URL: http://indico.ictp.it/event/a11164/session/25/contribution/19/material/0/0.pdf (accessed 04/09/2018).

Tsega, H., Lemmens, R., Lungo, J. and Kraak, M.J., (2013).Urban Context Modelling for Human Sensor Web, In Proceedings of the N-AERUS XIV workshop, Enschede, The Netherlands, pp. 12-14.

UN News Centre, (2013). Deputy UN chief calls for urgent action to tackle global sanitation crisis. URL:http://www.un.org/apps/news/story.asp?NewsID=44452&Cr=sanitation&Cr1=#.UU7Y2oaCmji (Accessed 30 April 2018)

Van Lamsweerde, A., (2009). Requirements engineering: From system goals to UML models to software (Vol. 10). Chichester, UK: John Wiley & Sons.

Vennapoosa, C., (2012). The Evolutionary Prototyping Model. URL: http://www.exforsys.com/career-center/project-management-life-cycle/the-evolutionary-prototyping-model.html (accessed 13 October 2018).

Vosloo, J.J., (2014). A sport management programme for educator training in accordance with the diverse needs of South African schools (Doctoral dissertation).

Wade, R.H. (2004). Bridging the Digital Divide: New Route to Development or New Form of Dependency, In: Avgerou, C. Ciborra, C. and Land, F. (eds), The Social Research of Information and Communication Technology Innovation, Actors, and Context. 185-206. New York: Oxford University Press.

Webinos, (2018). Xamarin – MonoTouch and Mono for Android, URL: http://webinos.org/crossplatformtools/xamarin-monotouch-and-mono-for-android/ (accessed 02/10/2018)

Welle, K., Williams, J. and Pearce, J., (2016). ICTs Help Citizens Voice Concerns over Water–Or Do They?, IDS Bulletin, 47(1).

Wisdom, J. and Creswell, J.W., (2013). Mixed methods: integrating quantitative and qualitative data collection and analysis while studying patient-centered medical home models. Rockville: Agency for Healthcare Research and Quality.

Wolber, D., Abelson, H., Spertus, E. and Looney, L., (2011). App Inventor. " O'Reilly Media, Inc.".

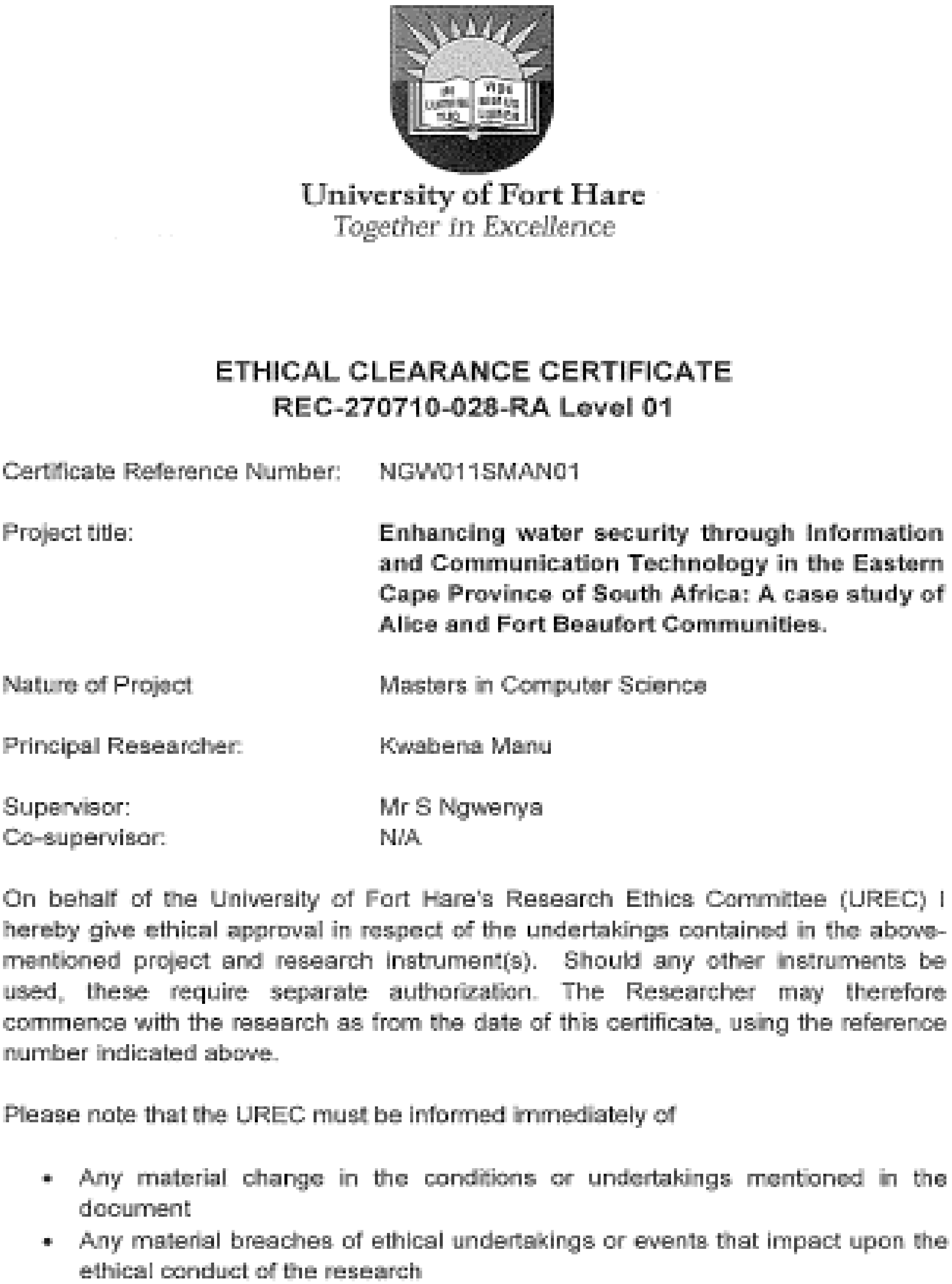
Wolber, D., Abelson, H., Spertus, E. and Looney, L., (2011). App Inventor. " O'Reilly Media, Inc.".

Xanthopoulos, S. and Xinogalos, S., (2013). A comparative analysis of cross-platform development approaches for mobile applications. In Proceedings of the 6th Balkan Conference in Informatics (pp. 213-220). ACM.

Zilpe, M.V. and Chatur, P.N., (2012). iNavigate-An Android Based Navigation Application. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), 1(4), pp.pp-89.

LIST OF APPENDICES

APPENDIX 1: ETHICAL CLEARANCE CERTIFICATE



University of Fort Hare
Together in Excellence

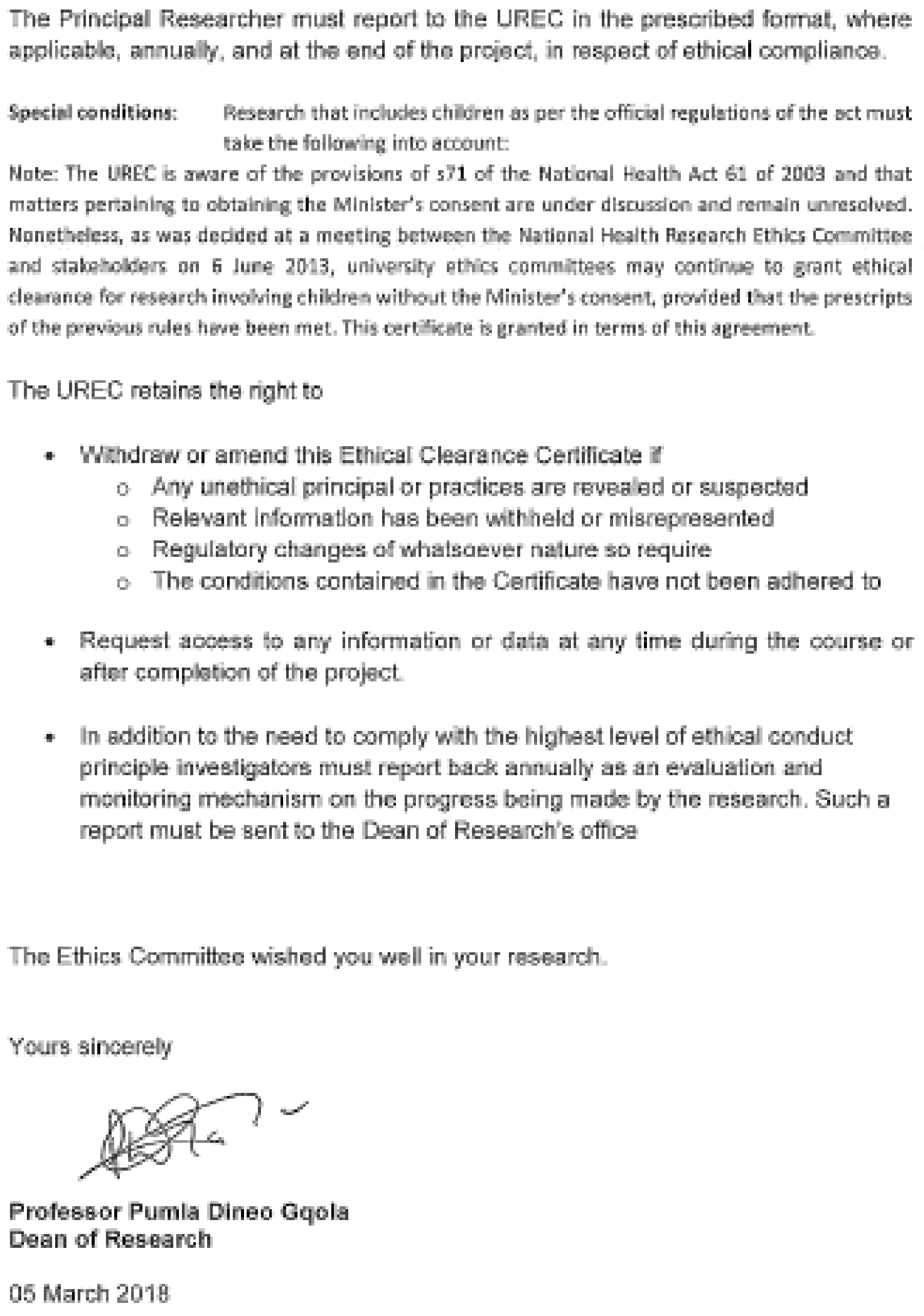
ETHICAL CLEARANCE CERTIFICATE
REC-270710-028-RA Level 01

Certificate Reference Number:	NGW011SMAN01
Project title:	**Enhancing water security through Information and Communication Technology in the Eastern Cape Province of South Africa: A case study of Alice and Fort Beaufort Communities.**
Nature of Project	Masters in Computer Science
Principal Researcher:	Kwabena Manu
Supervisor:	Mr S Ngwenya
Co-supervisor:	N/A

On behalf of the University of Fort Hare's Research Ethics Committee (UREC) I hereby give ethical approval in respect of the undertakings contained in the above-mentioned project and research instrument(s). Should any other instruments be used, these require separate authorization. The Researcher may therefore commence with the research as from the date of this certificate, using the reference number indicated above.

Please note that the UREC must be informed immediately of

- Any material change in the conditions or undertakings mentioned in the document
- Any material breaches of ethical undertakings or events that impact upon the ethical conduct of the research



The Principal Researcher must report to the UREC in the prescribed format, where applicable, annually, and at the end of the project, in respect of ethical compliance.

Special conditions: Research that includes children as per the official regulations of the act must take the following into account:

Note: The UREC is aware of the provisions of s71 of the National Health Act 61 of 2003 and that matters pertaining to obtaining the Minister's consent are under discussion and remain unresolved. Nonetheless, as was decided at a meeting between the National Health Research Ethics Committee and stakeholders on 6 June 2013, university ethics committees may continue to grant ethical clearance for research involving children without the Minister's consent, provided that the prescripts of the previous rules have been met. This certificate is granted in terms of this agreement.

The UREC retains the right to

- Withdraw or amend this Ethical Clearance Certificate if
  - Any unethical principal or practices are revealed or suspected
  - Relevant information has been withheld or misrepresented
  - Regulatory changes of whatsoever nature so require
  - The conditions contained in the Certificate have not been adhered to

- Request access to any information or data at any time during the course or after completion of the project.

- In addition to the need to comply with the highest level of ethical conduct principle investigators must report back annually as an evaluation and monitoring mechanism on the progress being made by the research. Such a report must be sent to the Dean of Research's office

The Ethics Committee wished you well in your research.

Yours sincerely

Professor Pumla Dineo Gqola
Dean of Research

05 March 2018

APPENDIX 2 : SURVEY QUESTIONNAIRE

Please take the time to answer questions for this survey about water security around the house and community.

Kindly note that all your response will be kept in confidential and your information will not be seen by anyone outside this project.

Great appreciation will be given to your open and honest response towards answering these questions.

This questionnaire will approximately take you 10 minutes to finish.

Please select your answers by indicating with **X**

Examples 1

	X			
Option 1	Option 2	Option 3	Option 4	Option 5

		X
Yes	No	Other

QUESTION ON WATER CONSERVATION

1. It is expected of me that I save water around the house and community.

Strongly disagree	Disagree	Neither agree or disgrace	Agree	Strongly agree

2. I would feel guilty if I didn’t save water around the house and community.

Strongly disagree	Disagree	Neither agree or disgrace	Agree	Strongly agree

3. There is agreement between the members of my household that engaging in everyday actions to save water around the house and community is a good thing to do

Strongly disagree	Disagree	Neither agree nor	agree	Strongly agree

		disagree		

4. How often do you check and fix leaking taps?

Never	Rarely	Sometimes	Almost always	Always

5. How often do you collect rain water to use on backyard garden?

Never	Rarely	Sometimes	Almost always	Always

6. How often do you turn off taps when brushing teeth?

Never	Rarely	Sometimes	Almost always	Always

7. How often do you use minimal water in kitchen (for cooking, washing up, rinsing)?

Never	Rarely	Sometimes	Almost always	Always

8. Have you ever noticed any public information program on water conservation?

Yes	No

QUESTIONS ON WATER DRINKING HABITS

9. What kind of water do you usually take at home?

Tap water	Bottled water	Filtrated water

10. What kind of water do you usually take at work?

Tap water	Bottled water	Filtrated water

11. How would you rate the existing water supply service?

Very good	Good	Fair	Poor	Do not know

QUESTION ON WATER LEAKS IN STREET/TOWN

12. Do you know Department/organisation to report street/town water leakage to?

Yes	No

13. Do you know the number to call to report a water leakage in the street/town?

I know	Do not know

14. Have you ever reported water leakage in the street/town before?

Yes	No

QUESTIONS ABOUT WATER RATES

15. Do you know how much you pay for each kilolitre of water?

Yes	No	Other :Specify________

16. What do you think about the current water rate?

Too high	Normal	Too Low	Do not know

17. When you compare the current water rate with other utility payments (example electricity), what do you think about the current water tariff?

Too high	Normal	Too Low	Do not know

QUESTION ON PHONES

18. Do you own a phone (eg. tablet, smartphone)?

Yes	No

19. How many mobile phones do you have, including both personal and company?

One	Two	Three	Other

20. Which of the following devices do you own?

iPhone	Android Phone	Windows Phone	Blackberry	Symbian Phone (Nokia)	Other

21. Does your phone have a camera?

Yes	No

22. What category of apps do you use and enjoy using most?

Social Networks	Games	Other

23. Have you ever downloaded apps for your device?

Yes Always free	Yes Always Paid	Yes Free & Paid	No

24. Can your phone access internet?

Yes	No

The End

Thank you for taking time to participate in this important research

APPENDIX 3: SURVEY ANALYSIS AND DISCUSSION

INTRODUCTION

To complete this research properly, it is important to analyse the data gathered in order to test the hypothesis and answer the research questions and attain the specified research objectives. An appendix 3 presents data from responses of the various participants in the Alice and Fort Beaufort communities. The data was collected through a series of interviews including observations and questionnaires that were administered. This section comprises analysis, presentation, interpretation and discussion of the findings resulting from the survey.

The analysis and interpretation of data is carried out in three phases. The first part is based on the results of the questionnaire in Alice community. The second, which is based on the results from the survey questionnaire in Fort Beaufort community and the last part, compares observational survey results from both Alice and Fort Beaufort. The results from this survey will help to get more insight from these two communities in order to come up with appropriate mobile application to manage some water security issues in the area.

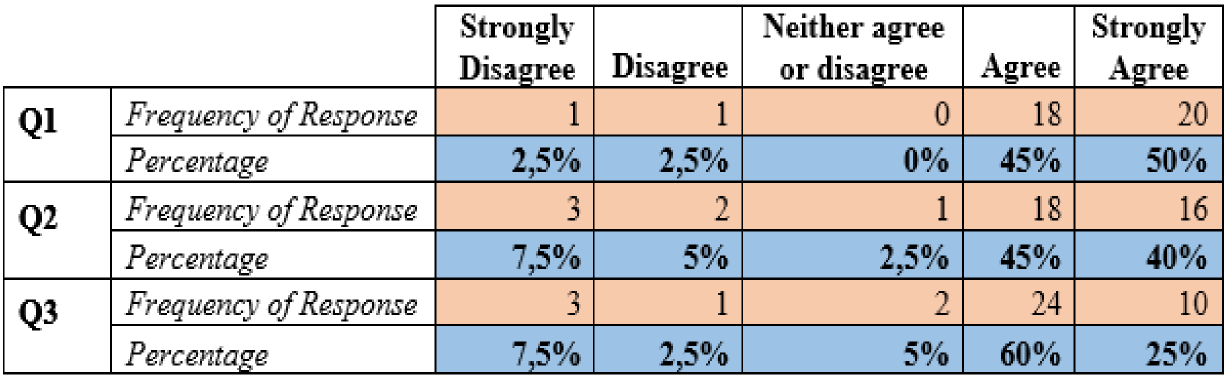
SURVEY IN ALICE COMMUNITY

The objectives of the survey were to understand the communities' perceptions and attitude towards water conservation practices within Alice community. A series of questions were developed which were aimed at understanding the public's knowledge and willingness to implement water conservation measures in the following areas: water conservation measures in the house and within the community, attitudes towards water conservation and water conservation awareness. Water drinking habits, water leakages in street or town, opinion on water rates and lastly, respondents' information on their mobile phones are been analysed and discussed.

WATER CONSERVATION MEASURES IN THE HOUSEHOLD AND COMMUNITY

Respondents were asked if it is expected of them to save water around the house and in the Alice community. Of the total number (50) of respondents that were interviewed, 95% either agreed or strongly agreed that they are expected to save water in their houses and communities as shown in Table 6.1. Out of this 95%, 50% strongly agreed for question (**Q1**). Only a total 5% of the respondents disagree (2.5%) and strongly disagrees (2.5%). In addition, participants were asked whether they would feel guilty if they did not save water in their various households and in the community at large. 45 % of the respondents agreed that they would feel guilty while 40% strongly agreed, as indicated in table 7.1 for question two (**Q2**). 5% disagree, 7.5% strongly disagree and 2.5% of the participants neither agree nor disagree.

Respondents were asked whether agreement between members of their household to engage in every day actions to save water in the house and community is a good thing to do. 60% of the participants agreed that it is a good thing to do, 25% strongly agreed it is an appropriate measure or practice for water conservation, shown in table 7.1 for question three (**Q3**). 2.5% respondents disagree, 7.5% strongly disagree and 5% neither agree nor disagree.



		Strongly Disagree	Disagree	Neither agree or disagree	Agree	Strongly Agree
Q1	*Frequency of Response*	1	1	0	18	20
	Percentage	**2,5%**	**2,5%**	**0%**	**45%**	**50%**
Q2	*Frequency of Response*	3	2	1	18	16
	Percentage	**7,5%**	**5%**	**2,5%**	**45%**	**40%**
Q3	*Frequency of Response*	3	1	2	24	10
	Percentage	**7,5%**	**2,5%**	**5%**	**60%**	**25%**

Table 7.1 – Water Conservation Measures in the house and in community (Alice)

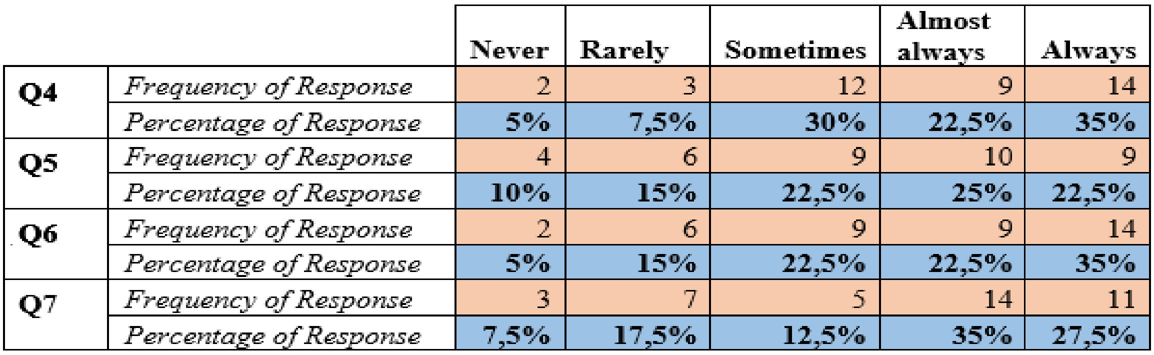
ATTITUDES TO WATER CONSERVATION IN ALICE

Respondents were asked about their attitudes towards leaking water taps at home, 35% of the participants always check and fix their leaking taps whereas 30% does it sometimes as shown in table 4.2 for question four (**Q4**). While 22.5% almost always check and fix leaking tap, 5% never checks and 7.5% rarely check and fix. Also, a question was asked about how often they collect rain water to use on their backyard garden. 25% indicated almost always, while 22.5% indicated always, 22.5% reported that they do it sometimes, 15% rarely use rain water for their garden and 10% never use rain water at all, as shown table 7.2 for question five (**Q5**).

In addition, participants were asked about how often they turn their taps off when brushing teeth. Of the 50 participants that answered the questionnaire, 35% of the respondents indicated that they turn off their taps when brushing teeth, 22.5% indicated that they ‘almost always’ turn off taps when brushing teeth as shown in Table 7.2 for question six **(Q6)**. 22.5%

also indicated they sometimes turn off their taps when brushing their teeth while 15% rarely turn off and 5% never turn off taps when brushing their teeth.

A question was asked about, how often respondents use minimal water in their kitchen. Majority of the respondents reported that they use minimal water in their kitchen, which is 27.5% always use minimal water and 35% almost always use minimal water in the kitchen, as indicated in table 7.2 for question seven (**Q7**). 12.5% respondents indicated they sometimes use minimal water in the kitchen whereas 17.5% rarely use minimal water and 7.5% indicated they never use minimal water in the kitchen.



		Never	Rarely	Sometimes	Almost always	Always
Q4	*Frequency of Response*	2	3	12	9	14
	Percentage of Response	**5%**	**7,5%**	**30%**	**22,5%**	**35%**
Q5	*Frequency of Response*	4	6	9	10	9
	Percentage of Response	**10%**	**15%**	**22,5%**	**25%**	**22,5%**
Q6	*Frequency of Response*	2	6	9	9	14
	Percentage of Response	**5%**	**15%**	**22,5%**	**22,5%**	**35%**
Q7	*Frequency of Response*	3	7	5	14	11
	Percentage of Response	**7,5%**	**17,5%**	**12,5%**	**35%**	**27,5%**

Table 7.2 – Attitudes to Water Conservation

WATER CONSERVATION AWARENESS

Respondents were asked whether they have noticed any public information program on water conservation (**Q8**). Majority of the respondents (85%) that were not aware of any public information on water conservation. Only 15% noticed public information on water conservation as shown Figure 7.1 below.

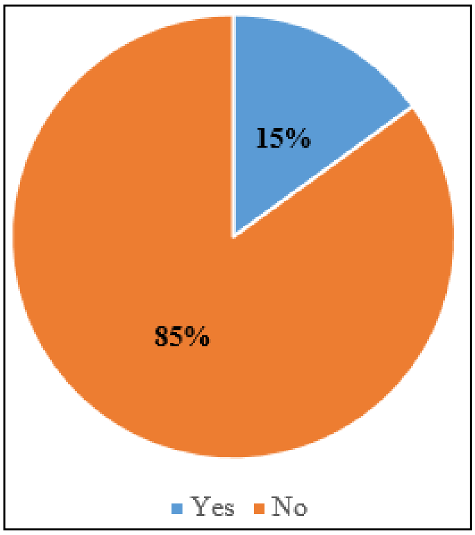


Figure 7.1 – Water Conservation Awareness

DRINKING WATER

Respondents were asked if they are more likely to drink water directly from the tap, drink bottled water or drink filtrated tap water at home or at work. Out of the 50 respondents, 77.5% indicated they preferred drinking water from the tap at home as shown in Figure 7.2 for question nine **(Q9**). Also, 65% of the respondent preferred drinking tap water at work and as many as 32.5% preferred drinking bottled water at work as shown below in Figure 7.2 for question 10 **(Q10).**

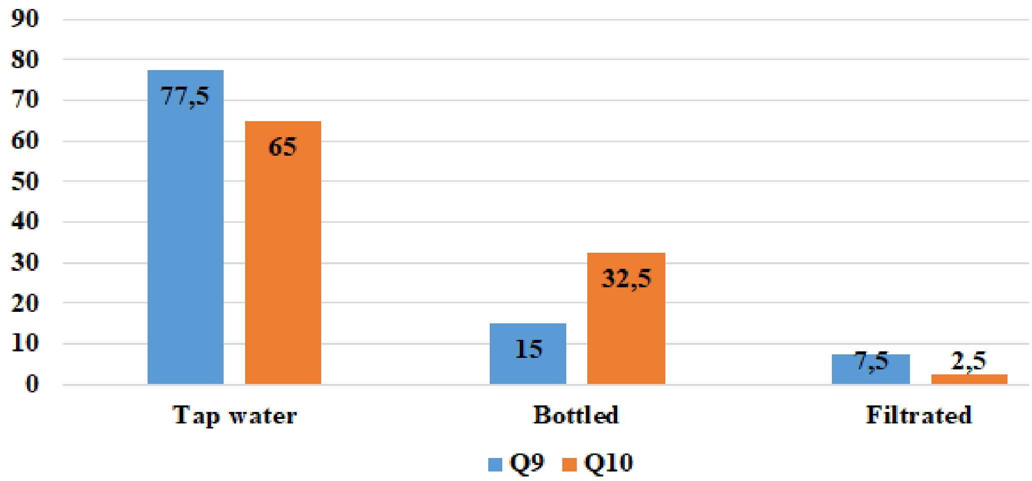


Figure 7.2 – Water Drinking Habits at Home and at Work

Respondents were asked about their perceptions on current water supply service in their community **(Q11).** 35% of the respondents indicated that the water supply service they receive was fair, 27% reported that it was good service, 25% reveal it was very good and 13% of the respondents rating the water supply service as poor, as shown in Figure 7.3.

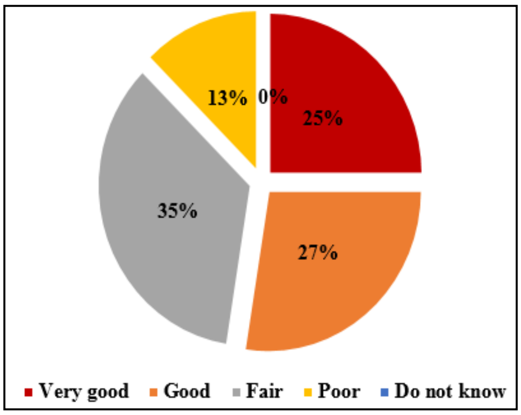


Figure 7.3 – Perceptions on Water Supply Service in Alice Community

WATER LEAKAGES IN STREET/TOWN

Respondents were asked if they know the organisation or authoritative body to report water leakages to in town (**Q12**). 77.5% of the respondents indicated that they didn't know, while only 22.5% reported that they knew the authoritative body to report water leakage to in town as shown in Figure 7.4.

In addition, respondents from Alice community were asked whether or not they have ever reported water leakage in the street or town to the local water authority **(Q13).** Figure 7.4 shows that the majority of respondents (85%) have never reported water leakage, while only 15% have reported water leakage from town before to the local water authority.

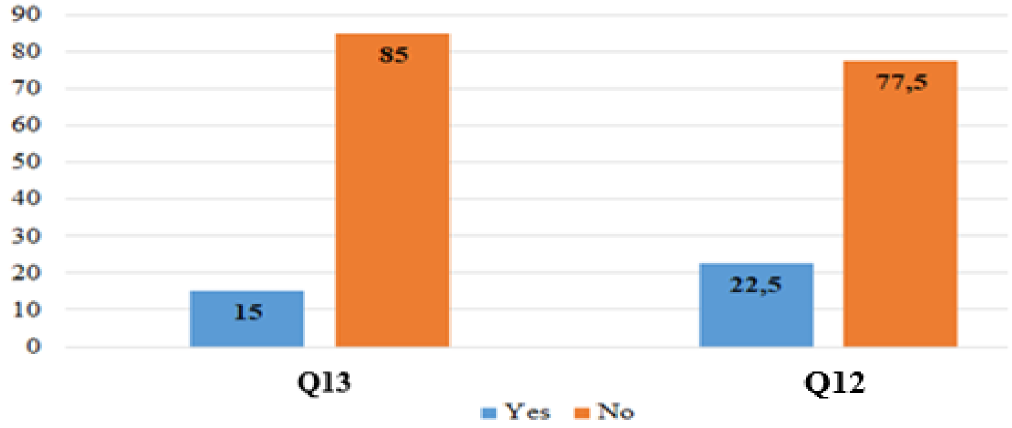


Figure 7.4 – Street Water Leakage

Respondents in Alice were asked to indicate whether or not they knew the contact number for the local water authority if there is water leakage or pipe burst in the street or town **(Q14)**. Findings are summarised in figure 7.5. Most of respondents (93%) reported that they do not know the contact number to call to report water leakages in the street or in town, if such problem occurred. Only 7% knew the number to call when the problem arises

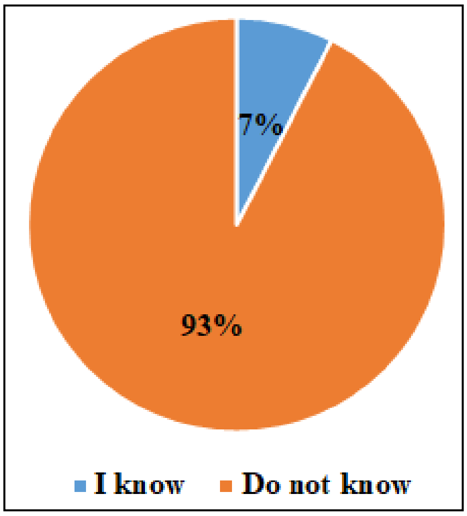


Figure 7.5 – Knowledge of Local water authority contact number

WATER RATES

Respondents were asked if they know how much they pay for 1 kilolitre (*1000 litre*) of water consumed (**Q15**). The findings showed that 80% of the participants do not know how much they pay per kilolitre of water and on other hand only 20% indicated they knew the amount as shown in Figure 7.6 below.

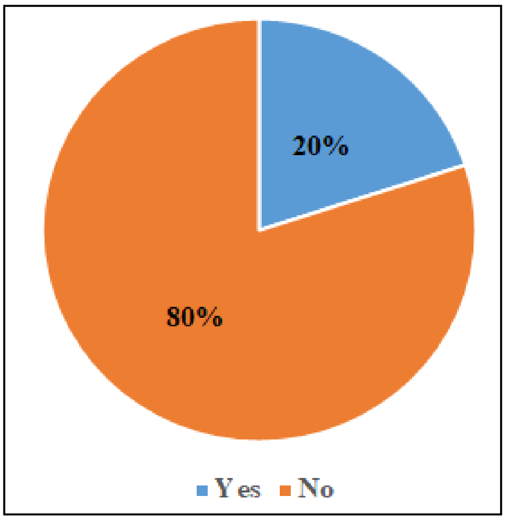


Figure 7.6 – Knowledge of Rate of 1 kilolitre of water

Respondents from Alice community were asked to rate the price of water supplied by their water authority. Figure 7.7 **(Q16)** is a graphical representation of the respondents' rating for the price of water services. More than half of the total sample (57.5%) believed that they did not know, while 25% said that it was normal. 12.5% said it was too high and only 5%

believed that it was too low. Again, when asked to give a comparison of the current water rates and other utility payments rate (example electricity) **(Q17)**, half of the respondents 50% said they did not know, 40% said it was normal when compared to other utility rates, while 5% believed it was too high and 5% said it too low.

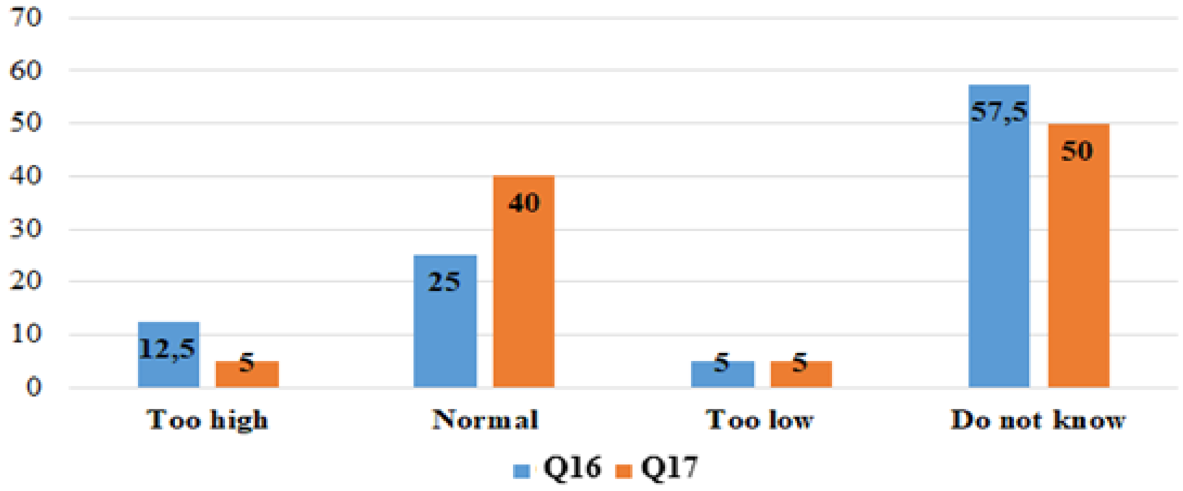


Figure 7.7 – Opinion on water rate

SURVEY ON MOBILE PHONES

Respondents were asked if they own a mobile phone **(Q18)**. 80% of the respondents said yes they own at least one mobile phone while only 20% do not own a mobile phone as indicated in Figure 7.8 below.

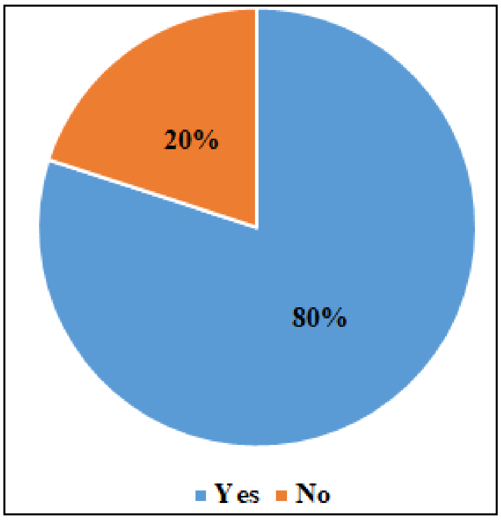


Figure 7.8 – Owners of Mobile Phone

Respondents were asked how many mobile phones they have, which includes both personal and business mobiles **(Q19)**. Majority (42%) of the respondents have at least one mobile phone, 28% indicated they have 2 mobile phones, 10% said they have 3 mobile phones and 20% indicated other, as shown in Figure 7.9.

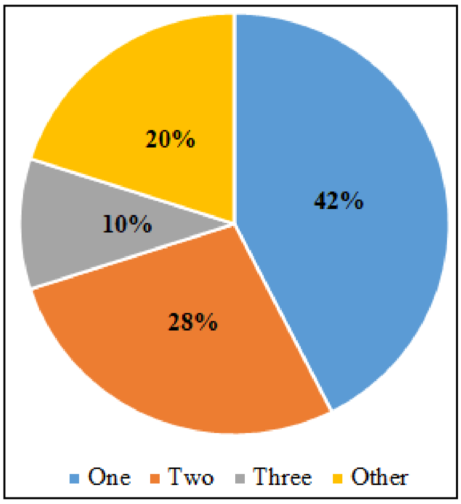


Figure 7.9 – Number of Mobile Phones Owned by Respondent

Question 20 focused on the type of mobile devices the respondents used. Most of the respondents (57.5%) own Android phones. 12.5% owns Windows phone, likewise 12.5% indicated they own other kind of mobile phone devices. 10% said they own Symbian (Nokia) phone. On the other hand, only 7.5% own iPhone type with none owning blackberry phone, as illustrated in Figure 7.10.

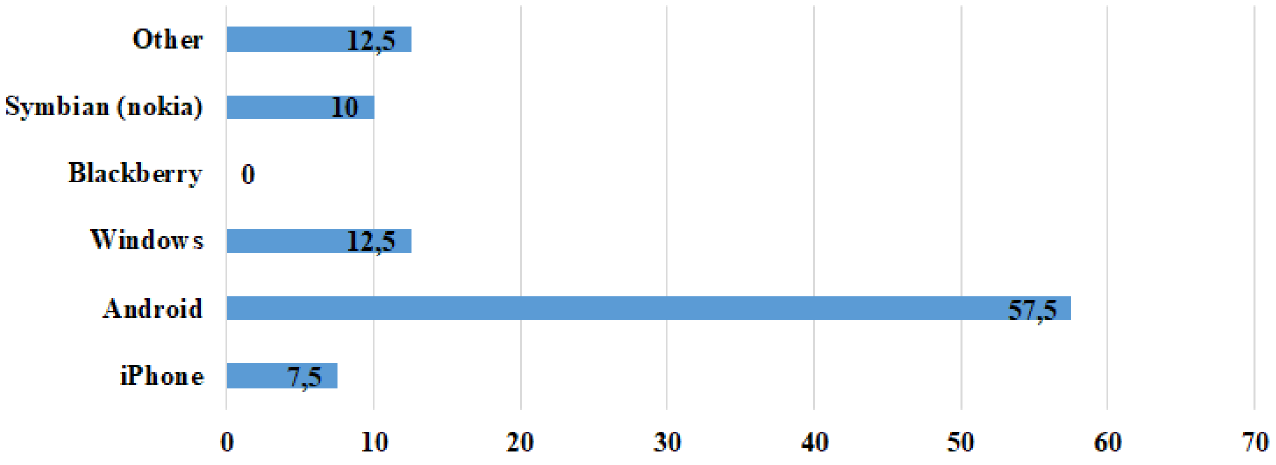


Figure 7.10 – Mobile Phone Type

Respondents were asked if their phone has a camera function (**Q21**). 75% indicated yes, while 25% indicated No, as shown in figure 7.11.

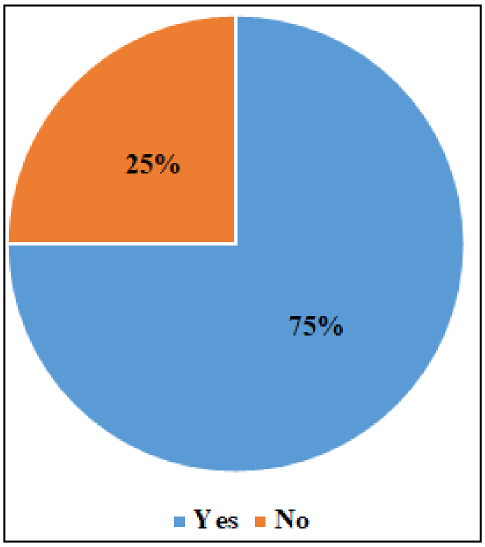


Figure 7.11 – Mobile Phones with Camera Functionality

Respondents were asked which category of applications do they use and enjoy using **(Q22)**. 55% indicated they enjoy using social networks while 20% said they enjoying game applications and 25% indicated they use other applications, shown in Figure 7.12 below.

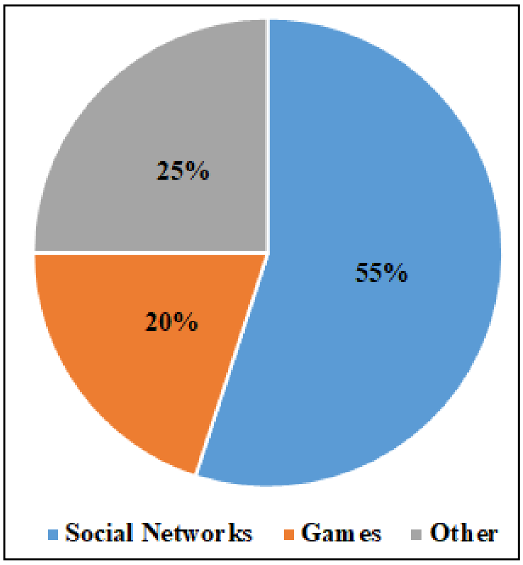


Figure 7.12 – Respondents Category of Mobile Phone Applications in Alice Community

Respondents were asked if they have ever downloaded application for their mobile device before (Q**23**). 45% indicated yes they download both free and paid applications, 17.5% indicated they always download paid application. On the other hand, 12.5% said they always download applications that are free and 25% indicated they have never downloaded application before, as shown in Figure 7.13.

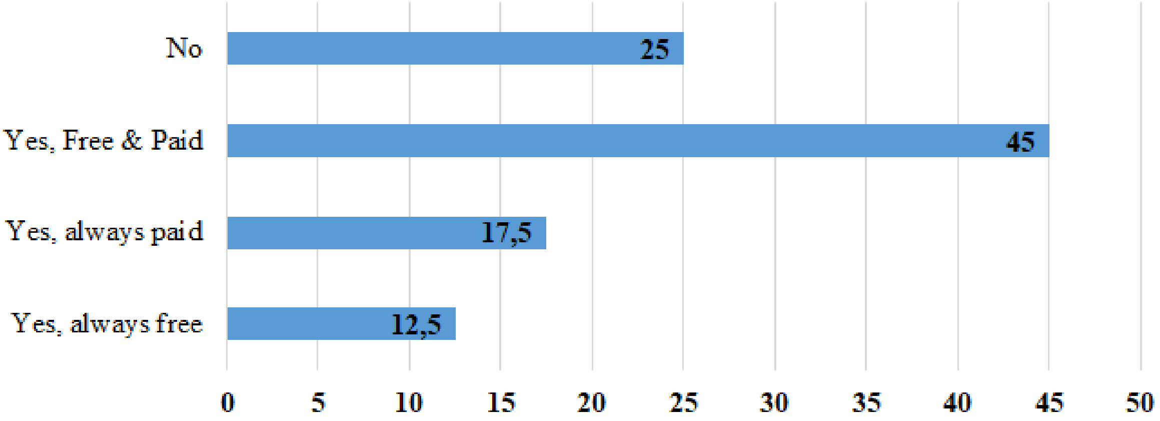


Figure 7.13 – Application Downloads in Alice Community

Respondents were asked if their mobile phone can have access to internet (**24**). 75% indicated yes while 25% indicated no, as shown in Figure 7.14.

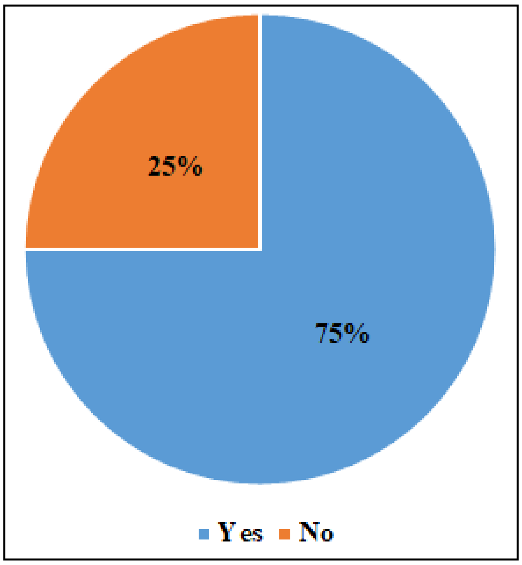


Figure 7.14 – Internet Accessibility

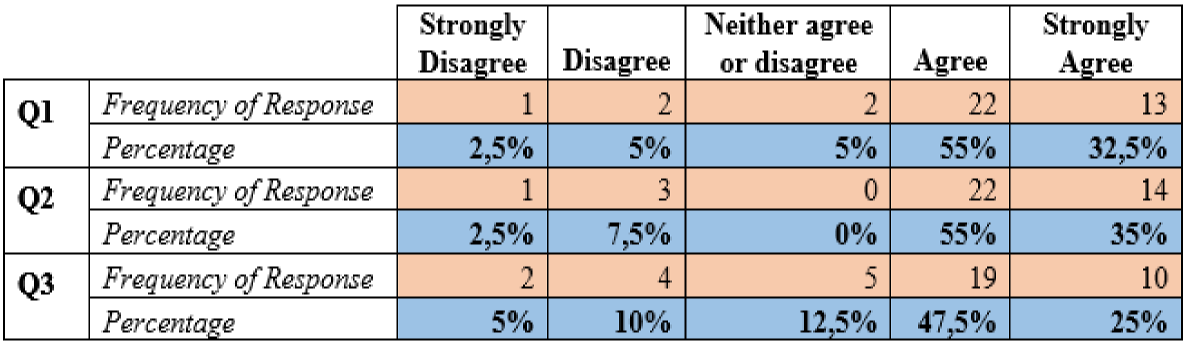
SURVEY IN FORT BEAUFORT COMMUNITY

WATER CONSERVATION MEASURES IN THE HOUSEHOLD AND COMMUNITY

Respondents were asked if they were expected to save water around the house and in the Fort Community. Majority of the participants agree or strongly agree, with 55% agreeing and 32.5% strongly agrees that is expected of them to save water in the house and in the community as appeared in table 8.1 for question one (**Q1**). 5% of the respondents disagree nor disagree while 5% disagree and 2.5% strongly disagree.

In addition, participants were asked whether they would feel guilty or regretful if they did not save water in their various households and in the community at large. 55 % of the respondents agreed that they would feel guilty while 35% strongly agreed, as showed in table 8.1 for question two (**Q2**). 7.5% disagree, 2.5% strongly disagree and none of the participants neither agree nor disagree.

Respondents were asked whether agreement between members of their household to engage in every day actions to save water in the house and community is a good thing to do. 47.5% of the participants agreed that it is a good thing to do, 25% strongly agreed it is an appropriate measure or practice for water conservation, as shown in table 8.1 for question three (**Q3**). 10% respondents disagree, 5% strongly disagree and 12.5% neither agree nor disagree.



		Strongly Disagree	Disagree	Neither agree or disagree	Agree	Strongly Agree
Q1	*Frequency of Response*	1	2	2	22	13
	Percentage	**2,5%**	**5%**	**5%**	**55%**	**32,5%**
Q2	*Frequency of Response*	1	3	0	22	14
	Percentage	**2,5%**	**7,5%**	**0%**	**55%**	**35%**
Q3	*Frequency of Response*	2	4	5	19	10
	Percentage	**5%**	**10%**	**12,5%**	**47,5%**	**25%**

Table 8.1 – Water Conservation Measures in the House and Community (Fort Beaufort)

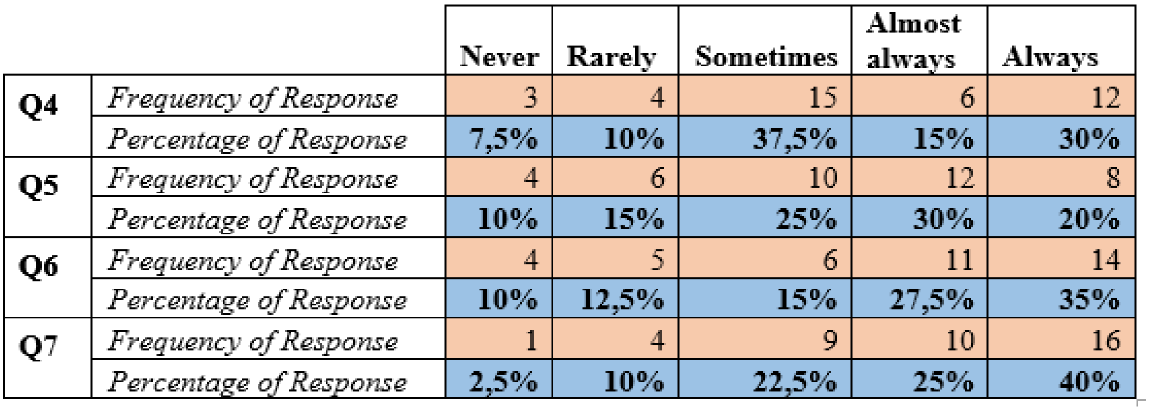
ATTITUDES TO WATER CONSERVATION IN FORT BEAUFORT

Respondents were asked about their attitudes towards leaking water taps at home, 30% of the participants always check and fix leaking taps whereas 37.5% does it sometimes as showed in table 8.2 for question four (**Q4**). 15% almost always check and fix leaking tap, 7.5% never checks and 10% rarely check and fix.

Also, question was asked on how frequently they gather rain water to use in their backyard garden. 30% indicated almost always while 20% indicated always, 25% reported that they do it sometimes, 15% rarely use rain water for their garden and 10% never use rain water at all, as shown in table 8.2 for question five (**Q5**).

In addition, participants were asked about how often they turn their taps off when brushing teeth. 35% of the respondents showed they always turn off their taps when brushing, 27.5% indicated that they do that almost always as shown in table 8.2 below for question six **(Q6)**. 15% also indicated they sometimes turn off their taps when brushing their teeth while 12.5% rarely turn off and 10% never turn off taps when brushing their teeth.

An inquiry was gotten on how frequently respondents utilize minimal water in their kitchen. Majority of the respondents reported that they use minimal water in their kitchen, which is 40% always use minimal water and 25% almost always use minimal water in the kitchen, as indicated in table 8.2 below for question seven **(Q7)**. 22.5% respondents indicated they sometimes use minimal water in the kitchen whereas 10% rarely use minimal water and 2.5% indicated they never use minimal water in the kitchen.



		Never	Rarely	Sometimes	Almost always	Always
Q4	*Frequency of Response*	3	4	15	6	12
	Percentage of Response	**7,5%**	**10%**	**37,5%**	**15%**	**30%**
Q5	*Frequency of Response*	4	6	10	12	8
	Percentage of Response	**10%**	**15%**	**25%**	**30%**	**20%**
Q6	*Frequency of Response*	4	5	6	11	14
	Percentage of Response	**10%**	**12,5%**	**15%**	**27,5%**	**35%**
Q7	*Frequency of Response*	1	4	9	10	16
	Percentage of Response	**2,5%**	**10%**	**22,5%**	**25%**	**40%**

Table 8.2 – Attitudes to Water Conservation in Fort Beaufort Community

WATER CONSERVATION AWARENESS

Respondents were inquired whether they have taken note of any public information program on water conservation (**Q8**). Over half of the respondents (55%) that were interviewed in Fort Beaufort community was not aware of any public information on water conservation while 45% were mindful of public information on water conservation as appeared in Figure 8.1.

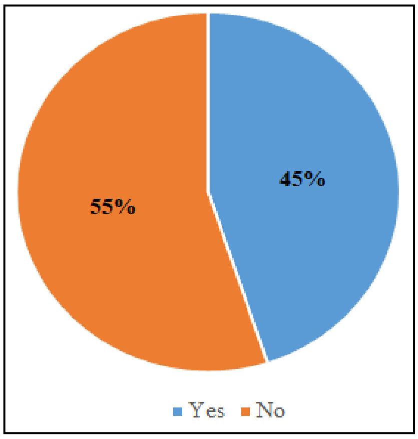


Figure 8.1 – Water Conservation Awareness

DRINKING WATER

Respondents were asked if they are more likely to drink water directly from the tap, drink bottled water or drink filtrated tap water at home or at work. Their preferred drinking water source at home was from the tap (55%). 20% prefer drinking bottled water and 25% favoured filtrated water as shown in Figure 8.2 for question **(Q9)**. Moreover, 50% of the respondent preferred drinking tap water at work and as many as 30% favoured drinking bottled water at work and 20% prefer Filtrated water as shown below in Figure 8.2 for question **(Q10).**

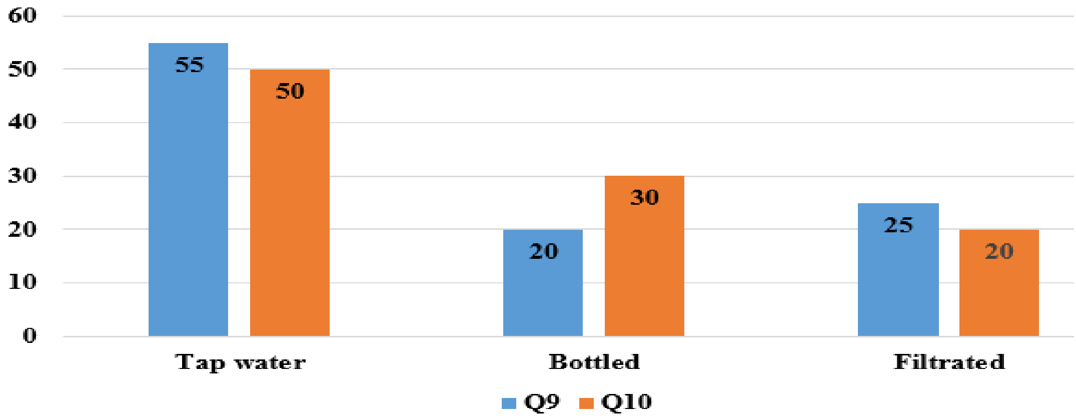


Figure 8.2 – Water Drinking Habits at Work and at Home

Respondents were inquired about their perceptions on current water supply service in Fort Beaufort community **(Q11).** 30% of the respondents demonstrated that the water supply service they receive was fair, 17% reported it has good service, 12% reveal it was very good and 28% of the respondents rating the water supply service as poor while 13% did not know how to rate the water supply service as shown in Figure 8.3.

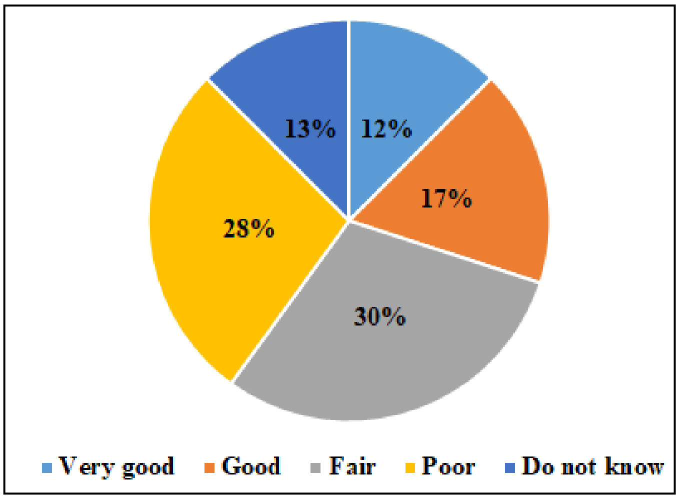


Figure 8.3 – Respondents perceptions on Water Supply Service in Fort Beaufort

WATER LEAKAGES IN STREET/TOWN

Respondents were inquired if they know the organisation or authoritative body to report street water leakages to (**Q12**). 67.5% of the respondents indicated No while only 32.5% said yes they know the authoritative body to report street water leakage to, as shown in figure 8.4.

In addition, respondents from Fort Beaufort were inquired whether or not they have ever reported street water leakage to the local water authority before **(Q13).** Figure 8.4 shows that the majority of respondents (95%) have never reported water leakage, while only 5% have reported street water leakage before to the local water authority.

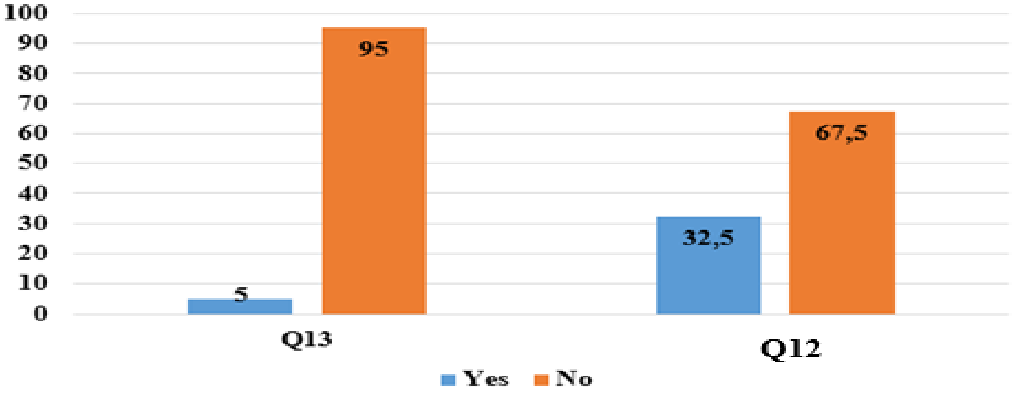


Figure 8.4– Street Water Leakages

Respondents in For Beaufort were asked to show whether or not they knew the number to contact at the local water authority if there is water leakage or pipe burst in the street or town **(Q14)**. Discoveries are outlined in Figure 8.5. Most of respondents (88%) reported that they do not know the contact number to call to report water leakages within the street or in town, if such problem occurred. Only 12% knew the number to call when the issue emerge.

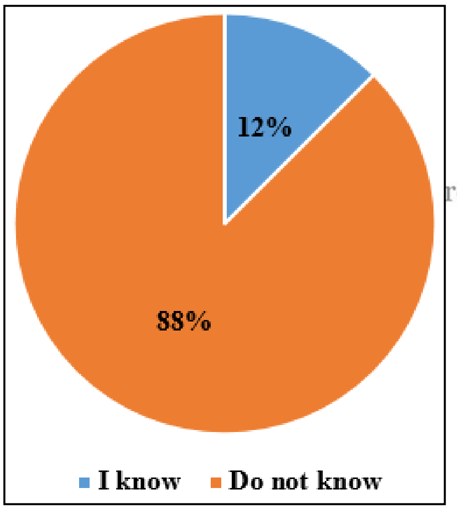


Figure 8.5 – Knowledge of Local Water Authority Contact Number

WATER RATES

Respondents were inquired if they know how much they pay for 1 kilolitre (*1000 litre*) of water consumed (**Q15**). The findings appeared that 70% of the participants do not know how much they pay per kilolitre of water and on other hand only 30% indicated yes they know the amount as shown in Figure 8.6 below.

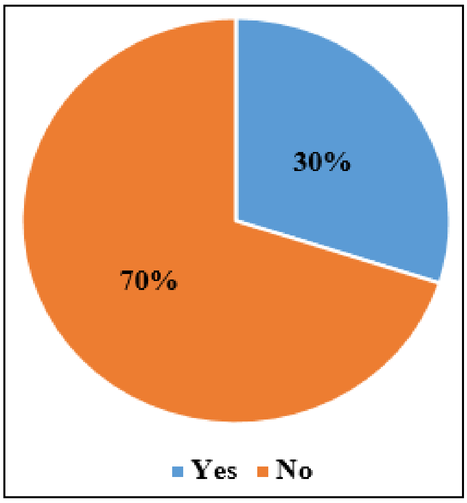


Figure 8.6 – Knowledge of Rate of 1 Kilolitre of Water

Respondents from Fort Beaufort community were asked to rate the cost of water supplied by their water authority. Figure 8.7 **(Q16)** below exhibits respondents' rating of the cost of water services. Close to half of the total sample (47.5%) accepted that they did not know while 42.5% said that it was normal. 7.5% said it was too high and only 2.5% believed that it was too low.

Once more, when inquired to give opinion on comparison of the current water rates and other utility payments rate (example electricity) **(Q17)**, 25% of the respondents said they did not know, 57.5% said it was normal when compared to other utility rates. 17.5% believed it was too high and 2.5% said it too low as appeared in Figure 8.7.

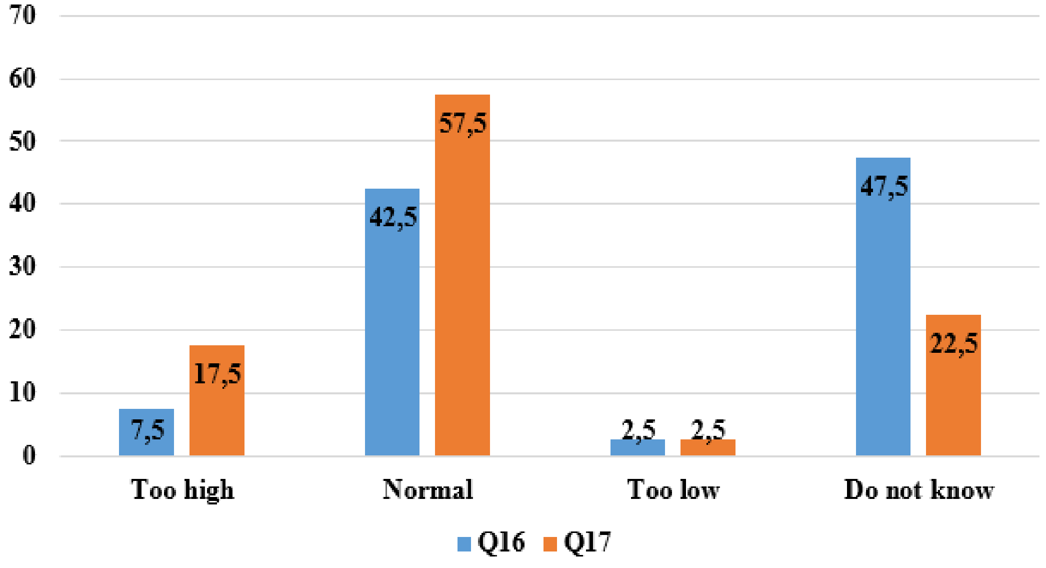


Figure 8.7 – Opinion on Water Rate

SURVEY ON MOBILE PHONES

Respondents were asked if they own a mobile phone **(Q18)**. 87% of the respondents said yes they own at least one mobile phone while only 13% do not possess a mobile phone as indicated in figure 8.8.

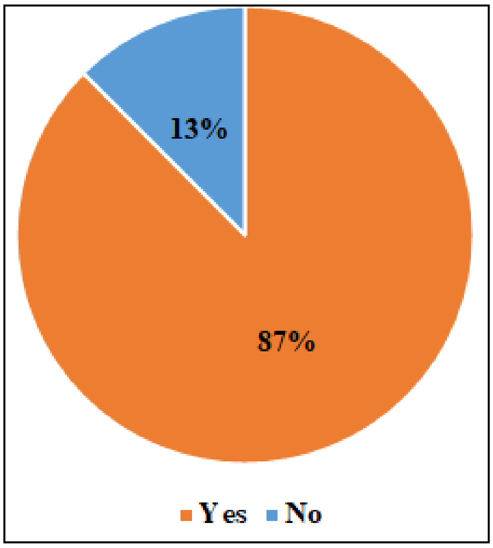


Figure 8.8 – Owners of Mobile Phone

Respondents were inquired how many mobile phones they have, which includes both personal and business mobiles **(Q19)**. Close to half (47%) of the respondents have at least one mobile phone, 27% indicated they have 2 mobile phones, 13% said they have 3 mobile phones and 13% indicated other, as appeared in Figure 8.9.

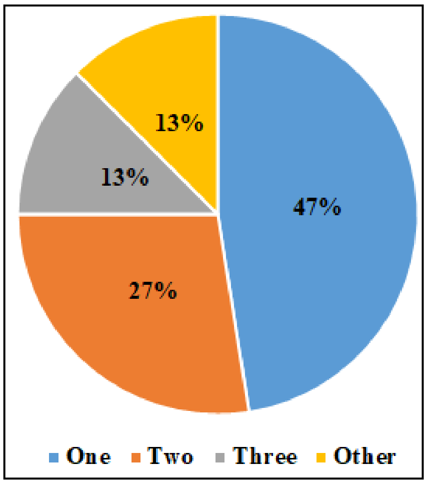


Figure 8.9 – Number of Phones Own by Respondents in Fort Beaufort

Question 20 focused on the type of mobile phone devices the respondent's possess. Close to half (47.5%) of the respondents own Android phones. 15% own iPhone, 22.5% indicated they own other kind of mobile phones. 2.5% said they own Symbian (Nokia) phone. 10% owns windows type with 2.5% owning blackberry phone, as illustrated in figure 8.10.

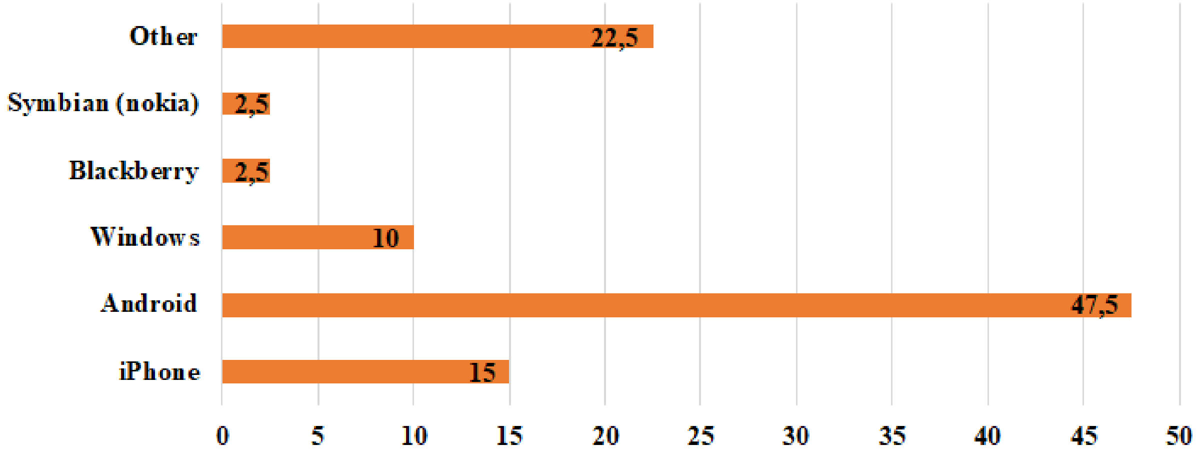


Figure 8.10 – Respondents Mobile Phone Type

Respondents were inquired as to whether their mobile phone has a camera functionality or not (**Q21**). 72% reported yes while 28% showed No, as appeared in Figure 8.10.

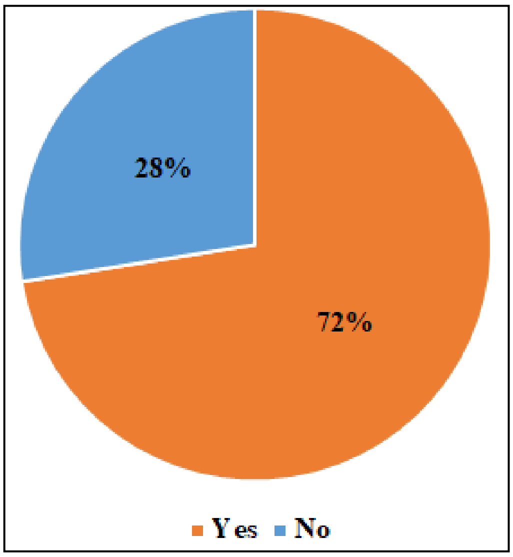


Figure 8.10 – Mobile Phone with Camera Functionality

Respondents were asked which category of applications do they use and enjoy using **(Q22)**. 57% indicated they enjoy using social networks while 20% said they enjoying game applications and 23% indicated they use other applications, as shown in Figure 8.11 below.

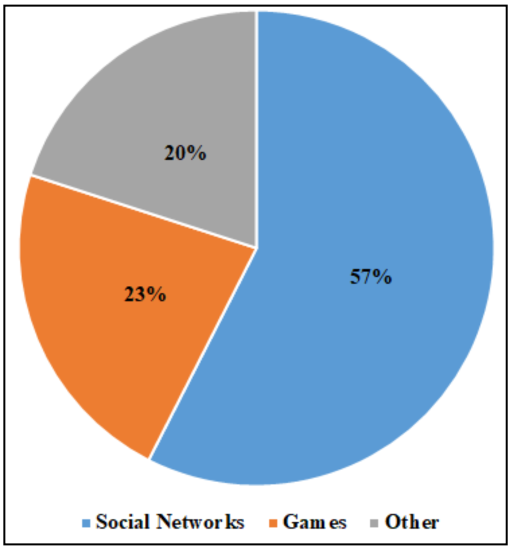


Figure 8.11 – Respondents Category of Mobile Phone Applications in Fort Beaufort

Respondents were inquired in the event that, if they have ever downloaded application for their mobile device before (Q**23**). 42% indicated yes they download both free and paid applications, 20% indicated they always download paid application. On the other hand 20% said they always download applications that are free and 17.5% indicated they have never downloaded application before, as shown in Figure 8.12.

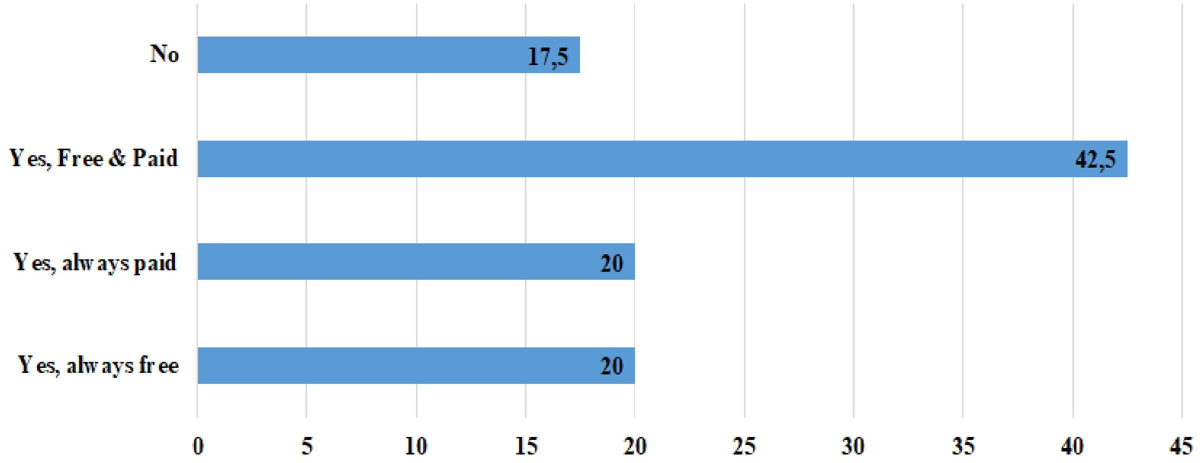


Figure 8.12– Application Downloads in Fort Beaufort Community

Respondents were inquired if their mobile phone can have access to internet (**24**). 82% indicated yes whereas while 18% indicated No, as shown in Figure 8.13.

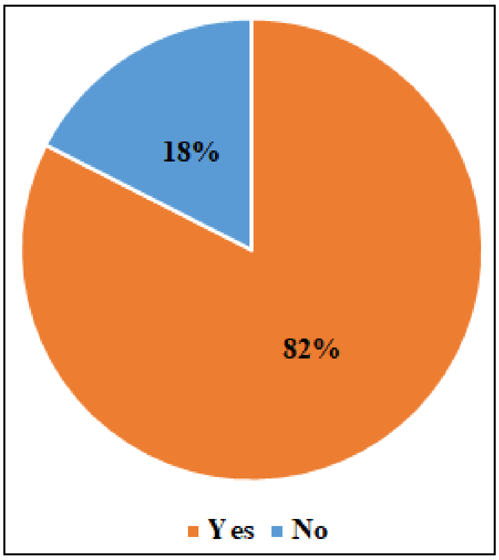


Figure 8.13 – Internet Accessibility

OBSERVATIONAL SITE VISITATION IN ALICE AND FORT BEAUFORT

From the visibility research carried out at the two research sites, water leakages in the street or town was seen as one of the contributing factors to water that is lost both in Alice and Fort Beaufort. Figure 8.14 and Figure 8.15 shows some street leakages in Alice and Fort Beaufort that were not attended to.

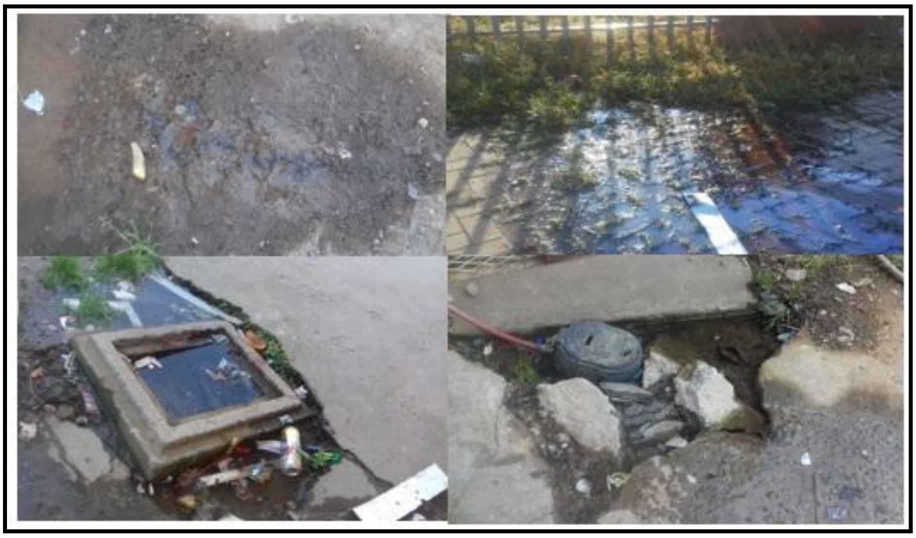


Figure 8.14 – Street/Town Leakages in Alice Community

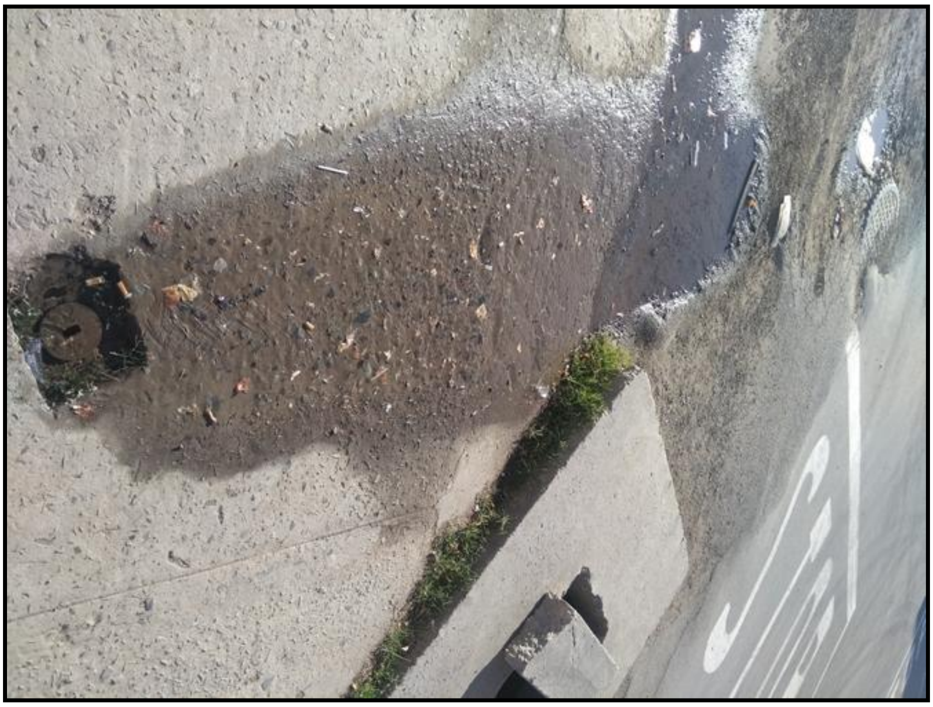


Figure 8.15– Street/Town Leakages in Fort Beaufort Community

The facilities at both sites were evaluated base on their physical qualities. Figure 8.16 shows some equipment and facilities used at Alice and Fort Beaufort water works.

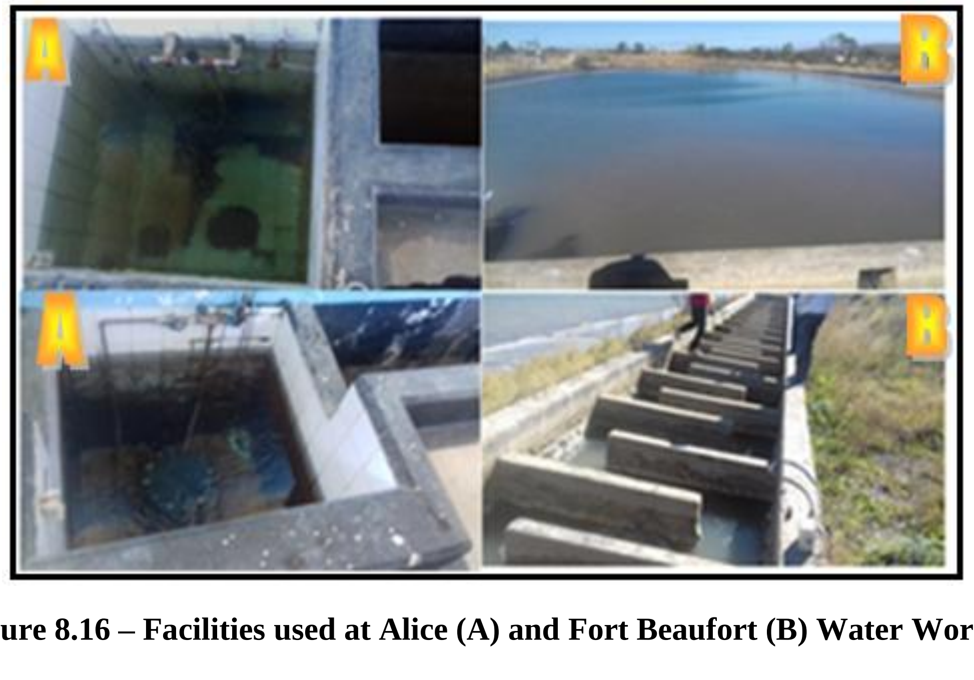
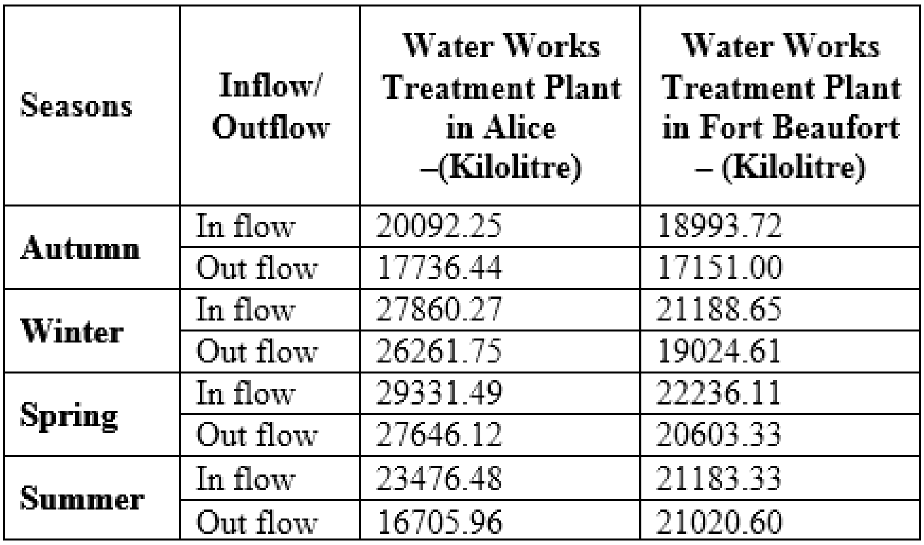


Figure 8.16 – Facilities used at Alice (A) and Fort Beaufort (B) Water Work

Data was collected for the period of one year from Jan to Dec 2017 for the amount of water that comes in (input flow) and what goes out (output) for both Alice and Fort Beaufort Water Works. Table 8.3 shows seasonal average water data for 2017 sampling period.



Seasons	Inflow/ Outflow	Water Works Treatment Plant in Alice –(Kilolitre)	Water Works Treatment Plant in Fort Beaufort – (Kilolitre)
Autumn	In flow	20092.25	18993.72
	Out flow	17736.44	17151.00
Winter	In flow	27860.27	21188.65
	Out flow	26261.75	19024.61
Spring	In flow	29331.49	22236.11
	Out flow	27646.12	20603.33
Summer	In flow	23476.48	21183.33
	Out flow	16705.96	21020.60

Table 8.3 – Seasonal Average Water Data in 2017 for Alice and Fort Beaufort Water Works.

DISCUSSION

This survey contributes to knowledge about the attitudes of both Alice and Fort Beaufort communities', on their perceptions and behaviours in relation to water conservation. The data analysis revealed optimistic attitude and perception of respondents in these communities. Overall respondents indicated positive practices, attitudes and perceptions towards water conservation. The results show that the majority of respondents (agree and strongly agree) in Alice (95%) and Fort Beaufort (85.5%) consider water as an important resource. These findings are in agreement with the research done by Onyenankeya et al (2015) at the University of Fort Hare in Alice, in which they found 80.5% of respondents were of the opinion that water conservation is important. Although respondents in these communities have favourable attitudes towards water conservation, however, attitudes and perceptions are not always translated into action. According to Dolnicar and Hurlimann (2010), attitudes are translated into action when it is easy or convenient to do so.

According to the results from the present research, most respondents (77.5% in Alice and 55% in Fort Beaufort) indicated that their usual source of drinking water at home is from the tap while (65% in Alice and 50% in Fort Beaufort) said their usual source of drinking water at work is from the tap as well. Clearly individuals use water generally in their day by day life therefore it is extremely essential for their wellbeing as well as for their cleanliness. The majority of the general respondents know about the significance of saving water. Despite their strong feelings about the importance of water conservation, very few respondents are aware of public information on water conservation. This is crucial considering the facts that majority of respondents (85% in Alice and 55% in Fort Beaufort) indicated that they were not aware of any public program on water conservation in their communities. However, there is still room for government, water authorities and stakeholders to reconsider the use of ICTs and its prospects to enhance the current conservation awareness campaigns in order to secure the objectives of the water conservation and water sustainability.

The respondents were with the opinion that the existing water supply service in their community is good (27% in Alice and 17% in Fort Beaufort) and 35% in Alice and 30% in Fort Beaufort felt the supply service is fair and are quite satisfied with the supply delivery. Even though, the people are satisfied with the water supply services, as many as 28% in Fort Beaufort and 13% in Alice rated the supply service as poor. It is worrisome that this large number of respondents was unhappy of this service. The department therefore needs to improve on their service with regards to water supply and water management services in these two communities. The services can be improved by putting in place measures that ensure easy and efficient communication between the department and the communities such as mobile applications.

Most of the respondents (77.5% in Alice and 67.5% in Fort Beaufort) said they did not know the relevant departments to report their concerns regarding water leakages in town. This fact points to the need of more awareness and education on what to do when they have problems in relation to water leakages in town. It is worthy of note that almost all the respondents (93% in Alice and 85% in Fort Beaufort) did not know the customer-service contact details for reporting water leakages in town. The relevant water authority needs to provide an improved alternative medium of communication for the residents to use to contact them. Most respondents (85% in Alice and 95% in Fort Beaufort) have never reported water leakage. This may be due to the fact that most respondents do not know the relevant body to report water leakages to and even those who are aware did not have the means of contacting them. There is a need to broaden water awareness campaigns in all the municipalities and districts (City of Cape Town, 2017). Authorities in these communities need to increase and expand its existing information and educational initiatives, so that an increased number of residents is aware of the service-contact details for queries relating to water leakages. In this digital era that we are in, the municipalities can make use of different mediums such as instant messaging, social networks and mobile application.

Also, households in Alice and Fort Beaufort that shared their experiences and perceptions over water rates, this revealed some gap between water authorities and the community. Lack of communication between consumers of water and suppliers of water has been identified as a major cause. For instance, majority of the respondents (80% in Alice and 70% in Fort Beaufort) do not have a clue of how much they are being charged per kiloliter of water. Considering the result that, when asked to rate current water tariff in communities (*Too high, Normal, Too Low and Do Not Know*), more than half the respondents (57% in Alice) said they don't know while 47% in Fort Beaufort said the same. This is a challenge considering the large number from both sites. The researcher is of the opinion that, when individual know how much they are been charge per little bit of water consumed, it will help them to make informed decisions on their consumption rate. Hence, it helps to reduce water consumption and in a way save households some money as well. Therefore, information on water consumption from water authorities needs to be disseminated and be transparent.

From the visibility research carried out at the two research sites, one of the factors that contribute to water lost was observed as result of water leakages at Alice and Fort Beaufort towns especially in Alice. The inadequate water has been going to waste without any effective or appropriate measures to tackle this issue and it cost both the consumers and the water works millions of rand. Figure 8.14 and 8.15 above shows some street leakages in both Alice and Fort Beaufort that were left unattended to for almost two weeks. Literature presented that, wireless sensor networks with combination of mobile applications can help in early detection of pipeline leakages. Therefore, the advancement in mobile phones with regard to sensors can be used in these areas thereby minimizing the amount of water waste.

The facilities at both sites were evaluated base on their physical qualities. Fort Beaufort facility is better as compare to Alice facility which needed to be attended to if the qualities of the water product is to meet the regulation and standard of Department of Water Affairs and Forestry. Also the capacity design for both facilities is seriously under stress, number of people that they are serving is far more than the people that the original capacity was design for. Lack of maintainers and upgrade of some tools was highlighted as a contributor to poor performance of both facilities. Figure 8.16 shows some equipment and facilities used at Alice and Fort Beaufort water works.

The survey revealed impact of climatic condition on the amount of water that comes in (inflow) and what goes out (outflow) of both facilities. These two water works and the consumers need to be water cautious if water is to be available for their day to day use. The effect of climatic condition is not only experience in the quantity of the water but also in the quality of the water, which amount to high cost for the water work in term of the chemical use and infrastructure that is destroyed through high temperature such as pipe leakage. Table 8.3 above shows the seasonal average of inflow and outflow rates of two water works facilities. The decline in inflow and outflow rates in summer at Alice revealed the negative impact of climatic change. The main factor that is attributed to the decline may be the higher temperatures that normally occur at Alice in summer. In chapter 2 of this research, many mobile phones tools were identified which can be applied to mitigate the effect of climate change on water availability. It is therefore recommended that reviews and reforms had to be made regarding policies and regulations on water management to incorporate mobile applications into this sector.

From the survey results it can be seen that majority of the respondents (87% in Fort Beaufort and 80% in Alice) own a mobile phone. 28% of respondents in Alice and 27% in Fort Beaufort own at least two mobile phones. The most common type of mobile phone that respondents uses (57.5% in Alice and 47.5% in Fort Beaufort) is android phone and 82% in Fort Beaufort and 75% in Alice have a phone that can access internet. It is evident from the survey results that mobile phones are common in these two communities. Therefore, it was seen as one appropriate tool that could be used to manage some of the water related issues that was found in the survey. Therefore, a mobile application has been developed to manage water loss and some water related issues and will serve as a feedback communication between the community and water sector.

The SaveAmanzi application is developed to take into consideration the challenges these two research sites are facing with respect to water issues. There is a need of the application in the sense that, water leakages can be reported hence ensuring that the quantity of water that goes to waste can be now reduced. The usage of the application can also help communities save money since information like tips of saving water and emergency guidelines to follow to minimize consumption rate is provided in the application. When less water is consumed it amounts to low water tariff for consumers. The application will also serve as a medium of communication between the water authorities and the community members. Therefore ensuring that information (water availability, maintenance etc…) is reached to the people at the right time. The application will also provide information on water tariffs. It is assumed when people are aware of how much they pay they will be able to make informed decisions concerning their daily water usage. SaveAmanzi app creates awareness to users on water conservation, as random notifications are sent to all users that have the app installed on their phones.

CONCLUSION

Appendix 3 has shown analysis and discussion of data that was collected from Alice and Fort Beaufort communities. This section has also revealed some water related issues in these sites and.

APPENDIX 4 : POST- TEST QUESTIONNAIRE

Question 1: Do you feel that you successfully completed all the tasks?

This question was asked with the idea of finding out if the participants were able to complete all the task scenarios mentioned above. The participant answers this question by indicating with X with preference of YES or NO.

Question 2: In relation to other application I have used, I found the SaveAmanzi application to be:

This question was asked in other to find out how difficult or easy it was to use this application comparing with other applications the participants has already used. The participants answer this question by indicating with X the following options: very easy, easy, neither easy nor difficult, difficult and very difficult.

Question 3: The buttons or labels were well organized and easy to find.

This questioned was asked to understand how participants feel about the action buttons and labels after using the application. The participants answer this question by indicating with X the following options: strongly disagree, disagree, neither agree nor disagree, agree and strongly agree.

Question 4: I found navigating around the SaveAmanzi Application screen to be:

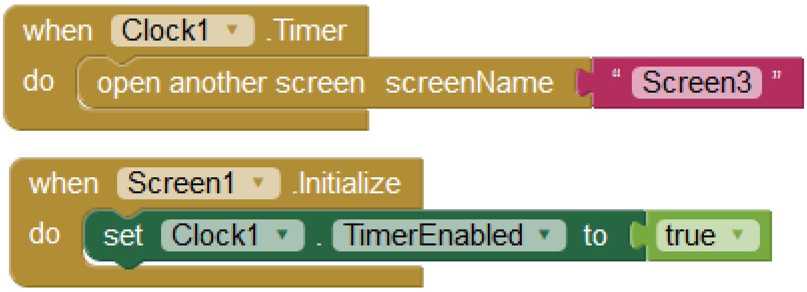
This question was asked to find out participants' thoughts and feelings of navigating from one screen to other. The participants answer this question by indicating with X the following options: very easy, easy, neither easy nor difficult, difficult and very difficult.

Question 5: My overall impression of the SaveAmanzi application is:

This question was asked to find out the participants' overall impression after using the application. The participants answer this question by indicating with X the following options: very negative, negative, neither negative nor positive, positive and very positive.

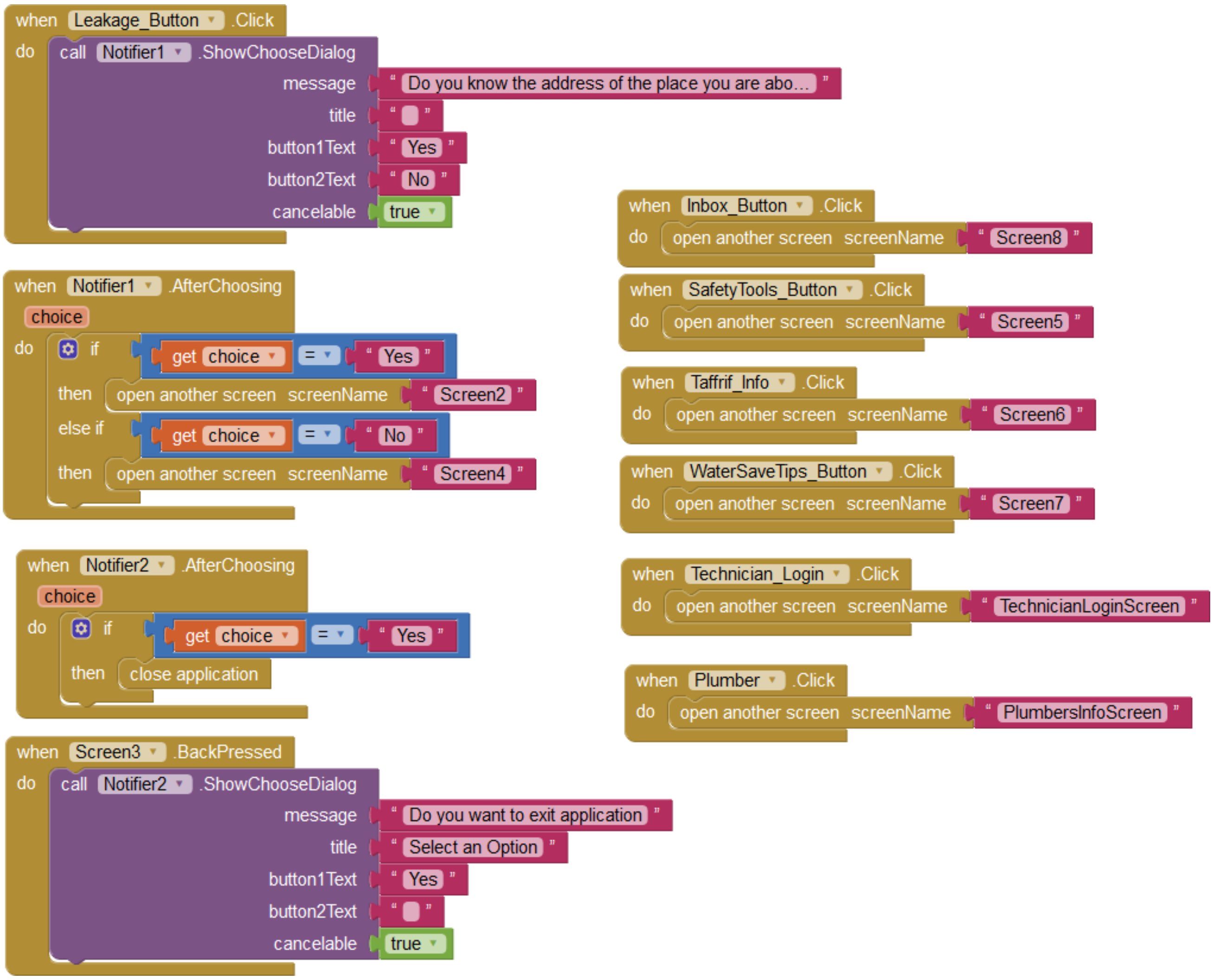
APPENDIX 5: SOURCE CODE BLOCKS FOR SAVEAMANZI MOBILE APPLICATION

Splash Screen Block Code



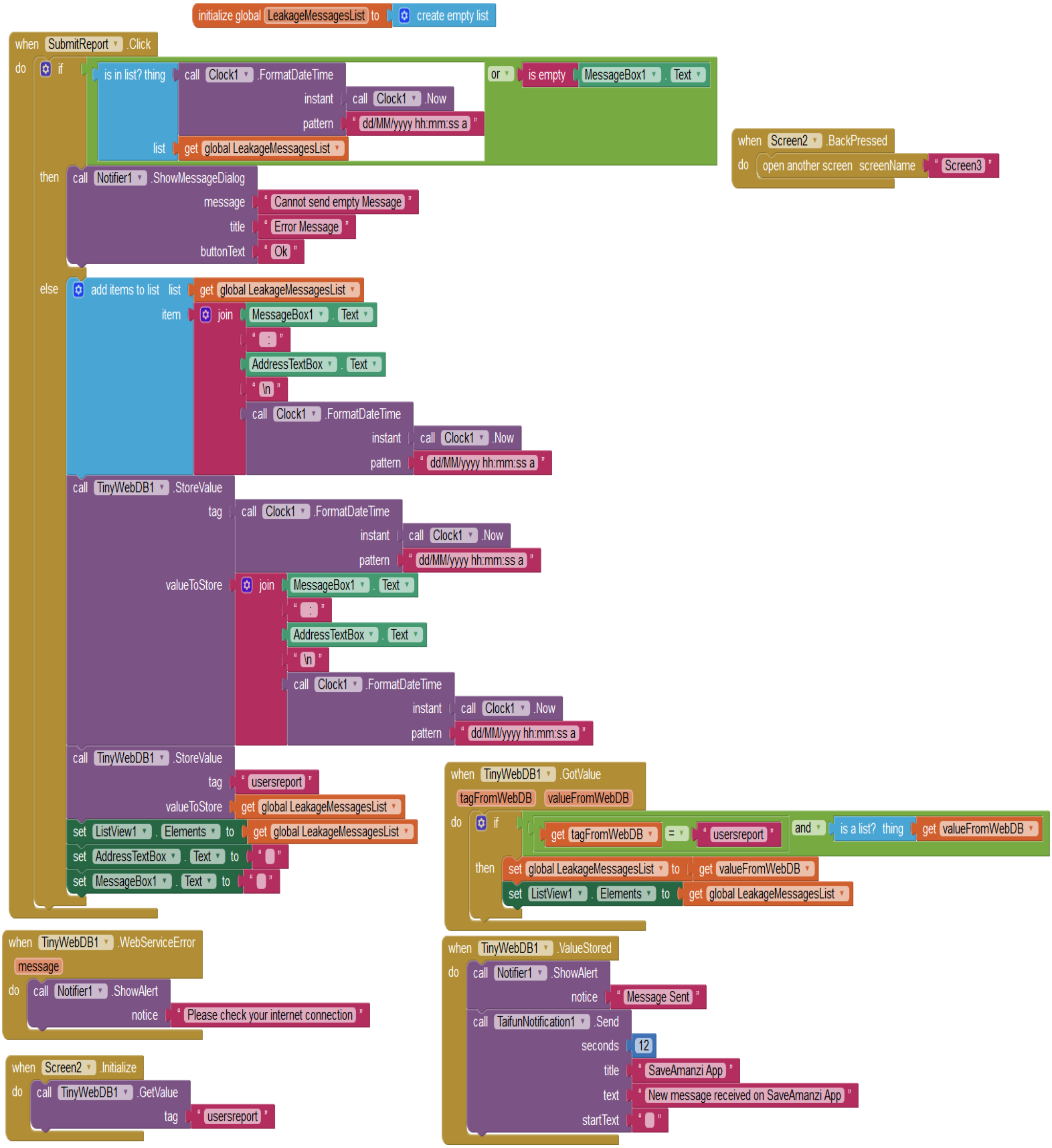
```
when Clock1 .Timer
do  open another screen  screenName  " Screen3 "

when Screen1 .Initialize
do  set Clock1 . TimerEnabled to  true
```

Home Screen Block Code



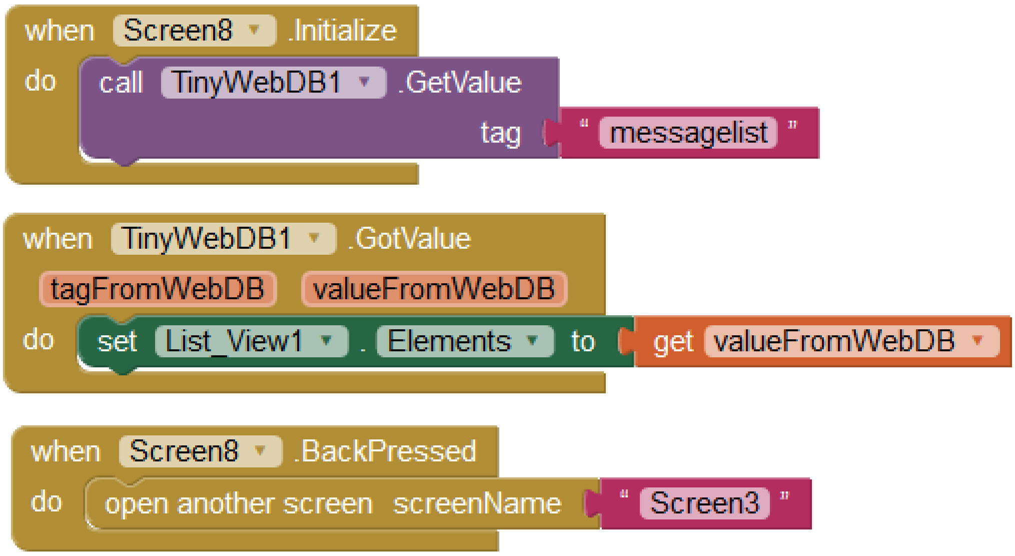
```
when Leakage_Button .Click
do  call Notifier1 .ShowChooseDialog
                 message  " Do you know the address of the place you are abo... "
                   title  "  "
             button1Text  " Yes "
             button2Text  " No "
              cancelable  true

when Notifier1 .AfterChoosing
 choice
do  if    get choice = " Yes "
    then  open another screen  screenName  " Screen2 "
    else if  get choice = " No "
    then  open another screen  screenName  " Screen4 "

when Notifier2 .AfterChoosing
 choice
do  if    get choice = " Yes "
    then  close application

when Screen3 .BackPressed
do  call Notifier2 .ShowChooseDialog
                 message  " Do you want to exit application "
                   title  " Select an Option "
             button1Text  " Yes "
             button2Text  "  "
              cancelable  true

when Inbox_Button .Click
do  open another screen  screenName  " Screen8 "

when SafetyTools_Button .Click
do  open another screen  screenName  " Screen5 "

when Taffrif_Info .Click
do  open another screen  screenName  " Screen6 "

when WaterSaveTips_Button .Click
do  open another screen  screenName  " Screen7 "

when Technician_Login .Click
do  open another screen  screenName  " TechnicianLoginScreen "

when Plumber .Click
do  open another screen  screenName  " PlumbersInfoScreen "
```

Reporting Water Leakages Block Code



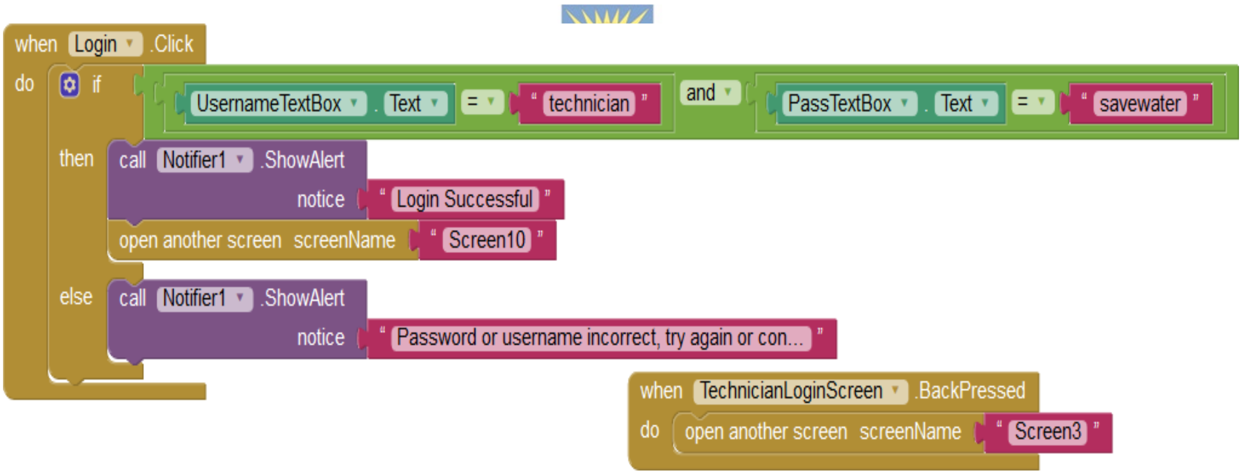
```
initialize global LeakageMessagesList to  create empty list

when SubmitReport .Click
do  if  is in list? thing  call Clock1 .FormatDateTime
                                  instant  call Clock1 .Now
                                  pattern  " dd/MM/yyyy hh:mm:ss a "
                           list  get global LeakageMessagesList
        or  is empty  MessageBox1 . Text
    then  call Notifier1 .ShowMessageDialog
                               message  " Cannot send empty Message "
                                 title  " Error Message "
                            buttonText  " Ok "
    else  add items to list  list  get global LeakageMessagesList
                             item  join  MessageBox1 . Text
                                         " : "
                                         AddressTextBox . Text
                                         " \n "
                                         call Clock1 .FormatDateTime
                                                   instant  call Clock1 .Now
                                                   pattern  " dd/MM/yyyy hh:mm:ss a "
    call TinyWebDB1 .StoreValue
                         tag  call Clock1 .FormatDateTime
                                        instant  call Clock1 .Now
                                        pattern  " dd/MM/yyyy hh:mm:ss a "
                  valueToStore  join  MessageBox1 . Text
                                      " : "
                                      AddressTextBox . Text
                                      " \n "
                                      call Clock1 .FormatDateTime
                                                instant  call Clock1 .Now
                                                pattern  " dd/MM/yyyy hh:mm:ss a "
    call TinyWebDB1 .StoreValue
                         tag  " usersreport "
                  valueToStore  get global LeakageMessagesList
    set ListView1 . Elements to  get global LeakageMessagesList
    set AddressTextBox . Text to  " "
    set MessageBox1 . Text to  " "

when Screen2 .BackPressed
do  open another screen  screenName  " Screen3 "

when TinyWebDB1 .GotValue
 tagFromWebDB  valueFromWebDB
do  if  get tagFromWebDB  =  " usersreport "  and  is a list? thing  get valueFromWebDB
    then  set global LeakageMessagesList to  get valueFromWebDB
          set ListView1 . Elements to  get global LeakageMessagesList

when TinyWebDB1 .WebServiceError
 message
do  call Notifier1 .ShowAlert
                     notice  " Please check your internet connection "

when TinyWebDB1 .ValueStored
do  call Notifier1 .ShowAlert
                     notice  " Message Sent "
    call TaifunNotification1 .Send
                         seconds  12
                           title  " SaveAmanzi App "
                            text  " New message received on SaveAmanzi App "
                       startText  " "

when Screen2 .Initialize
do  call TinyWebDB1 .GetValue
                         tag  " usersreport "
```

Checking Inbox Blocks Code (Message from water authority)



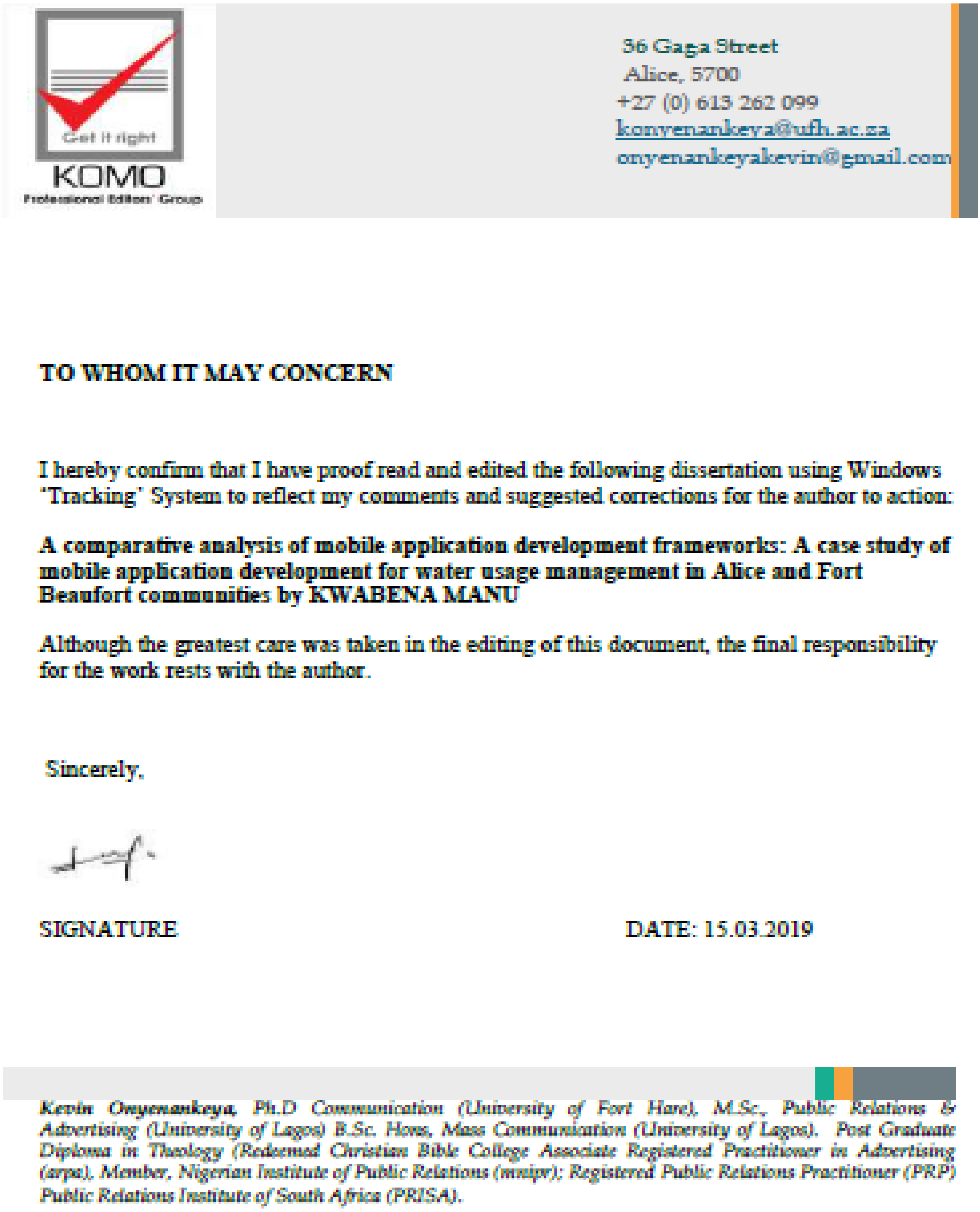
```
when Screen8 .Initialize
do  call TinyWebDB1 .GetValue
                       tag  " messagelist "

when TinyWebDB1 .GotValue
  tagFromWebDB  valueFromWebDB
do  set List_View1 . Elements to  get valueFromWebDB

when Screen8 .BackPressed
do  open another screen  screenName  " Screen3 "
```

Login as Technician Blocks Code



```
when Login .Click
do  if  UsernameTextBox . Text = " technician "  and  PassTextBox . Text = " savewater "
    then  call Notifier1 .ShowAlert
                       notice  " Login Successful "
          open another screen  screenName  " Screen10 "
    else  call Notifier1 .ShowAlert
                       notice  " Password or username incorrect, try again or con... "

when TechnicianLoginScreen .BackPressed
do  open another screen  screenName  " Screen3 "
```

APPENDIX 6: LANGUAGE EDITOR'S CERTIFICATE



Get it right
KOMO
Professional Editors' Group

36 Gaga Street
Alice, 5700
+27 (0) 613 262 099
konyenankeya@ufh.ac.za
onyenankeyakevin@gmail.com

TO WHOM IT MAY CONCERN

I hereby confirm that I have proof read and edited the following dissertation using Windows 'Tracking' System to reflect my comments and suggested corrections for the author to action:

A comparative analysis of mobile application development frameworks: A case study of mobile application development for water usage management in Alice and Fort Beaufort communities by KWABENA MANU

Although the greatest care was taken in the editing of this document, the final responsibility for the work rests with the author.

Sincerely,

SIGNATURE

DATE: 15.03.2019

Kevin Onyenankeya, Ph.D Communication (University of Fort Hare), M.Sc., Public Relations & Advertising (University of Lagos) B.Sc. Hons, Mass Communication (University of Lagos). Post Graduate Diploma in Theology (Redeemed Christian Bible College Associate Registered Practitioner in Advertising (arpa), Member, Nigerian Institute of Public Relations (mnipr); Registered Public Relations Practitioner (PRP) Public Relations Institute of South Africa (PRISA).