The Design, Development and Evaluation of Cross-platform Mobile Applications and Services Supporting Social Accountability Monitoring

Submitted in fulfilment
of the requirements of the degree of

Master of Science

of Rhodes University

Edward Robin Reynell

Grahamstown, South Africa
March 2016

Abstract

Local government processes require meaningful and effective participation from both citizens and their governments in order to remain truly democratic. This project investigates the use of mobile phones as a tool for supporting this participation. MobiSAM, a system which aims to enhance the Social Accountability Monitoring (SAM) methodology at local government level, has been designed and implemented. The research presented in this thesis examines tools and techniques for the development of cross-platform client applications, allowing access to the MobiSAM service, across heterogeneous mobile platforms, handsets and interaction styles. Particular attention is paid to providing an easily navigated user interface (UI), as well as offering clear and concise visualisation capabilities. Depending on the host device, interactivity is also included within these visualisations, potentially helping provide further insight into the visualised data. Guided by the results obtained from a comprehensive baseline study of the Grahamstown area, steps are taken in an attempt to lower the barrier of entry to using the MobiSAM service, potentially maximising its market reach. These include extending client application support to all identified mobile platforms (including feature phones); providing multi-language UIs (in English, isiXhosa and Afrikaans); as well as ensuring client application data usage is kept to a minimum. The particular strengths of a given device are also leveraged, such as its camera capabilities and built-in Global Positioning System (GPS) module, potentially allowing for more effective engagement with local municipalities. Additionally, a Short Message Service (SMS) gateway is developed, allowing all Global System for Mobile Communications (GSM) compatible handsets access to the MobiSAM service via traditional SMS. Following an iterative, user-centred design process, a thorough evaluation of the client application is also performed, in an attempt to gather feedback relating to the navigation and visualisation capabilities. The results of which are used to further refine its design. A comparative usability evaluation using two different versions of the cross-platform client application is also undertaken, highlighting the perceived memorability, learnability and satisfaction of each. Results from the evaluation reveals which version of the client application is to be deployed during future pilot studies.

Acknowledgements

First and foremost, I would like to thank my parents for their continual love and support throughout my lengthy academic career. I would like to express my utmost gratitude to my supervisors: Hannah Thinyane and Ingrid Siebörger. Your advice, feedback and guidance has been invaluable throughout this research project. It has been a privilege having you as my supervisors. I would also like to thank my girlfriend, Charis Futcher, for the endless smiles and support, as well as her tireless proof-reading efforts.

In addition, I would like to acknowledge the financial support from the National Research Foundation (NRF). I would also like to thank the Ford Foundation for their generous grant, which helped make this research project possible.

Finally, I would like to thank Rhodes University for providing me with the opportunity to pursue my Masters degree in Computer Science.

Contents

List of Figures

List of Tables

Glossary

API	Application Programming Interface
CN1	Codename One
CoGTA	Department of Cooperative Governance and Traditional Affairs
CPF	Cross-platform Framework
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
REST	Representational State Transfer
SAM	Social Accountability Monitoring
SDK	Software Development Kit
SMS	Short Message Service
SMSC	Short Message Service Centre
UI	User Interface
URI	JavaScript Object Identifier
UX	User Experience
WAP	Wireless Application Protocol
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Background

South Africa's constitution requires local government to perform certain tasks, including providing a democratic and accountable government for local communities; ensuring the sustainable provision of services; promoting social and economic development; as well as encouraging the involvement of communities and community organisations in matters of local government [69]. While the intentions of municipalities are often genuine in their desire to fulfil these tasks, many of these ideals have yet to be obtained [135]. An October 2014 review conducted by the Department of Cooperative Governance and Traditional Affairs (CoGTA) concluded that only 7% of South Africa's 278 municipalities were "doing well" [37, p.6], warning that "despite our achievements, our governance system is a cause for concern" [37, p.4].

Poor service delivery is often blamed on poor capacity, limited funds, corruption and an absence of effective leadership [135]. This is compounded by the fact that local municipalities seldom recognise and reward good performance, while at the same time are ineffective at dealing with under-performance [37]. In addition, "slow or inadequate responses to service delivery challenges are in turn linked to the breakdown of trust [between local government and the public]" [37, p.5], a symptom further agitated by public representatives often distancing themselves from the communities which they are obliged to serve [37].

As a result, citizen response often takes the form of protest and strike action, with an April 2012 article citing South Africa as the "protest capital of the world" [4, p.1], after more than 900 service delivery protests occurred within a single calendar year [21]. While protest and strike action are a means of participating in local government matters, they are not considered *meaningful* in that they alone do not bring about lasting positive change [21]. In spite of the challenges, however, local governments are structured in such a way as to increase meaningful citizen participation by "placing more power and resources at a closer and more easily-influenced level of government" [148, p.136]. As maintained by the Minister of Cooperative Governance and Traditional Affairs, Pravin Gordhan, during his budget vote speech in July 2014:

> "Local government is the crucible in which the complex processes of development, governance, transformation of life and living conditions is taking place daily." [66, p.1]

The key to successful public involvement is to ensure the "meaningful, informed, and effective participation" of citizens in government processes [203, p.170]. However, adequate mechanisms and skills are required in order to hold service providers accountable for their performance, or lack thereof. The social accountability monitoring (SAM) methodology is one such mechanism, an overview of which is provided next.

Developed by the Centre for Social Accountability at Rhodes University, the SAM methodology is a "rights-based and evidence-based framework for understanding and participating in government service delivery processes" [203, p.174]. While previously untested at local government level, the SAM methodology has been successfully used to monitor government at both national and provincial levels.

Figure 1.1: The Social Accountability Monitoring methodology (adapted from [180])

Implementing the SAM methodology allows resources to be effectively and accountably managed, and is done so by mapping five inter-dependent processes onto the public resource management system, as shown in Figure 1.1. Basic tools are provided by the methodology, allowing the public to engage in each process by interrogating the government documents which each of the processes produce: budgets, plans, financial reports, performance reports, audit findings and oversight committee minutes. Despite its successful use, the SAM methodology suffers from two limitations. Firstly, as the methodology is performed via the analysis of government documents, it is primarily suited to desk-based research and *post hoc* analysis. Secondly, in order to rigorously apply the methodology, a certain degree of literacy and analytical skills are required, often excluding those most affected by inadequate service delivery: the poor and marginalised. The development of additional tools to complement the SAM methodology is therefore needed.

While access to formal education remains poor and literacy rates low, South Africa's mobile phone penetration rate is comparable to that of many developed nations [76]. Although estimates vary, the GSMA has reported a mobile penetration rate of 65.7% while the Nielsen Group has reported a figure of 76% [77,94]. In addition, South Africa has been ranked fifth worldwide for mobile data usage, ahead of countries such as the United States, which ranks seventh [94]. The same study showed that more South Africans use mobile phones than any other modern and traditional ICTs, including computers (6 million), televisions (27 million) and radios (28 million) [94]. With this in mind, it is proposed that the SAM methodology could be further strengthened through real-time, user-generated data obtained from the public, submitted using their mobile phones. This would allow citizens to participate in issues of local government by providing them a direct means through which to interact with their municipality and express their needs and concerns. In addition, municipalities would be afforded a direct communication channel to their constituents, allowing for the delivery of timely, location relevant information to residents.

The research presented in this thesis forms a part of the MobiSAM (mobile social accountability monitoring) project, which aims to investigate the use of mobile phones as a platform to facilitate evidence-based engagement with local municipalities. In turn, this engagement aims to promote an increase in accountability as well as improve service delivery and meaningful participation. Currently being piloted within the Eastern Cape city of Grahamstown, MobiSAM introduces a mobile phone-based polling framework and service delivery complaint reporting tool which seeks to promote active citizen participation between residents and their local government, tasked with providing them basic services. The pilot project has partnered with Grocott's Mail, a local community newspaper, to broadcast collected information, with the aim to facilitate a two-way dialogue between citizens and their municipality.

As part of its recently launched (December 2014) *Back-to-Basics* framework, geared towards improving the functioning of local governments, CoGTA insisted that: "Municipalities *must* develop affordable and efficient communication systems to communicate regularly with communities and disseminate urgent information" [37, p.11]. In addition, the framework aims to provide "all South Africans [with] a basic set of tools by which they can hold their municipalities to account and measure whether they are living up to their promises" [37, p.9], with performance indicators being collated and published [37]. These are two roles which MobiSAM is well positioned to fulfil.

1.2 Project context

Makana Municipality is situated in the Eastern Cape Province, one of South Africa's poorest provinces. The area experiences a high unemployment rate (63.4%) and low levels of formal education (39.2% of residents have received a primary school education or none at all) [132]. As a result, 55.5% of Makana residents live below the poverty line [132]. Efforts to alleviate poverty are hampered by a sluggish economy and there is an increasing dependency on social grants, with half of all Makana residents receiving some form of government grant in 2009 [131]. This places a significant burden on the municipality, particularly in delivering basic services. The municipality is responsible for (amongst other things) municipal health services; municipal roads; sanitation; electricity reticulation; potable water; refuse removal; refuse dumps and solid waste disposal; child care facilities; and local tourism [70]. Despite a R266 million budget in 2009/10 [130], only 21% of Makana residents have access to potable water and only 35% to sanitation [131].

The municipality was recently ranked as the third worst performing municipality in the Eastern Cape, with 2,507 instances of misspending (R19.8 million), making up 9% of the total misspending in the province [141, 142]. The supreme audit institution has repeatedly found the municipality unable to adequately account for the use of public resources. In 2011/2012, the Auditor-General could not obtain sufficient appropriate audit evidence to support over R48 million of municipal expenditure [144]. This is the fourth successive disclaimer that Makana has received [143]. In August 2014 Makana Municipality was placed under provincial administration due to ongoing infrastructural and financial crises [145].

The situation in Makana Municipality is comparable to other local municipalities across South Africa, where "local government capacities are in short supply and financial sustainability is frequently in doubt. This hampers total government ability to perform traditional functions such as service delivery and regulation, collecting rates, user charges and fees" [148, p.137]. However, this is not a problem that is unique to South Africa, but has been demonstrated on numerous occasions and across a number of different contexts. A recent country wide, empirical evaluation found similar evidence across developing countries around the world [9].

1.3 Problem statement

The modern day mobile environment is diverse, with installable applications often targeting only a few specific platforms, thereby limiting their reach and perceived value. The research presented in this thesis aims to investigate a variety of techniques to allow client applications to be developed across heterogeneous mobile devices and platforms, thus providing access to the MobiSAM service. Particular attention is given to following known platform user interface and user experience guidelines in order to make the client applications as familiar as possible. Methods for potentially lowering the barrier of entry to using the MobiSAM service are presented, including: providing multi-language support; minimising client application data usage; as well as maximising mobile platform support (including low-end feature phones). In addition, the appropriateness of different navigation styles are explored, as well as how contrasting visualisations assist users to better understand and draw benefit from collated poll results. The effect of interactivity within these visualisations is also evaluated. In order to provide near universal access to the MobiSAM service, the ability for residents to report service delivery complaints via traditional SMS is also researched and developed.

1.4 Research goals

The goals and objectives of this research project are to:

1. Determine the current state of the art in the fields of e-government, m-government and ICT-facilitated civic engagement projects, as well as to investigate different visualisation principles, techniques and tools. Particular attention should be paid to the differences between traditional and mobile visualisations.

2. Perform an informal mobile phone retail outlet survey, as well as a comprehensive baseline study in an attempt to gain an understanding of how Grahamstown residents currently use mobile technology, and participate with local government around the area of service delivery.

3. Design and implement two platform-specific prototype client applications, enabling access to the MobiSAM service. Next, investigate and evaluate a variety of cross-platform mobile frameworks in an attempt to determine the most suitable means of targeting heterogeneous mobile devices and platforms. Leveraging a cross-platform framework, develop a client application, allowing residents to answer service delivery polls, as well as view and interact with collated poll results. Support need be extended to the most prevalent mobile platforms within Grahamstown. Develop a second cross-platform client application, streamlining the process of reporting *specific*, detailed service delivery issues, as opposed to answering service delivery polls. Visualisations need not be included. In addition, develop an SMS gateway allowing residents to report service delivery issues to their municipality via traditional SMS.

4. Briefly evaluate the prototype client applications. Conduct a thorough usability evaluation using the initial cross-platform client application, in order to determine how intuitively participants are able to navigate within the client application, as well as understand the different visualisation techniques employed. The effect of interactivity within the chart visualisations need also be assessed. Finally, conduct a comparative usability evaluation using the two cross-platform client applications, in an attempt to determine which version best meets participant needs and expectations, according to the usability criteria: learnability, memorability and satisfaction.

It should be noted that these goals are not strictly performed in the order in which they were presented.

1.5 Delimitation of scope

The research presented in this thesis forms part of a broader study, piloting the MobiSAM service within Grahamstown in collaboration with Makana Municipality. The project as a whole, aims to determine whether mobile phones offer a viable platform to promote and increase meaningful citizen participation within local government matters. However, only efforts relating to the client portion of the MobiSAM service are presented here. As the MobiSAM website and backend server were developed by a separate researcher, only basic information surrounding their use is provided (see Section 5.3 and Appendix A.1). It should be noted that the research conducted as part of this thesis does not evaluate whether MobiSAM is able to measurably increase citizen participation with local government. Instead, this research question has been designated as future work, to be evaluated as the pilot study progresses. In addition to this, comparisons between the mobile-friendly website and MobiSAM client applications are not presented. Due to ongoing concerns surrounding Grahamstown's potable water supply, the MobiSAM and Makana Municipality collaboration initially focussed on the monitoring of this basic service. Programming languages used when conducting the research presented in this thesis are limited to Java and C#, due to the researcher's proficiency therewith.

1.6 Research methodology

A user-centered approach was followed throughout the research presented in this thesis. By consulting users throughout the development process, they were able to influence and shape each application's design, potentially leading to more usable and satisfying solutions [2]. More specifically, the interaction design methodology was chosen, which encourages iteration, and consists of four activities: identify needs/establish requirements, (re)design, build interactive version and evaluate. Three complete iterations of the methodology performed while conducting the research.

1.7 Thesis structure

This thesis has been divided into eight separate chapters, with each focusing on a specific aspect of the research project.

Chapter Two investigates and discusses literature relating to the use of information and communication technologies (ICTs) in the public sector; explores a number of civic engagement projects using these technologies; as well as details a number of mobile technologies relevant to the MobiSAM project (including visualisation principles and techniques, as well as mobile adoptions thereof).

Chapter Three provides an overview of the methodology followed when conducting this research project. The interaction design life-cycle model and its use within the context of the project is introduced and expanded upon. The informal retail outlet and usability evaluations are also discussed.

Chapter Four presents the baseline study which attempts to empirically investigate how local residents use mobile technology and participate with local government around the area of service delivery. Study results are subsequently used during the requirements gathering process of the next iteration of the life-cycle model.

Chapter Five explores the development of two cross-platform client applications: MobiSAM and MobiSAM Report. The design and implementation of the SMS gateway application is also described here.

Chapter Six presents the results of the first usability evaluation, performed using the cross-platform MobiSAM client application. Shortcomings relating to navigation and visualisation are highlighted, along with miscellaneous usability concerns.

Chapter Seven presents the results of the comparative usability evaluation, performed using MobiSAM and MobiSAM Report cross-platform client applications. Results are analysed and discussed according to the usability goals: learnability, memorability and satisfaction.

Chapter Eight draws together the findings of the research project, revisiting and evaluating the project goals, bringing it to conclusion. Avenues for future research are also discussed here.

Chapter 2

Literature review

2.1 Introduction

This chapter investigates literature relating to the use of ICTs in the public sector (Section 2.2); explores and describes several local and international ICT-facilitated civic engagement projects (Section 2.3); and details a selection of mobile technologies relevant to the MobiSAM service (Section 2.5). Visualisation principles (Section 2.8.1), techniques (Section 2.8.2) and tools (Section 2.8.3) are then discussed. Finally, the differences between mobile and traditional visualisations are highlighted (Section 2.8.1).

2.2 ICTs in the Public Sector

The potential for ICTs to positively effect government was realised more than two decades ago with the foundation of the International Federation of Computer Science (IFIP) working group, Information Systems in Public Administration (ISPA) [96]. In his book titled *Reinventing Government in the Information Age*, Heeks identifies the effects of ICT-facilitated reforms in the public sector [83]. These include increased efficiency, more effective decentralised decision making, an increase in accountability information, as well as providing a "conduit for delivery of new forms of public service" [83, p.19]. Similar sentiments were echoed by the former President of South Africa, Thabo Mbeki, during his 2005 World Summit statement in which he maintained "we [the government of South Africa] have fully recognized the critical importance of modern ICTs as a powerful ally we have to mobilize" [24, p.4]. In their most basic form, ICTs allow governments to move away from paper-based records, instead digitising data, allowing for its easy storage and retrieval. In addition, advancements in mobile technology provide an ideal opportunity for governments to strengthen and improve their relationship with the public, using mobile devices as a potential tool for increasing transparency and improving service delivery [164].

As described in upcoming sections, the potential for ICTs to assist government operations is expansive, taking a variety of different forms. This section provides an overview of electronic government (e-government) and mobile government (m-government), as well as the different forms of m-government.

2.2.1 E-government

Considered by some to be "one of the oldest forms of ICT for accountability applications" [1, p.170], e-government is defined as "the use of ICT and its application by the government for the provision of information and public services to the people" [208, p.14]. Seen as a paradigm shift over traditional public administration approaches [149], e-government aims to provide an efficiently managed government, improved service delivery and an empowered electorate by streamlining internal processes through the use of ICTs [152].

E-government covers a variety of different activities and is categorised into four distinct areas, based on the segment served: government-to-government (G2G), government-to-citizen (G2C), government-to-business (G2B) and intra-government internal efficiency and effectiveness (IEE) [149]. G2C concerns itself with facilitating citizen interaction with government departments, a primary goal of e-government. G2B covers the procurement of goods and services by government. While G2G, argued by Nkwe to be "the backbone of e-government" [157, p.40], enables the sharing of data between the various central, provincial and local government departments. IEE differs from G2G in that it focuses on increasing the efficiency of back-office systems by coordinating available resources [6].

While the ideals of e-government promise several benefits when properly deployed, Schuppan argues that one has to be cautious when implementing such services within developing countries, stressing that "additional effort is necessary" [191, p.118]. Ndou maintains a similar stance, drawing attention to the fact that "there remain many challenges which hamper the exploration and exploitation of its [ICTs] opportunities" [152, p.12]. In addition, the popular press have expressed criticism for the pace at which South Africa's own e-government services are being rolled-out, highlighting "the country has slipped a whopping 36 places over two years [between 2008 and 2010] in the UN e-government index" [10, p.1]. Heeks emphasises the importance of implementing e-government systems which are country specific and not merely adaptations of existing systems used by other countries [84]. Furthermore, suitable leadership and input from key stakeholders has been identified as vital to successfully implemented e-government services [123].

2.2.2 M-government

Mobile phones are the most rapidly adopted technology in history [168]. By the end of 2014 the number of mobile cellular subscriptions worldwide was predicted to surpass 7 billion, while the African continent was forecast to reach a mobile penetration rate of 69% [98], up from 9% only a decade earlier [75].

M-government is defined as "strategy and its implementation involving the utilization of all wireless and mobile technology, services, applications and devices for improving benefits to the parties involved in e-government including citizens, businesses and all government units" [110, p.3]. It is often seen as both a subset and a natural extension of e-government [108,207]. While traditional e-government focuses on providing services to citizens via fixed Internet access methods such as landline telephone, m-government attempts to free users from geographical constraints by bringing mobility to e-government processes [164]. In theory, m-government allows both citizens and officials access to government information and services 24 hours a day, 7 days a week from any location within mobile network coverage [108].

To this end, m-government is seen as being particularly well suited to developing countries due to the high ratio of mobile subscriptions to fixed Internet services [98], thus affording access to citizens who may otherwise have been excluded [108]. While some have questioned whether m-government will replace the role of e-government, it should be noted that m-government services are by no means intended to substitute existing forms of online or offline service delivery [108]. Instead "[m-government] affords powerful and transformational capacity to the public sector" by both increasing access to existing services as well as allowing for the development and roll-out of new services [168, p.12]. In addition, m-government serves as an ideal platform which may be leveraged for citizen activism [108].

Similarly to e-government services, four distinct categories of m-government have been identified: G2G, G2C, G2B and IEE [79]. In their report titled *Making Government Mobile*, Raja, Cruse, Goldstein, Maher, Minges and Surya have further identified three forms of m-government, each of which is elaborated upon next [176]. It is important to note that instead of being rolled-out nationwide, South Africa's m-government services instead rely on the initiative of individual state departments and local municipalities to develop strategies for their implementation [27].

2.2.3 Forms of m-government

M-government services are usually deployed in a variety of different forms, with the aim of improving accountability, transparency and responsiveness to their electorate [176]. Three forms of m-government have been identified and will be described next.

Supplement The first form may be defined as mobile tools which include an additional channel to supplement the way in which government services are currently provided. An example of this would be in Korea, where the e-government services have been expanded by adding wireless portals and interfaces to access existing e-services [176].

Expand The second form helps to extend the reach of traditional government services, so as to include previously under served citizens. HealthLine, a telephonic service offering medical advice to citizens in Bangladesh, provides an example of expanding traditional e-government reach [176]. The service reduces the need to travel to clinics and waiting in health centre queues.

Innovate The third and final form are mobile tools which are used to "develop new services for service delivery and governance" [176, p.88]. The MobiSAM system falls into the third category, aiming to enable innovative ways for local government to involve and interact with citizens.

2.3 Local civic engagement projects

Several civic engagement projects leveraging ICTs such as mobile phones have been deployed within South Africa over the past decade. This section draws together some of these projects, detailing their purpose as well as their perceived successes and shortcomings. Three pilot studies conducted by Information for Community Oriented Municipal Services (iCOMMS) are first introduced, followed by local SMS hotlines, Mxit applications, Lungisa and City of Johannesburg's Find & Fix mobile application.

2.3.1 Information for Community Oriented Municipal Services (iCOMMS) Research

iCOMMS is an interdisciplinary research team based at the University of Cape Town's Civic Engineering faculty, whose primary aim is to "improve service delivery in developing countries through the use of appropriate tools" [95, p.1]. The group's research involves developing and evaluating systems based on the innovative use of existing technologies (such as mobile phones and other ICTs) which aim to "support basic service delivery in rural and under-serviced environments" [95, p.1]. Three of the group's recent projects are briefly discussed next.

Monitoring drinking water quality in South Africa

Forming part of the European Union-funded Aquatest project, this study investigated the use of a mobile phone-based information system for collecting water quality information [183]. The researchers maintained that the information systems currently implemented to monitor drinking water quality in South Africa are ill-suited for use by small and under-funded municipalities, instead being designed for more affluent municipalities "who have the means to implement policy and best practices" [183, p.24].

The project was based in rural South Africa and focused on a case study spanning four field sites, with each case focusing on a single water services authority [183]. Two mobile phone applications were developed: Water Quality Reporter (WQR) and Water Quality Manager (WQM), each to be used for data collection, information transfer and analysis purposes (see Figure 2.1). The WQR application was installed on a water-supply caretaker's handset, allowing them to answer questions and input parameters relating to the current quality of the water supply. Results were then uploaded to a central server via GPRS.

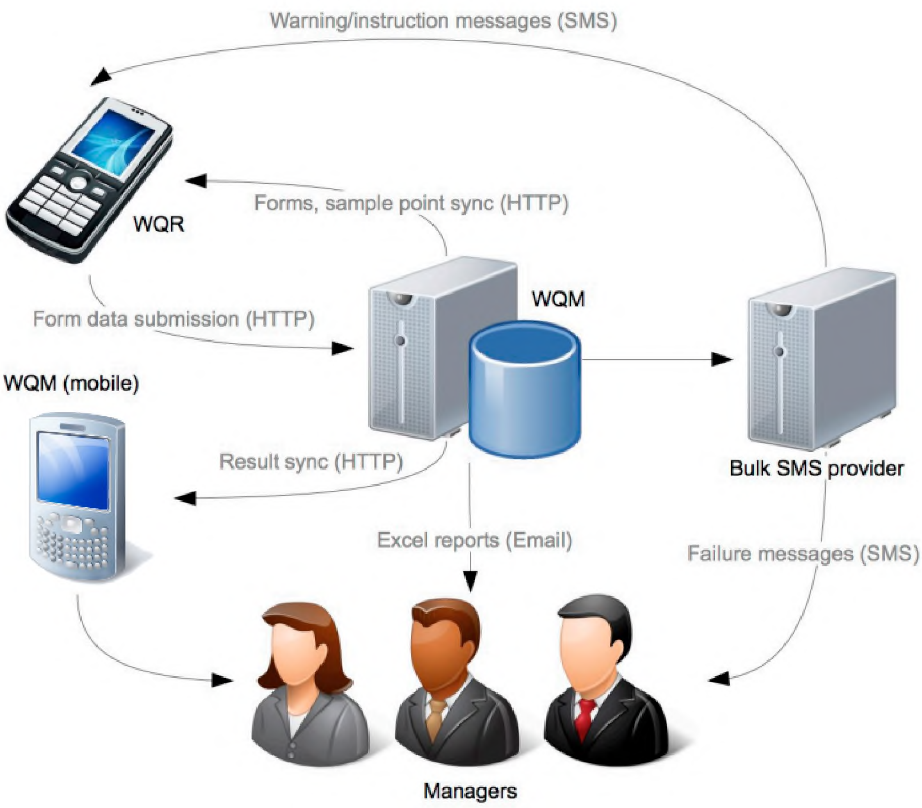


Figure 2.1: The WQR and WQM system architecture (taken from [183])

During the second phase of the project, the WQM application was installed on handsets which were used by water-quality monitoring managers. The application allowed managers to review the water quality status of their water sites at any time. Managers also received notifications alerting them when results submitted by a water-supply caretaker were outside accepted parameters. The researchers were very careful to design the system according to local needs, making sure the mobile application designs mirrored existing municipal work flows.

Ultimately, the study found that "all managers reported an increase in regular communication and an increase in awareness of the status quo of the water quality in the various field sites" [183, p.29]. As managers were able to assess sites from any location, the amount of time spent travelling to sites was reduced [183]. Managers also indicated that their confidence in matters relating to water quality had improved [183]. In addition, the movements and needs of water-supply caretakers were better understood by their managers. Furthermore, managers and caretakers both confirmed that the system helped them experience an increase in transparency and accountability [183].

'Drop drop'

Drop drop is an Android application which aims to "empower citizens in their position in the water system" [179, p.124] by attempting to help them better understand their water usage habits. The application provides an area for users to input their daily water usage, as read from their meter. Water usage history is presented as a bar chart, along with a textual description of the results. In addition, citizens are presented with educational information relating to the water system, such as how to read their water meter, allowing them to better track their own water consumption. Additional features such as 'Detect a leak' and 'Check my bill' are also included and take the form of 'wizards' which help users perform these activities. In an attempt to meet user requirements as closely as possible, an iterative process and user-centred design approach was followed during development [179].

A two week pilot study revealed that although participants were able to navigate Drop drop confidently, they encountered problems reading their water meters (primarily due the meter's analogue display) as well as entering readings into the application. Two participants did however detect a leak using the application. The researchers concluded that although their data was biased due to the small, demographically homogeneous study sample comprising of four female participants, "we can at least say that the engagement with a phone provides a platform to learn about water usage" [179, p.127]. Several enhancements such as improving meter reading entry, offering multiple languages and removing unused information were cited as possible future work.

WATER Alert!

In this study, iCOMMs researchers attempted to address the challenge of "reporting complex and critical water quality information in a way that is accessible to all South Africans" through the use of a symbol-based mobile application [22, p.1]. Currently, should citizens require water-quality information they need to request a paper-based report from their water service provider (WSP) which is often both difficult to interpret and understand. The WATER Alert! application attempted to simplify this information by issuing "locally relevant symbol-based message alerts and information to subscribers who can then forward the messages [...] to non-subscribers" [22, p.1].

Mobile phones were chosen as the preferred platform due to their "availability, widespread popularity and familiarity [...] as well as the speed, reach, lower associated costs of use and simplicity of use" [22, p.1]. Once a user has installed the application and subscribed to the service the application displays the current water quality status for their municipality or area of interest (see Figure 2.2). The user may then either select 'Advice' which offers tips relevant to the current status of their water (shown as a series of picture messages) or 'Report' which provides access to the detailed water quality report. Water quality reports are simplified through the use of a calendar metaphor and appropriate colour codes, thereby making them more understandable and better suited for display on mobile devices. The system was evaluated by five participants (two male and three female) each between the age of 18 and 45. During the evaluation participants were given a set of tasks to complete using a conventional paper-based water quality report followed by a similar set of tasks using the WATER Alert! application.

Figure 2.2: Current water quality status for Mandalay Municipality (taken from [22])

The researchers found that three out of five participants experienced an increased understanding of water quality information, one participant experienced a decrease while another experienced no change. In addition, participants were shown to complete tasks in much less time when compared to traditional paper-based reporting. All five participants correctly interpreted the symbol-based alert status, potential danger and advice given, with the researchers suggesting that the "use of visual metaphors in our prototype design contributed to the effectiveness of the application and empowered users" [22, p.8]. As future work researchers suggested implementing a feature allowing consumers to provide water quality feedback to their WSPs, thereby helping suppliers "satisfy their legal obligation to provide safe drinking water" [22, p.8].

2.3.2 SMS hotlines

Several SMS hotlines have been piloted by government departments within South Africa, allowing citizens to submit complaints via traditional SMS. Two such hotlines are described next.

During 2012, the Western Cape Department of Health launched an SMS hotline allowing citizens to lodge complaints about poor service at public healthcare facilities. The service was piloted for a period of three months during which time it proved successful, and according to the Western Cape government "will be rolled out across all facilities going forward" [71, p.1]. During the pilot study a Cape Town-based newspaper tested the newly launched service, receiving a response from call centre agents within 30 minutes. Several patients, however, were reportedly weary of using the service out of fear of being victimised by hospital staff [50].

A similar service, albeit targeting corruption, was deployed within the City of Johannesburg Metropolitan Municipality during the same year. The SMS hotline, which launched after a critical report by the watchdog organisation Corruption Watch[1], allows citizens to report fraudulent and corrupt behaviour within the Johannesburg Metro Police Department. It is envisioned that the service will help assist the city in taking disciplinary action against offending police officers. The service is also reachable via toll free voice call and all tip-offs are treated as confidential [187]. Results relating to the effectiveness of the system have yet to be released.

2.3.3 Mxit applications

Mxit is a locally developed mobile instant-messaging platform which initially became popular as a substitute for traditional SMS. As Mxit sends and receives messages using packet-data, the cost per message is significantly less than that of SMS. Several political campaigns have leveraged the platform in an attempt to engage with South Africa's youth. Two such campaigns are presented next.

Launched in May 2013, the African National Congress's MyANC Mxit application aims to improve the party's engagement with citizens [154]. The application, which primarily targets youth, has been downloaded over 717,000 times and as of November 2014 boasts over 430,000 active users, recently earning the ANC an award for "creating the largest political app on the social networking site Mxit" [196, p.1]. In the months preceding the 2014 National Elections, the Democratic Alliance launched a Mxit-based strategy game aptly titled DemocraCITY. The game urged citizens to make strategic decisions relating to issues of governance, showing them how these decisions influenced the happiness of citizens within the game [173].

Interestingly, Mxit had until recently banned politically motivated campaigns from its platform, with the company maintaining that it "did not want to be seen to endorse any particular

[1]Corruption Watch http://www.corruptionwatch.org.za/

party's policies" as Mxit was "purely an application for communication" [104, p.124]. Katz in his book, *Mobile Communication: Dimensions of Social Policy*, expressed surprise over Mxit's initial stance, and argued that "by excluding politically themed chats and user-generated content, Mxit's content policy curtailed opportunities for public participation on the most popular 'new media' platform in the nation" [104, p.124-125].

2.3.4 Lungisa

Lungisa, an isiXhosa word meaning "fix it", is a project which "aims to make it easy for ordinary people to raise publicly, concerns they have about service delivery problems" [127, p.1]. Run by Cell-Life, the service is a free community monitoring and reporting tool, allowing the public to submit reports via SMS, USSD, e-mail, Facebook, Twitter and Mxit. Lungisa acts as an intermediary, forwarding reports to the correct municipal contacts or partners, along with the contact details of the individual reporting the complaint. Feedback is usually provided within 48 hours, although as Lungisa is not a government operated service, they cannot guarantee that all reports will be attended to in a timely manner. Instead, Lungisa promise to "follow up persistently for you with the relevant government officials to make sure the problem you have identified is fixed" [125, p.1]. The service is currently available to residents of certain townships within the City of Cape Town, however, is soon said to be available elsewhere. As of November 2014 there had been 1,596 issues reported and 1,573 issues resolved, representing a fix rate of 98.6%. A breakdown of report types reveals that 'Sanitation' represents 16.8%; 'Electricity' represents 14.5%; 'Water' represents 12%; while 'Other' accounts for 31.8% of all received complaints [126]. No information is provided as to how quickly reported issues were resolved.

2.3.5 Find & Fix

Find & Fix (F&F) was launched in May 2014 by the Johannesburg Roads Agency (JRA) and allows citizens to report issues relating to the city's road network, such as potholes, traffic light outages and missing signs [90, 206]. The service aims to improve service delivery by allowing the JRA to obtain more timely, accurate information relating to reported problems. This, they believe in turn, "will have an improved effect on service delivery over time" [101, p.1]. Classified as an IEE service (see Section 2.2.2), F&F forms part of internal efforts to streamline its operations as well as reduce the administrative burden of dispatching repair crews [169]. Reports are submitted using the F&F mobile application, however, support is limited to Android, iOS and Windows Phone platforms [101], thus excluding citizens with low-end handsets. Although the service is free, the application makes use of packet-data. In addition, citizens are required to register before first time use. In the five months following its launch, 5,728 citizens had signed up to use the service, reporting a total of 9,749 issues [129]. As with Lungisa, no information is provided as to how quickly issues are resolved, although initial feedback has been positive, with a journalist claiming "anecdotally I can say that the pothole outside my house, which stayed open for weeks, was fixed within 24 hours" [169, p.1].

2.4 International civic engagement projects

This section presents an overview of several ICT-based civic engagement projects deployed abroad. The Arab Spring uprisings are first introduced (Section 2.4.1), followed by TXT CSC (Section 2.4.2), FrontlineSMS (Section 2.4.3), Ushahidi (Section 2.4.4) and SeeClickFix (Section 2.4.5).

2.4.1 The Arab Spring

While not considered a civic engagement project per se, the Arab Spring is the term given to the mass of protests that swept through the Middle East during early 2011, ultimately resulting in the fall of four of the region's most persistent dictators [92]. Each fell from power after unprecedented levels of protest, facilitated through the use of ICTs and social media networks, called for their dismissal. The democracy movement which began in Tunisia resonated with others across the region, soon inspiring Egypt, Libya and Yemen to follow their lead.

Traditional political actors, such as opposition parties and unions, did not spark the Arab Spring protests, instead it was young business owners, the urban middle classes, women's groups and many others who had long been considered unsuccessful at traditional political organisation [92]. According to Howard and Hussain, "the groups that initiated and sustained protests had few meaningful experiences with public deliberation or voting and little experience with successful protests" [92, p.3]. Importantly, however, they were "politically disciplined, pragmatic and collaborative" [92, p.3].

Over the past decade access to ICTs has become relatively widespread throughout Egypt, which is now home to more than 17 million Internet users and 4 million Facebook users [201]. Among these are an estimated 160,000 bloggers, a third of which are politically focused [201]. ICTs and social media are said to have played two critical yet overlapping roles in facilitating the Arab Spring uprisings. Firstly, these platforms were leveraged for organisational purposes, such as arranging protests. Secondly, they offered an ideal medium through which to disseminate information, particularly to international audiences where protesters' demands were published [201]. According to a March 2011 study, nine out of 10 Tunisian and Egyptian citizens used Facebook to create awareness and organise protests [93], proving that "information and communication networks can serve as powerful accelerators of social transformation" [201, p.2].

The Egyptian government's attempt to curb the use of ICTs and social media was considered "unprecedented in Internet history" [201, p.2]. After having blocked both Facebook and Twitter (to little effect), the Egyptian authorities then ordered telecoms providers to block Internet access. As a result, 93% of the country's networks and Internet addresses were shut down [201]. In response, citizens implemented workarounds such as the use of proxy servers, virtual private networks (VPNs) and Google's Voice-to-Twitter applications [201], thereby allowing them to maximise use of the remaining Internet capacity. Although the mainstream media focused on micro-blogging as the driver behind the Arab Spring uprisings (calling them the "Twitter revolutions"), Stepanova argues that its role has been overemphasised, especially when compared to ICTs such as mobile phones, video sharing sites (e.g. YouTube) and satellite television [201]. Instead, ICTs collectively provided a medium through which citizens came to realise their shared grievances, ultimately leading to political change.

2.4.2 TXT CSC

The Philippines, often referred to as "the text messaging capital of the world" [121, p.85], provides many examples of m-government services, several of which leverage the powerful simplicity of SMS. TXT CSC is a service launched by the country's Civil Service Commission (CSC) which aims to monitor and improve the quality of government service delivery to the public [175]. Corruption, poor service quality or inappropriate civil servant behaviour may all be reported using SMS and will be dealt with appropriately, according to the CSC [163]. The commission is obliged to reply to queries within 24 hours and usually responds with a personalised SMS [163]. TXT CSC has also been used to conduct public service delivery audits, whereby citizens use the service to rate their satisfaction of public services [111]. The low cost of using TXT CSC as well as the convenience it provides are cited as the primary reasons for its success [56].

2.4.3 FrontlineSMS

Launched in 2005, FrontlineSMS is an open source application which allows any desktop computer to be turned into an SMS-based communication hub. The software was primarily developed for use by non-governmental organisations (NGOs) as a means to "lower barriers to positive social change using mobile technology" [53, p.1] by helping these organisations bypass potential communication obstacles. Once installed, contacts are added to create groups, which the application is then able to broadcast messages to and receive replies from, all via traditional SMS [174]. FrontlineSMS does not require an Internet connection and instead makes use of a Global System for Mobile Communications (GSM) modem. In the event that the host computer has Internet access, FrontlineSMS is able to leverage online bulk SMS providers, thereby further reducing costs.

FrontlineSMS was originally developed to help conservationists in South Africa's Kruger National Park to engage and communicate with surrounding communities [174]. However, it has since been used to assist the spread of information relating to malaria in Cambodia [100], provide weather information to subsistence farmers in Zambia [150], help protect children against violence in Benin as well as enhance the monitoring of national elections in Nigeria, Afghanistan and the Philippines [12,46,177]. More recently (September 2014) the application has been used to collect and disseminate information relating to the Ebola outbreaks in West Africa, informing the public about what the disease is, how to avoid it and how to prevent it spreading [58].

2.4.4 Ushahidi

During the Kenyan election crisis of 2007, a media blackout meant that both local citizens and the outside world were unable to receive information about events unfolding on the ground. Ushahidi, a website created by a group of Kenyan activists, was developed in response and allowed citizens to submit (and receive) news reports via the Internet and SMS. This citizen journalism, made possible by ICTs and mobile phones, helped create an up-to-date overview of noteworthy events as well as a "time-indexed repository of reports" [166, p.59]. In addition, the service allowed reports to be geographically mapped in real-time, helping civilians avoid areas of conflict. While citizen reporting has the potential to be abused, Okolloh argues that "with enough volume, a 'truth' emerges that diminishes any false reports" [166, p.59]. Furthermore, the service allowed reports to be screened before being made public, hence reducing the likelihood of propaganda.

Since its initial deployment in Kenya and subsequent use in South Africa during the xenophobic violence in May 2008 [87], Ushahdi has been redesigned, enhancing its utility within humanitarian crisis situations [166]. In October 2008 the alpha version of the platform was launched in the Democratic Republic of Congo (DRC), in an attempt to cover the crisis unfolding in the eastern part of the country. While Ushahdi DRC received a large amount of international press coverage [40,134], its use by citizens was lower than anticipated [166]. The Ushahdi service has since been used by many other countries for disaster relief efforts [59,139,195]. Examples include: assisting post-earthquake rescue efforts in Chile and Haiti [47,88]; helping prevent and manage forest fires in Italy [138]; as well as visualising the effects of the 2011 floods in Queensland, Australia [186].

2.4.5 SeeClickFix

SeeClickFix (SCF) is a U.S.-based service which allows residents to report non-emergency concerns to their local government via smartphone application (supporting Android, iOS and Windows Phone) or the SCF website [192]. Launched in 2008, the primary goal of the service is simply: "to alert governments of issues that need to be fixed in the community" [140, p.3]. Traditionally, telephone and e-mail correspondence are used when reporting problems to local municipalities within the United States. While municipalities recognise these forms of communication, residents seldom receive acknowledgement of their reports, or whether they are being attended to [140]. This disconnect between the public and local municipalities regularly results in one-directional reporting, with the absence of any feedback mechanisms [140]. SCF attempts to address this by depicting reported issues in a transparent way, using a Google Maps mash-up, hopefully encouraging local governments to increase their accountability and responsiveness [140]. Residents are automatically informed about progress changes relating to their reports, thus enabling bidirectional reporting and “allowing for a full feedback cycle” [140, p.2]. SCF offers the delivery of basic e-mail reports at no cost to municipalities. More than that, municipalities are required to pay a monthly fee (from US$40 to US$400) depending on their subscription type.

SCF has been criticised for initially directing all of its attention towards citizens and none towards collaborating with municipalities [140]. The service was launched across the United States without prior knowledge of local governments, with the developers instead harvesting 16,000 email address from municipal websites. Citizen reports were then sent to these e-mail addresses [140]. A key problem to this approach was that:

> “In some localities, the report e-mails go to e-mail accounts that are not monitored, are ignored or marked as spam. The result is that citizens perceive their local governments as unresponsive and in turn public servants were surprised once they heard about their citizen complaints.” [140, p.15]

Some municipalities even went so far as to see it as a ‘guerrilla tactic’ which purposefully tried to make them look incompetent. Recent studies have highlighted government buy-in as a critical success factor for successfully implementing ICT related projects [27], thus helping avoid an ‘us versus them’ scenario.

2.5 Mobile technologies

The civic engagement projects described in Section 2.3 and Section 2.4 derive benefit from a variety of different ICTs. This section provides an overview of some of these technologies, including SMS, SMS gateways, unstructured supplementary service data (USSD), representational state transfer (REST) and JavaScript object notation (JSON).

2.5.1 Short Message Service (SMS)

Until recently (2006), SMS was the most widely used form of electronic communication [8]. The service is supported by all GSM handsets and most mobile networks worldwide, allowing subscribers to send and receive alpha-numeric messages of up to 160 characters in length [118]. In its most basic form, the architecture of an SMS-enabled mobile network consists of two additional elements: an SMS centre (SMSC) and an e-mail gateway. In addition, a short message entity (SME), usually taking the form of an application running on a mobile handset, is required for sending, receiving and storing messages. The primary role of the SMSC is to relay messages between SMEs as well as storing undelivered messages until offline SMEs become available (or the message's validity period expires). The e-mail gateway converts messages from SMS to e-mail and vice versa, as well as acting as an intermediary between SMS and Internet domains [118].

A considerable amount of literature describes investigations into how and why people make use of SMS [11,42,73,74,80,189,205]. Proponents of the service cite its low cost, ease of use, relative confidentiality, delivery confirmations and immediate dissemination of information as its major advantages [8,74]. Additionally, the inclusion of SMS support into even the most basic handsets has helped position it as the *de facto* messaging standard, making it an important communication channel for civic engagement projects to leverage. Although the total number of SMS messages sent worldwide during 2013 declined for the first time since its inception, a noteworthy 7.7 trillion messages were predicted to be sent during last year (2014) alone [55].

2.5.2 Short Message Service (SMS) gateways

Considered by some to be the quickest and most reliable way to send a large volume of text messages [103], an SMS gateway allows a computer coupled with a GSM modem to send and receive SMS messages via a mobile network. Additionally, SMS gateway services are available through Internet-based bulk SMS providers, who deliver messages on behalf of their customers. During the transmission of an SMS, the message is first relayed from an SMS gateway to the mobile operator's SMSC. From here the message is forwarded on to its destination SME. Should the destination mobile subscriber be unavailable (for example, their mobile phone is switched-off or has no network reception), the SMSC will store the message until the subscriber is available or until the message's validity period expires, thereby providing a 'store and forward' mechanism. SMS gateway providers are classified as either aggregator or Signalling System No. 7 (SS7) providers. Aggregator providers do not have direct access into the SS7 protocol (the protocol whereby SMS messages are exchanged) and therefore have limited visibility and control over message delivery [103]. SS7 providers are afforded direct access, and are therefore able to offer message delivery guarantees [103]. A selection of open and closed source SMS gateway solutions exist, two of which are briefly described next.

Kannel Kannel is an open source[2] Wireless Application Protocol (WAP) and SMS gateway for GSM networks [102]. The project is compatible with a variety of different GSM modems[3] and

[2]Kannel Repository http://redmine.kannel.org/projects/kannel/repository
[3]Kannel Compatibility http://www.kannel.org/compatibility.shtml

allows the details of each SMSC protocol to be abstracted, thereby simplifying the deployment of SMS-based services [48]. Kannel only offers support to Linux operating systems, however, the source code may be complied to run on Windows 9x/NT systems using Cygwin tools[4].

MOBITEK The MOBITEK SMS gateway is a proprietary combination of hardware and software developed by the Taiwanese company MOBITEK System[5]. The MOBITEK S63 GSM modem (pictured in Figure 2.3) comes bundled with an SMS gateway development kit, allowing developers to create SMS gateway services or integrate SMS functionality into existing applications [147]. The MOBITEK SMS Application Programming Interface (API) enables access to the modem's functionality such as initialisation, sending SMS messages and disconnecting from the mobile network. As the API is provided as a dynamic-link library (DLL) file, developers may access it using any programming language which is compatible with Microsoft's Component Object Model (COM) binary-interface standard [147]. Only Microsoft's Windows XP operating system and above are currently supported.

Figure 2.3: MOBITEK S63 GSM modem

2.5.3 Unstructured Supplementary Service Data (USSD)

Unstructured Supplementary Service Data (USSD) is a protocol implemented by GSM networks which allows mobile subscribers to communicate with their network operator's computer systems. While traditional SMS messages are limited to 160 characters, USSD messages may contain up to 182 characters and as with SMS, are handset independent. Unlike SMS messages, a real-time connection is created between the subscriber's handset and mobile network when sending USSD messages, allowing for a more interactive experience. To this end, USSD is known as a session-based protocol [178]. Sessions are usually charged at R0.20 per 20 seconds by network

[4] Cygwin http://www.cygwin.com/
[5] MOBITEK System http://www.mobitek-system.com/

operators within South Africa [72], making USSD unaffordable to some users. Nonetheless, USSD is leveraged by both Lungisa and Ushahidi projects [124, 128].

2.5.4 Representational state transfer (REST)

The concept of REST was first proposed by Roy Fielding in his Ph.D. dissertation and is described as a "way of judging architectures" [45, 182]. Several architectures are considered *RESTful* (and would score highly when judged according to Fielding's criteria), however, those of particular interest relate to the provision of web services or are resource orientated [182]. The REST architectural style is based on how the web works, making use of the Hypertext Transfer Protocol (HTTP) and Uniform Resource Identifies (URIs), with each unique URI representing an object (otherwise known as a resource) [62].

Interactions between clients and objects take place using HTTP verbs such as `GET` (to fetch an object), `POST` (to create an object), `PUT` (to update an object) or `DELETE` [62]. REST stipulates that developers should use HTTP verbs explicitly and as defined by RFC 2616 [184]. Similar to the HTTP protocol, *RESTful* web services are stateless and can therefore leverage HTTP cache and proxy servers, helping manage high traffic volumes and allowing improved scalability [62]. Although REST is not tied to any particular technology, a paper published by IBM titled *RESTful Web services: The basics* suggests that proper implementation of a *RESTful* web service should [184]:

- make explicit use of HTTP methods;
- be stateless;
- expose URIs as directory structure-like; and
- transfer XML and/or JavaScript Object Notation (JSON).

While traditional web services are considered to be "memory and processor intensive" [30, p.628], *RESTful* web services are lightweight, easy to invoke and quickly parsed [30], making them well suited to mobile environments. As *RESTful* web services leverage existing technologies, no specialised tools are required for development.

2.5.5 JavaScript object notation (JSON)

Described as the "fat-free alternative to XML" [34], JSON is a lightweight, human readable data-interchange format used to serve data between web applications and server [198]. Recent studies have shown JSON to offer significant performance improvements over data formats such as

XML [162], making it well suited to low-powered mobile devices [57]. In addition, JSON objects are significantly smaller than equivalent XML objects due to their terse structure (see Listing 1) and minimal operational overhead [162]. Furthermore, JSON is language independent, making it suitable for use across a wide variety of platforms [33]. Critics of the format, however, draw attention to its lack of `namespace` support, minimal input validation and limited extensibility as drawbacks [162]. Crockford, a programmer having popularised JSON, has since refuted these claims [34].

Listing 1 A simple JSON object describing the encoding of a name (adapted from [162])

```
{
        "firstname": "John",
        "lastname" : "Doe"
}
```

2.6 Mobile-optimised websites

With only 10% of South African households having fixed access Internet connectivity [199], many citizens are turning to their mobile phones for web access. Although mobile Internet access provides unprecedented flexibility and convenience, limited device resources coupled with expensive data tariffs present an array of challenges [216]. As most Internet content is designed for consumption using traditional desktop computers, it is ill-suited for older mobile handsets [216]. Although smartphones offer a much improved browsing experience when compared to their older counterparts, they only represented 30% of the installed user base within South Africa by 2015 [78].

The Wireless Application Protocol (WAP) 1.0 specification, introduced in April 1998, aimed to offer a solution to 'mobile-friendly' web content by providing a "single global standard for wireless data access for all hand-held mobile devices" [133, 171, p.98]. Ultimately, however, WAP failed to reach mainstream success with critics blaming its poor user experience (UX) [107, 109, 209], lack of end-to-end security [7], inappropriate revenue model [109] and flawed marketing approach [146]. One critic went so far as to label WAP as Wrong Approach to Portability [155]. As a result, mobile-optimised websites (leveraging existing web specifications and technologies) emerged, filling the role which WAP struggled to perform.

While Schmiedl, Seidl and Temper maintain that "the latest generation of mobile phones are reasonably suited to browse the full web" [190, p.71], they conclude that most users prefer and benefit from mobile-optimised web content. To this end, mobile-optimised websites are often provided alongside traditional websites, in an attempt to improve a user's browsing experience [190]. Of the civic engagement projects presented in Sections 2.3 and 2.4, only Ushahidi offers a mobile-optimised website, allowing users with low-end handsets to access the service and upload reports.

2.7 Mobile applications

Mobile applications, or apps, are installable native computer programmes designed to run on smartphones, tablets and other portable devices. They offer noteworthy advantages over traditional browser-based services, such as deeper integration with the host device and improved performance. However, often at the expense of platform support and increased implementation effort [210]. This section presents an overview of native mobile applications as well as cross-platform mobile frameworks.

2.7.1 Native applications

A multitude of different mobile platforms exist, each with their own set of APIs, design guidelines and implementation languages [210]. Native mobile applications inherently target a single platform, making them incompatible with other mobile operating systems. While these applications offer an unsurpassed overall user experience, they are often citied as being the most complex and expensive to develop [28, 210]. In addition, a diverse skill set is required to fully support the abundance of mobile platforms available today, as illustrated by Table 2.1. Differences amongst platform specific software development kits (SDKs) further adds to this complexity [28]. As an alternative, a variety of cross-platform frameworks exist which aim to abstract platform specific details and reduce the unnecessary duplication of work. A comparison of these frameworks is presented next.

Mobile platform	Skill set required
Apple iOS	Objective-C
Google Android	Java (Harmony flavoured)
RIM BlackBerry	Java
Java ME	J2ME
Windows Phone	C#

Table 2.1: Language skill set required (adapted from [28])

2.7.2 Cross-platform mobile frameworks

As reflected by Heitkötter, Hanschke, Majchrzak, "the market of mobile operating systems for smartphones is fragmented and rapidly changing" [86, p.120], requiring developers to redevelop the same application for each platform they wish to target [170]. Cross-platform frameworks aim to address this by allowing a single codebase to be compiled to run on multiple different platforms. Table 2.2 shows that available solutions are plentiful and while some merely act as a ‘wrapper’, running the application in a `Webview` component (accessing device hardware features programmatically through HTML5), others compile the codebase into multiple natively installable applications. The latter technique often results in improved, more fluid performance as applications are not hindered by the `Webview`’s rendering capabilities or JavaScript interpreter [28, 86].

In recent years there has been an increasing amount of literature covering cross-platform mobile development, however, studies have yet to identify a single ‘best’ solution [28, 81, 86, 165, 170]. An overview of three cross-platform mobile frameworks are presented next. These frameworks were chosen due to their expansive platform support and use of Java programming language, sidestepping the need for the researcher to learn platform specific languages.

Codename One

Codename One (CN1) is a collection of mobile application development tools used to simplify the often complex and fragmented task of targeting multiple mobile platforms. The framework is open source and allows developers to create native applications for Android, BlackBerry OS, iOS, Java ME and Windows Phone platforms, from a single shared codebase. Applications are written in standard Java 5 with user interface (UI) construction taking place in the CN1 designer or programmatically, using code [5]. An interactive simulator comes bundled with the CN1 framework and accurately emulates the supported platforms, allowing developers to rapidly prototype and debug applications. A default selection of device ‘skins’ are pre-installed with the simulator, providing an accurate preview of how the application will appear when running on a particular device/platform combination. Additional skins may be downloaded as new devices become available [5].

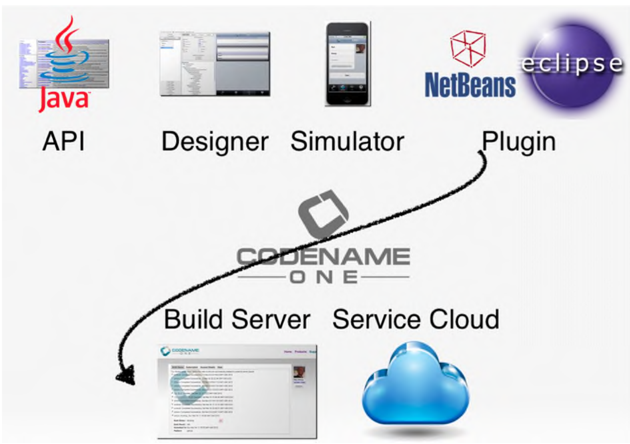


Figure 2.4: Codename One (CN1) cross-platform framework (taken from [5])

Framework	License	Programmed in	Supported platforms
Akula	Commercial	Visual	Android, BlackBerry, iOS, Windows Phone
Application Craft	Commercial	HTML, CSS, JavaScript	Android, BlackBerry 10, iOS, Windows Phone
Codename One	Open source and commercial	Java	Android, BlackBerry, iOS, Java ME, Windows Phone
Corona	Commercial	JavaScript	Android, iOS, Kindle, Windows Phone
J2ME Polish	Open source and commercial	Java ME, HTML, CSS	Android, BlackBerry, iOS, Java ME, Windows Phone
Flash Builder	Commercial	Flash	Android, BlackBerry Tablet OS, iOS
Feedhenry	Commercial	HTML, CSS, JavaScript	Android, iOS, HTML5, Windows Phone
Kony One	Commercial	HTML, CSS, JavaScript, RSS	Android, BlackBerry, iOS, Java ME, Windows Phone
LiveCode	Commercial	English-like	Android, iOS
M2Active	Commercial	Drag and Drop + Lua	Android, BlackBerry, iOS, Windows Phone
MobiForms	Commercial	Drag and Drop + MobiScript	Android, iOS, PC, Windows Mobile
MoSync	Open source and commercial	C/C++, HTML5/JavaScript	Android, BlackBerry, iOS, Java ME, Windows Phone
NeoMAD	Commercial	Java	Android, BlackBerry, iOS, Java ME, Windows Phone
Orubase	Commercial	ASP .NET	Android, iOS, Windows Phone
PhoneGap	Open source	HTML, CSS, JavaScript	Android, BlackBerry 10, iOS, Windows Phone
Qt	Open source and commercial	C++	Android, BlackBerry 10, iOS, Sailfish OS
Rhodes	Open source and commercial	Ruby, HTML, CSS, JavaScript	Android, BlackBerry, iOS, Windows Phone
Titanium	Open source	JavaScript	Android, iOS, Tizen, Mobile Web
trigger.io	Commercial	HTML5, JavaScript	Android, iOS, Windows Phone
webinos	Open source	JavaScript	Android, BlackBerry, iOS
webMethods	Commercial	Java ME	Android, BlackBerry, iOS, Windows Phone
Xamarin	Commercial	C#	Android, iOS, Windows Phone
XDK	Free	HTML, CSS, JavaScript	Android, iOS, Windows Phone, Windows RT

Table 2.2: Identified cross-platform mobile frameworks (adapted from [210])

CN1 is provided as an installable plug-in and supports NetBeans, Eclipse and IntelliJ integrated development environments (IDEs). Unlike traditional mobile application development, CN1 avoids the need of having to install multiple different platform SDKs, instead performing all application builds on the company's build-server (see Figure 2.4). The build-server produces platform-specific applications from the Java bytecode, ready to be installed on physical devices. While this approach may seem restrictive at first (as an Internet connection is required to perform builds), applications may be previewed offline, using the CN1 simulator [5]. Although the framework is open source and available at no cost, free users are limited to 100 builds per month. Should developers require more, a variety of different subscriptions are available ranging from US$19 to US$399 per month, per developer[6]. Support is provided via an active online discussion forum, while Pro and Enterprise subscribers in addition have access to e-mail support. Windows, Mac OS X and Linux operating systems are supported [5].

J2ME Polish

J2ME Polish comprises a collection of tools and technologies used for developing 'polished' J2ME applications. First released in 2004 by Enough Software, J2ME Polish is available under both GPL open source and commercial license [210]. The framework contains a variety of different modules, each focusing on a specific feature (*Lush*, for example, which allows developers to design attractive UIs outside of the application's source code, while *Touch*, enables access to server side content and communication with clients). The module responsible for transcoding J2ME applications to different platforms is known as *Janus*, and like CN1, allows developers to create Android, BlackBerry OS, iOS, Java ME and Windows Phone applications. *Janus* ports J2ME applications to Android and BlackBerry platforms automatically, while additional porting services (provided by Enough Software, for a fee) are required when targeting iOS and Windows Phone [41]. Since J2ME Polish is based on Ant, it is compatible with most popular IDEs. The framework is available for both NetBeans and Eclipse as installable plug-ins. Development is supported by any Java-enabled operating system and applications are written using a combination of Java, HTML and CSS [41]. No device simulator is provided.

NeoMAD

NeoMAD is a proprietary framework allowing developers to target a selection of mobile platforms from a single codebase. As shown in Figure 2.5, NeoMAD applications are written in Java 5 and XML, which is then compiled to produce installable Android, baba, BlackBerry OS, iOS, Symbian and Windows Phone applications [153]. The framework provides a generic API with native UI controls for each platform. Platform-specific code may be used within the Java codebase in the

[6] Codename One: Pricing http://www.codenameone.com/pricing.html

event that specific functionality is not provided by the API [39]. The framework includes tools for debugging as well as simulating applications. On-device debugging is also provided, allowing for the step-by-step execution of application code on a physical handset.

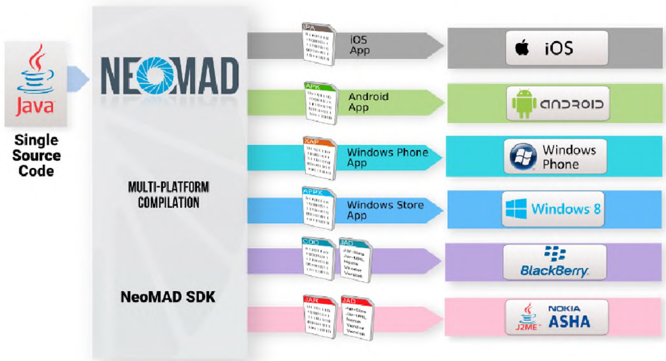


Figure 2.5: NeoMAD cross-platform framework (taken from [153])

NeoMAD offers improved low-level support when compared to the previously mentioned frameworks, such as providing access to a device's Bluetooth™ radio as well as sending and receiving SMS messages. Although NeoMAD does not require an Internet connection, the SDK of each targeted platform needs to be installed on the development workstation. In addition, unlike CN1, application UIs are constructed programmatically using either Java or XML code. Furthermore, the free version of the framework allows developers to target Android and iOS only, while targeting all platforms requires a subscription costing EUR 999 per year, per developer[7]. Windows and Mac OS X operating systems are supported.

2.8 Visualisation

"The purpose of visualization is insight, not pictures." — Ben Shneiderman [25, p.6]

Defined as "a graphical representation of data or concepts" [213, p.2], visualisation aims to solve one of today's most fundamental human-computer interaction (HCI) problems: information overload [26]. Yet choosing how to best visualise data so that it conveys meaning to the user is no easy task. As ICTs have become more widespread the amount of accessible data has increased dramatically, placing greater emphasis on the importance of well chosen visualisation techniques. Further still, resource-contrived mobile handsets pose a significant challenge when implementing visualisations, primarily due to their limited display size and scarce memory [29]. The different contexts in which these devices are used results in additional burden when designing mobile visualisations [29].

[7]NeoMAD: Pricing http://neomades.com/en/pricing

In their book titled *Information Visualization: Using Vision To Think*, Card, Mackinlay and Shneiderman maintain that visualisations assist users in understanding large datasets by making important patterns emerge, thereby allowing them to make complex decisions and identify areas which need further investigation [25,217]. Larkin and Simon demonstrated that visualisations (in the form of diagrams) help amplify cognition [112]. Similarly, Chittaro argues that visualisations on mobile devices assist with problem solving, allowing users to reach better decisions in less time. In addition, properly incorporated visualisation techniques have been shown to make mobile applications more intuitive, useful and productive [29]. The remainder of this section provides an overview of visualisation principles (Section 2.8.1), techniques (Section 2.8.2) and tools (Section 2.8.3). Comparisons between mobile and traditional visualisations are also made.

2.8.1 Visualisation principles

Traditionally, visualisations have been divided into two distinct categories: scientific visualisation and information visualisation. While scientific visualisation involves data of natural phenomena (such as a map indicating population density), information visualisation concerns itself with the presentation of more abstract data (such as financial figures or business data) which does not have any inherent spatial mapping [25, 26, 197, 212]. In addition, different types of data exist: numerical, ordinal and categorical data [197]. While data is often numerical in nature, this is not always the case. Ordinal data represents things which may be naturally ordered, such as months of the year. Conversely, categorical data has no order such as the names of countries.

Data type taxonomy

Having a clear understanding of different data types helps designers construct appropriate visualisation solutions which are representative of their underlying data. Thomas and Cook highlight the importance of this, stating: "A visualisation's quality is directly affected by the quality of the data representation underlying the visualisation" [204, p.12]. There are a variety of data type taxonomies proposed by different authors. Shneiderman provides a detailed taxonomy consisting of seven different data types which are defined as [194]:

1-dimensional. Otherwise known as 1D or linear data, examples of 1-dimentional data includes single scalar quantities such as a person's height or document text.

2-dimensional. 2D data, also known as a planar data, often takes the form of geographical maps. Geographical information systems make extensive use of this data type.

3-dimensional. Real-world objects such as buildings and biological molecules are represented by 3D data. A sphere is the most basic representation of a 3-dimensional object.

Temporal. This data type includes any of the previously mentioned types with an added time component which is used to explore data behaviour as time passes. Examples may be found in audio editing software.

Multi-dimensional. Higher-order data types contain more than 3-dimensions. Database data is often manipulated as multi-dimensional data. Relational database tables serve as a suitable example.

Tree. Also known as hierarchy structures, trees are groups of items where each item (excluding the root node) is linked to one parent item. These links as well as the items themselves may contain attributes.

Network. At times relationships between items cannot be depicted using a tree structure. Network structures are used to represent items connected to a collection of other items.

Although Ware provides a similar taxonomy, he approaches the topic more cautiously by arguing that "the classification of data is a big issue", and proposes that his classification is "not especially profound or all encompassing" [213, p.25]. Similarly, Shneiderman maintains that his taxonomy reflects "an abstraction of [...] reality" [194, p.4]. To this end, data classifications exist primarily to facilitate discussion surrounding their use, thus helping to lead to new discoveries. Keller and Keller as well as Keim define similar data type taxonomies, the details of which may be found in [106] and [105].

Visual variables

In order to effectively communicate information through the use of visualisations, a thorough understanding of graphics primitives and their associated properties is required [212]. Visual variables are the differences in graphics primitives as detected by the human eye. Ward, Grinstein and Keim have identified eight fundamental visual variables which may be used to encode visual information. These visual variables are an extension to those originally defined by Jacques Bertin in his seminal paper *Semiology of graphics: diagrams, networks, maps*, and include [13, 212]:

Position. Defined as the "first and most important visual variable" [212, p.137], position is the placement of marks within a specified display area. As users notice the spatial arrangement of a visualisation first, position is considered to have the most impact.

Mark. Visualisation data is represented by graphic primitives known as marks, of which there are four basic types: points, lines, areas and volumes. A variety of graphical objects may be used to represent marks including letters, words and symbols.

Size. This visual variable describes how small or large a mark will be drawn. Changes in size do not translate equally well to all marks, and designers need to be aware of this during visualisation construction.

Brightness. Otherwise known as luminance, this visual variable can be used to "provide relative difference for large interval and continuous data variables" [212, p.142]. Special attention needs to be paid when specifying this variable, as not all brightness value pairs are easily distinguished by the human eye.

Colour. This visual variable is defined by a combination of two parameters: hue and saturation. Data values are mapped onto individual colours (using colour maps) in order to correctly display information.

Orientation. Also known as direction, this property specifies how a mark is rotated in relation to a data variable. This visual variable cannot be used with all marks (such as dots) and those with a natural single axis are best suited.

Texture. Texture can be thought of as a "combination of many of the other visual variables" such as marks, colour as well as orientation, and is usually associated with polygons, regions or surfaces [212, p.145].

Motion. The final attribute, motion, displays information based on how a variable changes over time. Motion can be used in combination with any of the previously mentioned visual variables.

The effectiveness of a visualisation depends on how suitably these eight visual variables are combined to convey information. As such, visualisation designers need to bear in mind the limitations of the human perception system by carefully selecting visual variable parameters [212]. Interestingly, certain visual variables trigger pre-attentive processing, thereby enabling the low-level human vision system to detect them rapidly and accurately with very little cognitive effort [82].

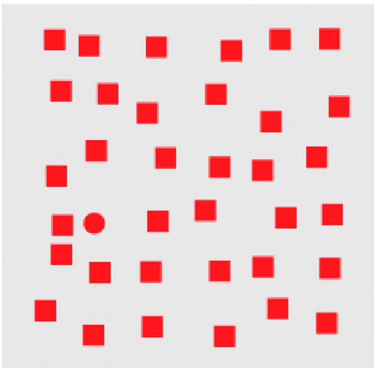


Figure 2.6: A pre-attentive visualisation (taken from [82])

The circle contained in Figure 2.6, for example, is found effortlessly within the group of squares as a result of pre-attentive processing.

Stages of visualisation

The modern visualisation process is a highly dynamic one, with control over virtually all stages being relinquished to the user. In an attempt to better understand the visualisation process, Ware has divided it into four basic stages, as illustrated in Figure 2.7. The entire process is self-contained and includes a variety of feedback loops.

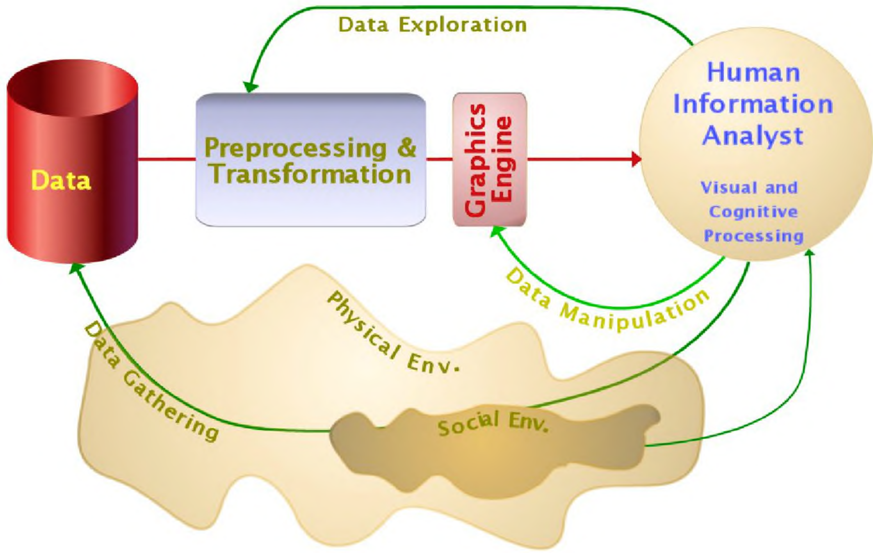


Figure 2.7: The stages of visualisation (taken from [213])

Each of the four stages are defined as [213]:

1. The gathering and storing of data.
2. Pre-processing transforms data into something more manageable.
3. Selected data is then mapped into a visual representation (which is then usually displayed as an image on a screen).
4. Finally, the human perceiver's visual and cognitive system process the displayed image.

The first stage, data gathering, is considered to be the longest feedback loop. Here the user collects as much data as deemed necessary. Next, the collected data is transformed in such a way as to reveal its underlying meaning, bringing to light new information. This may involve the system searching through a vast amount of data in an attempt to find a particular subset of this data. During stage three, the selected data is finally visualised and presented to the user. In addition, the entire visualisation process may be interactive in nature. The data gathering feedback loop is affected by both the social and the physical environment. While the physical environment represents a data source in itself, the social environment determines in a subtle yet intricate way what data is gathered and how the gathered data is then interpreted [213].

Visualisation interactivity

An important feature in many visualisations is interactivity, as it provides a means for users to directly engage with the visualisation process. Often there is a need to visualise more data than is able to fit on a single screen. At other times, in order to better understand visualisations, users need to vary the data-to-image mapping. Interactive visualisations achieve this by allowing users to modify several parameters, such as colour, orientation, size and visualisation type, in real-time. Should the interact-visualize-observe feedback loop iterate quickly enough, a user feels the sensation of 'navigating' through a visualisation. This sensation has been shown to strongly encourage and support the exploration process, thereby helping reveal the underlying information [202].

Visualisation and cognition

A 1987 study by Larkin and Simon demonstrated why suitably chosen visualisations have the potential to amplify human cognition, thus leading to more effective problem solving [112]. In their paper, the authors compare the performance of participants solving mathematics and physics problems using visual representations versus text-based representations. Particular attention is given to the amount of cognitive effort required when performing search, recognition and inference tasks. The study concludes that diagrams help aid cognition in three fundamental ways. Firstly, excess amounts of searching is avoided as information that is used together is often grouped together. Next, search and working memory load is reduced as location is used to group an element's information instead of symbolic labels. Finally, humans automatically find many of the perceptual inferences triggered by visualisations inherently easy to understand [112].

Mobile versus traditional visualisation

Mobile devices present unique constraints compared to that of traditional desktop computers, such as limited display size and variety of usage contexts [29]. Consequently, designers cannot simply port existing visualisations without first understanding the peculiarities and limitations which mobile devices pose. Chittaro in his paper *Visualizing information on mobile devices* provides a list of mobile device limitations, which serves as a useful reference during mobile visualisation design. When compared to desktop computers, mobile devices generally have [29]:

- limited displays (small size, low resolution, low bit depth);
- different aspect ratios (traditional desktop computers are usually 4:3 or 16:9);
- less powerful hardware (CPU, memory, buses, GPU);
- inadequate input peripherals for complex tasks (small keypad, rollers, mini-joysticks);
- differing input techniques (one-hand thumb-based input, keypad, touch screen);
- slower, less reliable connectivity (affecting interactivity of applications);
- massive differences in form factors, performance and input techniques across different devices; and
- few powerful, high-level graphics APIs and libraries.

Although there have been noticeable improvements in many of these areas in the eight years since the paper's publication, several of the author's observations still ring true. Especially when considering feature phones and low-end smartphones. Chittaro highlights this, noting: "From a purely technical point of view, developing visualisations on mobile devices will become easier due to on-going improvements such as new, possibly standard APIs and more powerful devices" [29, p.9].

In addition, the author provides a brief list of complications introduced as a result of the different contexts in which mobile devices are used. These complications include the extreme variability of the physical environment (for example, bright lighting making information difficult to decipher); the need for new and different applications which are not currently suitable for use in a mobile context; the reduction in cognitive resources that users are able to devote to tasks performed in a mobile context; as well as the simultaneous use of mobile devices and other activities resulting in possible safety issues [29]. As such, it is imperative that mobile applications convey the necessary information as quickly as possible. Mobile visualisations are seen as an ideal tool for this as they enhance the acquisition of information in less time than other methods [25].

2.8.2 Visualisation techniques

Datasets can be visualised through a variety of different visualisation techniques, from standard x-y plots, line graphs and bar charts to more elaborate and sophisticated graphing techniques [105]. Additionally, there are several ways in which to classify these visualisation techniques: based on the underlying data structure; based on the display dimension; or based on the tasks which are to be performed [43].

Classifications of visualisation techniques

Keim provides a detailed classification of visualisation techniques, dividing them into five distinct categories: [105]:

1. Standard 2D/3D displays, such as x-y plots and bar charts.
2. Geometrically transformed displays, such as landscapes and scatter plot matrices.
3. Iconic displays, such as those making use of different icons.
4. Dense pixel displays, such a recursive pattern and graph sketches.
5. Stacked displays, such as dimensional stacking or tree maps.

Fayyad, Vierse and Grinstein provide an alternative classification of visualisation techniques based on “whether their focus is geometric or symbolic, whether the stimulus is 2D or 3D, or whether the display is static or dynamic” [43, p.29]. Geometric representations often result in a physical model or simulation, and are used when the underlying dataset contains fields, which may be conveniently mapped to the axes of a Cartesian coordinate system. Such visualisations usually make use of scatter plots, lines, surfaces or volumes. Symbolic visualisation techniques focus on displaying the relationship amongst non-numeric data items through the appropriate use of pixels, icons and graphs. The network graph shown in Figure 2.8 serves as an example. With both geometric and symbolic representations, the final visualisation can be rendered into either 2D or 3D space. Finally, depending on how data are presented, visualisations can be either static or dynamic in nature, with the latter encouraging user interaction thereby helping uncover new information [43].

Although the above classifications serve as a useful guide, the authors emphasise that not all techniques fit neatly into a single category, stating that “many displays are hybrids of various techniques” [43, p.30].

Tasks in visualisation

Users interact with visualisations in order to extract insight and meaning from the data at hand. As every visualisation has its own set of potential tasks, a complete overview of all of tasks would be unimaginable [26]. Instead, Shneiderman provides a collection of seven abstract tasks which users perform. Collectively, they attempt to model the behaviour a user would follow when trying to extract insight and meaning from datasets [194]:

Overview. Gain an overview of the entire collection. For example, utilising a fish-eye strategy when browsing networks.

Figure 2.8: A network graph (taken from [85])

Zoom. Zoom into interesting items, thereby gaining a more detailed view. For example, clicking and holding down a mouse button to enlarge a portion of the display.

Filter. Reduce search size by allowing uninteresting items to be filtered out. For example, dynamic queries that are executed via buttons or sliders.

Details-on-demand. Get details when needed by selecting an item or item group. For example, a pop-up dialogue can display the details of a selected item.

Relate. View the relationships which exist among items. For example, select an item that is then able to show all other items which are somehow related to it.

History. Store a history of past actions which allow a user to undo, replay and make progressive refinement. For example, allowing a mistake to be undone.

Extract. Allow items or data to be extracted in a file format suitable for facilitating other uses, such as printing, sending via e-mail or importing into other applications.

Shneiderman proposes that a combination of these tasks should be supported by visual exploration applications for them to be considered effective [194].

2.8.3 Visualisation tools

This section introduces a selection of visualisation tools and libraries. The Google Charts API is first presented (Section 2.8.3), followed by Flot (Section 2.8.3) and AChartEngine (Section 2.8.3).

Google Charts

Google Charts allows developers to visualise data, providing a variety of different visualisation types. The service consists of two different plotting solutions, both of which will be described next.

Image charts Initially developed for internal use by Google, the Image Chart API was first made public in 2007 and allows developers to dynamically generate charts using specially formatted URIs. Charts take the form of a Portable Network Graphics (PNG) file, which is returned by the Google Chart API in response to an HTTP `GET` or `POST` request [63]. The returned image may then be embedded directly into a web page or application, or cached for subsequent offline use. As all processing is performed on Google's servers there is no overhead associated with rendering image charts, making it an attractive option for use on power-contrived mobile devices. Although image charts do not offer the interactive functionality which newer JavaScript-based charts provide, they have a distinct advantage when it comes to portability. While many entry-level handsets lack full JavaScript support, their ability to load and display image files is virtually guaranteed. In addition, image files are likely to occupy less memory than their equivalent JavaScript-based charts, as local charting libraries are not required. The Image Chart API supports many different chart types, including pie charts, bar charts, line charts and Venn charts, amongst others.

Interactive charts The interactive charts library allows developers to dynamically create JavaScript-based chart visualisations. Charts are loaded using the provided 'boilerplate' JavaScript code[8], which is embedded into a web page or `Webview` component [65]. The required Google Chart libraries along with the data to be visualised also need be loaded [65]. A variety of chart types are provided by Google, with each being exposed as a separate JavaScript class [65]. As all chart rendering is performed locally there is additional processing overhead when compared to Google's image charts. Support is limited to JavaScript-compatible devices thereby excluding low-end handsets where full JavaScript is not provided. Memory consumption is also increased when compared to image charts, as a result of the chart libraries which need to be loaded. Interactive charts do, however, potentially offer a more engaging user experience, thus increasing the understanding of presented data as well as its usefulness [217]. Google's interactive charts have been used in civic engagement projects such as the USAID funded *Afghan Civic Engagement Project* which aims to enable "greater governmental transparency and accountability" [99].

[8]Google Charts: Quick Start http://developers.google.com/chart/interactive/docs/quick_start

Flot

Described as a "pure JavaScript plotting library for jQuery" [114, p.1], Flot allows developers to create clear and concise interactive chart visualisations. The library is compatible with all browsers (and `Webview` components) which support the HTML5 `<canvas>` tag, and is currently the most widely used plotting library on the Android platform [113]. Flot provides a variety of different chart types such as lines, bars, points and filled areas, placing special emphasis on interactivity features [116]. The appearance of a chart is set by a variety of different basic options [115]. As with Google's interactive charts, rendering is performed on-device, posing additional processing overhead when compared to Google's image charts. Similar to other JavaScript-based plotting libraries, support for both entry-level and older mobile phones is restricted due to their absence of JavaScript. Flot has been successfully used to visualise the data of several different project types, such as the tracking of viral infection performance in the United Kingdom as well as combining the National Weather Service forecasts in Utah, USA [117]. The library is open source with repositories available on GitHub[9].

AChartEngine

Released in early 2009, AChartEngine is an open source charting solution, native to the Android platform. The library supports a variety of different chart types including line, area, scatter, time and bar charts. Developers are able to modify the appearance of each by adjusting individual chart properties. Many different interactive features are provided by AChartEngine, including panning, zooming and rotation. Unlike Google interactive charts and Flot, AChartEngine does not leverage a `Webview` component for rendering, instead making use of the host device's native graphics pipeline. This may potentially result in improved performance when compared to its counterparts. Although the library is native to Android, it has since been adapted for use with the CN1 cross-platform mobile framework. CN1 support, however, is limited to Android, iOS and Windows Phone [5].

2.9 Summary

This chapter presented an overview of literature relating to the use of ICTs in the public sector (particularly e-government and m-government) and discussed a variety of ICT-facilitated civic engagement projects, both local and international. Relevant mobile technologies as well as visualisation principles, techniques and tools were then described. The following chapter introduces and discusses the methodology followed while conducting this research.

[9]Flot repository http://github.com/flot/flot

Chapter 3

Methodology

3.1 Introduction

This chapter presents an overview of the methodology followed when conducting this research project. First, the concept of interaction design is introduced (Section 3.2). The interaction design life-cycle model and its use within the context of this project is then described (Section 3.3). Next, the informal retail outlet survey and informal usability evaluation are discussed. Finally, the baseline study, as well as usability evaluations are briefly detailed, and will be elaborated upon in Chapter 4, 6 and 7.

3.2 Interaction design

Defined as "designing interactive products to support people in their everyday and working lives" [193, p.6], interaction design takes a user-centred approach to development by consulting users early on in the development process to ensure that the end result matches their expectations and fulfils their needs. This allows users to influence and shape an application's design, ideally leading to a more usable and satisfying solution [2].

Preece, Rogers and Sharp highlight that "underlying good interaction design is the philosophy of user-centred design" [193, p.165], thus allowing development to be directed by the concerns of users, as opposed to the technologies being leveraged. As the term suggests, user-centred design (UCD) is described by Abras, Maloney-Krichmar and Preece as "a philosophy and methods which focus on designing for and involving users in the design of computerised systems" [2, p.12]. Originally coined by Don Norman in his book, *User-Centred Design: New Perspectives on Human-Computer Interaction*, UCD has been widely adopted by HCI professionals due to the cost-effective approach it provides to interaction design [2].

UCD was further expanded and elaborated upon by Norman in his second book, *The Psychology Of Everyday Things*, in which he improves the usability of designs by identifying specific requirements and interests of the user. Herein, Norman offers several recommendations on what constitutes a 'good' design, namely [159, p.188]:

1. Make it easy to determine what actions are possible at any moment.
2. Make things visible, including the conceptual model of the system, the alternative actions and the results of actions.
3. Make it easy to evaluate the current state of the system.
4. Follow natural mappings between intentions and the required actions; actions and the resulting effect; and information that is visible and the interpretation of the system state.

Similarly, Gould and Lewis highlight three key principles for creating usable designs: 1) place an early focus on users and tasks; 2) perform empirical measurement early on in the development process; and 3) adhere to an iterative design process. As such, when issues are found during user testing, they must be fixed, then retested as many times as necessary [68].

While many researchers praise the benefits of the UCD approach [2,32,161,193], others are less convinced about its use within the context of developing countries, citing potential language, literacy and cultural barriers [36,89,136]. Low-fidelity prototypes (often used in the UCD approach) are argued to be potentially misunderstood by participants, causing confusion due to their lack of inherent interactivity [137, 172]. Further still, complex socio-cultural issues are at times ignored by low-fidelity prototypes, only to be discovered at a later stage when design changes are less trivial to implement [136].

In this research, an assistant from Grahamstown was recruited to help mitigate these identified challenges. The assistant, who possesses strong community ties and is fluent in English, isiXhosa and Afrikaans, helped with the data gathering process and acted as a translator when needed. In addition, high-fidelity interactive prototypes were used exclusively during the evaluation stages of this research, potentially avoiding the pitfalls associated with their low-fidelity counterparts. While Maunder, Marsden, Dominic and Blake note that high-fidelity prototypes are more costly to develop (due to the additional time and effort required) [136], it was decided to leverage them within this research due to the potential benefits they presented.

3.3 Interaction design life-cycle model

Shown in Figure 3.1, the interaction design life-cycle model describes how the four activities of interaction design relate to each other. The life-cycle model encourages iteration and user focus, and was closely followed throughout this research. As such, the life-cycle model was iterated through an appropriate number of times, generating alternative designs and thereby shaping and refining the resulting applications. The importance of this process is highlighted by Preece et al. who argue that "generating alternatives is a key principle in most design disciplines, and one that should be encouraged in interaction design" [193, p.166]. The four activities of the interaction design life-cycle model are [185]:

1. Identifying needs as establishing requirements.
2. Develop alternate designs that meet these requirements.
3. Implement interactive versions of the designs so that they can be communicated and assessed.
4. Evaluate what is being built throughout the process.

In addition, several usability goals are at the centre of interaction design, including effectiveness, efficiency, safety, utility, learnability and memorability. Goals which are of particular importance need be identified and documented at the beginning of the development process. To better understand how the life-cycle model was followed during this research, Figure 3.1 has been annotated with numbers 1 to 4. Each of these numbers represents an activity within the life-cycle model and are elaborated upon next.

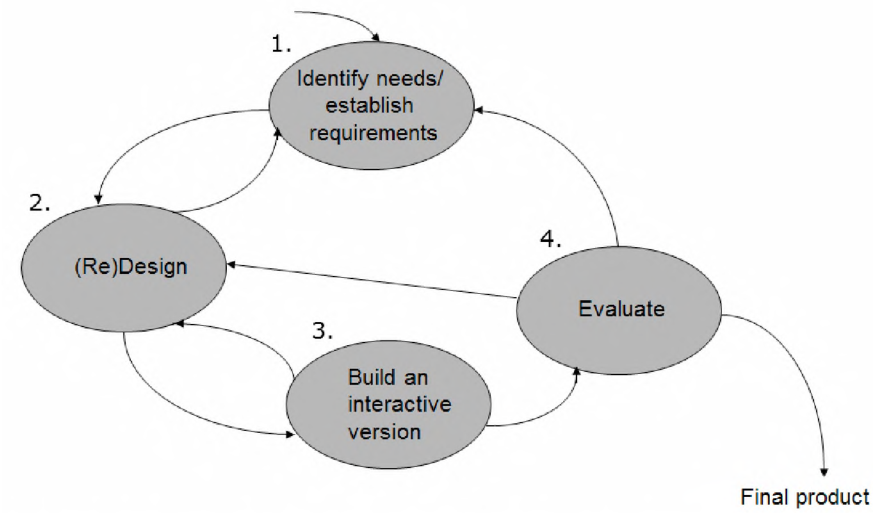


Figure 3.1: Interaction Design Life-cycle Model (adapted from [193])

3.4 Use of the interaction design life-cycle model

As previously highlighted, interaction design places the user at the centre of the design process in an attempt to narrow the gap between the researcher's perceived challenges and the actual challenges. Three iterations of the life-cycle model were completed during this research. It was decided to combine the development of the client applications and the SMS gateway within a single life-cycle model, as they collectively form part of the front-end of the MobiSAM system (see Section 5.3).

In accordance with the interaction design process, three primary usability goals for the client applications were identified and agreed upon upfront: learnability, memorability and satisfaction. The first goal, learnability, was chosen as users need to easily gain proficiency at using the client applications, without extensive training. As highlighted by Myers: "Time is valuable, [...] they [users] want to spend time accomplishing their goals, not learning how to operate a computer-based system" [151, p.74]. In addition, a survey of mobile application users conducted in 2012 found that, on average, users will spend five minutes or less learning a mobile application [49]. After this time, alternative applications may be sought. As there are likely to be long periods where the client applications are not used, each application needs to possess a high degree of memorability, thus ensuring that users are able to quickly re-establish their proficiency. The importance of this usability goal is reiterated by the previously mentioned survey, which found that nearly half of all installed mobile applications are only used once a month [49]. Finally, satisfaction was chosen in an attempt to ensure that users find the client applications pleasant to use, increasing the likelihood that they will be used again in the future, as opposed to seeking other avenues of complaint. In addition to the three previously mentioned usability goals, the utility of each client application was also kept in mind throughout. This attempted to ensure that the applications offered all the features needed by users. As maintained by Nielson: "Usability and utility are equally important [...] It matters little that something is easy [to use] if it's not what you want" [156, p.1]. When combined, the usability and utility of an application determines its usefulness [156].

Each iteration of the life-cycle model performed while conducting this research will now be discussed. Figures 3.2 to 3.4 provide a graphical representation of each iteration.

3.4.1 Iteration 1

The first iteration of the interaction design life-cycle model aimed to gather a broad overview of the problem space, including system requirements and related literature. It also saw the development of two high-fidenlity prototype client applications and the evaluation thereof.

Activity 1.1 Initial requirements were obtained from the MobiSAM project director and a project stakeholder, in this case Makana Municipality. While specifics are elaborated upon in Chapter 5, the overarching requirement was the need for a mobile client application enabling Grahamstown residents access to the MobiSAM service. This application would serve as an

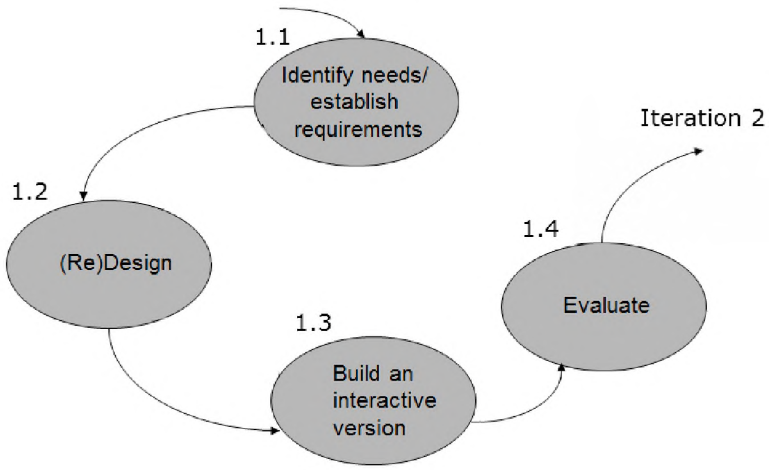


Figure 3.2: Iteration 1 of the life-cycle model

alternative to the MobiSAM website, allowing users to view and answer relevant service delivery polls, as well as visualise poll results in an easy-to-understand manner. Additionally, the client application needed to support a variety of different mobile platforms and interaction styles. An informal mobile-phone retail outlet survey (see Section 3.5) helped provide preliminary information as to which handsets (and thus platforms) needed targeting.

The procedures followed by Makana Municipality when logging and tracking service delivery issues were also examined in order to better understand the Municipality's internal workflow. This included identifying the most frequently reported service delivery issues, as well as determining the type of information gathered during the reporting process. Finally, a thorough literature review (see Chapter 2) was undertaken, allowing for a broader understanding of the problem space. The review attempted to investigate current literature relating to the use of ICTs in the public sector; explore and describe several local and international ICT-facilitated civic engagement projects; as well as detail a variety of technologies relevant to this research project. Visualisation principles, techniques and tools were also discussed, along with their use within a mobile context.

Activity 1.2 The second activity saw the design of two high-fidelity prototype client applications. Each client application was designed according to the requirements and identified needs gathered during Activity 1.1 (see Appendix A.2). Results from the informal retail outlet survey indicated that the Java ME and BlackBerry OS platforms were to be initially targeted (see Section 3.5). Both prototypes were designed to be functionally identical, allowing users to view and answer service delivery polls relevant to their suburb. Multi-language support was also offered. Poll result visualisations were designed to be legible in a variety of environmental conditions, and leveraged a simple numerical table and brightly coloured bar chart.

Activity 1.3 High-fidelity prototypes were then implemented using the necessary platform-specific tools and SDKs (see Appendix A.2.4), resulting in two fully interactive client applications, targeting what the informal retail outlet survey showed to be the two most popular mobile platforms. Each application was installed on a compatible handset (a Nokia 2630 feature phone and a BlackBerry 9800 smartphone), ready for Activity 1.4: an informal usability evaluation with two expert users.

Activity 1.4 The fourth and final activity of Iteration 1 involved evaluating each high-fidelity prototype client application (see Section 3.6). Two expert users participated in the informal usability evaluation. Both users were involved in providing the initial client application requirements, and were therefore able to offer a critical assessment of each high-fidelity prototype. Although the functionality of each prototype met user requirements, the evaluation highlighted several minor usability concerns. In addition, the shortcomings associated with developing multiple, platform-specific versions of the client application were revealed. This feedback influenced the redesign of the client applications, as described next.

3.4.2 Iteration 2

Feedback from the informal usability evaluation served as input into the second iteration, bringing to attention usability concerns as well as providing additional requirements. Iteration 2 aimed to refine the client applications and saw the redesign of the platform specific high-fidelity prototypes into a single cross-platform client application. However, before this could take place, numerous mobile cross-platform frameworks were investigated. A comprehensive usability evaluation was then conducted using the newly developed cross-platform client application, focusing specifically on navigation and visualisation.

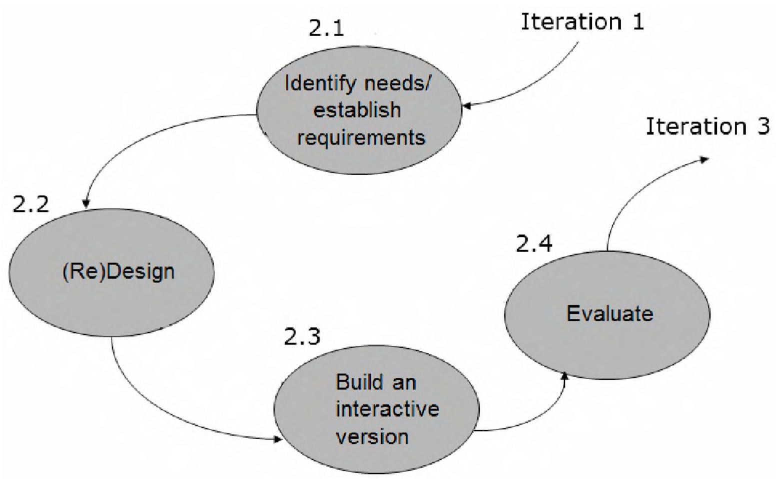


Figure 3.3: Iteration 2 of the life-cycle model

Activity 2.1 The first activity of Iteration 2 involved identifying additional needs and requirements, based on feedback obtained from the informal usability evaluation. Results from the evaluation highlighted several noteworthy usability concerns (see Section 3.6). In addition, comments from the expert users advised against developing multiple, platform-specific versions of the client application. As an alternative, a selection of cross-platform mobile frameworks were sought (see Section 2.7.2) and evaluated (see Section 5.5). The Codename One (CN1) cross-platform framework was found to be most suitable.

Activity 2.2 The high-fidelity Java ME and BlackBerry OS prototypes were then redesigned as a single cross-platform client application. The cross-platform client application's UI was designed using the 'drag and drop' CN1 designer tool. This allowed for the quick mock-up of multiple versions of the same screen, potentially leading to more fitting solutions. Previously identified usability concerns were also taken into account during this activity.

Activity 2.3 An interactive version of the newly designed cross-platform client application was then implemented (see Section 5.6). The CN1 framework allowed several platforms to be targeted from a single codebase, thus avoiding the unnecessary duplication of program code and simplifying the implementation process.

Activity 2.4 The fully functional cross-platform client application was then assessed via a usability evaluation involving 30 participants (see Chapter 6). Particular attention was given to navigation and visualisation aspects of the client application. Results from the study were used to update the requirements for the third and final iteration of the life-cycle model, presented next.

3.4.3 Iteration 3

Using the knowledge gained from Iteration 2, the third and final iteration aimed to further refine the existing cross-platform client application. In this iteration, a second cross-platform client application was developed (MobiSAM Report) as well as an SMS gateway application. The original (MobiSAM) and new (MobiSAM Report) cross-platform client applications were then evaluated side-by-side via a usability study.

Activity 3.1 Additional needs and requirements were obtained from four primary sources during the final iteration of the life-cycle model: a baseline study (see Chapter 4); a usability evaluation (see Chapter 6); Makana Municipality, as well as additional project stakeholders.

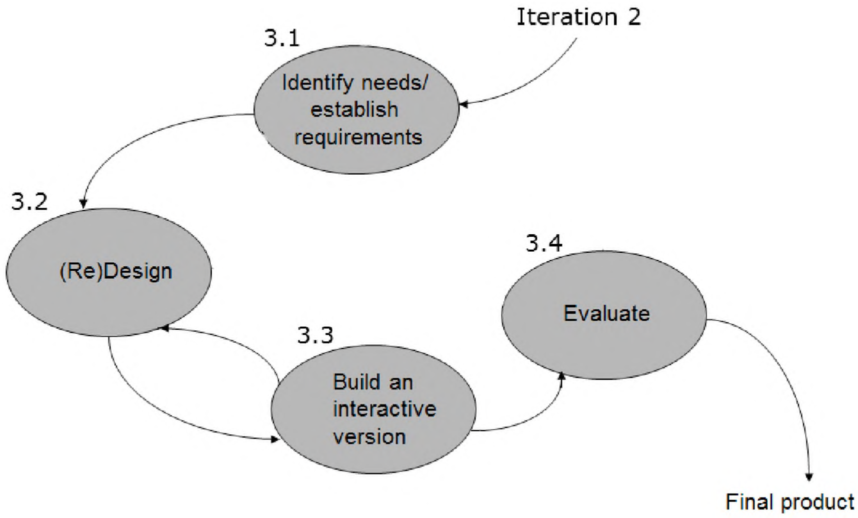


Figure 3.4: Iteration 3 of the life-cycle model

As is described in Section 6.7, results from the second usability evaluation (conducted during Activity 2.4) highlighted several additional usability concerns associated with the cross-platform client application. Furthermore, the project stakeholders felt that the focus of the existing cross-platform client application needed to be realigned. Instead of allowing users to answer service delivery polls, and subsequently presenting them with visualised poll results, stakeholders felt it was more important to streamline the process of reporting *specific* service delivery issues relating to water, roads, electricity and sanitation. As a result, it was decided to develop a second cross-platform client application, MobiSAM Report, according to their newly provided requirements.

Finally, feedback from Makana Municipality as well as the baseline study (see Chapter 4) emphasised the need to lower the barrier of entry to using the MobiSAM service. Accordingly, it was decided to develop an SMS gateway, thus providing any resident with access to a mobile phone the ability to report service delivery issues to Makana Municipality via traditional SMS.

Activity 3.2 During the second activity, the cross-platform MobiSAM client application was redesigned according to the updated needs and requirements (see Section 5.6), remedying the identified usability concerns and functional shortcomings. In addition, MobiSAM Report was designed anew, placing particular emphasis on the quick reporting of specific, predefined service delivery issues. Furthermore, the SMS gateway application was designed according to specifications obtained from Makana Municipality (see Section A.6.2).

Activity 3.3 The cross-platform MobiSAM and MobiSAM Report client applications were then implemented using the CN1 framework (see Sections 5.7 and 5.8), ready for the second usability evaluation. In addition, the SMS gateway application was implemented using the .NET framework, coupled with the necessary hardware (MOBITEK S63 GSM modem) and software libraries (SMS API v7).

Activity 3.2 After implementing changes to the cross-platform MobiSAM client application, it was decided to incorporate two additional features from MobiSAM Report. These included the ability to capture and include a photo of the service delivery issue being reported, as well as provide the option to obtain the user's exact position using their handset's built-in Global Positioning System (GPS) module (see Section 5.8.3), the latter being suggested by a two participants who took part in the second usability evaluation (see Chapter 6). In addition, these design modifications allowed for a more accurate comparative usability evaluation of the cross-platform client applications to be conducted, as both MobiSAM and MobiSAM Report now offered a very similar set of features. Finally, the UI of the SMS gateway application was re-designed using the third-party library MahApps.Metro, in an attempt to simplify its layout and improve its usability.

Activity 3.3 The two additional features (photo capture and geotagging) which were included in the cross-platform MobiSAM client application, were based on the implementation used in MobiSAM Report, and thus could be reused with minor modifications. The SMS gateway application required substantial changes, as much of its functionality had to be reimplemented using specific MahApps.Metro API routines. In addition, it was decided to restructure the SMS gateway application from a WinForms event-driven atchitecture to that of a WPF Model-View-ViewModel (MVVM) architecture. This decoupled the application UI from the application logic, allowing each to be modified independently, resulting in more manageable code.

Activity 3.4 Finally, the cross-platform MobiSAM and MobiSAM Report client applications were evaluated with 10 participants (see Chapter 7), in an attempt to determine which version participants found most intuitive and appealing. The client applications were assessed in a comparative usability evaluation according to the three usability goals highlighted in Section 3.4: learnability, memorability and satisfaction. As the SMS gateway application was to be primarily operated by the researchers, it did not undergo any formal usability evaluation.

3.5 Informal mobile-phone retail outlet survey

A brief survey was conducted during Iteration 1 of the life-cycle model (see Section 3.4.1), which attempted to determine the best-selling handsets within Grahamstown. Although anecdotal in nature, the collected data helped provide insight into which particular mobile devices and platforms were currently most popular. In addition, the survey results helped determine which handsets were to be purchased for application development and testing purposes.

Taking place over a two day period, the researcher visited a selection of local retailers known to sell mobile phones. Retailers included two Edgars stores, Nashua Mobile, MTN, Markham and PEP Stores. As PEP have recently emerged as the largest mobile phone retailer within South Africa [61], all Grahamstown branches were visited. Upon arriving at each store the researcher requested to speak to an employee who was most apt at answering mobile phone related questions. Usually one person was identified at each store. The identified employee was then approached and informed about the purpose of the informal study.

Next, they were asked to identify the best-selling low-end, mid-range and high-end mobile phone sold in their particular store. Handsets were considered low-end if they retailed for R400 or less, mid-range if they retailed between R401 and R1,000, while high-end devices retailed for over R1,000. Initially, employees were asked to identify the three best-selling handsets within each of these categories, however, this soon proved to be too time consuming. Instead they were simply asked to identify the single best-selling handset within each category. A simple form was used to capture employee responses, which included information relating to the brand, model and retail price (when available) of the handset, as well as additional written observations. Once the informal survey was complete, results were captured using Microsoft Excel.

	Brand	Model	Price	Mobile Platform	Internet Connectivity	Application Support
Low-end	Samsung	E1050	R129	Proprietary	N	N
Mid-range	Nokia	Asha 201	R699	Java ME (Nokia S40)	Y	Y
High-end	BlackBerry	9320/9360	R1,999	BlackBerry OS 7.0	Y	Y

Table 3.1: Best-selling handsets within Grahamstown

Descriptive statistical analysis was performed on the survey results. Mobile handsets which appeared most frequently in each of the three categories were noted, providing insight into the best-selling low-end, mid-range and high-end handsets within Grahamstown (see Table 3.1). The technical specifications of each identified handset were then determined by consulting GSMArena[1]. Information such as the type of mobile platform, Internet connectivity and installable application support were noted. Results from the survey helped reveal which brand and model of handsets sold well within Grahamstown, providing insight into which platforms the MobiSAM client applications needed to target.

3.6 Informal usability evaluation

An informal usability evaluation was conducted with two experts users, each familiar with the MobiSAM project, and attempted to highlight any usability concerns with the Java ME and

[1] GSMArena http://www.gsmarena.com/

BlackBerry OS prototype client applications. Instead of presenting each participant with a set of predefined tasks, the study took a more informal approach. The researcher requested that the participants navigate around each prototype client application, answering a variety of service delivery polls and viewing their results. During this process participants were asked to think-aloud, describing any usability concerns and functional shortcomings they encountered. Responses were transcribed verbatim by the researcher. Several noteworthy concerns were highlighted, including that:

- all available polls appeared on a single screen and were not organised by category;
- after completing a poll the prototype client application navigated back to the home screen instead of returning to the list of available polls; and
- no interactivity was included within either of the client applications.

The expert users suggested a remedy for each identified problem. These were then implemented during the second iteration of the life-cycle model, described in Section 3.4.2, and included:

- grouping available polls by category, thereby making it easier to find polls of interest;
- returning the user to the list of available polls after dismissing the results screen, allowing them to continue answering polls; and finally
- incorporating some form of interactivity within the chart visualisations, in an attempt to create a more engaging user experience;

In addition, the two expert users suggested that instead of targeting each mobile platform natively, using a separate codebase for each client application, different cross-platform mobile frameworks should be investigated. This would allow the researcher to target a selection of mobile operating systems from a single, common codebase resulting in quicker development times and easier maintenance.

3.7 Baseline study and client application usability evaluations

A detailed baseline study, as well as two usability evaluations were conducted as part of the research project. Chapters 4, 6 and 7 provide full details of these interventions, however, for ease of reference this section provides a brief overview of each.

3.7.1 Baseline study

As part of the requirements eliciting process (see Section 3.4.3), a detailed baseline study was conducted within Grahamstown. The study, consisting of 105 respondents, took the form of a four page questionnaire, and attempted to empirically investigate how residents currently use mobile technologies and participate with Makana Municipality around the area of service delivery. Results from the study helped provide an updated set of requirements used during Iteration 3 of the life-cycle model. Information relating to the study sample (Section 4.2), data-collection instruments (Section 4.3), method and analysis (Section 4.4) as well as the study results (Section 4.5) are presented in Chapter 4.

3.7.2 Evaluation: navigation and visualisation

The second usability evaluation, consisting of 30 participants, aimed to determine how intuitively participants were able to navigate within the client application. In addition, the evaluation aimed to gauge how effectively the chart visualisations were able to convey meaning to participants, and what role interactivity within the charts played in helping participants better understand the poll results. The usability of the client application was assessed according to the three previously identified criteria: learnability, memorability and satisfaction. Details relating to the study sample (Section 6.2); data-collection instruments (Section 6.3); method and analysis (Section 6.4); as well as results (Section 6.7) are presented is Chapter 6.

3.7.3 Evaluation: client application comparison

The third and final usability evaluation, consisting of 10 participants, aimed to determine which cross-platform client application participants preferred: MobiSAM or MobiSAM Report. A mixed method, comparative usability evaluation was conducted, providing both qualitative and quantitative feedback. Client applications were evaluated side-by-side according to the usability goals learnability, memorability and satisfaction. Further details surrounding the study are presented in Chapter 7, including the study sample (Section 7.2); data-collection instruments (Section 7.3); method and analysis (Section 7.4); as well as the study results (Section 7.7).

3.8 Summary

This chapter has provided an overview of the methodology followed when conducting this research project. The concept of interaction design was introduced, emphasising the importance of adhering to a UCD approach when conducting this research. The interaction design life-cycle model and its use within the context of the project was then described, including a detailed overview of the activities performed during each iteration. Finally, the informal retail outlet survey and informal usability evaluation were briefly discussed, followed by the baseline study and subsequent comprehensive usability evaluations. The next chapter provides detailed coverage of the baseline study, which aimed to investigate how residents currently use mobile technologies and participate with Makana Municipality around the area of service delivery.

Chapter 4

Baseline study

4.1 Introduction

This chapter provides an overview of the baseline study conducted within Grahamstown, the largest city within the borders of Makana Municipality. The study attempted to empirically investigate how local residents are currently using mobile technology, and participating with local government around the area of service delivery. Results of the study were used to partly inform the design of the cross-platform MobiSAM client application, in an attempt to ensure that user requirements and expectations were satisfactorily met. The study sample (Section 4.2); data-collection instruments (Section 4.3); as well as method and analysis (Section 4.4) are first presented, followed by the study results (Section 4.5). Finally, an analysis and discussion of the results is provided (Section 4.6).

4.2 Study sample

Grahamstown citizens represent a microcosm of life in South Africa. According to the 2011 South African National Census results for the city, of the more than 50,000 citizens: 72.78% described themselves as Black African; 14.29% as Coloured; 11.22% as White; and 0.94% as Indian/Asian [52]. The split in terms of gender in the city is 52.61% female and 47.39% male [52]. As such, it was decided to try and match the race and gender demographics of our study sample as closely as possible to these results. Stratified sampling with simple random sampling within each strata (race and gender demographics) was used, in an attempt to obtain as representative a sample as possible. One hundred and five ($n = 105$) participants took part in the study, of which 51.43% were female and 48.57% were male. As residents were required to give their consent before partaking, participants were limited to individuals 18 years and older.

4.3 Data-collection instruments

Two data-collection techniques were used during the study: a paper-based questionnaire and electronic survey. The first technique took the form of a four page questionnaire which comprised of three sections: 1) demographic information; 2) mobile phone and services usage; as well as 3) service delivery satisfaction and current participation with local government.

Section 1 consisted of nine questions which solicited information such as race, gender, education, employment status and monthly income. Section 2 consisted of sixteen questions soliciting feedback on mobile phone ownership and current usage thereof. These included questions about ownership or access to mobile phones, network choice, amount spent weekly on mobile airtime and use of applications on mobile devices. The final section consisted of fifteen questions soliciting feedback on the way which residents feel about services currently delivered by the Makana Municipality, as well as their current ways of participating with local government. The section also examined participants' engagement with traditional news channels (newspaper and radio). Questions pertaining to residents' current ways of participating with local government were informed by the Citizen Scorecard developed by the African Centre for Citizenship and Democracy [3].

In addition, the questionnaire was made available electronically, as a Google Forms survey. This allowed the researcher to target specific individuals when attempting to match the race and gender strata to that of the Grahamstown population demographics. The Google Forms survey was identical in structure and content to that of the paper-based survey. Participants were required to give their consent before starting the study (in this case electronically, by clicking an 'Agree' button) after which they were taken to the start of the survey. All questions were optional and participants were able to navigate both backwards and forwards through the survey, modifying their answers as required. Most ($n = 95$) baseline study responses were collected using paper-based questionnaires, with only 10 participants completing the study electronically. A copy of the paper-based questionnaire is provided in Appendix B.1.

4.4 Method and analysis

Before the study proceeded, ethics approval was sought and granted from Rhodes University (tracking number CS12-07), after which the researcher began the process of identifying participants who were needed for the strata outlined in Section 4.2. Participants were approached from several different areas in Grahamstown: communal areas in the informal settlement; common areas near the Assumption Development Centre (also in the informal settlement); pedestrians walking down main roads in Grahamstown; and by referral from existing participants.

Once a resident agreed to participate in the study, a consent form was signed (or given electronically). The paper-based questionnaire was completed verbally, with the researcher questioning and transcribing participant responses verbatim. In the process of piloting this questionnaire, this process was found to be quicker than asking participants to complete the form themselves. In addition, this ensured that written literacy levels did not interfere with the accuracy of the results obtained. Although the questionnaire was only available in English, interviews were conducted in either English or isiXhosa, as directed by the participant, with the research assistant translating responses when necessary. No incentive or remuneration was offered to citizens for their participation. Each participant was assigned a random identification number to ensure their responses remained anonymous. Upon completion, paper-based survey responses were captured electronically and analysed using Microsoft Excel. In addition, 10 participants were recruited via e-mail and completed the study electronically using the Google Forms survey. Only English speaking participants were targeted so as not to effect the accuracy of the results.

A selection of different analysis methods were used in order to make sense of the captured data, allowing for relationships to be identified and correlations drawn.

4.5 Results

The results of the study are presented next, and are organised into three separate sections: demographics (Section 4.5.1), use of mobile technology (Section 4.5.2) and current participation with local government (Section 4.5.3). Across all results, the following race groups are used to classify responses: Black African/Coloured, White and Indian/Asian. These combinations are performed for historical reasons as they refer to the classifications used by the former dispensation. Although Apartheid has long since passed, these classifications are used in order to reflect the ongoing inequalities in South Africa.

4.5.1 Demographics

The first section of the baseline questionnaire consisted of nine questions and attempted to gather basic demographic information from respondents, including age, gender, preferred language, highest level of education, current employment status and monthly income.

One hundred and five ($n = 105$) citizens took part in the study. Of the sample group, 83.81% described themselves as Black African/Coloured; 12.38% as White; and 3.81% as Indian/Asian. The race distribution of the study closely matched the results of the 2011 National Census (87.05% Black African/Coloured; 11.22% While; and 0.94% Indian/Asian). Participants fell into the following age groups: 40% were 18-30 years; 26.67% were 31-45 years; 28.57% were 46-60 years; while 4.76% where over 60 years. Of the participants, 51.43% were female and 48.57% were male, again closely matching the 2011 National Census gender split of 52.61% female/47.39% male.

Participants were asked about their levels of monthly income. Just under a third (31.43%) said they had no income; 5.71% earned between R1-R400; 4.76% between R401-R800; 15.24% between R801-R1,600; 11.43% earned between R1,601-R3,200; 17.14% between R3,201-R6,400; 4.76% between R6,401-R12,800; 7.62% earned R12,801 or more; while 1.9% were unsure of their monthly income.

In addition, participants were asked about their highest level of education. Of the study respondents, 34.29% had obtained their matric (a pass from the final year in high school); 27.62% had graduated from university; 21.9% had attended some high school; 13.33% had a primary school education; and 2.86% had a technikon or college diploma.

The current employment status across the sample group varied. 29.52% of participants were employed full-time; over a quarter (25.71%) described themselves as students; 18.1% were unemployed; 7.62% were temporarily unemployed; 5.71% were retired; 4.76% were employed part-time; 4.76% were self-employed; and lastly, 3.81% were temporarily employed.

The preferred language of each participant was asked, of which over half (56.19%) preferred isiXhosa; 39.05% English; 2.86% Afrikaans; one participant (0.95%) selected isiZulu; while another (0.95%) selected 'Other'. In addition, literacy levels were specified. 98.1% of participants were able to read in their preferred language, while 97.14% were able to write in their preferred language.

4.5.2 Use of mobile technology

The second section of the questionnaire, consisting of sixteen questions, solicited feedback relating to participants' use of mobile technology and services. Based on the findings derived from the survey of Grahamstown residents, seven key questions were able to be answered:

- Q1: How many respondents own or have access to a mobile phone?
- Q2: What are the most prevalent brands of mobile phones and mobile platforms amongst respondents?
- Q3: What is the average monthly expenditure on mobile phones per earning bracket?
- Q4: Which do respondents use most frequently: voice, SMS or data?
- Q5: Which online data services do respondents make use of?
- Q6: How often do respondents make use of the online data services?

Q7: How many respondents have experience installing mobile applications?

Collectively, these questions aim to determine how suitable mobile technologies are to facilitate participation between residents and the Makana Municipality. The answers to each question are presented next.

Q1: How many respondents own or have access to a mobile phone? Of the 105 participants, 95.24% have access to a mobile phone. 90.48% have their own handset, while 4.76% have access to a shared mobile phone. Therefore, of the study participants, only 4.76% do not have access to a mobile phone.

Q2: What are the most prevalent brands of mobile phones and mobile platforms amongst respondents? Figure 4.1 provides an overview of the brands of mobile phones that participants own. The most popular mobile phone brands are Nokia (37.63%), Samsung (26.88%) and BlackBerry (11.83%). 'Other' encompasses the brands which were less popular amongst respondents surveyed: Alcatel, Huawei, Sony Ericsson and ZTE.

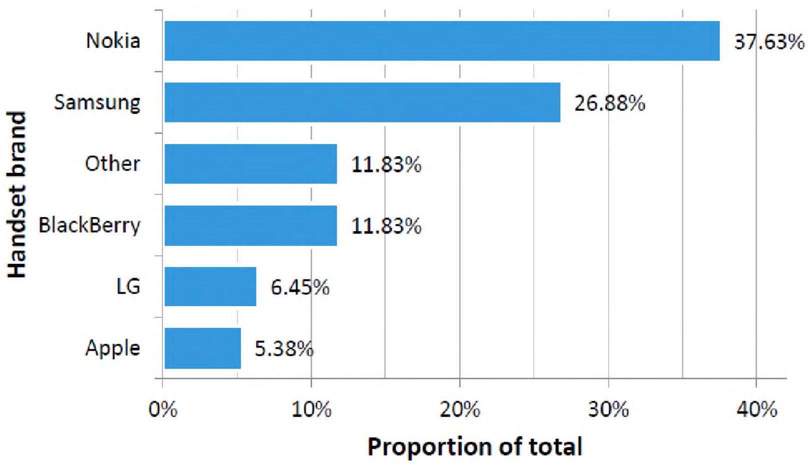


Figure 4.1: Most popular mobile phone brands ($n = 93$)

The study sought information relating to the make and model of each respondent's handset. Each newly identified handset was researched online to determine its operating system and features available. The GSMArena[1] website was used for most makes/models of handsets. In the event that the phone specifications were not available on this site, a simple Google search was used. In some cases, even though the exact make and model were captured, the operating system was simply referred to as "proprietary". In these cases, the results were grouped together, with a distinction instead made regarding the operating system's support for Java ME.

[1] GSMArena.com http://www.gsmarena.com/

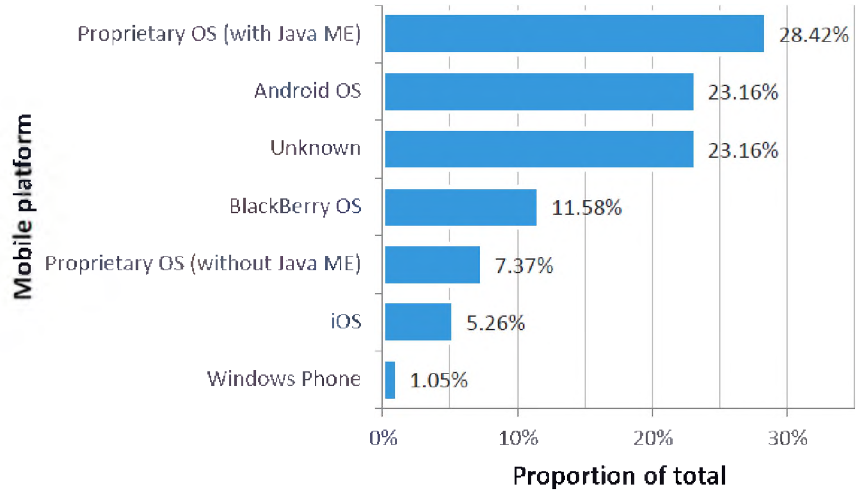


Figure 4.2: Identified mobile phone platforms ($n = 95$)

Figure 4.2 shows that the most prevalent mobile platform amongst participants is some form of proprietary OS supporting Java ME (28.42%); followed by Android OS (23.16%); and BlackBerry OS (11.58%). Devices grouped into the "Unknown" category are those where insufficient information was provided by respondents to accurately determine the platform of the device. The Nokia handsets (37.67% of mobile devices surveyed, see Figure 4.1) make up part of the following segments: Proprietary OS with Java ME support; Proprietary OS without Java ME support; or Windows Phone. This split was included to obtain a better understanding of the capabilities of the wide range of Nokia devices that were prevalent in the study.

When collecting information from participants, assistance was offered to help determine the make and model of the mobile phone. In the case of "Unknown" being recorded, either the handset was not with the participant, and the participant was unsure of the model (so only the make could be recorded), or the handset was with the participant but the model number could not be established.

Q3: What is the average monthly expenditure on mobile phones per earning bracket? In the survey, respondents were asked to indicate into which earning bracket they fell on a monthly basis (based on brackets given in the 2011 National Census). In addition, respondents were asked to indicate how much money they spent weekly on prepaid airtime (by specifying into which bracket they fell), or on their monthly contract. The weekly airtime expenditure was then extrapolated over the period of a month (approximately four weeks) and correlated to the earning bracket of each respondent.

This was done by grouping together the respondents who chose the same earning brackets, and then calculating the mode of expenditure for prepaid respondents, and the average monetary costs of contract respondents. In some cases where there was no mode for the prepaid brackets per earning bracket, the lowest common denominator (in terms of expenditure) was chosen. The results of this are presented in Table 4.1. It should be noted that blank cells in the "monthly expenditure" column for contract participants reflect that there was no one who fell into this category (rather than an expenditure of R0).

Prepaid ($n = 78$)		**Contract** ($n = 15$)	
Level of income	**Monthly expenditure**	**Level of income**	**Monthly expenditure**
I don't know	More than R16	I don't know	-
No income	Less than R5	No income	R277.25
R1-R400	R5-R15	R1-R400	-
R401-R800	R5-R15	R401-R800	-
R801-R1 600	R5-R15	R801-R1 600	-
R1 601-R3 200	More than R30	R1 601-R3 200	R240.00
R3 201-R6 400	R5-R15	R3 201-R6 400	-
R6 401-R12 800	R5-R15	R6 401-R12 800	-
R12 801 or more	R16-R30	R12 801 or more	R487.50

Table 4.1: Monthly income versus expenditure on airtime

It is interesting to note that participants who rated their level of income as "no income" were still able to purchase airtime. On closer examination of the results, exactly two thirds of those who said they had no income were also students, so it may be assumed that someone else was paying for their monthly airtime. The remaining third described themselves as temporarily unemployed.

Q4: Which do respondents use most frequently: voice, SMS or data? Participants were asked to rank the frequency which they use the data, voice and SMS services available on their mobile phone. Responses were ranked as most used (most), moderately used (mod) and least used (least). Table 4.2 shows that overall, data was ranked as the most used service by each of the race groups and across female and male respondents. However, for female and Black African/Coloured respondents, this was tied with voice calls.

	All	**Female**	**Male**	**W**	**I/A**	**BA/C**
Data	Most	Most	Most	Most	Most	Most
SMS	Mod	Mod	Least	Least	-	Mod
Voice	Mod	Most	Mod	Mod	Mod	Most

Table 4.2: Frequency of use: Data versus SMS versus Voice

Q5: Which online data services do respondents make use of? Participants were asked two questions relating to the services they use on their mobile phones. Firstly, they were asked to specify the messaging services that they currently use. In addition, they were asked to indicate any other online data services they use on their mobile phones.

	All ($n = 93$)	Female	Male	W	I/A	BA/C
SMS	74%	69%	80%	92%	75%	72%
WhatsApp	66%	56%	76%	92%	75%	61%
BBM	9%	2%	16%	8%	50%	7%
Mxit	16%	13%	20%	0%	0%	19%
E-mail	30%	19%	41%	77%	75%	20%
Facebook	61%	46%	76%	92%	50%	57%
Twitter	21%	13%	29%	38%	25%	18%
News	26%	19%	33%	69%	25%	18%
Wikipedia	17%	7%	27%	46%	0%	14%

Table 4.3: Use of messaging services and online data services

Table 4.3 shows that SMS is the most popular messaging service, while Facebook is the most popular online data service across all gender and race groups. Interestingly, female respondents are less likely to use messaging and data services on their mobile phones when compared with male respondents. This supports the earlier finding where females rated voice as most frequently used, alongside data.

Q6: How often do respondents make use of the online data services? Of the 100 respondents identified as either owning or having access to a mobile phone, 92 answered the question of how often they made use of the online data services. Figure 4.3 shows that most respondents (46.74%) access data services at least once per day, while slightly more than a quarter (28.26%) never make use of any online data services. Slightly less than a quarter (22.83%) access online data services a few times per week, while 2.17% of respondents access them a few times a month.

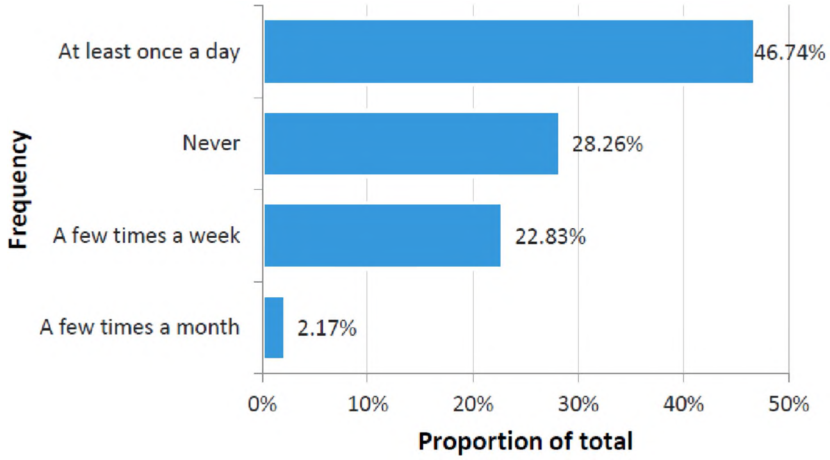


Figure 4.3: Frequency of online data services use ($n = 92$)

Q7: How many respondents have experience installing mobile applications? Of the 95 participants who own a mobile phone, 66 of these handsets were positively identified as being able to install applications. Out of these, 40 participants answered "yes" to having installed applications on their devices, translating into an installation rate of 60.6%. A distinction was also made between installing applications via the handset's store (i.e. Apple App Store, Google Play, Windows Store) or manually (i.e. downloading a Java ME `.jar` installation file from a website). Of the 66 participants with handsets capable of installing applications, 40.90% have installed applications using both their device's store, as well as manually. These results highlight the importance of providing adequate support and training to residents regarding the installation of the client applications. Additionally, selecting appropriate channels through which to distribute the client applications is vital, whether via the MobiSAM website, platform-specific application stores or a combination of these.

4.5.3 Current participation

The third and final section of the baseline questionnaire asked participants to specify how they currently participate with local government around the area of service delivery. In response to the question: "Have you ever made a complaint to your local municipality?", 58.1% of all participants stated that they had complained (the total responses for this question was 105, $n = 61$). Table 4.4 shows the results when analysing the different ways in which people complain by gender and race. In the 'Other' category, one participant stated that they had used Facebook to report a problem to the municipality.

When considering responses by gender, both female and male participants preferred to speak in person to staff at the municipality. Anecdotally, this may be seen to be due to the fact that municipal officials seldom answer the phone at the offices when they are called. When analysing by race, White residents showed a preference for signing petitions, with all other residents preferring to speak to someone in person at the municipality.

	All ($n = 62$)	Female	Male	White	I/A	BA/C
Petition	25%	23%	27%	80%	0%	20%
Spoke to municipality	46%	42%	50%	20%	100%	46%
Phoned municipality	7%	13%	0%	0%	0%	7%
Ward councillor	20%	19%	20%	0%	0%	22%
Letter	2%	3%	0%	0%	0%	2%
Other	2%	0%	3%	0%	0%	2%

Table 4.4: Complaints to municipality

The next question asked respondents if they had participated in a protest or demonstration within the last 12 months. One respondent chose to not answer this question ($n = 104$). As shown in Table 4.5, 39.42% of all respondents said that they had participated. When analysing by gender, females participated more than males. Interestingly, there is a large split in participation between race groups, with White and Indian/Asian respondents participating on average far less than their Black African/Coloured counterparts.

	All ($n = 104$)	Female	Male	W	I/A	BA/C
Yes	39%	45%	33%	15%	25%	44%
No	61%	55%	67%	85%	75%	56%

Table 4.5: Participation in protest

Another question asked to residents was how much agency they believed they had to change the system. One respondent did not answer this question ($n = 104$). Table 4.6 indicates that over half of all participants believed they could make "some" or "a great deal" of difference to improve the situation in local government. When considering the results by gender, males responded more negatively ("nothing") to the question, but paradoxically also responded more than females to the positive response ("a great deal").

	All ($n = 104$)	Female	Male	W	I/A	BA/C
Nothing	10%	6%	16%	31%	25%	7%
A small amount	33%	33%	33%	54%	25%	31%
Some	26%	31%	20%	8%	0%	30%
A great deal	28%	26%	29%	0%	25%	32%
I don't know	3%	4%	2%	8%	25%	1%

Table 4.6: How much agency for changing system

Interestingly, White participants responded more negatively that any other race group, with 84.62% of respondents saying an individual could make no change or "a small amount".

4.6 Analysis and discussion

The results from the baseline study highlighted several important findings which ultimately helped guide the design of the cross-platform MobiSAM client applications. This section draws together and discusses each of them.

4.6.1 Mobile phone prevalence and application support

Mobile phone use within Grahamstown is pervasive. With 95.24% of respondents indicating that they either own or have access to a mobile phone, it is clear that making MobiSAM accessible via these devices is vital to the project's success. Over two-thirds (69.47%) of participant handsets were positively identified as being able to install applications, indicating that this is a potentially viable means through which to provide access to the MobiSAM service. Although residents pursue many different avenues when lodging complaints with their municipality (see Table 4.4), mobile phones have the potential to help increase participation by "lessening the level of motivation that participation requires" [60, p.45]. In addition, with more than half (53.33%) the respondents believing that they can make "some" or "a great deal" of difference in helping improve instances of inadequate service delivery (see Table 4.6), it is likely that the majority of respondents will seek avenues of complaint (such as those provided by MobiSAM) when service delivery problems arise, rather than being apathetic.

4.6.2 Language preference

Examining demographics results from the study, 98.1% of participants preferred to use either isiXhosa, English or Afrikaans. Of the total participants surveyed, 97.14% were able to both read and write in their preferred language. This information provides two different critical pieces of information for the decision to develop client applications. Firstly, it provides important information on the literacy level of the sample population. It is interesting to see that the literacy level is higher than the reported levels for South Africa (97.14% versus 92.9%) by Statistics SA [199]. Secondly, it shows that since the client applications provide localised UIs in isiXhosa, English and Afrikaans, these applications effectively target the preferred language of 98.1% of respondents. It should be noted that the questionnaire only asked what language participants preferred to use, rather than to list all languages understood (and their respective literacy levels in each). This indicates that participants could potentially be served by providing support in these three languages, albeit not in their first language.

4.6.3 Mobile platform environment

Figure 4.1 highlights that Grahamstown is a heterogeneous mobile device environment. Five different mobile operating systems were identified, with the most popular found to be referred to as "proprietary" by GSMArea[2]. These typically correlated to low-end handsets such as those produced by Nokia and Samsung, nearly four-fifths (79.41%) of which were still shown to offer Java ME support. In the case where actual operating system details were available, the most popular operating systems were: Android OS (23.16%), BlackBerry OS (11.58%), iOS (5.26%) and Windows Phone (1.05%). Therefore, as support is provided for these four platforms, as well as Java ME, 69.47% of all surveyed mobile devices are potentially being reached. Referring back to Figure 4.2, it is important to remember that nearly a quarter (23.16%) of mobile devices were not able to be identified exactly as the model number was not available, therefore these figures represent the worst case scenario. As highlighted by Q7 in Section 4.5.2, of the 66 respondents owning handsets capable of installing applications, more than half (60.6%) have done so in the past. While this may seem like a low proportion, respondents may simply be unaware that their handsets support installable applications, particularly those with Java ME-enabled feature phones.

4.6.4 Mobile data and online services

Despite the wide variety of mobile devices and platforms identified, there are some commonalities in the way that participants described the use of their mobile phones. Firstly, across all genders and race groups, data was ranked as most frequently used (this was tied with voice calls for female and Black African/Coloured respondents). While at first this seemed surprising as it is a newer mobile product than voice calls and SMS, the reason becomes clearer when considering the price of mobile data. Although SMS is still the most popular messaging service (see Table 4.3), WhatsApp is ranked a close second by all genders and race groups. The widespread adoption of WhatsApp suggests two important pieces of information. Firstly, at least 65.71% of participants' handsets are capable of installing WhatsApp, suggesting that a similar proportion of devices are able to install the cross-platform MobiSAM client applications (as the identical platforms are targeted). Secondly, it indicates that most participants are comfortable using third party applications. Interestingly, the number of participants who make use of WhatsApp is greater than the number who indicated to have installed applications (65.71% versus 60.6%). This may be due to the fact that either a family member or friend installed the application for them. Alternatively, the participant may have installed WhatsApp themselves while being unaware that they were actually installing an application. In addition, some handsets (i.e. Nokia Asha 210) come with WhatsApp pre-installed.

[2] GSMArena http://www.gsmarena.com/

4.6.5 Universal access

The popularity of SMS across all gender and race groups highlights its importance and potential for providing access to the MobiSAM service. While the cost per message sent is relatively high when compared to other messaging services listed in Table 4.3, SMS offers the advantage of being available to all GSM handsets, thereby potentially lowering the barrier of entry to MobiSAM. Furthermore, unlike installable applications, many participants (74.29%) are already familiar with how to compose and send SMS messages from their devices.

4.7 Summary

This chapter presented the baseline study. First, the study sample; data-collection instruments; and method and analysis were described in order to provide the reader with a clear understanding of how the study was conducted. The study results were then presented in full. Finally, an analysis and discussion of the results was performed, drawing together and highlighting several important findings which ultimately helped guide the design of the cross-platform MobiSAM client applications. The next chapter draws on the results of the baseline study, presenting the design and implementation of the MobiSAM client applications and SMS gateway application.

Chapter 5

Design and implementation

5.1 Introduction

This chapter provides an overview of the design and implementation of the cross-platform MobiSAM client applications. First, background information relating to MobiSAM is provided (Section 5.2), in order to acquaint the reader with the service. The system's underlying architecture is then presented (Section 5.3), providing a detailed account of the MobiSAM front-end. Finally, the design and implementation of the cross-platform client applications, MobiSAM (Sections 5.6 and 5.7) and MobiSAM Report (Section 5.8), are discussed.

5.2 The MobiSAM service

As described in Chapter 1, MobiSAM is a mobile phone-based polling framework and service delivery issue reporting tool, which seeks to promote active citizen participation between residents and their local government. The service allows residents to answer service delivery polls, as well as report service delivery problems to their local municipality using a mobile phone or personal computer (PC). Depending on the chosen method, additional details such as a photo of the issue being reported, along with GPS coordinates may be included in reports.

In addition, polls may be periodically conducted in order to gather information used for ongoing service delivery monitoring purposes. Makana Municipality, for example, might conduct a poll asking citizens: 'How is your water pressure today?'. Resident responses would then be collated and visualised by suburb (see Figure 5.1), providing the municipality with timely, relevant information which may potentially be used to help improve service delivery.

Three service delivery poll types exist, the details of which are briefly discussed next.

- **Custom** A number of predefined responses are presented, of which the user may select only one. Predefined responses are set by a MobiSAM administrator when creating a poll. This poll type allows for the creation of: yes or no; true or false; and Likert type questions, amongst others.
- **Text response** Allows for an open-ended written text response, accepting any combination of characters but restricted in length.
- **Upload images** Allows for the uploading of images and includes the GPS coordinates, address (in human-readable format e.g. 1 High Street, Grahamstown Central) and description of the image being uploaded. This poll type allows for the reporting of specific service delivery problems.

A MobiSAM administrator selects an appropriate poll type when creating new polls. Once a poll has been created its type cannot be changed, instead requiring a new poll to be constructed. Only users with administrator privileges may create new polls.

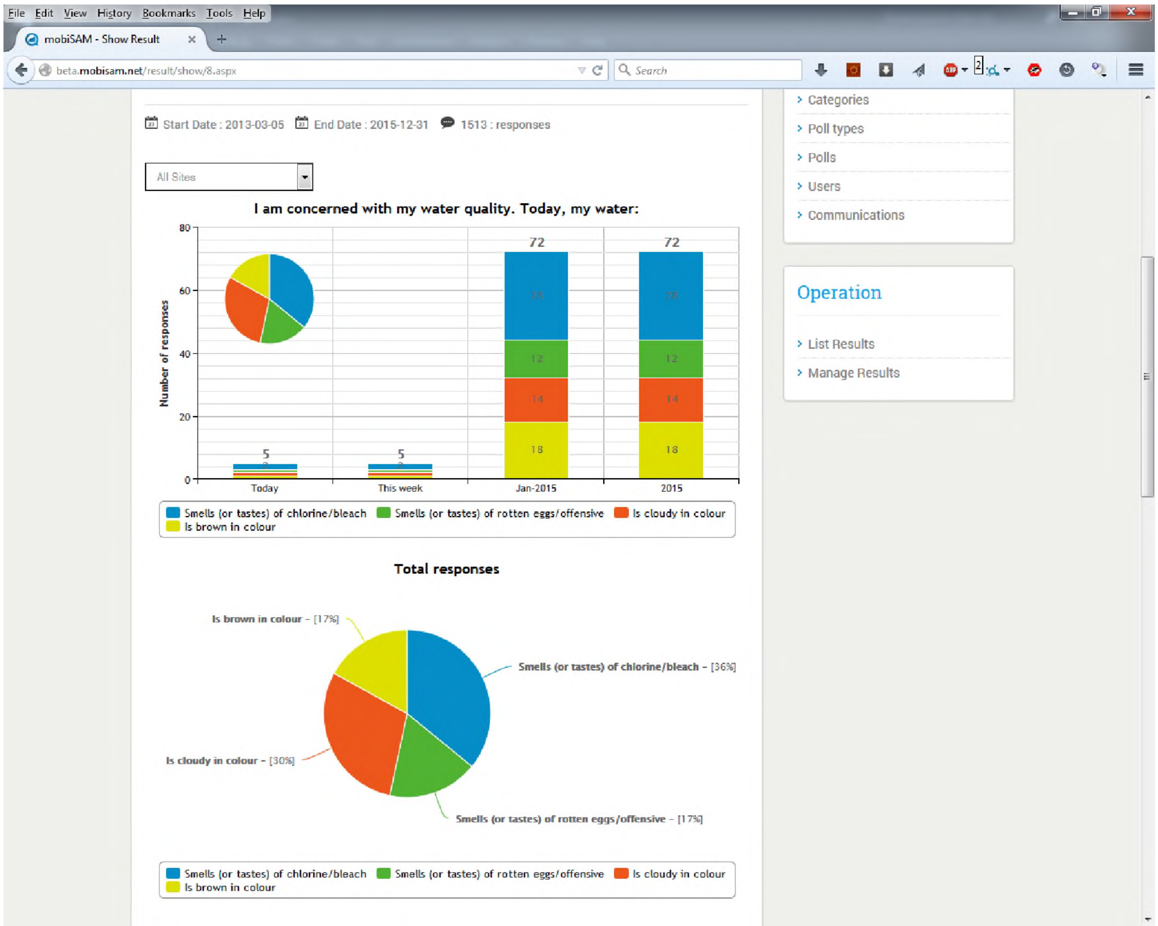


Figure 5.1: MobiSAM website: poll results visualisation

While MobiSAM shares similarities with the civic engagement project discussed in Chapter 2, many differences exist. Lungisa (see Section 2.3.4), like MobiSAM, aims to "monitor how public resources are spend, and to promote transparency, accountability, and better governance" [127, p.1], however, instead of encouraging local municipalities to interact directly with their service – thus creating a two-way dialogue – they instead forward service delivery complaints to the correct government department. Little emphasis is placed on the ongoing monitoring of the basic services provided by municipalities. Although Find & Fix (see Section 2.3.5) allows fault reports to be sent directly to the JRA, it provides no mechanism to monitor the quality of services which they offer. Platform support is also limited to Android and iOS, potentially excluding a large proportion of South Africa's population.

5.2.1 User registration

Before using the MobiSAM service, residents are first required to register via the website[1]. During the registration process, residents are asked to specify a username (unique), password, e-mail address (unique) and cellphone number, as well as basic demographic information such as first and last names, preferred language and optionally, their physical address. The information requested during registration corresponds to requirements obtained from Makana Municipality. User information is securely stored and used for communication purposes, as well as linking a user's poll responses and service delivery reports to a specific suburb within the municipal district. Once registered, an administrator reviews the provided information and manually activates the account before access to the service is granted. Currently, residents may only register via the MobiSAM website.

5.2.2 MobiSAM Facebook page

With over 9.4 million active users in South Africa [215], Facebook was seen as an important platform to help disseminate MobiSAM related information. Although not directly related to the MobiSAM service, a Facebook page[2] was created to establish MobiSAM's presence on the social network. The page is used to keep the public informed about potential service delivery problems within the Grahamstown area, thus providing information to residents who may not be registered MobiSAM users. As similar Facebook pages already exist (for example, *Grahamstown Municipal Services outage reporting*[3]), MobiSAM status updates are shared to these pages, helping keep residents informed while at the same time indirectly introducing MobiSAM to residents who have not yet heard of the service. Facebook users are able to 'like' the MobiSAM page should they wish to receive status updates. In addition, users are able to 'share' relevant MobiSAM

[1] MobiSAM Registration http://mobisam.net/user/register.aspx

[2] MobiSAM Facebook page http://www.facebook.com/mobisam.net

[3] Grahamstown Municipal Services outage reporting page https://www.facebook.com/pages/Grahamstown-Municipal-Services-outage-reporting/

status updates with their friends and followers. Information contained within MobiSAM status updates is obtained directly from Makana Municipality, Amatola Water (responsible for water reticulation within Grahamstown) and MBB Consulting Services (responsible for the bulk water supply within Grahamstown), thereby ensuring its authenticity and accuracy.

5.3 System architecture

When developing an application, it is vital that the system architecture aligns with the goals of the entire system. Although the research presented in this thesis concerns itself with the development of only a portion of the MobiSAM system, a visual overview of the entire system architecture is provided in Figure 5.2. From a high-level perspective, the MobiSAM system is comprised of two separate entities: a front-end (see Section 5.3.1), and a backend server (see Section 5.3.2). Furthermore, each entity is comprised of several different components. A client-server architectural model was followed throughout, with the backend server implementing its own *RESTful* API, leading to an increased separation of concerns. Importantly, this client-sever model helps facilitate the process of data exchange across different platforms [38], a characteristic vital to the success of the MobiSAM service.

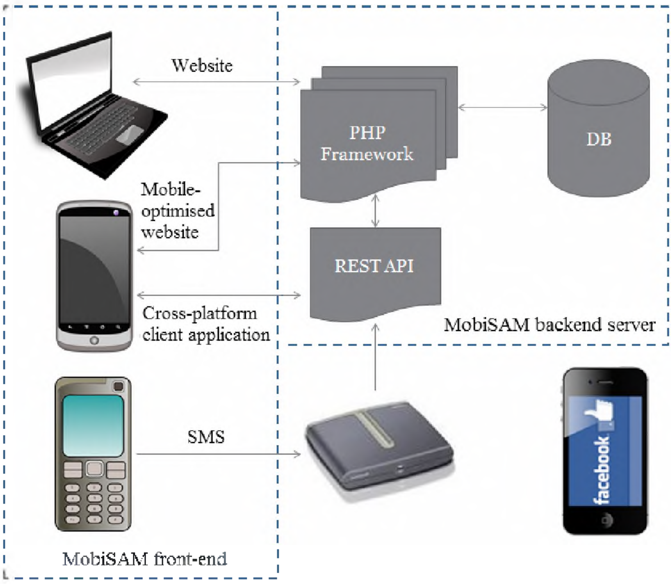


Figure 5.2: The MobiSAM system architecture

5.3.1 MobiSAM front-end

The front-end portion of the system concerns itself with how citizens interact with the MobiSAM service and consists of a variety of different methods. It was decided to include multiple access methods in an attempt to lower the barrier of entry, hence potentially maximising uptake of the service. This approach also allows the particular strengths of a particular platform to be leveraged, such as the simplicity of SMS or the GPS capabilities of a mobile device. During the requirements elicitation process, presented in Section 3.4, three distinct access methods were identified:

- A traditional and mobile-optimised website
- Installable client applications (apps)
- SMS

Each access method is briefly described next. Although the web portion of the system was developed by a separate researcher, a short discussion thereof is warranted, helping justify the need for additional access methods, such as client applications and SMS.

Website

Due to the prevalence of web access, as well as highlighted by the results of the baseline study (see Chapter 4), it was essential to provide access to the MobiSAM service via website. Accordingly, the website offers support for both mobile phones and PCs, adapting itself based on the host device. Although the traditional website offers full access to the MobiSAM service, functionality of the mobile-optimised website is restricted due to the different mobile web browser implementations of HTML5, as well as limited JavaScript support. As poll results, for example, are visualised using the Highcharts JS charting library[4] (see Figure 5.1), unsupported browsers do not provide any form of chart-based visualisation.

[4] Highcharts http://www.highcharts.com/

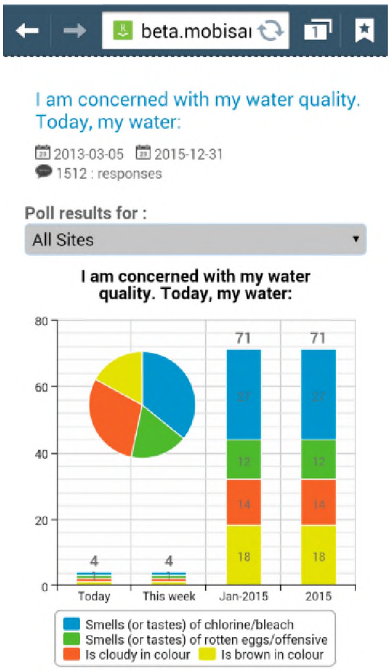


Figure 5.3: MobiSAM website: visualised poll results (running on an Android smartphone)

As an alternative, the mobile-optimised website offers limited functionality, while attempting to maximise browser support. Focus is instead placed on the answering of service delivery polls and reporting of service delivery problems. Should the browser accessing the MobiSAM website be supported by the Highcharts JS, poll results are visualised, as shown in Figure 5.3. Alternatively, a summary of the results appears in a numerical table, as illustrated in Figure 5.4. Highcharts JS does not provide a list of supported web browsers.

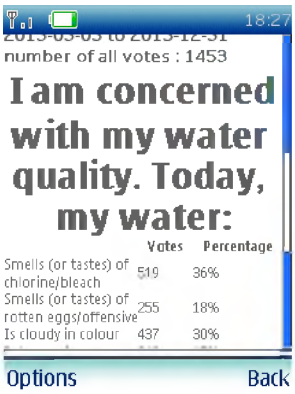


Figure 5.4: MobiSAM website: poll results (running on a feature phone)

Client applications

While the mobile-optimised website allows access to the MobiSAM service from a mobile device, its functionality and visualisation capabilities are wholly dependent on the handset's web browser. As an alternative, it was decided to investigate the use of platform specific, installable client applications. This decision was motivated by several factors, specifically:

- mobile applications potentially allow for a more engaging user experience;
- enhanced visualisation capabilities may be incorporated into mobile applications;
- built-in GPS and camera capabilities of mobile devices may be leveraged;
- potentially less bandwidth consumption, once a client application has been downloaded and installed; and
- browser quirks and inconsistencies are avoided.

Initially, two high-fidelity prototype client applications were developed natively (see Appendix A.2), targeting what results from the informal retail outlet survey (see Section 3.5) revealed to be the best-selling mobile platforms within Grahamstown: Java ME and BlackBerry OS. While this approach offered a great deal of flexibility, the time and effort required made the task impractical. The lessons learned from the native development process resonated with what Charland and Leroux wrote in their paper *Mobile Application Development: Web vs. Native*, quoting Google's VP of engineering: "even Google was not rich enough to support all of the different mobile platforms from Apple's App Store to those of the BlackBerry, Windows Mobile and the many variations of the Nokia platform" [28, p.50].

As an alternative, several cross-platform mobile frameworks were identified and evaluated, as is described in Section 5.5.

Short Message Service (SMS)

The pervasive nature and widespread use of SMS, highlighted by the results of the baseline study (see Chapter 4), made it an important medium through which to allow the reporting of service delivery problems. In addition, being able to report issues from *any* GSM handset meant that SMS had the potential to lower the barrier of entry to the MobiSAM service. As part of this research, an SMS gateway application was developed allowing residents to submit reports to MobiSAM via traditional SMS. Received reports are collated by the backend server and displayed chronologically via its web interface, as shown in Figure A.9. The design and implementation of the SMS gateway is presented in Section A.6.

5.3.2 MobiSAM backend server

Although development of the system's backend fell outside the scope of the research presented in this thesis, a brief description is warranted in order to provide the reader with an understanding of the system as a whole. Please refer to electronic Appendix A.1 for an overview of the backend server.

5.4 Prototype MobiSAM client applications

As part of initial development efforts, two high-fidelity prototype client applications were developed natively, using platform specific SDKs. Each application targeted what results from the informal retail outlet survey revealed to be the two best-selling mobile platforms within Grahamstown: Java ME and BlackBerry OS (see Section 3.5). Client applications were functionally identical and served as a basic proof of concept, allowing an informal usability evaluation to be conducted (see Section 3.6). Due to space restrictions, a detailed account of their design and implementation has been included as an electronic resource. Please refer to electronic Appendix A.2 for design and implementation details.

5.5 Cross-platform mobile framework selection

This section provides a brief overview of the cross-platform mobile framework selection process. The drawbacks of developing mobile applications natively are first discussed (Section 5.5.1), followed by the cross-platform mobile framework selection process (Section 5.5.2).

5.5.1 Native development drawbacks

Feedback gleaned from the informal usability evaluation (see Section 3.6), highlighted several shortcomings associated with developing client applications natively:

- developing multiple applications is resource intensive;
- a broad skill set is required to support and develop applications for multiple platforms; and
- ongoing maintenance requirements are high.

Similar disadvantages have been identified by other researchers [28, 81, 86, 165, 170, 210]. As a result, it was decided to redevelop the MobiSAM client application anew, using a cross-platform mobile framework. This approach sidestepped the need to develop the client application multiple times (albeit for different platforms), while at the same time allowing a greater number of platforms to be targeted.

5.5.2 Selection process

Given the multitude of cross-platform mobile frameworks available, a number of predefined criteria needed to be met before a framework was considered for use. These criteria were:

- provide support for Android, BlackBerry OS, iOS, Java ME and Windows Phone platforms;
- offer natively installable applications as opposed to leveraging JavaScript-based `Webview` components;
- support either Java or .NET C# as their implementation language; and
- be open source and/or offer affordable licensing terms.

Cross-platform mobile frameworks which leveraged embedded `Webview` components instead of offering natively installable applications were excluded due to patchy support from Java ME and BlackBerry OS. Since a `Webview` is essentially a 'chromeless' browser instance which is able to call native platform code from JavaScript [28], this additional overhead had to be avoided, especially when considering the limited processing capabilities of low-end handsets.

Table 2.2 lists the identified cross-platform mobile frameworks, 23 in total, of which only three met the specified criteria: Codename One (see Section 2.7.2), J2ME Polish (see Section 2.7.2) and NeoMAD (see Section 2.7.2). After conducting a thorough review of each framework's documentation and website, both Codename One (CN1) and J2ME Polish were earmarked as the most fitting solutions. NeoMAD was excluded due to its high licensing fees.

Given that the platform support and implementation language for CN1 and J2ME Polish were both similar, the final selection process needed to be based on additional merits. As such, a simple demonstration application was developed using both CN1 and J2ME Polish. The application attempted to determine how well each framework performed within the context of the MobiSAM service. For more information about the demonstration application please refer to [181]. The results of the evaluation highlighted five distinct advantages of CN1 over J2ME Polish:

- improved cross-platform support;
- cloud-based build server;
- rapid UI prototyping and construction via the CN1 designer tool;
- full Java implementation instead of restricted Java ME; and
- an active online community.

Consequently, CN1 was chosen for use when developing the cross-platform client applications, the design and implementation of which are described next.

5.6 Cross-platform MobiSAM client application

This section provides an overview of the design and implementation of the cross-platform MobiSAM client application. The updated requirements and specifications are first presented (Section 5.6.1). Details relating to the client application's UI and navigation (Section 5.6.2), visualisation (Section 5.6.3), as well as other miscellaneous features are then described (Section 5.6.4). The cross-platform client application was first developed during Iteration 2 of the life-cycle model (see Section 3.4.2).

Due to space restrictions, a detailed account of the technical implementation has been included as an electronic resource. Please refer to Appendix A.3 for more information.

5.6.1 Requirements and specifications

Feedback from the informal usability evaluation helped provide an updated set of requirements (see Section 3.6). These requirements were then clarified and documented, leading to the development of the corresponding specifications. Modifications included:

- organising available polls by category instead of presenting them all on a single screen;
- navigating to the list of available polls instead of returning to the *Home* screen after closing the *Results* screen; as well as
- including interactivity within the chart visualisations.

In addition, the two expert users taking part in the evaluation recommended transitioning from multiple platform-specific codebases to a single codebase, by leveraging a suitable cross-platform mobile framework.

5.6.2 User interface and navigation

As an application's UI is the primary means through which user interaction takes place, it is important that it offers the required functionality in a simplistic manner. Galiz, in his book *The Essential Guide to User Interface Design* contends that: "The user interface is the most important part of any computer system. Why? It *is* the system to most users" [54, p.1]. Accordingly, the client application follows a simple menu-driven UI style, in an attempt to guide the user through the process of answering service delivery polls. A strong contrast between UI elements was created, helping the client application remain legible in harsh lighting conditions. The variable usage context of the client was also kept in mind throughout its development, paying particular attention to externalities such as usage environment, time constraints and handset limitations. In addition, known platform UI and UX guidelines were strictly adhered to, in an attempt to maximise the client application's usability, as well as improve its familiarity. The CN1 designer was leveraged when constructing client application screens, thus separating design from implementation. Importantly, the speed and flexibility offered by the CN1 designer encouraged the construction and testing of multiple versions of the same screen, potentially leading to more usable solutions.

For information relating to the technical implementation details of the client application's UI, please see electronic Appendix A.3.2.

***Login* screen** After launching the client application, the user is presented with the *Login* screen, shown in Figure 5.5a. The screen contains the text 'Please Log in to Your Account' in its title, with the MobiSAM logo positioned below. Next, a text box where the user may enter their login credentials is provided. Text boxes are masked so as to only allow certain characters, thereby reducing potential security vulnerabilities. Two options are provided next. The first option ('Keep me logged in') allows the user to remain logged in between different invocations of the client application, while the second ('Lost password?') allows the user to request a new password be sent to their provided e-mail address. Following platform design guidelines, two soft keys ('Log in' and 'Exit') are visible when used on Java ME. These appear as regular buttons on the remaining platforms.

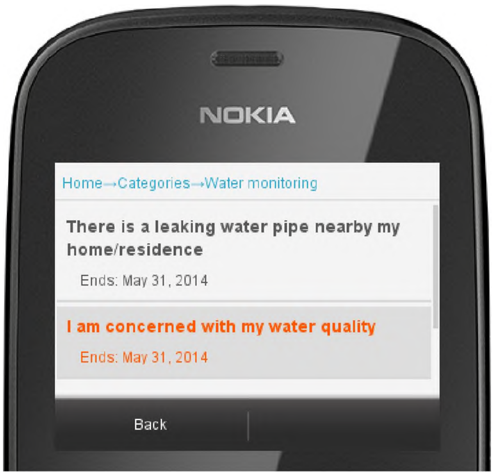
***Home* screen** The *Home* screen provides quick access to the client application's main functions: 'Take a poll', 'Update profile' and 'Log out'. Each menu option contains a second line of text, shown in Figure 5.5b, describing its purpose. While a 'Log out' soft key is visible on the Java ME platform, a 'Log out' button appears in the top-left corner when used on iOS, in accordance with platform conventions. This action is performed using the hardware back button on the remaining platforms. Before being logged out, the user is prompted to confirm their choice.

***Poll category* screen** Selecting the 'Take a poll' menu option navigates the user to the *Poll category* screen, shown in Figure 5.5c. This screen displays the available categories retrieved from the backend server, along with a short description of each. The screen differs from that of the prototype client applications (see Appendix A.2) in that polls are grouped according to specific categories, potentially allowing the user to find polls of interest in less time. Results from the informal usability evaluation (see Section 3.6) guided the redesign of this screen. In an attempt to simplify navigation, a 'breadcrumb' trail is provided in the screen title, outlining the steps followed to get to the current screen. A 'Back' soft key is visible on the Java ME platform,

(a) *Login* screen

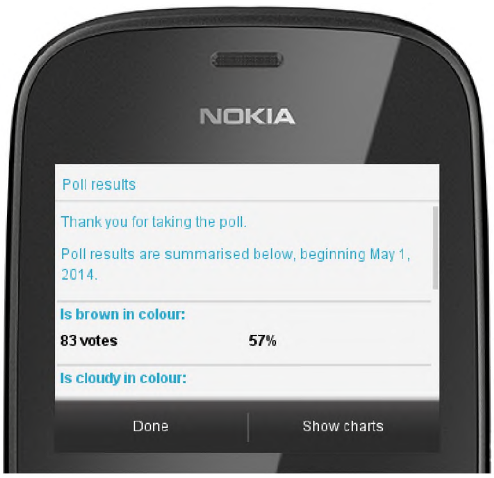
(b) *Home* screen

(c) Available categories



(d) Available polls with the 'Water monitoring' category

(e) Responding to a poll



(f) *Results* screen

Figure 5.5: Cross-platform client application

allowing the user to return to the *Home* screen. This action is performed using a regular button on iOS, or hardware back button on the remaining platforms.

***Available polls* screen** After selecting a category, the user is navigated to a screen displaying the available polls within the chosen category. The 'Water monitoring' category, for example, contains the polls shown in Figure 5.5d. Although three polls exist within the category, only two are shown due to the simulator's limited display resolution. As with the previous screen, a 'breadcrumb' trail of the steps taken is shown in the screen title.

***Poll response* screen** Having selected a poll, the user is presented with a screen where they are able to respond to the poll. Available responses are dependent on the type of poll the user is answering (see Section 5.2). While the *custom* poll type provides a set of predefined responses as radio buttons, a *text response* provides a text box into which an open-ended response may be typed. *Upload images* poll type is used exclusively for the reporting of service delivery complaints, as will be described in Section 5.8.3. The client application detects the poll type and adapts the *Poll response* screen accordingly. Figure 5.5e shows an example of a *custom* poll type containing four predefined responses, of which three are visible. The user selects the most fitting response and selects 'Submit' to confirm their answer or 'Cancel', which returns them to the previous screen.

***Results* screen** Once a poll response has been successfully submitted, the user is presented with the *Results* screen. In an attempt to promote transparency, a summary of all citizen responses to the given poll is provided (excluding *text response* and *upload images* poll types, due to their open-ended nature). Result summaries take the form of a numerical table (see Figure 5.5f), as well as pie and bar charts which are displayed by pressing the 'Show charts' soft key (see Figure 5.7). If running on Android, iOS or Windows Phone, interactive charts are provided. After closing the *Results* screen, the user is navigated back to the *Available polls* screen, allowing them to continue answering service delivery polls. This is unlike the prototype client applications (see Appendix A.2), which returned to the *Home* screen. The design of the chart visualisations is described in Section 5.6.3.

***Update profile* screen** The 'Update profile' menu option appears second on the *Home* screen. Selecting this menu option navigates the user to the screen shown in Figure 5.6a, allowing the user to modify their profile. Options include changing their preferred language (between English, isiXhosa and Afrikaans), as well as subscribing to, or unsubscribing from MobiSAM SMS and e-mail updates. Updates are sent to the user's contact details provided during registration. Changes are applied by selecting the 'Save' soft key on Java ME handsets, or 'Save' button on the remaining platforms.

(a) *Update profile* screen presented in isiXhosa

(b) *Information* dialog

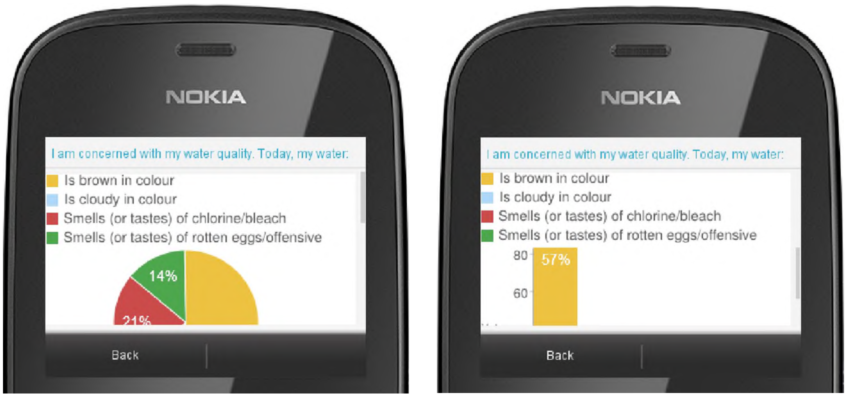
Figure 5.6: Cross-platform client application

The final *Home* screen menu option, 'Exit', closes the client application, releasing all allocated device resources. If the user selected 'Keep me logged in' during the login procedure, they will remain logged in after closing the client application. This causes the *Login* screen to be bypassed the next time the client application is launched, instead navigating the user directly to the *Home* screen. Java ME handsets display a 'Log out' soft key (see Figure 5.5b), allowing the user to safely log out of their account. This action is provided via a regular button on iOS, or hardware back button on the remaining platforms.

***Information* dialog** A generic dialog, shown in Figure 5.6b, appears should a problem arise during use of the client application. For example, should the user incorrectly enter their credentials when attempting to log on, or the backend server is unable to be reached, the necessary informational text is provided, keeping the user informed. All network related operations timeout after 30 seconds, after which time the user is also presented with the *Information* dialog.

5.6.3 Visualisation

A motivating factor behind the development of installable client applications was their potential to allow for a more immersive and engaging user experience when compared to the mobile-optimised website. Visualisations within the client application were designed to be clear and concise, in an attempt to swiftly convey poll results to the user. As such, three different visualisation types were included: a pie chart, a bar chart and a numerical table. While Few, in his article *Save the Pies for Dessert*, argues that pie charts are less apt for accurately conveying information when compared to bar charts [44], it was decided to include several different visualisation types within the client application. The effectiveness of each would then be assessed during the usability evaluation (see Chapter 6).



(a) Pie chart (b) Bar chart

Figure 5.7: Static image charts

Although CN1 supports a wide variety of mobile platforms, not all platforms offer identical functionality. Depending on the host platform, two different visualisation approaches were taken within the client application. The first approach is used when the client application is installed on a Java ME or BlackBerry OS device, and visualises poll results using static image charts. The second approach, which provides interactive chart visualisations, is used by the remaining platforms.

For more information relating to the technical implementation details of the chart visualisations, please refer to electronic Appendix A.3.3.

Static chart design

When designing the chart visualisations, several visualisation principles were paid close attention (see Section 2.8.1). Depending on the poll question, *custom* poll type results can be categorical or ordinal, 1-dimensional data (see Section 2.8.1). As such, appropriate chart types were chosen in order to best visualise it. Pie and bar charts held particular merit due to their widespread use and simple design. In order to leverage these chart types effectively, visual variables proposed by Ward *et al.* were consulted, including position, mark, size, brightness and colour (see Section 2.8.1).

As shown in Figure 5.7, a brightly coloured palette was used for each chart, in order to maximise legibility. Each series is represented by a unique colour and its corresponding percentage value is rendered within each. The limitations of mobile visualisations, highlighted by Chittaro [29], were kept in mind throughout the design process, making sure that the charts were able to convey meaning in a variety of physical environments.

Due to the limited display resolution of the Nokia Asha 201 handset, the user is required to scroll down to view the entire chart. This decision was made in order to provide suitably sized, clearly legible chart visualisations on low-end handsets.

Interactive chart design

Suitably chosen visualisations have been shown to enhance human cognition [112], while interactivity provides a means for the user to directly engage with the visualisation process. This interactivity both encourages and supports the exploration process, helping expose the underlying information [202]. Eager to accrue these potential benefits, it was decided to include basic interactivity within each of the chart visualisations. This functionality was then critiqued via usability evaluation in order to determine the real-world advantages it provided to the user (see Chapter 6).

(a) Overview (b) Automated panning and zooming (c) Information dialog

Figure 5.8: Interactive pie chart

Instead of presenting details directly on the charts themselves, it was decided to implement 'details-on-demand', a task defined by Shneiderman in his collection of seven abstract tasks (see Section 2.8.2). After responding to a service delivery poll, and selecting 'View pie chart', the user is navigated to a blank chart, shown in Figure 5.8a. This chart provides a graphical overview of the poll results, without detailing any actual values. The user is then able to tap areas of interest, which the screen then automatically pans and zooms into, as illustrated in Figure 5.8b. After the transition has completed, a dialog appears, revealing information about the selected series, as shown in Figure 5.8c. Information includes the name, value and percentage of the selected chart portion. The dialog is dismissed by pressing 'OK', after which the chart pans and zooms out, providing an overview of the entire chart once again.

Numerical table design

As an alternative to chart visualisations, poll results are arranged in a numerical table (see Figure 5.5f). The table is presented to the user directly after answering a poll and aims to provide a quick summary of responses to a particular poll. During the usability evaluation the numerical table was evaluated alongside the chart visualisations, in an attempt to determine which technique was more apt at conveying the necessary information (see Chapter 6). As with the static image charts, the user is required to scroll to reveal more information when used on low resolution devices.

5.6.4 Miscellaneous

Several miscellaneous features were implemented within the client application to ensure that user requirements were met. Three of these are described next. For more information relating to the technical implementation details of each, please consult electronic Appendix A.3.4.

Connectivity

The client application interacts with the backend server via a *RESTful* API (see Appendix C). The API is reached over HTTP, providing the REST routes required by the client. The necessary payloads are contained within the body of HTTP requests and responses are JSON-encoded. JSON was chosen over XML due to the performance gains it offers, as well as smaller object size (see Section 2.5.5). These characteristics make JSON well suited for low-powered handsets, while at the same time potentially reducing bandwidth consumption, and thus the data costs associated with using the client application.

Localisation

As highlighted by the 2011 National Census [52] and subsequently reiterated by the baseline study presented in Chapter 4, isiXhosa, English and Afrikaans are the most widely used languages within Grahamstown. It was therefore decided to offer support for each of them, potentially reducing any language barriers associated with using the MobiSAM service. The client application's language is modified from within the *Update profile* screen, depicted in Figure 5.6a. After a different language is selected and the 'Save' button pressed, all UI text is displayed in the chosen language. All poll questions retrieved from the backend server also appear in the preferred language.

Data persistence

Data persistence allows the client application to preserve its settings from one invocation to the next. When storing a preference, a key is specified along with its corresponding value. This value may then be retrieved by using its key. Within the client application, the user's preferred language, as well as SMS and e-mail alert preferences are saved to persistent storage. If 'Keep me logged in' is selected during log in, the necessary session information is also preserved.

5.7 Updated cross-platform MobiSAM client application

This section provides an overview of the modifications made to the cross-platform MobiSAM client application, guided by the results from the usability evaluation (see Chapter 6). First, the updated requirements and specifications are presented (Section 5.7.1). Design changes made to the UI and navigation (Section 5.7.2), as well as visualisation are then detailed (Section 5.7.3). Modifications discussed in this section were performed during Iteration 3 of the life-cycle model (see Section 3.4.3).

Due to space restrictions, a detailed account of the technical implementation has been provided as an electronic resource. Please refer to electronic Appendix A.4 for more information.

5.7.1 Requirements and specifications

The usability evaluation, presented in Chapter 6, aimed to determine how intuitively participants were able to navigate within the cross-platform client application. In addition, the study investigated how effectively different visualisations were able to convey meaning, as well as evaluate the role which chart interactivity played in helping participants better understand poll results. Feedback from the study identified several shortcomings relating to both the visualisation capabilities and usability of the client application (see Section 6.8). These shortcomings lead to an updated set of requirements, which were then clarified and noted, extending the client application's original specifications. The following updated requirements were identified:

- shift the focus of the client application away from the answering of polls, to the reporting of *specific* service delivery issues;
- remove any ambiguous category names;
- remove bar chart visualisations;
- improve affordance to better indicate chart interactivity;

- allow the user to update their current suburb from within the client application, as well as provide suburb-specific results; and finally,
- prompt the user to save changes before navigating away from the *Update profile* screen.

While not all feedback and suggestions were included in the updated requirements, all issues raised by participants were considered. Issues that appeared more than once were given particular attention.

5.7.2 User interface and navigation

As the MobiSAM service was initially envisioned as a mobile phone-based polling framework, enabling the monitoring of basic municipal services through the regular answering of service delivery polls, the 'Take a poll' menu option seemed fitting. Results from the first usability evaluation, however, revealed that participants were more concerned about being able to report *specific* service delivery grievances (see Section 6.8). Consequently, a selection of new screens were designed in order to reflect the client application's shift in focus. Each new screen, as well as modifications to existing screens are described next.

For information relating to the technical implementation details of the client application's UI, please see electronic Appendix A.4.1.

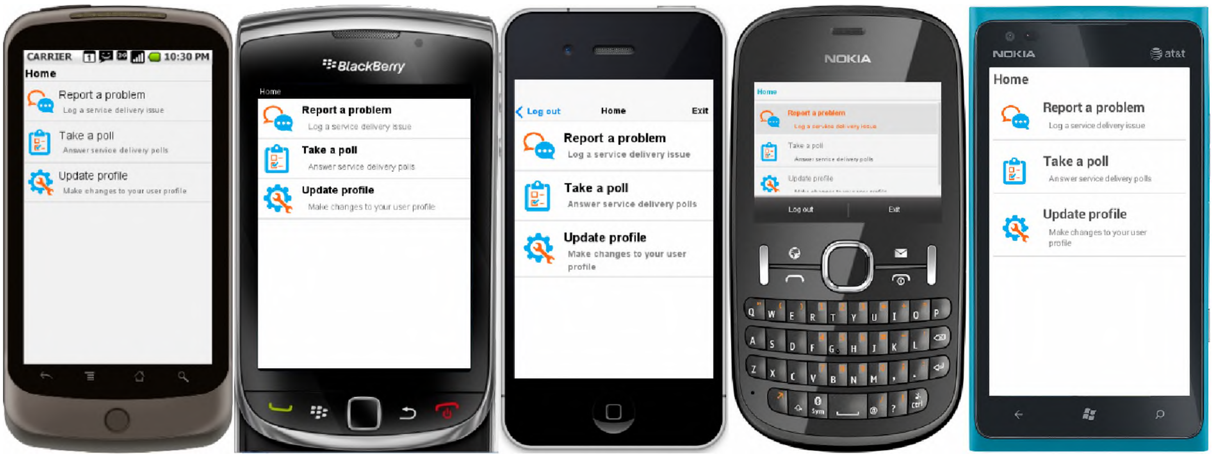


Figure 5.9: Updated cross-platform client application running on all supported platforms (not to scale)

***Home* screen** In an attempt to shift the client application's focus, the *Home* screen menu was modified to include the option 'Report a problem', as illustrated in Figure 5.9. Importantly, the updated client application continues to allow the user to respond to service delivery polls via the 'Take a poll' menu option. Stylised icons are now included alongside each menu option, further clarifying their purpose. In addition, the 'Exit' menu option has been removed and placed as a soft key on the Java ME platform. This action is performed using the hardware back button on Android, BlackBerry OS and Windows Phone, prompting the user to confirm their action. A regular 'Log out' button is provided on iOS.

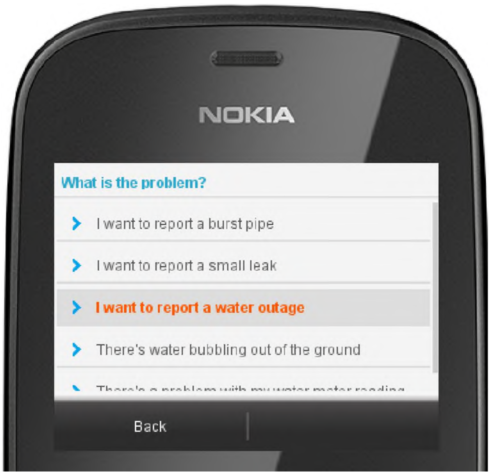
***Report category* screen** Pressing the 'Report a problem' menu option navigates the user to the newly designed *Report category* screen, shown in Figure 5.10a. The four basic services each municipality is tasked to provide are listed here: water, electricity, roads and sanitation. An additional 'Other' menu option is included for unrelated service delivery issues. Similar to the *Home* screen, descriptive icons are included alongside each menu option. In an attempt to help guide the user through the reporting process, it was decided to include context sensitive information in the title of the screen. The *Report category* screen, for example, shows the text 'What type of problem is it?' as opposed to detailing the steps taken: 'Report a problem→Report category'.

***Available reports* screen** After selecting the desired report category, the user is presented with a list of available reports for the chosen category. As illustrated in Figure 5.10b, a blue arrow is included alongside each report. Once again, the screen title contains context sensitive instructional text: 'What is the problem?'.

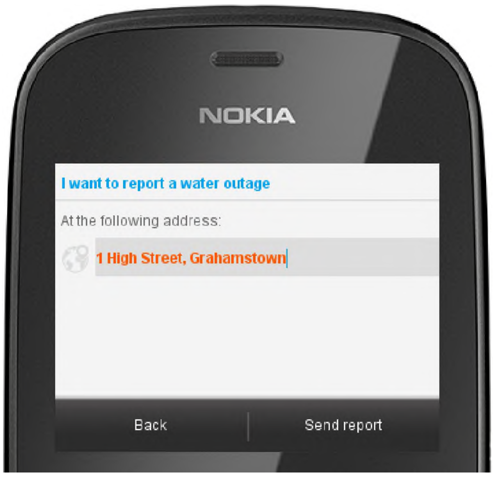
***Reporting* screen** Figure 5.10c illustrates the *Reporting* screen. The screen's minimal design is geared towards the reporting of service delivery problems, as opposed to the answering of service delivery polls. The name of the problem currently being reported appears in the title of the screen. For example, if 'I want to report a water outage' is selected from the *Available reports* screen, this text is shown in the screen title. A text box is provided into which the user may enter the address nearest to the problem being reported, taking the form of a *text response* poll type (see Section 5.2). This is required so as to inform the municipality about the exact whereabouts of the issue. Two soft keys, 'Back' and 'Send report' appear when running on the Java ME platform. The back action is accessible via the hardware back button on Android, BlackBerry OS and Windows Phone devices, while 'Send report' appears as a regular button. Both of these actions are represented by regular buttons on the iOS platform.

***Report sent* screen** After sending a report the user is presented with the *Report sent* screen, shown in Figure 5.10d. As the backend server does not provide a summary of recently received *text response* poll responses, no form of visualisation is provided. Instead, the screen simply informs the user that their report has been received and will be attended to within 24 hours. Selecting 'Done' returns the user to the *Home* screen.

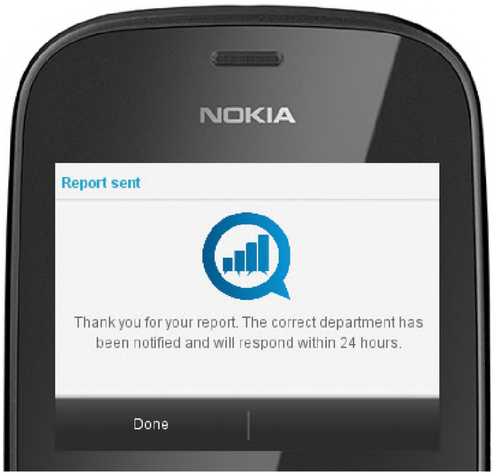
(a) *Report category* screen



(b) Available reports for 'Water'



(c) Reporting a water outage



(d) *Report sent* screen

Figure 5.10: Updated cross-platform client application

***Poll category* screen** As with the original cross-platform client application, selecting the 'Take a poll' menu option allows the user to respond to a variety of service delivery polls. First, the user is presented with the *Poll category* screen which displays all available poll categories retrieved from the backend server. While the screen remains virtually unchanged from the original client application, results from the usability evaluation highlighted the importance of providing unambiguous category names (see Section 6.8). In an attempt to remedy potential ambiguities, each category name and category description was renamed via the MobiSAM web-interface. In addition, blue arrows are now included alongside each available category, as shown in Figure 5.11a. The screen title text was also modified and instead of showing a 'breadcrumb' trail of the steps performed, now shows the context sensitive text: 'Select a category'.

***Available polls* screen** After selecting a category, the user is navigated to the *Available polls* screen. Modifications made to this screen include the inclusion of arrows alongside each available poll as well as updated screen title text (see Figure 5.11b). Instead of showing a 'breadcrumb' trail, context sensitive text takes its place: 'Select a category'.

***Poll response* screen** Selecting a poll navigates the user to a screen where they are able to respond to the poll. Design modifications made to this screen were cosmetic and included bolding the question text (for example, **'How is your water pressure today?'**) in order to better attract the user's attention. In addition, the screen title now provides instructional text to the user: 'Select a response'. Figure 5.11c depicts the updated *Poll response* screen.

***Results* screen** Finally, after submitting a poll the user is presented with the *Results* screen. As illustrated in Figure 5.11d, the screen contains a numerical table summarising other residents' responses. In accordance to feedback received from the usability evaluation, the screen now only shows responses from residents within the user's current suburb, as opposed to Grahamstown-wide results. In addition, minor cosmetic changes were made, including rephrasing of the screen text. A soft key named 'View chart' is visible when running on the Java ME platform. This appears as a regular button on the remaining platforms. Pressing 'View chart' takes the user to the updated *Visualisation* screen, which will be detailed in Section 5.7.3.

***Update profile* screen** The user is now able to update their current suburb from directly within the client application, as shown in Figure 5.11e. All other options remain unchanged.

5.7.3 Visualisation

A primary objective of the usability evaluation was to determine how effectively different chart visualisations convey poll results to users (see Chapter 6). In particular, whether interactive

(a) Available categories

(b) Available polls

(c) Answering a service delivery poll

(d) *Results* screen

(e) *Update profile* screen

(f) Static pie chart

Figure 5.11: Updated cross-platform client application

charts better help users understand poll results. While participant feedback relating to chart interactivity was positive, three-quarters (75%) of participants did not use this functionality (see Section 6.8.4). Upon further examination of the results, this may have been due to the fact that participants were unaware that the charts were interactive. Participants also indicated that instead of providing a blank chart (see Figure 5.8a), which users are required to 'tap' to reveal details, they would prefer basic values to be included on the charts.

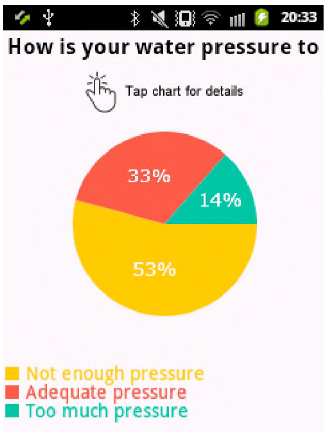


Figure 5.12: Updated interactive pie chart

This allows additional details to be revealed by tapping different areas of the chart. As participants showed a strong preference towards pie charts, bar charts were removed from the client application. As with the initial cross-platform client application, static charts are leveraged by Java ME and BlackBerry platforms, while interactive charts are used by the remaining platforms.

For more information relating to the technical implementation details of the chart visualisations, please refer to electronic Appendix A.4.2.

Static chart design

Figure 5.11f shows the newly designed static image chart. Sectors of the pie chart are arranged clockwise in order of magnitude, with the size of each depicted as a percentage. Sectors are coloured differently to the original pie chart, making use of a more vivid colour palette. This was done in an attempt to further improve the chart's legibility. The chart now also makes better use of the available display area, allowing it to be viewed in its entirety without scrolling, thus reducing load on the user's short term memory. In addition, the legend is now positioned alongside the chart when viewed on a handset with a landscape display. While appearing below the chart when viewed on a portrait display.

Interactive chart design

Modifications made to the interactive chart aimed to draw attention to its interactivity. As such, the text 'tap chart for details' along with a descriptive icon are now included above the chart (see Figure 5.12). When viewing an overview of the chart, percentages are now displayed within each sector. All other options remain unchanged.

5.7.4 Additional modifications

It is important to note that the GPS functionality (see Section 5.8.3), reverse geocoding (see Section 5.8.3) and photo capture capabilities (see Section 5.8.3), from the MobiSAM Report client application (discussed next), were incorporated within the updated cross-platform client application before conducting the final usability evaluation (see Chapter 7). This ensured that a similar set of features existed between the two cross-platform client applications, thus allowing for meaningful comparisons to be made between each. These modifications took place during Iteration 3 of the life-cycle model (see Section 3.4.3).

5.8 Cross-platform MobiSAM Report client application

This section provides an overview of the design and implementation of the cross-platform MobiSAM Report client application. Developed during Iteration 3 of the life-cycle model (see Section 3.4), MobiSAM Report offers an alternative to the MobiSAM client application, albeit with a single focus: the detailed reporting of *specific* service delivery problems. The requirements and specifications are first described (Section 5.8.1), followed by the design and implementation of the UI (Section 5.8.2), as well as additional miscellaneous features (Section 5.8.3).

Due to space restrictions, a detailed account of the technical implementation has been provided as an electronic resource. Please refer to electronic Appendix A.5 for more information.

5.8.1 Requirements and specifications

The original cross-platform MobiSAM client application (see Section 5.6) was developed as a mobile phone-based polling application, allowing for the ongoing monitoring of basic services, instead of reporting specific service delivery concerns. With this in mind, a stakeholder in the MobiSAM project suggested that a new version of the client application be developed, focusing on the reporting of specific service delivery problems. The stakeholder stated that emphasis should be placed on reporting, rather than the answering of polls and subsequent visualisation of results.

Requirements from the stakeholder were brief, in which they provided a simple 'mock-up' of the client application design. In addition, they expressed the need for residents to be able to upload photographs of issues being reported. This would allow them to prioritise responses to reported issues, based on knowledge of the severity of the situation depicted in the images. Furthermore, they requested the ability for the user to obtain their current position using their handset's built-in GPS capabilities (when supported). Accordingly, specifications were derived from these requirements, leading to the design of a new cross-platform client application: MobiSAM Report.

5.8.2 User interface and navigation

This section describes the UI, as well as the navigation approach taken by MobiSAM Report. The design of each application screen is presented. For information relating to the technical implementation details of the client application's UI, please see Appendix A.5.1.

Figure 5.13: MobiSAM Report running on all supported platforms (not to scale)

***Sign in* screen** After launching MobiSAM Report, the user is presented with the *Sign in* screen, shown in Figure 5.14a. The screen follows a very similar design to the *Login* screen of the cross-platform MobiSAM client application. Notable differences include use of the phrase 'Sign in' instead of 'Log in', an updated application logo, as well as the removal of the 'Username' and 'Password' instruction text, instead placing each as a hint within the text entry areas. The text 'Keep me logged in' was also shortened to 'Remember me', making it better suited to low resolution displays. As with the cross-platform client application, all screen title text throughout MobiSAM Report appears in bold, as does the text of currently selected UI components.

***Home* screen** Instead of providing a list of possible actions, the MobiSAM Report *Home* screen consists of four icons arranged in a 2x2 grid, as illustrated in Figure 5.13. These icons correspond to the four basic services which all municipalities in South Africa are mandated to provide: water, electricity, roads and sanitation. The MobiSAM Report logo appears at the top of the screen on all platforms excluding Java ME, due to the low resolution displays often used by feature phones. In addition, 'Sign out' and 'Settings' soft keys are included when running on Java ME. These actions are accessed via the hardware buttons on Android, BlackBerry and Windows Phone devices, while appearing as regular buttons on iOS.

***Reporting* screen** Selecting an icon on the *Home* screen navigates the user to the corresponding *Reporting* screen. Instead of providing a separate screen for each issue being reported (as is the case with the cross-platform client application), the *Reporting* screen contains a predefined set of common issues for a given report category (water, electricity, roads and sanitation). Each predefined issue was specified by the stakeholder and is represented by a radio button, as shown in Figure 5.14b.

Before submitting a report, the user is required to provide the address or position of the issue being reported. This can either be entered manually or obtained using the handset's built-in GPS (when supported). In addition, the user may capture and include a photo of the issue being reported by ticking the option: 'I'd like to take a photo and attach it to my report'. Once ticked, the handset's camera application is launched, allowing the user to capture a photo. Finally, a text entry area is provided where additional information may be included along with the report. Two soft keys, 'OK, send report' and 'Back', are included on the Java ME platform. These appear as regular buttons on the remaining platforms.

(a) *Sign in* screen (b) Reporting a water issue

Figure 5.14: MobiSAM Report

***Settings* screen** The *Settings* screen, shown in Figure 5.15, is analogous with the *Update profile* screen in the cross-platform MobiSAM client application. Here, the user is able to specify their preferred language and current suburb. Changes are applied automatically, without the user having to save changes.

***Report sent* screen** The *Report sent* screen is identical to that used by the cross-platform client application, shown in Figure 5.10d, and serves to inform the user that their report has been successfully sent. As the backend server does not make available a summary of received *photo uploads* reports, no form of visualisation is provided.

Figure 5.15: *Settings* screen

5.8.3 Miscellaneous

This section discusses the variety of miscellaneous features implemented within MobiSAM Report. Each of them are described next. For more information relating to the technical implementation details of each, please consult electronic Appendix A.5.2.

GPS functionality

When reporting an issue, the user is required to provide an address nearest to the issue being reported. When supported, the handset's built-in GPS may be used to obtain this positioning information. Access to this functionality is provided using the CN1 location API which allows developers to request a handset's position or track position changes.

Reverse geocoding

Handset coordinates returned by the location API are converted into a human-readable address through a process known as reverse geocoding [64]. The Google reverse geocoding service[5] is used to perform this conversion. As the service has limited resolution, suburb names are often excluded. This missing information is provided using an intersection algorithm, which returns the name of the suburb into which the user's position coordinate falls.

[5]Reverse Geocoding http://developers.google.com/maps/documentation/geocoding/#ReverseGeocoding

Photo capture

The ability to capture and attach photos to reports allows the municipality to effectively prioritise reported issues. This feature is supported by handsets which contain a camera module. In an attempt to reduce processing time and bandwidth consumption, images are captured as a width of 600 pixels, while image height is dependant on the aspect ratio of the handset's imaging sensor as well as the orientation of the device. Only a single photo my be attached per report.

Connectivity

The ability to upload a report, which includes a captured image as well as position coordinates and descriptive text is achieved using the *upload images* poll type (see Section 5.2). Several new polls were created on the backend server, each representing an available report listed on the *Reporting* screen.

5.9 Summary

This chapter presented the design and implementation of two cross-platform MobiSAM client applications. Background information relating to the MobiSAM service was first provided. Next, the MobiSAM system's architecture was described. The design and implementation of the cross-platform client applications, MobiSAM and MobiSAM Report, were then discussed. Updates made to the cross-platform MobiSAM client application were also detailed. The next chapter presents the first of two usability evaluations performed using the cross-platform MobiSAM client application.

Chapter 6

Evaluation: navigation and visualisation

6.1 Introduction

This chapter presents the usability evaluation performed using the cross-platform MobiSAM client application. The evaluation aimed to determine how intuitively participants were able to navigate within the client application. It also aimed to determine how effective different visualisation techniques were at conveying meaning to participants, as well as the role which interactivity played in helping participants to better understand visualised poll results. In addition, the evaluation intended to identify any usability concerns and functional problems to be addressed during subsequent development iterations.

The chapter begins by describing the study sample (Section 6.2), data-collection instruments (Section 6.3) as well as method and analysis (Section 6.4). It then details the study tasks (Section 6.5) and corresponding results (Section 6.7). Finally, it provides an in-depth analysis and discussion of the results (Section 6.8), highlighting interesting trends and important findings.

6.2 Study sample

A stratified sampling technique was employed in this evaluation, with opportunity sampling within each strata (age and gender demographics). This allowed the researcher to select participants who were most easily accessible. Instead of selecting a sample strictly representative of the Grahamstown population at large, it was decided to select participants from a broad cross-section of different age groups. This way, the researcher was able to determine how participants performed across a variety of different ages. Thirty ($n = 30$) participants took part in the usability evaluation. Due to the potentially lengthy process of having to ask a parent/guardian to sign a consent form for minors, participants were limited to citizens 18 years and older. Across all results, responses were classified into the following race groups: White, Indian/Asian and Black African/Coloured.

6.3 Data-collection instruments

In an attempt to capture as much useful data as possible, three separate questionnaires were developed which focused on gathering both quantitative and qualitative feedback from participants. The first, a pre-intervention questionnaire, comprised of two sections: demographic information and mobile phone usage habits. The first section consisted of four questions which solicited information such as race, gender, preferred language and highest level of education. The second section consisted of 15 questions soliciting feedback on mobile phone ownership and current use thereof. These included questions relating to the brand of mobile phone, network provider, amount spent on airtime per week, mobile services accessed, and the use of third-party mobile applications.

The second questionnaire contained the four study tasks to be performed by participants and used to evaluate the MobiSAM client application. Each task was designed to be representative of the client application's real-world use and when combined, spanned a broad spectrum of the client application's functionality. In the process of piloting the questionnaire, it became apparent that it was unrealistic to expect each participant to perform all four tasks in one sitting. Participants were instead randomly assigned a pair (2) of tasks. To ensure that every task combination was evaluated, tasks were grouped as follows: Task 1 & 2; Task 1 & 3; Task 1 & 4; Task 2 & 3; Task 2 & 4; and finally Task 3 & 4. Consequently, each task combination was performed by a total of five different participants ($n = 30$). It is important to note that while all task combinations were performed, not all permutations thereof where undertaken during the study.

A small number of statements and questions were posed directly after each task, quizzing participants on the task which they had just attempted. These aimed to quantify the level of task satisfaction. A five-point Likert scale was used to record responses to the statements, while a simple 'yes', 'no' or 'maybe' was recorded for each question. Both positively and negatively worded Likert statements were used in an attempt to mitigate the constant error caused by the acquiescence effect [122]. Additional space was provided below each statement or question allowing participants to elaborate upon their choice, if they so desired.

The third and final questionnaire consisted of two sections: application navigation and application visualisation. The first section comprised of four questions which aimed to gather information relating to the client application's ease of use and navigation, as well as how intuitive participants found the *Home* screen. The second section comprised of five questions soliciting feedback surrounding the effectiveness of the client application's visualisations, as well as determining which visualisation technique was found most useful by participants. An open-ended question was also included, requesting that participants bring attention to any additional suggestions or concerns relating to the client application.

In addition to the previously mentioned paper-based questionnaires, the client application was modified in order to log a variety of different task performance measures, including the number of log in attempts; task duration; and number of button press events made while performing a task. In order to prevent the variable latency of network operations potentially effecting timing results, all tasks were simulated. Details of the client application modifications are presented in Appendix A.3.5. General written observations were also made for each participant, such as the reason for a participant not completing a task, or when values were incorrectly read from a visualisation.

A copy of the study questionnaires are provided in electronic Appendix B.2.

6.4 Method and analysis

In accordance with University protocol, ethics approval was obtained (tracking number CS13-12) before the study proceeded. The researcher and research assistant then began recruiting participants required to constitute the study sample described in Section 6.2. A field setting was employed during the study in an attempt to mirror the variable physical environment in which the client application is likely to be used. In addition, this approach eased the recruiting process as citizens were able to be evaluated anywhere, any time, without prior arrangement. Each usability evaluation took between 20 and 40 minutes to complete, depending on the tasks given as well as the pace of the participant.

Participants were approached and enlisted in May 2014 from numerous public areas within Grahamstown: communal areas in the informal settlement; Rhodes University campus; and by referral from existing participants. Several of the research assistant's acquaintances also took part in the study. As each evaluation took significantly more time when compared to the baseline study questionnaire (see Chapter 4), it was not possible to enlist pedestrians as they were often going about their daily duties and unable to offer their time. Participants were not given any form of remuneration or incentive for their involvement.

Before agreeing to participate, citizens were informed about the purpose of the study and told what the study would entail. Once a citizen agreed to participate, a consent form was signed. The citizen was then presented with a pre-intervention questionnaire to complete before proceeding with the study tasks. Depending on the preference of each participant, questionnaires were either completed by themselves or verbally, with the researcher transcribing their responses. While the questionnaires and tasks were only available in English, the research assistant was able to translate between English and isiXhosa when required. All responses were kept strictly confidential with participants being assigned a number at random, making it impracticable to trace a specific questionnaire back to an individual.

After completing the pre-intervention questionnaire, the participant was given a choice of two possible handsets on which to perform the study tasks: a Nokia Asha 201 feature phone (Series 40 5th Edition) or a Samsung Galaxy Pocket smartphone (Android OS 2.3). Although the device display resolutions are identical, the Nokia handset is equipped with a QWERTY keypad while the Samsung makes use of a touchscreen display. In an attempt to determine whether participants preferred static or interactive chart visualisations, a different visualisation technique was used by each handset (see Section 5.6.3).

Having selected a handset, the researcher launched the client application, logged in and informed the participant that they had approximately two minutes to explore the application before starting the study. This exploration task aimed to increase the participant's familiarity with the client application, providing them with enough confidence to perform the study. Assistance was provided when needed. Once the time had expired, the participant was presented with the study task questionnaire, containing a pair of tasks (see Section 6.3 for possible task combinations). Each task was performed sequentially, therefore Task 1 was always performed first, irrespective of the task combination, whereas Task 4 was always performed last. This helped the researcher assess the learnability of the client application. After performing each task, the participant was then presented with several statements relating to the task (for example, "Logging in to the MobiSAM app was easy"). Statements were administered to participants immediately after each task in order to improve the accuracy of participant responses [97]. Responses to the statements took the form of a five-point Likert scale. Although no time limit was imposed on tasks, minimal guidance was offered to struggling participants when required. This assistance was recorded in the observational notes.

After performing both tasks and responding to the necessary post-task statements and questions, the participant was finally presented with the post-intervention questionnaire. This questionnaire aimed to gather detailed participant feedback relating to the client application's navigation and visualisation capabilities. Once the target number of participants had been obtained ($n = 30$), the raw results from the evaluation were captured electronically using Google Forms and analysed using Microsoft Excel. A detailed description of the analysis techniques is presented in Section 6.8.

6.5 Study tasks

Four tasks were constructed as part of the usability evaluation, each designed to be representative of the client application's real-world use. Tasks attempted to evaluate the main features of the client application: logging in; answering various poll types; interpreting returned poll results; and updating one's user profile. As the initial focus of the MobiSAM and Makana Municipality collaboration was the monitoring of potable water (see Section 1.5), all study tasks related to this basic service. Additional categories (roads, sanitation and electricity) were included at a later stage in response to formal requests made by Makana Municipality. Each task is described next.

6.5.1 Task 1: No water

Task 1 aimed to determine how easily participants were able to answer a *custom* (yes or no) poll type (see Section 5.2) as well as comprehend the returned poll results, using the chart visualisations. In this task, participants were instructed to use the client application to inform their municipality that they were without running water. The task consisted of four distinct steps:

1. First, participants were required to log in to the client application using the provided username and password combination.
2. Once logged in, participants were asked to follow the necessary steps to inform Makana Municipality that there was currently no water at their home or place of residence.
3. After having successfully submitted their poll response, participants were asked to use the chart visualisations to determine how many other residents (both the number and percentage) within their suburb were also without water.
4. Once participants had written down these figures, they were instructed to navigate back to the *Home* screen.

As poll results were presented to participants in three different forms (numerical table, pie chart and bar chart), participants were instructed to use the latter two when determining their response to step 3.

6.5.2 Task 2: Leaking pipe

Task 2 aimed to determine how easily participants were able to answer a *text response* poll (see Section 5.2) and instructed participants to use the client application to inform their municipality about a leaking pipe. The task comprised of three distinct steps:

1. First, participants were asked to log in to the client application.
2. Once logged in, participants were instructed to follow the necessary steps to inform Makana Municipality that there was a leaking water pipe at 5 High Street.
3. After having successfully submitted their response, participants were asked to return to the *Home* screen.

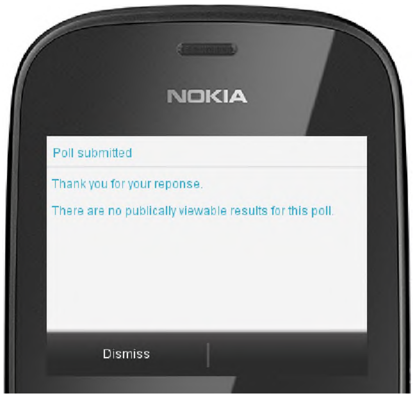


Figure 6.1: Generic *Feedback* screen for a *text response* poll type

As *text response* poll types do not return results (due to their elaborate, open-ended nature), a visual representation of the results was not presented to participants. Instead, they were shown a *Feedback* screen informing them that their response was successfully submitted, as illustrated in Figure 6.1.

6.5.3 Task 3: Dirty water

Task 3 aimed to determine how easily participants were able to answer a *custom* poll (see Section 5.2) and interpret results presented in the form of pie and bar chart visualisations. Participants were instructed to use the client application to report dirty water to their municipality. The task consisted of four distinct steps:

1. Participants were first asked to log in to the client application.
2. After having logged in, participants were asked to follow the necessary steps to inform Makana Municipality that their water was brown in colour.
3. Once their poll response was submitted, participants were instructed to make use of the chart visualisations to determine both the number and percentage of other residents within their suburb that were also experiencing brown water.

4. After having written down these figures, participants were finally instructed to navigate back to the *Home* screen.

As with Task 1, participants were instructed to make use of the pie and bar chart visualisations when determining the answer to step 3.

6.5.4 Task 4: Change application language

Task 4 was shorter than other tasks and aimed to determine how easily participants were able to modify their user profile. The task was split into three steps:

1. First, participants were asked to log in to the client application.
2. Having successfully logged in, participants were instructed to follow the steps required to change the client application's language from English to Afrikaans.
3. Finally, participants were asked to navigate back to the newly translated *Home* screen, now displayed in Afrikaans.

User profile changes needed to be saved before taking effect.

6.6 Task performance measures

Several parameters were recorded in an attempt to determine how well each participant performed during the evaluation. Measurements included task completion rate, task duration as well as the number of button press events made. In addition, a usability rating was calculated for each task, based on feedback from the post-task questionnaire.

6.6.1 Completion rate

The first and most basic usability metric employed was calculating what proportion of tasks attempted by participants were completed as expected. Tasks which were completed, although in an unexpected manner, were not considered. Instead, they were grouped together with those who did not complete the task. Please note that the phrases 'completed the task' and 'completed the task as expected' are used synonymously from here onwards.

6.6.2 Duration

Task times are a commonly reported usability metric [122]. Although not considered informative on their own, when combined with other measurements (such as task completion rates or button press events), they have the potential to reveal interesting usability trends.

As task times are often positively skewed (due to a small number of participants taking much longer to complete a given task than others), both the arithmetic mean as well as the median have been shown to be a poor measure of central tendency [188]. In their paper, *Average Task Times in Usability Tests: What to Report?*, Lewis and Sauro showed the geometric mean to have less error and bias when compared to the sample median [188]. As such, this technique was used when finding the mean participant duration for a given task. From this point forward, the geometric mean will simply be referred to as the mean.

6.6.3 Button press events

The number of button presses made by participants when performing each task was logged. Only presses which resulted in the navigation between different screens were recorded. For example, pressing the 'Update profile' menu option was counted, whereas changing the preferred language from the drop-down menu was not counted. In addition, the button presses of tasks which were not completed as expected were not considered.

In order to serve as a baseline, the minimum number of button presses required to perform each task are summarised in Table 6.1. In addition, the minimum number of button presses needed to find the information as instructed by the task questionnaire, using the chart visualisations, are presented. An expert user attempting Task 1, for example, was able to complete the task making a total of five button presses, irrespective of whether the feature phone or smartphone was used.

		Task		***Results* screen**	
	n	**Feature phone**	**Smartphone**	**Feature phone**	**Smartphone**
Task 1	15	5 presses		3 presses	4 presses
Task 2	15	5 presses		N/A	N/A
Task 3	15	5 presses		3 presses	4 presses
Task 4	15	3 presses		N/A	N/A
Median		**5 presses**		**3 presses**	**4 presses**

Table 6.1: Minimum number of button presses required per task

As the number of button presses per task differed, depending on which handset was used during the evaluation, two sets of figures are given. In an attempt to reduce the effects of outlier values, the median was used when calculating the average number of button presses made by participants.

6.6.4 Usability ratings

As previously described (see Section 6.4), several statements were presented to participants after having performed each task (for example, “Logging in to the MobiSAM app was easy”), with responses taking the form of a five-point Likert scale. Written comments elaborating reasons for a particular choice were also encouraged.

In an attempt to quantify how favourably participants answered each statement, a value from one to five was assigned to each Likert option. Therefore, if a participant indicated that they ‘Strongly agree’ to the previous statement, a value of five was scored. Conversely, selecting ‘Strongly disagree’ would result in a value of one. Assigned values were reversed in the case of negatively phrased statements.

A participant’s usability rating for each task was then calculated by adding the values of the responses to each statement presented to them and dividing the results by the total. For example, assuming a participant answered ‘Strongly agree’ to Q1; ‘Disagree’ to Q2; ‘Agree’ to Q3; and ‘Neutral’ to Q4, their overall usability rating for the task would be (assuming all statements are positively phrased):

$$5 + 2 + 4 + 3 = 14$$

Expressed as a percentage, the usability rating for the example task would be:

$$\frac{14}{20} = 0.7 * 100\% = 70\%$$

In addition, the usability rating for each statement was calculated by adding the individual Likert scores of all participant responses to the given statement. For example, assuming 10 participants responded ‘Strongly agree’ to the previous statement while five responded ‘Agree’, the resulting usability rating would be (assuming the statement was positively phrased):

$$(10 * 5) + (5 * 4) = 70$$

Converted to a percentage, the usability rating for the statement would be:

$$\frac{70}{5 * 15} = \frac{70}{75} = 0.9333 * 100\% = 93.33\%$$

Finally, in order to determine the mean usability rating for an entire task, the usability rating of each statement was added and then divided by the grand total. For example, assuming three statements were presented to 15 participants after having performed the task (therefore each statement would be out of a total of 75) and scored 70, 57 and 68 respectively. The mean usability rating for the entire task would be:

$$\frac{70 + 57 + 68}{(3 * 75)} = \frac{195}{225} = 0.8667 * 100\% = 86.67\%$$

A combination of different usability scores were used when comparing different participant subgroups as well as identifying usability weaknesses within the client application.

6.7 Results

This section presents the results of the pre-intervention and post-intervention questionnaires, as well as the four previously described study tasks. As discussed in Section 6.4, the evaluation was designed so that participants performed two out of four possible tasks, resulting in a total of 15 responses per task. Tasks were paired up so as to ensure that all six task combinations were evaluated five times, giving a total of 30 evaluations. As the study was performed using two different handsets, evaluations were split evenly between each model, resulting in 15 evaluations being performed using the feature phone and 15 evaluations being performed using the smartphone. Several different parameters were used as task performance metrics, as described in Section 6.6. Questionnaires presented to participants as part of the study are listed in Appendix B.2.

6.7.1 Pre-intervention questionnaire

The pre-intervention questionnaire was comprised of two sections (demographics and use of technology), the results of which are presented next.

Demographics

The first section of the pre-intervention questionnaire consisted of four questions and aimed to elicit basic demographic information such as age, gender, preferred language and highest level of education. The race group of each participant was also noted. Of the 30 participants, 43.33% were female while 56.67% were male. As reflected in Table 6.2, the majority of participants were between 18-30 years, while 16.67% were in the 31-45 years and over 60 years age split respectively.

60% of participants described themselves as Black African/Coloured; 36.67% as White; while one participant (representing 3.33% of the study sample) indicated that they were Indian/Asian.

Age group	n	**Percent**
18-30 years	17	56.67%
31-45 years	5	16.67%
46-60 years	3	10.00%
Over 60 years	5	16.67%

Table 6.2: Age distribution of participants

Participants were also asked to indicate their preferred language of which 80% specified English while the remaining 20% specified isiXhosa. Lastly, participants were asked about their highest level of education. Referring to Table 6.3, responses indicated that more than half the participants held a university degree.

Highest level of education	n	**Percent**
Some high school	1	3.33%
Matric	11	36.67%
Technikon/college diploma	1	3.33%
University degree	17	56.67%

Table 6.3: Education level of participants

Use of mobile technology

The second section of the pre-intervention questionnaire, consisting of fifteen questions, solicited feedback relating to participants' use of mobile technology and related services. Questions aimed to determine the level of prior experience participants had using mobile technologies and services, as well as their familiarity installing applications. The majority of questions in this section were adapted from the identically named section in the baseline study, presented in Chapter 4.

Of the 30 participants who took part in the evaluation, all 30 owned a mobile phone. When asked how long it had been since they obtained their first mobile phone, 96.67% of participants indicated six years or more (see Table 6.4).

Timespan	n	**Percent**
1-5 years	1	3.33%
6-10 years	16	53.33%
Over 10 years	13	43.33%

Table 6.4: Number of years since participants obtained their first mobile phone

Nokia and Samsung devices were owned by most participants, representing 53.3% and 26.7% of the participant handsets respectively. BlackBerry, HTC, Huawei and Vodafone-branded handsets were owned by a single participant each (3.33%), depicted as 'Other' in Figure 6.2.

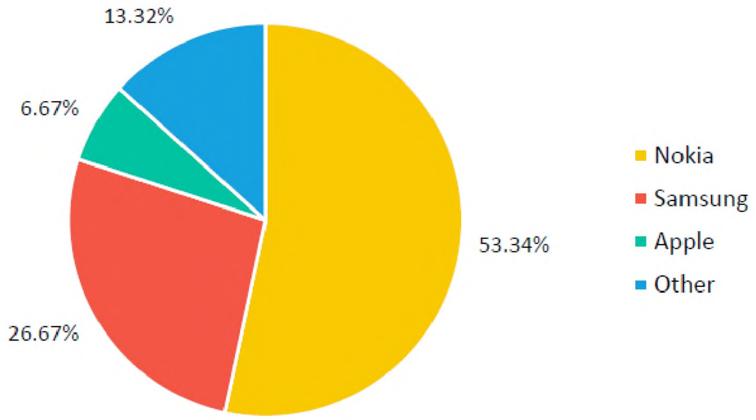


Figure 6.2: Most popular handset brands ($n = 30$)

Participants were asked to rank the frequency in which they use the data, voice and SMS services available on their handset. As illustrated in Figure 6.3, 63.33% of participants ranked data as most used; 60% ranked voice as moderately used; while SMS was ranked as least used by 53.33% of participants. Five participants indicated that they never used data services from their handset.

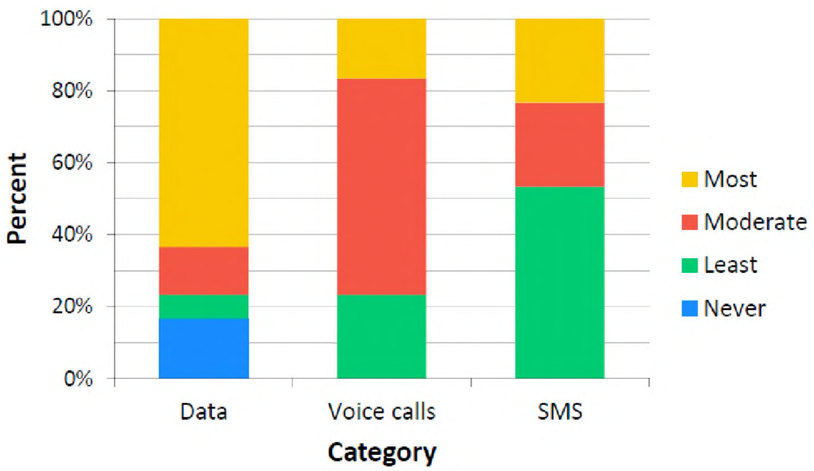


Figure 6.3: Frequency of use: data versus voice versus SMS ($n = 30$)

When asked to specify what messaging services participants used on their handsets, 96.67% indicated SMS; 76.67% indicated WhatsApp; 20% indicated Mxit; and 10% specified that they used BlackBerry Messenger. One participant (3.33%) said they used Facebook Messenger while another (3.33%) stated that they used Viber. Responses are reflected in Figure 6.4. Relating to their frequency of use, 90% of participants indicated that they use the specified messaging services on a daily basis; 6.67% use them a few times per week; while one participate (3.33%) stated that they make use of them a few times per month.

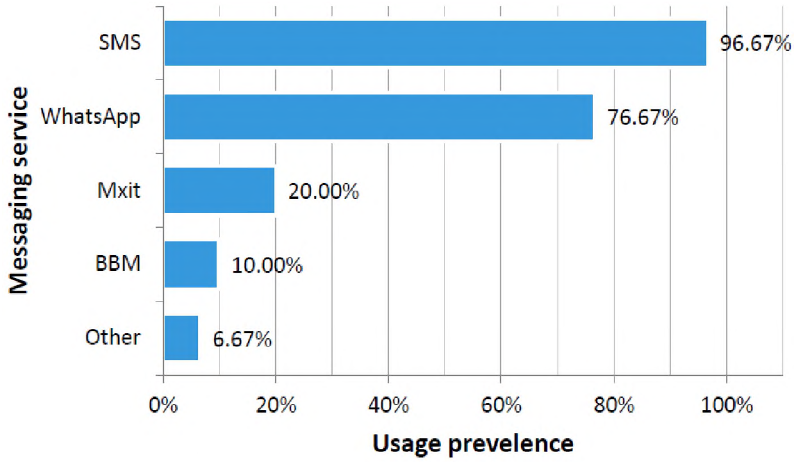


Figure 6.4: Use of messaging services ($n = 30$)

When asked to indicate what online data services participants accessed from their mobile phones, 70% specified e-mail; 66.67% specified Facebook; 33.33% said they read the news; 16.67% said they consulted Wikipedia; and 13.33% said they used Twitter. Responses are illustrated in Figure 6.5. In response to the question asking how often participants accessed the specified online data services, 66.67% indicated that they accessed them on a daily basis; 6.67% said they accessed them a few times per week; while the same number said they accessed them a few times each month.

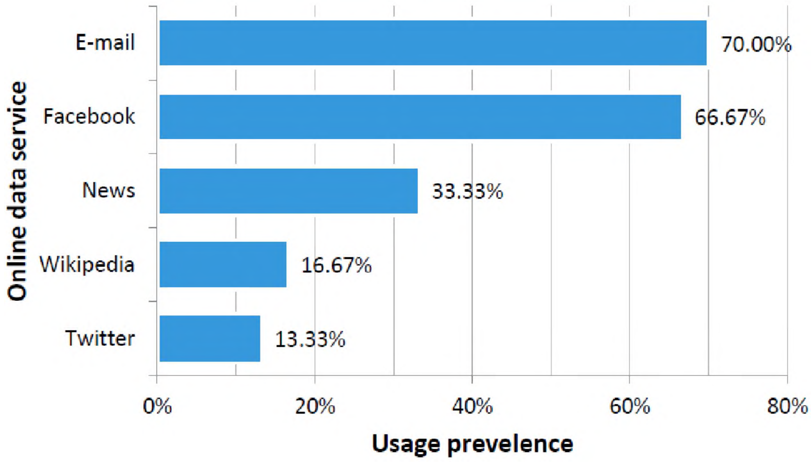


Figure 6.5: Use of online data services ($n = 25$)

20% of participants indicated that they do not make use of any online data services from their mobile device.

6.7.2 Task completion rates

Figure 6.6 illustrates what proportion of the tasks performed by participants were completed as expected, irrespective of the handset used.

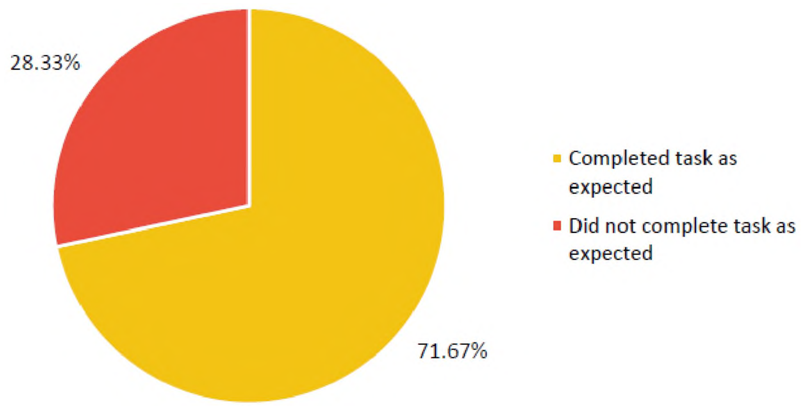


Figure 6.6: Proportion of tasks completed as expected ($n = 60$)

A breakdown of the results for each individual task is provided in Figure 6.7.

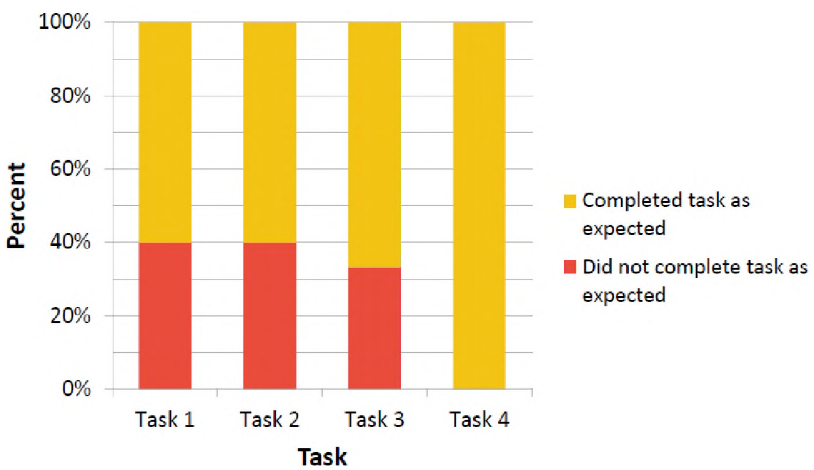


Figure 6.7: Completion rates of individual tasks ($n = 60$)

Examining these results further, Figure 6.8 illustrates that participants who used the feature phone ($n = 15$) completed a lower proportion of tasks compared to those who used the smartphone ($n = 15$). This is reflected in the mean usability rating, with tasks performed on the feature phone receiving a mean usability rating of 83.11%, while tasks performed on the smartphone received a mean usability rating of 86.72%.

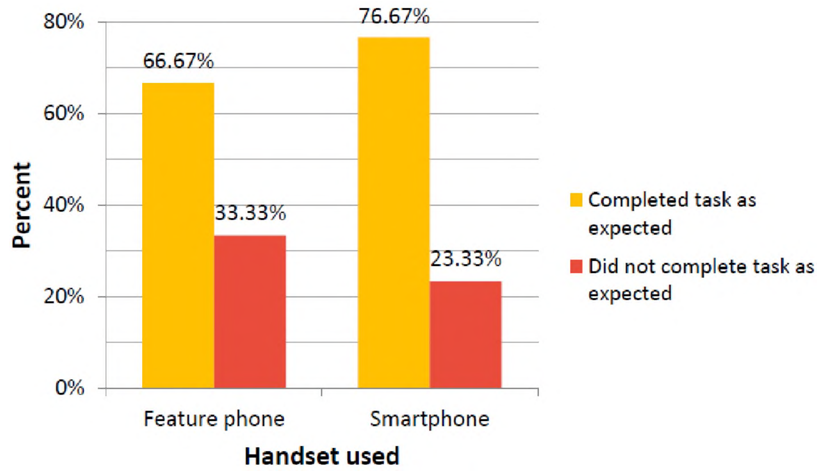


Figure 6.8: Comparison of task completion rates: feature phone versus smartphone ($n = 60$)

The results for each of the tasks described in Section 6.5 are now presented.

6.7.3 Task 1: No water

This section presents the results of Task 1: reporting a water outage. A summary of participant responses to the post-task questionnaire are presented in Figure 6.9.

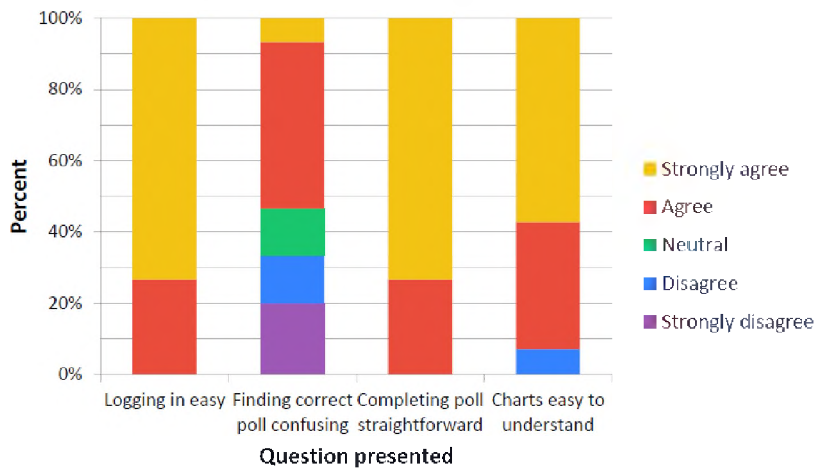


Figure 6.9: Summary of Task 1 responses ($n = 59$)

Of the 15 participants who attempted Task 1, over half (53.33%) completed the task on their own; one participant (6.67%) required some guidance; while 40% were unable to complete the task. Examining the results of log in procedure, 73.33% of participants managed to log in to the client application on their first attempt. One participant (6.67%) managed on their second attempt, while the remaining 20% took three attempts.

	Task 1		*Results* screen	
	Feature phone	**Smartphone**	**Feature phone**	**Smartphone**
Participant mean duration	120.59s	88.62s	155.16s	46.94s
Participant median	7 presses		3 presses	6 presses
Expert user	5 presses		3 presses	4 presses

Table 6.5: Average duration and number of button presses made when performing Task 1

The mean time taken by participants who completed the task was 120.59s for those who used the feature phone, and 88.62s for those who used the smartphone. The average number of button presses made while completing the task are presented in Table 6.5. It is interesting to note that even though participants, on average, made the same number of button presses regardless of the handset used during the study, participants using the smartphone took less time to complete the task. This notable difference (120.59s versus 88.62s) may have been as a result of the smartphone's touchscreen interface allowing participants to navigate more quickly when compared to the feature phone's QWERTY keypad, due to the more intuitive interaction method it provides.

Q1: "Logging in to the MobiSAM app was easy" Of the 15 participants who responded to the first statement, 73.33% answered 'Strongly agree'; while 26.67% answered 'Agree' resulting in a usability rating of 94.67% for the statement. No written comments were provided.

Q2: "Finding the correct poll was confusing" Responses to the second statement were generally unfavourable, with one participant (6.67%) answering 'Strongly agree'; 46.67% answering 'Agree'; 13.33% answering 'Neutral' as well as 'Disagree'; while 20% answered 'Strongly disagree'. This equated to a usability rating of 58.67% for the statement. Written comments from six participants (1, 4, 11, 23, 27, 30) cited confusion surrounding use of the word "poll". Many indicated that they were not sure whether they were required to 'Take a poll' as the questionnaire simply asked them to inform Makana Municipality about a water outage. In addition, Participant_9 indicated that it was possible to answer Task 1 via two different methods, both of which were valid, although one of which unexpected.

Q3: "Completing the correct poll was straightforward" The third statement aimed to assess how easily participants ($n = 15$) were able to answer the poll after having navigated to it. Responses were identical to that of question 1 with 73.33% of participants answering 'Strongly agree' while 26.67% answered 'Agree', resulting in a usability rating of 94.67%. Participants provided no written responses.

Q4: "The charts were easy to understand" Responses to the fourth statement were overall positive. More than half (57.14%) of participants responded with 'Strongly agree'; 35.71% with 'Agree'; while one participant (representing 7.14% of the sample) answered 'Disagree'. The resulting usability rating for the statement was 88.57%. As Participant_30 did not answer this statement, the sample size was $n = 14$. No written responses were given.

Q5: "I used the interactive features of the charts" The fifth and final statement was directed only to participants who performed the task using the smartphone handset ($n = 8$). Of these participants, 75% answered 'No' while 25% indicated that they did in fact use interactivity within the chart. Participant_4 commented that they did not realise they could interact with the chart to reveal more information, stating: "I didn't realise I could 'drill down' for details, my personal preference is for numbers/tables rather than charts".

Discussion The mean usability rating for Task 1 was 84.07%. Observations made by the researcher highlighted confusion surrounding the word "poll", with many participants being disorientated by the fact that they were required to select 'Take a poll' when asked to inform Makana Municipality about a water outage. Participant_27, for example, wrote: "'Take a poll' can seem odd when you're looking to report a problem" while Participant_23 said that 'Take a poll' was unclear and should be rephrased to 'Problems or comments?'. Three participants (7, 11, 23) had trouble understanding the term 'votes' shown on the *Results* screen, with alternative suggestions stating that 'people' or 'respondents' would be more appropriate. The six participants (1, 5, 6, 17, 29, 30) who did not complete the task, consequently, were not presented with the *Results* screen and were therefore unable to provide feedback regarding the chart visualisations.

6.7.4 Task 2: Leaking pipe

Of the 15 participants who attempted Task 2, 60% completed the task on their own, while the remaining 40% were unable to complete the task. The vast majority (80%) of participants were able to successfully log in to the client application on their first attempt, while 20% were successful on their second.

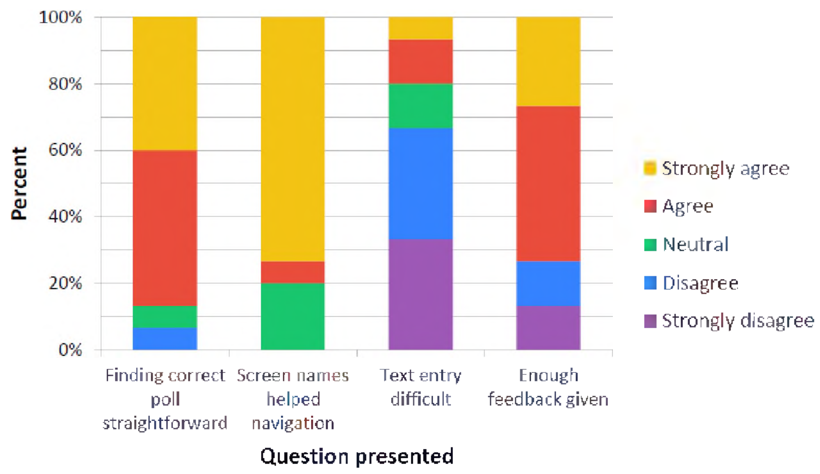


Figure 6.10: Summary of Task 2 responses ($n = 60$)

Examining the task durations for participants who managed to complete Task 2. A mean duration of 67.26s was recorded for those who used the feature phone and 74.65s for those who used the smartphone. An average of five button presses were made by participants when performing the task, regardless of the handset used. Results are depicted in Table 6.6.

	Feature phone	**Smartphone**
Participant mean duration	67.26s	74.65s
Participant median	5 presses	
Expert user	5 presses	

Table 6.6: Average duration and number of button presses made when performing Task 2

A summary of participant responses to the post-task questionnaire for Task 2 are illustrated in Figure 6.10.

Q1: "Finding the correct poll was straightforward" The first statement was answered by all 15 participants, of which 40% responded with 'Strongly agree'; 46.67% responded 'Agree'; one participant (6.67%) responded 'Neutral'; while another responded 'Disagree'. The resulting usability rating for the statement was 84%. Numerous written responses were given, of which four (Participant_15, Participant_20, Participant_21, Participant_22) described confusion surrounding the concept of taking a poll. Participant_20 wrote: "[...] I wouldn't call it a poll, rather problem reporting" while Participant_15 stated: "At first I wasn't sure what I had to do or where to go. I wasn't sure of the options on the homepage, even though the 'Take a poll' option was quite clear". Furthermore, Participant_22 highlighted that: "'Take a poll' wasn't intuitive to know this is where I needed to navigate to report an issue". However, Participant_23 indicated that it was "better the second time" due to familiarity. Thus potentially highlighting the learnability of the client application.

Q2: "Having the screen names at the top of the MobiSAM app made navigation easier" Responses to the second statement were generally positive with a usability rating of 90.67%. Of the 15 participants, 73.33% answered 'Strongly agree'; one participant (6.67%) answered 'Agree'; while the remaining 20% answered 'Neutral'. Three written responses from participants (11, 20, 23) indicated that they had not noticed the 'breadcrumb' trail at the top of the screen. Conversely, Participant_22 indicated that: "Breadcrumb trails are helpful for navigational understanding". Participant_15 shared similar feelings, stating: "It confirms where I am in the app and what task I'm carrying out".

Q3: "Entering text into the MobiSAM app was difficult" The third statement received mixed responses with one participant (6.67%) answering 'Strongly agree'; two participants (13.33%) answering 'Agree' and 'Neutral' respectively; and 33.33% answering 'Disagree' and 'Strongly disagree' respectively. The resulting usability rating for the statement was 74.67% ($n = 15$). Written responses from participants highlighted that difficulties arose partly as a result of them being unfamiliar with the handsets. Participant_24, for example, stated that: "The phone made it take longer to enter information because of wrong buttons pressed". Participant_22 indicated that they wished the UI was more instructive, writing: "Just specify what details are required to enter when reporting a problem". The participant then went on to suggest: "Rather ask the user what problem [they are experiencing] instead of stating 'I want to report..."'. On the other hand, Participant_15 stated that they found it "very straight forward".

Q4: "Enough feedback was given after submitting my report" Finally, the fourth statement also received varied responses from participants. Of the 15 participants, 26.67% responded that they 'Strongly agree'; 46.67% responded 'Agree'; while two participants (13.33%) responded with 'Disagree' and 'Strongly disagree' respectively. This was reflected in the usability rating of 72%. Examining written responses, four participants (3, 8, 11, 23) indicated that they would like more feedback. Specifically, Participant_11 wrote: "I would like to know if my report has been acted upon" while Participant_23 wanted to know if anyone else had already reported their particular problem. While the former suggestion would not be known at the time of submitting a report, the latter suggestion would be an important mechanism to put in place in order to prevent duplicate reports.

Discussion As with the previous task, the usability rating for each participant was calculated. The mean usability rating across all participants for Task 2 was 80.33%. Researcher observations again highlighted confusion surrounding the term "poll" with Participant_20 stating that they would "only use the [MobiSAM client] app when reporting a problem" and they "didn't want to take polls". Furthermore, Participant_2 indicated that they did not know what 'Take a poll' meant until they had selected it. Similarly, Participant_22 suggested that 'Take a poll' would not be clear to new users and that rewording the option to 'What service delivery problem would you like to report?' would be more meaningful. Four participants (3, 8, 11, 23) said that insufficient feedback was provided after answering a *text response* poll.

6.7.5 Task 3: Dirty water

As with previous tasks, participants were first required to log in to the client application. Of the 15 participants who performed Task 3, 86.67% managed to log in on their first attempt; one participant (6.67%) was successful on their second attempt; while another participant (6.67%) took three attempts. Two-thirds (66.67%) of participants completed Task 3, while the remaining third (33.33%) did not.

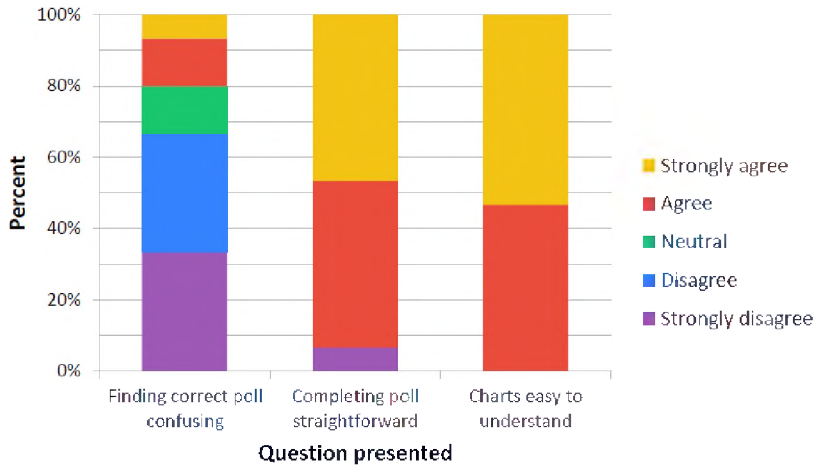


Figure 6.11: Summary of Task 3 responses ($n = 45$)

Participants who completed the task using the feature phone took a mean duration of 64.62s, while those who used the smartphone took slightly less, as reflected in Table 6.7. While the median number of button presses made by participants when viewing the *Results* screen was less than those made by an expert user, this is likely due to the fact that two of the three participants (18, 25) using the feature phone did not consult the chart visualisations. Instead, they read values from the numerical table appearing directly on the *Results* screen, hence only requiring a single button press. In addition, a single participant (Participant_1) who used the smartphone consulted the numerical table instead of the chart visualisations. Excluding this participant when calculating the median brings the number to 4 button presses, which corresponds to the minimum required. Figure 6.11 presents a summary of all the task responses.

	Task 3		*Results* screen	
	Feature phone	**Smartphone**	**Feature phone**	**Smartphone**
Participant mean duration	64.62s	48.57s	14.57s	38.81s
Participant median	5 presses	7 presses	1 press	3.5 presses
Expert user	5 presses		3 presses	4 presses

Table 6.7: Average duration and number of button presses made when performing Task 3

Q1: "Finding the correct poll was confusing" Again, varied responses were received from participants ($n = 15$). One participant, or 6.67% responded that they 'Strongly agree'; two participants (13.33%) responded 'Agree' and 'Neutral' respectively; 33.33% responded 'Disagree'; while the remaining 33.33% chose 'Strongly disagree'. The resulting usability rating for the statement was 74.67%. There were a few noteworthy written responses, with Participant_14 stating that they "think [it should] specify that I'm reporting water quality". Conversely, Participant_12 wrote: "[there were] few options to select through so [it] wasn't hard finding the poll".

Q2: "Completing the correct poll was straightforward" The second statement was answered positively by all but one participant, resulting in a usability rating of 90.67% ($n = 15$). Just under half (46.67%) of the participants responded with 'Strongly agree', while the same proportion (46.67%) responded with 'Agree'. One participant (6.67%) did not share these sentiments, instead responding with 'Strongly disagree'. There were no written responses.

Q3: "The charts were easy to understand" Responses to statement 3 were positive, with 53.33% of participants stating that they 'Strongly agree'. The remaining 46.67% responded with 'Agree' resulting in a usability rating of 90.67% ($n = 15$). Written comments from two participants (22, 24) who had used the smartphone handset (with interactive chart visualisations) when performing the study suggested including the actual value/percentage on each chart visualisation, instead of only making them viewable when 'tapping' the chart. Participant_22, for example, wrote: "Show the percentages and number of votes on the pie/bar chart" while Participant_24 stated: "Number[ed] figures in the pie charts would have been better".

Q4: "Which chart did you find easiest to understand?" The final question aimed to assess which chart type participants best understood. Of the 15 participants who answered the question, 46.67% stated that they found the pie chart easiest to understand; 20% stated that they preferred the bar chart; while 33.33% indicated that they were both equally understandable. Two noteworthy written comments were provided, the first by Participant_25 indicated that displaying the percentage value on the bar chart's y-axis would be preferred to the actual number of votes. The second comment, by Participant_10 posed a similar suggestion, stating: "the bar chart is a bit confusing because a user would expect the numbers on the y-axis to be percentages". Figure 6.12 depicts the misinterpreted bar chart visualisation. Participant_9 stated that they preferred the bar chart, and that "pie charts confuse some people". Conversely, Participant_12 maintained that the pie chart was "visually more appealing".

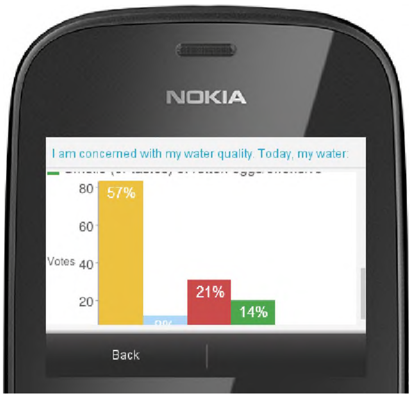


Figure 6.12: Misinterpreted bar chart visualisation

Discussion The usability rating for Task 3 was calculated for each participant, resulting in a mean usability rating of 85.33%. Observational notes made by the researcher highlighted that two participants (2, 10) did not notice the interactivity within the chart visualisations. When asked to determine how many other people were also experiencing brown water, Participant_25 was seen reading values from the numerical table instead of from the charts. Participant_14 had to perform the task again as they mistakenly dismissed the *Results* screen without noting the required values.

6.7.6 Task 4: Change application language

The fourth task began by asking users to log in to the MobiSAM client application. A vast majority (80%) of the 15 participants were able to successfully log in during their first attempt; two participants (13.33%) managed on their second attempt; while one participant (6.67%) took three attempts. All participants were able to complete the task, although two needed some guidance.

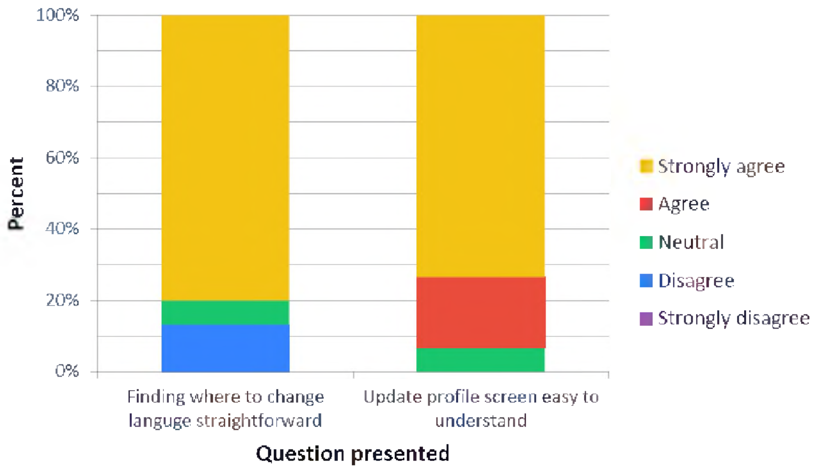


Figure 6.13: Summary of Task 4 responses ($n = 30$)

As shown in Table 6.8, Task 4 took participants on average 59.15s to complete when using the feature phone, and 26.26s using the smartphone. A median of three button presses were made by participants, irrespective of the handset used during the evaluation.

	Feature phone	**Smartphone**
Participant mean duration	59.15s	26.26s
Participant median	3 presses	
Expert user	3 presses	

Table 6.8: Average duration and number of button presses made by participants performing Task 4

Q1: "Finding where to change the MobiSAM app's language was straightforward" The first statement presented to participants ($n = 15$) scored highly with 80% of participants stating that they 'Strongly agree'; one participant (6.67%) stating 'Neutral'; while two participants (13.33%) answered with 'Disagree'. The statement received a usability rating of 89.33%. Three participants (14, 15, 30) provided written comments, all of which highlighted the ease at which they were able to perform the task. Participant_30 wrote: "[it] was very easy [for me] to navigate to profile and change the language" while Participant_14 felt that previous experience helped, stating: "It was easy for me because I know about profiles".

Q2: "The MobiSAM app's *Update profile* screen was easy to understand" The next statement, again, scored favourably, achieving a usability rating of 93.33% ($n = 15$). Nearly three-quarters (73.33%) of participants stated that they 'Strongly agree'; 20% stated 'Agree'; while one participant (6.67%) chose 'Neutral'. One written comment was provided by Participant_15 who stated that the UI was "very straight forward".

Q3: "Are there any other profile options you would like added to what is currently available?" The third and final question aimed to elicit additional options which participants might require within the *Update profile* screen. Responding to this question, 40% of participants answered 'Yes' while 60% answered 'No'. Four written comments indicated that participants (12, 19, 20, 30) wanted to be able to change their current suburb from within the client application, with Participant_12 stating: "For people who can't access the website regularly, it would be nice to [be able to] change addresses". Two participants (20, 27) suggested including the ability to obtain their position using their device's built-in GPS receiver, with Participant_20 writing: "Use GPS location for reporting problems". In addition, Participant_28 asked for the inclusion of additional languages, although they did not indicate which languages they required. Furthermore, Participant_30 felt it was necessary to include the mobile phone number of the account holder on the *Update profile* screen.

Discussion The mean overall usability rating for Task 4 was 91.3%. Researcher observations highlighted that Participant_5 and Participant_21 both needed to be reminded to press 'Save' in order to apply the changes made to their respective user profiles. Responses to the statements presented to participants after having performed Task 4 are summarised in Figure 6.13.

6.7.7 Post-intervention questionnaire

The final questionnaire was presented to all 30 participants and consisted of two separate sections. The first section focused on the navigational aspects of the client application, while the second focused on the application's visualisation functionality. Responses to each section are summarised in Figures 6.14 and 6.15 respectively. Similar to previously discussed tasks, a usability rating for each statement was calculated as well as an overall usability rating for each participant answering the questionnaire. (Please refer to Section 6.6 for an explanation on how these scores were calculated.)

Application navigation

This section was completed by all 30 participants and aimed to determine how intuitively participants were able to navigate within the client application.

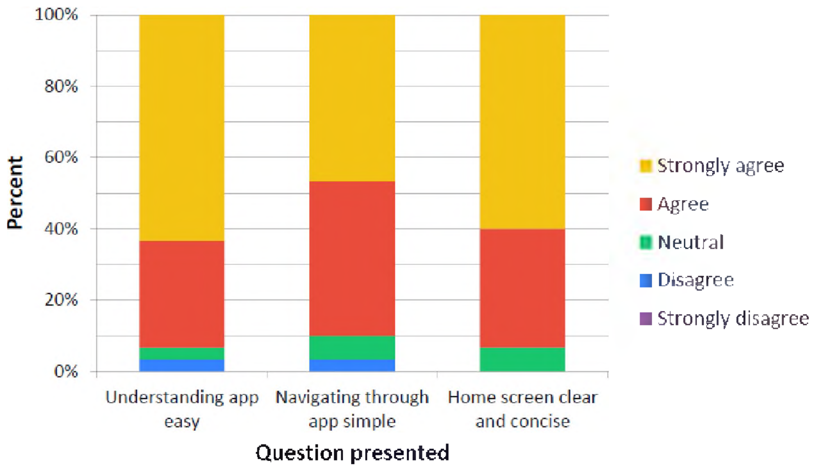


Figure 6.14: Summary of responses relating to client application navigation ($n = 90$)

Q1: "The MobiSAM app, in general, was easy to understand" In response to the first statement, 63.33% of participants selected 'Strongly agree'; 30% selected 'Agree'; while 'Neutral' and 'Disagree' were each selected by one participant (3.33%). This resulted in a usability rating of 90.67% ($n = 30$). Comments from Participant_14 again highlighted confusion with the concept of taking a poll, stating: "If the app [is] for reporting water [problems], it should specify that report instead of take a poll". Participant_22 said that they "misinterpreted the wording a lot" but found the application "very good" once they were familiar with it.

Q2: "Navigating my way through the MobiSAM app was simple" Of the 30 participants who answered the second statement, 46.67% indicated that they 'Strongly agree'; 43.33% indicated that they 'Agree'; two participants (6.67%) answered with 'Neutral'; while one (3.33%) chose 'Disagree'. A usability rating of 86.67% was achieved. Comments from two participants (14, 23) stated that they found navigation to be "fairly straight forward" and "relatively easy" respectively. Participant_11, on the other hand, stated that: "I was uncertain of the best option [to choose]".

Q3: "The *Home* screen of the MobiSAM app was clear and concise" Responding to the third statement, 60% of participants said that they 'Strongly agree'; 33.33% chose 'Agree'; while two participants (6.67%) chose 'Neutral'. The statement received a usability rating of 90.67% ($n = 30$). Written comments from two participants (22, 30) again indicated confusion with the phrase 'Take a poll', with Participant_22 stating: "'Take a poll' is confusing" however "the other two [menu options] are understandable". The same participant went on to elaborate that they found the layout "clear and simple", and that "the three [menu] options were well separated". Participant_30 suggested changing 'Take a poll' to two separate menu items: 'Report water problems' and 'Report other problems'.

Summary The resulting mean usability rating for the navigation portion of the post-intervention questionnaire was 89.33%. A summary of participant responses is illustrated by Figure 6.14.

Application visualisation

Due to the manner in which the study tasks were paired, only 25 participants performed tasks which included some form of visualisation. As such, the five participants who were assigned Task 2 and 4 did not answer this portion of the questionnaire.

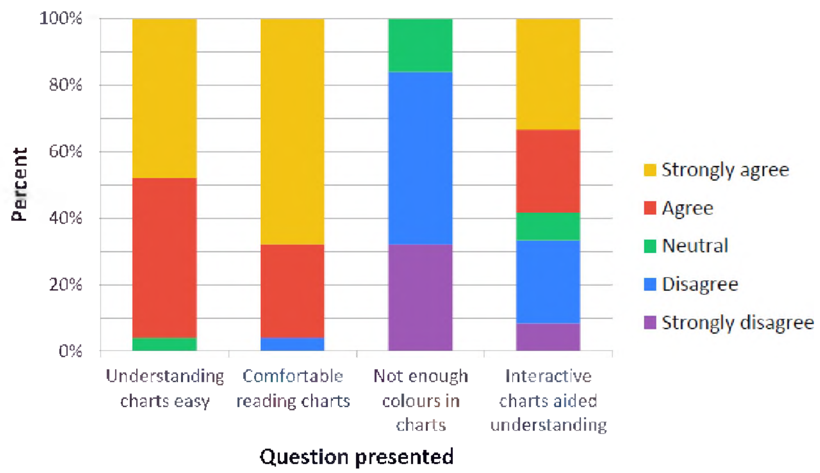


Figure 6.15: Summary of responses relating to application visualisation ($n = 87$)

Q5: "The charts, in general, were easy to understand" The first statement relating to chart visualisations scored highly, with 48% of participants stating that they 'Strongly agree'; 48% stating that they 'Agree'; while one participant (4%) chose 'Neutral'. The resulting usability rating was 88.8% ($n = 25$). Relating to comments from those who used the smartphone handset (i.e. interactive chart visualisations), Participant_22 requested that percentage values be included on the interactive charts. Furthermore, Participant_4 indicated that they found the charts understandable only once they had realised they were interactive. No written comments were provided by participants who used the feature phone.

Q6: "I felt comfortable reading the charts" Responses to the statement were mostly positive, with 68% of participants indicating that they 'Strongly agree'; 28% that they 'Agree'; while a single participant (representing 4% of the sample) said that they 'Disagree'. The statement was given a usability rating of 92% ($n = 25$). Two written comments from participants (4, 27) were provided and highlighted the benefits gleaned by visualising the returned poll results. Participant_4 wrote: "Even without knowing exact figures, it [the chart visualisation] provides a useful comparison of responses" while Participant_27 indicated that they found the charts to provide the "most compact information".

Q7: "Not enough distinct colours were used in the charts" Of the 25 responses received, 16% answered with 'Neutral'; 52% answered 'Disagree'; while 32% said that they 'Strongly disagree'. The resulting usability rating was 83.2%. A written comment was provided by Participant_22, stating: "The colour coding was very clear and appropriate to the keys they represented".

Q8: "The interactive charts made information more easily understandable" As the feature phone did not include interactive charts, only participants who performed the study using the smartphone handset were instructed to answer this statement, resulting in 12 responses. Of these, 33.33% chose 'Strongly agree'; 25% chose 'Agree' and 'Disagree' respectively; while one participant (representing 8.33%) chose 'Neutral' and 'Strongly disagree' respectively. A usability rating of 76.67% was achieved. Two participants (1, 11) indicated that they would prefer the actual values/percentages to appear on the charts instead of providing interactivity, with Participant_1 stating: "It is probably easiest to have numbers on the charts". Two participants (22, 30) stated that they did not realise the charts were interactive while Participant_12 mentioned that it would be "useful to have a loading indicator as the charts do take time to appear". Participant_30 suggested making the chart options more "distinct/visible for the user to find". Similarly, Participant_22 stated that: "There wasn't any affordance to know they [the charts] were interactive".

Q9: "Which of the visual representations did you find most useful?" Responses to the final question were mixed with 58.33% indicating that they preferred the pie chart; 12.5% chose the bar chart while 29.17% indicated that they found the numerical table most useful. As responses were not represented by a Likert scale, no usability rating was calculated. Participant_10 highlighted that "the pie chart doesn't have numerical labels" and that "having numerical figures on the pie chart can improve [its] usability".

Q10: "Is there anything else about the MobiSAM app that you would like to tell us?" As a final question, participants were asked to provide comments and suggestions relating to the client application. Participant_4 indicated that it would be useful to view poll results for a specific time frame, for example, the number of people without water today/this week/this month. Participant_15 suggested implementing a mechanism to inform users that their reported issue is currently being attended to. Finally, two participants (1, 11) indicated that MobiSAM would be very useful *if* Makana Municipality takes action when issues are reported.

Summary The mean usability rating across the 25 participants answering the visualisation section of the post-intervention questionnaire was 88%. Responses are summarised in Figure 6.15.

6.8 Analysis and discussion

Feedback obtained from the first usability evaluation highlighted a number of interesting findings. This section draws together and discusses each of them. Results are presented in four separate sections: demographics (Section 6.8.1); use of mobile technology (Section 6.8.2); navigation (Section 6.8.3); and visualisation (Section 6.8.4). Usability metrics across each task are examined (see Section 6.6), from which application-wide trends are then drawn.

6.8.1 Demographics

Examining the age groups of the study sample reveals that there were at least three participants across each group, with the majority of participants being 18-30 years (56.67%). The split between female and male participants was nearly equal, with 43.33% female and 56.67% male.

Furthermore, 80% of participants indicated that English was their preferred language while 20% specified isiXhosa. The majority of participants were Black African/Coloured (60%); 36.67% were White; while a single participant (3.33%) was Indian/Asian. All participants had at least attended "some high school", with 56.67% of the study sample having graduated from university.

With this in mind, it was decided to investigate how different participant groups performed based on their preferred language and age group.

Preferred language

As illustrated in Figure 6.16, participants who specified English as their preferred language completed 75% of the tasks they performed, while participants who chose isiXhosa were able to complete 58.33% of tasks.

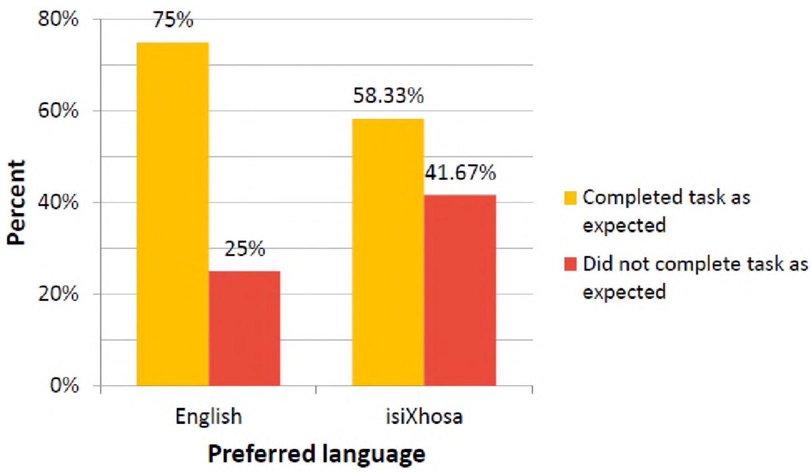


Figure 6.16: Task completion rates ($n = 60$)

This notable difference may have been caused by several factors. Firstly, as the client application's UI was only available in English during the study, isiXhosa preferred language participants may have been put at a disadvantage, resulting in a fewer number of tasks being completed. This reconfirms the need for multi-language support within each client application. It is important to note that participants were only asked to specify their preferred language, and not whether they were comfortable reading in this language. This may have further contributed to these results.

Secondly, 83.33% of isiXhosa preferred language participants indicated matric as their highest level of education, while 75% of English preferred language participants had obtained a university degree. This difference in education levels may have also influenced task completion rates. In addition, the sample size of the two participant groups was very different. While there were 24 English preferred language participants there were only six isiXhosa preferred language participants. As such, these results may not offer an accurate representation of the target audience.

	Feature phone		**Smartphone**	
	English	**isiXhosa**	**English**	**isiXhosa**
Participant mean	72.07s	75.60s	54.98s	39.10s ($n = 1$)

Table 6.9: Study-wide mean task duration

Referring to the study-wide mean task durations. When using a feature phone, English preferred language participants were slightly quicker than those who indicated isiXhosa to be their preferred language (see Table 6.9). This result is not unusual as the client application's UI was presented in English. Conversely, isiXhosa preferred language participants were quicker when using the smartphone. However, upon closer examination of the results, only a single isiXhosa participant (Participant_6) performed the evaluation using the smartphone. Therefore, the result of 39.1s cannot be considered to be truly representative.

	Feature phone		**Smartphone**	
	English	**isiXhosa**	**English**	**isiXhosa**
Participant median	5 presses	4 presses	5 presses	7 presses ($n = 1$)
Expert user median	5 presses		5 presses	

Table 6.10: Median number of button presses made per completed task

As summarised in Table 6.10, isiXhosa preferred language participants using the feature phone made a fewer number of button presses per task when compared to English participants (4 versus 5). In addition, they made fewer button presses compared to an expert user. A possible reason for this is that 60% of isiXhosa preferred language participants using the feature phone performed Task 4. This task only requires three button presses to complete, hence skewing the results. Conversely, the expert user median is calculated across all four tasks, thus resulting in a higher median number of button presses.

	Feature phone		Smartphone	
	English	isiXhosa	English	isiXhosa
Participant mean	85.25%	78.83%	87.02%	82.50%

Table 6.11: Study-wide mean usability rating

Finally, consulting the mean usability ratings presented in Table 6.11, English preferred language participants rated tasks more favourably than isiXhosa preferred language participants. Once again, lower task satisfaction levels amongst isiXhosa preferred language participants may have been due to difficulties arising as a result of the application's UI only being available in English.

Age group

In an attempt to determine whether age affected task performance, it was decided to split the study sample into two groups: 18-45 years (73.33%) and over 45 years (26.67%).

Referring to task completion rates illustrated in Figure 6.17, it can be seen that younger participants were less likely to complete tasks, with 68.18% of all tasks performed by participants 18-45 years being completed while 81.25% of all tasks performed by participants over the age of 45 were completed.

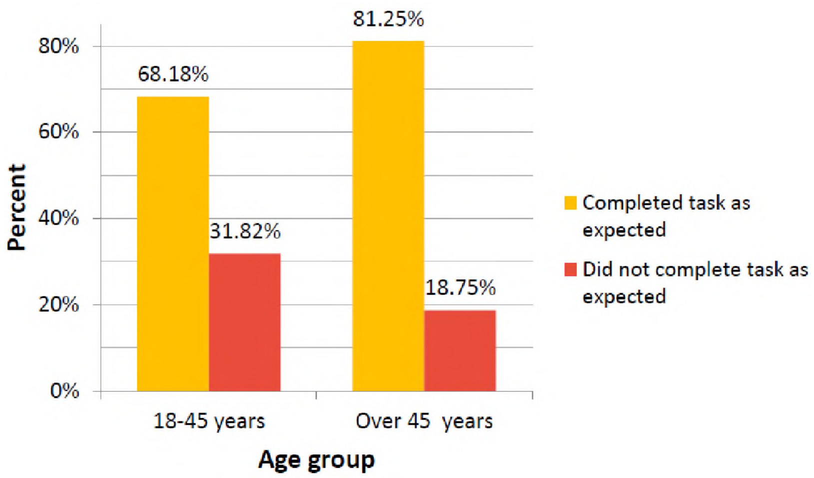


Figure 6.17: Task completion rates ($n = 60$)

Although these results seem to contradict the common held belief that younger individuals are more proficient at using modern technology than their older counterparts, the reason for this difference may be due to the fact that older participants exercised more caution when performing each task, carefully reading task instructions before proceeding. Conversely, younger participants were more likely to rush through the task in an attempt to finish as quickly as possible, resulting in a greater number of mistakes being made.

	Feature phone		Smartphone	
	18-45 years	Over 45 years	18-45 years	Over 45 years
Participant mean	60.65s	128.32s	50.98s	59.53s

Table 6.12: Study-wide geo mean task duration

The mean task durations shown in Table 6.12 confirm this, with participants aged 18-45 taking, on average, 60.65s to complete a task when using the feature phone, while those over 45 years took 128.32s. A similar trend was observed between the participant groups when using the smartphone.

	Feature phone		Smartphone	
	18-45 years	Over 45 years	18-45 years	Over 45 years
Participant median	5 presses	9 presses	6 presses	5 presses
Expert median	5 presses		5 presses	

Table 6.13: Median number of button presses made per completed task

Examining the median number of button presses made by participants when completing study tasks (see Table 6.13), younger participants who used the feature phone made fewer button presses than their old counterparts. Conversely, older participants who used the smartphone made, on average, fewer button presses than younger participants. A reason for this is likely due to the outlier value of 23 button presses, made by a younger participant (Participant_9) when performing Task 1. Removing this outlier reduces the value from 6 to 5, making the median number of button presses in both groups equal. There were no outlier values present within the older participant group sample.

	Feature phone		Smartphone	
	18-45 years	Over 45 years	18-45 years	Over 45 years
Mean per task	86.60%	69.17%	88.75%	82.67%
Overall	83.11%		86.72%	

Table 6.14: Study-wide mean usability rating

As summarised in Table 6.14, even though the younger participant group completed fewer tasks when compared to the older group, they gave a more favourable overall usability rating. A possible reason for this difference may be due to the older participants being more critical of ambiguities and imperfections encountered while performing tasks, whereas the younger participants were more forgiving, possibly as a result of their exposure to technology from a young age. In addition, younger participants may have rated tasks more favourably so as to convey a positive image of themselves to the researcher and research assistant.

6.8.2 Use of mobile technology

Examining responses relating to participants' prior use of mobile technology and services, all 30 participants indicated that they owned a mobile phone. Nearly all participants (96.67%) had used a mobile phone for six years or more, indicating that they were likely to be competent users. Relating to device brands, 53.33% owned a Nokia device, while 26.67% owned a Samsung handset. Of the participants who owned a Nokia device, 68.75% chose to perform the study using the Nokia-branded feature phone, while 75% of participants who owned a Samsung device chose to use the Samsung-branded smartphone.

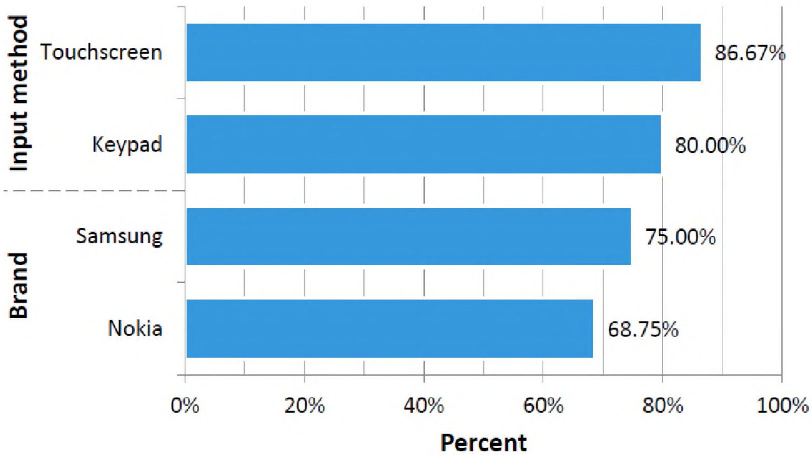


Figure 6.18: Proportion of participants whose personal handset and study handset matched: brand versus input technique ($n = 30$)

Reviewing these results, it seems as if participants chose handsets based on the input technique (keypad or touchscreen) they were accustomed to, irrespective of device brand. Referring to Figure 6.18, 80% of participants who owned a keypad device performed the study using the keypad-equipped feature phone while 86.67% of participants who owned a touchscreen device chose to use the touchscreen-driven smartphone.

Data usage

The majority (63.33%) of study participants ranked data services above both voice and SMS in terms of frequency of use, while five participants (16.67%) indicated that they never used data services on their mobile device. As such, it was decided to determine how the task performance measures (see Section 6.6) varied between these two groups.

Starting with the rate of task completion, data users completed a greater proportion of tasks compared to non-data users. As illustrated in Figure 6.19, 74% of tasks performed by data users were completed while this number stood at 60% for non-data users.

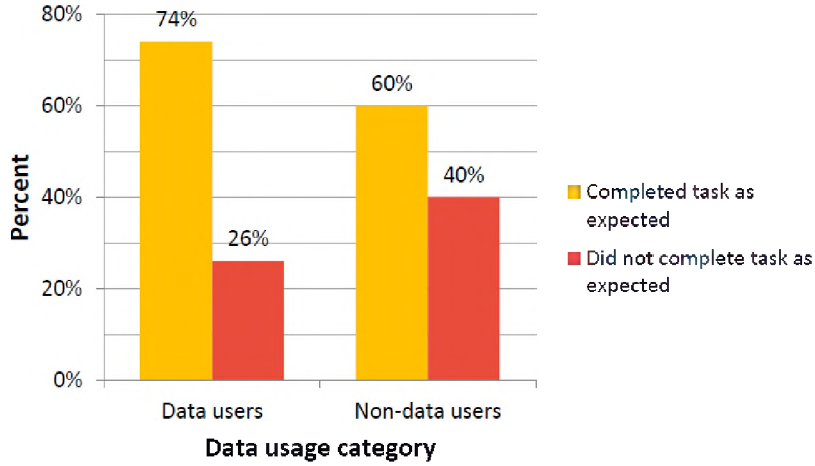


Figure 6.19: Task completion rates ($n = 60$)

It is hypothesised that the higher task completion rate amongst data users is likely due to previous experience with mobile applications and Internet services, thus allowing them to become familiar with the client application more quickly. Conversely, non-data users' mobile phone usage patterns are likely to have been limited to voice calls and SMS.

Four out of the five non-data users chose to perform the study using the feature phone, initially suggesting that they felt more comfortable using an entry-level device. However, upon further review of the results it became apparent that four out of the five non-data users owned a feature phone, a likely reason as to why they chose to perform the study using a similar handset. Interestingly, 80% ($n = 4$) of non-data users were over 45 years, playing to the stereotype that older individuals are less 'tech-savvy' when compared to their younger counterparts.

As reflected in Table 6.15, data users took less time when completing tasks, again, a possible indication that they were more comfortable using mobile applications.

	Feature phone		**Smartphone**	
	Data users	**Non-data users**	**Data users**	**Non-data users**
Participant mean	62.05s	103.16s	53.38s	75s ($n = 1$)

Table 6.15: Study-wide mean task duration

As summarised in Table 6.16, data users who completed tasks using the feature phone made, on average, fewer button presses when compared to non-data users. An identical number of button presses were made between each group when using the smartphone.

As there was only a single non-data user (Participant_1) who performed the study using the smartphone, their recorded task performance measures are likely to be an inaccurate representation of how the "average" participant would perform, given a larger sample size. In addition, Participant_1 did not make use of a smartphone in their personal capacity, further reducing the accuracy of these results.

	Feature phone		**Smartphone**	
	Data users	**Non-data users**	**Data users**	**Non-data users**
Mean participant median	5 presses	9 presses	5 presses	5 presses ($n = 1$)
Expert user mean	5 presses		5 presses	

Table 6.16: Median number of button presses made per completed task

Referring to Table 6.17, data users rated tasks more favourably when compared to non-data users. Data users' previous experience using mobile applications and accessing web content from their handsets it likely to have better prepared them for using the client application, hence the more favourable usability rating when compared to non-data users.

	Feature phone		**Smartphone**	
	Data users	**Non-data users**	**Data users**	**Non-data users**
Mean per task	86.29%	74.38%	87.56%	75% ($n = 1$)
Overall	83.11%		86.72%	

Table 6.17: Study-wide mean usability rating

Again, the usability rating given by non-data users, who used the smartphone, is likely to be a poor estimate, due to there only being one participant in this particular sample.

Platform correlation

As the client application is available for several mobile platforms, it was decided to determine how the task performance measures varied between participants whose personal handset's platform matched the platform which they used during the study (same platform participants), compared to those whose handset's platform did not match (different platform participants).

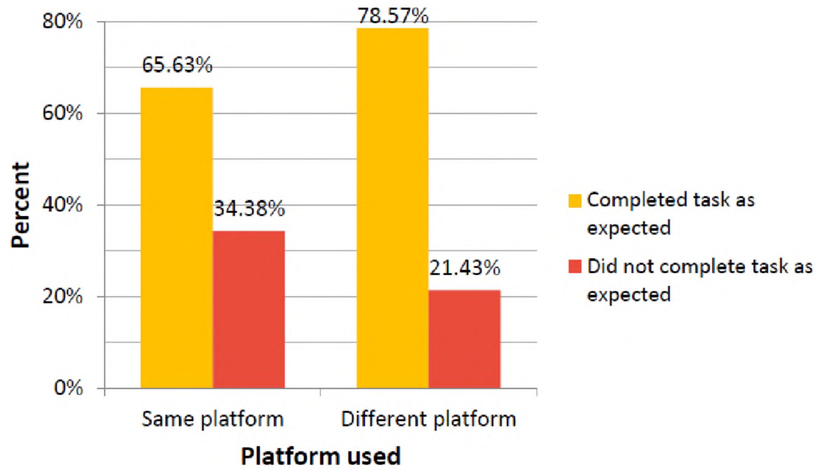


Figure 6.20: Task completion rates ($n = 60$)

Figure 6.20 shows that a lower proportion of tasks were completed by participants who performed the study using the same platform to what was found on their personal handset. However, upon examining these results further it became apparent that four out of the five non-data users were included in this group, as well as four out of six isiXhosa preferred language participants. Two confounding factors which may have influenced the results.

As described in Section 6.8.1, both of these groups did not perform as favourably as their counterparts, which may have resulted in a lower proportion of tasks being completed than one would expect. Although factoring out these participants reduced the sample size significantly (and therefore how representative the sample was), in doing so the task completion rate for same platform participants increased from 65.63% to 88.89%.

	Feature phone		**Smartphone**	
	Same	**Different**	**Same**	**Different**
Participant mean	74.61s	64.67s	55.93s	53.07s

Table 6.18: Study-wide mean task duration

In addition to same platform participants completing a lower proportion of study tasks, the tasks which they did complete took longer to perform when compared to different platform participants, regardless of the handset used (see Table 6.20).

	Feature phone		**Smartphone**	
	Same	**Different**	**Same**	**Different**
Median per task	5 presses	5 presses	5 presses	5 presses
Expert user	5 presses		5 presses	

Table 6.19: Median number of button presses made per completed task

Table 6.19 shows that both groups of participants made an identical median number of button presses per task, irrespective of the handset used during the study. These results highlight the possibility that, whether participants used the client application on a platform which they were familiar with, or not may have been irrelevant.

	Feature phone		**Smartphone**	
	Same	**Different**	**Same**	**Different**
Mean per task	81.75%	85.83%	86.53%	86.85%
Overall	83.11%		86.72%	

Table 6.20: Study-wide mean usability rating

The mean usability rating per task was also very similar between the two groups, again reiterating the previous statement.

6.8.3 Navigation

A primary aim of the usability evaluation was to determine how intuitively participants were able to navigate within the client application. The study hoped to highlight any areas of concern which would then be remedied during the next iteration of the life-cycle model. Instead of proceeding chronologically through each of the study's tasks (detailed in Sections 6.7.3 to 6.7.6), analysing and drawing conclusions from each, the main navigational issues emerging from the study as a whole are identified and discussed. Issues are ordered by their frequency of occurrence.

Issue 1: Answering service delivery polls

Starting with the most frequently cited issue, the concept of having to 'Take a poll' was often misunderstood by participants.

Consulting results from Task 1: Q2 ("Finding the correct poll was confusing"), exactly two thirds ($n = 10$) of participants answered 'Strongly agree', 'Agree', or 'Neutral', indicating that the majority of participants experienced some difficulty when attempting to find the correct poll. When asked to elaborate, six participants (1, 4, 11, 23, 27, 30) cited confusion surrounding the use of the term "poll", as well as the idea of having to 'Take a poll' when wanting to inform their municipality about a service delivery issue.

This confusion was reiterated by the low usability rating of 58.67% for this statement. Although Task 2: Q1 ("Finding the correct poll was straightforward") scored more favourably, written comments from three participants (15, 20, 22) also highlighted confusion of having to 'Take a poll'. Similarly, one written response to Task 3: Q1 ("Finding the correct poll was confusing") cited misunderstanding in having to answer service delivery polls, in which Participant_14 wrote: "I thought it [the client application] would specify that I'm reporting water quality".

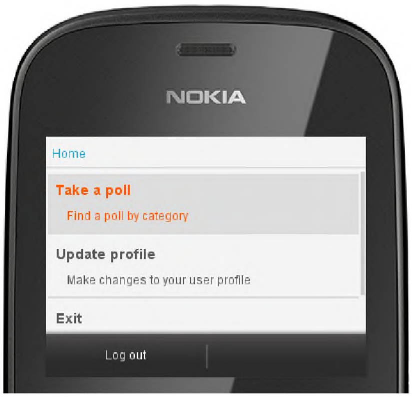


Figure 6.21: 'Take a poll' *Home* screen menu option

Studying participant responses more closely, two possibilities for the confusion were identified. The first and most likely cause was a simple mismatch between the wording used in the task questionnaires (see Appendix B.2) and the wording used on the client application's *Home* screen, shown in Figure 6.21. Participants, for example, were instructed to "inform Makana Municipality that you currently have no running water at your home/residence", which made no mention of the word "poll". However, in order to accomplish this task, participants were expected to select 'Take a poll' from the *Home* screen.

Secondly, a disconnect seemed to exist between the concept of answering service delivery polls (for example, "Do you have running water today?") and keeping Makana Municipality informed about service delivery concerns. This was typified by a comment from Participant_22, who wrote: "'Take a poll' wasn't intuitive to know this is where I needed to navigate to report an issue". While MobiSAM was originally proposed as a mobile phone-based polling framework used for the ongoing monitoring of basic services provided by Makana Municipality, it later became apparent that citizens were more concerned about being able to report *specific* service delivery grievances. Although, to the researcher, the phrase 'Take a poll' became synonymous with reporting a service delivery complaint (hence the menu option of the same name). In hindsight, it is understandable how this terminology confused first-time users.

Inspecting the median number of button presses made during Task 1 (see Table 6.5) reveals that participants made more than the minimum number of button presses required to complete the task, irrespective of the handset used. This may further indicate confusion surrounding the terminology used within the client application, although unfamiliarity (as it was participants' first time using the application) is also likely to have been a confounding factor.

As described in Section 5.7.2, a single design change was made to the *Home* screen in an attempt to sidestep this usability concern. The change aimed to draw a distinction between reporting a service delivery problem (a once-off occurrence), and answering service delivery polls (the continual monitoring of basic service delivery within Grahamstown). This was achieved by introducing an additional menu item: 'Report a problem'. Furthermore, a simple stylistic icon was included alongside each *Home* screen menu option in order to reinforce their purpose, as illustrated in Figure 5.9.

Issue 2: Ambiguous category names

The second most frequently encountered issue was caused by the ambiguous phrasing of certain menu options, particularly those found within the *Category* screen. This resulted in participants not completing the task as *expected*, even though the route navigated when performing the task constituted a logically valid answer.

Referring to Task 1, 40% of participants did not complete the task as expected. A deeper study of these results revealed that 83.33% ($n = 5$) of the unexpected answers were due to participants (1, 6, 17, 29, 30) selecting the incorrect category. Results from Task 2 showed a similar trend. Of the six participants (8, 21, 22, 24, 25, 26) who did not complete the task as expected, all of them were as a result of the incorrect category being chosen. The same held true for Task 3, where again all participants (3, 17, 19, 22, 28) who provided a unexpected answer ($n = 5$) did so as a result of choosing the incorrect category. This usability concern was highlighted by Participant_9 who commented on how the category 'Talk to your municipality' and 'Water monitoring' could ultimately be used to report the same issue, as there was no clear distinction between the two.

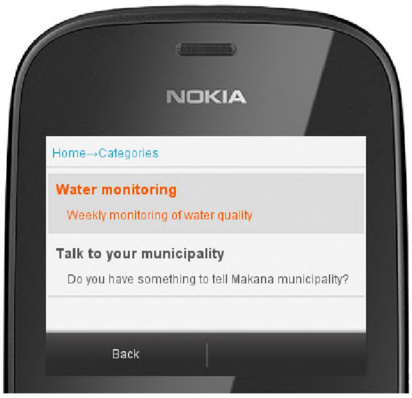


Figure 6.22: Ambiguous category names

This recurring pattern emphasised the importance of carefully selecting menu option names, while being mindful of potential ambiguities. For example, when participants considered what category to select, in order to inform Makana Municipality about a water outage (see Figure

6.22), both 'Water monitoring' and 'Talk to your municipality' seemed like valid choices, however only one was deemed an *expected* answer by the researcher (in this case 'Water monitoring'). In an attempt to mitigate this issue, each category and category description was rephrased, maintaining a clear distinction between each (see Figure 5.11a). The fact that 20% of study participants indicated that isiXhosa was their preferred language may have further exacerbated any potentially ambiguous menu options, re-emphasising the need for multi-language support.

Interestingly, although the same number of participants (40%) were unable to complete Task 1 and 2 as expected, their responses to the statement asking them about the difficulty experienced finding the correct poll varied hugely. Task 1: Q2 ("Finding the correct poll was confusing") received a usability rating of 58.67%, while Task 2: Q1 ("Finding the correct poll was straightforward") received a usability rating of 84%. This result may possibly be attributed to the fact that five out of the 15 participants who attempted Task 2 did so after having first attempted Task 1, and were therefore likely to have learnt from previous experience. Conversely, all 15 participants attempting Task 1 did so without having attempted any other tasks.

It is important to note that due to the manner in which tasks were paired (see Section 6.3), Task 1 was never performed after any other task during the evaluation, resulting in a total of 15 first-time exposures. Task 2, on the other hand, was presented first to participants a total of 10 times, while Task 3 was presented first a total of five times. Task 4 was always presented to participants last.

Issue 3: Forgetting to save changes

The third and final issue relates to Task 4 exclusively. Although all 15 participants managed to complete the task as expected, two required guidance in the form of having to be reminded to press the 'Save' soft key before returning to the *Home* screen (see Figure 6.23).

Further examination of the results revealed that both instances occurred with participants who performed the study using the feature phone, the design guidelines of which specifically require users to save changes before they are applied [158].

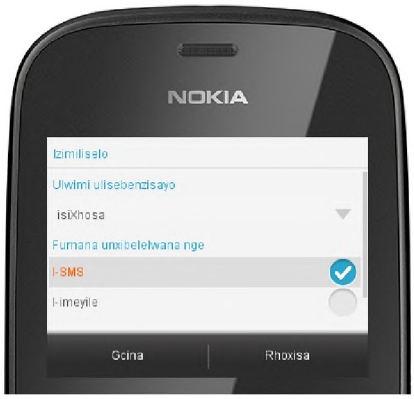


Figure 6.23: *Update profile* screen 'Save' soft key

Coincidently, both of these participants owned the identical handset to that of which they used in the study (a Nokia Asha 201), suggesting that their mistakes were likely not to have been caused by unfamiliarity with the device. Both participants however specified isiXhosa as their preferred language, which may have been a contributing factor. Interestingly, the issue was not experienced by any participants who performed the study using the smartphone, even though the same 'Save' option was mandatory. As a solution, the client application was modified so as to prompt users to save changes before navigating back to the *Home* screen.

The median number of button presses made by participants when attempting Task 4 was identical to the minimum number required (3), highlighting the ease with which the majority of participants were able to complete it as expected. This favourable result may have been due to the fact that the task was always performed last by participants. In addition, the task was simpler than other tasks, merely requiring participants to change the client application's language from within the *Update profile* screen.

Additional observations

It was decided to examine the number of log in attempts a participant made when performing their first assigned task, compared to their second task. Eight participants (1, 2, 5, 10, 12, 22, 26, 27) made fewer log in attempts when performing their second task, compared to four participants (14, 19, 20, 24) who made a greater number. This suggests that the client application possesses a certain degree of learnability. Similarly, 12 (1, 5, 6, 8, 19, 21, 24, 25, 26, 28, 29, 30) out of the 14 participants who did not manage to complete their first task were able to complete their second. While only a single participant (Participant_3), who managed to complete their first task, did not manage to complete their second. Again, highlighting the learnability of the client application.

6.8.4 Visualisation

Another aim of the usability evaluation was to determine how effectively different chart visualisations conveyed poll results to participants. Particular emphasis was placed on determining which charts participants found easiest to understand, as well as what role interactivity within the charts played in helping participants gain further insight into the data. Results are presented in three sections: chart preference (Section 6.8.4), interactivity and understanding (Section 6.8.4) as well as usability concerns (Section 6.8.4).

Chart preference

Two questions were posed to participants which aimed to determine which chart type they preferred. The first question ("Which chart did you find easiest to understand?") was presented to participants after having completed Task 3. Responses indicated that slightly under half (46.67%) of participants found the pie chart the easiest to understand; 20% indicated preference for the bar chart; while 33.33% rated them as both equal. The second question ("Which of the visual representations did you find most useful?") formed part of the post-intervention questionnaire, to which 46.67% of participants again indicated their preference for the pie chart.

Although these results cannot be considered unusual due to the self-explanatory nature of a pie chart, Cleveland, in his book titled *The Elements of Graphing Data*, argues that human perception is less accurately able to judge differences in area (the technique used by pie charts) compared to differences in length (the technique used by bar charts), possibly leading to inaccurate readings [31]. Conversely, pie charts are very effective at conveying the "part-to-whole relationship" [44, p.6], an important characteristic when comparing what proportion of residents within a given suburb are experiencing a particular service delivery issue.

Written responses from Participant_10 and Participant_25 highlighted confusion with the choice of unit used on the dependent variable (y-axis) of the bar chart. This may possibly have been a contributing factor as to why the pie chart took preference. Interestingly, many participants were observed using a combination of the charts, along with the numerical table when interpreting results, especially those who did not realise the smartphone charts were interactive. Ultimately, it was decided to keep both the pie chart and numerical table, and remove the bar chart, the technique rated least favourably by participants.

Interactivity and understanding

Study participants using the smartphone ($n = 15$) were presented with unlabelled, interactive charts. Participants were expected to 'tap' areas of interest within the charts, thereby revealing hidden information (such as the size of the selected sector). Part of the study aimed to determine whether this interactivity encouraged participants to explore the results, helping them gain further insight into the presented data.

As described in Section 6.5.1, three-quarters (75%) of participants who performed Task 1 using the smartphone indicated that they did not make use of the chart interactivity. Although this seemed surprising at first, observations revealed that many participants were unaware that the charts were in fact interactive, even though the screen title clearly stated: ‘Tap chart for more details’. Three participants (2, 22, 24) indicated that instead of having to interact with the chart, they would prefer the necessary information to appear directly on the chart itself, thereby allowing them to glean the necessary information more easily.

Interestingly, although only 25% of the participants who performed Task 1 using the smartphone indicated that they had used chart interactivity, the majority of responses to the post-intervention questionnaire statement “The interactive charts made information more easily understandable”, were positive. This resulted in a usability rating of 76.67%. A possible reason for this favourable rating is that participants who used the smartphone when performing the study ($n = 15$) were comparing this additional information provided to the blank chart initially presented (see Figure 5.8a).

Observations and concerns

After having performed Tasks 1 and 3 as expected, participants were presented with the *Results* screen, similar to that shown in Figure 6.24. The screen contained a numerical table, as well as the option to view two charts summarising responses to the given poll question. By examining these charts, participants were asked to note the number and percentage of people experiencing the exact issue they were experiencing.

Charts presented to participants who completed Task 1 displayed the number and percentage of people without water (answer: 131 people/83% of all votes). While charts presented to participants who completed Task 3 illustrated the number and percentage of people who were experiencing a water quality issue. Participants were asked to note those experiencing brown water (answer: 83 people/57% of all votes).

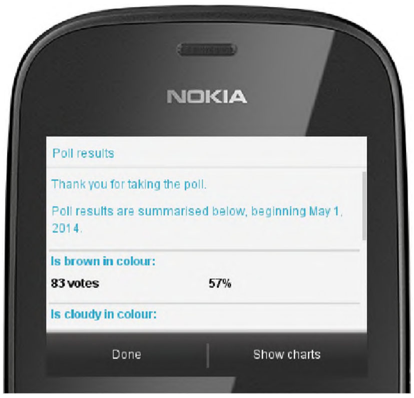


Figure 6.24: Numerical table summarising poll results

Consulting the written answers from participants who performed Task 1, it can be seen that only four participants (9, 10, 16, 17) correctly noted these figures. Furthermore, two of these participants (9, 10) were observed incorrectly consulting the numerical table, instead of the charts (as instructed by the questionnaire) when determining their answers.. Although this might seem like a low number, it should be noted that only eight out of the 15 participants performed Task 1 as expected, and as such only these participants were presented with the *Results* screen.

Interestingly, although only four participants correctly read the values from the charts, the majority of participants responded favourably when presented with the statement: "[t]he charts were easy to understand". This resulted in a usability rating of 88.57%. Two possible reasons for this result have been identified. Firstly, as participants were not told the correct answer after having performed the task, they may have been unaware that they interpreted the values incorrectly. Secondly, as with previous statements, participants may not have wished to convey a negative reflection of themselves, and therefore answered more positively than they otherwise would have.

Written answers for Task 3 were noticeably more accurate when compared to Task 1, with 66.67% ($n = 10$) participants correctly interpreting chart values. As previously mentioned, this improved result is likely to have been caused by participants' exposure to other tasks before performing Task 3, thereby improving their familiarity with the client application. This may potentially highlight the application's learnability. As a case in point, 10 out of the 15 participants who performed Task 3 had performed a prior task, while none of the 15 participants who performed Task 1 had done so. An equal number of English and isiXhosa participants performed each task, therefore language preference is unlikely to have affected the results. Observational notes made during Task 3 again highlighted that two participants (18, 25) read values from the numerical table instead of from the charts, as instructed. Participants responded slightly more favourably to Task 3's statement: "[t]he charts were easy to understand", resulting in a usability rating of 90.67%.

A minor usability concern highlighted by five participants (6, 7, 11, 20, 23) was the use of the term 'votes' on the *Results* screen. This issue is likely as a result of participants misunderstanding that when asked to inform Makana Municipality about a service delivery problem, they were in fact answering a service delivery poll. Alternative suggestions from participants included renaming votes to either respondents, people or comments.

6.9 Summary

This chapter presented the first of two usability evaluations, performed using the cross-platform client application. The study highlighted problems experienced by participants while navigating the client application, particularly misunderstandings surrounding the use of certain words ('poll', 'votes') and phrases ('Take a poll') as well as ambiguities within the menu options ('Water monitoring', 'Talk to your municipality'). The study also indicated which chart type participants found most understandable, as well as whether interactivity within the charts helped better convey meaning. The task performance measures from several different sample groups within the study were compared and discussed, drawing out interesting trends and findings. The following chapter details the comparative usability evaluation performed using the MobiSAM and MobiSAM Report cross-platform client applications.

Chapter 7

Evaluation: client application comparison

7.1 Introduction

This chapter presents the second and final usability evaluation and was performed using both the MobiSAM and MobiSAM Report client application. The study aimed to determine which of the two applications participants preferred, stating their reasons why. The study sample (Section 7.2), data-collection instruments (Section 7.3) as well as method and analysis (Section 7.4) are first described. The study tasks (Section 7.4) and task performance metrics (Section 7.6) are then detailed. Next, the results (Section 7.7) of each task are presented. Finally, a detailed analysis and discussion of these results is given (Section 7.8), centred around three of Nielson's five quality components of usability: learnability, memorability and satisfaction [156].

7.2 Study sample

Instead of conducting the study on a new set of participants, it was decided to employ a self-selection sampling technique, enlisting 10 participants who had taken part in the first usability evaluation (see Chapter 6). Although self-selection techniques indicate that the sample is not truly representative of the Grahamstown population at large (and therefore the target audience), it allowed the researcher to evaluate the memorability of the MobiSAM client application, gauging how quickly participants were able to re-establish their proficiency when using the application. Furthermore, as all 10 study participants had previously used the MobiSAM client application, they were able to reflect upon the design modifications made to the original. A detailed description of the study sample is presented in Section 7.7.1.

7.3 Data-collection instruments

Two new questionnaires were developed as part of the usability evaluation: a study tasks questionnaire and a post-intervention questionnaire. In addition, the pre-intervention questionnaire from the first usability evaluation was adapted and reused. The study tasks questionnaire consisted of four tasks spanning a cross-section of MobiSAM and MobiSAM Report's functionality. As with the previous evaluation, each task aimed to be representative of the client applications' real-world use.

A single question appeared after each task, simply asking participants: "Overall, this task was?" Responses to the question were represented by a five-point Likert scale with options ranging from 'Very easy' to 'Very difficult'. The question aimed to elicit information relating to each participant's task-performance satisfaction.

The post-intervention questionnaire consisted of seven questions and aimed to assess MobiSAM and MobiSAM Report according to three of Nielsen's five quality components: learnability, memorability and satisfaction [156]. The first component was chosen as client applications need to be easily learnt by citizens, without extensive training. Citizens also need to be able to quickly re-establish their proficiency at using the application, as it is likely that there will be long periods where the application is not used. The third component was selected in an attempt to ensure that citizens find the user interface pleasant to use and are therefore likely to use the application again in future (instead of seeking other avenues of complaint).

Finally, a question asking participants to describe their likes, dislikes and recommendations for each of the client applications was also presented. Responses to the post-intervention questionnaire were not restricted to predefined answers, instead eliciting open-ended responses from participants.

A copy of the study questionnaires are provided in electronic Appendix B.3.

7.4 Method and analysis

As per University protocol, ethics approval was granted (tracking number CS14-10) before proceeding with the study. Ten participants who had taken part in the first usability evaluation, and indicated that they were willing to participate in subsequent studies, were recruited using a self-selection sampling technique. Participants were informed about the purpose of the study and how it would proceed, after which they were asked to sign a consent form. As with previous studies, no remuneration or incentives were offered.

Participants were split into two groups, with five participants first performing the study tasks using MobiSAM and then MobiSAM Report (Participant_1A to Participant_5A). While the remaining five participants first performed the tasks using MobiSAM Report, followed by MobiSAM (Participant_1B to Participant_5B). Participants were provided with a Samsung Galaxy Pocket smartphone on which both MobiSAM and MobiSAM Report were installed. They were then shown both applications and given two minutes to explore each. Once the time had expired, they were asked to proceed with Task 1: logging into the application. After having completed the task as expected, participants were presented with the question: "Overall, this task was?", with responses being recorded using a five-point Likert scale.

The task duration as well as the number of display tap events were also recorded. Participants were instructed to perform Tasks 1 to 4 sequentially, with the researcher offering assistance when required. A within-groups study design was used whereby the same 10 participants were used to test both client applications. Once a participant had performed all four tasks using MobiSAM, they were asked to repeat the tasks using MobiSAM Report. The order in which each client application was presented to participants was counter-balanced in an attempt to mitigate learning effects. As participants performed a greater number of tasks when compared to the first study, the length of each task was substantially shorter, in an attempt to minimise participant fatigue.

7.5 Study tasks

Four tasks were constructed for the usability evaluation, spanning a broad cross-section of each client application's functionality. Each of the tasks are described next.

7.5.1 Task 1: Log in

The first task aimed to assess how easily participants were able to log in to each client application. The task began by presenting participants with the scenario: "You have just turned on the tap and realised that there is no running water", then:

1. Participants were instructed to log in to either MobiSAM or MobiSAM Report using the provided username and password combination.
2. Once successfully logged in, participants were asked to proceed to the second task.

7.5.2 Task 2: Report water outage

Task 2 instructed participants to report a water outage and consisted of two steps:

1. First, participants were asked to report a water outage to Makana Municipality, at their current position, obtained using GPS.

2. After having successfully reported the issue, participants were asked to navigate back to the *Home* screen.

As no results are provided after submitting a report using MobiSAM or MobiSAM Report, participants were not required to note any figures. Once the report had been successfully submitted, participants were instructed to perform Task 3.

7.5.3 Task 3: Report water bubbling out of the ground

In the third task, participants were presented with the scenario: "You have just seen water bubbling out of the ground at 1 High Street. Please report this problem to Makana Municipality". Again, the task consisted of two steps:

1. First, participants were instructed to report the issue to Makana Municipality at the provided address (1 High Street). They were asked to also include a photo of the issue being reported. To simulate participants discovering this issue, they were instructed to take a photo of a printed picture which depicted water bubbling out of the ground.

2. Once their report was successfully submitted, participants were instructed to return to the application's *Home* screen.

Similar to Task 2, no results were presented to participants. After completion of task 3 participants were asked to move on to the final task, Task 4.

7.5.4 Task 4: Change application language

Task 4 aimed to determine how easily participants were able to change their preferred language within MobiSAM and MobiSAM Report, and consisted of the following steps:

1. Starting at the *Home* screen, participants were asked to change the application language from English to isiXhosa.

2. Participants were then instructed to navigate back to the *Home* screen, now presented in isiXhosa.

Following the Android design pattern, changes were applied automatically, therefore no 'Save' button was provided.

Once participants had completed all four tasks using both client applications, they were presented with the post-intervention questionnaire. Participants were asked to complete the questionnaire verbally with responses being written down verbatim by the researcher. As with previous studies, questionnaires were only available in English with the research assistant acting as translator when required. Participant responses were guaranteed to remain anonymous and would not be traced back to them.

Each evaluation took between 15 and 25 minutes to complete, depending on the speed of the participant and the length of the responses given. Responses were captured electronically using Google Forms and later analysed with Microsoft Excel and IBM SPSS (a statistical analysis software package).

7.6 Task performance measures

Several different performance measures were recorded in an attempt to determine which client application participants preferred. Measures included task completion time (Section 7.6.1); number of display tap events when performing a task (Section 7.6.2); as well as task usability rating (Section 7.6.3). These quantitative results were used in conjunction with qualitative feedback in the form of open-ended responses from participants.

7.6.1 Duration

Task duration is a standard performance measure used in many areas of HCI research [122], and was recorded for each participant in the study. Although these times do not necessarily indicate how easy or difficult participants found a particular task, combined with other performance measures they allowed the researcher to perform a comparison between MobiSAM and MobiSAM Report.

Task timing was triggered by the first display tap event and continued until the participant had completed the task, and navigated back to the *Home* screen. A Wilcoxon signed-rank test was performed on the timings in order to determine whether there was any significant difference ($p<0.05$) in task durations between each client application. As the geometric mean has been shown to have less error and bias when compared to the sample median [188], it was used when finding the mean participant task completion times.

7.6.2 Display tap events

The number of display tap events made by each participant when performing a task was noted. All tap events, except those made when using the virtual keyboard, were recorded. As with task completion times, a Wilcoxon signed-rank test was performed on these values, indicating whether there was any statistically significant difference ($p<0.05$) between the number of tap events made when using MobiSAM and MobiSAM Report. The median was used when calculating the average number of tap events made by participants for each task. This decision was taken as Lewis and Sauro, in their paper titled *Average Task Times in Usability Tests: What to Report?*, only advocate using the geometric mean when calculating average task completion times, and not display tap events [188].

7.6.3 Usability ratings

As with the first usability evaluation, presented in Chapter 6, a value was assigned to each option in an attempt to quantify participant responses, with 'Very easy' representing a value of five, and 'Very difficult' representing a value of one. The overall usability rating for each task was then calculated by summing all participant responses for a participant task, then dividing the result by the total.

The resulting usability rating for each task was then compared across MobiSAM and MobiSAM Report in an attempt to determine which version participants preferred. As the summed Likert responses are considered interval data [14], the Wilcoxon signed-rank test could not be used to determine whether the differences in task usability ratings between the client applications was statistically significant.

7.7 Results

This section presents the results of the pre and post-intervention questionnaires, as well as the results of the four study tasks.

7.7.1 Demographics

As with the previously described studies (see Chapter 4 and Chapter 6), the first section of the pre-intervention questionnaire consisted of five questions and aimed to elicit basic demographic information from each participant.

Of the study sample, 60% of participants were female while 40% were male. Referring to participant age groups, the majority (70%) indicated that they were between 18-30 years; one participant (10%) indicated that they were between 31-45 years; while two participants (20%) indicated that they were over 60 years. 40% of participants indicated that they were Black African, while 60% were White. Examining participants' preferred language, 20% of participants specified isiXhosa, while 80% specified English. Furthermore, 20% of participants had obtained their matric, while 80% had graduated from university.

7.7.2 Use of mobile technology

The second section of the pre-intervention questionnaire solicited feedback relating to participants' use of mobile technology and related services.

Of the participant group, 90% owned a smartphone while a single participant owned a feature phone. 40% of participants had been using a mobile phone for 6-10 years, while the remaining 60% indicated that it had been over 10 years. The most prevalent brand of handset amongst participants was Samsung, representing 60% of the study sample. Nokia handsets were second most prevalent and represented 20% of the study sample, while the remaining two handsets were an Apple and a Huawei branded respectively.

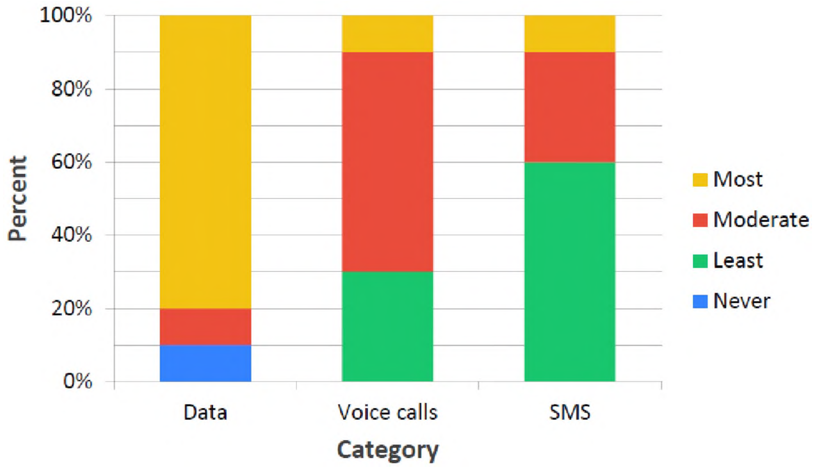


Figure 7.1: Frequency of use: data versus voice versus SMS ($n = 10$)

Participants were asked to rank the frequency in which they used the data, voice and SMS services available on their mobile phones. Responses are illustrated in Figure 7.1, with 80% of participants indicating that they used data most frequently; 60% indicating that they used voice calls second most frequently; while the same amount (60%) indicated that they used SMS least frequently. One participant never used data from their handset.

When asked to specify which messaging services participants used on their handsets, all participants indicated SMS; 90% of participants indicated WhatsApp; while a single participant used

BlackBerry Messenger (BBM). All participants indicated that they used these messaging services daily.

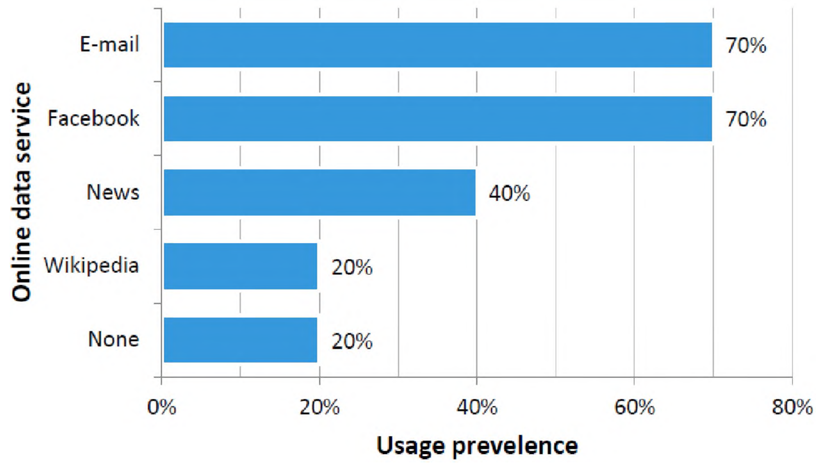


Figure 7.2: Use of online data services ($n = 10$)

Examining which online data services participants accessed from their handsets, 70% indicated both e-mail and Facebook; 40% indicated news; 20% indicated Wikipedia; while 20% indicated that they did not make use of any data services (see Figure 7.2). It should, however, be noted that of this 20% ($n = 2$), one participant (Participant_2B) did make use of WhatsApp, a messaging service which makes use of packet-data. This indicates that the participant possibly uses other non-specified data services which they do not realise are data services. Of the participants (80%) who made use of online data services, all indicated that they did so on a daily basis.

7.7.3 Task 1: Log in

The mean task completion time and median number of display tap events made by participants when completing Task 1 are summarised in Table 7.1. As each participant performed the task using MobiSAM and MobiSAM Report, both sets of results are provided. 80% of participants managed to log in to MobiSAM on their first attempt, while 20% managed to do so on their second attempt. All participants managed to log in to MobiSAM Report on their first attempt.

	MobiSAM	**MobiSAM Report**
Mean task completion time	18.08s	14.70s
Median tap events	2.5	2.5
Usability rating	94%	96%

Table 7.1: Duration, tap events and usability rating for Task 1

A Wilcoxon signed-rank test showed that there was no statistically significant difference ($p <$

0.05) in the Task 1 completion times between MobiSAM and MobiSAM Report ($Z = -0.614$, $p = 0.539$). In addition, there was no statistically significant difference ($p < 0.05$) in the number of tap events made ($Z = -0.962$, $p = 0.336$) between MobiSAM and MobiSAM Report. Results are illustrated in Table 7.2.

	Z-statistic	**p-value**
Task completion times	-0.614	0.539
Tap events	-0.962	0.336

Table 7.2: Wilcoxon signed-rank test for Task 1

The fact that there was no significant difference in any of the performance measures for Task 1 indicates that participants performed comparably well, irrespective of the client application used.

Q: "Overall, this task was?" When asked to rate the task-performance satisfaction using MobiSAM, 70% of participants selected 'Very easy' while 30% chose 'Easy', resulting in a usability rating of 94%. When asked to rate the same task performed using MobiSAM report, 80% of participants chose 'Very easy' while 20% chose 'Easy', equating to a slightly more favourable usability rating of 96%.

7.7.4 Task 2: Report water outage

Results for the second task are reflected in Table 7.3.

	MobiSAM	**MobiSAM Report**
Mean task completion time	37.60s	36.42s
Median tap events	6	4.5
Usability rating	100%	90%

Table 7.3: Duration, tap events and usability rating for Task 2

Performing a Wilcoxon signed-rank test indicated that there was no statistically significant difference ($p < 0.05$) in task completion times between MobiSAM and MobiSAM Report ($Z = -0.614$, $p = 0.539$). However, there was a significant difference ($p < 0.05$) in the number of tap events made when performing the task using MobiSAM and MobiSAM Report ($Z = -2.460$, $p = 0.014$).

	Z-statistic	**p-value**
Task completion times	-0.614	0.539
Tap events	-2.460	0.014

Table 7.4: Wilcoxon signed-rank test for Task 2

The results for each Wilcoxon signed-rank test are summarised in Table 7.4.

Q: "Overall, this task was?" All participants indicated that performing the task using MobiSAM was 'Very easy', resulting in a usability rating of 100%. The same task performed using MobiSAM Report scored less favourably, with 60% of participants rating it as 'Very easy'; 30% chose 'Easy'; while a single participant chose 'Neutral'. Due to the fact that MobiSAM Report presented participants with a single screen when reporting a problem, three participants (1A, 2B, 5A) had to be told they needed to scroll in order to reveal additional information contained further down the screen.

7.7.5 Task 3: Report water bubbling out of the ground

A summary of the results for Task 3 is presented in Table 7.5, and indicates that participants took slightly longer to perform the task using MobiSAM, while at the same time triggering more tap events. Performing a Wilcoxon signed-rank test on the completion times, however, showed that the difference between MobiSAM and MobiSAM Report was not statistically significant ($p < 0.05$).

	MobiSAM	**MobiSAM Report**
Mean task completion time	61.40s	54.10s
Median tap events	11.5	9.5
Usability rating	94%	88%

Table 7.5: Task 3: Duration, tap events and usability rating

There was however a statistically significant difference ($p < 0.05$) between the number of tap events made when performing the task using each client application, as summarised in Table 7.6. The median number of tap events made by participants performing Task 3 using MobiSAM was 11.5 compared to 9.5 when using MobiSAM Report.

	Z-statistic	**p-value**
Task completion times	-1.224	0.221
Tap events	-2.155	0.031

Table 7.6: Wilcoxon signed-rank test for Task 3

Q: "Overall, this task was?" 70% of participants indicated that Task 3 was 'Very easy' when performed using MobiSAM, while 30% indicated 'Easy'. Again, MobiSAM Report scored less favourably, with 40% of participants indicating 'Very easy' while 60% indicated 'Easy'. The resulting usability rating for each was 94% and 88% respectively.

Even though the questionnaire instructed participants to enter their address manually, Participant_1B performed this step correctly, but immediately afterwards chose to obtain their address using GPS, overwriting the correctly entered address. This oversight occurred when using both MobiSAM and MobiSAM Report. Participant_5A forgot to include a photo of the issue being reported when using MobiSAM Report. Finally, instead of selecting 'Water' from the MobiSAM Report *Home* screen, Participant_3B chose 'Roads' as they assumed the service delivery issue being reported related to road infrastructure.

7.7.6 Task 4: Change application language

Referring to Table 7.7, participants took slightly longer to perform Task 4 when using MobiSAM, however, this difference was shown not to be statistically significant ($p < 0.05$). Results to the Wilcoxon signed-rank test are summarised in Table 7.8.

	MobiSAM	**MobiSAM Report**
Mean task completion time	14.57s	9.15s
Median tap events	4	4
Usability rating	100%	96%

Table 7.7: Task 4: Duration, tap events and usability rating

An identical number of tap events were made when performing the task, irrespective of which application was used.

	Z-statistic	**p-value**
Task completion times	-1.790	0.074
Tap events	-0.552	0.581

Table 7.8: Wilcoxon signed-rank test for Task 4

Q: "Overall, this task was?" All participants indicated that performing Task 4 using MobiSAM was 'Very easy', resulting in a perfect usability rating. Participant_2A did however have difficulty navigating back to the MobiSAM *Home* screen due to unfamiliarity with the study handset. Once again, participants rated the task less favourably when performed using MobiSAM Report, with 80% of participants indicating the task was 'Very easy' while 20% indicated 'Easy'. The resulting usability rating was 96%.

7.7.7 Open-ended responses

After having completed the four tasks using both MobiSAM and MobiSAM Report, participants were asked a number of questions which aimed to assess the learnability, memorability and satisfaction of each client application.

Q1: "Which of the two apps did you find easiest to use?" Examining responses to this question, 80% ($n = 8$) of participants indicated that they found MobiSAM easiest to use; one participant indicated MobiSAM Report; while another indicated that they were both equally easy, although they preferred using MobiSAM.

When asked to elaborate on their choice, several different responses were given. Three participants (1A, 2A, 5B) indicated that they felt MobiSAM was better at 'guiding' them through the reporting process, while MobiSAM Report instead presented participants with a single screen. Participant_5B said: "[I] didn't like the scrolling up and down in MobiSAM Report, whereas MobiSAM used separate screens". Similarly, Participant_2A said that they "felt more secure using it [MobiSAM]" compared to MobiSAM Report. While Participant_1A stated: "It [MobiSAM] didn't leave room for me to pick a wrong option. It was more guiding than the other one [MobiSAM Report]". In addition, Participant_3B indicated that the instructions presented in MobiSAM were "more clear on what you need to do". Participant_4A felt that the MobiSAM *Login* screen was more intuitive, as it contained text specifying the purpose of each UI element. The participant also felt that the wording 'Keep me logged in' used in MobiSAM was more widely understood than 'Remember me'. Two participants (1B, 2B) indicated that previous learning may have made them feel more comfortable using MobiSAM, with Participant_1B mentioning that "MobiSAM Report was my learning stage", while Participant_2B felt that they were already partly familiar with MobiSAM due to previous exposure to the application.

Participant_3A, who indicated that they found MobiSAM Report easiest, said they "like[d] the use of pictures and graphics".

Q2: "Once familiar with the apps, do you feel you would be able to perform the tasks quicker using MobiSAM or MobiSAM Report?" Responding to the second question, 20% of participants indicated MobiSAM; 30% chose MobiSAM Report; while the remaining 50% of participants said that they would be able to perform the tasks equally quickly, regardless of the client application used.

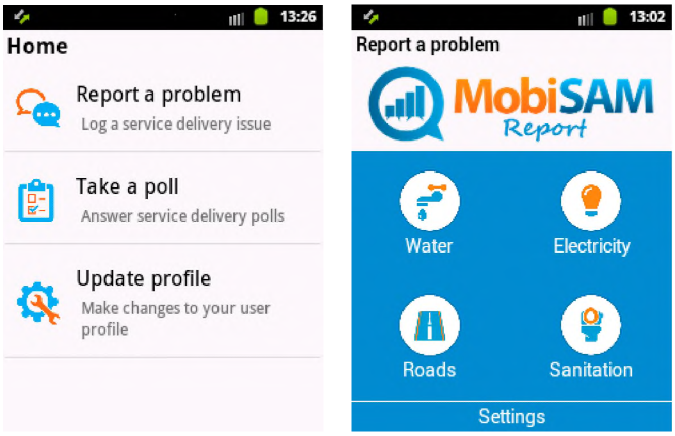
Of the participants who chose MobiSAM, Participant_4B said they felt they would be able to perform the task quicker because there was "less scrolling", therefore allowing the participant to "keep focusing on the screen". Participant_4A indicated that they preferred being shown "a clean page, and then another clean page, rather than scrolling" as it resulted in "less cognitive load when compared to MobiSAM Report". The participant also indicated that they found having a preview of the captured photo useful, a feature not included in MobiSAM Report.

Examining two of the responses from the three participants (2B, 3A, 3B) who chose MobiSAM Report, when asked the reason for their choice Participant_3A simply stated: "The interface is easier to use", while Participant_2B felt they would be able to perform tasks quicker using MobiSAM Report "because of the icons".

Finally, three responses were given by participants (1A, 1B, 2A) who indicated that they would be able to perform tasks equally quickly using both applications. Participant_2A said that although they felt they would be able to perform the tasks quicker using MobiSAM Report, they would still prefer to use MobiSAM. Both Participant_1A and Participant_1B felt that there would be no speed difference, as they had performed the tasks using both applications, which they were now familiar with. However, Participant_1B went on to conclude that: "For someone starting to use these apps, I think MobiSAM would be easiest".

Q3: "After having not used either of the apps for a while, which of the two would you feel most comfortable using again?" The majority of responses to Q3 were in favour of MobiSAM, with 60% of participants indicating that they would feel more comfortable using MobiSAM after having not reported an issue for a while. 30% of participants chose MobiSAM Report, while a single participant said that there was no difference between the two.

Describing their reasons for selecting MobiSAM, Participant_2A felt that it offered "way more guidance, whereas MobiSAM Report is more distracting". The participant also felt that only having three options on the *Home* screen was "less daunting" when compared to the grid layout used in MobiSAM Report. Participant_4B echoed this, saying: "[MobiSAM] doesn't present all the information at once", and therefore seemed less overwhelming. Participant_4A preferred the colour scheme used in MobiSAM saying: "The icons against the white background were easier on the eyes" while the icons used in MobiSAM Report "were more harsh" (see Figure 7.3).



(a) MobiSAM (b) MobiSAM Report

Figure 7.3: Client application *Home* screens

Examining the responses from participants (2B, 3A, 5B) who chose MobiSAM Report, Participant_2B said they felt more comfortable using it because there was less reading involved and a greater emphasis on the use of icons. Similarly, Participant_5B indicated that they preferred the grid layout of the *Home* screen icons compared to the menu-driven MobiSAM UI. In addition, Participant_3A felt that MobiSAM Report was "quicker to navigate through and easier to use".

Finally, Participant_1B said that they would feel equally comfortable using either client application as they were now familiar having used both.

Q4: "Do you feel more likely to make mistakes with MobiSAM or MobiSAM Report?" The fourth question presented to participants received mixed responses, with 20% indicating MobiSAM; 40% indicating MobiSAM Report; while the remaining 40% said that they were both the same.

Participant_3A felt that they were more likely to make mistakes using MobiSAM, as it placed less emphasis on the use of pictures to help guide the user. Participant_2B felt that because MobiSAM Report presented everything on a single screen, they were less likely to make mistakes.

Conversely, Participant_2A said that they were more likely to make mistakes using MobiSAM Report due to the fact that everything was presented on one screen, hence requiring more scrolling. Similarly, two participants (1A, 3B) felt that MobiSAM Report provided less guidance when compared to MobiSAM, with Participant_1A stating: "MobiSAM doesn't leave me any room to make mistakes. It leads me through everything". Furthermore, Participant_4A felt that MobiSAM Report was "more verbose", potentially leading to an increased likelihood of mistakes.

Finally, responses from the final group of participants (1B, 4B, 5A, 5B) indicated that they felt unlikely to make mistakes, regardless of the application used. Participant_1B said: "I don't see myself making mistakes", while Participant_5A indicated that if they did make a mistake, they felt it would be straightforward to backtrack and correct it.

Q5: "Which of the overall app designs do you find most pleasing and intuitive?" Examining answers to question five, 40% of participants found MobiSAM's design most pleasing and intuitive; 50% chose MobiSAM Report; while one participant indicated no preference.

Of the four participants (2A, 3B, 4A, 5A) who chose MobiSAM, two provided comments. Similar to previous comments, Participant_2A stated that the menu-driven UI style employed by MobiSAM helped guide them, letting them know that they were performing tasks correctly. Participant_3B also felt that MobiSAM helped 'walk' them through the steps, therefore making the client application more pleasing and intuitive to use.

The most frequently cited reason for participants having chosen MobiSAM Report was due to its use of iconography, with Participant_1B stating: "MobiSAM Report is a bit easier because it has pictures. Especially for somebody who's not familiar with phones". Similarly, Participant_3A liked the use of icons, Participant_4B preferred the grid layout of the *Home* screen, while Participant_5B simply stated: "It [MobiSAM Report] looks better". In addition, Participant_2B said MobiSAM Report was "very clear because of its use of icons rather than words".

The single participant (Participant_1A) who indicated no preference said that they felt that MobiSAM was more pleasing due to the guidance it offered users, while MobiSAM Report was more intuitive as the icons were bigger and therefore allowed for easier interaction.

Q6: "Do either of the apps better provide the features that you need?" Responses to question six indicated that 30% of participants felt MobiSAM better provides the features they need; 10% chose MobiSAM Report; while the remaining 60% had no preference: both applications provided the same functionality equally well.

Only three participants (1A, 2B, 4A) elaborated upon their choices. Participant_1A said they liked the fact that MobiSAM provides an 'Other' option, allowing them to report "more serious [service delivery] problems" should the need arise. In addition, they felt that the 'Update profile' menu option was more personal than 'Settings' option used in MobiSAM Report. Participant_4A preferred MobiSAM as it provided an icon next to the 'Update profile' menu option, whereas MobiSAM Report did not include an icon alongside 'Settings'.

Finally, Participant_2B indicated their preference for MobiSAM Report, simply stating it "provides better access in general".

Q7: "What are your likes, dislikes and recommendations for each of the apps?"

MobiSAM Four participants (1A, 2A, 4A, 4B) provided feedback and comments regarding the MobiSAM client application.

Examining aspects of the application which participants liked, Participant_4A said that MobiSAM was “clean” and “more simplistic”, and that they “preferred [the use of] less content”. Participant_1A liked the ‘Other’ menu option (which allows users to report service delivery issues not related to water, roads, electricity or sanitation) and felt that MobiSAM was “a little more guiding”. They went on to highlight that ‘Update profile’ seemed more personal than ‘Settings’, as provided in MobiSAM Report. Finally, Participant_2A appreciated the fact that they were able to “say something to the municipality without a huge hassle”, and that it “felt like I [had] accomplished something and it didn’t take much of my time”.

Participant_4B suggested that some of the options contained within the *Update profile* screen should instead be incorporated within a separate *Settings* screen.

MobiSAM Report Eight participants (1A, 1B, 3A, 3B, 4A, 4B, 5A, 5B) provided feedback and comments surrounding the MobiSAM Report client application.

Referring to the aspects of the client application which participants liked, Participant_1A, Participant_1B, Participant_3A and Participant_4B all indicated that they favoured the use of *Home* screen icons. Participant_1B went on to provide an anecdote highlighting the importance of pictures and how they aid understanding, stating: “When I was young I would read books but wasn’t interested [in books] without pictures. I want to see what I’m reading about”. Similarly, Participant_3A said: “[I] like that it [MobiSAM Report] was more graphical, whereas MobiSAM was more textual” while Participant_1A liked “the big buttons on the *Home* screen”. Participant_4B further added that they liked the grid layout of the *Home* screen.

Conversely, Participant_4A said that they found the colour of MobiSAM Report’s *Home* screen “overwhelming”, and the fact that they had to scroll through a single screen when reporting a service delivery issue resulted in “cognitive overload”. Similarly, Participant_1A felt that MobiSAM Report “doesn’t present content in manageable sizes”, instead showing multiple screens worth of information at once.

Finally, Participant_5B felt that MobiSAM Report should include a preview of the captured image, similar to MobiSAM.

7.8 Analysis and discussion

Usability is defined by Jacob Nielson as “a quality attribute that assesses how easy user interfaces are to use” [156, p.1], and consists of five quality components: learnability, efficiency, memorability, errors and satisfaction [156]. In an attempt to determine which client application participants preferred, three of these quality components were inspected, namely: learnability, memorability and satisfaction.

7.8.1 Learnability

The first quality component, learnability, is described as the ease with which users are able to accomplish basic tasks the first time they encounter a UI [156].

The first question ("Which of the two apps did you find easiest to use?") aimed to determine how MobiSAM and MobiSAM Report compared according to this quality component, with 80% of participants indicating that they found MobiSAM easier to use. MobiSAM's notable preference amongst participants is likely due to its menu-driven navigation style, as highlighted by a comment from Participant_1A stating: "It [MobiSAM] didn't leave room for me to pick a wrong option, [as it] was more guiding than the other one [MobiSAM Report]". Participant_3B stated that MobiSAM seemed more learnable as its instructions were clearer, and better described what needed doing. In addition, Participant_4A felt that MobiSAM's *Login* screen was more descriptive, and hence more learnable, as it contained text describing each UI component, while MobiSAM Report used a more minimalistic style. The participant also indicated that the description 'Keep me logged in' was better understood than 'Remember me'. Interestingly, Participant_1B felt that having first performed the study using MobiSAM Report helped them feel more at ease when using MobiSAM, stating: "MobiSAM Report was my learning stage". This suggests that the learnability of MobiSAM Report is possibly transferable to MobiSAM. The same participant went on to state that they "think MobiSAM would be easiest" for someone first using the client applications, further highlighting its learnability.

While MobiSAM follows a menu-driven navigation style, guiding participants through the reporting process by presenting them with several sequentially ordered screens, MobiSAM Report presents participants with a single screen which they are required to scroll through, filling in the necessary details. Although this UI design requires fewer tap events for a given task (see Tables 7.3 and 7.5), participants indicated that they felt less secure using it (due to the lack of guidance), which may have consequently hampered its learnability. This was typified by a comment from Participant_5, who stated that they "didn't like the scrolling up and down in MobiSAM Report, whereas MobiSAM used separate screens". Observations made by the researcher also indicated that three participants (1A, 2B, 5A) were initially unaware that the *Reporting* screen of MobiSAM Report was indeed scrollable. This left them unsure about how to proceed with the task. The single participant (3A) who found MobiSAM Report easiest to use cited its use of pictures and graphics on the *Home* screen for their preference.

Interestingly, inspecting the task performance measures indicates that there was no statistically significant difference ($p < 0.05$) in the degree of learnability of each client application. Examining the completion duration of Task 1 (the first task all participants were assigned) reveals that, while not statistically significant, participants on average took slightly longer when using MobiSAM (18.08s) compared to MobiSAM Report (14.70s). Furthermore, participants gave MobiSAM a slightly less favourable usability rating (94%) compared to MobiSAM Report (96%). However, the median number of tap events made when completing this task was identical between both client applications. As participants progressed through the tasks, however, their usability rating for MobiSAM improved, further suggesting that they were becoming acquainted with the client application. It is important to note that all 10 participants had previously used MobiSAM during the first usability evaluation (see Chapter 6), which may have influenced these results.

Drawing conclusions from participant responses and task performance measures, it is reasonable to deduce that MobiSAM provides a more easily learnt UI when compared to MobiSAM Report. This is primarily due to the guidance it provides users by 'walking' them through the reporting process, describing each step along the way.

7.8.2 Memorability

Memorability defines how easily users are able to re-establish their proficiency at using a UI after not having interacted with it for a period of time [156]. The third question presented to participants ("After having not used either of the apps for a while, which of the two would you feel most comfortable using again?") attempted to gauge how MobiSAM and MobiSAM Report compared according to this quality component.

The majority (60%) of participants (1A, 2A, 3B, 4A, 4B, 5A) chose MobiSAM over MobiSAM Report, indicating that the majority of participants thought it to be easier reacquainting themselves with MobiSAM. As all 10 study participants had previously performed tasks using MobiSAM during the first usability evaluation (see Chapter 6), the majority preference towards MobiSAM may have been a side-effect of this. Once again, participant responses indicated a preference for the menu-driven navigation style employed by MobiSAM, with Participant_2A stating: "[MobiSAM provides] way more guidance whereas MobiSAM Report is more distracting", a characteristic likely to reduce the memorability of MobiSAM Report. In addition, Participant_2B noted that MobiSAM felt "partly familiar" due to prior exposure during the first usability evaluation. Furthermore, it is worth noting that while three participants (4A, 5A, 5B) of the sample group were unable to complete all their assigned tasks during the first usability evaluation (see Chapter 6), these same participants managed to complete all assigned tasks during this study. Possibly highlighting the memorability of the client applications.

Examining participant responses reiterated the fact that they preferred being guided through the reporting process, thereby helping them re-establish their proficiency at using the client application. This preference was echoed by Participant_4B who stated: "[MobiSAM] doesn't present all the information at once", and is therefore seen as being less overwhelming. Participant_4A stated that they felt MobiSAM resulted in "less cognitive load" when performing tasks, further highlighting the manner in which the client application may enhance memorability.

As previously described, MobiSAM Report presented participants with a single *Reporting* screen, requiring them to scroll down, filling in information as required. Although this allowed for tasks to be performed in fewer steps, this increase in efficiency may have been at the cost of reduced memorability. This was echoed by Participant_1A who stated that MobiSAM Report "doesn't present content in manageable sizes", instead showing multiple screens worth of information on a single, scrollable screen. The 30% of participants (2B, 3A, 5B) who chose MobiSAM Report preferred its use of icons and the fact that it required fewer steps when performing a task. Participant_3A highlighted this by stating: "Quicker to navigate through and easier to use". The distinctive use of icons arranged in a 2x2 grid on the *Home* screen may have improved the memorability of MobiSAM Report for this group of participants. However, these benefits may have been partly negated by the lack of guidance provided on the *Reporting* screen.

Interestingly, although Participant_2B indicated that previous exposure to MobiSAM may have improved their task performance, they still indicated a preference toward MobiSAM Report when responding to Q3, stating that MobiSAM Report required less reading and placed greater emphasis on the use of icons.

Drawing together participant responses and task performance measures, although not statistically significant ($p < 0.05$), it seems as if the majority of participants found MobiSAM to be more memorable. This improved memorability is likely due to the manageable sizes in which MobiSAM presents content to users, as well as the step-by-step guidance provided to users when reporting a service delivery problem.

7.8.3 Satisfaction

As the name suggests, satisfaction aims to determine how pleasant a UI design is to use [156]. The fifth question ("Which of the overall app designs do you find most pleasing and intuitive?") attempted to gauge participant preference for each client application according to this quality component.

In contrast to previous questions, 50% of participants (1B, 2B, 3A, 4B, 5B) indicated a preference for MobiSAM Report, stating that the extensive use of icons and graphics helped improve its perceived satisfaction rating. However, this improvement is likely to have been diminished due to the excessive use of text on the *Reporting* screen. Participant_4A highlighted this, stating that MobiSAM Report was "more verbose" leading to "cognitive overload".

While MobiSAM, on the other hand, employed a more subtle use of icons, 40% of participants (2A, 3B, 4A, 5A) felt that its menu-driven navigation style helped improve its intuitiveness. This was highlighted by a response from Participant_2A, stating: "The steps guide the user and let them know they are doing things right". Participant_1A provided two different responses to the question, indicating that they found MobiSAM more intuitive, however, MobiSAM Report was more pleasing to use. In hindsight, question five should have been split into two separate questions as some participants are likely to have found one application more pleasing to use, while the other more intuitive.

All five participants who showed preference towards MobiSAM Report mentioned its effective use of icons and graphics as the primary reason for their preference. Interestingly, four (1B, 2B, 4B, 5B) out of five of these participants had performed the study tasks first using MobiSAM Report, followed by MobiSAM. While three (2A, 4A, 5A) out of the four participants who indicated a preference towards MobiSAM had first performed the study tasks using it, followed by MobiSAM Report. This suggests that the order in which client applications were presented to participants may have affected their perceived level of satisfaction.

Although participants made a statistically significant ($p < 0.05$), lower number of display tap events when performing Tasks 2 and 3 using MobiSAM Report, there was no mention of this having improved participant satisfaction levels. Instead, participants indicated the grid layout of MobiSAM Report, combined with its reliance on icons and graphics, to be the primary reason for their preference. As such, it may be worth combining the different approaches taken by MobiSAM and MobiSAM Report in a future release of the client application. In this way, the improved satisfaction levels gained from the grid layout may be coupled with the enhanced learnability and memorability of MobiSAM's menu-driven reporting process, potentially leading to a more usable client application.

7.9 Summary

This chapter presented the second and final usability evaluation which aimed to determine which client application participants preferred: MobiSAM or MobiSAM Report. The study saw participants perform four tasks using each client application which aimed to examine a broad range of their functionality. Comparisons between MobiSAM and MobiSAM Report were drawn using three of Nielson's five quality components (learnability, memorability and satisfaction) in an attempt to determine which client application was most preferred by participants. Results from the study indicated that although participants were able to perform the equivalent tasks in slightly less time (and with fewer display presses) when using MobiSAM Report, the improved guidance and assurance offered by MobiSAM flagged it as the preferred client application amongst participants.

The findings from this chapter have been used to derive a specifications list and possible design alternations for the next version of the MobiSAM client application, a brief outline of which is presented in Chapter 8. The next chapter provides a discussion and conclusion of the research undertaken in this thesis, revisiting each of the goals defined in Chapter 1, as well as presenting possible avenues for future work.

Chapter 8

Discussion and conclusion

8.1 Introduction

This chapter presents an overview and discussion of the research conducted as part of this thesis. First, the four research goals are revisited, revealing the contributions made by each (Section 8.2). A variety of possible future work is then described (Section 8.3), aimed at both evaluating and improving the client applications, as well as the MobiSAM service. Finally, a brief collection of concluding thoughts is provided (Section 8.4).

8.2 Research goals revisited

Four distinct research goals were described in Chapter 1. Each of these goals will now be revisited in order to determine how satisfactorily they were met, as well as the contributions they made to the relevant fields.

8.2.1 Goal 1: Literature review

The first research goal aimed to enable a broader understanding of the problem space. The literature review investigated the current state of the art in both e- and m-government arenas (see Section 2.2), as well as ICT-facilitated civic engagement projects, both locally and abroad (see Sections 2.3 and 2.4). A variety of applicable technologies were also detailed, including SMS, SMS gateways, USSD, REST and JSON (see Section 2.5). In addition, several different visualisation principles, techniques and tools were discussed (see Section 2.8). Particular attention was paid to the differences between traditional and mobile visualisations.

Contributions The completion of this goal provided a solid foundation upon which the remainder of the research was built. Additionally, it may serve as a useful resource to researchers implementing similar ICT-facilitated civic engagement projects.

8.2.2 Goal 2: Retail outlet survey and baseline study

The second goal was performed in order to determine the best-selling mobile handsets (and hence platforms) within the Grahamstown area (see Section 3.5), together with a comprehensive baseline study (see Chapter 4). The baseline study helped the researcher to gain an understanding of how Grahamstown residents currently use mobile devices and services, as well as participate with local government around the area of service delivery. Feedback from the informal retail outlet survey indicated Java ME and BlackBerry OS to be the two best-selling mobile platforms. As such, these mobile platforms were the chosen targets for the high-fidelity prototype client applications (see Appendix A.2). Results obtained from the baseline study confirmed these preliminary findings, as well as highlighting the prevalence of Android, iOS and, to a lesser extent, Windows Phone devices (see Section 4.6.3).

The study also indicated that data services are used either more frequently or the same amount as voice and SMS services, across all genders and race groups within Grahamstown (see Section 4.6.4). SMS was shown to be the most popular messaging service. In addition, almost two-thirds of respondents own handsets capable of installing the messaging application WhatsApp. This suggests that, as the cross-platform MobiSAM client application targets the identical platforms, a similar proportion of handsets could be reached by the installable client application. The majority of respondents also indicated that they believe that they can make either "some" or "a great deal" of difference in helping improve instances of poor service delivery (see Section 4.6.1). It is therefore plausible that these respondents will seek avenues of complaint, such as those provided by MobiSAM, as opposed to being apathetic to their current situation.

Contributions Combined, the results from the previously mentioned studies have contributed by providing the researcher with an accurate understanding of the current mobile environment within Grahamstown. Additionally, the extent to which residents currently engage with their local municipality, as well as the methods used were revealed. So too were their perceptions about being able to positively affect rates of service delivery within their town. In accordance with the UCD approach (see Section 3.2), this information was vitally important in informing the design and implementation of the client applications, as well as the SMS gateway (see Appendix A.2, as well as Sections 5.6, 5.7 and 5.8).

8.2.3 Goal 3: Develop MobiSAM client applications and SMS gateway

The third goal saw the development of two prototype client applications (see Appendix A.2), serving as an alternative to the MobiSAM website (see Section 5.3.1). Each acted as a basic proof of concept, and targeted the Java ME and BlackBerry OS platforms respectively. The development and maintenance of each client application revealed the effort required to target multiple mobile operating systems using platform-specific SDKs, thus emphasising the need for an alternative approach. Prototype client applications were developed during Iteration 1 of the life-cycle model (see Section 3.4.1).

The next portion of this goal attempted to identify the most suitable way of targeting Grahamstown's heterogeneous mobile platform environment. A survey of all cross-platform mobile frameworks was first undertaken, summarising information such as supported platforms, implementation language, as well as whether the framework leveraged web technologies or executed applications natively (see Section 2.7.2). Three frameworks were then short-listed (Codename One, J2ME Polish and NeoMAD) due to their supported platforms (Android, BlackBerry OS, iOS, Java ME and Windows Phone), implementation language (Java) and native application builds, as opposed to using web technologies (see Section 5.5). NeoMAD was excluded due to its high subscription fees. A demonstration application was then developed as a means to assess the remaining two frameworks. Results singled out Codename One as the most fitting solution.

A cross-platform client application was then developed, allowing access to the MobiSAM service, and supporting five different mobile platforms (see Section 5.6). The client application allows users to navigate different service delivery categories, answering polls of interest. Depending on the poll type, the results of answered polls are summarised in a numerical table, as well as visualised using a pie and bar chart. Additionally, if the client application is installed on a Android, iOS or Windows Phone device, interactive chart visualisations were provided, potentially leading to a better understanding of poll results (see Section 5.6.3). Development of the cross-platform client application was undertaken during Iteration 2 of the life-cycle model (see Section 3.4.2).

Following feedback from the usability evaluation (see Chapter 6), the cross-platform client application was updated, in an attempt to address the identified usability concerns and visualisation shortcomings (see Section 5.7). Modifications included: allowing the reporting of specific service delivery problems; the removal of ambiguous screen names, as well as the bar chart visualisation; and the redesign of the *Visualisation* screen, so as to offer improved affordance (see Section 5.7.1). In addition, participants are able to update their current suburb from within the client application. Furthermore, summarised poll results are relevant to the suburb within which the user currently resides, as opposed to the entire of Grahamstown. These modifications were implemented during Iteration 3 of the life-cycle model (see Section 3.4.3).

An additional cross-platform client application was developed in response to suggestions from a project stakeholder. The stakeholder felt that the focus of the existing cross-platform client application needed to be realigned. Instead of enabling users to answer service delivery polls and subsequently presenting them with visualised poll results, the project stakeholder indicated that it was more important to streamline the process of reporting *specific* service delivery complaints relating to water, roads, electricity and sanitation. As a result, a second cross-platform client application (MobiSAM Report) was developed according to their requirements (see Section 5.8). MobiSAM Report differs from the initial cross-platform client application in several ways (see Section 5.8.1). Firstly, it does not provide users with the ability to answer service delivery polls. Consequently, all forms of visualisation are absent. Secondly, instead of implementing a menu-driven UI, MobiSAM Report provides the user with a 2x2 icon grid, with each icon representing a different basic municipal service (see Figure 5.13). Selecting an icon takes the user to the *Reporting* screen, where they are prompted to select the issue they wish to report, from a list of predefined reports (see Figure 5.14b). Users are also able to capture and attach images to service delivery reports, as well as provide their current position, obtained using their handset's built-in GPS (when supported). These features were included in an attempt to further streamline the reporting process, as well as allow the municipality to gain greater insight into reported problems.

Finally, in an attempt to lower the barrier of entry to using the MobiSAM service, an SMS gateway was developed allowing residents to report service delivery complaints via traditional SMS (see Section A.6). The importance of providing SMS support was emphasised by results from the baseline study (see Chapter 4), which revealed SMS to be the most frequently used messaging service by Grahamstown residents. Furthermore, up to one-third of respondents did not have handsets capable of installing applications. Makana Municipality also stressed the need for SMS-based reporting.

Contributions The research undertaken for this goal contributes in several different ways. Firstly, it reveals that the development of platform-specific mobile applications is a viable alternative to the mobile-optimised MobiSAM website. However, a large number of resources are needed to target each platform natively. It then outlines a suitable method of targeting multiple mobile operating systems from a single, common codebase. It also reveals that, of the multitude of cross-platform mobile frameworks available, CN1 is able to provide native support for the greatest number of platforms (Android, BlackBerry OS, iOS, Java ME and Windows Phone), while at the same time extending its reach to entry-level Java ME-enabled feature phones. Secondly, the investigation shows that, within this specific use case, an application developed using a cross-platform framework is able to perform equally well when compared to those developed natively. A variety of suitable visualisation techniques were sought, allowing chart visualisations to automatically adapt to different display sizes and resolutions, as well as provide interactivity. The development of a cross-platform application capable of executing on five different mobile operating systems, while adhering to the UI and UX guidelines of each, was then detailed. This provides information to researchers aiming to target several mobile platforms using a single codebase. In addition, the implementation of novel features such as capturing and attaching photographs, as well as geotagging were presented. Finally, the process of developing an SMS gateway for use within the context of a civic engagement project was described.

8.2.4 Goal 4: Evaluate MobiSAM client applications

The fourth and final goal began with a brief informal usability evaluation involving two expert users. The evaluation highlighted several minor usability concerns within the high-fidelity prototype client applications (see Section 3.6). The expert users also reiterated the drawbacks associated with developing several client applications natively, suggesting that a suitable cross-platform mobile framework be sought instead (see Section 5.5.1).

A thorough usability evaluation using the cross-platform MobiSAM client application (see Chapter 6) formed the next part of this goal. The evaluation, consisting of 30 participants, had two primary objectives. Firstly, it aimed to assess how intuitively participants were able to navigate within the client application (see Section 6.8.3). Secondly, the study aimed to determine how effective different visualisation types were at conveying meaning to participants (see Section 6.8.4). In addition, the role which chart interactivity played in helping participants to better understand these visualised results was investigated. A several miscellaneous usability concerns and functional shortcomings were also revealed. Results from the study were used to provide an updated set of requirements (and hence specifications) for the cross-platform client application (see Section 5.7.1).

The final portion of this goal saw a usability evaluation be performed, using the updated cross-platform MobiSAM client application and newly developed MobiSAM Report (see Chapter 7). A mixed-method comparative usability evaluation involving 10 participants was conducted, providing both qualitative and quantitative feedback. The client applications were assessed side-by-side according to the usability goals learnability, memorability and satisfaction. Each participant performed four predefined tasks using both client applications, in alternating order (see Section 7.5). Feedback and results from the evaluation highlighted that the MobiSAM client application was preferred in terms of learnability and memorability (see Sections 7.8.1 and 7.8.2), while MobiSAM Report received higher satisfaction ratings (see Section 7.8.3). Several minor usability concerns were also highlighted within each of the client applications.

Contributions The completion of this research goal contributes by highlighting the impracticalities involved in developing multiple, functionally identical mobile applications natively. The ease at which a menu-driven UI is navigated by users, as well as the potential pitfalls of implementing such UI styles across heterogeneous platforms (ambiguous menu names, ignored screen titles, save confirmations). In addition, the effectiveness of different chart types was shown, as well as whether chart interactivity helped aid the understanding of poll results. The comparative usability evaluation indicates that a menu-driven UI navigation style better supports the learnability and memorability of the client application. While the icon-driven, grid layout was found to be more visually appealing, leading to higher levels of participant satisfaction. Reasons for participant preferences are also specified, potentially assisting other researchers create more robust and usable cross-platform mobile applications. The study also highlighted that the updated MobiSAM client application is the most suitable version to roll-out to the public, during future pilot studies.

8.3 Future work

There are several extensions which could be made to the research presented in this thesis. An overview of several of these are presented next. It is important to note that while all suggestions relate to the MobiSAM client applications, some rely on additional routes being added to the MobiSAM REST API (see Appendix C).

8.3.1 Combine client applications

Chapter 7 presented a comparative usability evaluation between the updated cross-platform MobiSAM client application and newly developed MobiSAM Report. While results and feedback from the evaluation indicated MobiSAM to be the preferred client application, half of the participants highlighted their preference for the icon-based, grid layout used by MobiSAM Report, resulting in higher levels of satisfaction. In an attempt to draw upon the strengths of both client applications, it is proposed that the favourable aspects of each could be combined into a single client application. Accordingly, the application could leverage an icon-based *Home* screen, while implementing a menu-driven step-by-step procedure for the reporting of service delivery problems. This would potentially enable more efficient reporting of service delivery problems, while at the same time allow for the answering of polls.

8.3.2 Pilot study with Makana Municipality

The MobiSAM service is currently (June 2015) being piloted with Makana Municipality. While residents are able to answer service delivery polls and report specific service delivery complaints via the MobiSAM website, the mobile client application and SMS gateway have yet to be rolled-out to the public. As such, the extent to which mobile phones are able to increase meaningful citizen participation with local government has yet to be evaluated. Conducting a thorough pilot study may also reveal additional usability concerns. These concerns are likely to become apparent due to the client applications being used by many residents in a real-world environment. A scenario which the usability evaluations were not able to directly emulate.

8.3.3 Ticketing system

Currently, there is no way in which a user is able to track the progress of their reported service delivery complaint from within the client applications. While the MobiSAM website provides a text enter area below each reported problem, where dialogue may take place, this functionality is absent from the client applications. Instead, after successfully submitting a service delivery complaint via the client application, the user is presented with a screen informing them that their report has been received and will be attended to timeously (see Figure 5.10d). In order to provide improved feedback to residents, it is proposed that a ticketing system be incorporated within the MobiSAM service. After reporting a service delivery complaint, the user could receive a report reference number, which they are then able to enter into the client application and/or website to obtain the current status of the complaint (received/in progress/resolved). In addition, a *Report history* screen could be implemented within the client applications, providing a list of all reports sent by the user, as well as their current status. This would help keep residents better informed, potentially leading to improved user satisfaction. Additional REST routes would be needed to enable this functionality.

8.3.4 Miscellaneous

Numerous smaller extensions which could be implemented within the client applications exist. A selection of these are presented next.

In app registration At present, residents are only able to register for the MobiSAM service via the website (see Section 5.2.1). Allowing a user to register within the client application would streamline this process, possibly improving the overall usability of the service. A new REST route is required to enable this functionality.

Time specific results Service delivery poll results are currently shown for the entire duration since the poll was first created. For example, if a user answers a poll on 25 February 2015, which was created on 5 February 2015, the returned results will provide a summary of the responses received over the duration of twenty days. While it is possible for a MobiSAM administrator to create new polls at regular, predefined intervals (i.e. once per week) and thus sidestep the issue, alternative methods are needed. One possibility would be to allow a user to view the results for a particular poll, based on a given time frame. After responding to a poll, the user could be presented with the *Results* screen, allowing them to select the time frame for which they would like to view the results: last 6 hours, last 24 hours, last week or the last month. This would allow residents to gain a better understanding of who else is experiencing a similar service delivery problem within their suburb. Additional REST routes are required to be able to implement this feature.

News section Including a simple news screen or section into the client application may help keep residents better informed. The municipality could post relevant information relating to basic service outages (planned or unplanned), provide warnings to residents (i.e. do not drink the tap water), amongst other things. The news could be either suburb-specific or Grahamstown wide. As newer mobile operating systems (Android, iOS and Windows Phone) support notifications, alerting a user to changes or updates within an application, this technique could be used to inform residents about news updates. Modifications would need to be made to the client applications, web-interface and REST API in order to implement this feature.

Heat/choropleth maps In addition to providing the user with a numerical table and a pie chart visualisation summarising poll results, either heat or choropleth maps could be included within the client application. The maps could then overlay different colour shadings for each suburb, based on the poll responses from each area, providing residents with a bird's-eye view of the current state of a particular municipal service. For example, a poll asking residents whether they currently have running water.

Get suburb from GPS Currently, the user is able to update their current suburb from within the client application's *Update profile/Settings* screen (see Figures 5.11e and 5.15). Here, they select their suburb from a drop-down list of predefined suburbs. This process could be improved by obtaining the user's position using their handset's GPS, and then translating this position into the corresponding suburb. As not all residents are aware in which suburb they reside, this feature would help avoid unnecessary confusion, ideally leading to an improved user experience.

Chat function Highlighted by the baseline study, messaging applications such as WhatsApp, as well as traditional SMS are the most frequently used services by mobile subscribers within Grahamstown (see Chapter 4). As such, it is proposed that this familiarity with messaging services may be leveraged to create a direct, real-time line of communication between residents and a municipal representative. To achieve this, a simple chat function could be implemented within the client application, allowing text messages to be exchanged between a user and support staff at the municipality. The potential benefits of this feature are twofold. Firstly, sending a text-based message is less expensive than voice calling the municipality, lowering costs for the user. Secondly, instead of physically visiting the municipality to speak to a representative (a method previously used by 47% of all baseline study respondents), the user is able to avoid the associated travel expenses by instead conducting their conversation via text messaging.

Reservoir level indicator Grahamstown's water is stored in several reservoirs, located at high-lying areas across the town. At times there are problems associated with pumping water from the source into these reservoirs, leading to demand outstripping supply. This results in unplanned water outages. By providing a reservoir level indicator within the client application, the user would be able to potentially pre-empt reservoir related water outages. In addition, the user will be able to adapt their usage pattern, thus helping stabilise the water supply. Implementing such a feature, however, would require special hardware to be installed at each of the Municipality's reservoirs, which then would automatically relay relevant data to the MobiSAM backend server at regular intervals. Alternatively, a municipal employee could be assigned to visit each of the reservoirs once per day, logging this data manually.

8.4 Concluding thoughts

The research presented in this thesis has helped established that the pervasive nature of mobile phones positions installable applications as viable tools for potentially increasing citizen engagement with local governments. Provided that all potential barriers of entry are identified and thus exceeded, including platform support; language constraints; and associated usage costs. While mobile applications are not likely to replace the existing methods of citizen engagement, they have the potential to provide an additional channel through which residents can hold their municipality to account. However, a pilot study of the client applications will be required before assessing their ability to increase meaningful citizen participation, and thus positively affecting levels of service delivery.

8.5 Summary

This chapter revisited the research goals outlined within Chapter 1, discussing how satisfactorily each goal was met, as well as the subsequent contributions it made. Several areas where the research may be extended were also presented, followed by the researcher's concluding thoughts.

References

[1] D. Aamoth. (2014, 18 August) *First Smartphone Turns 20: Fun Facts About Simon.* Time, Inc. [Online]. Available: http://time.com/3137005/first-smartphone-ibm-simon/ [Accessed: 20/09/2014]

[2] C. Abras, D. Maloney-Krichmar, and J. Preece, "User-centered design," *Bainbridge Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, vol. 37, no. 4, pp. 445–56, 2004.

[3] African Centre for Citizenship and Democracy. (2013) *African Centre for Citizenship and Democracy Citizen Scorecard.* [Online]. Available: www.accede.co.za/downloads/citizen%20scorecard%20DELFT%20web.pdf [Accessed: 03/12/2014]

[4] P. Alexander. (2012, April) *A massive rebellion of the poor.* Mail & Guardian. [Online]. Available: http://mg.co.za/article/2012-04-13-a-massive-rebellion-of-the-poor [Accessed: 24/01/2015]

[5] S. Almog, C. Fishbein, and E. Coolman. (2013, December) *Codename One Developer Guide.* Codename One. [Online]. Available: www.codenameone.com/developer-guide.html [Accessed: 01/11/2014]

[6] A. Anttiroiko and M. Mälkiä, *Encyclopedia of digital government.* IGI Global, 2007.

[7] P. Ashley, H. Hinton, and M. Vandenwauver, "Wired versus wireless security: The Internet, WAP and iMode for e-commerce," in *Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual.* IEEE, 2001, pp. 296–306.

[8] R. Atun and S. Sittampalam, "The role of mobile phones in increasing accessibility and efficiency in healthcare," *Berkshire, UK: Vodafone Group Plc*, 2006.

[9] P. K. Bardhan and D. Mookherjee, *Corruption and decentralization of infrastructure delivery in developing countries.* Boston University, Institute for Economic Development, 2000.

[10] J. Barnard, "Government IT's sad state of affairs," *Brainstorm Magazine*, vol. 11(8), pp. 18–21, 2012.

[11] A. Battestini, V. Setlur, and T. Sohn, "A large scale study of text-messaging use," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services.* ACM, 2010, pp. 229–238.

[12] BBC News. (2007) *Texts monitor Nigerian elections.* [Online]. Available: http://news.bbc.co.uk/2/hi/technology/6570919.stm [Accessed: 22/10/2014]

[13] J. Bertin, "Semiology of graphics: diagrams, networks, maps," 1983.

[14] D. Bertram. (2013) *Likert Scales are the meaning of life.* Faculty of Mathematics, University of Belgrade. [Online]. Available: http://poincare.matf.bg.ac.rs/~kristina/topic-dane-likert.pdf [Accessed: 04/03/2015]

[15] T. Bishop. (2014, January) *Android vs BlackBerry vs iOS vs Symbian vs Windows Phone in South Africa.* Deloitte Digital. [Online]. Available: http://deloittedigital.co.za/assets/pdf/mobile-operating-systems-south-africa.pdf [Accessed: 18/02/2015]

[16] BlackBerry Limited. *BlackBerry Java 7.1 SDK.* [Online]. Available: http://developer.blackberry.com/bbos/java/documentation/introducing_sdk_1968206_11.html [Accessed: 22/12/2014]

[17] ——. *CLDC platform and utilities APIs.* [Online]. Available: http://developer.blackberry.com/bbos/java/documentation/v71_cldc_category_1950148_11.html [Accessed: 22/12/2014]

[18] ——. *Localizing your apps.* [Online]. Available: http://developer.blackberry.com/bbos/java/documentation/localize_apps_2006594_11.html [Accessed: 28/12/2014]

[19] ——. *Signing your application.* [Online]. Available: http://developer.blackberry.com/bbos/java/documentation/code_signing_1977871_11.html [Accessed: 22/12/2014]

[20] ——. *UI components.* [Online]. Available: http://developer.blackberry.com/bbos/java/documentation/intro_to_ui_1969897_11.html [Accessed: 22/12/2014]

[21] S. Booysen, "With the ballot and the brick: the politics of attaining service delivery," *Progress in Development Studies*, vol. 7, no. 1, pp. 21–32, 2007.

[22] D. Brown, G. Marsden, and U. Rivett, "WATER alert!: using mobile phones to improve community perspective on drinking water quality in South Africa," in *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development.* ACM, 2012, pp. 230–240.

[23] F. Buschmann, K. Henney, and D. Schimdt, *Pattern-oriented Software Architecture: On Patterns and Pattern Language.* John Wiley & Sons, 2007, vol. 5.

[24] R. Capurro, "Information ethics for and from Africa," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 7, pp. 1162–1170, 2008.

[25] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think.* Morgan Kaufmann, 1999.

[26] D. A. Carr, “Guidelines for designing information visualization applications,” *Proceedings of ECUE*, vol. 99, pp. 1–3, 1999.

[27] M. Champanis, U. Rivett, S. Gool, and M. Nyemba-Mudenda, “ICTs in the water sector – where do we stand?” 2013, Water Research Commission.

[28] A. Charland and B. Leroux, “Mobile application development: web vs. native,” *Communications of the ACM*, vol. 54, no. 5, pp. 49–53, 2011.

[29] L. Chittaro, “Visualizing information on mobile devices,” *Computer*, vol. 39, no. 3, pp. 40–45, 2006.

[30] J. H. Christensen, “Using RESTful web-services and cloud computing to create next generation mobile applications,” in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications.* ACM, 2009, pp. 627–634.

[31] W. S. Cleveland *et al.*, *The elements of graphing data.* Wadsworth Advanced Books and Software Monterey, CA, 1985.

[32] A. Cooper and R. Reimann, “About Face 2.0,” *The Essentials of Interaction Design*, 2003.

[33] D. Crockford, “The application/json media type for javascript object notation (json),” 2006.

[34] ——, “JSON: The fat-free alternative to XML,” in *Proc. of XML*, vol. 2006, 2006.

[35] M. De Jode, *Programming Java 2 Micro Edition for Symbian OS: A developer's guide to MIDP 2.0.* John Wiley & Sons, 2004.

[36] A. Dearden, A. Light, S. Dray, J. Thomas, M. Best, C. Buckhalter, D. Greenblatt, G. Krishnan, and N. Sambasivan, “User centered design and international development,” in *CHI'07 Extended Abstracts on Human Factors in Computing Systems.* ACM, 2007, pp. 2825–2828.

[37] Department of Cooperative and Traditional Affairs. (2014, October) *Back To Basics: Serving Our Communities Better.* [Online]. Available: www.cogta.gov.za/summit2014/wp-content/uploads/2014/10/plgsummit-backtobasics-discussion-document.pdf [Accessed: 25/03/2015]

[38] S. Dustdar and W. Schreiner, “A survey on web services composition,” *International journal of web and grid services*, vol. 1, no. 1, pp. 1–30, 2005.

[39] C. Eberhardt and C. Price. *NeoMAD Overview.* PropertyCross. [Online]. Available: http://propertycross.com/neomad/ [Accessed: 09/11/2014]

[40] S. Ekine. (2009, 9 January) *Ushahidi Platform Used to Document Congo, Gaza Crises.* Public Broadcasting Service (PBS). [Online]. Available: www.pbs.org/mediashift/2009/01/ushahidi-platform-used-to-document-congo-gaza-crises009/ [Accessed: 19/09/2014]

[41] Enough Software. *Janus – The Platform Of Your Choice.* [Online]. Available: www.j2mepolish.org/janus/index.html [Accessed: 28/10/2014]

[42] X. Faulkner and F. Culwin, "When fingers do the talking: a study of text messaging," *Interacting with computers*, vol. 17, no. 2, pp. 167–185, 2005.

[43] U. M. Fayyad, A. Wierse, and G. G. Grinstein, *Information visualization in data mining and knowledge discovery.* Morgan Kaufmann, 2002.

[44] S. Few, "Save the pies for dessert," *Visual Business Intelligence Newsletter*, pp. 1–14, 2007.

[45] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[46] J. Fildes. (2009, 19 August) *Web tool oversees Afghan election.* BBC News. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/8209172.stm [Accessed: 22/10/2014]

[47] ——. (2010, 1 March) *Net puts Kenya at centre of Chile rescue efforts.* BBC News. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/8543671.stm [Accessed: 22/09/2014]

[48] A. Fink. *Kannel User's Guide: Overview of SMS.* [Online]. Available: www.kannel.org/download/kannel-userguide-snapshot/userguide.html [Accessed: 31/10/2014]

[49] D. Flood, R. Harrison, C. Iacob, and D. Duce, "Evaluating mobile applications: A spreadsheet case study," *International Journal of Mobile Human Computer Interaction (IJMHCI)*, vol. 4, no. 4, pp. 37–65, 2012.

[50] S. Fokazi. (2012, 14 August) *SMS hotline passes test.* Independent Online. [Online]. Available: www.iol.co.za/news/politics/sms-hotline-passes-test-1.1362079 [Accessed: 17/10/2014]

[51] B. Foust, *BlackBerry Java Application Development: Beginner's Guide.* Packt Publishing, 2010.

[52] A. Frith. (2012, 13 October) *Census 2011 – Main Place Grahamstown.* [Online]. Available: http://census2011.adrianfrith.com/place/264004 [Accessed: 03/12/2014]

[53] FrontlineSMS. *FrontlineSMS Overview.* The Social Impact Lab. [Online]. Available: www.frontlinesms.com/technologies/frontlinesms-overview/ [Accessed: 22/10/2014]

[54] W. O. Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques.* John Wiley & Sons, 2007.

[55] J. Garside. (2014, 13 January) *OMG! Number of UK text messages falls for first time.* Guardian News and Media Limited. [Online]. Available: www.theguardian.com/technology/2014/jan/13/number-text-messages-sent-britain-falls-first-time [Accessed: 19/10/2014]

[56] A. F. Ghyasi and I. Kushchu, "Uses of Mobile Government in Developing Countries," *Mobile Government Lab (mGovLab. org) Retrieved November*, vol. 1, p. 11, 2004. [Online]. Available: www.mgov.cn/lab/Archives/EuromGov2005/Uses%20of%20Mobile%20Government%20in%20Developing%20Countries.pdf [Accessed: 7/9/2014]

[57] B. Gil and P. Trezentos, "Impacts of data interchange formats on energy consumption and performance in smartphones," in *Proceedings of the 2011 Workshop on Open Source and Design of Communication.* ACM, 2011, pp. 1–6.

[58] K. L. Gilbert. (2014, 23 September) *Life-saving Ebola messages in 160 characters or less.* UMC.org. [Online]. Available: www.umc.org/news-and-media/life-saving-ebola-messages-in-160-characters-or-less [Accessed: 22/10/2014]

[59] A. Giridharadas. (2010, 14 March) *Africa's Gift to Silicon Valley: How to Track a Crisis.* The New York Times. [Online]. Available: www.nytimes.com/2010/03/14/weekinreview/14giridharadas.html?scp=1&sq=ushahidi&st=cse [Accessed: 19/09/2014]

[60] M. Gladwell, "Small change," *The New Yorker*, vol. 4, no. 2010, pp. 42–49, 2010.

[61] A. Goldstuck. (2014, 29 July) *Massive increase in SA smartphone purchases.* Mail & Guardian. [Online]. Available: http://mg.co.za/article/2014-07-29-smartphone-uptake-in-south-africa-rising-dramatically [Accessed: 30/11/2014]

[62] A. Goncalves, "Restful web services," in *Beginning Java EE 7.* Springer, 2013, pp. 495–538.

[63] Google, Inc. (2014) *Getting Started With Charts.* [Online]. Available: http://developers.google.com/chart/image/docs/making_charts [Accessed: 31/10/2014]

[64] ——. (2014) *The Google Geocoding API: Reverse Geocoding.* [Online]. Available: http://developers.google.com/maps/documentation/geocoding/#ReverseGeocoding [Accessed: 31/10/2014]

[65] ——. (2014) *Using Google Charts.* [Online]. Available: http://developers.google.com/chart/interactive/docs/index [Accessed: 31/10/2014]

[66] P. Gordhan. *Budget Vote 3: Cooperative Governance and Traditional Affairs.* National Assembly. [Online]. Available: www.cogta.gov.za/index.php/speeches/582-budget-vote-3-speech-by-minister-pravin-gordhan [Accessed: 25/03/2015]

[67] J. Gossman. (2005, 8 October) *Introduction to Model/View/ViewModel pattern for building WPF apps.* Microsoft Corporation. [Online]. Available: http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx [Accessed: 27/12/2014]

[68] J. D. Gould and C. Lewis, “Designing for usability: key principles and what designers think,” *Communications of the ACM*, vol. 28, no. 3, pp. 300–311, 1985.

[69] Government of South Africa. (1996) *Constitution of the Republic of South Africa: Act 151(1) of 1996.* [Online]. Available: http://www.info.gov.za/documents/ constitution/ 1996/96cons7.htm [Accessed: 19/09/2014]

[70] ——. (2000, November) *Municipal Systems Act, No. 32 of 2000.* [Online]. Available: www.energy.gov.za/files/policies/act_municipalsystem_32of2000.pdf [Accessed: 19/09/2014]

[71] ——. (2013, 29 May) *Western Cape MEC of Health Theuns Botha: Debate on the national Health Budget Vote at the National Council of Provinces.* [Online]. Available: www.gov.za/western-cape-mec-health-theuns-botha-debate-national-health-budget-vote-national-council-provinces [Accessed: 19/09/2014]

[72] Grapevine Interactive. (2014) *Services.* [Online]. Available: www.vine.co.za/services/ [Accessed: 09/11/2014]

[73] R. Grinter and M. Eldridge, “Wan2tlk?: everyday text messaging,” in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2003, pp. 441–448.

[74] R. E. Grinter and M. A. Eldridge, “y do tngrs luv 2 txt msg?” in *ECSCW 2001.* Springer, 2001, pp. 219–238.

[75] GSMA. (2011, December) *African Mobile Observatory.* [Online]. Available: www.gsma.com/spectrum/wp-content/uploads/2011/12/Africa-Mobile-Observatory-2011.pdf [Accessed: 13/03/2015]

[76] ——. (2012, 13 November) *Sub-Saharan Africa Mobile Observatory.* [Online]. Available: www.gsma.com/publicpolicy/wp-content/uploads/2012/03/SSA_FullReport_v6.1_clean.pdf [Accessed: 13/03/2015]

[77] ——. (2014) *The Mobile Economy: Sub-Saharan Africa.* [Online]. Available: www.gsmamobileeconomyafrica.com/GSMA_ME_SubSaharanAfrica_Web_Singles.pdf [Accessed: 13/03/2015]

[78] ——. (2015) *The Mobile Economy: Sub-Saharan Africa.* [Online]. Available: https://gsmaintelligence.com/research/?file=721eb3d4b80a36451202d0473b3c4a63&download [Accessed: 28/02/2016]

[79] N. Hafkin, “E-government in africa: Progress made and challenges ahead,” in *Conference presentation, Electronic/Mobile Government in Africa: Building Capacity in Knowledge Management through Partnership*, 2009.

[80] R. Harper, L. A. Palen, and A. Taylor, *The inside text: Social, cultural and design perspectives on SMS.* Springer, 2006, vol. 4.

[81] G. Hartmann, G. Stead, and A. DeGani, "Cross-platform mobile development," *Tribal, Lincoln House, The Paddocks, Tech. Rep*, 2011.

[82] C. G. Healey and J. T. Enns, "Attention and visual memory in visualization and computer graphics," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 7, pp. 1170–1188, 2012.

[83] R. Heeks, *Reinventing government in the information age: International practice in IT-enabled public sector reform.* Psychology Press, 1999, vol. 1.

[84] R. Heeks *et al.*, *Most eGovernment-for-development projects fail: how can risks be reduced?* Institute for Development Policy and Management, University of Manchester Manchester, 2003.

[85] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo." *Commun. ACM*, vol. 53, no. 6, pp. 59–67, 2010.

[86] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Evaluating cross-platform development approaches for mobile applications," in *Web information systems and technologies.* Springer, 2013, pp. 120–138.

[87] E. Hersman. (2008, 26 May) *The Ushahidi engine in South Africa.* Ushahidi. [Online]. Available: www.ushahidi.com/2008/05/26/the-ushahidi-engine-in-south-africa/ [Accessed: 19/09/2014]

[88] M. Hesse. (2010, 15 January) *Crisis mapping brings online tool to Haitian disaster relief effort.* The Washington Post. [Online]. Available: www.washingtonpost.com/wp-dyn/content/article/2010/01/15/AR2010011502650.html [Accessed: 19/09/2014]

[89] M. R. Ho, T. N. Smyth, M. Kam, and A. Dearden, "Human-computer interaction for development: The past, present, and future," *Information Technologies & International Development*, vol. 5, no. 4, pp. pp–1, 2009.

[90] T. Holmes. (2014, 1 May) *JRA launches mobile app for Joburg motorists.* Mail & Guardian Online. [Online]. Available: http://mg.co.za/article/2014-05-01-jra-launches-mobile-app-for-joburg-motorists/ [Accessed: 19/10/2014]

[91] B. Hopkins. (2007, June) *The Java ME GUI APIs at a Glance.* [Online]. Available: www.oracle.com/technetwork/systems/guiapis-155788.html [Accessed: 28/12/2014]

[92] P. N. Howard and M. M. Hussain, *Democracy's fourth wave?: digital media and the Arab Spring.* Oxford University Press, 2013.

[93] C. Huang. (2011, 6 June) *Facebook and Twitter key to Arab Spring uprisings: report.* The National. [Online]. Available: www.thenational.ae/news/uae-news/facebook-and-twitter-key-to-arab-spring-uprisings-report [Accessed: 22/10/2014]

[94] J. Hutton. (2011, 30 September) *Mobile phones dominate in South Africa.* The Nielsen Company. [Online]. Available: www.nielsen.com/us/en/insights/news/2011/mobile-phones-dominate-in-south-africa.html [Accessed: 13/03/2015]

[95] iCOMMS. (2014) *iCOMMS Research Team.* Department of Civil Engineering, Univerity of Cape Town. [Online]. Available: www.civil.uct.ac.za/icomms-research-team [Accessed: 15/11/2014]

[96] International Federation for Information Processing. (1988) *IFIP Working Group 8.5.* [Online]. Available: www.ifip.org/bulletin/bulltcs/tc8_aim.htm#wg85 [Accessed: 22/10/2014]

[97] International Organization for Standardization, "9241-11. ergonomic requirements for office work with visual display terminals (vdts)–part 11: Guidance on usability," 1998.

[98] International Telecommunication Union. (2014, April) *The World in 2014: ICT Facts and Figures.* [Online]. Available: www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf [Accessed: 19/11/2014]

[99] Internews Network. (2014, 24 April) *Afghan Journalists Put Data Journalism Skills into Practice.* [Online]. Available: http://internews.org/afghan-journalists-put-data-journalism-skills-practice [Accessed: 11/11/2014]

[100] IRIN. (2011, 6 September) *Using text messaging as weapon in malaria war.* ReliefWeb. [Online]. Available: http://reliefweb.int/report/cambodia/using-text-messaging-weapon-malaria-war [Accessed: 11/11/2014]

[101] Johannesburg Roads Agency. *JRA Find&Fix Mobile App: Frequently Asked Questions.* [Online]. Available: www.jra.org.za/index.php/find-and-fix-mobile-app [Accessed: 19/10/2014]

[102] Kannel Group. *Kannel: Overview.* [Online]. Available: www.kannel.org/overview.shtml [Accessed: 29/10/2014]

[103] V. K. Katankar and V. Thakare, "Short message service using sms gateway," *International Journal on Computer Science and Engineering*, vol. 2, no. 04, pp. 1487–1491, 2010.

[104] J. E. Katz, *Mobile communication: dimensions of social policy.* Transaction Publishers, 2011, vol. 1.

[105] D. A. Keim, "Information visualization and visual data mining," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, no. 1, pp. 1–8, 2002.

[106] P. R. Keller and M. M. Keller, *Visual cues: practical data visualization.* IEEE Computer Society Press Los Alamitos, CA, 1993.

[107] K. Kiili, "Evaluating wap usability: "what usability?"," in *Wireless and Mobile Technologies in Education, 2002. Proceedings. IEEE International Workshop on.* IEEE, 2002, pp. 169–170.

[108] M. Kumar and O. P. Sinha, "M-government–mobile technology for e-government," in *International conference on e-government, India*, 2007, pp. 294–301.

[109] V. Kumar, S. Parimi, and D. P. Agrawal, "Wap: present and future," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 79–83, 2003.

[110] I. Kushchu and H. Kuscu, "From e-government to m-government: Facing the inevitable," in *the 3rd European Conference on e-Government*, 2003, pp. 253–260.

[111] E. C. Lallana. (2008) *TXT CSC: SMS Service for the Philippines Civil Service Commission.* eGovernment for Development. [Online]. Available: www.egov4dev.org/mgovernment/resources/case/txtcsc.shtml [Accessed: 05/09/2014]

[112] J. H. Larkin and H. A. Simon, "Why a diagram is (sometimes) worth ten thousand words," *Cognitive science*, vol. 11, no. 1, pp. 65–100, 1987.

[113] O. Laursen. (2013, 3 July) *Flot: About.* IOLA. [Online]. Available: http://github.com/flot/flot/blob/master/README.md [Accessed: 09/11/2014]

[114] ——. (2014) *Flot: Attractive JavaScript plotting for jQuery.* IOLA. [Online]. Available: www.flotcharts.org/ [Accessed: 31/10/2014]

[115] ——. (2014) *Flot: Basic Options.* IOLA. [Online]. Available: www.flotcharts.org/flot/examples/basic-options/index.html [Accessed: 09/11/2014]

[116] ——. (2014) *Flot: Series Types.* IOLA. [Online]. Available: www.flotcharts.org/flot/examples/series-types/index.html [Accessed: 09/11/2014]

[117] ——. (2014) *Flot: Usage.* IOLA. [Online]. Available: http://github.com/flot/flot/wiki/Flot-Usage [Accessed: 09/11/2014]

[118] G. Le Bodic, *Mobile Messaging technologies and services: SMS, EMS and MMS.* John Wiley & Sons, 2005.

[119] R. Lerdorf, K. Tatroe, and P. MacIntyre, *Programming Php.* " O'Reilly Media, Inc.", 2006.

[120] Library of Congress. (2013, 11 January) *Codes for the Representation of Names of Languages.* [Online]. Available: www.loc.gov/standards/iso639-2/php/English_list.php [Accessed: 06/12/2014]

[121] D. Lombard, *The Second Life of Networks.* Odile Jacob Publishing Corp, 2008.

[122] S. Love, *Understanding mobile human-computer interaction.* Butterworth-Heinemann, 2005.

[123] S. C. Y. Luk, "The impact of leadership and stakeholders on the success/failure of e-government service: Using the case study of e-stamping service in hong kong," *Government Information Quarterly*, vol. 26, no. 4, pp. 594–604, 2009.

[124] A. O. Lungati. (2013, 10 March) *Ushahidi via USSD.* Ushahidi. [Online]. Available: http://www.ushahidi.com/2013/10/03/ushahidi-via-ussd/ [Accessed: 04/10/2014]

[125] Lungisa. (2014) *Frequently Asked Questions.* Cell-Life. [Online]. Available: www.lungisa.org/faq.php [Accessed: 21/10/2014]

[126] ——. (2014) *Home.* Cell-Life. [Online]. Available: www.lungisa.org/ [Accessed: 21/10/2014]

[127] ——. (2014) *Information.* Cell-Life. [Online]. Available: www.lungisa.org/information.php [Accessed: 21/10/2014]

[128] ——. (2014) *Report An Issue.* Cell-Life. [Online]. Available: www.lungisa.org/how-to-report-an-issue.php [Accessed: 21/10/2014]

[129] J. Mahlokwane. (2014, 7 October) *Highest honour for city's JRA find & fix app.* Southern Courier. [Online]. Available: http://southerncourier.co.za/43949/highest-honour-for-citys-jra-find-fix-app/ [Accessed: 19/09/2014]

[130] Makana Municipality. (2010, 30 June) *Annual financial statements.* [Online]. Available: http://mfma.treasury.gov.za/Documents/05.%20Annual%20Financial%20Statements/2009-10/02.%20Local%20Municipalities/EC104%20Makana/EC104%20Makana%20AFS%202009-10%20Unaudited.pdf [Accessed: 14/02/2015]

[131] ——. (2011, 31 August) *Integrated development plan review.* [Online]. Available: http://www.makana.gov.za/index.php?option=com_docman&task=cat_view &Itemid=26&gid=11 [Accessed: 8/3/2015]

[132] ——. (2014, 31 March) *Makana Municipality integrated development plan.* [Online]. Available: www.makana.gov.za/wp-content/uploads/2013/06/IDP-FINAL-2014-15.pdf [Accessed: 12/02/2015]

[133] S. Mann and S. Sbihli, *The Wireless Application Protocol (WAP): A Wiley Tech Brief.* John Wiley & Sons, 2002, vol. 12.

[134] A. Marwaha. (2008, 10 December) *Web tool maps Congo conflict.* BBC World Service. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/7773648.stm [Accessed: 22/09/2014]

[135] R. Mathekga and I. Buccus, "The challenge of local government structures in south africa: securing community participation," *Crit Dialogue Public Participation Rev*, vol. 2, no. 1, pp. 11–17, 2006.

[136] A. Maunder, G. Marsden, D. Gruijters, and E. Blake, "Designing interactive systems for the developing world-reflections on user-centred design," in *Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on.* IEEE, 2007, pp. 1–8.

[137] I. Medhi, A. Sagar, and K. Toyama, "Text-free user interfaces for illiterate and semi-literate users," in *Information and Communication Technologies and Development, 2006. ICTD'06. International Conference on.* IEEE, 2006, pp. 72–82.

[138] P. Meier. (2010, 5 May) *Ushahidi Used in Italy to Prevent Forest Fires.* Ushahidi. [Online]. Available: www.ushahidi.com/2010/05/05/ushahidi-used-in-italy-to-prevent-forest-fires/ [Accessed: 19/09/2014]

[139] ——. (2011, 6 March) *Using the New Ushahidi Platform to Crisis Map Lybia.* Ushahidi. [Online]. Available: www.ushahidi.com/2011/03/06/using-new-ushahidi-map-libya/ [Accessed: 19/09/2014]

[140] I. Mergel, "Distributed democracy: Seeclickfix.com for crowdsourced issue reporting," *Social Science Research Network*, 2012.

[141] P. Mini. (2011, 25 July) *Bad leadership drains municpal coffers.* Grocott's Mail. [Online]. Available: www.grocotts.co.za/content/bad-leadership-drains-municipal-coffers-25-07-2011 [Accessed: 24/01/2015]

[142] ——. (2011, 27 January) *Makana Municipality in a mess, says report.* Grocott's Mail. [Online]. Available: www.grocotts.co.za/content/makana-municipality-mess-says-report-27-01-2011 [Accessed: 24/01/2015]

[143] A. Mngxitama-Diko. (2013, 3 February) *Failed audit: "blame bosses".* Grocott's Mail. [Online]. Available: www.grocotts.co.za/content/directors-blame-failed-audit-03-02-2013 [Accessed: 24/01/2015]

[144] ——. (2013, 11 February) *Ghost employees rake in millions from Makana.* Grocott's Mail. [Online]. Available: www.grocotts.co.za/content/ghost-employees-rake-millions-makana-11-02-2013 [Accessed: 24/01/2015]

[145] A. Mngxitama-Diko and A. Mjekula. (2014, 28 August) *Gordhan seals Makana handover.* Grocott's Mail. [Online]. Available: www.grocotts.co.za/content/online-first-news-gordhan-seals-makana-handover-28-08-2014 [Accessed: 24/04/2015]

[146] MobileInfo. *Business Approach: NTTDoCoMo vs. European and American Telecoms.* [Online]. Available: www.mobileinfo.com/imode/buz_approach.htm [Accessed: 14/11/2014]

[147] MOBITEK System Sdn. Bhd. *SMS Gateway Development Kit.* [Online]. Available: www.mobitek-system.com/SMS_Gateway/SMS_Gateway.html [Accessed: 31/10/2014]

[148] T. M. Mogale, "Local governance and poverty reduction in south africa (i)," *Progress in development studies*, vol. 5, no. 2, pp. 135–143, 2005.

[149] A. Monga, "E-government in india: Opportunities and challenges," *JOAAG*, vol. 3, no. 2, pp. 52–61, 2008.

[150] R. Mumbi and P. Ghazi. (2011, 14 July) *Zambia: Climate Information Alerts Boost Poor Farmers.* AllAfrica. [Online]. Available: http://allafrica.com/stories/201107141317.html [Accessed: 22/10/2014]

[151] B. Myers, "Challenges of hci design and implementation," *interactions*, vol. 1, no. 1, pp. 73–83, 1994.

[152] V. Ndou, "E-government for developing countries: opportunities and challenges," *The Electronic Journal of Information Systems in Developing Countries*, vol. 18, 2004.

[153] Neomades SAS. *NeoMAD 3.8 documentation.* [Online]. Available: http://docs.neomades.com/en/current/neomad_at_a_glance/en/presentation.html [Accessed: 10/11/2014]

[154] T. Ngubeni. (2014, 20 January) *ANC Mxit app targets youth.* ITWeb. [Online]. Available: www.itweb.co.za/index.php?id=70198 [Accessed: 10/10/2014]

[155] J. Nielsen. (1999, 31 October) *Graceful Degradation of Scalable Internet Services.* Nielsen Norman Group. [Online]. Available: www.nngroup.com/articles/graceful-degradation-of-scalable-internet-services/ [Accessed: 07/01/2015]

[156] ——. (2012, 4 January) *Usability 101: Introduction to Usability.* [Online]. Available: www.nngroup.com/articles/usability-101-introduction-to-usability/ [Accessed: 21/02/2015]

[157] N. Nkwe, "E-government: challenges and opportunities in botswana," *International Journal of Humanities and Social Science*, vol. 2, no. 17, pp. 39–48, 2012.

[158] Nokia Corporation. *Series 40 Touch and Type Design Guidelines.* [Online]. Available: http://developer.nokia.com/resources/library/Touch_and_Type/ui-patterns/confirming-changes.html [Accessed: 21/02/2015]

[159] D. A. Norman, *The psychology of everyday things.* Basic books, 1988.

[160] ——, *The design of everyday things.* Basic books, 2002.

[161] D. A. Norman and S. W. Draper, "User centered system design," *New Perspectives on Human-Computer Interaction, L. Erlbaum Associates Inc., Hillsdale, NJ*, 1986.

[162] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of json and xml data interchange formats: A case study." *Caine*, vol. 2009, pp. 157–162, 2009.

[163] S. Odugbemi and T. Lee, *Accountability through public opinion: from inertia to public action.* World Bank Publications, 2011.

[164] A. P. Oghuma, M. Park, and J. J. Rho, "Adoption of mgovernment service initiative in developing countries: A citizen-centric public service delivery perspective," 2012.

[165] J. Ohrt and V. Turau, "Cross-platform development tools for smartphone applications," *Computer*, vol. 45, no. 9, pp. 0072–79, 2012.

[166] O. Okolloh, "Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information," *Participatory learning and action*, vol. 59, no. 1, pp. 65–70, 2009.

[167] Oracle Corporation. (2007, April) *JSR 118: Mobile Information Device Profile 2.0.* [Online]. Available: www.jcp.org/en/jsr/detail?id=118 [Accessed: 28/12/2014]

[168] Organisation for Economic Co-operation and Development, *M-Government: Mobile Technologies for Responsive Governments and Connected Societies.* OECD Publishing, 2011.

[169] A. Oxford. (2014, 7 October) *Joburg Roads' Find&Fix app logs nearly 10 000 potholes and problems.* htxt.africa. [Online]. Available: www.htxt.co.za/2014/10/07/joburg-roads-findfix-app-logs-nearly-10-000-potholes-and-problems/ [Accessed: 22/11/2014]

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

[171] J. Palomäki, "Case wap: reasons for failure," *Innovation Dynamics in Mobile Communications*, p. 98, 2004.

[172] T. Parikh, K. Ghosh, and A. Chavan, "Design studies for a financial management system for micro-credit groups in rural india," in *ACM SIGCAPH Computers and the Physically Handicapped*, no. 73-74. ACM, 2003, pp. 15–22.

[173] B. Phakathi. (2014, 8 April) *DA puts social media strategy in overdrive ahead of elections.* Times Media (Pty) Ltd. [Online]. Available: www.bdlive.co.za/national/politics/2014/04/08/da-puts-social-media-strategy-in-overdrive-ahead-of-elections [Accessed: 22/09/2014]

[174] M. Poblet, "Mobile technologies for conflict management: Online dispute resolution," *Governance, Participation. Springer, Berlin*, 2011.

[175] C. Z. Qiang, S. C. Kuek, A. Dymond, S. Esselaar, and I. S. Unit, "Mobile applications for agriculture and rural development," *World Bank, Washington, DC*, 2011.

[176] M. Raja, M. Cruse, J. Goldstein, K. Maher, M. Minges, and P. Surya, "Making government mobile," *Information and Communications for Development, ed*, pp. 87–101, 2012.

[177] L. Rakotomalala. (2014) *Technology for Transparency - Text Messages To Help To Protect Children Against Violence.* We-Magazine. [Online]. Available: www.we-magazine.net/we-volume-04/technology-for-transparency---text-messages-to-help-to-protect-children-against-violence/ [Accessed: 22/10/2014]

[178] D. Ralph and P. Graham, *MMS: Technologies, usage and business models.* John Wiley & Sons, 2004.

[179] P. Rein, M. Champanis, and U. Rivett, "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.* ACM, 2013, pp. 124–127.

[180] E. Reynell, H. Thinyane, and I. Siebörger, "Enhancing digital inclusion through mobile social accountability monitoring," in *South Africa Telecommunication Networks and Applications Conference (SATNAC).* SATNAC, 2013, pp. 253–258.

[181] ——, "Mobile social accountability monitoring in a connected society," in *Proceedings of 4th International Conference on M4D Mobile Communication for Development*, 2014, p. 196.

[182] L. Richardson and S. Ruby, *RESTful web services.* O'Reilly Media, Inc., 2008.

[183] U. Rivett, M. Champanis, and T. Wilson-Jones, "Monitoring drinking water quality in south africa: water quality," *Water&Sanitation Africa*, vol. 9, no. 3, pp. 24–27, 2014.

[184] A. Rodriguez, "Restful web services: The basics," *IBM developerWorks*, 2008.

[185] Y. Rogers, H. Sharp, and J. Preece, *Interaction design: beyond human-computer interaction.* John Wiley & Sons, 2011.

[186] N. Ross and M. Potts. (2011, 13 January) *ABC's crowdsourced flood-mapping initiative.* Australian Broadcasting Corporation. [Online]. Available: www.abc.net.au/technology/articles/2011/01/13/3112261.htm [Accessed: 17/10/2014]

[187] SABC News. (2012, 24 October) *Anti Corruption Hotline launched in City of Johannesburg.* [Online]. Available: www.sabc.co.za/news/a/ede61f804d32441cbc39fde570eb4ca2/Anti-Corruption-Hotline-launched-in-City-of-Johannesburg--20121024 [Accessed: 19/09/2014]

[188] J. Sauro and J. R. Lewis, "Average task times in usability tests: what to report?" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2010, pp. 2347–2350.

[189] D. J. Schiano, C. P. Chen, E. Isaacs, J. Ginsberg, U. Gretarsdottir, and M. Huddleston, "Teen use of messaging media," in *CHI'02 Extended Abstracts on Human Factors in Computing Systems.* ACM, 2002, pp. 594–595.

[190] G. Schmiedl, M. Seidl, and K. Temper, "Mobile phone web browsing: a study on usage and usability of the mobile web," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services.* ACM, 2009, p. 70.

[191] T. Schuppan, "E-government in developing countries: Experiences from sub-saharan africa," *Government Information Quarterly*, vol. 26, no. 1, pp. 118–127, 2009.

[192] SeeClickFix, Inc. *SeeClickFix: How It Works.* [Online]. Available: http://seeclickfix.com/how-it-works [Accessed: 19/09/2014]

[193] H. Sharp, *Interaction design.* John Wiley & Sons, 2003.

[194] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on.* IEEE, 1996, pp. 336–343.

[195] G. Skalski. (2013, 24 September) *Software company Ushahidi uses open source skills to help during Kenya mall siege.* Red Hat, Inc. [Online]. Available: http://opensource.com/life/13/9/ushahidi-kenya-mall-siege [Accessed: 19/09/2014]

[196] South African Press Association. (2014, 3 November) *ANC's Mxit app a winner.* Independent Online. [Online]. Available: www.iol.co.za/news/politics/anc-s-mxit-app-a-winner-1.1774729 [Accessed: 20/11/2014]

[197] R. Spence, *Information visualization.* Addison-Wesley, 2000.

[198] S. S. Sriparasa, *JavaScript and JSON Essentials.* Packt Publishing Ltd, 2013.

[199] Statistics South Africa. *General household survey 2013.* [Online]. Available: http://beta2.statssa.gov.za/publications/P0318/P03182013.pdf [Accessed: 13/11/2014]

[200] ——. (2011) *Local Municipality: Makana.* [Online]. Available: http://beta2.statssa.gov.za/?page_id=993&id=makana-municipality [Accessed: 01/11/2014]

[201] E. Stepanova, "The role of information communication technologies in the "arab spring"," *PONARS Eurasia Policy Memo*, vol. 159, pp. 1–6, 2011.

[202] A. C. Telea, *Data visualization: principles and practice*, 2nd ed. Ak Peters, 2014.

[203] H. Thinyane and D. Coulson, "MobiSAM: Mobile Social Accountability Monitoring in South Africa," *Proceedings of M4D 2012 28-29 February 2012 New Delhi, India*, vol. 28, no. 29, p. 170, 2012.

[204] J. J. Thomas and K. A. Cook, "A visual analytics agenda," *Computer Graphics and Applications, IEEE*, vol. 26, no. 1, pp. 10–13, 2006.

[205] C. Thurlow and M. Poff, "Text messaging," *Handbook of the pragmatics of CMC. Berlin and New York: Mouton de Gruyter*, 2011.

[206] Times Live. (2014, 14 May) *Johannesburg Find&Fix potholes app launches.* Times Media Group. [Online]. Available: www.timeslive.co.za/scitech/2014/05/14/johannesburg-find-fix-potholes-app-launches [Accessed: 19/09/2014]

[207] S. Trimi and H. Sheng, "Emerging trends in m-government," *Communications of the ACM*, vol. 51, no. 5, pp. 53–58, 2008.

[208] United Nations. *Global E-Government Readiness Report 2005: From E-Government to E-Inclusion.* [Online]. Available: http://unpan1.un.org/intradoc/groups/public/documents/un/unpan021888.pdf [Accessed: 19/09/2014]

[209] G. Vincent, "Learning from i-mode [packet-based mobile network]," *IEE review*, vol. 47, no. 6, pp. 13–18, 2001.

[210] R. Virkus, R. Gülle, T. Rouffineau, C. Brady, W. Kriesing *et al.*, *Don't Panic, Mobile Developer's Guide to the Galaxy*, 14th ed. Enough Software GmbH+ Co. KG, 2014.

[211] R. Virkus, *Pro J2ME Polish.* Springer, 2005.

[212] M. Ward, G. Grinstein, and D. Keim, *Interactive data visualization: foundations, techniques, and applications.* AK Peters, Ltd., 2010.

[213] C. Ware, *Information visualization: perception for design.* Elsevier, 2013.

[214] J. Winesett, *Web Application development with Yii and PHP.* Packt Publishing Ltd, 2012.

[215] World Wide Worx. (2014, October) *South African Social Media Landscape.* [Online]. Available: www.worldwideworx.com/wp-content/uploads/2013/10/Exec-Summary-Social-Media-2014.pdf [Accessed: 23/10/2014]

[216] D. Zhang, "Web content adaptation for mobile handheld devices," *Communications of the ACM*, vol. 50, no. 2, pp. 75–79, 2007.

[217] E. Zudilova-Seinstra, T. Adriaansen, and R. van Liere, *Trends in interactive visualization: state-of-the-art survey.* Springer, 2008.

Appendix A

Technical implementation details

A.1 MobiSAM backend server

The backend consists of three separate components: a PHP framework (Appendix A.1.1), REST API (Appendix A.1.2) and database (Appendix A.1.3). Each of these components will now be described.

A.1.1 PHP framework

Originally created by Rasmus Lerdorf, PHP is a server-side open source scripting language primarily designed for web development purposes. PHP has the advantage of being platform independent as well as supporting all major web servers and databases [119]. The MobiSAM backend leverages PHP5 through the use of the Yii framework which simplifies the implementation process by providing a substantial amount of base PHP code. Yii also supports a variety of caching mechanisms, thereby potentially reducing both client and server-side bandwidth consumption [214].

A.1.2 MobiSAM representational state transfer (REST) API

REST is an architectural style based on how the web works, making use of HTTP and URIs, with each unique URI representing a resource [62]. All interaction with MobiSAM takes place via its REST API, irrespective of whether SMS, client application or the MobiSAM website is used. Although still in beta release, the API allows clients to perform all the necessary operations, such as authorisation, authentication, fetching polls, fetching categories, responding to polls and the uploading of photos. Communication between the clients and REST API are conducted via HTTP requests which contain JSON-formatted strings embedded within each request body, providing information such as user responses or photo upload data. Similarly, all responses from the backend are JSON-encoded. The MobiSAM REST API specification is listed in Appendix A and includes examples of request and response messages.

A.1.3 Database

The backend server's database provides persistent storage capabilities to MobiSAM, storing all user profiles, polls, categories, responses and uploaded photos, along with additional information. The database was implemented using the MySQL relational database management system in conjunction with the MyISAM storage engine. Indirect access to the database is provided via the REST API which abstracts the implementation and storage details. This simplifies access to MobiSAM as well as ensuring that only privileged users are granted access to potentially sensitive data.

A.2 Prototype MobiSAM client applications

This section provides an overview of the design and implementation of the platform-native, high-fidelity prototype client applications. First, the Java ME and BlackBerry OS platforms are introduced (Appendix A.2.1). Next, the prototype client application requirements and specifications are discussed (Appendix A.2.2). The design (Appendix A.2.3) and implementation (Appendix A.2.4) of the prototype client applications are then described.

A.2.1 Mobile platforms

Java Platform, Micro Edition Java Platform, Micro Edition (Java ME) is a lightweight version of the Java Platform, Standard Edition (Java SE) which is suitable for the execution of applications (or *midlets*) on resource constrained devices with limited capabilities, such as mobile phones [35]. As Java ME is derived from Java SE it shares many similarities with its fully-fledged sibling, albeit restricted to the features of Java Runtime Environment (JRE) 1.3 and below.

Devices supporting Java ME are required to implement a configuration and a profile. A configuration provides "the lowest common denominator for a group of devices" and is comprised of a virtual machine and a set of class libraries which are "designed to provide the base functionality for a distinct set of devices with similar characteristics" [35, p.5]. There are two such configurations: Connected Device Configuration (CDC) and Connected Limited Device Configuration (CLDC). While the former is suited to automotive navigation and entertainment systems, the latter is often implemented by mobile phones. A profile, on the other hand, "adds an additional layer on top of the configuration providing APIs for a specific class of device" allowing devices to be adapted towards vertical markets [35, p.6]. The most common profile implemented by mobile handsets is the Mobile Information Device Profile (MIDP). There have been three major revisions of MIDP, all developed under the Java Community Process: MIDP 1.0 (JSR 37), MIDP 2.0 (JSR 118) and MIDP 3.0 (JSR 271). MIDP 2.1 (JSR 118 MR2) also exists, however, instead of introducing new features its focus is instead on reducing fragmentation among Java ME devices, thereby improving third party application compatibility [167].

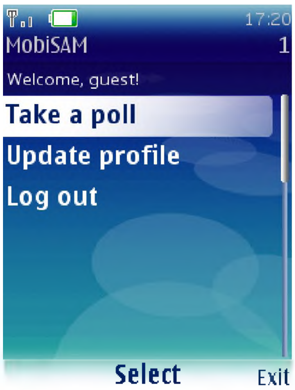


Figure A.1: Prototype Java ME client application: *Home* screen

The Java ME platform allows developers to create installable applications which, in theory, are able to run on any device which supports the targeted MIDP of the CLDC. Although Sun Microsystems provides a reference implementation for the Java ME specification, mobile phone manufacturers are required to provide binary implementations of the runtime environment, often leading to incompatibilities across different mobile devices. The Java ME MobiSAM client application, for example, refused to install on a Samsung E2100 feature phone, even though the handset offered Java ME support (CLDC 1.1/MIDP 2.1). The baseline study, presented in Chapter 4, highlighted the prevalence of Java ME compatible handsets within the Grahamstown area, with 28.42% of the handsets surveyed implementing either MIDP 2.0 or implementing MIDP 2.1.

BlackBerry OS Making its dêbut in January 1999, BlackBerry OS is a closed source mobile operating system developed by Research In Motion (RIM) for their BlackBerry range of smartphones. Leveraging a Java virtual machine based kernel, the platform allows developers to create installable applications using the Java programming language. Full multitasking is supported by the platform, enabling multiple applications to run simultaneously [51].

The BlackBerry Java SDK comes bundled with Java ME APIs as well as a collection of BlackBerry specific APIs. Although this allows Java ME applications to run, unmodified, on the

BlackBerry platform, RIM cautions that "they are not optimised for BlackBerry screens or functionality, and [are thus] not suitable for creating quality apps" [20, p.1]. Consequently, UI inconsistencies may arise due to the specialised input techniques used by many BlackBerry devices such as trackpads and trackballs [16]. Applications need to be digitally signed before gaining access to certain device functionalities. As of 2012, the digital signing process is free, requiring developers to register and request a code-signing key via the BlackBerry Developer website [19]. As with Java ME, the BlackBerry platform defines both a supported configuration and a profile, with the latest version of the platform (7.1) supporting CLDC 1.1 and profile MIDP 2.1 [16, 17].

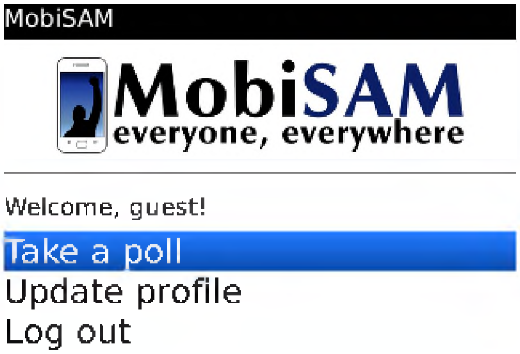


Figure A.2: Prototype BlackBerry OS client application: *Home* screen

It should be noted that although BlackBerry OS has since been superseded by the QNX-based BlackBerry 10 operating system, it was decided to support the original platform due to its ongoing prevalence within South Africa [15]. The baseline study, presented in Chapter 4, revealed that 11.58% of respondents' handsets leverage BlackBerry OS.

A.2.2 Requirements and specifications

As the requirements for the Java ME and BlackBerry client applications are identical, they are collectively discussed here. Specifications are also jointly discussed, where applicable.

As described in Section 3.4, the interaction design life-cycle model was followed when conducting this research. During the first iteration of the development life-cycle, the initial requirements for the client application were elicited from the MobiSAM project director and stakeholder. Although mobile access to the MobiSAM service was already provided via a mobile-optimised website, it was argued that a more engaging user experience could be achieved through the use of individual, platform-specific client applications. As such, the following broad set of requirements for the MobiSAM client application were defined:

- the application needs to target Java ME and BlackBerry OS platforms;

- multi-language (English, Afrikaans and isiXhosa) UIs must be provided; and
- improved poll result visualisation capabilities (when compared to the mobile-optimised website) need be included.

The resulting client application specifications defined which platforms were to be initially targeted (Java ME and BlackBerry OS), which languages were to be supported (English, isiXhosa and Afrikaans), as well as which visualisation techniques and tools would be employed (bar charts using the ChartComponent plotting library as well as the Google Image Chart API). This information provided a starting point for the design of each client application, the details of which are described next.

A.2.3 Design

Both Java ME and BlackBerry platforms provide specific UI and UX guidelines which developers are urged to follow when creating applications. Consequently, while the prototype client applications share identical functionality, their resulting designs differ slightly. The design of each is described next.

User interface and navigation The Java ME client application was designed first and followed a simple menu-driven UI style. A strong contrast between UI elements was created, helping ensure menu options remain legible in harsh lighting conditions. Figure A.1 depicts the *Home* screen of the Java ME prototype client application running on the Nokia Series 40 5th Edition simulator. A similar menu-driven *Home* screen was designed for the BlackBerry prototype client application, albeit with minor differences, as shown in Figure A.2.

Visualisation A primary argument for the development of installable MobiSAM client applications was the ability to offer improved visualisation capabilities over the mobile-optimised website, potentially helping users make better sense of the poll responses.

Most (93.7%) of Makana Municipality citizens aged 20 years and older have attended some form of schooling [200]. It was believed that a broad cross-section of the population have been exposed to basic charts through mathematics classes in both primary school as well as the first two years of high school. As such, it was decided to use a simple bar chart to visualise poll responses. In addition, poll responses were presented in a numerical table.

When visualised using a bar chart, each poll category appears as a rectangle, with its length proportional to the value it represents. Ward *et al.* define several fundamental visual variables which, when used appropriately, allow for effective and insightful visualisations [212]. A number of these were given special attention when designing the bar chart visualisation, including position, mark, size, brightness and colour. The potential caveats of mobile visualisation, highlighted by Chittaro [29], were also kept in mind (see Section 2.8.1).

Although both prototype client applications employed similar bar charts, their implementation details remained markedly different, as described in the Sections A.2.4 and A.2.4.

A.2.4 Implementation

While the design of the Java ME and BlackBerry prototype client applications shared many similarities, the implementation tools and techniques were notably different. As a result, the implementation details of each prototype client application are described separately. Although both platforms leverage Java ME as their runtime environment (with BlackBerry OS implementing additional, platform specific libraries), the client applications shared minimal program code. In addition, as the UIs of each client application were constructed using two different techniques, no common UI code existed. The implementation of each client application will be described next.

Java ME

Development environment The NetBeans IDE was installed on the development workstation, along with the Java ME 3.2 SDK and Nokia Series 40 5th Edition SDK. A conscious decision was made to install two SDKs as it allowed for the compilation and testing of the Java ME prototype client application using both the Java ME SDK and simulator, as well as the Nokia-centric Series 40 SDK and its corresponding simulator. Although similar, the Java ME SDK is wholly based on the Java ME specification, whereas the Series 40 SDK includes a selection of Nokia-specific libraries. The Java ME prototype client application targeted CLDC 1.1 and MIDP 2.0. As the physical handset used for debugging purposes (a Nokia 2630) ran the Nokia Series 40 5th Edition platform, the corresponding SDK was installed.

User interface In Java ME, the most basic UI elements are referred to as components. Both high- and low-level UI components are supported by MIDP implementations. The former are rendered by the mobile platform, adapting to the device's native 'look and feel', while the latter delegates the rendering to the developer, subsequently allowing for greater control over the UI. The use of low-level UI components is particularly appropriate where pixel-level control is needed, such as interactive games [91]. Since the prototype client application was designed to follow a simple menu-driven UI style, this level of control was unnecessary, potentially complicating the development process. As such, high-level UI components were used exclusively when constructing client application screens.

Figure A.3: Nokia Series 40 5th Edition soft keys

A variety of high-level UI components are made available to Java ME developers (such as `Alert`, `ChoiceGroup`, `DateField`, `Form`, `Gauge`, `ImageItem`, `List`, `StringItem`, `TextBox` and `TextField`), all of which are subclasses of the `Screen` and `Item` class [91]. Any combination of these components may be used to construct application UIs. Touch and key events are handled by `Commands` which are attached to particular UI components. As many feature phones make use of a hardware keypad for input, on-screen buttons known as soft keys are often used for navigation purposes (as shown in Figure A.3). These buttons are implemented through the use of `Commands` which are automatically mapped onto them by the platform.

Two primary techniques exist for the construction of Java ME application UIs:

- programmatically, using Java code; or
- using the Visual Mobile Designer (VMD) tool included in the NetBeans Mobility IDE package.

The second approach was used to implement the prototype client application UIs due to the speed and simplicity it offered. First, the client application UIs were constructed using the VMD 'drag & drop' editor. `Commands` were then added, allowing for navigation between different UIs. Figure A.4 shows the 'flow' view of the Java ME client application within the VMD. Each rectangle represents a different `Alert`, `Form` or `Screen` component, with lines between each representing a unique `Command`. UIs are modified by switching to the 'screen' view which allows the developer to add/remove as well as modify the position of individual UI components.

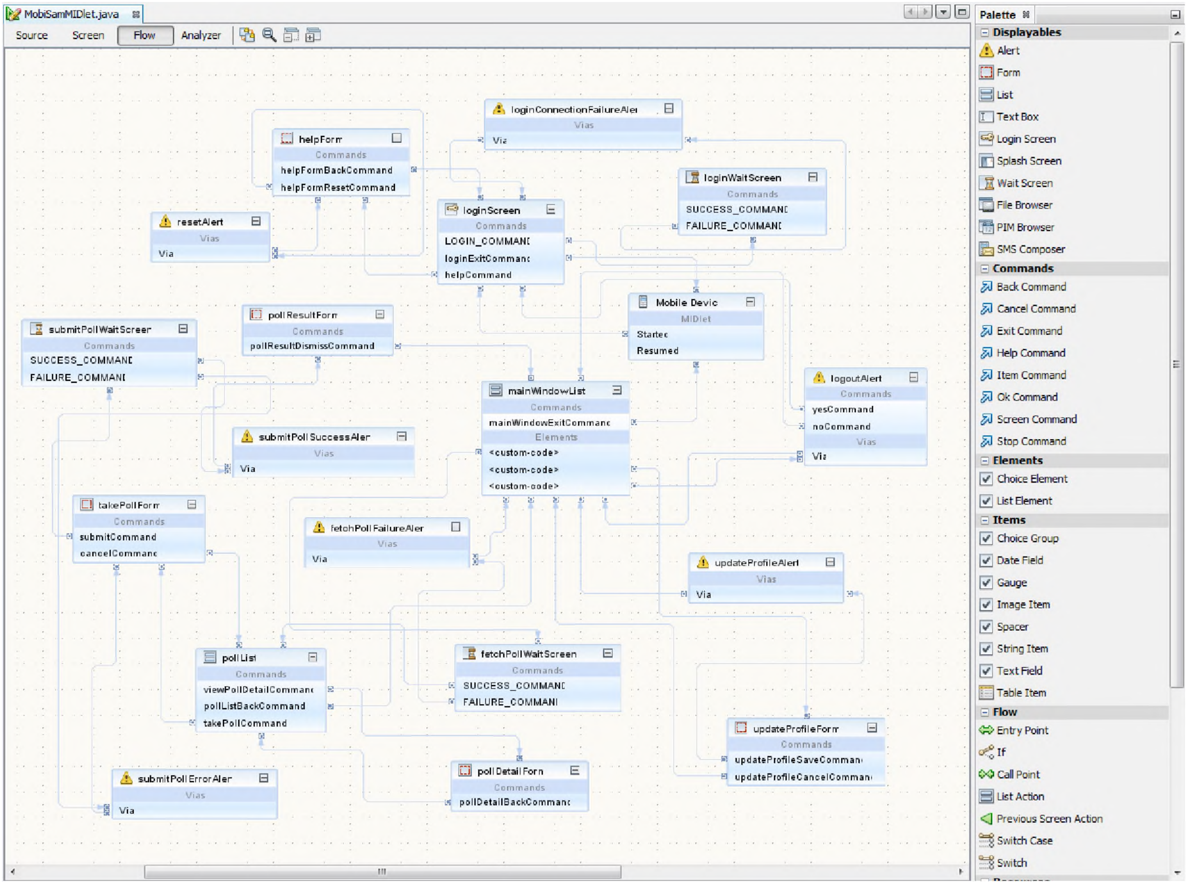


Figure A.4: NetBeans Visual Mobile Designer

Connectivity All interactions between the Java ME client application and backend server are performed using the MobiSAM REST API (see Section A.1.2). The Java ME CLDC Generic Connection Framework (GCF) provides a simplified, lightweight API for performing network related operations. Both the `Connector` class and `HttpConnection` interface form part of the GCF and when used together, allow the developer to initiate HTTP connections between client and server. This approach was exclusively used by the Java ME client application. As the Java ME platform does not provide support for the JSON data-interchange format (see Section 2.5.5), the external library JSON ME was used which provides a subset of the full JSON Java API.

Visualisation Due to the limited display size and resolution of many feature phones, particular attention was given to ensuring that visualised poll responses remained legible on these devices. As such, the ChartComponent[1] library was leveraged for visualisation purposes. The library, designed for use on low-end feature phones, offers support for MIDP 2.0 devices and above, and is capable of drawing line, bar and pie charts. A simple bar chart was chosen to depict poll

[1]J2ME ChartComponent http://www.beanizer.org/site/index.php/en/Software/J2me-ChartComponent-ver-1.5.html

responses within the Java ME client application. Chart interactivity was not offered due to library limitations.

Localisation Localisation is the process of translating an application's UI components to support different languages. The Java ME platform does not provide any specific techniques in which to achieve this, instead allowing the developer to choose a fitting implementation. To this end, the Nokia Developer website served as a valuable reference, providing many 'best practise' examples in its user-contributed *wiki*[2]. The technique implemented in the Java ME client application makes use of application attributes, represented by key-value pairs stored within its application descriptor (`.jad`) file. Each attribute represents a UI component's localised `String` and is added to the application descriptor file by supplying both an attribute name (for example, 'login-en') and a corresponding value (for example, 'Log in'). ISO-639-1 language codes were used to suffix attribute names, hence indicating which language they represent [120]

A UI component's localised `String` is retrieved by calling the method `getAppProperty(String string)` which returns the corresponding `String` stored within the application descriptor. For example, calling `getAppProperty("login-en")` returns the `String` 'Log in' while calling `getApp Property("login-af")` returns the Afrikaans equivalent: 'Teken in'. The benefit of this approach is that additional languages may be easily added via the NetBeans project properties, avoiding any additional code. Furthermore, separate application descriptor files may be created with each file representing a different locale. By offering only the required application descriptor file during download, data consumption may potentially be reduced.

Data persistence As the Java ME client application was required to preserve its settings from one invocation to the next, data persistence was implemented. Java ME provides a Record Management System (RMS) API which is used to achieve this functionality. The `RecordStore` class, found in the `javax.microedition.rms` package, allows developers to create a record store in which a collection of records can exist. While the `RecordStore` class may be instantiated to represent a single record store, it also provides several useful static methods.

Each record has its own unique identifier which is used for retrieval, deletion and updating purposes [211]. Three separate record stores are used by the Java ME client application. The first instance, `applicationKeyRecordStore`, stores the unique application key. The second, `accessCodeRecordStore`, stores the access code returned by the MobiSAM backend server, while `currentLanguageRecordStore` stores a user's preferred language (represented by the `String` en, xh or af). The `applicationKeyRecordStore` and `accessCodeRecordStore` together allow a user to remain logged in between client application invocations. All record stores are deleted when a user logs out of the client application.

[2] Localization on Java ME http://developer.nokia.com/community/wiki/Category:Localization_on_Java_ME

BlackBerry OS

Development environment The Eclipse IDE, BlackBerry Java plug-in for Eclipse and BlackBerry SDK, packaged as a single installer, was downloaded from the BlackBerry Developer website and installed on the development workstation. The BlackBerry OS 5.0 SDK was chosen in an attempt to support older BlackBerry devices, while at the same time supporting newer models via application backward compatibility. The necessary library components, Java runtime binary implementation and device simulator came bundled with the SDK. Application debugging and testing was conducted on the device simulator as well as a BlackBerry 9800 Torch handset, running version 6.0 of the BlackBerry OS.

User interface While the NetBeans IDE provides a 'drag and drop' UI editor, Eclipse instead requires developers to construct BlackBerry application UIs programmatically, using Java code. While this technique allows for a great deal of flexibility, it has the potential to be both cumbersome and time consuming, especially as an application grows in complexity. Two APIs exist for constructing BlackBerry application UIs: the MIDP UI API and the BlackBerry UI API. The former is used for developing applications which run on various MIDP-compatible devices (including BlackBerry smartphones), while the latter is used when developing applications exclusively for the BlackBerry platform. As the prototype client application targeted the BlackBerry OS, the BlackBerry UI API was used.

Fundamental to the UI implementation process, the `net.rim.device.api.ui` package contains three classes used by all UI applications: `Field`, `Manager` and `Screen`. The `Screen` class represents the main UI entity in a BlackBerry application, with only one `Screen` being displayed at a time. `Screen`s are 'pushed' onto the display stack when shown and 'popped' off when closed, a technique which was used for navigation within the BlackBerry prototype client application. Similar to the Java ME platform, the BlackBerry UI API provides several basic UI building blocks known as components. These include `ButtonField`, `DateField`, `LabelField` and `TextField`. In addition, the `net.rim.device.api.ui.container` package contains several classes used to arrange the layout of UI components. A combination of these UI components and layout managers were leveraged when implementing the client application.

Connectivity As BlackBerry OS supports MIDP 2.0, connectivity implementation was identical to that of the Java ME prototype client application, described in Section A.2.4. Although BlackBerry OS 6.0 and above provide native JSON support, the JSON ME library was used to allow support on version 5.0 of the platform. `JSONObject` and `JSONArray` instances were used to store backend server responses, while their built-in methods allowed easy access to the JSON-formatted data.

Visualisation The Google Image Chart API, discussed in Section 2.8.3, was used to visualise poll responses within the BlackBerry client application. Although the API supports a variety of chart types, the client application only leverages a basic bar chart. Special attention was paid to creating a strong contrast between bar chart elements. In addition, a legend is provided allowing the user to easily identify different categories. As with the Java ME client application, interactivity features were absent due to lack of library support.

Localisation The BlackBerry Java plug-in for Eclipse provides built-in localisation functionality, relieving developers from having to provide an implementation. `String`s may be localised through the creation of individual resource files (one for each language), hence avoiding the need to modify application code. Information and resources for a specific locale are stored in a `ResourceBundle` object. A collection of `ResourceBundle` objects are then contained in a `ResourceBundleFamily` object which an application then uses to display appropriately localised UI components. When a BlackBerry application is compiled, a separate `.cod` file is produced for each `ResourceBundle` object. These `.cod` files may then be installed on a BlackBerry device alongside the other necessary application files, depending on the language required [18]. During localisation of the BlackBerry client application, three resource files were created with each representing English, isiXhosa and Afrikaans. When specifying language resources, each resource content file name must match that of the resource header file, trailed by an underscore and the appropriate language code. For example, isiXhosa corresponds to the language code xh while Afrikaans is represented by af. The entire set of language codes are specified in ISO-639-1 [120].

Data persistence The BlackBerry platform provides two techniques in which developers may implement data persistence. The first approach is to use the regular *RMS* API provided as part of the Java ME MIDP 1.0 specification and described in Section A.2.4. Alternatively, developers may leverage BlackBerry Persistent Store. This approach was taken when implementing the BlackBerry client application and has the advantage of being able to store any object type, including complex data structures [51]. Unlike the *RMS*, serialisation routines are also provided. The `PersistentObject` class represents an object being stored in `PersistentStore`. The number of `PersistentObject` instances which may be stored is limited by the device's memory, though the BlackBerry client application makes use of only one. All BlackBerry client application preferences are stored in a `Hashtable` which itself is encapsulated within a `PersistentObject` instance and written to the `PersistentStore`. The client application needed to be digitally signed before gaining access to BlackBerry Persistent Store functionality on actual devices.

A.3 Cross-platform MobiSAM client application technical implementation

This section provides an overview of the technical implementation details relating to the cross-platform MobiSAM client application. The development environment (Appendix A.3.1), UI and navigation (Appendix A.3.2), as well as visualisation (Appendix A.3.3) are first described. Followed by additional miscellaneous features (Appendix A.3.4). Finally, modifications made to the client application in preparation for the usability evaluation are discussed (Appendix A.3.5). It is important to note that the implementation of the client application follows a controller pattern, as dictated by the CN1 framework [5]. As such, most functionality is contained within a single monolithic `StateMachine` class.

A.3.1 Development environment

The NetBeans IDE was chosen for development due to its seamless integration with the CN1 framework. As CN1 leverages Java as its implementation language, the JDK 1.7 was installed on the development workstation. No platform specific SDKs or libraries needed installing as all application builds take place on the CN1 build servers (see Section 2.7.2). The bundled device simulator, capable of accurately emulating all supported platforms, proved indispensable for debugging and testing purposes and was used extensively during development.

A.3.2 User interface and navigation

CN1 allows developers to create application UIs in much the same way as Java ME: either programmatically using Java code or via the CN1 designer tool. While programmatically constructed UIs allow for greater control, they are ill-suited for large applications due to potential organisation and maintenance issues. The CN1 designer was exclusively used when constructing client application UIs.

UI components Similar to Java ME and BlackBerry OS, components represent the fundamental UI elements within the CN1 framework. The most basic UI component is a `Container` which serves as a base class for many other components. The CN1 designer includes a variety of core components (such as `Button`s, `Label`s, `TextField`s, `RadioButton`s and `CheckBox`es) as well as more elaborate components (such as `Picker`s and `WebView`s). In addition, CN1 provides two components into which other components are placed: `Form` and `Dialog`. Only a single `Form` may be shown at any given time, while the UI components contained in it are arranged according to its layout manager. Every screen within the client application is represented by a `Form`. A

`Dialog`, on the other hand, is a special type of `Form` which occupies only a portion of the screen. These are also used throughout the client application, when displaying alerts and prompts.

Layout managers As previously hinted at, layout managers are used to position UI components with a particular `Container`, allowing for the construction of UIs which automatically adapt themselves to different display resolutions and orientations. CN1 provides six different layout manager types, a combination of which were used when constructing the client application UIs.

Actions and events Actions are performed in CN1 using events, of which five different types exist: action event, on create, before show, post show and exit form. Action events, for example, are bound to particular UI components such a `Buttons`, allowing their associated action methods to be invoked when pressed. The remaining event types are bound to particular `Forms`. The before show event, for example, being triggered *before* its associated `Form` is shown while the exit form event is triggered when a user navigates away from a particular `Form`. A combination of different event types were used extensively when implementing the client application. Pressing the 'Log in' button on the `LoginForm`, for example, triggers an action event which causes a network connection to be initiated. The client application then contacts the MobiSAM backend server and attempts to authorise the user. Should the user's credentials prove valid, the *Home* screen `Form` is displayed, if invalid, a `Dialog` is shown alerting the user.

Navigation Navigation between different screens is achieved through the `showForm(String resourceName, Command sourceCommand)` method which 'pops' the requested `Form` onto the display stack. Alternatively, the CN1 designer may be used to implement navigation between different `Forms` by specifying different action `Commands`. Navigating to a new `Form` causes the previous `Form` to be released from device memory, thereby saving device resources.

A.3.3 Visualisation

This section provides a overview of the techniques used when implementing the cross-platform MobiSAM client application's visualisation capaiblities.

Plotting library selection

Before a plotting library was chosen, those supported by CN1 were first identified. A thorough review of the CN1 website and discussion board was conducted, in which three prominent libraries were highlighted: Google Charts (see Section 2.8.3), AChartEngine (see Section 2.8.3) and Flot (see Section 2.8.3). As the Google Image Charts API is the only library to offer static image charts, it was chosen for use by the client application when installed on the Java ME or BlackBerry OS platforms. AChartEngine and Flot were evaluated for use of the remaining platforms. Several criteria were defined to help select the most suitable plotting library:

1. offer support for Android, iOS and Windows Phone platforms;
2. provide native rendering as opposed to JavaScript-based `Webview` components; and
3. allow for interactivity within charts.

While Flot satisfied the first and third criteria, AChartEngine met all three and was subsequently selected for use. The cross-platform client application determines which plotting library to use by calling the static method `Display.getInstance().getPlatformName()`, before navigating to the *Results* screen. If Android, iOS or Windows Phone, the AChartEngine library is used. Alternatively, the Google Image Charts API is used instead.

Google Image Charts API

The Google Image Charts API along with its basic usage is detailed in Section 2.8.3. Within the context of the client application, a pair of `ConnectionRequest` objects in conjunction with the `NetworkManager` class are used to send two HTTP `GET` requests to the Image Charts API. One requesting a pie chart and the other requesting a bar chart. The poll responses to be visualised along with their associated text labels are encoded within each request's URI. The Image Charts API returns each newly rendered chart as a PNG image file, which is stored by the client application as an `EncodedImage` object and displayed on the *Visualisation* screen `Form` (see Figure 5.7). The `getDisplayWidth()` method belonging to the `Display` object is used to match the requested image file width to that of the host device's display resolution.

AChartEngine

The AChartEngine library, introduced in Section 2.8.3, is used by the client application when installed on Android, iOS and Windows Phone platforms, enabling interactivity within chart visualisations.

Chart visualisations are rendered by following several distinct steps. Firstly, a `CategorySeries` data set is created from the returned poll results, using the `com.codename1.charts.models` package. This includes the value of each chart series, as well as its associated label description. A render object is then instantiated, from the `DefaultRenderer` class, enabling chart parameters such as font size and series colours to be defined. In the case of rendering a pie chart, a

`PieChart` object (also known as a chart view) is then instantiated, passing it the newly created `CategorySeries` and `DefaultRenderer` objects. Finally, before adding the chart to the screen, it is wrapped in a `ChartComponent` object. This is then added to the `Form` using the `addComponent()` method, before being displayed.

Interactivity is enabled in the charts by adding a listener to each of the chart series. In the case of the pie chart, a tapped segment becomes highlighted, the device display then pans and zooms into the segment (see Figure 5.8b). This is achieved by setting its corresponding series renderer's highlight member variable to `true`. Next, a `Rectangle` object in instantiated, which is used to represent the display viewport. The AChartEngine library method `zoomToShapeInChartCoords()` is then called, passing it the viewport `Rectangle`, as well as the duration of the transition. After trying various transition times, 500 milliseconds seemed most suitable.

Finally, after having highlighted, as well as panned and zoomed into the series of interest, a `Dialog` it displayed. This contains the name, size and percentage value of the selected series (see Figure 5.8c). Selecting the 'OK' button causes the `Dialog` to disappear, after which the device display returns to its previous state, providing once again an overview of the entire chart visualisation (see Figure 5.8a).

A.3.4 Miscellaneous

The implementation details of the primary portions of the cross-platform MobiSAM client application (connectivity, localisation and data persistence) are described next.

Connectivity

CN1 provides a `NetworkManager` class which alleviates the need for developers to manage separate network threads. The class is used in conjunction with a `ConnectionRequest` object, which facilitates web service requests and provides JSON parsing capabilities. Only HTTP/HTTPS connections are currently supported by CN1 due to restrictions placed on some mobile platform backends [5].

HTTP connections A `ConnectionRequest` object is used to create an HTTP connection by supplying it with the necessary URI, HTTP method type (i.e. `GET`, `POST`, `PUT`, `DELETE`), header content, along with additional information. The newly created object is then passed as a parameter to the `NetworkManager`'s `addToQueue(ConnectionRequest request)` method which queues and executes the connection asynchronously. The object's response listener is triggered once the network operation has completed, thereby notifying the client application and delivering with it the JSON-encoded response from the MobiSAM backend server.

Authenticating All network operations within the client application leverage the `NetworkManager` and `ConnectionRequest` class. When logging in using the client application, a `ConnectionRequest` object is created and initialised, providing it as parameters the request URI (`http://mobisam.net/api/associate`), HTTP method type (`POST`), content type (`application/json`) and connection timeout value (30 seconds). A JSON-encoded `String` is also embedded within the body of the `ConnectionRequest` and contains the client application key (a random 50 character hexadecimal `String`), username of the user accessing the service, as well as the SHA256 hash of the user's password keyed with the username. This data is used to associate the client application with a specific user account, allowing for future requests to the REST API. If the authentication process is successful, an `accesscode` is returned, embedded within the body of the response. This `accesscode` is stored (see Section A.3.4) and used for subsequent requests to the REST API.

Fetching available categories `ConnectionRequest` objects are used in a similar manner when querying other REST API routes, such as `GET api/categories`. This route returns all available categories which are then displayed on the client application's *Category* screen, as shown in Figure 5.5c.

In this example, a `ConnectionRequest` instance is created and provided with the request URI (`http://mobisam.net/api/categories`), HTTP method (`GET`), content type (`application/json`), connection timeout duration (30 seconds), as well as having the HTTP request header (where key=`X_MOBISAM_AUTH`, value=`accesscode`). If successful, the REST API returns all of the available categories which the requesting user has access to, along with a description of each. This data is contained within the body of the HTTP response as a JSON-encoded `String`. In addition, the REST API allows the user to request all of the available polls within a particular category.

Responding to a poll A similar process is followed when responding to a poll, albeit with a few noteworthy differences. A *custom* poll is used in this example (see Section 5.2). As with all network operations performed using CN1, a `ConnectionRequest` object is first created. The object's member variables are then initialised to the request URI (`http://mobisam.net/api/result/2/1`), HTTP method (`POST`), content type (`application/json`), connection timeout duration (30 seconds), as well as having its HTTP request header set (where key=`X_MOBISAM_AUTH`, value=`accesscode`). In this example, the request URI indicates that a poll with the identification number 2 is being responded to with choice 1. In addition, the request body contains a 64-character `checksum` value which is obtained by finding the SHA256 hash of the request URI keyed with the client application key. Upon successful submission, the REST API returns the Grahamstown-wide responses for the given poll, within the body of the response. These responses are then visualised, the process of which is described next.

Localisation

Support for isiXhosa, English and Afrikaans is provided by the cross-platform client application, the technical implementation details of which are provided next.

User interfaces The CN1 designer was used to localise client application UIs. First, a new resource bundle was created which acts as a large key-value pair dictionary, mapping a `String` from one language to another. Next, two additional locales were added to the resource bundle: isiXhosa (xh) and Afrikaans (af). ISO-639-1 language codes are used to specify different locales [120]. The CN1 designer was then used to automatically extract the text to be localised from each UI component. Appropriate isiXhosa and Afrikaans translations were obtained for each component and then manually entered into the corresponding field, with each bar representing a different locale, as shown in Figure A.5.

Key	af	en	xh
Any other information you would like to add:	Enige ander informasie wat u wil byvoeg:	Any other information you would like to add:	Nayiphi na into ofuna ukuyangeza apha:
App version:	App weergawe:	App version:	XH App version:
Are you sure you want to sign out?	Is jy seker jy wil uit teken?	Are you sure you want to sign out?	Uqinisekile ukuba ufuna ukuphuma?
At the following address:	By die volgende adres:	At the following address:	Kule dilesi ilandelayo:
Back	Terug	Back	XH Back
Burst pipe	Gebarste pyp	Burst pipe	Umbhobho ugqabhukileyo
Calculating address...	Berekening van adres...	Calculating address...	XH Calculating address...
Camera	Kamera	Camera	Ikhamera
Cancel	Kanselleer	Cancel	Rhoxisa
Check for updates	Kyk vir opdaterings	Check for updates	Khangela izihlaziyo
Checking for updates...	Soek vir opdaterings...	Checking for updates...	Sikhangela uhlaziyo...
Close	Sluit	Close	Vala
Comments or feedback? Write to the MobiSA...	Kommentaar of terugvoering? Skryf vir die M...	Comments or feedback? Write to the MobiSA...	Inkcazelo kwimpendulo? Bhalela kwiqela lakw...
Done	Klaar	Done	XH Done
Electricity	Elektrisiteit	Electricity	Umbane
Enter address here	Vul adres in hier	Enter address here	Fakela idilesi
Enter any details here	Gee enige besonderhede hier	Enter any details here	Naziphi na iinkcukhacha apha
Enter comments or feedback here	Verskaf kommentaar of terugvoer hier	Enter comments or feedback here	Fakela inkcazelo okanye impendulo apha
Exit	Sluit	Exit	Phuma
Finding position...	Kry posisie...	Finding position...	XH Finding position...
Forgot password	Wagwoord vergeet	Forgot password	Libele inombolo eyimfihlo
Forgot password?	Wagwoord vergeet?	Forgot password?	Uyilibele Inombolo Eyimfihlo?
Full/overflowing sewerage tank	Vol/oorstromende riooltenk	Full/overflowing sewerage tank	Isitishi selindle esigcweleyo/esiphuphumayo
GPS is turned off. Please turn GPS on and try...	GPS is afgeskakel. Skakel asseblief die GPS a...	GPS is turned off. Please turn GPS on and try...	iGPS ayisebenzi. Nceda uvule iGPS azame kw...
I want to report a:	Ek wil rapporteur:	I want to report a:	Ndinqwenela ukuchaza i:
I'd like to take a photo and attach it to my re...	Ek wil 'n foto neem en dit aanheg by my verslag	I'd like to take a photo and attach it to my re...	Ndinqwenela ukufaka umfanekiso kwinkcazel...
Large pothole (more than 50cm)	Groot slaggat (meer as 50cm)	Large pothole (more than 50cm)	Umngxuma omkhulu osendleleni (ongaphezul...
Leaking sewerage pipe	Lekkende rioolpyp	Leaking sewerage pipe	Imibhobho yelindle evuzayo
Log out	Teken uit	Log out	Phuma
No	Nee	No	Hayi
OK	OK	OK	Kulungile
OK, send report	OK, stuur verslag	OK, send report	Kulungile thumela ingxelo
Password	Wagwoord	Password	Inombolo Eyimfihlo
Please Sign in to Your Account	Teken in aseblief	Please Sign in to Your Account	Nceda ungene usebenzisa i-akhawunti
Please enter your email address:	Tik/verskaf asseblief jou e-pos adres:	Please enter your email address:	Nceda ufake idilesi yakho yomoya:
Please enter your username and password.	Verskaf asseblief u gebruikersnaam en wagw...	Please enter your username and password.	Nceda ufake Igama Olisebenzisayo ne Nomb...
Please indicate the whereabouts of the probl...	Dui asseblief die area van u probleem aan.	Please indicate the whereabouts of the probl...	Nceda ucacise ukuba yenzeka ndawoni na le ...
Please note	N.B.	Please note	Nceda ukhumbule
Power outage	Kragonderbreking	Power outage	Ukumka kombane
Problem with my electricity meter reading	Probleme met my elektrisiteitmeterlesing	Problem with my electricity meter reading	Iingxaki ngokufunda ibhokisi yam yombane
Problem with my water meter reading	Probleem om my watermeter te lees	Problem with my water meter reading	Ingxaki ngokufunda ibhokisi yam yamanzi
Remember me	Onthou my	Remember me	Ndikhumbule
Report a problem	Rapporteur 'n probleem	Report a problem	Chaza ingxaki
Report problem with electricity	Rapporteer probleem met elektrisiteit	Report problem with electricity	Chaza ingxaki yombane
Report problem with road	Rapporteer probleem met pad	Report problem with road	Chaza ingxaki ngendlela yemoto
Report problem with sanitation	Rapporteer probleem met sanitasie	Report problem with sanitation	Chaza ingxaki ngezangasese

Figure A.5: Localisation of client application UIs using the CN1 designer tool

When the client application is first launched, the English locale is loaded from the resource bundle into device memory by calling `UIManager.getInstance().setBundle(res.getL10N("Localization", "en"))`. Should the user change their preferred language during runtime, a new locale is loaded into memory, replacing the previous locale. Data persistence ensures that the user's preferred language is retained from one invocation to the next. Although most UI component text is automatically localised, some components require manual localisation by calling `UIManager.getInstance().localize(String key, String defaultValue)`. This method returns the localised `String` from the resource bundle should it exist, otherwise `defaultValue` is returned.

MobiSAM backend server The user's preferred language not only ensures that all client application UI text appears in their chosen language, but also indicates to the MobiSAM backend server which version of the service delivery poll question to send to the user. As shown in Figure A.6, when creating a new service delivery poll via the MobiSAM website, the MobiSAM administrator specifies the poll question text (for example, 'How is your water pressure today?') separately for each supported language. The backend server then sends the correct version based on the user's preferred language. This approach allows for the easy addition of languages when required.

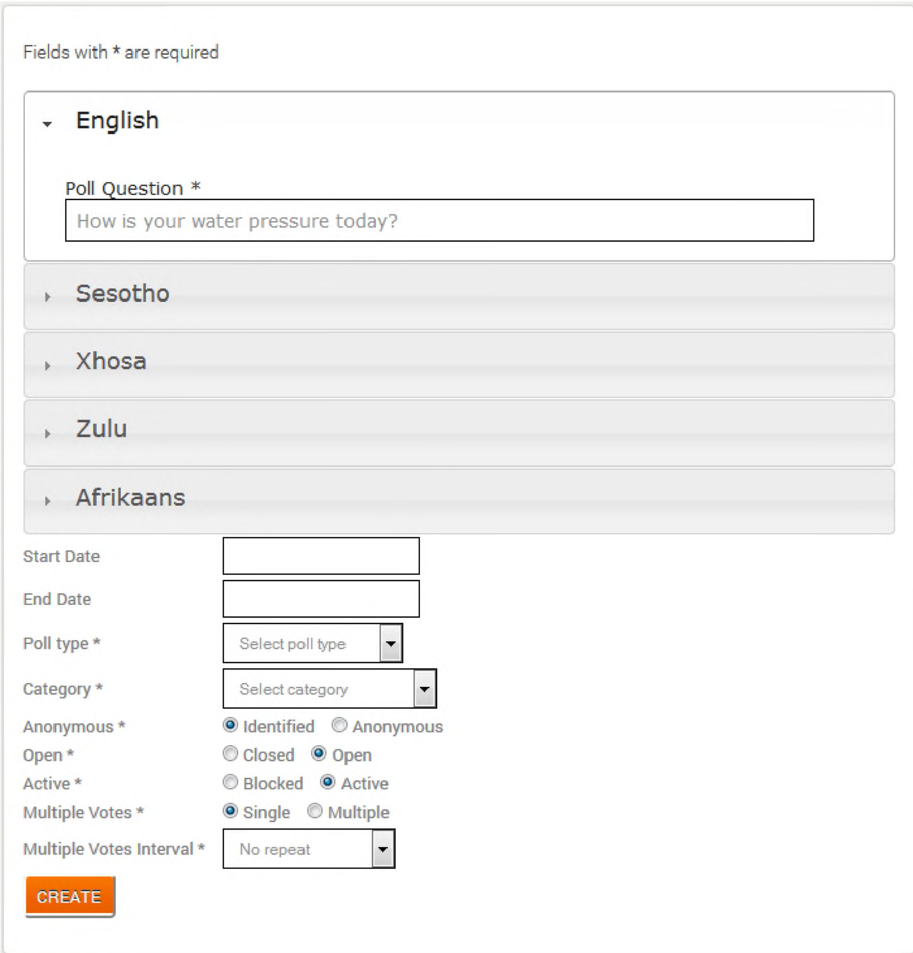


Figure A.6: MobiSAM web-interface: specifying different versions of the same question

Data persistence

Data persistence is provided by the `com.codename1.io.Preferences` class which enables the storage, retrieval and deletion of simple variables such as `Boolean` and `String` values. When storing a preference value, a key is specified along with its corresponding value, for example, `Preferences.set("username", "guest")`. This value may then be retrieved by using its key:

`String s = Preferences.get(“username”)`. If the provided key does not exist, `null` is returned. Stored session information includes the client application's 50 character application key as well as the assigned `accesscode`. When the user logs out of the client application, the preference named `keepMeLoggedInCheckBox` is set to false, ensuring that the *Login* screen will be shown next time the application is launched.

A.3.5 Usability evaluation modifications

Several modifications were made to the cross-platform client application in order to keep track of different task performance measures employed during the usability evaluation (see Chapter 6). Recorded measures were:

- the number of login attempts;
- task duration; and
- number of button presses made while performing a task.

In addition, all tasks were simulated in an attempt to eliminate confounding factors such as variable network latency and transfer rate when communicating with the MobiSAM backend server.

The logging of each task is started manually by pressing the '#' key on the feature phone or the hardware menu button on the smartphone, which brings up the *Task logger* screen. Here, the current task number being performed is selected from a `ComboBox`, and 'Start' is pressed. A countdown timer `Dialog` is then shown (see Figure A.7) which alerts the participant that the evaluation is about to start, disappearing after three seconds. Logging continues until the participant has performed the task as expected. For example, submitting a response to a particular poll or changing their preferred language. Gathered task performance measures were used when analysing the results of the usability evaluation (see Section 6.8).

Figure A.7: Cross-platform client application: *Countdown* dialog

Login attempts The number of log in attempts made by the participant when performing a task was logged. This was implemented using a simple `int` variable which is incremented each time the participant fails to log in to the client application.

Task duration The time taken for the participant to perform a particular task as expected was also logged. This was achieved by logging the task start and end times. The start time was then subtracted from the end time, along with the cumulative duration the loading `Dialog` was displayed. This resulted in the total duration taken to perform a given task as expected.

Button press count Once again, an `int` variable was used to keep track of the number of button presses made by the participant while performing a task. Only button presses which resulted in navigation between different `Forms` were logged. For example, pressing the 'Update profile' menu option does count as a button press whereas changing the preferred language from the `ComboBox` within the screen does not.

A.4 Updated cross-platform MobiSAM client application technical implementation

This section details the technical implementation modifications made to the cross-platform client application. Changes to the UI and navigation are first described, followed by those made to the client application's connectivity and visualisation.

A.4.1 User interface and navigation

***Login* screen** Design changes were first implemented on the *Login* screen. First, the `Title` font theme constant was set to bold, ensuring that the title text appears bold throughout the client application. The `TextField` theme constant was then changed so as to bold the currently selected item. Finally, the 'Keep me logged in' `CheckBox` tick was also moved to the right hand side of the text.

***Home* screen** Each menu option on the *Home* screen is represented by a `Container` housing two `TextAreas` (used to represent `Line1` and `Line2`) and a `Button` (used to listen for presses via its `ActionListener`). When combined, these form a `MultiButton` component. In addition, an `icon` is included alongside each menu option, and is implemented by setting each `Button`'s `icon` property. Accordingly, an additional menu option was added to the top of the *Home* screen: 'Report a problem'. Furthermore, the 'Exit' menu option was removed and instead implemented as a `Command`, appearing as a soft key on Java ME, or after pressing the hardware menu button on the remaining platforms.

***Report category* screen** As with the *Home* screen, each *Report category* screen menu option is implemented using a `Container`, `TextArea` and a `Button` component, forming a `MultiButton`. Similarly, `icons` have been included within each menu option (by setting each `Button`'s `icon` property), helping clarify the purpose of each. Unlike the *Home* screen menu, only a single `String` of text is included within each `MultiButton`. As such, five `MultiButton` menu options were implemented ('Water', 'Electricity', 'Roads', 'Sanitation' and 'Other'). The screen's `Title` component text was set to 'What type of problem is it?'.

***Available reports* screen** After selecting the desired report category (for example, 'Water'), all available reports corresponding to the chosen report category are retrieved from the backend server and displayed. As with previous screens, each available report is represented by a `Container` housing a `TextArea` and `Button`, which together form a `MultiButton`. An arrow `icon` is included alongside each available report, implemented by setting the `Button`'s `icon` property.

***Reporting* screen** The chosen report's text is shown in the `Title` of the *Reporting* screen (for example, 'I want to report a water outage'). Next, a `SpanLabel` is used to present the instruction text: 'At the following address:'. A `TextField` is positioned below this, allowing the user to enter an address. A `Label` appears to the left of the `TextField`, which has its `icon` property set to a stylistic depiction of the World (representing position/address). Depending on the platform, either a `Command` or `Button` is provided to 'Send report'.

***Category* screen** After selecting 'Take a poll', the user is presented with a list of the available categories which are retrieved from the backend server and displayed. As with other screens, each available category is implemented using a single `TextArea` and `Button`, to form a `MultiButton`. Each `Button`'s `icon` has been set to an arrow image.

***Available polls* screen** Once a category of interest has been chosen, the user is presented with a list of available polls within the category, fetched from the backend server. Again, each poll in the list is represented by a `MultiButton`, which is composed of two `TextArea`s (representing `Line1` and `Line2`) and a `Button` component. The first line of the `MultiButton` (`Line1`) is set to the poll question text (for example, 'How is your water pressure today?') while the second line (`Line2`) reflects the end date of the poll (for example, 'Ends: May 31, 2014'). As with previous screens, `icons` are implemented by setting each `Button`'s `icon` property.

***Poll response* screen** The `Title` component now provides the text 'Select a response', instead of providing a 'breadcrumb' trail of actions performed. Below, the poll question text appears as a `SpanLabel`, followed by either a list of predefined `RadioButton` options (which collectively belong to a `RadioButtonGroup`) or a `TextField`, depending of the poll type (see Section 5.2). The poll type is determined by the client application by checking whether the JSON-formatted response from the backend server contains an `options` field. If so, each option is displayed as a `RadioButton`. The currently selected `RadioButton` option appears in orange, bold text. In the event that no `options` field is included in the server response (indicating that the poll type is *text response*), a `TextField` is included on the screen, into which the user may enter their open-ended response. Text responses are limited to 255 characters. Depending on the platform, the 'Submit' option appears either as a soft key `Command` or as a regular `Button`.

***Update profile* screen** The user may update their current suburb using the `ComboBox` component. The component has its `ListItems String` property value set to the respective suburbs of Grahamstown. In addition, the user is now prompted by a `Dialog` to save changes before returning to the *Home* screen.

A.4.2 Visualisation

This section provides a brief overview of the techniques used when implementing the updated cross-platform MobiSAM client application's visualisation capaiblities.

Static chart

The Google Image Chart API was leveraged when implementing the image charts, as described in Appendix A.3.3. There are, however, a few notable differences. First, a single `ConnectionRequest` object is created due to the fact that only a pie chart visualisation is now included within the client application. The *custom* poll responses to be visualised along with their associated text labels are encoded within each request's URI. The `chdlp` chart legend parameter was modified so as to position the legend below the chart when a portrait device resolution (width $<$ height) is detected, while to the right of the chart when a landscape device resolution is detected (width $>$ height). Chart sectors are also sorted by size (from largest to smallest) within the URI. Colours are chosen sequentially from an array of possible colours, so as to ensure the colour of adjacent sectors do not clash.

Interactive chart

A simple `Label` containing the text 'Tap chart for details' was positioned above the chart component. The `Label` has its `icon` property set to an image indicating that the chart may be 'tapped'.

The chart's `renderer` object method `setDisplayValues(true)` is called so as to display the corresponding percentage value within each chart sector.

A.4.3 Connectivity

As a result of the newly introduced reporting feature, several additional *text response* polls were created on the backend server, where each poll represents an available report (for example, 'I want to report a water outage'). Reports are sent in an identical fashion to what is described in Appendix A.3.4, albeit a *text response* is answered instead of a *custom* poll type.

A.5 Cross-platform MobiSAM Report client application technical implementation

This section provides an overview of the technical implementation details relating to the cross-platform MobiSAM Report client application. The UI and navigation are first described, followed by GPS functionality, reverse geocoding, photo capture and connectivity.

A.5.1 User interface and navigation

***Home* screen** The *Home* screen consists of a `Form` containing two `Container`s. The first `Container` holds the MobiSAM Report logo. The `Container` below holds four `icon`s arranged in a 2x2 `GridLayout`, each represented by a `Button` component. A stylistic image and text description was added to each `Button` using its `icon` and `text` properties. Depending on the platform used, two soft key `Command`s are also included on the *Home* screen: 'Sign out' and 'Settings'. The former signs the user out of their account, navigating them back to the *Sign in* screen, while the latter navigates the user to the *Settings* screen.

***Reporting* screen** The *Reporting* screen consists of a `Form` and multiple `Container`s. Screen content is created dynamically, depending on the report category selected from the *Home* screen. A predefined list of possible issues relating to the chosen report category is provided at the top of the screen. Each issue is represented by a `RadioButton` which, together, form part of a larger `RadioButtonGroup`.

Next, a `TextField` is provided which is used to input the address or position of the issue being reported. The address may be entered manually by the user or obtained using the handset's GPS, by ticking the `CheckBox` 'Use GPS to calculate my position instead' (see Section 5.8.3). When ticked, an `ActionListener` is triggered which causes the handset to fetch its position via

GPS (when supported). The retrieved coordinates then undergo reverse geocoding (see Appendix A.5.2).

Another `TextField` is positioned below, into which the user may enter any additional details relating to the report. Finally, a `CheckBox` in provided, allowing the user to capture and include a photo within the report (see Section 5.8.3). The two soft keys, 'OK, send report' and 'Back' are implemented as `Commands` on Java ME and as regular `Buttons` on the remaining platforms.

***Report sent* screen** The *Report sent* screen was implemented using a `Form` and two `Containers`. The first Container holds the image displayed after successfully submitting a report (see Figure 5.10d), while the second contains a `SpanLabel` which informs the user that the report was submitted. The 'Done' soft key is represented by a `Command` on Java ME and a `Button` on all other platforms.

***Settings* screen** The *Settings* screen consists of a `Form` and a single `Container`. The `Container` holds four UI components: two `SpanLabels` and two `ComboBoxes`. Each `SpanLabel` includes text which describes the purpose of the `ComboBox`. The first `ComboBox` is used to represent the user's preferred language and contains three `ListItems`: English, isiXhosa and Afrikaans. The second `ComboBox` is used to specify the user's current suburb and contains the name of each suburb within Grahamstown. Both the preferred language and current suburb settings are automatically saved when modified, by triggering the `ActionListener` of each `ComboBox`. The newly selected preferred language is reflected immediately, with all the necessary UI component text being updated. This avoids the user having to manually save changes before navigating back to the *Home* screen.

A.5.2 Miscellaneous

The implementation details relating to the primary portions of the updated cross-platform MobiSAM client application (GPS functionality, reverse geocoding, photo capture and connectivity) are presented next.

GPS functionality

The `LocationManager` class provides access to the device's built-in GPS, allowing a handset's position to be determined either synchronously or asynchronous. The former is 'blocking' and may take some time to return with an accurate position, while the latter makes use of a `LocationListener` which triggers once a position has been obtained, thereby informing the application about the coordinate change. All GPS requests within the MobiSAM Report are performed using the latter technique.

Reverse geocoding

The static method `String getSuburbName(Coord point)` belonging to the `SuburbReverse-GeocoderService` class, determines which suburb the passed coordinate falls into. The method makes use of an intersection algorithm in order to determine whether a given coordinate falls within a particular set of coordinates (otherwise known as a polygon). If so, the name of the suburb into which the coordinate falls is returned as a `String`, and appended to the address `TextField` component on the *Reporting* screen. If the coordinate does not correspond to any known suburb, an empty `String` is returned. In an attempt to improve the user experience, positioning requests timeout after 30 seconds. Once expired the user is shown a `Dialog` requesting that they enter the address manually instead.

Photo capture

The CN1 `Capture` class provides a static method, `String capturePhoto(int width, int height)`, which is used to capture images. The method returns a path to the recently captured image file. Before uploading, the captured image file is converted to a Base64 encoded `String` which is included within the body of the HTTP request when submitting the report.

Connectivity

Before newly captured images are uploaded they are first converted to their Base64 `String` representation, as required by the MobiSAM REST API. This is achieved using the `encodeNoNewline (byte[] in)` method, provided by the `com.codename1.util.Base64` class. The resulting Base64 encoded `String` is then embedded within the body of the `ConnectionRequest` and uploaded. The position coordinates indicating the whereabouts of the report, as well as the human-readable address (either entered manually or obtained via reverse geocoding) are included alongside.

A.6 MobiSAM SMS gateway

This section presents the design and implementation of the MobiSAM SMS gateway. The SMS gateway consists of two components: a MOBITEK S63 SMS modem (see Section 2.5.2) and a Windows-based application. The application interfaces with the SMS modem, which receives service delivery reports via the mobile network in the form of traditional SMS messages. The application then relays these service delivery reports to the backend server where they are displayed chronologically via the MobiSAM web-interface. This ensures that all residents who own (or have access to) a mobile phone are able to submit service delivery complaints, regardless of the handset's capabilities.

A.6.1 Requirements and specifications

While Makana Municipality highlighted the need for residents to be able to submit service delivery reports via SMS, they provided no additional requirements, instead relying on the researcher to develop an appropriate solution. As such, several key requirements for the application were defined by the researcher, with a particular focus on creating a robust and usable system:

- provide a simple UI allowing a MobiSAM administrator to connect/disconnect the SMS modem to/from the mobile network;
- display clear information relating to the status of the SMS gateway;
- allow an administrator to modify how often the SMS gateway 'checks' (polls) for incoming messages;
- ensure that in the event of system downtime, undelivered incoming SMS reports are retained;
- alert an administrator via e-mail should any issues arise with the SMS gateway (e.g. no airtime; problem with the GSM connection to the mobile network; connectivity issues to the backend server);
- retain a detailed system log of all actions performed by the SMS gateway, written to disk; and finally,
- allow for autonomous, unassisted operation as much as possible.

Furthermore, several requirements were defined with regards to using the SMS-based service delivery reporting service:

- the user need not be a registered MobiSAM user to be able to submit service delivery reports via SMS;
- service delivery issues are entered in plain text without special formatting; and
- receipt confirmation is provided via SMS, thus helping keep the user informed.

Figure 5.2 illustrates how the SMS gateway is positioned within the MobiSAM system architecture. An SMS modem and the SMS gateway application are installed on an Internet-enabled workstation and together, listen for incoming service delivery reports via SMS. These messages are then relayed to the backend server using its *RESTful* API, where they are collated and presented to a MobiSAM administrator via the web-interface. As the SMS gateway application interacts with the backend server in an identical manner to that of the client applications, no special access rights were granted. The design and implementation of the SMS gateway application is presented next.

A.6.2 Design

This section presents a brief overview of the design of the SMS gateway application. First, the architectural pattern and UI are discussed. Application connectivity, as well as logging and data persistence are then described. Due to space restrictions, a detailed account of the technical implementation has been included as an electronic resource. Please refer to electronic Appendix A.6.3 for more information.

Architectural pattern

A software architecture pattern "describes a particular recurring design problem that arises in specific design contexts, and presents a well-proven generic scheme for its solution" [23, p.8]. As Microsoft encourage use of the Model-View-ViewModel (MVVM) pattern within event-driven programming paradigms, it was chosen for implementation of the SMS gateway application. Based on the model-view-controller (MVC) pattern, the MVVM pattern helps developers "cleanly separate the business and presentation logic of [their] application from its user interface" [67, p.1]. This separation between logic and UI helps make the application easier to test, maintain and expand upon.

User interface

The SMS gateway application consists of a single screen, shown in Figure A.8, through which all user interaction takes place. The UI presents information relating to the current state of the SMS modem (e.g. mobile network name; signal strength; total number of SMS messages sent/received) as well as providing an area where all recently performed actions are detailed (for example, connecting to the GSM network; authenticating with the MobiSAM server; and polling for incoming SMS messages). The Windows operating system UI guidelines were consulted in an attempt to maximise the application's consistency.

In addition, a selection of UI controls are provided, allowing the user to set the SMS gateway online/offline; modify the incoming SMS poll interval and COM port value; enable/disable auto scrolling of the log window as well as clear its contents. The online/offline toggle button changes colour based on the SMS gateway's current state, thus providing visual feedback to the user. Application specific help (in the form of a pop-up dialog) is offered by clicking the question mark alongside the 'COM Port' text. Furthermore, the current status of the SMS gateway ('Online', 'Offline', 'Connecting...', 'Authenticating...') is shown in the application title bar, allowing it to be monitored even while minimised.

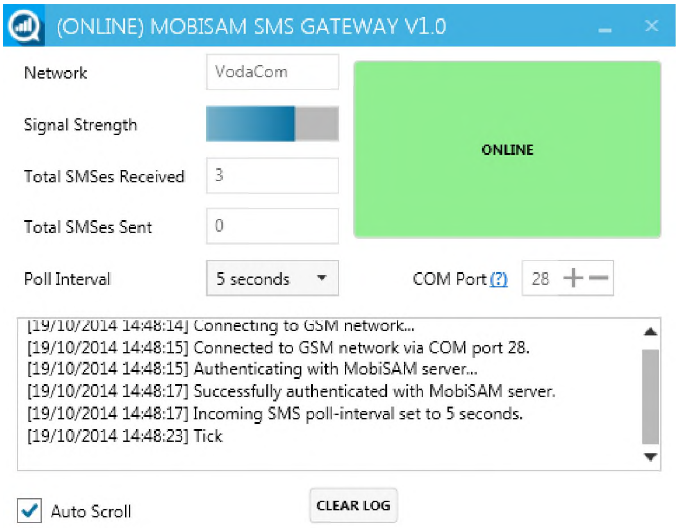


Figure A.8: MobiSAM SMS gateway application

Certain UI components are disabled depending on the status of the SMS modem, thus imposing what Norman refers to as constraints [160]. For example, it is only possible to change the 'Poll Interval' when the SMS modem is online, at all other times the UI component is disabled. Conversely, it is only possible to change the 'COM Port' when the SMS modem is offline. It should be noted that most UI controls are updated via data bindings, thereby ensuring that the most recent values are always reflected. If, for example, the received SMS count variable is incremented, this new value will be automatically reflected within the corresponding UI control.

A confirmation dialog is presented to the user if they attempt to exit the SMS gateway application before going offline, reminding them that the SMS gateway will no longer receive SMS messages. In the event that the SMS gateway is offline, all incoming SMS reports are queued on the GSM network's SMSC and only delivered to the SMS gateway once online again.

Connectivity

The application's connectivity may be viewed as two separate parts: connectivity to the GSM network (via the SMS modem) and connectivity to the MobiSAM backend server (via the host workstation's Internet connection). A detailed account of the implementation of each is provided in electronic Appendix A.6.3.

Application logging and data persistence

As the SMS gateway was developed to run unattended, a logging mechanism was implemented, which may be used for troubleshooting in the event of a problem. The application logs all performed actions, including: changes in the SMS modem's status, port number and poll interval;

incoming/outgoing SMS details; as well as any errors which may occur. In addition, the application's settings (e.g. COM port number; SMS poll interval; total SMS messages sent/received) are retained between invocations of the application. Please refer to electronic Appendix A.6.3 for a technical implementation details.

A.6.3 Implementation

This section presents the technical implementation details relating to the MobiSAM SMS gateway. First, the development environment and UI are discussed. Connectivity, as well as logging and data persistence are then described.

Development environment

Microsoft Visual Studio was used for development purposes due to the SMS modem's support for the .NET framework as well as the researcher's familiarity with the C# programming language. Before connecting the SMS modem to the development workstation, a virtual COM port driver was installed. This allowed the USB SMS modem to appear as a COM port, which was then referenced from code. Next, the MOBITEK SMS API v7 was installed. The API offers support for Visual Basic, .NET C# as well as C++ programming languages and is provided as a COM/ActiveX dynamic-link library (`.dll`) file. This `.dll` file provides an API used to perform GSM network related operations, and was referenced from within the SMS gateway application project. Next, the API needed to be referenced from within the program code, and was achieved by including `using MobitekSMSAPI7;` at the beginning of the main window's code-behind file (`MainWindow.xaml.cs`). In an effort to create a more visually appealing and usable UI, the MahApps.Metro package was installed. This package provides additional modern looking UI controls to Windows Presentation Foundation (WPF) applications.

User interface

The SMS gateway application consists of a single UI window which was constructed using Visual Studio's 'drag & drop' editor. The UI is represented by a XAML file (in this case `MainWindow.xaml`) which may be modified both programmatically or using the editor. The MahApps.Metro namespace was referenced in the opening `Window` tag in order to allow for its UI controls to be leveraged by the application. In addition, the opening `<Window...` tag was changed to `<Controls:MetroWindow...` thereby enabling a MahApps.Metro styled main window.

The `MainWindow` UI controls are arranged in a `Grid` layout panel consisting of seven rows and four columns in order to ensure uniform spacing. Rows 0, 1, 2, 3 and 4 of column 0 each contain a single Label. These Labels provide a description of each UI control appearing to their immediate right (see Figure A.8). Starting at the top of the UI, the 'Network' TextBox control displays the name of the GSM network currently being used. The 'Signal Strength' ProgressBar control displays the current reception strength. The 'Total SMSes Received' and 'Total SMSes Sent' TextBoxes indicate how many SMS messages have been received/sent. Finally, the 'Poll Interval' ComboBox control allows the user to select the frequency which the SMS modem is checked (or 'polled') for incoming SMS messages.

Rows 0, 1, 2 and 3, of columns 2 and 3 are occupied by a large Button control which allows the user to connect/disconnect from the GSM network. The Button is coloured grey when the SMS gateway is disconnected from the GSM network; disabled while connecting/authenticating; and coloured green when connected to the GSM network, and successfully authenticated with the backend server. Below a NumericUpDown control is provided which allows the user to select the COM port number to which the SMS modem is connected. This control is disabled while the SMS gateway is online.

A RichTextBox control spans the entire of row 5 and is used to display all recently performed actions, the contents of which is written to a text file. A Button is positioned directly below the RichTextBox which allows the user to clear the log window (though the contents of the text file remains intact). Finally, a CheckBox control is positioned to its left which allows the enabling/disabling of auto-scrolling.

Connectivity

Connecting/disconnecting A connection to the GSM network is created by first instantiating a Modem object. The object's method `Init(short intPort, String pin)` is then called, passing it the COM port number to which the SMS modem is connected, as well as the PIN code of the SIM card as parameters. The method returns `1` if a connection is successfully established, otherwise `0` is returned. Once successfully connected to the GSM network the application UI is updated, reflecting the GSM network name and signal strength. The incoming SMS poll frequency is also displayed and may be modified as required. The SMS modem is disconnected from the GSM network by calling the `ShutDown()` method.

Incoming SMS messages As the SMS modem does not provide any form of action listener or interrupt, it instead needs to be 'polled' for incoming SMS messages on a regular basis by calling an SMS object's `ReadSMS()` method. First, an SMS object is instantiated. A DispatcherTimer object is then created whose EventHandler method is triggered either every 5, 60, 300 or 600 seconds depending on the ComboBox component value. The `ReadSMS()` method is then called from within the EventHandler. If there is an incoming SMS message, `true` is returned, otherwise `false` is returned. The instantiated `SMS` object stores the received SMS message along with the sender's number within its `Msg` and `MN` member variables. The `TotalSMSesReceived` counter variable is incremented accordingly.

Outgoing SMS messages The SMS gateway is able to send receipt confirmations in response to every SMS message received, thereby informing citizens that their service delivery report has been noted. This is achieved by calling the `SMS` object's `SendSMSText(String strMobileNumber, String strMessage)` method, specifying the necessary parameters, including the message: 'Thank you for your SMS report. The correct municipal department has been notified and will respond within 24 hours'. Receipt confirmations are disabled by default due to their associated cost.

Backend server communication As previously mentioned, the SMS gateway application interacts with the MobiSAM backend server in an identical fashion to that of the client applications, using the MobiSAM REST API. A class named `MobiSAMWebService` has been implemented and is responsible for all interactions with the backend server. During initialisation of the SMS gateway application, a `MobiSAMWebService` object is instantiated.

Figure A.9: MobiSAM website: incoming SMS report

After the SMS gateway application has successfully connected to the GSM network, it attempts to authenticate itself with the backend server. This is achieved using the method `AuthenticateMobiSAMSMSGatewayAsyncAwait(string username, string password)` which, if successful, updates the `accessCode` member variable. This value is then used for all subsequent requests to the MobiSAM API. Incoming SMS reports are submitted to the backend server as *text response* poll types (see Section 5.2), which are then collated and displayed chronologically via the MobiSAM web-interface, as illustrated in Figure A.9. All network related operations are performed asynchronously so as not to adversely effect the application's performance.

E-mail notifications As the SMS gateway has been designed to operate unattended, it was decided to include a mechanism to alert the user should any potential problems arise. Alerts are sent via e-mail and cover three scenarios:

- The SMS modem has unexpectedly disconnected from the GSM network;
- the SMS gateway application is unable to submit a received SMS report to the MobiSAM backend server; as well as when

- there is no available airtime remaining on the SIM card used by the SMS modem.

The transactional e-mail provider Postmark[3] is used to send e-mail alerts from the alias smsgateway@mobisam.net.

A `static` class named `Utils` was constructed which contains several general utility methods, including `SendSystemStatusMailMessageAsync()` and `SendTimeoutMailMessage()`. The first method is used to alert the user in the event that the SMS modem has unexpectedly disconnected from the GSM network. This is achieved by first instantiating a new `MailMessage` object, providing it with both the user's as well as the sender's e-mail address (in this case, smsgateway@mobisam.net). The object's `Subject` and `Body` member variables are then assigned appropriate `string`s. Next, an `SmtpClient` object is created by passing the constructor an SMTP server address (smtp.postmarkapp.com) and corresponding port number (587). The delivery method and credentials are then set. Finally, the e-mail is sent asynchronously using the `SmtpClient` object's `SendMailAsync()` method.

Similarly, the `SendTimeoutMailMessage()` method is used to notify the user that the SMS gateway application was unable to submit an SMS report to the MobiSAM backend server. The SMS gateway application attempts to submit the SMS report to the backend server a total of ten times (waiting 2 minutes between each attempt) before the method is called. The method operates in an identical fashion to the previously detailed method, however, additional information is included within the e-mail report, including the contents of the SMS message which triggered the problem; the sender's number; as well as a textual description of the problem. The SMS gateway application then changes state to offline, with all subsequent SMS reports being retained on the GSM network's SMSC. In the event that the SMS modem is unable to send an SMS message due to a lack of airtime, a simple e-mail is sent to the user informing them about the issue.

Application logging and data persistence

Action logs The SMS gateway application saves to persistent storage, the details of all performed actions, including: changes in the SMS modem's status, port number and poll interval; incoming/outgoing SMS details; as well as any errors which may occur. Details are saved in a plain text (`.txt`) file within the `...\Logs\` directory of the SMS gateway application. A new file is created at the start of each month and named according to the month and year (MM-yyyy), for example, the log file for September 2014 was named `09-2014.txt`.

Appending text to the log file is achieved by calling the method `UpdateSystemLog(string str)`. The method first tries to open the current log file, however, if the file does not exist it creates it

[3]Postmark http://postmarkapp.com/

anew. This is done by instantiating a `TextWriter` object and passing it the log file's location on disk. The method `WriteLine(string value)` is then called, passing it the timestamped `string` to be appended to the log file. In addition, transaction logs are shown in the log output window (see Figure A.8) which takes the form of a `RichTextBox` UI component. This is achieved by calling the `AppendText(string textData)` method and allows the user to view, chronologically, the sequence of actions which the SMS gateway has performed since its last invocation.

Application settings The COM port number, incoming SMS poll interval as well as the total number of SMS messages sent and received are retained between invocations of the SMS gateway application. This ensures that the user does not need to reselect the COM port each time they start the application. It also provides the user with an accurate idea of how many SMS reports have been received, as well as how many receipt confirmations have been sent.

Individual values are saved as application settings, a form of persistent storage provided by the .NET framework. First, four new settings were created via the project 'Properties' menu: `COMPort`, `PollIntervalIndex`, `TotalSMSesReceived` and `TotalSMSesSent`. These settings are then updated in appropriate locations throughout the application. The COM port value, for example, is saved to application settings only once the SMS modem has successfully connected to the GSM network. This is achieved by calling the code shown in Listing 2.

Listing 2 Saving COM port value to application settings

```
// Save COM port
MobiSAMSMSGatewayWPF.Properties.Settings.Default.COMPort = COMPort;
MobiSAMSMSGatewayWPF.Properties.Settings.Default.Save();
```

Furthermore, the application retains the total number of SMSes sent/received until purposefully reset by the user. Resetting of values is done by clicking the `TextBox` UI component containing the value which the user wishes to reset (see Figure A.8). A dialog box is shown before the value is reset to zero, confirming the user's action.

Appendix B

Questionnaires

B.1 Baseline study

This page has intentionally been left blank.

B.1.1 Consent form

CONSENT FORM

Project Title Social Accountability Monitoring using Mobile Phones

Project Description In this project, the researcher hopes to determine whether mobile phones are a viable technology to facilitate increased participation in local government processes. This questionnaire aims to gain feedback regarding citizen's cellphone usage habits as well as their current level of engagement with local government. Participants will be asked to complete a series of related questions; feedback obtained will serve as a baseline for the study.

Researcher Mr Edward Reynell

- I have received information about this research project.
- I understand the purpose of the research project and my involvement in it.
- I understand that I may withdraw from the research project at any stage.
- I understand that participation in this baseline study is done on a voluntary basis.
- To the best of my knowledge I have no physical impediments that will stop me from completing this questionnaire.
- I understand that while information gained during the study may be published, I will not be identified and my personal details will remain confidential.

Name of participant ....

Signed .... **Date** ....

I have provided information about the research to the research participant and believe that he/she understands what is involved.

Researcher's signature and date ....

P.T.O.

MobiSAM Project

Rhodes University and Grocott's Mail newspaper are starting a project to begin monitoring service delivery projects in Makana (Grahamstown) municipality. Would you like to be a part of this project? Please note there is no payment for participation.

☐ Yes ☐ No

If **yes**, please provide your contact details below. Your questionnaire answers will not be linked back to you.

Name: ....

Cellphone number: ....

Address: ....

....

....

B.1.2 Baseline study questionnaire

MobiSAM Baseline Questionnaire

Please answer all the questions in the space provided or tick where appropriate.

Section 1 deals with basic demographic information, **Section 2** deals with your cellphone usage habits and **Section 3** relates to your service delivery satisfaction and current participation within local government matters.

Please note: *No personally identifiable information will be linked to your answers.*

Section 1 – Demographic Information

1. What is your gender?
   - ☐ Male ☐ Female
2. In which age group do you belong?
   - ☐ 18 – 30 years ☐ 31 – 45 years ☐ 46 – 60 years ☐ Over 60 years
3. What is your preferred language?
   - ☐ Xhosa ☐ English ☐ Afrikaans ☐ Zulu
   - ☐ Ndebele ☐ Sotho ☐ Swazi ☐ Tswana
   - ☐ Tsonga ☐ Venda ☐ Other (please specify)

   Other: ....
4. Which of the following are you able to do in your preferred language? (tick all that apply to you)
   - ☐ Read ☐ Write ☐ Speak
5. How long have you lived within the Makana (Grahamstown) municipality?
   - ☐ Less than 1 year ☐ 1 – 5 years ☐ 6 – 10 years ☐ Over 10 years
6. What is your highest level of education?
   - ☐ Primary school ☐ Some high school ☐ Matric ☐ University degree
   - ☐ Other (please specify)

   Other: ....
7. What best describes your current employment status?
   - ☐ Working full-time ☐ Working part-time ☐ Temporarily employed
   - ☐ Unemployed ☐ Temporarily unemployed ☐ Homemaker full-time
   - ☐ Student ☐ Self employed
   - ☐ Other (please specify) ☐ Retired/pensioner

   Other: ....

8. Have you ever worked for Makana municipality?

☐ Yes ☐ No

If **yes**, please provide details (duration, job position etc.): ....

....

9. What is your monthly income? (after tax)

☐ No income ☐ R1 – R400 ☐ R401 – R800
☐ R801 – R1 600 ☐ R1 601 – R3 200 ☐ R3 201 – R6 400
☐ R6 401 – R12 800 ☐ R12 801 or more ☐ I don't know

Section 2 – Cellphone Usage Information

10. Do you own a cellphone?

☐ Yes ☐ No

11. If **no**, do you have access to a cellphone that you can use regularly?

☐ Yes ☐ No

If **no**, please skip to **Section 3**.

12. How many years have you used a cellphone?

☐ 1 – 5 years ☐ 6 – 10 years ☐ Over 10 years

13. What brand is your current cellphone?

☐ Nokia ☐ Samsung ☐ Motorola ☐ Sony Ericsson
☐ LG ☐ BlackBerry ☐ Other

Model: .... (Please ask for help if you don't know)

14. Who is your cellphone network operator?

☐ Vodacom ☐ MTN ☐ Cell C ☐ Virgin Mobile
☐ Telkom Mobile ☐ Other (please specify)

Other: ....

15. Is your cellphone on pre-paid (e.g. Pay-As-You-Go) or contract?

☐ Pre-paid ☐ Contract

16. If **pre-paid**, please specify how much airtime you spend per week, on average.

☐ Less than R5 ☐ R5 – R15 ☐ R16 – R30 ☐ More than R30
☐ I don't know

17. If **contract**, please specify how much you spend per month.

R .... ☐ I don't know

18. What do you use your cellphone for? Please rank in order of most used (**1** – most, **2** – moderate, **3** – least, **blank** – never)

☐ Voice calls ☐ SMS ☐ Data services (WhatsApp, Facebook, MxIT, browsing etc.)

19. Do you use any of the following messaging services? (tick all that apply to you)

☐ SMS ☐ BBM (BlackBerry) ☐ MxIT
☐ WhatsApp ☐ Other (please specify)

Other: ....

20. How often do you use these messaging services?

☐ At least once a day ☐ A few times a week ☐ A few times a month
☐ Never

21. Do you access any of the following services from your cellphone? (tick all that apply to you)

☐ Facebook ☐ Twitter ☐ News
☐ E-mail ☐ Wikipedia ☐ Other (please specify)

Other: ....

22. How often do you access these services from your cellphone?

☐ At least once a day ☐ A few times a week ☐ A few times a month
☐ Never

23. Have you ever installed an application on your cellphone using its application store? (e.g. Google Play, Nokia Store, BlackBerry World, AppStore)

☐ Yes ☐ No ☐ I don't know

24. Have you ever manually installed an application on your cellphone? (e.g. using your phone's web-browser)

☐ Yes ☐ No ☐ I don't know

Section 3 – Service Delivery Satisfaction & Current Participation

25. How satisfied are you with the delivery of the following services in your area?

Service	Very satisfied	Fairly satisfied	Not very satisfied	Not at all satisfied	I don't know
Electricity					
Water					
Sanitation					
Refuse removal					
Parks and recreation facilities					
Roads and sidewalks					

26. Did you vote in any of the following **local government** (municipal) elections? (tick all that apply to you)

☐ 1995 ☐ 2000 ☐ 2006 ☐ 2011

27. If you did **not** vote in one or more of the above elections, please indicate why below:

☐ I was not registered to vote
☐ I did not want to vote
☐ I was unable to vote
☐ I don't know
☐ Other
Please specify: ....
....

28. Who is the Ward Councillor for your area?

Name: .... ☐ I don't know

29. Have you ever attended a meeting organised by your ward committee?

☐ No, never ☐ Yes, once or twice ☐ Yes, often ☐ I don't know

30. If **yes**, what is the reason for attending meetings?

....
....
....
....

31. If **no**, what is the reason for not attending meetings?

☐ I don't have transport
☐ I don't have the time
☐ They will not listen to my opinion
☐ I am not interested
☐ I don't listen to lies
☐ They are not visible to us
☐ I have not thought about it
☐ I am not aware of this committee
☐ I don't have any information about the meetings that are held
☐ It will make no difference, nothing will change
☐ There are no meetings/they are not active in this area
☐ I don't know

32. Have you served as a ward committee member?

☐ Yes ☐ No

33. Have you ever read a copy of the following documents produced by Makana municipality?

	Yes	No
Integrated Development Plan (IDP)		
Budget		
Service Delivery Budget Implementation Plan (SDBIP)		
Annual Report		
Auditor-General's Report		

34. If you have never read a copy of these documents, why?

- ☐ I don't know what they are
- ☐ I have never heard of them
- ☐ I could never get a copy
- ☐ I'm not interested in reading them
- ☐ I cannot understand them
- ☐ There is no point in reading them
- ☐ Other
  Please specify: ....
  ....

35. Have you ever made a complaint to your local municipality? (tick as many as are applicable)

- ☐ No, I have never made a complaint
- ☐ I signed a petition
- ☐ I spoke with somebody in the municipality
- ☐ I phoned someone at the municipality
- ☐ I raised the issue with my Ward Councillor
- ☐ I wrote a letter
- ☐ Other
  Please specify: ....
  ....

36. Have you ever reported poor service delivery to anyone outside the municipality? (tick as many as are applicable)

- ☐ No, I have never made a complaint
- ☐ The media
- ☐ The police
- ☐ Presidential Hotline
- ☐ Public Service Commission Hotline
- ☐ Public Protector
- ☐ Auditor-General
- ☐ South African Human Rights Commission
- ☐ Advice Offices, NGOs
- ☐ LegalAidSA

37. Have you taken part in a protest or demonstration within the last 12 months?

- ☐ Yes
- ☐ No
- ☐ I don't know

38. When there are problems in how local government is run in your area, how much do you think an ordinary person can do to improve the situation?

- ☐ Nothing
- ☐ A small amount
- ☐ Some
- ☐ A great deal
- ☐ I don't know

39. Do you read the Grocott's Mail newspaper?

- ☐ Yes, often
- ☐ Sometimes
- ☐ Never

40. Do you listen to Radio Grahamstown?

- ☐ Yes, often
- ☐ Sometimes
- ☐ Never

The questionnaire is now completed. Thank you for your time.

B.2 Usability evaluation: navigation and visualisation

B.2.1 Consent form

CONSENT FORM

Project Title Social Accountability Monitoring using Mobile Phones

Project Description In this project, the researcher hopes to determine whether mobile phones are a viable technology to facilitate increased participation in local government processes. This user study aims to gain feedback regarding the MobiSAM applications' navigation and visualisation capabilities. Participants will be asked to complete a series of tasks, with each task assessing a different aspect of the application. Participants will complete these tasks on a handset which most closely resembles their own, ensuring the operation of an unfamiliar handset doesn't burden them unnecessarily. Feedback obtained from the study will help further refine and improve the MobiSAM mobile applications.

Researcher Mr Edward Reynell

- I have received information about this research project.
- I understand the purpose of the research project and my involvement in it.
- I understand that I may withdraw from the research project at any stage.
- I understand that participation in this user study is done on a voluntary basis.
- To the best of my knowledge I have no physical impediments that will stop me from completing this study (e.g. colour blind).
- I understand that while information gained during the study may be published, I will not be identified and my personal details will remain confidential.

Name of participant ....

Signed .... **Date** ....

I have provided information about the research to the research participant and believe that he/she understands what is involved.

Researcher's signature and date ....

B.2.2 Pre-intervention questionnaire

Pre-intervention Questionnaire

Please answer all the questions in the space provided or tick where appropriate.
Section 1 deals with basic demographic information while **Section 2** deals with your cellphone usage habits.

Please note: *no personally identifiable information will be linked to your answers.*

Section 1 – Demographic information

1. What is your gender type?
   ☐ Male ☐ Female
2. In which age group do you belong?
   ☐ 18 – 30 years ☐ 31 – 45 years ☐ 46 – 60 years ☐ Over 60 years
3. What is your preferred language?
   ☐ Xhosa ☐ English ☐ Afrikaans ☐ Zulu
   ☐ Ndebele ☐ Sotho ☐ Swazi ☐ Tswana
   ☐ Tsonga ☐ Venda ☐ Other (please specify)

   Other: ....
4. What is your highest level of education?
   ☐ Primary school ☐ Some high school ☐ Matric ☐ University degree
   ☐ Other (please specify)

   Other: ....

Section 2 – Cellphone usage information

5. Do you own a cellphone?
   ☐ Yes ☐ No
6. If no, do you have access to a cellphone that you use regularly?
   ☐ Yes ☐ No

   If you don't own or have access to a cellphone, please skip to the end of the questionnaire.
7. How many years have you used a cellphone for?
   ☐ 1 – 5 years ☐ 6 – 10 years ☐ Over 10 years
8. What brand is your current cellphone?
   ☐ Nokia ☐ Samsung ☐ Motorola ☐ Sony Ericsson
   ☐ LG ☐ BlackBerry ☐ Other

   Model: .... (Please ask for help if you don't know)

P.T.O.

9. Who is your cellphone network operator?
   - ☐ Vodacom
   - ☐ MTN
   - ☐ Cell C
   - ☐ Virgin Mobile
   - ☐ Telkom Mobile
   - ☐ Other (please specify)

   Other: ....

10. Is your cellphone on pre-paid (i.e. Pay-As-You-Go) or contract?
    - ☐ Pre-paid
    - ☐ Contract

11. If **pre-paid**, please specify how much airtime you spend per week, on average.
    - ☐ Less than R5
    - ☐ R5 – R15
    - ☐ R16 – R30
    - ☐ More than R30
    - ☐ I don't know

12. If **contract**, please specify how much you spend per month.

    R..

    - ☐ I don't know

13. What do you use your cellphone for? Please rank in order of most used (1 – most, 2 – moderate, 3 – least, blank – never)
    - ☐ Voice calls
    - ☐ SMS
    - ☐ Data services (WhatsApp, Facebook, Twitter, browsing etc.)

14. Do you use any of the following messaging services on your cellphone?
    - ☐ SMS
    - ☐ BBM (BlackBerry)
    - ☐ MxIT
    - ☐ WhatsApp
    - ☐ Other (please specify)
    - ☐ I don't use any of these

    Other: ....

15. How often do you use these messaging services on your cellphone?
    - ☐ At least once a day
    - ☐ A few times a week
    - ☐ A few times a month
    - ☐ Never

16. Do you access any of the following services from your cellphone?
    - ☐ Facebook
    - ☐ Twitter
    - ☐ News
    - ☐ E-mail
    - ☐ Wikipedia
    - ☐ Other (please specify)

    Other: ....

17. How often do you access these services from your cellphone?
    - ☐ At least once a day
    - ☐ A few times a week
    - ☐ A few times a month
    - ☐ Never

18. Have you ever installed an application on your cellphone using its application store? (e.g. Google Play, Nokia Store)
    - ☐ Yes
    - ☐ No
    - ☐ I don't know

19. Have you ever manually installed an application on your cellphone? (e.g. using a USB cable & computer)
    - ☐ Yes
    - ☐ No
    - ☐ I don't know

The questionnaire is now complete. Thank you for your time.

B.2.3 Usability evaluation tasks

User Study Tasks & Questionnaires

MobiSAM is a service which allows you to answer polls relating to the delivery of basic services, as well as report service delivery problems to your municipality.

This user study aims to gather feedback regarding the MobiSAM mobile application's navigation and visualisation capabilities. This feedback will be used to further refine and improve the application's design.

Please complete the tasks presented below.

Please note: *no personally identifiable information will be linked to your answers.*

Task 1 – No water

1. Please log in to the MobiSAM application ("app") using the username **user**, and password **pass**.
2. Once logged in, please follow the necessary steps to inform Makana municipality that you currently have **no running water at your home/residence**.
3. Once the poll has been successfully submitted, please use the chart visualisations to see how many other people in your suburb also have **no water**.

   Answer: .... people / .... %
4. Please navigate back to the 'Home' screen once you are done.

(Please turn over the page and answer the questions relating to the task you have just completed.)

P.T.O.

Task 1 questionnaire

Please answer all questions in the space provided. Additional comments are encouraged.

1.	Logging in to the MobiSAM app was easy

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

2.	Finding the correct poll was confusing

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

3.	Completing the correct poll was straightforward

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

4.	The charts were easy to understand

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

5.	I used the interactive features of the charts (only answer if using a touchscreen cellphone)

☐ Yes ☐ No

Additional comments:

....

....

P.T.O.

Task 2 – Leaking pipe

1. Please log in to the MobiSAM app using the username **user**, and password **pass**.
2. Once successfully logged in, please follow the necessary steps to inform Makana municipality that there is a leaking water pipe at **5 High Street**.
3. Please navigate back to the 'Home' screen once you are done.

(Please turn over the page and answer the questions relating to the task you have just completed.)

P.T.O.

Task 2 questionnaire

Please answer all questions in the space provided. Additional comments are encouraged.

1.	Finding the correct poll was straightforward

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

2.	Having the screen names at the top of the MobiSAM app made navigation easier

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

3.	Entering text into the MobiSAM app was difficult

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

4.	Enough feedback was given after submitting my report

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

Task 3 – Dirty water

1. Please log in to the MobiSAM application ("app") using the username **user**, and password **pass**.
2. Once successfully logged in, please follow the necessary steps to inform Makana municipality that **your water is brown in colour**.
3. Once the poll has been successfully submitted, use the chart visualisations to see how many other people in your area also have **brown water**.

   Answer: .... people / .... %
4. Please go back to the 'Home' screen once you are done.

(Please turn over the page and answer the questions relating to the task you have just completed.)

P.T.O.

Task 3 questionnaire

Please answer all questions in the space provided. Additional comments are encouraged.

1. Finding the correct poll was confusing

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

2. Completing the correct poll was straightforward

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

3. The charts were easy to understand

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

4. Which chart did you find easiest to understand?

☐ Pie chart ☐ Bar chart ☐ Both equal

Additional comments:

....

....

Task 4 – Change application language

1. Please log in to the MobiSAM app using the username **user**, and password **pass**.
2. Next, please follow the necessary steps to change the MobiSAM app's language **from English to Afrikaans**.
3. Take note of whether the 'Home' screen is now displayed in Afrikaans.

(Please turn over the page and answer the questions relating to the task you have just completed.)

P.T.O.

Task 4 – Change application language

Please answer all questions in the space provided. Additional comments are encouraged.

1. Finding where to change the MobiSAM app's language was straightforward

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

2. The MobiSAM app's 'Update profile' screen was easy to understand

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

3. Are there any other profile options you would like added to what is currently available? For example: change your suburb/address, change your password etc. (If yes, please provide examples below.)

☐ Yes ☐ No

Examples:

....

....

....

The questionnaire is now complete. Thank you for your time.

B.2.4 Post-intervention questionnaire

Post-intervention Questionnaire

Please answer all the questions in the space provided. Additional comments are encouraged.

Please note: *no personally identifiable information will be linked to your answers.*

Section 1 – Application navigation

1. The MobiSAM app, in general, was easy to understand

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

2. Navigating my way through the MobiSAM app was simple

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

3. The 'Home' screen of the MobiSAM app was clear and concise

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

4. I would most likely use the MobiSAM app again in the future

☐ Yes ☐ No ☐ Maybe

Additional comments:

....

....

P.T.O.

Section 2 – Application visualisation

5. The charts, in general, were easy to understand

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

6. I felt comfortable reading the charts

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

7. Not enough distinct colours were used in the charts

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

8. The interactive charts made information more easily understandable (only answer if using a touchscreen cellphone)

☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Additional comments:

....

....

9. Which of the visual representations did you find most useful? (i.e. numerical, pie chart, or bar chart)

☐ Numerical ☐ Pie chart ☐ Bar chart

Additional comments:

....

....

Is there anything else about the MobiSAM app that you would like to tell us?

....

....

The questionnaire is now complete. Thank you for your time.

B.3 Usability evaluation: client application comparison

B.3.1 Consent form

CONSENT FORM

Project Title Social Accountability Monitoring using Mobile Phones

Project Description In this project, the researcher hopes to determine whether mobile phones are a viable technology to facilitate increased participation in local government processes. This usability evaluation aims to measure how changes made to the original MobiSAM application are perceived by users. A side-by-side comparison will be made between MobiSAM and a newly developed version, namely, MobiSAM Report. Participants will be asked to complete four tasks using each version of the app; participants will then be asked a series of questions relating to the tasks. Responses from this study will ultimately influence the decision as to which version of the MobiSAM app will be released into the public domain.

Researcher Mr Edward Reynell

- I have received information about this research project.
- I understand the purpose of the research project and my involvement in it.
- I understand that I may withdraw from the research project at any stage.
- I understand that participation in this user study is done on a voluntary basis.
- I understand that my responses to questions will be recorded.
- I understand that while information gained during the study may be published, I will not be identified and my personal details will remain confidential.
- To the best of my knowledge I have no physical impediments that will stop me from completing this study (e.g. colour blind).

Name of participant ....

Signed .... **Date** ....

I have provided information about the research to the research participant and believe that he/she understands what is involved.

Researcher's signature and date ....

B.3.2 Pre-intervention questionnaire

Usability evaluation: pre-intervention questionnaire

Please answer all the questions in the space provided or tick where appropriate.
Section 1 deals with basic demographic information while **Section 2** deals with your cellphone usage habits.

Please note: *no personally identifiable information will be linked to your answers.*

Section 1 – Demographic information

1. What is your gender type?

☐ Male ☐ Female

2. In which age group do you belong?

☐ 18 – 30 years ☐ 31 – 45 years ☐ 46 – 60 years ☐ Over 60 years

3. What is your preferred language?

☐ Xhosa ☐ English ☐ Afrikaans ☐ Zulu
☐ Ndebele ☐ Sotho ☐ Swazi ☐ Tswana
☐ Tsonga ☐ Venda ☐ Other (please specify)

Other: ....

4. What is your highest level of education?

☐ Primary school ☐ Some high school ☐ Matric ☐ University degree
☐ Other (please specify)

Other: ....

Section 2 – Cellphone usage information

5. Do you own a cellphone?

☐ Yes ☐ No

6. If no, do you have access to a cellphone that you use regularly?

☐ Yes ☐ No

If you don't own or have access to a cellphone, please skip to the end of the questionnaire.

7. How many years have you used a cellphone for?

☐ 1 – 5 years ☐ 6 – 10 years ☐ Over 10 years

8. What brand is your current cellphone?

☐ Nokia ☐ Samsung ☐ Motorola ☐ Sony Ericsson
☐ LG ☐ BlackBerry ☐ Other

Model: .... (Please ask for help if you don't know)

P.T.O.

9. Who is your cellphone network operator?
   - ☐ Vodacom
   - ☐ MTN
   - ☐ Cell C
   - ☐ Virgin Mobile
   - ☐ Telkom Mobile
   - ☐ Other (please specify)

   Other: ....

10. Is your cellphone on pre-paid (i.e. Pay-As-You-Go) or contract?
    - ☐ Pre-paid
    - ☐ Contract

11. If **pre-paid**, please specify how much airtime you spend per week, on average.
    - ☐ Less than R5
    - ☐ R5 – R15
    - ☐ R16 – R30
    - ☐ More than R30
    - ☐ I don't know

12. If **contract**, please specify how much you spend per month.

    R ....

    - ☐ I don't know

13. What do you use your cellphone for? Please rank in order of most used (1 – most, 2 – moderate, 3 – least, blank – never)
    - ☐ Voice calls
    - ☐ SMS
    - ☐ Data services (WhatsApp, Facebook, Twitter, browsing etc.)

14. Do you use any of the following messaging services on your cellphone?
    - ☐ SMS
    - ☐ BBM (BlackBerry)
    - ☐ MxIT
    - ☐ WhatsApp
    - ☐ Other (please specify)
    - ☐ I don't use any of these

    Other: ....

15. How often do you use these messaging services on your cellphone?
    - ☐ At least once a day
    - ☐ A few times a week
    - ☐ A few times a month
    - ☐ Never

16. Do you access any of the following services from your cellphone?
    - ☐ Facebook
    - ☐ Twitter
    - ☐ News
    - ☐ E-mail
    - ☐ Wikipedia
    - ☐ Other (please specify)

    Other: ....

17. How often do you access these services from your cellphone?
    - ☐ At least once a day
    - ☐ A few times a week
    - ☐ A few times a month
    - ☐ Never

18. Have you ever installed an application on your cellphone using its application store? (e.g. Google Play, Nokia Store)
    - ☐ Yes
    - ☐ No
    - ☐ I don't know

19. Have you ever manually installed an application on your cellphone? (e.g. using your phone's web-browser)
    - ☐ Yes
    - ☐ No
    - ☐ I don't know

Thank you. The questionnaire is now complete.

B.3.3 Usability evaluation tasks

Usability evaluation tasks

MobiSAM

Task 1

You have just turned on the tap and realised that there is no running water. Please log in/sign in to the app with the username *user* and the password *pass*.

1.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Task 2

Please report that there is a water outage at your *current* position, obtained using GPS. Once successfully reported, please navigate back to the Home screen.

2.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Task 3

You have just seen water bubbling out of the ground at *1 High Street*. Please report this problem to Makana Municipality. Include a photo of the issue. Once successfully reported, please navigate back to the Home screen.

3.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Task 4

Please change the app's language from English to isiXhosa. Then, return to the Home screen.

5.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

P.T.O.

MobiSAM Report

Task 1

You have just turned on the tap and realised that there is no running water. Please log in/sign in to the app with the username *user* and the password *pass*.

1.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Task 2

Please report that there is a water outage at your *current* position, obtained using GPS. Once successfully reported, please navigate back to the Home screen.

2.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Task 3

You have just seen water bubbling out of the ground at *1 High Street*. Please report this problem to Makana Municipality. Include a photo of the issue. Once successfully reported, please navigate back to the Home screen.

3.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Task 4

Please change the app's language from English to isiXhosa. Then, return to the Home screen.

5.) Overall, this task was?

Very difficult — Very easy

1	2	3	4	5

Thank you. All tasks are now complete.

B.3.4 Post-intervention questionnaire

Usability evaluation: post-intervention questionnaire

1. Which of the two apps did you find easiest to use? Why?

....

....

....

2. Once familiar with the apps, do you feel you would be able to perform the tasks quicker using MobiSAM or MobiSAM Report?

....

....

....

3. After having not used either of the apps for a while, which of the two would you feel most comfortable using again? Why?

....

....

....

4. Do you feel more likely to make mistakes with MobiSAM or MobiSAM Report? Where do you foresee these mistakes happening? If you made a mistake during the tasks, were you easily able to recover from them?

....

....

....

5. Which of the overall app designs do you find most pleasing and intuitive? Why?

....

....

....

6. Does either of the apps better provide the features that you need? Which features are these? Why?

....

....

....

7. What are your likes, dislikes and recommendations for each of the apps?

....

....

....

Appendix C

MobiSAM RESTful Web Services API

REpresentational State Transfer (REST)
Application Programming Interface (API)

Version : 0.4
23rd July, 2014

Compiled by : Mamello

MobiSAM RESTful Web Services API

Document Overview

This is version 0.4 of the MobiSAM REST API specification. This is for the pre-production version of MobiSAM and as such will be getting updated frequently prior to the launch of MobiSAM. The document gives a brief overview of how external applications and services can interact with MobiSAM via the REST API.

MobiSAM REST API Overview

The RESTful web service API is accessible to uniquely identified applications that are tied to user accounts. Users can have multiple applications linked to their accounts. Users need to first be registered on MobiSAM via the web interface before they can link their applications to their accounts.

Communication between the applications and the API is via HTTP requests that embed JSON messages for further information required by the API. Examples of the request and response messages are given in the document.

The REST API is currently beta and is accessible via the MobiSAM beta sub-domain : `http://beta.mobisam.net`

API routes

The table below describes the routes that are defined in the API. All data exchange with the API is via JSON :

Table 1 – MobiSAM REST API routes

REST route	Description
`GET api/polls`	Returns all the polls that the user is authorized to access, along with detailed information associated with the polls. *JSON response:* `[{"pollid":"2","question":"What colour is your water today?","start":"2013-02-07","end": "2014-02-01","category":"1", "options": {"1":"Clear","2":"Cloudy","3":"Slightly brown","4":"Brown"}} … ]` -- updated in version 0.2 to return category ID instead of full category name, also returns an array of options

GET api/categories	Returns poll categories with their corresponding IDs. *JSON response:* [{"categoryid":"1", "name":"Health", "description":"Polls related to health and also to medical issues", "pollscount":"3"}, …]
GET api/polls/<category> *e.g. api/polls/2* -- updated in version 0.2 to use category ID instead of category name.	Returns the polls in a particular category specified by the category identifier <category>. *JSON response:* [{"pollid":"2","question":"What colour is your water today?","start":"2013-02-07","end": "2014-02-01","category":"1", "options": {"1":"Clear","2":"Cloudy","3":"Slightly brown","4":"Brown"}} …] -- updated in version 0.2 to return the poll information plus an array of options
GET api/poll/<id>	Returns information about a poll with the specified id. *Example :* GET api/poll/3 to get information about poll number 3 *JSON response:* {"pollid":"2","question":"What colour is your water today?","start":"2013-02-07","end":"2014-02-01","category":"1", "anon":"0","open":"1","multiple":"1","options":{"1":"Clear","2":"Cloudy","3":"Slightly brown","4":"Brown"}} -- updated the response to include an array of options in version 0.2
POST api/result/<poll>/<choice>	Responds to the poll identified by its unique identifier <poll> with the user's response choice specified by <choice>. If the response is to a poll with a text feedback or for uploading images, the <choice> has to be set to 0 (zero) and the text feedback and other parameters are embedded into request. This has to be authenticated and authorized accordingly as described below. *Example :*

	`POST api/result/4/3` to respond to poll number 4 with a choice number 3 selected. *JSON response:* {"hot":"15","not":"2"} *or* {"message":"thank you for voting"} -- updated to include the results in the response (for polls where the results are accessible in version 0.2 updated in version 0.4 to handled the uploading of images for the relevant poll types
`POST api/associate`	This associates an application with a specific user. *(See "Authentication" section below)*

Authentication

The initial process in using MobiSAM REST API is to associate the requesting application with a specific user account. Therefore the user must have registered via the web interface beforehand. The association is a one-step processes where a request embedding information that identifies both application and the user is send to MobiSAM. The route for this call is as follows :

`POST api/associate`

with the following JSON formatted information embedded in the body of the request :

```
{
        "appkey": "<appkey>",
        "description":"<description>",
        "user": "<username>",
        "checksum":"<code>"
}
```

where :

`<appkey>` = unique hexadecimal string of length 50. This is generated by the application requesting to be associated with the user. This code should be kept private and secure and is only to be used to keying the hash of the `POST api/result/<poll>/<choice` as described further below

`<description>` = a user- supplied description of the application (e.g. name of the application). This is used on the web interface to allow the user to manager their authentication applications and services.

`<username>` = the username of the person accessing the service

<code> = the SHA256 hash of the user password, keyed with the username. In othe words, `hmac ('sha256', <password>, key=<username>)`

If successful this returns an accesscode in a JSON format

{
"accesscode":<accesscode>"
}

the accesscode is to be stored and used for subsequent requests to the API.

Otherwise the following error messages are returned, with the HTTP Response Code and the JSON Message embedded in the body of the response :

Table 2 : Authentication response messages

Response Code	Message	Error
400	{"message":"missing data"}	There is no information embedded in the body of the request
400	{"message":"invalid parameters"}	Invalid parameters embedded in the request, this could be because of missing parameters or parameters that do not conform to the specification.
404	{"message":"incorrect parameters"}	Incorrect parameters (e.g. credentials) have been passed
403	{"message":"application already associated"	The application (identified by the <appkey>) has already been associated with the user.
500	{"message":"could not associate due to a technical issue"}	The account/user association could not be completed due to an internal application error

Authorization

Every request to the API, except the `POST api/associate` request must have the `X_MOBISAM_AUTH` header set with the value of the assigned `<accesscode>` from the association process above.

Calls to the `POST api/result/<poll>/<choice>` route must also embed within the body of the request a SHA256 hash of the request URI, keyed with the unique private `<appkey>` generated in the association process above. Where the response is for a textual feedback poll, the text response has to be embedded in the body of the request and the text has to be appended to the URI to generate the keyed hash.

Example :

- for a choice response

```
POST api/result/27/3
```

{"checksum":"2f033a52d2fecdb64dd6ad976da519ba3b88d7ea235ff427a0103583f2d3471f"}

the checksum value is `hmac('sha256', '27/3','9EWVFmHpHN6n2YKW9QtvUqX3xbsFQUBovlrFddqnF7fpcSDA2q')`

- for a text response (the <choice> has to be set to 0)

```
POST api/result/27/0
```

{"checksum":"590f33bd50c9b3027a2125fd0be33dcd66eca45a4219bae244732f8de62c71dc", "response":"the water is brown"}

the checksum value is `hmac('sha256', '27/0_the water is brown','4mwisugu2afeezwkp2b6z3zlab83j96dad7nri787qa96008ea')`

- for an image upload response (the <choice> has to be set to 0)

```
POST api/result/34/0
```

{"checksum":"4369b28e3b8c8b855338c57a9987573129452858af24d550a2cd62c2fde95fb8",
"response" : "testing",
"location" : "12.33232,32.5343",
"locationstring" : "25 Hill Street, Grahamstown",
"image" : "/9j/4AAQSkZJR....Nr4zGlPL8ueMwS4v//Z",
"comments":"there is pipe that's burst"}

- ***checksum** value is hmac('sha256' , '34/0_testing' , '627a524063277e2458387b4c4a5259473c454d48652373774a')
- **image** is a Base64 encoded image (only jpeg. jpg, and png supported)
- ***location** is the longitude and latitude
- ***locationstring** is the physical address
- **comments** are any extra user comments
- ** = required parameters*

If the request is authenticated and authorized, the response would be as per the MobiSAM REST API routes table above. Otherwise the following error messages are returned.

Table 3 : Authorization response messages

Response Code	Message	Error
400	{"message":"missing header information"}	There required header information is missing from the request.
403	{"message":"invalid request"}	Invalid information for identifying the requesting application.

API Error Messages

Response Code	Message	Error
404	{“message”:“no items were found for that category”}	There are no polls available for the specified category
400	{“message”:”Missing information, the category must be specified”}	The category ID must be specified for the request
404	{“message”:“No items were found for <x>”} {“message”:“The requested item was not found”} {“message”:“Specified poll is not available”}	None of the requested polls or categories are available to the user
400	{“message”:“Missing POST data”}	The request is missing the required POST body
400	{“message”:“incorrectly formatted request”}	The request doesn't comply to the specification
400	{“message”:“inconsistent request”}	The request URI doesn't match the encoded data
400	{“message”:“invalid poll choice selection”}	The specified choice is not valid for the selected poll
400	{“message”:“Not a correct JSON format, missing user response”}	The user's text response is missing from the JSON request
401	{“message”:“Already voted”}	The user has already voted for the selected poll
500	{“message”:“Could not save”}	The user's voting could not be captured due to an internal or technical MobiSAM error
400	{“message”:“missing data”}	The information that is required is missing
403	{“message”:“application already associated”}	The application (identified by the AppKey) has already been associated for the user