



University of Fort Hare
Together in Excellence

**Implementing an integrated
e-Government functionality for a
marginalized community in the
Eastern Cape, South Africa**

A thesis submitted in fulfillment of the requirements of the degree of

Master of Science

in

Computer Science

University of Fort Hare

by

Bobby Tichaona Jakachira

November 2009

Declaration

I hereby declared that the content of this research work is my original work and information obtained from other sources is acknowledged as such.

November 2009

Acknowledgements

During the research of this project, I received immense support from a number of individuals and sponsors. I would like to thank my co-supervisor, Ronald Wertlen. He has been a good and efficient person to work with. I would like to thank him for his cordial help, encouragement and patience throughout the duration of this research. I would like also to thank Prof Alfredo Terzoli and Prof HN Muyingi for providing support while I undertake my Masters studies.

I would like to acknowledge the NRF for their generosity in awarding me an NRF Africa scholarship which made it possible for me to undertake my Masters studies.

I would like also to acknowledge the Centre of Excellence for financial and technical support during the course of research. Special thanks go to various individuals in the Computer Science Department who have been there for me when I need help. These people include the Head of Department and the class mates.

I would also like to extend my gratitude to my family members, with their words of wisdom and encouragement. These have been a source of inspiration to me throughout the period of this research.

Abstract

Traditional methods of providing public services to disadvantaged rural communities in South Africa have, over the years, proven to be inefficient and in most such communities, simply non-existent. Although the South African government has taken initiatives to make these public services cheaply and conveniently available online at national level, access at local municipal level is still lacking. The goal of this study is to develop a cost-effective e-government system that will contribute to improved provision of public services to the Dwesa area, a rural community in the Eastern Cape province of South Africa, by the government.

A prototype construction approach was used, to develop a cost-effective four-modular web application. Interviews were conducted in the field, resulting in four e-government system modules, based on open-source software, developed and integrated to form a single, dynamic web component that will act as a one-stop shop for Dwesa community members. These are the Dwesa Online Application Centre (DOAC) to apply for important government documents and grants, the Dwesa Online Reporting Centre (DORC) to report various grievances to the responsible agencies, the Dwesa Forum Corner (DFC), a digital community, and the management back-end module. The Dwesa e-government portal was developed using Linux-Apache-MySQL-PHP (LAMP) technology, a Zoop framework to model the individual components and a JQUERY JavaScript library to increase the responsiveness of the user interfaces.

The most significant contributions of this thesis have been the development of a cost-effective, integrated e-government functionality, applicable to disadvantaged communities, and the greater understanding this has given of the tools and methodologies that can be used to deliver public services efficiently to citizens. The final evaluation of this e-government system gives significant evidence that the e-government portal provides a solid foundation that will allow e-government implementation to raise the provision of public services to a higher level.

Table of Contents

Chapter One: Introduction

1	Introduction.....	2
1.1	Background Information.....	2
1.2	Problem Statement.....	2
1.3	Research Questions.....	4
1.4	Research Objectives.....	4
1.5	Delineation and Limitations.....	5
1.6	Project Context.....	6
1.7	Underlying Assumptions	6
1.8	Significance of the Study	7
1.9	Brief Chapter Overviews	8
1.10	Conclusion	9

Chapter Two: Related Work

2	Introduction.....	11
2.1	Definition of Integrated E-Government Functionality	11
2.2	The Nature of E-Government	13
2.2.1	Growth Models of E-Government Systems.....	14
2.2.2	The Meta-Synthesis Approach.....	17
2.3	Major areas of E-Government Development.....	18
2.3.1	Government-to-Citizen (G2C) Service Delivery	18
2.3.2	Government-to-Government (G2G) Service Delivery	21
2.3.3	Government-to-Business (G2B) Service Delivery	21
2.3.4	Government-to-Employee (G2E) Service Delivery.....	21
2.4	Benefits in the Adoption of Electronic Government	22
2.4.1	E-Government's Benefits to Citizens	22
2.4.2	E-Government's Benefits to the State.....	22
2.5	E-Government Initiative Perspectives	23
2.5.1	Citizen Perspectives.....	23
2.5.2	Government Perspectives.....	24
2.6	A History of E-Government in South Africa.....	24
2.7	E-Government Impacts in South Africa	25
2.8	E-Government Challenges in South Africa	26

2.9	Potential Models of e-Government Access in South Africa.....	27
2.9.1	Model One: Smart Service Gateway Service Points.....	27
2.9.2	Model Two: Smart Plug-In	27
2.9.3	Model Three: M-Services	28
2.9.4	Model Four: Government Online.....	28
2.9.5	Model Five: Centre Services Gateway Service Centre.....	29
2.9.6	Model Six: Talk-to-Government plain old telephone.	29
2.9.7	Model Seven: Computerized Counter Services	30
2.9.8	Proposed E-Government Access Model	30
2.10	Conclusion	30

Chapter Three: System Methodology, Requirements and Technologies

3	Introduction.....	33
3.1	Background of the Dwesa E-Government Initiative.....	33
3.2	E-Government Software Development Method	34
3.2.1	Aspects of RAD Methodology.....	34
3.2.2	The Advantages of RAD Methodology	34
3.3	E-Government System Requirements and Assessment	37
3.3.1	E-Government System Functional Requirement	38
3.3.2	E-government System Non-Functional Requirements	38
3.3.3	E-government System Modularity.....	40
3.4	Dwesa E-government System Functionality Flow Control	40
3.4.1	Data Access Objects (DAOs).....	41
3.4.2	Model Classes	41
3.4.3	Action Classes.....	41
3.4.4	Zone Templates.....	41
3.5	E-Government Software Development Technologies	41
3.5.1	Dwesa E-Government System OS Platform.....	42
3.5.2	Dwesa E-Government System Web Server	42
3.5.3	Dwesa E-Government System Database Server	43
3.5.4	Dwesa E-Government System Model Framework	47
3.5.5	E-Government System Development Environment.....	49
3.5.6	JQUERY JavaScript Library.....	49
3.5.7	Cascade Styling Sheets (CSS)	50
3.5.8	PDF File Filler	50

3.5.9	Digital Signatures.....	51
3.5.10	Process Scheduling	51
3.5.11	Forms Transfer Using FTP	52
3.6	Conclusion	52

Chapter Four: Dwesa E-Government System Administration

4	Introduction.....	54
4.1	The Distribution of the Dwesa E-Government System	55
4.2	The architecture of Dwesa E-Government System.....	56
4.2.1	System Page Zones	57
4.2.2	Action Classes in the Dwesa E-Government System	58
4.2.3	System Zone Templates.....	59
4.2.4	System Model Classes	59
4.2.5	Data Access Objects	59
4.2.6	System Support Classes	60
4.3	Dwesa E-Government System Database Design	60
4.3.1	Database Tables	60
4.3.2	Database Views.....	63
4.3.3	Stored Procedures in Dwesa E-Government Database.....	64
4.3.4	Triggers in Dwesa E-Government Database.....	65
4.3.5	Events in Dwesa E-Government Database	66
4.4	Dwesa E-Government System User Management.....	68
4.4.1	User Accounts Management.....	68
4.4.2	Users Details Management	70
4.5	The Dwesa Online Application Centre (DOAC) Portal Management.....	70
4.5.1	Government File Servers Configuration.....	70
4.5.2	Updates of Government Forms	72
4.5.3	Applications Digital Signatures	74
4.5.4	DOAC Usage Monitoring	75
4.5.5	Dwesa E-Government Services Management	75
4.5.6	New and On-Request Government Forms	77
4.5.7	Application Notifications.....	77
4.5.8	Uploading New Forms	78
4.6	The Dwesa Online Reporting Centre (DORC) System Management	78
4.6.1	DORC System Performance	78

4.6.2	DORC Portal Access.....	79
4.6.3	DORC Usage Monitoring	79
4.6.4	Responding to Villagers' Reports.....	79
4.7	Dwesa Forum Corner (DFC) System Management.....	80
4.7.1	DFC System Access.....	80
4.7.2	DFC Usage Monitoring.....	81
4.7.3	Moderating DFC User Posts	81
4.7.4	Public Announcements Verification and Broadcasting	81
4.8	Conclusion	82

Chapter Five: E-Government Functionalities Design and Implementation

5	Introduction.....	84
5.1	The Dwesa Online Application Centre (DOAC)	84
5.1.1	Listing Government Forms on the DOAC Portal	85
5.1.2	Downloading Government Forms on the DOAC	86
5.1.3	Filling in Forms.....	87
5.1.4	Signing Applications.....	88
5.1.5	Saving Applications	90
5.1.6	Sending Applications	91
5.1.7	Tracking Applications.....	94
5.1.8	Requesting Other Forms	96
5.1.9	Viewing Applications	97
5.1.10	DOAC Security Features	97
5.2	The Dwesa Online Reporting Centre (DORC)	99
5.2.1	User Requirements for Reporting	99
5.2.2	Sending User Reports	100
5.2.3	Validation of User Reports	102
5.2.4	Routing User Reports.....	102
5.2.5	Receiving of Reports.....	103
5.2.6	Government Agency Responds.....	104
5.2.7	DORC Security Features.....	105
5.2.8	Latest Reports.	106
5.3	The Dwesa Forum Corner (DFC)	106
5.3.1	Public Announcements	107
5.3.2	Public Notice Board.....	107

5.4 Conclusion	107
----------------------	-----

Chapter Six: Evaluations, Results and Conclusion

6 Introduction.....	110
6.1 Summary of Findings.....	110
6.2 Evaluation of the Dwesa E-Government System.....	111
6.3 Dwesa E-Government System Evaluation Criteria.....	113
6.3.1 Functionality Evaluation.....	113
6.3.2 Accessibility Evaluation	117
6.3.3 Usability Evaluation.....	119
6.3.4 Other Success Factors or Enablers.....	121
6.4 Conclusion	121
6.4.1 Summary of Contributions.....	122
6.4.2 Suggestions for Further Research	123
6.4.3 Discussion of Constraints and Challenges	123

References

References.....	125
-----------------	-----

Appendice

Appendix A: System Installation and Setup.....	133
Appendix B: System Classes and Procedures.....	135
Appendix C: DwesaCentre Database Objects.....	139
Appendix D: Questionnaire and Feedback.....	141

List of Figures

Figure 3.1 The Rapid Development Method (Marner 1997).....	36
Figure 4.1 Dwesa E-Government System Back-end Interface.	54
Figure 4.2 The Front-End Interface.	55
Figure 4.3 Dwesa E-Government System Distribution.	56
Figure 4.4 Dwesa E-Government Database ERD.....	60
Figure 4.5 Users Database Tables.....	61
Figure 4.6 Forms Database Table.	62
Figure 4.7 Dwesa Reports Database Table.	63
Figure 4.8. Forum Database Table.....	63
Figure 4.9 Login View.....	64
Figure 4.10 Screening Report Stored Procedure.....	65
Figure 4.11. Users_bi_del Trigger.....	66
Figure 4.12. Clean Up Database Event.....	67
Figure 4.13. Screening Reports Event.	68
Figure 4.14 User accounts management.	69
Figure 4.15. Sending the user a new Password.....	69
Figure 4.16. Editing User Details.	70
Figure 4.17. List of Government File Servers.	71
Figure 4.18. Government File Server Settings.....	72
Figure 4.19. Updating Forms.....	72
Figure 4.20. Updating Forms.....	73
Figure 4.21. Links of forms extracted on Department of Home Affairs Site.	73
Figure 4.22. Links of forms extracted on Department of Social Work Website.	73
Figure 4.23. Example of a Private Key.	74
Figure 4.24. Example of a Public Signature.	75
Figure 4.25. User allocated memory calculation.	75
Figure 4.26. Services Management Portal.	76
Figure 4.27. Activation of Services.	76
Figure 4.28. Integrating New Services to the system.	77
Figure 4.29. Requesting a Government form.....	77
Figure 4.30. SMS functionality.....	78
Figure 4.31. Responding to Reports.	80
Figure 4.32. Making an Announcement.	81
Figure 5.1: The DOAC portal.	84
Figure 5.2 Listed Forms on Birth Category.....	86
Figure 5.3: Downloading a Form.....	87
Figure 5.4: Filling in a Pdf Form Online	88
Figure 5.5: Signing an application.	88
Figure 5.6: Example of a Private Key.....	89
Figure 5.7: Example of a Public Key.	89
Figure 5.8: Creating a Signature.	89
Figure 5.9: Validating a Signature.....	90
Figure 5.10: Saving an application.	90
Figure 5.11: Sending an application.	92
Figure 5.12: Government Server Simulation.....	92
Figure 5.13: Validating an application.....	93
Figure 5.14: Adding an application.	93

Figure 5.15: Transmitting an application.....	94
Figure 5.16: Tracking an application.....	95
Figure 5.17: Creating a tracking number.	95
Figure 5.18: Requesting an application.	96
Figure 5.19: Sent Applications.	97
Figure 5.20: Saved Applications.....	97
Figure 5.21: Waiting Applications.....	97
Figure 5.22: Validating a form.....	98
Figure 5.23: Validating an application.....	98
Figure 5.24: The DORC Portal.	99
Figure 5.25: Sending a report.....	100
Figure 5.26: Sending reports.....	101
Figure 5.27: Validating a report.....	102
Figure 5.28: Live Reports from Dwesa Village.....	103
Figure 5.29: Responding to a report.	104
Figure 5.30: Responding to a report.	105
Figure 5.31: Viewing latest reports on the DORC.....	106
Figure 5.32: The DFC Portal.	107
Figure 6.1 Participants' responses for question 8.	114
Figure 6.2. Participants' responses for question 18.	115
Figure 6.3. Participants' responses for question 19.	115
Figure 6.4. Participants' responses	118
Figure 6.5. Participants' responses for question 20.	120

List of Tables

Table 4.1: System Users Zones.....	57
Table 4.2: System Functionalities Zones.	58
Table 4.3 System Action.....	58
Table 4.4: Model Classes.....	59
Table 6.1: Dwesa Field Trips and Activities.	112
Table 6.2 Functionality Evaluated options.	116
Table 6.3: Accessibility Evaluated Options.....	118
Table 6.4: Usability Evaluated Options.....	120

List of Acronyms

ATM	- Asynchronous Transfer Mode
DFC	- Dwesa Forum Corner
DOAC	- Dwesa Online Application Centre
DORC	- Dwesa Online Reporting Centre
ICT	- Information and Communication Technology
Wi-Fi	- Wireless Fidelity
WiMAX	- World Interoperability
WLAN	- Wireless Local Area Network
SITA	- State Information Technology Agency
G2C	- Government-to-Citizen
RAD	- Rapid Application Development

Chapter One

Introduction

This chapter provides a general background for the development of a web-based e-government portal for marginalized communities. The chapter highlights the aims, objectives and significance of this research initiative.

1 Introduction.

The chapter will give an introduction to the research being undertaken. It will highlight the problem and questions the research is trying to solve and answer. This chapter will emphasize the objectives as well as the aims behind the research work. The contribution that the research will make to the citizen's social life and to governance in South Africa will be discussed. We also state in this chapter the underlying assumptions and the limitations of this study.

1.1 Background Information

It is estimated that, in 2002, only one in every ten people in South Africa were able to use the Internet (CSPI 2003). Since then, the growth in value of computer and telecommunications technologies and the proliferation in applications of the Internet and world-wide web has been overwhelming. As a result, we see that citizens, policymakers, and others have raised their expectations about the delivery of government services digitally (Heeks 2006). The challenge for all governments is to embrace the opportunities that the online world provides and ensure that community needs and expectations are met, while at the same time ensuring program and cost effectiveness.

E-government systems are primarily concerned with making government services available to citizens of the country through the innovative use of information and communication technologies (ICT). Also, an e-government system should strive to deliver public services in ways that citizens want them, using the internet and other technologies as enablers (Reffat 2006). The systems can be seen in terms of the processes implemented on ICT platforms connected to the internet (Heeks 2006). The benefits of implementing online service provision have been recognized for the past decade, both in developed countries, such as Australia's Federal Government, as well as in developing countries such as South Africa. The Federal Government of Australia, for example, released its Government Online Strategy in 2000. The aim was to provide a strategic framework to assist agencies in meeting key online commitments.

1.2 Problem Statement

Between 2003 and 2004 there was a significant growth of government services in South Africa available to South African citizens over the Internet (IST-Africa Consortium 2005). However, access to government services on the internet is skewed towards cities and towns, where

services and ICT infrastructure are available. It is also noted that poverty is geographically concentrated, with 72% of the poorest households residing in South Africa's rural areas, especially in the Eastern Cape Province (CSPI 2003). In this context, we have realized that the implementation of e-government access strategies is likely to have a strong bias towards cities and provincial towns, where the majority of the population resides. It is with this understanding that this research is being undertaken to address the needs of citizens in the remote Dwesa rural area in the Eastern Cape. The "development divide" in South Africa is the major stumbling block to the flourishing of e-government strategies. This research aims to develop an e-government system aligned to an area that is almost excluded from mainstream society and where most citizens suffer from lack of access to government services (Heeks & Arun 2007).

Government can do more to make access to these new technologies available to citizens who are not already online (Reffat 2006). It is a fact that in South Africa, most such citizens are those living in rural communities. The Dwesa community is one of these areas, where villagers walk more than 40km to visit different government offices at the Willowvale Business Centre and at least 80km to the Idutywa Business Centre to access public services. These services range from obtaining a form to applying for a government permit at the Home Affairs department. This becomes a real problem when we consider that a large proportion of the population in these rural communities comprises elderly people who cannot walk such long distances.

The research will be based on the development of an e-government portal. The research will review the nature of e-government access and the models of e-government development, synthesizing these to find the most applicable model and form of e-government strategy for a rural area such as Dwesa. While numerous models for rural e-government access currently exist, few of these models have successfully provided services to citizens in a way that treats them as individuals and offers them personalized services. From the existing South African e-government systems, rural users may want to launch valid grievances to the government but cannot because they do not know of any contacts to send their reports to. In this study, we seek to develop a prototype of a one-stop e-government portal that provides services in a different way to each citizen (Reffat 2006). This portal will help to promote the rebuilding of government-citizen relationships. As an initiative to reach out to its citizens, government will

be able to use the new technologies to treat citizens as individuals and provide them with personalized services (Reffat 2006).

1.3 Research Questions

This thesis seeks to demonstrate the practical application of a web-based integrated e-Government system in the Dwesa rural area. The overall development and design of the e-government system will be based on a qualitative approach. The study is specifically designed to assess the various links existing between networks of institutions, organizations, state and non-state actors to form a digital village enacted by ICT programs (Navarra & Ciborra 2003). Therefore the research questions posed here are:

- 1) *How can a web-based e-government system be developed?*
- 2) *Can public services be provided to disadvantaged communities through the Internet?*
- 3) *Why is a web-based e-government system needed in the Dwesa rural area?*
- 4) *What can we do to combat the lack of public services access in Dwesa rural community?*

The questions present a challenging research agenda (Jukic & Vintar 2008), drawing on issues that are debated theoretically and practically across various disciplines. The choice of the Dwesa rural area as a site for research is relevant to the exploration of the debate that the present study will address.

1.4 Research Objectives

The objective of this research is to develop and implement a prototype of an integrated e-government web-based proxy system that best suits the Dwesa rural area. The e-government system will be divided into three main e-government system modules. These are the Dwesa Online Application Centre (DOAC), the Dwesa Online Reporting Centre (DORC) and the Dwesa Forum Centre (DFC). According to Fang (2002), an e-government system should be database-driven, where all information is listed in the databases as it is placed online, since this provides the most effective solution for categorizing government information online. To avoid a permanent e-government project failure, it is best to prototype the e-government system with a pilot project (Heeks 2008). This project will integrate the two ideas on its way to meeting its objectives and we will use Dwesa community area as a test bed. Our stated objectives will require us to build a strong and user-defined e-government system that will answer the research questions we have posed.

We have also realized that there must be explicit objectives for social development and local economic development. The provinces must have their own e-government initiatives aligned with those objectives and their own priorities (Furber 2007). The research will also consist of a field test and assessment of the e-government prototype in the Dwesa rural area. The e-government system will bring a collection of basic government services to the rural public electronically, making these services more available and more efficient. These will be services that people actually need. The e-government portal will formulate a communication channel for the government to reach out to the Dwesa community villages in the Eastern Cape. The e-government system will comprise of systems modules depicting government services that will be available online so that Dwesa community people are able to access them.

The objectives of this research are explained below:

- To deploy a cost-effective e-government solution in a timely fashion (IST-Africa Consortium 2005).
- To ensure that the e-government solutions are aligned to the needs of the government and the citizens (IST-Africa Consortium 2005).
- To eliminate administrative bottlenecks, thus reducing the turn-around time in the provision of most government services (Heeks 2006).
- To link all the possible spheres of service delivery thereby providing a single view of government (IST-Africa Consortium 2005).
- To offer the citizens universal access to vital information at all times.
- To develop an e-government service that provides a user-friendly communication and transaction channel between the public organizations and the citizens (a government-to-citizen relationship) (Ezz 2006).
- To ensure a common, effective, look and feel for citizen interaction with government (Cook 2000).
- To create a digital community where information is shared, which is something that is often lacking in public sector organizations (Heeks 2006).

1.5 Delineation and Limitations

As much as this research will strive to bring government services to the Dwesa rural citizens, this study will not deal with a complete, fully phased e-government strategy. The work will consider online provision of basic government services as a first priority, and will treat the

gradual improvement of the e-government strategy as future work. This piece of research work will not look into the localization of the e-government system as this is being undertaken on a parallel research project and any further initiatives taken to localize the system shall be deemed future work.

The research work will not consider the applicability of the e-government system to urban centres, as it strives to address the need to bridge the development divide between rural and urban areas. Furthermore, the area targeted will be the Dwesa rural area situated in the Eastern Cape.

1.6 Project Context

This project is part of a larger framework project being implemented in the Dwesa rural area. This study is meant to be integrated into an on-going, multi-purpose ICT project that has been partly deployed in the area. The e-government strategy will address the lack of provision of services to citizens by the government. The system will strive to provide Dwesa rural villagers with vital information and online transactions with the government. The e-government system will target the following users:

- Dwesa rural villagers, who will be able to get application forms for important documents like identity documents, birth and death certificates, and various types of government grants. Apart from the users' capability to make local announcements and social discussion through the system, they will also be able to launch reports and complaints with the various government departments. In such case, the e-government portal will give users the facility to report on the matters without having to know which agency it should be submitted to.
- Government departments will be able to receive, through the proxy system, reports made online by the citizens. The research will start by implementing a system that can link with a repository of government forms at the Home Affairs Department. The e-government strategy will also provide a Report Centre that allows the government agencies to receive reports and complaints from citizens and respond to them.

1.7 Underlying Assumptions

The research assumes that citizens will benefit from the cost-effective delivery of public services to solve their lack of information access at local municipal level. The e-government

system will improve the accessing and processing of public information by Dwesa community members.

1.8 Significance of the Study

The research work contributes by creating an e-government application system product that aims to alleviate the lack of government services access in Dwesa rural area. Thus, the study will be a significant endeavor in promoting efficient online delivery of public services to the Dwesa rural community. The concept of an e-Government initiative is relatively new at local municipal level, where the majority of the rural populace access public services. Whilst many e-government initiatives have successfully provided public services online to the users, this study looks to create an e-government portal that reflects government services in a personalized way for each user type. The view of the e-government portal differs between rural villagers and city dwellers. What rural villagers anticipate from online public services is not what urban population expect. Therefore, research which explores the different citizens' perspectives of online public service delivery will help to promote awareness among those who are unacquainted with its benefits at rural level.

The success of this research will significantly contribute to improved public service provision by the South African government. The software product will enable citizens to download application forms and apply online for various necessary identity documents and social grants from the Department of Home Affairs. The system will also allow citizens to launch reports and complaints without the need to know which government agency they should be submitted to. These will be received by the various government institutions, which include the municipal offices based at the Idutywa Business Centre. Dwesa community villagers will also be able to send through announcements on the Dwesa Forum functionality of the system, and to discuss social issues on the forum. What makes this web-based system unique is that it is a rural, user-defined system that integrates a number of government services, making them easily accessible from the rural villages.

The research initiative signifies the beginning of efficient and cost-effective public service delivery at local communal level. The e-government system will be the first initiative applicable to the Dwesa community that consider the little ICT awareness the Dwesa rural citizens will initially have when accessing the portal (Misra 2007). The Dwesa e-government

portal will be the first citizen oriented system to provide online public services to the Dwesa community at local level.

1.9 Brief Chapter Overviews

The second chapter will explore and review prior work by academia and international development organizations. The main reason for this will be to establish the current state of the art in projects associated with the development of e-government strategies. We will explore in particular the history, nature and importance of e-government facilitation in rural areas. Finally, the chapter will look in depth at various e-government access models as well as giving an analysis of each of them. The chapter will suggest a synthesized e-government access model applicable to the Dwesa rural area, drawn from the models available in the literature. The chapter will literally identify the nature of the links that exist among major areas of e-government strategies and priority will be given to these links in the context of this study.

Chapter three will discuss the methods applicable to the e-government strategy to be designed. The chapter will highlight the reasons for implementing the chosen methodology. Requirements and software technologies will be highlighted in this chapter.

Chapter four will describe and explain the management of the Dwesa e-government system. Various designs of administration processes are explained in chapter four. The chapter will give an overview of the system structure and the back-end of the e-government functionality. The design of the Dwesa e-government back-end database is discussed in this chapter.

Chapter five will look into the implementation and design of the three main functionalities of the Dwesa e-government system. The chapter will describe the real core importance of the e-government strategy built. The chapter explains various security features implemented on each of the functionalities.

Chapter six will conclude the study by describing the deployment of the e-government system. System demonstrations done and field test results will be illustrated in chapter six. An evaluation of the research work will also be done in this chapter, listing all the problems encountered during the phased development of the system. The contributions of this research will then be given. It will give some suggestions as to possible future work that can be extended from the e-government strategy developed.

1.10 Conclusion

This chapter gave an overview of the project. It discussed the background and the problem of the research. The project aim and objectives were explained, in line with other projects being undertaken.

Chapter Two

Related Work

This chapter reviews the work currently being done to support the growth of e-government initiatives. It also examines different types of e-government system models already available.

2 Introduction

In every country, each state government provides its rural and urban citizens with a wide range of services. Reffat (2006) states that all governments seek structural and administrative improvements and do not want to be left behind in embracing the electronic delivery of information and services to their citizens. The service delivery methods that have been used for the past years are similar across all governments ICT platforms world-wide. Reffat (2006) suggests that it is vital to review what other people and organizations are doing to solve problems pertaining to inefficient public service delivery. He believes that such analysis will help in the design and building of an appropriate e-government system.

However, much of the available literature deals with e-government far more broadly than a specific focus on access issues and very often from the perspective of changes required in government systems, rather than issues facing citizens (Jakachira, Muyingi & Wertlen 2008). Traditionally, a citizen would have to visit government department offices to get the services they need. Government partners and agencies have moved on to deploying e-government platforms on the internet, which present a convenient channel for government services delivery.

2.1 Definition of Integrated E-Government Functionality

For one to fully understand the concept behind e-government, one must first examine what government is in general terms (Reffat 2006). On its own, government is a dynamic mixture of goals, structures and functions and only when the government begins to use ICT innovations in their day-to-day activities it shifts to e-government (Reffat 2006).

We will split the term under discussion into three parts and provide each with a definition in the context of this research.

Integrated – In this context, the terms *integrated*, *portal*, *one-stop* and *single-view* will refer to an entity derived as a result of aggregating two or more public services in one place. Caldow (1999) defines a portal as a window to an array of web-based content. He argues that once an aggregating strategy is in place, starting with vital public services, it will be easy to add other components such as community services. It is important to reiterate that the development and deployment of an e-government portal should not be held up until all the public services are integrated, as has been noted by Caldow (1999).

E-government – Kawatananga (2006) defines e-government as a scenario where government agencies are working together to use technology so they can better provide citizens and businesses with government services and information. He reiterates that it is not a massive ICT project, as much of it is about establishing common standards across government, delivering services more effectively, and providing ways for agencies to work together using ICT technology (Kawanatanga 2006).

Wimmer and Tamborius (2002) defines e-government as an application of ICT to innovate and modernize the field of public administration. Whilst this definition clearly suggests that government has a role to play in the e-government ICT projects, it is biased in the sense that it not only government agencies that are spearheading the delivery of government services, and it is also not correct to suggest that e-government is not a huge ICT project, when developing countries, for instance South Africa, are still lagging behind in rural ICT innovations.

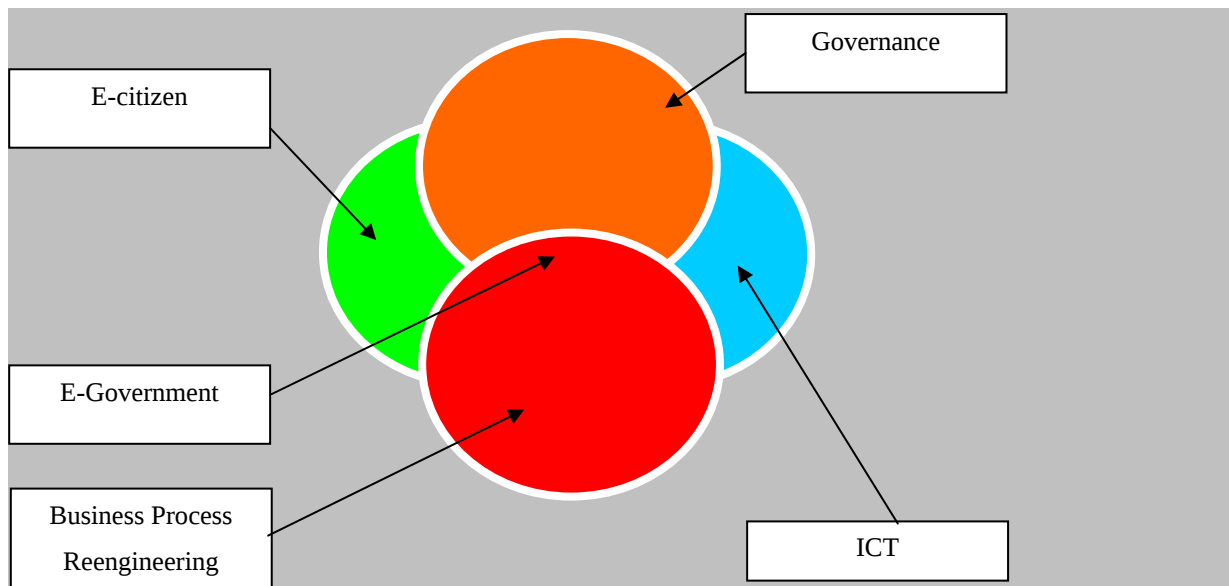
Fang (2002) describes e-government as a method to use the innovative ICT to provide public services to the citizens and businesses. He emphasizes that the most delivery method used is by means of web-based Internet applications.

A more concise definition states that e-government is the use of information technology to free the movement of information to overcome physical bounds of traditional paper and physical based systems (Wikibooks 2008). In other words, e-government refers to the use of technology to enhance access to and delivery of government services to benefit citizens, government business partners and employees.

Functionality – In the context of information technology, functionality is the sum of any aspect of what a product such as a software application or computing device can do for a user. It enables a user to have a set of capabilities around the product. For instant, a software application product can contain a vast number of interfaces or modules that a user can work on, all combined into one system or product.

Therefore, e-government functionality is a software application type of product that uses information technology to bring a group of government services to citizens, and to business partners and employees of government without them physically having to contact the relevant government departments.

In figure 2.1, Misra suggests that e-government encompasses four different sectors. An e-government is derived from the inter-connections of the e-citizen, ICT, business and Governance (Misra 2007). He argues that e-government functionality will not succeed if any of the four components is missing.



1) Defining E-Government (Misra 2007).

This research component deals with ensuring effective online delivery of specific government services to the disadvantaged Dwesa community area in South Africa. The e-government functionality will be deployed, in conjunction with specific South African Departments, on the multi-purpose ICT platform already deployed at Dwesa.

2.2 The Nature of E-Government

In many cases, research has been carried out on how government can make services available online to citizens. Several researchers dwell on how a government-to-citizen relationship can be improved, failing to recognize that electronic government does not only follow a single channel but exists between several groups and state departments. Yigitcanlar (2004) stresses that e-government performs at municipal, regional and national government levels. This is further stated by Peters, Janseen and Engers (2004) that public administration is understood as different types of business processes. They noted that the fragmented nature of government was important to an understanding of the distribution of public administration procedures.

E-government initiatives at local municipal level are highly complex as they encompass every aspect of everything that local authorities and other public service providers do (Yigicanlar 2006). This is a problem, as South African rural communities access public services mostly at local municipal level. Furber (2007) rightly states that several South African local municipalities lag behind in e-government services delivery. This is no coincidence, as explained by Yigicanlar (2006), that maintaining a primary focus on the business of local government, and not technology, is difficult. We note at this point that for an e-government initiative to flourish, technology infrastructure should first find its way into the local municipalities and even deep into the rural areas such as Dwesa.

Whilst providing state services to the citizens through ICT platforms remains a big aim for several governments, there is need to cushion other services delivery channels. This research project considers not only the government-to-citizen service delivery.

2.2.1 Growth Models of E-Government Systems

There are several growth models of the e-government systems. The complexity of an e-government solution can be evaluated along two domains. Gupta & Jana (2001) mentions that the e-government systems follow certain precise stages. Figure 2.2 shows Gupta and Jana's categorization of e-government as it develops in complexity.

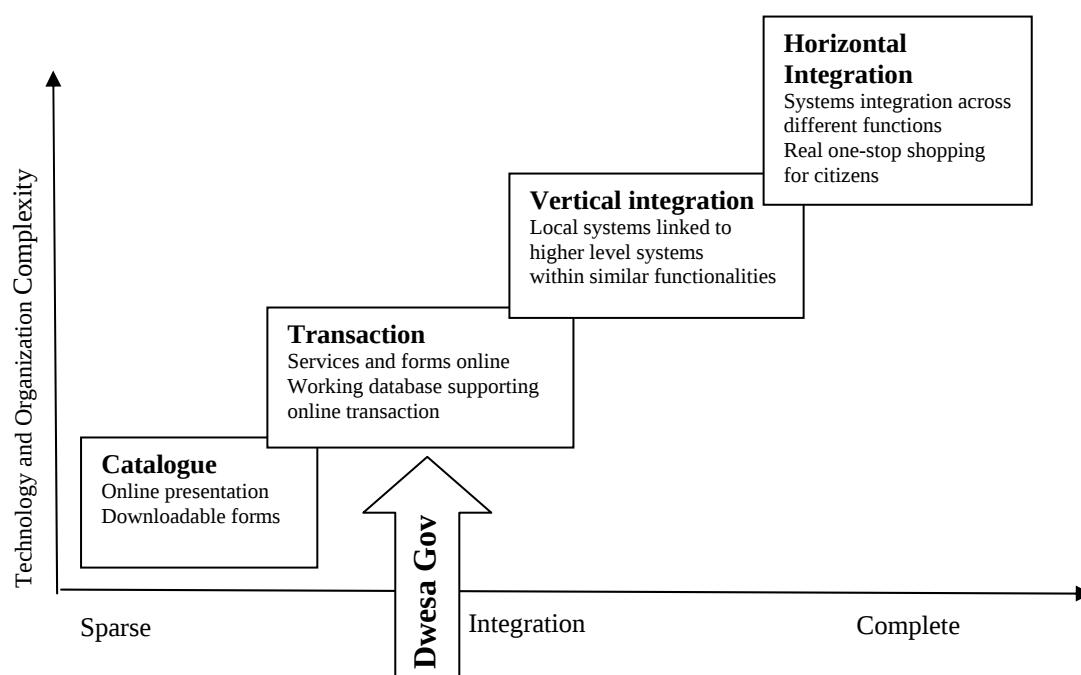


Figure 2.2: Development Phases of an E-Government.

The first stage for an e-government development is a catalogue. Gupta & Jana (2001) describes this stage as marking the beginning of online provision of public services because processes at this stage are not complex. Online presentation of downloadable content such as forms happens at this stage.

In the transaction stage, the process will begin to get complex, as genuine data exchange will be happening between the government and its citizens (Gupta & Jana 2001). At this stage, an online database will provide support for the storage of transacted data. For instance, citizens will be able to fill in the forms and upload them to public institutions.

The vertical integration stage involves local systems that provide similar services, linked to higher-level systems (Layne & Lee 2001). An e-government system at this stage will allow citizens to access a state service from a local-level portal (Reffat 2006). Reffat (2006) says that an e-government system at this stage encompasses local, state, and federal counterpart systems all communicating with each other. The systems could be linking to a central database or several deployed databases at various government departments.

At the horizontal stage, e-government systems that provide different public services are integrated to form a one-stop portal (Reffat 2006; Heeks 2003). Databases that hold data on different functional areas are interconnected to share information. Reffat (2006) says that the information obtained by one agency, will spread throughout all e-government systems. He also assumed that at this stage the e-government system development and growth will be complete.

However, Kamar cites three stages in the implementation of an e-government system.

- Initial 2-way interaction – downloadable forms that can be submitted offline;
- Online transactions – payments, creating and submitting information such as tax returns;
- Comprehensive government portals – one-stop transactions.

(a) Gartner's Four-Stage Model

A four-stage model involving web presence, interaction, transaction, and transformation was proposed by the Gartner group (Siau & Long 2005). The four stages are explained below

- Web presence – at this stage, government agencies provide a website to post necessary information to the public.
- Interaction – at this stage, citizens are able to communicate with government departments through websites (e.g. e-mail) or do self-service requests (e.g. download documents).
- Transaction – at this stage, citizens can perform entire transactions online.
- Transformation – at this stage, operational processes are transformed by governments to ensure that the services are efficient, integrated, unified and within the constitution of the country.

(b) Deloitte's six-stage model.

The model proposed by Deloitte and Touche ensures that the e-government strategies are to serve citizens as customers and to build a long term relationship with citizens. The model consists of six stages and is described below (Siau & Long 2005).

- Information publishing/dissemination – the citizens' access to information is increased by government departments.
- "Official" two-way transaction – information and communication technologies such as digital signatures and security keys are used to secure the interaction between governments and users through agencies.
- Multi-purpose portals – government uses a single portal to provide universal service across multiple departments.
- Portal personalization – citizens have the choice of customizing portals according to their desires.
- Clustering of common services – government enhances collaboration and reduces intermediaries in order to provide a smooth unified and seamless service.
- Full integration and enterprise transaction – every citizen is provided with sophisticated, unified and personalized services, according to their own needs and preferences.

(c) Hiller and Belanger's five-stage model and Moon's five-stage model

Hiller and Belanger identified a five-stage model – information, two-model communication, transaction, integration, and participation. Moon adapted the model proposed by Hiller and Belanger. The model he suggested consists of the following:

- Simple information dissemination (one-way communication). This is the most basic form of e-government, which spreads the information by simply posting it on the websites.
- Two-way communication (request and response). The government departments and the citizens will be interacting in a user-friendly way.
- Services and financial transactions. Transactions occur both between governments and citizens (G2C) and between governments and business (G2B).
- Vertical and horizontal integration. This stage combines (Layne & Lee 2001) the last two stages of their four-stage model. This stage ensures that separate systems at different levels (vertical) and from different departments (horizontal) are integrated.
- Political participation. This stage promotes citizen participation in politics through services such as online voting and surveys.

The models explained above will help us to derive a synthesized model by combining them. A synthesized model will form common framework for this research and a common point of reference (Siau & Long 2005).

2.2.2 The Meta-Synthesis Approach

Meta-synthesis is a research method used to produce interpretive translations, ground narratives or theories by integrating and comparing the findings or metaphors of different qualitative studies such as described above. We have chosen to follow in this regard the meta-synthesis approach with a view to deriving a common approach that contextually adapts to what we aim to achieve (Siau & Long 2005).

In our study, we use the meta-synthesis approach to compare, interpret, translate, and synthesize different research frameworks. The approach when used can achieve interpretive synthesis, rather than simply an aggregative summary of the findings. As a relatively new approach, meta-synthesis is not widely used in the area of information systems management. However, we have decided to facilitate the theory-building procedure through systematic synthesis. In our case, we are trying to arrive at a synthesized model which depicts the e-government development process. The model will be comprehensive enough to include the main ideas of previous models. The overall vision of e-government strategy is highlighted

- i. *Web Presence*: This phase will provide the most basic form of e-government. Limited information and downloadable forms will be posted through the e-

government websites. At first, most of the information will be seen as static. However, the advancement of e-government capability will ensure that citizens will have dynamic information posted and regularly updated. This phase does not involve user interaction as do other higher phases.

- ii. *Interaction.* Citizens will interact well with the government under this phase. The citizens will get official form downloads, online crime-reporting systems, etc. Interaction can be regarded as a process development which the e-government system will undergo to shift from a simple web presence to complete transaction.
- iii. *Transaction:* The phase enables complete online transactions to be conducted successfully by citizens. Citizens can perform self-services online, such as identity document applications.
- iv. *E-democracy:* This phase is about how the citizen interacts with government or influences the legislature or public sector process.

Vosloo (2005) notes that most of South Africa's e-government initiatives fall into the initial stage of cataloguing. He predicts that a move towards the more advanced e-government stages was already underway.

2.3 Major areas of E-Government Development

The development of e-government can be visualized in four major areas, which are government-to-citizen (G2C), government-to-business (G2B), government-to-government (G2G), and government-to-employee (G2E). Among the four areas, G2C and G2E enable cooperation and interaction between government and individuals, while G2B and G2G promote the relationship between government and organizations. A description of each area of e-government development follows.

2.3.1 Government-to-Citizen (G2C) Service Delivery

No doubt the most common type of service delivery channel implemented over the years, and found in several provinces, addresses only a part of a country's population (Xavier & Gupta 2003). Fang (2002) insists that G2C service delivery provides the momentum to put public services online, in order to provide information and communications. Whilst many governments have different formats for their e-government platforms, certain innovations are needed to differentiate the category of citizens. Xavier and Gupta (2003), for example, states that there was a need for innovations to the e-government initiatives so that it suits the Indian

population situated mainly in rural areas. They mentioned that e-government initiatives in India face the twin challenges of automating government departments and taking on-line services to the common man.

South African rural area citizens face the same problem, as e-governance is still in its infancy in this country. There have been a number of G2C e-government functionalities deployed in various provinces across South Africa. However, closer analysis shows that there is a dominance of e-government systems in only two provinces, namely the Western Cape and Gauteng. According to the information released by State Information Technology Agency (SITA) in 2002, G2C governance was commonly done in the following six examples of initiatives in South Africa:

Kwa-Thema Information Kiosk – An e-government portal launched in Gauteng Province in 2006 to provide a one stop service point from which communities in Gauteng can access various public services and products. The information stated on the government site stated that the portal is easily accessible to the nearby Gauteng communities (Info Gov Site 2009).

KHANYA – The Khanya Project is an e-governement initiative of the Western Cape Education Department. It was it established in 2001 to determine the contribution that technology could make towards addressing the growing shortage of educator capacity in schools. It has provided each school with a computer for students to access internet and e-mail facilities (KHANYA Org, 2008).

Gauteng Online – This is the Gauteng Provincial Government's internet portal, launched in 2005. It aims to give members of the public easy access to certain government services and information and is an expression of the provincial government's commitment to build a caring and effective government. It provides internet access to all schools within the province.

In Gauteng, a driver's license booking system has been introduced. E-recruitment service is also provided that enables online application for Gauteng Provincial Government jobs. The portal provides a status guide for ex-employees to view and track progress on termination without having to visit an physical office. It also offers an online employee leave-balance enquiry with the aim of giving employees immediate access to leave information and reducing the number of queries that need to be dealt with by its staff.

Education services such as a schools search facility, viewing and retrieving grade 12 results and the ability to view and download past examination papers are also offered on Gauteng Online. This includes a children's portal for playing games and allowing other activities for children. Moreover, the service provides information on social welfare services through the Batho Pele portal.

Whilst this portal has a wide range of services for citizens, the major drawback of this entire project is the fact that whilst it is deployed online and accessible to everyone in the country, it only applies to Gauteng citizens. In areas like Dwesa, citizens are not aware of these services, not to mention that they do not have access to computers and the internet.

PALS – Provincial automated library service of Western Cape. The portal provides only the comprehensible library services mainly in the Western Cape.

With the exception of Cape Gateway e-government initiative which has successfully presented the public services in three languages to Western Cape citizens which are English, Xhosa and Afrikaans, most e-government portals still have too much information presented in English.. The strategies failed to realize the importance of localizing the information centres. Thus, when strategizing the e-government systems, efforts should be put to create a system model that is easy to modify and that appeals individually to every citizen. The Information Kiosk e-government strategy had its contact information provided; responsiveness to e-mails was poor or non-existent. Most of these e-government strategies had outdated information and rely enormously on *pdf* files (Furber 2007).

In central India, the Gyandoot project installed Internet kiosks in rural villages staffed with trained locals who helped villagers print land records, apply for government documents, and access information on poverty programs. Further, ICT tools enable governments to enact programs more effectively, for example by allowing much more accurate, up-to-date maps of poverty, and transferring that data quickly and uniformly throughout the public sector (CEG 2000).

2.3.2 Government-to-Government (G2G) Service Delivery

The G2G service delivery model is now common between South African State Departments. According to information released by SITA (2002), a single G2G government functionality has been deployed and is in full use in South Africa. The Doc Warehousing is an e-government system derived from integrating 102 systems to enable information sharing for ministers in various state departments. The involvement of different government organizations, departments, and authorities in ICT communication has enhanced communication between them. According to the Wikipedia website, the model is common in United Kingdom, though South Africa has been taking initiatives to strengthen the channeling of information among its departments and ministries.

The G2G systems fall into two categories namely, internal facing and external facing. Internal facing e-government portals help to improve internal efficiencies and many other critical tasks between local governments, and between department-level and attached agencies and bureaus. External facing government portals provide a link between two or more governments' IS systems. External facing portals can also be used as an instrument of international relations and diplomacy (SITA 2002).

2.3.3 Government-to-Business (G2B) Service Delivery

Siau and Long (2005) notes that government-to-business involves external communication and collaboration between government departments and outside business institutes. This service delivery model exists between local and central Government and the commercial business sector for online non-commercial interactions. The services channeled through the G2B model include dissemination of policies, memos, rules and regulations (Siau & Long 2005). Business services range from obtaining current business information to payment of taxes (Reddick 2004).

2.3.4 Government-to-Employee (G2E) Service Delivery

This area of e-government development, as explained by Siau and Long (2005), promotes cooperation between governments and their employees. The interaction between the governments and employees is internal (Siau & Long 2005). The Hong Kong Immigration Department proposed in April 2004 that an e-Leave application would be introduced as a government-to-employee flagship project. The Department would implement the web-based

new application to process leave applications as well as calculation and recording of leave records.

2.4 Benefits in the Adoption of Electronic Government

The adoption of electronic government has benefited both the state and its citizens.

2.4.1 E-Government's Benefits to Citizens

- **More Convenience.** It will be possible to access e-government services in any location with an internet connection at any hour of the day or night that is convenient to the citizen.
- **Better Customer Service.** The online delivery of services may be more user-friendly than that delivered by traditional means. Instead of getting in line, citizens can get online.
- **More Information Access:** Deployment of information in the real-time public web servers guarantee that the fetching of information can be done at any time. The access is not restricted to business hours.
- **Equal Opportunity to Access Information:** Kamar and Millicent (2007) states that citizens will have equal accessibility to the information despite geographic location or physical disabilities. They reiterated that barriers of distance are overcome with diversified service points (Kamar & Millicent, 2007).

2.4.2 E-Government's Benefits to the State

Decreased Cost. Undertaking an e-government strategy will bring dividends in terms of lower costs. Government will save on subsidies as the online transactions are much cheaper than traditional ones, with significant savings in paper, printing, postage and associated staff costs.

Increased Efficiency. Better delivery of information and services has a potential to improve the uptake of services and increase citizens' satisfaction with government services.

Increased Esteem: Ireland government demonstrated leadership potential through the development of the Public Services Broker model, as well as through its performance in the EU benchmarking of available on public services.

Reduced Corruption Cases: As rightly noted by Kamar and Millicent (2007), the accountability and transparency of e-government transactions reduces corruption in state departments. This is because all transactions are done systematically online and this allows users to present a proof and backup when problems occur.

2.5 E-Government Initiative Perspectives

To reap the benefits stated above, an e-government system should not be allowed to fail. Many e-government strategies fail before they have started. Heeks (2006) states that it is imperative to draw perceptions of e-government from the point of view of government and citizens. From these perceptions, vital public services can be planned and implemented to the satisfaction of citizens.

2.5.1 Citizen Perspectives

Reffat (2006) states that citizens expect government to operate more like commercial entities. He reiterated that access to the e-government services should be virtually open to all citizens, available at all times and convenient to use. Citizens do not want restrictions on accessing the services although they want their information and content to be secure against outside intruders.

It is important to note that citizens are not interested in the personal details of the public official who is responsible for a specific government program or public service (Reffat 2006). For instance, when filing grievances to a public office, citizens are not interested to know who will receive the report or where will it be received. Caldow (1999) further states that citizens are not worried about the name of the department that will actually process their transaction.

Citizens prefer personalized public services. Over-the-counter service delivery ensures that a citizen communicates in his own language to the public official behind the counter. Despite the fact that official languages are taken as communication bridges between two individuals of different ethnic groups, citizens expect the public official to communicate in a language understood by both of them. As stated by Tolbert (2004), e-government can facilitate more individualized communication between citizens and government agencies.

Because of this, citizens need an e-government service delivery portal that meets their communication needs. E-government initiatives already deployed in Gauteng have their content written only in English (Furber 2007). Reffat (2006) agrees that to provide personalized services to its citizens, government should provide information and services within a single integrated source.

Reffat (2006) points out that citizens who do not have time to go to committee hearings or village meetings in order to participate in public debates, can send an email or contribute to an online discussion forum. The participation in these public debates by the rural villagers will be relatively low for the next few years, and it will be gradually boosted by the growing of e-government readiness among community villagers. Thus, Reffat (2006) mentions the need to create a digital community for citizens. It should be noted that such a digital community will alleviate communication problems and help to spearhead the public's ICT awareness, especially in rural areas like Dwesa. Similarly, Caldow (1999) points out that the internet has brought a new dimension to the traditional meaning of community.

2.5.2 Government Perspectives

Public institutions view e-government as an initiative to improve quality of public service delivery among citizens (Reffat 2006). Reffat (2006) underlines that governments are likely to embrace any e-government initiative because they assume that it will assist in regaining the trust of the citizens. He states that rebuilding a relationship with citizens requires provision of services without long waits and complex procedures. Reffat (2006) insists that public institutions have strong perception that citizens require reliable public services, but these institutions do not value the time spent by a citizen in accessing public services.

It is then accurate to conclude that government values the availability of public services more than the efficiency of service delivery. Reffat (2006) mentions that an e-government initiative is likely to be welcomed by staff in public office as it will reduce employee time spent on non-citizen related activities.

2.6 A History of E-Government in South Africa

South Africans have had to obtain government services by visiting the respective government departments until recently, when households could get access to computers. To obtain a social grant application form, a citizen would have had to visit the nearest government office. In 2003, 13.6% of households owned computers although only 9.1% of all households had access to the internet (Furber 2007). In 1992, only one in 778 people could use the internet; and by 2002 the figure had risen to one in every 10 South Africans (CSPI 2003).

By 2006, the number of South Africans who had access to the internet had shot up drastically to 3.56 million. The prices of computers remained a barrier to private access, and few ICT e-

government systems were available at the time. Those that were running could only provide minimal services to the limited number of South Africans who had internet access (Louis 2007).

2.7 E-Government Impacts in Developing Countries.

E-government impacts in South Africa can be classified into two categories. The impacts depend on whether the e-government project has been successful or not. Furthermore, the contribution of these e-government projects differs with the sectors in which they are running. Basically, their contributions vary between the following sectors.

- **On Economy.** When an e-government project succeeds, it brings the whole benefits together with growth in economy. Heeks (2003) assumes that failure of e-government does not necessarily have negative impacts on economy. He points out that there are valid indirect benefits associated with a failed e-government project. He mentions that infrastructural foundations would have been layered in the event that the whole project fails. This will, he explains, be the foundation of future projects. However, the economy will directly suffer from opportunity costs. The money that could have been spent on the project would have been channeled to solve some alternative problems faced in the country (Heeks & Anul 2006).
- **On Governance.** The success of an e-government brings all the benefits associated with to the government. Reffat (2006) mentions that adopting an e-government initiative can re-build government-customer relationship. Government can use new technology to treat citizens as individuals and provide personalized services (Reffat 2006). In the event that it fails, Heeks (2003) states that whilst the governments endure political costs, for instant loss of image, it also benefits indirectly. Heeks and Anul (2006) points out that skills acquisition from a failed or successful e-government project reduces the number of unskilled personnel in the country. The state organizations will benefit from situational learning done by the project team. Heeks (2003) explains that the process of analysis and design can help better understanding of organization's processes, structure and culture.
- **On social life** There are enormous positive impacts of the e-government strategies on the social life of the citizens. The provision of services online streamline the problems associated with gender and race differences.

- **On democracy.** Reffat (2006) argues that e-government can lead to more direct democracy. This is, according to Macintosh (2004), achieved through citizen participating in debates and discussions forums supported by municipalities such as Idutywa. Municipalities can then use the functionalities to inform citizens of decision-making processes.

2.8 E-Government Challenges in South Africa

E-government initiatives face serious challenges in South Africa when compared to e-commerce and e-learning. The implementation of e-government requires the participation of various stakeholders and parties. Below, we describe some of the problems faced in moving from the development phase to the serving stage.

Initiatives and access: As Furber (2007) states that, in his ITWeb report, preliminary results of a study into the effectiveness of e-government at the municipal level in South Africa's Gauteng province show that most municipalities are still only at the very early stages of e-delivery. The chief reason is the cost of software packages.

Political Issues: Several governments have hesitated to embrace e-democracy because of political constraints. This applies even to developing countries such as South Africa. The influences of citizens on legislative or public sector processes remain a key to democracy in many countries. E-government can even play an important role in electronically delivering citizen's participation in government legislature. The incorporation of online voting into any e-government ICT initiative will enhance the credibility of the results of any such government polls. However, it should be noted that in several instances possible abuses by states have cast doubt on the authenticity of such ICT processes.

Social Issues: Rural dwellers suffer from a lack of information. Kamar mentions that there is reluctance to share information among rural villagers. Many who reside in rural areas think that a computer can only be used by educated people (Kamar 2007).

ICT Resources: Most rural dwellers in South Africa have low information technology literacy. There is uneven distribution of internet facilities between rural areas and urban centres in South Africa (Dalvit *et al.* 2007). Access to services and infrastructure is skewed towards cities and towns. If they are available, internet facilities will be limited and costly for the ordinary

citizens. Insufficient funds from the government hinder the spread of ICT awareness in remote areas (Kamar 2007).

2.9 Potential Models of e-Government Access in South Africa

A CPSI e-government access study released in 2003 by South African Department of Public Service and Administration and the SITA cites seven e-government access models adaptable to South Africa. Below, we review each of them.

2.9.1 Model One: Smart Service Gateway Service Points

The model enables the provision of much needed and necessary services to citizens. Future extensions that may be needed on the services modeled around the Smart Service gateway can easily be made. The study notes that while the model is fundamentally transactional with a degree of interactivity, citizens can also potentially access limited government information from the point at which the transaction was done. In the initial phases, the study suggests, previously prepared information can be printed off the terminal device (CPSI, 2003).

Technology appropriate for use in this model is the same as that of widely accepted ATMs, and the use of debit cards and smart cards at point-of-sale devices could become popular. The study reported that, using the model, the Department of Home Affairs had taken the initiative of designing a citizen smart card to provide a major platform for e-services and e-transactions. More and more Public Internet Terminals have been deployed at Post Offices and these could serve as Gateway Service Points with a more limited range of services. The e-government model is best suited to cities, towns and townships where access to technology is not a problem. This model would also be affordable to customers.

Our argument against this model is that the provision of technology to local municipalities at e-government access level is very limited.

2.9.2 Model Two: Smart Plug-In

This model foresees a closer interaction with the extensive telecommunications and ICT networks of agencies. Currently, South Africa boasts the highest deployment of ICT platforms in the continent. Agencies such as Uthingo, the Post Office and Post Bank have a wide reach at local level throughout the country. The model emphasizes the need to integrate some plug-ins with, for instance, the Uthingo network (CPSI 2003).

While the model has a good geographical reach, it does not meet the objectives of convenience, improved service, efficiency and specific service enhancements. Moreover, the study reports that this type of e-government access model is relatively expensive for government and other agencies to implement, while it should carry no cost for the citizen.

2.9.3 Model Three: M-Services

The model embraces the rapid expansion of mobile subscribers in South Africa. Currently there are 14 million mobile subscribers in South Africa, the report says. Hence, it is a cornerstone to the implementation of e-government functionalities. One such example of e-government strategy rooted within this model is the rapidly growing popularity of cell-phone banking being implemented by banks and mobile phone companies. Government could also integrate e-government functionalities before the mobiles reach the customers. The model addresses the objectives of extension of services to under-serviced areas (CPSI 2003).

The problem that most e-government providers face when adapting this model is the limited display and networking capacity of mobile devices (Poslad *et al.* 2001). A solution suggested by Poslad *et al.* (2001) is to create an adaptation of services and contents to individual interests through a filtering process. They argued that such a process will be based on the user profile, which gives the description of his interests, abilities and characteristics. We note that this is not a viable solution that can guarantee the use of the M-Services model. Poslad *et al.* (2001) argued that the perspective of a citizen is to get public services in less time and at less cost. They argue that citizens do not want to surf through a large number of web pages. Even so, Poslad *et al.* (2001) reiterates that a huge problem facing their proposed e-government model is the acquisition of knowledge about citizens before applying the filtering process.

Furthermore, citizens would face high costs of using this model of e-government system. Moreover, service level agreements with service providers are needed, or institutional management of such deployed services. The model requires effective management at government level. Therefore, this model contradicts the aim of achieving independence of systems that are in remote parts of the country.

2.9.4 Model Four: Government Online

Public internet access in South Africa is increasing with the availability of initiatives such as MPCC telecentres, Gateway Service Centres, the Citizen's Post office, public libraries, school-

based access and other initiatives. The study reiterated that the model has a potential to give an e-government strategy that will cover all the objectives of e-government. In the early stages of the model, there will be limited access, therefore limited ability to address the objectives of extension to under-serviced areas and of citizen convenience.

The model offers efficiency and convenience. Moreover, with adequate resources, the model can provide much better access for under serviced areas such as rural parts of the country. Major institutional and management challenges in moving government services online, and the training and orientation is the prerogative to the full usage of the services. Having saying that, our study has resolved to use model four as the e-government access model applicable to Dwesa rural area.

2.9.5 Model Five: Centre Services Gateway Service Centre

This model is reported to be the most visible face of e-government service delivery. The Gateway Service Centre will provide one-stop interaction to the citizen. The model emphasizes the need to consider the location, which should ease the citizens' movements, proximity to other services and traffic flow. However, the study reported, the model realizes only the need for centres to be confined to high population density areas. It is therefore more applicable in big cities and high density suburban areas (CPSI 2003).

2.9.6 Model Six: Talk-to-Government plain old telephone.

Apart from being an appropriate model for under-serviced areas, this model has the capability of transactions if there is a payment system in place with effective authentication mechanisms (CSPI 2003). The study suggests that the model can be referred to as an information service model, which can be based on two applications – IVR and call centres. The application of IVR will be best suited to rural areas, the study notes. However, many villages have extremely low levels of telephony and consequently also of connectivity for the purpose of electronic communication, such as, for instance, in the Dwesa rural area, in the Eastern Cape.

Moreover, the model has a high cost of initiative and strategy design. Furthermore, business planning has to ensure that risk analysis and risk mitigation measures are in place. The study reported that the component of the strategy is best placed to be initiated by departments and agencies of the government (CPSI 2003).

2.9.7 Model Seven: Computerized Counter Services

The model is a key one when e-government is in its initial stages of implementation. The main objective of strategies behind this model is to reduce repeated visits by citizens. However, the location of the e-government functionality will still be in government offices, though computerized. This conflicts with the objective of e-government strategy to bring the services closer to the citizens' doorsteps (CPSI 2003).

2.9.8 Proposed E-Government Access Model

The seven models described above each has its own advantages and disadvantages. Some models ensure lower costs of initiating e-government projects while others prioritize service value and quality before money. Few of the models require a development timeframe of between 18 and 24 months, while others, as the study noted, such as mobile e-government may take anything from 5 to 8 years. Our study has noted that several e-government strategies are skewed towards one model, which is not sufficient for the development of Gateway and long-term e-government services.

Our study strives to serve remote rural communities and effort is needed to come up with a strategy that conforms to at least two models. From the literature reviewed on the e-government access models, model four and model seven are well placed to strategize our e-government service in the rural parts of South Africa. We have noted the importance of Public Internet Access through initiatives such as school-based access and the Gateway Service centre. We have realized that the internet, as a model for e-government access, will offer efficiency and convenience for under-serviced areas in the Eastern Cape, because our strategy is still in the early stages. We have resolved to initiate the e-government strategy through the computerized counter services model to provide an entry point for government to respond to the citizens' complaints and their needs.

2.10 Conclusion

The conclusion drawn from the work assembled and reviewed suggests that a successful e-government strategy will be possible if a well-synthesized e-government model is implemented. We have reported on the nature of e-government strategies, the major areas in the development of online service provision systems and on the certain key focus areas where government services to citizens could benefit from the application of ICT.

The work reviewed cited the impacts of e-government strategies not only on citizens but on the government and the business world. Thus, we have concluded that e-government access is a complex issue, affected by several factors, not only technological options.

The review of the work has also concluded that the following issues should be considered in an e-government strategy in order to comprehensively cover the issues of access in the short to long term.

- Adopt a sufficiently broad concept of access in the design of the e-government system.
- The strategy should present a clear direction to achieve “real access” in South Africa.
- Vital to the success of e-government strategy is the development of an understanding of what citizens want – in terms of specific service enhancements, efficiency, service and convenience.

Having discussed the nature of e-government initiatives in this chapter, the arguments stated supported the proposal of the nature of an e-government system required in the Dwesa community area. Of the three e-government access levels discussed in this chapter, we noted that our e-government initiative will provide public services accessed at a local municipal level. This will help us to identify requirements fit to design an e-government system applicable to the Dwesa community. We then discussed four areas of e-government development. Government-to-Citizen service delivery was argued to be the best, as it describes the partners involved in the provision of public services – the South African government and its Dwesa citizens.

The study of the existing information about models of the e-government systems in South Africa helped us to choose a suitable model for the e-government system for the Dwesa community. The Government Online model, the fourth of the seven models discussed, was chosen for the reasons stated in this chapter.

We have looked at government and citizen perspectives of the provision of public services. Several governments highlighted in this chapter are concerned only about provision of more public services to the citizens. Citizens do not want just provision of services but they also consider their efficiency and reliability.

Chapter Three

System Methodology, Requirements and Software

Technologies

This chapter will give an overview of the software methodology, requirements and software technologies which will be utilized in the development of the e-government system

3 Introduction

This chapter will discuss the system methodology and the software technologies that will be used to implement the e-government system. Our methodology should reflect the realities in business: speed, cost effectiveness, and quality so that the e-government system will not have to wait years for competitive improvements.

Various factors and constraints bound the choice of hardware and software tools. These include the size of the system, type of users, financial constraints, and the network bandwidth usage and type of services provided by the system. Further discussion of these tools and the functionalities that come with the system follows below.

3.1 Background of the Dwesa E-Government Initiative

The e-government functionality will be deployed in the Dwesa communal area upon completion. Dwesa is a rural community located in the Eastern Cape Province of South Africa. As this project is part of an on-going major ICT project, the e-government system will utilize the ICT infrastructure already deployed at the four schools located in the area.

Mpume School: Because of the initial presence of electricity and its location, the school was chosen as the main point of presence for the major project (Dalvit 2007). The access of the e-government portal will require the presence of an internet connection. The VSAT CPE installed at this school will provide this. Other equipment includes a WiMAX CPE, a server (LTSP, HTTP and MySQL), 6 client PCs, an 8 port DLink switch and a VoIP phone (Dalvit 2007). The e-government system will be deployed on the web server available at this school.

Ngwane School: This school is not very far from Mpume. Dalvit (2007) wrote that this school was also chosen as one of the schools where a WiMAX CPE was installed. He stated that unlike other schools, Ngwane had sourced their computer lab. This will be one of the points the Dwesa villagers will access the South African public services delivered by the e-government system. The lab has 20 PCs and a printer. This printer will be utilized to print government forms so that villagers can fill them in by hand, if needed.

Ntokwane and Nondobo Schools: The VoIP and Internet services at these two schools are provided by Mpume School (Dalvit 2007). There are client PCs and WiMAX CPE at these two schools.

The two prerequisites for the e-government platform to be effective are an internet connection and basic training in computer skills. Internet connectivity is provided to Mpume via VSAT and then extended to the other schools via WiMAX CPE (Dalvit 2007). The basic training in computer skills will be given by a group of young researchers from the University of Fort Hare and Rhodes University who pay regular monthly visits to Dwesa. The training lasts for approximately one week each time.

3.2 E-Government Software Development Method

The Dwesa e-government system will be developed using Rapid Application Development (RAD) and incremental prototyping methods. Thus, when we chose the Rapid Application Development method we sought a development path that will attack mostly quality and time challenges head-on by providing a means for developing systems faster, while reducing cost and increasing quality (Benyon-Davies *et al.* 1999). The incremental prototype method promotes the development of large web applications in phases. The combination of the two methods will yield favourable results. The latter method will avoid the delay between the development of the e-government system and its delivery (Howcroft & Carroll 2000).

3.2.1 Aspects of RAD Methodology

The aspects of our RAD methodology are:

- Fixed time delivery.
- Understanding and mastering the dynamic, changing environment the system will be on.
- Using evolutionary prototypes that are eventually transformed into the final product.
- Using workshops, instead of interviews, to gather requirements and review design.
- Usage of demonstrations, instead of interviews, to gather requirements and reviews (Howcroft *et al.* 2000).

3.2.2 The Advantages of RAD Methodology

Users are now regarded as shapers of successful systems. Rapid Application Development (RAD) methodology is said to be most suitable for applications with a strong element of user

interface and with a lack of complexity regarding both requirements and computation (Mackay *et al.* 2000).

Apart from this, the method is ideal because it uses object-oriented programming methodology and allows the release of working prototypes at frequent intervals (Karen *et al.* 2001). This ensures that the working prototypes can be deployed briefly and tested. This allows splitting of the e-government prototypes into several independent modules that can be revisited and modified individually with no major changes required for modules links.

The method is ideal for e-government systems development because the e-government systems go through several phases (Karen *et al.* 2001) during their development life cycle. The e-government system that will be developed will strive to meet the transactional phase, i.e. the second of the four phases available. The shift from the first cataloguing phase should be facilitated through use of the RAD method.

Figure 3.1 suggests that when the RAD system development method is used, the system prototypes can be iterated to launch working versions of the system, thereby saving the resources for system development (Marner 1997). The construction iterations occur within a fixed time-frame that regulates the resources that can be expanded.

The RAD process requires heavy user involvement of customers in the phases of planning, analysis, and testing, and then over and over in the prototyping process.

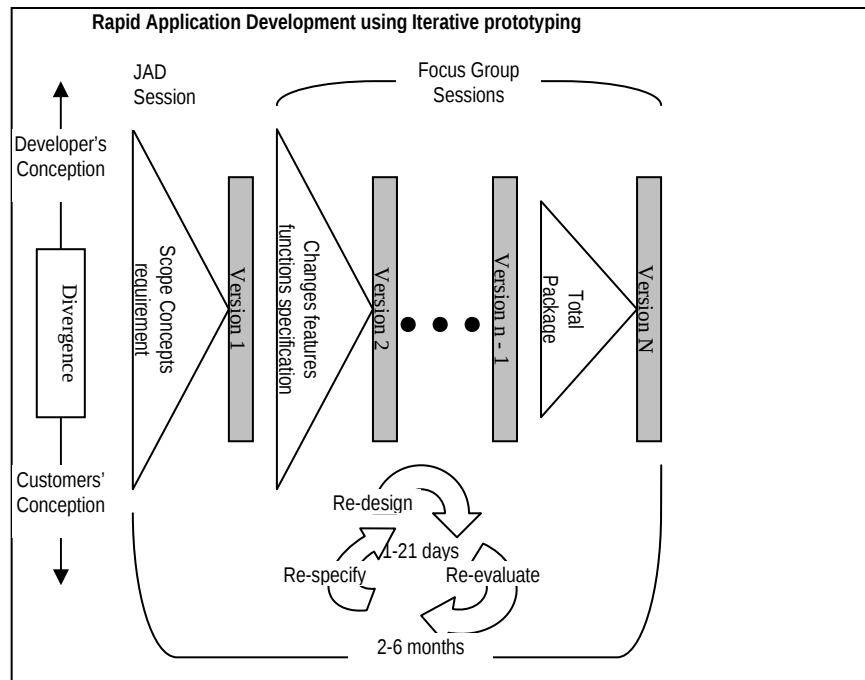


Figure 3.1 The Rapid Development Method (Marner 1997).

Figure 3.1 shows the iterative approach used by RAD to design, build and test activities in the system development stage. The construction process supports the implementation of the requirements according to their value or risk and the incorporation of user comments and requirements into the system (Marner 1997).

The construction tasks include:

- **User interface design:** The RAD method specifies that the user interface should support the government processes in the development of e-government strategy. It incorporates the design of the components visible to the user (Marner 1997).
- **Processing flow design:** In this stage, the RAD method enables the development of web applications system to enter phases, making it easy to evaluate each prototype in the field. This stage is taken as an iterative process.
- **Database design:** Data and process network distribution design. Data and process distribution are primary concerns associated with client/server systems such as web-based ones. The task is vital as the data will be centrally located (Denr 1999).

The RAD methodology dictates conclusion with the roll out stage. This step follows the completion of the construction phase. This step has processes that create procedures that should specify training programs for the users and convert the application into production quality, shifting the development process into a production mode.

3.3 E-Government System Requirements and Assessment

The Dwesa e-government system will be a web-based system that caters for online provision of government services. The system integrates different citizen portals that offer various functionalities. The selection of e-government system functionalities will be best done by comprehensive review of related case studies as well as from the wide range of needs gathered from interviews with the Dwesa rural people. During the numerous visits to the Dwesa rural area, we managed to interview the villagers and to run some questionnaires with the teachers and pupils of Mpume and Ngwane Junior Secondary schools and their suggestions were noted and considered in the development of the e-government pilot system.

It is during these interviews that we gathered the importance of developing an e-government initiative that would meet the villagers' needs. In the interviews, we noted their need to obtain important applications forms from government offices, hence the Dwesa Online Application Centre (DOAC) was developed. We realized the critical processes that the system should handle properly, which include the ability to file a grievance complaint and to monitor its status, and the ability to apply for a variety of government certificates (Pujari 2002). We also noted the lack of communication between citizens and municipal officials should there be anything that needs to be brought to their attention through the Dwesa Online Reporting Centre (DORC). We also gathered from the interviews that 80 percent of the Dwesa rural villagers spend less than R20 a day on most days. Hence, the need to develop a cost-effective e-government system based on open-source software. During the visits we also learnt the type of networks deployed and the system distribution structure.

Based on these observations and facts gathered, the e-government system will be developed to run on an open-source Ubuntu Linux Operating system. The Dwesa e-Government system became a web-based software application after we realized the need for system distribution among several client machines.

The system, running on an Apache web-server, will be written in PHP scripting language and will make use of database tier of a MySQL 5 database tier. Thus, the system is modelled with complete LAMP software. Further observations suggest that the system will require to be modified in future and extra functions added to achieve the final phase of the e-government development. The system follows the MVC framework policies (Nerisa *et al.* 2007).

3.3.1 E-Government System Functional Requirement

Informal interviews were conducted at the initial stage to determine the public services that will be provided by the e-government system. Students, teachers and community villagers were picked randomly at the four different schools in the Dwesa village to participate in the interviews. Participants were merely asked the public services they would want from the government. Some of the questions asked were later included in the evaluation questionnaire listed in Appendix D. Feedback from the users included the difficulties they face to access the public services at the Government departments in Willowvale. From the feedback attained from these informal interviews we managed to put them into four categories of public services. Each category will form its own component module in the development of the first e-government prototype.

Initial requirements attained from the interviews were firstly used to design a prototype with RAD method that was used to get more user requirements of the final e-government system. After the deployment of the first prototype a questionnaire was administered after the to get more e-government system requirements. The questionnaire is listed in Appdendix D.

The e-government system to be developed will integrate four functionalities, and extra functionalities can be added easily to the current system. Drawn from the interviews conducted taken in the field, the four components the system will have are as follows:

- Dwesa Online Application Centre (DOAC).
- Dwesa Online Reporting Centre (DORC).
- Dwesa Forum Corner (DFC).
- Forms Downloading Portal (part of DOAC).

Each functionality will have its own features and some features will be shared among them. The functionalities will be developed to ensure the best delivery of e-government services to the Dwesa villagers.

3.3.2 E-government System Non-Functional Requirements

The following system development factors will be considered throughout the implementation of the e-government system.

3.3.2.1 Users System Permissions

The e-government system allows three types of users to access the system functionalities. The main user is the Dwesa villager. The administrator and the third party agents will also have access to the system. The third party agents are the people who will process Dwesa villagers' application forms and respond to their complaints and administrators will manage the e-government system.

3.3.2.2 Secure Transactions

The first requirement for e-government is for it to be extremely robust and reliable. Much work has to be done to ensure secure uploading and downloading of application forms. The users who participate in uploading and downloading of files are grouped as follows

- The system will be able to automatically provide government form updates. Administrators may have to upload up-to-date application forms to the system.
- Dwesa Villagers will download important forms from the system and fill them in online before uploading them to the government department. The users will be able to save their application forms on the system and the system's Dwesa Online Application Centre (DOAC) service will forward the transaction to the Home Affairs office through a proxy.
- Government officials will have the filled-in forms on their servers, and will need to upload them back to the villagers should any corrections need to be made before the processing of the application.

Therefore necessary software tools will be used in the development of the e-government system to ensure secure environments to access and utilize the form files.

3.3.2.3 Network Efficiency

This is required to ensure that the system developed will efficiently utilize the multi-purpose ICT network facilities deployed at Dwesa. We discovered that there is a need to limit network traffic in order to minimize the costs of sustaining the system. The traffic of large files being uploaded and downloaded over the network can increase data overheads during transmission. It is a requirement to control the data flow on the network so as to ensure that it stays uncongested during the busiest times of the day. Various technologies will be brought into the development of the e-government system to strictly maintain optimal usage of the network we have in Dwesa. The files will be stored outside the Web server and the files' metadata on the

MySQL database server, hence there is the need to efficiently implement policies on both the database and web servers.

On the database server side, the policies were enforced through the use of MySQL database objects that include stored procedures and triggers. These are explained in the sections that follow. On the web server, we realized the need to use a proxy server to transmit user-filled application files to the Home Affairs servers using File Transfer Protocol (FTP).

While the use of javascript makes the system responsive, it creates an increased bandwidth usage (Aptivate Org 2008). The system will minimize the use of javascript to implement critical processes that require instant communication with the server. For example, when a user is sending a report, he should provide a valid security code, otherwise the javascript will notify the user automatically that the system requires a valid security code before the report is sent to the municipality. The validation will not wait for the user to send the report but the client browser will communicate with the web and database servers, validating the code number behind the scenes. The process of minimizing javascript usage to the most critical parts of the system ensures that bandwidth costs are kept very low (Aptivate Org 2008).

3.3.3 E-government System Modularity

This is a vital requirement as the e-government system will have to allow extension when necessary. The system will be developed in phases and the need to consider cross network vertical system integration led us to believe that the e-government system should be modelled around an MVC system architecture framework.

3.4 Dwesa E-government System Functionality Flow Control

The functionality flow control will be achieved by implementing zones. A zone is a group of web pages with similar function or subject. For example, the system will have a set of web pages for administration, user or third parties agents. Using a Zoop framework, the controller portion allows the logic of any web page to be within a method called *pageXXXX* in the *zone_administration* class, which is generally defined in a file called *zone_administration.php*.

Therefore, we call the *zone_administration.php* an object called *zone_administration*. Inside this object exist many functions, one for each separate page. These functions are called page functions, specifically designed for the administration functionality.

3.4.1 Data Access Objects (DAOs)

The DAOs provide an abstraction layer through which one can access the database. These are PHP classes and functions created to manipulate the e-government database. The access objects access the database on behalf of the system. The data access objects use stored procedures, views, triggers and simple MySQL statements to communicate with the database. The DAOs promote efficient data exchange between the system's model classes and the database (Digre 1998).

3.4.2 Model Classes

These are PHP classes written to handle the system's components modularity. The four model classes in the Dwesa e-government system interface the action classes. Each model class has a corresponding data access class (DAO) that communicates with the database. Therefore each action class will have a model class (Waterson 2003).

3.4.3 Action Classes

These are object-oriented classes written in PHP to carry out tasks for the e-government system. The tasks will interact with the database through the use of model classes. Each action class has a model class. Each zone has an action class that handles the processes specific to that zone.

3.4.4 Zone Templates

These are HTML pages that deal with the system's presentation. Zone templates are grouped according to the zone they exist in. For example, all templates pages pertaining to the administration zone are grouped together. This ensures security and eases the design of system user accessibility. The advantages of creating zones in this system development are that user access to different zones of the system will be strictly controlled. The design will also allow system modularity and easy modification. Extra functionalities can be added easily to the available system. System management is also made easy by zones. For instance, the administrator can easily take a specific zone offline without major changes to other zones (Zoop Inc 2008).

3.5 E-Government Software Development Technologies

The development of an e-government system requires careful consideration of the technologies that will be used. Because this system is part of an existing ongoing project implemented in

open-source software, the e-government system will be developed using open-source proprietary software. We have chosen various software technologies to make the system reliable and secure to the users, particularly to Dwesa villagers.

3.5.1 Dwesa E-Government System OS Platform

The current multipurpose ICT platform deployed as a part of on-going project uses Ubuntu Linux operating system. Our aim is to develop a system that can run on our current Ubuntu Linux platform. Hence, the e-government system will be developed and will be open-source software which will be open to all improvements and future versions.

The platform chose Ubuntu Linux to make the project cost-effective and self-reliant. The Ubuntu Linux operating system is very secure when compared to other operating systems. Thus, it creates a very secure environment for running the web servers with our e-government system. There are very few viruses for the Linux operating system and thus it requires minimal virus protection software compared to what one would require on a Windows platform.

3.5.2 Dwesa E-Government System Web Server

Various web servers exist today. We chose to develop a web system that uses open-source software (Hu, Nanda & Yang 1999). The Apache web server is an open-source software that is used extensively in the world of information systems (Abrahams & Newton-Reid 2008). The Netcraft survey in November 2008 covered over 3 million web sites, and shows that Apache accounted for 57% of these. (Netcraft Org 2008). For the development of Dwesa e-government, we chose the Apache Web Server 2.0 because it provides a free license for us to access the source, the right to use and modify it as well as, most importantly, the right to distribute it.

The open-source web server Apache provides a high level of security, higher than that of proprietary web servers such as Microsoft Internet Information Services (IIS). The Netcraft survey justified the statement on the basis of the good history of Apache with regard to the number and nature of security bugs found. As of September 2009, the Apache web server had served over 54.4% internet sites and 66% of the million busiest sites (Wikibooks 2009). These figures show the strength and versatility of the Apache web server.

One of the greatest advantages of the Apache web server is its modular architecture. In the Apache Web Server, you can add or remove functionality as dictated by the system's requirements. Apache is also extensible, thus if there is any feature that you want but does not exist in Apache, you can write your own server module to implement it because Apache server and API source code are open to the public (Kamthan 1999).

Apart from this, the Apache web server is efficient because its Apache's C code has been optimized for performance. As a result, the web server consumes less system resources than many other servers. The administration of the Apache web server is very convenient because one can control the server remotely from a command line. This is possible because the Apache configuration files are in ASCII, have a simple format, and can be edited using any text editor, thus, they are transferable, so one can effectively clone a server (Kamthan 1999).

3.5.3 Dwesa E-Government System Database Server

The Dwesa e-government system will be a database-driven web application. In the development of this e-government system, we will make use of a MySQL 5.1 database server. The MySQL database server is characterized as a fast, robust database server with a good set of features. The server is best suited for use on systems that are designed to be portable and reliable. The server has an active development team for user support and capabilities are constantly being added that will always be available as open-source software. The current release is MySQL 5.1 and that is the version the Dwesa e-government system will be using to store user and system data. The e-government system will use a set of features that comes with the MySQL Database (MySQL 2008). These are explained below.

I. Type of Database Tables: It is vital to describe and explain the storage engine that will be used to create the Dwesa e-government system database. This will help to highlight the reason why some processes were designed in the way that they were. The MySQL server supports two types of database storage engines: the InnoDB and MyISAM. Whenever a table is created, the database engine is chosen from one of these two storage engine types. The MyISAM storage engine is the default for the MySQL server.

We would want our e-government system database to have better performance, rather than other criteria like data integrity. We thus chose to use the MyISAM storage engine to create the database tables for the e-government system. As stated by Buckler (2009) in his article,

benchmarks have shown that myISAM tables beat every other general-purpose database in terms of speed, and use fewer resources. The chief reason for using the myISAM tables is that they support a full-text search functionality. It is imperative to provide a functionality that will help e-government system users to perform a search on the content stored in the database. The most efficient way, other than using server-side search algorithms, is to embrace the full-text search feature provided by the myISAM tables (Buckler 2009).

However, as pointed out by Buckler (2009), myISAM tables do not enforce data integrity. Unlike InnoDB tables, which support referential integrity enforcement, Buckler (2009) stated that server-side algorithms can be scripted to achieve proper data integrity in myISAM tables. By using myISAM tables, we will not be able to run transactions and perform cascading deletes. We chose myISAM to create a balance between the resources that we have and the ones needed by myISAM tables. Furthermore, should the need arise to implement InnoDB tables, Buckler (2009) suggested that a database can contain both types of tables.

II. Triggers: These are named database objects associated with a table, that are activated when a particular event occurs for the table. This database object is vital to the e-government system because they maintain the integrity of one table that might rely on another table. For instance, if the *Users* table references the *Applications* table, and a field value from the *Applications* table gets deleted, a trigger can be set up to handle the data in the *Users* table. Triggers will also enforce referential integrity between two or more database tables. Trigger objects are crucial to cleaning the database tables and maintain data consistency in the database. An example of a Trigger is given below:

```
CREATE TRIGGER registeredDate
    AFTER INSERT ON Users
    FOR EACH ROW BEGIN
        INSERT INTO History (latestHistoryDate) SELECT DateSubmitted
        FROM Users;
    END;
```

The trigger named *registeredDate* inserts the date of registration of the user into the *History* table. This is only after the user has submitted his details successfully to the *Users* table. Triggers have restrictions that state that a table can have only one trigger per event. The trigger

can occur either before or after a query. If a trigger that handles INSERTs is created, it can be set up to run before the database has had a record added, or after. The e-government system developed requires critical database management. The database will hold files that will increase its size considerably and triggers will help to automate database clean-up tasks to avoid over-filling it with unneeded data.

(III) Stored Procedures: These are also database objects that are implemented on databases. A stored procedure is a set of SQL commands that has been compiled and stored on a database server. Stored procedures are proprietary programs written in procedural languages. A stored procedure is a program that runs directly on the database server. MySQL 5.0 has added the capability to create stored procedures. The e-government system uses them to improve performance and help with the ease of development (Gulutzan 2005). An example of a stored procedure is given below:

```
CREATE PROCEDURE validateApplications
    DECLARE x AS INTEGER;
DO
BEGIN
    SELECT * FROM users WHERE status=2;
    SELECT requirements INTO x;
    IF x > 4 THEN
        SET status = 3; -- all requirements are met
    ELSE
        SET status = 0; -- not all requirements are met
    END IF;
    UPDATE users SET Status=3 WHERE app_ID = ID;
```

The stored procedure given above validates applications that are ready to be sent to the remote government file server. Normally, this process can be achieved by writing server-side algorithms which will require two database connections to run the validation process. This increases network traffic between the database server and the web server. Because the whole stored procedure is executed in the database server and only necessary bits of information are sent to the web server, the process runs relatively faster. Thus, stored procedures are required in this development process to limit network traffic and CPU load.

The e-government system will allow users to upload and download files from the database. If the number of users increases, the operations may slow the performance of the database because more data will need to travel over the network. This will happen when the system has a script that gets called often or uses any looped queries, generating a lot more network traffic than the system should have. Stored procedures will cut down on long queries being sent over the network by turning a potentially long query into a short alias. Basically, only vital bits of information are pushed through the network (Gulutzan 2005).

IV. Views: Database views are stored SQL statements to perform a specific task on a database. They are usually referred to as virtual tables or just queries. They use the Data Dictionary that the RDBMS defines and they are stored in the database. Views are only instantiated when they are called. The RDBMS then evaluates the SQL statement that used the view's name in its *from* clause. The view's name is used just as one would use other regular table names. The MySQL database server allows the creation of powerful and versatile database views that can ease integration and backwards compatibility. Views simplify complex SQL queries and ensure location transparency (Slazinski, 2001).

The Dwesa e-government system will use database views to implement its security policies. Together with stored procedures, they will ensure database abstraction, restricting the data that a user is entitled to see. A view can be created from several types of SELECT statements. In the following example, a villager is allowed to view his own submitted and saved applications, and should not be able to see anyone else's.

```
CREATE VIEW All_Applications AS
Select * from Applications
Where Username = User;
```

Database Views will be used in this e-government system to insulate the system from possible future database schema changes. This will also be done inline with the Rapid Application Development (RAD) method used to develop the e-government system (Slazinski 2001).

V. Database Events. These are database objects that run queries or stored procedures that aggregate or gather information on a regular basis. Examples of such tasks are statistics

gathering, data polling and data normalization (Wijayawardhana 2008). The Dwesa e-government system will have several processes that require a certain event to happen so that they are invoked. The events occur when a user interacts with the Dwesa e-government interfaces. There will be other processes that require to be invoked at a predefined, scheduled time. The processes that will maintain our database efficiency and its integrity will be scheduled using the Event scheduler in MySQL. The events will run processes during the night when the data transaction stress on the database server is less. The scheduled tasks called events can also be set to run at certain intervals of time rather than as a onetime task.

For example, when a user clicks a button to deregister, his details will be deleted from the users table. However, since we created a trigger that will also remove all records from other tables linked to this users table, once a record is deleted on the users table the process is likely to put stress on the server if it is run during the day. The system's performance will be heavily compromised if more than one user considers deregistering at the same time. Because these types of processes can complete the task on the database server, the Event Scheduler of MySQL is the key to schedule the process to run on off-peak hours of the network traffic. The deregistering will be done on the group of users who have deregistered during the day. The following code shows the event to deregister the users:

```
CREATE EVENT deregister
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_TIMESTAMP
DO DELETE FROM users WHERE DELETED = 1;
```

The above event, once created, will start to delete all deregistered users at any interval of 1 day, and this process will recur for ever.

3.5.4 Dwesa E-Government System Model Framework

The Zoop framework is an object oriented PHP web application framework based on a front controller (Jakachira *et al.* 2008). The Zoop Model-View-Controller ensures separations of content on the development of the e-government system. The business logic of the system is separated from the presentation content as well as the data layer. Thus, the framework is designed to be extensible, modular and flexible.

The framework, when used to develop systems, is capable of producing very secure web applications (PHPframeworks Inc 2009). It is also suitable for the RAD method as the coder can make secure web systems quickly, making it easier to do rapid prototyping. Zoop, an open-source software, is considered one of the most full-featured frameworks available (Zoop Inc 2009) and these features include the following:

- PHP4 and PHP5 are supported. The developer is able to mix the two PHP versions without creating compatibility problems.
- MVC is a Model-View-Controller setup that comes with the Zoop framework. The setup separates PHP code from HTML and CSS presentation code. It also creates data layer abstraction to allow the data layer to be flexible to modifications.
- A Multiple Database handling capability is included in the Zoop framework. This allows the use of more than one database without having to change anything.
- An Inbuilt Template Engine ensures efficient system usage. The engine will quicken the response of the Apache web server, when a user calls a web page. The inbuilt template engine also makes for portable systems and eases the maintenance of complex systems.
- A Caching functionality is an added feature in Zoop framework. The e-government system strives to minimize the usage of bandwidth, however, if an increased number of users download and upload application forms from the portal at the same time. The caching functionality allows the creation of web applications that conform to low bandwidth usage. The system being developed in this research will implement caching of large application forms.
- The Inbuilt Validation or filtering component is the most vital feature that comes with the Zoop framework. The use of this component guarantees data consistency in the system database. It also provides a very strongly secured system. When the filtering component is implemented, it avoids SQL injections in the database by malicious users (Palleter 2008). SQL injections are Data Manipulation Language (DML) queries submitted to the database through web forms with the aim of extracting vital information stored on the system's database. The Zoop inbuilt Validation component filters special characters from data posted from the forms before submitting to the database.

The system will be developed using PHP5 programming language, and the Zoop framework is compatible with PHP5 language.

3.5.5 E-Government System Development Environment

A software development environment is a programming environment that provides a group of software add-ins and is packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder. Software development environments are crucial tools when RAD methodology is used because of the need to rapidly produce working system prototypes (Boehm *et al.* 1984). The Netbeans IDE Early Access for PHP is an integrated development environment (IDE) for PHP web and mobile applications that has been made available by Sun Microsystems and the Netbeans community. The Netbeans IDE is an open-source software that brings all software development modules to one place. The business logic of the e-government system will be done in Netbeans IDE (Netbeans Inc 2008).

The system development will also use JavaScript technology, and NetBeans 6.1 IDE provides a tighter integration with the MySQL database. The IDE delivers code completion and faster start-up benefits to the development of the system. It also includes support for embedded CSS and HTML. Furthermore, the system development environment provides debugging support using XDebug plug-in (Netbeans Inc 2008).

3.5.6 JQUERY JavaScript Library

JQUERY is a lightweight JavaScript library that is easy to learn as it uses familiar CSS syntax to design responsive web applications (Eisenberg & Volder 2004). Unlike PHP, JavaScript is a client-side scripting language. The development process will make use of a JQUERY library because it is a lightweight JavaScript library and can be easily extended.

An example of how this library will be used is shown below. The java script sends name and surname to the server-side script in *send.php* asynchronously.

```
var txtname    = $('#txtnm').val();           //txtnm = id of the text field
var txtsurname = $('#txtsurnm').val();        //txtsurnm = id of the text field
$.ajax({
    url: 'send.php',
    dataType: 'json',
    data: {name:txtname; surname: txtsurname},
```

```

        success: function(data, StatusText){
            $('div').text(data.result);
        }
    });

```

When the send.php script receives the data posted, it creates a response and sends it back as JSON data type. The data is displayed in a DIV html element.

```

<?php
    $firstname = $_POST['name'];
    $lastname = $_POST['surname'];
    $result['result'] = "{ $firstname } { $lastname } is a student";
    echo json_encode($result);
?>

```

The use of JavaScript in this research is to write embedded functions to run locally in the user's web browser and increase the system web page's response time to user actions, making the web application feel more responsive. (Dyakalashe *et al.* 2008). The use of JavaScript will be restricted to critical processes of the e-government system only.

3.5.7 Cascade Styling Sheets (CSS)

The e-government system will be designed to standardize its look for the Dwesa community. More work will be needed to localize the system at a later stage. We decided to start at a preliminary level to design the look and feel of the e-government system interfaces. This will be done at the presentation level using the Cascade Styling Sheets (CSS) and HTML code. CSS and HTML are both referred to as mark-up languages (Glass *et al.* 2004).

3.5.8 PDF File Filler

The e-government system to be developed will let the Dwesa villagers download important forms and fill them in online and then upload them to the appropriate South African government department. Most of the important files are stored in read-only PDF format, thus they are read-only when they are downloaded by the users. Few forms are writable or can be easily filled in using text editor software.

We sought for a method to automate the conversion of read-only PDF files to files that can be filled in on the computer, leaving the structure and contents of the files intact. The PDF Filler allows users to complete the PDF files the way they do on text files or web forms. Obviously,

our choice will be left on whether we will use third-party PDF Fillers or rapidly develop our stand-alone PDF Filler (PDFOnline Inc 2008).

The e-government system will utilize the stand-alone PDF Filler. The disadvantage, however, of this integration is that the PDF filler is not free. We realized that incorporating this filler system into the system will lessen the time needed to develop such a stand-alone system specifically for the Dwesa e-government system. We also noted that it is important to have the necessary support that this stand-alone system will provide to our e-government system.

3.5.9 Digital Signatures

Like handwritten signatures, computer digital signatures are a specialized way to enhance security issues such as tampering with and impersonation of data (Balacheff *et al.* 2001). Non-repudiation is critical for services online as it contributes to the reciprocal trust in the system of the parties involved (Bruschi *et al.* 2003). The e-government system will allow users to upload their applications to the government department file servers. The transmission of such applications to the remote government server through the World Wide Web (WWW) means the files may be prone to illegal tampering and impersonation. We use the Cryptograph PHP library to digitally sign user application files.

Detecting document tampering and other authentication methods rely on a mathematical function called a one-way hash (Rivest *et al.* 1978). The crucial part of signing any data is the use of a private key and a public key (Rivest *et al.* 1978). Anyone who knows the corresponding public key may verify the integrity of the signer (Bruschi *et al.* 2003). Because the private key is only accessible to its owner, a digital signature is deemed non-repudiate. The signing software creates a one-way hash of the data, and then uses the private key to encrypt the hash. It is this encrypted hash and other related information that is referred to as digital signature (Bruschi *et al.* 2003).

Confirming the identity of the signer requires that the public key belongs to the applicant. This is achieved by using username-and-password authentication in our e-government system.

3.5.10 Process Scheduling

The time that will be spent by Dwesa e-government users to accomplish forms data movements will be minimized and strictly controlled. Also because the data movement is not a productive

activity, we will require a dedicated tool to schedule such types of activities (Kettimuthu *et al.* 2007). The Job scheduler is an open-source light-weight plug-in feature that will be integrated in the Dwesa e-government system. The plug-in will allow us to schedule the processes that should be run at a convenient time as well as at appropriate intervals. To move files between the Dwesa e-Government Web Server and the South African file servers, the files will be transmitted as a batch.

(a) Updating Government Forms on DOAC: This process will always run at midnight to get updates of government forms.

(b) Sending Applications on DOAC: The applications will be batched and transmitted during off-peak hours to minimize overloading the network with data traffic during daytime.

1. **(c) Tracking Applications Information:** A process will run every 6 hours to gather small details of applications already residing on the government server. The details will be stored in the database so that any users tracking their applications will be furnished with up-to-date information about the processing of their applications.

3.5.11 Forms Transfer Using FTP

File Transfer Protocol (FTP) is a communications protocol that controls the transfer of files from one computer to another over a network. FTP can be used to upload and download files between a client machine and a file server. To make use of the protocol between the Dwesa e-government server and the respective government department file server, we connect to an FTP server on the remote government department server and send it requests (Valade & Ballad 2008:204). The protocol is utilized to transfer files from one server to another server, connected through a WAN network, which in this study will be a government file server. The e-government system will be able to handle efficiently the possibly large number of application forms through the use of a PHP in-built FTP library (Valade & Ballad 2008:204). For this e-government pilot system, a simulation government file server will be created using FileZilla to mirror the Home Affairs server. The computer machine will be able to receive completed application forms from the users through the proxy.

3.6 Conclusion

This chapter stated the requirements and needs that are vital to the development of a successful e-government system. It explained various technologies that will be used to develop the system. It explained how the requirements were found and mentioned how these requirements

will be met. To summarize all the steps necessary for the development of the e-government system, the following steps have been discussed in this chapter:

- Gathering citizen requirements using workshops or focus groups during Dwesa trips. The needed e-government functionalities were determined through consultations with Dwesa community villagers.
- Prototyping and early iterative user testing of e-government designs.
- The re-use of available e-government software components.
- A favourable schedule that defers design improvements of the user interfaces to the next product version.

Chapter Four

Dwesa E-Government System Administration and User Management

The chapter describes and explains the management of the e-government system. It also discusses the distribution of the Dwesa e-government system. The design of the Dwesa e-government database is discussed in this chapter.

4 Introduction

This chapter will discuss the design and implementation of the system database and the Dwesa e-government back-end interfaces. The back-end portal provides system administration and user management capabilities. The chapter will explain the design of the back-end system architecture. It will explain the system database design and how the three major functionalities are managed.

Dwesa e-government system back-end interfaces are administrative interfaces that run processes against the input collected from the front-end interfaces. Back-end interfaces are used by administrators and the third parties. Third parties agents are government agencies which process some of the input from the front-end interfaces.

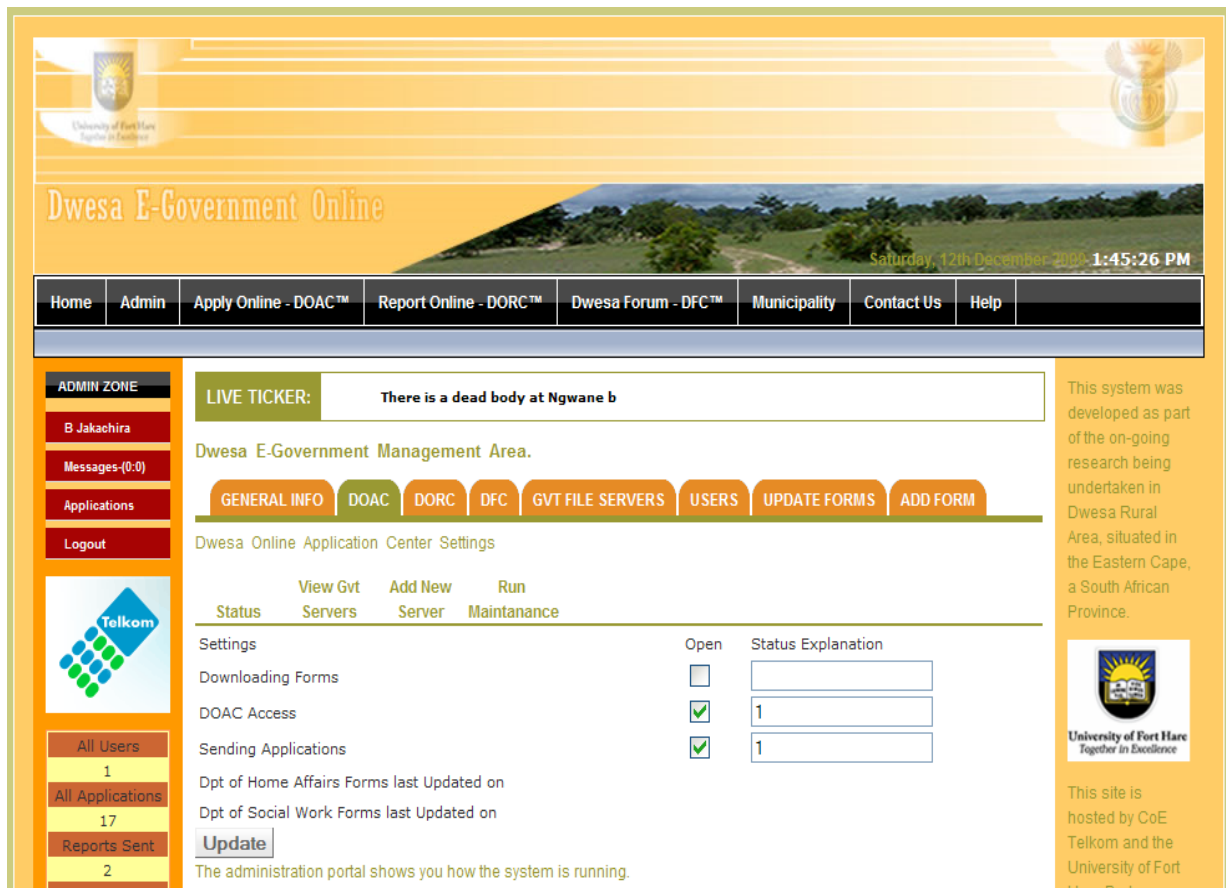


Figure 4.1 Dwesa E-Government System Back-end Interface.

Front-end interfaces are interfaces that stand between the Dwesa community members and the back-end interfaces. These interfaces collect the input from the users.

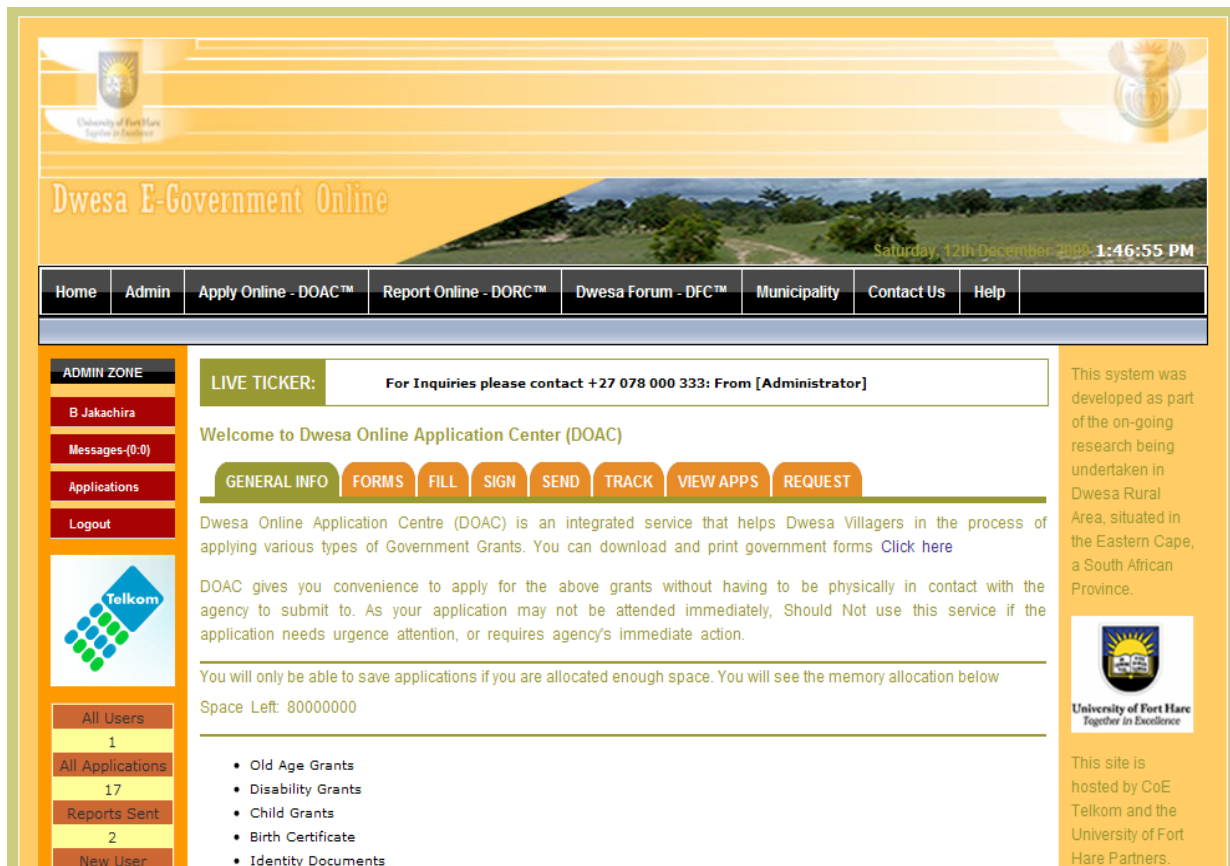


Figure 4.2 The Front-End Interface.

4.1 The Distribution of the Dwesa E-Government System

This section gives an explanation of the distribution of the e-government system. Figure 4.3 describes the system distribution. It shows that the Dwesa Online Reporting Centre (DORC) and Dwesa Forum Centre (DFC) run on the local Ubuntu Linux server.

The Dwesa e-government system uses a central server at Ngwane Primary School. This is done to ensure that the management of the system is handled at single point. The distribution of the Dwesa e-government system promotes information sharing among the Dwesa community members. This also minimizes the usage of the resources, as the data is only stored at one point, avoiding data redundancy.

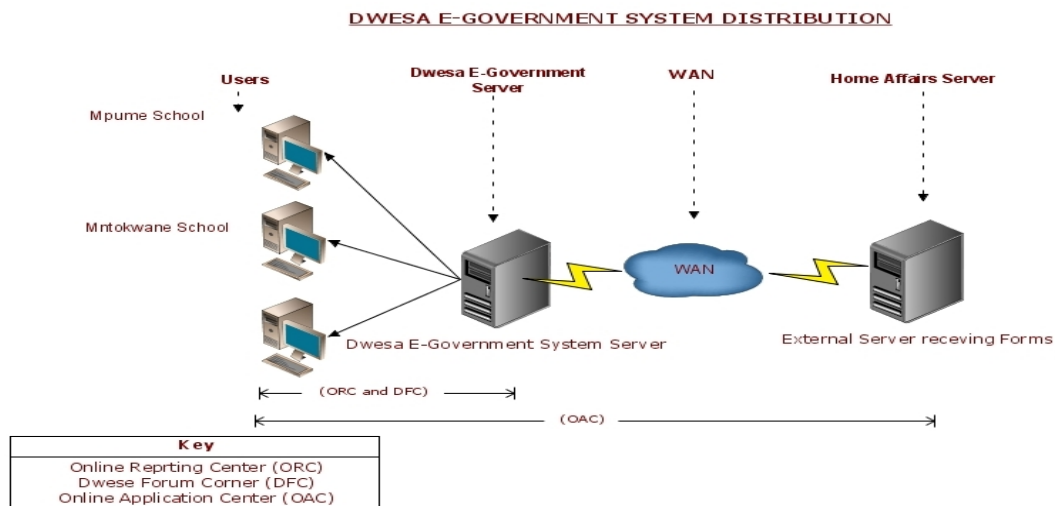


Figure 4.3 Dwesa E-Government System Distribution.

Figure 4.3 depicts the existence of a Wide Area Network (WAN) through which the Dwesa Online Application Centre (DOAC) is able to communicate with a remote government file server. The users, Dwesa villagers and students from the four schools, access the e-government system from the server at Mpume Junior School through a proxy. The schools have computers that are running the Ubuntu operating system. These are deployed on a network accessing the outside world through a VSAT link.

4.2 The architecture of Dwesa E-Government System

The e-government system was developed using LAMP (Linux, Apache, MySQL and PHP) system components. This means that the e-government system was developed using the following technologies

- Ubuntu Linux Operating system.
- Apache web server 2.0.
- MySQL 5.0.1 Database.
- PHP 5 scripting language.
- FileZilla Version 0.9.31.

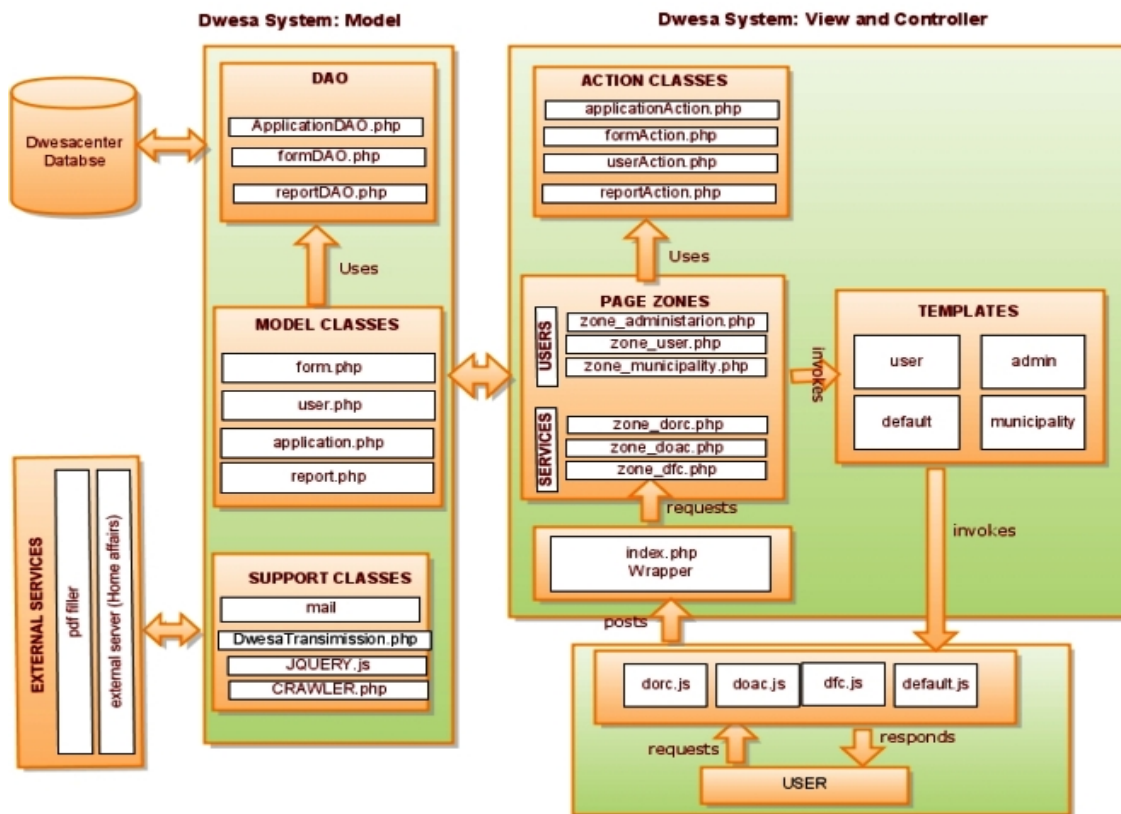


Figure 4.1. Dwesa E-Government Components and Structure.

In the sections that follow, we explain the data flow among the e-government system components. The system architecture makes use of the software technologies highlighted in chapter three.

4.2.1 System Page Zones

Page zones are system classes that pass dynamic data to the *html* templates contained in zones. The remote browser sends a request to the wrapper, which is the default page of the web application. The wrapper then requests a page from the page zones container. A specific zone is then called, which uses other components described below to generate the page requested by the user. Table 4.2.1 shows the three zones that the Dwesa E-government system has.

System user	Description	File Name
Administration	For the administration	Zone_administration.php
User	User main portal	Zone_user.php
Municipality	For the government agencies	Zone_municipality.php

Table 4.1: System Users Zones.

System Functionality	Description	File Name
DOAC	The Dwesa Online Application Centre	Zone_doac.php
DORC	The Dwesa Online Reporting Centre	Zone_dorc.php
DFC	Dwesa Forum Corner	Zone_dfc.php

Table 4.2: System Functionalities Zones.

Each Zone has a number of templates. Each template may invoke its own *page* and *post* functions to generate a web page. A page function displays the web page. A post function handles the data posted from the forms. Therefore a typical zone in this system comprises of sets of page and post function combinations.

4.2.2 Action Classes in the Dwesa E-Government System

There are three action classes in the design of the Dwesa e-government system. Action classes are used by zones. Each zone can use one or more of these action classes. The classes form a part of the view and controller components of the Dwesa system. Table 4.2.3 shows the action classes designed that are utilized by the system zones.

Action Class	Description of the Action Class
UserAction.php	It carries all actions on a Dwesa system user. The class handles user management processes. For example logging in and out, deregistering, activating, blocking, communication, user details management, authenticating, etc.
applicationAction.php	All actions on an application form are handled by this class. For example, filing forms, uploading completed forms, tracking applications, receiving feedback, etc. The class is used extensively in the Dwesa Online Application Centre (DOAC).
reportAction.php	The class is designed to do actions on a report. For example, announcing, sending, responding, deleting, etc. It is widely used in the Dwesa Online Reporting Centre (DORC).
formAction.php	It's not used widely in the system design. Its aim is to do some action on a form. For example, downloading, updating, deleting, etc.

Table 4.3 System Action.

4.2.3 System Zone Templates

Zone templates are pages written in Hyper Text Markup Language (HTML) language. The templates have a *.tpl* file extension. They render the content to the users as web pages. Each zone has its own set of templates. These templates are only accessible to an authorized user at zone level. The templates utilize Zoop GUI functionalities. The data and some bits of information are communicated between the zone pages and zone templates using Zoop GUI components.

4.2.4 System Model Classes

The system has model classes designed specifically to model the system entities. The classes return characteristics about the entities. Each model class belongs to an entity. In this context, an entity is a Dwesa e-government system object. Such objects include form, user, report and application. These objects have characteristics such as name, size, type, etc. Each object has a corresponding data access (DAO) class that accesses data stored in the database for that object. For example the *application* class has a data access class called *applicationDAO*. These characteristics are defined in the model classes. Table 4.4 states and describes the four entities.

Dwesa System Entity	Model Class	Description
User	User()	Provides a model for the e-government user
Form	Form()	Handles all form characteristics and metadata
Application	Application()	Each application is modeled through this class.
Report	Report()	Each report sent by the user is modeled by Report class

Table 4.4: Model Classes.

4.2.5 Data Access Objects

Dwesa Data Access Objects (DDAOs) are specialized system classes in the design of the system to manipulate the MySQL database on behalf of the action classes. The objects query that database when an action from an Action class attempts to utilize the database. The DDAO are part of the Dwesa e-government system model. The objects make use of database objects such as stored procedures, views, triggers and simple SQL statements. The use of Data Access

Objects provides database abstraction, which is necessary when the database is constantly being accessed and modified.

4.2.6 System Support Classes

Support classes are designed to support the Dwesa e-government system in several ways. The system receives data and information from horizontally integrated systems. For example, the application forms catalogue is kept updated by the information obtained from the government forms repositories. The information fetched is handled by the support class called *Crawler*. Another support class is *DwesaTransmission* that defines a FTP connection to a government department file server.

4.3 Dwesa E-Government System Database Design

The E-government system is driven by a back-end MySQL database system that holds the content of the e-government system. The database also holds all vital user information. The database has 25 database tables, four triggers, three stored procedures, two events and one view. These are all explained below.

4.3.1 Database Tables

The e-government system is supported by 25 linked MySQL database tables. The important database tables are related to each other as depicted by the entity relational diagram (ERD) in Figure 4.4.

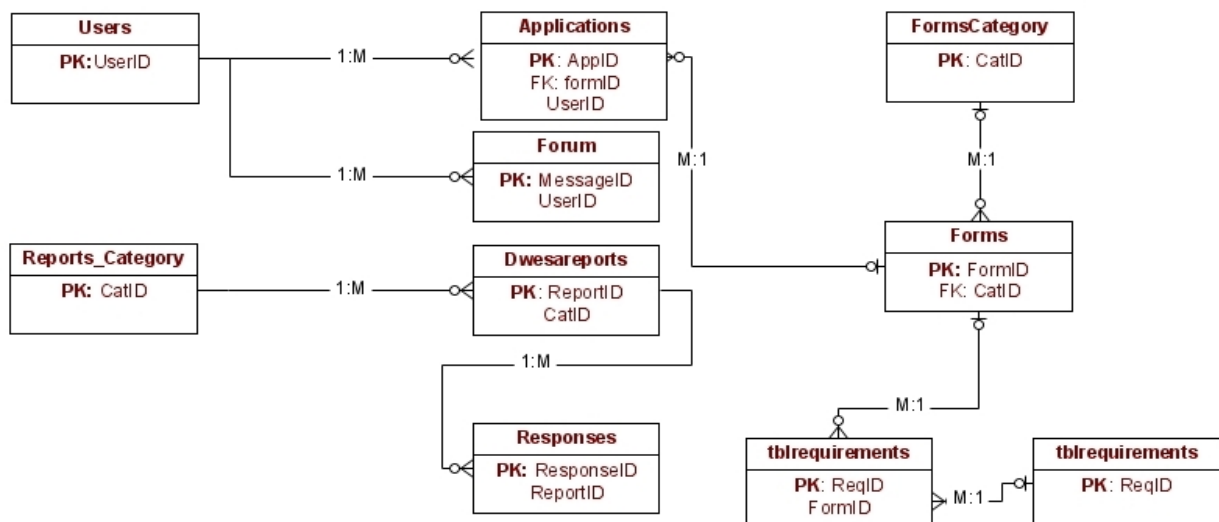


Figure 4.4 Dwesa E-Government Database ERD.

The ERD diagram in Figure 3.1 shows that a user can have zero or more document files saved in the database. However, each of these document files for this user can only point to one form. Thus a user cannot have more than one document file of the same form. This enforces the system policy that a user can not apply twice for the same important document. These tables are explained below.

(a) Users Table: This table contains user details. All visitors to the e-government portal must complete the registration process before they make use of the services available. The table has 12 fields and all are explained in the table 4.3.2.

```
mysql> describe users;
```

Field	Type	Null	Key	Default	Extra
Uid	int(4) unsigned zerofill	NO	PRI	NULL	auto_increment
Username	varchar(45)	NO	UNI	NULL	
Password	varchar(45)	YES		NULL	
Name	varchar(45)	YES		NULL	
Surname	varchar(45)	YES		NULL	
EmailAddress	varchar(45)	NO		NULL	
PostalAddress	varchar(45)	YES		NULL	
PhysicalAddress	varchar(45)	YES		NULL	
Gender	varchar(45)	YES		NULL	
DateOfBirth	varchar(45)	YES		NULL	
Country	varchar(45)	YES		NULL	
Telephone	varchar(45)	YES		NULL	
Town	varchar(45)	YES		NULL	
DateRegistered	varchar(45)	YES		NULL	
DateLogged	varchar(45)	YES		NULL	
RegStatus	varchar(45)	YES		NULL	
Conditions	varchar(45)	YES		NULL	
NewsLetter	varchar(45)	YES		NULL	
BlockingStatus	varchar(45)	YES		NULL	
Cell	varchar(45)	YES		NULL	
IDnumber	varchar(45)	YES		NULL	
LastLogged	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
RegType	smallint(50)	YES		1	
MessageCode	varchar(50)	NO		XXXXXX	
System_Access	int(11)	YES		0	
Signature	varchar(200)	YES		NULL	
DELETED	tinyint(4)	YES		0	

Figure 4.5 Users Database Tables.

The most important fields, apart from user identification fields, are *RegType*, *BlockingStatus*, and *LastLogged*. These fields make the user administration processes run efficiently.

The *BlockingStatus* field shows if the User account has been activated or not. Every user who registers on our system should have his account activated by the administrator before he could log into the system. This is done to enhance the security system and ease the management of users.

The *RegType* field is needed to identify users who have fully or partially registered on the system. The e-government system allows users to quickly register should the user want to report an urgent matter. The user should provide a username, email address and a password. An

email message is then send to the user to activate his account. The user is furnished with a Message Security Code (MSG) which he should use to send reports on the ORC system.

The *LastLogged* field shows the last date the user logged in.

As a method to enhance the system usability and friendliness, users are provided with a quick and easy way to register on the system. There are two possible ways, a long method and a shorter method, for a user to log into the system. A user who has registered using the shorter way is regarded as a partial user whilst a user who has registered using a longer way is treated as a fully registered member.

(b) Forms Table: The table holds information about the important government application forms available in the system. All attributes of each form are kept in this table.

```
mysql> describe forms;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
Size	varchar(45)	NO		NULL	
Content	mediumblob	NO		NULL	
DateFiled	varchar(45)	NO		NULL	
Version	varchar(45)	YES		NULL	
Description	varchar(250)	YES		NULL	
Downloads	tinyint(3) unsigned	YES		0	
cat_id	int(11)	YES		NULL	
Extra_Name	varchar(155)	YES		NULL	
Price	varchar(250)	YES		NULL	
Requirements	int(11)	YES		NULL	

Figure 4.6 Forms Database Table.

(c) Dwesa Reports Table: Reports sent by villagers are kept in this table. The table has a field named *Urgency*. The field shows if the report is urgent or not. This allows the system to decide the method to use when sending the report to the relevant authorities.

```
mysql> describe dwesareports;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
CatID	tinyint(3) unsigned	YES		NULL	
username	varchar(45)	NO		NULL	
email address	varchar(45)	NO		NULL	
Urgency	varchar(45)	YES		NULL	
Receipient	varchar(45)	YES		NULL	
Reply	varchar(45)	YES		NULL	
Message	varchar(100)	YES		NULL	
Type	varchar(100)	YES		NULL	
Title	varchar(100)	YES	MUL	NULL	
Announced	varchar(20)	NO		NULL	
TimeReported	varchar(200)	YES		NULL	
uid	int(11)	YES		NULL	
Screened	int(10) unsigned	YES		0	

Figure 4.7 Dwesa Reports Database Table.

(d) Forum Database table: This table holds all opinions and notices posted on the Dwesa Forum Corner. Each message has an Identity number. Each message can be a notice or an opinion or both. The table has a field that depicts whether the message is an illegal one or not. Illegal messages are flagged and are not shown on the Dwesa Forum Corner functionality.

```
mysql> describe forum;
```

Field	Type	Null	Key	Default	Extra
MessageID	mediumint(8) unsigned	NO	PRI	NULL	auto_increment
ResponseID	int(11)	YES		NULL	
Username	varchar(45)	NO		NULL	
Emailaddress	varchar(45)	NO		NULL	
Message	text	YES		NULL	
ForumName	varchar(100)	YES		NULL	
IllegalMessage	int(10) unsigned	YES		0	
DatePosted	varchar(45)	YES		NULL	
MessageType	varchar(70)	YES		NULL	
Title	varchar(45)	NO		NULL	
Response	varchar(50)	YES		NULL	
Receiver	varchar(200)	YES		NULL	
Uid	int(11)	YES		NULL	

Figure 4.8. Forum Database Table.

4.3.2 Database Views

The Dwesa e-government authentication system uses MySQL database views to query the database. The *login* view queries the database each time the user tries to log into the system and returns a virtual table with two fields, Username and Password, as depicted in Figure 4.9. The view screens all users from the *users* database table.

```
mysql> describe login;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Username   | varchar(45)   | NO   |     | NULL    |       |
| Password   | varchar(45)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Figure 4.9 Login View.

The view does not output users that qualify in the following cases:

- account blocked by the administrator;
- account still to be activated;
- expired accounts;
- if the account is an administrator's one.

The administrator's account is not included in the view results for security reasons. Each time a user tries to login, the authentication system looks in the view results to see if the user with the username and password supplied exists.

Thus, the view was created to allow grouping of users and to hide some accounts from users' direct access. Only users that are registered, query the database during login, thus preventing unnecessary queries from running on the database server. A simple SQL statement, to check the existence in the login view of a user who tries to login, does not slow down the database server.

4.3.3 Stored Procedures in Dwesa E-Government Database

The e-government system has two stored procedures, created to allow effective handling of crucial database server processes.

(a) screenReports: This procedure is utilized in the DORC system. It uses the MySQL full-text searching feature to screen the content of reports before they are sent. The procedure is called by our database event scheduled to run every 10 minutes. The procedure deletes all reports with inappropriate content after every hour. This process is part of our security mechanism implemented on the DORC portal. Users who send illegal reports are also suspended. Figure 4.10 shows how this was implemented.

```

1 DELIMITER $$
2 DROP PROCEDURE IF EXISTS `screenReports` $$
3 CREATE DEFINER=`root`@`localhost` PROCEDURE `screenReports`()
4 BEGIN
5     DECLARE rep_id INTEGER;
6     DECLARE done INTEGER DEFAULT 0;
7     DECLARE id INTEGER;
8     DECLARE tit INTEGER;
9     DECLARE message text;
10    DECLARE Uuid INTEGER;
11    DECLARE score INTEGER;
12    DECLARE matchd TEXT;
13    --illegal words hidden to publish this procedure
14    UPDATE dwesareports SET Screened = 2 where MATCH(Title,Message)
15        AGAINST(illegal words IN BOOLEAN MODE)
16        and Screened = 0;
17    UPDATE dwesareports SET Screened = 1 where Screened = 0;
18
19
20    select Uid, Screened from dwesareports;
21 END $$
22 DELIMITER ;

```

Figure 4.10 Screening Report Stored Procedure.

(b) **validateUser**: this procedure takes username and password as parameters and outputs the username and a code number between 1 and 5. It validates the user before login. The code number describes the result of validation.

(c) **changepassword**: This procedure was designed to allow the user to modify his password.

4.3.4 Triggers in Dwesa E-Government Database

The e-government system makes use of designed MySQL triggers. This is to ensure data integrity. We have designed three database triggers that perform on different user tasks. When a user makes an application on the DOAC portal, the first action that occurs in the database may be that an application is inserted into the applications table. By a way of triggers, this action initiates a chain reaction of events in other four tables associated within the database.

(a) **Applications_bi_trg** Trigger: The trigger, listed in Appendix C, performs updating tasks across four database tables. When a user saves or sends the application, the details of that application are inserted into the applications database table. The insertion event invokes our trigger to run first. The trigger updates user's disk space size, number of applications saved or sent in the performance table.

It also updates the number of applications sent to this particular government server to the *departments* database table. We also store the date the application is sent so that we track the last activity on the government file server.

(b) **Forms_af_trg** Trigger: This database trigger performs tasks after a record is inserted in the forms database table. When a new form is inserted into the forms table, the trigger updates the number of forms on the government department that this particular form originates from. We also store the date on which the form has been stored. The date shows the time we receive an update from this government department.

(c) **Users_bi_del** Trigger: One of the very important database trigger that is invoked when a user chooses to de-register in the system, it intercepts all DELETEs MySQL commands on the users table and just flags the record as deleted.

```
1 DELIMITER ||
2 DROP TRIGGER IF EXISTS users_bi_del
3 ||
4 DELIMITER $$
5 Create Trigger users_bi_del
6     BEFORE DELETE ON users
7     FOR EACH ROW
8 BEGIN
9     DELETE from applications where Uid = New.Uid;
10    DELETE from dwesareports where Uid = New.Uid;
11    DELETE from forum where Uid = New.Uid;
12    DELETE from tblperformance where Uid = New.Uid;
13    DELETE from responses where Uid = New.Uid;
14 END$$
```

Figure 4.11. Users_bi_del Trigger.

The record waits for the *clean_up_evnt* database event object to delete it. This is important, because allowing a delete command on the users table that is linked to all tables in the database can heavily slow down the database server every time a user chooses to deregister his account.

4.3.5 Events in Dwesa E-Government Database

With the aim of maintaining our database integrity and to ensure data consistency, we designed two database events that run on an hourly and weekly basis, cleaning the system database and authenticating the reports before they are sent. The *clean_up_evnt* and *screening_evnt* events are scheduled to run on a weekly and hourly basis respectively.

These are also vital database objects, as they restrict database server resource-consuming processes to a specified time. The reason to implement database events was to lessen the web server activities. We realized that these processes can run and achieve their goals in the database server. By doing this we also ensure that the network traffic is kept at minimum levels. Database events are pre-compiled small programs that are scheduled to run in the database server at a particular time in the future.

The Dwesa e-government system has a database event scheduled for a clean-up process in the database once every week. The *clean_up_evnt* event, when deleting users who have deregistered on users table, invokes the *Users_bi_trg* trigger, which initiates a chain of events that deletes all information about this user in the database.

```

1  DELIMITER ||
2  DROP EVENT IF EXISTS clean_up_evnt;
3  ||
4  DELIMITER $$
5  CREATE EVENT clean_up_evnt
6  ON SCHEDULE
7      EVERY 1 WEEK
8      ON COMPLETION PRESERVE
9  DO
10 BEGIN
11     DELETE from users where Deleted = 1;
12 END $$

```

Figure 4.12. Clean Up Database Event.

Because myISAM tables do not support cascading deletes, we designed the event to remove all records with broken links in the database. For instance, when a user de-registers in the system, all his DOAC, DFC and DORC data continue to exist in the database until this event runs.

The *screening_evnt* event runs every ten minutes as depicted in Figure 4.13. The event calls the *screenReports* stored procedure that screens all submitted reports. The screening is done to ensure that each report's message is in appropriate language. When the report is submitted to the database, they wait for 10 minutes for screening before they are sent. A ten minute interval is given to allow more reports to accumulate before screening. The stored procedure uses the MySQL FULLTEXT search feature on the title and the body of the report. When inappropriate words are detected, the stored procedure puts the report in the deletion queue. Valid reports are tagged appropriate and can be displayed to the Government Receiving Interface. The invalid

reports wait for an hour before they are deleted. The delaying tactic was designed to minimize huge overheads on the database server.

```
2 DELIMITER ||
3 DROP EVENT IF EXISTS screening_evnt;
4 ||
5 DELIMITER $$
6 CREATE EVENT screening_evnt
7 ON SCHEDULE
8     EVERY 10 MINUTE
9     ON COMPLETION PRESERVE
10 DO
11     CALL screenReports();
12 END $$
```

Figure 4.13. Screening Reports Event.

4.4 Dwesa E-Government System User Management

Integrating three large system functionalities to form one e-government system has resulted in a complex system that attracts users with different ambitions. For efficient delivery of services to each user, we implemented a portal to manage all users who register in the system.

The management of the user accounts is the role of a user who has been granted administrative rights. The user should be logged in to the e-government system as an administrator to administer user accounts.

4.4.1 User Accounts Management.

As shown in Figure 4.14, an administrator can administer user accounts.

GENERAL INFO
DOAC
DORC
DFC
GVT FILE SERVERS
USERS
UPDATE FORMS
ADD FORM

User Accounts

Number of Registered users: 1

Search by

Lillyosa Jakachira

Username	Lilly	 Account Open Deactivate
Email Address	ljakachira@gmail.com	
Date Registered	24-Oct-2009; 10:22PM ;+0300 and last login was on 24-Oct-2009; 10:23PM To manage Lillyosa's Forum Messages Click here .	
Memory Available	DOAC: 76.29 MB DORC: 976.56 KB DFC: 976.56 KB	
DOAC Status	DEACTIVATE	
DORC Status	DEACTIVATE	
DFC Status	DEACTIVATE	
No Picture Available		
Send Password Deregister Send Message		

Figure 4.14 User accounts management.

Activating the user on the DOAC, we give the user permission to use the DOAC portal as an administrator. The user can then send reports from the system to:

- Deactivate or activate the user account.
- Send the user a new password.
- Deregister the user.
- Send private messages to the user.

```

226      //create a temporary password and update it
227      $a_user = new UserAction();
228      $pwd = $a_user->setMessageCode();
229      $user->setPassword(md5($pwd));
230      $task = $a_user->addUser($user);
231
232      $message->setMessage("Dear {$user->getName()}. \n\nYour Password
233                          has been changed to: {$pwd}. \n\nThank You.
234                          \n\nThe Dwesa E-Government Admin.");
235      $message->setType(3); //message from Admin
236      //transmission
237      $transmit = new DwesaTransmissionAction();
238      $result = $transmit->sendbyEmail("", $message);

```

Figure 4.15. Sending the user a new Password.

4.4.2 Users Details Management

Users manage their own details in the system. Users register in the system using our quick registration interface. Upon successful registration, the user is provided with a System Message Code (SMC), which is used in the DORC portal.

The user interface for managing user details is shown in Figure 4.16.

Welcome to Dwesa E-Government Portal

GENERAL INFO EDIT PROFILE CHANGE PASSWORD MESSAGES YOUR REPORTS CONTACT ADMIN

Your Account Details - Administrator

Name	Bobby
Surname	Jakachira
System Access	OPEN
Country	South Africa
Email Address	btjakachira@gmail.com
Cell No	unknown
Postal Address	unknown
Physical Address	unknown
Gender	unknown
IDNumber	unknown

Edit the Cell

0720

Update

Figure 4.16. Editing User Details.

4.5 The Dwesa Online Application Centre (DOAC) Portal Management

The management of the e-government system is done on the back-end interface by the administrator. There are a few processes that the system automatically runs to achieve certain results. For instance, the sending of document files over the network. The system runs the cron-jobs to send the files on OAC system to the Home Affairs server. Most of the DOAC management processes require human interaction to achieve their goals.

4.5.1 Government File Servers Configuration

Government file servers are servers that receive and store applications of DOAC users at the South African government department that has downloadable forms listed on the DOAC portal. These servers are created at the government department as follows.

At the Government Department

1. Install file server software, e.g. FileZilla.
2. Configure the server to allow only the connection from our Dwesa e-government Server.
3. Create a single directory known as “dwesausers”.

At the Dwesa E-government System

1. Open Admin Interface and navigate to DOAC Management Portal.
2. Choose to add new Server.
3. Details of the Department that host the server are also provided, such as Department Location, Name, and Number of forms online.
4. Provides all the settings of the new Government File Server and submit them to the database.
5. Run an application test to the Government Server. A folder named Admin is be created at the Government server. Inside this folder, three folders are created namely Received, Processing and Processed. A test application form is the found in the “Received” directory.
6. The administrator is responsible for the configuration of the remote servers.

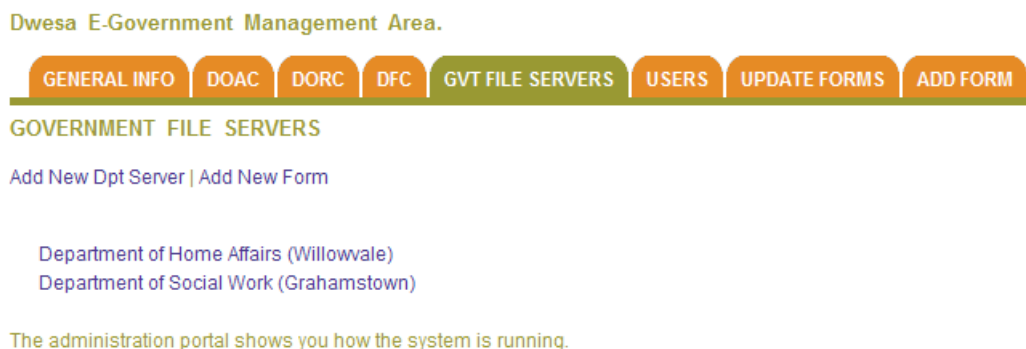


Figure 4.17. List of Government File Servers.

The e-government system has a back-end interface designed to manage these government servers. The interface allows the administrator to add new government servers that can receive user applications.

GOVERNMENT FILE SERVERS

[Add New Dpt Server](#) | [Add New Form](#)

[Department of Home Affairs \(Willowvale\)](#)

Location: Willowvale Department Forms
 Number: 23 | Updated On: 23/12/2009 | Downloaded: 0

Telephone: 0476226374 Applications
 Received: 27 | In Process: 0 | Processed: 0

Description: Last Application Sent On: 02-Oct-2009; 04:57PM

Host Name: 127.0.0.1

Username: newuser [Edit](#)

Port: 21

TIMEOUT: 90 seconds

The test message will be shown here...

[Run Connection Test](#) [Run Application Test](#) [Break Connection](#)

[Department of Social Work \(Grahamstown\)](#)

Figure 4.18. Government File Server Settings.

4.5.2 Updates of Government Forms

All downloadable government forms on the DOAC portal are up-to-date. The process implemented to update the forms ensures the use of accurate government forms at all times. This guarantees that the forms on DOAC portal are the same as the ones available at the government offices so to avoid sharing information that is inconsistent

(a) Design: The updating of forms is done by both an administrator and a cron-job. Figure 4.19 shows the interface an administrator uses to update forms. The interface shows the administrator whether the government forms repository is online and if the DOAC system can get updates from it. The web page also shows the date the last update was done to allow the administrator to see if updates run by the cron-job are going well.

Dwesa E-Government Management Area.

Dwesa E-Government Management Area.

GENERAL INFO **DOAC** **DORC** **DFC** **GVT FILE SERVERS** **USERS** **UPDATE FORMS** **ADD FORM**

The link to home affairs forms repository is configured in "config/db.php". The link hardly changes but in case if it changes this status below will tell you if this system can link to it.

Please note you can only update the forms if the repository status is open

Home Affairs Repository Status: **OPEN**

This means that the repository is still available and you can proceed and update the forms.

Date Lasted Updated: 23/12/2009

NB: Before you update please make sure that "availableforms" folder contains no old forms (files/availableforms)

[Update Forms \(This process takes several minutes\)](#)

The administration portal shows you how the system is running.

Figure 4.19. Updating Forms.

When the updates are possible, the administrator can click the “Update forms” link and the process begins. All forms that are updated are listed as shown on Figure 4.20.

(b) Implementation: We designed a class called *Crawler*. This class accepts a valid URL to the government website as a parameter every time it is declared. The class has protected methods that attempt to read a web page residing on another server, in this case on the government website. We managed to get a link to the website of Home Affairs. The method *get(“links”)* filters all links tags from the metadata of this page using regular expressions. When all links are retrieved, the method then get links that point to the forms. We use regular expressions to get these links. The links with *.pdf* and *.doc* are valid as they point to a file. Each Home Affairs form has a code number that starts with “bi-” followed by 2 or more digits. For instance, bi-154 is the code number for a birth certificate. We also use this code number to filter our links as every link of the form on the Home affairs site specifies it. The code snippet in Figure Figure 4.20 is run by the system updating schedule to update the forms.

```

533      $updater = new formAction();
534      $task = $updater->updateDwesaForms();
535      $upd['message'] = $task['message'];
536      $upd['DateofUpdate'] = date("d-M-Y; h:iA");
537      $SQL = db_insert_array($upd, "form_date");

```

Figure 4.20. Updating Forms.

Figure 4.21 shows a part of the links extracted by the Crawler from the South African Home Affairs site.

```

[25] => href='documents/bi-196.pdf' target='_blank'
[26] => href='documents/bi-84.pdf' target='_blank'
[27] => href='documents/bi-159a.pdf' target='_blank'
[28] => href='documents/bi-159b.pdf' target='_blank'
[29] => href='documents/bi-159c.pdf' target='_blank'

```

Figure 4.21. Links of forms extracted on Department of Home Affairs Site.

```

[12] => href="GSSC%20-%20Health%2017191.pdf" target="_blank"
[13] => href="GSSC%20-%20Social%20Development%2017189.pdf" target="_blank"
[14] => href="gpg%20bursary%20application%20form%202004.pdf" target="_blank"
[15] => href="SCHOLARSHIP%20APPLICATION%20FORM%20WITH%20LOGO.pdf" target="_b
[16] => href="Z143%20Application%20for%20Spouse-Orphan%20Pension.pdf" target
[17] => href="Z300%20Funeral%20Benefits%20Form.pdf" target="_blank"
[18] => href="Z864%20Updating%20of%20Personal%20Particulars%20of%20Member.pd
[19] => href="Z894%20Updating%20of%20Banking%20Details.pdf" target="_blank"
[20] => href="Z102%20Withdrawal%20from%20fund%20Application.pdf" target="_b1
[21] => href="Z125%20Admission%20to%20Fund.pdf" target="_blank"

```

Figure 4.22. Links of forms extracted on Department of Social Work Website.

Once the filtering is done, the class method returns an association array of links, as shown on Figure 4.21 and Figure 4.22. Each array item contains the link name – a form name, the link to that form, the code number and the size of the file. A second method is called to compare the size of each form with the one already on our system. If the size is different, the method copies the file from its link to our server. New forms are also downloaded. Once all updated files are downloaded, the *updateForms()* method of *formAction* class iterates the associative array, creating a form object of each array item. The method uses the *addForm()* method of the same class, which takes a form object and updates each of the form details already stored in the database table called forms. The method returns an array of form objects that have been updated. The names of the forms are shown to the administrator as depicted in Figure 4.22.

4.5.3 Applications Digital Signatures

Users of the DOAC portal digitally sign the documents they send to the Home Affairs government department. It is important to securely store the user's document signatures in the system. When a user chooses to digitally sign his application, the system uses the public key provided by the government department to sign the form. **Figure 4.23** shows a private key that we used in our simulation government server.

```

65      $private_key = <<<EOD
66      -----BEGIN RSA PRIVATE KEY-----
67      MIIBOgIBAAJBANDiE2+Xi/WnO+s120NiiJhNyIButVu6zxqlVzz0wy2j4kQVUC4Z
68      RZD80IY+4wliX2YxKBZKGnd2TtPkcJ/ljkUCAwEAAQJAL151ZeMKHEU2c1qdRKS9
69      sTxCcc2pVwoAGVzRccNX16tfmCf8FjxuM3WmLdsPxYoHrwb1LFNxiNk1MXrxjH3R
70      6QIhAPB7edmcjH4bhMaJBztebNE1VRCEi/bisAwiPPMq9/2nAiEA3lyc5+f6DEIJ
71      h1y6BWkdVULDsm+jpi1XiV/DevxuijMCIQCAEPGqHsF+4v7Jj+3HAgh9PU6otj2n
72      Y79nJtCYmvhoHwIgNDePaS4inApN7omp7WdXyhpZhBmulnGDYvEoGJN66d0CIHra
73      l2SvDkQ5CmrzkW5qPaE2oO7BSqAhRZxiYpZFb5CI
74      -----END RSA PRIVATE KEY-----
75      EOD;
```

Figure 4.23. Example of a Private Key.

This public key would have been created by a private key which is only known to officials at the South African government department. The signature on the user application is then verified at the government department using the private key.

```

77      $public_key = <<<EOD
78      -----BEGIN PUBLIC KEY-----
79      MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANDiE2+Xi/WnO+s120NiiJhNyIButVu6
80      zxqlVzz0wy2j4kQVUC4ZRZD80IY+4wliX2YxKBZKGnd2TtPkcJ/ljkUCAwEAAQ==
81      -----END PUBLIC KEY-----
82      EOD;
```

Figure 4.24. Example of a Public Signature.

4.5.4 DOAC Usage Monitoring

The Dwesa Online Application Centre (DOAC) system has a process that monitors the usage of system resources. This monitors the usage of the disk space by each user in the OAC. Users upload and save documents and files in the DOAC system, and we realized the importance of limiting the total size of memory a user can use to store his documents files. When the user uses up his allocated space, the user is not able to save document files or make any application. In such a case, the DOAC system asks that the user free up some space. Deleted files and documents are zipped and archived elsewhere. The archive is only accessible to the administrator. This was implemented to avoid data overflow on the e-government system server.

(b) Implementation: The code snippet shown in Figure 4.25 shows how the calculation of memory allocation was implemented.

```
223 | $gui->assign('appsize',zone_user::ByteSize($sql3));  
224 | $gui->assign('limit',zone_user::ByteSize($limit));  
225 | $gui->assign('leftsize',zone_user::ByteSize($limit - $sql3));
```

Figure 4.25. User allocated memory calculation.

The *zone_user* class has a function that converts the figures into megabytes and it is called here to pass the correct values to the three parameters that state the figures for each user.

4.5.5 Dwesa E-Government Services Management

The e-government system requires a dedicated functionality that manages all the services. The objective of the functionality is to allow an administrator to control the running of Dwesa e-government services. Services are managed independently of each other. The functionality also shows a service that is being used regularly. This allows the administrator to allocate enough system resources to such a service. The screenshot in Figure 4.26 shows the services that constitute the Dwesa e-government system.

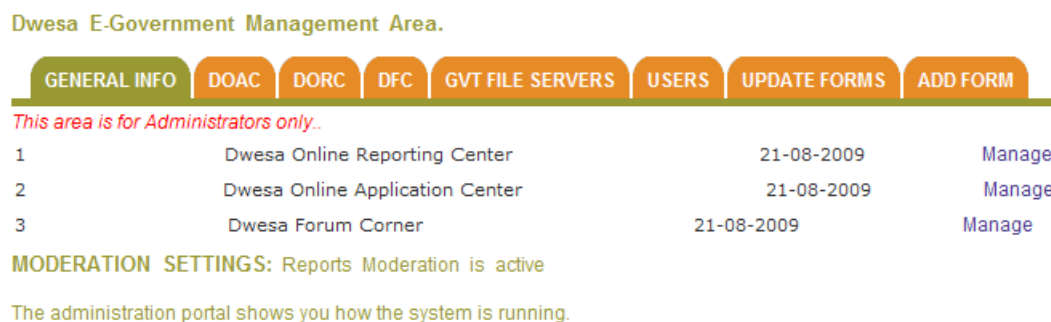


Figure 4.26. Services Management Portal.

As shown in figure Figure 4.26, each service has a launch date and an expiry date. This is important to remind the administrator when a service is supposed to be taken offline. The last date the service was used is also stated for debugging purposes. The number of users that has used each service to this day is also shown. The screenshot also shows services that are offline or online.

```

535  $Sid = $post['id'];
536  $act['Status'] = 0;
537  $SQL = "SELECT Status FROM services where ID='$id'";
538  $status = sql_fetch_one_cell($SQL);
539  if($status==1)
540  $act['Status'] = 1;
541  $suc = db_update_array($act,"services","ID",$id);
542  if($suc)
543  {
544  $sGlobals->color="#00FF00";
545  }

```

Figure 4.27. Activation of Services.

As the e-government system grows, more and more services will require to be integrated into the system. Functionality exists in the system that allows for the adding of new services. New services should be developed and integrated as plug-ins with the e-government system.

The administrator provides the name of the service being launched. A description of the service is required. For implementation of the service, the name of the default page is required. This name is passed on as the URL that the users need to access it. The piece of code shown in Figure 4.28 shows how the functionality was implemented.

```

349 $query = "INSERT INTO services (ServiceName, Description,
350       DateLaunched,ServiceUrl ) VALUES ('$serv', '$desc','$dat', '$app' )";
351 if(sql_query($query)){
352     $gui->assign("smessage", 'The service has been Launched');
353 }
354 else{
355     $gui->assign("smessage", 'An error occured');
356 }

```

Figure 4.28. Integrating New Services to the system.

4.5.6 New and On-Request Government Forms

Because not all government forms are listed on the DOAC portal, a user who seeks an unavailable form can send a request to the Home Affairs servers for that form. Two processes run to find the requested form. The first process is to update the list of forms in the DOAC system. When the updates contain the form, a message is sent to the user with the link to download the form. If the updates fail to get the form, the second process of sending a request to the Home Affairs officials is run.

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO FORMS FILL SIGN SEND TRACK VIEW APPS REQUEST

This is where you will request certain forms from various South African government Departments

Type the name form

Figure 4.29. Requesting a Government form.

4.5.7 Application Notifications

The Dwesa Online Application Centre (DOAC) has an added functionality that allows users to receive notifications during all stages of application processing at a government department. These notifications are received through emails and SMS, using the user email address and cell phone number respectively. The user is notified of the processing stage his application is at. Figure 4.30 shows how the SMS functionality was implemented in the system. The SMS delivery method is only available when credits are available from the SMS Gateway server which is provided by the TM4B (Text-Messaging for Business) SMS Gateway. The feature depends upon the availability of financial resources in the project.

```

63 //initialize the request variable
64 $request = "";
65 $param["username"] = SMS_USERNAME;
66 $param["password"] = SMS_PASSWORD;
67 $param["msg"] = $message->getMessage();
68 $param["to"] = $message->getCell();
69 $param["sim"] = SMS_SIM;
70 $reporter = $message->getSender();
71 $param["from"] = $reporter->getCell();
72 $param["route"] = SMS_ROUTE;
73 //url of the gateway's interface
74 $url = "http://www.tm4b.com/client/api/send.php";
75 $ch = curl_init();
76 curl_setopt($ch, CURLOPT_URL, $url);
77 curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
78 curl_setopt($ch, CURLOPT_POST,1);
79 curl_setopt($ch, CURLOPT_POSTFIELDS,$param);
80 $response = curl_exec($ch);
81 curl_close($ch);

```

Figure 4.30. SMS functionality.

4.5.8 Uploading New Forms

The administrator can manually upload forms online in the DOAC Management Portal. The forms are instantly made available for users to download them. Information required to add a new form includes the version number of the form, a description of the form, the department which the form originates from and the server identity number that should receive applications made using the form.

4.6 The Dwesa Online Reporting Centre (DORC) System Management

In an effort to ensure reliability and efficiency of the Dwesa Online Reporting Centre (DORC), we implemented processes that allow easy management of the portal. The DORC portal has a system performance monitoring back-end interface, accessible to a user with administration privileges.

4.6.1 DORC System Performance

Whenever a problem arises in the DORC portal, the administrator is notified through an email of that problem. Each problem has a number associated with it for easy user assistance. This number allows the administrator to respond quickly to problems encountered by the users.

An error thrown into the DORC System is identified by three digits. For instant, 115 refers to the error when the user is trying to send a report with an incorrect Security Message Code. The

error number 119 shows that the SMS delivery method of forwarding reports is not available. The error number 121 depicts a report delivery failure to the administrator.

4.6.2 DORC Portal Access

The Dwesa Online Reporting Centre (DORC) is accessed by community members as well as by the administrator. Community members are given access to the portal as soon as their registration process is finished. The portal is accessed after successfully signing in.

The *zone_dorc* class on initial running, checks to see if the user object has its *ORCstatus* property set as true. If it is true, the user is granted access to any page that falls in the DORC portal.

4.6.3 DORC Usage Monitoring

The usage of the DORC portal by users is carefully monitored to ensure that the system's resources are used sparingly. DORC portal users are allowed to send a specified size of reports. The limit of disk space that a user can use is defined by the administrator in the configuration settings. Users who have no space left are allowed to delete old reports.

4.6.4 Responding to Villagers' Reports

Each registered and logged-in user can respond to all reports in the system. These reports are made to highlight grievances of the Dwesa villagers. The system's screening process then moderates all the responses. The screenshot in Figure 4.31 shows the interface used to respond to the message sent by the administrator. The *Denounce* link shows that the report was previously announced and may now be removed. Only when an administrator is logged in can reports be announced or removed from the bulletin board.

Welcome to Dwesa Online Reporting Center (DORC).

[GENERAL INFO](#) [SEND A REPORT](#) [SMC](#) [VIEW REPORTS](#) [CONTACTS](#)

Number of Reports: 2

Sent To: Police

The urgency is normal

Message : There is a man next door who beats his child evryday. We... [Boyd : Jakachira (bjakachira@gmail.com)]

Sent on : 11-Dec-2009; 04:29PM RESPONSE: pending

[Delete](#) | [Respond](#) | [View Responses](#)

Please type a short respond below for this report. Long responds will be not send. Illegal responds will be deleted. Thank you.

[Respond](#)

Figure 4.31. Responding to Reports.

4.7 Dwesa Forum Corner (DFC) System Management

The e-government system provides an interface for the management of the Dwesa Forum Corner. Administrative processes that run on this interface include the activation of the DFC portal, assigning memory to the DFC users, and portal statistics.

4.7.1 DFC System Access

DFC system access is restricted to authenticated users. Because the DFC system is an independent program module with its own processes, we implemented features that manage access to the portal.

The DFC portal is only accessible to a user who has an account activated to access it. A user with administration rights can, for any particular reason, choose to bring the DFC portal offline, leaving the DOAC and DORC systems still running. This is important when there is maintenance that needs to be done on the DFC portal.

All DFC users are required to log into the e-government system before they can use the portal. Users should have a DFC access status of 1 if they are to use the DFC portal. Users with a status of 2 should have had their accounts suspended for some reason only known to the administrator and the DFC portal. Users who misuse the system are suspended by the DFC system screening process.

4.7.2 DFC Usage Monitoring

The use of the DFC portal, as with other portals, is monitored independently. Users are allowed to post irrelevant discussion topics. Only topics that our system considers valid are accepted in the DFC portal. We designed a process that determines the memory disk space usage by each user. Each user, upon registration, is designated a pre-defined amount of disk space to use. The size of the space is determined by the number of records the user has so far. We calculate the total byte size of each user post each time a user visits the DFC portal. Inadequate disk space means the user cannot participate in the discussion forum until some space is freed up by deleting old reports.

4.7.3 Moderating DFC User Posts

The Dwesa Forum Corner (DFC) was implemented to allow public discussions among Dwesa villagers. Public notices and public alerts are also posted on the DFC as announcements. Before a user message is posted to the public, the DFC system screens the message to verify that it does not contain abusive language. This is handled by a database event object that runs every ten minutes. The event invokes a stored procedure that screens the title and the body of the message and the number of illegal messages is accumulated. After this screening process, all messages meant for discussions are passed as legal. Public notices and alerts await the approval of the administrator before they are made public.

4.7.4 Public Announcements Verification and Broadcasting

Users have an option to make public announcements on the Dwesa Online Reporting Centre (DORC) system or the Dwesa Forum Corner (DFC) system. The announcements are created as messages in the DFC system or as reports in the DORC system. These types of messages are verified by the DORC portal's verification system and by the administrator. A message posted can only qualify to be an announcement with the approval of the administrator. A process was designed to flag a message as an announcement. The code in Figure 4.32 shows a part of this process.

```
238     case 13: //announcing report..
239         $r_action = new DwesaReportAction();
240         $id = $inPath[2];
241         $report = $sGlobals->reps[$id];
242         $result = $r_action->announceReport($report);
243         unset($sGlobals->reps);
244         break;
```

Figure 4.32. Making an Announcement.

4.8 Conclusion

The e-government system is a web-based system dedicated to serving the Dwesa villagers with little human intervention. To achieve this goal; the implementation of the system back-end interfaces was crucial. This chapter described how the system back-end interfaces were designed and implemented. The back-end interfaces show all processes that perform the user management and system management. This chapter described how these processes were implemented. The chapter described the database server objects designed, which include views, events, stored procedures and triggers.

Also, the chapter discussed the system architecture and all components that the Dwesa e-government system is made up of. It explained the distribution of the e-government system.

Chapter Five

Design and Implementation of the Dwesa E-Government System's Functionalities in Front-End Interfaces

The chapter will describe the implementation and design of the three main functionalities of the Dwesa E-Government System. The security features implemented on each of the functionalities will also be explained.

5 Introduction

This chapter will discuss the design and implementation of the presentation and transactional phases of e-government system. The chapter will explain the processes in the Dwesa Online Application Centre (DOAC), the Dwesa Forum Corner (DFC) and the Dwesa Online Reporting Centre (DORC). The chapter will then describe the security mechanisms applied on the DOAC, DFC and DORC front-end portals.

5.1 The Dwesa Online Application Centre (DOAC)

The Dwesa e-government system offers an online application centre functionality. The portal is on the front-end, allowing the Dwesa community members, to download, sign and send important documents to the Department of Home Affairs. Figure 5.1 shows the DOAC portal.

A user needs to follow one or more steps to apply for a document. The four steps required to successfully apply for any document from any government department are stated below.

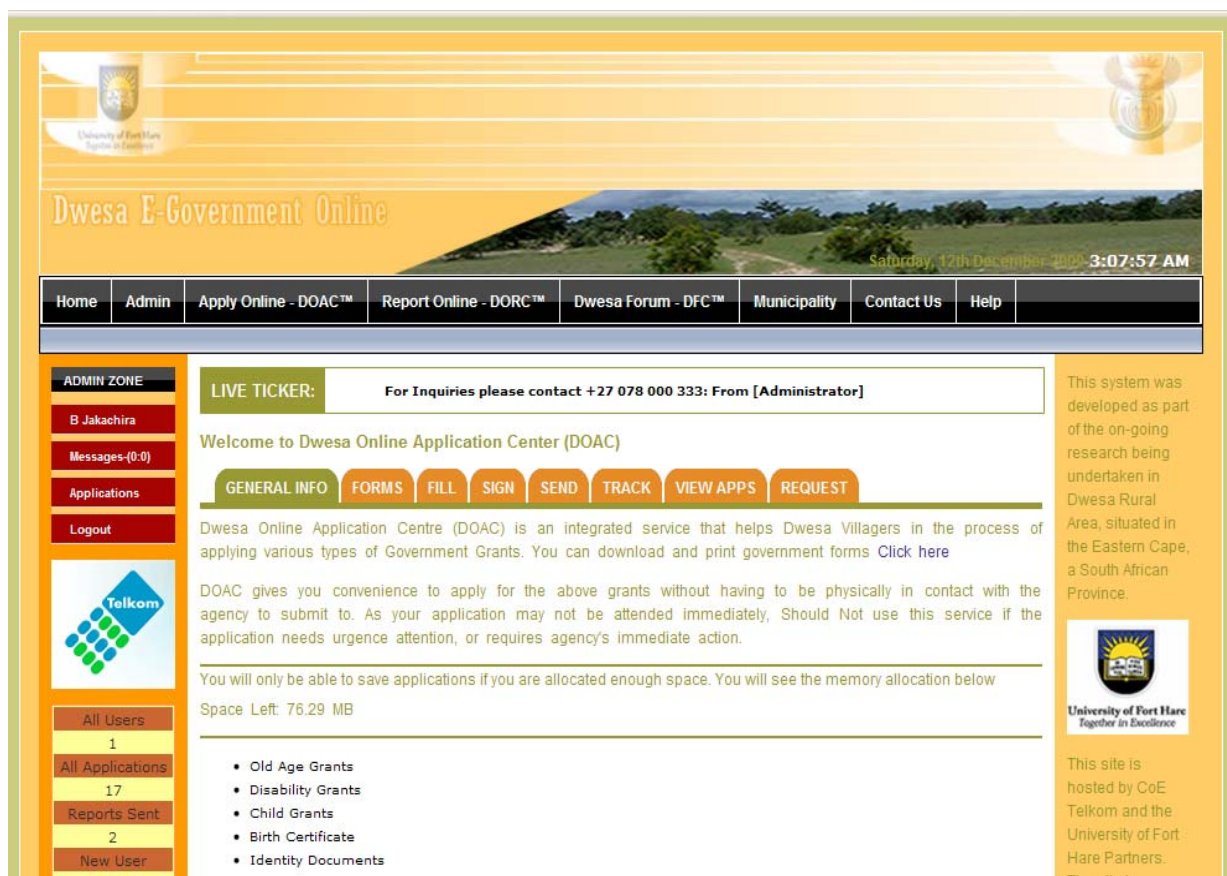


Figure 5.1: The DOAC portal.

Step 1: Download the Application Form from the list of forms. The user downloads the form of the document he wants to apply. He then prints the form to fill it by hand or he can choose to fill it online using the PDF Online Filler system.

Step 2: If the form filling-in is completed, the user then browses to the location of the filled-in form on his machine and selects it to save or send.

Step 3: The user is required to choose the type of application he wants to send. This ensures quick processing of the application form and as well as to check if the user has applied for this document before.

Step 4: At this stage, the user can choose to save the filled-in application form on the system or send it to the relevant government department.

5.1.1 Listing Government Forms on the DOAC Portal

The process was implemented to provide a portal with a list of downloadable government forms. The forms come from various government institutions. This allows the Dwesa e-government system to present itself as a government forms one-stop shop, dedicated to the community members.

(a) Design: All forms are categorized according to the types of government documents they apply to. All South African government forms bear a code number (for instance bi-84 is the code number for a form to apply for a death certificate). Each form is identified on the DOAC portal by this code name and its actual description. This was done to allow users to easily identify the forms they may need. Each form is listed together with its size. Figure 5.2 shows the forms listed on the birth category

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO	FORMS	FILL	SIGN	SEND	TRACK	VIEW APPS	REQUEST
--------------	-------	------	------	------	-------	-----------	---------

South African Government Forms.

Birth	Death	Grant	Passport	ID	Foreign
-------	-------	-------	----------	----	---------

These Home Affairs forms are hosted by Dwesa System (last updated on)

BI-1682 - Application For Insertion Of Natural Father'S Particulars In The Birth Register Of Child : (71.13 KB)

BI-59 - Application For Amendment Of Birth Registration Of A Child Born Out Of Wedlock : (48.25 KB)

BI-154 - Application For A Birth Certificate : (67.30 KB)

BI-862 - Application For Dual Citizenship [9(2)] : (104.34 KB)

BI-1666 - Application For Exemption In Regard To The Loss Of SA Citizenship [26(4)] : (42.48 KB)

BI-1664 - Application For Retention Of SA Citizenship : (70.95 KB)

BI-529 - Determination Of Citizenship Status : (103.39 KB)

BI-175 - Application For Resumption Of SA Citizenship : (45.78 KB)

Figure 5.2 Listed Forms on Birth Category.

(b) Implementation: Two PHP classes were designed to achieve proper listing of forms. The *form* class defines a form object and *formAction* class possesses all method tasks that can be carried out on the form object. The latter has a method called *getForms()* which takes a form object as an optional parameter. We call this method to get forms data stored in the *dwesaCentre* database forms table. The method returns an associative array of forms objects. This whole process is runs once only if the *zone_doac* class is run at the first session and we store every variable in the session. We then list all forms together with their respective details.

5.1.2 Downloading Government Forms on the DOAC

This was implemented to allow users to download government forms from the DOAC portal in the comfort of their labs in the Dwesa rural community. We designed it in a way that gives the Dwesa users more information about what they are about to download. The downloading is made quicker as these forms reside on our server.

(a) Design: When a user clicks on a form link, more information about that form is displayed. The name, file type and the code number of that form are displayed together with the size of the form. The DOAC attempts to calculate an estimated time that the form takes to download and state it together with the estimated user bandwidth size. It also states the date the form was last updated. The form is downloaded once the user clicks the download button. Figure 5.3 shows the details of a *bi-73* form.

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO FORMS FILL SIGN SEND TRACK VIEW APPS REQUEST

South African Government Forms.

Birth Death Grant Passport ID Foreign

You are about to download a form with the following details:

Application for amendment of birth registration of a child born out of wedlock.

The form originates from a South African Department that is not mentioned.

This form was last updated on 21-Sep-2009; 05:57PM.

You should verify this code on your form: **BI-59.**

Estimated Bandwidth: 542.7375kb/s

Estimated Time of Download: 0 seconds

 Size: 48.25 KB

DOWNLOAD

Figure 5.3: Downloading a Form.

(b) **Implementation:** Once a user clicks on a form link, a page with the form details is displayed. All the form details are retrieved from a form object stored in the session. The system calculates the time this page has taken to load just by subtracting the start time from end time. With that we estimate the kbps which we assume to be the bandwidth, and then we run arithmetical calculations of how long that form can take to download. We implemented this using JQUERY JavaScript library.

5.1.3 Filling in Forms

When applying for important documents, the user should fill in the form, preferably online. Several downloadable application forms are in a read-only *PDF* format, whilst a few are in word document format. The e-government system developed ensures that users are able to fill in the forms in word document format. Those that are read-only can be filled in at PDF Online Filler website. Figure 5.4 shows the PDF Filler Interface that the Dwesa E-government system is linked to. This allows smooth filling in of *pdf* documents online. The form shown below is for requesting for Taxpayer Identification Number and Certification and was used as a sample. The user saves the document on his machine before he uploads it to the system.

Figure 5.4: Filling in a Pdf Form Online

The implementation of the online form filling process was made possible through the use of web services. The user, upon downloading the form, is asked if he wants to fill in the document online or just print it. If he chooses to fill it in online, the user is re-directed to a web service that allows him to fill in the document online. After filling in the form, the user now can upload it to the Home Affairs server.

5.1.4 Signing Applications

We implemented this process on the DOAC portal to ensure secure and safe file transfer between the Dwesa central server and the government servers. This is part of our concerted effort to provide a secure system that does not deprive users of their privacy.

(a) Design: Our e-government system uses a private and public key provided by the government departments to sign the documents. Users should provide the application identity number that identifies the application that should be signed. Only applications that are filled in and saved are signed.

Figure 5.5: Signing an application.

(b) Implementation: a function called *signApplication()* would take an application object as a parameter and uses a PHP encryption library to sign the document content and if successful returns the signed application as an object. Figure 5.6 and Figure 5.7 show examples of a private key and a public key. Figure 5.8 shows how this is done.

```

65     $private_key = <<<EOD
66     -----BEGIN RSA PRIVATE KEY-----
67     MIIBOgIBAAJBANDiE2+Xi/WnO+s120NiiJhNyIButVu6zxqIVzz0wy2j4kQVUC4Z
68     RZD80IY+4wliX2YxKBZKGnd2TtPkcJ/ljkUCAwEAAQJAL151ZeMKHEU2c1qdRKS9
69     sTxCcc2pVwoAGVzRecNX16tfmCf8FjxuM3WmLdsPxYoHrwb1LFNxiNk1MXrxjH3R
70     6QIhAPB7edmcjH4bhMaJBztcNE1VRCEi/bisAwiPPMq9/2nAiEA3lyc5+f6DEIJ
71     h1y6BWkdVULDSM+jpi1XiV/DevxuijMCIQCAEPGqHsF+4v7Jj+3HAgh9PU6otj2n
72     Y79nJtCYmvhoHwIgdNdPaS4inApN7omp7WdXyhPZhBmulnGDYvEoGJN66d0CIHra
73     l2SvDkQ5CmrzkW5qPaE2oO7BSqAhRZxiYpZFb5CI
74     -----END RSA PRIVATE KEY-----
75     EOD;

```

Figure 5.6: Example of a Private Key.

```

77     $public_key = <<<EOD
78     -----BEGIN PUBLIC KEY-----
79     MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANDiE2+Xi/WnO+s120NiiJhNyIButVu6
80     zxqIVzz0wy2j4kQVUC4ZRZD80IY+4wliX2YxKBZKGnd2TtPkcJ/ljkUCAwEAAQ==
81     -----END PUBLIC KEY-----
82     EOD;

```

Figure 5.7: Example of a Public Key.

Figure 5.8 shows how a digital signature for a Dwesa application is created. The code lines are part of the *signApplication()* method (Fully listed in Appendix B)

```

29     $form      = $application->getForm();
30     $rsa_obj    = new Crypt_RSA;
31     $error_handler($rsa_obj);
32     $rsa_obj->setErrorHandler($error_handler);
33     $signature  = $rsa_obj->createSign($form->getFormContent(),
34                                     $key_pair->getPrivateKey());
35     if(!$signature){
36         $result['message'] = "Your application could not be signed.
37                             Try again later";
38         $result['result'] = false;
39         return $result;
40     }

```

Figure 5.8: Creating a Signature.

Figure 5.9 depicts how a signature on an application is validated. The method *validateSign()* of RSA Crypt class object accepts the content of the signed document, the signature and the public key. These are validated to see if the signature on the file corresponds to the one the system has.

```

42 // signature checking
43 $result['result'] = $rsa_obj->validateSign($form->getFormContent(),
44                                         $signature,
45                                         $key_pair->getPublicKey());
46 $application->setSignature($signature);
47 $result['returned'] = $application;
48 return $result;

```

Figure 5.9: Validating a Signature.

The four lines of code shown in Figure 5.9 are part of a DwesaValidator class method called *validateApplication()* (Listed in Appendix B).

5.1.5 Saving Applications

Saving user applications is vital for future reference. It also provides flexibility to the process of doing long applications. Users are able to take breaks between processes and resume later with their previous information still intact.

(a) Design: Users can choose either to send their application or save it in the middle of applying. Figure 5.10 shows how users save their application.

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO FORMS FILL SIGN SEND TRACK VIEW APPS REQUEST

This where you will send your application. You are also required to have the applications requirements at hand (as files or images). You will be asked to send these requirements as well.

Below you will click on Browse and find your file from the directory that you saved and select it.

Your Application has been saved. Please note this Application ID Number: 6.

Choose Option? Save ☒ Sent ☐

Browse to your form C:\xampp\htdocs\final\dwesagov\ Browse...

Application Type foreign

Choose the Version bi-159f

There are no requirements to send this application. You can proceed.

Submit Your Form Reset

Figure 5.10: Saving an application.

Users are notified if their application is saved, in which case an Application identity number is given to the user. This number is required when accessing or signing the application.

(b) Implementation: When a user wants to save an application, he only submits the form of that application, other details about his will be already in the system prior to applying. This is where everything started to get interesting. When the user selects his form, he should also choose the type of the application in the drop down menu; another drop down menu appears and asks the user to select the code number of that form. The code number always appears on the top right corner of the front the form.

The *application* and *applicationAction* classes were designed to handle these user requests. An application object is created upon clicking submit. The user's application details are stored in the *applications* database table, along with a user identity number. The *saveApplication()* function takes an application object as its parameter and inserts the application into the database if the Status object property of application is set to 1 (a status of 1 on an application shows the application is still on our servers). The form is stored in the user's directory called "saved" ("files/username/saved/..").

5.1.6 Sending Applications

The sending of applications is one of the major processes implemented for DOAC portal. Applications are sent to the respective government department together with their requirements which could be files as well. Users are excused from knowing the department which receives their application. They are only told where their application has been sent. This process completes one of our goals of providing a complete public service online.

(a) Design: When a user wants to send an application, he should choose the option of sending as shown on Figure 5.11. The user then selects the form that is already signed and filled. He then chooses the type of document which he is applying for before he can choose the form identity code in the drop-down box that appears.

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO FORMS FILL SIGN SEND TRACK VIEW APPS REQUEST

This where you will send your application. You are also required to have the applications requirements at hand (as files or images). You will be asked to send these requirements as well.

Below you will click on Browse and find your file from the directory that you saved and select it.

Your application has been sent to the Department of Home Affairs. Tracking Number: DR322720

Choose Option? Save ☐ Sent ☒

Browse to your form

Application Type

Choose the Version

There are no requirements to send this application. You can proceed.

Figure 5.11: Sending an application.

If the application has requirements, the user can select the files, which could be scanned images. The user then clicks the ‘submit’ button to send the application. If the system can send the application, the user is provided with a tracking number. If the system cannot send the application for some reason, the user is notified that the application is placed in a queue of applications that is sent by the cron-job.

(b) Implementation: To implement the process of sending files, we set up a simulation of a remote government file server, using FileZilla. The `addApplication()` method of `applicationAction` class would attempt to upload an application to this remote server if the Status property of the application object is set to 3. We designed `DwesaTransmission` and `DwesaTransmissionAction` classes. The `DwesaTransmission` class defines the properties of a FTP connection.

```
(000001) 12/11/2009 16:38:54 PM - (not logged in) (127.0.0.1)> 220-FileZilla Server version 0.9.31 beta
(000001) 12/11/2009 16:38:54 PM - (not logged in) (127.0.0.1)> 220-written by Tim Kosse (Tim.Kosse@gmx.de)
(000001) 12/11/2009 16:38:54 PM - (not logged in) (127.0.0.1)> 220 Please visit http://sourceforge.net/projects/filezilla/
(000001) 12/11/2009 16:38:54 PM - (not logged in) (127.0.0.1)> USER newuser
(000001) 12/11/2009 16:38:54 PM - (not logged in) (127.0.0.1)> 331 Password required for newuser
(000001) 12/11/2009 16:38:54 PM - (not logged in) (127.0.0.1)> PASS *****
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> 230 Logged on
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> CWD Received
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> 250 CWD successful. "/Received" is current directory.
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> CWD Boyd
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> 250 CWD successful. "/Received/Boyd" is current directory.
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> MKD bi-862
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> 257 Directory created successfully
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> CWD bi-862
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> 250 CWD successful. "/Received/Boyd/bi-862" is current directory.
(000001) 12/11/2009 16:38:54 PM - newuser (127.0.0.1)> disconnected.
```

Figure 5.12: Government Server Simulation.

The latter class has methods called *sendbyFTP()* and *sendbyCURL()* that use the File Transfer Protocol to send files. The former method always runs first, the latter acts as a second optional method if the first fails. These functions take an application object as a parameter and attempt to put the file on the remote server. The application is then sent to the remote server, but if this fails, the application Status property is changed to 2 (a status of 2 shows that the application is put in a sending queue).

In both cases, the application is saved in the database, showing its status and the user is furnished with an 8-digit tracking number to track the application. Then, the user is notified accordingly to the status of his application. Applications are sent using three methods. The main method used is through the File Transfer Protocol. This class is utilized by DwesaTransmissionAction PHP class. The FTP functions are used to upload the file to the remote server through a dedicated password-secured connection. Upon successful uploading, the department official is able to open the directory that contains this application. The directory name is an application identity number that uniquely identifies that application.

When the FTP fails, the system attempts to send the application as an email attachment to the respective department. This method however depends on the availability of a government department's valid email address dedicated to receive these applications. Such email addresses can be added to the system whenever they are available.

When the email method fails, the system publishes the application on the Dwesa system portal. The application becomes available on the back-end interface dedicated only to this department. The department official has to download it from the Dwesa servers.

```
394 //validate application and send if it is valid..
395 $result['application'] = $validator->validateApplication($application);
```

Figure 5.13: Validating an application.

```
401 if($result['result']){
402     $result = $appAction->addApplication($application);
403     if($result['result']) unset($GLOBALS->user_apps);
404 }
```

Figure 5.14: Adding an application.

The snippet of PHP code in Figure 5.15 shows part of the *addApplication()* function implemented.

```

312 //transmitting the application to the remote server
313 private function transmitApplication($application)
314 {
315     $transmit = new DwesaTransmissionAction();
316     //using ftp
317     $result['ftp'] = $transmit->sendbyFTP($application);
318     //using email if ftp connection fails
319     if(!$result['ftp']['result']) $result['email']=$transmit->sendbyEmail($application);
320     //sending to the back-end is the last option
321     if(!$result['email']['result']) $result['dwesasys']=$transmit->sendbyDwesaSys($application);
322     return $result;
323 }

```

Figure 5.15: Transmitting an application.

When the application is successfully sent, a tracking number is provided for that application. The applicant can then track the processing of his application.

5.1.7 Tracking Applications

Tracking applications are vital to keep our users up-to-date on what is happening at any time with their applications.

(a) Design: To track an application, a user needs a tracking number like the one shown on Figure 5.16. Users who track their applications are those who, when applying, had chosen the option of sending the application, because this is the only case when the user can get a tracking number. When a user enters a tracking number, for instance as on the interface shown in Figure 5.16, he is notified of the status of his application.

The image below shows that an application is at the Government Department and it is still to be processed.

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO **FORMS** **FILL** **SIGN** **SEND** **TRACK** **VIEW APPS** **REQUEST**

This is where you will track your application you made to South African Departments. When you applied, you were provided with this track number. You can view the previous application by clicking "View Apps" on the tabs. You will find the track number if you have any previous applications made to the Government Department.

Tracking Number

Application for a study permit

Your application is still to be sent.

Your application will be sent in 24 hours

Date Applied: 11-Dec-2009; 09:44PM.

Figure 5.16: Tracking an application.

(b) **Implementation:** The function *trackApplication()* when called takes the application object as its parameters and return the tracker that depicts the status of the application at that time. Part of this function is shown in Figure 5.17.

```
311 public function trackApplication($application)
312 {
313     $valid = strlen(trim($application->getTrackingNumber()));
314     if($valid != 8)
315     {
316         $result['app_name'] = "<em>Not available.</em>";
317         $result['stage'] = "Unknown";
318         $result['note'] = "<em>The track number is invalid - 100.</em>";
319         return $result;
320     }
321
322     //..... more code of trackers left out....
323
324     if($application->getArrivalDate() == $application->getDateLastModified()) //tracker[3]
325     {
326         $tracker['color'] = "#00FF00";
327         $tracker['note'] = "Your application is now being processed. Check in two weeks";
328         $tracker['stage'] = "Attention";
329     }
330     return $tracker;
331 }
```

Figure 5.17: Creating a tracking number.

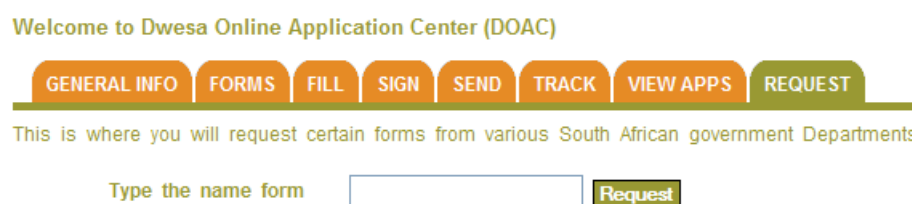
Our object of application class has a property called status that keeps track of the location of the application between our server and the remote server. A status of 1 is set whenever a user decides to save the application. A status of 2 is defined only when a user attempts to send the application and the system decides that the time is not convenient to send the application or the remote server is not responding. The system then saves the application with a status of 2. Users

are allowed to send their applications at any time, provided network conditions allow this. If the network is not overloaded, the *transmitApplication()* method of *applicationAction* class attempts to send it, and if successful the status is set to 3. The application is then waiting to be sent by a cron-job. The cron-job runs to send applications as well as gather more information on the remote server about applications that have a status of 3. The information gathered is used to determine the status of applications that were sent previously. Our process just checks the directory in which the application is situated – received, processing or finished directories. All the information is updated in the database at night when the cron-job does its work. An application that is being processed is depicted by a status of 4. If the application has been processed, the application status changes to 5. All this gives more details about the status of the user application.

The user can send a secure message through the system to inquire about the application form.

5.1.8 Requesting Other Forms

The e-government system provides a downloading portal for all application forms of important documents. While it strives to list all the necessary forms for users to download, the system may not be able to get some of the application forms that can only be provided upon request to the Home Affairs Department. The reason some of the forms are not listed could be because of government laws and policies enforcing the Department not to divulge such application forms to everyone. The other reason for not locating a government form could be because the system has not been able to provide an updated version for that form for a period of time and has hence discontinued it until the most recent and latest version is obtained. Figure 5.18 shows the interface where a user can provide information of the form he is requesting.



Welcome to Dwsa Online Application Center (DOAC)

GENERAL INFO FORMS FILL SIGN SEND TRACK VIEW APPS REQUEST

This is where you will request certain forms from various South African government Departments

Type the name form Request

Figure 5.18: Requesting an application.

The implementation of online form requesting was done by forwarding users' requests to the Home Affairs Department. This is only done when the Application Updating Manager fails to provide a form that is requested

5.1.9 Viewing Applications

Users should be able to see all the applications they have made for easy future reference. The DOAC portal provides a user with a view of all his applications. This is also important as a user may be asked to re-send the application to the government department.

(a) Design: A user is allowed to view all his applications. These could be those that are saved, sent or are waiting to be sent. Each saved application is listed with application identity number while a sent or a waiting application has a tracking number as shown in Figure 5.19. Users who forget their tracking number or application identity number can easily check them.

Welcome to Dwesa Online Application Center (DOAC)

GENERAL INFO	FORMS	FILL	SIGN	SEND	TRACK	VIEW APPS	REQUEST
--------------	-------	------	------	------	-------	-----------	---------

SENT APPLICATIONS		
Track No	Application	
DF262570	7: Application for work permit: Self-employment -11-Dec-2009; 09:25PM	Download More Delete
DR322720	8: Application for a study permit -11-Dec-2009; 09:44PM	Download More Delete

copyright & reserved 2008

Figure 5.19: Sent Applications.

SAVED APPLICATIONS				
Appl ID	Application	Code	Date	
1	bi-159c_0085.pdf	bi-288	11-Dec-2009; 04:53PM	Download More Delete
2	bi-288_0085.pdf	bi-59	11-Dec-2009; 04:54PM	Download More Delete

Figure 5.20: Saved Applications.

APPLICATIONS WAITING TO BE SENT		
Track No	Application	
DE643383	3: Power of attorney in respect of an application/extension of (an) immigration permit(s) or temporary residence -11-Dec-2009; 05:00PM	Download More Delete
DB275946	4: Application for work seeker's permit -11-Dec-2009; 05:01PM	Download More Delete

copyright & reserved 2008

Figure 5.21: Waiting Applications

(b) Implementation: When a user visits the DOAC portal, the *getUserApplications()* method of the *applicationAction* class retrieves all applications for the user from the database and stores them in the session variable.

5.1.10 DOAC Security Features

The DOAC portal has security features implemented to provide a secure and safe e-government service to community members and government agencies. The content which the DOAC portal

receives from the all users is also checked to see if it does not contain malicious content that may be harmful to the DOAC system or its users.

(a) Design: The DOAC portal is accessible only to the users defined by the administrator. Users who visit the e-government website are able to register to access general information about the three integrated functionalities, one of which is the DOAC. All forms and applications are scrutinized before they are submitted to the DOAC system.

(b) Implementation: We designed a special class called *DwesaValidator* that validates content which the system receives from our users. When a user is saving or sending applications, two methods of this class are called to validate the content.

The method *validateForm()* takes a form object as a parameter and checks that the file size is not unnecessarily big and that it is actually a file. It also checks whether the file is executable, in which case it is not accepted. The method checks the type of the file which should be a pdf or a word document.

```
376 //validating the form used for the application
377 $validator = new DwesaValidator();
378 $result['form'] = $validator->validateForm($form);
```

Figure 5.22: Validating a form.

The name of the file is also restricted to a specified number of characters. Most importantly, the method gives the file a new name that is going to be used from now on in the server. If the form passes all these tests, we declare it as an application object which should also be validated.

```
394 //validate application and send if it is valid..
395 $result['application'] = $validator->validateApplication($application);
```

Figure 5.23: Validating an application.

An application should contain a valid form and should contain a digital signature. The *validateApplication()* method takes an application object as its parameter. It verifies the digital signature using a private key provided by the government department. If all that passes, it returns the validated application that is ready to be submitted to our system or to be sent to the government department.

5.2 The Dwesa Online Reporting Centre (DORC)

The Dwesa Online Reporting Centre (DORC) is an e-government system functionality that allows the villagers to launch reports of grievances to various government agencies' contacts. The portal permits users to send reports concerning road damage, environmental problems, child and women abuse, lost identity documents and crime. The DORC portal is one of the three standalone e-government functionalities that our system has.

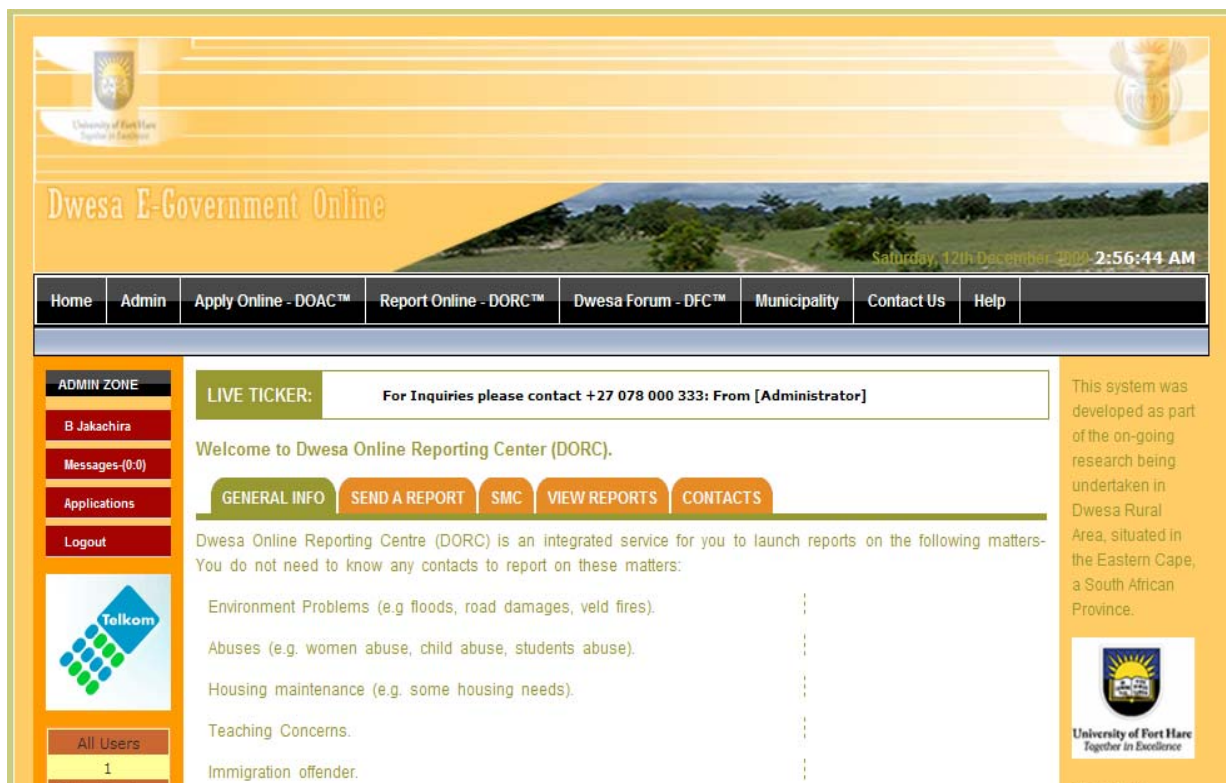


Figure 5.24: The DORC Portal.

5.2.1 User Requirements for Reporting

To send report, users are required to provide a valid message code number. Upon registering on the e-government system, a user is given a message code. This is an important process to avoid non-repudiation of reports by the villagers. The user is not required to provide details of the intended recipient of the report. The DORC portal intelligently routes the reports to the government agency.

The ORC portal accepts all the reports launched to the municipality by the user and creates an automatic knowledgebase.

5.2.2 Sending User Reports

The reports are sent to a government agency once the user clicks the send button. Users do not need to know the government agency they should report to.

(a) Design: A user provides a valid message code, email address, cell number and the message. He should also specify whether the report is an urgent one or not. Urgent reports are sent using SMS delivery methods. Those that are not urgent are sent to the government agency's back-end interface. The sender is notified of the government agency that is going to attend to his report.

Figure 5.19 shows the interface to send reports.

Welcome to Dwesa Online Reporting Center (DORC).

GENERAL INFO SEND A REPORT SMC VIEW REPORTS CONTACTS

Type your report below and the system will send it to the relevant department. The message should not contain abusive, swearing or defamatory language. This service is properly monitored and any misuse is prohibited

Message Code

Email Address

Cell Number

Message Subject

How Urgent is it? High ☐ Normal ☐

Your Report



Send Report

Figure 5.25: Sending a report.

(b) Implementation: Sending of reports is well-handled by *DwesaReport* and *DwesaReportAction* classes. The *DwesaReport* class object defines the properties of a valid report received from the Dwesa community village. These properties include the sender and the recipient. The *addReport()* method of *DwesaReportAction* class accepts the report object as a parameter and send the report to the relevant authorities as defined by the recipient property of the report object. The recipient is determined by the routing function that will be discussed in the sections that follow.

The *sendbySMS()* method of the *DwesaTransmissionAction* class accepts the report object as a parameter and sends the message as an SMS to the relevant authorities. This method is only used if the report is an urgent one. The method uses an SMS Gateway defined in our system configuration. For simulation purposes, we used the SMS4U gateway to send SMSs. The *sendbyEmail()* method of the same class sends the report using an SMTP server. This method requires the email address of the government agency. The other method, *sendbyDwesaSys()*, sends the report to the back-end interface of the DORC system so that the agency sees the report directly from our servers. The code snippet in Figure 5.26 shows how sending of reports was implemented.

The sending and receiving of reports was implemented on the same server. Thus, the users and the government municipality use the same server to communicate. The DORC has a front-end portal specifically designed for users only. The portal has a back-end interface accessible to municipal officials only. Users receive their responses via the DORC Portal, SMSs, and emails. The piece of code in Figure 5.2.3 shows the implementation of report sending and receiving.

```

287      //get the resport details
288      $report = new DwesaReport();
289      $report->setMessage($post['reportmessage']);
290      $report->setTimereported($dateposted);
291      $report->setUser($sGlobals->user);
292      $report->setAnnounced(0);
293      $report->setCatid($post['complainttype']);
294      $report->setUrgency($post['radiobutton']);
295
296      //sending the report
297      $report_action = new DwesaReportAction();
298      $result = $report_action->sendDwesaReport($report);

```

Figure 5.26: Sending reports

The code for the routing function is listed in Appendix C. The mode of sending depends on the urgency of the report. If the report is very urgent, it becomes imperative for the report to be delivered immediately. This is sent by an SMS to the respective official. If the report is not critical, it is sent by email.

5.2.3 Validation of User Reports

Users should send reports that conform to our e-government system standards. This ensures that reports reach their destination efficiently. Validation is important so that the government agencies receive reports that contain relevant content.

(a) **Design:** The *DwesaValidator* class has a method called *validateReport()* that validates a report object passed to it as parameter. All reports are validated before they are submitted. The actual message of the report is validated so that it does not contain abusive content or invalid characters. The method ensures that urgent reports have a specified number of characters that are supported by SMS technology. When a user finishes typing a report, he should choose the level of urgency of the report. The e-government system's DORC portal determines the mode of response, depending on the level of urgency. The DORC system was designed to send urgent reports through all available channels of communication to ensure quick and guaranteed delivery. Responses are made through the DORC portal or telephonically.

(b) **Implementation:** The *validateReport()* method of *DwesaValidator* class accepts a report object and validates all the properties of the report. If all properties are valid then the report is allowed to be sent otherwise the sender is notified of any invalid data provided.

```
268 //validating the dwesa report
269 $r_validator = new DwesaValidator();
270 $result = $r_validator->validateReport($report);
```

Figure 5.27: Validating a report.

5.2.4 Routing User Reports

The DOAC is a portal that allows villagers to launch their grievances to the government. Our system assumes the reporter has no contacts to the government and therefore when a user submits a report, the system determines the recipient of that report. Reports are sent to various government departments.

(a) **Design:** To route reports we designed a mask of keywords that each government department specializes with. The table below shows part of our knowledge database we built to route the reports.

Receiver	Keywords in Report Message and Title
----------	--------------------------------------

Police	Death, killed, dead, murder, rape, steal, stolen, stealing, noise, destroy, beat, abuse,
Municipality	Evicted, marriage, divorce
Ministry of Education	Teachers, school, examination, fees, books
Administration	Test, Testing, Password

(a) Implementation: The *findReceiver()* method (listed in Appendix C) of *DwesaReportAction* class uses regular expressions to match our keywords that determine the recipient by the words contained in a user's report. If a pattern of words is found, the system attempts to forward the report to the relevant authorities associated with the keywords. The recipient receives this message on their report board and emails. In case the rules match several departments, the administrator will manually elect the receiver of the report.

5.2.5 Receiving of Reports

The government agencies receive reports from the users via emails, SMSs, web-enabled phones and the Dwesa back-end interface.

(a) Design: Reports are shown live as they come in on the back-end of the Dwesa e-government portal. The agent looking at the reports sees the reports as they arrive.

Figure 5.28 shows a report received at the back-end of the Municipality offices.



Figure 5.28: Live Reports from Dwesa Village.

(b) Implementation: Reports are displayed as they come in from the Dwesa village. The government agent can simply watch the reports board because new reports are automatically displayed without a page refresh AJAX.

5.2.6 Government Agency Responds

When community members send reports to the government agencies, they also wait for responses. This is important to allow complete interaction between community members and government.

(a) Design: A government agency can respond to reports through the DORC portal. The interface shown in Figure 5.2.6 allows the agency to respond to an incoming report. An agency can also contact the reporter through other traditional communication channels such as through emails, SMS, or telephoning.

Welcome to Dwesa E-Government Municipality Receiving Area.

[GENERAL INFO](#) [LIVE REPORTS](#) [RESPOND REPORTS](#) [CONTACT INFO](#)

LIVE REPORTS FROM DWESA

Number of Reports: 1

Sent To: Police

The urgency is normal

Message : There is a dead body at Ngwane bridge [Boyd : Jakachira (bjakachira@gmail.com)]

Sent on : 02-Oct-2009; 04:39PM **RESPONSE:** pending

[Denounce](#) | [Delete](#) | [Respond](#) | [View Responses](#)

Please type a short respond below for this report. Long responses will be not send. Illegal responses will be deleted. Thank you.

Respond

Figure 5.29: Responding to a report.

(b) Implementation: Replies are sent to reporters in the same way that reports are sent to the government agencies. Thus, a response is treated in the same way as a report. The `sendDwesaReport()` method also accepts a response as a parameter and uses either `sendbySMS()` or `sendbyDwesaSys()` methods of `DwesaTransmissionAction` class to send the response to the reporter. The piece of code in Figure 5.29 depicts how the response is sent.

```

221 case 11: //send a response..
222     $r_action = new DwesaReportAction();
223     $id = $inPath[2];
224     $response = new DwesaReport();
225     $response->setReportID($id);
226     $response->setMessage($post['txtmsg']);
227     $response->setUser($sGlobals->user);
228     $response->setTimereported(date("d-M-Y; h:iA"));
229     $result = $r_action->sendDwesaReport("", $response);
230     break;

```

Figure 5.30: Responding to a report.

5.2.7 DORC Security Features

It is important that the government agencies receive correct reports. The DORC portal provides a validation mechanism that ensures all reports that are sent out are accurate. Reports that contain inaccurate information are deleted from the system. All authenticated reports bear the sender and receiver's contact details to avoid repudiation of reports.

(a) Design: Users who do not have a Security Message Code (SMC) cannot send reports. The SMC is provided during registration on the e-government system. Because a user's login details are stored in the session until the user logs off, some users may leave the web page without signing out. Any person who next uses the computer can then pretend to be the original user and send reports. The SMC code is not stored in the session, thus the user will have to provide it whenever the system requires it. Users need to enter a valid email address. Those who do not have one can enter their cell number. Users are notified accordingly if any of these are invalid. The DORC portal has a moderating process that was implemented to deny sending of reports that contain malicious code, or abusive and rude language. It verifies the content of each report.

(b) Implementation: All reports are handled by *validateReport()* method of *DwesaValidator* class to screen all illegal reports. The method uses four metrics to screen the reports. The report should have a valid SMC code, and then reports are screened according to the level of urgency. This allows the timely broadcast of very urgent reports. All urgent reports are relayed by the SMS delivery method if it is activated in the system. If it is not activated, they are sent using emails and are broadcast live on the Reports Receiving Interface.

5.2.8 Latest Reports.

All users who visit the ORC Centre are able to see the latest reported matters. This avoids duplications of reports. The latest report is also shown on the front page and for a certain period of time, scrolls across the top of the home page as an announcement. This, however, requires the administrator to announce such a report to the global front page.

Welcome to Dwesa Online Reporting Center (DORC).

[GENERAL INFO](#) | [SEND A REPORT](#) | [SMC](#) | [VIEW REPORTS](#) | [CONTACTS](#)

Number of Reports: 2

Sent To: Police

The urgency is normal

Message : There is a man next door who beats his child evryday. We... [Boyd : Jakachira (bjakachira@gmail.com)]

Sent on : 11-Dec-2009; 04:29PM RESPONSE: pending

[Delete](#) | [Respond](#) | [View Responses](#)

Sent To: Police

The urgency is normal

Message : There is a dead body at Ngwane bridge [Boyd : Jakachira (bjakachira@gmail.com)]

Sent on : 02-Oct-2009; 04:39PM RESPONSE: pending

[Delete](#) | [Respond](#) | [View Responses](#)

Page:: 1

Figure 5.31: Viewing latest reports on the DORC.

5.3 The Dwesa Forum Corner (DFC)

The Dwesa Forum Corner is a discussion forum that serves to announce notices and convey alerts in the village. We assume that it will help to alleviate communication problems in the Dwesa community. The announcements can be posted by any registered user and if approved by the administrator, the announcement is broadcast. This functionality is still in its early stages and we have implemented it using emails as well as system postings as a mode of communication. We are, however, working to put in an SMS communication mode to speed up the delivery of announcements.

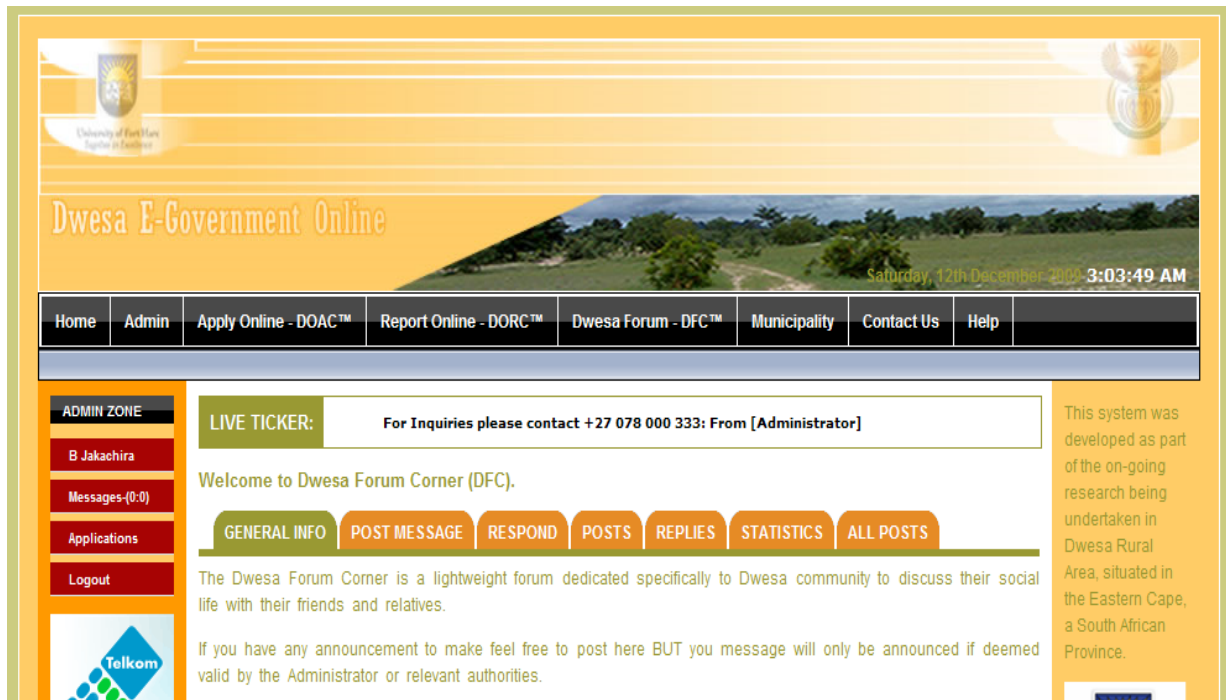


Figure 5.32: The DFC Portal.

5.3.1 Public Announcements

The Dwesa Forum Corner portal allows the villagers to post public announcements. The user is allowed to post an announcement but it must be approved by the administrator. Figure 5.32 shows the interface which the user can use to post announcements and notices.

5.3.2 Public Notice Board

The Dwesa Forum Corner (DFC) portal provides a public notice board where notices are posted. These may be announcements, alerts or village notices.

Every user who visits the Dwesa Forum Corner is able to read the Public Notice Board. The user is also capable of posting on the notice board by selecting the public notice radio button. The public notice board is monitored by a moderator.

5.4 Conclusion

The Dwesa e-government system has front-end and back-end interfaces. The designed front-end interfaces are the Online Application Centre (DOAC), the Dwesa Online Reporting Centre (DORC) and Dwesa Forum Corner (DFC). This chapter has described the implementation and

design of these three portals, together with their features. The portals together form e-government functionality, implemented as described and explained in this chapter.

Chapter Six

Evaluations, Results and Conclusions

Chapter six gives a conclusion to the research undertaken. The chapter will discuss and explain the evaluations and the results of the research. The chapter will finally give recommendations for possible future work.

6 Introduction

In this chapter, we conclude the project with an explanation of the evaluations of the Dwesa e-government portal. Two evaluation methods were combined to measure the outcome of this project. They will be described in this chapter. This will involve an examination of the questionnaire conducted at the end of the pilot use of the portal. The chapter will narrate the constraints and challenges encountered during the development and deployment of the Dwesa e-government portal in the field. It will then weigh up the project using an external evaluation method (Golubeva *et al.* 2005). The chapter will then make recommendations on possible future work on the Dwesa e-government strategy.

6.1 Summary of Findings

The findings suggest that the investment in e-government functionality was justified as the rural communities are likely to embrace it as a solution to get closer to their government. This has result in the development of an e-government system applicable to the marginalized communities.

The findings show that it is possible to create a single view of government services and make it available to remote parts of the country. Furthermore, the e-government portal was designed to provide just the kind of information the community members deemed was vital. However, answers from the questionnaire show that rural communities are also skeptical whether the government will attend to them through an e-government portal. This shows that public awareness and program changes have not done a lot to educate rural people on the principles of providing public services online.

Results show that the project relied mainly on infrastructure existing within our bigger ongoing project in Dwesa. The result of deploying the e-government portal in Dwesa showed that community members are beginning to realize that the government benefits as well from the e-government solutions. Government can lead to direct democracy through supporting online debates and discussions at Dwesa Forum Corner (DFC) portal.

The deployment of the e-government portal also shows that major areas of e-government strategies that can apply in poor and remote parts of the country such as Dwesa are mainly Government to Citizen (G2C). This is because few community members have needs that

require a Government-to-Business (G2B) communication channel. The polls conducted pointed out that 80% of users require communication with the government. In this case, the designed system took the initiative to promote G2C communication links in the rural community.

6.2 Evaluation of the Dwesa E-Government System

The fact that the development of the Dwesa e-government system went through various phases of growth, means the portal is still not fully evolved. Any further meaningful evaluation of the Dwesa e-government portal will be done as more user feedback is received, once users are fully acquainted with its use. Having seen meaningful evaluations of Palkka.fi portal (Selkainaho 2006) due to the method they applied to evaluate the success of their portal, we decided to use a similar method in our evaluation. This was done to get a more objective evaluation of the e-government portal.

This project realized the importance of user consultation, ensuring user participation in the process of e-government implementation. Table 6.1 shows some of the visits made to the Dwesa field in order to interact with the users. Informal interviews were carried out on Trip A to determine the public services vital to the Dwesa community members.

Trips B and C were aimed at giving the Dwesa community members some tutorials on how to use computers as well as the benefits of using computers. This would be done in the four computer labs situated at four different schools. The schools are Ngwane, Mntokwane, Nqabara and Mpume Junior Secondary Schools. Because this research project is a part of a bigger on-going ICT project in Dwesa, other visits which were not entirely aligned to this study were made, promoting public awareness by giving training on how to use the computers.

Trip Type	Trip Dates	Trip Activities
A	31 March – 6 April 2007	Introductions and Informal Interviews with the Dwesa community members, teachers at Mpume and Ngwane. Notes of the public services were taken for implementation.

B	19 August – 24 August 2007 7 October – 12 October 2007	Implementing the system. Giving tutorials to the Dwesa Community members at four different schools, Ngwane, Mpume, Mntokwane and Nqabara using our Siyakhula Living Computer Labs.
C	30 November - 5 December 2008	Test of the first prototype of the Dwesa E-government portal and administering the Evaluation Questionnaire. To test for DOAC and DORC portal we simulated a government server that could accepted users' applications files.

Table 6.1: Dwesa Field Trips and Activities.

The aim of the Trip C was to deploy the e-government portal prototype and then do some field evaluations. We administered a short questionnaire at the end of the demonstrations of the portal on the 2nd of December 2008 at Ngwane. We got 22 Dwesa community members to participate in the demonstrations and the questionnaire. The members varied from school pupils to adults. Our questionnaire had 19 questions. The first ten questions were asked to get background information and factual type of questions. Answers were received on what users expect to get from the government as well as their general view of the demonstrated e-government portal. The system-evaluating questions that were asked at this phase tried to determine the usability of the Dwesa e-government system.

More feedback about the e-government system will be obtained gradually as community members get accustomed to the interfaces. The e-government portal has a discussion forum, DFC, which allows users to discuss and post their subjective views and opinions about the system. Further improvements to the current e-government portal will be done basing on the users' feedback on the DFC portal. Thus questionnaire questions which could have been meaningful to ask were left out and the effort was put on basic matters the participants would understand. The system-evaluating questions, with options to choose from, that were raised at this phase would seek the functionality, usability accessibility of the Dwesa e-government system. We felt that to get more meaningful evaluations, we could apply an objective evaluation method together with the questionnaire.

Observing the behaviour of the Dwesa users during the demonstration sessions, while they tried to fulfill different tasks, provided us with best opportunity to see whether the system was user-

friendly or not for the same users. We noted that we have managed to create a common, generally effective, look and feel for citizen interaction with government.

An external evaluation method was used to evaluate the current more refined e-government system. In their paper, Golubeva and Merkryeva (2005) used external and internal evaluation methods to evaluate the development of e-government in their area. The method evaluated websites designed for e-government purposes in a way that we felt was exhaustive and could draw accurate measurements of their system. In this context, external evaluation is an evaluation of the integrated e-government system carried out by the questionnaire respondents, who are Dwesa community members. Thus, the external and objective evaluation methods yield similar results to the internal and subjective evaluation methods. Upon seeing insightful evaluations attained on their project using this method, we decided we could apply this method to the evaluation of the Dwesa e-government system. The objective evaluation would be performed in conjunction with the subjective questionnaire conducted at Dwesa. The deployment of the e-government pilot system was achieved through numerous visits to Dwesa village.

6.3 Dwesa E-Government System Evaluation Criteria

The evaluation method used was centred on the main features of the three functionalities designed in the Dwesa e-government portal. Evaluation was also carried out on the accessibility and usability of the portal. Each of the functionalities will have features defined to provide some measurement of user-orientation, coverage, information currency and accuracy and accessibility.

6.3.1 Functionality Evaluation

The e-government questionnaire (given in Appendix D) that was administered in the Dwesa community produced results that can back the external evaluation's findings. We got 22 Dwesa rural community members to respond to the questionnaire. Questions were asked in the questionnaire in a way we deemed users would understand. We used simple terms, for instance in question 8, the term "the computer" refers to the e-government system. These questions were asked after and while we demonstrated the three functionalities of the e-government portal. The following questions are some of those that sought user evaluation of the functionality of the system.

Question 8. Do you think you can apply for an ID or Birth Certificate on the computer?

- A. Agree.
- B. Strongly Agree.
- C. Not Sure.
- D. Disagree.
- E. Strongly Disagree.

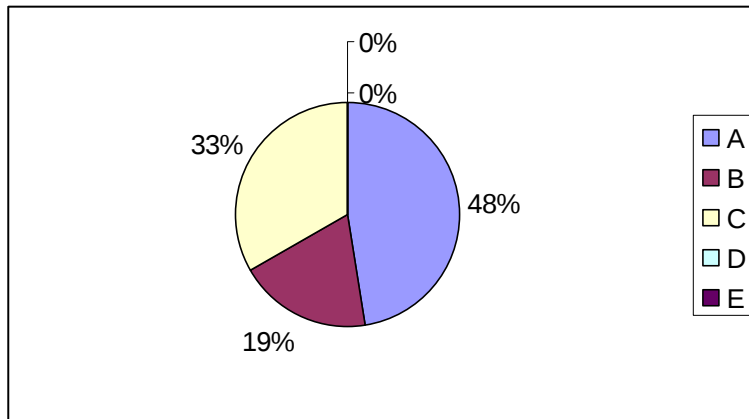


Figure 6.1 Participants' responses for question 8.

67% of the participants showed that they are confident that they can apply for important government documents on the DOAC portal of the e-government system. The 19% of participants who strongly agreed with this shows that they could have been convinced about the benefits of obtaining public services online. 33% were not sure whether they could successfully apply an for ID or any other government document online. The explanation for this is that some of the community members may still have to participate in our public awareness program at the time.

Question 18. Do you think you can use the computer to report anything to the government?

- A. Agree
- B. Strongly agree
- C. Not sure
- D. Disagree
- E. Strongly Agree

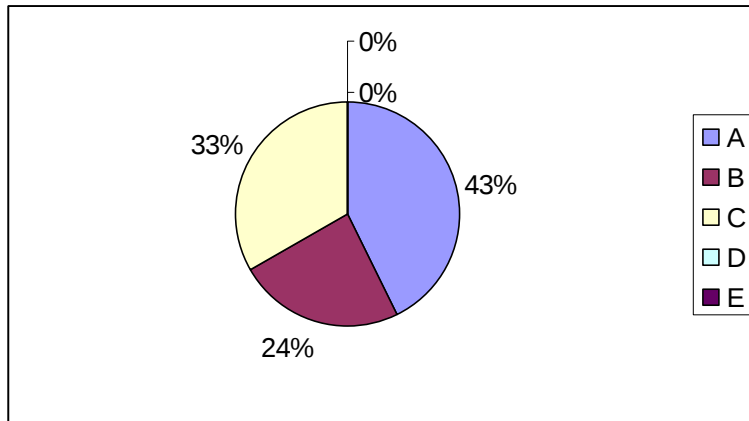


Figure 6.2. Participants' responses for question 18.

As depicted in the chart in Figure 6.2, 43% of the participants simply agreed and 33% strongly believed that this will create a communication link between them and the government. This shows that the Dwesa Online Report Centre (DORC) part of the e-government portal has covered much ground in achieving its goal to allow Dwesa community members to report any grievances to the relevant government department.

The functionality of the Dwesa Online Application Centre portal was evaluated in the questionnaire by the following questions:

Question 19. Do you think the government will be happy to receive your passport, ID or Birth Certificate application form from your computer?

- A. Yes
- B. No
- C. I don't know

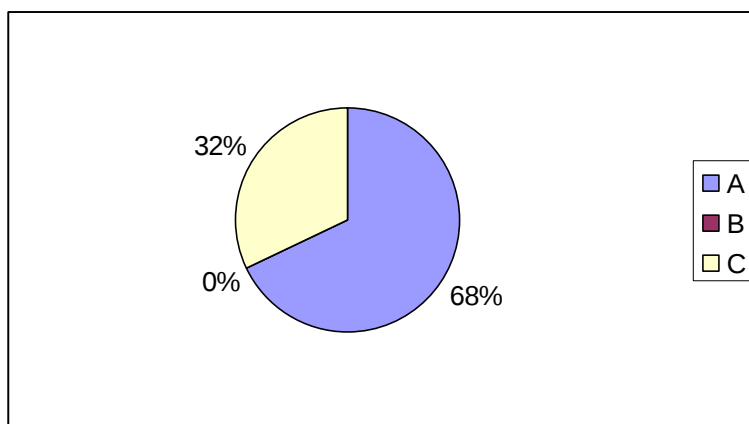


Figure 6.3. Participants' responses for question 19.

As the pie chart above indicates, 68% of the participants agreed that they had confidence in working with their government departments that issue important documents and permits through the DOAC portal, which is part of the e-government portal. Table 6.2 examines further the results using the external evaluation method.

Feature	Criteria	Evaluated Options
Functionality	Users Orientation	Contact information, Physical location, Agency description.
	Coverage	Description of services Quality of databases Downloadable forms News Feedback from officials
	Information Currency and Accuracy	Updates on information and data Relevancy and consistency of content. Pages responsibility Grammatical and spelling errors Number of pages under construction.
	Interactivity	Performance of email and subscription service. Live effects. Completeness of transactions (form and reports submissions).

Table 6.2 Functionality Evaluated options.

The portal managed to provide necessary contact information for the government agencies on its three functionalities through its global contact form, thereby creating a digital community. Because the e-government system provides equitable, universal and affordable access to the Dwesa rural community, an increased participation of the village members on the DFC portal builds a strong digital community.

To be specific, the evaluation of the Dwesa Forum Corner (DFC) and Dwesa Online Reporting Centre (DORC) showed that administrative bottlenecks would have been reduced, thereby achieving one of our objectives, namely, to lessen the turn-around time in the provision of most government services. Users showed interest in launching reports from the comfort of their community computer lab. These will be launched on the DORC portal, and community members are only required to provide the report and our system will route it to the relevant

government agencies. Thus, we achieved one of our goals that of implementing an e-government system that will require less input from users but deliver more.

The Dwesa e-government portal managed to present all downloadable content in a very efficient and secure way. When we simulated the process of applying for a document, a simulated government server could receive users forms digitally signed using encryption technology. The downloadable content includes forms and application files. Uploading to the portal was also handled efficiently, as is supported by the questionnaires filled in by users who attended the demonstration. The portal utilizes SMS technology to deliver urgent reports. During simulation of the process, we noted that our e-government portal could intelligently route reports to relevant government authorities. This suggested that villagers who do not know where to send reports or complaints could just launch their grievances on the DORC portal. It has a live ticker banner that can be used to provide announcements through Dwesa Online Reporting Centre (DORC) and Dwesa Forum Corner (DFC). This ensures that information is delivered quickly.

Evaluation of the e-government portal done after the simulation of the processes showed that the accuracy of the information and data is in line with the information the government is giving out. Evaluating the e-government portal, we have seen that it has its web pages organized between the three functionalities. Thus, the three functionalities act as independent modules integrated on one portal, making it possible to adjacently incorporate further services. This will then foster a single view of the government services.

The Dwesa e-government portal, successfully implemented, provides live effects because it runs the Ajax technologies. For instance, on the DORC portal, reports are sent and received live. At the DOAC portal, sending of applications is live, which means users do not bear the refreshing of the web-page every time they try to send or download a form. The DOAC portal also has a live form update process that, if completed, will run on the server on daily basis updating the list of government forms residing in our system database.

6.3.2 Accessibility Evaluation

The evaluation showed that the accessibility of the Dwesa e-government portal at Dwesa computer labs is not aligned to specific types of users. The portal and its content are available to a wide range of rural users with varying level of skills and technology. This is supported by

the number and type of users who participated in the demonstration of the e-government functionalities at Siyakhula computer labs in Dwesa. The community members could access the client machines in the labs without any problems, as shown by the answers given by the 22 questionnaire respondents to Question 17, below:

Question 17. Did you use a school computer before?

- A. Yes
- B. No
- C. I will never use it
- D. We are not allowed

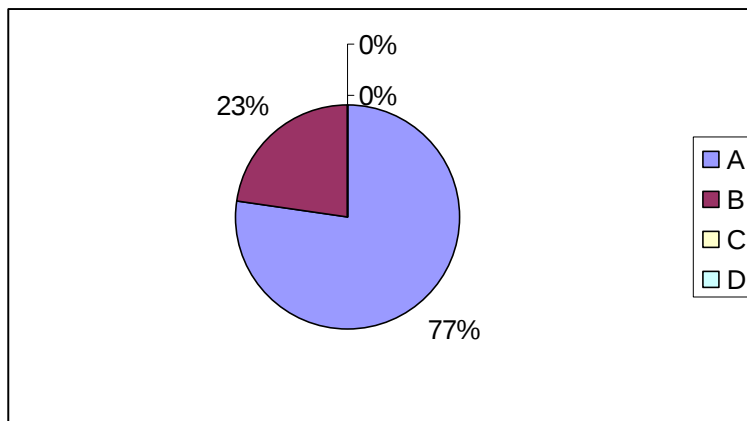


Figure 6.4. Participants' responses

The chart shows that 77% of participants, who included community members, have used a computer before and this suggests that the community has access to the computer lab deployed at this school. This in turn means that the developed e-government system has a solid foundation to take it to the next e-government development phase.

Table 6.3 depicts evaluated options on each features' criteria.

Feature	Criteria	Evaluated Options
Accessibility	Access to site	Use of symbols. Consistency of format. Layout (page lengths, organization)
	Access to site content	Relevancy of provided links. Absence of dead links. Descriptive comments. Frequent user options.

Table 6.3: Accessibility Evaluated Options.

Assessing the accessibility to the content, we found out that our system provides enough instruction on how to use each of the three functionalities. A Helpdesk system is also currently being developed to provide around-the-clock assistance to visitors of the e-government portal. The presence of multiple languages on an e-government portal provides more accessibility to the content. Our e-government project waits to be localized culturally and linguistically. Thus, in this case the project proved to have foresight on upcoming possible extensions to the current e-government portal. The current system is in English.

The evaluation found that the project had consistently tried to adopt best practice, and to take full cognizance of the needs of the Dwesa villagers in providing access to government information online. The evaluation concluded that this project was successful in providing a platform to carry e-government awareness to the next level.

Looking at the case study of the Cameroon Government Web Portal, the portal suffered a lack of access to the internet in rural areas. Our project ran the same risk, but at its initial implementation, the project boasts four public computer labs that are accessible to community members. These four labs are part of the Siyakhula Living Lab infrastructure, which provides the portal with necessary ICT.

6.3.3 Usability Evaluation

The usability of the e-government portal can be hindered by several factors. The page length can hide valuable information from the users. Evaluations of our e-government portal shows only three web pages have a length big enough to fall outside the standard client machine screen. All other web pages present the information without the need to scroll down and up to get what is wanted.

Question 20. After you used the demonstrated system, did you better understand how to use the interfaces and would you be able to work on your own without any assistance?

- A. I agree
- B. Strongly agree
- C. I disagree
- D. Strongly disagree

E. Not sure

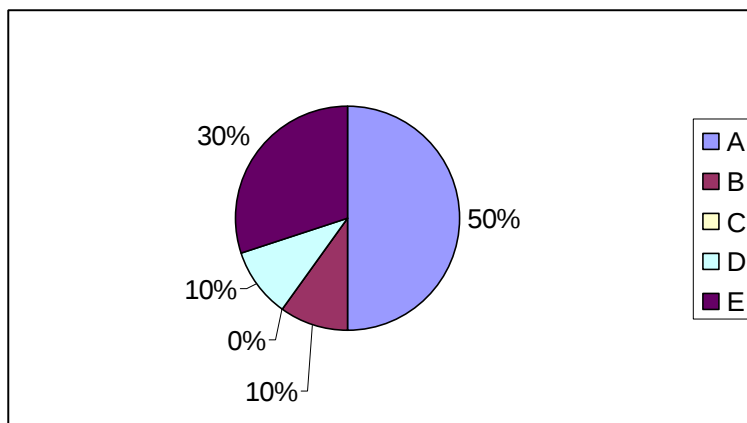


Figure 6.5. Participants' responses for question 20.

Question 20 of the questionnaire evaluated whether community members had understood the interfaces of the e-government portal. As shown on the pie chart above, a total of 60% of the participants, of which 10% of them strongly agreed, felt that they had understood the usability of the e-government portal.

The evaluation of the functionality, accessibility and usability of the e-government explained above shows that we managed to cover much ground in our objectives. Thus the basic functionalities were provided at lower cost and users showed some approval of this technology-based solution to the provision of government services online.

Table 6.4 depicts evaluated options on each features' criteria.

Feature	Criteria	Evaluated Options
Usability	Architecture, design and layout	Readability Use of symbols Consistency Advanced display features
	Links	Absence of dead links. Descriptive comments. Relevancy of provided.
	Navigability	Local search engines Site map
	Metadata	Descriptive page titles

Table 6.4: Usability Evaluated Options.

6.3.4 Other Success Factors or Enablers

This project received support from the highest political level. The Deputy Minister of Science and Technology, Derek Hanekom made a visit to the Dwesa project in September 2008. This support was required to lobby for some approval and recognition from the government side. The presentation of this project was held in the presence of the deputy minister to highlight the aim and objectives of this project. That was done to show the minister that such an e-government portal will positively improve the relationship between the government and its citizens. The deputy minister pledged to assist the bigger on-going project, to which this e-government project belongs, by mentioning its existence in his meetings with the Department of Science and Technology.

Several individuals at the SATNAC conference provided support for this initiative. Top-level expertise in the ICT field at this conference recommended vertical integration of this project to existing e-government strategies. This project was seen as a major step towards bringing the rural people and their government into interacting closely.

The project also received much support from a very competent hybrid team. The team was made up of students researching other projects aligned to Dwesa.

As this project research is a part of a bigger project going on in the Dwesa rural community, the e-government portal benefited from prior formal structures already deployed at the start of the bigger project. The project has made sound progress relatively quickly because it has been able to utilize the existing ICT network infrastructure deployed across four secondary schools in the rural community. This ensures that prototypes would be deployed for evaluation at the most convenient time.

6.4 Conclusion

From the observations made, we concluded that e-government systems will provide convenient government-to-citizen (G2C) communication between the rural citizens and the government. Rural citizens will benefit from the e-government initiatives more than their urban counterparts do. Factors that influence the choice of the e-government access models used to build e-government systems in rural areas differ from those in urban areas. The software that is used to

build e-government systems differs as we move away from the cities. This is because of differences in users' cultural beliefs, GDP contributions as well as educational status.

6.4.1 Summary of Contributions

An integrated e-government system capable of delivering public services efficiently to the Dwesa rural area has been developed. The success in creating an integrated e-government system has helped to answer the research questions posed at the beginning.

How can a web-based e-government system for a marginalized community be developed? The development of the Dwesa e-government system has contributed to the understanding of software tools and methodology that can be used to deliver public services cost-effectively to the disadvantaged communities. The study has shown how web systems can be developed to solve access to government services problems.

Can public services be provided to the disadvantaged community through the internet? The provision of public services through the three integrated portals of the Dwesa e-government has shown that the internet can be used effectively to promote government-to-citizen services delivery.

Why is a web-based e-government system needed in Dwesa? The initiative has contributed by expanding the knowledge in the study of delivery models applicable when government is providing public services to its remote communities. The study has successfully shown that a web-based e-government system can successfully bring government closer to its citizens' needs.

What can be done to combat the lack of public services access in the Dwesa community? The study ended by deploying a cost-effective e-government system in Dwesa that has contributed to a significant improvement of public services access in Dwesa.

Following evaluations done, we conclude that the e-government portal provides a solid foundation to take to another level of e-government implementation. The project's aim was to start a pilot system that will be incorporated into the existing e-government strategies or further implemented to make an e-government system that fully covers the needs of the Dwesa rural community.

The project, as part of major on-going research in Dwesa, has provided solid public awareness of the possibility of interacting with their government departments. It will remain the principal mechanism needed in the Dwesa community to access vital public services.

6.4.2 Suggestions for Further Research

To make this project even more successful in other public domains, a champion in the government should be found. Correct information on e-government should be heard by the champion, who in this case may be a person who holds a high rank in government (Levin 2003).

Promoting public/private partnerships will give rise to e-government systems that are vertically integrated, linking all possible government services in one place. This will create a single view of government. A focal point where the citizens, mainly from rural areas, link with their government will then be possible (Olivier 2002).

Awareness and change programs that are being run at Dwesa can be enhanced to make the final e-government portal more successful (Levin 2003).

6.4.3 Discussion of Constraints and Challenges

The outcome of this project did not come smoothly. The project survived some difficult strategic implementation. The challenges this project faced in the development did not deter the outcome of this study. This was due to determination of the project hybrid team and the approval of an e-government portal for the Dwesa community members.

6.4.3.1 Resistance

The development of an e-government system went without any glitch. However, promoting public awareness of the system proved to be a big challenge. Dwesa rural villagers could not easily embrace the idea. The system incurred some form of resistance from users who were skeptical whether the government would acknowledge the content emanating from the portal. The project team could have invested more of its time advancing public awareness about the e-government functionalities and other related projects.

The project also struggled to get support from the government side. The support sought could have been a public approval of such a strategy by a top government official. After long and cumbersome lobbying to the government, the team hosted the deputy minister of Science and Technology in Dwesa where this project was one of several projects that were presented. This, however, occurred towards the end of the project and was thus not much help.

6.4.3.2 Implementation Gaps

The main implementation gap found during the development of the system was that most of the downloadable documents are non-interactive. Much effort will be needed to convert these forms so that users are able to fill them in online.

REFERENCES

- Aptivate Organisation. *Web Design Guidelines for low Bandwidth*. Available at <http://www.aprtivate.org/webguidelines/Scripting.html>. [Last accessed: 12/12/2008].
- Balacheff, B., Chen, L., Plaquin, D., and Proudler, G. 2001. *A trusted process to digitally sign a document*. In Proceedings of the 2001 Workshop on New Security Paradigms. Cloudcroft, New Mexico, Public-Key Cryptography, Available at: <http://docs.sun.com/source/816-6154-10/contents.htm>. Last Updated: September 10 - 13, 2001. NSPW '01. ACM, New York, NY, 79-86. DOI= <http://doi.acm.org/10.1145/508171.508184>.
- Boehm, B. W., Penedo, M. H., Stuckle, E. D., Williams, R. D., and Pyster, A. B. 1984. A Software Development Environment for Improving Productivity. *Computer* 17, 6 (Jun. 1984): 30-44. DOI= <http://dx.doi.org/10.1109/MC.1984.1659160>.
- Bruschi, D., Fabris, D, Glave, V. & Rosti, E. 2003. How to unwittingly sign non-repudiable documents with Java applications. In Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC '03), pp. 192-196, IEEE Computer Society.
- Buckler, C. 2009 - MySQL: the Pros and Cons of InnoDB tables. Available: <http://www.sitepoint.com/blogs/2009/03/26/mysql-innodb-table-pros-cons/>. Last accessed: 26 October 2009.
- Caldow, J. 1999. *The Quest for Electronic Government: A Defining Vision*. Institute of Electronic Government, IBM Corporation.
- CEG (Centre of Electronic Governance). 2000. *An Evaluation of Gyandoot E-Government (G2C) project*. Indian Institute of Management, Ahmedabad, India. CEG, India. pp. 3-6
- CPSI. 2003. A study conducted for the Centre for Public Service Innovation, *Citizen Access to E-government Services*, Department of Public Service and Administration, WITS Link Centre and Sangonet.

- Cook, M. E. 2000. What Citizens Want from E-Government. Center for Technology in Government, University of Albany, SUNY, Email Address: info@ctg.albany.edu. Available at: <http://netcaucus.org/books/egov2001/pdf/citizen.pdf>. Last accessed 13-Nov-09.
- Dalvit, L., Terzoli, A., Muyingi, H., Thinyane, M., 2007 - The deployment of an e-commerce platform and related projects in a rural area in South Africa, *International Journal of Computing and ICT Research*, 1(1) 2007: 9-17.
- Darie, C. and Brinzarea, B. 2006. *AJAX and PHP: Building Responsive Web Applications*. Packt Publishing, 32 Lincoln Road, Olton, Birmingham, B27 6PA, UK. ISBN 1-904811-82-5.
- Digre, T. 1998. Business Object Component Architecture, *Data Access Technol*, USA, October 1998.
- Ezz, I.E.A. 2006. *Inter-Organisational collaboration towards Process Integration in Public Sector*. E-Government Collaboration in Egypt. HICSS.
- Fang, Z. 2002. *E-government in Digital Era: Concept, Practice, and Development*. School of Public Administration (NIDA), Thailand.
- Fourie, L. 2007. *Enhancing the Livelihoods of rural poor: The role of information and communication technologies*, 2007.
- Furber, P. 2007. Municipality lag in e-delivery. Available: <http://ww2.itweb.co.za/events/egovernment>. Last Accessed: 26 October 2009.
- Glass, M., Scouarnec Yann, L., Naramore, E., Mailer, G., Stolz, J., Gerner, J. 2004. *Beginning PHP, Apache, MySQL Web development*. Wiley Publishing. ISBN: 0-7645-5744-0.
- Golubeva, A., Merkuryeva, I., and Shulakov, N. 2005. *Development of E-Government in St.Petersburg: Evaluation of websites performance and usability*. Public Administration Department, School of Management, St. Petersburg State University, St.Petersburg, Russia,

2005. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.7119&rep=rep1&type=pdf>. Last Accessed: 13 November 2009.
- Gulutzan, P. 2005. *Why Stored Procedures?* MySQL 5.0 Stored Procedures, New Features Series – Part 1. A MySQL Technical Paper, Page 4.
- Gupta, M.P. and Jana, D. 2001. *E-government Evaluation: a framework and case study*. Stages of e-government. Indian Institute of Technology, Department of Management Studies, Delhi, 110016, India.
- Heeks, R. 2006. *Implementing and Managing E-government: An International Text*. London, Thousand Oaks, Calif.: Sage. ISBN: 0761967923. Online: <http://worldcat.org/wcpa/isbn/0761967923>
- Heeks, R. & Arun, S. 2007. IT Social Outsourcing as a Development Tool: IT Outsourcing to Social Enterprises for Poverty Reduction and Women's Empowerment in Keral. Development Informatics Group, IDPM, University of Manchester, 2007, pg. 2.
- Heeks, R. 2008. *eGovernment for Development Information Exchange*, Institute for Development Policy and Management, University of Manchester, United Kingdom, Last Updated: 19 October 2008.
- Holzner, S 2006. *Ajax for Dummies*. For Dummies. Computer/Tech.
- Howcroft, D & Carroll, J 2000, A Proposed Methodology for Web Development. IS Research Centre, University of Salford, Salford.
- Griessal, A 2006, Kwa-Thema Multi-Purpose Community Centre. Issued by Office of the Premier, Gauteng Provincial Government, and South Africa. Available at: <http://www.info.gov.za/speeches/2006/06072811151001.htm>. Last Accessed: 18 November 2009.
- Jakachira, B.T., Muyingi, H., Wertlen, R. 2008. Implementation of an e-government proxy for a marginalized rural population, SATNAC, Department of Computer Science, University of Fort Hare, Alice, South Africa.

- Jukic, T. & Vintar, M. 2008. Ex-ante evaluation of e-government projects: which methods to use? Faculty of Administration's Research Centre, University of Ljubljana, Ljubljana, Slovenia.
- Kamar, N. 2007. Impact of e-Government on Management and use of Government Information in Kenya. Government Information and Official Publications.
- Kamar, N. and Ongo'ndo, M. 2007. Implementation of E-government systems – website creation. Egerton University, J.D. Rockefeller Research Library, Njoro, Kenya.
- Kenhango Tazo Olivier. 2002. *The Cameroon Government Web Portal, Success/Failure Case Study* No.9, Central Africa, August. Available at: <http://www.egov4dev.org/success/case/camportal.shtml> Last Accessed: 13 November 2009.
- Kettimuthu, R., Allcock, W., Limng, L., Navarro, JP. & Foster, I., 2007. *GridCopy: Moving Data Fast on the Grid, Mathematics and Computer Science Division*. The University of Chicago, Chicago, IL 60637.
- Layne, K. and Lee, J. 2001. *Developing fully functional E-government: A four stage model*. University of Nevada Las Vegas, NV 89154-60009, 2001.
- Levin, A and Dingley, R. 2003. Evaluation: Failure or Success, eTransparency Case Study No.1, Opening Government Information via the Cape Gateway. Case Editor: Richard Heeks. November 2003. Available at: www.egov4dev.org/transparency/case/cape.shtml . Last Accessed: 13-Nov-09.
- Macintosh, A. 2004. *Characterizing E-Participation in Policy-Making*. International Teledemocracy Centre, Napier University, Edinburgh, EH01 5DT, Scotland, UK.
- Marner, W. 1997. *Rapid Application Development*. Department of Computer Science, Bowling Green State University, Revised in 1997. Available at <http://csweb.cs.bgsu.edu/maner/domains/RAD.htm>.

Misra, D.C. 2006. Ten Emerging E-government Challenges Today, 4th International Conference on E-governance (ICEG DEC 2006), Keynote Session: E-gov Brass Tasks Independent E-government Consultant, New Delhi, India.

Misra, D.C. 2007. Defining E-government: A Citizen-centric Criteria-based Approach. Available at: <http://topics.developmentgateway.org> and <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.4809&rep=rep1&type=pdf> Last accessed: 13 November 2009.

MySQL Website. Available at: <http://www.mysql.com>, Last Accessed on 14 July 2009.

Navarra, D.D. & Ciborra, C. 2003. *Good Governance and Development Aid: Risks and Challenges of E-Government in Jordan*. Information Systems Perspectives and Challenges in the Context of Globalization, 2003: 387-402.

Necrafft. Available at: http://news.netcraft.com/archives/web_server_survey.html Last Accessed: December 2008.

Netbeans Org. Available at: <http://www.netbeans.org/features/php/index.html#> Last Accessed: December 2008.

Pankaj, K 1999, Apache at your Web Service, Features and Advantages of Apache, August 1999. Available at: <http://www.irt.org/articles/js177/index.htm>. Last Accessed: 14 January 2009.

PDF Filler Inc. Available at www.pdfFiller.com. Last Accessed: December 2008.

Petricek, V., Escher, T., Cox, I.J., Margetts, H. 2007. *The Web Structure of E-government – Developing a Methodology for Quantitative Evaluation*, UCL Computer Science, London.

Peters, R.M., Janssen, M., and van Engers, T.M. 2004. Measuring e-government impact: existing practices and shortcomings. In *Proceedings of the 6th international Conference on Electronic Commerce*. Delft, The Netherlands, October 25-27, 2004. M. Janssen, H.G. Sol,

- and R. W. Wagenaar, Eds. ICEC '04, vol. 60. ACM, New York, NY, 480-489. DOI=<http://doi.acm.org/10.1145/1052220.1052281>
- PHPframework Inc. 2009. ZooP. Available at: <http://www.phpframeworks.com/php-frameworks/index.php?id=7>. Last Accessed: 16 November 2009.
- Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P. and Zipf, T. 2001. CRUMPET: Creation of User-friendly mobile services personalized for tourism, Queen Mary, University of London, UK.
- Pujari, A.K. 2002. *Planning Web-Enabled Services for Citizens in Orrisa*. South Asia.
- Raffat, R.M. 2006. *Developing a Successful e-Government*. University of Sydney, NSW, Australia, Email Address: rabee@arch.usyd.edu.au. Available at: <http://publications.ksu.edu.sa/Conferences/eGovernment%20Conference/E16.pdf>. Last Accessed: 13 November 2009.
- Reddick, C. G. 2004 - "A two-stage model of e-government growth: Theories and empirical evidence for U.S. cities," *Government Information Quarterly* 21, no. 1 (2004): 51-64.
- Rivest, R. L., Shamir, A., & Adleman, L. 1978. *A method for obtaining digital signatures and public-key cryptosystems*. Commun. ACM 21, 2 (Feb. 1978), 120-126. DOI=<http://doi.acm.org/10.1145/359340.359342>. 1998.
- Selkainaho, R. 2006. *Evaluating E-government: The case of Palkka.fi portal*, Swedish School of Economics and Business Administration, A Case Study, 2006. Available at: <http://www.pafis.shh.fi/graduates/ritsel04.pdf> Last Accessed: 13 November 2009.
- Siau, K. and Long, Y. 2005. *Synthesizing e-government stage models: a meta-synthesis based on meta-ethnography approach*. Department of Management, College of Business Administration, University of Nebraska-Lincoln, USA.
- Slazinski, E.D. 2001. *The 'other' database object*. Department of Computer Technology, Purdue University. West Lafayette, ISBN 47907-1421, USA.

- Te Komihana O Nga Tari Kawanatanga. 2006. *Definition of E-Government*. New Zealand Interoperability Framework Document, version 3.1. State Services Commission. November.
- Vosloo, S. 2005. E-Government and the E-Readiness of Non-Profit Organisations in the Western Cape, South Africa. Provincial Government of the Western Cape and University of Cape Town, South Africa.
- Waterson, K. 2003. *The Model*. Model View Controller (MVC). Available at <http://www.phppro.org/tutorials/Model-View-Controller-MVC.html#6>
- Welling, L. and Thomson, L. 2004. *PHP and MySQL Web Development*. (3rd edition). Sams.
- Wijayawardhana, D. 2008. MySQL 5.1 Features: The Event Scheduler, MySQL Community Team.
- Wikibooks. 2009. The open-content textbooks collection. Available: <http://en.wikibooks.org/wiki/E-government/Definition>. Last Accessed: 12 September 2009.
- Wimmer, M.A. and Tamborius, E. 2002. Online One-Stop Government: A working framework and requirements. In proceedings of the IFIP World Computer Congress, August 26-39, 2002, Montreal.
- Withrow, J., Brinck, T. and Sperdelozzi, A. 2005 *Comparative Usability Evaluation for an e-Government Portal*. Diamond Bullet Design in Collaboration with the National Information Consortium. Dec 2005. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.1831&rep=rep1&type=pdf>. Last Accessed: 13 November 2009.
- Xavier, M.J. & Gupta M.P. 2003. *Government to citizen (G2C) service delivery: Models, Selected Case Studies and Lessons for Future Developments*. Institute for Financial Management and Research.
- Yiming, H., Nanda, H., Yang, Q. 1999. *Measurement, Analysis and Performance Improvement of the Apache Web Server*. Department of Electrical and Computer Engineering &

Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030. Available:
<http://pkp.sfu.ca/ojs/doxygen/html/main.html>

Zoop Inc. Year of Posting or update. *Zoop Framework*. Available from:
<http://zoopframework.com/>. Accessed: 29 November 2008.

KHANYA Org. KHANYA Annual Report Year of update: 2008 Available at
http://www.khanya.co.za/projectinfo/docs/khanya_annual_rep_2008.pdf Last Accessed: 26
March 2010.

Appendix A

System Installation and setup

To install the e-Government system on a computer the following system requirements are needed first.

- i. Apache web server.
- ii. MySQL database server (Any version above 5.1).
- iii. FileZilla.
- iv. PHP (Version above 5).

When all the requirements are set properly, the root directory of the e-government can be copied to the Apache web server virtual directory (either www or htdocs). The database can be installed by importing the dump file into the MySQL database server.

The *config.php* Configuration File.

The **config** file is located under the root directory. Normally you do not need to change anything in this file. However, when installing the e-government system on a different OS platform, valid paths to the PEAR libraries should be defined. The statements are shown in the file below. The system is supported on both UNIX and Windows.

```
31 ///////////////////////////////////////////////////////////////////
32 //                      PEAR Stuff                      //
33 ///////////////////////////////////////////////////////////////////
34 //  uncomment one line if using the zoop/libs instead of your systemwide pear libraries.
35
36 //  ini_set('include_path',ini_get('include_path').':'. zoop_dir . '/lib/pear:'); // FOR UNIX
37   ini_set('include_path',ini_get('include_path').':'. zoop_dir . '/lib/pear:'); // FOR WINDOWS
38
```

The *db.php* Configuration File.

The **db** file located in “root_directory/config/” has all the necessary configuration settings. The following settings are set in this file

- **Database settings:** Database Server name, database name, username and password are required for the system to operate.
- **Emailing settings:** The system can be configured to send emails or not, on these settings. It can also be configured to route all emails to the admin.
- **File Servers settings:** When a new government server is required to be set up on the system, its settings are set up on this configuration file.

- **System Functionalities settings:** Three portals are integrated to form the Dwesa e-government. An administrator can set various management settings for this configuration file.
- **Administration Settings:** Admin username and passwords are stored in this file.

Appendix B

System Classes and Procedures

The Dwesa e-government system is comprised of several sections. The screen shot below shows all the zones available to the system.

```
27  ////////////////////////////////////////////
28  //  include all zone subclasses here      //
29  ////////////////////////////////////////////
30
31  $zoop->addZone('default');
32  $zoop->addZone('admin');
33  $zoop->addZone('user');
34  $zoop->addZone('administration');
35  $zoop->addZone('municipality');
36  $zoop->addZone('doac');//Dwesa Online Application Center
37  $zoop->addZone('dfc');// Dwesa Forum Corner
38  $zoop->addZone('dorc');//Dwesa Online Reporting Center
39
```

Other Dwesa E-government System Classes.

The image shown below gives a list of some of the classes implemented in the development of the e-government system. We use a function called *addObject(classname)* of a Zoop object to add the classes.

```
59  $zoop->addObject('DwesaTransmission'); // a class that defines a connection to a government file server.
60  $zoop->addObject('DwesaTransmissionAction'); // an action class that carries transmission of user data to remote locations.
61
62  $zoop->addObject('DwesaValidator'); // a class that validates all e-government entities
63  $zoop->addObject('Crawler'); // a class that updates Government forms on Dwesa E-gov System
64
65
66  $zoop->addObject('Category'); // a class that describes the category of a form
67  $zoop->addObject('CategoryAction'); // a class that performs action on a category
68
69  $zoop->addObject('DwesaMessage'); // a class that defines a Dwesa e-government message object
70  $zoop->addObject('messageAction'); // this class defines all actions on a message.
71
72  $zoop->addObject('Files'); // the class offers vital methods to manipulate forms on the system
73  $zoop->addObject('DwesaUtility'); // a class that provides utilities methods.
```

A short description is given next to each of the classes on the image shown above.

DORC *FindReceiver* Method

Dwesa community villagers do not require a government contact to file a grievance. The DORC portal intelligently determines the recipient department of the report. This method routes all Dwesa users' reports to the relevant government departments' back-end receiving area.

```
private function findReceiver($report)
{
    $report->setReceipient("Administration");
    $split = explode(" ", $report->getMessage());
    $municipal = "destroy|drink|damages|road|flood";
    $police = "abuse|beat|thieves|steal|break|stolen|lost|kill|dead|accident";
    $grant = "take|took|chase|cheat";
    $education = "exams|teacher|classes|school";
    $admin = "error|password|email address|testing|bug";
    $works = "fired|chased from|work";

    if(ereg($municipal, $report->getMessage()))
    {
        $report->setReceipient("Municipality Officials");
    }
    if(ereg($police, $report->getMessage()))
    {
        $report->setReceipient("Police");
    }
    if(ereg($grant, $report->getMessage()))
    {
        $report->setReceipient("Grant");
    }
    if(ereg($education, $report->getMessage()))
    {
        $report->setReceipient("Department of Education");
    }
    if(ereg($admin,$report->getMessage()))
    {
        $report->setReceipient("Administration");
    }
    if(ereg($works,$report->getMessage()))
    {
        $report->setReceipient("Department of Public Works");
    }
    return $report;
}
```

As shown above, if a router function fails to find the receiver of a report, the report is sent to the administrator, who will have to then manually forward the report to its intended destination.

DOAC *signApplication* Method

The method shown below originates from the `applicationAction` class (path “root_directory/objects”). The method accepts an application object as a parameter and digitally signs it.

```
public function signApplication($application)
{
    // creating an error handler
    $error_handler = create_function('$obj',''); // 'echo "error: ", $obj->getMessage(), "\n";
    // 1024-bit key pair generation
    $key_pair = new Crypt_RSA_KeyPair(1024);
    $form      = $application->getForm();
    $rsa_obj    = new Crypt_RSA;
    $error_handler($rsa_obj);
    $rsa_obj->setErrorHandler($error_handler);
    $signature  = $rsa_obj->createSign($form->getFormContent(), $key_pair->getPrivateKey());
    if(!$signature)
    {
        $result['message'] = "Your application could not be signed. Try again later";
        $result['result'] = false;
        return $result;
    }
    // signature checking
    $result['result'] = $rsa_obj->validateSign($form->getFormContent(), $signature, $key_pair->getPublicKey());
    $application->setSignature($signature);
    $result['returned'] = $application;
    return $result;
}
```

DOAC *validateApplication* Method

The `verifySignature()` is an `applicationAction` class method that validates a digital signature on an application form. The method accepts an application object, a valid signature and a public key as parameters as checks if the digital signature used to sign the document was created using the public key. If it was, then the application signature is deemed valid.

```
public function verifySignature(object $application, $signature, $public_key)
{
    $form = $application->getForm();
    $rsa_obj    = new Crypt_RSA;
    $ok = $rsa_obj->validateSign($form->getFormContent(), $signature, $public_key);
    if ($ok == 1) {
```

```
$result['result'] = true;
$result['message'] = "signature ok (as it should be)";
} elseif ($ok == 0) {
    $result['result'] = false;
    $result['message'] = "This application has incorrect signature. There are changes to your application
contents, sign it again";

    } else {
        $result['result'] = false;
        $result['message'] = "The application has no signature. Please sign it.";;
    }
    return $result;
}
```

Appendix C

DwesaCentre Database Objects

1. Applications Before Trigger:

```
Create Trigger applications_bi_trg
    BEFORE INSERT ON applications
    FOR EACH ROW
BEGIN
    DECLARE tot_size Integer;
    DECLARE tot_apps Integer;
    DECLARE tot_forms Integer;
    DECLARE dptid Integer;
    DECLARE catid Integer;
    DECLARE last_update VARCHAR(150);
    SELECT COUNT(*) into tot_size FROM tblperfomance WHERE Uid=New.Uid;
    SELECT COUNT(*) into tot_apps FROM tblperfomance WHERE Uid=New.Uid;
    select DateFiled into last_update from forms order by id desc limit 1;
    --SELECT COUNT(*) into tot_apps FROM tblperfomance WHERE Uid=New.Uid;
    IF New.Status = 3 Then
        SELECT Department into dptid FROM form_category inner join forms
        on form_category.cat_id = forms.cat_id WHERE forms.id=New.id;
        SELECT COUNT(*) into tot_forms FROM forms inner join form_category
        on forms.cat_id=form_category.cat_id where Department = dptid;
        UPDATE tbldepartments
            SET Applications_received = 1 + Applications_received,
                Application_last_sent = New.DateApplied,
                forms_count = tot_forms,
                forms_last_update = last_update
            WHERE Dpt_id = dptid;
    END IF;
    IF tot_apps > 0 Then
        UPDATE tblperfomance
            SET total_size = New.Size + total_size
            WHERE Uid = New.Uid;
    ELSE
        INSERT INTO tblperfomance
            (Uid, total_size, total_apps)
        VALUES(NEW.Uid, New.Size, 1);
```

```
END IF;  
END$$
```

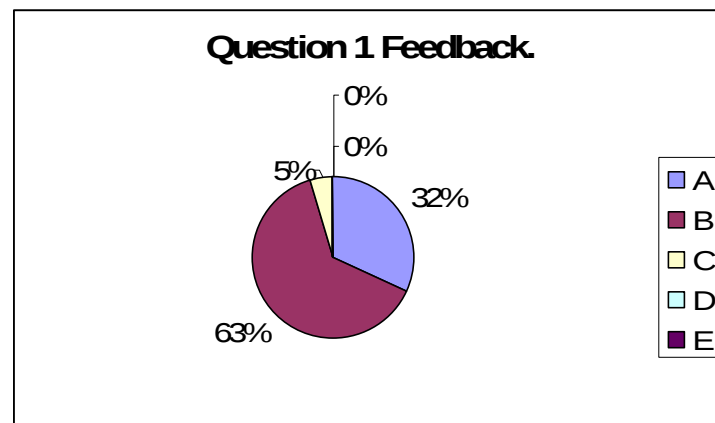
Appendix D

Questionnaire and Feedback

The questionnaire was aimed at recognizing the government services that are inaccessible to the Dwesa rural villagers. It is done as a part of the research being undertaken to develop a multifunctional e-government portal for the Dwesa rural community. The questionnaire followed the development of a first prototype of the e-government system and the data sought was used in the refining of the then prototype into the final version.

1. Which offices do you use to apply for an ID, Birth Certificate?

- A. Idutywa.
- B. Willowvale.
- C. East London.
- D. I don't know.

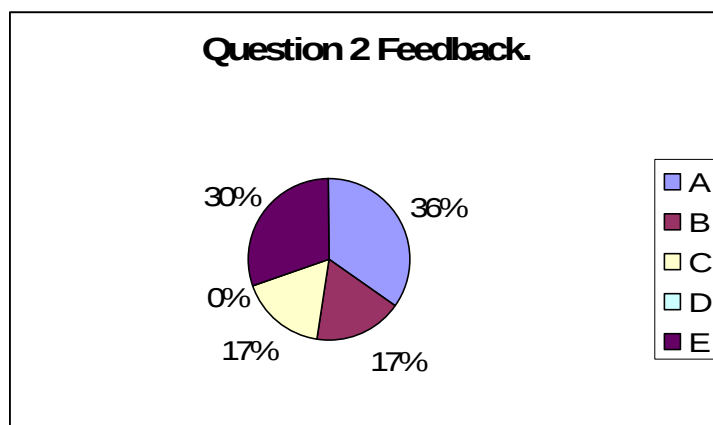


63% of the villagers have access to the Willowvale public department. This means users travel or walk for 40km to apply for an identity document or a form requisition. 32% choose to access the services as far as East London city. We found out that this is because villagers think it is quicker to have their ID applications done in East London. East London is also the main city conveniently accessible from Dwesa community.

2. Did the government give you an ID and a birth certificate?

- A. Yes, both documents.
- B. No, birth certificate only.
- C. No, I want to apply for an ID.

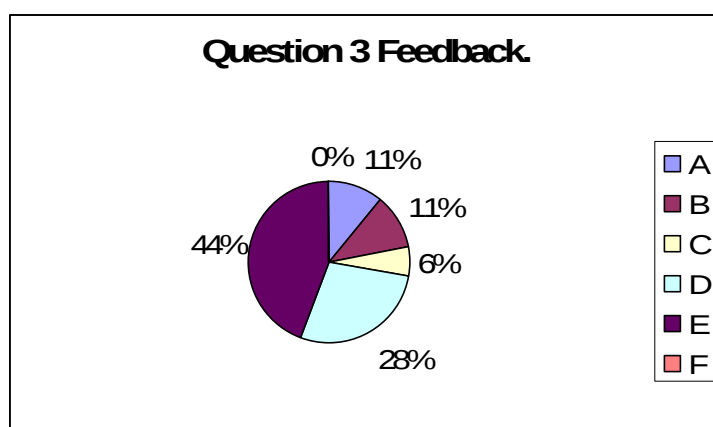
D. I don't have any.



Looking at the feedback from Question two, we noted that 64% of the participants did not have an ID. Only 36% of the participants had identity documents. We established that the 64% who did not have the documents did not resources to access to services at Home Affairs. These resources included bus fare and time.

3. Which of the following documents do you wish to apply for (mark the ones you like)?

- A. ID
- B. Birth Certificate
- C. Death Certificate for a relative
- D. Passport
- E. Drivers License
- F. None of the above.



All documents were required by the Dwesa community villagers. This was shown on the feedback received from the participants who represented the community.

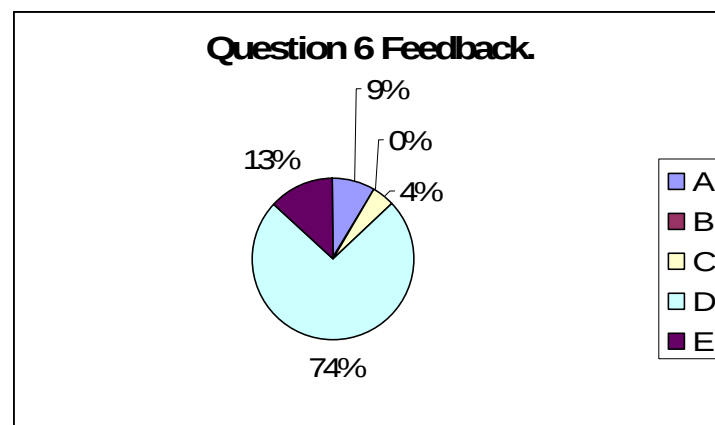
4. Which other service(s) do you wish to get from the government or municipality?

5. Between which age did you get your birth certificate?

- A. 0-6.
- B. 7-16.
- C. 17-28.
- D. 29 and older.

6. How many days did you spend applying for your ID?

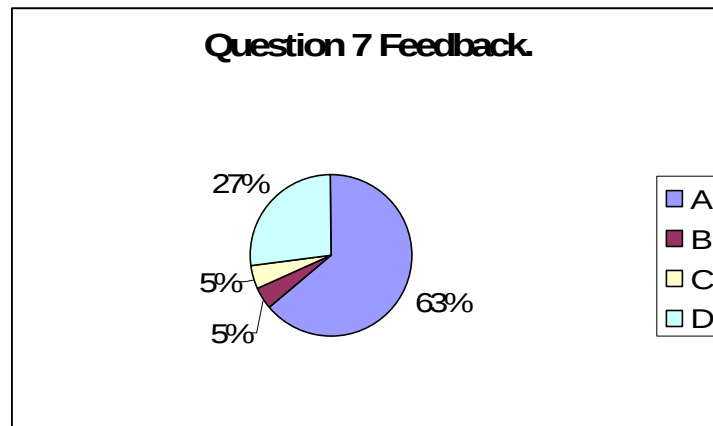
- A. 1 week.
- B. 2 weeks.
- C. 3 weeks
- D. 4 weeks or more.
- E. I don't have any ID.



The feedback showed that 74% of the participants expect the process of applying for a government document to last for more than 4 weeks. This shows that the development of an e-government can help to speed up the process of applying the government documents.

7. Do you think you can get a Passport Application form on the school computer?

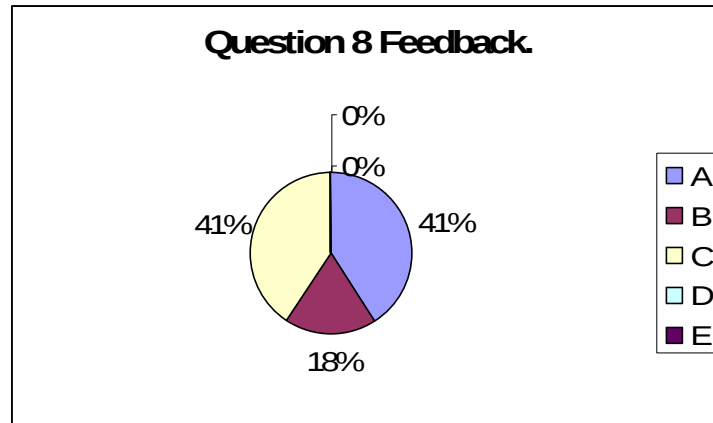
- A. Yes.
- B. No, it's not allowed.
- C. No, it's not possible.
- D. I don't know.



The feedback to this question showed that the Dwesa community villagers are confident that they can get a government form on a computer. 63% of the participants accepted that they can obtain a passport application form on the e-government system. 27% could not be sure.

8. Do you think you can apply an ID or Birth Certificate on the computer?

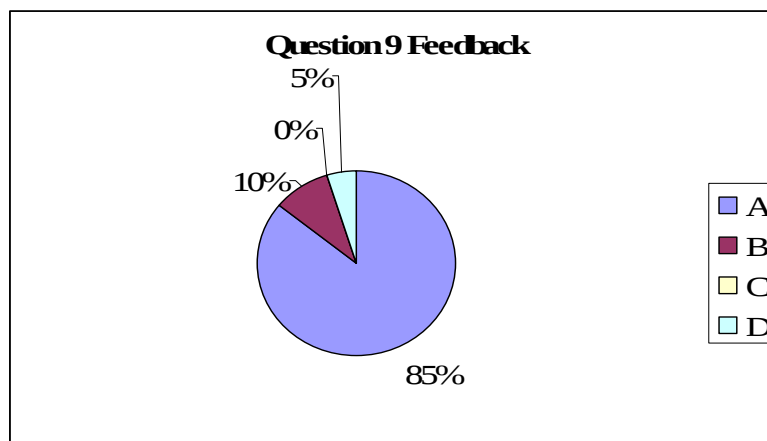
- A. Agree.
- B. Strongly agree.
- C. Not sure.
- D. Disagree.
- E. Strongly disagree.



A total of 59% of the participants agreed that they can apply for an Identity document or a birth certificate using their computer lab. These figures shows that large percentage of the Dwesa community was prepared to embrace the introduction of the Dwesa e-government system that would provide the community members an easy access to the South African government services.

9. Do you usually have village meetings?

- A. Yes.
- B. No.
- C. Not often.
- D. I don't know.

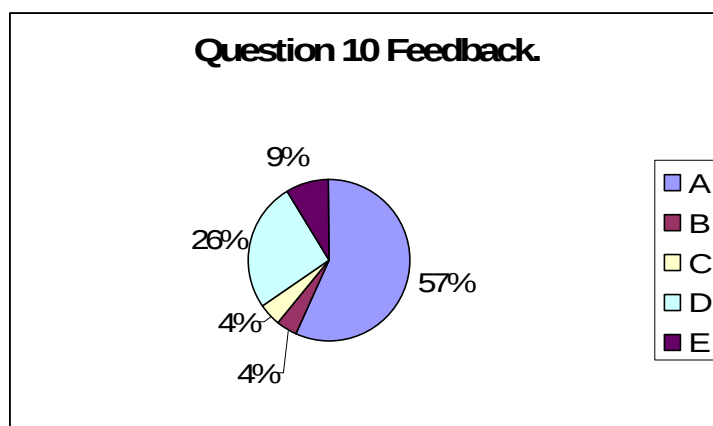


The feedback to this question showed that Dwesa community members hold regular meetings. The development of the Dwesa Forum Corner sought to enhance a digital community where the villagers can spread vital information across the community.

10. How do you know if you are having a village meeting?

- A. School announcement.

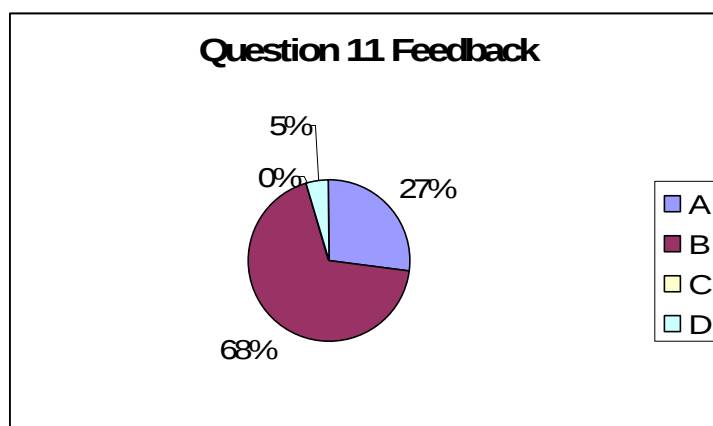
- B. Telephone.
- C. Rumour.
- D. Messenger.
- E. We don't have village meetings.



It is interesting to note that villagers are informed of a village meeting through largely by school announcement and by a messenger. 83% of the participants stated these as their most source of information when a village meeting is going to happen. We found out that the DFC portal will help to post announcements that can also be broadcasted by SMS should the method is financially supported.

11. Where do you report when you are robbed?

- A. Idutywa Police Station.
- B. Village head.
- C. Municipality.
- D. We do not report.

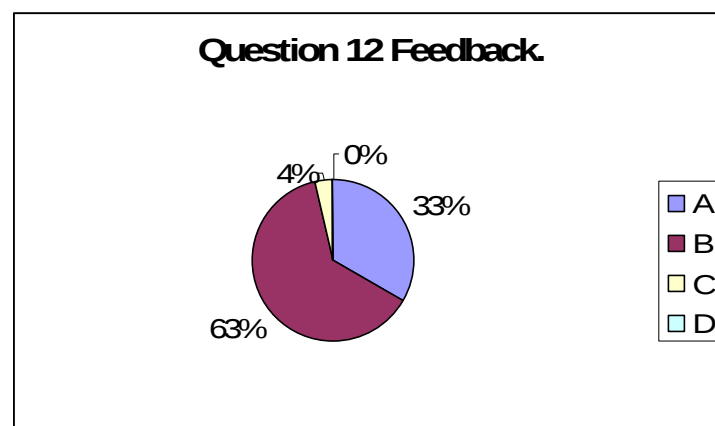


The feedback suggests that villagers report most of the grievances to the village head. 68% stated that they report to village head when they are robbed. Normally these reports should be

directed to the Willowvale or Idutywa Police station. We found out that, from the feedback, users are likely not to contact the crime prevention unit at these two business Centres due to the fact that they do not know who to report to or call to if they have cell phone access.

12. How do you report village matters, theft, violence or any misconduct?

- A. Use telephone.
- B. Walk to the offices.
- C. Email.
- D. Send a letter.



The majority of participants walk to the offices to report village matters, theft, violence or any misconduct if they can not contact the village head. 33% use telephone to contact the police offices at Willowvale business Centre. Very few use email as a method to send reports.

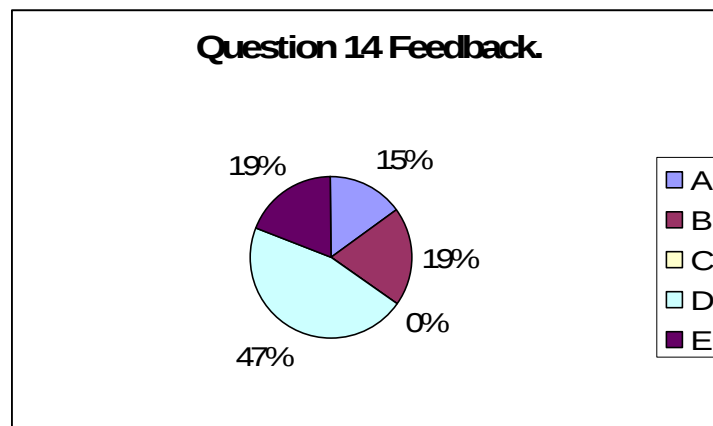
13. Are you receiving a grant from the government?

- A. Yes.
- B. No.
- C. No, but I want to apply.
- D. Once.

14. Which government grants do you want to apply for?

- A. A child grant.
- B. Old age grant.
- C. Disabled grant.
- D. Study grant.

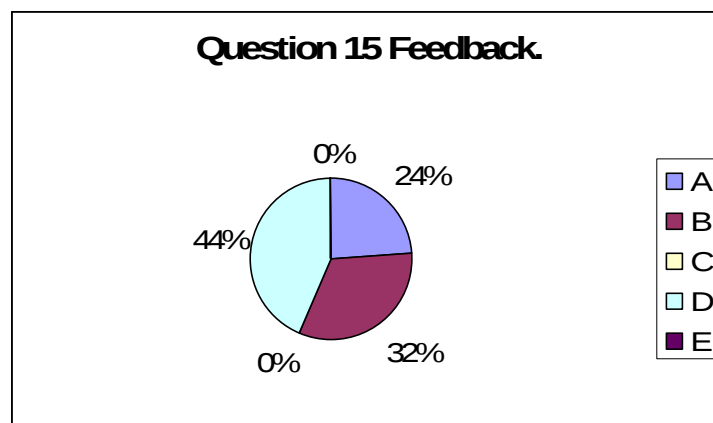
E. None of the above.



The feedback to this question shows that only 19% of the participants had nothing to apply to the Department of Social Work. The 81% of them had various types of grants they wished to apply and we found out that with the introduction of the DOAC portal, they would be able to do so.

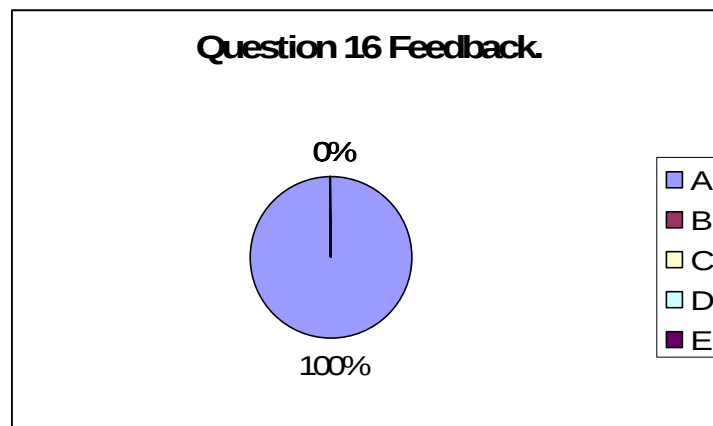
15. Do you think you can apply for a government grant on your computer?

- A. Agree.
- B. Strongly Agree.
- C. Not sure.
- D. Disagree.
- E. Strongly disagree.



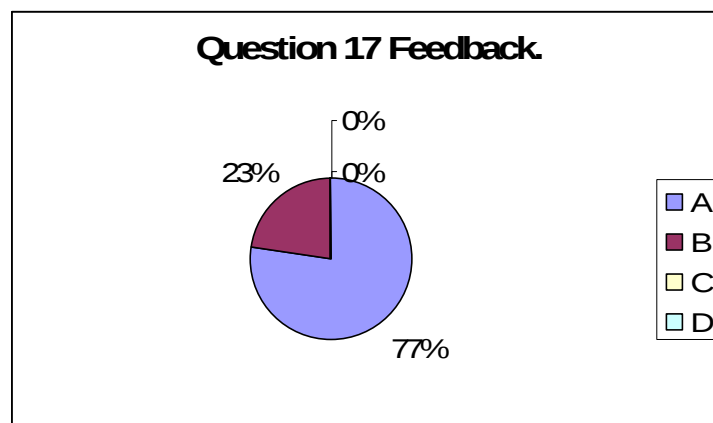
16. Do you have a cell phone?

- A. Yes.
- B. No.
- C. My parents have one.
- D. I used to have one.



17. Did you use a school computer before?

- A. Yes.
- B. No.
- C. I will never use it.
- D. We are not allowed.

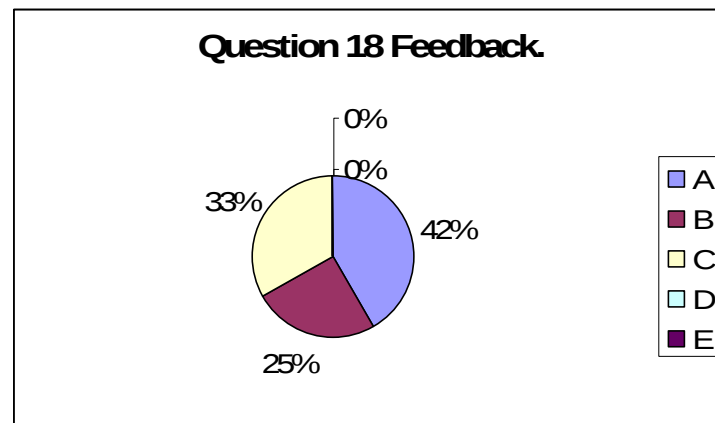


77% of the participants had used a computer before. This figure shows that the community members have a general access to computers. We found out that almost all those who had used a computer had done so in the computer labs we have in the field.

18. Do you think you can use a computer to report of anything to the government?

- A. Agree.
- B. Strongly agree.
- C. Not sure.
- D. Disagree.

E. Strongly Agree.



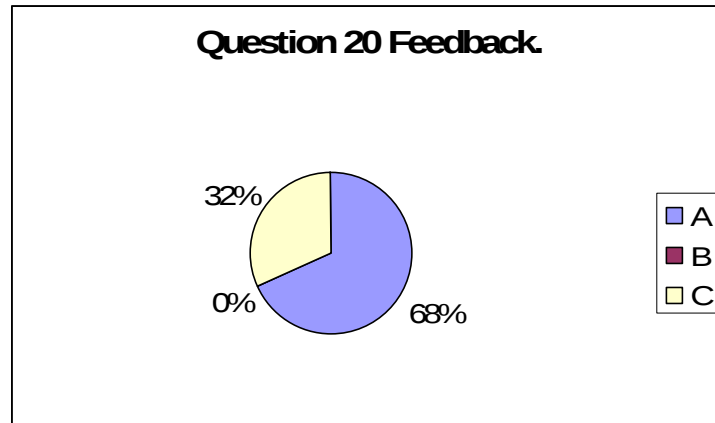
A total of 67% participants thought that they can report anything to their government or to the village head through a computer. We found out that the development of the Dwesa Online Reporting Centre had benefited the community.

19. Do you think the government will be happy to receive your passport, ID or Birth Certificate application form from your computer?

- A. Yes.
- B. No.
- C. I don't know.

20. After you used the demonstrated system, did you better understand how to use the interfaces and would you be able to work on your own without any assistance?

- A. I agree
- B. Strongly agree
- C. I disagree
- D. Strongly disagree
- E. Not sure



The feedback obtained from question 20 shows that after demonstration 68% of the participants agreed to have understood how to use the interfaces and thought they would access the services on the Dwesa e-Government system by themselves.