University of Fort Hare
Together in Excellence

Design and Implementation of a Network Revenue Management Architecture for Marginalised Communities

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

Master of Science

In

Computer Science

University of Fort Hare

by

Paul Tarwireyi

December 2007

Acknowledgments

One of the most crucial payoffs in pursuing this MSc degree was to be associated with intelligent and assistive minds. I am most fortunate and grateful in this regard. First of all, I would like to express my very special thanks to my supervisors – Professor Alfredo Terzoli and Professor Hyppolyte Muyingi for the invaluable guidance, advice, help and encouragement they gave me for this work.

I would like to acknowledge the invaluable help given by Mr Ronald Wertlen and Mr Mamello Thinyane on several implementation issues, the enthusiasm and farsighted guidance they provided for my research.

The financial assistance from the Zimbabwean government, Centre of Excellence in Developmental E-commerce (University of Fort Hare) and the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the University of Fort Hare or the funders.

This work was undertaken in the Centre of Excellence at the University of Fort Hare

I would also like to thank the many friends and colleagues who have helped make life during my MSc studies at the University of Fort Hare an enjoyable journey, sharing their experience and love - Thank you all! Furthermore, I would like to mention that the time and work involved completing this thesis has not come without sacrifice. I apologise for not always being there for you.

Finally, I am deeply grateful to my wonderful family which has supported me unconditionally throughout my studies, especially my mother for her immeasurable love.

Declaration

I acknowledge that all references are accurately recorded and that, unless otherwise stated, all work herein is my own.

Abstract

Rural Internet connectivity projects aimed at bridging the digital divide have mushroomed across many developing countries. Most of the projects are deployed as community centred projects. In most of the cases the initial deployment of these projects is funded by governments, multilateral institutions and non-governmental organizations. After the initial deployment, financial sustainability remains one of the greatest challenges facing these projects.

In the light of this, externally funded ICT4D interventions should just be used for “bootstrapping” purposes. The communities should be “groomed” to take care of and sustain these projects, eliminating as soon as possible a dependency on external funding.

This master thesis presents the design and the implementation of a generic architecture for the management of the costs associated with running a computer network connected to the Internet, The proposed system, called the Network Revenue Management System, enables a network to generate revenue, by charging users for the utilization of network resources. The novelty of the system resides in its flexibility and adaptability, which allow the exploration of both conventional and non-conventional billing options, via the use of suitable ‘adapters’. The final goal of the exploration made possible by this system is the establishment of what is regarded as equitable charging in rural, marginalized communities - such as the community in Dwesa, South Africa.

Table of Contents

Appendices

List of Figures

List of Tables

Acronyms

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AAA	Authentication, Authorisation and Accounting
AJAX	Asynchronous JavaScript and XML
API	Application programming interface
ATM	Asynchronous Transfer Mode
BSS	Business Support System
CDR	Call detail record
CHAP	Challenge-handshake authentication protocol
CRUD	Create, read, update and delete
CSIR	Council for Scientific and Industrial Research
DBMS	Database Management System
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
FTP	File Transfer Protocol
FOSS	Free and Open Source Software
GNU	GNU's Not UNIX
GUI	Graphical user interface
HTML	Hypertext Markup Language
HTTP	Hyper Text Terminal Protocol
HTTPS	Hyper Text Terminal Protocol Secured??
IP	Internet Protocol
ICT	Information and Communications Technology

ICT4D	ICT for development
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IPDR	Internet Protocol Detail Record
IPSec	IP security
ISP	Internet service provider
L2TP	Layer 2 Tunnelling Protocol
LAMP	Linux, Apache, MySQL, PHP
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MGD	Millennium Development Goals
MVC	Model View Controller Architecture
NAT	Network Address Translation
NAS	Network Access Server
NGO	Non-governmental organization
NSE	Network Service Element
PAP	Password authentication protocol
PKI	Public key infrastructure
PPPoE	Point-to-Point Protocol over Ethernet
PPTP	Point-to-point tunnelling protocol
QoS	Quality of Service
RADIUS	Remote Dial In User Protocol
RFC	Request For Comments
SSL	Secure Sockets Layer
TCP	Transport Control Protocol

TSL	Transport Secure Layer
UAM	Universal Access Method
URL	Uniform Resource Locator
UNDP	United Nations Development Programme
VSAT	Very small aperture terminal
VPN	Virtual private network
VoIP	Voice over Internet Protocol
WiMAX	Worldwide Interoperability for Microwave Access
WWW	World Wide Web
WPA	Wi-Fi Protected Access

Chapter 1

Research Introduction

1.1. Introduction

Information and Communications Technology (ICT) for development has gained wide scholarly interest and support from governments, non-governmental organisations and private companies. ICTs are believed to play a crucial role in poverty alleviation and strengthening positive social dynamics, hence are viewed as enablers of community development [1]. As a result, there has been an increase in the number of projects that seeks to deploy ICT in rural hinterlands.

Over the last decade, the South African Universal Service Agency and other organizations have been searching for ways of providing ICTs within the reach of all citizens [2]. Over a period of time, community networks with wireless backbones, supported by wired networks, have proven to allow a more pervasive ICT penetration in marginalised communities. Such networks have sprouted up across South Africa, making it possible for the rural hinterlands to be connected and enjoy the benefits of the information society [3].

1.2. Research problem

As developing nations strive to bridge the digital divide, governments, multilateral institutions and non-governmental organizations, have taken the lead in sponsoring the deployment of ICT projects in marginalised communities [4-6]. Conceding the fact that infrastructure or start-up costs have been footed by these organisations, very few of the deployed projects planned for long-term financial sustainability [7]. This, in many ways, has been the Achilles' heel of ICT4D interventions. Most of these projects are characterised by financial short-sightedness and lack of measures and strategies to ensure long-term sustainability [3]

If a project is not financially sustainable, its economic and social benefits are unlikely to be realized. So, many projects promoting the use of ICTs in marginalized rural communities have failed in the past, partially due to economic sustainability problems [7].

Even though it sounds like a utopian goal in the light of many experiences, projects should aim for financial sustainability and hosting communities should be groomed to take care of and sustain these projects.

This process could be helped by identifying means and ways that can be used to raise revenue in these projects. The generated revenue will then be used to meet financial obligations.

Generally, ICT interventions need to be sustained in every sense; that is culturally, socially, technically and economically or financially. However, the focus of this research is on financial sustainability.

1.3. Research objectives

The primary objective of this research was to prototype an architecture to explore ways of cost recovery on a deployed network that are seen fair by the community hosting the network and therefore accepted. The architecture not only should allow experimenting with charging users on traditional billing metrics, such as bandwidth and time, but also incorporate contextualised billing mechanisms and attempt to integrate the community's conceptualisation of fairness.

When entering a community we will not be having the knowledge of what they regard as fair ways of pricing or sharing costs. We can explore this untapped knowledge by developing a system which allows us to bill based on any factor, such as income, age, number of dependents and monthly household expenditure, or even number of cows.

The overall objective maps into a series of sub-objectives the most important of which are:

- Building an authentication infrastructure – this deal with Authentication, Authorisation and Accounting issues. Users need to be authenticated and authorised before they are granted access to the network resources. As they use the resources, the system should be able to collect information on resource usage so that at the end of the usage the system will be able to ascertain what the user did.
- Building a billing and rating infrastructure – after ascertaining what a user has done on the network, there is need for functionalities which will enable the determination of the cost of usage. After this cost is determined invoices should be generated and sent to the respective users.
- Building a general framework where adapters for conventional and non-conventional billing metrics can be deployed and tested.

The guiding design principle for the system architecture was to provide flexibility and extensibility.

1.4. Research context and scope

This project was undertaken within the context of the Siyakhula project, which is undertaken in Dwesa. The Siyakhula project is an ICT4D project, initially with the aim "to develop and field-test the prototype of a simple, cost-effective and robust, integrated e-commerce / telecommunications platform for marginalised communities in South Africa" [8]. It is run jointly by the University of Fort Hare (UFH) and Rhodes University (RU). This project has deployed a wireless network in Dwesa. The deployed network runs on an Open Source platform. This platform is essentially a multi-functional, distributed communications system designed to integrate different functionalities.

The current deployment in Dwesa is based in schools which are connected to a WiMAX backbone. The schools were just identified as the best locations to put the resources, but not necessarily the final locations. This is due to the fact of them having electricity and being learning centres. They act as central locations which serve communities in their vicinity.

For the past two years the project has been running with the financial support of the Centre of Excellence (UFH and RU) through the partnership with the industry. However, the philosophy of this specific project can be understood by using the analogy of nurturing a child, until he is able to provide for himself and then others. So this project is to be nurtured until it is mature and able to sustain itself, and hopefully can also support other, upcoming projects. This philosophy is similar to the ideology behind Innovation Hubs [9]. They scout talent and nurture it until it becomes a viable business.

In the light of this view, my research aimed at putting in place the infrastructure to enable the deployed network to raise or generate revenue. In a way, the act of charging means that the users are sharing the costs amongst themselves. This research does not determine what is regarded as fair, when it comes to the issue of sharing costs in a specific community. Rather, it provides the infrastructure that would enable the exploration and experimentation needed in a specific community to find a way to charge customers regarded as fair.

1.5. Thesis outline

The rest of the thesis is structured as follows:

Chapter 2 presents some introductory and background material. Specifically, it explores the digital divide which is the context in which this research was carried out. Reasons why ICT can be an important tool for development are given. Then, this chapter reviews work related to Internet pricing, then charging and accounting systems for communication services.

Chapter 3 explores the research objectives in more detail. It describes who will use the system, how and what the system will do. It also details the requirements and objectives of the system. During this process, the methodology followed in this research is also highlighted.

Chapter 4 provides a detailed description of the system architecture. It delves into a conceptual solution that fulfils the system requirements by highlighting the design scope and identifying and describing the functional blocks that compose the system.

Chapter 5 discusses the technological choices that were made with regard to the implementation of the various components of the prototype built. A brief discussion of how each technology works is conducted and reasons for choosing the various technologies are given.

Chapter 6 describes how the various tools are used in the implementation of the Network Revenue Management System. It is in this chapter where we delve more into how the essential components of the prototype were developed and implemented. In particular, this chapter discusses how the non-conventional billing options are made available. A system walkthrough to demonstrate how the system works concludes the chapter.

Chapter 7 is a general discussion of the overall prototype implementation. It also gives an overview of the Internet usage patterns in Dwesa as observed over a period of seven months.

Chapter 8 summarizes the work presented in this thesis and articulates the extent to which the research meets the research goals. Furthermore, suggestions of further research that can be conducted are outlined.

Chapter 2

Background and related work

2.1. Introduction

This chapter introduces the concept of digital divide, which is the context in which this research was carried out. Models which have shown the potential to bridge the digital divide are discussed. This chapter also gives a synopsis of ICT4D projects and why ICT is an important tool for development. The sustainability of ICT projects and the factors which influence its achievement are also discussed. This is followed by an introduction of Dwesa-Cwebe, which is the community where this research was conducted and applied. Finally, we discuss the state-of-the-art in the area of charging and accounting for telecommunication services.

2.2. Digital divide

Difference in access to information technology, or what is frequently referred to as the digital divide, has become an important focus area both in policy and research. There is a substantial amount of literature that focuses on the differences between those with access to information technology and those without access. These studies have been instrumental in understanding the inequalities brought about by the differences in access to ICTs. The spread of digital communications and information technology resulted in the digital divide. The digital divide seems an intractable gap that came as a result of the creation, possession, distribution of technology. Accordingly, bridging the digital divide essentially refers to the efforts dedicated to provide information technology resources to those who do not have them.

There are different views on the effectiveness of ICTs in eliminating socio-economic inequalities. The instrumentalist view considers ICTs as a powerful instrument with the capacity to catalyze social transformation, whilst the structuralist view believes that the structure of the social actions, attitudes, perceptions and processes determine the adoption of ICTs within a society. Each view has a positivist side and a fatalist facet [10].

The positivist instrumentalist view believes that ICTs have the capacity to leapfrog development. On the other hand, the structuralist view believes that ICTs will level the playing field, thus changing the structure of societies.

On the fatalist side, the general view is that ICTs will benefit the prevailing power structures. There is no guarantee that the diffusion of technology in marginalized communities will equally benefit everyone in those communities. In the process of bringing ICTs to marginalized communities we will be creating other gaps as it is inevitable that not all the members of the community will embrace ICTs as an opportunity. There are gaps everywhere in our economies: the income gap, the literacy gap, the electricity divide, the achievement gap, and many more. The question is why only single out the digital gap? Which gap should we really close first? Is it possible to close without closing the others? Is technology more important to a poor man than basic necessities such as food and clean water?

In his first news conference after his appointment as chair of the Federal Communications Commission (US), Michael Powell expressed scepticism about his agency's role in addressing the digital divide. In his statement he said:

I think the term "digital divide" sometimes is dangerous in the sense that it suggests that the minute a new and innovative technology is introduced in the market; there is a divide unless it is equitably distributed among every part of society, and that is just an unreal understanding of an American capitalist system… I think there's a Mercedes Benz divide, I'd like one, but I can't afford it…..[11].

These sentiments are the same as those of Wade, who stated that ICTs can only help people absorb knowledge gained somewhere and apply it to local needs [12]. These views suggest that ICTs are being touted as the "magic formula" for development. It is like saying cheaper books will help eradicate illiteracy. Wade also suggested that efforts to bridge the digital divide may have the effect of locking developing nations into a new form of dependency on the West since the technology and its strategies are mostly designed and developed in developed countries [12]. However, the positivist side holds a different perspective.

2.3. ICT for development

With regard to ICT for development, the positivists believe that ICTs play a crucial role in poverty alleviation and strengthening social dynamics. Most scholarly interest has taken this attitude towards digital divide. Even though there is lack of empirical evidence to support the effectiveness of ICTs in alleviating poverty [3, 4, 6, 7, 13-17], they are working against all odds to reap the fruits of bridging the digital divide. Essentially, the idea is that one cannot just sit and do nothing about the digital divide because of the lack of guarantee that ICTs will benefit everyone equally on our target communities. The fatalist notion clearly accepts that at least some people who were previously not benefiting from ICTs will benefit if ICTs are deployed in their area. As a result, the belief is that ultimately, the sum, multiplicity and dynamics of all those small achievements is a great achievement. Even though ICTs and their technologies are mainly designed and developed in developed countries, when deploying them in developing countries, all that is needed for them to work is a strategy for integrating indigenous knowledge, adaptation and contextualisation.

Most developing countries and developed countries alike have also taken the positivist attitude towards the digital divide. They view ICTs as an enabler for community development [1]. An example of this attitude was articulated by the then President of the Republic of Korea Kim Dae-jung in his statement where he mentioned that the success or failure of nations as well as the prosperity of mankind, depends on our ability to wisely develop our human resources [18]. He went on to say that:

During the twentieth century tangible elements, such as capital, labour and human resources were the powers that boosted economic development. But in the new century, intangible elements such as information and creativity will give nations a new competitive advantage. Consequently, if we are able to develop the potential of our citizens by fostering a spirit of creative adventure, individuals as well as nations will become rich, even if they do not have much capital, manpower or natural resources

Not only do these proponents agree that ICT is not the cure for all problems, but they also believe that the key to participation in the radically changing global economy lies in possessing the right skills to access the right tools. It is in the light of this reality that we have seen developing countries, NGOs and private institutions redoubling their efforts to prepare the marginalized communities to harness the benefits brought about by the information society. This is witnessed in global partnerships such as the UN's Millennium Development Goals (MGDs), which promote poverty reduction and education as part of its eight goals [19]. As a consequence we have seen many ICT4D projects sprouting up across many developing countries.

Researchers and organizations have formed a coalition with a single purpose - to alleviate digital divide. Extensive work has already been done to study, understand and define what the digital divide really is. The digital divide is manifested in many different ways. These include infrastructure divide, access, language, usability, jobs and demographics divide [20]. Therefore, to address the digital divide, a comprehensive solution or framework which will address all the different sub-categories is needed.

2.3.1. ICT4D in South Africa

In South Africa, since the 1980s, the South African Universal Service Agency and other organizations have been searching for ways of providing ICTs within the reach of all citizens [2]. According to Benjamin [5], small, profit centred solutions have sometimes been successful, whilst none of the implemented donor funded solutions have shown a model that is sustainable. In the light of this reality, community-driven wireless networking has emerged as a promising model for supporting universal access in disadvantaged and underserviced communities [6, 15]. The model is often implemented through some form of community access point for information and communications services, often known as a community telecentre or telekiosk [4-6].

According to the UNDP [21], a telecentre is a community centre that offers shared access to ICTs for the purposes of community development. In this model, the infrastructure and the Internet are delivered as community resources that should be financially sustained by the community. This model has shown a lot of potential as an answer to the digital divide woes. Hence, it has emerged as the foremost means of using ICTs for delivering services to the marginalized communities. However, uncertainty about how to finance telecentres has hindered their spread. This means that robust, community-based financing strategies will be needed to ensure the wide spread and sustainability of ICT interventions.

Examples of ICT4D at work in South Africa include projects being run by the Council for Scientific and Industrial Research (CSIR) in South Africa [22]. CSIR is one of the leading scientific and technology research, development and implementation organisations in Africa. It undertakes directed research and development for socio-economic growth. Under the theme of ICT for rural development, the CSIR icomtek developed a wireless network system that could provide a solution for the roll-out of ICTs to rural communities. At a pilot installation in the Eastern Cape, the network provides point-to-multi-point connectivity within two villages linking a school, clinic, hospital, police station, and a community telecentre thus forming a community Intranet [22].

2.4. Sustainability

For these projects to gain wide acceptance, they need to be sustainable. Pade, discusses ICT project failure and sustainability by exploring literature related to rural development and ICT [14]. She also emphasises the importance of being proactive in managing ICT projects so as to ensure their long term survival.

Sustainability is a complex and highly contested issue and there is considerable debate over its actual meaning. The term sustainability was popularised by the Bruntland Report [23]. In the context of development, this was articulated to mean 'the development that meets the needs of the present without compromising the ability of future generations to meet their own needs". Sustainability implies continuity, or the ability of something to continue over a long time [24]. It is a property of the project's capacity to cope with an uncertain future. In other words, we can say a project is sustainable if its net benefits endure as expected throughout the lifetime of the project. Sustainability is also a property which arises out of interactions with stakeholders. This means that project leaders have to interact with the community in order to gain community buy-in. As a matter of fact, projects are managed better if the owners have some stake in them. Once they are given the sense of owning the projects, efforts should then be directed on how they can nurture those projects for future generations.

2.5. Research environment

2.5.1. Dwesa-Cwebe

Dwesa-Cwebe is located in the remote Wild Coast of the former homeland of Transkei, in the Eastern Cape province of South Africa. It is situated between the Nqabara River (32^0 12 S; 28^0 58 E) and the Ntlonyana River (32^0 20 S; 28^0 48 E), is composed of both communal land and state land, and its size is approximately 235 km2 [25]. Although Dwesa and Cwebe are different areas, they are usually collectively referred to as Dwesa-Cwebe. Dwesa and Cwebe are separated by the Mbashe river. They fall under the Mbashe Municipality which in turn falls under the jurisdiction of the Amatole District Municipality based in East London. Willowvale and Idutywa are the towns closest to Dwesa, 50 and 75km respectively, whilst the closest towns to Cwebe are Elliotdale and Umtata, 50 and 100km respectively [25].

The population of Dwesa-Cwebe is approximately 15 000 people, housed in 2 270 households [25]. Dwesa-Cwebe was the site of one of the first land restitution projects in post-apartheid South Africa [26]. This area is in many ways a reflection of many rural realities in both South Africa and Africa as a whole [3, 16].

According to [17] the Dwesa-Cwebe community is characterized by:

- Scarcity and absence of public transport on the 50 km dust road from Willowvale to Dwesa there is no regular bus servicing that route;
- Scarcity of technical personnel;
- Low level of literacy due to a large number of school dropouts. (The educational situation of the Eastern Cape population aged 20 years and above is illustrated by Table 2-1 and Figure 2-1. The diagrams show that 22.8% of this age group did not attain any schooling, whilst only 6.3% managed to finish higher education. Further exploration in Table 2-2 reveals that the black population is the one which is most affected);
- Sub-standard educational facilities. For instance Figure 2-2 and Figure 2-3 below show the administration office and grade one classroom at Entubeni Junior Secondary School, which is one the schools in Dwesa;
- Low level of economic activities. Even agricultural activities are not that high in this area, probably due to reasons attributed to the terrain of the area;
- Low income per capita – about R9800 per annum, on average [27]. The people rely on pensions/grants, remittances, local resource utilization, livestock production and cultivation for livelihood [26];
- Absence of electricity in most parts of the area [28];
- Poor telecommunications coverage. For example there are no working Telkom landlines and there is poor cell phone coverage (this is improving over time);
- Difficult topographical conditions (semi mountainous) which render the construction of wire networks costly and;
- Poor infrastructure in general.

	No Schooling	Some Primary	Complete Primary	Some Secondary	Std 10/ Grade 12	Higher	Total
Eastern Cape	743583	644101	240396	963217	459434	204509	3255240
% in EC	22.8	19.8	7.4	29.6	14.1	6.3	100
South Africa	4567275	4084259	1623536	7846096	5200415	151180	24472761
% in SA	17.9	16.0	6.4	30.8	20.4	8.4	100

Table 2:1: Level of education in the Eastern Cape Province and South Africa from [29]

This table is realized graphically in figure 2-1 below.

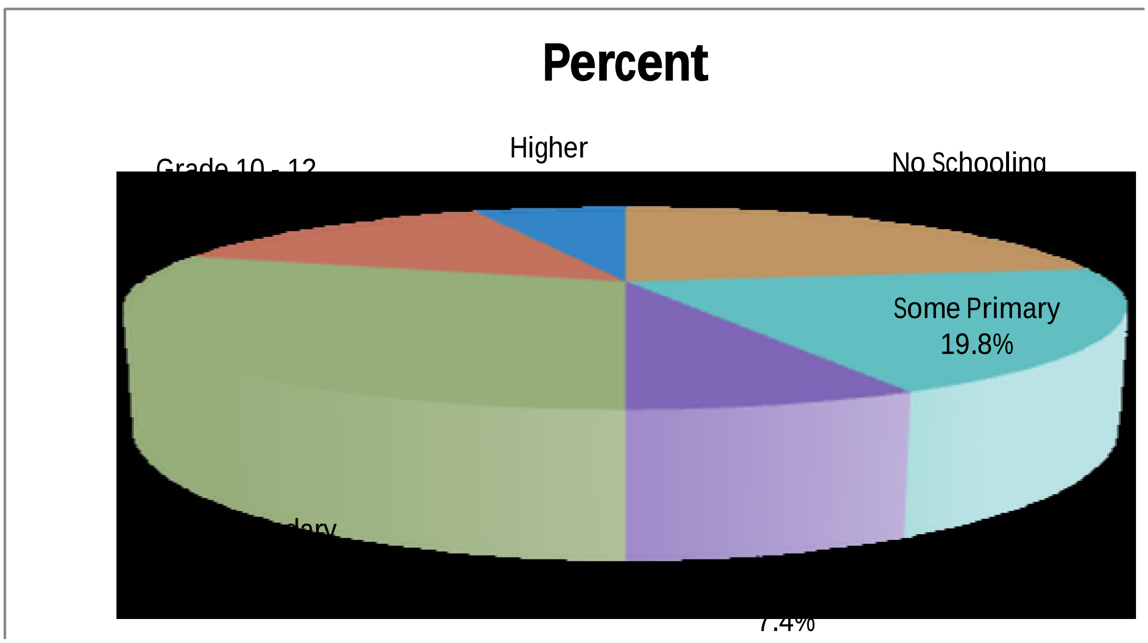


Figure 2:1: Level of education in the Eastern Cape Province and South Africa

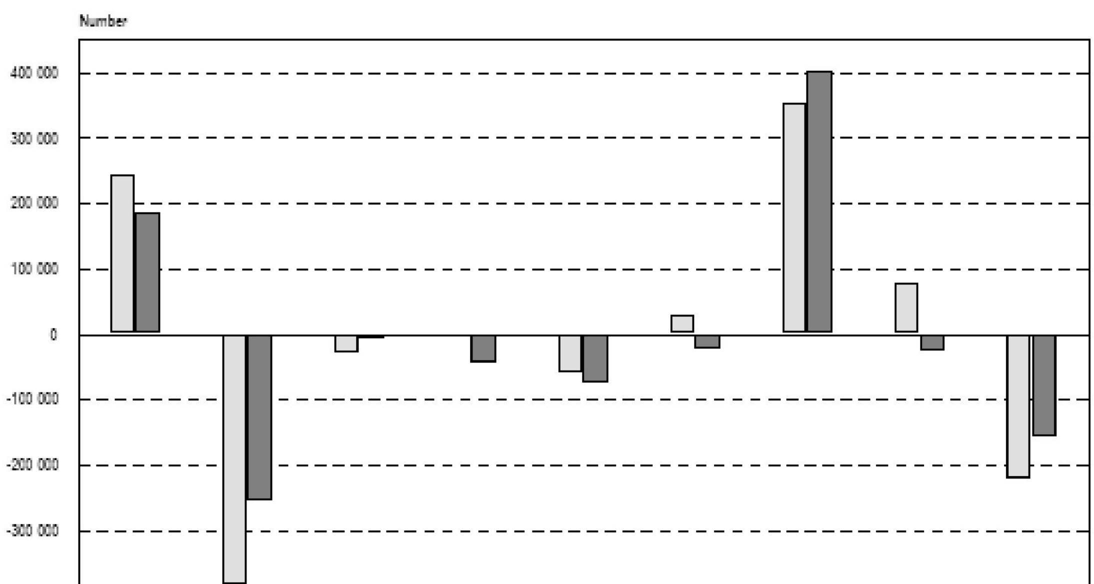
Level of Education	Black African		Coloured		Indian/ Asian		White		Total for EC	
	#	%	#	%	#	%	#	%	#	%
No schooling	715837	26.1	25187	9.2	301	2.5	2257	1.0	743383	22.8
Some primary	590357	21.5	50704	18.5	545	4.5	2495	1.1	644101	19.8
Complete Primary	209696	7.6	28515	10.4	350	2.9	1835	0.8	240396	7.4
Some Secondary	776304	28.3	111187	40.5	3565	29.6	72161	32.1	963217	29.6
Std 10/ Grade 12	316431	11.5	46940	17.1	4131	34.3	91932	40.9	459434	14.1
Higher	135096	4.9	11960	4.4	3141	26.1	54312	24.1	204509	6.3
Total	**2743721**	**100**	**274493**	**100**	**12034**	**100**	**224993**	**100**	**3255241**	**100**

Table 2:2: Educational levels in the Eastern Cape Province by population groups from [29]

Figure 2:2: Entubeni JSS Administration office

Figure 2:3: Entubeni JSS grade one classroom

Like most rural areas, lack of adequate productive employment in Dwesa-Cwebe has made life difficult for the locals. This is readily evident in the propensity of the youth to migrate to cities in other provinces, Gauteng being the main destination. The Eastern Cape has the highest net loss in internal migration as illustrated by Figure 2-4 [29]. According to the Labour Force Survey of September 2004, the working-age population of the Eastern Cape was estimated at just over 4 million. Out of this total, only 45% were economically active. The unemployment rate for the province was 30% and the highest numbers of unemployed people were black Africans. In 2001, about 21% of the population of Eastern Cape left the province for other provinces whilst only 6% migrated to the Eastern Cape. However, among the migrants who leave the Eastern Cape, many lack proper qualifications to get employment, or will secure employment in relatively low paying jobs. As a result, a lot of social issues occur, such as psychological isolation and discrimination. Some of these social issues have led to high crime rates in South Africa. Out of the total number of crimes reported in South Africa for 2003, Eastern Cape accounted for 11,2% which was the fourth highest in the country.



	Western Cape	Eastern Cape	Northern Cape	Free State	KwaZulu-Natal	North West	Gauteng	Mpuma-langa	Limpopo
1996	244 232	-384 892	-29 184	-6 262	-60 715	30 530	351 490	78 040	-223 239
2001	184 999	-254 791	-7 445	-44 259	-75 333	-22 172	403 326	-26 992	-157 333

Figure 2:4: Net gains/losses in internal migration by province, South Africa, 1996

Dwesa-Cwebe was chosen as the setting of research efforts run in cooperation by the University of Fort Hare and Rhodes University, in which a complete, distributed and Internet-enabled computing system was deployed in early 2006. The aim of the project is to explore the effects of ICT in a deep rural setting, as an enabler of economic growth and better quality of life [3]. A group of researchers from both Universities visits Dwesa once every month. Each monthly visit last for about a week.

2.5.2. The network and Computing Infrastructure

The network was provided as a community resource and the deployment was centred on schools, which act as points of access for the community at large. It is a converged IP network consisting of both wired and wireless sections. In the setup, VSAT provides the backhaul connectivity to the Internet, while. IEEE 802.16 (WiMAX) is the last-mile, local-loop access to the various schools [17, 30]. Currently, the points of presence are four schools: Mpume, Ngwane, Mtokwane and Nondobo, as shown in Figure 2-5 [3].

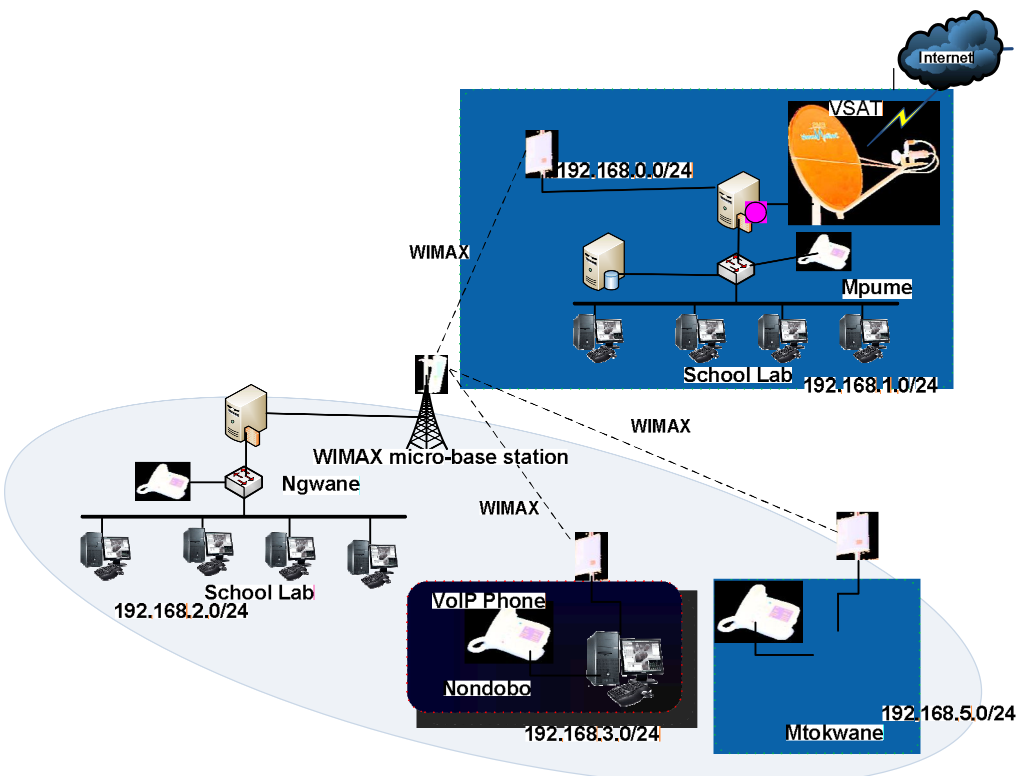


Figure 2:5: Dwesa Points of Presence

Mpume: This was the first point of presence for the project. Due to its strategic location and availability of electricity, it was chosen as the site for VSAT installation.

Ngwane : This is in Line Of Sight (Los) with Mpume and other schools. Because of its location, it was chosen as the site for the WIMAX base station installation.

Mtokwane and Nondobo (schools C and D): These schools are in line of sight with Ngwane, and this enabled the extension of the Internet to them.

The detailed network topology is illustrated in Figure 2-6 shown below [31].

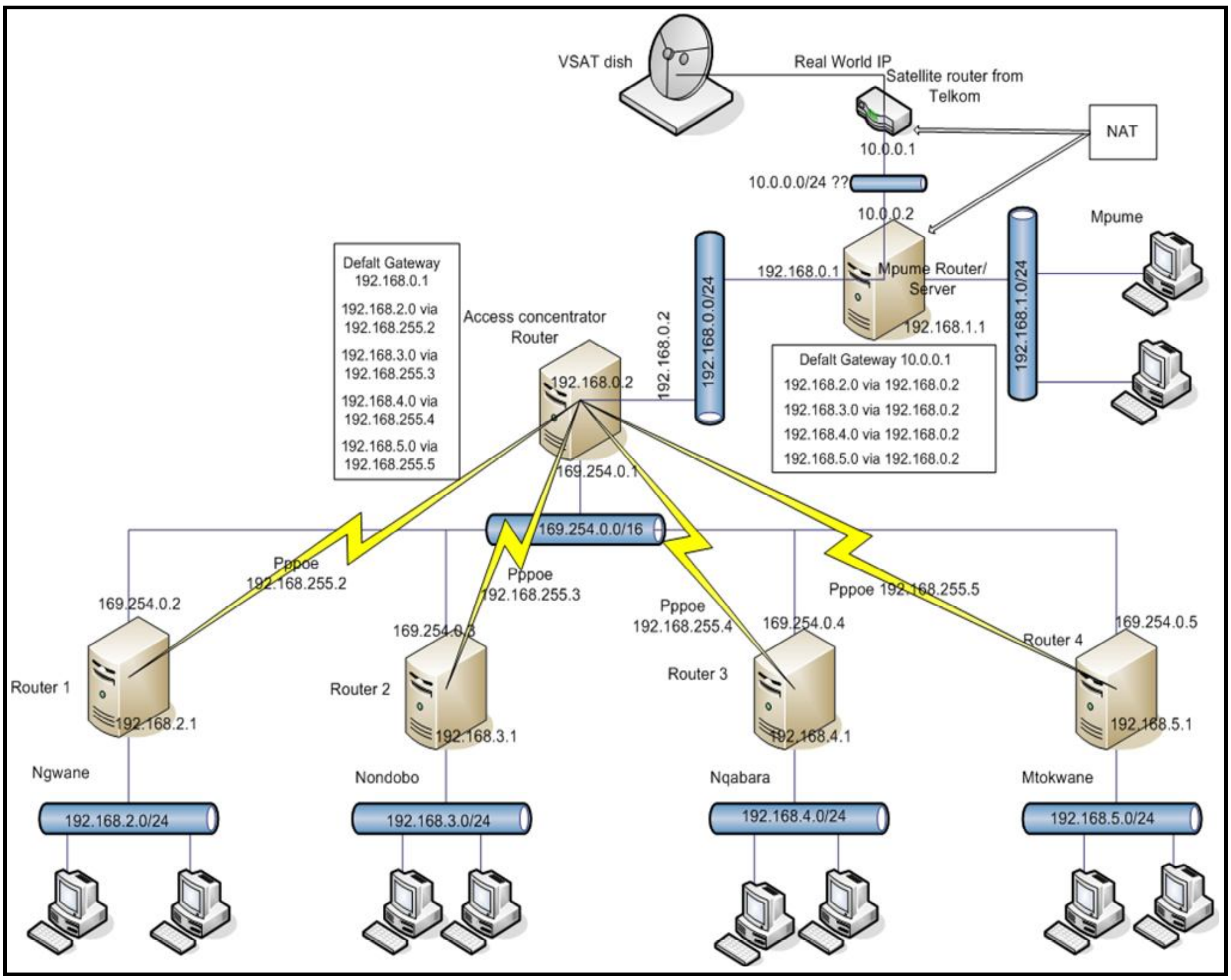


Figure 2:6: Dwesa Network topology

As documented in the Dwesa Networking document [31], at each of the schools there is a LAN, with a server, a router and clients.

For the routers low end Intel Pentium III FreeBSD 6.1 machines are used. Each router has two interface cards: an internal and an external network interface. For example, the Ngwane router has interface with the IP address 192.168.2.1 as its internal interface whilst 169.254.0.2 is on the external interface. Each router is configured such that when it receives packets on its internal interface that are destined for other networks, including the Internet, it will forward them to the access concentrator.

A FreeBSD 6.2 Pentium III 500MHz processor machine is used as the access concentrator. This acts as the router for the WiMAX test bed network. It establishes communication tunnels with other routers on the WiMAX network. It runs a PPPoE service for the clients on the network. Each of the routers then uses the “raw” network to establish a PPPoE session, authenticating itself with a username and password to the access concentrator, which checks the credentials (against a plain-text password list) and establishes the PPPoE session. The access concentrator provides each of the school routers with a PPPoE IP address in the range of 192.168.255.0/24. It will pass on all the Internet traffic, as well as traffic for the Mpume local network, to the Mpume server. The Mpume server has three network cards. One connects to the backbone network via the access concentrator, with 192.168.0.1 as its IP address. The second network interface binds an IP address (192.168.1.1) on the LAN at Mpume, allowing the school’s local computers to communicate with the server. The local IP block for the LAN at Mpume is 192.168.1.0/24. The third and final network interface connects to the VSAT and so to the Internet.

This network interface binds an IP within the 10.0.0.0/8 range and is configured via DHCP. Incoming network traffic will be either intended for other computers on the LAN, or the Internet, or needs to be sent to one of the other school networks. If traffic is for the LAN, then the server will route it locally; if it is intended for one of the other school networks, then it will forward it on to the access concentrator. The Mpume server is configured such that its default gateway to the VSAT connection.

2.5.3. Sustainability in Dwesa

A study was conducted in the early phases of the Dwesa project to identify and explore factors promoting sustainability in rural ICT projects[14]. The preliminary investigation revealed that most researchers link the problem of sustainability to financial sustainability. However, sustainability is not just about the financial side of projects but also encompasses the multiplicity of other issues such as cultural, social and political [32] factors. Sustainability for the Dwesa project was classified into social and cultural, institutional, economic/financial, political and technological sustainability [14, 32].

2.5.3.1. Social and cultural sustainability

This looks at the social and cultural context in which the project operates and the capacity of the project to adapt to such context. When a project shows consideration for the community's social and cultural aspects, the people feel empowered, and therefore are more willing and active in seeking ways to keep it running [32]. This means that the project should identify the different social strata, their needs, expectations and perceptions and then address these proactively. As far as the Dwesa project is concerned, a number of interventions have been undertaken to ensure the cultural and social sustainability of the project. These include the linguistic and cultural localization [16] of HCI interfaces on software deployed in Dwesa. It also entails contextualising the deployed solutions by integrating indigenous knowledge in the solutions [33].

2.5.3.2. Political sustainability

Political challenges often hinder the progress of ICT projects. It is important for a project to be accepted by the main governing bodies of the community. This was true in the Dwesa project. The Nature Reserve management was first approached to host the VSAT. However, all efforts to make this a success were fruitless. The project later approached the Mpume community which welcomed the project.

2.5.3.3. Technological sustainability

As technology is ever-changing, it is imperative that the technology chosen will serve for an extended period of time. Training the local community can be seen as a measure to ensure that there's technological/technical expertise in the area, which would enable technological sustainability.

2.5.3.4. Economic sustainability

This is the ability of a project to develop economic or cost recovery / sharing mechanisms. This can be achieved by having a project which is able to generate revenue that will surpass the operating costs.

2.5.3.5. Financial sustainability

Even though a lot of emphasis has been placed on the importance of ICTs in supporting development, it seems many initiatives have stalled [7]. Conceding that the initial start up costs have to be footed by either governments, international agencies, NGOs or private companies, very few projects even plan for long-term sustainability, and even fewer achieve it. If a project is not financially sustainable, its economic benefits will be difficult to realize.

2.6. Developmental goals vs. the need to generate revenue

An investigation highlighted that it is obvious that a business telecentre should be financially sustainable [32]. Even though it sounds like a utopian goal in the light of many experiences, developmental telecentres should aim for financial sustainability as well. This is good because it will earn the project self reliance and reduces its dependency on outside funding which often dries up.

As highlighted in [21], projects aiming to bridge the digital divide have been implemented in different forms, most notably cybercafés and telecentres, or multipurpose community telecentres. Each of the different interventions may have different owners. The most common types of owners are NGOs, communities, aid agencies, universities, private companies and governments. Figure 2-7 below depicts the dynamics between generating revenue and achieving developmental goals.

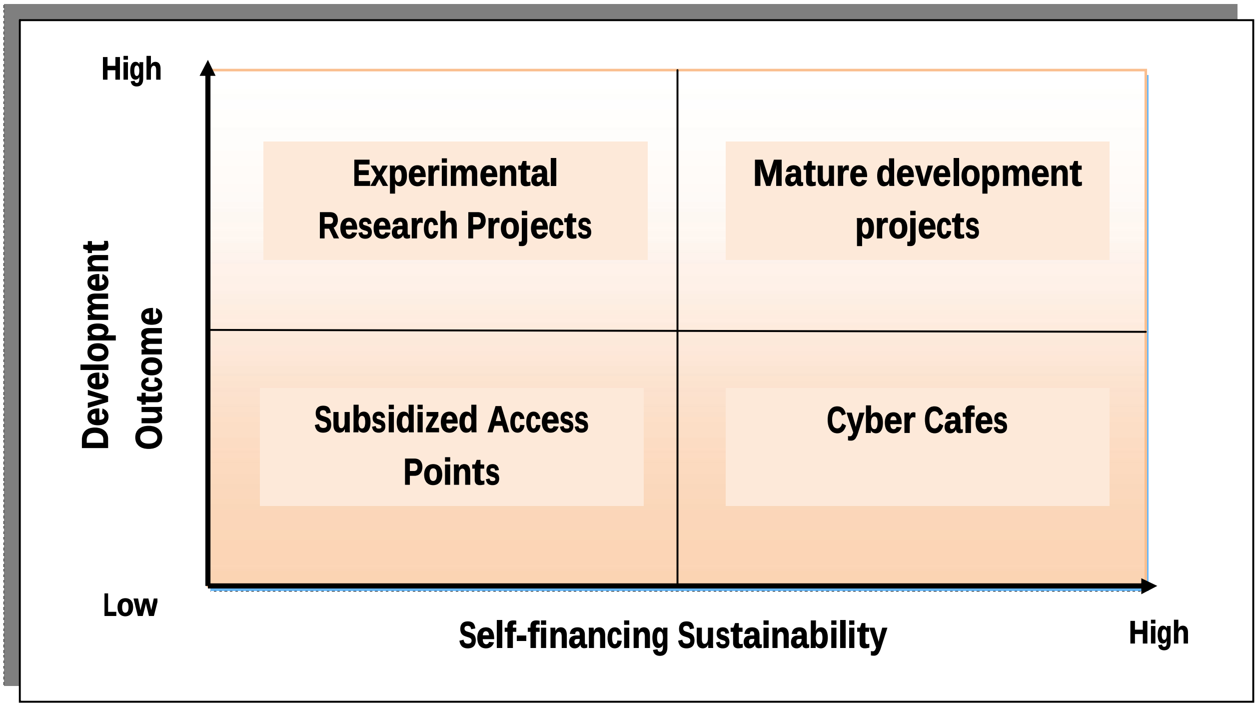


Figure 2:7: Sustainability-Development Matrix (adapted from [21])

It is noteworthy that the more an implementation is required to generate revenue in its initial stages, the less emphasis will be placed on supporting developmental goals, and hence this ultimately resembles a cybercafé. This shows that it is imperative to strike an optimal balance between financial support and revenue generation in order to achieve long term sustainability along all dimensions [21].

This is also significant with respect to ascertaining who benefits and/or pays for the costs and why. While some researchers advocate for donors and /or governments to pay [34], the majority believes that projects are managed better if the owners have some stake in them. They stress that the owners of the project should contribute towards its sustainability [5, 6, 24]. This means that we should implement the "recovery of some of the project costs from the project beneficiaries" aspect of financial sustainability pointed earlier.

2.6.1. Cost recovery

According to [35], the introduction of user charges to finance project expenditures can be one way of recovering costs. This process involves four important issues, namely;

1. The economic effect of the charges,
2. The degree of revenue generation or cost recovery,
3. The application of prices between new and existing users, and
4. The affordability of the prices by different user classes

This form of cost recovery can be implemented through charging users for use of the Internet and other resources on the network. This means that a way to account, charge and bill users for their usage is needed.

2.7. State-of-the-art in charging and accounting

Billing systems allow the generation of revenue for the network services provided. This means that such systems should provide functionality for service provision, tracking, billing plan selection, rating, discounting, invoice creation, payment management and customer management. These systems are the link between the network and the stakeholders [36].

According to [37], accounting is a process which involves tracking a user's activity while accessing network resources. Activities tracked include time spent on the network, amount of data exchanged and services accessed. In other words this is the collection of usage information from different network elements and linking it to the respective user.

In a typical billing process, the next process that follows after the usage details are collected is rating. Rating is defined as the evaluation of something in terms of quality, quantity or some combination of both. In telecommunications this is the process of determining the cost of use of a service. It traditionally involves converting usage-related data (bits, bytes, time, origin and destination, for example) into a monetary equivalent [38].

The rated usage records are associated with their respective users and aggregated. Billing now becomes the process of generating an invoice, which is a collection of the rated or aggregated usage records and making it available to the user.

There has been extensive research on the economics of the Internet and members of the technical, academic, and user communities have shown interest in the areas of charging and accounting. An extensive overview of work on charging and accounting, covering also customer care aspects is given in [39]. It is acknowledged that although solutions for methods of charging and accounting for single service class networks like telephone networks, exist and are successfully applied, integrated services networks require a different approach. According to [40], currently, there is debate in the network billing, user and academic communities on the best way charging and accounting. Charging and accounting protocols, convergence and integration of telecommunication services, incentive-compatible pricing models for Integrated Services networks, an environment for trading communities and open billing systems were highlighted as areas which still needed further research.

This gave birth to number of research initiatives investigating these different areas. Examples include the CATI [41] project, INDEX [42], M3I [43] project and other various reviews which talk about the different pricing schemes for the internet. All these projects indicate that the most fundamental expectations of users are transparency and predictivity of pricing schemes.

The Charging and Accounting Technologies for the Internet (CATI) project was set up to focus on the design, implementation, and evaluation of charging and accounting mechanisms for Internet services and Virtual Private Networks (VPN). Its consists of two main projects, one focuses on Internet protocols to support charging and accounting for existing and future Internet services, and the other demonstrates these technologies with real-world applications [41]. The goals of the first project were to design and implement charging and accounting mechanisms based on the current Internet protocol suite, design and implement a VPN configuration service including accounting and charging functions and also to develop a generic API for Internet-based open e-commerce. The second project aimed to develop an IP phone which makes use of the developed charging and accounting functionalities, investigate and define business models for the Internet and to evaluate the developed charging and accounting mechanisms and business models [41]. This project produced a lot of publications discussing Internet charging and accounting issues defined as the goals of the project.

According to [42], the Internet DEmand eXperiment (INDEX) project was started in 1998 to investigate, amongst other things, the user reaction to the different pricing schemes. Findings from this research indicate that flat fee based schemes were favoured by users. However, the leaders of the project argue that this model in its purest form tend towards a wastage of resources, unfairness amongst users and revenue loss for ISPs [44].

Another project, the Market Managed Multi-service Internet (M3I) was started with the aim of designing, implementing and testing a next-generation system that will allow Internet resource management through market forces, specifically enabling differential charging for multiple levels of service. These capabilities will increase the value of Internet services to customers through greater choice over price and quality. Consequently for the network provider or ISP, flexibility will be improved and management complexity reduced [43].

The main organization in the development of Internet standards, the Internet Engineering Task Force (IETF), has also addressed the issue from different angles by having several working groups working on the different aspects of the subject area. For example, the network group has a sub-group Real-Time Traffic Flow Measurement (RTFM) [45] which defined an architecture and mechanisms for traffic measurement and collection.

The Internet Protocol Detail Record (IPDR) [46] recently came into play to address the same issues in different way. The IPDR was termed after the traditional telecommunications term Call Detail Record (CDR). IPDR promises to be the future standard for exchanging accounting related data between different providers and different accounting systems. The main goal of this project was to enable cost-effective data exchange and usage data management, thereby facilitating the deployment of next generation services [46]. Its reference model defines various layers, some of which are elucidated below.

1. The Network and Service Elements (NSE) layer. This layer consists of all physical devices and systems that are needed to configure these devices. These are the components which specifically provide IP based services to customers.

2. The Business Support System (BSS), which support the business operations that are pertinent to the service provider. Some of these business operations might include billing and customer management functions.

3. Mediation layer is the layer that sits between the NSE and BSS layers. It provides the necessary interfaces to allow these two layers to communicate. It receives usage data from the NSE layer, decodes it and formats it into a format that can be understood by the various BSS functions.

The specification also highlights five attributes that should be present in any usage record. These are highlighted in the table below.

Who	This is the entity which is responsible for the usage on the network. User ID is usually used to identify this entity
When	This deals with the time spent using a network service. It usually uses attributes such as *start* and *end* time or event time.
What	This identifies the service that was used, its usage measures, QoS measures and state information
Where	This deal with the traceability for the network service usage, by looking at elements such as source identifiers and destination identifiers.
Why	Event trigger type - why is the network and service element reporting this data.

Table 2:3: IPDR usage record attributes

After accounting for usage, the next question is how to price or charge the customer. Pricing stands as one of the greatest challenges for the next generation Internet. Even though the success of the Internet is mainly attributed to the flat rate charging, there might be a need to introduce usage based pricing schemes and differentiate services to counteract congestion. According to [40], pricing can be an effective means not only to recover costs but also to permit users to select among different service options in a controlled manner.

2.8. Internet Pricing

Substantive research has been conducted in the area of Internet pricing. This is evidenced by the proliferation of many pricing models, some of which are presented in [5]. [47], articulates that, suitable pricing models represent one of the essential prerequisites for a successful running and implementation of a charging and accounting tool. A couple of dimensions are identified in order to classify, characterize and distinguish different Internet pricing proposals. The most crucial dimensions noted include [47];

- **Service categories:** In this dimension packet-based pricing models and their distinctions are highlighted. These include the Differentiated Services (DiffServ) and Integrated Services (IntServ) approaches. It is also argued that packet based models were the first ones to distinguish between several qualities of services.
- **Pricing components and resulting basic approaches:** Access fees, setup fees and usage fees are the three fundamental elements of traditional telecommunications pricing. Combining these three leads to the classification of pricing schemes into, usage based, flat fee, reservation based, volume based, bandwidth based and so on.
- **Pricing parameters:** This concerns the issue of which parameters should be available for use in accounting and charging.
- **Pricing classification:** In this dimension the basic reason for pricing has to be identified. Pricing can be used to maximize revenue for the provider, control access or maximize user satisfaction.

The pricing models can be put into different classifications and an overview of some of these is given below.

2.8.1. Traditional pricing models

- **Packet pricing:** This is mainly used in packet switched networks. The customer is charged for the packets of data exchanged.
- **Metered pricing:** The subscriber is charged a connection fee, usually on a monthly basis then also charged for metered usage.

- **Flat rate pricing:** The subscriber pays a certain fixed fee then can enjoy all the services that come with the subscription.

According to [48], the proposed Internet pricing models over the past decade can be classified into two major categories:

- Pricing for Best Effort Service. Examples include, Paris Metro, congestion pricing, priority pricing, zone-based pricing and edge pricing; and
- Pricing with Quality of Service guaranteed. The examples include IntServ and DiffServ) pricing schemes.

2.8.2. Pricing for Best Effort Service

Congestion pricing scheme: proposed in [48], reiterates that congestion pricing has drawn a lot of attention in the Internet pricing arena. The use of the network by one user means the time he has used would have been used by some else. Thus we say one user's use of the network imposes negative externalities or social costs to other users. These externalities might be in the form of delays or loss caused to their traffic.. However, when the network is not congested the marginal cost of additional traffic will be zero. In this model the user will therefore pay when the network is congested [49].

Smart market pricing scheme: In this model each packet is padded with a bid field in its header to indicate the amount the sender is willing to pay for sending it. The packet will be allowed to pass if its bid price is greater than the current marginal cost of transportation on each router. The users will pay the bid of the lowest-priority admitted packet rather than their own bid [50].

Shadow pricing scheme: This scheme is applied to model a network in which a resource has the capacity to deal with a specified number of equal sized packets in each time period. Each packet arriving in the overloaded slots is marked and charged a small fixed amount called the "shadow price" and this mark is sent to the users. As congestion increases, the shadow prices will increase thereby forcing users to adjust their usage based on the feedback [51].

Edge pricing scheme: In this scheme subscribers are charged based on the expected congestion along the path of their traffic. The expected congestion is derived from previous usage patterns during different times of the day or other historical metrics.

Congestion discount pricing scheme: This scheme uses prices as incentives to shift subscribers from congested times to non-congested times. When users try using resources during the congested times, they are offered a discount incentive to return after the peak time. They will have an opportunity to accept or decline the offer. Declining the offer means they will continue with their usage as intended but at the high congestion rate [52].

Zone based cost sharing pricing model: In this model [53], advocates for sharing of costs between senders and receivers. To implement the model, [53] proposes an additional field in the IP header to indicate whether either the sender or the receiver is willing to pay for better than best-effort QoS. The Internet is divided into zones in which users choose the zones for which they are willing to pay.

Paris Metro pricing scheme: This models stems from the rail system practise of sub-classing the train into first class and second class. First class customers pay more and enjoy travelling in less congested and luxurious conditions. Odlyzko [54], suggests applying this model to Internet pricing by sub-classing the network into different layers, each with a fixed fraction of the network capacity. The network is expected to route packets using the usual TCP/IP transport protocols, therefore there is no formal guaranteed QoS. However, it is expected that expensive classes will be less congested due to self-regulation, therefore will deliver better QoS.

Priority pricing scheme: In this scheme, subscribers flag their packets with a priority. Packets with a higher priority are the ones which will be processed first, hence lowering their delay in congestion times.

2.8.3. Pricing with Quality of Service Guarantee

Charging for Integrated Services using RSVP: Karstern at al [55] proposed a charging model that can be integrated into RSVP architecture. In this architecture the sender initiates communication by sending a PATH message to the receiver. The receiver then sends a RESV message followed by a PATH message along the same route. If either the receiver or the sender or both have certain prices they are willing to pay, they embed this information in the PATH and RESV messages. This means that PATH and RESV messages are sent to communicate pricing information and establish a contract between the sender and the receiver. RESV messages are sent to reserve resources if the senders and receivers establish a contract. At each hop in the outgoing link, the market price of the sender's requested QoS is added to the price. However, the final price varies in case of dynamic pricing schemes.

Charging for Differentiated Services: This defines an architecture for implementing scalable service differentiation in the Internet by aggregating traffic classification state which is communicated by means of IP-layer packet marking. The architecture consists of functional elements implemented in network nodes, including per-hop forwarding behaviours, a language with which to express per-hop forwarding behaviours, traffic policing and packet classification functions [48].

When evaluating these pricing models, it is clear that flat rate pricing and congestion based pricing schemes are on opposite ends of the continuum. In his visual 3-Dimensional evaluation model, Nguyen et al [48] highlights that congestion pricing maximizes economic efficiency and social welfare. However, the costs of implementing these pricing schemes are very high. Another concern with these pricing schemes is their technical complexity. In contrast, flat rate pricing is easy to implement and has no major implementation costs but provides low economic efficiency and social welfare. Consequently, there is a need to strike a balance between different attributes; hence we say, an optimal pricing scheme is one which trades off between technical complexity, economic efficiency and social well-being.

It is further highlighted that self-regulating schemes such as Paris Metro might not work under competition. This shows that even though there have been so many proposals for economic pricing models, some of these proposals are not practical.

2.9. Charging and accounting frameworks

A lot of work has been done in implementing charging and accounting architectures. Several standardization organizations have developed or are still developing billing architectures in both the telecommunications and Internet area [56]. Groups such as the Authentication Authorization and Accounting ARCHitecture Research Group (AAAARCH) of the Internet Research Task Force (IRTF), Authentication Authorization and Accounting (AAA) Working Group of the Internet Engineering Task Force (IETF), SA5 of the 3rd Generation Partnership Project (3GPP) and the UTMS forum and the SUSIE [57] project are researching on the implementation of robust and flexible charging and accounting architectures [58]. Each of these groups proposed or developed different architectures. The architectures for circuit switched voice services are mature, hence the place where architectures for Internet charging are drawing their lessons. Some of the architectures are reviewed below.

2.9.1. The traditional charging and accounting framework

The traditional view of accounting systems sees the framework as a set of layered components which perform different functions. Upper layers depend on the data produced by lower layers. The ultimate goal is to produce bills which will be sent to their respective subscribers. This framework is depicted in Figure 2-8 shown below [56].

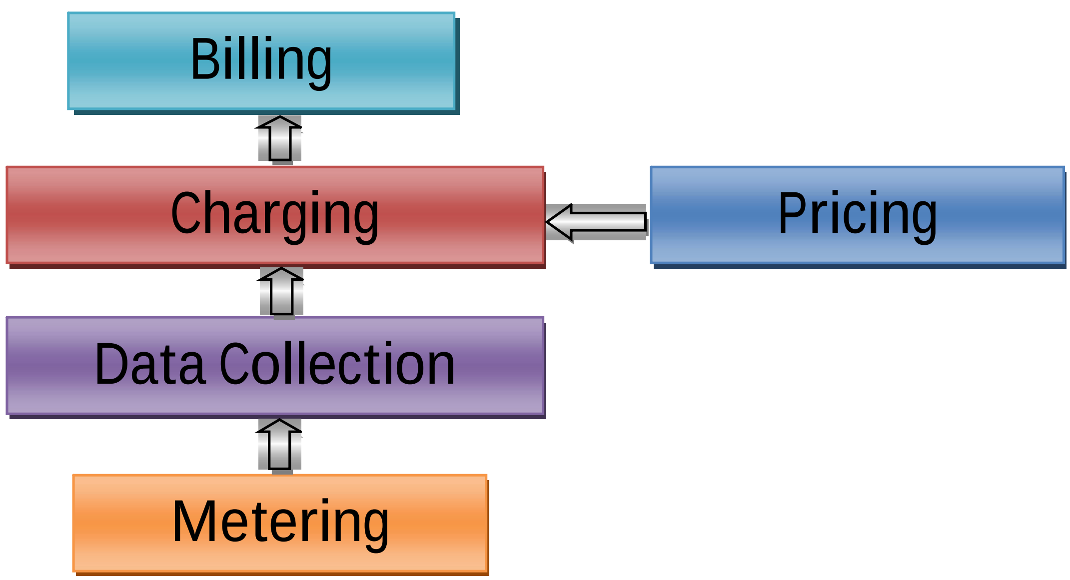


Figure 2:8: Traditional Accounting framework

In this model network activity is metered to collected data which will be used for pricing and charging purposes. When the prices have been applied to the collected data, bills or invoices are made and sent to the subscribers.

2.9.2. Internet Account and charging frameworks

The IRTF Generic AAA Architecture

The Internet Research Task Force (IRTF) is a sister group to the Internet Engineering Task Force (IETF) which aims to promote important research on the evolution towards future Internet. This is done through creating focused, long-term and small Research Groups working on topics related to Internet protocols, applications, architecture and technology [56]. In IRTF work related to AAA is carried out in the AAAARCH research group which is working to define a Next Generation AAA architecture. This architecture depicts high level components which are crucial for AAA purposes.

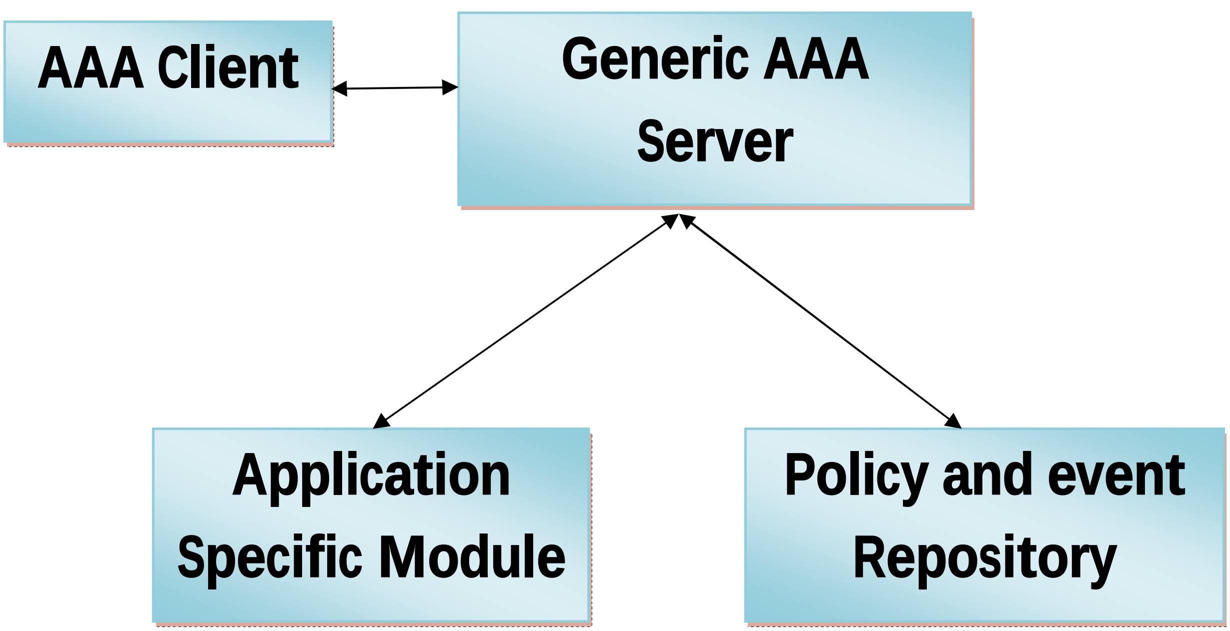


Figure 2:9: IRFT Generic AAA Architecture

In this framework, the user or another AAA server contacts the AAA server to get authorization. The AAA server checks for authorization against the Policy and Event Repository (PER), which is a database containing the available services and resources about which authorization decisions can be made. The AAA server provides an interface to communicate with the Application Specific Module (ASM). The communication between various components is performed using protocols which are yet to be defined. The ASM is the one which is responsible for executing the application specific services [59].

The IETF Real Time Flow Measurement Architecture (RTFM)

The RTFM working group focuses on the development of standardized solutions for metering Internet traffic flows and collecting the metered data for accounting purposes [56]. The RTFM architecture provides a distributed set of network entities that measure traffic flows and build tables of data flow. In this architecture, traffic flows are arbitrary collections of packets specified in terms of their end-point attributes. Figure 2-10 shown below is used to depict this framework.

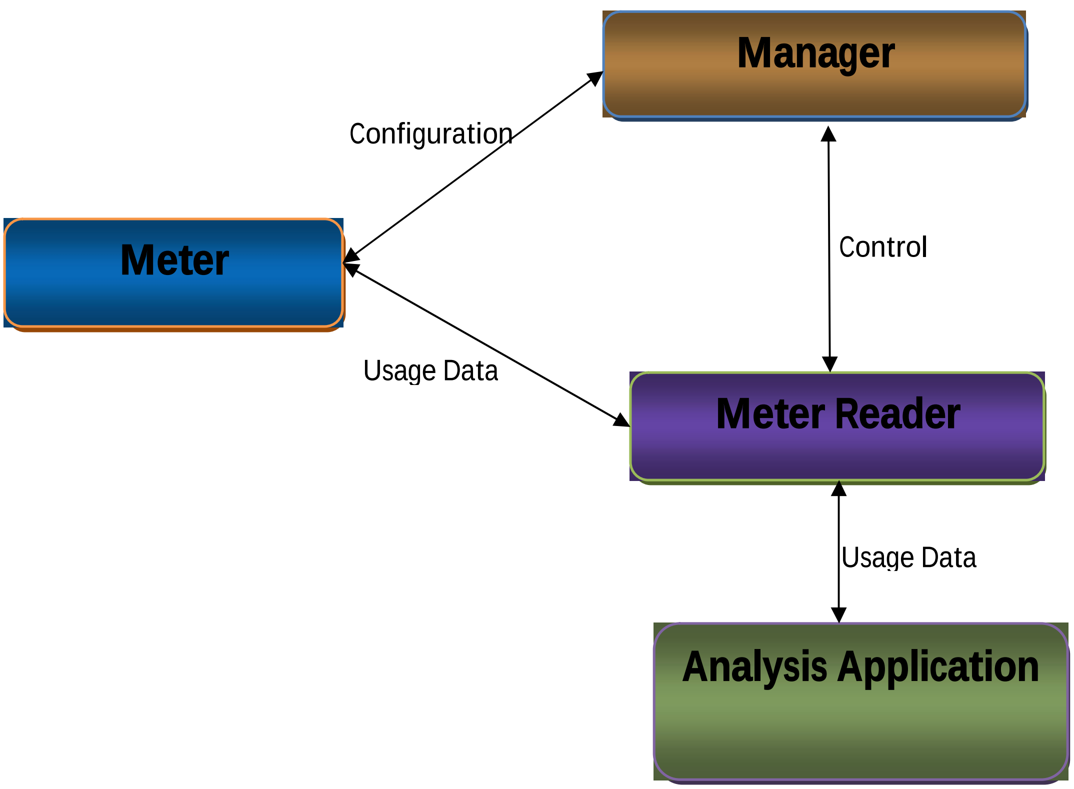


Figure 2:10: Real Time Flow Measurement Architecture

The Authentication, Authorisation and Accounting[1] (AAA) working group of the IETF focuses on developing Authentication, Authorisation and Accounting requirements for network access. Figure 2-11 show the way this group view accounting [60].

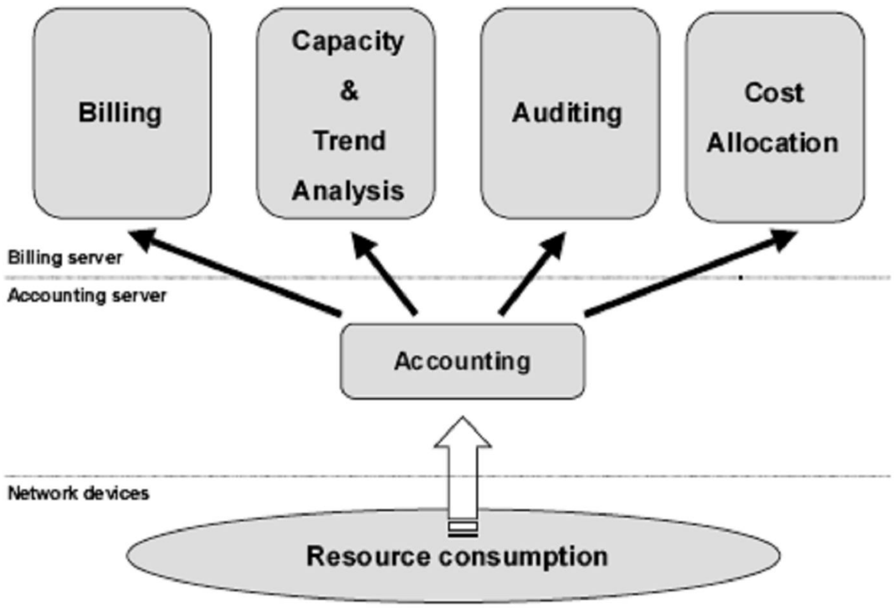


Figure 2:11: View of AAA group

[1] http://www.ietf.org/html.charters/aaa-charter.html

In this framework, meters are placed at suitable convenient places where they capture network activity as per their configuration settings. Meters can as well aggregate and process the data before storing it. This processed data is referred to as usage data. Meter readers are responsible for transporting the usage data to analysis applications. The traffic measurement manager is an application which configures the meters and meter readers. It monitors and controls the operation of meters and meter readers. The analysis application component is responsible for processing usage data so as to perform reporting operations necessary for network planning and management purposes. NetTraMet is an open-source implementation of this architecture. This package employs passive methods collect flow-based traffic information from networks.

Policy-based Architecture from the SUSIE Project

The SUSIE project is a project co-funded by the European Union under the ACTS program. This project looks into the development of a charging approach for QoS-enhanced IP services or Premium IP services provided over ATM networks. This includes the design of charging schemes as well as developing applications which implement those schemes [57]. The architecture is depicted by the diagram below [56].

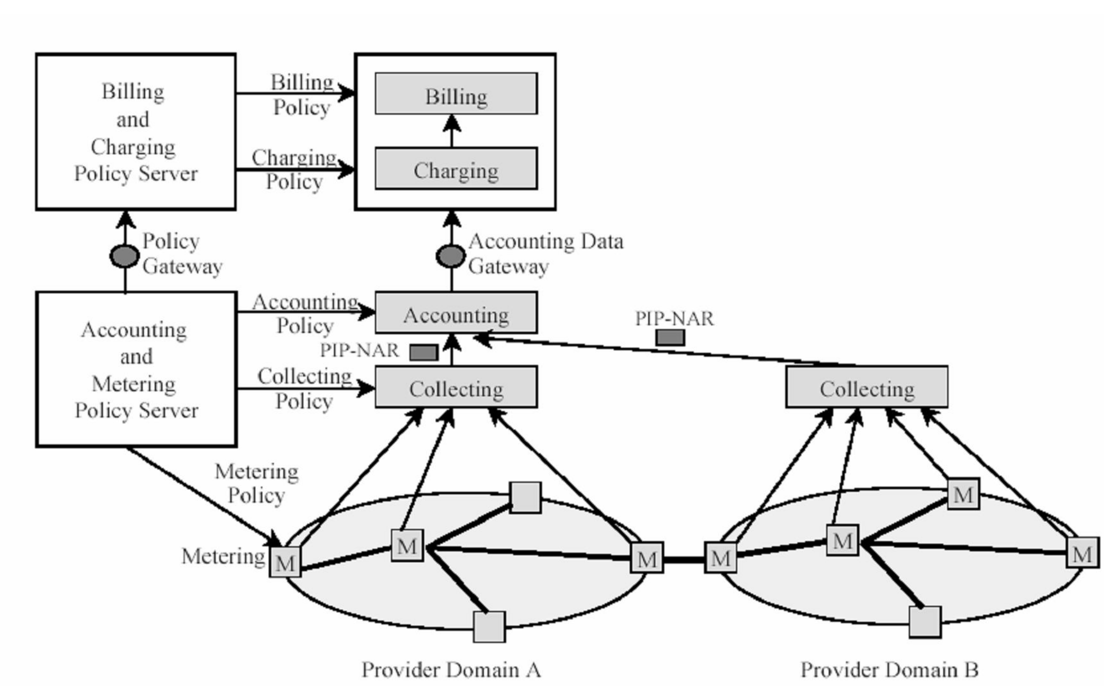


Figure 2:12: SUSIE policy based Architecture

This project developed a reference model for classifying charging, accounting and closely related processes and for describing their interaction. This model can be configured to meet the requirements of both DiffServ and IntServ. The accounting system is divided into five layers: metering, meter reader, accounting, charging and billing layer. Each layer is linked to a policy. Figure 2-13 depicts this relationship [56].

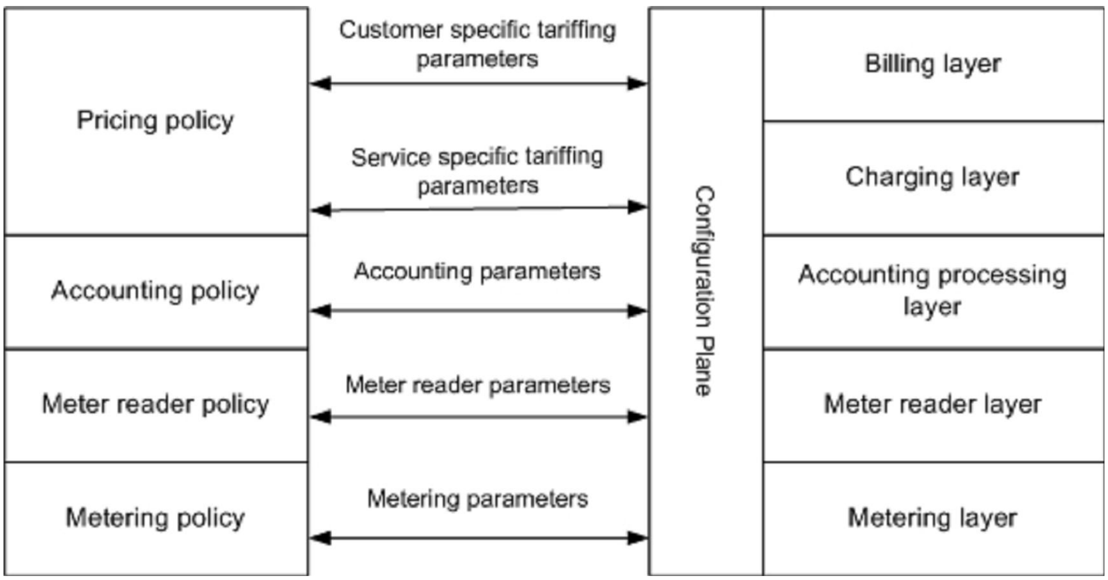


Figure 2:13: SUSIE reference model

The metering layer measures resource usage. It allows the distinction of two types of two resources: reservation of network resources and actual usage of network resources. The meter reader layer involves all the elements that access data from the metering layer and forwards it to the Accounting processing layer for further processing. The Accounting processing layer processes the data from meter readers consolidates it based on service parameters and then creates accounting data sets which will be forwarded to the next layer. The charging layer derives charges for accounting data sets based on service specific tariff parameters. The billing layer translates the calculated costs into monetary units and generates bills for subscribers.

Other architectures

The list of charging and accounting architectures is a long one. Among these architectures are [56]:

- Cisco NetFlow [61] and
- The GigaABP Architecture [62]

2.10.Conclusion

Projects aiming to bridge the digital divide have mushroomed across developing countries. However, conceding the fact that setup costs would have been met by another party which in most cases is a non-governmental organization, a university, a government or a private company, financial sustainability remains a big challenge facing these projects. A lot of literature discusses this problem from a theoretical point of view. This is done through defining the digital divide in different contexts and the various kinds of sustainability that are associated with the deployed projects. However, there is limited literature proposing or discussing practical solutions to the problem.

The Internet community proposed a lot of pricing schemes. However, most of these schemes are not practical and some are too complex to implement hence we have not seen of lots of implementations of the proposed charging schemes. It is evident that these pricing schemes came as a result of disparate efforts by different members of this community. There are no defined standards for pricing telecommunication services. Hence where there are no standards people will just find what works for them and stick to it.

In the area of charging and accounting architectures, a lot of work has been done by various standardization bodies and projects, hence the maturity that is demonstrated in the area. Considering the fact that these architectures were created by different organizations or projects for different purposes, there are some variations in their nature. Yet some components and aspects are common to all these proposals.

Using the information and lessons from these diverse fields it is possible to take what we want and use it to define our charging and accounting architecture which shall be referred to as the Network Revenue Management Architecture. This architecture highlights the functional components of a system that will be used to collect revenue and share costs in the Dwesa project.

Traditional billing metrics such as usage based metrics might not encapsulate the local communities' idea of fairness of cost sharing, which could in reality be associated with socio-economic factors. Therefore, it is imperative for this system to also include non-conventional billing metrics.

Chapter 3

Methodology and Requirements

3.1. Introduction

This chapter answers the questions of *who* will use the system, *what* the system will do, and *where* and *when* it will be used. The approach adopted in implementing the Network Revenue Management System described in this thesis was based on a variety of software development methodologies. This chapter describes how I applied and tailored these methodologies and best practices in the implementation of the Network Revenue Management System.

3.2. Evolutionary prototyping

Considering the facts that the project was complex, involved various research disciplines and had special target users, it was impossible to understand all system requirements and come up with an all-inclusive system requirements specification in the early phases of the project. There is another project in Dwesa which aimed at looking at what are the best requirements elicitation methods in the context of projects like Dwesa where traditional methods might not be appropriate [63], but its results are not yet available. My project required a systematic and thorough investigation to discover and introduce new requirements. This made this project a good candidate for evolutionary prototyping development methodologies.

When developing a system using evolutionary prototyping, the system goes through a continuous cycle of building, deployment and testing, and refinement. In this paradigm a prototype is built in a structured manner and is allowed to evolve through testing and refinement [64].

Following this paradigm, an easily modifiable and extensible working model of the proposed system, which provided a functional representation of key parts of the system before implementation, was implemented. This prototype enabled me to visualize and gain hands-on experience with the system. The prototype also allowed me to experience the system from a user's perspective and ultimately generate more requirements. Hence this prototype became a mechanism for identifying the actual system requirements. The process followed in this research can be illustrated in Figure 3-1 shown below.

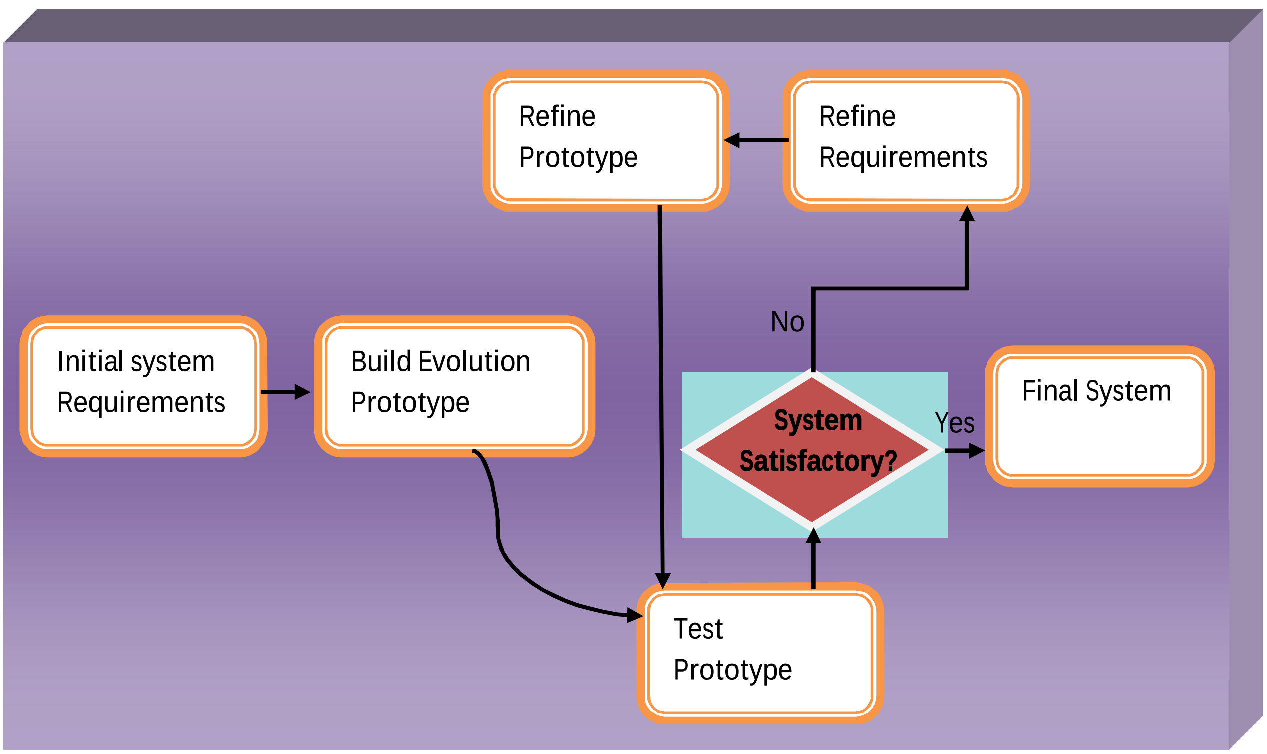


Figure 3:1: Rapid Prototyping Development cycle

The figure above demonstrates that in this model, the specifications, design, implementation, evaluation and testing phases are inter-twined and the system is developed as a series of increments that are evaluated at each stage. As shown in the diagram, a spiral model of rapid prototyping development was adopted. Using iterative or rapid prototyping enables the most crucial components of the system and its business functionalities to be scoped through a series of back and forth reviews.

It should be noted however, that not all requirements were possible to prototype. Examples of these include non-functional specifications which will be discussed later in this chapter.

3.3. Domain analysis

The first step was to perform an investigation on the domain of the project. This was an important prelude to other processes. This investigation made it possible to acquire sufficient knowledge in the subject area. This was achieved mainly through:

- Discussions with my supervisors and mentors;
- Internet searches and
- Library searches.

My discussions with supervisors suggested that although there are several ICT projects being implemented in developing nations, most of them have been implemented as open access networks, hence did not include revenue generating/cost recovery/sharing systems.

The literature survey confirmed that a reasonably substantial amount of work has been done in areas related to bridging the digital divide, Internet charging schemes and charging and accounting systems. However, it was found that there is not a lot of literature around the subject of revenue generation and cost recovery/sharing systems in ICT4D projects.

This domain analysis activity continued throughout the project.

3.4. Requirements elicitation

The Dwesa research team visits Dwesa once every month. For each visit it stays there for about one week. During these visits the community and teachers are trained how to use the ICT resources deployed and also handle whatever troubleshooting that needs to be done on the network. It was during my first visit when I got myself acquainted with the topology of the network deployed there [3]. The roles of the different stakeholders in the project were also investigated. As highlighted in the Dwesa network topology diagram, at Mpume Junior Secondary School there is the VSAT which provides backhaul connectivity to the internet. The internal connectivity between the different schools in Dwesa is a point-to-point WIMAX local loop. The WIMAX base station is hosted at Ngwane Secondary School which routes all Internet traffic through to the VSAT at Mpume Junior Secondary School.

The major operational costs in this network are the VSAT and electricity costs. The contract options with Telkom, which is the service provider of VSAT, are summarized in Table 3-1 shown below. The option which was chosen for the project is highlighted.

TELKOM USER PACKAGES							
Users will have a choice of one of the following five access packages:							
Access Medium	Receive Data Rate (kbps)	Trans-mit Data Rate (kbps)	Monthly Volume Caps*	No. of IP Address	Internet /Data	Voice Lines	Access Price Vat inclusive
Space Stream Express	64	16	500 MByte	4	Yes	None	R998.00
Space Stream Express	128	32	1 GByte	4	Yes	None	R1244.00
Space Stream [redacted]	256	64	2 GByte	8	Yes	None	R1824.00
[redacted]							
Stream Office			MByte			to 4	
Standard Installation							**R3100.00**

Table 3:1: Telkom VSAT contract options

Electricity costs vary from school to school ranging from R40 – R200 per month. The variation in electricity costs largely depends on the type of electrical equipment used.

I conducted preliminary data collection through several informal interviews via emails and personal conversations with the stakeholders. Questions asked in these interviews where mostly related to sustainability and cost sharing issues.

The main purpose of the preliminary interviews was to generate information on how the community has sustained its other projects, how they have managed to contribute towards community projects or other community functions or activities, how they view the methods of contribution used and how they keep track of their contributions.

The interviews also reviewed some novel revenue generation mechanisms within the community, especially at the schools where the infrastructure was deployed. For example, the schools were charging community members to re-charge their cell phones. On average, a 3 rand fee was charged per phone. The proceeds were then used to buy more electricity, while the surplus was used for buying electrical equipment and for other purposes.

Besides the interviews, observation was another technique which was used to gather information pertaining to the system. This process involved observing the community's utilisation of the infrastructure. This included observing things such as the community's usage patterns for electricity and computers. This technique also provided an effective way of gauging the responsiveness of different social groups to technology and how social dynamics played a role in the project. Observation also enabled me identify the different stakeholders in the project.

Data was also collected from the servers to access the usage patterns of the Internet by the communities.

The information from this phase was analysed and the results of the analyses became the basis for the initial systems requirements.

3.5. Requirements analysis

As mentioned before, the Network Revenue Management System is a system which will handle charging users of the network. The revenue generated will be used to cover the costs of running the network and other associated costs.

The system will provide easier control of users' access to and usage of network resources by abstracting the underlying operations and processes required to perform these tasks. It will also provide the users with a pool of pricing options from which they may choose. The system will encapsulate context-sensitive and unconventional pricing metrics in its charging schemes.

It was anticipated that this would enable the exploration of different pricing models to determine what is regarded as a fair way of charging in the rural marginalized communities of developing countries.

This meant that decisions had to be made on how each requirement highlighted was to be handled. A divide-and-rule approach was taken. Based on other charging and accounting systems reviewed in chapter 2, we broke and categorized the system into various components, each performing a specific task. These involved identifying groups of related activities and assigning them to a particular component. The results of this phase are summarized in Figure 3-2.

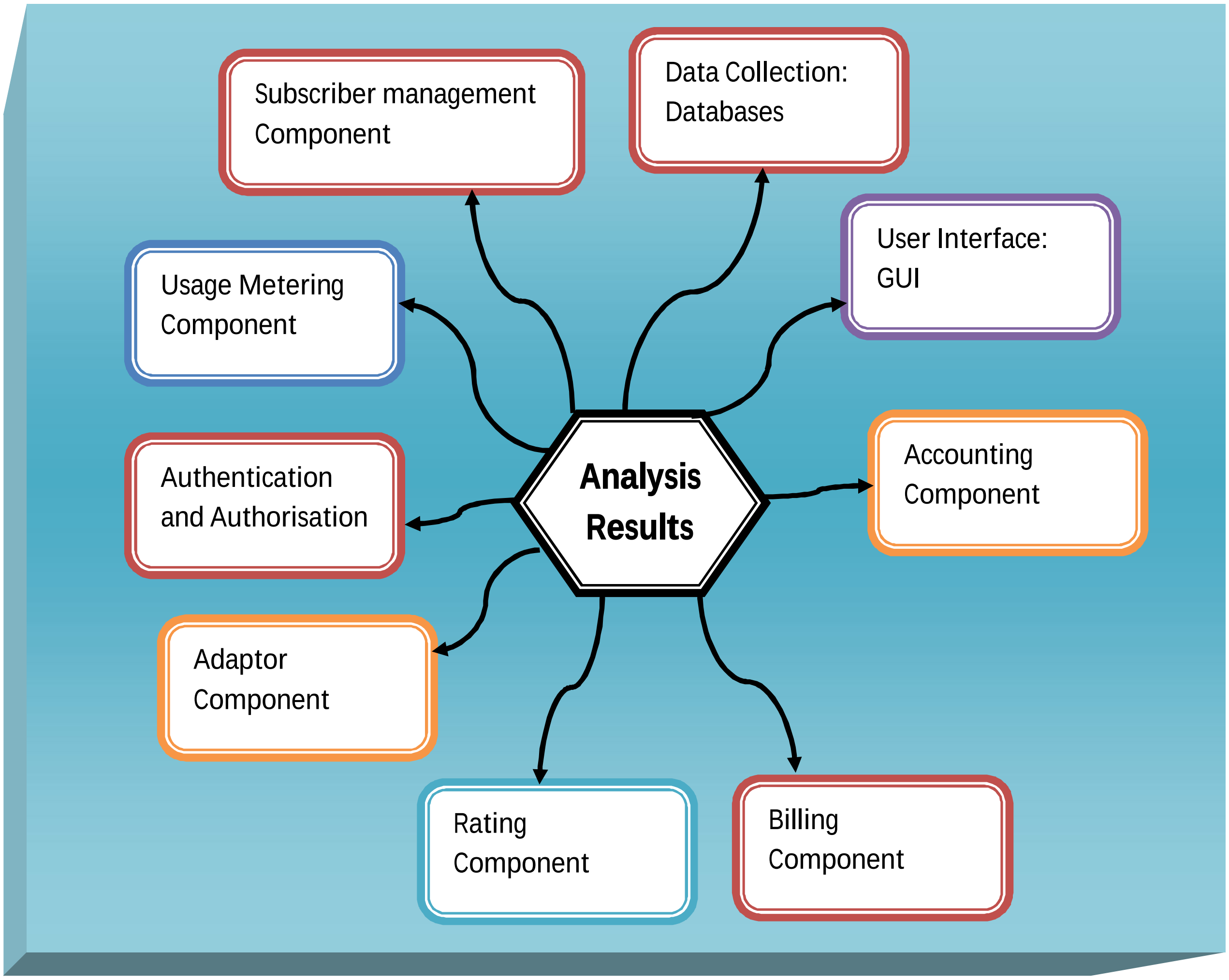


Figure 3:2: Analysis results

We need a way of knowing who the users (subscribers) of the network are, so we need a way of registering and managing our subscribers. A subscriber management component should be one aspect of the system. After the users are registered they would use the system. A way is needed for us to authenticate and authorise them as required. As they are using the system, it is also important for us to track their usage and assign it to their profiles (accounting). There is also a need for storing different kinds of data that are of importance to the system. For this purpose databases are needed. A billing component is needed to extract the different data stored in databases and manipulate it to produce invoices for users. Decisions need to be made also on the type of user interfaces to adopt.

McEwen notes that, *requirements are not requirements unless they are written down. In other words, neither hallway conversations nor "mental notes" constitute requirements* [65]. This means that this phase culminated into the requirements specification document. The complete document is contained in APPENDIX A. Here below some of the requirements identified are elaborated.

3.6. Overall requirements

This section highlights important points of the system's services and constraints.

3.6.1. Definitions, acronyms and abbreviations

Access Control

Access Control is any mechanism that a system uses to grant or deny the right to access some data. Normally the users first identify themselves to the system by issuing an authentication token (usually a username and a password) and the system then verifies the claimed identity (Authentication).The access control mechanism controls what the user may or may not do on the system. In any access control model, the entities that perform actions are called subjects, and the entities representing resources to which access may need to be regulated are called objects. Access Control systems provide Identification, Authentication, Authorization and Accountability [66].

Authorization

Determines what the subjects can or not do on the system. It can be defined as a process of determining the privileges which can be granted to a particular user. These privileges define the resources which the user is allowed to access and the actions he is allowed on those resources [66].

Accounting

Determines what a subject did. It is concerned with the collection of information on resource usage. This means that this is a process of identifying all that was done by subjects by using components such as audit trails and log files. The usage details are then linked to the controlling user [67].

Billing and rating

For a better understanding of the system, the distinction between rating and billing needs to be clarified.

Rating is defined as the assessment or evaluation of something, in terms of quality, quantity or some combination of both. In telecommunications this is the process of determining the cost of usage of a service. Traditionally, it involves converting usage-related data (for example, bits, bytes, time, origin and destination) into a monetary equivalent [66].

The rated usage records are associated with their respective users (Accounting) and aggregated.

Billing is the process of generating an invoice which is a collection of the rated or aggregated usage records and sending or making it available to the user.

Free and Open Source Software

Free and open source software (FOSS) is software that gives users the privilege to run, copy, distribute, study, change, and improve it as they wish, without them having to ask for permission from or making additional payments to anyone [68].

3.6.2. Assumptions

At this stage, it will be assumed that the system will have two kinds of users, administrators and general users. General users are the subscribers of the network. These are the people from the community or visitors who use the network for various purposes. Administrators are responsible for administering subscribers. These maybe selected teachers from the schools or appointed community members.

3.6.3. System features

The system will be based on a client server architecture. The servers will be placed on various strategic locations on the network. Clients send request to servers for services like DHCP, AAA, Netboot, HTTP, HTTPS and VoIP. The responsible servers will handle the requests, perform specified actions and then send back the responses to the requesting clients. The system will be developed using PHP, Perl, java script and bash as languages of choice. To avoid reinventing the wheel, FOSS applications will be used where appropriate.

3.6.4. User characteristics

The user of the system should be capable of interacting with computer software easily. For the client side this is fairly easy, because it is a matter of submitting authentication, checking current balances and usage statistics. In order to facilitate ease of use on the administration side, the user will have to be familiar with billing operations. If needed, the user should be trained on various networking aspects such as bandwidth and time as measures of usage.

The system should be made such that it is intuitive for the users, so while there is still a need for some basic training on the parts of the users, the system should be intuitive enough as well and contextualized to the users profile.

3.6.5. Design and Implementation Constraints

Since the system is has a client-server architecture, it will require a web server capable of handling PHP applications and SSL/TSL, a RADIUS server, and an access controller. Since the network deployed in Dwesa runs on a Linux platform, the system implementation is also based on the same platform. Given the fact that the system is a subsystem of the ecommerce platform project, it needs to be integrated with other systems to minimize the maintenance of duplicate data where possible. In particular input from other systems should be done by common subsystems, to reduce the burden of data entry on the end users.

3.6.6. Functional requirements

3.6.6.1. Authentication infrastructure

Authentication is a two step process to determine who can log on to the system and the association of users and their software subjects. This means that a client wishing to access a system submits its identity along with a set of credentials as proof of the supplied identity. The supplied credentials are then used by the system to verify if the identity actually belongs to that user [67].

The authentication modes that need to be supported can be subdivided into two main classes [67].

i. A mode accommodating users that arrive at a fixed location, such as the LAN with fixed devices already connected to the network. In this case users simply need to log on to the network using their credentials.

ii. A mode accommodating mobile users carrying their mobile devices to gain access to the network and be able to use its resources. In this case both the user device and the user have to be authenticated on the network.

As an illustration of mode (i), Figure 3-3 is used. In step 1 the user tries to access the Internet which is a resource outside his LAN. Since the user is not yet authenticated, he is redirected to a login page using page redirection techniques. The user then supplies his authentication credentials which are then checked against a backend as shown by step 2. This backend could be any number of services such as LDAP, Active Directory, Kerberos, a database or UNIX password files.

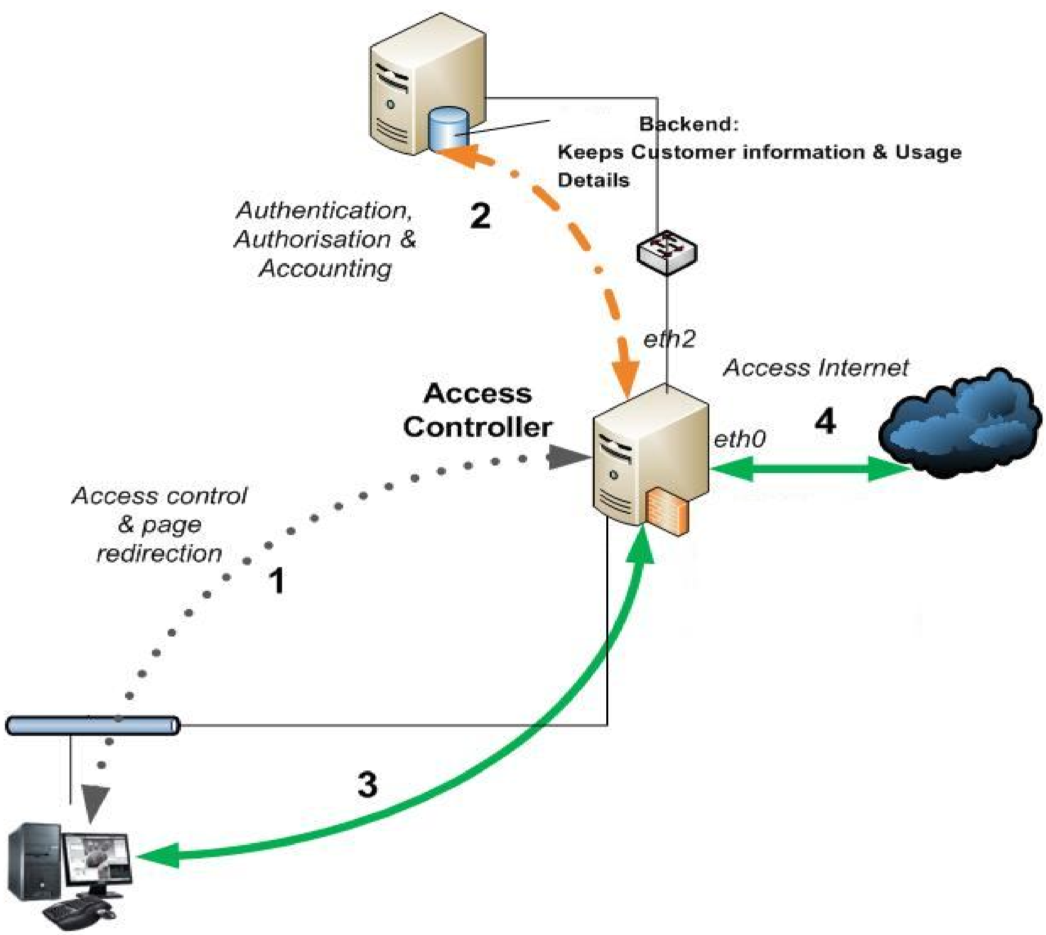


Figure 3:3: Access control and page redirection

A basic scenario for mode (ii) is when a user sets up his wireless device to join the network. This is usually performed through associating with an access point and issuing a DHCP request to the network DHCP servers. The server will respond with a specified DHCP offer. After receiving an IP address the client is then able to access services like HTTP and HTTPS as long as it has been authenticated.

Three-Party Authentication Model

In this model, the various elements that participate in the authentication process are identified and classified as three parties as shown in Figure 3-4 [67, 69].

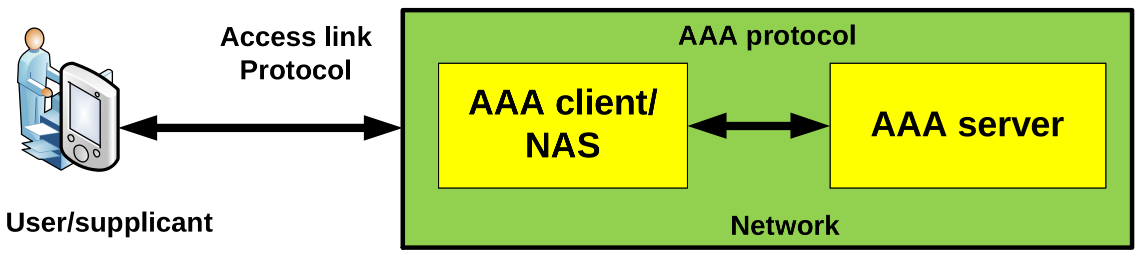


Figure 3:4: Three-Party Authentication Model

1. The supplicant

This is the user trying to gain access to network resources. In our case these are teachers, students and the community users.

2. The Authenticator

This is the device which is at the edge interacting with the user. This party has no authority by itself. It is like the security guard at the entrance of a corporate company who asks authorities whether he should let in a particular visitor or not. This entity is called the network access server (NAS) and it acts as an AAA client.

3. The Authentication server

This is the device which has real authority and the necessary information to make decisions regarding granting access to specified users. This role is played by the AAA server. The sequence diagram show in Figure 3-5 summarises the roles played by each part in this scenario.

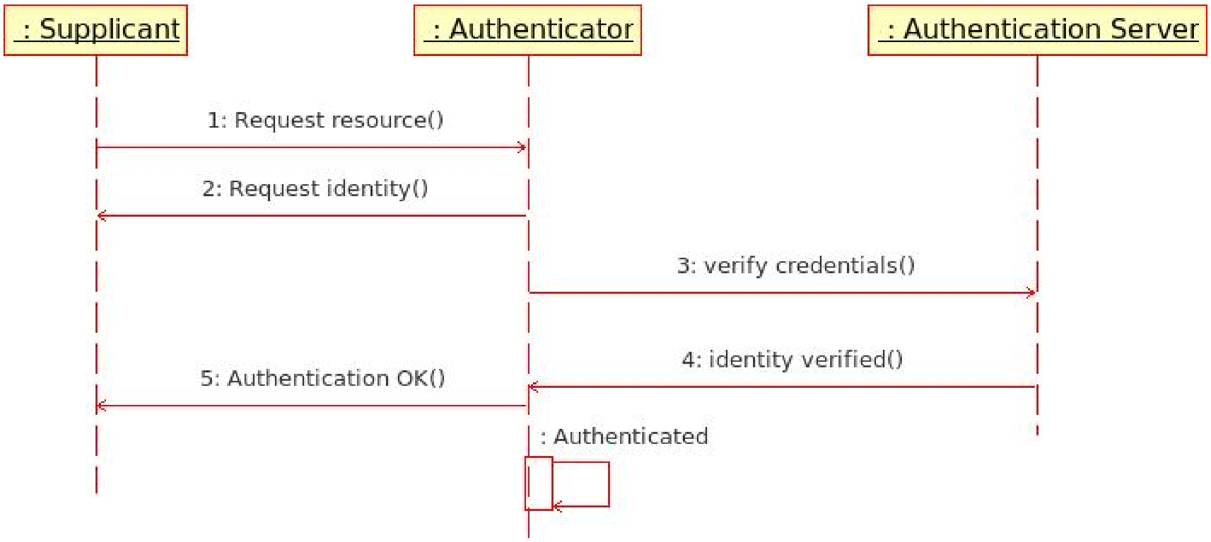


Figure 3:5: Three-Party Authentication Model sequence diagram

Authentication Mechanisms

Authentication mechanisms are based on data matching techniques. They describe how to authenticate users or devices [66]. Some of these mechanisms are:

- **Password**

This is the oldest and the most common authentication method. A password is something that you know and can be several characters long.

- **One time password**

This is an improvement of the traditional password mechanism to guard against sniffers and eavesdroppers sitting somewhere in the network to obtain authentication information that could be used later in replay attacks.

- **Challenge / response**

This mechanism thrives on a shared secret which is used to authenticate different parties to the network.

- **Other mechanisms**

Other mechanisms include the Public Key Infrastructure (PKI) and biometric techniques. An overview of authentication mechanisms can be highlighted as shown below.

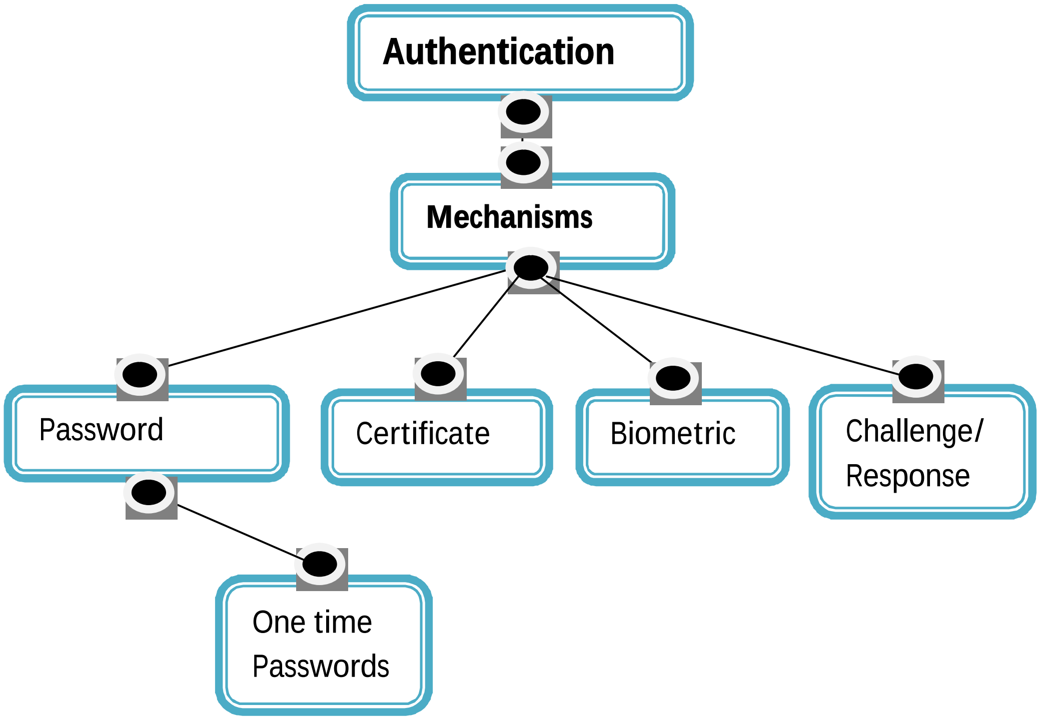


Figure 3:6: Authentication Mechanisms

Authentication Protocols

There many protocols available for use in the communications between the different parties involved in an authentication session. These range from simple text based methods to more advanced authentication against DIAMETER servers. Some of the protocols that might be implemented in this system include:

1. Password Authentication Protocol (PAP) is a simple Link Control Protocol in the Point to Point Protocol suite, which provides a simple method for the peer to establish its identity using a 2-way handshake. PAP transmits unencrypted ASCII passwords over the network hence, it is considered insecure [70].

2. Challenge Handshake Authentication Protocol (CHAP) is a three way handshake protocol that is considered more secure than PAP [70].

3. Extensible Authentication Protocol which supports multiple authentication methods and is used between a dial in client and a server to determine what authentication method will be used [71].

4. Remote Authentication Dial In User Service (RADIUS) is an AAA protocol for applications such as network access or IP mobility. It is specified to work in both local and roaming situations [72].

3.6.6.2. Non-conventional billing options

One of the most important requirements of the system is to allow the billing of users based on non-conventional billing options. The motivation for the inclusion of non-conventional billing metrics is that usual metrics (e.g. usage based metrics) might not encapsulate the local communities' idea of fairness of cost sharing, which could in reality be associated with socio-economic factors. There are financial systems already that factor in socio-economic considerations. In South Africa, the government helps disadvantaged students with study finances through the National Research Foundation. This institution makes use of a technique called the "means test" to determine the amount a particular student is legible to receive from their coffers and the amount that is expected to come from his/her parents' or guardians' income. This is a formal financial assessment method which has been applied by many institutions in different parts of the world for many years. In the UK it is used in the health system. Applying similar techniques in Internet billing in marginalised communities will enable us to assess the degree of need of each member and offer him discounted or subsidised packages when appropriate. Of course, this is just a basic example. The system should also allow for the implementation of evolving billing metrics/schemes, i.e. should provide a pluggable billing modules interface.

The following are scenario for a few different ways of billing. They do not want to represent the space of all possible billing methods, but just give some ideas of what could be involved, to help design an appropriate interface into which plug-in modules that capture non-conventional metrics.

Usage scenario 1

A user comes to use the Network resources, but because he is poor or in financial difficulty, we want to charge him based on his income. The system needs a way to incorporate or extract the information which will be needed to make this billing decision. In this case, the user's income details should stored somewhere where they will be readily available to the system.

So, the system should provide the functionality for capturing the various data which will be used for non-conventional billing options. Examples of the possible non-conventional billing metrics include:

- Billing based on percentage income spent on food or groceries;
- Billing based on one's age;
- Billing based on household income;
- Billing based on number of dependents;
- Billing based level of competence with computers and so on.
- Billing based on amount spent on beer;

Usage scenario 2

There is someone who is on our network using our system to make money or run a profitable business like arts and crafts e-business. How can we make such kind of people contribute towards the sustainability of the project? The system needs a way to calculate and evaluate the proceeds from sales. After that, mechanisms should be implemented so as to determine how much should be charged or deducted from the entrepreneur's earnings in a manner that is considered fair both to the entrepreneur and the rest of the community, based on the value he derives from the network.

Or we can use the opposite point of view: let us give discounts to those who are making use of the system for entrepreneurial purposes as this will encourage all the other community members to use the system to generate revenue and improve their socio-economic well-being.

This means that the system needs ways of communicating with the various e-business applications to collect the appropriate information.

Usage scenario 3

Ever since the deployment of the project, the community contributed money towards several project requirements. An example includes contributing towards the purchasing of printers. The system needs a way of recording all such things and be able to use it for selectively charging it users. In this scenario for example, we have two users whom for the sake of this scenario we will assume have the same socio-economic level, that is, they have the same income and the same expenses, the same number of dependants and so on. In such a case, the system may charge more the one who has contributed less to the project. In this way we will be encouraging all the members of the community to participate and contribute equally where possible.

3.6.6.3. Other functional requirements

The section gives a summary of important requirements for the system.

Number	Functional requirement
1	The system should allow mutual authentication between the NAS and the AAA server
2	The system should keep usage activity data in a database
3	The system should allow subscribers to check their balances and usage statistics
4	The system should allow the user to end his session when he has finished using the resource
5	The administrator should be able to administer subscribers (registration and management)
6	The system should allow administrators to view and manage subscriber usage activity records
7	The system should allow the administrator to configure system wide settings
8	The system should allow the creation of prepaid cards
9	The system should provide conventional and non-conventional billing options and billing plans
10	The system should allow the administering of invoices, bills and payments

Table 3:2 : Functional requirements

3.6.6.4. Use cases

According to [73], in the mid-1980s, Ivar Jacobson put forward the idea of usage cases and usage scenarios. Recently these have become to be known as use cases and use case scenarios. A use case is simply a reason to use a system. For example, you need a cell phone to make calls, receive calls, send an SMS and so forth. In this context you are known as the actor, the cell phone is the system and what you can do with the cell phone are the goals. Therefore, use cases describe how a particular user utilises a system to achieve a particular goal. A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is achieved [74]. Use case diagrams are used to identify the primary elements and processes that form the system. These diagrams can be used to further illustrate the functional requirements of the system. The use cases will be presented from the view point of the subscribers and administrators, which we identified as the two main actors in our system.

Subscribers

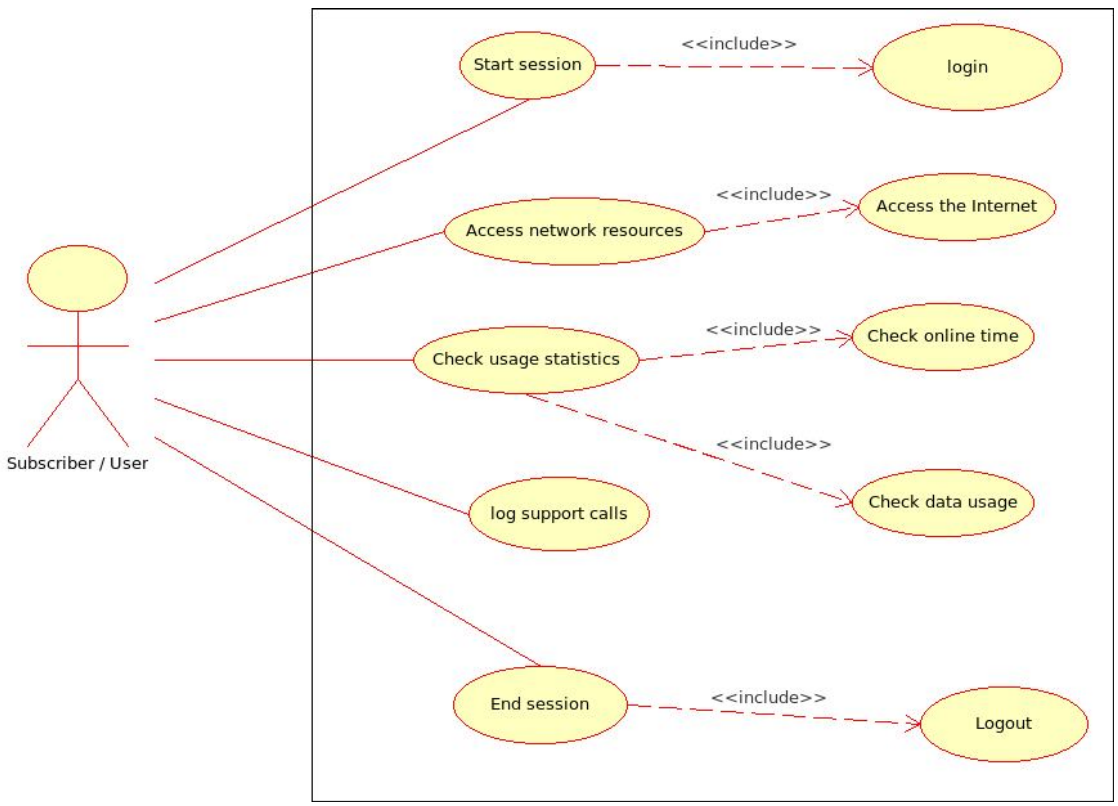


Figure 3:7: Subscribers' use case diagram

The diagram above shows the various requirements that need to be satisfied for the user to use the system. Users have to be able to interact with the system to get access to network resources.

Administrators

On the admin side, the administrators should be able to perform all the tasks that are necessary for the management and smooth running of the network. These activities and interactions are summarized in the use case diagram shown in Figure 3-8.

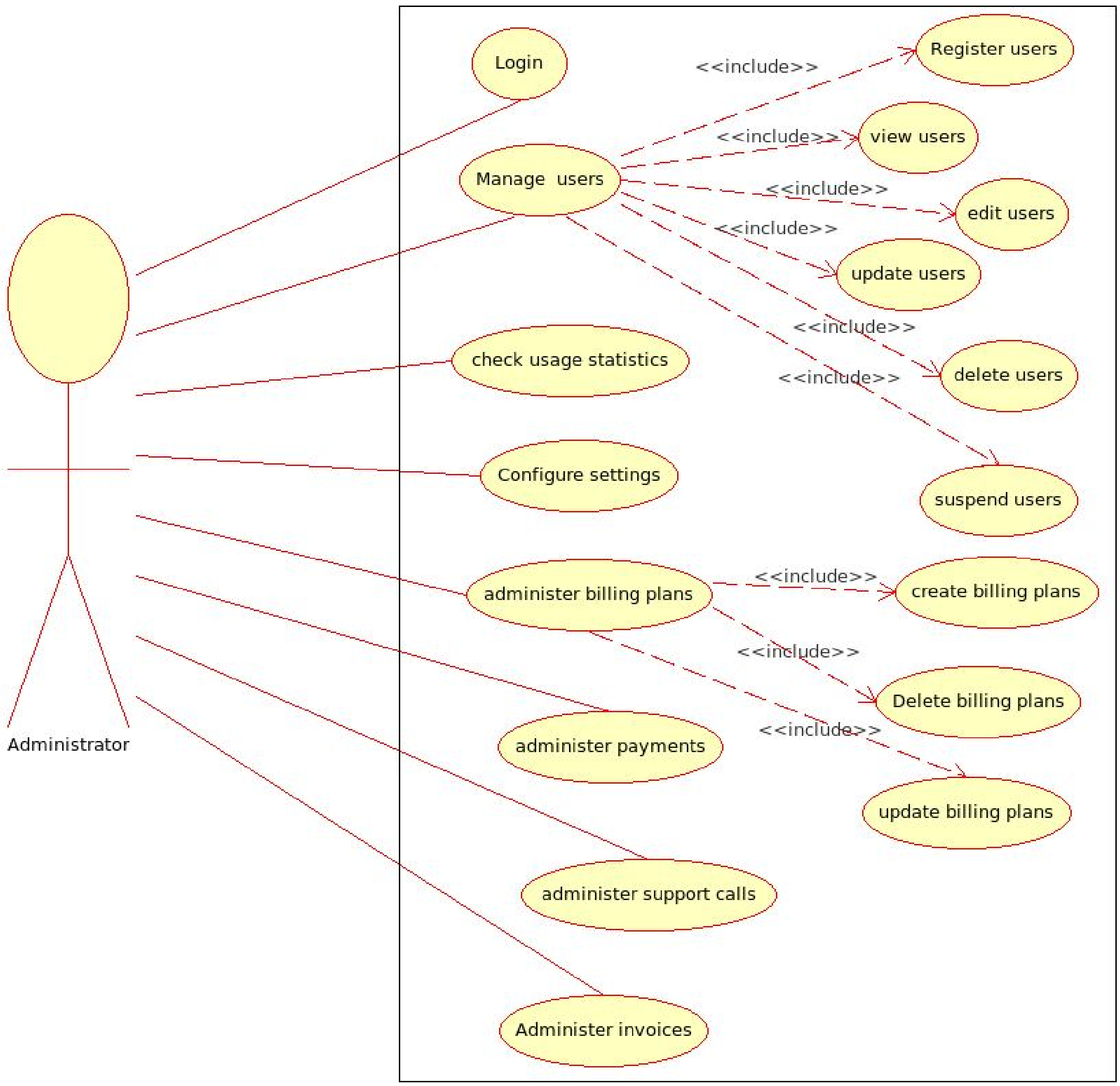


Figure 3:8: Administrators' use case diagram

3.6.7. Non-functional requirements

Non-functional requirements are about constraints and qualities. Qualities are properties of the system that the stakeholders are concerned about and hence will affect the extent of their satisfaction with the system. Constraints are normally not subject to negotiation and, unlike qualities, are off-limits during design trade-offs [66]. Some of the constraints and qualities are highlighted below:

Scalability

The system should scale gracefully. Scalability is a desirable property of a system or a network, which indicates its ability to either handle growing amounts of work in a graceful manner. It is the ability of the system to maintain its availability, reliability, and performance as the amount of traffic load increases. This means it should be able to handle more subscribers and increased traffic load.

Extensibility

Extensibility is a design principle which takes into consideration the future growth. This means that the system should be developed with hooks or mechanisms for enhancing the system, making the addition of new features or components easier.

Flexibility

User requirements are always changing. Flexibility is the ability of the system to change in response to different user requirements. If today their favourite billing option was income, after a year they might want to bill based on things such as number of cows. The system should be easy to adapt to these requirements.

Security

Security is all about confidentiality, integrity, availability and non-repudiation. The system should ensure that information provided by subscribers and other system information is not disclosed to unauthorized processes or persons. The system should maintain the correctness and consistency of the information it stores.

Redundancy

The system should be designed without single points of failure. This is achieved by deploying redundant servers which will be failover systems to take over when one of the systems breaks down. This means we should have redundant database and authentication systems.

Roaming

The system should allow subscribers to roam from one location to another. This enables users to register at one school and still be able to access network resources from any other school which is in the same network.

Low cost

Given our target users, the system should low cost to ensure that it is affordable for the users. This can be implemented by the use of free and open source software as no license or software acquisition fees need to be paid.

3.7. Conclusion

In this chapter the steps which were taken to gather user requirements were discussed. A blend of the iterative development models were applied in the actually realization of the system. As a result, the system was built and refined through a continuous process, even if this report's organization does not reflect it fully.

The identified requirements are both functional and non-functional. Authentication, accounting and billing are identified as the major requirements of the system. More importantly the system should allow for billing models which are based on non-conventional charging methods. These options should be provided as optional functionalities to augment the traditional billing options.

Hopefully, this chapter has given a deeper understanding of the problem and at the same time provided a foundation for the design phase.

Chapter 4

Design

4.1. Introduction

After the analysis phase, the project proceeded to the detailed design phase, where the requirements were translated into a representation of the system. In this chapter we present an overview of our approach to the design and implementation of the Network Revenue Management System. This emphasises a conceptual solution that fulfils the system requirements. We begin by restating the system components identified and analysed in chapter 3. We then clarify the architectural design of the system. Thereafter, we elaborate on the different components of the system and how they interact with each other. Finally we describe the billing engine which meets the important requirement of contextualised or custom billing modules.

4.2. The design scope

In this project, on account of time and human resources constraints and in line with current practice, we tried not to reinvent the wheel wherever possible. So, the system is derived from and designed to take advantage of currently available applications and technologies. Specifically, the system makes use of different open source applications to implement the functionality requirements of components where appropriate.

4.3. System Architecture

IEEE Standard 1472000 [75], the IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, defines architecture as

the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.

The standard also defines a system as *a collection of components organized to accomplish a specific function or set of functions*.

The architecture design process established a basic structural framework for the Network Revenue Management System. The model established in Chapter 3 will now be explored to include definitions of each of the components or subsystems. Figure 4-1 shows the architecture of the Network Revenue Management System.

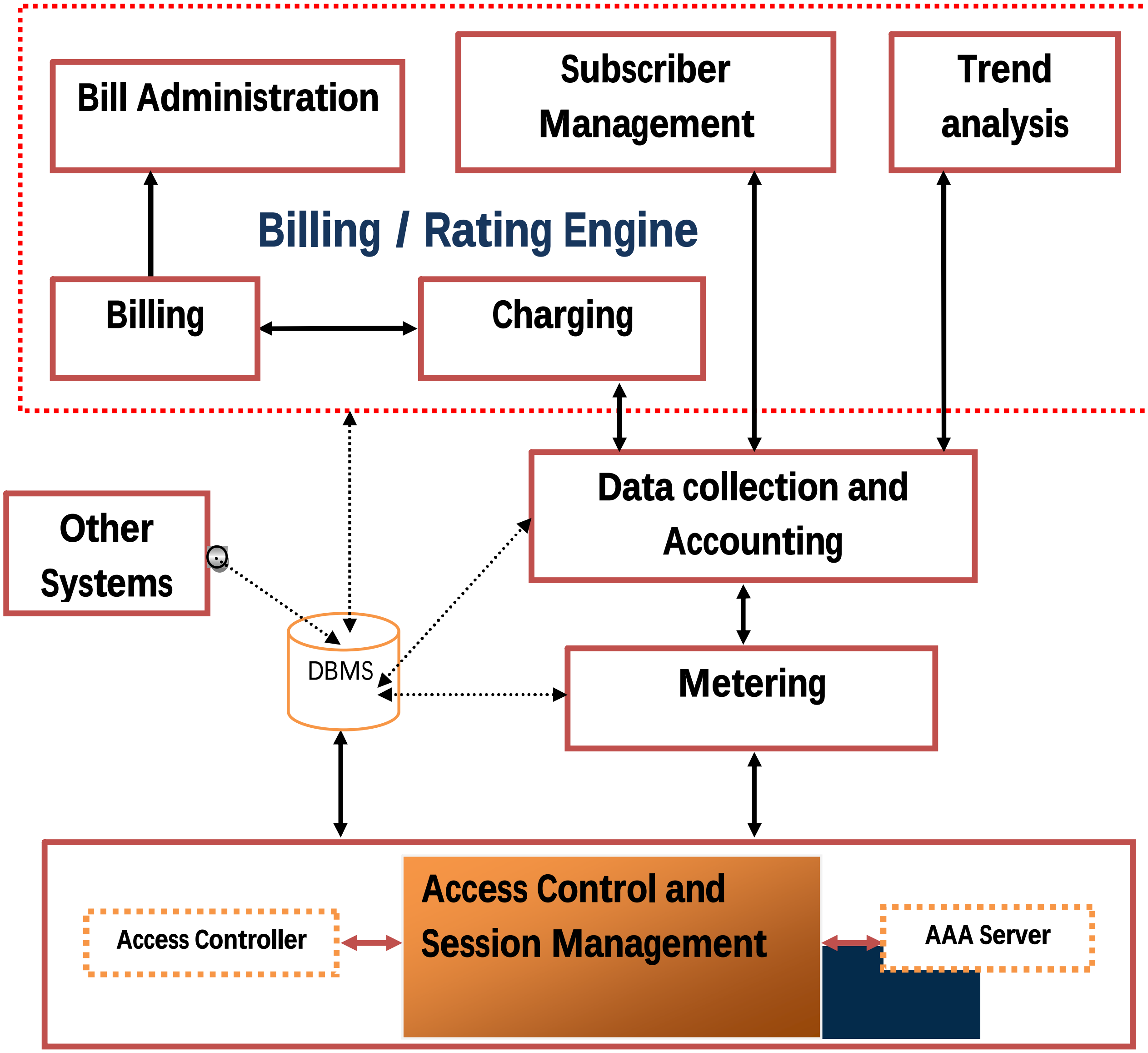


Figure 4:1: Conceptual architecture for the Network Revenue Management System

4.3.1. Access Controller and session management

This component addresses the functional requirement of the authentication infrastructure. Some of the tasks that are performed by the access control and session management component are illustrated in the sequence diagram shown below. This component acts as a proxy that sits between the user and the AAA server. In the three-party authentication model highlighted in chapter 3, it is partly the supplicant and the NAS or AAA client. It accepts the user credentials and forwards them to the AAA server which verifies them against a specific backend. Once the user's details have been verified, the AAA server returns information elaborating what the user is allowed to and not to do. At the end, when the user has finished using the resource for which is authorized, or his session has timed out, the session is ended.

4.3.2. AAA Server

This is the application server that deals with the user request for access to network resources. It provides the authentication, authorization and accounting functionalities. As highlighted in chapter 3, the AAA server is the device which has real authority and the necessary information to make decisions regarding granting access to specified users. It interacts with AAA clients or NAS using an AAA protocol. The Remote Authentication Dial-In User Service (RADIUS) protocol is the current standard used for such purposes. The AAA server also interacts with directories, files and databases containing user information.

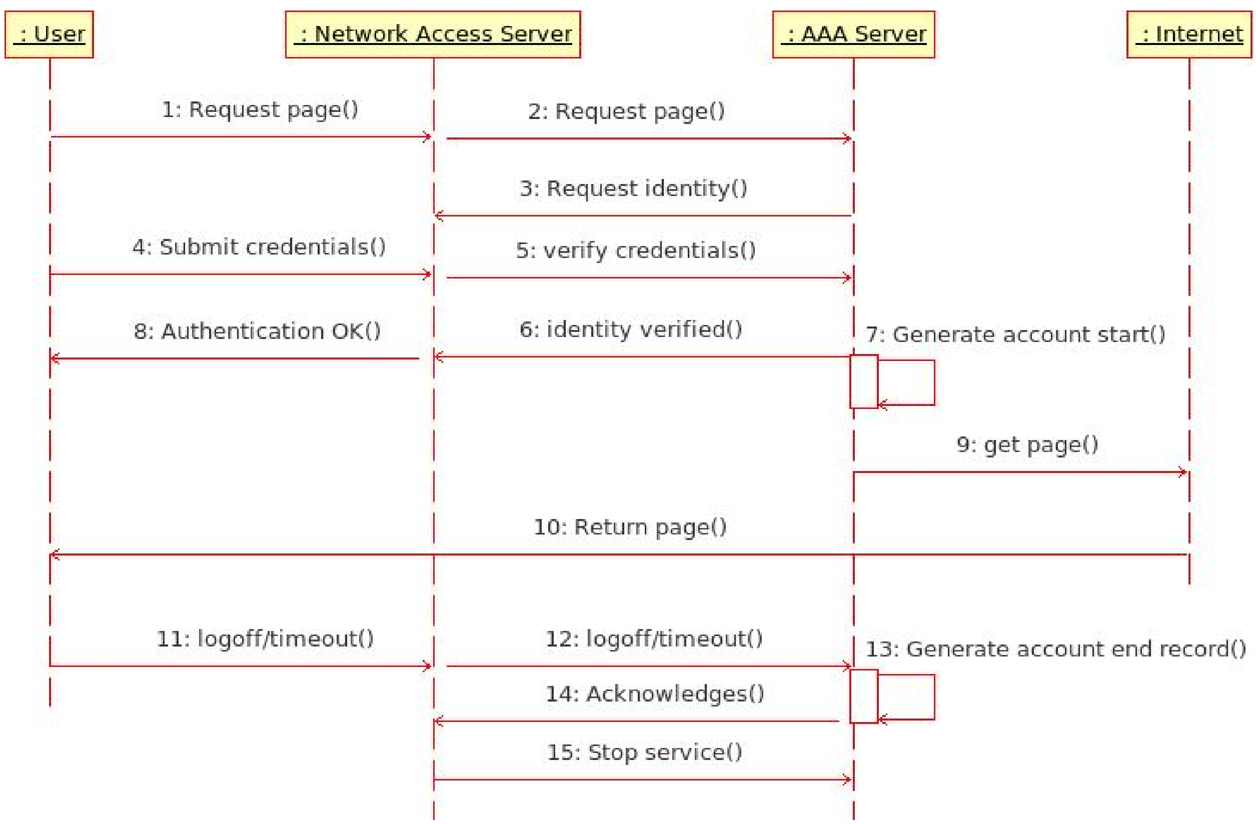


Figure 4:2 : Authentication and Access control sequence diagram

4.3.3. Data collection and accounting

Each user session produces one or more records describing the activities that took place during the life of the session. The data collection and accounting component is responsible for formatting and aggregating the captured network activity data, assign it to the respective user and then store it in a database. The records of this database can then be manipulated by upper layers to perform specified actions.

4.3.4. Other systems

These are other systems that form part of the Dwesa project, whose data will be used for billing and other purposes by the Network Revenue Management System. Examples include the Dwesa e-commerce portal. This system is an e-commerce system which allows the local people to sell their art and craft artefacts online [76] and in future will support micro-tourism initiatives.

4.4. Billing engine

This component abstracts the technical details and manipulates the data coming from other systems as well as generated by the system to display different kinds of reports to the user through a web interface. Its primary responsibility is to calculate and assign prices to records, based on different billing metrics or inputs. Some of the rating engine's inputs are depicted in the diagram below.

The engine gets its inputs from different places. The obvious inputs are the outputs of the data collection and accounting component, whilst the less obvious ones are ones which have been specified as non-conventional billing options in Chapter 3. These might include income, number of dependents, history of contributions, proceeds from sales and others.

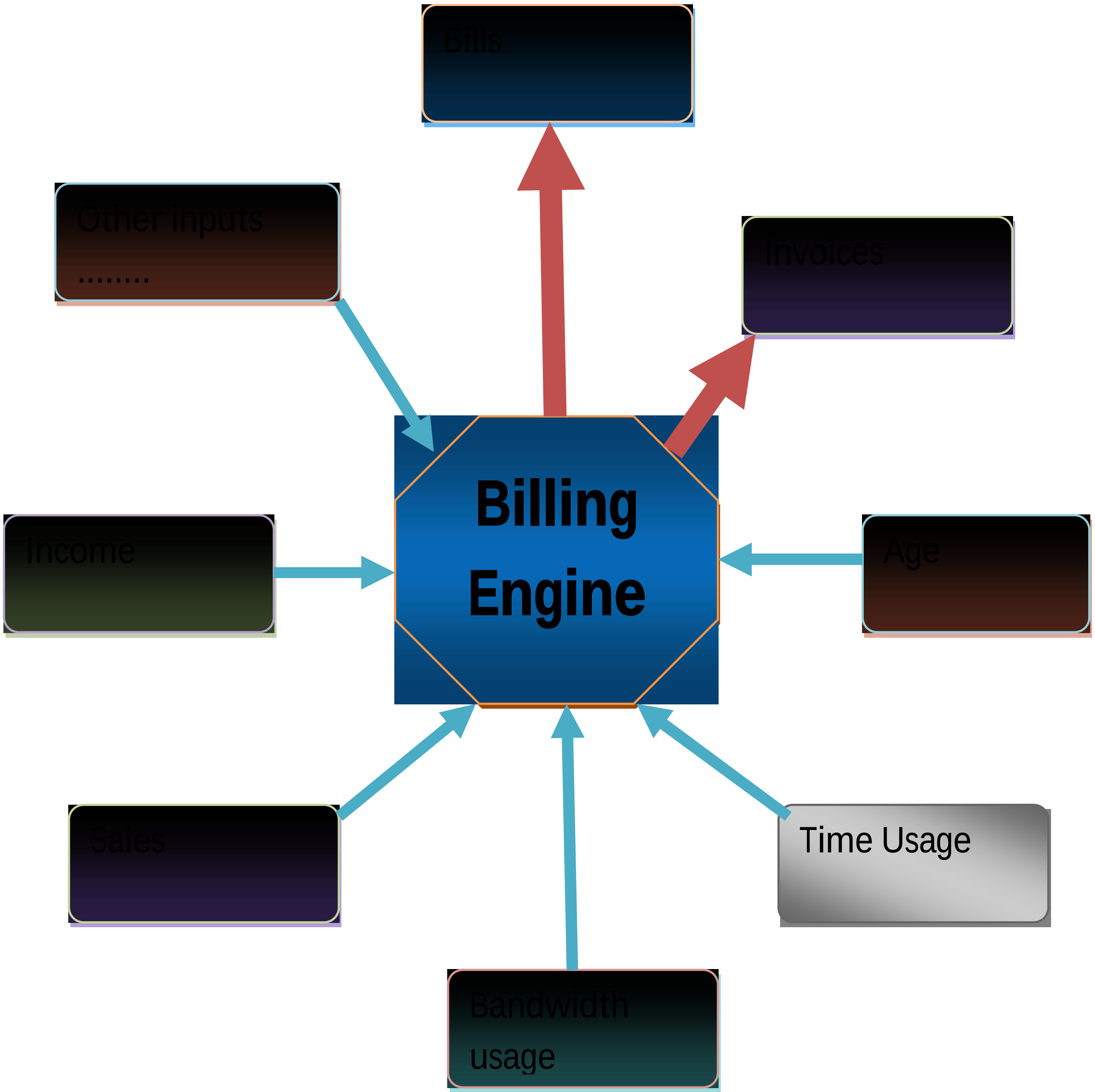


Figure 4:3: Billing engine inputs and outputs

4.4.1. Model View Controller design pattern

Systems which contain jumbled presentation code, business logic code and data access code are usually difficult to maintain. To avoid such problems, the well known Model View Controller

(MVC) design pattern was chosen as the underlying framework for this system. According to Burbeck [77], the MVC pattern separates the modelling of the domain, the presentation, and the actions based on user input into three different categories.

In [78], Deacon highlights that any application is likely to evolve its interface as time goes by, but the underlying application might remain fairly constant. The unchanging parts of the application are identified as the ***model***. These consist of the set of classes which model and support the application. According to Burbeck [77], the model manages the behaviour and data of the application domain, responds to requests for information about its state and responds to instructions to change state.

The interface is identified as the ***views***. These consist of a set of classes which gives visual results, for example GUIs/ widgets and command line interfaces.

The ***controller*** handles and processes user events and instructs the model and or the view to change accordingly. It provides a single point of entry through which requests for resources pass. It also provides interfaces between the model and the views.

A bird's-eye-view into the billing engine is given in Figure 4-5 shown below.

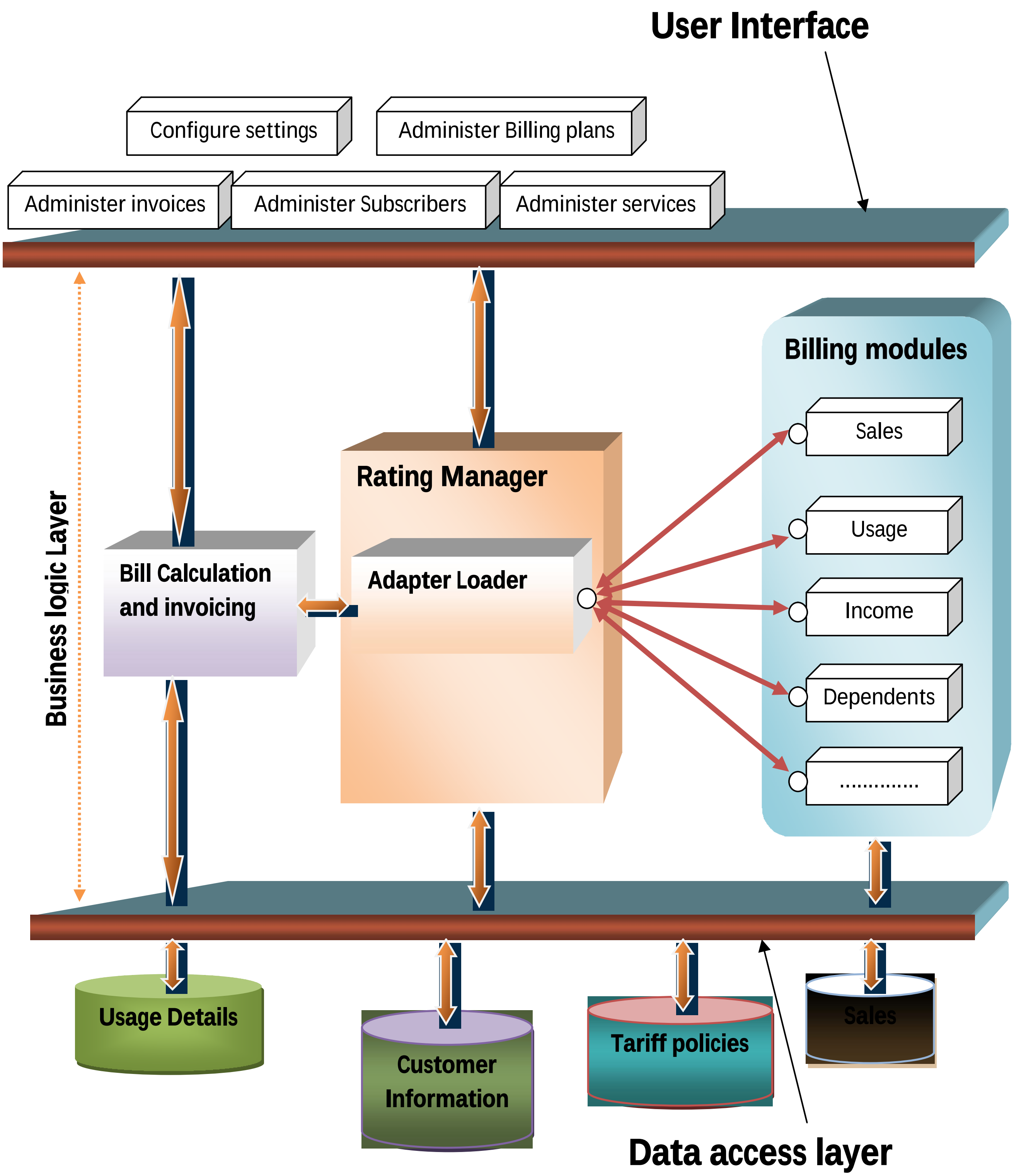


Figure 4:4: Conceptual architecture for the Billing engine

The conceptual architecture highlights the structure of the MVC architecture where the different application concerns are separated. The different application layers collaborate to provide sufficient functionality for the Network Revenue Management System.

4.4.2. Data Access layer

Data access layer provides interfaces for interacting with data stored in databases. This layer consists of classes that represent business data and the business rules that govern access to and updates to this data. This is where the data models are defined and implemented. These data models represent the different relationships of the data stored in the database or XML files. The diagram below highlights the database model to be manipulated by this layer.

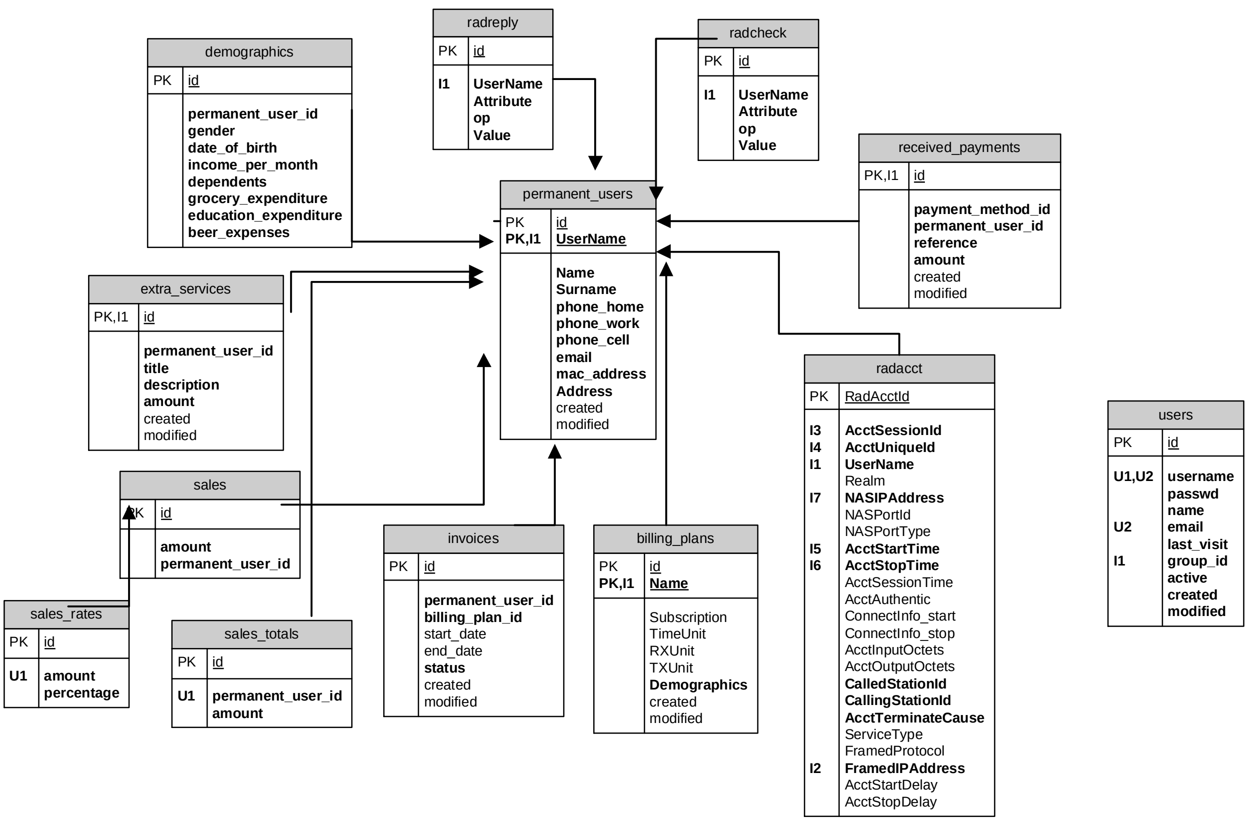


Figure 4:5: Database Model

The permanent user entity is created when a user comes to register to use the Network. His demographics details are also captured at that moment. There should also be a table to store the details of a special system user, the system administrator. The process of registering users can be depicted by the following sequence diagram.

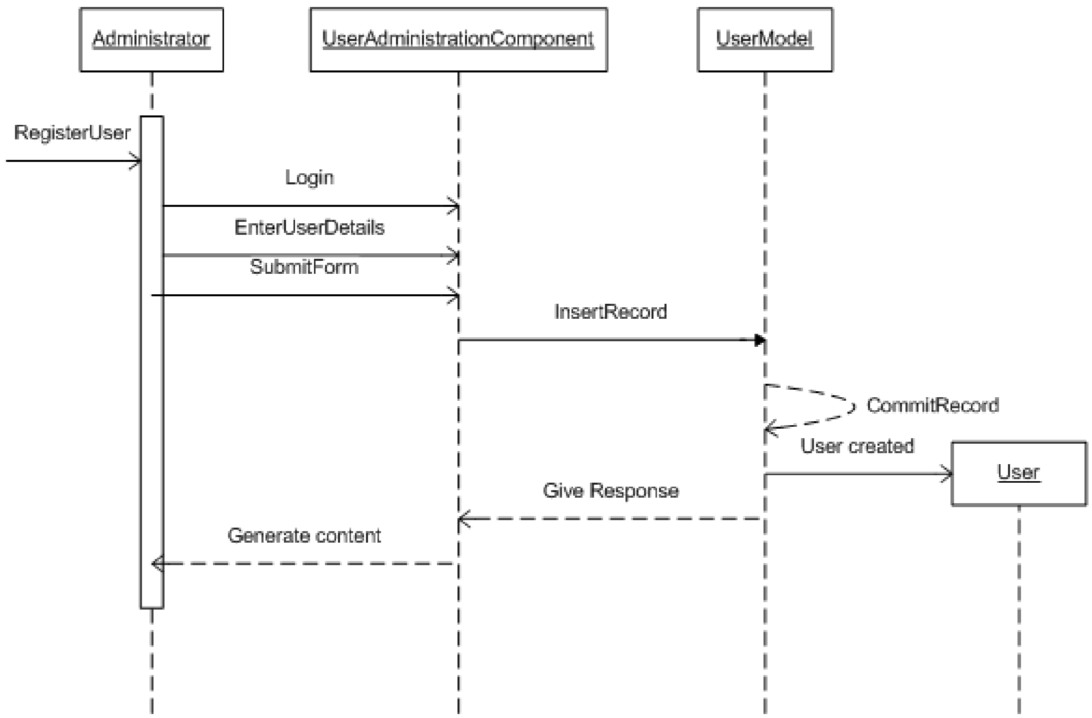


Figure 4:6: User registration sequence diagram

In this database model we should also have tables for storing data from other systems such as *sales*. The *radacct* table is used to store various usage related information such as accounting start and stop information and octets in and out. This means that the model becomes a set of classes that abstracts the various database operations and will have methods to select, connect and disconnect from data sources. This is highlighted by the diagram shown in Figure 4-8.

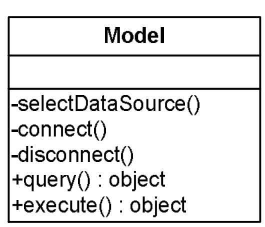


Figure 4:7: Structure of models classes

4.4.3. Business logic layer

The business logic layer consists of classes that are responsible for performing business logic operations. They handle and process user requests and send the output back to the user. They do this by calling appropriate models and generating appropriate dynamic content on the views. Classes that are of importance to this research are those that enable the introduction of non-conventional billing options into the picture. In our architecture this is implemented by the rating manager component which has the capability to load the different classes which perform the non-conventional billing. Figure 4-9 highlights the structure of such classes.

Figure 4:8: Rating manager and billing modules classes

The business logic layer also consists of methods which handle the Create, Retrieve, Update and Delete (CRUD) operations on behalf of the user. These are mainly methods to create, update, delete and view the data stored in the database. For instance, the billing and invoicing component consists of methods which provide the ability to administer invoices.

4.4.4. User interface

Just like any other system, a good user interface is essential for the success of the Network Revenue Management System. This layer consists of HTML files which create the user interface for the user. This includes menus, buttons, text and other graphics that are displayed and meant to interact with the user. This layer provides the user with the ability to login, configure system wide settings, administer subscribers, administer billing plans, and administer payments and so on.

4.5. Conclusion

The system design phase defined the architecture of the Network Revenue Management System. The system components and subsystems were put into context and in addition to this; their interactions and relationships were identified and defined. This was followed by a brief description of the MVC architecture and its benefits. More importantly, attention was given to the billing engine which is the component that enables the implementation of non-conventional billing options. Finally, the database structure and the classes that are used to handle operations were highlighted. The next chapter gives details on the implementation of the system.

Chapter 5

Technological choices and motivation

5.1. Introduction

In Chapter 4 the overall architecture for the prototype was established. However, low level decisions such as the choice of programming languages, tools and platforms to use were not made. In this chapter, such decisions will be presented and, where necessary, discussed.

5.2. Deployment diagram

According to Ambler, a deployment diagram serves to model the hardware of the system, the software that is installed on the hardware, and the middleware used to connect disparate machines to each other and capture the configuration of the run-time elements of the application [79]. This diagram was designed to give a better overview of the system and how the components are interconnected.

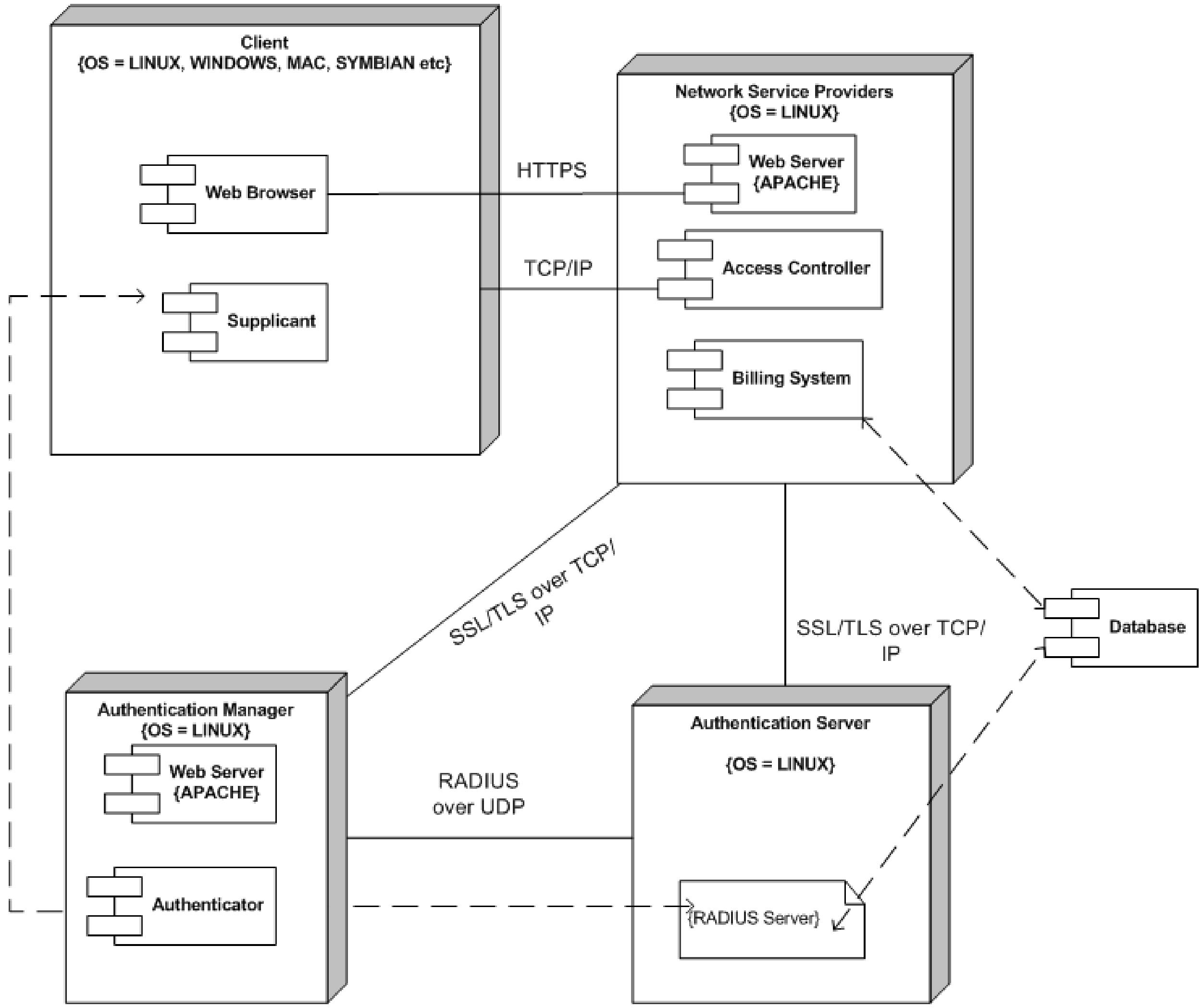


Figure 5:1: Network Revenue Management System Deployment Diagram

The deployment diagram shows that the clients can run on any operating system, for example, Microsoft Windows, Linux, Mac or Symbian. The clients interact with the access controller via the TCP/IP protocol and with the service provider's web server via HTTPS. All communications concerning authentication are done over the Secure Sockets Layer. Finally, it is also highlighted that the RADIUS protocol is used for the communication between the Authentication manager and the Authentication server.

5.2.1. Secure Sockets Layer (SSL)

SSL is a cryptographic protocol that is commonly used to provide security of communication on the Internet for web browsing, instant messaging and other data transmissions. It aims to provide authentication, privacy and data integrity between two communicating endpoints. It enables the communicating partners to do so in a manner which minimises security risks such as eavesdropping and spoofing [80]. SSL has become a de facto standard, and recently evolved into Transport Layer Security (TLS) [81, 82].

SSL runs on a layer which lies between reliable protocols such as TCP and application protocols such as HTTP and FTP. Hence, it can be used to add security to any application protocol that uses a reliable protocol. On World Wide Web (WWW), it is commonly used to secure web pages where it is used with HTTP to form HTTPS [82].

In this deployment, the web server's identity is ensured or verified first, whilst the user remains unauthenticated. This means that the user is able to verify if the server they are about to send their credentials to is authentic.

5.2.2. Remote Authentication Dial-In User Service (RADIUS)

As highlighted in the Authentication, Authorisation and Accounting section of chapter 4, RADIUS was used for AAA purposes. According to Microsoft TechNet in [83], RADIUS is an industry standard protocol illuminated in RFC 2865 [72] and RFC 2866 [84]. It enables network access servers to communicate with a central server using RADIUS messages in order to authenticate users. The RADIUS messages are sent as User Datagram Protocol (UDP) messages [83].

Some of the RADIUS messages, as defined by RFCs 2865 and 2866, are summarised in Table 5-1.

Value	Message	Description
1	Access-Request	Sent by a RADIUS client to request authentication and authorization for network connection
2	Access-Accept	Sent by a RADIUS server in response to the request. This message notifies the client that the connection attempt is authenticated and authorized
3	Access-Reject	Also sent by a RADIUS server in response to the request. But, this one is sent to inform the client that the connection attempt has been rejected.
4	Accounting-Request	Sent by a RADIUS client to specify accounting information for a connection that was accepted.
5	Accounting-Response	Sent by the RADIUS server in response to 4. This message acknowledges receipt and processing of the accounting request message.
11	Access-Challenge	This message is used by the RADIUS server to issue challenges which clients must respond to.

Table 5:1: RADIUS protocol messages

5.3. Tools used

5.3.1. Apache2 Web server

According to the Netcraft Web Server Survey, Apache has been the leading web server software since April 1996. As of November 2007, Apache was serving more than 50% of all web sites [85]. It supports various features which range from server-side programming language support to authentication schemes. Moreover, it provides support for languages like Perl and PHP and also includes features such as SSL/TLS and URL rewriting.

5.3.2. Access controller

As highlighted in the requirements, analysis and design phases, an access controller is needed to enable the system to force users to authenticate before gaining access to network resources. This section describes the solution implemented to perform this functionality.

5.3.2.1. Captive Portal

According to the Personal Telco Project, the Captive portal technique forces an HTTP client on a network to be redirected to a special web page before surfing the internet [86]. Captive portals are usually implemented by placing an authentication gateway in front of the network to control network and services connectivity [87]. Such gateway intercepts and blocks all packets destined for the outer interface until the client opens a web browser and try to access the Internet. The user is redirected to a web server called the "authentication server", that displays an authentication page to the user. If the user supplies correct credentials, the authentication server instructs the gateway to forward that user's packets to the desired interface. The gateway is responsible for opening and closing firewall rules, thus selectively allowing users to access network resources [87]. The communications between the different parties involved in this process have been summarised in the Authentication and Access control sequence diagram shown in Figure 4-2. This prototype uses a captive portal application called Chillispot.

5.3.2.2. Chillispot

Chillispot is an open source captive portal or local area network access point controller [88]. It is a user-land daemon whose primary tasks involve carrying IP packets from one network interface over to another. In a typical configuration, it sits between two networks: one is the *inner network* and the other one is the Internet *outer network.* Hence, the computer that runs Chillispot needs at least two network interface cards, such as *eth0* and *eth1*. Chillispot supports two authentication methods: the Universal Access Method (UAM) and Wireless Protected Access (WPA).

5.3.3. RADIUS Server

As mentioned in chapter 3, the RADIUS server has real authority and the necessary information to make decisions regarding granting access to specified users. Our prototype uses FreeRADIUS as the RADIUS server of choice.

5.3.3.1. FreeRADIUS server

FreeRADIUS is the premiere open source RADIUS server [89]. FreeRADIUS server has been developed by people who have many years of collective experience in developing RADIUS software.

It runs on a Linux server as a daemon process. It includes support for DBMSs such as MySQL, Postgress and Oracle. It also includes support for LDAP, RADIUS proxying, failover, load balancing, and nearly 100 vendor dictionary files [89]. In our prototype FreeRADIUS Version 1.1.7 was used.

5.3.4. Billing system

As highlighted above, the captive portal forces users to authenticate before they are allowed to access the system. When they are authenticated, a session is started. FreeRADIUS, which is employed for AAA purposes, account for their usage, by logging their usage data into a MySQL database. After this data is stored in the database, a way is needed to manipulate this data for billing purposes. In our case, we want to use this data in combination with other data from external sources to make billing decisions. A system for doing such operations was needed and for our prototype we chose Hotcakes.

5.3.4.1. Hotcakes hotspot manager

Hotcakes [90], is an Open Source, LAMP (Linux Apache Mysql PHP) hotspot management solution based on the CakePHP framework [91]. The first beta version of Hotcakes was released on Cakeforge in May 2007. As of this writing, it is on the fifth beta. This application is a billing solution with a lot of features.

Hotcakes features an interface which is used to manage subscribers, usage statistics and invoicing. It also supports bandwidth shaping through Chillispot. It offers traditional billing options, specifically data usage and time online, out-of-the box.

Hotcakes has three types of users:

- Administrators: They can do virtually anything on the system
- Cashiers: These are less privileged administrators, who only have the right to manage subscribers, payments and other day- to-day operations.
- Clients: These are the subscribers of the system. They use the system to gain access to the internet. They can also log into the system to check information pertaining to their accounts such as usage activity, payments and personal details.

Hotcakes comes with most of the features which were stipulated in our requirements, hence it was chosen as the most suitable front end candidate for refinement and adaptation to meet our requirements.

To adapt Hotcakes to our needs, it was studied to find where we could to hook into its functionality to implement the non-conventional billing mechanisms. However, prior to this, it was imperative to learn the CakePHP framework to better understand how Hotcakes works.

5.3.4.2. CakePHP framework

CakePHP is a free and open source rapid development framework for PHP. It is a structured collection of libraries, classes and run-time components for developers creating web applications. This framework was originally inspired by the Ruby on Rails framework. It employs widely known design patterns such as ActiveRecord, Association Data Mapping, Front Controller and MVC. Its goal is to enable developers to work in a structured and rapid manner and develop robust web applications without losing flexibility [91]. Since CakePHP is based on the MVC design pattern, it separates out database and business logic from the presentation layer.

A class called *Dispatcher* lies at the heart of CakePHP. It is responsible for turning requests into actions. Specifically, it figures what controller to use, sets up the controller; makes sure that the appropriate models, components and so on are loaded and takes care of error handling issues. However, this class is usually transparent to developers. The diagram shown below can be used to highlight a simplistic representation of how CakePHP handles incoming requests [92]. The controller handles all the requests and interacts with models to manipulate data stored in the database and send the returned data to the views.

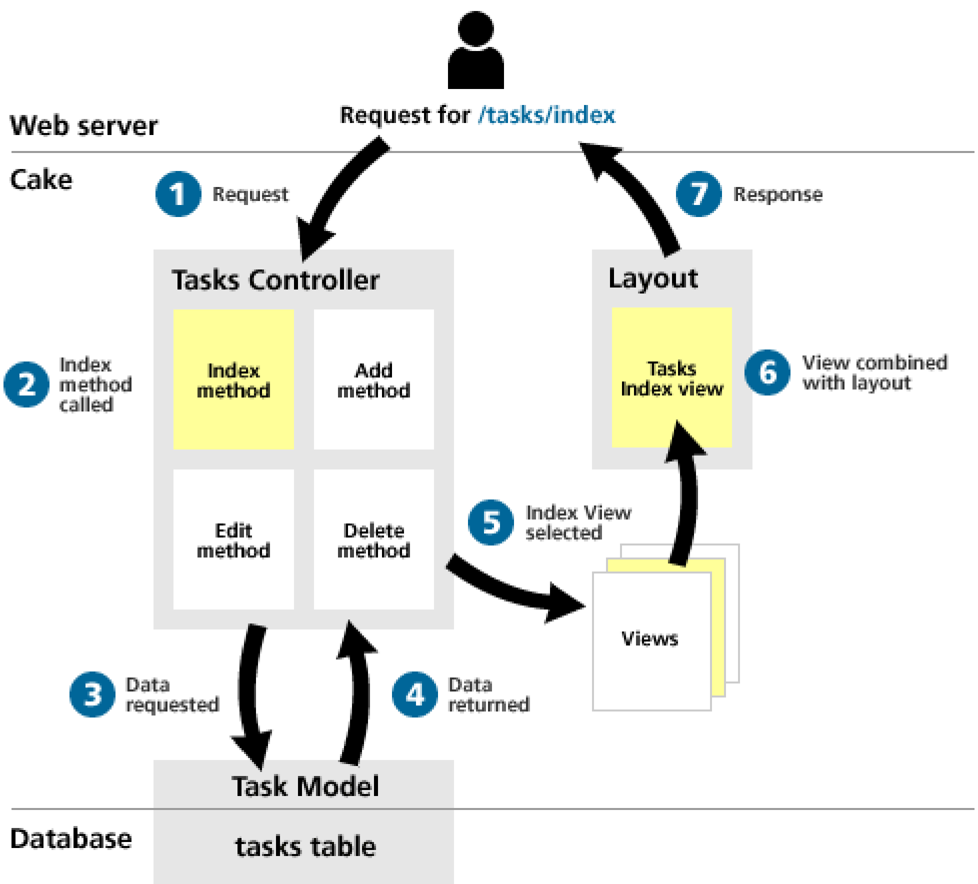


Figure 5:2: How CakePHP works [92]

Almost all requests in CakePHP follow a similar pattern. The diagram below shows a "bare-bones" MVC request in the CakePHP framework. For illustrative purposes, let us assume that the user has just clicked the 'tasks' link on your application [40].

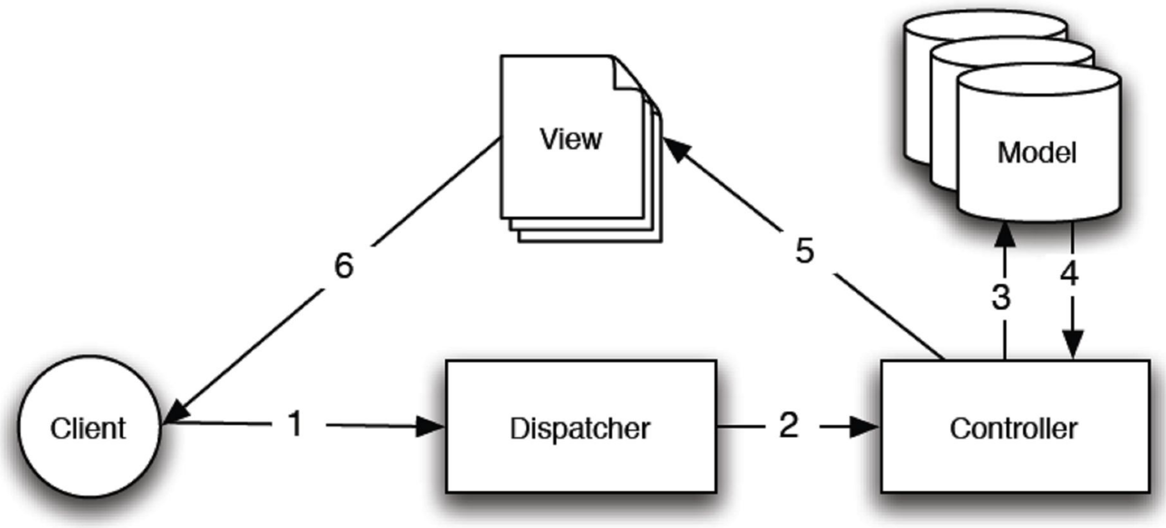


Figure 5:3: A Basic MVC Request [40]

1. The user clicks on the link http://YourSite/tasks/index, and his browser makes a request to the Web Server.

2. The Dispatcher checks the request URL (/tasks/index), and directs the request to the appropriate Controller.

3. The Controller performs application specific logic and.

4. Uses models to gain access to the application's data.

5. After manipulating and processing the data, it hands it over to a View.

6. Once the View has used the data from the controller to build a fully rendered view, the contents of the view are returned to the user's web browser.

5.4. Motivation for the choices

There are several platforms and tools that could have been used to implement the prototype. In this section, the motivation for choosing our solution is explained.

5.4.1. Why Free and Open Source Software

According to Bruce Perens of the Open Source Initiative [93], the philosophy of free software is not a new one. In the early days of computers in universities, software was freely passed around, and programmers were paid for programming not for the programs themselves. Later on programmers began restricting rights to their software and charging fees for each copy. In 1984, Richard Stallman began popularising free software as a political idea by forming the Free Software Foundation and its GNU Project [93].

Open Source Software is free software that is released under a special license where the user is granted the right to view the source code, modify it, and be able to redistribute the software. It is developed in a public, collaborative manner. In actual fact, open source software does not just mean software with access to the source code. Its distribution terms should also comply with the criteria set in the Open Software Definition [93] which is the bill of rights for the users.

Why Open Source? Why not? Our main reason for using open source software is the fact that the Network Cost Management is implemented in a network which runs Free and Open Source Software platforms. This means that the system needs to fit in currently existing infrastructure. Consequently, this approach inherently comes along with the benefits of open source in general.

The advantages of FOSS have been advocated in much literature. Its implications are not just technical but political, social and economic as well. FOSS has been used by many flourishing and profitable businesses worldwide. This is evidenced by the popularity of software such as Firefox, Apache Server, Mysql and Linux. For example the diagram below shows that Apache has been the leading Web Server since 1996 [85].

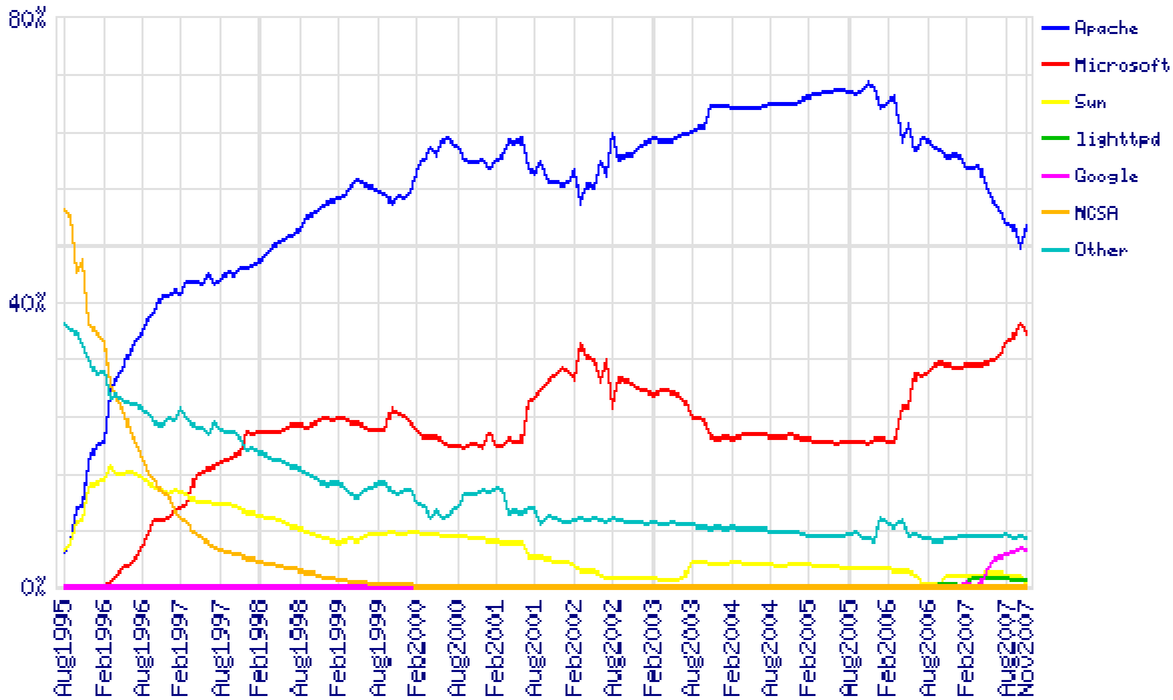


Figure 5:4: Market Share for Top Servers across All Domains August 1995 - November 2007 [85]

Advantages of FOSS

- **Low software costs** – open source requires no license fees. In addition to this, it also should not have high maintenance fees, though this is not always true in practice if one does not have access to developers. Therefore, open source software is good for everyone, especially those rural and marginalised communities whose capacity to generate income is very low.

- **No vendor locking** – users are not tied to stick with one vendor. The choices are open for the users to choose from different vendors of their liking.

- **Encourages hands-on** – the use of open source encourages people to have hands on experience even if they are not programmers.

- **Support** – Open source software is supported by large communities. Most of the times, the problem you are dealing with has already been experienced and possibly solved by someone else. So by posting the problem on the forums or searching the knowledge base, you are likely to get the solution or get someone to answer the query for free.

- **Security** and **reliability** – the Linux environment is considerably more secure than the Microsoft Windows environment. After deploying ICT in rural areas, experts will not always be there to deal with Microsoft Windows' recurrent problems such as viruses. Even though Linux at times get viruses, the viruses are too few to be a major threat.

- **Availability** – given the fact that you do not have to reboot your machine every time you get new patches, catch viruses and experience system errors, the downtime for Linux is relatively lower and compared to Microsoft Windows. (The difference is becoming less pronounced the time passing, though.)

The only disadvantage that one who wants to venture into free and open source solutions might face is that one has to dedicate sometime in setting up and configuring and troubleshooting the applications. In addition, to readily enjoy the support and benefits of the open source community, one should also devote time for getting acquainted with its etiquette.

5.4.2. Why Chillispot

There are many captive portal software we could have chosen from. This section highlights some of the solutions looked into and why Chillispot for our purposes was more appropriate.

The prototype being discussed in this thesis considered the following open source captive portal solutions.

- **NoCatAuth** – is an open source centralised authentication system that can be implemented in various wireless network deployments and is based on Perl [94]. The system is composed of a gateway server, authentication server and an access point. The last version of NoCatAuth was released in May 2004, its C port, NoCatSplash, last released in February 2005 and by this time it was missing a lot of the functionality of NoCatAuth and only had beta support for authenticated users. There was no active development of this package over the past two years.

- **Public IP's Zone CD** – Public IP provides a complete open source hotspot solution. However this solution needs to completely reformat the hard drive and install. Given the fact that infrastructure is one of the greatest challenges in bridging the digital divide, we cannot afford solutions which will prohibit us from running as many applications as we want on one server to minimise the total number of servers needed. The Zone CD was mainly developed for the implementation of free hotspots. Hence, it is not easily customisable for billing purposes such as ours [95]. This was the same scenario with LESS networks [96].

- **Other** – Other captive portal solutions such m0nowall and SweetSpot were left out because of platform and implementation related issues.

- **Chillispot** - Chillispot is under active development and maintained by Jens Jakobsen. It is developed in C. There are many projects using it for authentication purposes, and so there is a large community base, increasing the support base. It is a more complete implementation than NoCatAuth. It features advanced authentication methods such as the Universal Access Method (UAM) and Wireless Protected Access (WPA), authentication with RADIUS servers, as well as reporting functionality. It also manages connections, sets bandwidth-throttling and firewall rules.

5.4.3. Why Hotcakes

There are other applications that could have been used to provide an interface for the management of client accounts, but due to time constraints, for this prototype we considered the following:

- **PHPMyprepaid** – This is an interface to support the creation and management of user accounts for the billing system. It is designed to work with a Mysql database and FreeRADIUS server. This system provides features such as the JavaScript pop up after login, which displays user statistics such online time, bandwidth usages statistics and so on. Closing this pop up will end your session. It also provides functionalities for generating prepaid cards generate PDF invoices and also include the concept of locations, where a user should belong to a particular location.

  However this project's development activity has been very low in relation to its age. From 2004 it has achieved 5 major realises. Its code base is based on the traditional PHP coding techniques. It is a collection of PHP files calling each other using PHP <*includes*>. Although it is very flexible enough, it lacks a well defined structure. Furthermore, it is not well documented; therefore the learning curve is huge.

An advantage of using a development framework, which PHPMyprepaid does not offer, is on security. When coding using a framework like CakePHP, there is usually no need to worry about security because the framework takes care of it. However it should be admitted that this does not mean that frameworks do not have security issues and exploits.

Nevertheless, this was my first choice application during the first year of my research, and it gave me good hands-on experience on managing user accounts and understanding the underlying billing operations.

- **Hotcakes** - A new comer into this arena has come with a lot of features which surpassed PHPmypaid. It provides all the functionality provided by PHPMyprepaid and on top of that it enjoys all the benefits of being built on a well defined robust, CakePHP framework. It provides an elegant interface for the creation and management of user accounts on the network.

  It features multi user levels, client's own personal page, multi languages support, and realms, which are similar to the location concept in PHPMyprepaid.

  It also provides an easy local DNS howto where user just type 'exit' in the address bar to log off, 'info' to go to their personal pages. However, you can change this to whatever phrases you want and you can even add more. Moreover, Hotcakes also use AJAX to actively display usage statistics to the user during a web session.

  Some of Hotcakes' inherited advantages are: active, friendly community, integrated CRUD for database interaction, application scaffolding, built-in validation, email, cookie, security, session and request handling components, data sanitation, flexible view caching and it provides helper for features such AJAX, JavaScript and HTML forms [40].

  Given its modularity and extensibility, it was easy to implement our non-conventional metrics in the billing processes without affecting the normal operation of the system. Its structure makes its components reusable in new applications.

5.5. Conclusion

In this chapter, the different technologies selected to assemble the different components of the system were described.

A deployment diagram was used to elaborate on the components and protocols that were used to achieve secure communication between different elements of the system. Illustrations of how the components work were also given.

Finally, justifications for the main technological choices were given.

Chapter 6

Implementation

6.1. Introduction

Chapter 5 has highlighted the tools that were used to build the prototype of the Network Revenue Management System. This chapter describes how these tools were used in the implementation of the system. The system prototype described in this section serves as a proof of feasibility of the proposed architectural design.

6.2. Configuring Apache to support SSL

For the components to communicate via SSL as shown in the deployment diagram in Figure 5-1, the web server had to be setup appropriately. The following steps were used

1. Install the openssl package

   ```
   sudo apt-get install openssl
   ```

2. Generate the SSL certificate

   ```
   sudo make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
   ```

3. Enable the SSL module

   ```
   sudo a2enmod ssl
   ```

4. Add the ssl port *443* to *ports.conf*

   ```
   echo "Listen 443" >> /etc/apache2/ports.conf
   ```

5. Create and enable the SSL site

   ```
   sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/ssl
   ```

6. Edit the SSL site created in 4 to reflect the correct location of the SSL certificate

7. Enable the SSL site

    sudo a2ensite ssl

8. Restart Apache

    sudo /etc/init.d/apache2 force-reload

6.3. Configuring Chillispot

As highlighted in Figure 3-3, Chillispot takes control of the internal interface (*eth1)* using a kernel module called *vtun* to bring up a virtual interface *(tun0)*. Actually, vtun is used to move IP packets form the kernel to the user mode to enable Chillispot to operate without any non-standard kernel modules. To enable the routing of the IP packets between the two interfaces *eth0* and *eth1,* packet forwarding needs to be enabled in the kernel. In our prototype this was achieved by un-commenting the

net.ipv4.conf.default.forwarding=1

line in the "/etc/sysctl.conf" file. Furthermore, to enable network address translation, NAT was also configured as follows:

iptables -t nat -A POSTROUTING -o ethX -j MASQUERADE

By default, the configuration file of Chillispot is a file called chilli.conf in the "/etc" directory. Some of the most important configurations are highlighted in the table shown below:

Attribute	Comment
uamserver	Chillispot needs to know the location of the authentication web server. In other words this is the URL of the page to which users are redirected to authenticate.
uamsecret	This is a shared secret between Chillispot and the authentication web server.

radiusserverX	This designates the location of the radius server used, where X will have the values 1 and 2. This means you will have the option to specify two radius servers.
radiussecret	This is a shared secret between Chillispot and the radius server

Table 6:1: Chillispot configuration attributes

6.4. Configuring FreeRADIUS server

FreeRADIUS server has many configuration files which need to be configured. Some of the files are used for authentication purposes and some for accounting purposes. The first file to configure is the /etc/freeradius/clients.conf. This is where the radius secret is defined, for authentication purposes between Chillispot and FreeRADIUS server. The important section should look like this [97]:

```
client 127.0.0.1 {
        secret      = myradiussecret
        shortname   = localhost
        nastype     = other
}
```

FreeRADIUS server comes preconfigured for Mysql so all you need is to point it to the right database in /etc/freeradius/sql.conf. You can also configure it to work with Postgres, Oracle or MS-SQL.

FreeRADIUS comes preconfigured to store accounting data in files. For our purposes it had to be configured to use Mysql instead of files. To accomplish this, the accounting section in /etc/freeradius/radius.conf was configured to look like this:

```
accounting {
        unix
        radutmp
        sql
}
```

There is also an authorise section which needs to be configured to look like this [97]:

```
authorize {
        preprocess
        chap
        mschap
        suffix
        sql
        noresetcounter
}
```

6.5. Tapping into Hotcakes

Hotcakes comes with well documented installation and configuration instructions. Hence, they will not be elaborated in this thesis.

Hotcakes uses profiles to control how users should access the Internet. Profiles store information specifying:

- The maximum upload and download speeds.
- The allowed idle time for an account.
- The number of allowed simultaneous instances for the same username.
- Frequency in which accounting data will be updated for the accounts using that particular profile.
- The maximum data cap.
- Expiry date and other quality of service parameters.

A sample profile is shown in figure 6-2.

Attributes Used

Attribute Name	Tool Tip Text	Check or Reply	Units
WISPr-Session-Terminate-Time	2008-06-09T19:00:00+00:00 (to teminate 09 June 2008 7PM)	Reply	Text String
WISPr-Bandwidth-Max-Up	Maximum transmit rate (b/s) - From Hotspot to Internet	Reply	Bits/Second
WISPr-Bandwidth-Max-Down	Maximum receive rate (b/s) - From Hotspot to Internet	Reply	Bits/Second
Idle-Timeout	(Try 900)- After switchoff => auto disconnect	Reply	Seconds
Simultaneous-Use	Limit the amount of machines / given time	Check	Text String
Acct-Interim-Interval	(Try 60) - The time in seconds which accouinting data will be u	Reply	Seconds

Table 6:2: Hotcakes profile attributes

The attributes used are extracted from the FreeRADIUS server and Chillispot configuration files. If there is need to include other vendor specific attributes, these can be found under the '/usr/share/freeradius' directory, where the dictionary files for different vendors are stored.

After a profile is created, user accounts can be created and then be associated with that profile. Billing plans are drawn and then used to generate invoices, normally based on the QoS parameters specified in the profile. The actual charging is performed by the Invoices Controller. Figure 6.1 gives the class diagram of the Invoices Controller.

Hotcakes provides two types of user accounts; permanent and prepaid. Permanent users are those users which are charged on a postpaid basis whilst prepaid users are charged on a prepaid basis (using prepaid cards).

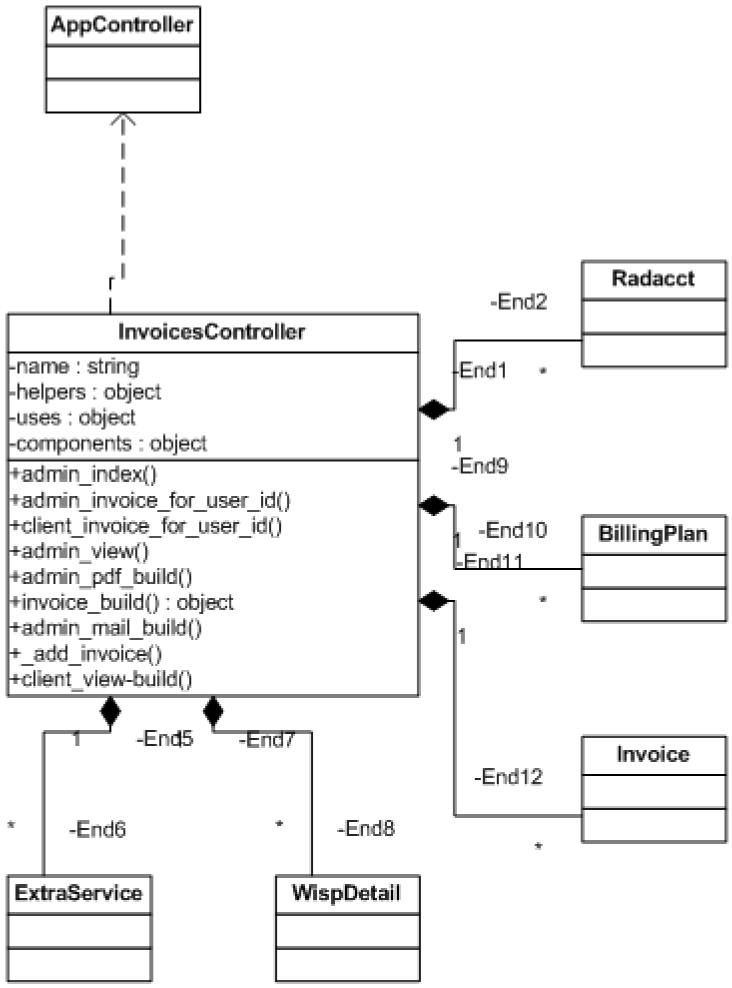


Figure 6:1: Invoices controller class diagram

The class shown in the next page is used for creating a data structure containing all the information that will be displayed on the invoice.

```
*/
class InvoicesController extends AppController {

    var $name = 'Invoices';
    var $helpers = array('Html', 'Form','fpdf','Image', 'Info' );
    var $uses    = array('Invoice','Radacct','RatingManager','PermanentUser','BillingPlan','WispDetail','ExtraService',
    'ConfigurationItem');

    var $components = array('SwiftMailer','InvoiceData');

    function index() {

    function invoice_for_user_id($id=0) {

    function view($id = null) {

    function add() {

    function edit($id = null) {

    function delete($id = null) {

    function create(){

    function view_build($invoice_id){

    function pdf_build($invoice_id){

    function mail_build($invoice_id){

    //Creates a data structure containing all of the info of the invoice
    function invoice_build($id = null){


    function _add_invoice($user_id,$start_date,$end_date,$billing_plan_id){

    function _date_clash_check($user_id,$start_date,$end_date){
```

The information is built in the *invoice_build* function and includes:

- Service provider details
- Invoice specific details such as invoice numbers
- User details
- Activity Details
- Billing plan details
- Extra services and
- The total amount to be paid.

What we wanted was the ability to include non-conventional billing options in our invoice calculations or to choose different billing options based on specified configurations or decisions.

To achieve this, we created new functions. The generic functions which perform common operations were placed in the *rating_manager-controller* class and adapter specific operations were placed in the specific Adapter. The structure and relationships of these classes are shown in the diagram below.

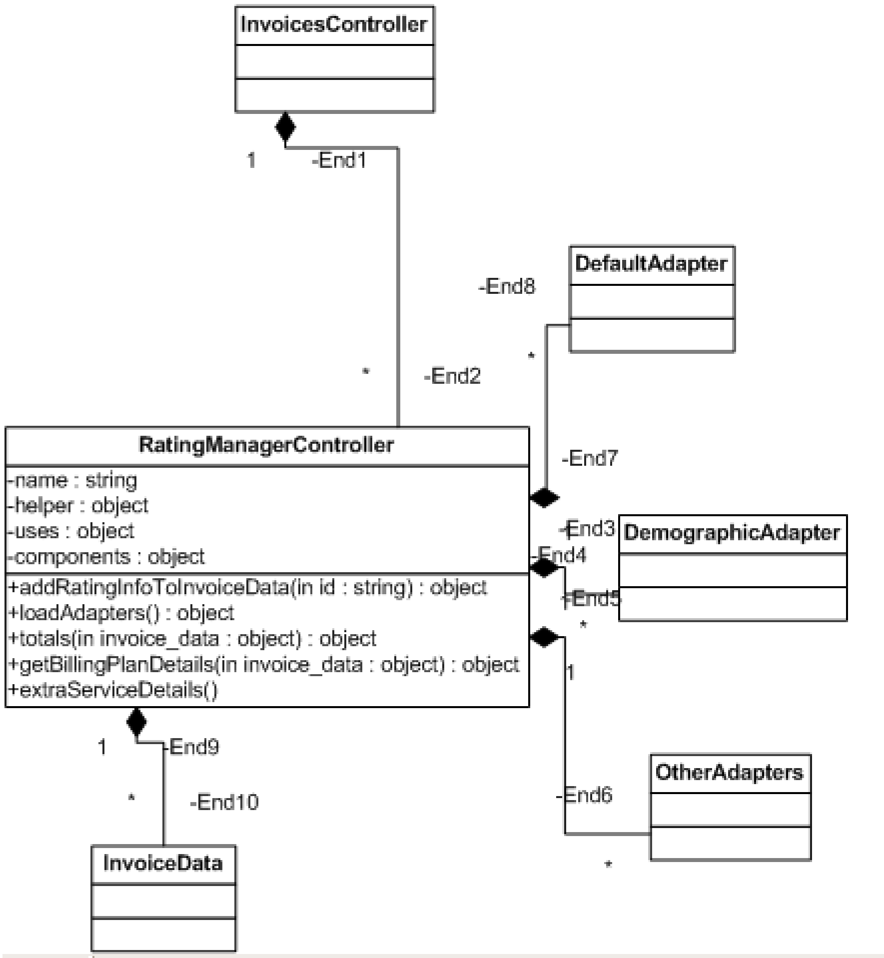


Figure 6:2: Rating Manager Controller

The *RatingManagerController* is responsible for adding rating information passed to it by the *InvoicesController* to the invoice data. This information includes billing plan details, extra services and other details. It also loads the adapters which are responsible for calculating charges based on different metrics. In other words, it acts as our central engine for making charging and pricing options. The code snippet of the rating manager is shown on the next page.

```php
<?php
/*
   Copyright (c) <2007> Paul Tarwireyi (<ptarwireyi@gmail.com>).
   Created 2:23:01 PM   at  in  2007
   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 2 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software
   Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
    Window - Preferences - PHPeclipse - PHP - Code Templates
 */
class RatingManagerController extends AppController {
       var $invoice_data;

   var $name = 'RatingManager';
   var $helpers = array('Html', 'Form','Info' );
   var $uses    = array('Invoice','Radacct','PermanentUser','BillingPlan','WispDetail','ExtraService','Demographic','ConfigurationIter

   var $components = array('DefaultAdapter','InvoiceData','DemographicAdapter');



    function addRatingInfoToInvoiceData($id)

   function loadAdapters()


   function totals($invoice_data)

   function getBillingPlanDetails($billing_plan,$invoice_data)

   function extraServiceCosts($invoice_data)

}
?>
```

The *LoadAdapters* function is responsible for loading the adapters. In a real set up, this function will read all the specified adapter classes and their attributes into an array or object. For the purposes of this prototype a simple solution was implemented, just to demonstrate feasibility. This is shown in the code snippet below.

```php
    function loadAdapters()
    {
        //This is where i load my adapters
        //This is also where the order of ADAPTERS COME INTO PLAY

        $adapters = array();

        //Adapter 1
        $adapters[0][0] = "DefaultAdapter";
        $adapters[0][1] = array();// In this array you pass the parameters you want

        //Adapter 2
        $adapters[1][0] = "DemographicAdapter";
        $adapters[1][1]= array("income");

        //Adapter 4
        $adapters[2][0] = "DemographicAdapter";
        $adapters[2][1]= array("dependent");

        //Adapter 3
        $adapters[3][0] = "DemographicAdapter";
        $adapters[3][1]= array("beer");

        return $adapters;
    }
```

Later when we need the adapters, we should be able to iterate through the adapter object and call the adapters, as shown below.

```php
        for($i=0;$i<count($adapters);$i++) {
            //print_r($adapters);
            $adapterClass=$adapters[$i][0];
            $parameters=$adapters[$i][1];
            $invoice_data=$this->$adapterClass->getData($invoice_data,$billing_plan,$parameters);

        }

```

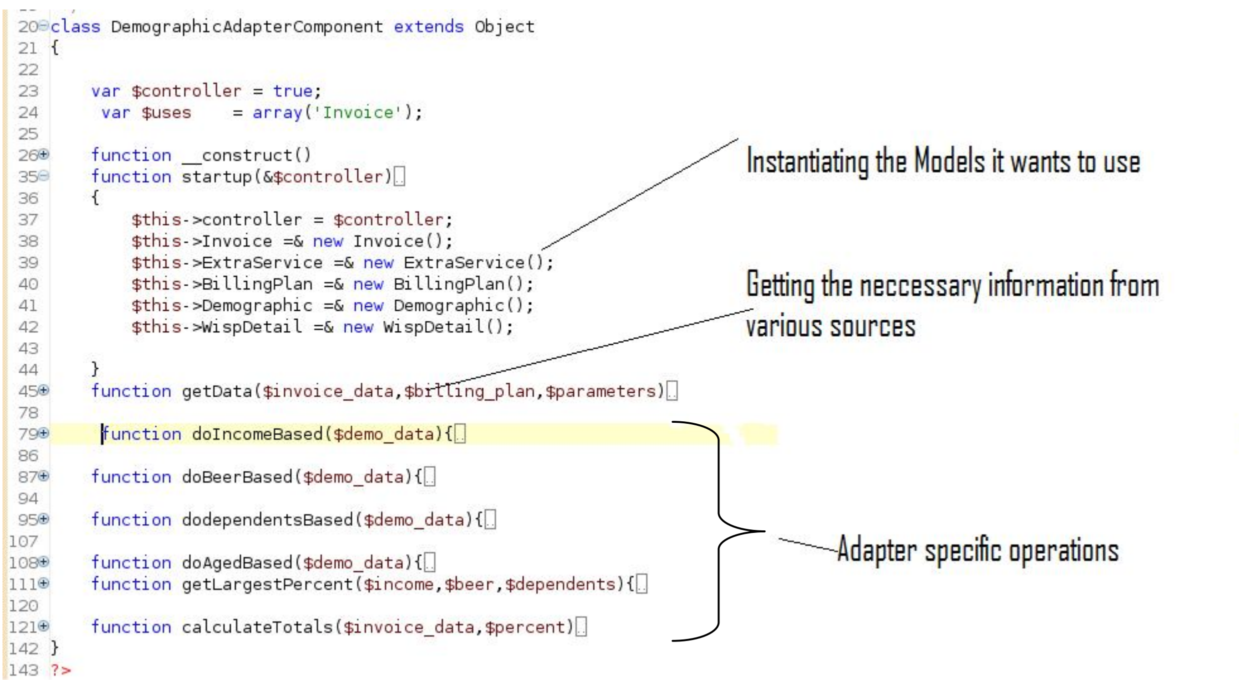
This implementation makes it possible to introduce new adapters or billing options without a need to change how the system was functioning. For example, there is an adapter called demographics; this adapter has many dimensions to deal with and we can selectively specify the dimension we want to consider by supplying it as a parameter as shown with income, dependents and beer. To add a new adapter in demographic context say, ‘number of cows’, the logic that will be performed by the adapter has to written in the *DemographicAdapter* and the new adapter has to be added in the *LoadAdapters* function. For example:

```php
$adapters[4][0] = "Cows";

$adapters[4][1]= array();
```

To implement the adapters we relied on CakePHP's 'components'. In the CakePHP framework, components are used to aid controllers in specific situations. Instead of extending Cake's core libraries, special functionality can be simply implemented in components. A component makes it possible to write a new class which can help controllers in specific situations. The advantage of using components lies in their sharability and reusability [91]. The diagram below illustrates a simple component.

```
*/
class DemoComponent extends Object
{
    var $MyVar = null;
    var $controller = true;

    function startup(&$controller)
    {
        // This method takes a reference to the controller which is loading it.
        // Perform controller initialization here.
    }
    function doSomething()
    {
        $this->MyVar = 'Demo';
    }
}
?>
```

The *RatingManagerController* acted as the call point for the new components, as shown in Figure 6-2. The code snippet shown below depicts the structure of adapters. The demographics adapter is used only as an example.



```php
class DemographicAdapterComponent extends Object
{

    var $controller = true;
     var $uses     = array('Invoice');

    function __construct()
    function startup(&$controller)
    {
        $this->controller = $controller;
        $this->Invoice =& new Invoice();
        $this->ExtraService =& new ExtraService();
        $this->BillingPlan =& new BillingPlan();
        $this->Demographic =& new Demographic();
        $this->WispDetail =& new WispDetail();

    }
    function getData($invoice_data,$billing_plan,$parameters)

     function doIncomeBased($demo_data){

    function doBeerBased($demo_data){

    function dodependentsBased($demo_data){

    function doAgedBased($demo_data){
    function getLargestPercent($income,$beer,$dependents){

    function calculateTotals($invoice_data,$percent)
}
?>
```


According to the CakePHP manual, the *startup()* function shown in the above listing is used by the Dispatcher during the CakePHP bootstrap process and this gives the component access to the controllers that load it [91]. This means that all our adapters get access to the *RatingManagerController* and are able to instantiate the different models they will use in their calculations. For instance, if our adapter needs to enable the provision of free services for a user who is below 15 years, the *DoAgedBased* function needs to have this logic encapsulated and return the charges.

As an example of discouraging spending money on unnecessary things such as beer, you might have a rule that say that if your beer expenditure as a proportion of income is very high, then you might not enjoy certain benefits available to others or you are charged more.

However, the focus of this project is not to impose billing options on the community. These are just examples of what is possible to do with this system. The mandate to create billing options lies with the community. After a few adapters have been implemented, there are myriad ways in which other researchers can carry out investigations to find out what is considered as fair ways of billing in these communities.

Data from other systems

A central requirement for the system was to get data from other systems and utilise it for billing. In this section we will explore the extraction and use of data obtained from the Dwesa e-commerce portal for illustrative purposes.

To collect this data, a table was created to store all the individual sales for a specified period of time. A trigger was written on the orders table of the e-commerce portal to replicate committed orders to the billing system's sales table. The trigger is shown below.

```
CREATE TRIGGER sales_tracking_tr AFTER INSERT ON orders
FOR EACH ROW INSERT INTO radius.sales
(
amount,
permanent_user_id
)
VALUES (
NEW.amount, NEW.permanent_user_id
)
```

A model called 'Sale' was created to manipulate this data. A snapshot of this model is shown below. As it can be seen in the snapshot, it was possible to define the relationships between this model and other models. For example, it is shown that a sale should belong to a permanent user. Furthermore, the *validate* variable is used to specify constraints on the fields, for example, "id=>VALID_NOT_EMPTY". This ensures that data is validated before being inserted into the databases.

```php
class Sale extends AppModel {

    var $name = 'Sale';
    var $validate = array(
        'id' => VALID_NOT_EMPTY,
        'amount' => VALID_NOT_EMPTY,
        'permanent_user_id' => VALID_NOT_EMPTY,
    );

    //The Associations below have been created with all possible keys, those that are not needed can be removed
    var $belongsTo = array(
            'PermanentUser' =>
                array('className' => 'PermanentUser',
                        'foreignKey' => 'permanent_user_id',
                        'conditions' => '',
                        'fields' => '',
                        'order' => '',
                        'counterCache' => ''
                ),

    );

    var $hasOne = array(
            'SalesTotal' =>
                array('className' => 'SalesTotal',
                        'foreignKey' => 'permanent_user_id',
                        'conditions' => '',
                        'fields' => '',
                        'order' => '',
                        'counterCache' => ''
                ),

    );
function getTotals()
```

Given the fact that the Dwesa e-commerce portal is hosted outside Dwesa and the billing system will be situated in Dwesa, what if an order is made but the billing system is down? There is the need for a safer way to synchronise the data between the two systems.

At the end of a certain period, say a month, we run a *cron* job which will aggregate all the sales for the users and come up with the amount of total sales for each user in that period. Once we have the totals, we can utilise them in the billing process. As mentioned before, we could deduct a certain percentage from the proceeds of each user and record it as a sales tax/commission.

There are no limitations on how charging based on sales can be done: any business logic can be coded in. A very different option from the one just described might be to award bonuses to a seller who achieves a certain amount of sales. This helps motivate the community to participate in the project, promote the idea of income generating projects and might ultimately alleviate poverty in these communities.

Realms

Hotcakes provides a *realms* feature. This feature enables the creation of separate administrative blocks. A username becomes <username>@<realm> and a realm is managed by a particular cashier [90]. We can apply this feature in our situation, for example by treating each school as a separate realm. For example all users belonging to the Mpume Junior secondary school will follow the pattern <username>@mpume. This enables the different schools to manage their own clients.

Belonging to a particular realm means that you cannot access network resources from a realm other than yours. To enable your users to roam freely, you need to register them as global users, that is, users who do not belong to a particular realm. The pattern for these users will be just <username>. Just like what Telcos do, you might even want to charge more on the roaming clients.

Other sources of revenue

A major revenue stream for most websites is advertising online. Why not also charge for advertising on the Dwesa websites? This means we can also charge the community users for advertising their services and wares on the Dwesa websites. For example, someone who is selling fresh vegetables in the Mpume village might put up adverts on the Mpume websites or other sites. This revenue generating idea can also be applied to the Dwesa e-commerce portal. Other sources of revenue include charging for:

- Printing
- Typing
- Computer literacy courses
- Community contributions

Naturally, each of the activities listed above will be recorded appropriately for each user, and appropriate adapters written.

6.6. System walkthrough

This section provides a walkthrough of the Network Revenue Management System. It illustrates how the system works by showing key screen shots explaining the system functionality.

A testing environment was set up in the University of Fort Hare computer science lab. This environment reproduces the essential elements of the network in Dwesa. The infrastructure deployed in this environment is illustrated by the diagram below.

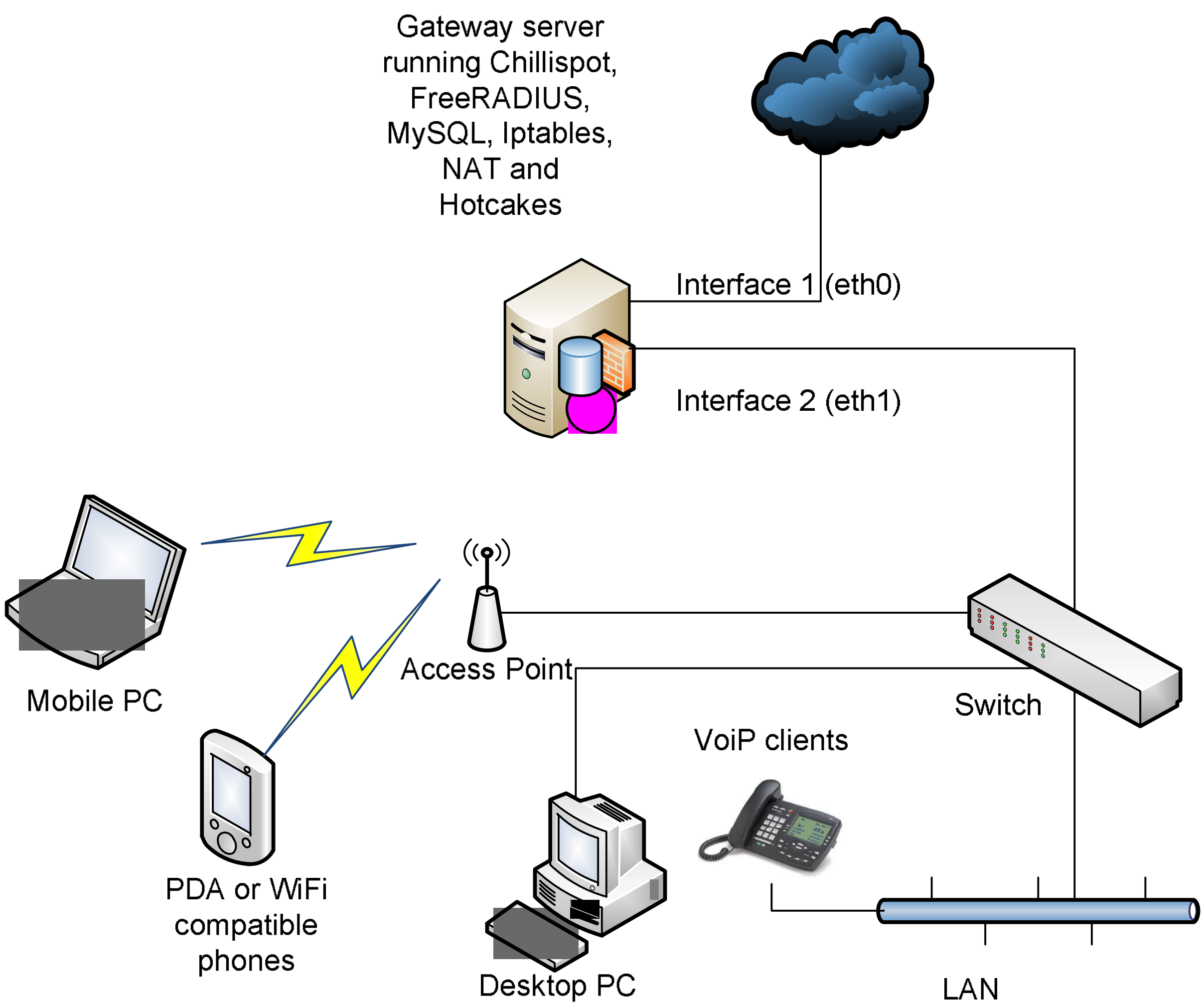
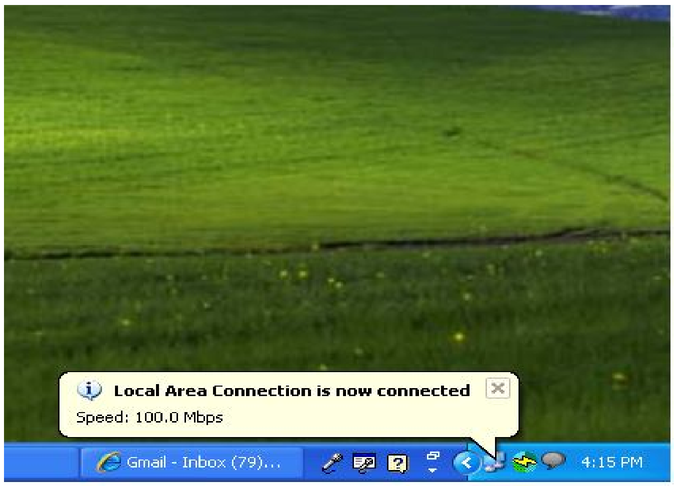


Figure 6:3: Testing environment

In the set up, a 2.80 GHz, Intel Celeron machine with 512Mb of RAM was setup with Ubuntu 7.10 (Gutsy Gibbon). This machine acts as the gateway server which runs Chillispot, FreeRADIUS, MySQL, Iptables, NAT and Hotcakes. It is equipped with two network cards, with interfaces *eth0* and *eth1,* where *eth0* is connected to the external network and *eth1* to the internal network. As highlighted in the requirements, the system allows clients to join the internal network.

After the client machine has successfully connected and joined a network, the user will see a connection success notification screen, shown below for a Microsoft client.

Figure 6:4: Successful connection



When a user tries to connect to the Internet or network resources outside his LAN, his TCP/IP session is intercepted by the captive portal and redirected to a login page or a home page explaining to the user how to gain access to the desired resource. Just before a user can be taken to these pages, though, he needs to verify the server by accepting the certificate issued by the server as highlighted by the snapshots shown below.

Figure 6:5: Accepting Certificate on a Nokia N80 phone

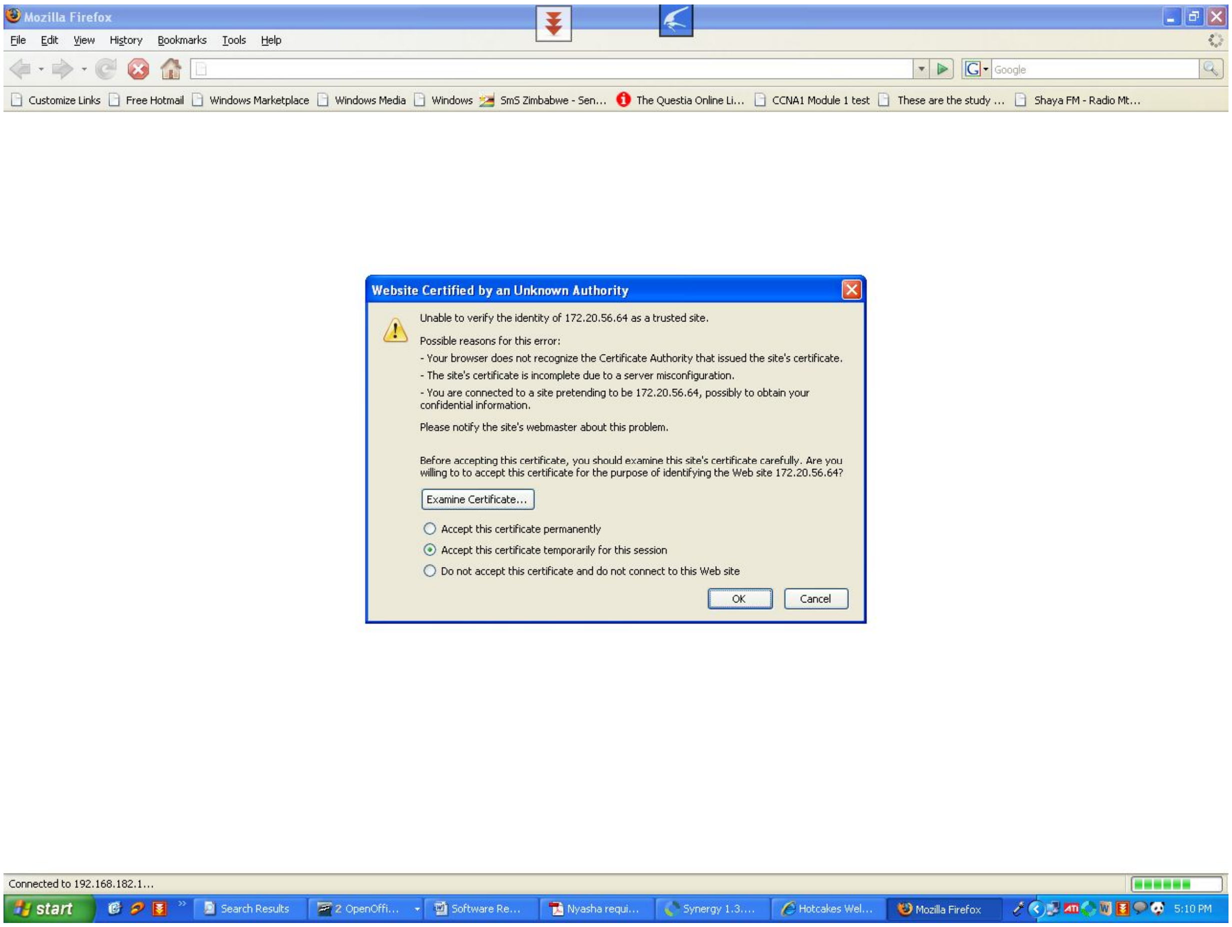


Figure 6:6: Accepting Certificate on a desktop client

After accepting the certificate, the user will be redirected to the login page shown below. A user enters his username and password. A NAS in the LAN he belongs to will collect these credentials and sends them to the RADIUS server, together with any required NAS details. The RADIUS server checks the credentials against its backend and creates a tunnel for user to access the requested resources if the authentication is successful.

Figure 6:7: Login screen on a desktop client

Figure 6:8: Login screen on a on a Nokia N80 phone

After the user is successfully authenticated, a pop up is displayed, showing the user's usage details such the current online time and total bandwidth used, after which he is redirected to the requested page.

(By making use of Chillispot's *macauth* configuration option, the system also supports devices such as VOiP phones, which do not support web browsers. This client can then is registered in the system with its mac address and a password. For example, '00-16-76-23-1F-C4' and password '1234'.)

Figure 6:9: Pop up page on desktop client

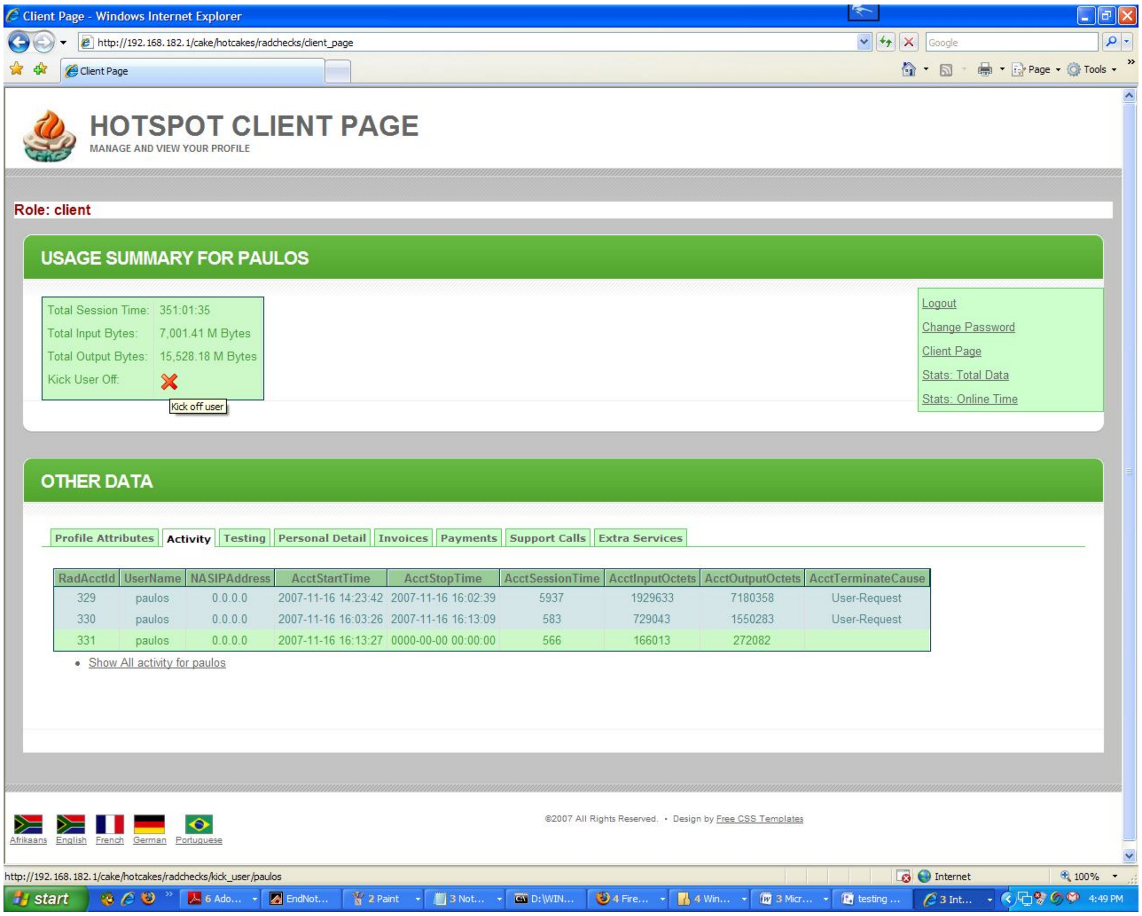


Figure 6:10: Client page on desktop client

Users are also able to check their accounts at any time by just typing ‘info’ in the address bar of their browser and logging in with their credentials. The diagram above shows the page that clients receive once logged in. On this page they can, among other things, change their passwords, check their usage statistics, and check if all their payments are reflecting.

After the user has finished using the network resources the user will log off. His session will be ended and his usage data will be logged into the *radacct* table. The following snapshot gives a sample of the information stored in each record.


```
    <FramedIPAddress />
    <AcctStartDelay>0</AcctStartDelay>
    <AcctStopDelay>0</AcctStopDelay>
  </radacct>
- <radacct>
    <RadAcctId>288</RadAcctId>
    <AcctSessionId>46c52a7e00000000</AcctSessionId>
    <AcctUniqueId>b388144ef80d8367</AcctUniqueId>
    <UserName>paulos</UserName>
    <Realm />
    <NASIPAddress>0.0.0.0</NASIPAddress>
    <NASPortId>0</NASPortId>
    <NASPortType>Wireless-802.11</NASPortType>
    <AcctStartTime>2007-08-17 06:56:48</AcctStartTime>
    <AcctStopTime>2007-08-17 07:02:34</AcctStopTime>
    <AcctSessionTime>346</AcctSessionTime>
    <AcctAuthentic />
    <ConnectInfo_start />
    <ConnectInfo_stop />
    <AcctInputOctets>350014</AcctInputOctets>
    <AcctOutputOctets>417183</AcctOutputOctets>
    <CalledStationId>00-13-D3-C3-43-E8</CalledStationId>
    <CallingStationId>00-16-76-23-1F-C4</CallingStationId>
    <AcctTerminateCause>User-Request</AcctTerminateCause>
    <ServiceType />
    <FramedProtocol />
    <FramedIPAddress>192.168.182.2</FramedIPAddress>
    <AcctStartDelay>0</AcctStartDelay>
    <AcctStopDelay>0</AcctStopDelay>
  </radacct>
- <radacct>
    <RadAcctId>286</RadAcctId>
```


In the XML snapshot, we can see different kinds of information regarding a usage record. Information such as AcctStartTime, AcctStopTime, AcctInputOctets and AcctOutputOctets is used for making usage based calculation.

As mentioned before, hotcakes provide two kinds of administrators, cashiers (less privileged administrators) and administrators (super administrators). Although these administrators have different privileges, here we highlight the common tasks both type of administrators perform, as illustrated in the screenshot below.

- administer users both permanent and prepaid (i.e. create/register, edit, update, manage and delete users),
- administer usage details,
- administer invoices,
- administer helpdesk calls,
- administer system wide configurations,
- administer external data from external sources for example, sales,
- administer payments
- administer profiles
- administer realms

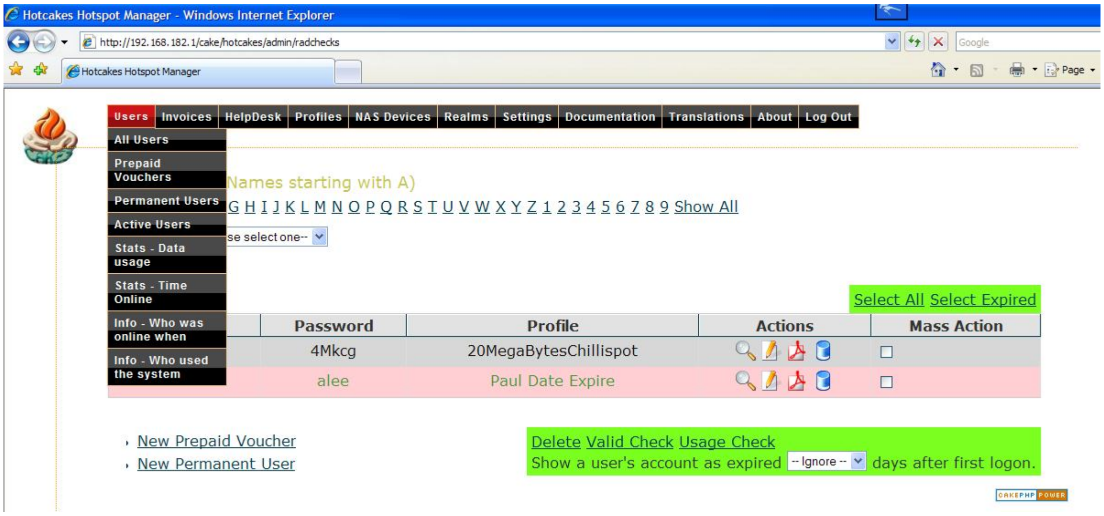


Figure 6:11: User administration

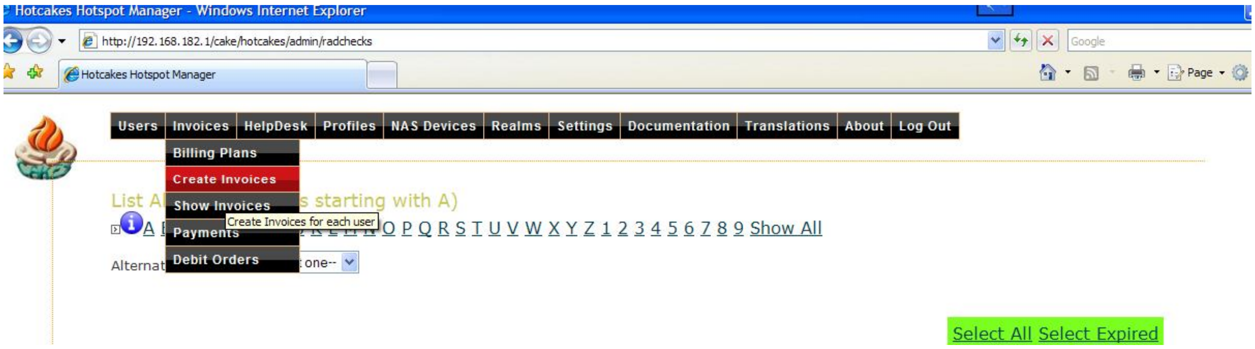


Figure 6:12 : Invoice and billing plan administration

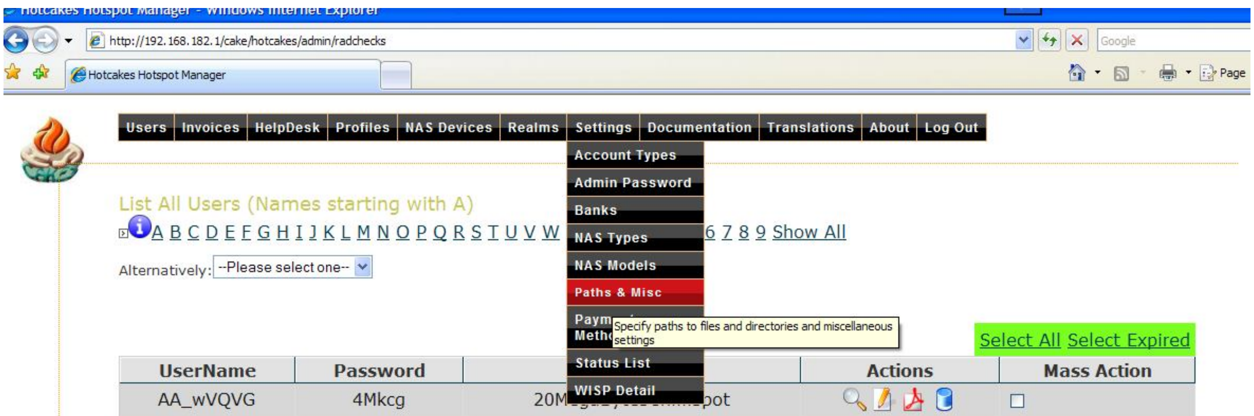
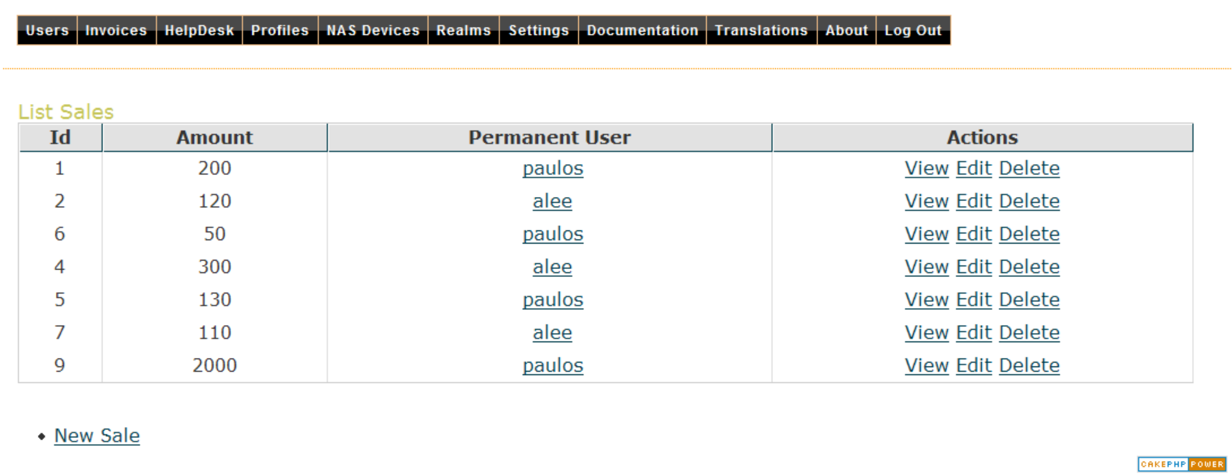


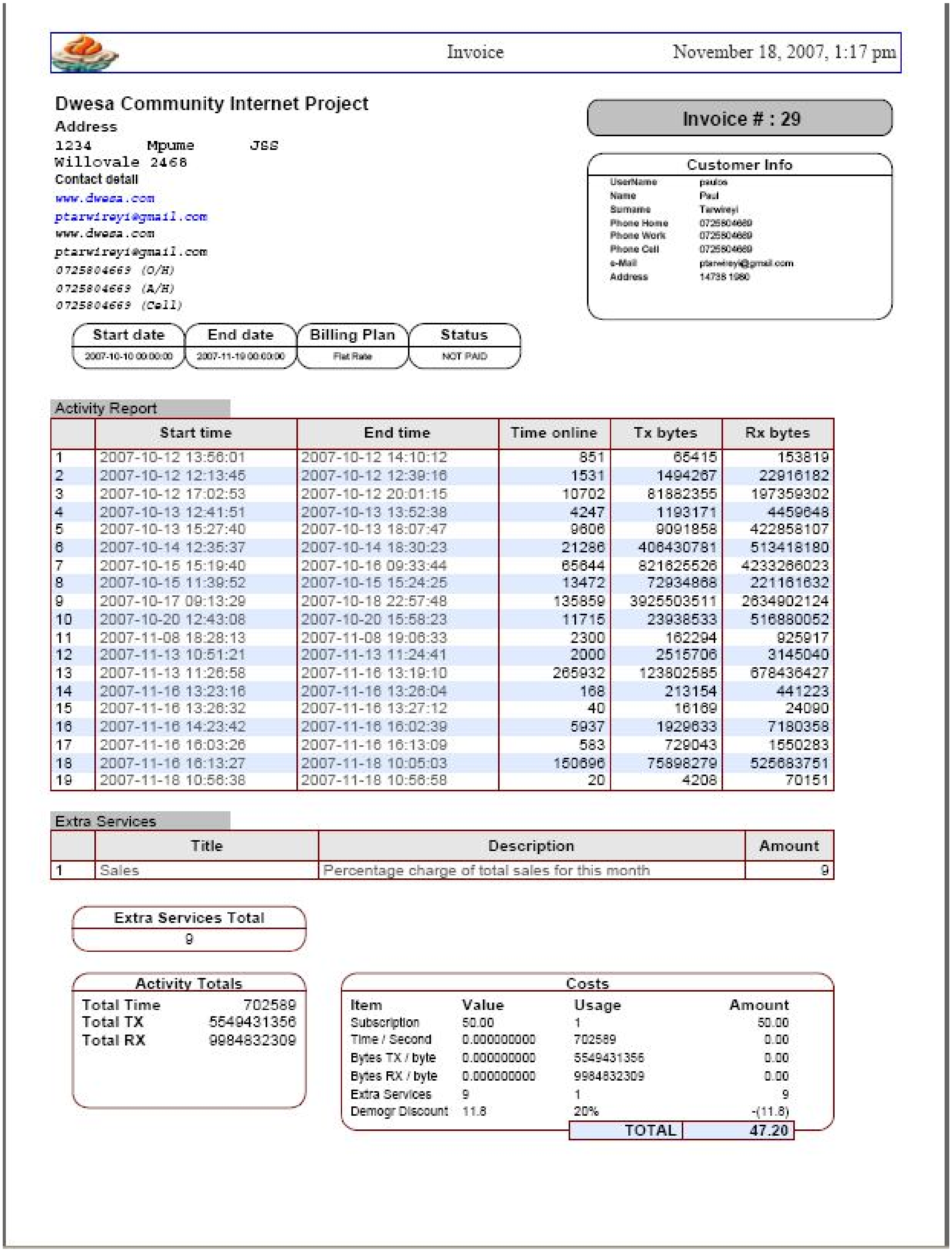
Figure 6:13 : System wide configuration administration



Id	Amount	Permanent User	Actions
1	200	paulos	View Edit Delete
2	120	alee	View Edit Delete
6	50	paulos	View Edit Delete
4	300	alee	View Edit Delete
5	130	paulos	View Edit Delete
7	110	alee	View Edit Delete
9	2000	paulos	View Edit Delete

- New Sale

Figure 6:14 : Sales administration



Invoice | November 18, 2007, 1:17 pm

Dwesa Community Internet Project

Address
1234 Mpume JSS
Willovale 2468
Contact detail
www.dwesa.com
ptarwireyi@gmail.com
www.dwesa.com
ptarwireyi@gmail.com
0725804669 (O/H)
0725804669 (A/H)
0725804669 (Cell)

Invoice # : 29

Customer Info

UserName	paulos
Name	Paul
Surname	Tarwireyi
Phone Home	0725804669
Phone Work	0725804669
Phone Cell	0725804669
e-Mail	ptarwireyi@gmail.com
Address	14738 1980

Start date	End date	Billing Plan	Status
2007-10-10 00:00:00	2007-11-19 00:00:00	Flat Rate	NOT PAID

Activity Report

	Start time	End time	Time online	Tx bytes	Rx bytes
1	2007-10-12 13:56:01	2007-10-12 14:10:12	851	65415	153819
2	2007-10-12 12:13:45	2007-10-12 12:39:16	1531	1494267	22916182
3	2007-10-12 17:02:53	2007-10-12 20:01:15	10702	81882355	197359302
4	2007-10-13 12:41:51	2007-10-13 13:52:38	4247	1193171	4459648
5	2007-10-13 15:27:40	2007-10-13 18:07:47	9606	9091858	422858107
6	2007-10-14 12:35:37	2007-10-14 18:30:23	21286	406430781	513418180
7	2007-10-15 15:19:40	2007-10-16 09:33:44	65644	821625526	4233266023
8	2007-10-15 11:39:52	2007-10-15 15:24:25	13472	72934868	221161632
9	2007-10-17 09:13:29	2007-10-18 22:57:48	135859	3925503511	2634902124
10	2007-10-20 12:43:08	2007-10-20 15:58:23	11715	23938533	516880052
11	2007-11-08 18:28:13	2007-11-08 19:06:33	2300	162294	925917
12	2007-11-13 10:51:21	2007-11-13 11:24:41	2000	2515706	3145040
13	2007-11-13 11:26:58	2007-11-16 13:19:10	265932	123802585	678436427
14	2007-11-16 13:23:16	2007-11-16 13:26:04	168	213154	441223
15	2007-11-16 13:26:32	2007-11-16 13:27:12	40	16169	24090
16	2007-11-16 14:23:42	2007-11-16 16:02:39	5937	1929633	7180358
17	2007-11-16 16:03:26	2007-11-16 16:13:09	583	729043	1550283
18	2007-11-16 16:13:27	2007-11-18 10:05:03	150696	75898279	525683751
19	2007-11-18 10:56:38	2007-11-18 10:56:58	20	4208	70151

Extra Services

	Title	Description	Amount
1	Sales	Percentage charge of total sales for this month	9

Extra Services Total
9

Activity Totals

Total Time	702589
Total TX	5549431356
Total RX	9984832309

Costs

Item	Value	Usage	Amount
Subscription	50.00	1	50.00
Time / Second	0.000000000	702589	0.00
Bytes TX / byte	0.000000000	5549431356	0.00
Bytes RX / byte	0.000000000	9984832309	0.00
Extra Services	9	1	9
Demogr Discount	11.8	20%	-(11.8)
		TOTAL	47.20

Figure 6:15 : A sample invoice for a user

Figure 6-15 shows an invoice generated for a user. The invoice contains information concerning the user, the billing plans, the amounts charged and the various types of charges. To enable future explorations of the billing plans and billing metrics, all the invoices generated are kept in the system and a summary of billing metrics used on that particular invoice is stored in a database.

In this invoice, the user earned a 20% off discount after the demographics adapter was applied. This information can be recorded to see the effects of the adapters and hence allow for improvements and modifications.

6.7. Setting up local DNS

In the initial versions, the user was forced to keep open the browser page that comes out after successful login (Figure 6-9). In one of the implementations closing this pop up meant that session was also terminated. In the other one closing this pop meant that a user will not be able to log off and the session was terminated after timing out. These implementations were not good enough. We can imagine situations where we charge based on time and the user accidentally closes the pop up. In the second implementation, this means that a user will be charged for time he did not use.

Fortunately, this problem was solved in the latest implementations of hotcakes by setting up a local DNS. The code snippet shown below shows the zones that were set up to enable the user to either login, logout or go to any page on the billing system anytime regardless of the pop up. This means that closing the pop up does not have any effect on the session.

```
dwesa.com. IN SOA al-tsc-fs01.ufh-domain.local.dwesa.com. admin.dwesa.com.(
        2007031001
        28800
        3600
        604800
        38400
)

dwesa.com. IN NS al-tsc-fs01.ufh-domain.local.dwesa.com.

al-tsc-fs01.ufh-domain.local IN A 192.168.182.1
login      IN A 192.168.182.1
info       IN A 192.168.182.1
exit       IN A 192.168.182.1
```

In our testing environment we used the university name server `al-tsc-fs01.ufh-domain.local.` When a user issues a browser request by typing ‘logout’ in the address bar, the request is handled by the local DNS and the user is redirected to the logout page, otherwise the request is forward to `al-tsc-fs01.ufh-domain.local`.

6.8. Other system features

The system provides other tools which enable easy administration of users using the network resources. For example, before the user can do anything on the network he has to be logged on to a local machine. This means that there is need for easy user administration. To facilitate this we created a script which adds Linux users from a web client. As users come to register for on the network, both a Linux account and a captive portal account are created.

The code snippet shown below shows that after entering user details on a web form and pressing submit, the chosen password is encrypted using the DES algorithm and stored in a variable $crypt. After we have the password in its encrypted form, the PHP shell_exec command is executed with the various commands necessary for creating the user. However, Apache has to be given the necessary privileges to perform this task by adding it as a NOPASSWD SUDOER user. The necessary security mechanisms also needs to be implemented.

```
...
...
       $crypt = crypt($pass,gensalt(des));
        //other sanity check

       echo "<pre>". shell_exec("/usr/bin/sudo /usr/sbin/useradd -d
/home/".strtolower($login)." -p".$crypt." -g dwesa -s /bin/bash -m -c
\"".$fullname."\" ".strtolower($login))."</pre>";

...
...
```

6.9. Dwesa deployment plan

To deploy the system in Dwesa, each school needs to have an access controller which forces users to authenticate. The RADIUS server is located at a central place, say where the access concentrator is (Figure 2-6). However, a second RADIUS server may be placed on any strategic location and failover configurations set to enable one RADIUS server to takeover when one is down.

The backend against which the RADIUS server will authenticate users may also be installed on the access concentrator. However, for redundancy purposes, each school might have its own local copy of the database. With this, failure of one backend does not affect the whole network. This is also true with the RADIUS server. Because the RADIUS server supports failover and load balancing, servers might be located in strategic locations to create a robust distributed billing solution.

6.10.Conclusion

In this chapter we presented the implementation details of a prototype of our system. Different open source components were used as building blocks of the system. Hotcakes provided most of the functionality needed. We have shown how we hooked into hotcakes to include non-conventional billing metrics. These help contextualise the billing processes and start the process of finding billing strategies that are considered fair by the community.

In the next chapter we discuss issues which need to be solved after the billing system is in place and other issues related to the capacity or power of the system to produce the desired effects and its limitations.

Chapter 7

Discussion

7.1. Introduction

The thesis has discussed the development and implementation of a prototype of a Network Revenue Management System and has shown how most of the requirements mentioned in chapter 3 have been met. This chapter gives a general discussion of issues related to the implemented prototype. It highlights the effectiveness of the prototype as a solution to the problem of revenue generation for sustainability purposes in ICT4D projects. Moreover, some of the limitations of the system prototype are also highlighted.

7.2. Efficacy of the architecture and system prototype

The proposed architecture and the subsequent prototype were based on the set of requirements presented in Chapter 3. The efficiency and effectiveness of the designed system relies in its ability to satisfy the functional and non-functional requirements posed in the requirements stages.

The proposed architecture is based on the MVC architecture, making it possible to design well modularised and decoupled applications.

We used FOSS as it is the most viable solution for poor and marginalised communities whose capacity to generate income is very low. The costs of the network have to keep as minimal as possible. This means proprietary solutions are not an option since there will be the need to pay for licences.

Our generalised, flexible billing architecture enables us to design billing applications where billing information can be derived from all sorts of information repositories. This means that in addition to the traditional billing options, we are able to create custom and contextualised billing options. With a pool of billing options available, it is possible to experiment with different options and see their effects. This makes it possible to include different types of incentives aimed at motivating the community hosting the ICT facilities to use them and pay for them.

The general architecture of the system components makes it easy to add new point-of-presence (POP) onto the network, an important need of a live distributed installation. Each POP will need an access controller as the minimum requirement to connect to the billing system. This makes the system distributed and scalable.

By keeping all the statistics and information about invoices generate, it is possible to study the billing options used and see their implications either on usage patterns or on revenue generated.

7.3. Limitations

Due to time constraints we did not manage to deploy the system in Dwesa. Instead a small testing environment deployment, described in the system a walk through section, was set up in a lab.

The Dwesa network has a star topology. Even though failover and redundancy mechanisms can be implemented, the topology implies that failure of central systems affects the whole network.

Captive portals provide a rather low security mechanism where there is only initial authentication of users but no subsequent authentication of the data sent during the session. Furthermore, this solution does not offer encryption of traffic, hence the need for overlaying such a solution with VPN technologies such as PPTP, L2TP and IPSec [87] in future versions of the system.

The documentation of CakePHP is not very good and there are few tutorials which work with the latest version of the framework. This means that one might need to dig deeper into the source code and go back and forth through the API. This is also the case with the other tools used in this prototype (Chillispot and FreeRADIUS). However, in most cases, as long you are doing your things in the standard way, solutions will be found in forums.

The prototype application is based on PHP and so there may be at times unacceptable overhead on the web server, since PHP is interpreted on the fly, as opposed to other programming languages such as .NET, where the application is compiled once and from there onwards work from the deployed executable. The problem can be alleviated by making use of caching, but caching can create its own problems, such as inconsistencies in data.

After the billing infrastructure is in place, there are a lot of other issues which needs to be addressed. In this chapter we discuss some of these issues and suggest some tentative solutions. A more complete discussion can be found in APPENDIX B.

7.4. Dwesa Internet usage patterns

After the deployment of WiMAX, some of the main schools were not switching on their servers, thereby prohibiting Internet access and other inter-network communications in the network. For example, if the school hosting the VoIP server does not switch it on, there will not be any VoIP service running on the network. If the school hosting the access concentrator does not switch on its server, the schools will not be able to communicate at all.

Data was collected for the months February to August 2007, to analyse the Internet usage patterns in this network. The table shown below shows the usage of the Internet in February 2007.

Logon time	Logoff time	Service	Duration	NAS	Port	Address	Bytes Downloaded	Bytes Uploaded	Disconnect reason
Wed Feb 28 14:34:09 2007	Thu Mar 1 14:34:09 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	39 762 356	12 261 082	Session-Timeout
Tue Feb 27 14:20:05 2007	Wed Feb 28 14:20:05 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	8 835 209	2 066 627	Session-Timeout
Mon Feb 26 14:19:08 2007	Tue Feb 27 14:19:08 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	20 684 649	2 774 446	Session-Timeout
Fri Feb 23 10:06:11 2007	Sat Feb 24 10:06:11 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	85 428	12 088	Session-Timeout
Wed Feb 21 13:51:48 2007	Thu Feb 22 13:51:48 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	16 245 638	2 652 239	Session-Timeout
Tue Feb 20 09:50:11 2007	Wed Feb 21 09:50:11 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	76 339 403	8 082 181	Session-Timeout
Tue Feb 20 09:45:32 2007	Tue Feb 20 09:47:44 2007	SSG	132	196.43.10.237	16777431	10.24.6.112	18 407	3 760	User-Request
Mon Feb 19 08:27:16 2007	Tue Feb 20 08:27:16 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	31 076 924	3 115 548	Session-Timeout
Thu Feb 15 13:52:19 2007	Fri Feb 16 13:52:19 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	44 166 663	6 177 164	Session-Timeout
Wed Feb 14 13:48:19 2007	Thu Feb 15 13:48:19 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	71 539 619	9 270 417	Session-Timeout
Tue Feb 13 13:42:29 2007	Wed Feb 14 13:42:29 2007	SSG	86400	196.43.10.237	16777431	10.24.6.112	5 617 583	1 840 988	Session-Timeout
Tue Feb 13 11:03:22 2007	Tue Feb 13 12:59:58 2007	SSG	6996	196.43.10.237	16777431	10.24.6.112	5 029 957	591 681	User-Request

Table 7:1: Internet usage statistics for February 2007

The data shows that in February, the Internet was used on 12 days and the days were working days. The usage patterns depicted in figure 7-1 shows that the connection stayed on for 24hrs until it timed out automatically. This shows that the users were not logging off from the VSAT; rather, they were just switching off the machines and leaving. Consequently, no reliable conclusion can be drawn from such data as we are uncertain of the actual number of productive hours that were spent online. (This problem will go away when the Network Revenue Management System will go live on the Dwesa network).

The usage patterns based on the number of days the Internet was used can be summarised by the diagram shown below.

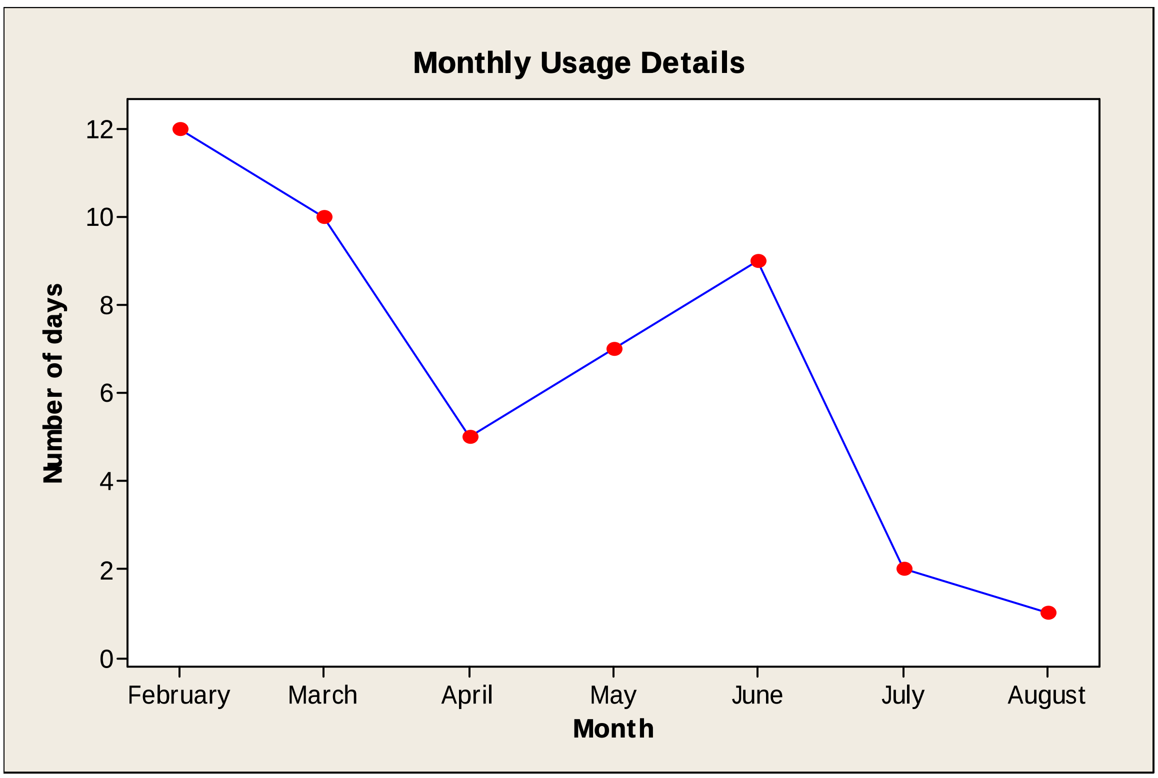
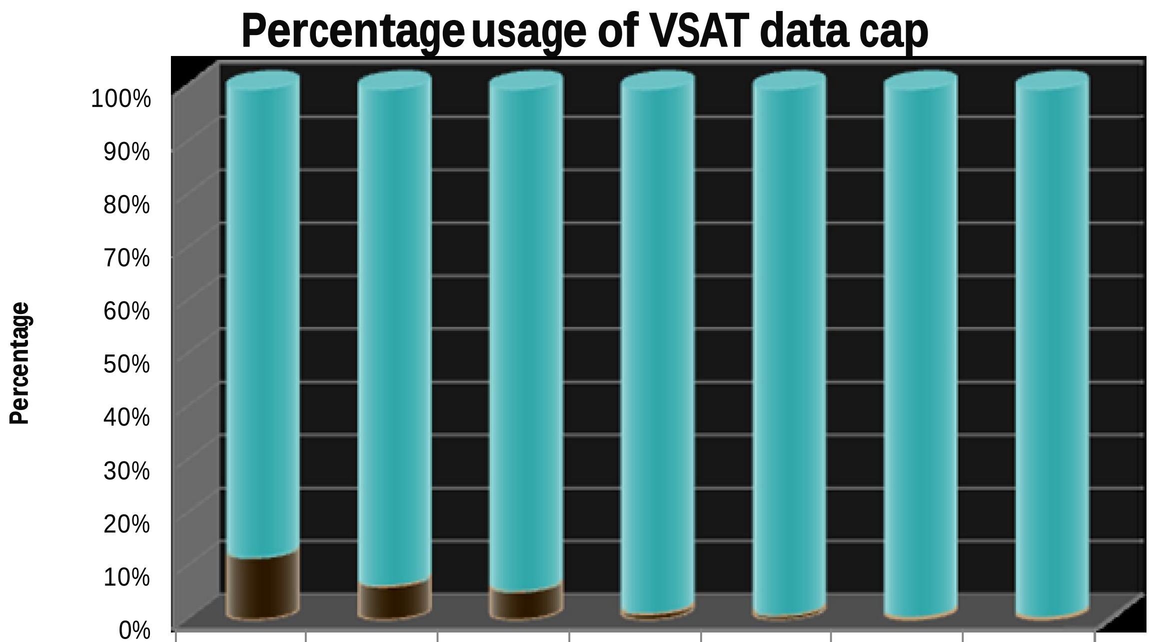


Figure 7:1: Number of days of Internet usage per month

This shows that the frequency of Internet usage was very low.

Another source of interesting information is the percentage usage of the VSAT data cap. (The 'data cap' is the maximum amount of traffic allowed per month, in the case of Dwesa 3 GB.) Again, this is based on the data for months February to August 2007.



	February	March	April	May	June	July	August
Not used	88.3	93.7	94.7	98.9	99.2	99.6	99.6
Used	11.7	6.3	5.3	1.1	0.8	0.4	0.4

Figure 7:2: Percentage usage of VSAT data cap

The diagram above shows the percentage usage of the VSAT data cap. As one can see, the account is extremely underutilised. This raises lots of questions such as: was this the right choice of account for a start? Should we keep on using this account option? What are the causes of such pattern? Maybe a lesson that can be drawn from this is to start with smaller and cheaper packages, and only upgrade to meet growing demand. This will effectively lower the costs.

A maybe surprising observation is that in the Dwesa community fewer men were using the infrastructure than women. Another notable pattern is the absence of older people at the facilities, the few who were coming were coming for organisational issues, but most stopped coming at all.

Chapter 8

Conclusion

8.1. Introduction

This chapter sums up the work undertaken in this thesis, articulates the extent to which the research meets the research goals, and it makes suggestions for future work.

As many schools of thought have come to realise that ICTs play an important role in the socio-economic development of marginalised communities, many developmental ICT projects have sprouted across nations. Recent trends are increasingly showing that shared connectivity models have become the generally accepted model of delivering sustainable ICT infrastructure in marginalised communities. Even though it sounds like utopia in the light of many experiences, the ICT infrastructure in the target communities should be financially self-sustaining, as they are in any other community, if they are to gain wide acceptance and diffusion, and realise their goals of upliftment and development. On the other hand, it is probably inevitable that these projects need external capital injection during their infancy stages. This implies that rural ICT developmental projects should be nurtured to reach a self-sustaining level.

As part of addressing the sustainability problem, this research implemented a platform for cost recovery, cost sharing and ultimately, revenue generation. Specifically, it provided an architecture and the implementation a proof-of-concept prototype based on this architecture. The system prototype will enable the Dwesa community to find ways, hopefully, to generate revenue and share the costs of running the network.

8.2. Achievements

To build this system, a comprehensive literature survey and requirements elicitation was conducted. This culminated into a requirement specification document. This specification was then used as the basis for the design of Network Revenue Management Architecture. Concomitant to the architectural design, a prototype system was implemented to test and demonstrate the architecture.

The design of the architecture was based on the MVC architecture which separates the modelling of domain, presentation and actions. This helps reduce coupling between the different components. The efficient modularity of the design allows any of the components to be swapped around as the user or developer wishes. Furthermore, because the modules are decoupled, these changes will not affect other parts of the system. This means that, the architecture is inherently flexible and capable of easy growth, which makes it applicable to various scenarios.

To demonstrate the feasibility and validity of the proposed architectural solution, a system prototype was implemented. This prototype was derived from and designed to take advantage of the currently available applications, technologies and protocols. Specifically, it is built upon existing, good, open source, authentication and billing infrastructure. More importantly, this prototype features a novel and flexible rating engine which allows for users to be charged based on conventional and non-conventional billing metrics. In other words, we implemented a generalised, flexible billing system where billing information could be derived from all sorts of information repositories.

From an implementation point of view, the system satisfies the objectives of the project as specified in Chapter 1 and the requirements specification. The implementation chapter highlighted that the system is a flexible, low cost, open source solution which allows for charging for network resource use. The system provides features such as:

- An interface for managing system users. This interface allows for the administration of subscribers, cashiers and administrators of the system.
- Features for authenticating users. The users are forced to authenticate when necessary before being allowed access to network resources.
- Accounting infrastructure. The system is able to track usage of resources for the duration of a user's session.

- It provides a billing infrastructure which is centred around a rating manager which allows the charging of users based on either conventional or non-conventional billing metrics or both.

Besides playing a central role in allowing cost sharing and revenue generation, the system can be used to collect data on computer use habits in marginalized communities, which can be studied and analyzed. This is essential as this might as well send indicators on the areas that researchers need to pay careful attention or act on quickly as they venture rural ICT projects.

Due to time constraints and the fact that this research is still in its infancy stages, our system has a number of limitations. All of them represent directions for future work, once the system is deployed in Dwesa.

8.3. Areas of further research

The most obvious future work is the implementation of more adapters for billing. Ultimately, this will give the community a pool of billing options from which to choose, to determine what is regarded as a fair way of charging.

This work requires attending to subtasks, such as:

- A study of the network resources usage habits and patterns to make planning decisions
- A study of the responsiveness of the users to different incentives and prices.
- Given the fact that we want to use other information such as income, age, and expenditure, an investigation must be carried out on how we are going to get authentic data in this respect.

Once the adapters are written and tested in Dwesa, work will have to be done to validate them in other settings.

More future work on the technical side lies in the replacement of FreeRADIUS with a DIAMETER server - DIAMETER protocol is a next generation protocol that was derived from the RADIUS protocol. It offers improvements in a number of aspects. This protocol was designed specifically to meet the requirements IMS authentication, AAA, the IETF, 3GPP and 3GPP2 [98].

8.4. Overall conclusion

This thesis has described the design of a novel Network Revenue Management Architecture and the subsequent implementation of a system prototype in the Dwesa /Cwebe community. Most importantly, this thesis outlines a system which combines traditional and customer-oriented billing methods, to provide a larger pool of billing options which can be used for experimenting on ways of charging for network resource usage.

It is important to note that the work described in this thesis was conducted adhering to the assumption that the implementation of ICT projects in marginalised communities should not only be regarded as a technical process but also an extremely contextualised social process which takes into account trade-offs and prioritisation of goals. If it proves successful, within its scope, the thesis demonstrates the validity of this assumption, to which the author totally subscribes.

References

[1] Department of Education and Department of Communication, "Strategy for Information and Communication Technology in Education," Government Printer, Pretoria 2001, http://www.info.gov.za/otherdocs/2001/ict_doe.pdf.

[2] S. Parkinson, "Telecentres, Access and Development: Experience and Lessons from Uganda and South Africa," IDRC, 2005, http://www.idrc.ca/en/ev-87255-201-1-DO_TOPIC.html.

[3] P. Tarwireyi, A. Terzoli, and M. Thinyane, "Implementation of an Internet Access Cost Management system for disadvantaged communities," in *SATNAC* Mauritius, Sugar Beach Resort, 2007.

[4] M. L. Best and C. M. Maclay, "Community Internet Access in Rural Areas: Solving the Economic Sustainability Puzzle," Oxford University, Oxford 2003, http://edev.media.mit.edu/SARI/papers/gitrr2002_ch08.pdf.

[5] P. Benjamin, " African Experience with Telecenters," *OnTheInternet,* October 2000, http://www.isoc.org/oti/printversions/1000benjamin.html.

[6] S. Parkinson, *Telecentres, Access and Development: Experience and Lessons from Uganda and South Africa*: IDRC, 2005.

[7] G. Hearn, M. Kimber, J. Lennie, and L. Simpson, "A Way Forward:Sustainable ICTs And Regional Sustainability," *The Journal of Community Informatics,* vol. 1, pp. 18-31, 2005, http://ci-journal.net/index.php/ciej/article/viewFile/201/160.

[8] A. Terzoli and H. Muyingi, "THRIP application," *Telkom Centre of Excellence in developmental e-commerce, University of Fort Hare* 2004, http://cs.ufh.ac.za/coe/.

[9] BLACKROCK COMMUNICATION, "Innovation Hub incubation success," in *ITWEB*, 11 November 2004 http://www.itweb.co.za/sections/business/2004/0411110930.asp?A=INV&S=Innovations&O=FPIN.

[10] M. F. T. Mendoza, " THE GLOBAL DIGITAL DIVIDE: EXPLORING THE RELATION BETWEEN CORE NATIONAL COMPUTING AND NATIONAL CAPACITY AND PROGRESS IN HUMAN DEVELOPMENT OVER THE LAST DECADE ". vol. Doctor of Philosophy: Tulane, 2001, http://studentweb.tulane.edu/~mtruill/diss/Final.pdf.

[11] American Library Association, "New FCC Chair Downplays Digital Divide," 2001, http://www.ala.org/ala/alonline/currentnews/newsarchive/2001/february2001/newfccchairdownplays.cfm.

[12] R. H. Wade, " Bridging the digital divide: new route to development or new form of dependency?," *Global Governance,* vol. 8, 2002.

[13] G. Accascina, "Information technology and poverty alleviation: The role of information and communication technologies in rural development and food security," United Nations (FAO), 2000, http://www.fao.org/sd/CDdirect/CDre0055h.htm.

[14] C. I. Pade, B. Mallinson, and D. Sewry, "An Exploration of the Categories Associated with ICT Project Sustainability in Rural Areas of Developing Countries: A Case Study of the Dwesa Project," in *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries* Somerset West, South Africa: South African Institute for Computer Scientists and Information Technologists, 2006.

[15] E. H. Blake and W. D. Tucker, "Bridging communications across the Digital Divide," in *SATNAC*, 2002, http://pubs.cs.uct.ac.za/archive/00000156/01/finalBridging.pdf.

[16] L. Dalvit, M. Thinyane, H. Muyingi, and A. Terzoli, "The Deployment of an e-Commerce Platform and Related Projects in a Rural Area in South Africa," *International Journal of Computing and ICT Research,* vol. 1, pp. 9-18, 2007, http:www.ijcir.org/issue1/article 2.

[17] G. S. V. R. K. Rao, A. Terzoli, H. Muyingi, and M. T. Mandioma, "A study on Addressing Digital Divide through ICT initiatives in South Africa, ICTe Africa ", 2006.

[18] UNDP, "Human Development Report," 2001.
[19] UNDP, "Millennium Development Goals - MDGs," 2007, http://www.undp.org/mdg/.
[20] Raj Reddy, Rahul Tongia, Eswaran Subrahmanian, N. Balakrishnan, and V. S. Arunachalam, "Sustainable ICT for Emerging Economies Mythology and Reality of the Digital Divide Problem," Carnegie Mellon University, Pittsburgh, 2004.
[21] UNDP, "Telecentre Sustainability. Financing ICTs for the poor," in *APDIP e-Note 15*, 2007, http://www.apdip.net/apdipenote/15.pdf.
[22] CSIR, "DST and CSIR showcase three South African ICT projects during WSIS," 2003, http://www.meraka.org.za/news/DST_and_CSIR.html.
[23] G. Bruntland, "Our Common Future: The World Commission on Environment and Development," Oxford University Press, Oxford 1987.
[24] Batchelor and Norrish, "Sustainable Information and Communication Technology (ICT)," D. o. I. Development, Ed., 2002, http://www.sustainableicts.org/Sustainable.htm.
[25] H. G. Timmermans, "Rural livelihoods at Dwesa/Cwebe : poverty, development and natural resource use on the Wild Coast, South Africa." vol. Masters of Science Grahamstown: Rhodes, 2004, p. 196, http://eprints.ru.ac.za/78/01/timmermans-msc.pdf.
[26] R. Palmer, H. Timmermans, and D. Fay, "From conflict to negotiation: nature-based development on South Africa's Wild Coast," in *Human Sciences Research Council* Pretoria, 2002.
[27] A. R. D. R. Institute, "Rural livelihoods survey in the Mbashe Municipality," Univerversity of Fort Hare, Alice 2001.
[28] T. L. S. Organisation, "Rights enshrined but right denied? Post-settlement struggles in Dwesa-Cwebe in the Eastern Cape," 2006, http://www.landcoalition.org/cd_ILC/documenti/EXPANDING_ADVOCACY_AND_POLICY_WORK/Legislation_and_Regulations/TRALSO_Rights_enshrined_but_rights_denied.pdf.
[29] P. Lehohla, "Provincial Profile, Eastern Cape," *Statistics South Africa,* 2004.
[30] M. T. Mandioma, G. S. V. R. K. Rao, A. Terzoli, and H. Muyingi, "A study on WiMAX Implementation at Dwesa-Cwebe Rural Areas of Eastern Cape of South Africa," in *IEEE TENCON* Hong Kong, 2006.
[31] Ingrid Siebörger and Barry Irwin, "Dwesa Networking document," October 22, 2007.
[32] K. STOLL, "Telecentres Sustainability: What does it mean?," 2003, http://topics.developmentgateway.org/ict/sdm/previewDocument.do~activeDocumentId=442773?activeDocumentId=442773.
[33] Ndlovu G, A. Terzoli, and G. Pennels, "Indigenous knowledge: a pathway towards developing sustainable ICT solutions," in *SATNAC conference* Cape Town, South Africa, 2006.
[34] V. N. Rideout and A. J. Reddick, "Sustaining Community Access to Technology: Who should pay and why?," *The Journal of Community Informatics,* vol. 1, 2005, http://ci-journal.net/index.php/ciej/article/view/202/161.
[35] Asian Development Bank, *Guidelines for the Economic Analysis of Projects*, 1997.
[36] A. Ofrane and L. Harte, *Introduction To Wireless Billing; Usage Recording, Charge Processing, System Setup, And Real Time Billing*: Althos, 2003.
[37] internet.com, "Accounting " in *Webopedia*, 2007, http://www.webopedia.com/TERM/A/AAA.html.
[38] Wikipedia, "Telecommunications rating," 2007, http://en.wikipedia.org/wiki/Telecommunications_rating.
[39] B. Stiller, G. Fankhauser, B. Plattner, and N. Weiler, "Pre-study on Customer Care, Accounting, Charging, Billing, and Pricing," in *The World Wide Web and Databases* Valencia, Spain: Springer, 1998.
[40] D. Clarke, "A Model for Cost Allocation and Pricing in the Internet," in *Internet Economics*: MIT Press, 2005.
[41] B. Stiller, T. Braun, M. Gunter, and B. Plattner, "The CATI Project: Charging and Accounting Technology for the Internet " in *Multimedia Applications, Services and Techniques - ECMAST'99: 4th European Conference, Madrid, Spain, May 1999. Proceedings*: Springer Berlin / Heidelberg, 1999.
[42] B. Stiller, P. Reichl, J. Gerke, and P. Flury, "A Generic and Modular Internet Charging System for the Cumulus Pricing Scheme," *Journal of Network and Systems Management,* vol. 9, pp. 293-325, 2001.
[43] M3I, "Market Managed Multi-service Internet (M3I)," 1999, http://www.tik.ee.ethz.ch/~m3i/.
[44] R. Edell and P. Varaiya, "Providing Internet access: What we can learn from the INDEX trial," *IEEE,* vol. 13, pp. 18-25, 1999.

[45] The Internet Society, "Realtime Traffic Flow Measurement: Applicability Statement," 1999.
[46] IPDR Consortium, "Internet Protocol Detail Record," 2007, http://www.ipdr.org/about/faq.html.
[47] B. Stiller and P. Reichl, "Pricing Models for Internet Services," in *CATI, Charging and Accounting Technology for the Internet*, 1999.
[48] T. T. T. Nguyen and G. J. Armitage, "Pricing the Internet - A visual 3-Dimensional Evaluation Model," in *Australian Telecommunications, Networks and Applications Conference (ATNAC)*, Sheraton Towers, Southbank, Melbourne, 2003, http://atnac2003.atcrc.com/ORALS/NGUYENTTT.pdf.
[49] T. Henderson, "Internet Economics – A literature Review," April 2000.
[50] K. J. MacKie-Mason and R. H. Varian, "Pricing the Internet," 1993.
[51] F. P. Kelly, "Charging and Accounting for Bursty Connections," in *MIT workshop on Internet Economics*, March 1995.
[52] N. Keon and G. Anandalingam, "A new pricing model for Competitive Telecommunications Services using Congestion Discounts," 2000.
[53] D. Clark, "Combining Sender and Receiver Payments in the Internet," in *Telecommunications Research Policy Conference*, October 1996, www.gta.ufrj.br/diffserv/csrp-ddc.ps.gz.
[54] A. M. Odlyzko, "Paris Metro Pricing for the Internet," in *The 2nd Internation conference on Information and Comupation Economies*, November 1999.
[55] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz, "An Embedded Charging Approach for RSVP," in *Proceedings of 6th IEEE/IFIP International Workshop on Quality of Service*, Napa, CA, USA, 1998, pp. 91--100, citeseer.ist.psu.edu/karsten98embedded.html.
[56] H. Chen, "Accounting Facilities in Mobile and Wireless Internet." vol. MSc: University of Twente, 2003, p. 90, http://dacs.ewi.utwente.nl/assignments/completed/master/reports/Thesis_Hong_Chen.pdf.
[57] G. Carle, F. Hartanto, M. Smirnov, and T. Zseby, "SUSIE - Charging and Accounting for QoS-enhanced IP Multicast," in *Proceedings of the IFIP TC6 WG6.1 & WG6.4 / IEEE ComSoc TC on on Gigabit Networking Sixth International Workshop on Protocols for High Speed Networks VI*, August 25-27, 1999 pp. 151-168.
[58] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, and L. Merakos, "A Holistic Solution for Charging, Billing & Accounting in 4G Mobile Systems," in *Vehicular Technology Conference,*, VTC 17-19 May 2004 IEEE 59th, pp. 2257 - 2260
[59] C. d. Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence, "RFC 2903 - Generic AAA Architecture," The Internet Society August 2000.
[60] Sander Hille, Henk Jonkers, Andrew Tokmakoff, and Martin Wibbels, "State-of-the-Art in Electronic Accounting, Billing and Payment," Telematica Instituut, 2000.
[61] Wikipedia, "Netflow," 2007, http://en.wikipedia.org/wiki/Netflow.
[62] Sander Hille, Henk Jonkers, Roel Stap , and Arjen Bakker, "A Flexible Architecture for Inter-domain Accounting, Billing and Payment," Telematica Instituut, 2002, https://doc.telin.nl/dsweb/Get/Document-19825/.
[63] N. Isabirye, R. Roets, and H. Muyingi, "Eliciting user requirements for ecommerce applications for entrepreneurs in rural areas," in *SATNAC conference* Cape Town, South Africa, 2006.
[64] A. M. Davis, "Operational Prototyping: A new Development Approach," *IEEE Software,* p. 71, September 1992.
[65] Scott McEwen, "Requirements: An introduction." vol. 2007: IBM, 2005, http://www.ibm.com/developerworks/rational/library/4166.html?S_TACT=105AGX78&S_CMP=HP.
[66] L. L. A. Wangensteen, "Using SIM for strong end-to-end Application Authentication," Department of Telematics, Norwegian University of Science and Technology, 2006.
[67] Madjid Nakhjiri and Mahsa Nakhjiri, "AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility," John Wiley & Sons, Ltd, 2005.
[68] The MITRE Corporation, "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense," 2003.
[69] P. Congdon, B. Aboba, G. Zorn, A. Smith, and J. Roese, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS)," The Internet Society 2003, http://www.ietf.org/rfc/rfc3580.txt.
[70] W. Simpson and B. Lloyd, " PPP Authentication Protocols," in *Request for Comments: 1334*: The Internet Society, October 1992.

[71] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, " Extensible Authentication Protocol (EAP)," in *Request for Comments: 3748* The Internet Society, June 2004.
[72] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," in *Request for Comments: 2865* The Internet Society June 2000, http://tools.ietf.org/html/rfc2865.
[73] Jason Gorman, "Use Cases - An Introduction," Parlezuml, 2006.
[74] R. Malan and D. Bredemeyer, "Functional Requirements and Use Cases," Bredemeyer Consulting, 2006.
[75] IEEE Computer Society, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," *IEEE Standard 1472000,* 2000.
[76] S. Njeje, A. Terzoli, and H. Muyingi, "Software Engineering of a Robust, Costeffective Ecommerce Platform for Disadvantaged Communities," in *SATNAC conference* Cape Town, South Africa, 2006.
[77] Burbeck Steve, "Application Programming in Smalltalk-80: How to use Model-View-Controller (MVC)," *University of Illinois in Urbana-Champaign (UIUC) Smalltalk Archive,* 1992, http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html.
[78] John Deacon, "Model-View-Controller (MVC) Architecture," *JOHN DEACON Computer Systems Development, Consulting & Training,* 2005, http://www.jdl.co.uk/briefings%5CMVC.pdf.
[79] Scott W. Ambler, "UML 2 Deployment Diagrams," Ambysoft Inc. 2006.
[80] A. O. Freier, P. Karlton, and P. C. Kocher, "The SSL Protocol," *Internet Draft,* 1996, http://wp.netscape.com/eng/ssl3/ssl-toc.html.
[81] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," *INTERNET-DRAFT* October 2007 http://www.ietf.org/internet-drafts/draft-ietf-tls-rfc4346-bis-06.txt.
[82] B. Cusack, "Secure Sockets Layer," *SearchSecurity.com Definitions* 06 Mar 2007, http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci343029,00.html.
[83] Microsoft, "RADIUS protocol," *Microsoft TechNet,* Jan 2005, http://technet2.microsoft.com/windowsserver/en/library/9ecf38e5-3200-490d-83d8-2c624da94d8b1033.mspx?mfr=true.
[84] C. Rigney, "RADIUS Accounting," in *Request for Comments: 2866*: The Internet Society June 2000, http://www.faqs.org/rfcs/rfc2866.html.
[85] Netcraft, "October 2007 Web Server Survey," 2007, http://news.netcraft.com/archives/web_server_survey.html.
[86] Personal Telco Project, "Captive Portal," 2007, http://wiki.personaltelco.net/index.cgi/CaptivePortal.
[87] L. Veltri, "WLAN Security Mechanisms," Department of Information Engineering, University of Parma – Italy 2005.
[88] ChilliSpot, "ChilliSpot," 2007, http://www.chillispot.info/.
[89] The FreeRADIUS Server Project, "FreeRADIUS Server," 2006, http://www.us.freeradius.org/.
[90] D. v. d. Walt, "Hotcakes Hotspot Manager," 2007, http://cakeforge.org/projects/hotcakes/.
[91] CakePHP, "What is CakePHP," 2007.
[92] Graham Bird, "How Cake Works," 2006, http://grahambird.co.uk/cake/tutorials/howitworks.php.
[93] Bruce Perens, "The Open Source Definition," *Open Source Initiative*, http://perens.com/OSD.html.
[94] S. Erle, R. Flickenger, R. Gibson, and R. Weeks, "NoCat.net," 2005, http://nocat.net/.
[95] Public IP, "Zone CD," 2007, http://www.publicip.net/.
[96] M. P. Thinyane, "Investigating the viability of a framework for small scale, easy deployable and flexible hotspot management systems," in *Department of Computer Science*. vol. MSc in Computer Science Grahamstown: Rhodes University, 2005.
[97] Asommer, "WifiDocs / ChillispotHotspot," 2007, https://help.ubuntu.com/community/WifiDocs/ChillispotHotspot.
[98] Jeffrey Liu, Steven Jiang, and Hicks Lin, "Introduction to Diameter," *IBM, developerWorks, WebSphere,* January 2006.
[99] Global Insight, "Provincial Growth and Development Plan:Socio-economic profile, Eastern Cape: 2004-2014," 2007, http://mdg-guide.undp.org/popup/files/south_africa/South%20Africa_Provincial%20Growth%20and%20Development%20Plan_Section%202_Socioeconomic%20Profile.pdf.
[100] Statistics South Africa, "A natinal poverty line for South Africa," *National Treasury,* 21 February 2007.

Appendix A

Network Revenue Management System

Software Requirements Specification

Version 3.0

Follows IEEE Standard 830

Revision History

Name	Date	Reason For Changes	Version
Paul Tarwireyi	10/05/2006	Initial system requirements	1.0
Paul Tarwireyi	27/09/2007	Updating the System Requirements Specification	2.0
Paul Tarwireyi	18/10/2007	Making corrections to some of the sections and fixing typo errors after feedback from Mamello.	3.0

1. Introduction

1.1. Purpose

This document is intended to capture the requirements that must be embodied in the "Network Revenue Management System, referred to as "the system" hereafter. It gives a description of the behaviour of the system. Presentation of what the system should do is done in a systematic manner that facilitates understanding by the users of the system. Consequently this document will be used by end users and developers to validate that all requirements for the system are clear and well understood.

The requirements highlighted in the document will be used as the basis for subsequent design and development of the Network Cost Management System. The intended audience is the analyst, programmer, project managers and testers of the systems.

1.2. Project Scope

The proposed name of the system is "Network Revenue Management System". The software product will be capable of charging users for the use of network resources in order to generate revenue. The revenue generated will be used to cover the costs of running the network and other associated costs, thus contributing towards economically sustaining the network.

The system will provide easier control of users' access to and usage of network resources by abstracting the underlying operations and processes required to perform these tasks. It will provide the users with a pool of pricing options from which they can choose. The system will encapsulate unconventional pricing metrics in it charging schemes. It is anticipated that this will enable the exploration of different pricing models to determine what is regarded as a fair way of charging in the rural marginalized communities of developing nations. The system should also allow users to roam from one WISP to another.

This system will include existing applications and databases, such as Freeradius Server, Chillispot, Hotcakes and the Dwesa e-commerce portal and will have interfaces for process intercommunications.

1.3. Definitions, Acronyms, and Abbreviations

1.3.1. Free and Open Source Software (FOSS)

Free and open source software is software that gives users the privilege to run, copy, distribute, study, change, and improve it as they wish, without them having to ask for permission from or making additional payments to anyone (The MITRE Corporation, 2003).

1.3.2. ICT for development

This is the exploration of ICT as an enabler for community development. It encompasses the implementation of projects that aim to leapfrog economic growth and overall community social well-being.

1.3.3. Sustainability

Sustainability is a characteristic of a process or state that can be maintained at a certain level indefinitely. This means that "sustainable development" would be development of economic systems that last indefinitely (Wikipedia, 2007).

1.3.4. Access control

Access Control is any mechanism that a system uses to grant or deny the right to access some data. Normally the users first identify themselves to the system by issuing an authentication token (usually a username and a password) and the system then verifies the claimed identity (Authentication).The access control mechanism controls what the user may or may not do on the system.

In any access control model, the entities that perform actions are called subjects, and the entities representing resources to which access may need to be regulated are called objects.

Access Control systems provide Identification, Authentication, Authorization and Accountability (Wikipedia, 2007).

1.3.5. Identification and Authentication

This is a two step process to determine who can log on to the system and the association of users and their software subjects (Wikipedia, 2007).

1.3.6. Authorization

Determines what the subject can do on the system.

1.3.7. Accounting

Determines what a subject did. This is a process of identifying all that was done by subjects by using components such as audit trails and log files. The subjects are then linked to the controlling user (Wikipedia, 2007).

1.3.8. Billing /Rating

For better understanding of the system, the distinction between rating and billing needs to be clarified.

Rating is defined as the assessment or evaluation of something, in terms of quality, quantity or some combination of both. In telecommunications this is the process of determining the cost of usage of a service. It involves converting usage-related data (bits, bytes, time, origin and destination) into a monetary equivalent value (Wikipedia, 2007).

The rated usage records are associated with their respective users (Accounting) and aggregated.

Billing is the process of generating an invoice which is a collection of the rated or aggregated usage records and sending or making it available to the user.

1.4. References

The MITRE Corporation. (2003). *Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense.*

Wikipedia. (2007). *Access Control*. Retrieved August 28, 2007, from Wikipedia: http://en.wikipedia.org/wiki/Access_control

Wikipedia. (2007). *Telecommunications rating*. Retrieved August 28, 2007, from Wikipedia: http://en.wikipedia.org/wiki/Telecommunications_rating

1.5. Overview

This document is structured in way that should clearly portray the developer's view of the system. The remaining sections of this document describe the requirements of the system both from the external perspective of end users and interfacing systems, and from the internal perspective of functional requirements.

Section 2 of this document gives an overall description of the system with particular focus on the environment in which it will operate. Key system functions are introduced and characteristics of potential users are explored. Any constraints and assumptions made by the developer will be stated.

Section 3 gives the specific requirements of the product and discusses the external requirements and gives detailed descriptions of functional requirements.

The software requirements specification (SRS) enables us to apply due-diligence to analyze needs and requirements and provides a solid framework within which to develop applications which facilitates future development efforts.

2. Overall Description

2.1. Product Perspective

The application described in this SRS is a community-based Network Accounting and Charging System that will be deployed in the Dwesa / Cwebe area of Transkei. The system will focus on providing a way of recovering the costs of running the Internet network that was deployed in that area. It also provides a modularized plug-in architecture to explore various billing schemes. It is expected that through this billing infrastructure, the users of the network will be able to cater for the up-keep of the network thus ensuring its continued existence and sustainability.

The product is part of a project being undertaken jointly by the University of Fort Hare and Rhodes University to develop and field-test a prototype of a simple, cost-effective and robust, integrated e-commerce / telecommunications platform for marginalized communities in South Africa.

The project will manage the network that was deployed in Dwesa in 2006.

2.2. Product Features

The system will function as a Network billing system for the network. The system will be based on a client server architecture. The servers will placed on various strategic locations on the network. Clients send request to servers for services like DHCP, AAA, Netboot, HTTP, HTTPS and VoIP. The responsible servers will handle the requests, perform specified actions and then send back the responses to the requesting clients.

2.3. User Characteristics

The user of the system should be capable of interacting with computer software easily. For the client side this fairly easy, because it is a matter of submitting authentication credentials and maybe checking current balances and usage statistics. In order to facilitate ease of use on the administration side, the user will have to be familiar with billing operations. If needed, the user should be trained on various networking aspects such as bandwidth and time as measures of usage.

2.4. Design and Implementation Constraints

Since this system is deployed as a client-server architecture, it will require a web server capable of handling PHP applications and SSL/TSL, a RADIUS server, and an access controller. Since the network deployed in Dwesa run on a Linux platform, the system implementation is also based this platform. Given the fact that the system is a subsystem to the ecommerce platform project, it needs to be integrated with other systems to minimize duplicate data maintenance where possible. In particular input from other systems should be done by common subsystems to reduce the burden of data entry on the end users.

2.5. Assumptions and Dependencies

For now, it will be assumed that the system will have two kinds of users, an administrator and a general user.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

User Interfaces will employ graphical icons, visual indicators along with text and labels which are clear and free of jargon to represent information and actions available to a user. The actions will be performed through direct manipulation of the graphical user interfaces (GUI). This enhances the usability of the underlying logical design contained in the program.

3.1.2. Software Interfaces

The system will use free radius server for AAA and use Mysql as its backend database. It will also need to communicate with other systems like the e-commerce portal to get its input data for communal based charging methods.

3.1.3. Communications Interfaces

The users will access the system with Windows, Linux based or other suitable devices through the LAN or WAN. Standard protocols likely to be used in this project include TCP/IP, HTTP, HTTPS, ARP, PAP, CHAP and EAP.

3.2. Functional requirements

3.2.1. User class 1 (General user – user that accesses and use network resources)

3.2.1.1. Feature 1 (Join the Network)

3.2.1.1.1. Introduction/Purpose of Feature

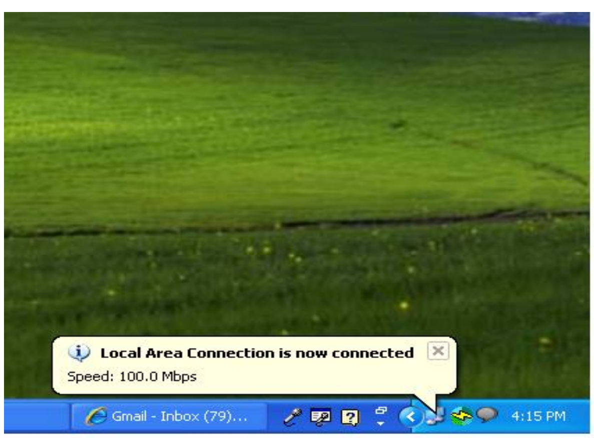
This feature describes what happens when a user joins the LAN.

3.2.1.1.2. Stimulus/Request sequence

When a client machine starts up or join the network, it will issue a new DHCP request to the servers' virtual interface tun0 which is a PPPO interface. The server will respond with a specified DHCP offer. After receiving the IP address the client is then able to access services like HTTP and HTTPS hosted by the server.

3.2.1.1.3. Associated functional requirements

After the client machine has successfully connected and joined a network, the user will see a



connection success notification screen such as below. In the case of Linux clients, there are various ways of checking connectivity.

3.2.1.2. Feature 2 (Make a connection outside LAN)

3.2.1.2.1. Introduction/Purpose of Feature

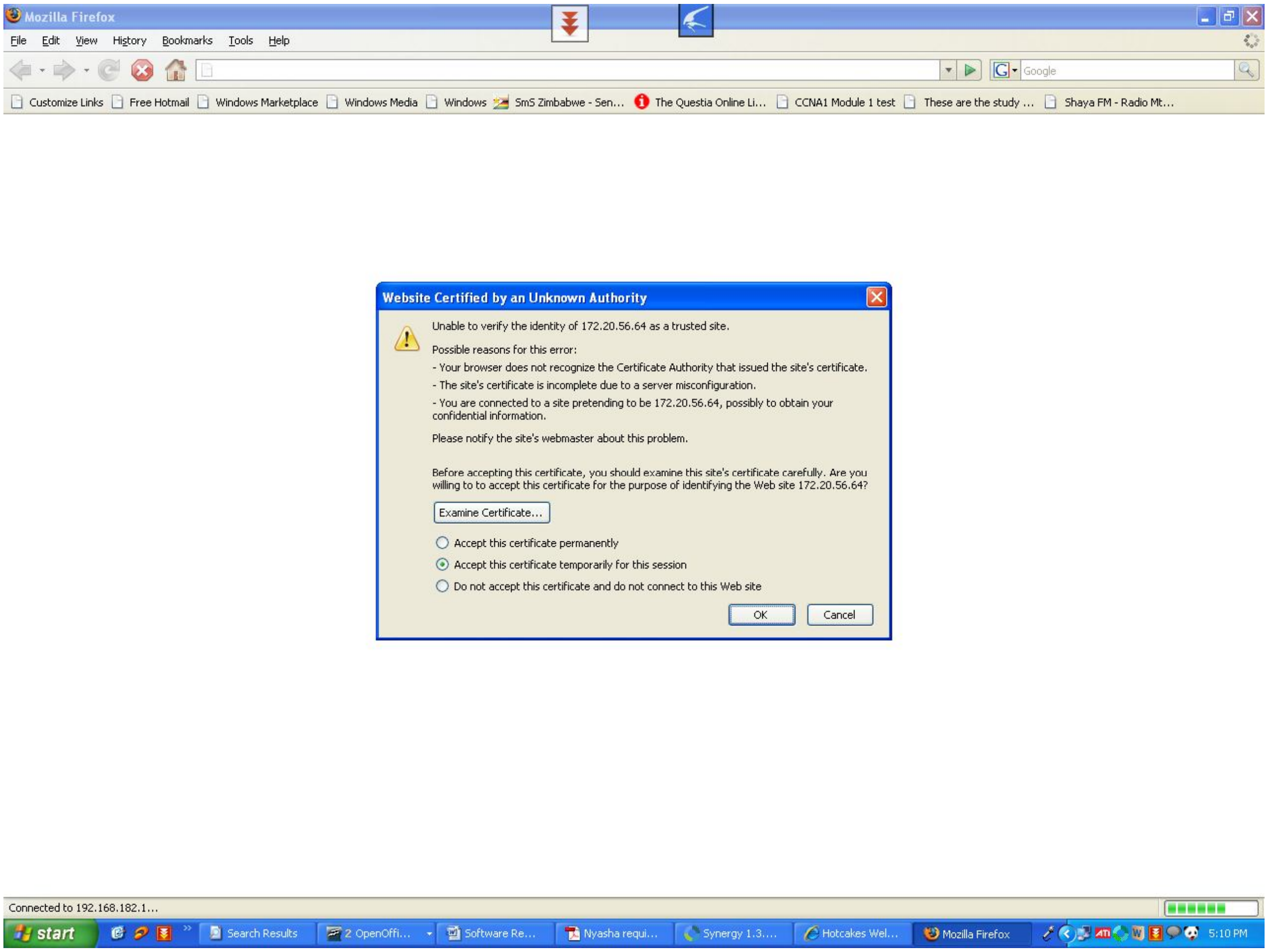
This feature describes what happens when a user want to access network resources outside his LAN.

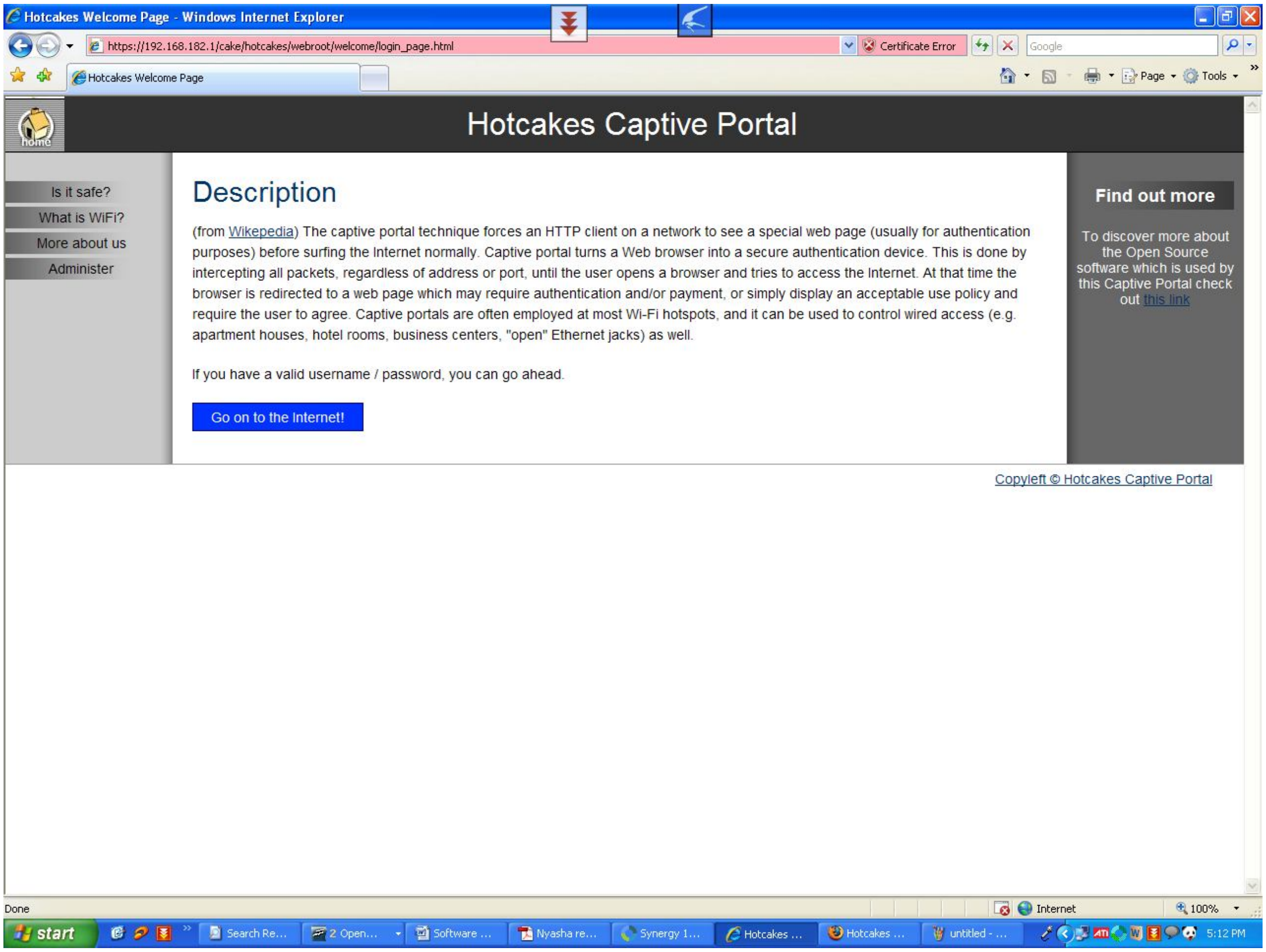
3.2.1.2.2. Stimulus/Request sequence

When a user tries to connect to the Internet or network resource outside his LAN, his TCP/IP session is intercepted by the captive portal and redirected to a login page or a home page explain to the user how to gain access to the desired resource. Just before a user can be taken to these pages he needs to verify the server by accepting the certificate issued by the server (mutual authentication).

3.2.1.2.3. Associated functional requirements

When a client fires request for a resource he is redirected to another page shown below


Mozilla Firefox
File Edit View History Bookmarks Tools Help
Customize Links Free Hotmail Windows Marketplace Windows Media Windows SmS Zimbabwe - Sen... The Questia Online Li... CCNA1 Module 1 test These are the study ... Shaya FM - Radio Mt...
Website Certified by an Unknown Authority
Unable to verify the identity of 172.20.56.64 as a trusted site.
Possible reasons for this error:
- Your browser does not recognize the Certificate Authority that issued the site's certificate.
- The site's certificate is incomplete due to a server misconfiguration.
- You are connected to a site pretending to be 172.20.56.64, possibly to obtain your confidential information.
Please notify the site's webmaster about this problem.
Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the Web site 172.20.56.64?
Examine Certificate...
Accept this certificate permanently
Accept this certificate temporarily for this session
Do not accept this certificate and do not connect to this Web site
OK
Cancel
Connected to 192.168.182.1...
start
Search Results
2 OpenOffi...
Software Re...
Nyasha requi...
Synergy 1.3....
Hotcakes Wel...
Mozilla Firefox
5:10 PM


Hotcakes Welcome Page - Windows Internet Explorer
https://192.168.182.1/cake/hotcakes/webroot/welcome/login_page.html
Certificate Error
Hotcakes Captive Portal
Is it safe?
What is WiFi?
More about us
Administer
Description
(from Wikepedia) The captive portal technique forces an HTTP client on a network to see a special web page (usually for authentication purposes) before surfing the Internet normally. Captive portal turns a Web browser into a secure authentication device. This is done by intercepting all packets, regardless of address or port, until the user opens a browser and tries to access the Internet. At that time the browser is redirected to a web page which may require authentication and/or payment, or simply display an acceptable use policy and require the user to agree. Captive portals are often employed at most Wi-Fi hotspots, and it can be used to control wired access (e.g. apartment houses, hotel rooms, business centers, "open" Ethernet jacks) as well.
If you have a valid username / password, you can go ahead.
Go on to the Internet!
Find out more
To discover more about the Open Source software which is used by this Captive Portal check out this link
Copyleft © Hotcakes Captive Portal
Done
Internet
100%
start
5:12 PM

3.2.1.3. Feature 3 (Login)

3.2.1.3.1. Introduction/Purpose of Feature

The main purpose of this feature is to enable users to authenticate against the system and be allowed to access the network resources according to the specific users' profile.

3.2.1.3.2. Stimulus/Request sequence

A user enters his username and password. A NAS in the WISP he belongs to collects these credentials and sends them to the RADIUS server, together with any required NAS details. The RADIUS server checks the credentials against its backend and creates a tunnel for user to access the requested resources if authentication is successful.

3.2.1.3.3. Associated functional requirements

Only authenticated users will have access to the network resources. Once a user is authenticated the system will allow him access to network resource according to his profile.

3.2.1.4. Feature 4 (Check balance)

3.2.1.4.1. Introduction/Purpose of Feature

The main purpose of this feature is to allow the user the self-service of checking their own balances anytime they want.

3.2.1.4.2. Stimulus/Request sequence

A user clicks the check balance link and the system responds by giving him his invoice showing all the charges. The system should regularly display to the user information about current usage and remaining time.

3.2.1.4.3. Associated functional requirements

It should be possible for users to check their current balances and current usage statistics. This will help them adjust their usage habits as they see fit.

3.2.1.5. Feature 5 (Logout)

3.2.1.5.1. Introduction/Purpose of Feature

The main purpose of this feature is to allow a user to logout of the system at any point in time.

3.2.1.5.2. Stimulus/Request sequence

A user clicks on the logout link to logout of the system.

3.2.1.5.3. Associated functional requirements

The user should be able to kill his session when he has finished using the resource. This is essential because we don't want the user to continue being billed when he is no longer using the resource.

3.2.2. User class 2 (administrator)

3.2.2.1. Feature 1 (Login)

3.2.2.1.1. Introduction/Purpose of Feature

The main purpose of this feature is to allow the administrator to gain access to system so that he can perform administrative tasks.

3.2.2.1.2. Stimulus/Request sequence

An administrator enters username and password.

3.2.2.1.3. Associated functional requirements

Only authenticated administrator should be given access into the system.

3.2.2.2. Feature 2 (Logout)

3.2.2.2.1. Introduction/Purpose of Feature

The main purpose of this feature is to allow the administrator logout of the system at any point in time.

3.2.2.2.2. Stimulus/Request sequence

An administrator clicks on the logout link to logout of the system.

3.2.2.2.3. Associated functional requirements

The administrator should be able to end his session when he has finished his tasks.

3.2.2.3. Feature 3 (User administration)

3.2.2.3.1. Introduction/Purpose of Feature

This feature controls the users of the system. The main focus is on the creation of new users, deletion of existing users, modification of user status and passwords.

3.2.2.3.2. Stimulus/Request sequence

An administrator chooses a user management option from the various options.

3.2.2.3.3. Associated functional requirements

It should be possible to have someone responsible for managing the people who use the network or system. In so doing the security of the system is maintained.

3.2.2.4. Feature 4 (View user activity records)

3.2.2.4.1. Introduction/Purpose of Feature

This feature allows the administrator to view and manage user activity records such as usage statics.

3.2.2.4.2. Stimulus/Request sequence

An administrator chooses a particular user and sees that user's usage history.

3.2.2.4.3. Associated functional requirements

It should be possible for the administrator to view how a particular user is using resources.

3.2.2.5. Feature 5 (Administer prepaid cards)

3.2.2.5.1. Introduction/Purpose of Feature

This feature allows the administrator creates, view and modify prepaid cards.

3.2.2.5.2. Stimulus/Request sequence

The administrator clicks on prepaid vouchers to create, view, edit and delete prepaid cards.

3.2.2.5.3. Associated functional requirements

It should be possible for the administrator to administer prepaid cards. This includes activation and deactivation.

3.2.2.6. Feature 6 (Configure settings)

3.2.2.6.1. Introduction/Purpose of Feature

This feature allows the administrator to configure and manage system wide settings such as paths, servers and other miscellaneous details.

3.2.2.6.2. Stimulus/Request sequence

An administrator chooses the setting menu from which he will choose various configuration menus.

3.2.2.6.3. Associated functional requirements

It should be possible for the administrator to configure and manage system wide settings.

3.2.2.7. Feature 7 (Creating Billing Plans)

3.2.2.7.1. Introduction/Purpose of Feature

This feature allows the administrator to create and administer billing plans.

3.2.2.7.2. Stimulus/Request sequence

The administrator clicks on billing plans and view options to view, edit, add, modify and delete.

3.2.2.7.3. Associated functional requirements

It should be possible for the administrator to create and manage billing plans.

3.2.2.8. Feature 8 (Administer Invoices)

3.2.2.8.1. Introduction/Purpose of Feature

This feature allows the administrator to generate and administer invoices.

3.2.2.8.2. Stimulus/Request sequence

An administrator chooses the invoices menu from which he will choose various options.

3.2.2.8.3. Associated functional requirements

It should be possible for the administrator to generate, view and manage invoices.

3.2.2.9. Feature 9 (Administer Payments)

3.2.2.9.1. Introduction/Purpose of Feature

This feature allows the administrator to record, view and modify a payment made by a user.

3.2.2.9.2. Stimulus/Request sequence

An administrator chooses the payments menu from which he will choose various options.

3.2.2.9.3. Associated functional requirements

It should be possible for the administrator to record, view and modify a payment made by a user.

3.3. Performance Requirements

No performance requirements have been identified.

3.4. Design constraints

The system uses Mysql database running on Linux servers.

Our preferred web server is Apache2.

Our preferred captive portal is Chillispot.

Our preferred radius server is FreeRADIUS.

3.5. Security Requirements

Since users will use the web to enter sensitive data like usernames and passwords. Such data streams must be encrypted.

3.6. Software System Attributes

The system will be developed using PHP, Perl and bash as languages of choice. To avoid reinventing the wheel, FOSS applications will be used where appropriate.

3.7. Other Requirements

No performance requirements have been identified

Appendix B

- Pricing of the services- how can we price the network services in a way that is fair, affordable and profitable or (break even)?

- How are the operating costs going to be shared amongst the different communities?

- How are the collected funds going to be managed?

- How are the operating expenses going to be paid?

Pricing of services

One of the biggest challenges in charging for the use of ICT infrastructure in marginalised communities lies in how to price these services. The motive behind charging in the Dwesa project is to cover the costs of running the network, i.e. operational costs. However, it is not clear how to go about pricing the use of services to achieve this goal because there are several questions which need answers before we can come up with the prices. These questions include:

- **How much should we charge to break even?**

The answers to this question are many. Generally, it depends on how many people are using the network. If we know the number of people using the network, different amounts can be approximated for each user in order to achieve desired profit margins. Most of the operational costs are fixed. Costs such as the VSAT costs are fixed whilst electricity costs also do not change a lot. This implies that the marginal costs of additional consumption of resources are almost zero. Therefore, by performing break-even analysis and forecasting, it can be shown that the more the number of people using the network the better are the changes of generating more revenue. For example, if on average, person pays R20 per month for use of network resources, 5 people will pay R100, 10 will pay R200.

From another angle we can say, this also depends on the usage patterns of these users. If we are charging based on time, the more the time they spend on the network then the higher the revenue will be generated from them.

For the purposes of this demonstration, we assume the simple time based scenario where usage is charged per hour. Furthermore, we assume that the charge for this hour is similar to the charges of the South African Post Office's 'Public Internet Terminal', R5/per hour. Moreover, we should also approximate the total number of hours that are spent online on the network. This means the more the hours spent online the higher will be the revenue generated.

In addition to this, the fixed costs have to be calculated. Assuming that the average costs of electricity and other sundry costs are fixed at about R150, and also that the Telkom VSAT costs are fixed as shown in the Telkom VSAT contract. This gives us an estimated total of R2000 as fixed costs. The following diagram can be used to depict the relationship between the number of hours spent on the network and the revenue generated per month. The prices attached to services also affect the manner in which we will raise revenue.

The break-even analysis and forecasting of these costs per month can be summarised as shown below.

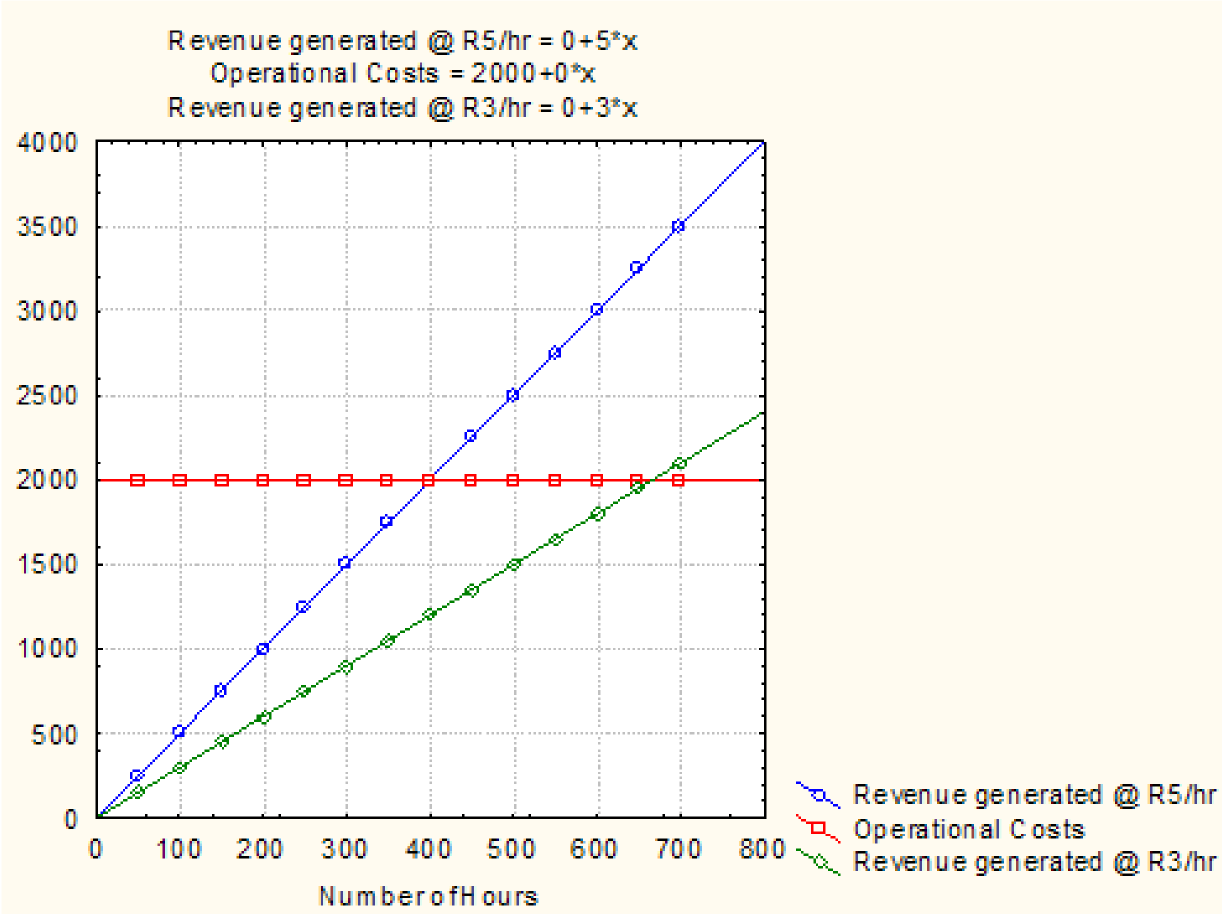


Figure B-0:1: Break-even analysis and forecasting

From the above diagram it can be seen that 400 hours per month at R5/hr and 650 hours per month at R3/hr can enable the network to break even. Hence, the higher the number of hours, the lower the prices can be.

Taking the same analogy, but now applying it to proceeds of sales made on the e-commerce portal. If we are charging based on sales, the more the items they sell online, the more the revenue the system will derive from them. This means that the system will get more revenue when more arts and craft items are sold.

- **How much should affordable prices be?**

Again, the answers to this question are similar to those provided for the above questions. For example, as can be seen from the break even analysis diagram, the higher the number of hours the lower the prices can be, hence the more affordable the prices will become. This also depends on the value that the users place on the utility of network resources as equated to the money they pay. This can also be a factor of comparing the charges in other parts of the country and the socio-economic status of the community under consideration. The underdeveloped rural hinterlands, particularly the former homeland areas of Transkei and Ciskei are poverty stricken [99]. According to the Global Insight, almost 64% of households in the Eastern Cape live below the poverty line [99]. The poverty line is a measure of income adequacy, expressed in monetary terms. It is composed of an aggregate cost of minimum goods basket[100]. If the majority of people in the Eastern Cape are living in poverty, this implies that their ability to pay is low, and so for the service to be affordable to them, the prizes have to be lower compared to other places. However , a complication arise because the cost of providing ICTs in these areas is very high, hence for the users to pay for the costs of running these networks, they have to endure the high prices, unless if they can use the resources in their numbers and share the costs as previously discussed. This takes us back to the dilemma of balancing the need to generate revenue and development goals discussed in chapter 2. The more an implementation is required to generate revenue in its initial stages, the less the success will be achieved on supporting developmental goals. The following section portrays the Internet usage patterns in Dwesa.

Cost sharing

The other questions that need to be answered relate to cost sharing.

- **How are the costs going to be shared?**

There are different ways in which costs can be shared in this network. First of all, we look at other community developments which show the ability and willingness of the community to sustain the infrastructure.

Community paying

When we installed the Mpume lab, we did not supply them with a printer. However, the teachers saw the urgent need of a printer and mobilised the community to contribute money to buy one. To our surprise it was a heavy duty printer (Konica Minolta Bizhub 162) which we had not expected that they would buy. Other schools as well managed to follow the suit in organising their own printers. This initiative shows that the communities are willing to give whatever little they have for the success of the project. So another source of revenue is through community contributions.

Schools paying

The schools themselves may also pay a certain amount of money because they are the ones who stand to benefit the most due to the fact that the computer labs are hosted there. They can use the computers to enhance their educational programmes. If schools have managed to sustain other costs, like electricity, they will do whatever they can to sustain the ICT infrastructure provided there is a definite benefit they can derive. Just to give a simple illustration, we assume that the operation should be split equally amongst the benefiting schools or organisations. The diagram below shows the costs each school will have to pay decreases as the number of schools involved increase.

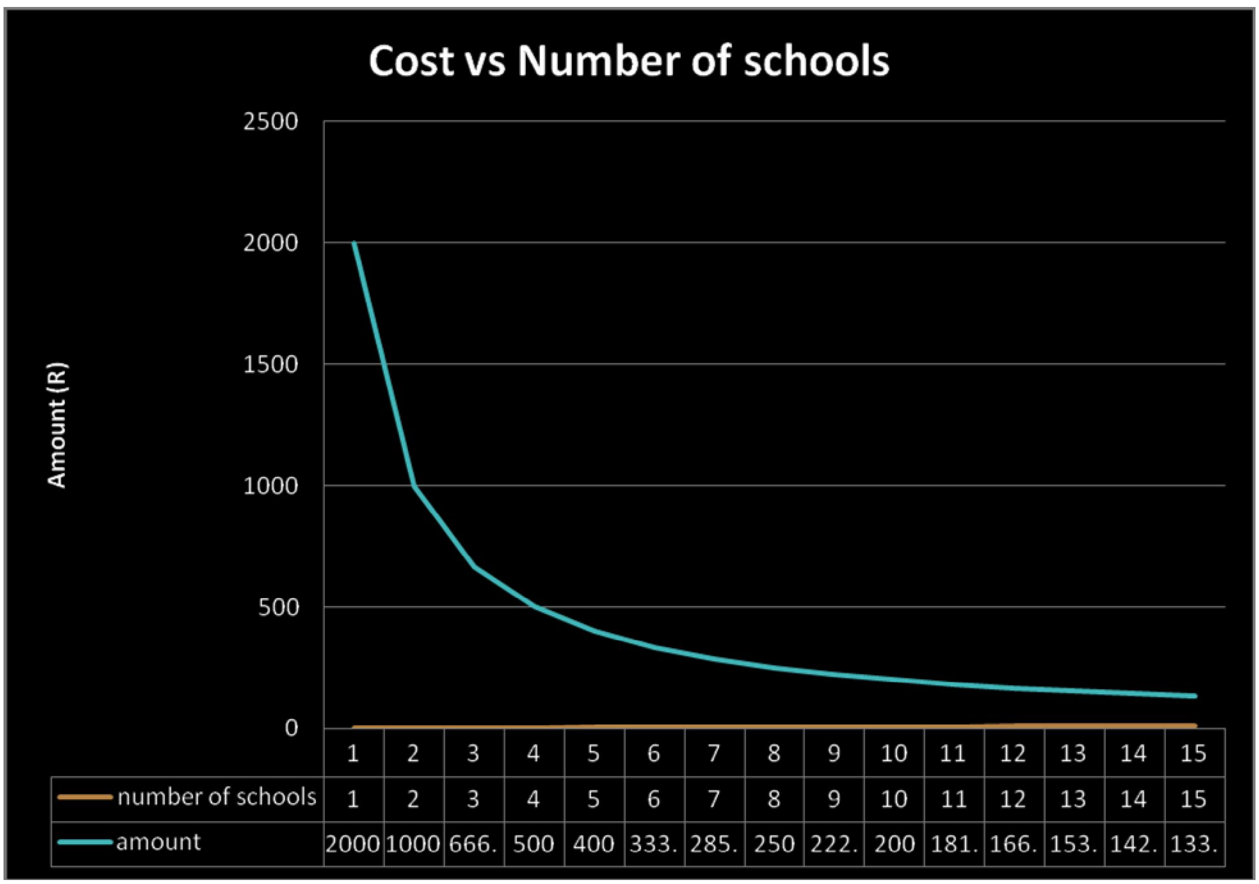


Figure B-0:2: Cost vs number of schools

Figure B-2 shows that as the number of schools increase, the amount each school will have to pay decreases significantly. Hence, if we connect more schools, the infrastructure will be cheaper for each school to manage. But, and important point note is the fact this amount decreases at a decreasing rate.

As far as the schools' budget for the ICT infrastructure is concerned, a possible solution would be to appropriate some of the travelling and communication funds into the ICT budget. There is always a need for schools to constantly communicate with other schools for organising sporting activities, school functions and meetings, and to notify each other of other important work related issues. With ICTs, these schools can communicate efficiently and save time, which is a good motivation of appropriating some of the travel and communication funds into the ICT budget.

Learners paying

Another option is for the learners to pay. If we can look in most of the school related developmental projects in developing countries, for example, the building of new classrooms, durawalls and other infrastructure, the students of those schools contribute towards the costs of the projects mostly in the form of levies. If it a durawall, it will be called a durawall levy. However, in our case, it is ICT, and so why not also introduces an ICT or computers levy.

There are many ways in which this levy might be introduced, for example, the learners could pay according to the usefulness of computers in their studies, and naturally, senior students should pay more than junior students. This means we would need to classify our students into different classes which will then pay different ICT levies. Again, in this scenario the more the students the lower will be charges.

Money from other sources of revenue

As highlighted in chapter 5, there are other sources of revenue. Once the network has matured, it is likely that these revenues will grow. This revenue can be used to subsidise costs. Besides these, there is also another innovative income generation method being carried out by all schools - cell phone charging. This method has already been highlighted in chapter 3. They use the proceeds from charging cell phones to buy electricity and other electrical equipment. This means that surplus can easy be used to cover other operational costs on the network.

Going back to the break even analysis (figure B-1), when the money from other sources is used to cushion expenses, the effect it has on the graph is an upward shift in the horizontal axis as shown in the diagram below.

The diagram depicts revenue from other sources as R1000. This has an effect of shifting the revenue generated upward. Consequently the break even points change to that 200 hours per month at R5/hr and 350 hours per month at R3/hr.

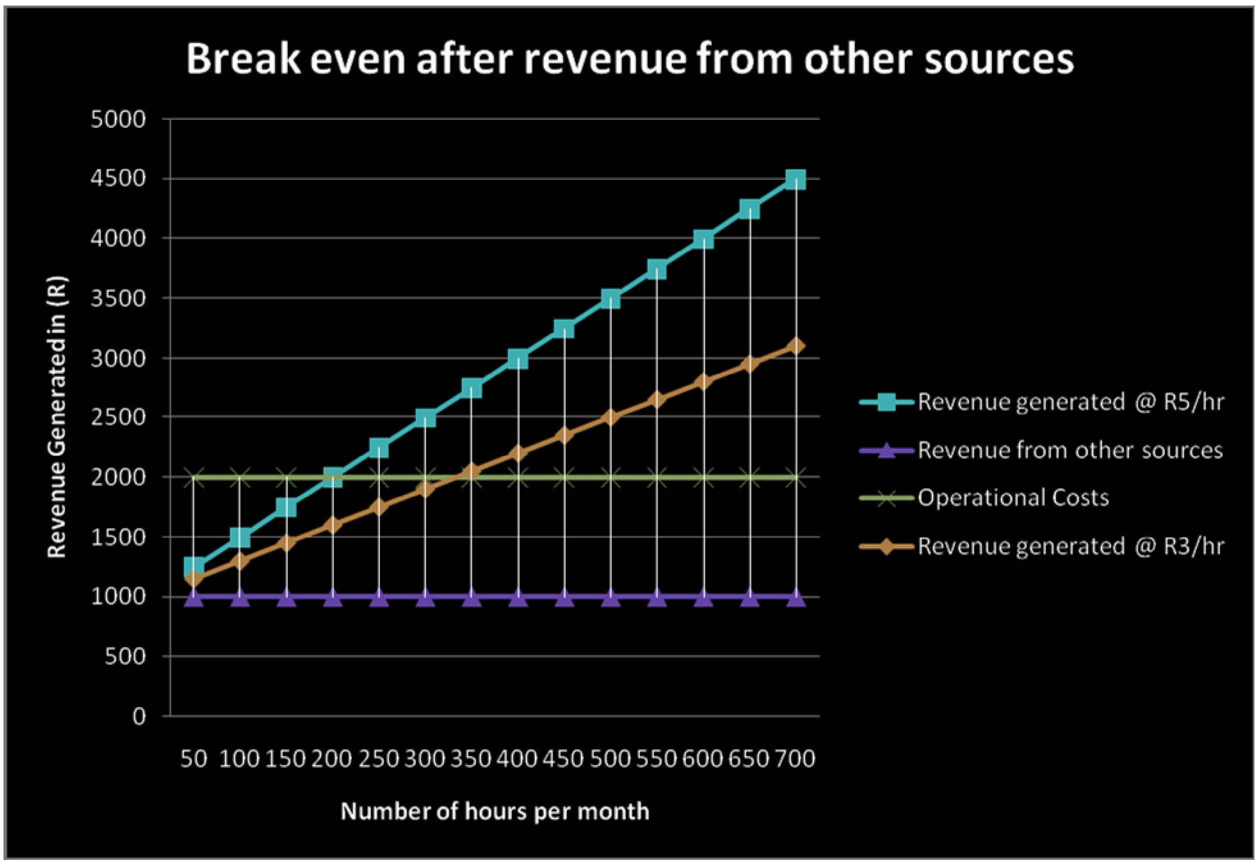


Figure B-0:3: The effect of revenue from other sources on break even

Fairness

However, given the nature of equipment installed, different schools have different responsibilities on the network. This is highlighted by the cases where other schools depended on other schools for services and connectivity. The schools which everyone depends on have the burden of hosting more powerful servers which consume more electricity. Therefore, as a way of introducing some sort of fairness in the cost sharing process, these schools should receive a certain amount of money from the schools that depend on them.

Managing the funds

There is no one clear way to manage the infrastructure financially. One way of doing it is to follow some kind of wholesaler-retailer model. Mpume which is the school connected to the VSAT acts a wholesaler who sells services to other schools.

Another way is to say, at the beginning of each month, each school pays all expenses with the hope of recovering them from its supporting community. That is as the community users come to use the Internet we charge them and raise some revenue for our school. This might actually have the effect of motivating the schools to encourage the communities in their vicinity to come and use the infrastructure because, without this they have to allocated the money from the school's own funds.

However, there are situation where some communities are smaller than others, and expecting them to contribute in same manner as the bigger communities might not be fair.

The schools might accumulate and keep these funds until they are needed to make payments for the operational costs.

Problems faced in the Dwesa project

The project was under constant threat from a lot of problems ranging from political to technical. Some of these problems include: poor connectivity and computer failures. Furthermore, power failures and interruptions were also experienced due to bad weather conditions. There were also power problems due to the fact that some of the principals felt that computers were consuming lots of electricity. In some schools where principals were groaning about electricity consumption, the teachers felt intimidated to use the computers.

Conclusion

In this section we have explored questions relating to pricing of services, cost sharing and management of funds. In this discussion we highlighted the problems and their tentative solutions. It is clear from this discussion that there are no clear cut ways of answering these questions. The whole cost sharing process can be very complicated to manage, therefore a more constructive approach is needed to device the best possible methods of achieving this.

This is a problem which is open to any interpretation, therefore can lead to very innovative and intuitive strategies if the community is engaged in the process of determining the best methods. There are many ways in which the funds may be managed, but likewise, the community also needs to be engaged to come up with tentative solutions.

One thing which is clear in this discussion is that the community has to see the value of ICTs and be willing to utilise the resources and contribute towards the sustainability of the project. The more people use the infrastructure, the higher will be the chances of generating more revenue.