UNIVERISTY OF ZAMBIA
DEPARTMENT OF COMPUTER STUDIES

UNIVERSITY OF ZAMBIA CLINIC RECORDS MANAGEMENT SYSTEM.

BY

Malama Kasanda Jr.

ABSTRACT

The purpose and essence of any Records Management system is the right information in the right place in the right order, at the right time for the right person at the lowest cost [1]. For this feat to be achieved, an integrated, highly efficient and effective records management system is needed.

The Major objective of the project was to design and develop a RMS that would automate patient records Management and give direct benefit for the University Clinic section in terms of fast information retrieval, enhanced decision-making (patient diagnosis) whilst avoiding any confusion that would jeopardize the quality of patient care. The RMS was designed as a client/server and web-based system and implemented using open source solutions that include MySQL as the database and back-end storage engine, and PHP, HTML and JavaScript as the programming languages.

The system was developed using Extreme Programming methodology. An extensive evaluation of the project determined that the project achieved many of its predefined objectives however, the major limitation of the project was the scope covered. From a proper analysis and assessment of the designed system, it can be concluded that the system developed is an efficient, usable and reliable records management system.

CHAPTER ONE: INTRODUCTION TO THE RESEARCH

1.1 Introduction

Hospitals and Clinics deal with the life and health of their patients. Good medical care relies on well-trained doctors and nurses and on high quality facilities and equipment. Good medical care also relies on good record keeping. Without accurate, comprehensive and up to date and accessible patient notes, medical personnel may not offer the best treatment or may in fact misdiagnose the condition, which can have serious consequences. Associated records, such as x-rays, specimens, drug records and patient registers, must also be well cared for if the patient is to be protected. Good records care also ensures the hospitals administration runs smoothly; unneeded records are transferred or destroyed regularly, keeping storage areas clear and accessible; and key records can be found quickly, saving time and resources. Records also provide evidence of the hospital's accountability for its actions and they form a key source of data for medical research, statistical reports and health information systems.

Managing Clinic Records addresses the specific issues involved in managing clinical and non-clinical hospital records. A comprehensive records management system in a hospital helps to ensure that staff have access both to clinical information and to administrative records on a wide range of issues, including policy, precedents, legal rights and obligations, personnel, finance, buildings, equipment and resources.

Good record-keeping is an integral part of governance, professional practice and the mark of the skilled and safe practitioner. The University of Zambia Clinic Records Management System (UNZACRMS) is a platform and utility for efficient record-keeping as well as offering a comprehensive medical dispensary tracking system.

As well as underpinning good patient care, complete, accurate and timely records allow a clear picture of events to be obtained, which is imperative for managing claims and complaints, and for auditing practice and remaining proactive in improving records management practices.

A management information system (MIS) is a system or process that provides the information necessary to manage an organization effectively. An MIS should be able to influence decision making. A records management system while incorporating aspects of a MIS should be able to influence decision making in an institution/ organization.

An information system (IS) is any combination of information technology and people's activities using that technology to support operations, management, and decision-making. In a very broad sense, the term information system is frequently used to refer to the interaction between people, algorithmic processes, data and technology. In this sense, the term is used to refer not only to the information and communication technology (ICT) an organization uses, but also to the way in which people interact with this technology in support of business processes and is therefore relevant to the development of a records management system.

A management system is a proven framework for managing and continually improving an organization's policies, procedures and processes.

The ultimate goal of UNZACRMS is to enable the clinic provide efficient record-keeping technique's to better the services that it provides.

Therefore a good and efficient records management system should be able to incorporate specific aspects of the systems mentioned above in order to provide and efficient means of records storage and management.

1.2 Motivation and Importance of Thesis

The author came up with the UNZACRMS with the view of creating a fast and efficient records management system for the university clinic, by allowing members of staff go about with their daily operations but only this time assisted by the system to make information and get information as and when it is available in a fast manner. The main motivation stemmed from the need to implement particular view points on issues concerning the management of records at the clinic; furthermore, it was imperative to manage other attributes of the system that encompassed the clinic such as payment methods. The inspirations came from the concepts being implemented at National level through the Smart-Care project to which most of the modules implemented on the system where gotten. Thus, this project is an investigation into the possibility of such an implementation in our local setting here at the University of Zambia.

1.3 Scope

The scope provides for the boundary of the research in terms of depth of investigation, content, and methodology, geographical and theoretical coverage.

The system was exclusively designed and developed for the UNZA Clinic records Management Department in general section in particular. The UNZA Clinic records section is solely responsible for keeping medical and related records for patients and keeping track of this information.

The records management system was designed in such a way that makes it possible to access it through any web browser programme. This serves as the user interface. The web browser supported interface created is dynamic and as a result backed by a database system that enables users to have the ability to input, access, manipulate and delete data from the database.

HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets) were used as the languages of preference for the design of user interfaces. In the interfaces, Java script was used as the client side validation tool.

PHP was used as a scripting language for linking the interfaces to the SQL database(s). PHP is a server-side scripting language that enables one the ability to insert into a web interface instructions that web server software would execute before sending a response to the web browser [2].

SQL was used as the programming language for developing the database. SQL is the de facto standard language used to manipulate and retrieve data from these relational databases.

The proposed system is a simple web platform for the University of Zambia Clinic to enable clinic staff keep track of their patient's records, raise dispensary requisitions, keep track of medical charts of the patients. Thus, the system will be Web-based.

1.4 Problem Statement

The system design and development was undertaken in order to eliminate the problem of redundant, erroneous and incomplete data that was escalating the inefficiencies in data retrieval. These limitations were mainly caused by the fact that data, under the previous manual recording system was entered into books and paper files and was later stored in overcrowded storage rooms that made retrieval of archival records close to impossible.

1.5 Aims and Objectives

The main aim of the project was to build an electronic patients medical record information system that provides easy and instant access to patient information. In addition, the new system will obtain a more efficient support for clinical work, better possibilities for integration with future systems, and lower cost of operation.

Objectives

1.5.1 General Objective

To design and develop a records management system for the University of Zambia Clinic that would enable faster and more efficient storage, retrieval and updating of clinical records.

1.5.2 Specific Objectives

The projects specific objectives where:

- To carry out a feasibility study for the possibility of developing a records management for the University of Zambia Clinic.
- To design and develop a records management system for the University of Zambia Clinic
- To test and validate the records management system for the the University of Zambia Clinic
- To implement the records management system for the University of Zambia Clinic

1.6 Organization of the Thesis

The thesis is organized in a way that's easy to manuover and read through: Chapter One clearly introduces the concept under research, stepping through Chapter Two where a review of literature has been tackled with particular interest in Software systems that share certain implementable similarities to the UNZACRMS. Chapter Three on the other hand gives an insight of Methodologies used to implement the UNZACRMS. Furthermore Chapter Four and Five gives insight analysis of the findings of the previous chapters and concludes the discussion.

1.7 Summary

To crown it all, this chapter was an introductory note to the design and implemenatation of the UNZACRMS. It highlighted the major Aim and Objectives of this project, shed more light on the scope and problem statement under discussion.

CHAPTER TWO: LITERATURE REVIEW AND RELATED WORKS

2.1 Introduction

In order to understand the concepts associated with records management and or computer based records management systems, it is imperative to examine and analyze published material from experts regarding the field. The purpose of this review is to analyze and examine and obtain experience as regards the creation and archival processing of electronic records. The review is based on an exhaustive assessment of the literature on computerized electronic management and electronic records, and contains an overview of the main concepts associated with the creation of an electronic records management system from the perspective of published experts.

2.2 Literature Review

2.2.1 Background Theory

In this section of the research, location and analysis of the existing knowledge related to the subject of inquiry are explored and cited. It also sells at the relationship of the proposed research for purposes of good representation and critical review of the existing literature. Martin (1976) data within an organization is increasingly being regarded as a basic resource needed to run the organization. As with other basic resources, professional management and organization of data are needed. The importance of efficient use of data for planning, predicting and other functions will become so great in a computerized organization that it will have a major effect on growth and survival of co-operations. In relation to the above argument, the presence of an automated data management system at UNZA clinic's efficiency, timely decisions and responses will be achieved.

ERs are part of an organisation’s memory that is invaluable to the current and future functioning of an organisation or a corporate body. To consider ERs as formal records with intrinsic value, these must go through stringent information management policies. For the sake of reliability and authenticity, Ers must adequately capture and describe the actions these represent. The record should not only preserve the ‘content’ but also the ‘context’. Therefore, Stephens and Wallace [2] have indicated that now most of the archival repositories are undergoing transition from paper to electronic format. Archival institutions worldwide have plans to protect the integrity and ways to retrieve archival value of the Ers [4].

E-Health is described as a revolutionary new paradigm for health care that has evolved as a result of advances in information, telecommunication, network technologies and information management. These technologies have transformed the way that health care is delivered [4]. Today's technology has the capability to support people in managing almost all aspects of their health care, from seeking general health information to clinical consults without ever having to leave their homes; yet even the most basic personal health information, like specific results of tests, contained in medical charts is not currently readily available through existing technologies (like the Internet) to most consumers of health care [4]. This inaccessibility makes it difficult for consumers of health care to be active participants in their own health and wellness. There is consensus that in order for patients to be true partners in the health care encounter, they must have access to their own personal clinical health information [4]. that is commonly stored in institutional electronic health records (EHRs). An Electronic Health Record (EHR) is a computerized version of an individual's health record that may contain a person's full health and medical record or can be used for certain records, such as lab results, in conjunction with a more traditional paper-based patient chart.

The EHR may be accessible online from many separate, interoperable automated systems within an electronic network and it can facilitate the electronic integration of health care providers by enabling the retrieval of information about patients when and where it is most needed [5]. The ability to provide patients with access to their personal health information can be facilitated through the use of emerging technologies, most commonly through the Internet [6]. This type of access to one's own health information can help prepare individuals to better manage and cope with their health status. In turn, this type of access may have a positive impact on the health care system that can be recognized through more efficient use of resources resulting in health care savings [4]. Access to personal health information is a fundamental right supported by the law [5] and the emergence of new technologies alters how that right can be fulfilled. It has been demonstrated that access to one's health information using these technologies is desired by many health care consumers [4,5].

Although technology exists to support patient access to their medical record, and the law supports individuals' rights to access their health record the health care community has been slow to adopt the use of patient accessible electronic health records. Although several reasons could be cited for this slow uptake process, the absence of a key technology, the EHR, is a significant obstacle to making progress in providing patient accessible electronic records. A 2003 white paper from the American Association of Medical Informatics identifies "the lack of ubiquitous EHR usage" as the main environmental barrier to patient accessible health records [15].

There has recently been a noticeable increase in institutional interest in and adoption of EHRs and Canada Health Infoway has set the target of 50% of Canadians having their electronic health records available to their healthcare providers by 2010 [15]. The adoption of systems that provide patient access to these EHRs, such as patient portals or personal health records (PHRs) has been slower to follow. Systems such as these capture either elements of data or all the data stored in the EHR [5] and can be easily provided to the consumer. A PHR system can also incorporate data entered by patients themselves. Slower growth in this area may result from physicians' reluctance to embrace the use of information and communication technology (ICT) solutions [11].

Review of Important Concepts

Model View Controller

The Model-View-Controller (MVC) pattern separates the modelling of the domain, the presentation, and the actions based on user input into three separate classes [5]:

- **Model:** The model manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).

- **View:** The view manages the display of information.

- **Controller:** The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate.

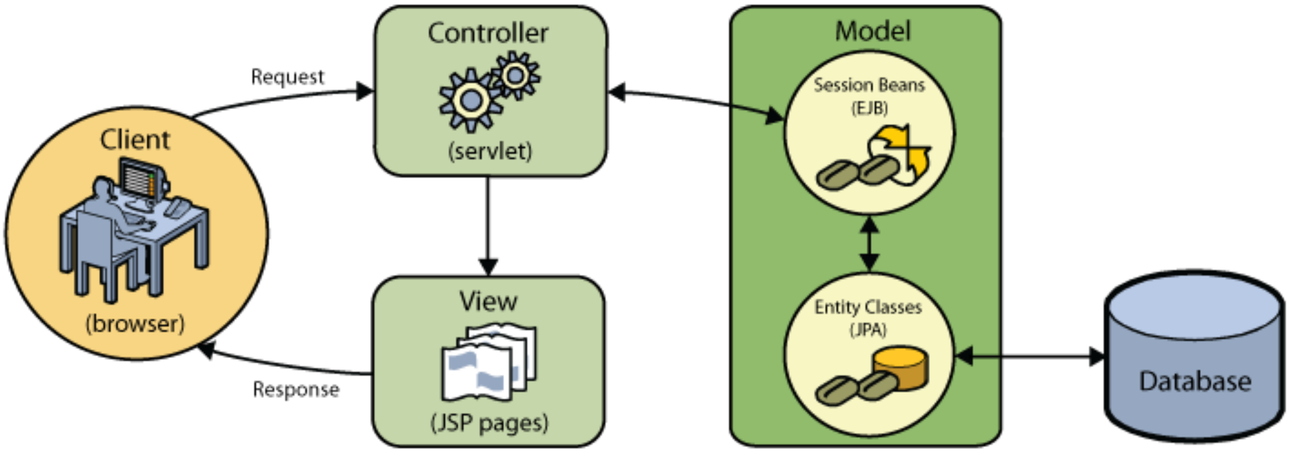


Fig: 2.1: MVC Class Structure

It is important to note that both the view and the controller depend on the model. However, the model depends on neither the view nor the controller. This is one the key benefits of the separation. This separation allows the model to be built and tested independent of the visual presentation. The separation between view and controller is secondary in many rich-client applications, and, in fact, many user interface frameworks implement the roles as one object. In Web applications, on the other hand, the separation between view (the browser) and controller (the server-side components handling the HTTP request) is very well defined. Model-View-Controller is a fundamental design pattern for the separation of user interface logic from business logic. Unfortunately, the popularity of the pattern has resulted in a number of faulty descriptions. In particular, the term "controller" has been used to mean different things in different contexts. Fortunately, the advent of Web applications has helped resolve some of the ambiguity because the separation between the view and the controller is so apparent. In a nutshell, separating functionality like this entails that medical records can be stored in a Relational database (Model) and separate the business logic of the Clinic using the controller, this being generic allows for the evolutionary of the system [5].

Web-based Communication and Information Exchange

The web is a system of interlinked hypertext documents accessed via the internet. It was created by British engineer and computer scientist, Sir Tim Berners-Lee sing concepts from earlier hypertext systems.

The terms Internet and World Wide Web are often used in daily speech without much distinction. However, the Internet and the World Wide Web are different. The Internet is a global system of interconnected computer networks while the Web is one of the services that run on the internet. The web is a collection of interconnected documents and other resources, linked by hyperlinks and URLs (Universal Resource Locators). In short, the web is an application running on the internet. Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser, or by following a hyperlink that page or resource. The web browser then initiates a series of communication messages, behind the scenes, in order to fetch and display it.

First, the server-name portion of the URL is resolved into an IP address using the global, distributed Internet database known as the Domain Name System (DNS). This IP address is necessary to contact the web server. The browser then requests the resource by sending an HTTP (Hypertext Transfer Protocol) request to the Web server at that particular address, the diagram below illlustrates this:

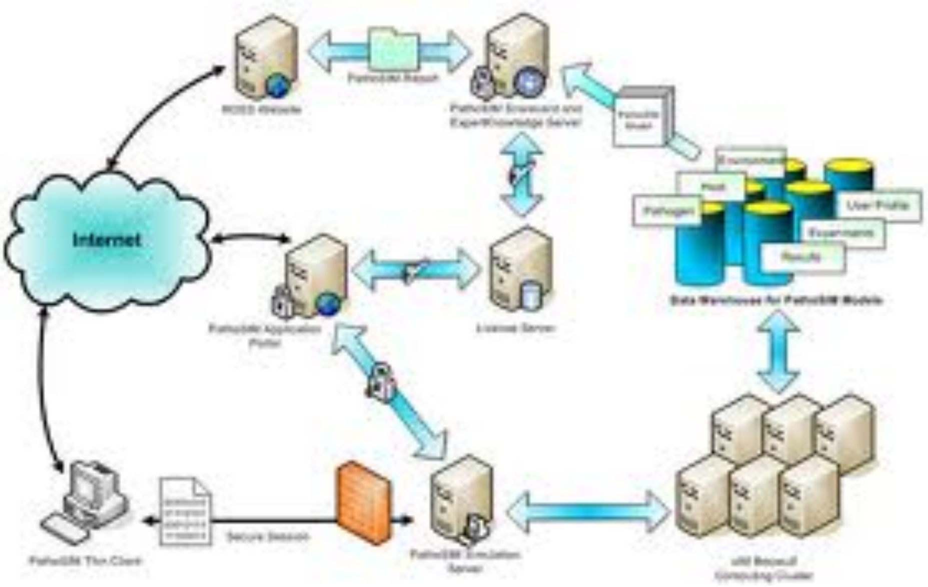


Figure 2.1: The role of a Name Server in Name-to-IP resolution.

In the case of a typical web page, HTML text of page is requested first and parsed immediately by the web browser, which then makes additional requests for images and any other type of files that make part of the page. Statistics measuring a websites popularity is usually based either on the number of page views or associated server hits (file requests) that take place.

The UNZACRMS uses the same technology. While receiving these files from web server, browsers may progressively render the page onto the screen as specified by the HTML, CSS, and JavaScript. Any images and other resources are incorporated to produce on-screen web page that the user sees. Most web pages will themselves contain hyperlinks to other related pages and perhaps to downloads, source documents, definitions and other resources. Such a collection of useful, related resources, interconnected via hypertext links, is what was dubbed a “Web” of information.

Networks

The UNZACRMS will run on a network. A Network is a system of interconnected autonomous machines [7]. A computer network can allow sharing of resources and information among interconnected devices. A computer network, often simply referred to as a network, is a collection of computers and devices among users and allows users to share resources. Networks maybe classified according to a wide variety of characteristics. Web based information systems can be implemented on many types of network's. The networks range from LAN, VPNs, MANs and WANs.

Local area networks, generally called LANs, are privately owned networks within a single building or campus of up to a few kilometres in size. They are widely used to connect personal computers and workstations in a company office and factories to share resources (e.g printers) and exchange information.

A metropolitan area network. or simply a MAN covers a city. The best known example of a MAN is the cable television network available in many cities in developed countries.

A wide area network, or simply just WAN spans a large geographical area, often a country or continent. It contains a collection of machines intended for running user or application programs. The host are connected by a communication subnet, or just a subnet for short. The hosts are owned by the customers, whereas the communication subnet is typically owned and operated by a telephone company or internet service provider. The job of the subnet is to carry messages from host to host, just as the telephone system carries word from speaker to listener. Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts), greatly simplifies the complete network design. Devices employed in WANs include routers that transmit information between networks.

The network of networks forms the internet.

Computer networks can be used for several purposes:

Facilitating Communications. Using a network, people can communicate efficiently and easly via mail, instant messaging, chat roos, telephone and video conferencing.

Sharing Hardware. In a networked environment, each computer on a network may access and use hardware resources on the network, such as printing a document on a shared printer or a server.

Sharing files, data, and infromation. In a networked environment authorized user may access data and information stored on other computers on the network. The capability of providing access to data and information on shared storage devices is an important feature of most networks.

Databases

A database is a collection of logically related data and a description of this data, designed to meet the information needs of an organisation. A databases consists of an organized collection of data for in or more uses, typically in digital form. One way of classifying databases involves the type of their contents, for example: bibliographic. Digital databases are managed using a Database Management System, which store database contents, allowing data creation and maintenance, and search and other access.

AJAX

AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS and Java Script.

Ajax uses XHTML for content and CSS for presentation, as well as the Document Object Model and JavaScript for dynamic content display. Conventional web application trasmit information to and from the sever using synchronous requests. This means you fill out a form, hit submit, and get directed to a new page with new information from the server.

With AJAX when submit is pressed, JavaScript will make a request to the server, interpret the results and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.XML is commonly used as the format for receiving server data, although any format, including plain text, can be used. AJAX is a web browser technology independent of web server software.

A user can continue to use the application while the client program requests information from the server in the background. Intuitive and natural user interaction. No clicking required only Mouse movement is a sufficient event trigger. Data-driven as opposed to page-driven.

The UNZACRMS makes use of AJAX for Rich Internet Application Technology, because it is the most viable RIA technology thus far [8]. Its getting tremendous industry momentum and several toolkit and frameworks are emerging. But same time JAX has browser incompatibility and it is supported by Java Script which is hard to maintain and debug [8].

Web 2.0

The term Web 2.0 is used in association with Web Applications, techiniques and technologies that facilitate interactive information sharing, interroperability, user-centric design, and collaboration on the World web. A Web 2.0 Website or application gives its users the free choice to interact or collaborate with each other in a social media dialogue as creators of user-generated content in a virtual county, in contrast to Websites where users are limited to the passive viewing of content that was created for them. Good examples of 2.0 include social-networking sites, blogs, wikis, video-sharing sites, hosted services and web applications.

Web 2.0 websites and applications allow users to do more than just retrieve information. Ny increasing what was already possible in the supposed Web 1.0, they provide the user with more interactive user-interface, software and storage facilities, all through their web browser. This has been called “Network as Platform” computing. Users can provide the data that is on Web 2.0 site and exercise some control over that data. These sites may have the architecture of participation and involvement by the users that encourages users to add value to the application as they use it.

The concept of Web-as-participation-platform captures many of these characteristics web 2.0 has been described by some as the “Participatory Web” and with this regard Web 1.0 following the Web-as-information-source paradigm.

Web 2.0 makes extensive use of AJAZ. All this relates to this project in that this paradigm allows web users to interact with their data by being able to add, view, edit and delete the content.

2.3 Related Works

2.3.1 Records Management System for Mbarara Hospital in Uganda

For the last few years the hospital employees have been able to collect data from agents by providing them with a piece of paper with required fields to fill. Its routine for every health worker to collect data, this should be processed and stored completely. They avail the right information and knowledge to the right person and institution in the form at the right time and place. The information ranges from individual patient reports to disease re-balance to mortality rate in the right persons and institutions which include the counties that use the health service, the service provider at local level, ministry of health and the donors. The company's employees and patients are straining to process lots of policy documents every day. Integrating and streamlining policy Application and document processes would ease administrative headaches for patients and greatly strengthen relationships with their customers Streveler (2004) grouped the component making HIS into 2 which are information involves processing data and collection, management. transmission, Information processing, processing analysis and presentation of information for use in patient care and health care management decisions. Health management system cannot exist alone but as functional unit aimed at improving the health of individuals.

The problems which Mbarara Hospital faced before the system was developed are similar to that of the ones being faced at the UNZA Clinic. As records accumulated, management became so difficult that some were even lost in the process. The system was developed using PHP (cakePHP as the framework using the MVC architecture and MySQL as the back-end database.

2.3.2 St Jansdal Hospital in the Nertherlands

The hospital wanted to reduce time spent on desk work, such as re-entering patient data into the hospital's systems, so that nurses could focus more on bedside care.

St Jansdal deployed CSC's Lorenzo solution, an electronic record management system that nurses now use to capture data electronically at the bedside. Lorenzo automatically triggers actions based on patient data and generates reports for hospital management. This system was built using C# and MS SQL as back-end database. Nurses spend more time with patients and less on deskwork and other healthcare staff can react more swiftly to patient needs which helps improve outcomes.

2.3.3 Munster Hospital in USA

The Health Information Technology for Economic and Clinical Health (HITECH) Act, part of the American Recovery and Reinvestment Act of 2009, designated funding to modernize the health care system by promoting and expanding the adoption of health information technology, according to the U.S. Department of Health and Human Services.

Aside from offering Medicare and Medicaid incentive payments to hospitals and doctors who use electronic health records, the act authorizes grant programs and contracts that support health information technology adoption.

2.3.4 SmartCare Project in Zambia

The SmartCare project in Zambia is a portable intergrated Electronic Health Record system currently being deployed in Zambia, Ethiopia, and South Africa.

SmartCare offers a touch screen Graphical User Interface (GUI) to minimize the learning curve and enable rapid data entry by clinicians and the Electronic Health Record is copied to Smart Cards for clients to own permanently addressing broadband costs and infrastructure failures.

The UNZACRMS can be thought of a small scale SmartCare project, implemening some of the modules that the larger scale project is offering.

2.4 Summary

The chapter gave an insight into other peoples view of the area of research, typically coin phrasing what methodology of use for this system is. The chapter illlustrated the importance of the the MVC architecture and the vital role it as to play in constructing this project.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

Methodology is a term used to describe a process, technique or manner in which an action is performed. Under the development a system, a methodology refers to the process that was taken to ensure that a system is effectively and efficiently developed.

The Software development used was a hybrid between incremental, as well as XP and evolutionay approaches at various stages of the development process. High priority requirements were tacled first followed by decreasingly lower priority requirements. The evolutionary approach involves adding requirements as they come up and the incremental approach involves developing the application as a series of increments. This development model was used because advancement in the project was easily achieved by production of milestones and deliverables at each stage of the process.

In designing the records management system for UNZA Clinic, the following system development methodology was used.

3.1.1 System Development Methodology

The systems development methodology is used to describe the process for building systems, intended to develop systems in a very deliberate, structured and methodical way.

Extreme programming was used as the methodology of choice in developing a records management system for the University of Zambia Clinic.

Extreme programming is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles. This is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. The main goal of XP is to lower the cost of change in software requirements.

Extreme programming is carried out in the following manner; the phases are carried out in extremely small steps. First, one writes automated tests, to provide concrete goals for development.

Next is coding (by a pair of programmers). Design and architecture emerge out of refactoring, and come after coding. Design is done by the same people who do the coding. The incomplete but functional system is deployed or demonstrated for the users. At this point, the practitioners start again on writing tests for the next most important part of the system.

XP Features.

Extreme programming has the following features/ core practices:

- Fine scale feedback which involves, Test driven development, Planning game, Whole team and Pair programming.
- Continuous process rather than batch. This also involves, Continuous Integration, Design Improvement ,and Small Releases.
- Shared understanding including Simple design, System metaphor, Collective code ownership and Coding standards or coding conventions.
- And Programmer welfare that involves Sustainable pace that is forty hour week.

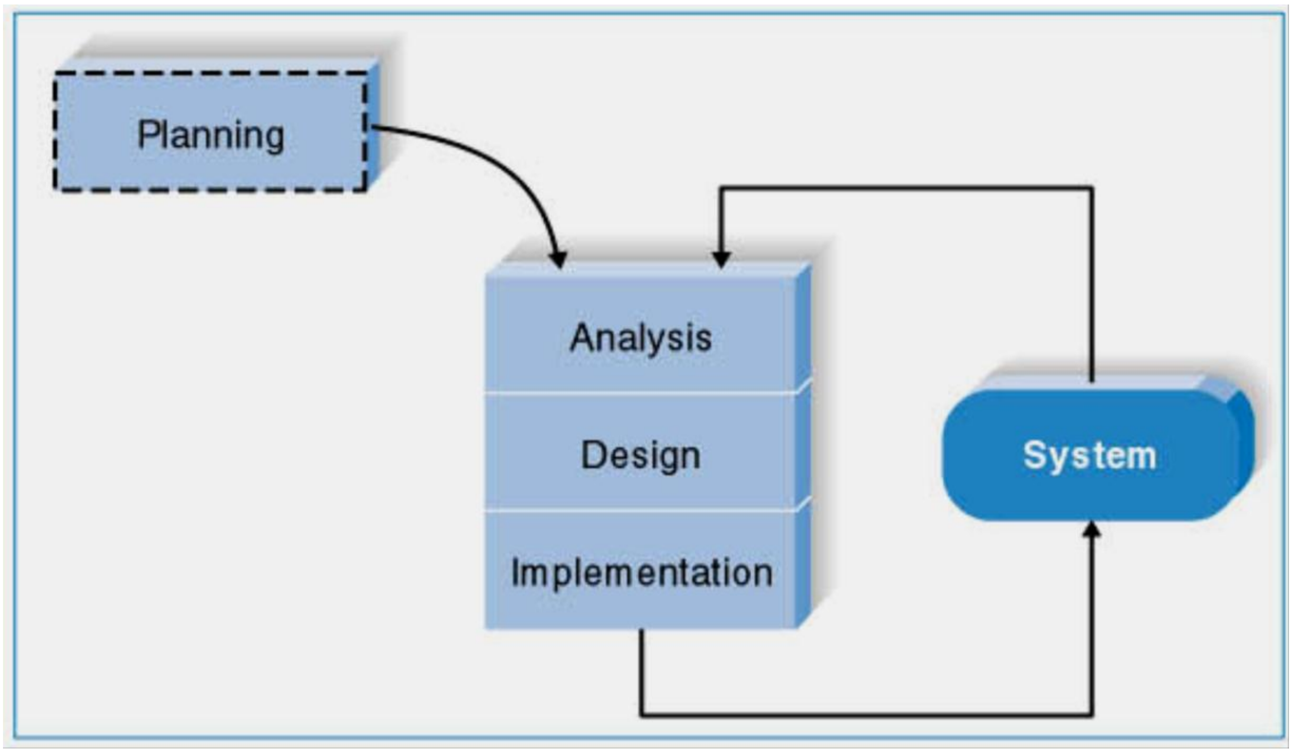


Figure 3.1.1 XP Life Cycle

3.1.2 System Development Lifecycle

In developing the UNZA Clinic records Management System, the following steps were taken;

1. **Planning**

A project plan was developed as well as other planning documents. It provided the basis for acquiring the resources needed to achieve a solution. This phase ensured that the problem solved was the one that needed to be solved and that the initial description was complete and consistent.

Under the planning phase of the project, a project timeline and work plan was developed. (Please refer to appendices). Under this phase;

- The project team was formed which consisted of two people
- The system flowcharts were prepared
- The characteristics of the proposed system were defined and identified

2. **Analysis**

At this point, the system in place was analysed to determine where the problem was in an attempt to fix the system. This step involved breaking down the system in different pieces to analyse the situation, analysing project goals, breaking down what needed to be created and attempting to engage users so that definite requirements could be defined.

Under analysis, Requirement gathering is the most crucial aspect as many communication gaps arise in this phase and this leads to validation errors and bugs in the software program.

Therefore, the following techniques were used to gather information

- **Semi-structured interviews:** Semi-structured interviews are conducted with a fairly open framework which allow for focused, conversational, two-way communication. They can be used both to give and receive information. In the process of developing the system, the development team interviewed the data entrants at the UNZA Clinic in order to identify the processes, obtain specific quantitative and qualitative information from the interviewees, obtain general information relevant to data entry , and to gain a range of insights on the process of records management.

This tool was used as a data collection methodology of choice because it is; less intrusive to those being interviewed as the semi-structured interview encourages two-way communication.

- **Direct (Reactive) Observation:** Direct Observation is a method in which a researcher observes and records behaviour / events / activities / tasks / duties while something is happening. This was used in correspondence to interviewing in order to gain a more holistic view of the Clinic's current records management system.

Direct observation was used as a research methodology of choice in designing the records management system for the UNZA Clinic because; Observations give additional, more accurate information on behaviour of people than interviews or questionnaires. They can also check on the information collected through interviews especially on sensitive topics.

- **Using available information:** This is a data collection method that involves the process of examining and evaluating already existent literature material to obtain facts and data regarding a specific subject. Locating these sources and retrieving the information can help in data collection.

In the development of the records management system, this research methodology was mainly used in the analysis and design phases of the system development process. This is because it permitted the researcher(s) to analyse changes in trends.

3. **Design**

In systems design the design functions and operations was described in detail, including screen layouts, business rules, process diagrams and other documentation. The output of this stage described the new system as a collection of modules or subsystems. The design stage took as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements was produced as a result of interviews, workshops, and/or prototype efforts.

Design elements described the desired system features in detail, and generally included functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary.

4. **Implementation Phase**

Here all the iterations were brought together and integrated to make one working system. Modular and subsystem programming code was accomplished during this stage. Unit testing and module testing was done in this stage.

3.2 System requirements

The system requires a client-server architecture where a server is necessary to host the application and the database .The users will access the server to retrieve information from their desktops through their web-based interfaces. For this to work, the following will be required;

3.2.1 Non-functional Requirements

Non-functional requirements are described as the constraints on the services the system provides and they include;

- Users must login in order to access the system resources.
- All staff who intends to use the system should undergo training.

3.2.2 Functional Requirements

The Functional Requirements of a system defines the capabilities and functions that a System must be able to perform successfully. The functionalities of the system include:

- Keep and track patients medical records
- Add and delete system users
- Make payments

- Generate and View reports
- Search and retrieve patient records

3.2.3 Hardware Specifications

- Processor Pentium II, Pentium III, Pentium IV or higher
- RAM 64 Mb or Higher
- Disk Space130 Mb or higher
- LAN Ethernet 10/100Mbps card/bus.

3.2.3 Software Specifications

- **Operating System**: Win-XP, Windows Vista, Windows 7 or Higher
- **Web Browser**: Any, but preferably Mozila Firefox
- **Database:** MySQL version 5.0.1 or higher
- **Webserver:** Apache 2.0 as web server

3.3 System Design

The system is designed in the following manner. The Records Management system has a backend engine that consists of a MYSQL database, PHP as the programming language and Apache as the webserver and the user interface modules. The system architecture is illustrated in Figure 3.0 below.

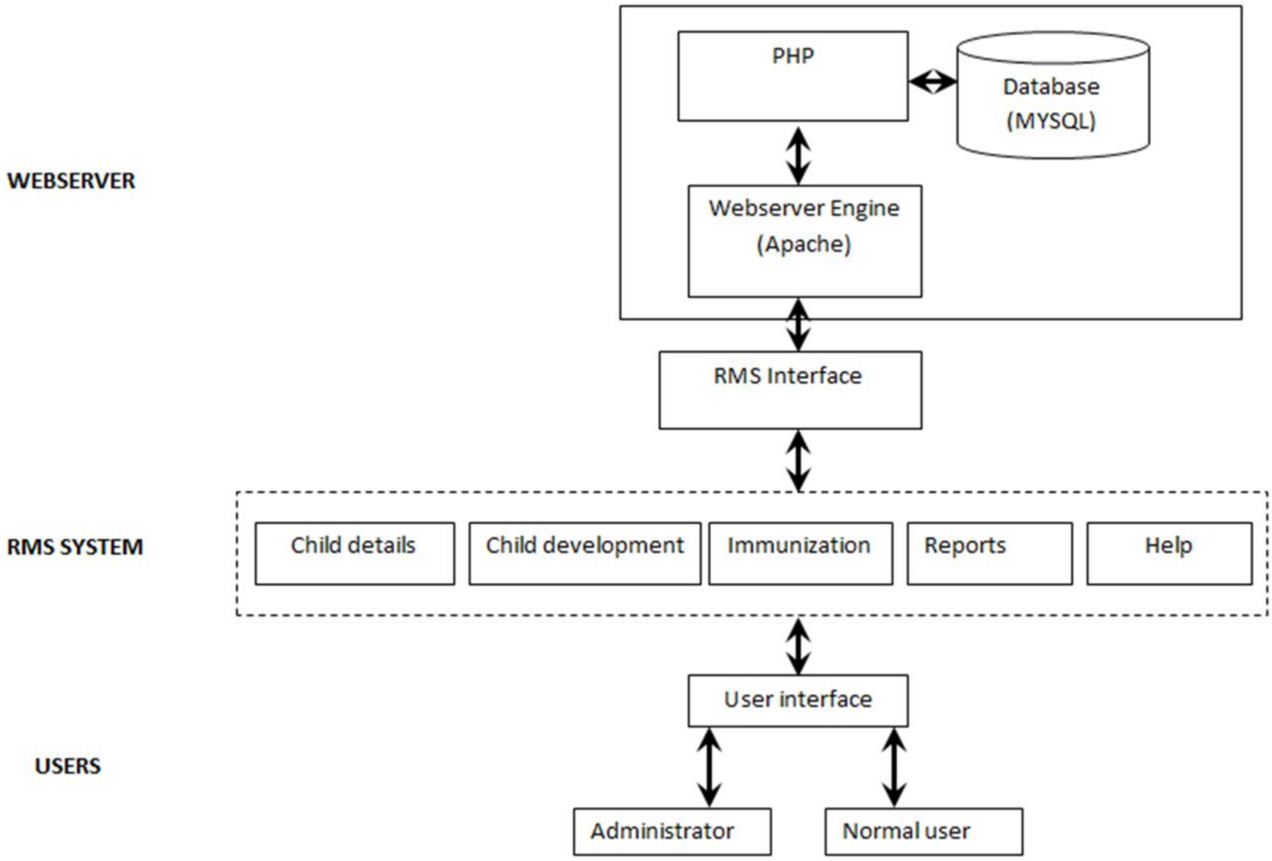


Figure 3.0 System Architecture for the UNZACRMS

3.3.1 System Analysis and Design

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. Unified Modeling Language (UML) was used to model this system in the analysis phase.

- **Use Case Diagram for the System**

Figure 3.1 shows the Use Case diagram for UNZACRMS, evidently the actors at play in this system are the Clerk, Doctor and a Specialized Clinical officer.

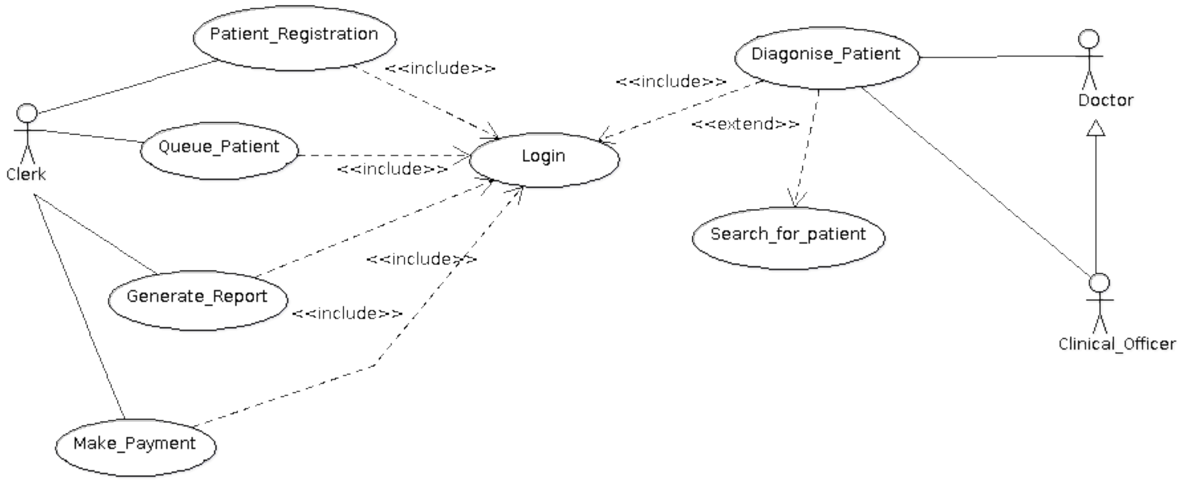


Figure 3.1 Use Case Diagram

- **Class Diagram**

Figure 3.2 shows the systems class diagram;

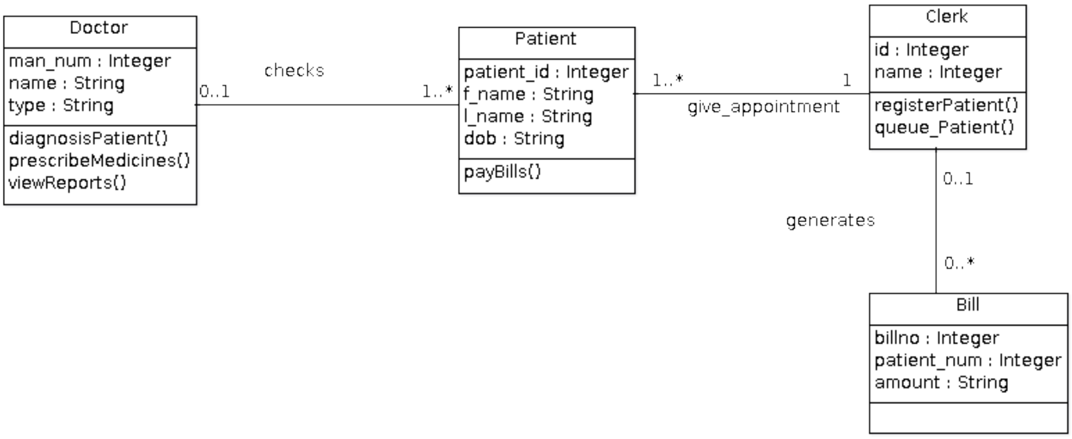


Figure 3.2: Class Diagram

- **Sequence Diagram for Patient Registration**

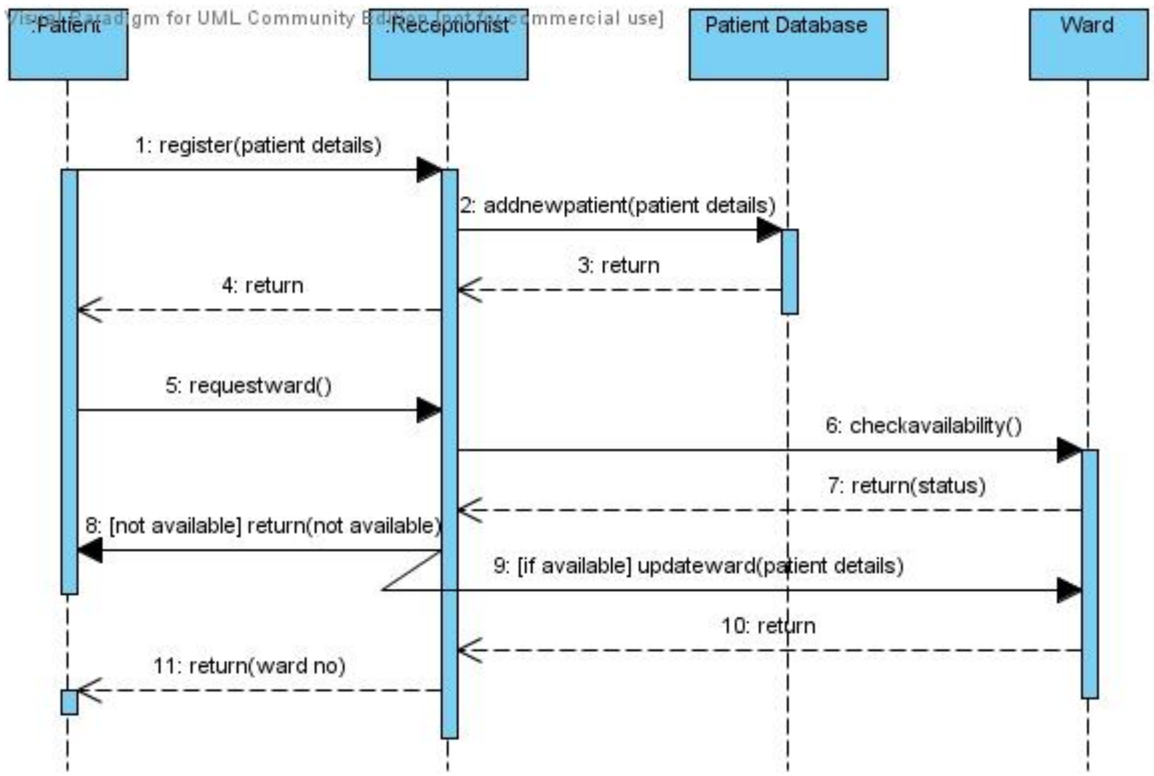

Visual Paradigm for UML Community Edition [not for commercial use]
:Patient
:Receptionist
Patient Database
Ward
1: register(patient details)
2: addnewpatient(patient details)
3: return
4: return
5: requestward()
6: checkavailability()
7: return(status)
8: [not available] return(not available)
9: [if available] updateward(patient details)
10: return
11: return(ward no)

Figure 3.3 Sequence Diagram for Patient Registration

- **State Chart for a Patient**

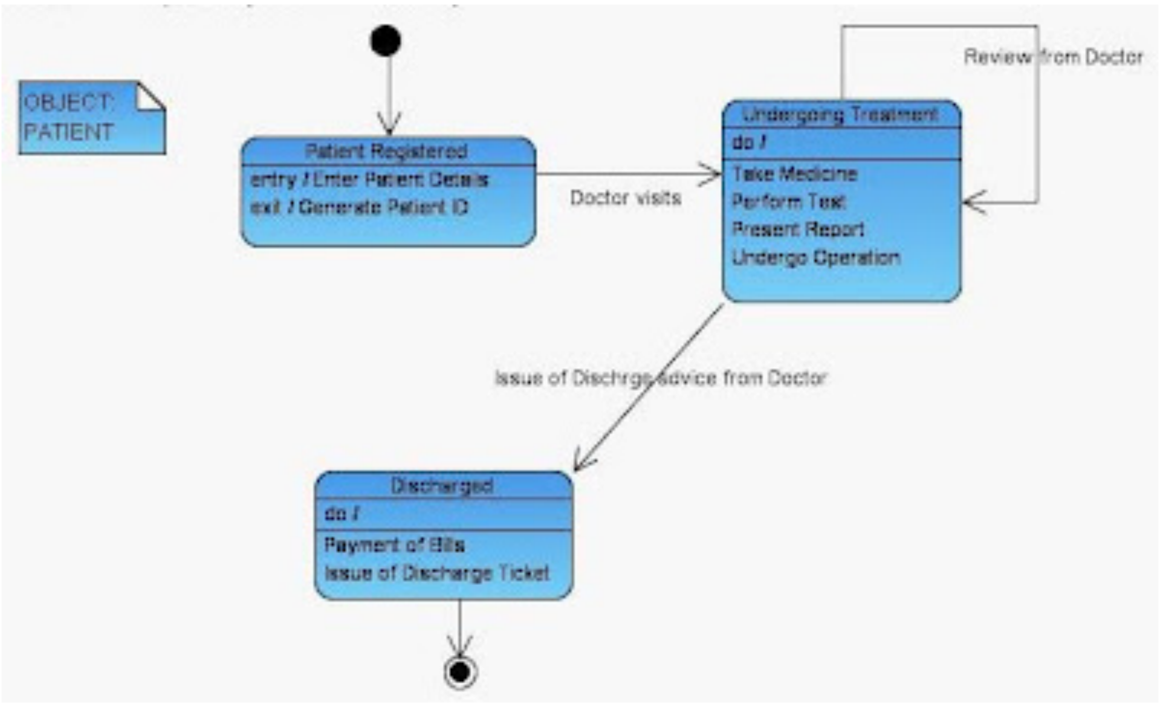
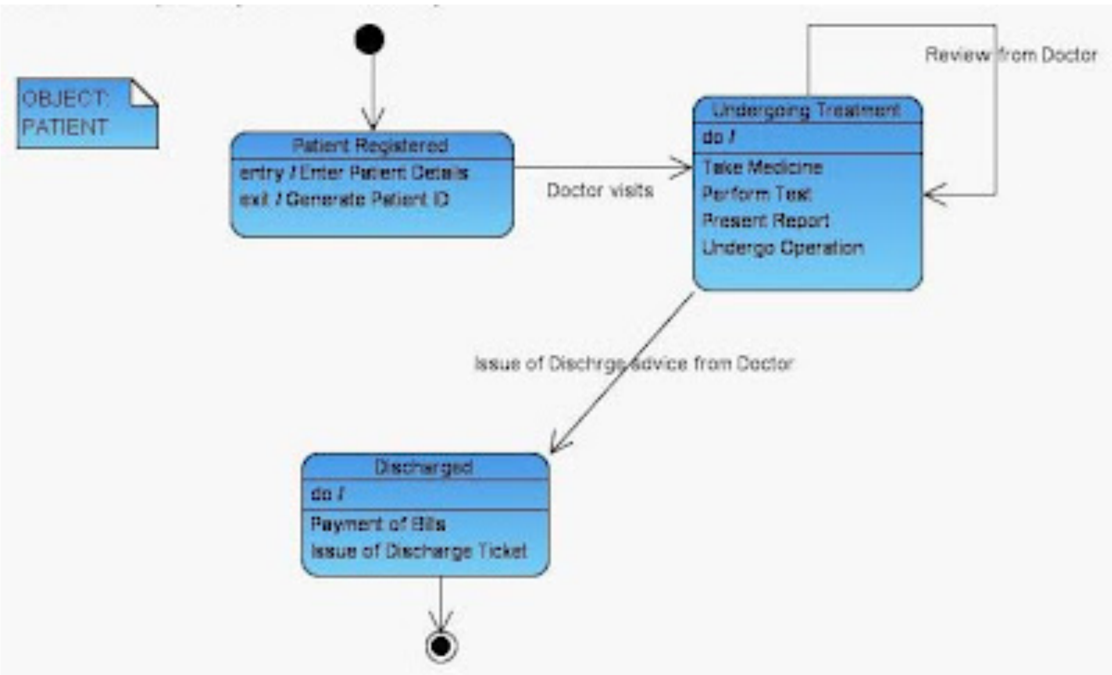


Figure 3.4 State Chart for a Patient

- **State Chart for a Doctor**



- **Activity Diagram for Patient Treatment**

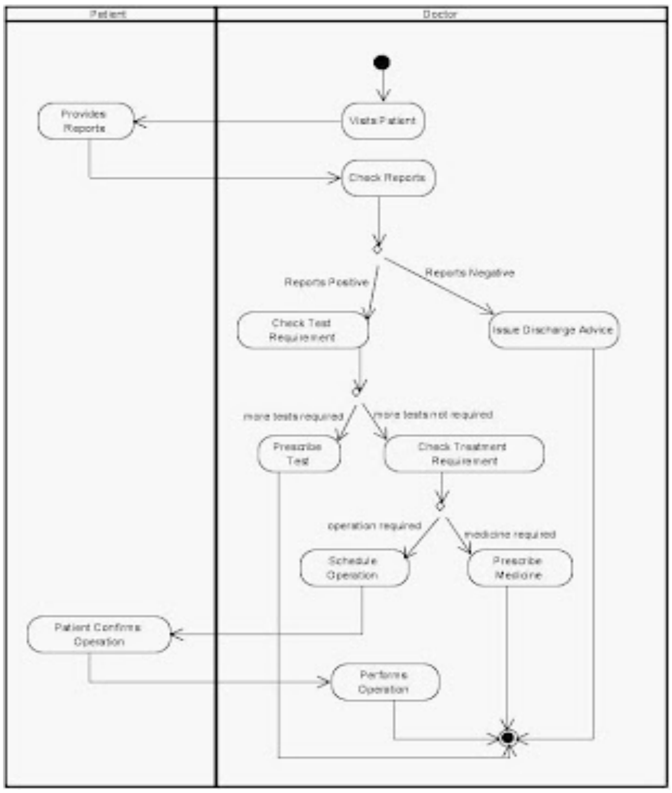


Figure 3.5: Activity Diagram for Patient Treatment

- **Activity Diagram for Patient Registration**

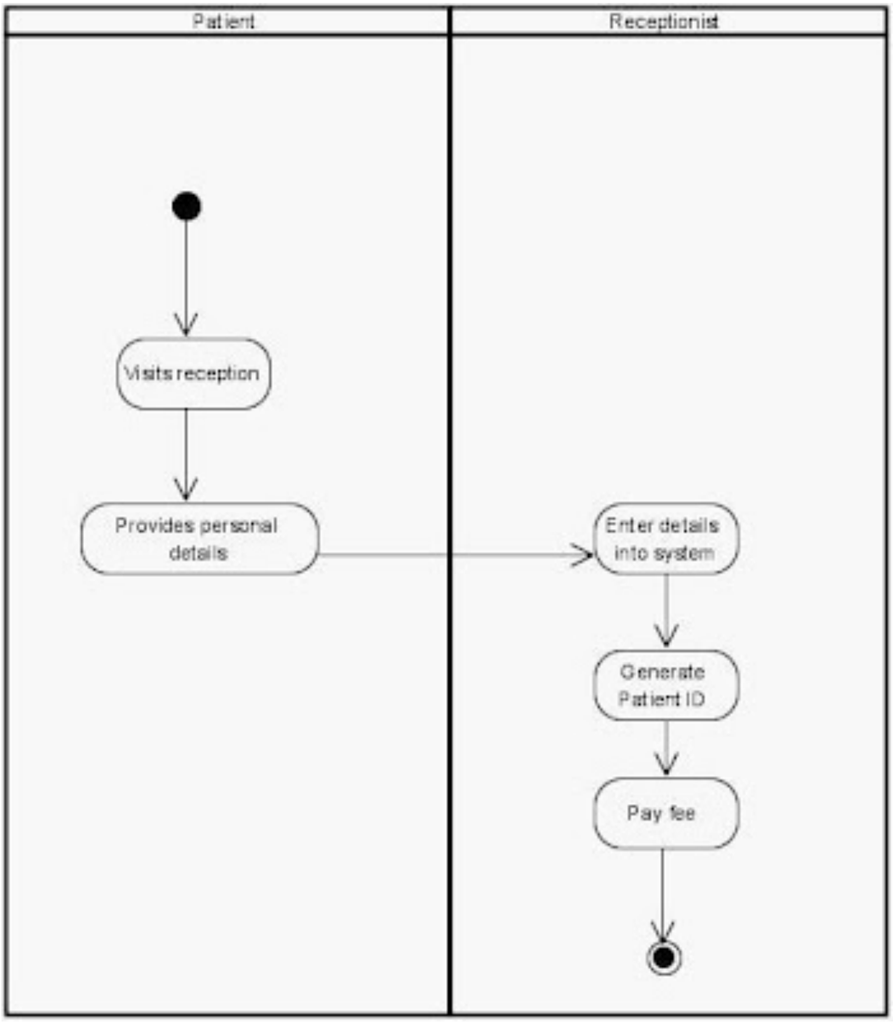


Figure 3.6: Activity Diagram for Patient Treatment

- **System Collaboration Diagram**

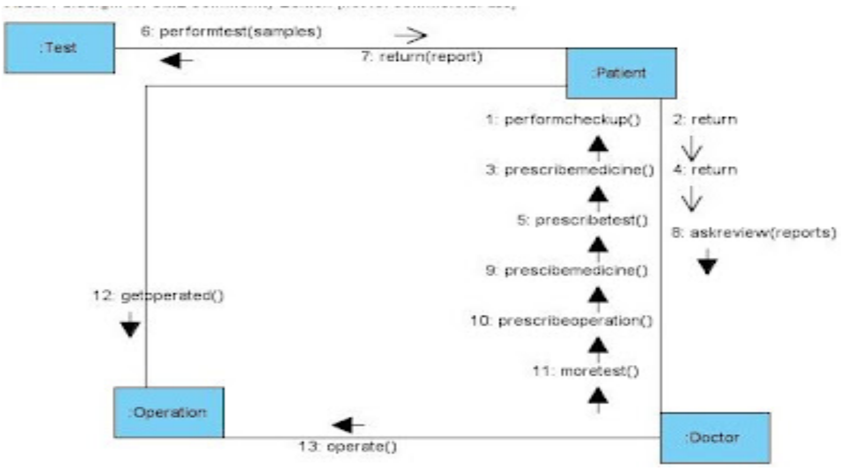


Figure 3.6: System Collaboration Diagram

3.3.2 Logical Database Design

The logical database design is meant to describe the representation of the database in terms of its entities in form of tables and the existing relationships. Below is an illustration of the systems logical design as generated by the MYSQL workbench design tool.

3.4 System Implementation

Implementation is a very important aspect in the development of any computerized system, and this also applies to the development of a records management system. Pro-development Implementation usually involves two main steps, these are;

- **System Construction:** The system is built and tested to make sure it performs as designed.
- **Installation:** Preparation is made to support the installed system. This involves associated documentation.

The System encompasses all the activities associated with the recording of patient medical history and development progress all of which are integrated in the Clinic Records Management System.

The main functionalities available in this system are:

- Maintaining patient details records
- Maintaining payment records of patients

All these features include the ability to create, update (edit), retrieve through search results and truncate obsolete records. It also contains a report generation system that can be saved in a pdf file format.

The system works in the following manner:

3.4.1 Accessing the System:

A user starts the process by logging into the system by means of a valid username / password combination. A new user has to first be registered in order to obtain access to the system.

Users with administrative privileges reserve the exclusive authorization to register new system users. A default administrative account has been provided by the system designers in order to enable the administrator to access exclusive privileges such as registering new users with either limited (normal user) or unlimited (administrative) privileges.

During the process of user registration, all users are issued with a unique user name and password combination as well as a specific user type (limited or unlimited). This combination is then used by the registered user to access the system resources that fall under their privilege level.

A user gains access to the system resources after a username password combination has been verified as accurate after which they are redirected to the homepage. The system homepage serves as the gateway to the entire records management system. Therefore, once a user is logged into the system they can access all system resources available to them based on their privilege level.

Once logged into the system, the user can create, manipulate and truncate records. However, the amount of manipulation that a user can perform with regards to the records is dependent on user privilege levels as explained below:

User Privileges

The system maintains two levels of users:-

- Administrator Level
- User Level-Data Entry Operator

Administrator Level

The administrator level is reserved for the database administrator. The administrator maintains the exclusive privilege to access ALL system resources and therefore has unlimited access to the system. In the designed system, the administrator has the following exclusive privileges;

- Granting and revoking access to system resources to users. This involves registering and deregistering users.
- Maintaining all tables with fixed data values for example, table with list of immunisable diseases which only needs to be updated once in a few years or never at all.
- Truncating Obsolete Records.

User Level

The user level is reserved for the data entry operator. This user's privileges are limited to only entering in data, specifically, data collected regularly about the children. They have the ability to add and edit data that they have access to but cannot truncate any record.

3.4.2 Pseudo Code

Log in to system

```
Startup system
Enter login name and password
On clicking the login button
Connect to database
Query database to know whether user credentials are correct
```

```
If not
Deny access and return login page with an error message
If correct
Check if credentials are for administrator
If yes
Allow login
Set admin session
Redirect administrator to admin home page
If no
Allow login
Set user session
Redirect user to user home page
```

Add New User

```
Check if administrator is logged in
If correct
Check if all fields entered are correct
If correct
Check if unique field value entered already exists
If correct
System message: user already exists
If not
Registration of user successful
```

Add New Record

Enter Record Details

If record exists

Return record already exists

If not

Registration of Record successful

Edit Record

Click on edit button

Query the database to retrieve details

If record exists

Return record details

Check if all fields entered are correct

If not

System message: fields incorrect

If correct

System message: record successfully edited

Delete Record

Check if administrator is logged in

If not

System message: no sufficient rights to perform this operation

If correct

enter recordID

If record ID exists

Delete record from table

If record ID does not exist

System message: sorry! record does not exist

CHAPTER 4: SYSTEM TESTING AND RESULTS

4.1 Introduction

Testing is critical for a newly developed system as a prerequisite for it being put into an environment where the end users can use it. Exhaustive testing is conducted to ensure accuracy and reliability and to ensure that bugs are detected as early as possible. In the process of designing the RMS, three levels of testing were conducted, namely, unit testing, integration testing and system testing.

4.2 System Test

Experiment 1: Operation and Performance of System, Creation, Deletion of Patient

AIM:

To check if the Creation and deletion scripts are working according to the requirements and if also their performance is acceptable.

OBJECTIVES:

- Check if the scripts are working – able to create and delete patients and users from the system
- Assess the performance of the scripts with different workloads

TOOLS NEEDED:

The NetBeans IDE (Version 6 and above), Apache Web Server, Web browser, The UNZA Clinic RMS and configuration files, Google's Speed Tracer, and Apache JMeter either as a NetBeans plugin or as a seperate application.

METHODOLOGY

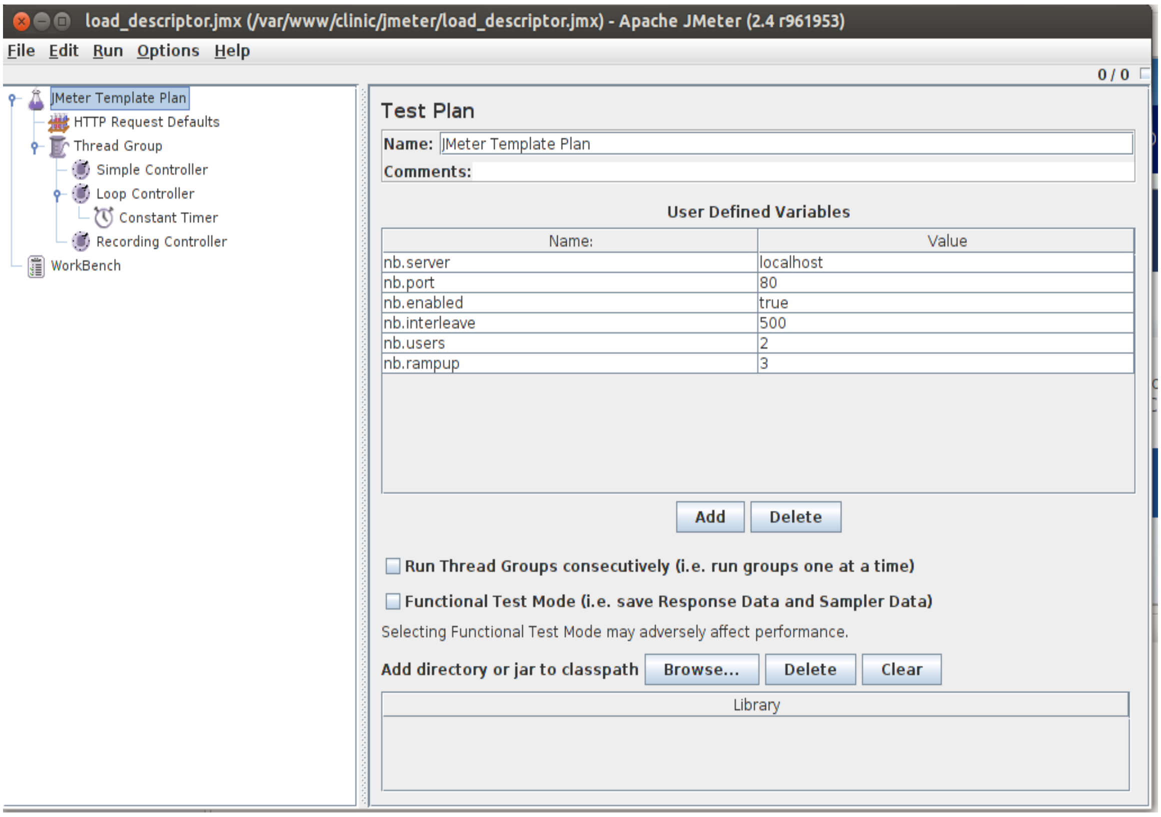
First, the IDE s started. After which it is fully loaded, the JMeter load generating script is edited and configured to provide a particular load on the test plan. A proxy server is then configured on JMeter to run the tests locally.

In this experiment, the load was in terms of the number of users using the system at the same particular point to perform the operations listed in Table 4.0. Speed tracer is started the moment the application is loaded. The different test cases are listed below

Table 4.1: Test Cases

USE CASE	SERVER REQUEST
Creating User	http://clinic/addUser.php
Delete User	http://clinic/deleteUser.php?id

Create Patient	http://clinic/addPatient.php



Test Plan

Name: JMeter Template Plan

Comments:

User Defined Variables

Name:	Value
nb.server	localhost
nb.port	80
nb.enabled	true
nb.interleave	500
nb.users	2
nb.rampup	3

□ Run Thread Groups consecutively (i.e. run groups one at a time)

□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

Figure 4.1: JMeter Test Template Page

The application is started with a single user session. All the cases in the table above are simulated by doing each of the individual use cases. Results are recorded in a table for whether each of the operations was successful. The process is repeated for different loads ranging from 1-700

For each of the tests in Jmeter, the interleave period (nb.interleave) was swt to 5000ms. The value for nb.interleave is used in the user delay Test action that was created, and represents the duration of a pause in miliseconds. Therefore, the test will pause for 5 seconds between each user-initiated request simulated. The results are recorded for whether the user/patient was created and/or deleted. Also the performances for different loads are recorded.

DATA COLLECTED:

Table 4.2: Operation Results of experiment 1

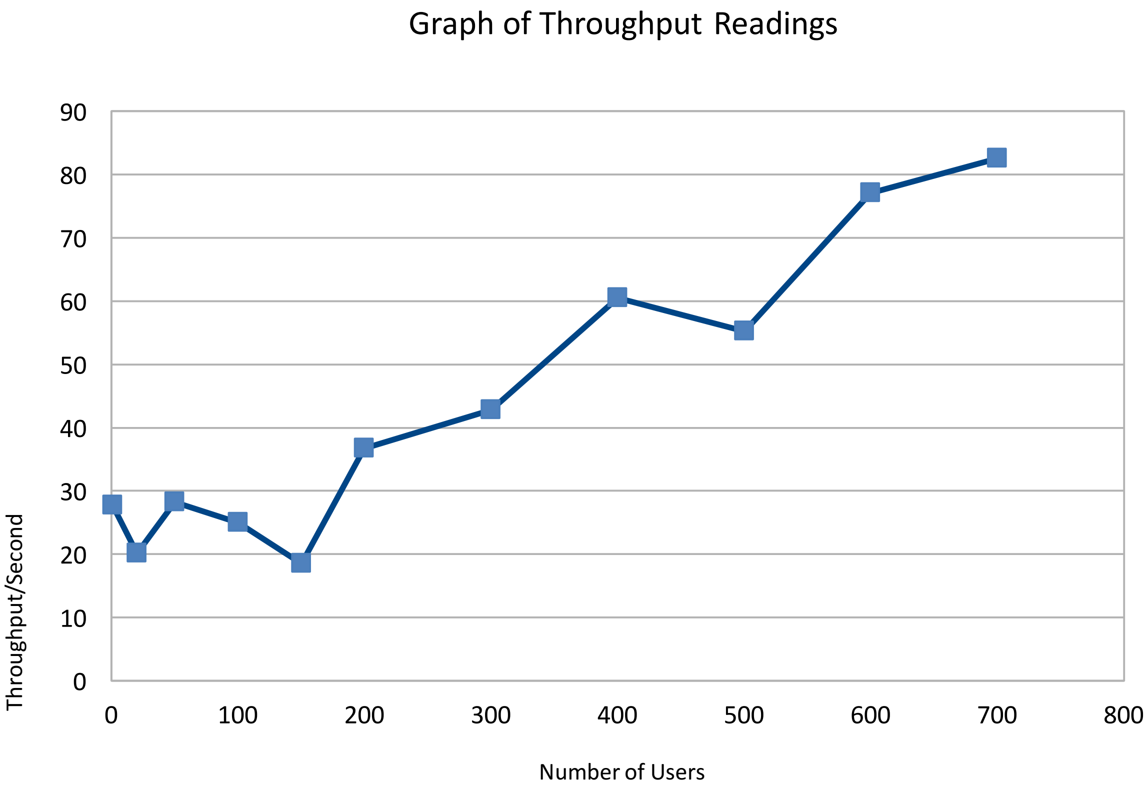
Test	**Result**	**Implication**
Creating User	Successful	The user script created the user on the system

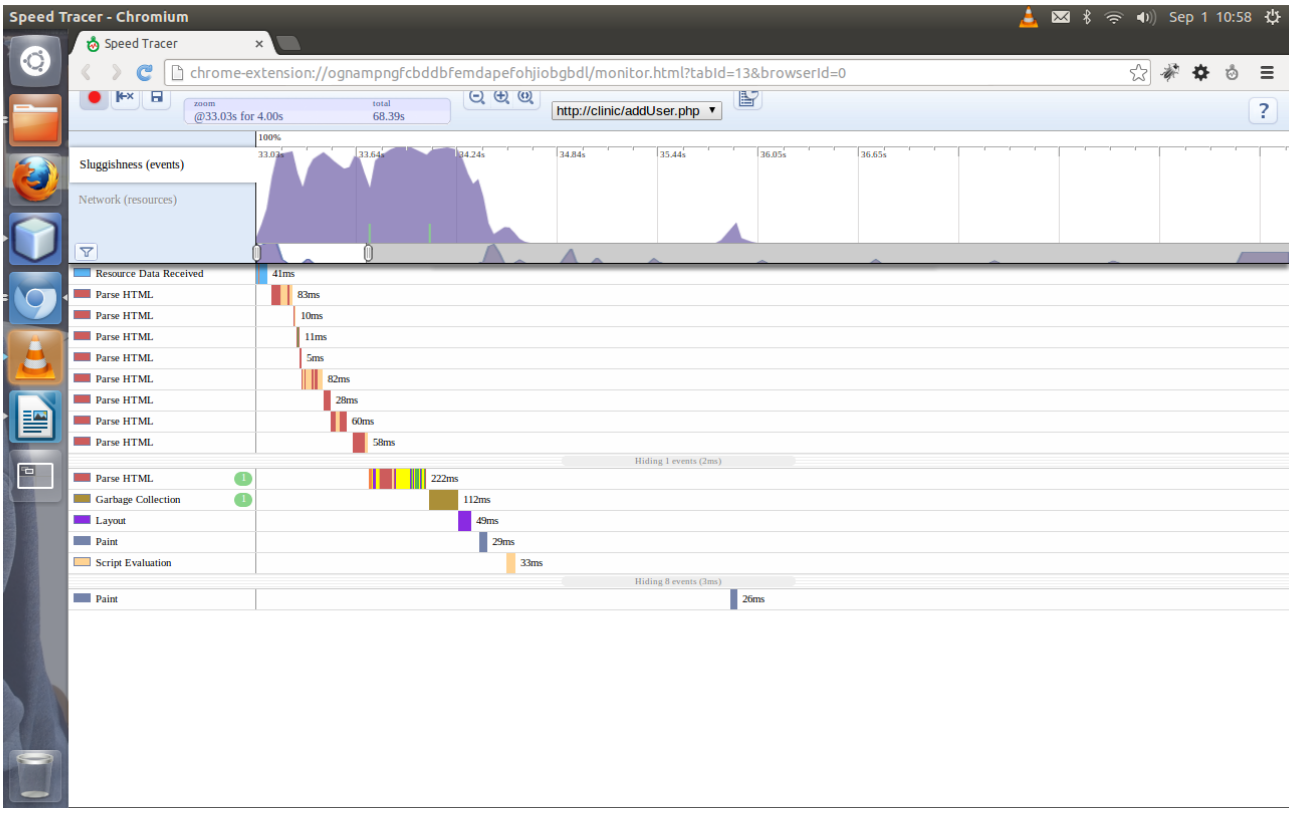
Delete User	Successful	The delete script deleted the user off the system
Create Patient	Successful	The create patient script did create the patient on the system

Table 4.3: Performance Test Results

Users	**Ramp-Up (Seconds)**	**Average (Miliseconds)**	**Throughput**	**Error %**
1	1	11	27.2	20.000
20	20	9	20.1	20.000
50	25	8	28.2	20.000
100	25	8	25.0	20.000
150	25	8	18.5	20.000
200	25	7	36.7	20.000
300	25	6	42.8	20.000
400	25	120	50.5	20.000
500	25	355	55.2	20.000
600	25	369	77.1	20.000
700	25	447	82.5	20.000

Figure 4.2: Graph of Throughput readings





```
Resource Data Received   41ms
Parse HTML               83ms
Parse HTML               10ms
Parse HTML               11ms
Parse HTML               5ms
Parse HTML               82ms
Parse HTML               28ms
Parse HTML               60ms
Parse HTML               58ms
                         Hiding 1 events (2ms)
Parse HTML           1   222ms
Garbage Collection   1   112ms
Layout                   49ms
Paint                    29ms
Script Evaluation        33ms
                         Hiding 8 events (3ms)
Paint                    26ms
```

Figure 4.3: Summary chart of add/delete user with Speed Tracer

4.3 Results and Evaluation of Experiment

The experiment had both quantitative and qualitative results. The qualitative results emphasised on whether or not the operational goals of this project, were achieved or not and the quantitative part aimed at finding out the performance of these operations. For the qualitative part Table 4.2 shows that all the scripts for deletion and creation of patients and/or users were able to do their designates tasks.

- The Add Scripts were able to handle both the Creation of Users and Patients
- The Delete Script was able to handle the deletion of Users from the system

For the quantitative part, the results recorded in this were arrived at by running tests on a computer with 2GHz Intel core Dual core processor and 2GB of DDR3 memory. Naturally, the results obtained depend on the computers performance, and in each likely not match those on a different machine.

In each of the readings for a particular number of users, as can be seen from Table 4.3, the following information is recorded:

- Samples – Number of user requests
- Minimum, Maximum and Average Time for request
- Standard Deviation

- Error – error in the request being processed
- Throughput – Number of requests handled in a particular period of time
- Bandwidth – Speed at which request was processed.

The total throughput of the application for each of the different number of users is in Table 4.3. the server when loaded with a single user provides a total throughput approximately 41 per minute. In other words, the server is capable of serving all requests within the test plan 41 times within a minute. According to the Jmeter User's manual, the throughput is calculated as: (number of requests)/(total time), and includes any delay inserted between samples, as it is supposed to represent the load on the server.

The ramp-up time (nb.rampup) varable is used in the RMS Users Thread Group panel in Jmeter specifies the ramp-up time in seconds. In this case, if the test is for 20 users and the ramp-up period is 20 seconds, hten a new user will begin sending requests every second. For 200 users and above, the tests started taking longer. At times Jmeter became unresponsive and it was necessary to shut it down form the task manager and the tests redone.

4.4 Summary

This chapter gave a more realistic detail of the performance of the application against the server constraints. Jmeter was used typically to test both dynamic and static resources of the application. Speed tracer was used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. Both these tools were used to make graphical analysis of performance and test script behaviour under concurrent load .

CHAPTER 5: DSCUSSION AND CONCLUSIONS

5.1 Introduction

The experiments conducted in the previous chapter showed very interesting results. Experimentation is essential because before delivering any application, it is important to ensure that it functions properly, and that it can perform acceptably for the demands expected of it. Web applications, especially Web 2.0 interaction applications, by their nature provide concurrent access to shared resources. In other words, the servers on which they are hosted must be able to respond to multiple users requesting the same resources over the same period of time. It is very important for a developer to be mindful of this fact when during development the application appears to behave properly as a user clicks through web pages in a browser. How will the application behave when handling 100 users simultaneously? Are there memory leaks that will degrade the server's performance after the application has been running for long periods of time?

5.2 Discussion

The experiment showed that the application was able to properly create and handle users, submit and retrieve information, which was the main aim of creating the application. However, this is just a *black box test* on the application showing that it does what it needs to do without considering the program code and database interaction behind the functionality. This is why a quantitative performance test was essential for each of the requests to the server associated to the entire qualitative black box in order to have an indirect white box test. This is basically a load test. Load testing a web application involves making concurrent requests for resources, typically by simulating multiple users, and then examining the servers behaviour. The same test plan, was used with adjustments made to te number of users and other setting to determine how the server behaves under the given workload.

The load testing in this case was performed with specific goals in mind, such as determine the throughput for the anticipated average and peak numbers of users. It is also worthwhile to assess the response time for reqests, to ensure that the application users are not having to wait too long to be served. Once again, the requests recorded in this and the following sections were arrived at by running tests on a computer with 2.0GHz Intel Dual Core processor and 2.0 GB DDR3 memory. The performance test for a single user was as shown below:

Table 5.1: Performance results for 1 user in JMeter

Label	Sample	Average	Throughput	Min	Max	Std. Deviation	KB/Sec	Error %
/clinic	1	41	24/sec	41	41	0.00	44.5	0.00
index.php	1	44	22.7/sec	44	44	0.00	43.4	0.00

addUser.php	6	24	28.2/sec	57	8	19.36	16.08	0.00
queuePatient.php	1	70	25.0/sec	36	36	0.00	53.06	0.00
SearchPatient.php	1	36	18.5/sec	70	70	0.00	1.04	0.00
report.php	2	45	36.7/sec	28	40	0.00	8.03	100.00
makePayment.php	1	31	42.8/sec	45	45	0.00	35.07	100.00
searhpatient.php	1	77	50.5/sec	33	23	0.00	83.05	0.00
Viewpaymen..php	2	56	55.2/sec	36	36	0.00	53.06	0.00
logout.php	1	33	24.1/sec	10	52	21.00	27.00	0.00

From the results displayed above, we can observe that:

- The server, when loaded with a single user, provides a total throughput of about 27.9 per minute. In other words, the server is capable of handling all server requests within the test plan 27.9 times within a minute. Once more, according to the Jmeter user manual, the throuput is calculated as: (number of request)/(total time), and includes any delay inserted between samples, as it is supposed to represent the load on the server.

- The searchpatient.php request, recorded at 77 milliseconds, takes much longer to process than most requests. This likely due to the fact that, the searching the databases takes awfully longer than any of these requests.

- According to the Error % column, several requests had errors that occurred from running the test. In other words, **not** all server responses included an HTTP 200 status, some HTTP 404 status.

Referring to the Speed Tracer diagram, it can be realised that, the entire system has an overall out mate performance of a role that's implementable. The turnaround time for the entire system is 28.5 Seconds which is admissible.

5.3 Conclusion

The ultimate goal and aim of the project was to create a Records Management system that provides quick and instant access to patient's records. The researcher can confidently say that this has been met; additionally the objectives of this project were met.

Furthermore the resources listed in this report suffice for this project to be implemented, as well as the qualitative results performed have shown that the application can withstand a great deal of load.

The researcher can confidently conclude that the project was a success; the application meets all the user requirements and is usable in a local environment such as the university of Zambia clinic to provide a quick access to the patients records as well as management.

5.4 Future Works, Challenges Faced and Recommendations

This system was version 1, a lot has to be done and this includes, adding an announcing mechanism for next patient in the queue. Provide adequate, FAQ functionality on the system so that new users don’t need to be trained on how to use the system.

The greatest challenge the researcher faced understood the requirements which kept on changing. Furthermore, load testing software was difficult to find, the ones that came through are not reliable, those are reliable are expensive. Additionally, the researcher had a problem with logistical funds, as the sponsor did not come through.

The researcher recommends that, the powers that maybe should put in place measures to have logistical money be credited to the next team to work on the project. Additionally, certain load testing software should be provided by the powers that maybe.

5.5 Summary

This chapter gave a summary and conclusive remark on the entire project. It resounded the findings of the experiment conducted on the application. It is worth noting otherwise that despite all the challenges faced, the project was a total success!

APPENDICES

APPENDIX 1 – INSTALLING APACHE WEBSERVER.

Apache is the most commonly used Web Server on Linux systems. Web Servers are used to serve Web Pages requested by client computers. Clients typically request and view Web Pages using Web Browser applications such as *Firefox*, *Opera*, *Chromium*, or *Mozilla*.

Users enter a Uniform Resource Locator (URL) to point to a Web server by means of its Fully Qualified Domain Name (FQDN) and a path to the required resource. For example, to view the home page of the clinic.cs.unza.zm a user will enter only the FQDN

The most common protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). Protocols such as Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS), and File Transfer Protocol (FTP), a protocol for uploading and downloading files, are also supported.

The *Apache2* web server is available in Ubuntu Linux. To install Apache2:

```
sudo apt-get install apache2
```

Apache2 is configured by placing *directives* in plain text configuration files. These *directives* are separated between the following files and directories:

1. *apache2.conf:* the main Apache2 configuration file. Contains settings that are *global* to Apache2.
2. *conf.d:* contains configuration files which apply *globally* to Apache2. Other packages that use Apache2 to serve content may add files, or symlinks, to this directory.
3. *envvars:* file where Apache2 *environment* variables are set.
4. *httpd.conf:* historically the main Apache2 configuration file, named after the *httpd* daemon. Now the file is typically empty, as most configuration options have been moved to the below referenced directories. The file can be used for *user specific* configuration options that globally effect Apache2.
5. *mods-available:* this directory contains configuration files to both load *modules* and configure them. Not all modules will have specific configuration files, however.
6. *mods-enabled:* holds *symlinks* to the files in /etc/apache2/mods-available. When a module configuration file is symlinked it will be enabled the next time *apache2* is restarted.
7. *ports.conf:* houses the directives that determine which TCP ports Apache2 is listening on.
8. *sites-available:* this directory has configuration files for Apache2 *Virtual Hosts*. Virtual Hosts allow Apache2 to be configured for multiple sites that have separate configurations.
9. *sites-enabled:* like mods-enabled, sites-enabled contains symlinks to the /etc/apache2/sites-available directory. Similarly when a configuration file in sites-available is symlinked, the site configured by it will be active once Apache2 is restarted.

In addition, other configuration files may be added using the *Include* directive, and wildcards can be used to include many configuration files. Any directive may be placed in any of these configuration files. Changes to the main configuration files are only recognized by Apache2 when it is started or restarted.

The server also reads a file containing mime document types; the filename is set by the *TypesConfig* directive, typically via /etc/apache2/mods-available/mime.conf, which might also include additions and overrides, and is /etc/mime.types by default.

APPENDIX 2 – INSTALLING MySQL DATABASE SERVER

MySQL is a fast, multi-threaded, multi-user, and robust SQL database server. It is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

To install MySQL, run the following command from a terminal prompt:

```
sudo apt-get install mysql-server
```

During the installation process you will be prompted to enter a password for the MySQL root user.

Once the installation is complete, the MySQL server should be started automatically. You can run the following command from a terminal prompt to check whether the MySQL server is running:

```
sudo netstat -tap | grep mysql
```

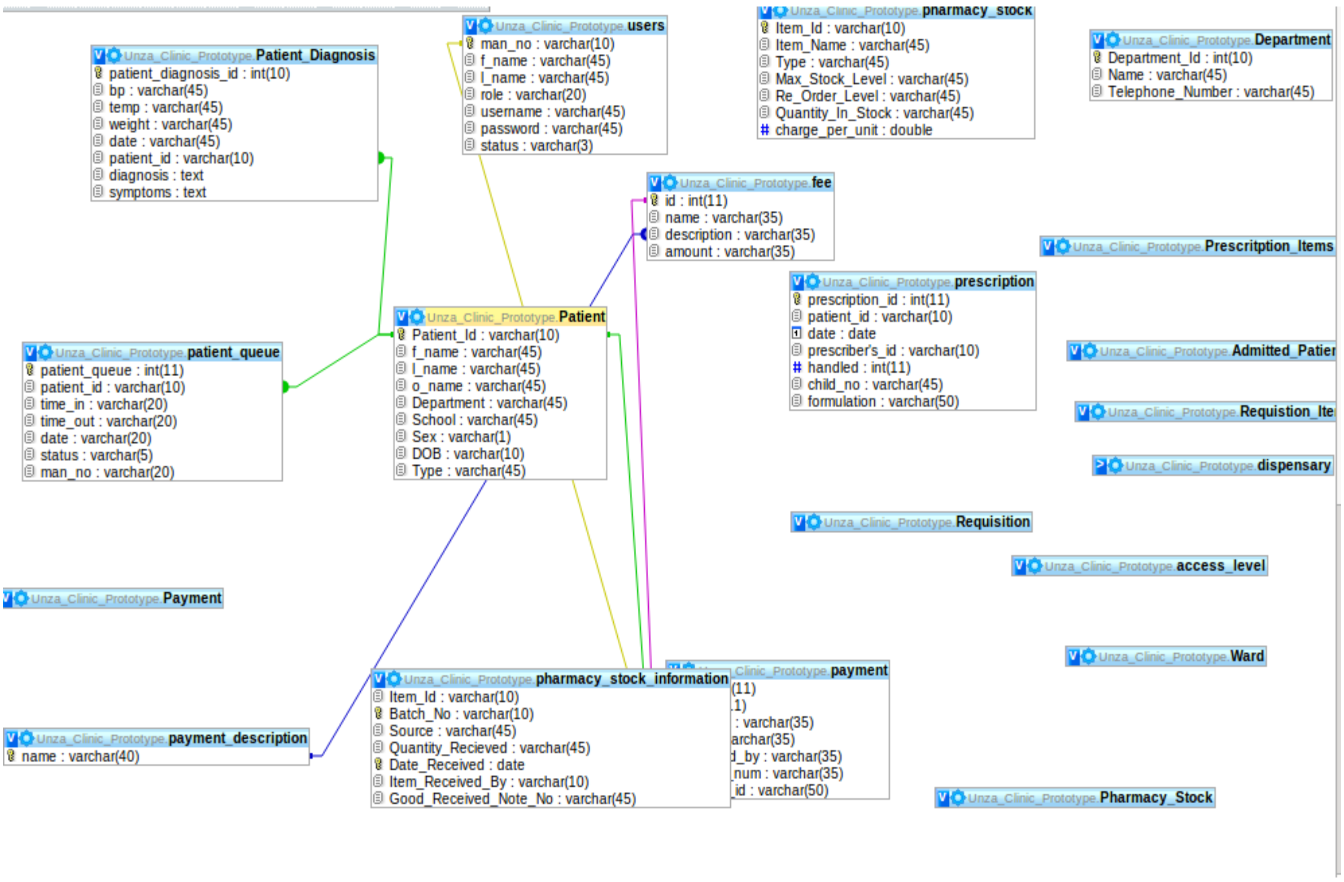
When you run this command, you should see the following line or something similar:

```
tcp        0      0 localhost:mysql        *:*                 LISTEN
2556/mysqld
```

If the server is not running correctly, you can type the following command to start it:

```
sudo service mysql restart
```

APPENDIX 3 – DATABASE SCHEMA



Unza_Clinic_Prototype.Patient_Diagnosis
patient_diagnosis_id : int(10)
bp : varchar(45)
temp : varchar(45)
weight : varchar(45)
date : varchar(45)
patient_id : varchar(10)
diagnosis : text
symptoms : text

Unza_Clinic_Prototype.users
man_no : varchar(10)
f_name : varchar(45)
l_name : varchar(45)
role : varchar(20)
username : varchar(45)
password : varchar(45)
status : varchar(3)

Unza_Clinic_Prototype.pharmacy_stock
Item_Id : varchar(10)
Item_Name : varchar(45)
Type : varchar(45)
Max_Stock_Level : varchar(45)
Re_Order_Level : varchar(45)
Quantity_In_Stock : varchar(45)
charge_per_unit : double

Unza_Clinic_Prototype.Department
Department_Id : int(10)
Name : varchar(45)
Telephone_Number : varchar(45)

Unza_Clinic_Prototype.fee
id : int(11)
name : varchar(35)
description : varchar(35)
amount : varchar(35)

Unza_Clinic_Prototype.Prescritption_Items

Unza_Clinic_Prototype.prescription
prescription_id : int(11)
patient_id : varchar(10)
date : date
prescriber's_id : varchar(10)
handled : int(11)
child_no : varchar(45)
formulation : varchar(50)

Unza_Clinic_Prototype.Patient
Patient_Id : varchar(10)
f_name : varchar(45)
l_name : varchar(45)
o_name : varchar(45)
Department : varchar(45)
School : varchar(45)
Sex : varchar(1)
DOB : varchar(10)
Type : varchar(45)

Unza_Clinic_Prototype.patient_queue
patient_queue : int(11)
patient_id : varchar(10)
time_in : varchar(20)
time_out : varchar(20)
date : varchar(20)
status : varchar(5)
man_no : varchar(20)

Unza_Clinic_Prototype.Admitted_Patier

Unza_Clinic_Prototype.Requistion_Ite

Unza_Clinic_Prototype.dispensary

Unza_Clinic_Prototype.Requisition

Unza_Clinic_Prototype.access_level

Unza_Clinic_Prototype.Payment

Unza_Clinic_Prototype.Ward

Unza_Clinic_Prototype.pharmacy_stock_information
Item_Id : varchar(10)
Batch_No : varchar(10)
Source : varchar(45)
Quantity_Recieved : varchar(45)
Date_Received : date
Item_Received_By : varchar(10)
Good_Received_Note_No : varchar(45)

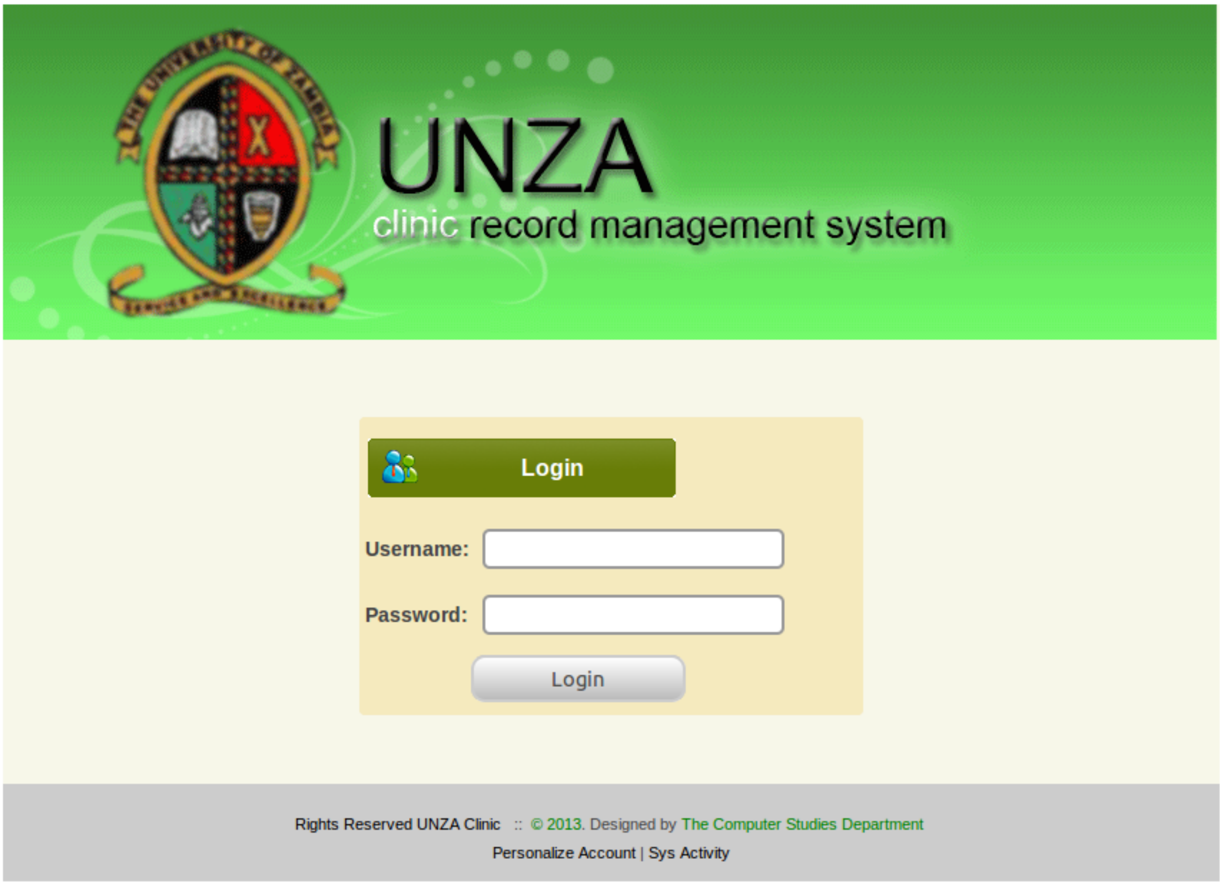
Clinic_Prototype.payment
(11)
1)
: varchar(35)
archar(35)
d_by : varchar(35)
num : varchar(35)
id : varchar(50)

Unza_Clinic_Prototype.payment_description
name : varchar(40)

Unza_Clinic_Prototype.Pharmacy_Stock

APPENDIX 4 – SAMPLE SCREEN SHOTS

Login screen



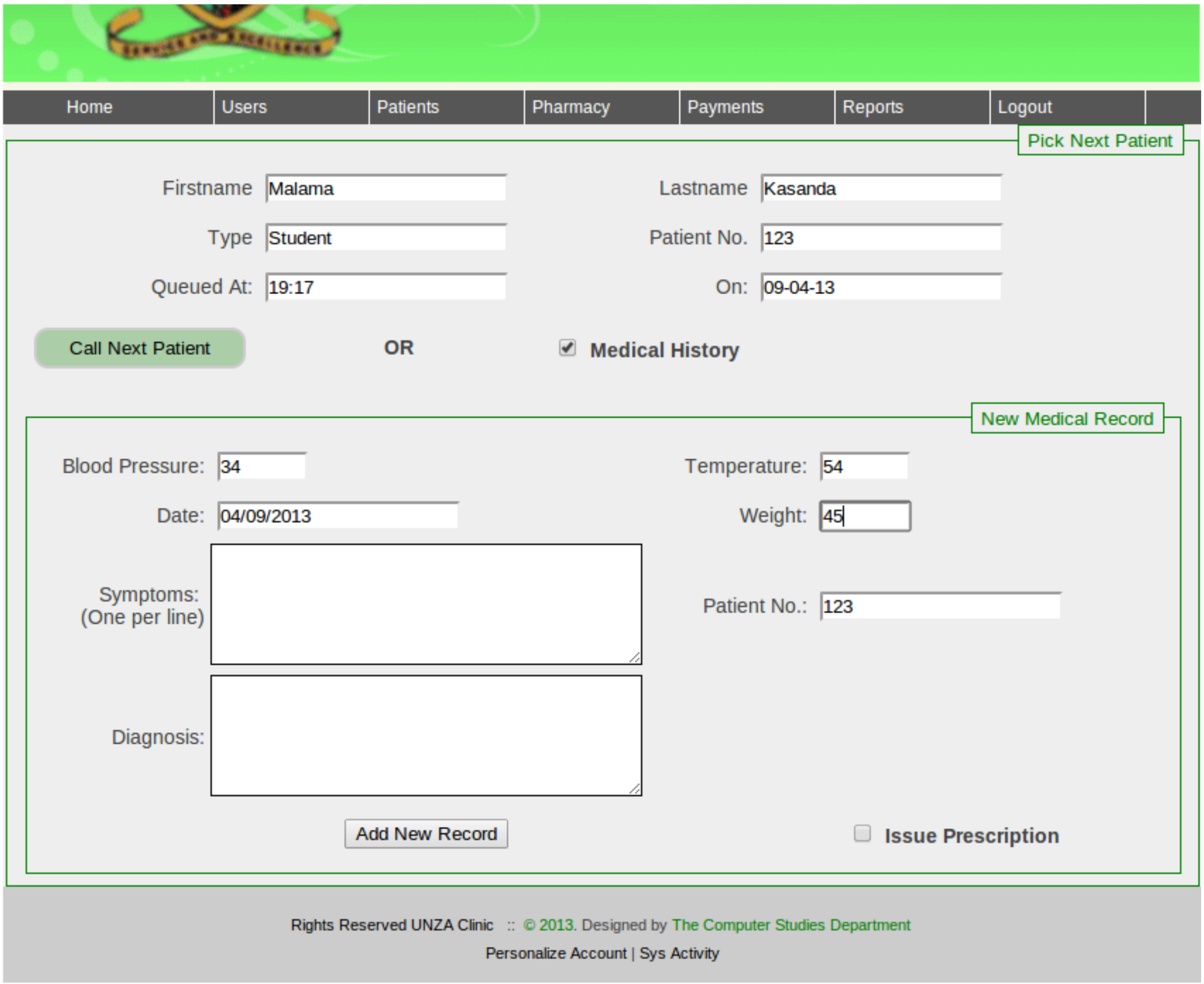
Username:

Password:

Rights Reserved UNZA Clinic :: © 2013. Designed by The Computer Studies Department

Personalize Account | Sys Activity

Add Medical History



Firstname: Malama

Lastname: Kasanda

Type: Student

Patient No.: 123

Queued At: 19:17

On: 09-04-13

☑ Medical History

Blood Pressure: 34

Temperature: 54

Date: 04/09/2013

Weight: 45

Symptoms: (One per line)

Patient No.: 123

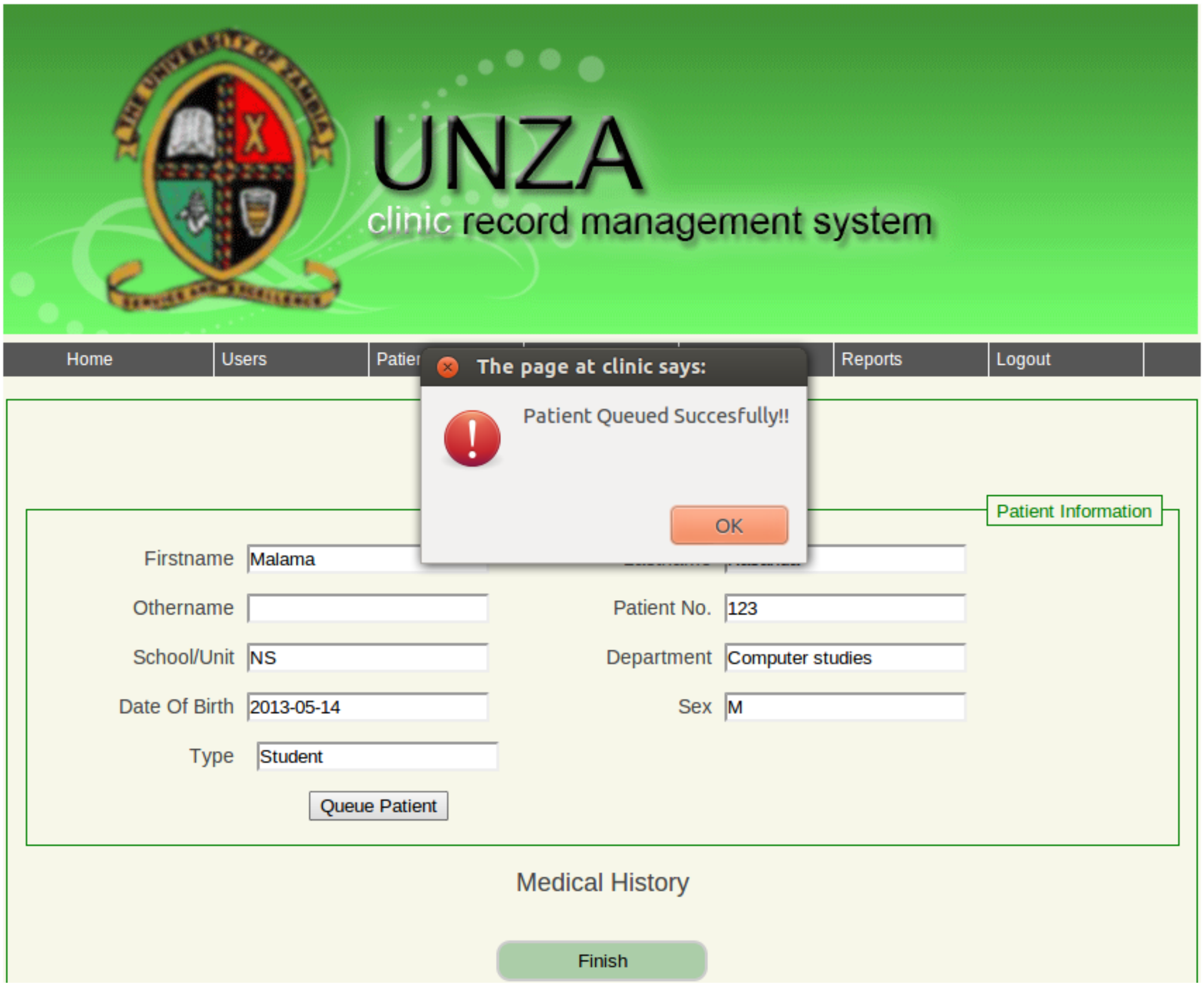
Diagnosis:

□ Issue Prescription

Rights Reserved UNZA Clinic :: © 2013. Designed by The Computer Studies Department

Personalize Account | Sys Activity

Queue Patient



The page at clinic says:

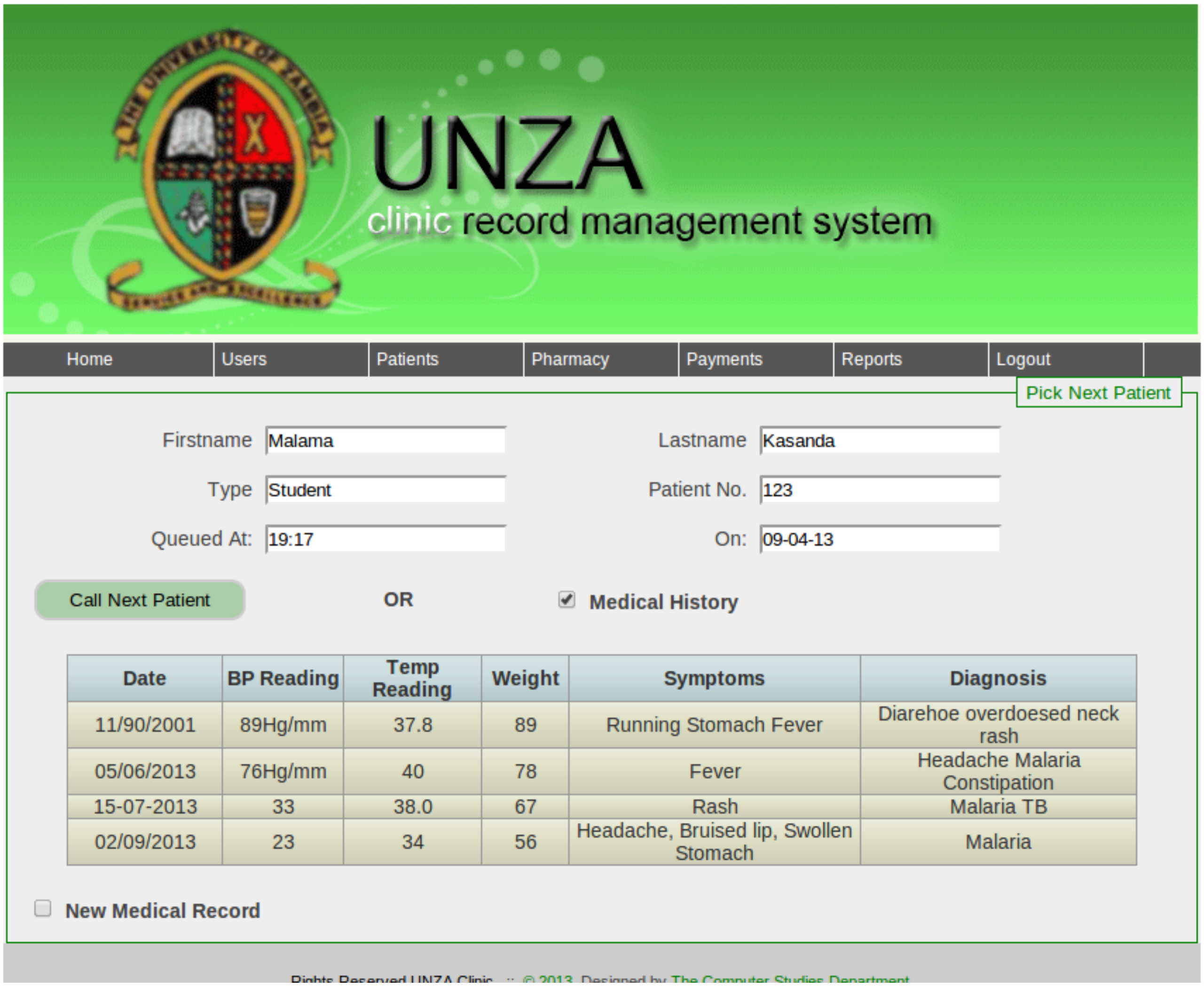
Patient Queued Succesfully!!

Patient Information

Firstname	Malama		
Othername		Patient No.	123
School/Unit	NS	Department	Computer studies
Date Of Birth	2013-05-14	Sex	M
Type	Student		

Medical History

Medical History



Firstname	Malama	Lastname	Kasanda
Type	Student	Patient No.	123
Queued At:	19:17	On:	09-04-13

☑ Medical History

Date	BP Reading	Temp Reading	Weight	Symptoms	Diagnosis
11/90/2001	89Hg/mm	37.8	89	Running Stomach Fever	Diarehoe overdoesed neck rash
05/06/2013	76Hg/mm	40	78	Fever	Headache Malaria Constipation
15-07-2013	33	38.0	67	Rash	Malaria TB
02/09/2013	23	34	56	Headache, Bruised lip, Swollen Stomach	Malaria

□ New Medical Record

Rights Reserved UNZA Clinic :: © 2013 Designed by The Computer Studies Department

REFERENCES

[1] S. I. Allen, G. O. Barnett, and P. A. Castleman. Use of a time-shared general purpose file handling system in hospital res. Proc.IEEE,54, 2008.

[2] G. O. Barnett, R. D. Zielstorff, J. McLatchey, et al. Costar: A comprehensive medical information system for ambulatory care. In Symposium on Computer Applications in Medical Care, volume 6, pages 8-18. IEEE Press, 2011.

[3] S. M. Bobrowski. MySQL 7 & Client/Server Computing. Sybex, San Francisco, CA, 2010.

[4] C. J. Date. A Guide to the SQL Standard: A User's Guide to the Standard Relational Language SQL. Addison-Wesley Publishing Company, Reading, MA, 2007.

[5] T. Connoly, C. Begg. Database Systems: A Practical approach to Design, Implementation and Management. 3rd Edition. Page 274

[6] I. S. Kohane. Getting the data in: three year experience with a pediatric electronic medical record system. In Symposium on Computer Applications in Medical Care, volume 18, pages

[7] National Library of Medicine. UMLS Knowledge Sources, seventh experimental edition, 1996.

[8] Oracle Corp. Oracle Database Designer. http://www.oracle.com/products/ tools/dbdes/html/.

[9] D. Meier. The Theory of Relational Databases. Computer Science Press, 2002.

[10] Understanding the MVC: www.mvcexplaned.com/mvc.htnl

[11] Q. E. Whiting-O'Keefe, A. Witting, and J. Henke. The STOR clinical information system. M.D. Computing, 5(5):8-21, 2003.

[12] Bardram, J. (2005). I Love the System – I just don’t use it!. ACM Digital Library

LIST OF PUBLICATIONS

[1] Baroudi, J., J., Olson, M., H., & Ives, B., (2004). An Empirical Study of the Impact of User Involvement on System Usage and Information Satisfaction. Communication of ACM, 29(3),
232-238

[2] Carr, M., D., (2001). Serving Physician Information Needs, A model for the transition to Electronic Medical Record, Journal of Health Information Management, 13(3),.73- 84.

[3] Gefen, D., (1998). The Impact of Developer Responsiveness on Perceptions of Usefulness and Ease of Use: An Extension of Technology Acceptance Model. ACM Digital Library.