MAASAI MARA UNIVERSITY

DEPARTMENT OF COMPUTING AND INFORMATION SCIENCES

TITTLE:

HEALTH CENTRE MANAGEMENT INFORMATION SYSTEM, ITS DESIGN AND IMPLEMENTATION

(A Case Study of Maasai Mara University Health Unit)

RACHEL WANJIKU MAINA

IS01/025/2012

Submitted in Partial Fulfillment of the Requirements of a Degree in Bachelor of Science in Information Sciences in Maasai Mara University

May 2016

DECLARATION

I hereby declare that this project proposal submitted by me, Rachel W. Maina in partial fulfillment for the award of the degree of B.Sc. Information Sciences is a record of original work and has not been submitted either in part or in full, for the award of any other degree or award in this or other university.

Registration Number: ……

Sign : ……..Date: ……..……

UNIVERSITY SUPERVISOR APPROVAL

This Project has been submitted for examination with my acknowledged approval as the University Supervisor.

Mr. Kemboi

Lecturer, Maasai Mara University, Department of Computing and Information Sciences

Signature……..Date……..…….

DEDICATION

This project is dedicated to my best friend, a friend without whom this would not have been possible.

ACKNOWLEDGEMENT

I would love to acknowledge God Almighty first of all, the giver of all life, wisdom, intellect and ability. Secondly, I would like to acknowledge my family for making provision for me to conduct this research and my supervisor, Mr Kemboi for his time to refine my project. For that I am very grateful. My fellow school mates in Computer Science also deserve a mention for their invaluable support. The university's health unit staff cannot go unmentioned. Thank you for your support. Lastly, but not least, I would love to acknowledge my best friend who helped make this project a realization in its very first stages. You are a constant reminder of loyalty, commitment and dedication.

ABSTRACT

The continued expansion and growth in global technologies has led to the development of many new electronic health information management systems to improve efficiencies and quality of care within the field of medicine. Maasai Mara University is a growing world class university. There is a necessity for competent administration in the university's health unit.

This project suggests the computerizing of all the records about patients, staff and drug/equipment supplies in the health center. The health unit information management system will manage patients and staff and will automatically generate medical reports relevant to the clinic and provide a computerized stock inventory.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

Terminology	**Meaning**
HMIS	Health Management Information System
PMS	Medical Practice management Software
EHR	Electronic Health Record
EMR	Electronic Medical Record
RDBMS	Relational Database Management System
JRE	Java Virtual Machine

CHAPTER ONE

INTRODUCTION

1.0 Introduction

The maintenance of complete and accurate medical records is a requirement of health care providers and is generally enforced as a licensing or certification prerequisite. Handwritten paper medical records may be poorly legible, which can contribute to medical errors. Pre-printed forms, standardization of abbreviations and standards for penmanship were encouraged to improve reliability of paper medical records. Electronic records may help with the standardization of forms, terminology and data input. Digitization of forms facilitates the collection of data for epidemiology and clinical studies.

Automating healthcare activities such as patient care, scheduling, billing, claims processing and other related operations for medical practices streamlines office workflow and doctors and their staff are able to spend more time with patients and less time on administrative tasks.

1.1 Background of the Study

Kenya's health care system is structured in a step-wise manner so that complicated cases are referred to a higher level. Gaps in the system are filled by private and church run units. The structure thus consists of Dispensaries and private clinics, Health centers, Sub-district hospitals and nursing homes, District hospitals and private hospitals, Provincial hospitals and National hospitals. All government health centers have a clinical officer as the in-charge and provide comprehensive primary care.

The scope of services offered at Maasai Mara University Health Center is registered as **"Maasai Mara University Health Centre"** with the **Medical Practioners and Dentists Board,** affiliate sections (**Pharmacy and Laboratory**) registered with relevant Boards. Its main function is to provide quality health care through curative, preventive, supportive and psychological care to students' fraternity, staff and their dependants, disciplined forces and civil servants through **National Hospital Insurance Fund** policy as an income generating facility.

The health Centre has the following departments:

- Administration block where patients register and all correspondence and resources are managed.
- Out-patient consultation rooms where patients are seen and examined by clinical officers.
- In-patient ward (observation room) where very sick patients can be admitted.
- Laboratory where diagnostic tests are done. Pharmacy
- Maternal and child health

1.2 Problem Statement

There is a major crisis in managing health records in Maasai Mara University health Center using the traditional manual system. It is difficult to categorize and shelf patient's file. Even after successfully shelving those hundreds of these files have to be processed and retrieved once they are required. The space required to store these files is huge indeed. Not to mention the time required to sort and retrieve them. A case of multiple entire of case files for patients whose files cannot be found is inevitable.

There is a need to organize patient records to ensure that patients undergo proper treatment and to lessen the workload of fetching a single file amidst hundreds of files. The patients' information also needs to be secure and available for future reference.

1.3 Objectives of the Study

1.3.1 Main Objectives

To develop an operational Information Management System for Maasai Mara University Health Center

1.3.2 Specific Objectives

- To provide a quick and efficient means of organizing records.
- To ensure that confidentiality of medical records is maintained.
- To improve reliability of medical records by standardization of input data.
- To generate medical and administrative reports

1.5 Scope of the Study

The system will be designed for use in Maasai Mara University Health Unit. The system will have the following features:

1. Provide Patient and Staff registration.
2. Provide instant access by the entire health care team to all the important and necessary information about a patient in real time on authorization and authentication.
3. Monitor stock and drugs.
4. Present data and reports in an organized manner.

1.7 Significance of the Study

The system will have a positive impact on how business is conducted in a university health unit in the following ways.

i. In an emergency, a clinician will have immediate access to a patient's complete medical history.
ii. During a visit, a clinician can schedule and order the tests needed by the patient directly through the system and access test results immediately.
iii. A clinician can send prescriptions directly to the pharmacy, where a patient picks them up as they leave.
iv. The system will overcome the problem associated with retrieving patient records amidst a myriad of many other such records. It will save on space and create a sense of organization.
v. The pharmacy manual inventory system will also be greatly improved. Current Drug stock can be automatically calculated from processed prescriptions.
vi. The senior medical officer can generate medical records and make informed decisions through the same.

1.8 Definition of Operational Terms

Health Care: Efforts made to maintain or restore health especially by trained and licensed professionals

Health Unit: One of a network of clinics staffed by a group of general practitioners and nurses providing healthcare services to people in a certain area. Typical services covered are family practice and dental care, but some clinics have expanded greatly and can include internal medicine, pediatric, women's care, family planning, pharmacy, optometry, lab, and more.

Medical Record: The systematic documentation of a single patient's medical history and care across time within one particular health care provider's jurisdiction. The term is used for both the physical folder that exists for each individual patient and for the body of information found therein.

Medical Practice Management Software (**PMS**): A category of healthcare software that deals with the day-to-day operations of a medical practice. Such software frequently allows users to capture patient demographics, schedule appointments, maintain lists of insurance payers, perform billing tasks, and generate reports.

Electronic Health Record (**EHR**): or **Electronic medical record** (**EMR**): Refers to the systematized collection of patient and population electronically-stored health information in a digital format.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

In this section the research, location and analysis of the existing knowledge related to the subject of inquiry are explored and cited.

2.1 Information systems

2.1.1 Definitions of Information Systems

According to Encyclopedia Britannica (britannica.com), an Information system is an integrated set of components for collecting, storing, and processing data and for delivering information, knowledge, and digital products.

From Wikipedia, an information system is any organized system for the collection, organization, storage and communication of information. More specifically, it is the study of complementary networks of hardware and software that people and organizations use to collect, filters, and process, create and distribute data.

A computer Information System (IS) is a system composed of people and computers that processes or interprets information. The term is also sometimes used in more restricted senses to refer to only the so (information-system) software used to run a computerized database or to refer to only a computer system.

Information system is an academic study of systems with a specific reference to information and the complementary networks of hardware and software that people and organizations use to collect, filter, process, create and also distribute data. An emphasis is placed on an Information System having a definitive Boundary, Users, Processors, Stores, Inputs, Outputs and the aforementioned communication networks.

2.1.2 Components of Information Systems

The six components that must come together in order to produce an information system is:

i. **Hardware**

The term hardware refers to machinery. This category includes the computer itself, which is often referred to as the central processing unit (CPU), and all of its support equipment. Among the support equipment are input and output devices, storage devices and communications devices.

ii. **Software**

The term software refers to computer programs and the manuals (if any) that support them. Computer programs are machine-readable instructions that direct the circuitry within the hardware parts of the system to function in ways that produce useful information from data. Programs are generally stored on some input / output medium, often a disk or tape.

iii. **Data**

Data are facts that are used by programs to produce useful information. Like programs, data are generally stored in machine-readable form on disk or tape until the computer needs them.

iv. **Procedures**

Procedures are the policies that govern the operation of a computer system. "Procedures are to people what software is to hardware" is a common analogy that is used to illustrate the role of procedures in a system.

v. **People**

Every system needs people if it is to be useful. Often the most over-looked element of the system is the people, probably the component that most influence the success or failure of information systems. This includes "not only the users, but those who operate and service the computers, those who maintain the data, and those who support the network of computers." (Kroenke, D. M. (2015). MIS Essentials. Pearson Education)

vi. Feedback

It is another component of the IS, that defines that an IS may be provided with a feedback, although this component isn't necessary to function.

2.1.3 Types of Information Systems

Transaction Processing Systems

Transaction processing systems automate the handling of transactions, which are individual simple events in the life of an organization.

When a transaction processing system processes an organization's transactions, each transaction is available for recall later. More importantly to the organization, the number and volume of transactions can be calculated for a given time period. Transactions also provide the official record of business activities, which drive other systems such as those which bill customers, pay vendors and employees, and reorder inventory or raw materials or stocked goods.

Management Information Systems

Management information systems (MIS) are designed to take the relatively raw data available through a TPS and convert them into a summarized and aggregated form for managers, usually in a report format. Several types of reports can be produced. Summary reports present all activity over a given time period, geographic region, or other categorization in aggregate form. Exception reports only present information that is out of normal ranges. On-demand reports present anticipated summaries only when a manager wants or needs to check the status of activities. The precise contents of on-demand reports may change depending on the manager's immediate need. Ad hoc reports provide specific information as needed, the contents of which may change depending on the manager's needs. Ad hoc reports are unanticipated and may be one-time in nature.

Decision Support Systems

Decision support systems (DSS) are designed to help organizational decision makers make decisions. DSS usually have three major components: a database, a model base, and a dialogue module (Sprague, 1980).

- The database contains data relevant to the decision to be made.
- The model base contains one or more models that can be used to analyze the decision situation.
- The dialogue module provides a way for the decision maker, usually a non-technical manager, to communicate with the DSS.

Expert Systems

Expert systems (ES) attempt to codify and manipulate knowledge rather than information. Typically users communicate with an ES through a dialogue during which the ES asks questions and the user supplies the answers. The answers are then used to determine which rules apply and the ES finishes with a recommendation based on its rules. One of the most difficult parts in building an ES is acquiring the knowledge of the expert in the particular problem domain.

Office Automation Systems

Office automation systems are usually quite basic and include such tools as word processing and accounting information systems. Integrated office systems that include electronic mail, calendaring features, and reminder files in addition to word processing are also available. Electronic mail (e-mail) allows office workers to send each other messages and files directly from their computers and is usually more convenient than trying to reach someone by telephone. Calendaring features allow office workers to coordinate their schedules, to reserve conference rooms, and to schedule meetings. Reminder files provide a means for conveniently reminding ourselves of meetings and other commitments. Office systems are rarely if ever developed in-house, but instead are purchased or leased from hardware or software producers.

2.2 Medical practice management software (PMS)

2.2.1 Definition of PMS

According to Wikipedia a PMS is a category of healthcare software that deals with the day-to-day operations of medical practice. Such software frequently allows users to capture patient demographics, schedule appointments, maintain lists of insurance payers, perform billing tasks, and generate reports. The desktop-only variety is intended to be used only on one computer by one or a handful of users sharing access. Client-server software typically necessitates that the practice acquire or lease server equipment and operate the server software on that hardware, while individual users' workstations contain client software that accesses the server. Client-server software's advantage is in allowing multiple users to share the data and the workload; a major disadvantage is the cost of running the server. Internet-based software is a relatively newer breed of PMS. Such software decreases the need for the practice to run their own server and worry about security and reliability. However, such software removes patient data from the practice's premises, which can be seen as a security risk of its own.

Medical Practice Management software automates activities such as patient care, scheduling, billing, claims processing and other related operations for medical practices. By streamlining office workflow, doctors and their staff are able to spend more time with patients and less time on administrative tasks. Medical Practice Management applications can be utilized by private practices, community health centers, long term care and behavioral health facilities, hospitals and other health care organizations.

2.2.2 Components of Practice Management Software

Most practice management software contains systems that allow users to enter and track patients, schedule and track patient appointments, send out insurance claims and patient statements as part of the collection process, process insurance, patient and third party payments, and generate reports for the administrative and clinical staff of the practice. Typically, using a PMS also involves keeping up to date large sets of data including lists of diagnosis and procedures, lists of insurance companies, referring physicians, providers, facilities, and much more.

Patient demographics

The capture of patient demographics often starts when a new patient fills out a patient information chart. This information includes the patient's name, address and contact information, birthdate, employer, and insurance information. Practice staff typically enters this information into the software. The software may automatically verify the patient's eligibility for receiving benefits with the insurance company using a standard electronic data interchange connection.

Appointment scheduling

Practice Management systems often include a calendaring or scheduling component that allows staff to create and track upcoming patient visits. Software is often differentiated by whether it allows double-booking, or whether it uses a scheduling or a booking model. Schedules are often color-coded to allow healthcare providers (i.e. doctors, nurses, assistants) to easily identify blocks of time or sets of patients.

Claims and statements

If the patient carried a valid private or public insurance policy at the time these services were provided, the charges are then sent out as an insurance claim. The process of sending charges may happen on paper. In most cases, electronic claims are submitted using an automated software process.

In cases where a patient did not have proper insurance, or where insurance coverage did not fully pay the charges, the practice will usually send out patient statements. Practice management software often contains a way for a practice to print and mail their own statements (or other correspondence), and may even contain a way to interface to third-party patient statement printing companies.

2.3 'Topographic Map' of a Health Information System

A health information system can be divided into two major parts:

i. a clinical part
ii. An administrative/statistical part.

The difference between these two is the way the data are used. In clinical work they are linked to a real patient either via a name or a unique personal identifier. Decisions concerning this patient are made based on this information; thus there is a high demand for data accuracy and correctness. In the administrative/statistical part of the system, data are separated from the patient and are no longer used in decision making concerning individual patients. Thus the demand for absolute correctness in each individual case is not so high. Instead, the usefulness of the data at this level is highly dependent on definitions, comparable use of codes etc. The data from an administrative/statistical system may not in any case be traced back to the patient again. This principle is also the key to the data security of an HIS.

The clinical part of an HIS is composed of two elements:

i. the clinical databases that keep the clinical data in order and are always linked with the correct patient
ii. an evidence-based decision support and expertise system that provides the clinician with the latest critically reviewed scientific information

The administrative/statistical part of the system has three elements

i. a business reporting system that provides information on the production and use of services for benchmarking purposes and also for management at local, regional and national levels – this has traditionally been called 'healthcare statistics'
ii. an epidemiological information system that contains information on the incidence and prevalence etc. of diseases or health conditions, living habits and health hazards and that also serves the purposes of disease surveillance and early warning mechanism
iii. A quality system that ensures the high quality of services provided to the population.

2.4 Patient Record Keeping in Health Care Units

Health care practitioners must always be very careful before engaging in record keeping of the patient because many small mistakes that could be detrimental could occur. A health record represents the hospital's responsibility in the life of a patient from which health reports, research and statistical reports could be drawn. 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 record care also ensures that only the information necessary to the patients' health is recorded and that the storage area is freed from unnecessary and unambiguous information. This saves time, resources and the personnel can access the information quite easily.

2.4.1 Types of Records Required in Health Care Units

Most patients who go to a clinic first visit the out-patient department. Here, they first encounter the hospital nurses who give them a basic hospital form, locally referred to as the hospital card. In this form/file is written the patients' name, age, sex, marital status and the location of residence). Note that in practice, not all of these records are necessarily similar to other hospitals. However, the same form can be used during another visit to a different hospital and is usable until it is completely full.

The patient then goes ahead to tell the nurse all the signs and symptoms of the illness and this information is recorded in a totally different file. This file remains in the hospital and will be assigned to patients on the very first visit at the hospital. In Kenya, one can visit any health care center and hospital in the country, regardless of his location of residence.

The nurse could also write down case notes. After the dialogue, the nurse assesses the situation and could either administer treatment herself or advance the patient to the doctor in charge.

The doctor will then administer treatment and record it in the patients file. Prescriptions for medication will be written by hand in the patient's hospital card. When discharged from the hospital, the patient takes this card to the pharmacy to buy the medication. If the patient is referred to the in-patient ward, the medication comes from the hospital's store and is included in the patient's final bill.

2.4.1.1 Pharmacy and Drug Records

The prescription and supply of drugs generates a variety of records, including pharmacy stocks, ordering and dispensing records, requests for drugs from wards and departments, drug administration records and prescriptions for individual patients. The receipt and issue of all drugs must be recorded. The pharmacy and the ward must also keep a copy of all the records of drugs so that cross-referencing and uniform management can be documented well.

Information about dangerous or 'controlled' drugs especially the strong kind of painkillers as well as medication used in the operating rooms, such as Propanol must be recorded in particular detail in both the hospital pharmacy and in individual wards and departments, in order to avoid drug abuse among members of the staff as well as the patients. Such medication is rarely administered to patients while they are not under medical personnel supervision, like in a hospital ward.

2.4.1.2 Patient Case notes

Case notes are little pieces of information that most members of the health care system will use to write important bits of information before finally writing this in the patient's medical record. They could include basic information such as the signs and symptoms, diagnostic test results and temperature blood pressure. They could also include complex information such as patient histories and allergies, as well as records of operations and other forms of treatment. At the regular hospital, all the case notes, doctor's notes, nurse's review and other hospital records available from other health care centers are kept together in one file bearing the patient's name and other personal details. With time, all these documentations form a complete medical history of the patient.

2.5 Workflow of Maasai Mara University Health Centre

From the Maasai Mara university health unit procedure manual, the workflow of the health center can be outlined as follows:

A patient visiting the center for the first is registered if they are not students. Students are registered as patients once they are bona fide students of the university; otherwise their medical file is retrieved. A patient can be treated as an inpatient and given a prescription or they can be allocated a bed as they are treated in the observation room. Alternatively, they may be referred to another facility (outpatient) and the resulting invoices/billings generated.

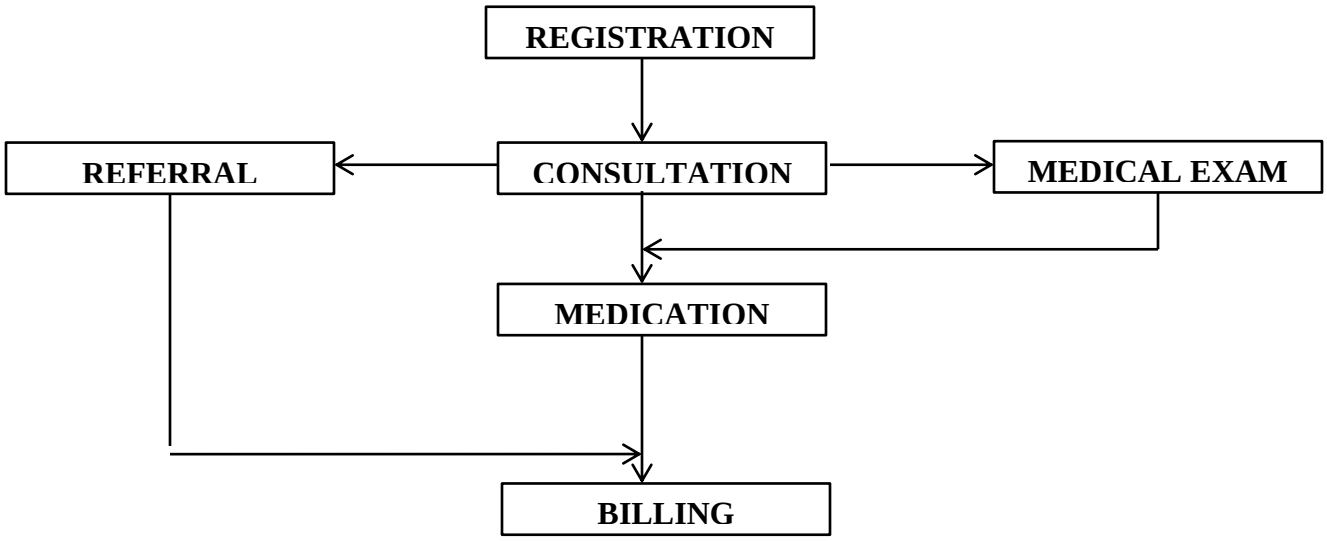

Figure 1: Work Flow of Maasai Mara University Health Unit

2.5.1 Advantages of a computer- based information system.

According to Gordon (2006), the following are what he identified as the advantages of computer- based information system.

i. They are user friendly and the navigation is very easy.
ii. They help in organizing and managing documents effectively.
iii. Since the data is stored in a highly organized manner, accessing necessary data is very easy. It helps save time.

iv. People are able to access data needed in real time thus enabling them access detailed information.

v. Accurate, current and reliable data is provided.

vi. As data can be analyzed correctly and plans made can be implemented at astounding speed due to proper automated systems.

vii. They are installed to improve internal efficiency of the organization.

viii. They increase security and protect the data from being misused.

ix. They are extremely useful, especially during disaster recovery, as paper documents can be lost, causing a business millions of dollars in losses.

2.5.2 Weakness of a computer- based information system.

Hackers: information sent by use of the internet can easily be hijacked and terminated by unauthorized persons before reaching its destination.

Viruses: this can destroy files by replicating themselves in the document hence losing the meaning of the file.

2.6 System Development Methodologies

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.

2.6.1 Waterfall

The Project is divided into sequential phases, with some overlap and splash back acceptable between phases. Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time. Tight control is maintained over the life of the project through the use of extensive written documentation, as well as through formal reviews and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase.

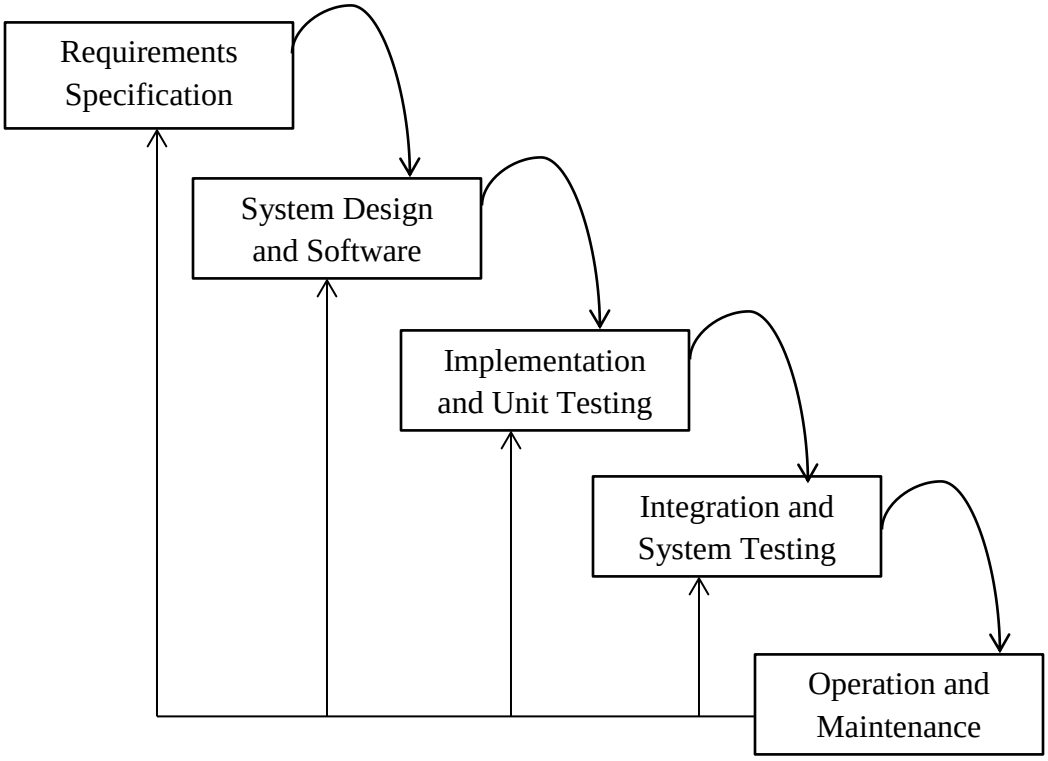

Figure 3: Waterfall Model

Situations where most appropriate:

1. Project is for development of a mainframe-based or transaction-oriented batch system.
2. Project is large, expensive, and complicated.
3. Project has clear objectives and solution.
4. Pressure does not exist for immediate implementation.
5. Project requirements can be stated unambiguously and comprehensively.
6. Project requirements are stable or unchanging during the system development lifecycle.
7. User community is fully knowledgeable in the business and application.
8. Team members may be inexperienced.
9. Team composition is unstable and expected to fluctuate.
10. Project manager may not be fully experienced.

2.6.2 Prototyping

It is not a standalone, complete development methodology, but rather an approach to handling selected portions of a larger, more traditional development methodology (i.e., Incremental, Spiral, or Rapid Application Development (RAD)). It attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process. The user is involved throughout the process, which increases the likelihood of user acceptance of the final implementation. Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users' requirements. While most prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system. A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problem.

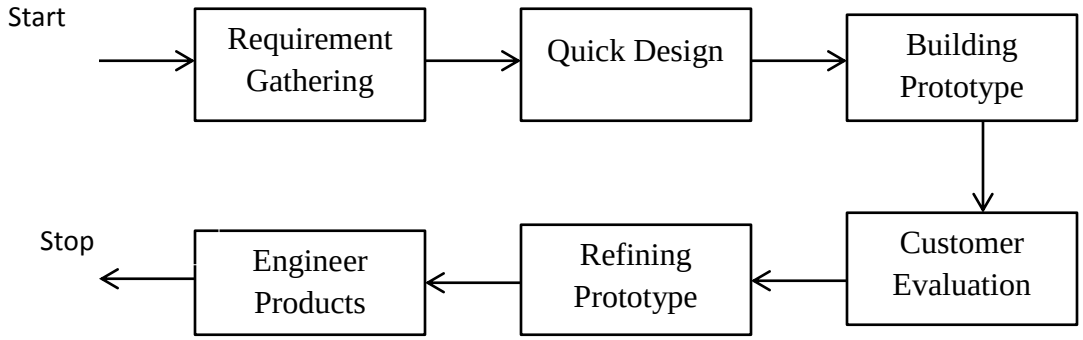

Figure 4: Prototyping Model

Situations where most appropriate:

1. Project is for development of an online system requiring extensive user dialog, or for a less well-defined expert and decision support system.
2. Project is large with many users, interrelationships, and functions, where project risk relating to requirements definition needs to be reduced.
3. Project objectives are unclear.
4. Pressure exists for immediate implementation of something.
5. Functional requirements may change frequently and significantly.

6. User is not fully knowledgeable.
7. Team members are experienced (particularly if the prototype is not a throw-away).
8. Team composition is stable.
9. Project manager is experienced.
10. No need exists to absolutely minimize resource consumption.
11. No strict requirement exists for approvals at designated milestones.
12. Analysts/users appreciate the business problems involved, before they begin the project.
13. Innovative, flexible designs that will accommodate future changes are not critical.

2.6.3 Incremental

A series of mini-Waterfalls are performed, where all phases of the Waterfall development model are completed for a small part of the system, before proceeding to the next increment. Alternatively overall requirements are defined before proceeding to evolutionary, mini Waterfall development of individual increments of the system. Again the initial software concept, requirements analysis, and design of architecture and system core are defined using the Waterfall approach, followed by iterative Prototyping, which culminates in installation of the final prototype (i.e., working system).

Situations where most appropriate:

1. Large projects where requirements are not well understood or are changing due to external changes, changing expectations, budget changes or rapidly changing technology.
2. Web Information Systems (WIS) and event-driven systems.
3. Leading-edge applications.

2.6.4 Spiral

Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle. "Each cycle involves a progression through the same sequence of steps, for each portion of the product and for each of its levels of elaboration, from an overall concept-of- operation document down to the coding of each individual program." (Boehm, 1986)

Each trip around the spiral traverses four basic quadrants: (1) determine objectives, alternatives, and constraints of the iteration; (2) evaluate alternatives; identify and resolve risks; (3) develop and verify deliverables from the iteration; and (4) plan the next iteration. (Boehm, 1986 and 1988). Each begins cycle with an identification of stakeholders and their win conditions, and end each cycle with review and commitment. (Boehm, 2000).

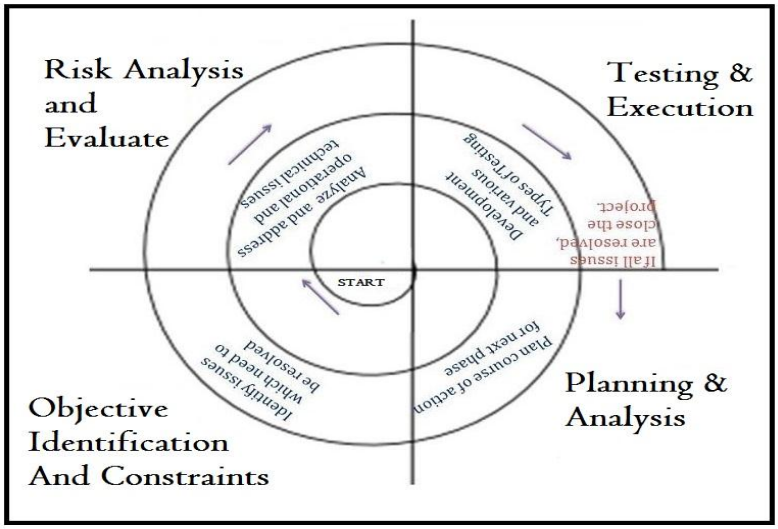

Figure 5: Spiral Model

Situations where most appropriate:

1. Real-time or safety-critical systems.
2. Risk avoidance is a high priority.
3. Minimizing resource consumption is not an absolute priority.
4. Project manager is highly skilled and experienced.
5. Requirement exists for strong approval and documentation control.
6. Project might benefit from a mix of other development methodologies.
7. A high degree of accuracy is essential.
8. Implementation has priority over functionality, which can be added in later versions. .

2.6.5 Rapid Application Development (RAD)

The key objective is for fast development and delivery of a high quality system at a relatively low investment cost. It attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

It aims to produce high quality systems quickly, primarily through the use of iterative Prototyping (at any stage of development), active user involvement, and computerized development tools. These tools may include Graphical User Interface (GUI) builders, Computer Aided Software Engineering (CASE) tools, Database Management Systems (DBMS), fourth-generation programming languages, code generators, and object-oriented techniques.

Key emphasis is on fulfilling the business need, while technological or engineering excellence is of lesser importance. Project control involves prioritizing development and defining delivery deadlines or "timeboxes". If the project starts to slip, emphasis is on reducing requirements to fit the time box, not in increasing the deadline. Generally includes Joint Application Development (JAD), where users are intensely involved in system design, either through consensus building in structured workshops, or through electronically facilitated interaction.

Situations where most appropriate:

1. Project is of small-to-medium scale and of short duration (no more than 6 man-years of development effort).
2. Project scope is focused, such that the business objectives are well defined and narrow.
3. Application is highly interactive, has a clearly defined user group, and is not computationally complex.
4. Functionality of the system is clearly visible at the user interface.
5. Users possess detailed knowledge of the application area.
6. Senior management commitment exists to ensure end-user involvement.
7. Requirements of the system are unknown or uncertain.
8. It is not possible to define requirements accurately ahead of time because the situation is new or the system being employed is highly innovative.
9. Team members are skilled both socially and in terms of business.
10. Team composition is stable; continuity of core development team can be maintained.

11. Effective project control is definitely available.
12. Developers are skilled in the use of advanced tools.
13. Data for the project already exists (completely or in part), and the project largely comprises analysis or reporting of the data.
14. Technical architecture is clearly defined.
15. Key technical components are in place and tested.
16. Technical requirements (e.g., response times, throughput, database sizes, etc.) are reasonable and well within the capabilities of the technology being used. Targeted performance should be less than 70% of the published limits of the technology.
17. Development team is empowered to make design decisions on a day-to-day basis without the need for consultation with their superiors, and decisions can be made by a small number of people who are available and preferably co-located.

2.7 Popular High-Level Programming Languages

Table 1: High Level Programming Languages

Language	Language Description
Ada	Named for Ada Lovelace, who worked on mechanical general-purpose computers. The Ada language was developed for the Department of Defense and is used mainly in defense projects.
BASIC	Beginner's All-purpose Symbolic Instruction Code. It was designed to be learned and used easily by beginners.
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	An object-oriented language, based on C.
C#	It is a hybrid of Java and C++ and was developed by Microsoft
FORTRAN	FORmula TRANslation. Popular for scientific and mathematical applications.
Java	Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.
Pascal	Named for Blaise Pascal, who pioneered calculating machines in the seventeenth century. It is a simple, structured, general-purpose language primarily for teaching programming.
Python	A simple general-purpose scripting language good for writing short programs.
Visual Basic	Developed by Microsoft and it enables the programmers to rapidly develop graphical user interfaces

2.8 Implementation of the Health Centre Management Information System

The Health Centre Management Information System is a system that can manage multiple users of the system and can have the track of the right assigned to them. It makes sure that all the users function with the system as per the rights assigned to them and they can get their work done in efficient manner. It is a Customizable and strong administration system i.e. changes of password of users at the administration point. The information management system will be able to capture information about an old patient the information captured will be easily managed by the administrators more easily.

2.8.1 Problems in Implementing the Health Centre Management Information System

According to Gordon the following are possible problem to be encountered while implementing the system

i. It is not suitable for computer illiterate people.
ii. The user must be a member in order to make use of the system.
iii. The systems do not do away with paper work completely; the papers are still used at some point.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This is a description of methods chosen to achieve the objectives of the system. It will go on to describe the techniques of data collection that will be employed in the research study of the system.

3.1 Approach for the Development

The system development life cycle (SDLC) to be used will be the iterative waterfall model. In this model, the system follows a series of events from the requirement definition, system and software design, implementation and unit testing, integration and system testing and operational maintenance.

This model will be chosen because:

The orderly sequence of development steps and strict controls for ensuring the adequacy of documentation and design reviews will help ensure the quality, reliability, and maintainability of the developed software.

i. Progress of system development is measurable.
ii. It conserves resources.
iii. The project has clear objectives and solution.
iv. Pressure does not exist for immediate implementation.
v. Project requirements can be stated unambiguously and comprehensively.

Different aspects from other models such as prototyping will be used to come up with the system.

3.1.1 Data Collection Methods

Two data collection methods will be used. They are:-

i. Direct observation
ii. Interviewing

Direct observation

Being a bona fide student of the university, the researcher has at many times been involved in most of the clinic's patient-treatment activities. These activities are examination, takin laboratory tests, obtaining prescribed drugs.

From the above the research is able to identify some of the activities that can be incorporated to generate a working clinic management system.

Interviewing

In this method, there will be interaction between the researcher and the Staff. Interviews will be conducted with the medical officer and some potential employees to find out what difficulties they encountered with the existing system.

3.2 Database Design.

A relational database design will be used to design the database. A relational database management system (RDBMS) is an excellent tool for organizing large amount of data and defining the relationship between the datasets in a consistent and understandable way. A RDBMS provides a structure which is flexible enough to accommodate almost any kind of data. Relationships between the tables will be defined by creating special columns (keys), which contain the same set of values in each table. The tables can be joined in different combinations to extract the needed data. A RDBMS also offers the flexibility that enables redesign and regeneration of reports from the database without need to re-enter the data.

The database will be developed using Structured Query Language (SQL) and MYSQL Database Management System.

3.3 Programming Language

The user application programs and interface will be developed using Java. Java has significant advantages over other languages and environments that make it suitable for just about any programming task. One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels. Its other advantages are:

i. Java is easy to learn.

ii. Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

iii. Java is object-oriented.

iv. This allows one to create modular programs and reusable code.

CHAPTER FOUR

SYSTEM ANALYSIS AND REQUIREMENT MODELING

4.0 Introduction

This is an analysis of the existing system and the new system, a description of the system requirements and how they are modelled.

4.1 System Analysis

System Analysis is the process of gathering and interpreting facts, diagnosing problems (if any), using information to recommend improvements to the system. There are four basic elements of system analysis: - Output, Input, Files, processes. For computerization of any system, the existing system must be thoroughly understood to determine "how the computer can be best used to make its operation most effective". This is acquired by analyzing existing system.

4.1.1 Feasibility Study

Feasibility study is the process of determining whether or not a project is worth doing. The contents and recommendations of this feasibility study gave a basis for deciding how to proceed with the project. It helped in making decisions such as which software to use, hardware combinations, etc.

1. Technical Feasibility

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

This application in going to be used in a desktop application in a networked environment. It is also able to work on distributed environment.

A relational database design will be used to design the database. A relational database management system (RDBMS) is an excellent tool for organizing large amount of data and defining the relationship between the datasets in a consistent and understandable way. A

RDBMS provides a structure which is flexible enough to accommodate almost any kind of data. Relationships between the tables will be defined by creating special columns (keys), which contain the same set of values in each table. The tables can be joined in different combinations to extract the needed data. A RDBMS also offers the flexibility that enables redesign and regeneration of reports from the database without need to re-enter the data.

The database will be developed using Structured Query Language (SQL) and MYSQL Database Management System.

The user application programs and interface will be developed using Java. Java has significant advantages over other languages and environments that make it suitable for just about any programming task. One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels. Its other advantages are:

i. Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
ii. Java is object-oriented.
iii. This allows one to create modular programs and reusable code.

2. **Economic Feasibility**

Economic feasibility determines whether there are sufficient benefits in creating the system and whether the costs are acceptable.

The system is feasible and is economical regarding its pre-assumed cost for making a system. The costs to be incurred will be derived from the cost of purchasing hardware components and software resources. This is a onetime cost.

3. Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period.

The schedule for the system is as shown below.

Table 2: Project Schedule

Activities		Months						
	February	February	February	March	March	April	April	May
Project Proposal								
Requirement Engineering								
System design								
System development								
System Testing								
System Deployment								

4. Operational Feasibility

Operational feasibility criteria measure the urgency of the problem (survey and study phases) or the acceptability of a solution (selection, acquisition and design phases). The systems front end is developed using GUI. So it is easy for users to enter the necessary information. Users need to have knowledge on basic computer operation.

4.2 Existing System

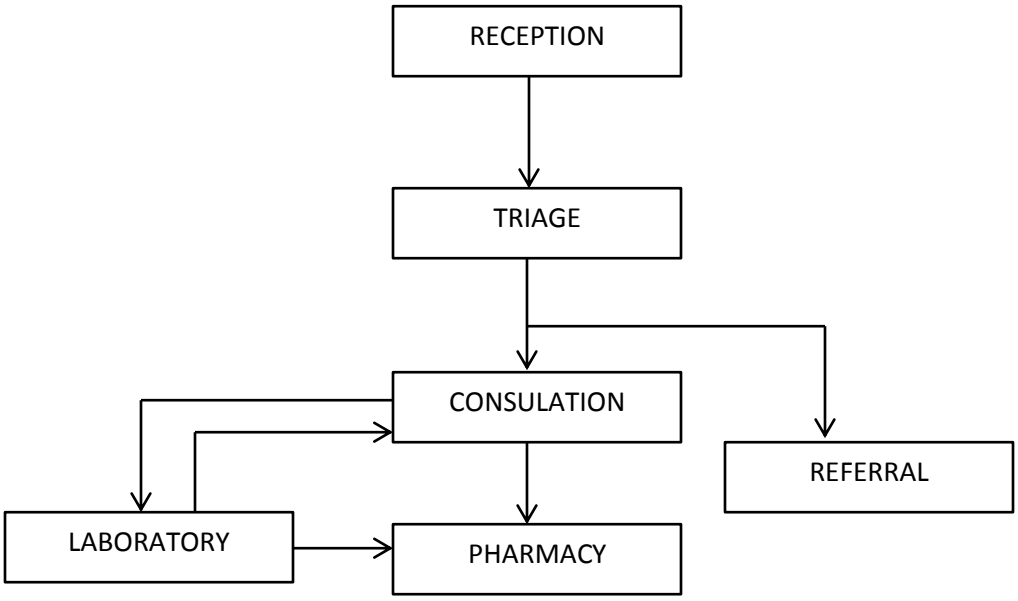

Figure 6: Existing System

Currently, the work flow of the University Health Unit is as shown in the figure above. A patient visits the health unit. If the patient is not registered, their details are taken and the patient is registered. There are three kinds of patients-students, staff members of the university and people from the local community who are registered under NHIF. Otherwise the patient's file is retrieved at the reception unit where all patient files are kept.

They then proceed to have their vital signs recorded by a nurse. From here, they proceed to the consultation room where they are examined. This examination determines whether the patient will be treated, sent to the laboratory to take tests or whether they will be referred to another health institution. Referral is given in instances where the services required by the patient are not available in the health unit.

If no tests are required the patient is prescribed drugs which they pick on their way out, otherwise the doctor notes down the tests to be taken and the patient proceeds to the laboratory. Here, the tests are taken. The laboratory technician notes down the test results and any other

observations and the patient goes back to the consultation room where the doctor prescribes drugs.

A prescription note is given to the patient. With this note the pharmacist can prescribe drugs to the patient. There are instances where the drug prescribed by the clinician is not in stock. In such a situation, the pharmacist will prescribe a different drug or ask the patient to purchase it from a local pharmacy.

Limitations of Existing System

- ***It is tedious to retrieve and maintain records.***

  There are three categories of patients-students who are in session at different periods of an academic year, staff members and people from the local community. All these files have to be classified and maintained. It is tedious to retrieve and to maintain them. A lot of time is consumed when retrieving them leading to a delay in the flow of services offered.

- ***It is difficult to compile medical and administrative report.***

If data was required for epidemiology and/or clinical studies such data would not be available and even if it was available it would not be complete and organized. Such data would require a manual processing of patient files and/or record books. In such cases data redundancy cannot be avoided.

- ***No security is afforded on medical records***

Medical records are confidential. A breach of confidentiality may attract legal charges. There are no security measures that ensure that the confidentiality of patients' files is maintained. A records officer for example is able to retrieve a patients' file and view the contents. In the event of a natural disaster such as a fire or floods, there is no guarantee that records will be protected from damage.

4.3 Proposed System

The system computerizes all information and data related services of the health unit. All records are stored in a database. Views are used to achieve security and data independence. A user will only see what they are required to see. A nurse for example cannot tell the drugs prescribed to a patient or the drugs in stock.

The system will have a positive impact on how business is conducted in the University Health Unit in the following ways.

vii. In an emergency, a clinician will have immediate access to a patient's complete medical history.

viii. During a visit, a clinician can schedule and order the tests needed by the patient directly through the system and access test results immediately.

ix. A clinician can send prescriptions directly to the pharmacy, where a patient picks them up as they leave.

x. The system will overcome the problem associated with retrieving patient records amidst a myriad of many other such records. It will save on space and create a sense of organization.

xi. The pharmacy manual inventory system will also be greatly improved. Current Drug stock can be automatically calculated from processed prescriptions.

xii. It will also be easier to compute patient bills.

xiii. The senior medical officer can generate medical records and make informed decisions through the same.

Limitations of the system

- Requires expensive resources-both hardware and software
- Basic computer knowledge is a prerequisite for system users

4.4 Requirements Analysis

Functional Requirements

Generally the functional requirements of the system give provisions for:

- A user to login into the system.
- A records officer to search and/or add a new patient
- A nurse to take and record vital signs of a patient
- A clinician to examine a patient, record examination results, request a test, schedule a referral and/or prescribe drugs
- A pharmacist to process and record prescriptions and to take a stock inventory of available drugs and add more stock. They can print a note of prescription to be purchased from another pharmacy.
- A laboratory technician to take and record test results and to take a stock inventory of available equipment and add more stock. They can print test results
- The head clinician to view and print medical reports.

Non-Functional Requirements

These include:

- Security-User authentication on login.
- The system will have consistent interface formats and button sets for all forms in the application.
- Drug and stock available within the health unit.
- Referral centers recommended for the health unit.

4.5 Data Flow Diagram of Proposed System

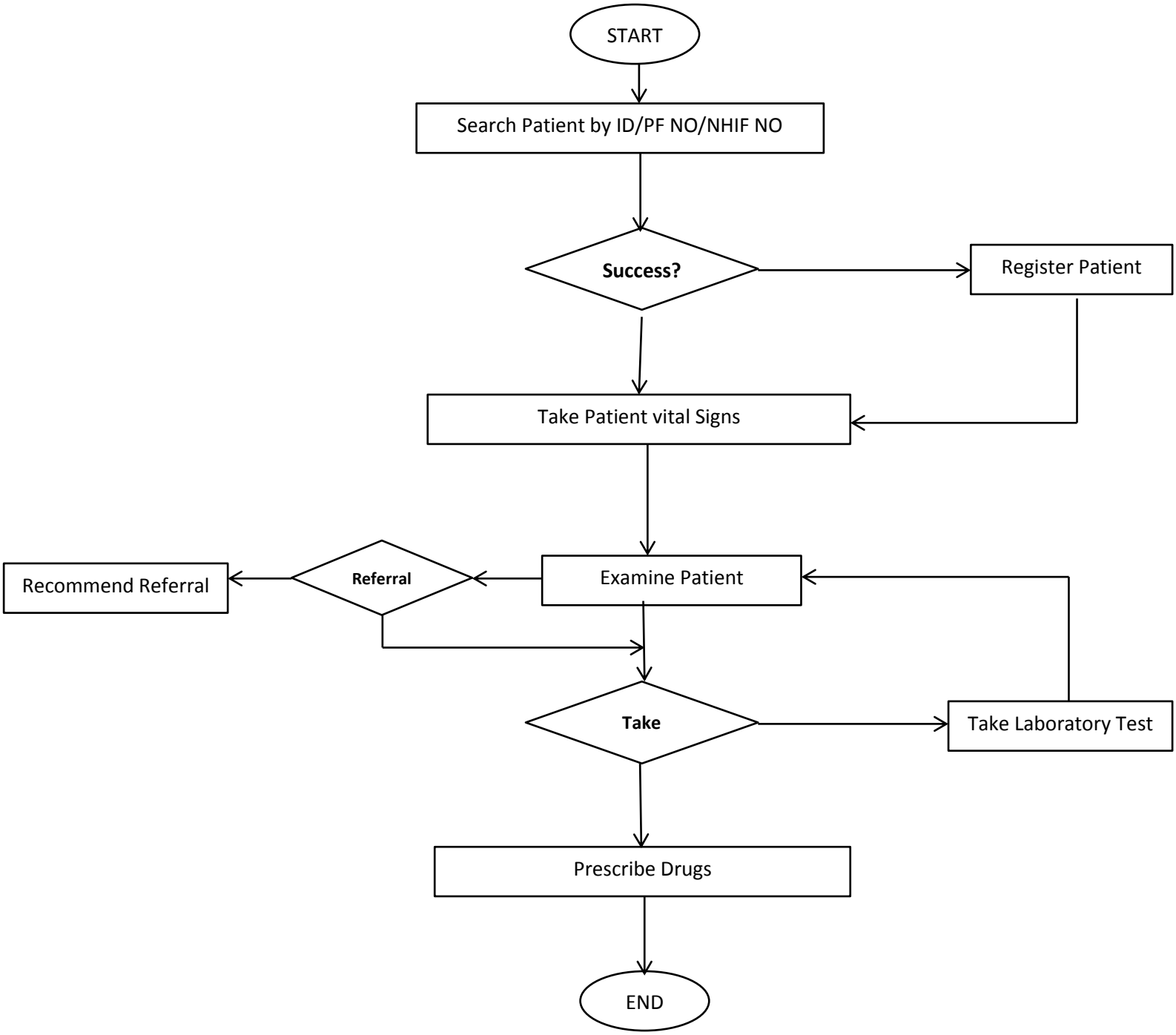

Figure 7: Data Flow Diagram of Proposed System

4.6 Context Diagram of Proposed System

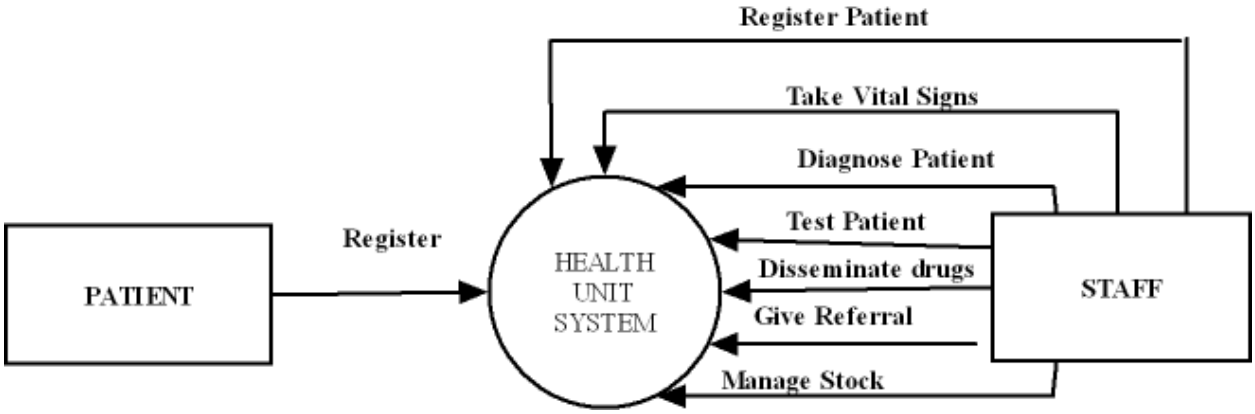

Figure 8: Context Diagram of Proposed System

4.7 Entity Relationships Diagram of Proposed System

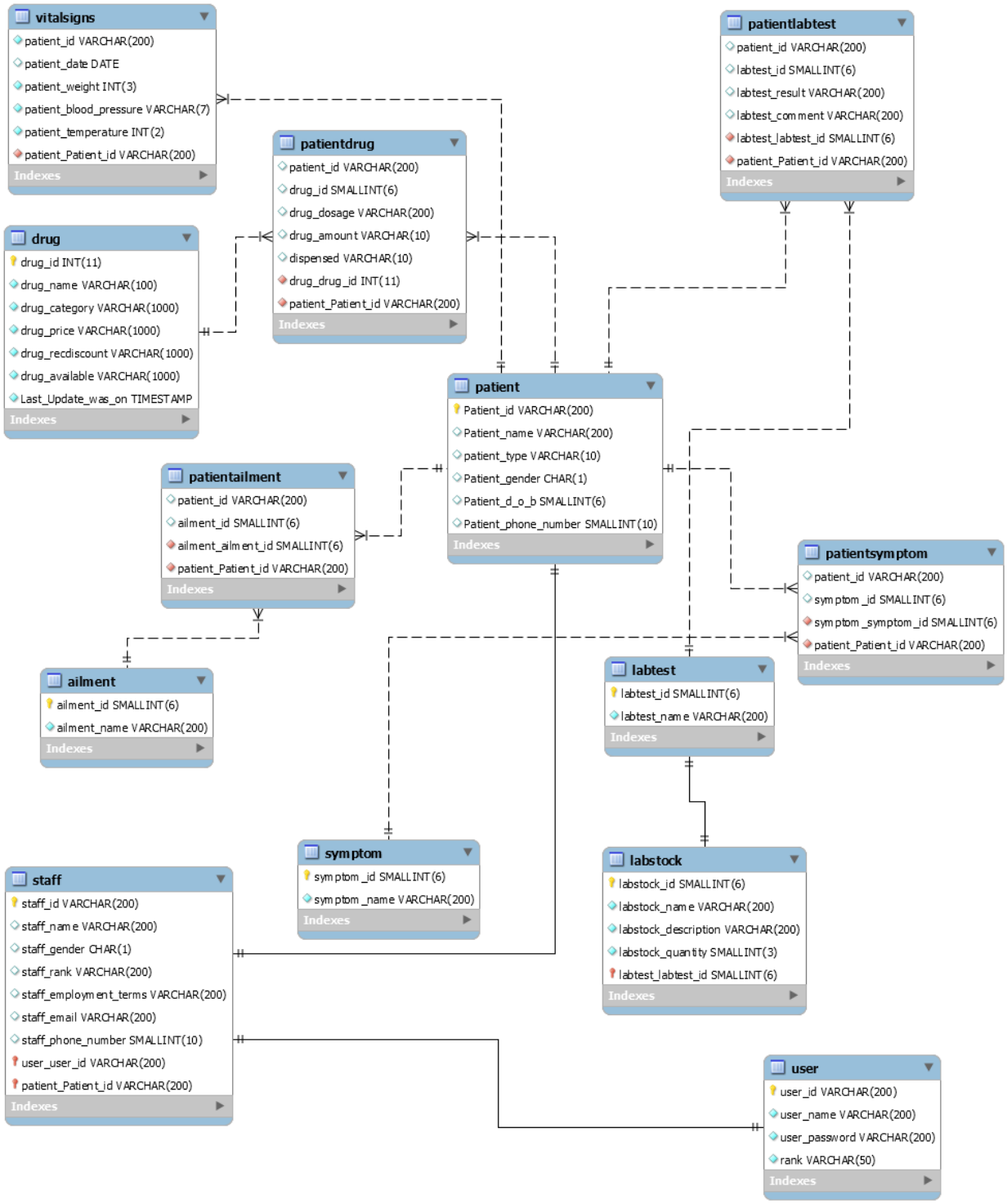

Figure 9: Entity Relationships Diagram of Proposed System

CHAPTER FIVE

SYSTEM DESIGN

5.1 Design specification

Design of software involves conceiving, planning out and specifying the externally observable characteristics of the software product. Data design, architectural design and user interface design are explained in the following section. The goals of the design process are to provide a blue print for implementation, testing, and maintenance activities.

5.2 Data design

The primary activity during data design is to select logical representations of data objects identified during requirement analysis and software analysis. A data dictionary explains the elements of the data structure. A data dictionary should be established and used to define both data and program design.

5.3 Design methodology

The two basic modern design strategies employed in software design are;

1. Top down design approach

2. Bottom up design approach.

Top down Design is basically a decomposition process, which focuses on the flow of control. At later stages it concern itself with the code production. The first step is to study the overall aspects of the tasks at hand and to break it into a number of independent modules. The second step is to break each one of these modules further into independent sub-modules. The process is repeated to obtain modules, which are small enough to group mentally and to code in a straightforward manner. .Only the necessary data and control that must be called back and forth over the interface are defined

In a bottom-up design one first identifies and investigates parts of design that are most difficult and necessary designed decision are made and the reminder of the design is tailored to fit around the design already chose for crucial part. It vaguely represents a synthesis process explained in previous section. One storage point of the top-down method is that it postpones details of the decision until the last stage of the decision. It allows making small design changes when the design is half way through. There is danger that the specifications will be incompatible and this will not be discovered until late in the design process. By contrast the bottom-up strategy first focuses on the crucial part so that feasibility of the design is tested at early stage. In mixing top-down and bottom-up design it often appears that we start in the middle of the problem and work our way both up and down there. In a complex problem, it is often difficult to decide how to modularize the various procedures in such cases one might consider a list of system inputs and decide what functions are necessary to process these inputs. This is called back to front design. Similarly one can start with the required outputs and work backwards evolving so called front-back design. I have applied both the top down and bottom up approach in our design approach.

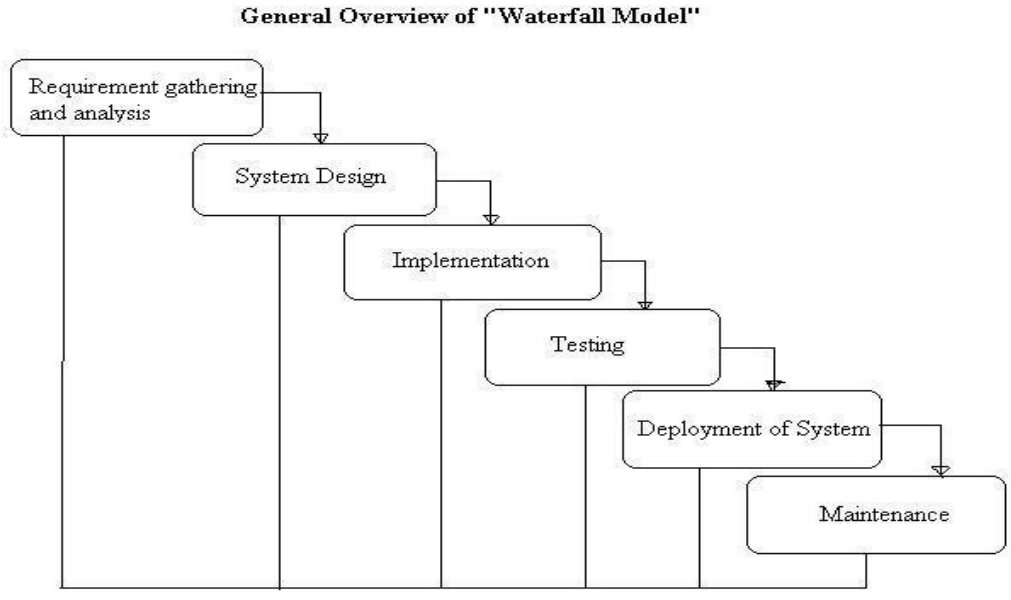

Figure 10: Waterfall Model

5.4 Interface Design

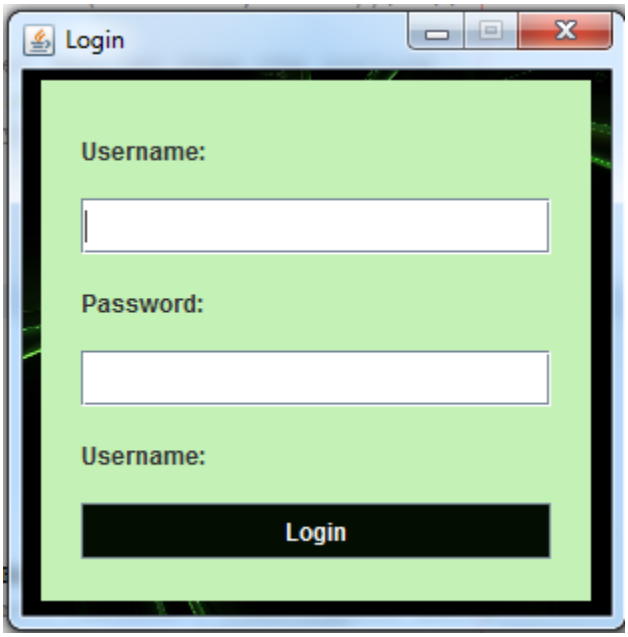

Figure 11: Log in Form

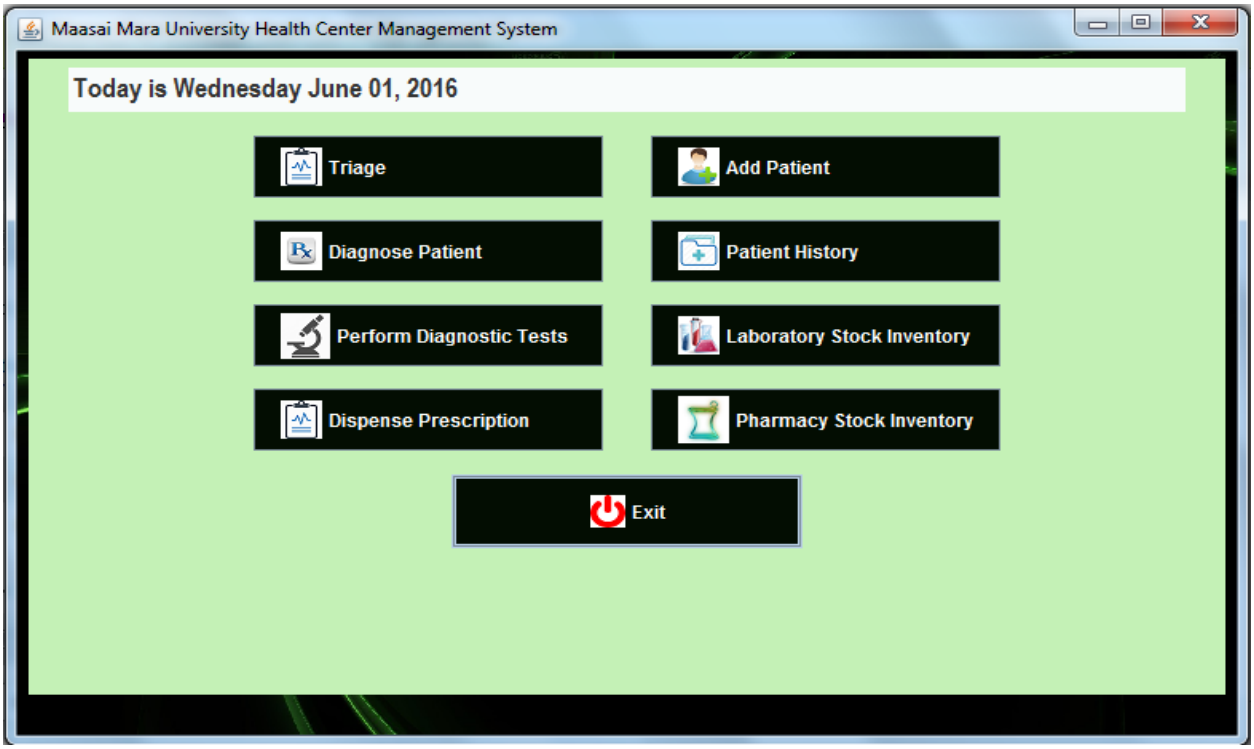

Figure 12: Main Menu

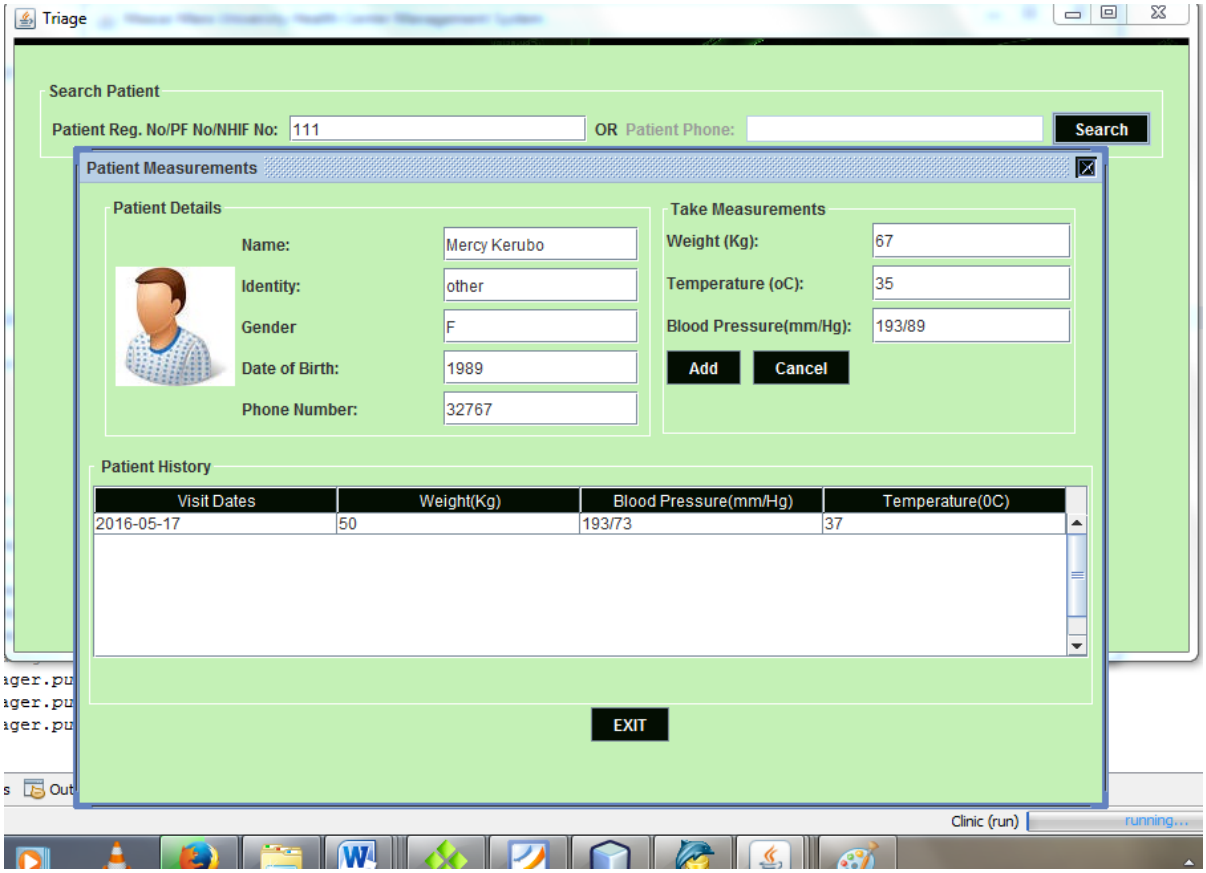

Figure 13: Get Patient Vital Signs

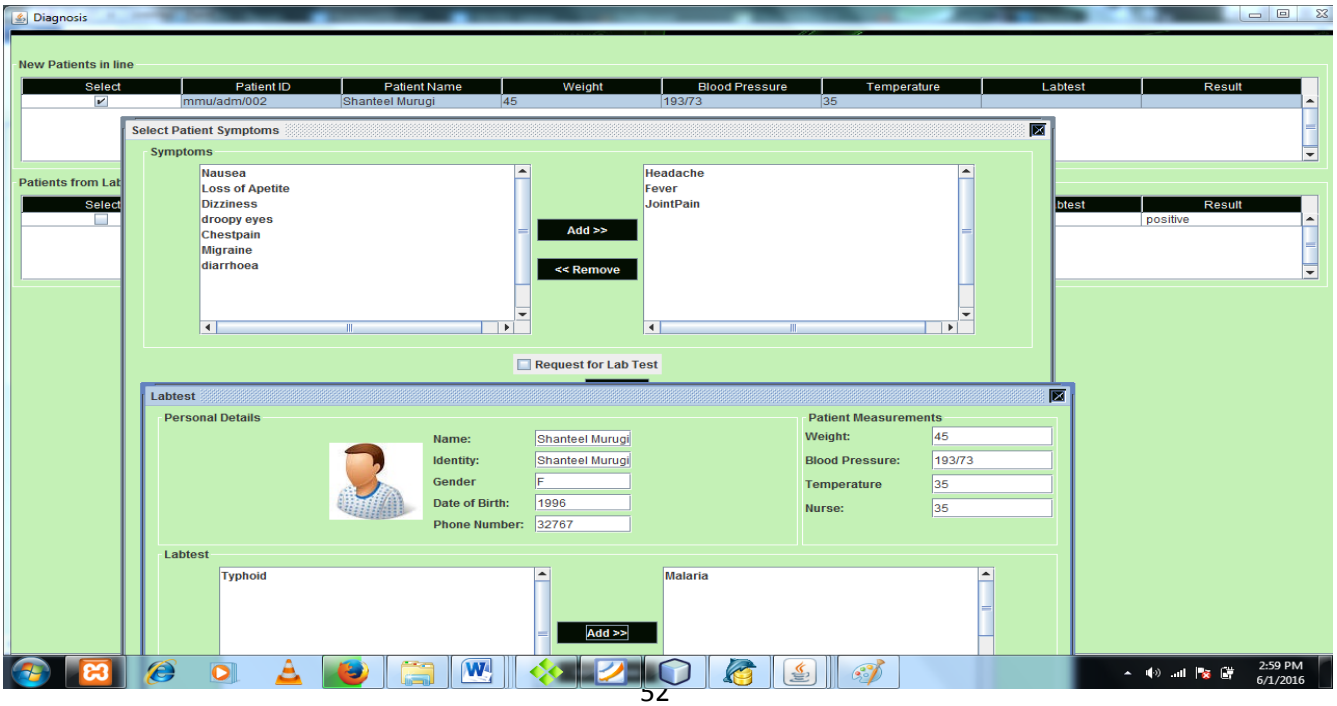

Figure 14: Diagnose Patient

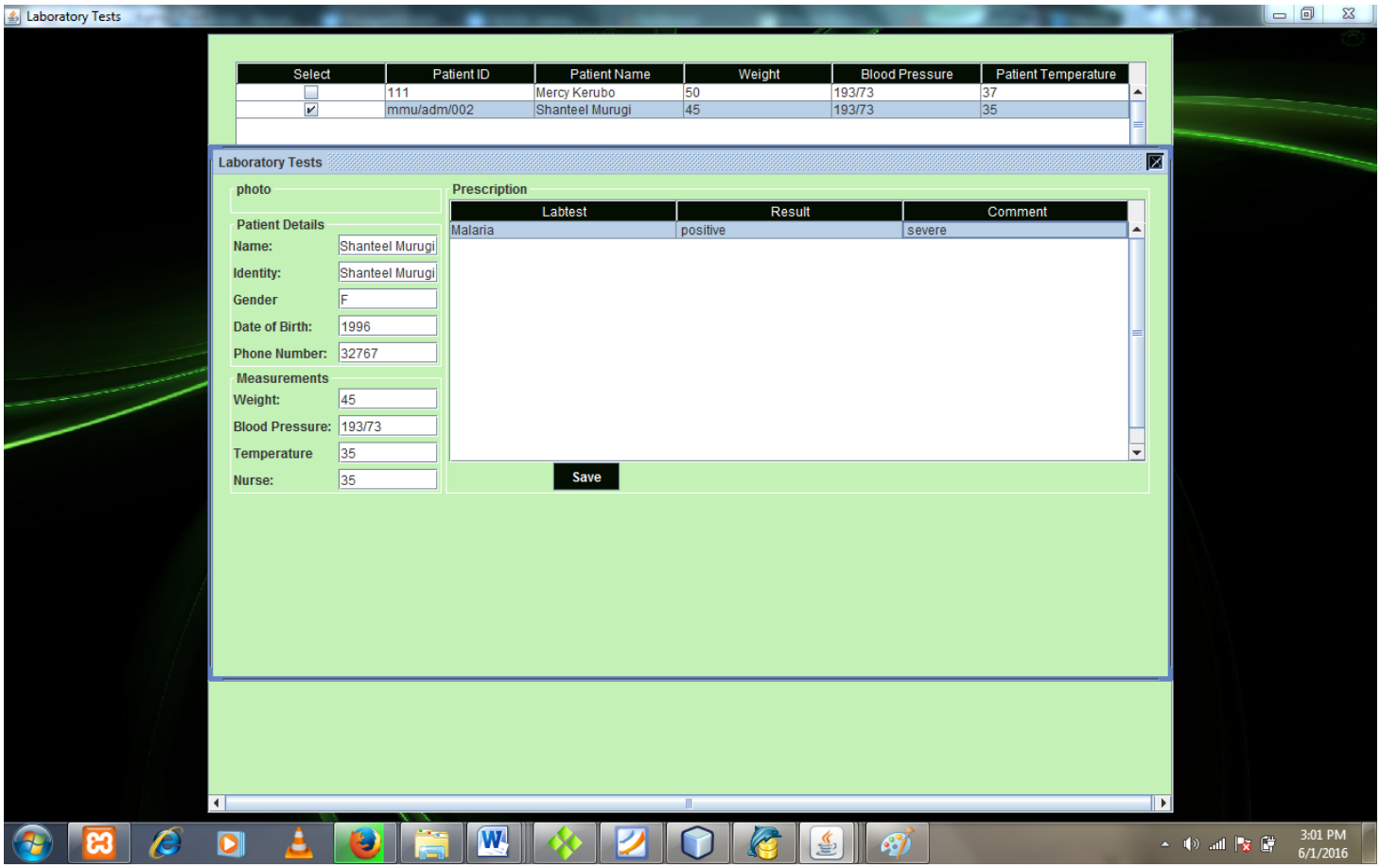

Figure 16: Laboratory Tests

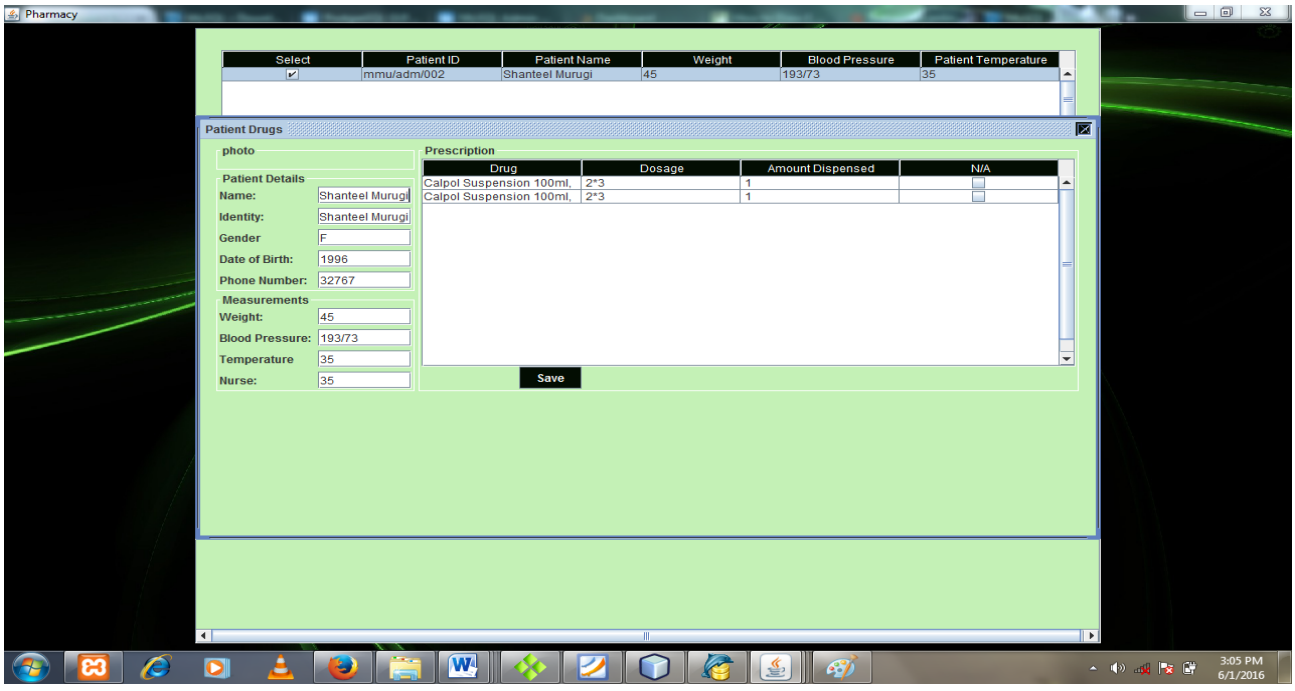

Figure 15: Issuing Drugs

Figure 17: Pharmacy Stock

Figure 18: Equipment Stock

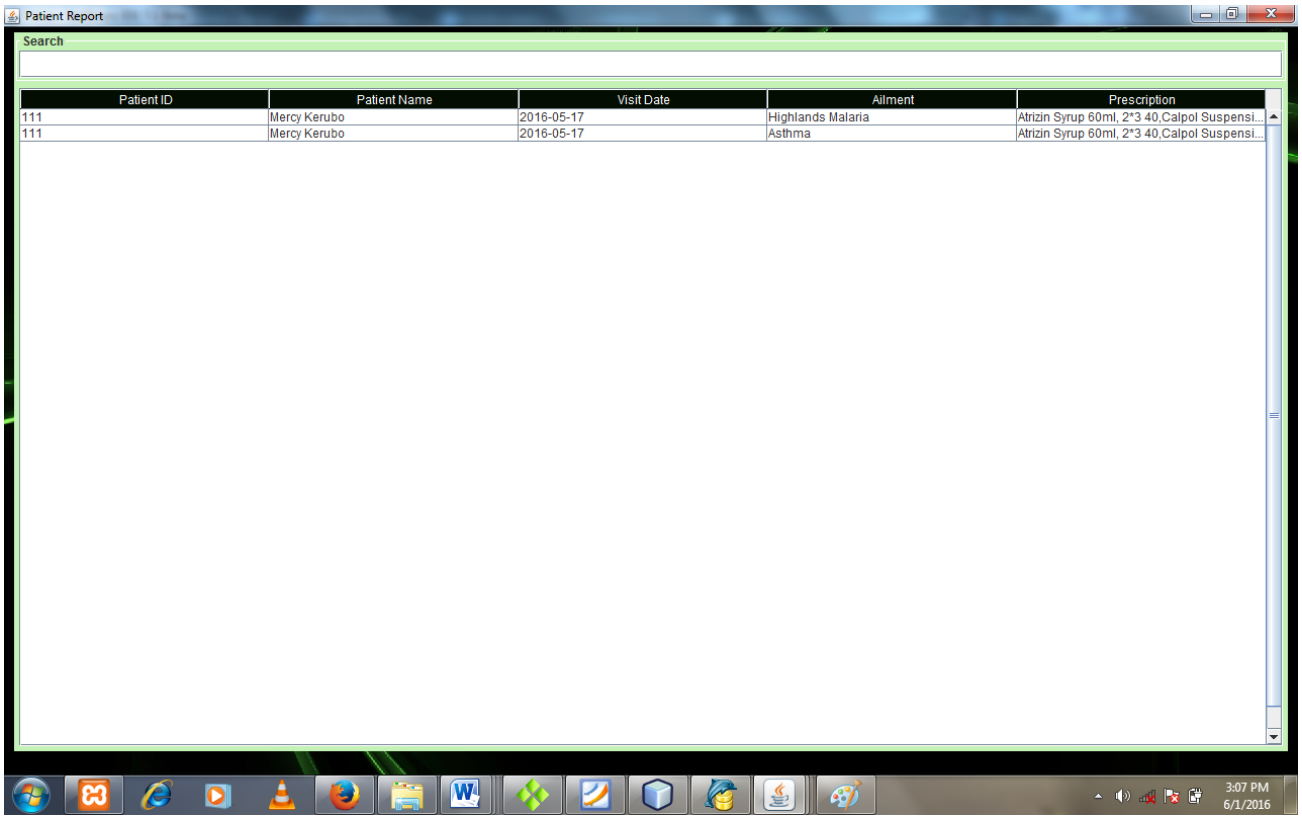

Figure 19: Patient Report

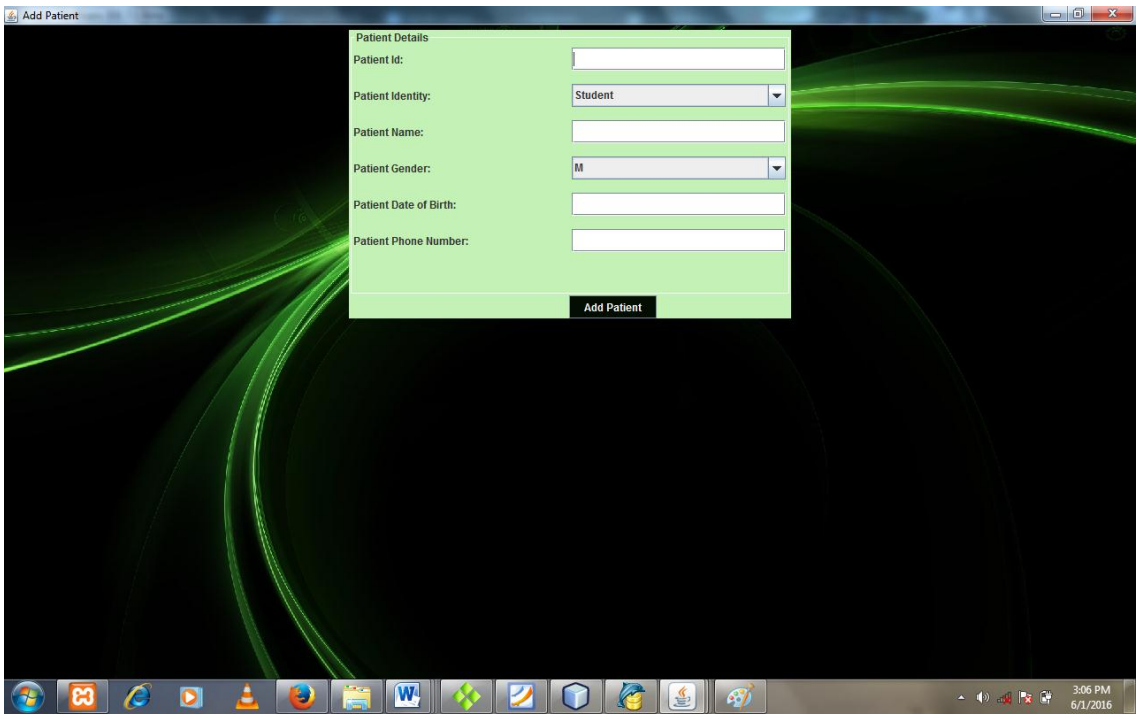

Figure 20: Add Patient

CHAPTER SIX

SYSTEM IMPLEMENTATION

6.1 System development

After designing the overall system architecture, the design was converted into actual system through coding various modules

6.2 Integration and testing

Component/unit testing

The different modules/component of the system was tested independently. During development each module was tested on completion to establish if it conformed to its specification. Its goal was to expose faults in these components

Integration testing

It exposes defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system. During this stage different modules of the system were integrated and tested for compatibility.

6.3 Validation testing

It explains how test and test results will be mapped to documented system requirements. It describes the procedure to ensure that the requirements are validated. Here test data was identified, what was to be tested, the expected output, the actual results after inputting the test data, problems or failures found and suggestion or notes.

6.4 Implementation and deployment

At this stage the system should be rolled out into a production environment

6.4.1 Installation

It will involve both hardware and software installation.

6.4.2 Training

The training phase of implementation will involve introducing the staff at the organization to the system and its functionality. The training will be delivered through workshop case study, tutorials and through practice.

6.4.3 Execution

It will involve the actual operation of the system in parallel with the manual system. The organization management team will perform the execution phase so that they can confirm to what the system is performing as required by the users.

6.4.4 Conversion Plan

There were various steps in the conversion plan before the commencement of the operations.

Hardware Installation

It will involve the purchasing of cables, computers, printers and their related accessories that will all be set at the organization tested and certified as being operational.

Software Installation

It will involve the installation of operating system, a JRE, MySQL DBMS and finally the developed system.

6.4.5 Review

During the review phase the system developer, a moderator and related stake holders who will analyze, inspect the system, compare the manual system to the automated system and draw conclusions on the changeover.

6.4.6 Data Conversion

This will involve changing the data from the old system to a format compatible with the new system. Since the old system was manual this stage will thus involve copying all the data in the manual files and keying it to the computer systems'.

Justification for the Changeover Method

Parallel changeover will be used in replacing the old system with the new system. Certain modules of the new system will be implemented over time and the old system is phased out as functions are successfully replaced by the new system.

The users will be able to learn the new system step by step and still be able to carry out the normal operations until they are comfortable with the new system. In this method the developer will run the new automated system in parallel with the existing manual system until there is a certainty that the new system is working correctly.

Parallel running is likely to be expensive since it involves doing the work twice for a period of time, however it is the safest.

Why use Parallel Changeover

Continuity of organization operations: it will enable development team to curb the errors that might arise without having to completely alter all the operations of the organization.

Convenience: this is because it is not possible for the whole organization operations to come to a halt waiting for the whole system to be implemented.

Creates time to train users: it will be easier to train the users as the old system is running so that some of the data is keyed in the computer during the training while other user are using the manual system to record the data.

Secure: this is because all the data that was in the old manual records will be keyed in to the new system in the process of training the users on the operation of the new system.

6.5 System Testing

Goals and objectives

Software meets requirements: To demonstrate to the developer and the customer that the software meets its requirements.

Find defect: This is the classic objective of testing. A test is run in order to trigger failures that expose defects.

Maximize bug count: The distinction between this and "find defects" is that total number of bugs is more important than coverage.

Minimize technical support costs: Working in conjunction with a technical support or help desk group, the test team will identifies the issues that lead to calls for support. These are often peripherally related to the product under test for example, getting the product to work with a specific printer or to import data successfully from a third party database might prevent more calls than a low-frequency, data corrupting crash.

Assess conformance to specification: Any claim made in the specification is checked. Program characteristics not addressed in the specification are not (as part of this objective) checked.

Conform to regulations: If a regulation specifies a certain type of coverage (such as, at least one test for every claim made about the product), the team creates the appropriate tests. If the regulation specifies a style for the specifications or other documentation, the test team probably checks the style. In general, the test group is focusing on anything covered by regulation nothing that is not covered by regulation.

Minimize safety-related lawsuit risk: Any error that could lead to an accident or injury is of primary interest. Errors that lead to loss of time or data or corrupt data are checked.

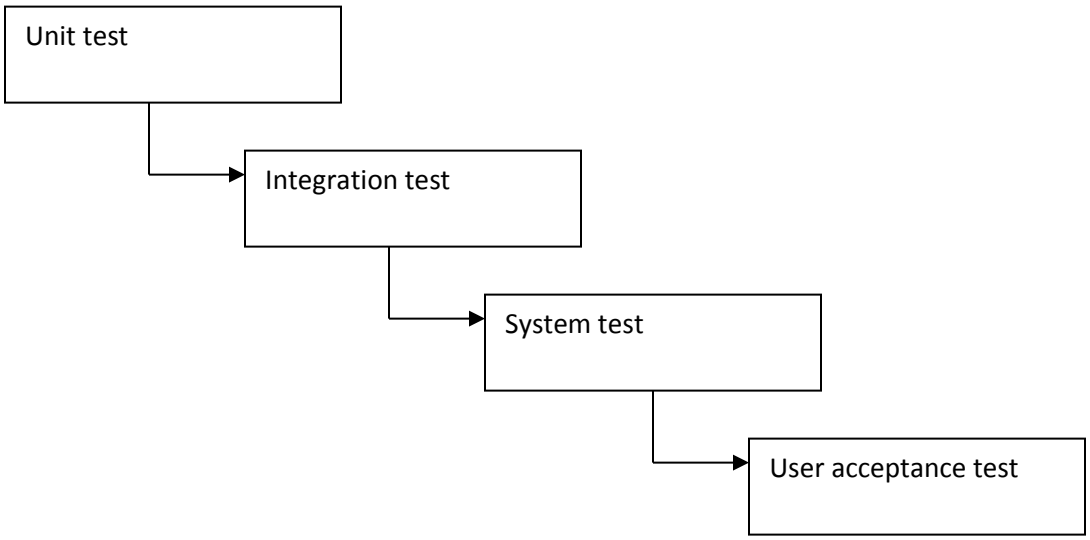

Figure 20: Testing strategy

6.6 Actual Implementation

The following are the steps that will be performed during the actual implementation of the system.

Phase 1

Acquisition and installation of required hardware.

Installation of the new system to all machines.

Setting up and configuring the database.

Configuration of directories for applications

Creation of user and administrator accounts

Phase 2

User training and orientation

Defining procedure for adding new users

Phase 3

Collect processing, performance and output results of the new system.

Monitor and review system performance and problems.

Compare the result of the new system with the current system.

Evaluation of both systems.

Phase 4

Stopping the operation of the old system and putting the new secure computerized information system

CONCLUSION

The project Maasai Mara University Health Unit Management System is for computerizing the working the university's health unit. The software takes care of all the requirements of the health unit. It provides easy and effective storage of information related to patients that visit to the clinic; students, staff and the surrounding community. It also generates reports; patient reports and stock reports

LIMITATIONS

Requires expensive resources-both hardware and software

Basic computer knowledge is a prerequisite for system users

RECOMMENDATION

The system was not fully implemented due to time constraints. Not all the modules mentioned were implemented. Given adequate time, all the modules would be fully designed and implemented.

If system were to be implemented in into the system additional features that make the system distributable should be incorporated. The DBMS used should be powerful, preferable a version of SQL server. The database should be set up on a server that will be accessed by other machines.

REFERENCES

1. *Clinical Information System*. (n.d.). Retrieved 11 01, 2015, from biohealthmatics: http://www.biohealthmatics.com/technologies/his/cis.aspx
2. *Clinical Information System*. (n.d.). Retrieved 11 01, 2015, from biohealthmatics: http://www.biohealthmatics.com/technologies/his/cis.aspx*8.*
3. Government of Kenya, 2011. National Health Accounts, 2009/2010. Ministry of Medical Services and Ministry of Public Health and Sanitation.
4. Health Policy Project, Futures Group. Devolution of healthcare in Kenya assessing county health system readiness in Kenya: a review of selected health inputs. July 2014.
5. *Health Information System*. (n.d.). Retrieved 10 30, 2015, from The Free Dictionary: http://medical-dictionary.thefreedictionary.com/health+information+system
6. *Health Unformation System*. (n.d.). Retrieved 10 30, 2015, from The Free Dictionary: http://medical-dictionary.thefreedictionary.com/health+information+system
7. *information-system*. (n.d.). Retrieved 02 10, 2016, from Encyclopedia Britannica: http://www.britannica.com/topic/information-system
8. *Medical Record*. (n.d.). Retrieved 10 30, 2015, from Wikipedia: https://en.wikipedia.org/wiki/Medical_record
9. *Medical Records/ Policy/ Policies & Publications/ College of Physicians and Surgeons of Ontario*. (n.d.). Retrieved 10 29, 2015, from cpso: http://www.cpso.on.ca/policies-publications/ Policy/ medical-records
10. Maasai Mara University Health Unit Procedure Manual, 2013.
11. MOH. Kenya Health Policy 2014-2030.
12. Kroenke, D M. (2008). Experiencing MIS. Prentice-Hall, Upper Saddle River, NJ
13. O'Brien, J A. (2003). Introduction to information systems: essentials for the e-business enterprise. McGraw-Hill, Boston, MA
14. Y. D. Liang(2004), *Introduction to Java Programming.* Prentice Hall.

APPENDIX

Sample Code

Main page

```
package clinic;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.*;
public class Default extends JFrame {
   public Default() {
      setContentPane(new ImagePanel());
      createComponents();
   }

   public void createComponents() {
      //Layouts
      LayoutManager layout4 = new GridLayout(4, 2, 30, 15);
      LayoutManager layout2 = new FlowLayout(FlowLayout.LEFT, 0, 0);
      LayoutManager layout3 = new FlowLayout(FlowLayout.CENTER, 0, 0);
      LayoutManager layout1 = new FlowLayout(FlowLayout.CENTER, 6, 6);



//Create Jpanels
      JPanel panel1 = new JPanel();
      JPanel panel2 = new JPanel();
      JPanel panel3 = new JPanel();
      JPanel panel4 = new JPanel();
      panel1.setBorder(BorderFactory.createEmptyBorder(10, 100, 10, 100));
      panel4.setPreferredSize(new Dimension(750, 430));
```

```
//Create a Layout Manager
panel1.setLayout(layout4);
panel2.setLayout(layout2);
panel3.setLayout(layout3);
panel4.setLayout(layout1);

//Create icons for the buttons to be created
Icon addPatient = new ImageIcon("src/image/e.png");
Icon diagnosis = new ImageIcon("src/image/f.png");
Icon tests = new ImageIcon("src/image/a.png");
Icon prescription = new ImageIcon("src/image/d.png");
Icon history = new ImageIcon("src/image/b.png");
Icon labStock = new ImageIcon("src/image/c.png");
Icon pharmStock = new ImageIcon("src/image/g.png");
Icon triage = new ImageIcon("src/image/d.png");
Icon exitFrame = new ImageIcon("src/image/exit.png");

//Create Jbuttons and set the texts to display
JButton button1 = new JButton("Add Patient", addPatient);
JButton button2 = new JButton("Diagnose Patient", diagnosis);
JButton button3 = new JButton("Perform Diagnostic Tests", tests);
JButton button4 = new JButton("Dispense Prescription", prescription);
JButton button5 = new JButton("Patient History", history);
JButton button6 = new JButton("Laboratory Stock Inventory", labStock);
JButton button7 = new JButton("Pharmacy Stock Inventory", pharmStock);
JButton button8 = new JButton("Exit", exitFrame);
JButton button9 = new JButton("Triage", triage);
```

```
button1.setHorizontalAlignment(SwingConstants.LEFT);
button2.setHorizontalAlignment(SwingConstants.LEFT);
button3.setHorizontalAlignment(SwingConstants.LEFT);
button4.setHorizontalAlignment(SwingConstants.LEFT);
button5.setHorizontalAlignment(SwingConstants.LEFT);
button6.setHorizontalAlignment(SwingConstants.LEFT);
button7.setHorizontalAlignment(SwingConstants.LEFT);
button9.setHorizontalAlignment(SwingConstants.LEFT);

button8.setPreferredSize(new Dimension(220, 50));

//Add the JButtons to panel1
panel1.add(button9);
panel1.add(button1);

panel1.add(button2);
panel1.add(button5);
panel1.add(button3);
panel1.add(button6);
panel1.add(button4);
panel1.add(button7);
panel2.add(button8);


//Create JTextPanes textpane1 and textpane2 and their preferred settings and contents

JTextPane textPane2 = new JTextPane();

Font font = new Font("Arial Narrow", Font.BOLD, 18);
textPane2.setFont(font);
```

```
JTextPane textpane2 = new JTextPane();
textPane2.setBackground(new Color(248, 251, 251));
Date date = new Date();
SimpleDateFormat sdf = new SimpleDateFormat("EEEE MMMM dd, yyy");
textPane2.setText("Today is " + sdf.format(date));
textPane2.setEditable(false);

textPane2.setPreferredSize(new Dimension(700, 30));
//Add the JTextPanes to panel2

panel3.add(textPane2);
panel4.add(panel3);
panel4.add(panel1);
panel4.add(panel2);

//Create a frame

setTitle("Maasai Mara University Health Center Management System");
setSize(780, 500);
setLocationRelativeTo(null);
setDefaultCloseOperation(DISPOSE_ON_CLOSE);
add(panel4);
setVisible(true);



//Menu Bar
JMenuBar menuBar = new JMenuBar();
// setJMenuBar(menuBar);
JMenu file = new JMenu("File");
```

```
        JMenu patient = new JMenu("Patient");
        JMenu report = new JMenu("Report");
        JMenu help = new JMenu("Help");
        menuBar.add(file);
        menuBar.add(patient);
        menuBar.add(report);
        menuBar.add(help);
//File
        JMenuItem print = new JMenuItem("Print");
        JMenuItem exit = new JMenuItem("Exit");
        file.add(print);
        file.addSeparator();
        file.add(exit);

//Patient
        JMenuItem add = new JMenuItem("New Patient");
        JMenuItem search = new JMenuItem("Search Patient");
        JMenuItem about = new JMenuItem("Patient History");
        patient.add(add);
        patient.addSeparator();
        patient.add(search);
        patient.addSeparator();
        patient.add(about);

//Reports
        JMenuItem numberOfPatients = new JMenuItem("Patients");
        JMenuItem patientDiagnosis = new JMenuItem("Diagnosis");
        JMenuItem labtests = new JMenuItem("Tests");
        JMenuItem pharm = new JMenuItem("Drugs");
        report.add(numberOfPatients);
        report.addSeparator();
```

```
        report.add(patientDiagnosis);
        report.addSeparator();
        report.add(labtests);
        report.addSeparator();
        report.add(pharm);

//Set font

//add action listeners
        button9.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                Nurse t = new Nurse();


            }
        });
      button1.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
              // Register.register();
            }
        });
              button2.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
              Diagnosis d=new Diagnosis();
            }
        });
         button3.addActionListener(new ActionListener() {
```

```
    @Override
    public void actionPerformed(ActionEvent e) {
        Lab l= new Lab();
    }
});
   button4.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
       Pharmacy p=new Pharmacy();
    }
});

     button6.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
       Equipment eq=new Equipment();
    }
});

     button1.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
       Register r=new Register();
    }
});

 button7.addActionListener(new ActionListener() {
```

```
        @Override
        public void actionPerformed(ActionEvent e) {
            Pharmstock ps= new Pharmstock();
        }
      });

  button8.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
          System.exit(1);
        }
      });
    button5.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            PatientReport report= new PatientReport();
        }
      });
  }
}
```