

**FRAMEWORK FOR MODELLING AND REUSE OF MEDICAL KNOWLEDGE IN
SOUTH AFRICA**

BY

KIKUNGA MUHANDJI
(STUDENT NUMBER: 24088935)

DISSERTATION SUBMITTED IN FULLFILMENT OF THE REQUIREMENT FOR THE
AWARD OF DEGREE OF MASTER (MSc) OF COMPUTER SCIENCE



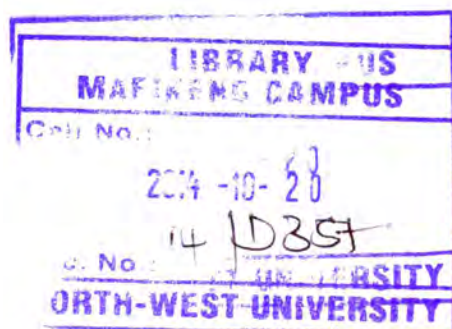
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF MATHEMATICAL&PHYSICAL SCIENCES
FACULTY OF AGRICULTURE, SCIENCE AND TECHNOLOGY
NORTH WEST UNIVERSITY – MAFIKRNG CAMPUS

SUPERVISOR: PROF. OBETEN OBI EKABUA



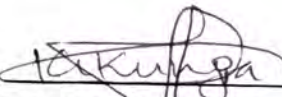
NOVEMBER, 2013

North-West University
Mafikeng Campus Library



DECLARATION

I declare that this Research Project on **Framework for Modelling and Reuse of Medical Knowledge in South Africa** is my work, and has never been presented for the award of any degree in any university. All the information used has been dully acknowledged both in text and in the references.

Signature 
Muhandji Kikunga

Date 18/09/2014

Approval

Signature _____

Date _____

Supervisor: **Prof. O.O. Ekabua**
Department of Computer Science
Faculty of Agriculture, Science and Technology
North West University, Mafikeng Campus
South Africa

DEDICATION

This research dissertation is dedicated to:

My beloved children:

Preddie Mikwimi Mbodila
Graddi Mabanza Mbodila
Merdi Zuka Mbodila

My husband:

Mony Munienge Mbodila

and

My parents:

Bosco Mushinga Kikunga
Josephine Kapinga Kikunga

ACKNOWLEDGEMENTS

I wish to first express my gratitude and praise to the Almighty God, for making everything possible. I bless him for everything he has done for me by giving me life, knowledge, wisdom and making it possible for the completion of this research project and my program to be successful.

I am immensely grateful and thankful to Prof Obeten O. Ekabua, my supervisor, for his motivation, support, advice, discussions and useful criticism while carrying out this research. I am also acknowledging his support and encouragement in pursuing this degree.

I am also grateful to the lecturers and staff of the Department of Computer Science, North West University, Mafikeng Campus, for their help and support.

I also bless the Lord for the help, support, encouragement and prayers of all the elders in my church- Andre & Lynette Van Nierkerk, Oupa & Ouma Palmer, Etienne & Matjeen Van Nierkerk, and all the rest of the church members and friends – Stanley & Memory Kaseke, Sean & Erin Van Nierkerk, Emmanuel & Becky Ruwona, Norman & Joyce, pastor Jean- Baptiste and Maman Lilie Sumbela, pastor Rigobert Katende, prophete Nehemie, Prince and Laurette Tshisuaka, Renee and Lawrence Okuacha, Thandiswa, Bassey Isong, Nosipho dladlu, Thuso Moemi, Nnnena Eric, Ifeoma Ohaeri and to all Pamodzi ladies club.

I am very grateful to all my family - my father Jean Bosco Mushinga Kikunga, my mother Josephine Kapinga Kikunga and my siblings- Rock, Jully, Passy, Nancy, Muzzy, Aristote, Naith, Blanchard, Francine, Chadrack and Jessica Kikunga for their love, support and care whilst undertaking this program and to my mother in law Yvonne Mikwimi Mabanza, Thethe Mabanza, Sarah Mabanza and Niclette Nsakua Mabanza.

Finally, I wish to express my love and gratitude to my husband, Mony Munieng Mbodila, for his love, support and encouragement and to my children, Preddie Mikwimi Mbodila, Graddi Mabanza Mbodila and Merdi Zuka Mbodila for appreciating everything I do for them by saying “Mama you are the best ever in the world and no one else is like you” and for inspiring me, even in the midst of their distractions. You guys are a blessing and a gift from GOD. I love you and I bless the Lord for you being my children.

Table of Contents

DECLARATION	i
DEDICATION	ii
ACKNOWLEDGEMENTS	iii
List of Figures	vii
List of Tables.....	ix
List of Acronyms.....	x
ABSTRACT	xi
CHAPTER 1.....	1
INTRODUCTION AND BACKGROUND.....	1
1.1 Introduction	1
1.2 Background.....	2
1.3 Problem Statement.....	3
1.4 Research Questions (RQs).....	4
1.5 Research Goal.....	4
1.6 Research Objectives	4
1.7. Motivation	4
1.8 Research Methodology.....	5
1.9. Key Terminology.....	5
1.10 Research Contribution.....	7
1.11 Included Publication.....	7
1.12 Dissertation Summary	7
CHAPTER 2.....	8
LITERATURE REVIEW.....	8
2.1 Chapter Overview.....	8
2.2 Medical knowledge	8
2.3 Importance of Managing Medical Knowledge.....	8
2.4 Medical Knowledge Base Construction	10
2.4.1 Medical knowledge Acquisition	11
2.4.2 Medical Knowledge Representation.....	12
2.4.3 Medical Knowledge Acquisition and Representation Tools	12
2.5 Medical Diagnosis.....	13
2.6 Medical Diagnosis Problem	14
2.7 Medical Diagnosis System	14
2.8 Expert System for medical diagnosis	17

2.9 Artificial Intelligent in medical diagnosis	19
2.10 Health care system.....	20
2.11 Challenges of health care system management	21
2.12 Health Care Technology Management	21
2.13 Health Information System (HIS)	22
2.14 Health care Patient Management.....	23
2.15 Some related software application for managing patients.....	23
2.16 Chapter Summary	24
CHAPTER 3.....	26
REQUIREMENT ANALYSIS AND FRAMEWORK DESIGN	26
3.1 Chapter Overview.....	26
3.2 Requirement Analysis	26
3.2.1 System Requirement	26
3.2.2 Hardware requirement	26
3.2.3 Technologies used to develop the system.....	27
3.3 The design of the System	29
3.3.1 Data Collection	29
3.3.2 The 3 tier architecture	30
3.3.3 The Medical Diagnosis Framework Architecture.....	31
3.3.4 Role and functionality of users in the system.....	32
3.3.5 Relationship between the three hospitals connected to the system.....	37
3.3.6 The knowledge Base	38
3.4 MySQL database design.....	40
3.4.1 MySQL Workbench.....	41
Figure 3.6 EER - Diagram.....	42
3.4.2 MySQL Database Tables	42
CHAPTER 4.....	46
SYSTEM IMPLEMENTATION.....	46
4.1 Chapter Overview.....	46
4.2. Medical Diagnosis System's Operations.....	46
4.2.1 The Registration Page	46
4.2.2 The login Page	47
4.2.3 The Administrator Page	48

4.2.4 Doctors' Page.....	58
4.3 Results	61
4.3.1 Consulting a patient for the first time	61
4.3.2 Searching for information in the Knowledge Base (KB).....	62
4.3.3 Treating patient using information in the KB	63
4.3.4 The results of the treatment.....	64
4.3.5 The next Consultation	64
4.3.6 The results of the next treatment.....	65
4.4 Chapter Summary.....	66
CHAPTER 5.....	67
SUMMARY, CONCLUSION AND FUTURE WORK	67
5.1 Summary.....	67
5.2 Conclusion.....	67
5.3 Future Work.....	68
REFERENCES.....	69
APPENDICES.....	74

List of Figures

Figure 2.1:	The knowledge life-cycle model of Siemens healthcare.....	9
Figure 3.1:	The 3- tier architecture.....	30
Figure 3.2:	The medical diagnosis framework.....	31
Figure 3.3:	Role and functionality of users.....	33
Figure 3.4:	The administrator’s flow chart.....	34
Figure 3.5:	Doctors’ flow chart.....	36
Figure 3.6:	Enhanced Entity Relationship (EER)-Diagram.....	42
Figure 4.1:	The Registration Page.....	46
Figure 4.2:	Login Page.....	47
Figure 4.3:	Administrator Page.....	48
Figure 4.4:	Creation of doctor’s accounts.....	48
Figure 4.5:	Results of creation of doctor’s account	49
Figure 4.6:	Searching for doctor’s accounts.....	50
Figure 4.7:	Viewing the lists of doctors.....	50
Figure 4.8:	Registering a patient.....	51
Figure 4.9(a):	Results of patient registration in Limpopo.....	52
Figure 4.9(b):	Results of patient registration in Gauteng.....	52
Figure 4.10:	Viewing all patients.....	53
Figure 4.11(a):	Search for patients using First name or Surname.....	54
Figure 4.11(b):	Search for patients in Gauteng Province.....	54
Figure 4.11(c):	Search for a patient using health care number.....	54
Figure 4.12:	Adding disease name and history.....	55
Figure 4.13:	Adding the disease symptoms.....	56
Figure 4.14:	Adding disease’s treatment.....	56
Figure 4.15:	Adding prescription of the disease.....	57
Figure 4.16:	Viewing the list of diseases in the KB.....	57
Figure 4.17:	Viewing the information of a disease in the KB.....	58
Figure 4.18:	Doctor’s profile.....	59

Figure 4.19:	Consulting a patient.....	59
Figure 4.20:	Changing password.....	60
Figure 4.21:	Consulting a patient for the first time.....	61
Figure 4.22:	Searching for information in the KB.....	62
Figure 4.23:	Treating the patient using information in KB.....	63
Figure 4.24:	Results of the treatment.....	64
Figure 4.25:	The next consultation day.....	64
Figure 4.26:	The results of the next treatment.....	65

List of Tables

Table 3.1:	Login table.....	43
Table 3.2:	Administrator table.....	43
Table 3.3:	Doctors table.....	44
Table 3.4:	Patients table.....	44
Table 3.5:	Knowledge Base table.....	45

List of Acronyms

- AHP:** Analytic Hierarchy Process
- AI:** Artificial Intelligence
- ANN:** Artificial Neural Network
- BMDS** Blackboard –Based Medical Diagnosis System
- CDSS:** Clinical Decision Support System
- CMCS:** Computer Mediated Communication System
- CMDS:** Contract Net Based Medical Diagnosis System
- DR:** Doctor
- EER:** Enhanced Entity Relationship
- HCP:** Health Care Provider
- HCS:** Health Care System
- HIS:** Health Information System
- KA:** Knowledge Acquisition
- KB:** Knowledge Base
- KBS:** Knowledge Based System
- KM:** Knowledge Management
- KR:** Knowledge Representation
- MD:** Medical Diagnosis
- MDS:** Medical Diagnostic System
- MH:** Medical History
- MRD:** Medical Record Diagnosis
- PMIS:** Public Medical Information System
- PMS:** Practice Management Software
- RDMS:** Relational Database Management System
- SA:** South Africa
- UML:** Unified Modelling Language
- WHO:** World Health Organization

ABSTRACT

In today's medical world, there is large amount of information and knowledge about patients and diseases that have been collected while providing healthcare. This information includes patient's medical history, diseases, and diagnosis and treatment methods. However, the problem of making this medical knowledge and data sharable among medical practitioners and reusable over applications and for several purposes is a serious challenge. Though different computer technologies have emerged as leverage in the medical industries, most health institutions are yet to effectively utilize them to manage patients' information and medical knowledge for fast decision making. In South Africa (SA), there is a rapid development of medical institutions and services, which require effective exchange of patient's medical histories and information. But information exchange among medical information systems is difficult and it can sometimes go against medical ethics of privacy and confidentiality. In this dissertation, we developed a framework for medical knowledge-based system to be used by physicians in order to make fast diagnostic decision and share information among the practitioners. The system is a web-based application making use of a central database which links hospitals in three provinces (Limpopo, Gauteng and North West) in South Africa. The advantage of this system is its ability to solve the challenges faced by healthcare practitioners in identifying common ground where relevant medical information can be utilized on a real time basis.

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Introduction

In today's medical society, the discipline of medicine is known to incorporate massive amounts of existing and ever-increasing medical knowledge and information about patients. This information includes patients' medical history, diseases, diagnostics and treatment methods. As a result, medicine is directly or indirectly becoming increasingly a science saturated with information and managing this medical knowledge and information is posing serious challenges for health-care practitioners. The major challenge health-care providers or practitioners face is finding and using the relevant information at the right time [1]. Accordingly, knowledge representation (KR) is another area that is used to solve important problems in today's science world especially where the knowledge has to be reasoned out effectively as part of a decision support system. KR which is described within the abundance of existing expert knowledge plays a key role in the medical domain as practically each of its specializations has a constantly growing and interacting number of relevant guidelines. Basically, the long term goal of KR is the representation of this knowledge in a format that can be used by systems in support of medical decision making. An approach of this undertaking is needed to facilitate systematic representation of different types of medical knowledge that can be used for various types of reasoning [2].

In the medical world today, there are several healthcare systems, and managing a patient in a share-care context is referred to as a knowledge intensive activity. With the knowledge embedded in the systems, health-care providers are able to apply their medical knowledge for making various clinical decisions, such as prognosis, diagnosis, and therapeutic related problem solving and prediction treatment effects [3, 4]. The complex nature of the medical field has resulted in medical knowledge growing continuously and exponentially, which in turn adds to the complexity of existing medical problem solving. Dilemmas are becoming increasingly challenging and failure-prone even for specialists in a highly specialized domain [5]. For instance, conventional medical diagnosis in clinical examinations relies heavily upon physicians'

experience, and these physicians have to intuitively apply knowledge based on symptoms that were found in previous patients.

In everyday practice, medical knowledge grows steadily, such that it becomes increasingly difficult for physicians to keep up with essential information gained from the practice. Thus, for physicians to be able to quickly and accurately diagnose a patient there is a critical need for computer technologies [6, 7, 8]. Computer technologies have become important tools in assisting firstly inexperienced physicians in making medical diagnosis and secondly experienced physicians in supporting complex decisions, retrieving medical information as well as making decisions to overcome medical complications that are prevalent in today's world [9]. An example of this technology is the Medical Diagnostic System.

1.2 Background

In South Africa (SA), the development of medical services, scaling up, grouping of medical institutions, the exchange of patients' medical histories and information is increasingly gaining momentum. In particular, the area of patients' medical histories and information is facing a huge challenge, due to the fact that the existing medical information systems are heterogeneous in development, architecture and supply. Consequently, information exchange among medical information systems is difficult (i.e. the manual method of exchange of patients' information). This also goes against medical practices of privacy and confidentiality [10]. This means that managing medical knowledge and information is a serious challenge for health-care practitioners in SA. The underlying problem is to identify a common ground, where relevant medical information can be utilized at the right time and effectively. Hence, applying uniform standards for medical information in SA is crucial.

In this research work, part of our objective is to provide a solution to the existing problems in SA medical services, particularly effective management of patients' medical information and medical knowledge for practitioners. Our intention is to harness and connect both privately and publicly held relevant information such as patients' personal data, medical reference books, websites, research information and statistical reports to a general medical information system in order to effectively assist health-care providers to make consistent and reliable decisions.

Consequently, this research therefore intends to develop a generic framework and its associated system to enhance the modelling and reuse of medical knowledge for medical diagnosis in SA.

1.3 Problem Statement

In the face of the rapid development of medical services, scaling and grouping of medical institutions in South Africa is increasingly obvious, hospitals and medical insurance organizations frequently need to exchange patients' medical histories and information with each other. Because most medical information systems are developed independently by different software vendors, the heterogeneity in platform layer, system and data layer frequently leads to challenges of information exchange between medical information systems from the same hospitals. In a worst case scenario, a patient's medical information may be spread out over a number of different medical institutes which do not interoperate against the backdrop of privacy and confidentiality. This usually leads to most South African medical institutes or organizations exchanging the patients' information through manual methods which are insufficient, unsafe and go against the rule of medical practice of privacy and confidentiality [11]. Therefore, managing medical knowledge and information becomes an increasing challenge for the health-care practitioners. Medicine as a science that incorporates an enormous amount of existing and ever-increasing medical knowledge and information about patients' medical history, diseases, diagnostic, and treatment methods are necessarily becoming a science of information [12, 1]. Therefore, using uniform standard for medical information in South Africa is imperative.

The real problem faced by patients and health care providers is to identify a common ground where relevant medical information can be utilized at the right time. The main goal is to connect privately held patient personal data, such as medical, diagnosis, treatment plans and outcomes of the treatment to a general public medical information system, medical reference books, websites, research information and statistics reports so that consistent and reliable decisions can be made [13, 8, 1]. This research therefore intends to advance this knowledge in the context of the South African medical environment by developing a generic framework and its associated system to enhance the modeling and reuse of medical knowledge for medical diagnosis in South Africa.

1.4 Research Questions (RQs)

In order to achieve the main goal, this research provides answers to the following research questions (RQs):

RQ1: What aspects of medical knowledge are worth computerizing?

RQ2: What are the design criteria for building useful computer-based systems that will improve the application of various kinds of knowledge in medical care?

RQ3: From knowledge about the design criteria, can we develop a generic framework incorporating the necessary aspects of medical information that can enhance designing a medical information system?

RQ4: As a proof of concept, can we develop a medical information system that would enhance decision making and reuse of information within South African medical practitioners?

1.5 Research Goal

The goal of this research is to develop a generic framework and its associated medical Record System for modeling and reuse of medical knowledge to enhance medical diagnosis in South Africa.

1.6 Research Objectives

To achieve this goal, we set the following objectives:

- i. Combining different literature on designing a medical diagnosis system.
- ii. Building a framework that will be used for Modeling and Reuse of Medical knowledge.
- iii. Designing a medical diagnosis system that will offer physicians all the required information to get adequate medical diagnosis.
- iv. Implementing the system as proof of concept.

1.7. Motivation

The sharing and reuse of medical knowledge has become more difficult, and to have correct information about some medical diagnosis for different diseases has also been a serious problem even for specialists in a highly specialized domain.

To overcome this challenge, it is imperative to design a medical diagnosis system that will assist in providing all the required information needed in medical diagnosis.

1.8 Research Methodology

The research methods to be used in this project are as follows:

1. **Literature Survey:** This method involves surveying the background of the area of interest and analyzing related works. Previous and existing applications of medical diagnosis systems will be reviewed from a study of existing methods, the approach proposed in this research was developed.
2. **Framework Development:** This entails building a generic framework that connects the privately held data such as medical record diagnosis, treatment plans, and treatment outcomes into a single generic approach for a medical information system that enhances relevant decision making by medical practitioners.
3. **Design of the System:** From the framework developed, we designed a medical diagnosis system to assist health practitioners in making diagnosis decisions. The system was designed using Java codes and MySQL as the database.
4. **Implementation of the System:** As a proof of concept to this research, the medical diagnosis system was implemented to conceptualize the proposed idea.

1.9. Key Terminology

A framework: It is a real conceptual structure intended to serve as a support or guide for building of something that expands the structure into something useful.

A Model: It is a representation containing the essential structure of some object or event in the real world.

Medical Model: Refers to a technique used in behaviour therapy in which a person learns a desired response by observing and imitating the behaviour.

Diagnosis: Refers to the identification of the nature of an illness or other problem by examination of the symptoms.

Medical Diagnosis: This is the determination of the cause of a patient's illness or suffering by the combined use of physical examination, patient interview, laboratory tests, review of patient's medical records, knowledge of the cause of observed signs and symptoms, and differential elimination of similar possible causes.

Medical Diagnostic System: A computerized System that is used for medical diagnosis.

Knowledge: This refers to information and skills acquired through experience or education, the theoretical or practical understanding of a subject.

Knowledge Representation (KR): It is an area of artificial intelligence research aimed at representing knowledge in symbols to facilitate making inferences from those knowledge elements, creating new elements of knowledge.

Knowledge Base (KB): is a special kind of database for knowledge management. It is an information repository that provides a means for information to be collected, organized, shared, searched and utilized. It can be either machine-readable or intended for human use.

Knowledge Based System (KBS): Refers to a computer program that reasons and uses knowledge to solve complex problems.

Medical Knowledge: Medical knowledge is the ability of a client to remember and interpret information.

Reuse: Means to use an item again after it has been used.

Health Care: It is the diagnosis, treatment, and prevention of disease, illness, injury, and other physical and mental impairments in humans. Health care is delivered by practitioners in medicine, optometry, dentistry, nursing, pharmacy, allied health, and other care providers. It refers to the work done in providing primary care, secondary care, and tertiary care, as well as in public health.

Health Care System: Organization of people, institutions, and resources to deliver health care services to meet the health needs of target populations.

Health Information System (HIS): refers to any system that captures, stores, manages or transmits information related to the health of individuals or the activities of organisations that work within the health sector.

1.10 Research Contribution

The main contribution of this research to academia and the research community is the development of a generic framework and a medical diagnosis system to assist health care practitioners or physicians to make effective and precise medical diagnosis decisions.

1.11 Included Publication

Part of the research reported in this thesis has been accepted for publication. The paper is:

Muhandji Kikunga and Obeten Ekabua. "Towards a framework for Modeling and Reusing Medical Knowledge in South Africa", *in proc. of the International Conference on Information and Knowledge Engineering (IKE), WORLDCOM'2013*, Las Vegas, Nevada, July 22-25, 2013.

1.12 Dissertation Organization

The remainder of this research project is organized as follows:

Chapter 2 is on review of related literature and looks at what has been done on using medical framework and medical diagnosis systems.

Chapter 3 introduces different methods that were used to design the framework and Medical diagnosis system. It explains how the system works, the databases and its entities.

Chapter 4 presents the implementation of the framework and results obtained from the developed system.

Chapter 5 is the concluding chapter of this research report. A summary emphasizing this project's contributions is presented followed by recommendations. Suggestions for further work are pointed out.

CHAPTER 2

LITERATURE REVIEW

2.1 Chapter Overview

This chapter presents a review on related work done in designing a framework and medical diagnosis system using different methodologies.

2.2 Medical knowledge

Medical knowledge is the ability of a client to remember and interpret information. Medical knowledge is the certain information about relationships that exist between symptoms and symptoms, symptoms and diagnoses, diagnoses and diagnoses and more complex relationships of combinations of symptoms and diagnoses to a symptom or diagnosis [14, 15]. Medical knowledge, in general terms, has to be considered from two points of view: first, Expert Knowledge related to the physician's description of different relationships between symptoms and diagnoses, symptoms and symptoms, and diagnoses and diagnoses and secondly, Patient Information is collected from each patient (patient data collection and structuring) Medical knowledge is used by physicians to make a variety of clinical decision tasks such as diagnosis, prognosis and prediction of treatment effect [3]. Medical knowledge often suffers from different forms of information imperfections (i.e., uncertainty, imprecision, ambiguity, etc.). In addition to the different types of information imperfections, the information can be quantitative (numerical or binary) and qualitative (nominal and ordinal) [16, 17, 14]. Thus the heterogeneity and imperfection of medical knowledge must be taken into consideration when designing medical support systems.

2.3 Importance of Managing Medical Knowledge

Managing medical knowledge and information becomes an increasing challenge for health-care professionals. Having the right information at the right time can become a very difficult problem due to the sheer amount of ever-expanding knowledge. Clinicians, administrators, industry managers and research scientists are facing a growing body of knowledge that they have to routinely access, absorb, and utilize [18,13].. The volume of knowledge doubles itself every 17

years. There is also the issue of the various formats in which information exists as well as the diverse disparate medical information sources. Taking these aspects into account, it is not surprising that knowledge management is attracting so much attention.

Just as we manage our organisation's other key assets, important knowledge needs to be actively managed. The required expertise of most employees is related to economics, business or technical issues; hence medical knowledge has to be actively managed. Siemens healthcare sector is one of the world's largest suppliers to the healthcare industry and a trendsetter in medical imaging therapy, laboratory diagnostics, medical information technology and hearing aids. It offers its customers products and solutions for the entire range of patient care from a single source, prevention and early detection to diagnosis, and on treatment and aftercare. Siemens healthcare takes an active approach to Medical Knowledge Management by executing a series of strategies to improve how knowledge is managed, including a branding strategy focused on mobilising awareness and support of the Knowledge Management initiative. The organization places value on the tacit knowledge that individuals have and combines it with explicit knowledge (e.g. scientific journals, clinical workflows, guidelines) thus improving communication, collaboration and information transfer. Siemens healthcare consists of a dedicated team of Knowledge management (KM) workers who actively manage the operation of KM processes. Their aim is to improve knowledge creation and sharing process in the organization and also co-ordinate the basic processes of the knowledge life cycle which comprise the identification of need, creation, sharing, collection, storage and update [18]. The knowledge life-cycle model of Siemens healthcare is shown below:



Figure 2.1: The knowledge life-cycle model of Siemens healthcare [18]

1. **Identification of Need:** The first thing the organization does is to identify its medical knowledge needs. These needs are determined by profound analysis of the current and future healthcare market, demographic developments as well as screening of most life-threatening but preventable diseases like coronary artery diseases, stroke and/or lung cancer in connection to Siemens Healthcare mission fighting the most threatening diseases. In addition, the identification and interviewing of medical experts inside and outside the firm that have a broad view in the above mentioned disease areas are key drivers in the identification of disease trends. Once the trends have been identified and researched, the findings are compared with staff skills and experiences, thus medical knowledge needs can be identified.
2. **Creation:** The creation process is initiated if the needed medical knowledge is not available. Here the organization focuses on human-centric view of knowledge creation to enable the constructive and subjective nature of such a process. In this regard the utilization of diverse knowledge sources, drawing upon different organizational functions and professional disciplines, is an essential success factor. Medical Knowledge is provided by dedicated Clinical Competence Centers that provide the necessary breadth of medical knowledge combined with extensive clinical expertise.
3. **Collection and storage:** The new information together with existing information are collected and stored in a web-based database named Clinical Knowledge Base. Any input to the knowledge base is evaluated and commented on by the medical experts and thus adapted to the needs of the organization.
4. **Sharing:** Siemens Healthcare approach in successful medical knowledge sharing is the implementation of push-pull strategies within the organization. KM services offered are combination of pull (self-service) and push services (facilitate transfer) specifically designed to meet the needs of the organization.

2.4 Medical Knowledge Base Construction

Constructing and maintaining a good knowledge base is one of the main factors that determine the success of a medical decision support system. The objective of the medical knowledge base

construction is to perform reliable information modelling of the medical knowledge description, expressed by physicians, according to a predefined knowledge representation scheme. The medical knowledge base system construction reduces diagnosis error risks, as well as helping physicians to make high quality and reliable medical decisions [19]. The process of building the knowledge base consists of collecting the relevant medical knowledge from the specific domain, its systematization and technical formalization. The challenges in constructing and maintaining the knowledge base are numerous. First of all there is a big variety of information sources and it is crucial to identify the most relevant ones, including both theoretical and practical. Another problem is that it is not easy to present all details of the expert knowledge in a structured way and establish relations between the specific facts. Once constructed, a knowledge base must be continuously updated with new facts from the latest research, which are sometimes contradictory. The possibility of life-threatening consequences due to incorrect conclusions brings additional risks and responsibilities to this process. Building the knowledge base and keeping it up to date requires a tremendous human expert effort and time.

A prototype of a knowledge- based system was developed to help people living with diabetes in South Africa, especially in rural areas to get advice and knowledge of their conditions without consulting the doctor [20]. The JESS (Java Expert System Shell) was used as a tool for designing the knowledge base expert system.

In order to construct a knowledge Base system the followings steps outlined below are required:

2.4.1 Medical knowledge Acquisition

Medical knowledge acquisition is the first and very important step in building the medical knowledge base. It is a process of extracting, structuring and organizing knowledge from a specific domain so that it can be used in knowledge- based software systems. Important tasks before starting the process are: identifying and evaluating a well-defined knowledge domain, identifying the knowledge sources from the specific domain and identifying the specific knowledge acquisition techniques to be used (interviews, questionnaires, observations, decision trees, rules development, etc.) [21].

A key concept in the knowledge acquisition process is ontology. Within health informatics, ontology is a formal description of the health- related domain and provides a standard vocabulary for biomedical entities [22]. While creating an ontology is not a required step for building the

knowledge base, it facilitates standardization, re-using and easy integration of medical knowledge. The GAMES is a knowledge acquisition environment that sees the process of knowledge acquisition like the construction of two models: the epistemological model and the computer model [23]. PROTÉGÉ environment helps developers in setting up available problems solving methods available [24]. These methods are applied to domain ontologies and give knowledge acquisition tools for specific tasks. M-KAT (Medical Knowledge Acquisition Tool) is conceived to help the user, usually a specialized medical doctor, to build and maintain an expert system executing a medical reasoning. It allows the user to build a knowledge base system working on the results obtained from the analysis at the cognitive level [25].

2.4.2 Medical Knowledge Representation

The next step in the knowledge base construction is formal representation of the acquired medical knowledge in a form that can be processed efficiently [26]. The key principles for effective knowledge representation are [27]:

- i. The expert knowledge must be presented in a human understandable but computer-interpretable manner;
- ii. The clinical knowledge models must provide means of expressing different kinds of knowledge in an unambiguous way to tackle complexity;
- iii. Clinical knowledge should be shareable and re-usable between different organizations and portable between different clinical settings and system platforms.

The use of the possibility theory as a global framework was proposed to construct the medical knowledge representation model [14]. This possibilistic model was applied as a knowledge representation approach, to represent the relationship (Modality) – (Diagnosis), as well as in the construction of the medical knowledge base. This possibilistic representation transformed the linguistic knowledge into a model useable by a decision support system.

2.4.3 Medical Knowledge Acquisition and Representation Tools

By its nature, knowledge construction is a complex process and it is frequently described as being a major bottleneck of the CDSS development [28]. For that purpose, there are many computer tools developed to assist the knowledge acquisition and representation process. These tools serve as interface between a user and a knowledge base by maintaining two different representations of the knowledge: one representation for the knowledge base and one representation for the user [29]. The issues that a knowledge acquisition and representation tool

must be able to address are identifying the domain knowledge that is to be formally encoded, mapping that knowledge into the encoding utilized by the computer, validating that the encoding accurately reflects what the domain expert was trying to express and validating that what was expressed is indeed the appropriate knowledge to apply [29].

2.5 Medical Diagnosis

Medical diagnosis is the determination of the cause of a patient's illness or suffering by the combined use of physical examination, patient interview, laboratory test, review of patient's medical records, knowledge of the cause of observed signs and symptoms, and differential elimination of similar process. Medical diagnosis has always been an art. An artist is a person that can carry out something that others cannot, and that is exactly what a good physician does during a medical diagnosis procedure. He (she) uses his education, experience, and talent to diagnose a disease. A diagnosis procedure usually starts with the patient complaints and the medical doctor observe more information acquired about the patient's condition interactively during an interview, as well as by measuring some metrics such as blood pressure or body temperature. The diagnosis is then determined by taking the whole available patient status into account. Then depending on that, a suitable treatment is prescribed and the whole process might be iterated. In each iteration, the diagnosis might be configured, refined, or even rejected [30, 31]. Medical diagnosis is a categorization task that allows physicians to make prediction about features of clinical situations and to determine appropriate courses of action. It involves a complex decision that involves a lot of vagueness and uncertainty management, especially when the disease has multiple symptoms [32]. Medical diagnosis has undergone different phases of research from statistical methods, which saw the application of Bayesian inference, utility theory, Boolean logic and discriminant analysis. When it was evident that statistical tools could not deal with most complex medical problems, Artificial Intelligence (AI) principles were applied [33].

Making a diagnosis is the pivotal cognitive activity of every practicing doctor. A correct diagnosis will in most cases lead to appropriate treatment. With the high cost of health care, increased patient awareness, medico- legal and insurance pressures, every doctor must be empathic, accountable and cost-effective in patient care. Diagnosis must always be logical and defensible based on considerations of the dynamic internal and external environment of a living human being.

2.6 Medical Diagnosis Problem

The major task of medical science is to prevent and diagnose disease. In 2001, Brause highlighted that almost all physicians are confronted during their formation by the task of learning to diagnose [34]. Here they have to solve the problem of deducting certain diseases or formulating a treatment based on more or less specified observation and knowledge. Some medical diagnosis difficulties that have to be taken into account are as stated below:

- i. The basis for a valid diagnosis, a sufficient number of experienced cases, is reached only in the middle of a physician's career and is therefore not yet present at the end of the academic formation.
- ii. The quality of diagnosis totally depends on the physician's talent as well as his/her experience.
- iii. Emotional problems and fatigue degrade the doctor's performance.
- iv. The training procedure of doctors, in particular specialists, is a lengthy and expensive one. So even in developed countries we may feel the lack of medical diagnosis.
- v. Medical science is one of the most rapidly growing and changing fields of science. New results disqualify the older treatments, new cures and drugs are introduced day by day. Even unknown diseases turn up every now and then, so a physician should always try hard to keep him/herself up to date.

To answer the above problems, and also many others, computers have been employed widely to help in medical diagnosis. From local and global patient and medicine databases to emergency networks, or as digital archives, computers have served well in the medical sector. Regarding the complexity of the task in medical diagnosis, it has not yet been realistic to expect a fully automatic, computer-based, medical diagnosis system. However, recent advances in the field of intelligent systems are going to materialize in a wider usage of computers, armed with Artificial Intelligent (AI) techniques, in that application. A computer system never gets tired or bored, can be updated easily in a matter of seconds, and is fairly cheap and can be easily distributed.

2.7 Medical Diagnosis System

Medical Diagnosis System is a man-computer integrated system which faces the whole process of clinical diagnosis and treatments. It aims to be an assistant for doctors and nurses. Medical Diagnosis System provides expected knowledge, experience, method and model to help doctors to diagnosis and make treatment accurately, and reduces medicine expense effectively [35].

Nowadays, an advanced Medical Diagnosis System must be context-aware. The character of context-awareness makes Medical Diagnosis System to be convenient and humanist. Medical Diagnosis System is always heterogeneous and complex. Modelling, Simulation and verification are important methods for preliminary design of a complete and safe system.

A framework for the development of a low cost cell phone based application was proposed [8]. The framework employs an inference mechanism based on Fuzzy logic and analytic hierarchy process (AHP), in the diagnosis of common tropical diseases such as typhoid fever, diarrheal diseases, pneumonia, Tuberculosis, malaria and amebiasis. The system offered patients the ability to just enter their symptoms on the cell phone and receive immediate advice on a preliminary diagnosis of the system. It was mentioned that symptoms of some tropical diseases often overlap and become confusable due to ambiguity, vagueness and imprecision of symptoms of these diseases and because of this confusion it becomes a challenge for physicians especially the inexperienced ones to make a provisional diagnosis. Therefore fuzzy logic and AHP tools were combined to help with differential diagnosis in an accurate and timely manner, while resolving the ambiguities and uncertainties associated with these symptoms.

Iantovic [36] developed a Blackboard Based Medical Diagnosis system (BMDS) for solving medical diagnosis problems that were based on combinations of illness [36]. The system allowed physicians with a medical specialization plan treatment, to cure illnesses that were in advanced stages. However, the system had some limitations because the treatment to cure illnesses in the less advanced stages was not included in the system. BMDS system is composed of medical expert system agents and different classes of assistant agents. The system showed that medical expert agents can be used successfully as members of a diagnosis multi-agent system. It also stated that Medical expert agents required future improvement in order to increase their autonomy and flexibility in problem solving. Later on the author proposed a Contract Net Based Medical Diagnosis system (CMDS) that can solve medical problems randomly [37]. The system allowed medical expert system agents and physicians to be capable of elaborating medical diagnosis. It was composed of physicians, medical expert system agents and assistant agents. The physicians and artificial medical agents have limited diagnostic knowledge. The advantage that CMDS has over BMDS is in its autonomy and flexibility in handling medical diagnosis problems.

Rao et.al designed a Computer Mediated Communication System Framework for Collaborative Medical Decision Making (CMCS) to enable the storage, retrieval, display and transmission of medical data records from distant locations from a patient's primary health care institution thus enabling physicians to collaborate and generate diagnosis [38]. They mentioned that their CMCS framework attempts to minimize the inherent cognition complexities associated with traditional phases in medical decision making which entail the process of differential diagnosis. It also facilitates intelligent medical collaboration and attempts to minimize cognition related to medical information.

Shujun et. al developed a Medical Diagnosis System for the health- care to be available at any time and any place. They designed a context-awareness framework and introduced UPPAAL as a new tool for modeling, simulating and verifying a real-time system in the medical diagnosis system. This system used a context Aware Database that was responsible for storing information of doctors, nurses, patients and diseases, completing context information processing. The Database was able to store patients' information, doctors' information, nurses' information, relations between doctors and patients, and between nurses and patients; the trust- worthiness of the patient to the doctors and nurses, and disease diagnosis. They stated that their model of Medical Diagnosis System was still simple and that it had to be improved and extended in future.

A fuzzy medical diagnosis system was designed to support physicians in non-invasive data analysis and medical diagnosis [39]. The objective of the system was to define an architectural model in order to build an open, distributed medical diagnosis system whose quality of feedback relies on the acquisition of medical knowledge. Their final result was the building of a framework that represented a flexible and dynamic architecture for the diagnostic processing of clinical data. Their approach was based on building a distributed model that integrated different medical domains in order to enable the access and consultation of different pathologies through a conceptual representation of medical knowledge (ontologies).

A system was developed to be used, compared, exchanged, and discussed with clinical staff of other hospitals with different cultures [40]. This system facilitates reuse of critical software components and introduced small variations, which are useful in a medical context.

A pattern recognition system for the diagnosis of gonorrhoea disease was designed to support doctors in diagnosing of sexual transmitted diseases (STDs) like gonorrhoea [41]. The system intended to assist physicians to determine precisely whether or not a patient is infected with gonorrhoea. This system presented the application of pattern recognition for the diagnosis of gonorrhoea with better performance, reliability and increased efficiency and availability.

2.8 Expert System for medical diagnosis

Expert system is a form of Artificial Intelligence program that simulates the knowledge and analytical skills of one or more human experts [42]. There are many existing medical expert systems to assist physicians in diagnosing which may shorten the time spent in correcting diagnosing errors. At the same time the physician may obtain information on the symptoms of each of the diseases and pathologic syndromes contained therein. An expert system provides advice derived from its knowledge base using a reasoning process embedded in its reference engine, the thinking part of the system. MYCIN is one of the first rule based medical expert systems that was developed at Stanford University in 1972 [43]. MYCIN is intended to provide physicians with advice about diagnosis and drug therapy for bacterial infections. It identifies the organisms responsible for an infection from information concerning the patient's symptoms and test results. This expert system was designed to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for the patient's body weight. It uses backward chaining procedure. The main limitation of MYCIN was its incomplete knowledge base which does not cover a full spectrum of infectious diseases. This is mainly because full spectrum knowledge requires more computing power than most hospitals could afford at that time.

ONCOCIN was a rule- based medical expert system for oncology protocol management developed at Stanford University, 1981. It was designed to assist physicians in treating cancer patients receiving chemotherapy [44]. ONCOCIN is able to recommend treatment for cancer patients because it has a large knowledge base of oncology protocols. At the highest level, it is used to describe the ordering, or algorithm, of the various chemotherapies and radiotherapies being administered to the patient. ONCOCIN reasons about the precise treatment that should be given based on its knowledge of the protocol, the patient's past response to treatment, and current

patient data. ONCOCIN uses the same rule-based approach as MYCIN but it uses a difference inference engine. ONCOCIN uses the forward chaining inference engine to reason. The main advantage of ONCOCIN over MYCIN is that it allows an interaction with previous information or historical data.

Computer Aided Diagnostic (CADIAG-2) expert system is a knowledge representation of fuzzy concepts for medical diagnosis that was designed to assist physicians during diagnosis. Its inference engine uses fuzzy set of theory based methods [45]. The main advantage of CADIAG-2 is its simplicity and it has also been shown to give reasonable results with less error. The common weakness of this expert system is that it does not focus on several independent rules giving the same diagnosis with an equal certainty.

DIABNET is a knowledge-based system that was developed to help medical experts with therapy planning in gestational diabetes [46]. DIABNET helps physicians to interpret the data. DIABNET helps patients with diet planning, insulin therapy adjustments and any schedule alteration. The biggest disadvantage of DIABNET is that it sometimes gives patients risky advice which could be dangerous to them. However, this risk could be managed by delaying any decision until sufficient information is available, generally on the patient's next visit.

GIDEON is another expert system that is a very useful resource for medical practitioners that provides decision support for generating a differential diagnosis [47]. It is used for difficult-to-identify infectious diseases for patients with a travel history. It clarifies unusual disease patterns and helps to plan effective treatment. To assist medical professionals with diagnosis, GIDEON generates a ranked list of possible diseases based on signs, symptoms, laboratory tests and exposure history for any infectious disease, in any country of the world.

XpertMalTyph is an expert System for Medical Diagnosis of complications of malaria and Typhoid (XpertMalTyph) to help patients and assist doctors to make a medical diagnosis decision. The system was developed using Uniform Modeling Language (UML) to have a detailed description of functionality of the system and understand the basic requirement. Java Expert System Shell (JESS) was used to implement the expert module of the medical diagnosis system to allow ease of processing of rules. This work was also able to enhance the expert

system by implementing a system which was a hybrid of expert system and artificial neural network (ANN).

2.9 Artificial Intelligent in medical diagnosis

Intelligence can be defined as the ability to learn and understand, to solve problems and make decisions. However, Artificial Intelligence (AI) is the display of intelligence by machines, and its main goal as a field is to make machines do things that would require intelligence if done by humans. The central problems of AI include such traits as *reasoning, knowledge, planning, learning, communication, perception* and the ability to *move and manipulate* things. In the early 90s, AI achieved its greatest application, despite all its previous setbacks. It was successfully applied in logistics, data mining and medical diagnosis. The success of AI started being revived with the commercial success of expert system in 2003 [48]. Artificial intelligence is a branch of computer science capable of analysing complex medical data. Its techniques can be applied in almost every field of medicine. Their potential to exploit meaningful relationships within a dataset can be used in diagnosis, treatment and prediction in many clinical scenarios.

Fuzzy is one of the artificial intelligence techniques. It deals with uncertainty in knowledge that simulates human reasoning in incomplete or fuzzy data. Fuzzy logic has become an important field of study with a wide spread of applications in diversified fields including medical diagnosis. To diagnose a patient quickly and accurately, there is a critical need to employ computerized technologies to assist in diagnosis and access the related information. The complexity of medical practice makes traditional approaches of analysis inappropriate. Most medical diagnosis is full of uncertainty and imprecision. Fuzzy logic which is one of the soft computing techniques can render precise what is imprecise [49]. Fuzzy logic provides the opportunity for modeling conditions that are imprecisely defined. Fuzzy techniques in the form of approximate reasoning provide decision support and expert systems with powerful reasoning capabilities. The Fuzzy Expert System has proved its usefulness significantly in medical diagnosis for the quantitative analysis and qualitative evaluation of medical data, consequently achieving correct results. The types of human agents that can play a role in artificial intelligence in medical diagnosis are as follow:

- i. **The medical meeting agent:** This is considered as a high level hierarchical agent. It is a set of agents defined to identify the group of physicians. The communication protocol

between agents follows human communication procedures based on personal conversations; that is, somebody intervenes and the others remain silent and attentive. Once the person has finished, another one intervenes, and so on. The leadership position can be assumed by the medical doctor who sits down face-to-face with the patient (Physician in charge of the case).

- ii. **Medical speciality agents:** They define a classification based on the different medical specialities, i.e., human agents such as radiologists, neurologists, pathologists and others. Agent knowledge is heterogeneous. The agents divide the tasks among themselves and share data about the patient. Each one of these specialists can observe only one part of the “outside” (i.e., the patient).
- iii. **The patient:** A human agent who has an active role in his/her treatment. He/she requests medical appointments, takes part in his/her cure, and asks for information. He/she permits the clinical context to be established.

Rahman et.al proposed a design for an Intelligent Patient Management Software (IPMS) system in developing countries to help physicians in patient history, diagnosis, prescription and maintaining proper lists of patients [50]. The primary motivation and purpose of their research work was to ensure more quality and efficient medication to people, as physicians have to take a lot of responsibilities. The Intelligent Patient Management (IPMS) was intended to help physicians to manage their chamber and patients. It features a comprehensive waiting patient list management, setup of diagnosis, advice, diseases and surgery, database management, intelligent medication and multi-language support. They mentioned that they would continue to extend their work and hoped to have an impact on the improvement of public health facilities.

2.10 Health care system

Health care is the maintenance and restoration of health by prevention, treatment, and management of illness and the preservation of mental and physical well-being through the services offered by the medical and allied health professions.

Health care services depend on effective diffusion of health technology to meet their variety of needs. It uses technology to gather information necessary for appropriate diagnosis; to process this information and present it in comprehensive forms; to treat disorders effectively when diagnosed; to monitor treatment and evaluate its efficacy.

2.11 Challenges of health care system management

Health care is a special commodity in emerging economies, in both economic and social terms. South Africa is no exception to the rule. For many countries, health care is a problem on a priority scale dominated by poverty, a growing population and rural to urban migration. About 70-80 percent of the population in most African countries lives in rural areas. Although South Africa is economically advanced compared to other African countries, it still has a large rural population and the benefits of health care services have not been fully realized because of infrastructural constraints. Most rural areas in South Africa receive inadequate health services, because most health services are concentrated in urban areas and cities. The use of telecommunication to provide health care services has brought hope of providing health care services to the rural population [51, 52].

Strategic planning and management of technological investments in health care systems, hospitals and clinics are the most difficult challenges facing health policy makers and planners in South Africa. Although the public health care system is far from being homogenous, it is evident that two large groups of technologies are considered simultaneously, namely, technology for primary health care services and hospital-based technologies for curative services. The effective management of the two groups of technologies will lead to competitive health services delivery – higher outputs at lower costs. It is therefore imperative that investments in health technology infrastructure must take into account technological relationships and dynamics between various levels of health care delivery to ensure the cost-effectiveness of technological investments [51].

2.12 Health Care Technology Management

Health technology management has been defined as an accountable, systematic approach to ensuring that cost effective, efficacious, safe, and appropriate equipment is available to meet the demands of quality patient care. Modern health care institutions, even in emerging economies like South Africa, are increasingly coming to the view that technology is an integral part of all major health service delivery decisions. Technological innovations in the 21st century have reshaped the field of medicine and the delivery of health care services. Recent advances in health technology have provided a wide range of diagnostic, therapeutic, and rehabilitative tools and instruments that are now routinely used in the cure of specific disease and illness. In the process, modern hospitals have evolved as technologically sophisticated health care facilities serviced by

technologically specialized personnel. The changes have contributed significantly to the quality of health care provided by various health facilities, from teaching hospitals to rural clinics; but have also brought new challenges in health technology in South Africa [51].

MedWeaver is a unique web-based facilitator, integrating powerful differential diagnosis tools with diverse information resources to generate personalized answers to healthcare professionals. Through the differential diagnosis and disease lookup modes, clinicians are guided to the focused content they need. The result is better decision making. Studies show that clinical judgement combined with the use of differential diagnosis tools increases diagnostic quality and accuracy.

2.13 Health Information System (HIS)

Health Information System is a discipline of information science, computer science, and health care. It deals with the resources, devices and methods required to optimize the acquisition, storage, retrieval and use of information in health and biomedicine. Health Information Systems are expected to play an important role for improvement of health services in society, however very few systems in developing countries meet that demand. Availability of reliable, relevant, comprehensive and timely health information is widely recognized and recounted as an essential foundation for any public health intervention [53].

Several HIS researchers have mentioned that health information systems in developing countries have been considered as obstacles that hinder the provision of quality health services, rather than supporting it. For example, data is of poor quality and there are problems associated with target setting [53, 54]. HIS in developing countries have in recent years gained increasing attention as more effort by governments, international agencies, non-governmental organizations, donors and other development partners seek to improve health care as a way to reverse disease trends in these countries. There is also an indication that computerization in health information systems has become increasingly apparent. The authors stated that computer technology can greatly enhance and expedite data processing and presentation, and hence facilitate the interpretation and use of information for decision making. The United Nations identifies eight development goals to be achieved by all countries by 2015 and three of them are directly related to health and well-being [55]. It was also identified a long time ago by World Health Organization (WHO) that HIS is critical for achieving health for all [56]. Reports from health facilities and surveillance allow

determination and monitoring of the presence of a disease in a population. The timely use of the reliable information is essential to make informed decisions, set priorities, and improve the quality of services or track epidemics. However, HIS of developing countries, as emphasized by different information system researchers, are not good enough to give the required management support.

2.14 Health care Patient Management

The patient: A human agent who has an active role in his/her treatment. He/she requests medical appointments, takes part in his/her cure, and asks for information. He/she permits the clinical context to be established [3]. Ensuring a computer based quick and accurate patient management system has been a challenging goal for many years. This adds additional challenges in undeveloped countries where the doctor-patient ratio is higher than in developed countries. With the advancement of software engineering and information technology (IT), a lot of practical, professional and personal fields are becoming largely automated by software applications. The medical sector and health care is one of them which is already a nascent field in this respect, and day by day it is engaging more and more with software applications [50].

In general, IT allows health care providers to collect, store, retrieve, and transfer various types of information electronically. IT plays a vital role nowadays because it has the potential to improve the quality, safety and efficiency of health care. Quality health care relies on physicians, nurses, patients and their families, and others having the right information at the right time and using it to make the right decisions. However, the health information needed to make these decisions changes frequently; the guidelines and clinical evidence continually evolve, as does knowledge about the condition of the patient. IT may provide a tool to store, integrate, and update this information base. Thus IT is continuously helping health care in terms of security, quality, efficiency, integrity, prudentially [50].

2.15 Some related software application for managing patients

A category of software that deals with the day-to-day operations of a medical practice is known as Practice management software (PMS) [50]. Such software frequently allows users to capture patient demographics, schedule appointments, maintain lists of insurance payers, perform billing tasks, and generate reports. PMS is often connected to electronic medical record (EMR) systems. While some information in a PMS and an EMR overlaps - for example, patient and provider data

- in general the EMR system is used for assisting the practice with clinical matters, while PMS is used for administrative and financial matters. Dental Management and Patient Record - Open Dental is the first Open Source dental management package with very broad capabilities on record management, patient scheduling and dental office management. FreeMED is a practice management and electronic and computer records system. It allows the tracking of medical data, in detail, with preservation not just of the diagnosis but the reasons for medical encounters. FreeMedForms is a full set of medical applications (EMR, prescriber, drug interaction checker). GP Desk has exciting and not limited features such as storing a complete electronic record of your patients' medical history. GP Desk includes allergies, problems, diagnoses, procedures, tests, medications, immunization, and pregnancy, events/encounters which include symptoms, signs, joint mannequin, and a skin display. GP Desk includes a scheduler for patient appointments, a disease monitoring system, a patient recall system, clinical tools (e.g. cardiac risk calculator), a billing system, a stock control system, and a user messaging system. Another popular software is PMP – Patient Management Program. Its main features include ability to Schedule patients and book appointments easily, Process patient activity and manage patient accounts, Merge patient data to produce personalized communications, Generate statistical reports to help you analyze and improve your practice. EzMedPro is another Medical Practice Management Software that helps clinicians automate their practice. This Medical Management Software integrates data entry, scheduling, Records Management, billing and reporting. Features like Manages Multiple Practices, Multiple providers per practice, Electronic Claims, Electronic Medical Records, CPT and ICD Code Management, Data Backup and Data Restoration are available on this.

2.16 Chapter Summary

This chapter focused on reviewing existing approaches on designing a framework for modelling and reusing medical knowledge. However, most of the previous work done on modelling and reusing of medical knowledge has stated that managing the medical knowledge and information has become an increasing challenge for health care professionals. Medicine as a science that incorporates an enormous amount of an existing and ever-increasing medical knowledge and information about patient medical history, diseases, diagnostics and treatment methods, is increasingly becoming a science of information. It also mentioned that the real problem faced by patient and health care providers is to find and utilize the relevant information at the right time.

This research, therefore, designed a framework and a medical diagnosis system to improve in managing medical knowledge and information about patient medical history, disease, diagnostics and treatment methods and assist health care practitioners to find and utilize the relevant information at the right time.

CHAPTER 3

REQUIREMENT ANALYSIS AND FRAMEWORK DESIGN

3.1 Chapter Overview

This chapter introduces different methods that were used in order to design the framework and describes every technology used for the development of the system. The database and its entities are also shown and explained. Finally, it explains how the system works.

3.2 Requirement Analysis

The requirement analysis defines the required hardware, software, people and environment for the system to work effectively. The medical diagnosis system is designed and developed using open source software running on windows 7 Professional Operating System. The system is developed using open source technology. The reason for choosing open source software is because they are free of cost, easy to use and learn, and has high performance.

3.2.1 System Requirement

The system is designed and developed to allow physicians/doctors to use the information stored in the knowledge base in order to a make an effective decision on diagnosing patients. The doctors and the system administrator are the only authorized users that have access to the system. The administrator is allowed to register patients and create accounts for doctors and enter information in the knowledge base.

3.2.2 Hardware requirement

The hardware devices suitable for the running of the system are the Personal computers (PC) (i.e. Desktop/ Laptop Computer). The specification for the PC is as follows:

- i. Operating System: Windows XP, VISTA, and 7 Professional or latter versions
- ii. Processor and Speed: Intel (R) core (TM) 2 Duo CPU P8600 @ 2.40 GHz
- iii. Memory: 2.00 GB
- iv. System type: 32 bit operating system
- v. Hard disk space: 2.11 GB
- vi. Internet browser and version (MS Internet Explore)

In all, any hardware (PC) having these capabilities will be able to run the system that is to be developed in this research project.

3.2.3 Technologies used to develop the system

Different technologies were used in order to design and develop the system and these are explained below.

3.2.3.1 MySQL

MySQL (pronounced My- ESS- Que_ Ell) is a very fast robust, relational database management system (RDMS) that enables the user to store, search, sort and retrieve data. The MySQL server controls access to the data to ensure that multiple users can work with it concurrently to provide fast access to it and ensure that only authorized users can obtain access. Hence MySQL is a multi- user threaded server. It uses SQL (Structured Query Language), the standard database query language worldwide that stores, updates and accesses data easily. MySQL handles large databases, in the area of more than 50 million records. MySQL is now available under an open source license, but commercial licenses are also available.

3.2.3.2 Java Programming Language

Java is a simple, object oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high- performance, multithreaded and dynamic language. It also provides database connection to enable search, random generation of questions and other database related operations. Java is an extension of C and C++ with the added feature of garbage collection and improved memory management. It deals with objects and their behavior and has an extensive library of routine for coping easily with TCP/IP protocols like HTTP and FTP. Java is intended for writing programs that must be reliable in a variety of ways and for use in network distributed environments. Java bytecodes are translated on the fly to native machine (interpreted) and not stored anywhere. Java programs carry substantial amounts of run time information. Its bytecode is more efficient and its interpreted nature provides high performance. The features and advantages of java are explained in detail below:

A. Features of Java

- i. **Simple:** Java is an extension of C and C++ with the added feature of garbage collection and improved memory management.

- ii. **Object-oriented:** Object oriented programming deals with objects and their behaviors and hence an analogy of real world can be found in programs.
- iii. **Network-savvy:** Java has an extensive library of routines for coping easily with TCP/IP protocols like HTTP and FTP. This makes creating network connections much easier
- iv. **Robust:** Java is intended for writing programs that must be reliable in a variety of ways. Java puts a lot of emphasis on early checking for possible problems, later dynamic (runtime) checking, and eliminating situations that are error prone.
- v. **Secure:** Java is intended for use in networked/distributed environments. Toward that end, a lot of emphasis has been placed on security. The changes to the semantics of pointers make it impossible for applications to forge access to the user's hard disk.
- vi. **Architecture neutral:** To enable a Java application to execute anywhere on the network, the compiler generates an architecture-neutral code – bytecode which is executable on many processors, given the presence of the Java runtime system.
- vii. **Portable:** There are no "implementation independent" (machine/processor dependent) aspects of the specification. The sizes of primitive data types (integer, float) are specified.
- viii. **Interpreted:** Java bytecodes are translated on the fly to native machine instructions (interpreted) and not stored anywhere.
- ix. **High performance:** Java Bytecode is more efficient and its interpreted nature provides high performance.
- x. **Multithreaded:** Multithreading is a way of building applications with multiple threads (more than one processes running at the same time). It is useful for better interactive responsiveness and real-time behavior.
- xi. **Dynamic language:** Java programs carry substantial amount of run time information.

B. Advantages of using Java programming language

- i. **Fast execution:** Java executes rapidly, since the native code of a Java program is continuously re-optimized, not only for executing hardware, but also with regard to program usage, i.e. what parts of the software are actually used, and how, at any given moment. There is no other commercial language that would be faster.
- ii. **Time-to-market:** Java easily lends itself to a robust programming style. It gives high quality code from relatively short development cycles, and enables us to offer very short

time-to-market periods. There is also a great selection of high-quality development and productivity tools available for Java, all of which ease development and maintenance. Other language environments only offer a fraction of the range of tools that are available for Java.

- iii. **Large systems:** Java is especially suited for large systems with long life cycles. Java encourages a lucid programming style which makes Java programs easy to maintain and support with low total cost of ownership.
- iv. **Flexibility:** Furthermore, the continuing development of the Java Virtual Machine (JVM) means that our products can often take advantage of new Java performance enhancements without changing anything in our code, without even re-compiling it, by simply deploying it using a new JVM.

3.3 The design of the System

The structure of the system shows how the framework or the system was developed from the beginning to the end. It describes all the steps required in order to obtain the results. The structure and steps on how the system was designed are described in details below.

3.3.1 Data Collection

The collection of the data is the first phase in designing the system and it helps in developing the medical diagnosis system. This phase involves interaction with medical practitioners in order to obtain experimental knowledge on the diagnosis of diseases. This is done through structured interviews and questionnaires. A major aspect of data collection is the process of establishing the significance of specific symptoms of the diseases. Usually, a combinatorial analysis of symptoms determines possible diseases. The combinatorial analysis is obtained from the physicians' experiences in diagnosis of diseases. In this stage we had to interview many different experienced physicians from different hospitals to get a better understanding of the diseases and their symptoms, how these different diseases could be prevented and treated, and at the same time we also tried to get some information of the history of each disease and how this can be developed in a patient's body. The data we collected were from different hospitals and different doctors. As mentioned earlier, this was done through interviews and questionnaires. The reason for interviewing different doctors was to get different information from each doctor and compare the results in order to come up with the conclusions.

3.3.2 The 3 tier architecture

The 3 tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. The 3 tier Architecture consists of client tier, server tier (middle tier) and database tier. Each of these are explained below.

- **The client tier:** this consists of a web browser as user interface. It receives input from users and sends a request to the server tier.
- **The Server tier:** receives and processes the request, passing it to the database tier (MySQL server)
- **The database Server:** retrieves the data and sends the data back to the server tier. Finally the server tier receives the data, executes the MySQL –query and passes the results to the client tier.

The medical diagnosis framework is designed as a web application with 3 tier architecture that also consists of client tier, server tier and data tier as shown in Figure 3.1.

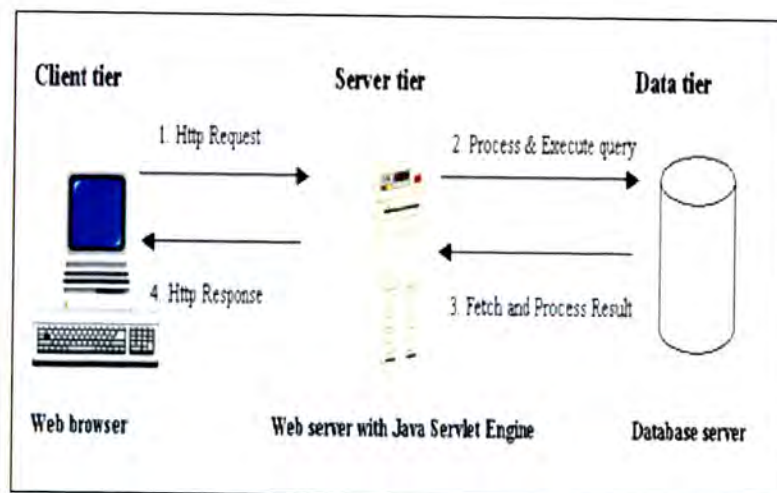


Figure 3.1: The 3-tier architecture

The 3-tier architecture shows how our medical diagnosis framework works. First the client sends a request to the server and then the server processes and executes the query to the database, then

when the database gets the message from the server, it responds to the message and sends back to the server to be delivered to the client.

3.3.3 The Medical Diagnosis Framework Architecture

As mentioned earlier the medical diagnosis framework is designed as a web application that work as 3-tier architecture consisting of client tier, server tier and database tier. The design of the medical diagnosis framework is shown in Figure 3.2.

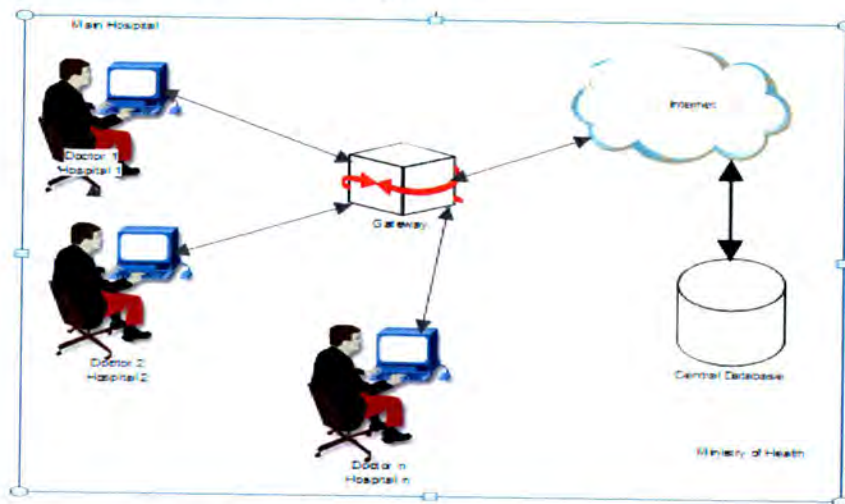


Figure 3.2: The Medical Diagnosis Framework

The design of the framework consists of four main components:

3.3.3.1 The Doctor's Interface

A Desktop/Laptop computer serves as the doctor's interface to send and receive information from the central database. In this system there are three different hospitals that are connected to the central database to send and receive information. In order for each doctor/physician in the three hospitals to send data from his/her desktop/laptop computer to the database, the data needs to pass through the gateway, which in return send that data on the network (internet) finally the internet sends it to the database. Similarly, in order for the doctor to receive any information from the database, the information needs to pass through the internet to the gateway and from there to the desktop/laptop computer.

3.3.3.2 The gateway

The gateway is a node that allows physicians to gain entrance into a network on the Internet. The gateway serves as the bridge between the Internet and the doctor's interface (computer). In

order for the computer to receive information from the central database through the internet, the gateway allows the entrance to the internet. In the same way, in order for the information to be sent in the database through the internet, the gateway is also there to link the computer to the internet. The gateway allows only doctors to have access into the network; no any unauthorized user is allowed into the network. Therefore the gateway is also used as a firewall to prevent any unauthorized person from accessing into the network since the integrity, confidentiality, and privacy of medical records is necessary required to protect the patient's information.

3.3.3.3 The Internet

The Internet is a global network that connects millions of computers. This system shows that there are three different computers in three different hospitals that are connected to the Internet in order to exchange information in the database. Here the Internet plays a crucial role because no computer can send or receive any information from/to the database nor can the database send or receive any information from/to the computer without passing through the Internet. Therefore, we can say that the internet allows the physician interfaces to communicate with the central database. The type of Internet browser that is used in each hospital is the Microsoft Internet Explorer. It is only through this that doctors/physicians are able to send or receive information in the database.

3.3.3.4 The Central database

The central database is located in the South African Health Ministry or any government agency responsible for keeping medical records. The Health Ministry ensures integrity, confidentiality and privacy of medical records. The database keeps the administrator's information, patients' information, doctors' information and the knowledge based information.

3.3.4 Role and functionality of users in the system

The system is designed to accommodate two different types of authorised users. These users are the administrator and the doctors. These two users use desktop/laptop computers as their interfaces. The administrator is the manager of the system and the doctor uses the information in the system to make a medical diagnosis decision and treat patients. The administrator first login using a correct username and password after that he is given access to the system and from there he can then create accounts, view, add and update records in the system. Similarly, the doctors also have to login with their username and password and then they can also view records, consult

and diagnose patients. The role and functionality of users in the system is shown in the Unified Modelling language (UML) diagram in Figure 3.3. It shows the role and functionality of the administrator and the doctor in the same diagram

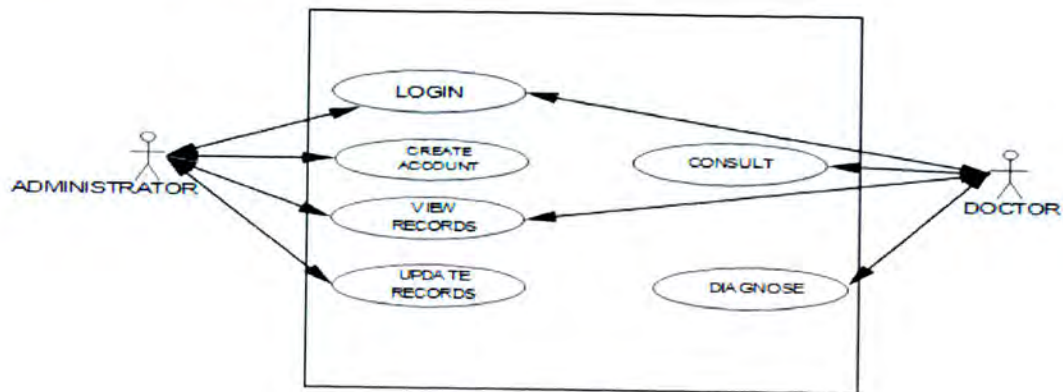


Figure 3.3: Role and functionality of users

The role or task of the administrator and the doctors in the system is explained in details in figure 3.3 above.

3.3.4.1 The administrator

The administrator is the manager of the system and is responsible for the overall maintenance of the system. His tasks involve registering new patients in the system, creating accounts for doctors and adding knowledge to the medical diagnosis system. The administrator has an account that consists of a username and a password that allows him to login as an administrator. When patients visit the hospital for the first time, the first thing the administrator does is to register them in the system and create a unique health care number for each patient as unique identification. This health care number will always be used to retrieve each patient's information in the system. Every time the patient goes to the hospital, he/she has to show his identification number that doctors can use to retrieve his/her information for consultation purposes. The administrator can view patient's information through the patient's identification number, edit patients' profiles, update patients' information and add new patients in the system. The administrator creates an account for each doctor which consists of a username and a password. This account allows the doctor to access the patients' information and the medical diagnosis records. The administrator can view, update and delete doctors' information to the system. The last task of the administrator is to add knowledge in the medical diagnostic system. For instance if there is a new disease that is to be added in the system, this will be done by the administrator,

since he is the manager of the system. He can add new knowledge and delete any record in the medical diagnostic if necessary. The administrator is limited when it comes to the diagnostics of the patients, however he can only view the patients' information but when the doctor diagnose a patient with a particular disease that information becomes confidential and only the doctors have access to that and it cannot be accessed by anyone else or the administrator, due to security, confidentiality and privacy considerations. Finally the administrator is also responsible for calculating statistics of the patients' daily, weekly and monthly appointments. Figure 3.4 illustrates the tasks that are performed by the administrator in the system, represented in the form of a flow chart.

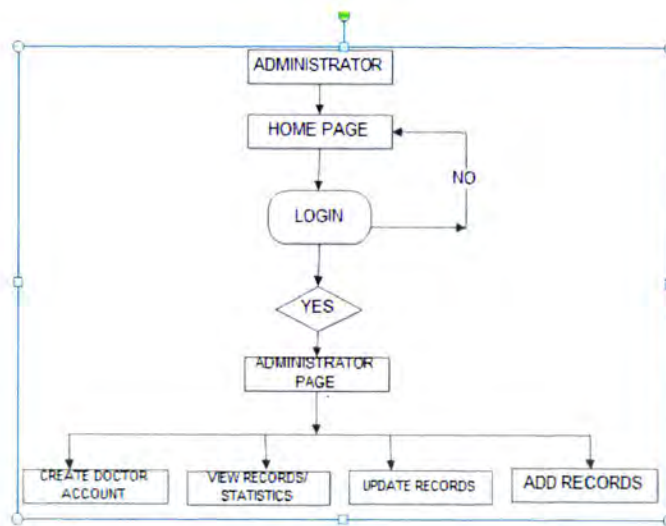


Figure 3.4: The administrator's flow chart

The flow chart above explains the tasks that the administrator performs in the system. From the home page the administrator has to log in using his username and password. If the username and password are incorrect it notifies the administrator that the information entered is incorrect, otherwise the administrator's page opens. From this page, the administrator is able to create the doctor's account, view records/statistics, update records, and add record in the system. The types of tasks performed by the administrator are explained below:

1. **Creating accounts:** This option allows the administrator to create accounts for doctors in order for them to have access to the system. These accounts consist of a username and a password. It also allows the administrator to register patients that are visiting the

hospital for the first time and create for them a unique health care number that is used as a unique identification number.

2. View record: This option allows the administrator to view the details of both the patients and the doctors. The patient's information is viewed through his/her unique identification number.

3. Add record: This option allows the administrator to add new information in the patients or doctor's accounts.

4. Update record: the administrator uses this option to update new information or modify any old information of the patients or doctors.

3.3.4.2 The doctors

The doctors are situated in the three particular hospitals connected to the central database. They are first registered in the system by the administrator and given an account that consists of a username and a password that allows them to access patients' information and the medical diagnosis record for the consultation process. When a patient meets with the doctor, the first thing that the doctor does is to retrieve the patient's information from the database through his/her unique health care number. After retrieving the patient's information, the doctor can then start with the consultation process by interviewing the patient, for example by asking the patient how he/she is feeling, and what symptoms the patient has. After getting all the complaints or symptoms from the patient, the doctor then goes to the medical diagnosis system to get knowledge and make a precise diagnosis decision by entering the symptoms shown by the patient in the system and waiting for the response from the system. If the symptoms match those in the system, then the information displays on the doctor's screen. Then the physician/doctor can then use that information to diagnose the patient. The information that the doctor gets from the system after entering the symptoms will include the name of the disease, history of that disease, more symptoms of that particular disease, treatment, prevention and prescription. After the doctor has finished consulting the patient, he enters all the information of the patient's diagnosis and next appointment with the patient in his computer and that particular information is sent to the central database through the internet. The doctors connected to the database have access to all the information stored in the system. They can view the entire patient's information and all the knowledge base information. They can add the diagnosis information of the patient in the system and ensure privacy, confidentiality and integrity of the patient's diagnosis

information. The role of the doctors in the system is represented in the form of a flow chart in Figure 3.5:

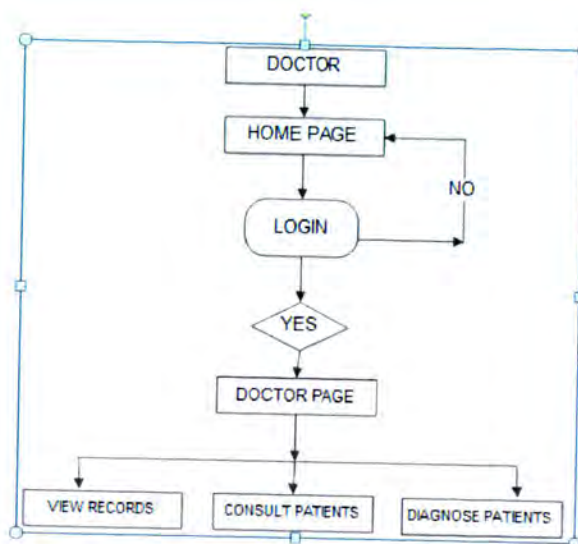


Figure 3.5: Doctors' flow chart

The flow chart above illustrates that from the home page the doctors need to provide their username and password in order for them to login in the doctor's page that allows them to view the patient's information, consult and make a diagnostic. They will only be allowed in the system if they enter the correct username and password otherwise the system refuses them access. The information available to the doctor is outlined below.

1. **View patient records:** This option allows doctors to view the patients' personal and medical information through their health care number for consultation purpose
2. **Consult patients:** When the patients go to see the doctor it is for consultation purposes. Therefore, this option allows the doctor to consult the patients through interviews and enter the patients' complains or symptoms on the system for them to get treatment.
3. **Diagnose patients:** After consultation, the doctor needs to treat the patient and make a diagnosis decision. Therefore this option allows the doctors to make a final decision on the diagnosis of the patient.

3.3.5 Relationship between the three hospitals connected to the system

The system is designed to work as knowledge based system. The information in this system is sharable among physicians so that they can make a precise diagnosis decision. We have connected three different hospitals around South Africa to share and reuse the information in this knowledge based system. The three hospitals are named as hospital 1 which is the main hospital and this is where the manager of the system, i.e. the administrator is situated. The second and third hospital are named as hospital 2 and then hospital 3. The administrator first registers all the doctors in each of the three hospitals and then creates an account consisting of username and password for each doctor that will allow them to have access to the knowledge based system. Patients in the three hospitals are also registered in the database by the administrator. Each patient has a registered unique health care number. This number is used irrespective of the environment/province where the patient is located. The unique health care number consists of a letter and a number. The letter represents the province of initial registration of the patient and the number represents a unique identification number. For example if a patient health care number is L10, L represent Limpopo province and 10 is a unique identification number of that particular patient. Another patient's health care number may be G3 or NW20, G representing Gauteng province and NW representing North West province. Patients who are located in the same province, for instance in Limpopo province will have the health care numbers with the same letter but a different number, because the number is a unique identifier for each patient. Similarly GP and NWP, patients from these provinces will also have the same letters but different numbers.

This is how the system works in such a way that if a patient who has been getting treatment in hospital 1 moves to an environment where hospital 2 is located, and wants to continue with the treatment, the doctors in that hospital will be able to retrieve that patient's information using his/her unique health care number before treating the patient. Similarly, if a patient in hospital 3 goes to hospital 1, doctors in hospital 1 should be able to retrieve that patient's information through his/her health care number to continue with the treatment where the last doctor who consulted the patient stopped. All these doctors will also be able to share knowledge among themselves. If for instance doctors in hospital 3 find a new cure, treatment or medication for a particular disease, they can also share it with hospital 1 and 2 so that it can be added in the system by the manager of the system, that is the administrator. The information that is in the

system can only be shared among doctors who are located in the provinces registered by the system. In this case, for example, if a patient being treated in Limpopo moves to a province where the hospital is not registered in the system, he/she will need to start with a new application or treatment, because doctors in that particular hospital will not be able to retrieve the medical history of that particular patient.

3.3.6 The knowledge Base

The knowledge base consists of all the information that helps physicians/doctors to share and re-use the knowledge and make a medical diagnosis decision. This information is shared by doctors in three different hospitals in South Africa. The knowledge base information is stored in a central database which is located in the South African health ministry. It also contains all the medical history of patients that are registered in the system in the three provinces. The South African health ministry ensures security, integrity, confidentiality and privacy of all the information that is in the system. The information in the system is managed by the administrator. He is responsible for adding or updating any medical information in the system if necessary. The doctors in the three hospitals have the right to make suggestions anything concerning the system. If for instance hospital 2 finds another way of treating or preventing a certain disease that is already in the system, that hospital has the right to discuss this with the two other hospitals. If they all agree, that information can then be added or updated in the system by the administrator. Another example for instance is if there is a disease that is not in the system and the three hospitals agree to add it as a new disease with all other information accompanied with it, then they have the right to do it. The information found in the knowledge base consists of:

- i. **The name of the disease:** This option consists of the name of different types of diseases such as HIV/AIDS, cancer, diabetes, malaria or Tuberculosis. When a patient goes to consult a doctor when he/she has an expectation that the doctor will be able to determine the name of the disease he is suffering from. After consultation the doctor should be able to determine the disease the patient has. This information allows the doctor to make a precise medical diagnosis and provide a solution for the patient. .

- ii. **The history of the disease:** this part gives brief information on each disease that is in the system, for instance information on how a particular disease can attack a person, where and when the particular disease was first discovered, on which continent, in which country or province that disease is mostly found, how that particular disease has affected the world or people. It states also how each disease can destroy cells in the human body. It also determines whether the disease is contagious or hereditary, and the methods by which it can be transmitted from one individual to another. The age of most people who suffer from the particular disease can also be given.
- iii. **The symptoms of the disease:** This is the most important part of our system. It is used as the primary key in the system. It is from the symptoms shown in the patient that doctors are able to determine or make a decision on the diagnosis of the patient or determine the name of the disease. When the patient is interviewed by the doctor during the consultation process, he is being asked questions on the symptoms that he experiences. After the doctor has gotten the response from the patient on the symptoms he/she is having such as fever, cough, headache, night sweat or diarrhoea, the doctor can then go into the system and enter at least three symptoms shown in the patient and then the information will display on the screen of the computer showing all the information linked to the particular symptoms. From the particular information, the doctor can now be able to diagnose the patient with a particular disease. There are many symptoms that can be linked to one disease, or there could be different diseases that can have one or two symptoms that are common. Because symptoms should not overlap, we have chosen only three major symptoms to be entered by the doctor when searching for the results to diagnose a patient. The system can contain more than three symptoms for a particular disease but if the three symptoms are entered and they match with those in the system, the results display on the system. But if the symptoms do not match then there will be no response from the system.
- iv. **Treatment of disease:** This information enables doctors to treat the particular disease that the patient is diagnosed with. After the doctor has entered the necessary symptoms in the system, the response he gets back displays the name of the disease the patient is suffering from and the treatment required or recommended to be given to the particular patient.

- v. **Prescription required:** After the patient's disease is known, the doctor needs to prescribe some medications to treat that particular disease. Therefore, this part will consist of different types of medications that can be used to treat the particular disease. From this information, the doctor/physician will know which medication to prescribe for the patient.

3.4 MySQL database design

My MySQL database is the world's most popular open source database for cost-effectively delivering reliable, high performance, scalable, online transaction processing, and embedded database applications. It is an integrated, transaction safe, ACID-compliant database with full commit, rollback, crash recovery, and low-level locking capabilities. MySQL delivers the ease of use, scalability, and high performance, as well as a full suite of database drivers and visual tools to help the developers to build and manage their MySQL applications. The MySQL database provides the following features:

- i. **High performance and Scalability:** To meet the demands of exponentially growing data loads and users.
- ii. **Self-healing Replication Clusters:** To improve scalability, performance and availability
- iii. **Online schema change:** To meet changing business requirements.
- iv. **Performance schema:** For monitoring users and application level performance and resource consumption.
- v. **SQL and NoSQL Access:** For performing complex queries and simple, fast key value operations.
- vi. **Platform independence:** To give users flexibility to develop and deploy on multiple operating system.
- vii. **Big data interoperability:** Using MySQL as the operational data store for Hadoop and Cassandra.

The MySQL database design consists of MySQL Workbench and the MySQL database tables that are explained in detail as follows:

3.4.1 MySQL Workbench

MySQL Workbench is a unified visual development and administration platform that includes advanced tools for database modeling and design and testing, server configuration and monitoring, user and security administration and wizard-driven database migrations. MySQL Workbench provides a graphical tool for working with MySQL servers and databases. It provides three main areas of functionality:

- i. **SQL Development:** Enables users to create and manage connections to database servers and configure connection parameters. MySQL Workbench provides the capability to execute SQL queries on the database connections using the built-in SQL Editor.
- ii. **Data Modeling:** Enables the user to create models of the database schema graphically, reverse and forward engineer between a schema and a live database, and edit all the aspects of the database using the comprehensive Table Editor. The Table Editor provides easy- to-use facilities for editing tables, columns, indexes, Triggers, partitioning, options, inserts and privileges, routines and views.
- iii. **Server Administration:** Enables users to create and administer server instances.

The Enhanced Entity Relationship (EER) model created by MySQL Workbench is shown in the figure below:

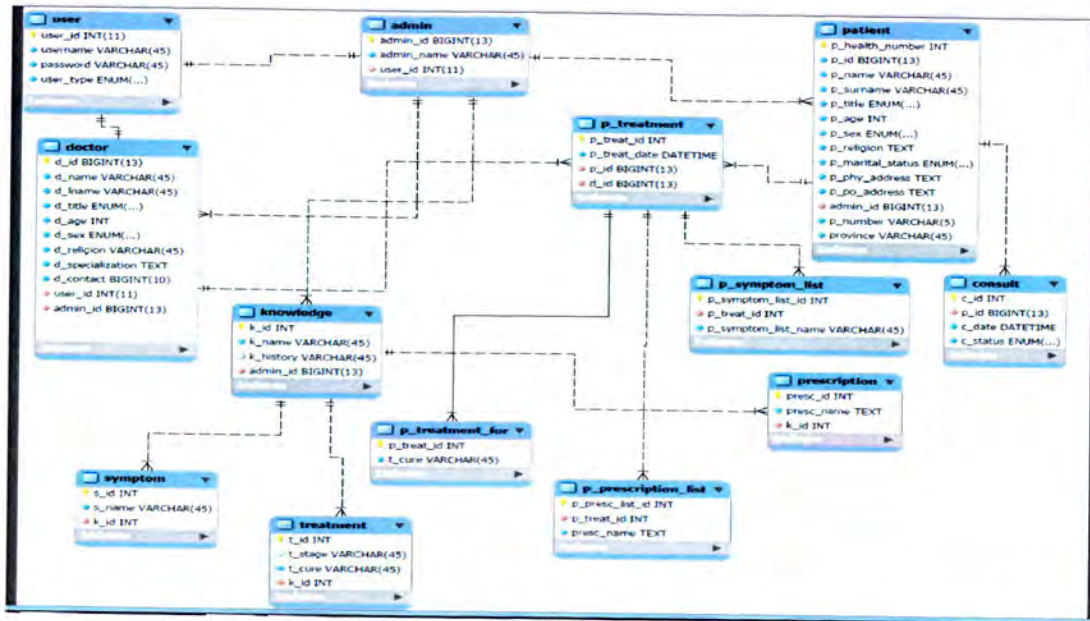


Figure 3.6 EER - Diagram

The Enhanced Entity Relationship (EER) model is a high-level or conceptual incorporating extension to the original entity-relationship (ER) model, used in the design of databases. The EER model is created, edited and exported in MySQL Workbench. Therefore the EER diagram shown in Figure 3-6 shows the whole database design. It shows the relationship between different entities. In the diagram we can see that the administrator creates doctors' accounts and registers patients, and adds information in the knowledge base, while doctors use the knowledge base to diagnose and treat patients. The diagram shows that each entity type (subclass) is an entity of another entity type.

3.4.2 MySQL Database Tables

The MySQL database tables are all tables which were designed in this project. The tables below are some of the MySQL database:

3.4.2.1 Login table

Table 3.1 the login table

	user_id	username	password	user_type
▶	1	Blandine	hHkox08KZAmJMLMPijj0Y/5Sjo=	admin
	2	Mbodila	GWbmlLrZBoZFRb5nN9DKAD9yjkpA=	doctor
	3	Van Nierkerk	ZKt/7dQS0ibMJLzz78mk6CHJWtQ=	doctor
	4	Coppessam	c35gvNoy2glQbZSwXrkzwmZNIvk=	doctor
	5	Ekabua	OIG0wmKOPmMhzfYi/TfsZikxoVk=	doctor
	6	Preddie	b1Bn/gdTsvNfTjOskRnWHA4/IV0=	doctor
	7	Abudalay	L90C+PEIKsWZbslhyP9m1iaVsRw=	doctor
*	NULL	NULL	NULL	NULL

The login table stores the username and password that allow the administrator and doctors to login in the system. Here the password is encrypted as a security measure to stop unauthorized users from accessing information in the database. The administrator is the one who opens accounts for doctors, and at first he will know their passwords, but if any doctor decides to change his password, that password becomes encrypted and even the administrator, the manager of the system, would not be able to view it, since security in this system is very high. As shown in the table above, the login table consists of usernames and passwords of two different types of users, namely the administrator and the doctors.

3.4.2.2 Administrator table

Table 3.2: Administrator table

	admin_id	admin_name	user_id
▶	811605	Muhandji	1
*	NULL	NULL	NULL

The administrator table stores the administrator's name and identity number and the user identity, which in this case is 1 since there is only one administrator. This is the information that the administrator supplied for first registration on the system. After filling in the required information such as his name and identity number, he had to create a username and password so that he could login the system.

3.4.2.3 Doctors table

Table 3.3: Doctors table

d_id	d_name	d_lname	d_title	d_age	d_sex	d_religion	d_specialization	d_contact	user_id	admin_id
82209	Freddie Mkwim	Moodia	Dr	25	female	christian	pediatrician	151567393	6	811605
571604	Mohammed	abudulay	Dr	56	male	muslim	generalist	740954231	7	811605
651711	Nadia	Coppessam	Dr	43	female	Hindou	Neurologist	848923154	4	811605
702106	obeten	Okreua	Dr	42	male	christian	surgeon	820403265	5	811605
781010	Munvenge Mony	Moodia	Dr	35	male	christian	gynaecologist	726721476	2	811605
831204	Ebenne	Van Nierkerk	Dr	30	male	christian	urologist	837054236	3	811605
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The doctor's table stores the information of each doctor in the system. This consists of the doctor's identity, name, surname, title, age, sex, religion, specialization, contact number, and the administrator's identity since he is the one who creates all the doctors' accounts.

3.4.2.4 Patients table

Table 3.4: Patients table

p_health_number	p_id	p_name	p_surname	p_title	p_age	p_sex	p_religion	p_marital_status	p_phy_address	p_po_address	admin_id	p_number	province
7	290846837	Nene	Awa	Mrs	29	female	Muslim	Single	80, leue street/Mokong flat unit 19/Fuatenburg	PO BOX 3976/Fuatenburg030	811605	NW7	NorthWest
8	3509053489	Tshebale	Gumede	Mr	35	male	buddist	Married	3, krogth Emeniesicoveto	PO Box 5908/Soweto0980	811605	GP8	Gauteng
1	1019050366	gacdi	Moodia	Mr	3	male	christian	Single	57, cellers street/let no 20/ous treant	P.O BOX 4326/ous richard0320	811605	L1	Impopo
4	452097656	Natasha	Nardoo	Mrs	47	female	buddist	Married	6, chakode street/Santopretoria	Private Bag X2908/Pretoria 0256	811605	GP4	Gauteng
6	8320120987	akuwani	Tshikoro	Ms	30	female	muslim	Single	59, Plesk street/Turkop/Pobokwane	PO BOX 4058/Pobokwane0940	811605	L6	Impopo
10	8912109865	Mulaudi	Tshikoro	Mr	24	male	christian	Single	5, krogth street/jobeth flats unit 34/Mokopane	P.O BOX 5667/Mokopane0970	811605	L10	Impopo
3	8612083456	Nosoph	deudu	Ms	27	female	christian	Single	90, Protea street unit 17, the willows flats/Melkang	P.O BOX 2341/Melkang 1010	811605	NW3	North-West
5	90121109876	Ruth	kadeng	Mrs	23	female	christian	Married	37, long market street unit 21, zheyandime flats/Thohoyendou	Private BAG X5050/Thohoyendou0925	811605	NW5	North-West
2	22807908	tembo	kikungu	Ms	11	female	christian	Single	45, beverly road/Etsevele Jochenesturg	P.O BOX 6907/Jochenesturg2245	811605	GP2	Gauteng
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The patient's table stores the information of all the patients in the three hospitals that are connected to the system. This consists of the patient's health care number, this unique number is generated by the system as soon as the administrator finishes registering the patient's identity number, name, surname, title, sex, marital status, religion, physical address, postal address, the

administrator's identity since he is the one registering all the patients in the system, and the province where the patient is located.

3.4.2.5 Knowledge Base table

Table 3.5: Knowledge Base table

	k_id	k_name	k_history	admin_id
▶	1	Malaria	-start from a bite of infected mosquitoes	811605
	2	Typhoid	Eating contaminated food	811605
	3	Cholera	eating food and drinking water contaminated	811605
	4	cancer	growth of cells in the body	811605
	5	Tuberculosis (TB)	coughing for more than 2 weeks	811605
	6	HIV/AIDS	sexual contact	811605
*	NULL	NULL	NULL	NULL

The knowledge table consists of the name of the disease and the history of each disease. These are the two types of information that are entered first. This is followed by other information such as symptoms, treatment and prescription which are stored in their own table, but linked to this table, and finally the administrator's identity since he is the one who enters the information in the knowledge base.

3.5 Chapter Summary

This chapter presented the design of the framework and the analysis requirement such as hardware and software requirement. It described in details the hardware and software that were used. The personal computer and its details were used as hardware requirement and MySQL and Java programming language were used as software requirement. In this chapter we also explained how the data was collected in order to design the framework. We mentioned that this framework was designed as three tier architecture and we explained how it works in detail and we also presented the MySQL Workbench Enhanced Entity Relationship Model and the MySQL database tables. The administrator and doctors tasks in the system were also shown in the form of a flow chart using the Unified Modeling Language diagram.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 Chapter Overview

This chapter presents the actual implementation of the system. Here we present and discuss all the interfaces and results obtained from the implementation. Based on the requirements and design discussed in the previous Chapter, the system's interfaces show how the system can be manipulated to achieve the desired objectives. We describe the results and the interfaces in detail for better understanding of the basic operation of the medical diagnosis system and they are discussed in subsequent sections.

4.2. Medical Diagnosis System's Operations

In this section, we give detailed explanation of the system's operations describing the individual interfaces that make up the entire system. In this system, only two users are identified: the Doctor and the Administrator. They are the only users charged with the responsibility of using the system with the functionalities defined, in order to ensure confidentiality of patient's information.

4.2.1 The Registration Page

The screenshot shows a registration form with the following fields and values:

Field	Value
Your name	MURAD
Your id number	0123456
Username	Murad
password	****

Buttons: Fresh, Clear, close

Figure 4.1: Registration page

In our system, the registration page shown in Figure 4.1 is the first page which consists of the information that the administrator needs to complete in order for him to be registered in the system. It consists of the name, identity number, a username and the password. After the

administrator has finished completing this form, an account is created for him to be able to login to the system using the username and password.

4.2.2 The login Page



Figure 4.2: The login page

The login page (Figure 4.2) is the second page that appears after the administrator has created an account using the first page. Two fields are involved: the username and password which upon authentication, opens the administrator's page that allows the administrator to perform his/her actual tasks such as *registering patients*, *creating accounts for doctors* and *entering knowledge-based information* (see Figure. 4.3). After the administrator has created accounts for the doctors, a separate login page is created that allows them to authenticate themselves before giving access to the system. Based on the design of our system, no physician/doctor is allowed access to the system without a valid username and password. The login page is used as the front page of the system. It is used by both the administrator and the doctors with valid usernames and passwords.

4.2.3 The Administrator Page



Figure 4.3: The administrator page

In the administrator's page, several functionalities exist such as *doctors' accounts*, *patients' account*, *knowledge base* and *statistics*. Based on the functionalities, it allows the administrator to create accounts for doctors and patients, enter information in the knowledge base, and view the statistics of all the patients that were consulted by any doctor on a particular date. He/she can also view the lists of all patients and doctors after registering them in the system and update any information of doctors, patients or in the knowledge base and finally search for doctors or patients by their names, surname or health care number. The interfaces for the above functionalities within the administrator's page are discussed below:

4.2.3.1 Creation of doctor's account



Figure 4.4: Creation of doctor's account

Figure 4.4 shows how the administrator creates doctors' accounts. To create an account, he/she first clicks on the doctors' account button which displays a form with different fields where information such as username, surname, first name, title, age, gender, religion, identity number, the specialization of the doctor, password and so on can be entered. (see Figure 4.5) After completing the form, the CREATE button is clicked to create an account for the particular doctor. Figure 4.5 displays a doctor's profile resulting from Figure 4.4. The information in white and orange fields can be updated or changed while those in green coloured fields cannot be changed or updated.

4.2.3.2 The results of creation of doctor's account

Blondine (admin)	
Doctor Accounts Patient Accounts Knowledge Base Statistics	
<input type="text" value="write Doctor"/> <input type="button" value="All Doctors"/> <input type="button" value="Search"/> <input type="button" value="Change Password"/>	
Doctor Username	M0000
Name	Muhammad Nizam
Surname	M0000
Title	Dr
ID Number	791019
Age	28
Gender	Male
Religion	Islam
Area of specialization	Emergency
Contact number	001234567
<input type="button" value="close"/> <input type="button" value="logout"/>	

Figure 4.5: Results of creation of doctor's account

The figure 4.5 displays the information of the doctor after the creation of his/her account. After the administrator has completed filling the form by entering all the necessary information for the doctor's account, he clicks on create and the Figure above displays showing all the information provided.

4.2.3.3 Searching for doctors

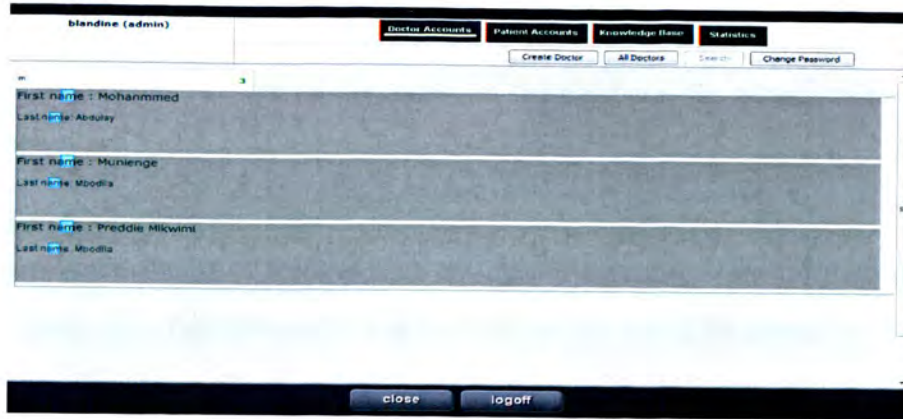


Figure 4.6: Searching for doctors' accounts

The search button in the doctors' account option, allows the administrator to search for any doctor stored in the database. If for instance the letter m is typed in the search textbox, all the doctors' first names or last names starting with the letter m display, then the administrator chooses the name of the particular doctor he is searching for and clicks on that to get his/her information. In Figure 4.6, number 3 that appears on the textbox together with the letter m indicates that there are three doctors that have either first names or last names starting with the letter M.

4.2.3.4 Viewing the list of doctors

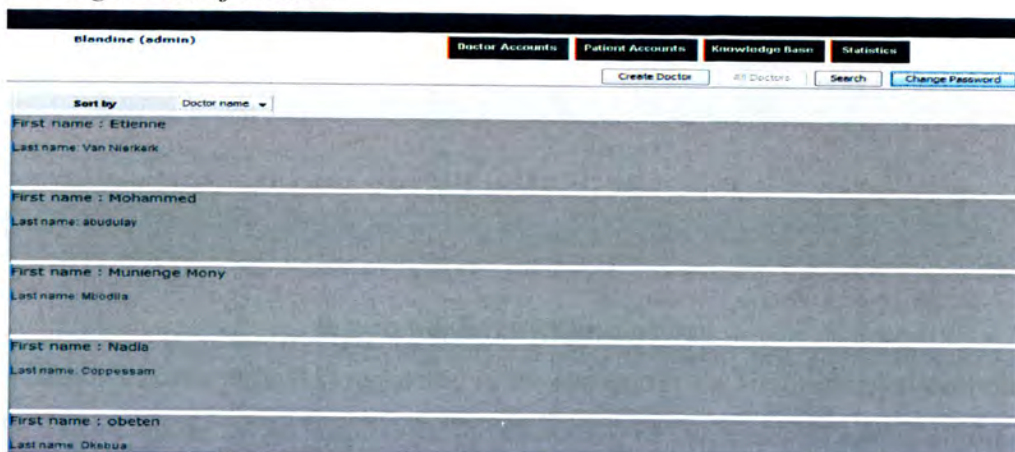


Figure 4.7: Viewing the list of Doctors

Figure 4.7 displays the names and surnames of the doctors that are stored in the system. To view the lists of doctors in the database, the administrator will click on the all doctors button to display the list of all the doctors. To see all the names in the database, the administrator has to scroll down the page.

4.2.3.5 Patient Accounts

The patient accounts consist of features such as *register a patient*, *view all patients* and *search patient* (see Figure 4.8). All these three functionalities allow the administrator to either register a patient for the first time, view all the patients stored in the system or search for a particular patient by his first and last name or healthcare number. The tasks performed by the administrator in the patient accounts are shown in Figure 4.8.

4.2.3.6 Registering patients

The screenshot shows a web application interface for a hospital system. At the top, there is a navigation bar with the user name 'Blandine (admin)' and four tabs: 'Doctor Accounts', 'Patient Accounts', 'Knowledge Base', and 'Statistics'. The 'Patient Accounts' tab is selected. Below the navigation bar, there are three buttons: 'Register a Patient', 'View all Patients', and 'Search a Patient'. The main form is divided into several sections. On the left, there are input fields for 'ID Number' (1019050966), 'Surname' (Mbodla), 'Title' (Mr), and 'Gender' (with radio buttons for 'Female' and 'Male', where 'Male' is selected). Below these are two text areas: one containing 'christian' and another containing '57, celliers street, flat no.20, louis trichardt'. At the bottom left of this section is a 'Register' button. On the right side of the form, there are dropdown menus for 'Province' (Limpopo) and 'Marital Status' (Single), and input fields for 'Name' (greddi) and 'Age' (3). At the bottom of the form, there are two buttons: 'close' and 'logout'.

Figure 4.8: Registering a patient

Figure 4.8 shows how a patient is registered in the system for the first time by the administrator. When the patient comes to the hospital for the first time, he/she needs to supply all the required information to complete the form. The form also states the province where the patient is located. As mentioned earlier that this system is designed to be used by three provinces of South Africa

which are Limpopo, Gauteng and North-West. After the administrator has finished with the registration, he then clicks on register and the information in Figure 4.9(a) and 4.9 (b) below displays:

4.2.3.7 Results of patient Registration

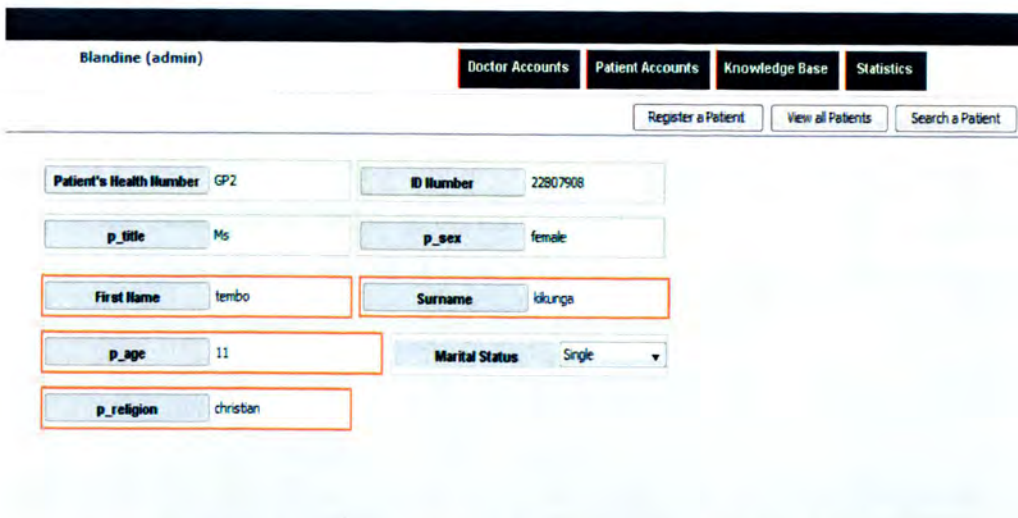


The screenshot shows a web interface for a healthcare system. At the top, there is a navigation bar with the user name 'Blandine (admin)' and four menu items: 'Doctor Accounts', 'Patient Accounts', 'Knowledge Base', and 'Statistics'. Below the navigation bar, there are three buttons: 'Register a Patient', 'View all Patients', and 'Search a Patient'. The main content area displays the registration details for a patient in Limpopo. The fields are as follows:

Patient's Health Number	L1	ID Number	1019050966
p_title	Mr	p_sex	male
First Name	gradi	Surname	Mbodla
p_age	3	Marital Status	Single
p_religion	christian		

Figure 4.9(a): Results of patient registration in Limpopo

Figure 4.9(a) displays the results of the patient after being registered in the system by the administrator. It displays the health care number L1 of which the letter L is the name of the province (Limpopo), and number 1 which is a unique number that identify the patient.



The screenshot shows the same web interface as Figure 4.9(a), but for a patient registered in Gauteng. The fields are as follows:

Patient's Health Number	GP2	ID Number	22807908
p_title	Ms	p_sex	female
First Name	tembo	Surname	ikunga
p_age	11	Marital Status	Single
p_religion	christian		

Figure 4.9(b): Results of patient registration in Gauteng

Figure 4.9 (b) displays another patient form with a healthcare number GP2 of which GP represent the province of Gauteng and 2 represent the unique number that identify the patient. Patients that are in the same environment or province will have the letter such as L, GP or NW to represent the province where they are located but each one of them will have a unique number that only represents them.

4.2.3.8 Viewing all Patients

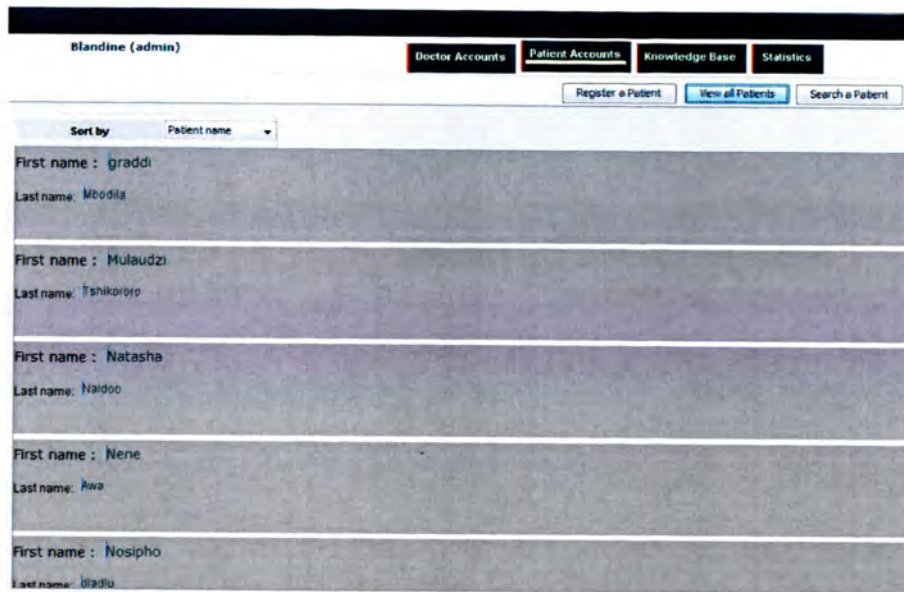


Figure 4.10: Viewing all patients

Figure 4.10 displays the lists of patients that are stored in the system. If the administrator wants to view the list of all patients, he clicks on the view all patients' option under patient Accounts, then the list of all patients displays. If he decides to view the information of any patient, he then clicks on that particular patient and all his/her information will display. Here the list of patients can be sorted by patient's name, surname or health care number.

4.3.2.9 Searching for patient



Figure 4.11(a): Search for patients using first name or surname

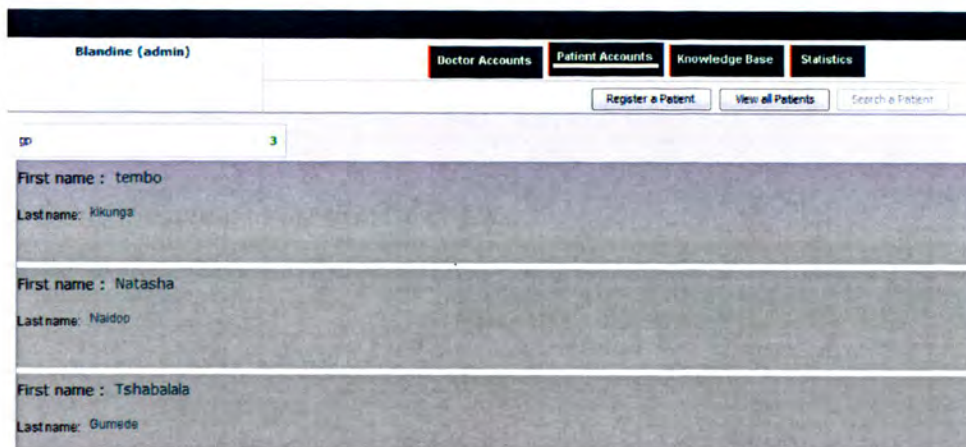


Figure 4.11(b): Searching for patients according to provinces

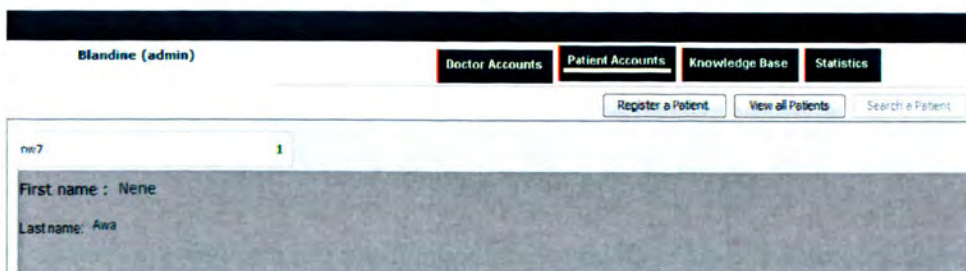


Figure 4.11(c): Search for a patient using his health care number

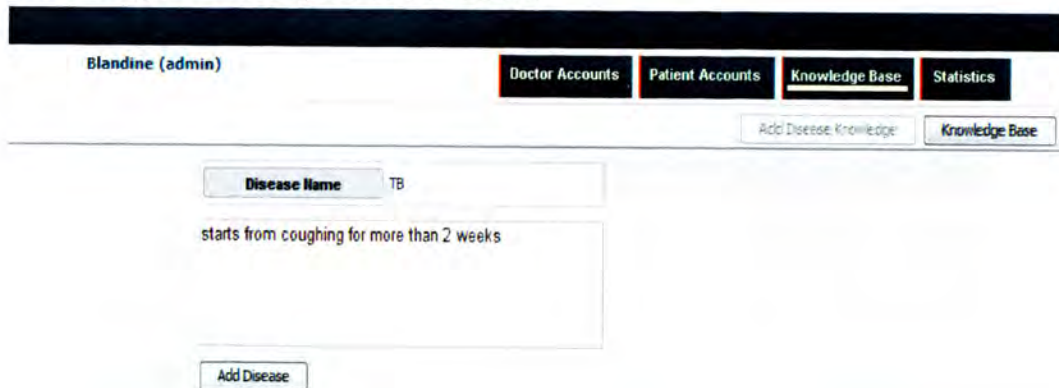
Figures 4.11(a), 4.11(b), and 4.11(c)) show how patients can be searched for in the system. Figure 4.11(a) searches for patients according to their first name or surname having N as starting letter or wherever there is an N in their names and number 4 in the search box shows that there

are only 4 patients with names or surnames having the letter N. Figure 4.11(b) searches for patients in a particular province which is Gauteng Province (GP) and there are only 3 patients found in GP. Finally Figure 4.11(c) searches for patients according to their health care numbers which consist of letters and numbers. As shown in the Figure 4.11(c) there is only one patient found in the Northwest province and his/her health care number is NW7. As mentioned earlier, there cannot be two or more patients having the same health care numbers.

4.3.2.10 Knowledge Base

The knowledge base (KB) functionality consists of two features which are *adding diseases knowledge* and view lists of diseases that are stored in the knowledge base. If the administrator clicks on the add diseases knowledge, it displays a form where disease information can be entered, such as the name of a disease, history, symptoms, treatment and prescription for that particular disease and if he clicks on knowledge base button he can view all the type of diseases that have been stored in the database. Figure 4.12 below shows how the information about the disease is entered in the system.

4.3.2.11 Entering the name of the disease in KB



The screenshot shows a web application interface for an administrator named 'Blandine (admin)'. At the top, there are navigation tabs: 'Doctor Accounts', 'Patient Accounts', 'Knowledge Base' (which is selected), and 'Statistics'. Below the tabs, there are two buttons: 'Add Disease Knowledge' and 'Knowledge Base'. The main content area contains a form with a 'Disease Name' input field containing 'TB', a text area containing 'starts from coughing for more than 2 weeks', and an 'Add Disease' button at the bottom.

Figure 4.12: Adding disease name and history

Figure 4.12 shows how information about any disease and its history is entered in the system. As shown, we can see TB is the name of the disease and the next text box shows its history. After all the information is entered then the *add disease button* is clicked in order to go to the next page shown in Figure 4.13.

4.2.3.12 Adding Symptoms of the disease

Blandine (admin) Doctor Accounts Patient Accounts Knowledge Base Statistics

Add Disease Knowledge Knowledge Base

Symptom 1 heavy cough

Symptom 2 shortness of breath

Symptom 3 chest pain

Symptom 4 fatigue

Symptom 5 vomiting blood

Add more

Save and continue

Figure 4.13: Adding the disease symptoms

After entering the name of the disease and its history, the next page appears which allows us to enter different symptoms that are linked to the disease as it is shown in Figure 4.13. There are different types of symptoms for TB. There are two buttons; the *Add more button* which allows us to add more symptoms if we want and the *save and continue button* which allows us to save our information and continue to the next page.

4.2.3.13 Adding treatment of the disease

Blandine (admin) Doctor Accounts Patient Accounts Knowledge Base Statistics

Add Disease Knowledge Knowledge Base

Treatment 1 skin test Stage 1

Treatment 2 chest x-ray Stage 2

Treatment 2 selva test Stage 3

Save and Continue

Figure 4.14: Adding disease's treatment

Figure 4.14 shows how TB can be treated in each stage. When entering the information about the treatment of TB, it also requires the stages of TB treatment to be entered.

4.2.3.12 Adding prescription of the disease

Blandine (admin) Doctor Accounts Patient Accounts Knowledge Base Statistics

Add Disease Knowledge Knowledge Base

Prescription 1 acetaminophen

Prescription 2 etambutol

Prescription 3 rifampin

Prescription 4 pyrazinamide

Add more

Save and continue

Figure 4.15: Adding the prescription of the disease

Figure 4.15 displays different prescriptions of TB. Here more prescriptions can be added if the add more button is clicked.

4.2.3.13 Viewing the list of diseases in the KB

Blandine (admin) Doctor Accounts Patient Accounts Knowledge Base Statistics

Add Disease Knowledge Knowledge Base


Malaria
Typhoid
Cholera
cancer
Tuberculosis (TB)
HIV/AIDS

Figure 4.16: Viewing the list of diseases in the knowledge Base

Figure 4.16 displays the list of diseases in the knowledge Base. Each disease is linked with information such as history, symptoms, treatment and prescription of the particular disease. As soon as a particular disease is clicked, the information attached to the name of the disease

displays. For instance if the doctor clicks on the name of the disease Tuberculosis (TB), the information shown on Figure 4.17 will be displayed.

4.2.3.14 Viewing the information of the disease in the KB



Blandine (admin)		Doctor Accounts	Patient Accounts	Knowledge Base	Statistics
				Add Disease Knowledge	Knowledge Base
Disease Name	Tuberculosis (TB)	Symptom	heavy cough		
Disease History	coughing for more than 2	Symptom	shortness of breath		
<input type="button" value="Update"/>		Symptom	night sweats		
		Symptom	fatigue		
		Symptom	chest pains		
		Treatment	skin test		
		Treatment	chest x-ray		
		Treatment	saliva test		
		Prescriptions	acetaminophen		
		Prescriptions	etambutol		
		Prescriptions	rifampin		
		Prescriptions	pyrazinamide		

Figure 4-17: Viewing the information of a disease in the KB

Figure 4.16 displays the information linked to the name of the disease Tuberculosis and this is how all the diseases in the knowledge base are classified and this is the information that doctors use in order to diagnose and treat patients. It shows the name of the disease, the history, symptoms, treatment and prescription of the disease.

4.2.4 Doctors' Page

As mentioned earlier, there are two users of this system. These users are the administrator and the doctors. We have already explained the task of the administrator and now we shall to explain the task of the doctor in this system. From the login page, the doctor needs to enter his username and password. If they are valid, then the doctor's page displays showing the functionalities such as doctor's profile, consult, and change password. The three options in the doctor's page are described in subsequent sections.

4.2.4.1 The Doctor's profile

The doctor's profile displays information such as the doctor's full Name, identity number, religion, contact number and specialization. Some of the information such as age and sex are stored in the database and are not visible in the profile. Figure 4.18 displays the doctor's profile.

Preddie (doctor)		Doctor	Consult
		My Profile	Change my password
My ID Number	82209		
Full Name	Dr Mbodila Preddie Mikwimi	Specializing In	pediatrician
Religion	christian		
Contact	151567393		

Figure 4.18: Doctor's profile

4.2.4.2 Consultation

Preddie (doctor)		Doctor	Consult	LOGOUT
gp#	1	Personal Details:		
Patient's Health Number	GP4			
Patient Name	Mrs Naidoo Natasha			
ID Number	4620097656			
Patient Marital Status	Married			
		Last treatment Mrs Naidoo		
Patient does not have treatment history.				
Enter Symptom Name	Being treated for			
	Symptom	Prescription		
	More Symptoms	More Prescriptionw		
	Treat Patient	Reset		

Figure 4.19: Consulting a patient

The consult feature allows doctors to view and document patients' information during consultation. Firstly, the doctor has to search for the patient by his/her health care number with the result displaying as in Figure. 4.19. The result shows that Doctor Preddie wants to consult patient Mrs Naidoo Natasha by first searching for her health care number which is GP4. It shows the patient's location which is Gauteng province, her identity number and marital status and also her medical history. Here it shows that the patient has no treatment history since it is the first time that has come for consultation. In the figure (4.19), there is also an option that allows the doctor to search for information in the KB in order to consult the patient.

4.2.4.3 Changing password



The screenshot shows a user interface for a doctor named Preddie. At the top, there is a navigation bar with the user's name 'Preddie (doctor)' on the left and two buttons, 'Doctor' and 'Consult', on the right. Below the navigation bar, there are two buttons: 'My Profile' and 'Change my password'. The 'Change my password' button is highlighted. Below these buttons, there are three input fields for passwords, each with a label and a set of dots representing the password characters. The labels are 'Old Password', 'New Password', and 'Confirm Password'. Below the 'Confirm Password' field, there is a 'Confirm' button.

Figure 4.20: Changing password

The change password option allows doctors to change their passwords if they wish to. When the administrator creates doctors' accounts he provides them with a username and password that they use to login in the system. After the accounts have been created, doctors can change their passwords for confidentiality, integrity and privacy. In order for a doctor to change his password, he first needs to enter his old password then the new password that he wants, after that he confirms the new password, which has to match with the first new one, and he finally confirms. After the doctor has changed his password, even the administrator cannot have access to his account anymore.

4.3 Results

Our research objectives were to design a framework and a medical diagnosis system that will assist doctors in making a precise medical decision. The framework in Figure 3.2 has been shown and explained in chapter three and the following are the results of how the system assists doctors to use the information in the knowledge base for them to diagnose and treat patients.

4.3.1 Consulting a patient for the first time

The screenshot shows a web interface for a doctor named Van Nierkerk. At the top, there are buttons for 'Doctor' and 'Consult'. Below this, the interface displays patient information for 'Ms Kikunga tembo'. The patient's health number is GP2, ID number is 22807908, and marital status is Single. The last treatment recorded is 'Ms Kikunga'. A message states 'Patient does not have treatment history.' At the bottom, there is a section for 'Being treated for' with input fields for 'Symptom' and 'Prescription', and buttons for 'More Symptoms', 'More Prescription', 'Treat Patient', and 'Reset'.

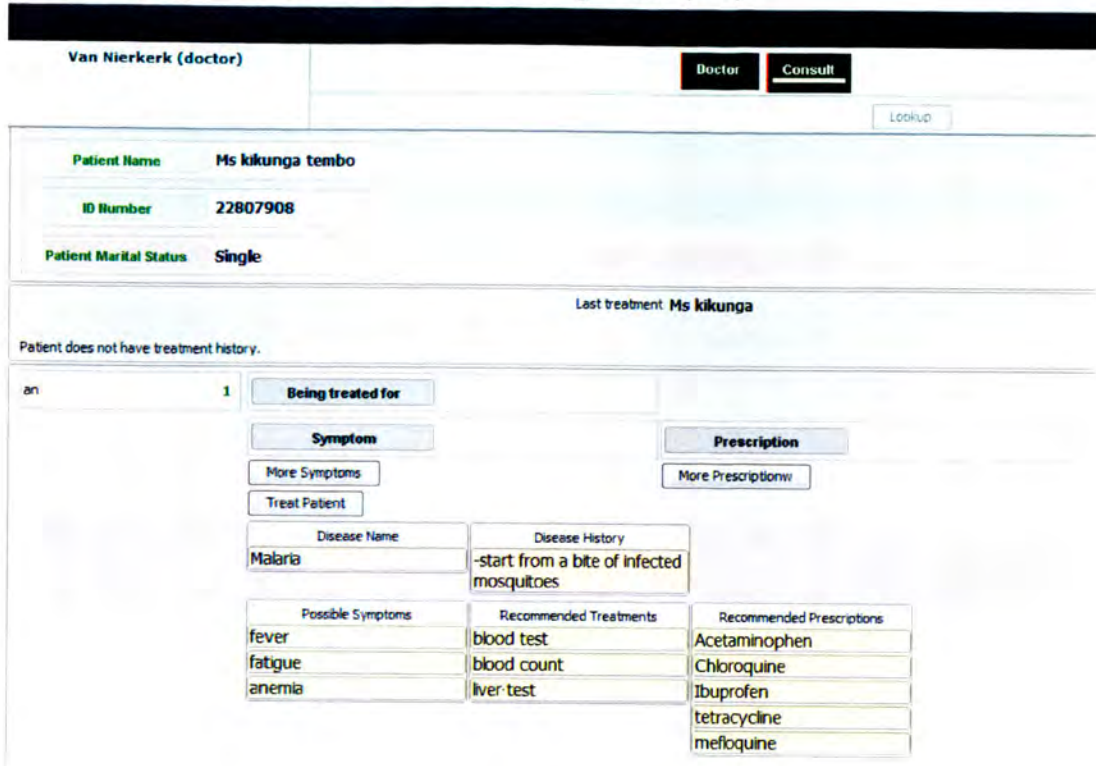
Figure 4.21: Consulting patient for the first time

When a patient comes to the hospital for consultation, he first needs to open a file and then he can be able to see the doctor. As mentioned earlier in chapter 3 on the framework design (3.3.3.4 central database server), our system keeps the information of patients, doctors, the administrator and medical information that assists doctors to treat patients. Therefore, when a patient comes for the first time to the hospital, he needs to be registered in the system and be given a unique health care number that he will always show every time he comes for consultation. The patient's healthcare number is used as his unique identification for the doctor to retrieve his personal details and medical history.

As shown in the Figure 4.21, Doctor Van Nierkerk is consulting patient Tembo Kikunga for the first time. The doctor first retrieves the patient's information through her health care number which shows that she is located in Gauteng Province. The patient's information displays showing

her personal details and treatment history. In this case the patient has no treatment history. It also shows how the doctor can use the information in the knowledge base to treat the patient.

4.3.2 Searching for information in the Knowledge Base (KB)



Van Nierkerk (doctor) Doctor Consult

Lookup

Patient Name Ms Kikunga tembo

ID Number 22807908

Patient Marital Status Single

Last treatment: Ms Kikunga

Patient does not have treatment history.

an 1 Being treated for

Symptom **Prescription**

More Symptoms More Prescriptionw

Treat Patient

Disease Name	Disease History	
Malaria	-start from a bite of infected mosquitoes	

Possible Symptoms	Recommended Treatments	Recommended Prescriptions
fever	blood test	Acetaminophen
fatigue	blood count	Chloroquine
anemia	liver-test	Ibuprofen
		tetracycline
		mefloquine

Figure 4.22: Searching for information in the KB

After the doctor has interviewed the patient and obtained all the necessary information, for instance the symptoms that the patient described such as headache or fever, the doctor then enters three main symptoms in the text box below “patient does not have treatment history” as shown in Figure 4.22, to get the information in the KB for the diagnosis.

4.3.3 Treating patient using information in the KB

Van Nierkerk (doctor)

Patient Marital Status Single

Last treatment: **Ms kikunga**

Patient does not have treatment history.

an	1	Being treated for Malaria		
		Symptom fever		Prescription Acetaminophen
		Symptom fatigue		Prescription chloroquine
		Symptom anemia		Prescription tetracycline
		Symptom vomiting		Prescription Ibuprofen
		<input type="button" value="More Symptoms"/>		<input type="button" value="More Prescription"/>
		<input type="button" value="Treat Patient"/>		

Disease Name	Disease History	Recommended Prescriptions
Malaria	-start from a bite of infected mosquitoes	Acetaminophen
fever	blood test	Chloroquine
fatigue	blood count	Ibuprofen
anemia	liver test	tetracycline
		mefloquine

Figure 4.23: Treating the patient using information in the KB

After getting the diagnosis from the KB in figure 4.22, the doctor now needs to use that information to treat the patient. In Figure 4.23 the doctor enters the name of the disease that he has diagnosed the patient for, the symptoms presented by the patient and the prescription required for the particular disease and then clicks on treat patient. The Figure 2.24 shows how to treat a patient using the information in the KB.

4.3.4 The results of the treatment

Van Nierkerk (doctor) Doctor Consult

GP2 1 Lookup

Personal Details:

Patient's Health Number: GP2

Patient Name: Ms Kikunga tembo

ID Number: 22807908

Patient Marital Status: Single

Last treatment: Ms Kikunga

Last treated on: 2013-08-17 03:02:55

by: Dr Van Nierkerk

Treated for: Malaria

Diagnosed Symptoms	Given Prescriptions
fever	Acetaminophen
fatigue	chloroquine
anemia	tetracycline
vomiting	ibuprofen

Enter Symptom Name: Being treated for

Figure 4.24: The results of the treatment

After the doctor treats the patient the results of the treatment are stored in the system so that the next time the patient comes for consultation the doctor on duty that day will be able to see the date of last treatment and the name of the doctor who treated the patient. Figure 4.24 shows the results of treatment. It shows the last date that the patient was treated, the name of the doctor, the name of the disease treated for, the diagnosed symptoms and prescription given.

4.3.5 The next Consultation

Ekaba (doctor) Doctor Consult

Last treatment: Ms Kikunga Lookup

Last treated on: 2013-08-17 03:02:55

by: Dr Van Nierkerk

Treated for: Malaria

Diagnosed Symptoms	Given Prescriptions
fever	Acetaminophen
fatigue	chloroquine
anemia	tetracycline
vomiting	ibuprofen

10 2 Being treated for Typhoid

constipation Ampicillin

stomach pains ceftriaxone

loss of appetite ciprofloxacin

vomiting

Disease Name	Disease History	Recommended Prescriptions
Typhoid	Eating contaminated food	Ampicillin
constipation	blood test	Ceftriaxone
stomach pains	stool test	Ciprofloxacin
loss of appetite	bone marrow	

Figure 4.25: The next consultation day

Figure 4.25 shows that the patient has been treated for a particular disease and it also shows the date and the name of the doctor that treated the patient. The next treatment will be when the

patient goes to the hospital for consultation again. Any doctor who will be on duty in the patient's next consultation will be able to view the last treatment given to the patient and consult the patient again for any disease. The figure 4.25 shows the next consultation of patient Kikunga. On top it shows the date of the last treatment by Dr Van Nierkerk and this time the patient is going to be treated by a different doctor who is Dr Ekabua. This doctor will still use the information in the knowledge base to treat the patient and at the same time he is able to see the last treatment given to the patient and the name of the doctor who treated the patient.

4.3.6 The results of the next treatment

The screenshot shows a web interface for a doctor named Ekabua. At the top, there are buttons for 'Doctor' and 'Consult', and a 'Logout' button. The patient's GP2 number is 1. The 'Personal Details' section includes: Patient's Health Number (GP2), Patient Name (Ms kikunga tembo), ID Number (22807908), and Patient Marital Status (Single). The 'Last treatment' section, performed by Ms kikunga, shows the date 2013-08-19 06:08:22, the doctor Dr Okebua, and the diagnosis Typhoid. A table lists 'Diagnosed Symptoms' (constipation, stomach pains, loss of appetite, vomiting) and 'Given Prescriptions' (Ampicillin, ceftriaxone, ciprofloxacin). A 'View All 2 treatments' link is present. At the bottom, there is a 'Being treated for' section with 'Symptom' and 'Prescription' dropdowns, 'More Symptoms' and 'More Prescriptions' buttons, and 'Treat Patient' and 'Reset' buttons.

Figure 4.26: The results of the next treatment

The result of the next consultation is shown in Figure 2.26. The patient is being treated on another day and by another doctor and he now has two treatments in his history. The treatment shown is the most recent one but if any doctor wants to view the previous treatment of the patient he can click on “view all the 2 treatment” and this will display the two treatments with all the details such as the date of treatment, the diagnosis, prescription and the name of the doctor who treated the patient on that particular date.

4.4 Chapter Summary

This chapter presented the implementation of the medical diagnosis system. It showed the interfaces and the results of how the system works. The system could assist doctors from three provinces of South Africa (Limpopo, Gauteng and North West) to make an adequate decision on diagnosing patients. The system was able to store patients, doctors and the administrator and the knowledge base's information. Our system could generate a unique health care number for each patient after registration. This health care number consists of letters representing the patient's location or province and numbers representing the patient's unique identifier. The health care number is used by the doctors when they want to retrieve the patient's information and for consultation purposes. Our system worked perfectly in that doctors/physicians could just enter at least three types of symptoms shown in the patient in order to get all the possible information linked to those symptoms for them to be able to diagnose and treat patients. The information in the knowledge base can be shared and reused by doctors in the three provinces of South Africa.

CHAPTER 5

SUMMARY, CONCLUSION AND FUTURE WORK

5.1 Summary

In today's medical world, due to the large volume of medical information that exists, it has been observed that medical practitioners find it difficult to speed up the rate at which they attend to patients. To overcome the challenges faced by physicians in making a precise diagnostic decision, a medical diagnostic system has been built. The medical diagnostic system is used for keeping medical record since managing patients' information manually, such as patients' files, can cause a lot of chaos, for instance patients' files can be misplaced or lost. This can lead to patients' information being exposed to unauthorized persons and this goes against the medical practices of privacy and confidentiality. Therefore, the medical diagnosis system makes it easier to manage patients' information since this information will be kept in a digital format which makes it difficult to be misplaced or lost and eliminates exposure to anyone else except authorized people to have access to the patients' information or any other medical information that needs higher security, privacy and confidentiality.

The medical diagnosis system keeps information that can be shared and reused amongst different physicians. This information gives physicians more knowledge on how to treat a particular disease instead of having to guess or giving inaccurate diagnostics. The medical diagnostic system has improved the keeping of medical records, and the provision of high quality health care services.

5.2 Conclusion

The effective management of medical knowledge and information has become a serious challenge for health-care providers especially when it comes to finding and using the relevant information at the right time, and SA is not an exception. SA is faced with rapid development of medical services and institutions which leads to problems regarding effective exchange of patient's medical histories and information. The exchange of information between medical information systems is difficult and sometimes goes against medical practices of privacy and confidentiality.

In everyday practice, medical knowledge grows steadily. It becomes increasingly difficult for physicians to keep up with essential information gained from the practice. Thus, for physicians to be able to quickly and accurately diagnose a patient there is a critical need for computer technologies. Computer technologies have become important tools in assisting firstly inexperienced physicians in making medical diagnosis and secondly experienced physicians in supporting complex decisions, retrieving medical information as well as making decisions to overcome medical complications that are prevalent in today's world. An example of this technology is the Medical Diagnostic System.

In conclusion, this research has built a framework for a medical knowledge- based system to be used by physicians in order to make fast diagnostic decisions. The framework has been designed to solve the ugly situation faced by health-care practitioners and introduce uniform standards for medical information handling. As this system is implemented, medical knowledge will be utilized for effective diagnostic decisions and patient's medical information accessed and shared across medical institutions while preserving the privacy and confidentiality principles.

5.3 Future Work

Regarding the background of health information system, our future work will focus on enforcing the privacy and security of patient's data by ensuring that the patient's data will not be accessed and used for unwanted purposes for example, discrimination on loan or insurance coverage based on information about patient's medical history and also prevent the information in the database not to be accessed by hackers.

REFERENCES

- [1] L. Aleksovska-stojkowska and S. Loskovska, "Clinical Decision Support System Medical Knowledge Acquisition and Representation Methods," in *electro/Information Technology (EIT), IEEE International Conference on Digital Object Identifier*, 2010, pp. 1- 6.
- [2] A. Jovic, M. Prcela and D. Gamberger. "Ontologies in Medical Knowledge Representation," in *Information Technology Interface, 29th International Conference on Digital Object Identifier*, 2007, pp. 535-540.
- [3] A. Aguilera and A. Subero, "A multi-agent architecture and system for dynamic medical knowledge acquisition," in *Computational Intelligence for Modeling Control & Automation, International Conference on Digital object identifier*, 2008, pp. 165- 170.
- [4] G. Lanzola, S. Falasconi and M. Steffanelli, M. "Cooperative Software Agents for Patient Management", *Artificial Intelligent medicine, 5th Conference on Artificial Intelligence in Medicine in Europe*, 1995, vol. 934, pp. 173-184.
- [5] A. Kabakcioglu. "Artificial Intelligence for medical Knowledge representation/ reasoning/ acquisition," *Processing of International Biomedical Engineering days*, 1992, pp. 186-191.
- [6] P. Meesad and G. G. Yen. "Combined Numerical and Linguistic Knowledge Representation and its Application to medical Diagnosis," *Systems, Man and Cybernetics, Part A, Humans, IEEE Transaction on Digital object identifier*, 2003, pp. 206-222.
- [7] D. Gamberger et al. "Medical Knowledge representation within Heartfaid platform," *In Proc. of Biostec, International Joint Conference on Biomedical Engineering Systems and Technologies*, 2008, pp.205-217.
- [8] F. M. E. Uzoka ,J Osuji, F. O Alaji and O. U. Obot. "A Framework for Cell Phone Diagnosis and Management of Priority Tropical Diseases," *IST-Africa Conference Proceedings*, 2011, pp. 1-13.
- [9] D.O Kang, H.J Lee, E.J KO, K. Kang and J. Lee. "A Wearable Context Awareness System for Ubiquitous HealthCare," *Engineering in Medicine and Biology Society, Proceeding of the 28th IEEE EMBS Annual International Conference*, 2006, pp. 5192-5195.
- [10] E. Coiera. "Guide to Health Informatics", Arnold, London, October 2003, (2nd Edition), Chapter 25.
- [11] C. Caiping and W. Huijing "A Framework for Medical Information Integration Based on ontology and web services," *Information Engineering and Computer Science (ICIECS), 2nd International Conference*, 2010, pp. 1- 4.

- [12] H. Lieberman and C. Mason Media Laboratory, MIT, Cambridge, MA, USA, University of California, Berkeley, CA, USA, "Intelligent-Agent Software for Medicine" *Stud HealthTechno*, 2002, pp. 80: 99- 109.
- [13] N. Wickramasinghe and E. Geisler. "Epistemetrics: Conceptual Domain and Applications of Knowledge Management (KM) in Health Care," *PICMET '07. Portland International Conference on Management of Engineering & Technology*, 2007, pp. 1056 - 1061.
- [14] M. Alsun. L. Lecornu and Basel solaiman. "Possibilistic Medical Knowledge Representation Model," *Journal of Advances in Information Fusion*, 2 December 2012, vol. 7, No. 2, pp. 101 - 113.
- [15] R. Seising, C. Schuh and K. P. Adlassing. "Medical Knowledge, fuzzy sets and expert Systems," *Workshop on Intelligent and Adaptive Systems in Medicine*, Prague 2003, Homepage: <http://cyber.felt.cvut.cz/EUNITE03;BIO/pdf/Seising.pdf>.
- [16] T. kiselovia. "A theoretical comparison of disco and CADIAG-II like systems for medical diagnosis," *kybemetika*, Vol. 42, 6 (2006), pp. 723- 748.
- [17] P. Szolovits, "Uncertainty and Decision in medical informatics, methods," *Methods of Information in Medicine*, 34, 1/2 (1995), pp. 111- 121.
- [18] B. Stroetmann, and A. Aisenbrey. "Medical Knowledge Management in Health industry" *World Academy of Science, Engineering and Technology*, Vol. 64, 2012, pp. 557-562.
- [19] E. S. Berner. "Clinical and Decision Support Systems Theory and Practice", Health Informatics series, Springer Science, 2007.
- [20] K. Makhubele. "A knowledge Based Expert System for Medical Advice provision," Honours Project Report, University of Cape Town, South Africa, 2012.
- [21] P. H. Jones. "Knowledge Acquisition," J. R. Barret, D. D. Jones. *Knowledge Engineering in Agriculture*, ASAE Monograph No. 8, ASAE, St. Joseph, MI, 1989.
- [22] S. M. Huff. "Ontologies, vocabularies, and data models," in R.A Greenes, editor. *Clinical decision Support: The road ahead*. Amsterdam: Academic Press, 2007, pp. 307- 324
- [23] G. Van Heijst, G. Lanzola, T. Schreiber and M. Stefanelli, "Foundations for a methodology for medical KBS development," *Knowledge Acquisition*, Vol. 6, 1994, pp. 395- 433.
- [24] M.A Musen, "Domain ontologies in Software engineering: use of protégé with the EON architecture," *Methods Inf Med.*, Vol. 37, 1998, pp. 540- 550.

- [24] G. Lanzola, S. Falasconi and M. Steffanelli, "Cooperative Software Agents for Patient Management," *Proceedings of the 5th Conference on Artificial Intelligence in Medicine in Europe: Artificial Intelligence Medicine*, LNCS, Vol.934, Springer-Verlag, 1995.
- [26] J. H. Carter, "Clinical Decision Support System," in E. S. Berner, Ed. Springer-Verlag, New York, 1999, pp. 169 - 198.
- [27] K. Verlaene, W. Joosen and P. Verbaene, "Arriclides: An Architecture Integrating Clinical Decision Support Models," *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, 2007, pp. 135c.
- [28] G. Potamias, V. Moustakis, G. Charissis, "Interactive knowledge based construction and maintenance," *Applied Artificial Intelligence*, vol. 11, 1997, pp. 697 - 717.
- [29] A. R. Puerta, H. Erksson, J. H. Gennari and M. A. Musen., "Toward ontology frameworks for knowledge-acquisition tools," *Proceedings of Eight Knowledge – Acquisition Workshop for Knowledge –Based Systems*, Banff, Alberta, Canada, February 1994.
- [30] S. Moein, S. A. Monadjemi and P. Moallem, "A Novel Fuzzy- Neural Based Medical Diagnosis System," *International Journal of Biological and Life Sciences*, Vol. 27, 2009.
- [31] J. G. Wolff. "Medical Diagnosis as Pattern Recognition in a Framework of Information Compression by Multiple Alignment, Unification and search," *Decision Support Systems* Vol. 24, no.2, 2006, pp. 608- 625.
- [32] C. Ugwu, N. L. Onyejebu, I. C Obagbua, "The Application of Machine Learning Techniques for Malaria Diagnosis", *Nigeria Computer Society 23rd National Conference 2009*, pp. 151-158.
- [33] O. U Obot, O. C Akinyokun and S. S. Udoh. "Application of Soft Computing Methodologies to the Management of Hypertension", *Nigeria Computer Society 22nd National Conference*, 2008, pp. 339 - 348.
- [34] T. Zrimec and I. Kononenko. "Feasibility analysis of machine learning in medical diagnosis from aura images," *University of Ljubljana*, 2004.
- [35] Z. Shujun, W. Kaiyu and Y. Zongyuan. "Modelling and Verifying of Medical Diagnosis system Based on Context-awareness Framework," *Fifth International Conference on frontier of Computer Science and Technology*, 2010, pp. 299 -304.
- [36] B. Iantovics "A novel diagnosis system specialized in difficult medical diagnosis problems solving Understanding," *Complex Systems*, Springer-Verlag Heidelberg, 2006, pp. 187- 197.

- [37] B. Iantovics. "The CMDS Medical Diagnosis System," symbolic and Numeric Algorithms for scientific Computing," *SYNASC 9th International Symposium*, 2007, pp. 246 - 253.
- [38] G. R Rao, B. A Suresh, M. Turoff and S. R Hiltz, "Issues in the Development of a Computer Mediated Communication System framework for Collaborative Medical Decision Making," engineering in Medicine and Biology Society, Engineering Advances New opportunities for Biomedical Engineers, *Proceeding of the 16th Annual International Conference of the IEEE*, Vol. 2 , 1994, pp. 1354- 1355.
- [39] G. Fenza, V. Loia and S. Senatore, "Fuzzy modeling of medical diagnosis through a service-oriented architecture," Fuzzy Information Processing Society, Annual Meeting of the North American, 2008, pp. 1- 6.
- [40] M. Dojata and F. Pachet "Effective Domain-dependent Reuse in Medical Knowledge Bases," Fuzzy Information Processing Society, NAFIPS, Annual Meeting of the North America, 2008, pp. 1- 6.
- [41] U. A Umoh, A. A. Umoh, G. G. James, U. U. Oton and J. J. Udoudo, "Design of Pattern Recognition System for the Diagnosis of Gonorrhoea Disease," *International Journal of Scientific & Technology Research*, Vol. 1, Issue 5, June 2012, pp. 74 -79.
- [42] S. A Fatumo, E. Adetiba and J. O Onaolapo. "Implementation of XpertMalTyph: An Expert System for Medical Diagnosis of the Complications of Malaria and Typhoid," *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol. 8, Issue 5, pp. 34 - 40, Jan. – Feb. 2013.
- [43] J. Fisher. "MYCIN Clinical Decision Support System," Available from <http://neamh.cms.cns.edu/MedInfo/mycin.html>
- [44] E. H. Shortliffe, A. C. Scott, M. B. Bischoff, A. B. Campbell, W. V. Melle and C. D. Jacobs. "ONCOCIN: An Expert System for Oncology Protocol Management," Heuristic Programming Project, Stanford University, pp. 876-881.
- [45] L. Lhotska, V. Marik and T. Vlcek. "Medical applications of enhanced rule-based expert systems," *International Journal of Medical Informatics* Vol. 63, 2001, pp. 61-75.
- [46] M. E. Hernando, E. J. Gomez, R. Corcoy and F. Del Pozo. "Evaluation of DIABNET, a decision support system for therapy planning in gestational diabetes," *Computer methods and programs in biomedicine* Vol. 62, 2000, pp. 235- 248.
- [47] J. Sliwa. "Scientist study bacterial communities inside us to better understand health and disease," 3 June 2008: Available from: http://www.eurekalert.org/pub_releases/2008-06/asfm-ssb052908.php.
- [48] S. J. Russel and P. Norvig. "Artificial Intelligence: A Modern Approach," 2nd ed. *Upper Saddle River*, New Jersey: Prentice- Hall, 2003.

- [49] S. S. Sikchi, S. Sushil and M. S. Ali. "Artificial Intelligence in Medical Diagnosis," *International Journal of Applied Engineering Research*, Vol. 7 No 11 (2012), pp. 1-5: Available: <http://www.ripublication.com/ijaer.htm>.
- [50] R. Rahman, M. D. Rashidujjaman Rifat, S. Moutushy and H.S. Ferdous. "Toward a Customizable Effective Patient Management System for Ensuring Quality Medical Service," *IEEE Student Conference on Research and Development*, 2011, pp. 415-419.
- [51] M. O. Katchienga. "Technological Innovations in Health Care: Challenges of Managing Telehealth Technology in South Africa," Graduate School of Technology Management, University of Pretoria, South Africa, 2008, pp. 1-5.
- [52] M. O. Katchienga. "Health technology assessment in Sub-Saharan Africa: A Cross-National Study of Kenya and South Africa," Doctoral Thesis, University of Cape Town, South Africa, 1999.
- [53] Z. Damtew. "Using Local knowledge in Information Gathering Process: Case study from Ethiopia," *IST-Africa Conference Processings*, Paul Cunningham and Miriam Cunnigham (Eds), International Information Management Corporation, ISBN: 978-1-905824-19-9, 2010, pp. 1-7.
- [54] G. Lemecha. "Addressing Health Management Information System in Ethiopia: a business case," A dissertation Submitted to the University of Warwick in partial fulfillment to the degree of Master of Science. Warwick Medical School.
- [55] United Nations. "The Millennium Development Goal Report, New York, 2008.
- [56] WHO expert Committee. "Information Support for new public health action at district level, technical report series, Geneva: Switzerland, 1994.

APPENDICES

```
import com.gui.Form;
import com.gui.FormattedForm;
import com.gui.LightButton;
import com.gui.LightButton2;
import com.gui.PField;
import com.gui.Run;
import com.gui.TField;
import com.objects.User;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.JComponent;
import javax.swing.Timer;

/**
 *
 * @author TJ
 */
public class Login_Form extends Form implements KeyListener, ActionListener {
    public static TField username;
    public static PField password;
    JComponent comps[];
    Thread check = null;
    Timer checkSleep;
```

```

public static boolean go = false;
int timeNow = 0;
LightButton2 login, clear;
private final int[][] matrix = {{1, 0}, {1, 0}, {1, 1}};
static User user;

public Login_Form() {
    init();

    comps = new JComponent[]{username, password, login, clear};
    FormattedForm theForm = new FormattedForm(comps, matrix);
    add(theForm);
    loginCheck();

    theForm.setPreferredSize(new Dimension(350, getPreferredSize().height));
}

public static void logoff() {
    go = false;
}

public static boolean isLoggedIn() {
    return go;
}

void init() {
    username = new TField("Username", 20);
    password = new PField("password", 20);
    username.addKeyListener(this);
    password.addKeyListener(this);
    login = new LightButton2("Login");
    clear = new LightButton2("Clear");
    login.addActionListener(this);
}

```

```

clear.addActionListener(this);
}
private void loginCheck() {
check = new Thread(new Runnable() {
@Override
public void run() {
if (checkSleep == null) {
checkSleep = new Timer(1000, Login_Form.this);
checkSleep.start();
}
while (true) {
if (Login_Form.go) {
checkSleep.stop();
check.interrupt();
check = null;
checkSleep = null;
Run.firstPage(user);
break;
}
if (timeNow >= 10) {
checkSleep.stop();
timeNow = 0;
check.interrupt();
check = null;
checkSleep = null;
break;
}
}
}
}

```

```

    }
}
});
check.start();
}
@Override
public void keyTyped(KeyEvent e) {
}
@Override
public void keyPressed(KeyEvent e) {
    attempt();
    if (KeyEvent.VK_ENTER == e.getKeyCode()) {
        formValidate();
    }
}
@Override
public void keyReleased(KeyEvent e) {
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() != login) {
        if (e.getSource() == clear) {
            username.setText(null);
            password.setText(null);
            Run.focus(username);
        } else {
            timeNow += 1;

```

```

    }
} else {
    attempt();
    formValidate();
}
}

public void attempt() {
    timeNow = 0;
    if (checkSleep == null) {
        checkSleep = new Timer(1000, Login_Form.this);
    }
    if (!checkSleep.isRunning()) {
        checkSleep.start();
    }
    if (check == null) {
        loginCheck();
    }
}

public void formValidate() {
    String uName = username.getText().toString();
    String pWord = new String(password.getPassword());
    user = User.login(uName, pWord, true);
    if (user != null) {
        go = true;
    } else {
        Run.error("Username or password incorrect.");
    }
}

```



```

    }
}
Do import com.gui.Main;
import com.gui.Run;
import com.gui.Tab;
import com.gui.admin.CreateDocAccount_Form;
import com.gui.admin.EditDocAccount_Form;
import com.gui.admin.SearchDoctor_Form;
import com.gui.admin.ViewAllDoctors_Form;
import com.gui.doctor.ChangePassword_Form;
import com.objects.Admin;
import com.objects.Doctor;
import com.objects.User;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.math.BigInteger;
import java.util.Arrays;
import javax.swing.JButton;
/**
 *
 * @author TJ
 */
public class DoctorsTab extends Tab implements ActionListener {
    public static JButton createDocAccount;

```

```

JButton viewDocAccount, viewAllDoctors, changeMyPassword;
CreateDocAccount_Form createForm;
SearchDoctor_Form searchForm2;
ViewAllDoctors_Form viewAllForm;
int id = 0;
private ChangePassword_Form pChangeForm;
public DoctorsTab() {
    setBackground(super.getBackground());
    setLayout(new FlowLayout());
    createDocAccount = new JButton("Create Doctor");
    viewAllDoctors = new JButton("All Doctors");
    viewDocAccount = new JButton("Search");
    changeMyPassword = new JButton("Change Password");
    createDocAccount.addActionListener(this);
    viewDocAccount.addActionListener(this);
    viewAllDoctors.addActionListener(this);
    changeMyPassword.addActionListener(this);
    super.subButtons = new Component[]{createDocAccount, viewAllDoctors, viewDocAccount,
changeMyPassword};
    super.setSubButtons();
    actionPerformed(new ActionEvent(createDocAccount, 0, null));
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == createDocAccount) {
        createForm = new CreateDocAccount_Form();
        if (!Arrays.asList(createForm.create.getActionListeners()).contains(this)) {
            createForm.create.addActionListener(this);

```

```

    }
    Run.focus(createForm.d_id);
    if (!Arrays.asList(getComponents()).contains(createForm)) {
        clean();
        add(createForm);
        refresh();
    }
}
}
if (e.getSource() == changeMyPassword) {
    clean();
    pChangeForm = new ChangePassword_Form();
    pChangeForm.confirmChangeP.addActionListener(this);
    pChangeForm.setDefaultSize();
    add(pChangeForm);
    refresh();
}
if (pChangeForm != null && e.getSource() == pChangeForm.confirmChangeP) {
    String oldP = new String(pChangeForm.oldP.getPassword());

    String newP = new String(pChangeForm.newP.getPassword());
    String confirmP = new String(pChangeForm.cNewP.getPassword());
    if (!Run.Current.checkPassword(oldP)) {
        String msg = "Password change cannot be done, you must enter correctly your current
password.";
        Run.error(msg);
        return;
    }
    if (!newP.equals(confirmP)) {

```

```

String msg = "New passwords do not match.";
Run.error(msg);
return;
}
if (Run.Current.changePassword(newP)) {
    Run.info("Password has been changed.");
    Main.top();
} else {
    Run.error(Run.Current.getError());
    return;
}
}
if (createForm != null && e.getSource() == createForm.create) {
    if (createForm.username.getText().length() == 0) {
        createForm.username.setError();
        Run.error("Username cannot be empty.");
        return;
    }
    String username = createForm.username.getText();
    if (createForm.password.getText().length() == 0) {
        createForm.password.setError();
        Run.error("Password cannot be empty.");
        return;
    }
    String password = new String(createForm.password.getPassword());
    if (createForm.d_name.getText().length() == 0) {
        createForm.d_name.setError();

```

```

    Run.error("Doctor's name cannot be empty");
    return;
}

String d_name = createForm.d_name.getText();
if (createForm.d_lname.getText().length() == 0) {
    createForm.d_lname.setError();

    Run.error("Doctor's cannot be empty");
    return;
}

String d_lname = createForm.d_lname.getText();
if (createForm.d_title.selected == null || createForm.d_title.selected.length() == 0) {
    //JOptionPane.showMessageDialog(this, "Please select title.");

    Run.error("Please specify title of the doctor.");
    return;
}

String d_title = createForm.d_title.selected;
if (createForm.d_id.getText().length() == 0) {
    createForm.d_id.setError();

    Run.error("Doctor's id cannot be empty");
    return;
}

BigInteger d_id = new BigInteger(createForm.d_id.getText());
if (createForm.d_age.getText().length() == 0) {
    createForm.d_age.setError();

    Run.error("Please fill in doctor's age");
    return;
}
}

```

```

int d_age = Integer.parseInt(createForm.d_age.getText());
if (createForm.d_sex.getSelected() == null) {
    //JOptionPane.showMessageDialog(this, "Please select gender.");
    Run.error("Please select gender.");
    return;
}
String d_sex = createForm.d_sex.getSelected();
if (createForm.d_religion.getText().length() == 0) {
    createForm.d_religion.setError();
    Run.error("Please specify religion");
    return;
}
String d_religion = createForm.d_religion.getText();
if (createForm.d_specialization.getText().length() == 0) {
    createForm.d_specialization.setError();
    Run.error("Specify the doctor's area of specialization");
    return;
}
String d_specialization = createForm.d_specialization.getText();
if (createForm.d_contact.getText().length() == 0) {
    createForm.d_contact.setError();
    Run.error("Doctor's contact number cannot be empty");
    return;
}
BigInteger d_contact = new BigInteger(createForm.d_contact.getText());
User user = new User(username, "doctor", password);
if (!user.create()) {

```

```

        System.out.println(user.getError());
        return;
    }
    user.user_id = user.last_inserted();
    Run.progress("user created user_id: " + user.user_id);

    Doctor doctor = new Doctor(d_id, d_name, d_lname, d_title, d_age, d_sex, d_religion,
d_specialization, d_contact, user.user_id,
(Admin.get_by_user_id(Run.Current.user_id)).admin_id);
    if (!doctor.create()) {
        user.delete();

        System.out.println(doctor.getError());
        return;
    }
    doctor = Doctor.get_by_id_num(doctor.d_id);
    clean();

    EditDocAccount_Form edf = new EditDocAccount_Form(doctor, false);
    setPreferredSize(new Dimension(getPreferredSize().width, edf.getPreferredSize().height));
    add(edf);

    Main.currentTab.setPreferredSize(new Dimension(getPreferredSize().width,
getPreferredSize().height));
    refresh();

    Run.progress("Doctor created.");
    Run.info("Doctor's accout has been created created.");
}
if (e.getSource() == viewDocAccount) {
    clean();

    setLayout(null);

    if (searchForm2 == null) {
        searchForm2 = new SearchDoctor_Form();

```

```

    }
    add(searchForm2);
    searchForm2.setBounds(0, 0, searchForm2.getPreferredSize().width,
searchForm2.getPreferredSize().height);
    refresh();
}
if (e.getSource() == viewAllDoctors) {
    clean();
    setLayout(null);
    viewAllForm = new ViewAllDoctors_Form();
    add(viewAllForm);
    viewAllForm.setBounds(0, 0, viewAllForm.getPreferredSize().width,
viewAllForm.getPreferredSize().height);
    refresh();
}
if (e.getSource() != viewDocAccount && e.getSource() != viewAllDoctors) {
    if (getLayout() == null) {
        setLayout(new FlowLayout());
        refresh();
    }
}
}
public void setPreferredSize(Dimension d) {
    super.setPreferredSize(d);
}
}

```

```

public void clean() {

```

```

    if (getComponentCount() != 0) {
        removeAll();
        repaint(50L);
        revalidate();
    }
}

public void refresh() {
    repaint(50L);
    revalidate();
}
}ctor Tab

import com.gui.Run;
import com.gui.TField;
import com.gui.Tab;
import com.gui.admin.AddKnowledge_Form;
import com.gui.forms.AddPrescription_Form;
import com.gui.forms.AddSymptom_Form;
import com.gui.forms.AddTreatment_Form;
import com.gui.forms.ListKnowledge;
import com.objects.Admin;
import com.objects.Knowledge;
import com.objects.Prescription;
import com.objects.Symptom;
import com.objects.Treatment;
import java.awt.Component;
import java.awt.FlowLayout;

```

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
/**
 *
 * @author TJ
 */
public class KnowledgeTab extends Tab implements ActionListener {
    JButton addKnowledge, knowledgeLookup;
    AddKnowledge_Form addKnowledgeForm;
    AddSymptom_Form addSymptomForm;
    AddTreatment_Form addTreatmentForm;
    AddPrescription_Form addPrescriptionForm;
    public KnowledgeTab() {
        setBackground(super.getBackground());
        setLayout(new FlowLayout());
        addKnowledge = new JButton("Add Disease Knowledge");
        knowledgeLookup = new JButton("Knowledge Base");
        addKnowledge.addActionListener(this);
        knowledgeLookup.addActionListener(this);
        super.subButtons = new Component[] {addKnowledge, knowledgeLookup};
        super.setSubButtons();
        actionPerformed(new ActionEvent(addKnowledge, 0, null));
    }

```

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == knowledgeLookup) {
        Run.progress("List all knowledge");
        clean();
        ListKnowledge form = new ListKnowledge();
        knowledgeLookup.setEnabled(true);
        add(form);
        refresh();
    }
    if (e.getSource() == addKnowledge) {
        clean();
        if (addKnowledgeForm == null) {
            addKnowledgeForm = new AddKnowledge_Form();
            addKnowledgeForm.createKnowledge.addActionListener(this);
        }
        add(addKnowledgeForm);
        refresh();
    }
    if (addKnowledgeForm != null && e.getSource() == addKnowledgeForm.createKnowledge) {
        Run.debug("try adding knowledge");
        if (addKnowledgeForm.k_name.getText().toString().trim().length() == 0) {
            addKnowledgeForm.k_name.setError();
            return;
        }
        String k_name = addKnowledgeForm.k_name.getText().toString().trim();
        if (addKnowledgeForm.k_history.getText().toString().trim().length() == 0) {

```

```

    addKnowledgeForm.k_history.setError();
    return;
}
String k_history = addKnowledgeForm.k_history.getText().toString().trim();
if (Admin.get_by_user_id(Run.Current.user_id) == null) {
    Run.error("Somehow admin id could not be retrieved, cannot proceed.");
    return;
}
Knowledge k = new Knowledge(k_name, k_history,
(Admin.get_by_user_id(Run.Current.user_id)).admin_id);
if (!k.create()) {
    Run.debug(k.getError());
    return;
}
int k_id = k.last_inserted();
k = null;
k = Knowledge.get_by_id(k_id);
if (k == null) {
    Run.debug("That knowledge you last inserted could not be retrieved.");
    clean();
    refresh();
    return;
}
if (addSymptomForm == null) {
    addSymptomForm = new AddSymptom_Form(k_id);
}
addSymptomForm.saveAndContinue.addActionListener(this);
addSymptomForm.skip.addActionListener(this);

```

```

clean();
add(addSymptomForm);
refresh();
addKnowledgeForm = null;
}
if (addSymptomForm != null && e.getSource() == addSymptomForm.skip) {
}
if (addSymptomForm != null && e.getSource() == addSymptomForm.saveAndContinue) {
    List<TField> symptomsFields = addSymptomForm.symptoms;
    List<Symptom> symptoms = new ArrayList<Symptom>();
    for (int i = 0; i < symptomsFields.size(); i++) {
        if (symptomsFields.get(i).getText().toString().trim().length() <= 0 && i < 3) {
            symptomsFields.get(i).setError();
            return;
        } else {
            symptomsFields.get(i).setSuccess();
        }
        if (symptomsFields.get(i).getText().toString().trim().length() > 0) {
            symptoms.add(new Symptom(addSymptomForm.k_id,
symptomsFields.get(i).getText().toString()));
        }
    }
    List<Integer> newSymptomsIDs = new ArrayList<Integer>();
    for (int i = 0; i < symptoms.size(); i++) {
        /**
         * At this point we must roll back all changes
         */
        if (!symptoms.get(i).create()) {

```

```

Run.debug("Fail to insert symptom " + i);
for (int k = 0; k < i; k++) {
    if (!symptoms.get(k).delete()) {
        Run.debug("Failed to rollback changes on Symptom: " + symptoms.get(k).s_name);
    }
}
return;
}
symptoms.get(i).k_id = symptoms.get(i).last_inserted();
newSymptomsIDs.add(symptoms.get(i).last_inserted());
}
clean();
addTreatmentForm = new AddTreatment_Form(addSymptomForm.k_id);
addTreatmentForm.saveAndContinue.addActionListener(this);
add(addTreatmentForm);
addSymptomForm = null;
refresh();
}
if (addTreatmentForm != null && e.getSource() == addTreatmentForm.saveAndContinue) {
    List<Treatment> treatments = new ArrayList<Treatment>();
    for (int i = 0; i < addTreatmentForm.t_cure.size(); i++) {
        if ((addTreatmentForm.t_cure.get(i).getText().toString().trim().length() <= 0 ||
addTreatmentForm.t_stage.get(i).getText().toString().trim().length() <= 0) && i < 3) {
            addTreatmentForm.t_cure.get(i).setError();
            addTreatmentForm.t_stage.get(i).setError();
        }
        return;
    } else {
        addTreatmentForm.t_cure.get(i).setSuccess();
    }
}

```

```

        addTreatmentForm.t_stage.get(i).setSuccess();

        treatments.add(new Treatment(addTreatmentForm.k_id,
addTreatmentForm.t_cure.get(i).getText().toString().trim(),
addTreatmentForm.t_stage.get(i).getText().toString().trim()));
    }
}
for (int i = 0; i < treatments.size(); i++) {
    /**
     * At this point we must roll back all changes
     */
    if (!treatments.get(i).create()) {
        Run.debug("Fail to insert Treatment " + i);
        for (int k = 0; k < i; k++) {
            if (!treatments.get(k).delete()) {
                Run.debug("Failed to rollback changes on Treatment: " + treatments.get(k).t_cure);
            }
        }
        return;
    }
    treatments.get(i).k_id = treatments.get(i).last_inserted();
}
if (addPrescriptionForm == null) {
    addPrescriptionForm = new AddPrescription_Form(addTreatmentForm.k_id);
}
addPrescriptionForm.saveAndContinue.addActionListener(this);
clean();
add(addPrescriptionForm);
refresh();

```

```

    addTreatmentForm = null;
}
if (addPrescriptionForm != null && e.getSource() == addPrescriptionForm.saveAndContinue)
{
    List<TField> prescriptions = addPrescriptionForm.prescriptions;
    List<Prescription> finalPrescriptions = new ArrayList<Prescription>();
    for (int i = 0; i < prescriptions.size(); i++) {
        if (prescriptions.get(i).getText().toString().trim().length() == 0) {
            prescriptions.get(i).setError();
            return;
        } else {
            prescriptions.get(i).setSuccess();
            finalPrescriptions.add(new Prescription(prescriptions.get(i).getText().toString().trim(),
addPrescriptionForm.k_id));
        }
    }
    for (int i = 0; i < finalPrescriptions.size(); i++) {
        /**
         * At this point we must roll back all changes
         */
        if (!finalPrescriptions.get(i).create()) {
            Run.debug("Fail to insert Prescription " + i);
            for (int k = 0; k < i; k++) {
                if (!finalPrescriptions.get(k).delete()) {
                    Run.debug(finalPrescriptions.get(k).getError());
                }
            }
        }
    }
    return;
}

```

```

    }
    finalPrescriptions.get(i).k_id = finalPrescriptions.get(i).last_inserted();
}
addPrescriptionForm = null;
JOptionPane.showMessageDialog(this, "Disease knowledge has been added.");
actionPerformed(new ActionEvent(addKnowledge, 0, null));
}
}
}

```

```

import com.constants.WindowSize;
import com.gui.Form;
import com.gui.PField;
import com.gui.TField;
import java.awt.Dimension;
/**
 *
 * @author TJ
 */
public class ConsultLookup_Form extends Form {
    public TField searchIDBox;
    public ConsultLookup_Form() {
        setSize();
        init();
        append();
    }
    void init() {
        searchIDBox = new TField("Search by Id number", 20);

```

```

    }
    void append() {
        add(searchIDBox);
    }
    void setSize() {
        Dimension size = new Dimension(WindowSize.MAIN_W, WindowSize.MAIN_H);
        setPreferredSize(size);
        setMinimumSize(size);
        setMaximumSize(size);
    }
}

import com.gui.Form;
import com.gui.FormattedForm;
import com.gui.JCombo;
import com.gui.PField;
import com.gui.JRadio;
import com.gui.TArea;
import com.gui.TField;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JComponent;
/**
 *
 * @author TJ
 */
public class CreateDocAccount_Form extends Form {

```

```

public TField username;
public PField password;
public TField d_name;
public TField d_lname;
public JCombo d_title;
public TField d_id;
public TField d_age;
public JRadio d_sex;
public TField d_religion;
public TArea d_specialization;
public TField d_contact;
public JButton create;
private int formMatrix[][] = {{1, 1}, {1, 1}, {1, 1}, {1, 0}, {1, 1}, {1, 0}, {1, 0}, {1, 0}};
public CreateDocAccount_Form() {
    setLayout(new FlowLayout());
    setSize();
    init();
    JComponent comps[] = new JComponent[]{username, password, d_name, d_lname, d_title,
d_id, d_age, d_sex, d_religion, d_specialization, d_contact, create};
    add(new FormattedForm(comps, formMatrix));
}
void init() {
    username = new TField("Username", 20);
    password = new PField("Password", 20);
    d_name = new TField("First Name", 20);
    d_lname = new TField("Surname", 20);
    d_title = new JCombo("Title", new String[]{"...", "Dr", "Mr", "Mrs", "Ms", "Miss", "Prof"},
new String[]{"", "Dr", "Mr", "Mrs", "Ms", "Miss", "Prof"});
}

```

```

    d_id = new TField("Id Number", 20);
    d_id.numOnly(13);
    d_age = new TField("Age", 20);
    d_age.numOnly();
    d_sex = new JRadio("Gender", new String[]{"Female", "Male"}, new String[]{"female",
"male"});

    d_religion = new TField("Religion", 20);
    d_specialization = new TArea("Specialization", 30, 15);
    d_specialization.tip("Area of Specialization...");
    d_contact = new TField("Cell number", 20);
    d_contact.numOnly(10);
    create = new JButton("Create");
}
void setSize() {
    setDefaultSize();
}
}

```

```

import com.gui.Form;
import com.gui.FormattedForm;
import com.gui.JCombo;
import com.gui.JRadio;
import com.gui.TArea;
import com.gui.TField;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JOptionPane;

```

```

/**
 *
 * @author TJ
 */
public class CreatePatient_Form extends Form { public TField p_id;
public TField p_name;
public TField p_surname;
public JCombo p_title, province;
public TField p_age;
public JRadio p_sex;
public TArea p_religion;
public JCombo p_marital_status;
public TArea p_phy_address;
public TArea p_po_address;
public JButton create;

private int formMatrix[][] = {{1, 1}, {1, 1}, {1, 1}, {1, 1}, {1, 0}, {1, 0}, {1, 0}, {1, 0}};

public CreatePatient_Form() {
    setLayout(new FlowLayout());
    setSize();
    init();

    JComponent comps[] = new JComponent[] {p_id, province, p_surname, p_name, p_title,
p_age, p_sex, p_marital_status, p_religion, p_phy_address, p_po_address, create};

    add(new FormattedForm(comps, formMatrix));
}

void init() {
    p_id = new TField("ID Number", 20);
    p_id.numOnly(13);
    p_name = new TField("Name", 20);

```

```

    p_surname = new TField("Surname", 20);

    p_title = new JCombo("Title", new String[]{"...", "Dr", "Mr", "Mrs", "Ms", "Miss", "Prof"},
new String[]{"", "Dr", "Mr", "Mrs", "Ms", "Miss", "Prof"});

    province = new JCombo("Province", new String[]{"...", "Limpopo", "Gauteng",
"Mpumulanga", "Northwest", "FreeState", "Northern Cape", "Kwazulu-Natal", "Western Cape",
"Eastern Cape"}, new String[]{"", "limpopo", "Gauteng", "Mpumulanga", "NorthWest",
"FreeState", "NorthernCape", "KwazuluNatal", "WesternCape", "EasternCape"});

    p_age = new TField("Age", 20);

    p_age.numOnly();

    p_sex = new JRadio("Gender", new String[]{"Female", "Male"}, new String[]{"female",
"male"});

    p_religion = new TArea("Religion", 30, 15);

    p_religion.tip("Religion...");

    p_marital_status = new JCombo("Marital Status", new String[]{"...", "Single", "Married",
"Widowed", "Engaged"}, new String[]{"", "Single", "Married", "Widowed", "Engaged"});

    p_phy_address = new TArea("Physical Address", 30, 15);

    p_phy_address.tip("Physical Address...");

    p_po_address = new TArea("Postal Address", 30, 15);

    p_po_address.tip("Postal Address...");

    create = new JButton("Register");
}

void setSize() {
    setDefaultSize();
}

```