



**ADDIS ABABA UNIVERSITY  
SCHOOL OF PUBLIC HEALTH AND  
SCHOOL OF INFORMATION SCIENCE**

**HEALTH INFORMATICS PROGRAMME**

**DEVELOPING WEB-BASED REFERRAL INFORMATION MANAGEMENT  
SYSTEM FOR HOSPITALS AND HEALTH CENTERS IN ADDIS ABABA**

**By**  
AkaleRegassa (B.Sc.)

June 2012  
Addis Ababa



**ADDIS ABABA UNIVERSITY  
SCHOOL OF PUBLIC HEALTH AND  
SCHOOL OF INFORMATION SCIENCE**

**HEALTH INFORMATICS PROGRAMME**

**DEVELOPING WEB-BASED REFERRAL INFORMATION MANAGEMENT  
SYSTEM FOR HOSPITALS AND HEALTH CENTERS IN ADDIS ABABA**

**By:** AkaleRegassa

**Advisors:** Alemayehu Mekonnen (MD, MPH)  
Gashaw Kebede (PhD)

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATES STUDIES OF ADDIS  
ABABA UNIVERISTY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE  
DEGREE OF MASTER OF SCIENCE IN HEALTH INFORMATICS

June 2012  
Addis Ababa

**ADDIS ABABA UNIVERSITY**  
**SCHOOL OF GRADUATE STUDIES**

**DEVELOPING WEB-BASED REFERRAL INFORMATION MANAGEMENT  
SYSTEM FOR HOSPITALS AND HEALTH CENTERS IN ADDIS ABABA**

By  
AkaleRegassa

HEALTH INFORMATICS PROGRAM  
FACULTY OF INFORMATICS, ADDIS ABABA UNIVERISTY  
APPROVED BY THE EXAMINING BOARD

---

CHAIRMAN, DEPARTEMENT GRADUATE COMMITTEE

ALEMAYEHU MEKONNEN

\_\_\_\_\_

ADVISOR

GASHAW KEBEDE

\_\_\_\_\_

ADVISOR

**1. Dr. Getinet Mitike** \_\_\_\_\_

**2. Ato Ermias Abebe** \_\_\_\_\_

EXAMINER

## **ACKNOWLEDGEMENT**

First, my deepest thanks go to almighty God who makes possible all things in my life. Had not be God with me, my dreams would not have been real. I am very glad to tell the great care and love of God for anyone who will access my paper.

I would like to put my deepest and greatest gratitude to my advisors, Alemayehu Mekonnen and Gashaw Kebede, who offer me with all the support, valuable comment and guidance to make this moment real.

My deepest appreciation is also goes to my great and best friend Ashenafi Demesse, what can I say about you? Your sympathy, value, personality and knowledge are priceless. Best wishes and success in your future life. Thanks my dear, Abiot Desta, you are the one who makes me feel confident, strong and determined in doing my research. Thanks my friends Askale Kassa, Amanuel Tekalign and Haileeyesus Kinde for your support and comments. I am proud of having you.

I am indebted to my mother Abuye; you are best, brave, valuable and determined mother. The success which I got today is because of your efforts, love you much aba, thank you my brothers Dawit Regassa ,Yosef Regassa and sister Ayantu Belayneh.

Finally, I would like to thank Yodit who helped me in providing information for my work at Yekatit 12 Hospital, and my class mate friends Tomas Habte, Anteneh Aklilu and Worku Tadesse. Thank you all of my instructors in school of Information Science and School of Public Health.

**God Bless You All!**

## **Acronyms**

AAAB – Addis Ababa Administration Bureau

ASP – Active Server Page

BR –Business Rule

CCO – Coordinated Care Organization

CGI – Common Gateway Interface

CSS – Cascading Style Sheet

DBMS –Data Base Management System/Software

DDL – Data Definition Language

DML – Data Manipulation Language

EHGT – Electronic Health Global Technology

E-Health – Electronic Health

EHR – Electronic Health Record

E-Mail – Electronic Mail

EMR – Electronic Medical Record

E-Referral – Electronic Referral

GPs – General Practitioners

GUI – Graphical User Interface

HIPAA – Health Insurance Portability and Accountability Act)

HSC – Health Service Center

HTML – Hyper Text Markup Language

IRIS – Internet Referral Information System

IPA – Inter-Hospitals Professionals Association

ISS - Internet Information Server

IP –Internet Protocol

IT – Information Technology

IVR – Intelligent Voice Réponse

JAR –Java Archive

Java EE – Java Enterprise Edition

JSP – Java Server Page

MHV – Microsoft Health Vault

MySQL – My Structured Query Language  
NGO – Non Governmental Organization  
NCCP – National Cancer Control Program  
NGT – Nominal Group Technique  
OO – Object Oriented  
OOA – Object Oriented Analysis  
OOD – Object Oriented Design  
OOA/D – Object Oriented Analysis and Design  
OOP – Object Oriented Programming  
OOSD – Object Oriented System Development  
PCP – Primary Care Provider  
PHP – Hypertext Pre Processor  
SFGH – San Francisco General Hospital  
SQL – Structured Query Language  
SSL – Secure Shield Line  
TCP – Transmission Control Protocol  
UC – Use Case  
UI – User Interface  
UK – United Kingdom  
UML – Unified Modeling Language  
US – United States  
USA – United State of America  
WRIMS – Web- based Referral Information Management System  
WRIMSAA – Web-based Referral Information Management System in Addis Ababa  
WWW - World Wide Web  
ZEPRS – Zambia Electronic Perinatal Record System

<b>Table of Contents</b>	<b>Pages</b>
ACKNOWLEDGEMENT	IV
ACRONYMS	V
TABLE OF CONTENTS	VII
LIST OF TABLES	XI
LIST OF FIGURES	XI
ABSTRACT	XII
CHAPTER ONE: INTRODUCTION	1
1.1 Background	1
1.2 Statement of the Problem	2
1.3 Objective of the study	3
1.3.1 General Objective	3
1.3.2 Specific Objectives	3
1.4 Research Questions	4
1.5 Rationale for the study	4
1.6 Application of the study	5
1.7 Scope of the Study	5
1.8 Organization of the Document	6
1.9 Operational Definition	6
CHAPTER TWO: LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Web-based System	7
2.3 Referral Management System	7
2.4 The Role of Medical Referral System for Referral Decision	8
2.5 Web-based Referral Information Management System	8
2.5.1 South Korea	9
2.5.2 USA	9
2.5.3 Denmark	11
2.5.4 Ireland	11
2.5.5 Zambia	12

2.6 Online versus Paper- based Referral	13
2.6.1 UK	13
2.6.2 USA	13
2.7 Selected Web- based Referral System	14
2.7.1 Web-based Patient Placement Referral System	14
2.7.2 Internet Referral Information System	15
2.7.3 Physician –to –Physician Community Referral System	16
2.7.4 E-Health Access Referral Management System	16
2.7.5 Strata Pathways	17
2.7.6 Health Insurance Portability and Accountability Act	18
2.7.7 Zambia Electronic Perinatal Referral System	18
2.8 Web-based Referral System Methodologies	19
2.8.1 Refer A Smoker Org	19
2.8.1.1 Study Design	19
2.8.1.2 Setting and Sample	20
2.8.2 A New Methodology For Developing Web-based System	20
2.9 Summary	22
2.10 Related Works in Ethiopia	22
CHAPTER THREE: METHODOLOGY	23
3.1 Introduction	23
3.2 Conceptual Framework of the Study	23
3.3 Web-based System Development Methodologies	24
3.3.1 Object – Oriented Analysis and Design Methodologies	25
3.4 Web-based System Development Phases	26
3.4.1 Object – Oriented Analysis	26
3.4.2 Object – Oriented Design	27
3.4.3 Implementation	27
3.5 Data Collection Methodology	28
3.5.1 Data Source	28
3.5.2 Sampling	28
3.5.3 Data Gathering Technique	28

CHAPTER FOUR: WEB-BASED SYSTEM DEVELOPMENT	31
4.1 Web-based Architecture for WRIMSAA	31
4.2 Requirement Analysis of the Existing System	32
4.2.1 Use case	32
4.2.1.1 Essential Use case Modeling	33
4.2.2 Non Functional Requirement	37
4.2.3 Business Rule Identification	38
4.3 Object- Oriented Analysis	38
4.3.1 System use case Modeling	39
4.3.2 Conceptual Modeling	45
4.3.3 Data Modeling	49
4.3.4 User Interface Flow Diagramming	51
4.4 Object – Oriented Design	53
4.4.1 Sequence Diagram	54
4.4.2 Class Diagram	60
4.4.3 User Interface Design	62
4.4.4 Database Design	66
4.5 Implementation	68
4.5.1 Class Implementation	68
4.5.2 Data Base Implementation	69
4.5.3 Web Application Implementation	69
4.5.4 Integration and Testing	69
4.6 Results	70
4.7 Evaluation of WRIMSAA	70
4. 7.1 Evaluation Result	71

CHAPTER FIVE: CONCLUSION and RECOMMENDATION	72
5.1 Conclusion	72
5.2 Contribution of the Work	72
5.3 Strength and Limitation of the Study	73
5.3.1 Strength of the Study	73
5.3.2 Limitation of the Study	73
5.4 Recommendation	74
REFERENCES	75
ANNEXES	78

<b>List of Tables</b>	<b>Pages</b>
Table 4.1: Conceptual/Data modeling concepts and attributes	41
Table 4 .2: Data modeling entities and attributes	45

<b>List of Figures</b>	<b>Pages</b>
Figure 3.1: Conceptual framework of the research	23
Figure 4.1: 3 –tier architecture for Web-based application	28
Figure 4.2: Essential use case diagram for WRIMSA	34
Figure 4.3: System use case diagram for WRIMSAA	39
Figure 4.4: Domain/Conceptual model for WRIMSAA	48
Figure 4.5: Data model for WRIMSAA	51
Figure 4.6: User Interface Flow diagram for WRIMSAA	53
Figure 4.7: Login Sequence diagram for WRIMSAA	55
Figure 4.8: Refer patient sequence diagram for WRIMSAA	55
Figure 4.9: Register referral result sequence diagram for WRIMSAA	56
Figure 4:10 Edit referral sequence diagram for WRIMSAA	56
Figure 4.11: View referral sequence diagram for WRIMSAA	57
Figure 4.12: Allocate patient sequence diagram for WRIMSAA	57
Figure 4.13: Add user sequence diagram for WRIMSAA	58
Figure 4.14: Schedule Appointment sequence diagram for WRIMSAA	58
Figure 4.15: View feedback sequence diagram for WRIMSAA	59
Figure 4.16: Generate Report sequence diagram for WRIMSAA	59
Figure 4.17 Class diagram for WRIMSAA	61
Figure 4.18: Home screen display	62
Figure 4.19: Login screen display	63
Figure 4.20: Main menu screen display	63
Figure 4.21: Referral data list screen display	64
Figure 4.22: Referral detail screen display	64
Figure 4.23: Edit referral form display	65
Figure 4.24: Add user form display	65
Figure 4.25: Physical Database modeling for WRIMS	67

## **ABSTRACT**

**Background:** Web-based referral Information management system presents a multitude of potential benefits for health system especially for those in developing countries. It facilitates consultation, reduces unnecessary face to face specialist visits, makes handling of referred patient data efficient, and improves quality of care in referral unit through timely and informed decision.

**Objective:** This project, therefore, aimed to develop a Web-based referral Information management system for hospitals and health centers in Addis Ababa, WRIMSAA. By taking referral centers in Addis Ababa, attempt was made to introduce a Web-based referral Information management system that may substitute the existing traditional referral system.

**Methodology:** In order to meet the objectives of the project, the researcher has performed requirement gathering and analysis prior to employing an Object Oriented Analysis and Design methodology. Analysis, design and implementation of the proposed system, WRIMSAA is performed. UML tools for analysis, design phases and programming tool Java, web application scripting language JSF, JSP embedded with HTML and MySQL database language was employed. Integration of developed modules and testing was the final task performed to bring the system functional.

**Result:** Therefore, in this project the researcher developed a Web-based System, WRIMSAA that helps referral service providers to refer, allocate and redirect patients to alternative facilities with minimal time. A database is developed with integration to web application that improves management of referred patient data at all referral centers, using developed databases the users at referral center can access referral data, anywhere and anytime. Finally, WRIMSAA tested and evaluated by users at referral centers.

**Conclusion:** Improving the existing referral system using Web-based Referral Information Management system plays an important role in enhancing the quality of referral service provision in Ethiopia. It has a great impact in reducing morbidity and mortality rate by providing on time referral service which can be achieved through effective communication, coordination, sharing of patient medication data using Web-based system.

**Key terms:** Web-based Referral Information Management System, Object-Oriented Analysis and Design, Hospitals, Internet, Addis Ababa

# CHAPTER ONE: INTRODUCTION

## 1.1 Background

Ethiopia is the second populous country in Africa that shows the lowest health status in the world.<sup>1</sup> Ethiopia's health care system is among the least developed in Sub-Saharan Africa and is not at present able to effectively cope up with the significant health problems facing the country. Widespread poverty, poor nutritional status, low education levels and poor access to health services have contributed to the high burden of the country .<sup>2</sup>

Addis Ababa city Administration Health Bureau (AAAB) was established in 1993 and is authorized to organize, coordinate and regulate public health activities in the city through improving the health status of the residents of Addis Ababa by providing quality and comprehensive health service. Some of the problems faced by the bureau include poor health information, lack of integration and co-ordination in service delivery that resulted in poor health service delivery and quality care, failing to meet the satisfaction level of the end users.<sup>3</sup>

In countries like Ethiopia with scarce specialized human resource for health, patients are usually referred with the purpose of contributing to the effort to reduce the critical shortage of specialized human power<sup>4</sup>. The process of generating a clinical referral for a patient and the resulting transfer of information among hospitals and health centers are key components in the struggle to deliver less costly and more effective clinical care.

On top of this, modern patient care requires collaboration among health care workers belonging to various functional units, departments and hospitals. However, the qualities of health care in Ethiopia are still undermined by weak referral systems. The current referral system in Addis Ababa organized to include all health centers and hospitals. The referral done first at health centers and hospitals accept the referred patient based on their catchment area. The major gaps seen in the existing referral system includes transferring referred patients data with incomplete socio-demographic and clinical information which make the diagnosis process in hospital difficult, delay in patient placement process, paper - based referred patient data recording system at health centers, poor communication and fragmented service delivery among referral health facilities. More importantly the inefficiency seen in referral facilities is worse when the referral case is complex and emergency.<sup>4</sup> So it becomes apparent that bringing technology to support referral system which is effective, accessible, integrated and efficient has a great importance.

Referral management systems may include Web-based referral, services directories, and referral tracking systems and secure Electronic mail (E-mail) facilities to enable communication between health professionals in primary and secondary care.<sup>5</sup>

The Web-based technology that is facilitated by internet provides a multifaceted benefit in improving the existing referral system in Ethiopia. Some of the benefits it brings include timely provision of service to save life through well-organized communication, efficient allocation of patients, a means of sharing and retrieving patient medication details related to referral services provision that enhance timely service delivery.

In general sharing patient referral data over the web is cost effective and fast by giving health care providers easy access to patient medication information from any locations. Therefore, it is necessary to design and develop a Web-based referral system for health care providers in Ethiopia.

## **1. 2 Statement of the Problem**

In order to obtain appropriate medical care, patients can be referred to or transported from one hospital to another based on the capacity, capability and quality of medical care provided by hospitals. Referral systems instituted without appropriate protocols that indicate when, where and how to refer and transfer patient data are bound to face problems during implementation. So ensuring access to medical treatment in case of complication needs to consider "time" factor.<sup>4</sup>

The data collected from Yekatit 12 hospitals referral department through observation on operational time shows that, on average 5 of the referral forms sent to the hospitals were not complete and 2 additional was unreadable in a day which results in difficulty to understand the referral case where the patient was sent to. In addition, a large number of patient requests for service which was much more than the capability to handle by the hospital and no communication between hospitals and health centers to solve information gaps appear in the process.

In general, it was observed and identified by the researcher and indicated by the practioneris was that

- Patient records in Addis Ababa referral hospitals are kept and shared locally using desktop applications which results referral information to be shared at hospitals which did not integrate health centers.

- Health centers use a paper-based patient record system, which is usually incomplete and time taking to access and process.
- Patients allocated based on the catchments areas defined but there is no mechanism to plan redirect patient to alternative health facility when varied number patient refer in different hospitals which brings load in some hospitals.
- Poor communications prior to, in the process and after service delivery and delay in the placement of patients contributing to its inefficiency.

Therefore having a WRIMS (Web-based Referral Information Management System) that introduces a well-organized management of patient data, timely sharing of referral related data, flexibility to patient allocation and fast communication among health care service provider is important.

## **1.3 Objective of the Study**

### **1.3.1 General Objective**

The purpose of this project is to develop a Web-based Referral Information Management System for hospitals and health centers in Addis Ababa.

### **1.3.2 Specific Objectives**

The specific objectives of the study are:

- To work with referral users to define user and system requirements that addresses users' needs.
- To analyze system requirement of the system under study
- To design system for WRIMSAA.
- To develop a prototype for the referral system.
- To test WRIMSAA by referral practioners at referral facilities.

## **1.4 Research Questions**

This study tries to answer the following questions.

1. What are the user requirements of referral centers in Addis Ababa to develop WRIMSAA?
2. What functionalities should WRIMSAA should have?
3. What data should WRIMSAA generate and provide?

## **1.5 Rationale for the Study**

In countries like Ethiopia strengthening the referral system is absolutely essential to reducing morbidity and mortality. However, in developing countries including Ethiopia, the referral system is often weak.<sup>4</sup> It is the task of health referral management system to quickly and easily deliver treatment for patient who is referred. The consequence of not getting timely treatment results in loss of life which can be improved through an effective, efficient and integrated WRIMS that incorporates health care providers.

In response, referral health care providers in Ethiopia have to implement WRIMS for providing service that

- Brings faster and simpler referral system which facilitates management of referred patient medication data and timely communication among service provider.
- Helps referral service provider to make possible redirection of patient to different facility prior to service delivery.
- Makes easy to share best practices among technical staff and timely report generation.

By taking the advantage introduced by WRIMS, the researcher strongly believes that developing WRIMS has an enormous benefit in providing a better referral system in particular and health care system in general in Ethiopia.

## **1.6 Application of the Study**

The result of this study can be used to take Referral Information Management system domain one step ahead by:

- Improving the referral service to be delivered to clients with minimal time through Internet, as time greatly affects referral service provision.
- Improving the patient medication data management system through implementing a database management system for hospitals and health centers which can be shared and accessed by all interested.
- Facilitating patient allocation system to be flexible in contrast to fixed allocation of patients in hospitals through prior information which helps health centers to redirect patients to alternative facilities.
- Improving the reporting mechanism much easier and on time which can help to make on time decision.

In addition, it can be adopted at national level which incorporates a large number of health care providers.

## **1.7 Scope of the Study**

This study was limited to developing WRIMS for all hospitals and health centers found in Addis Ababa because of time and budget constraint. The main focus of the study was to develop a system which can handle and transfer patient medication data in outpatient department of referral service unit starting from the time where the patient being referred from health center until the patient get diagnosis at hospitals. Financial issues which are related to referral tasks are not considered in a system as it depends on external factors beyond health facilities.

## 1.8 Organization of the Document

This thesis is organized in five chapters.

**Chapter One:** Discusses background to referral system, problems it faces, the objective set to be achieved by the study, research questions for the study, the reason for the study, application of the study and the tasks that are included under the study.

**Chapter Two:** A thorough reviews of literature on problems, concepts, technologies, methodologies and finding of similar researches related to Web-based information management system is discussed.

**Chapter Three:** Describes the methodology employed to conduct the research, object oriented methodology, which is a well-known methodology and develop the system through phases of analysis, design and implementation and the conceptual framework of the study.

**Chapter Four:** Describes the system development steps followed to develop the desired Web-based referral information management system.

**Chapter Five:** The final chapter provides conclusion, recommendation, and contribution of the study, strength and limitation of the study. References and annexes are also part of the study.

## 1.9 Operational Definition

1. **Data Base Management System:** An application that facilitates data management in a computer system.
2. **Health Care System:** A system organized to meet health care service needs of the whole population.
3. **Programming Language:** It is an application language that used to write codes in developing software.
4. **Referral:** The process of transferring patient to another facilities based on the limitation in capability at referring institution.
5. **Referral System:** An organizational network for providing referral related service among health care providers.

## **CHAPTER TWO: LITERATURE REVIEW**

### **2.1 Introduction**

The literature review enabled the researcher to gain a better understanding of the research topic. In this study a wide range of resources were examined including articles, journals, reports, magazines and books. Basic theoretical concepts about Web-based and referral system are discussed. Various aspects of WRIMS and related issues including the advantage it brings, their implementation requirement, methodology followed, technology employed and system developed discussed.

### **2.2 Web-based System**

The World Wide Web (known as "WWW" or "Web") is the source of information over the Internet. It is growing at a phenomenal rate. Delivering data over the web is cost effective and fast, and gives Internet users easy access to databases from any locations. For this reason, databases are now largely Web-based. It is believed that the integration of the web and database technology will bring many opportunities for creating advanced information management applications.<sup>6</sup>

Web-based technology platforms allow for seamless, integrated delivery of desired services to users with requisite access rights and privileges. Web-based and Web-enabled tools provide greater capacity to achieve efficiency in collecting and analyzing data and disseminating results.

### **2.3 Referral Management System**

Referral management is a tool used to monitor, direct or control patient referrals. In health care delivery systems, referral is a set of activities undertaken by a health care provider or facility in response to its inability to provide the quality or type of intervention suitable to the need of the patient. It is identified that using referral management services, including the tracking of referrals, reduce waiting times for uncomplicated cases and allows appropriate redirection of some referrals where an alternative service provider can arrange a more convenient appointment.<sup>7</sup>

The activities at referral hospitals can be seen at two levels. Referral management centers receive referrals from primary care. In addition to analyzing referral data (as do the more basic referral information services), they may link with patient booking services and decide the treatment route for patients. On another level referral includes referral from the community to the highest level and back, that is, a two-way referral system. To be effective referral should be a two-way process

that requires coordination and information exchange of patient data between the referring facility (usually at the primary care level) and the first referral hospital.

## **2.4 The Role of Medical Referral System for Referral Decision**

Having a medical referral systems helps to achieve high standards of care by improving patient outcomes and decreasing costs through optimal use of medical services. The success of medical referral system is highly dependent on the quality of referral decisions. A referral decision is a clinical decision by which physicians determine referral indication, type of required services and selection of appropriate providers for referred patient. This inherently complicated process depends on a complex mix of patient, provider and healthcare system determinants. By considering this, improving referral process requires communication and coordination of referred patient information among service provider using IT (Information Technology).<sup>8</sup>

## **2.5 Web-based Referral Information Management System (WRIMS)**

Design and development of Web-based systems is different from development of 'traditional' IT-based systems. Web-based system is mainly an interaction medium in a WWW environment and new applications will be developed and assembled by cloning existing components. Web-based information systems are also different from traditional information systems in the sense that they require new approaches and often are results of grass-root efforts. The interactive aspect of web system designing get explicit focus on web applications that mediate interactions among multiple distributed collaborating actors.

A Web-based application is used usually in conjunction with a website and database. It helps to automate and handle the entire process, enabling applicants to file their applications online and track the progress. The users have the ability to administer the application, update the status and progress and communicate with the applicants and assessors all online.

Web-based system enables nurses, health agencies, hospitals, rehabilitation facilities, and other extended care organizations to more efficiently track and manage all inbound patient referrals and their associated medical documents by automating the entire referral management process.<sup>9</sup>

The next section discusses the literature which shows some countries who implemented Web-based referral management system.

### **2.5.1 South Korea**

In South Korea there were numerous problems in referral service prior to Web-based information sharing, including missing or delaying clinical information when referring patients to a nearby secondary or tertiary care hospital. Patient referral is usually done at random but no reliable information was provided for appropriate care. Furthermore, no follow-up information was sent back to referring institutions when the care was completed such as discharge summary.

The Web-based information sharing service implemented at patient referral center in Asian Medical Center, a tertiary care teaching hospital located in Seoul, Korea serviced 11,587 patients in 1998. During hospitalization, each referring physician is given access code to check status of the referred patient. When the patient is discharged or transferred back to the referring institute, discharge summary and other adequate information are provided through simple file transfer.<sup>10</sup>

In general, Web-based referral management system and referral network improve information flow and communication so each party involved in patient referral has the information they need and when they need it.

### **2.5.2 USA (United States of America)**

E-referrals with specialists have been found to improve care coordination in the Green Field Health primary care practice in Portland, Oregon .E-referrals (Electronic referrals) which is implemented on Web-based system has a number of advantages in Portland; it can hasten access to specialists, reduce costs, and improve care coordination. Some practices have implemented E-referral systems in which primary care physician's E-mail data which includes patient's medical history, physical examination, laboratory tests, and radiographic results which make the specialists to ask specific questions about the patient. If those questions can be answered without the need to see the patient, the specialist E-mails back the response.<sup>11</sup>

The implementation of e-referral systems for gastroenterology, cardiology, and other specialties at San Francisco General Hospital (SFGH) has markedly reduced waiting times for specialty appointments. Group Health Cooperative of Puget Sound uses a secured-messaging system through its electronic medical record in which primary care physicians can send non urgent electronic consult requests to specialists and receive a response within 24 hours. This system appears to reduce unnecessary face-to-face specialty visits while improving the coordination of care.

In January 2007, SFGH launched an E-referral portal that incorporated sub specialist triage, iterative communication with referring providers, and existing Electronic Health Record (EHR) data to improve access to subspecialty care.<sup>14</sup> The key attributes of the SFGH E-referral system include integration of existing EHR demographic and clinical data into E-referrals, centralized triage of referrals by designated sub specialty, and back-and forth communication between referring providers and a sub specialist reviewer. Referring providers complete an electronic template, to which existing relevant EHR information is automatically appended. Sub specialists review requests within seventy two hour and choose one of the following options:

- Schedule next available regular appointment,
- Schedule an urgent appointment, or
- Do not schedule. When reviewers do not schedule an appointment, they request additional workup or information, or suggest alternative appointment.

The referring and reviewing providers can communicate in an iterative fashion until the reviewer decides to either schedule an appointment or they both agree that the patient does not need one. When the appointment is granted, this decision is electronically transmitted to the clinic scheduler, who makes the appointment. The hospital EHR system then generates a letter to the patient and E-mail to the referring provider alerting him/her to the appointment. The E-referral portal keeps a database of all submitted referrals, which serves as a tracking mechanism for both referring and sub specialist providers and clinics.<sup>12</sup>

Another area where e-referral implemented in USA is safety net program, where the sub specialist shortage is severe. E-referral improves access to sub specialty care in safety net settings. The adoption of E-referrals could mitigate the effects of the scarcity of sub specialists especially for the under and uninsured who have fewer choices among sub specialists and longer wait times for appointments. In referral hospitals, longer wait times are associated with delays in diagnosis, greater costs, and worse outcomes. Compared to paper referrals, E-referrals improve the transfer of administrative and clinical information; they may reduce duplicate test-ordering, and improve both the referring and subspecialty physician's ability to make treatment decisions. These attributes could lead to rational allocation of sub specialty visits, improving clinical outcomes while minimizing wasted resources.<sup>13</sup>

In USA the process of generating a clinical referral for a patient and the resulting transfer of information from the primary care physician to the specialist and back again, are key components in the struggle to deliver less costly and more effective clinical care. In response, computer-based outpatient clinical referral application which runs on a Web-based system developed in Atlanta general Hospital. So it facilitates in identifying an appropriate specialist, collecting the clinical, demographic, and financial data required to generate a referral and transferring the information between the specialist and the primary care physician. Preliminary results indicate that the new computer-based referral process is faster to use than conventional methods.<sup>14</sup>

### **2.5.3 Denmark**

In Denmark all non-emergency patient referrals to secondary care are routed through a General Practitioner (GP). Over 90% of GP and all hospitals are able to send and receive E-referrals in Denmark and on average it takes staff a total of 8.5 minutes to process a paper referral and 6.3 minutes to process an e-referral. The average processing cost shows that, on average, it costs hospital staff a total of €9.66 to process a paper referral and €6.48 to process an e-referral. This suggests that, on average, paper referrals cost €3.18 more to process paper-based referrals than e-referrals.<sup>15</sup>

### **2.5.4 Ireland**

In Ireland, the National Cancer Control Program (NCCP) in collaboration with a broad range of stakeholders has developed electronic cancer referral for breast, prostate and lung cancer in 2008.

The objective was to develop an online system for GP, so that they could refer patients with symptoms of cancer directly to the cancer center. This ensures rapid referral of patients with suspected cancer in a secure manner. Once the GP sends an electronic cancer referral, an immediate acknowledgment is given. In addition, the cancer teams will send a response to the GP, with the date of the patient's appointment within five working days.<sup>16</sup> NCCP uses E-referral for a range of cancer types. The referrals are available as:

- An online referral form (Health link Online) from the GP directly to the cancer teams.

- An integrated referral form which goes from GP management system directly to the cancer teams.

There have been 4,300 electronic cancer referrals sent to the eight nominated cancer centers in 2011 and it improves the referral process in a number of ways, including by

- Streamlining the cancer referral process.
- Providing rapid access for patients who are being referred with a suspected cancer.
- Aiming to increase the number of patients diagnosed at an early stage and maximizes the potential for cure.
- Providing automatic confirmation of receipt of GP referral.
- Providing direct access for GPs to the cancer teams at the eight designated cancer centers.
- Reducing communication difficulties.
- Reducing costs for the GP.

The recent report by the health care service provider in Ireland states that implementation of E-referrals will improve patient referral pathway. <sup>16</sup>

### **2.5.5 Zambia**

The patient referral system, which was the first step towards the full Zambia Electronic Perinatal Record System (ZEPRS), was developed with high levels of Lusaka user input and involvement of local technical staff. The launch of the electronic-first referral system generated great excitement at the clinics in Zambia because clinic staff was now being informed of their patients' outcomes after pregnancy complications: whether they survived, whether they delivered their babies, what sex the babies were, and other outcomes. This ability for two-way communication about patients between the clinics and Hospitals was, in fact, a great incentive for clinic staff to fill out the information when they referred a patient.

According to observations and monthly reports presented to CIDRZ, not all patient referrals were entered within an "immediate" (emergency) status; staff seemed not to abuse the notion of priority patients. Having this acknowledgment and disposition data filled out for hospitals had been a key motivator for clinic staff to use the referral application. The greatest problem in usage of the referral application was that the hospital staff sometimes lacked the time to fill out the referral acknowledgement and dispositions.

The development of the Web-based electronic-first perinatal patient referral system was launched in May 2004 as an interim step toward the full ZEPRS.<sup>17</sup>

## **2.6 Online versus Paper-based Referral**

This section discusses the advantages of using online referral service over traditional paper-based one. It presents different experience from countries that have implemented Web-based referral system.

### **2.6.1 United Kingdom (UK)**

The study conducted in UK hospitals shows that accurate handover of information is very important. Previously, referrals were notoriously difficult to track and review patient documents due to poor quality of written paper-based recorded information for handover (illegibility, incomplete paper trail, repetition of information and loss of patients records). There is strong evidence to suggest that the poor quality of paper-based referral letters. In a survey conducted in London indicates that thirty eight percent of specialists in outpatient departments reported referral letters which contain inadequate information. Information gaps include:

- A poor explanation of why the patient is being referred and the desired outcome.
- Missing clinical information, including the results of examinations and investigations, provisional diagnosis and psychological and social details.

Significant problems of illegibility and missing information were identified with the paper-based referrals. In comparison, hundred percent documentation was achieved with the online referral system.<sup>18</sup>

### **2.6.2 USA**

In a study conducted on Hospital for Sick Children (Sick Kids) in USA, new online referral system support more timely and efficient management of referral health care. It holds great benefits for the referring professionals, child and hospital. The online referral system at Sick Kids makes the entire referral process faster and simpler by providing automated responses to let users know:

- When a referral has been received
- Whether it has been redirected, and to whom and

- When the patient has been booked for an appointment

In addition, patients and health care service provider benefit because:

- The referral is immediately directed to the right clinic
- Patients get the right care at the right time
- Staffs spend less time on the phone, because clinics are not calling for more information and you always know the status of the referral.<sup>19</sup>

The conclusion made from the studies indicates that the traditional written paper-based referrals do not provide an acceptable level of documentation and recommended to use a fully integrated Web-based system that may offer a solution.

## **2.7 Selected Web-based Referral Systems**

The next section describes some of the examples of Web-based systems in use. It tries to highlight the typical functionality and system feature being used.

### **2.7.1 Web-based Patient Placement Referral System**

A patient placement referral system in USA allows continuing care providers to instantly receive referrals and inquiries directly into referral management system that were submitted through forms on users Web sites. It includes the following capabilities to improve visibility and efficiency.

- Electronically E-mails to staff alerts such as "New Referral Received", providing automatic reminders to ensure staff is on top of every referral.
- Sends automatic updates back to the referral source, reducing phone tag.
- Provides a real-time view into the status of all referral activity for one location or all locations with control of all inbound patient referral activity.
- Includes reporting and analysis at every level - by location, by referral source, by user, by line of business, by payer and more.

In addition, custom reporting allows for workflow decisions based on current, accurate referral information. According to the study done on Atlanta, the patient placement system capture referrals instantly and effortlessly translates to quicker response times and greater revenue potential. Admissions teams in the system received an alert of the new referral, and with the

referral already entered into the system, the staff can immediately assign and track approval activity to streamline the entire admission process.<sup>19</sup>

Referral submission forms provide a simple mechanism to increase the number of patient referrals. When a "contact-us" or similar form is submitted through an organization's Web site, Web-to-Referral automatically transfers the information into referral management system. None of the referral information provided in the form fields needs to be re-entered, which eliminates manual effort, typographical errors and delayed response times.

Another study done in UK shows that Patient Placement referral management system an easy and low cost Web-based software solution is designed. It is designed for nursing homes, home health agencies, hospice services and other long-term care providers to automate and accelerate admissions and intake. Patient Placement system designed the low cost Web-based subscription service specifically to automate every step of the long term care admission process, including insurance verification, form creation, and reporting. Inbound faxes convert automatically to electronic documents eliminating long fax waits and mountains of paper.<sup>20</sup>

In general, the Patient Placement Systems developed helps continuing care providers to admit more patients, generate more revenue, compete better and respond faster with simple, efficient Web-based healthcare software that automates and accelerates admissions and referral management.

### **2.7.2 Internet Referral Information System (IRIS)**

IRIS is a Web-based patient referral system that incorporates clinical rules of the practice's design, along with flexible order processing and management capabilities. IRIS has been designed for patient referrals between Primary Care Providers (PCP), Hospitals and Specialists. IRIS was developed to allow providers to interact easily by removing obstacles such as IT system incompatibilities, operational policies, and training procedures.<sup>21</sup> IRISs' sole requirements are a workstation with Internet access and Microsoft Internet Explorer 6.0 or newer.

IRIS

- Clarify the reasons for the referral.
- Facilitate patient condition communication with in referral health care providers.
- Identify and insure complete workup.
- Communicate standards of care in a community.

- Direct referrals to the most appropriate service.
- Match provider/payer acceptance before order placement.
- Assign priorities based on clinical conditions.
- Identify and manage all documentation requirements.

### **2.7.3 Physician-to-Physician Community Referral System**

The Portland Inter hospital Physicians Association launches ‘Portland Inter hospital Physicians Association (IPA) Provider Connect’ as a pilot project to connect two large primary care clinics and their referral network of specialists.

On January 9, 2012 the Portland IPA, Oregon’s largest physician group comprising more than 2,500 primary care and specialty physicians launched an easy-to-access system to assist with generating and tracking referrals, tracking patient utilization and facilitating an expedited process for obtaining prior authorizations and ordering diagnostic tests for physicians and health organizations.<sup>22</sup>

The web tool can be customized for each physician practice, health site or community clinic and fit in with each clinic or health organization’s existing referral process to ensure tracking, decision support and follow-up for each patient. Each physician or administrator can customize it per their current referral patterns. When fully implemented, Portland IPA Provider connect and integrates as many as 300 primary care and specialty physicians, hospitals, nurse practitioners, and allied providers. The designed system is not only accountable and efficient in managing patient referrals but it also transforms typically fragmented physician-to-physician communications and facilitates physician relationships to maximize timely and tracked care delivery.

This easy-to-use, Web-based technology is a simple platform that works among public and private physicians, hospitals and other health organizations regardless of the entity’s current health technology capabilities and regardless of what kind of EHR system an organization has or doesn’t have. As health providers and insurers focus their efforts toward creating Coordinated Care Organizations (CCO), a connecting piece improves how health workers process referrals, manage patient information and track outcomes.

#### **2.7.4 E-health Access Referral Management System**

E- Health access referral management system is designed to electronically manage patient referrals and the new patient intake process. It allows for appointments to be more expeditious, increases staff efficiency, increases data accuracy and reduces costs. Rochester Electronic Health Global Technologies, Inc. (EHGT), New York (NY), is one of an organization that works on developing referral management system. It continues to make the process of integrating outside medical records and imaging into referral patient workflow more cost effective and efficient, while improving the coordination of care for all patients transitioning to a new facility.

It is a value-added capability that enhances existing Electronic Medical Record (EMR) and scheduling systems.<sup>23</sup> This system enables electronic and Web-based management of new patient referrals, while seamlessly integrating referred patient information into hospital scheduling systems. E-health (Electronic - health) Access Referral Management System leverages a unified environment for managing fax, phone, or electronic orders, while managing all related patient documents and images in a patient-centric easy-to-use web interface. Workflow is built in, and allows for easy collaboration on new patient referrals for the entire care team.

E-health Access Referral Management System saves valuable staff time and greatly reduces the potential of manual entry errors by eliminating the need to manually input patient information into various hospital systems, in addition to converting a paper-based process to an electronic one. No more paper faxes and integration with internal scheduling and EMR systems. Further it reduces cost, increases staff efficiency, and allows for quicker turnaround for patient appointments. Real-time management reports provide insight into who is referring patients, plus the value of each referral, leading to more effectiveness. E-health Access Referral Management System enables the care team to automatically route reports, information, status updates and send confirmations to the referring physician. These can be accessed online, cutting staff time and dramatically improving service to these physicians.

#### **2.7.5 Strata Pathways**

Strata Pathways is a Web-based interface which can be seen as a stand-alone, Web-based application or as a fully integrated component of an EHR referral system developed in UK. Strata Pathways supports a “Connect All”, rather than a “Replace All” strategy, which minimizes both change management and expense while optimizing clinician workflow, usability and

adoption. As new EHR components are added or integration capabilities increase, Strata Pathways design allows for increased interaction with other EHR components.<sup>24</sup>

Some of the Strata Pathways system functions include:

- Acute care clinicians initiating referrals from their hospital or community information system, with single sign-on and patient context.
- Clinicians sending e-referrals from their existing EMR, with both single sign-on and patient context.
- Referral status updates being sent from Strata Pathways to existing hospital or community information systems.
- Service providers receiving e-referrals in their own information system, or by using the Strata Pathway Web-based interface
- Patients and their families utilizing patient portals such as Microsoft Health Vault (MHV) to check on the estimated wait times, service provider information, and status of outstanding referrals.

## **2.7.6 Health Insurance Portability and Accountability Act (HIPAA)**

A free online referral case management system to the clients designed known as HIPAA offers a variety of useful tools for clients to manage their referrals online. Through password-protected system, clients can:

- Request an appointment
- Upload electronic records for physician review
- Upload cover letters for physician review and comment
- Check on status, record and review referrals
- View, download and print final reports and invoices
- Send a message to referral staff or physician regarding a case
- Search for a physician
- View physician credentials, including CV, testimony list, and biography

In addition, supervisors and/or managers may also view all assignments submitted by their team. Closed cases remain online so that copies of final reports and/or invoices may be viewed, downloaded or printed at any time.<sup>24</sup>

### **2.7.7 ZEPRS (Zambia Electronic Perinatal Record System)**

This referral application was released as a prototype for the main ZEPRS application. It was the first electronic-first perinatal referral system in sub-Saharan Africa and enabled users to enter patient data in real time.<sup>17</sup> Some important features of ZEPRS include:

- Centralized authentication and authorization scheme,
- Notification to referral clinic or hospital for incoming patients,
- Provision of critical information to prepare for patients' arrival,
- Notification to referring clinic and hospitals about patient status, and
- Record management of all patient referrals.

## **2.8 Web-based Referral System Methodologies**

In this section the methodology employed by different organization in developing a Web-based system from different literature is discussed.

### **2.8.1 Refer ASmoker.org**

ReferASmoker.org is a point-of-care e-referral portal that allows providers to e-refer smoking patients to an online smoking-cessation portal. It uses the following methodology to develop the system.<sup>25</sup>

#### **2.8.1.1 Study Design**

Four-step usability and pilot testing approach applied which consisted of

##### *Phase I: System Conceptualization*

System conceptualization was done using Delphi technique which is a systematic forecasting method for reaching consensus regarding prediction of usability and feasibility. It helps to identify key functionalities that would overcome barriers in providing referrals for smoking. Delphi technique was a useful communication strategy that provides a structured process for the reliable and creative exploration of ideas suitable for decision making. Controlled opinion feedback sessions were used to establish expert consensus without certain social interactive behaviors that can hinder opinion forming in a typical group discussion. A panel of 8 experts that included physicians and psychologists with expertise in health services, tobacco control, and informatics participated in the Delphi process.

### *Phase 2: Programming and Usability testing*

Web system programming and refinement using agile methodology, was used to iteratively strategize and plan the programming of the ReferASmoker.org e-referral system. Agile software development was advantageous because developers can adapt to changing requirements based on the short-term goal setting and collaboration. This approach has also been demonstrated to reduce development time and risk. The ReferASmoker.org Web-based system was programmed using Microsoft's ASP.NET version 3.5 and C# technology. Microsoft SQL Server version 2000 was used as the database. The study used programming best practices in the form of design patterns and modular architecture.

Usability of the system was assessed using the "think-aloud" approach. In this approach, while participants are reviewing the system's content and interacting with the program, they are asked to vocalize thoughts, feelings, and opinions.

### *Phase 3: Implementation Planning*

Implementation planning was done using the Nominal Group Technique (NGT) for the effective deployment of the system. Once the primary processes were identified, conducting NGT session to collect feedback on the referral system and plan for implementation in practices was done. NGT is a highly structured, multi-step, consensus-building procedure often used in formative research to elicit and prioritize group responses to a specific question. It is a consumer-oriented formal brainstorming or idea-generating technique used to foster creativity and to effectively prompt group members to articulate meaningful disclosures.

### *Phase 4: Pilot Implementation and Evaluation*

Pilot testing was done to identify practice recruitment and system-use barriers. The study tested implementation of the system to identify recruitment barriers and areas of refinement in the system. Health care provider staff recruitment is done from some selected clinics to participate in the pilot study. Practitioners in the pilot were representative of participants in planned larger trial.

#### **2.8.1.2 Setting and Sample**

For system conceptualization, experts were recruited from multiple academic institutions. For usability and pilot testing, the study selected practitioners that would represent the sample in

planned randomized trial. Thus, physicians and nurses from community-based practices across several states in the US were recruited. For implementation planning sessions, recruitment of physicians from a university setting is done. The study was approved by the institutional review boards at the University of Alabama at Birmingham, the Johns Hopkins University School of Medicine, and the University of Massachusetts Medical School.

### **2.8.2 A New Methodology for Developing Web-based Systems**

Another study by Omer and his colleagues proposed a methodology for the development of Web-based systems, which they called “Towards a new methodology for web-based system”. Their methodology for developing Web-based system for intranet and web application and is spiral one divided into four stages discussed below.<sup>26</sup>

#### ***Analysis phase:***

The first phase in developing the Web-based system and it includes the following steps.

- Collect customer requirement and content needed to be showed.
- User classification.
- Determine and identify web application page/pages needed to each user requirement.
- Assign development priority to each user requirement according to the customer need.
- Develop information or content architecture which includes a conceptual design of the overall site structure, structure of individual sub-sections, overall navigation methods as well as cross-linking between section.
- Determine test cases for dynamic pages.
- Conduct environment analysis including identifying technology for Web-based system such as Linux, Apache, MySQL (My Structured Query Language) and PHP(Hyper Text Preprocessor), Windows, Microsoft IIS(Internet Information Server), Microsoft SQL(Structured Query Language) Server and ASP(Active Server Page).

#### ***Design and Development phase:***

- Design tasks i.e. Images, animations- headers, menu and flash.
- Development tasks i.e. HTML (Hyper Text Markup language), Script and CSS (Cascading Style Sheet).
- Tools i.e. Adobe Photoshop, Dream weaver or any other tool.

- Design and development of the page

***Test phase:***

- CSS validation, Links validation, Script test, HTML test, Apply test cases (intranet and web application) and user test.

***Integration and Release phase:***

- Integrate and link pages together and perform integration test.
- Publish the release.

## **2.9 Summary**

The literature review above tries to show a number of implementation aspects for referral Web-based information management system including the technology, methodology employed to develop the system, the problems associated with using the traditional referral system and the solution made through using Web-based system. Overall, it tries to make clear the importance of developing a Web- based system for integration and coordination of referral service provider for delivering better, fast and timely service for referred patient which finally improves clinical outcomes.

## **2.10 Related Works in Ethiopia**

To the knowledge of the researcher, no study has been found in developing a WRIMS in Ethiopia. Even if desktop application is developed for referral hospitals in Addis Ababa which helps health care providers to transfer referred patient data locally and some research indicates the need for E-referral and a need to have a multi-agent decision making in referral service provision, there is none done towards Web-based referral system that integrate all stakeholders involved in referral service provision. Hence, this study tries to introduce a new application for referral information management system which can contribute towards a better approach for integration and coordination of referral service providers in Addis Ababa.

# CHAPTER THREE: METHODOLOGY

## 3.1 Introduction

Web-based applications are becoming business critical, and that this would be reflected in the organization of the development work. The methodology applied to develop any system greatly affects the system efficiency in various aspects. Through taking this into consideration, this study employed a Web-based methodology rooted from the literature reviewed in previous chapter, a new methodology for Web-based system with some modification. The selected methodology helps the researcher to understand the existing referral system as well facilitates the development of the new system, WRIMSAA.

In this chapter the researcher outline

- The conceptual framework of the study
- The methodology applied in this study and the rationale for adopting the research methodology.
- The process of collection and analysis of data followed to understand the existing referral system.
- The methodological steps followed in the study during the development of the WRIMS for Addis Ababa.

## 3.2 Conceptual Framework of the Study

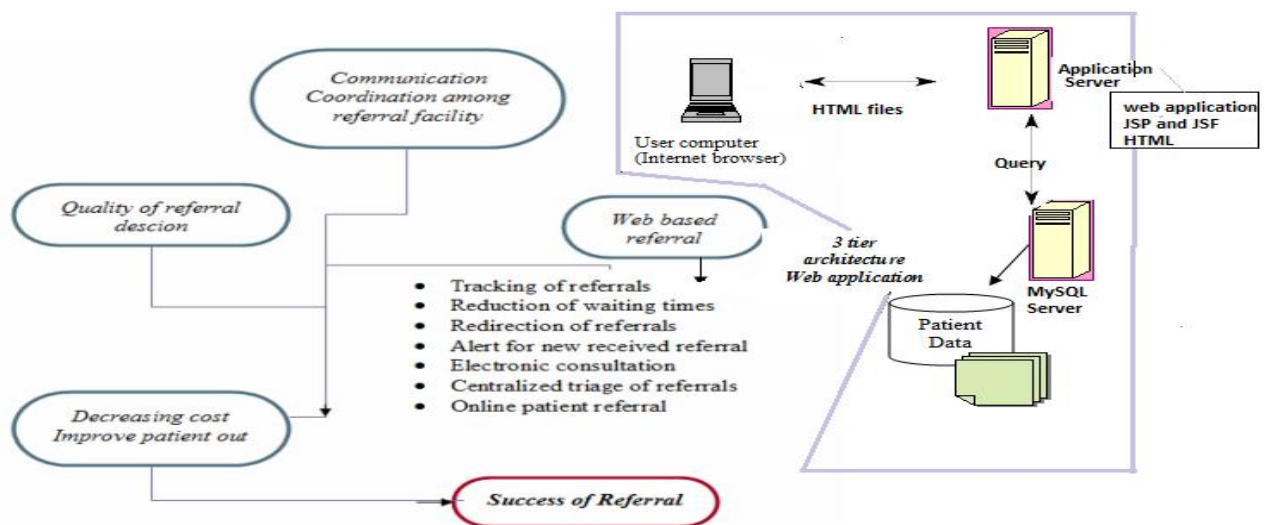


Figure 3.1: Conceptual Framework of the study

Referral system success relies on different issues which bring to achieve users' goal. The above conceptual frame work shown in fig.1.1 tried to indicate some of the factors which contribute to success of referral system.

- **Communication and Coordination of Referral Centers**

One of the major problems seen in Ethiopia referral system now days is lack of integration in providing referral service. It greatly affects referral facilities capability to make timely decision for referral cases. Having a fast and timely communication plays an important role in reducing morbidity and mortality for referred patients.

- **Quality of Referral Decision**

Any decision which lies on unreliable information leads to wrong decision in any system. Quality decision results from effective, reliable, accurate and timely communication among referral facilities which empowers professional to make the right decision at the right time.

- **Web-based referral system**

An internet based system facilitates handling of patient data form anywhere, at any time. It becomes the solution for many systems especially for service oriented organization. In relation to this, the need for developing and implementing Web-based system get attention of system developers. Web-based system integrate web with database that differs from the known web application systems. It provides a greater advantage for users in referral system and some of the benefits it brings described above in fig. 1.1.

### **3.3 Web-based System Development Methodologies**

A system development process methodology describes an approach to building, deploying, and possibly maintaining software. There are different methodologies for developing system including Web-based application, Database application, User Interface (UI) designing and Usability designing.<sup>27</sup>

For this study, the researcher used an OOA/D (Object Oriented Analysis and Design) methodology for developing a Web-based application for referral centers in Addis Ababa. Referral system is composed of interacting objects including physicians, health centers and

hospitals. Therefore implementing OOA/D methodology which focuses on interacting objects in the real world provides a number of advantages.

### **3.3.1 Object-Oriented Analysis and Design methodology**

OOA/D methodology applied in this study integrate referral data used in referral centers and process concerns which acts on a referral data in providing the required service for clients, patients, in referral system into constructs called objects. It follows a modular approach for each concepts identified in referral systems which is easier to understand, correct and modify, that makes it more popular and preferable methodology for this study.

The methodology employed helps to analyze and design a system requirement identified from requirement document from an object-oriented point of view. There are many tools available for designing and developing OO systems whereas this study uses Java, a well-known object oriented programming languages, for developing WRIMS for Addis Ababa referral hospitals. In relation to employing an OOA/D methodology for developing WRIMS, UML (Unified Modeling Language) artifacts is used to model the system's objects with pictures with various perspectives such as structure and behavior. The following are some of the advantages in using OOA/D in the study to develop the WRIMS.

#### ***Increased reusability***

The OO (Object Oriented) paradigm provides opportunities for reuse through the concepts of inheritance; polymorphism, encapsulation, modularity, coupling, and cohesion are all straight forward concepts that lead to better design. For instance, any requirement changes in referral systems can easily integrate to the designed system which makes easy to reuse for future developers. Although the use of OO does not guarantee you will develop reusable software or that you will, in turn reuse software yourself, it does offer significantly more opportunities for reuse than the structured paradigm.

#### ***Increased Extensibility***

Because referral objects was organized as classes, systems which have both data and functionality when you add new features to the system you only need to make changes in one place, the applicable class. This is different than in the structured world where a change in a single business rule could affect many programs.

### ***Improved Quality***

Quality systems are on time, on budget, and meet or exceed the expectations of their users. One of the difficulties seen in system development using structured system methodology is inability to integrate any change in the process of system development which makes inefficient. OOA/D methodology allows users in referral system to have a greater opportunity to participate in system development and integrates any requirement change in development process. More importantly, it contributes to success of the referral system through meeting users' goal by implementing a prototype of the system that is subject to modification.

### ***Financial Benefit***

Reusability, extensibility and improved quality are all technical benefits. While they all sound like good things, the reason they are important is because they lead to the business benefits of OO. From the point of view of our users, the real benefits are we can build systems better, faster, and cheaper.

### ***Reduced maintenance burden***

Since the methodology supports to use a well-defined documentation tools and object orientation reuse change incorporation is easy the maintenance cost is reduced.

## **3.4 Web-based System Development Phases**

In developing WRIMS the study used an OOSD (Object Oriented System Development) life cycle, a series of life cycles followed in OOA/D methodologies. It allows developing the system under study progressively with a representation of referral system component or object through the phases of object oriented analysis, object oriented design and implementation.<sup>27</sup>

### **3.4.1 Object-Oriented Analysis (OOA)**

OOA phase in developing WRIMS emphasizes on investigation of the problem and requirements, for referral facilities rather than a solution. It is also called requirement analysis. It focuses on issues like *how will the existing referral system be used? What are its functions?* To create the best solutions, the study follows a detailed need analysis of data from requirement document for determining user's requirements (i.e., defining *what* the system is supposed to do). Compiling a

list of system requirements guides the researcher to analyze the system. To capture what a proposed system should do, the study employed a technique and tool from UML artifacts including:

- Conceptual modeling that helps the researcher to capture concepts in the referral system with their attributes independent of implementation details.
- Data modeling used to represent persistent database objects in referral system.
- UI flow diagramming is applied to show the high level architectural view of system UI.

### **3.4.2 Object-Oriented Design (OOD)**

The study uses OOD phase to design a conceptual solution (hardware and software) that satisfies user requirement (i.e., deciding *how the system should do it*), rather than its implementation. During object oriented design phase of developing WRIMS this study emphasis on

- Designing software objects, classes with their attributes, methods, interfaces and access specifiers using UML artifacts class diagram and sequence diagram. Class diagram represent the static view of the system where as sequence diagram employed to depict the dynamic view of the system under study, WRIMS.
- Designing entity objects in a referral database with their attributes, primary and foreign keys, and association between entity objects and cardinality specified using data modeling.
- Designing/prototyping UI for WRIMS where a user in referral system interacts.

### **3.4.3 Implementation**

In previous phases of developing WRIMSAA, the study used an OOA/D methodology for analyzing and designing purpose. Ultimately, designs can be implemented, and the implementation such as code expresses the true and complete realized design and it includes the choice of programming tool used to develop the required system.

So for this phase the study employed the programming tool:

- Java Enterprise Edition (Java EE) application, an object oriented programming language is used to implement classes which can be best used in designing WRIMS.
- JSF (Java Server Faces), JSP (Java Server Page) a server side scripting language, HTML, a client side web development language is in use for implementing web application.

- MySQL Enterprise Edition for developing object database which mainly contains referred patient medication, system user and patient allocation data on referral network with all data management mechanisms , access, store, update and delete, is defined.

After developing the prototype of the system using the above application tools separately integration and testing of application is performed. Testing is last task performed in OOSD life cycle which brings the system to deployment. The task in testing includes checking the correctness of codes written at development stages to verify the system full and functional.

### **3.5 Data Collection methodology**

#### **3.5.1 Data source**

The data sources for this specific research are the referral unit service staff including physicians, health officers, nurses and technical staffs who involves in tasks related to referral service, documents used for processing referral data and referral service provision environment at health centers and hospitals in Addis Ababa governmental health care providers. For the purpose of requirement gathering which provides an input for OOA/D stages of developing the system and understanding the existing system, the study used the following data gathering tools.

#### **3.5.2 Sampling**

As tasks performed and rules and policy procedures followed to process referral information by individual at referring and referred facility are similar, the study used random sampling techniques to selects sample referral centers. Then 4 hospitals, Zewditu, Yekatit 12 hospitals, Ras Desta and Saint Pawlos and 5 Health centers, Woreda 24, Beletshachew, Lideta, Addis Ketema and Akaki health center managers were selected randomly and interviewed. In addition 2 Health officer from selected hospitals and 1 Nurse and 1 physician from health centers were involved in the process of developing the system under study through providing comments if there was a requirement missed or incorrectly included.

#### **3.5.3 Data Gathering Techniques**

The study employed the following data gathering techniques for collecting data, user requirements, from sampled subjects of the population.

### ***Interview***

Most analysts use interviewing as a primary way of gathering requirement in information system projects. The study conducted an in-depth interview on 4 hospitals and 5 health centers, 9 managers to gather facts and business processes followed to provide referral service in the existing system and additional requirement need to be filled with the new system being developed. Indeed, interviews, open- ended type are important methods to obtain rich information; the study used open- ended type which has 3 parts.

- The first part includes questions which helps the researcher to clearly understand the overall view of the current referral system.
- The second part tries to identify issues related to the performance of the existing system.
- The final part focuses on identifying the changes required in the existing system which can be integrated with the system under study.

### ***Observation***

The study employed observation for getting some issues which can be gathered through other data gathering techniques from referral centers. It includes the waiting time for patients to get service for understanding the efficiency of the existing system, the availability of infrastructure and the difficulties seen in the referral service provision process.

### ***Document Analysis***

In addition to the data gathering techniques employed above the study carried out document analysis including forms such as referral slip, referral register book forms, patient allocation guides known as catchment areas and other formats where referral staff uses to capture referral services related information for delivering required service.

Finally the response for interview was summarized and used in requirement document. Then it is described using essential use case modeling that helps the researcher to understand the existing system very well, helps to identify difficulties for current system in detail and helps identify gaps to be filled by the new designed system. More importantly, the requirement document used as an input for the next stage of system development, OOA. The documents, referral forms and feedbacks forms, analysis help the researcher to design user interface for the system.

To conclude, the study mainly used an approach, which is user-centric that ensured and targeted to involve users in developing the new system. The data gathering techniques followed to capture the required data for understanding the existing system and developing WRIMS. In addition an object-oriented system development principle and methodology provides a lot of advantage for system users at referral department and researchers who involve in improving referral system in the future time.

# CHAPTER FOUR: WEB-BASED SYSTEM DEVELOPMENT

## 4.1 Web-based Architecture for WRIMSAA

Applications in Web-based system are usually broken into logical chunks called "tiers", where every tier is assigned a role. Traditional applications consist of only 1 tier, which resides on the client machine, but web applications lend themselves to n-tiered approach by nature. Though many variations are possible, the most common structure is the three tiered application. In its most common form, the three tiers are called *presentation*, *application* and *storage*.<sup>28</sup>

In this study 3 tier architecture is applied for developing WRIMSAA. A web browser is the first tier (presentation), an engine using some dynamic web content technology (such as ASP (Active Server Page), CGI (Common Gateway Interface), COLDFUSION, PHP (Hypertext Preprocessor), Python and JSP/Java) is the middle tier (application logic), and a database is the third tier (storage). The web browser sends requests to the middle tier, which services them by making queries and updates against the database and generates a user interface.<sup>29</sup> Fig.4.1 below shows a 3 tier application with application server that supports and runs JSP, JSF (Java Server Faces) files, with servlet engines and Enterprise Java Beans (EJBs), where the application server is connected to a database at the back end.

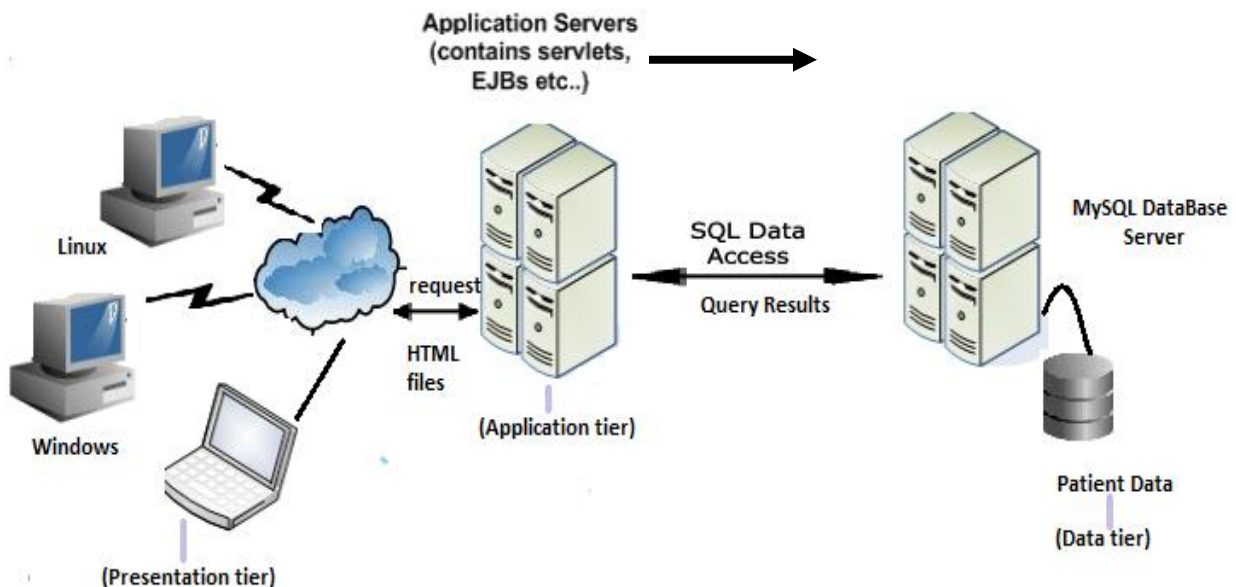


Figure 4.1: 3 - tier architecture for Web-based System

**Presentation tier:** provides user interface on clients' browser and used to translate tasks and results to something the user can understand.

**Application tier:** sometimes called logic tier that coordinates the application, process commands, make logical decisions and evaluations, and performs calculations. It also moves and process data between the two surrounding tiers.

**Data (storage) tier:** store and retrieve information from database system. The information is then passed back to the application tier for processing and then eventually back to the user.

## **4.2 Requirements Analysis of the Existing System**

Requirement phase of system development life cycle identify requirement of the existing referral system and use that as input for the succeeding system development phases, OOA. Requirement analysis is done based on the requirements gathered from interviews and document analysis with potential users of the referral system users and finally organized under requirement specification for WRIMSAA. Use case, UML notation that visualize user requirement is used in this phase.<sup>26</sup>

### **4.2.1 Use cases**

Use cases are used to discover and record functional or behavioral requirements of referral system by writing stories of using a system to fulfill user goals. It is a key requirements input to classic OOA/D. Important value of use cases is that it emphasizes on the user goals and perspective; we ask the question "*who is using the system, what are their typical scenarios of use, and what are their goals?*" This is a more user-centric emphasis compared to simply asking for a list of system features. Use cases can be seen as a collection of related success and failure scenarios that describe an actor using a system to support a goal. A scenario is a specific sequence of actions and interactions between actors and the system.<sup>30</sup>

#### **Actor**

Actor is something with behavior, a person, identified by role, including the system under discussion itself when it calls upon the services of other systems. Actors are roles played not only by people, but by organizations, software, and machines. For instance: Patient, Physician, Referral, Feedback and Database connection.

There are two types of use cases, essential and system use cases. In this study, essential use case was dealt with in requirement analysis and system use case is included in object oriented analysis phase of OOA/D system development life cycles.

#### 4.2.1.1 Essential use case

Essential use case is used to explore usage-based requirements in the existing referral system. It is technology independent. It describes the fundamental business task of the existing referral system without bringing technological issues into account. Often referred to as a task case model or an abstract use case model, models a technology- independent view of your behavioral requirements. Use-case can be shown using use case description which is the textual scenario and use case diagram, pictorial representation of use case stories.

In developing WRIMSAA the study used a requirement document and identified 6 use cases; *refer patient, receive referral, check payment, register patient, register referral result, search patient* and *login* in the existing referral system. The identified use cases are described using textual description and use case diagramming as follows.

#### A. Use case Diagram for essential use case modeling

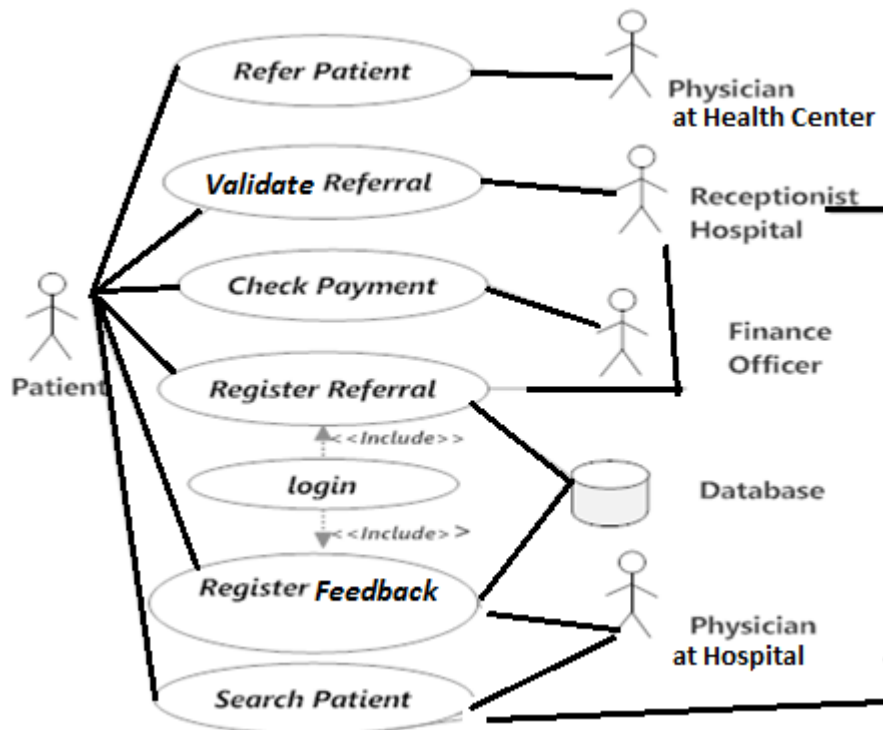


Figure 4.2: Essential Use case diagram for WRIMSAA

## ***B. Use case description for essential use case modeling***

**Use case UC1:** Refer patient

**Scope:** Referral at Health center

**Primary actor:** Patient

**Pre-condition:** Physician determine on referral indication of patient.

**Successes guarantee (post-condition):** The patient is referred and allocated.

**Main Success scenario (Basic flow):**

1. The use case starts when the triage selects the appropriate provider.
2. The triage registers patient referral data on referral form.
3. The triage registers patient general medication information on referral register book.
4. The triage at health center sends referral form to hospital.
5. The use case ends.

**Use case UC2:** Validate referral

**Scope:** Referral at Hospitals

**Primary actor:** Patient

**Pre-condition:** Patient with referral form.

**Successes guarantee (post-condition):** The patient referral data being approved.

**Main Success scenario (Basic flow):**

1. The use case starts when the triage checks the catchments area for referred patient.
2. If the referred patient is from the right catchments area. [Alt 2.1]
3. The triage verifies the completeness of referral data from referral form.
4. If the referral forms data is complete and correct.[Alt 4.1]
5. The triage registers the treatment route on referral form.
6. The triage registers patient general information in register referral book.
7. The use case ends.

**Alternative Scenario:**

- 2.1 The triage returns the patient to referring facility.
- 4.1 The triage returns referral form to referring facility.

**Use case UC3:** Check payment

**Scope:** Referral at Hospitals

**Primary actor:** Patient

**Pre-condition:** Patient with routed referral form.

**Successes guarantee (post-condition):** The patients payment condition is approved.

**Main Success scenario (Basic flow):**

1. The use case starts when the cashier request for payment.
2. If the patient has no letter of payment free.[Alt 2.1]
3. The cashier request for pay.
4. The cashier registers a stamp on referral form.
5. The use case ends

**Alternative scenario:**

- 2.1 The use case checks the letter for free of payment.
- 2.2 The use case goes to 4.

**Use case UC4:** Register patient

**Scope:** Referral at hospital

**Include:** Login

**Primary actor:** Patient

**Pre-condition:** Patient with approved, routed referral form and settled payment.

**Successes guarantee (post-condition):** The patient is registered.

**Main Success scenario (Basic flow):**

1. The use case starts when the receptionist opens the referral main menu page on a computer.
2. The system displays registration form.
3. The receptionist input patient data from referral paper.
4. The system adds the data to the database.
5. The use case ends

**Use case UC5:** Register feedback

**Scope:** Referral at hospital

**Primary actor:** Patient

**Include:** login

**Pre-condition:** The referral form reaches to physician

**Successes guarantee (post-condition):** The patient gets treatment result.

**Main Success scenario (Basic flow):**

1. The use case starts when the physician opens the referral main page.
2. The system displays information for referred patient data (referral form).
3. The physician checks referred patient data from referral form.
4. The physician input feedback (treatment result) information on referral form.
5. The system adds the data to the database.
6. The use case ends.

**Use case UC6: Search patient**

**Scope:** Referral at Hospital and Health center

**Include:** login

**Primary actor:** Patient

**Pre-condition:** Referred patient get treatment previously.

**Successes guarantee (post-condition):** The referred patient data is accessed successfully.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays the referral main page.
2. The receptionist select search for patient.
3. The system display search form with search index.
4. The receptionist input patient name and mobile number.
5. The system displays searched patient data.
6. The use case ends

### **Use case UC7: Login**

**Scope:** Hospital and health center

**Primary actor:** Receptionist (hospital) and physician

**Pre-condition:** Receptionist needs to register patient and physician need to refer and access patient data.

**Successes guarantee (post condition):** The user is successfully logged in.

#### **Main Success scenario (Basic flow):**

1. The use case starts when the receptionist opens the referral main page on a computer.
2. The system displays login form and requests account information.
3. The receptionist input user name and password.
4. If account information correct.[Alt 4.1]
5. The system display main menu window.

#### **Alternative Scenario:**

- 4.1 The system displays invalid user name screen.
- 4.2 The use case goes to step 2.

### **4.2.2 Non-Functional Requirements**

Nonfunctional also called technical requirements that describe the quality aspect of the system.

This study identifies the following technical requirements in developing WRIMSAA.

- The system should be available to all appropriate individuals along with their corresponding privileges.
- The system should be able to recovers from any kind of failure like power supply.
- The system is expected to have less response time.
- The system should use password and user name for authentication to enter the system and security password for the different functionalities of the system.
- The system should be implemented on a high speed computer.

### 4.2.3 Business Rule Identification

**BR1:** *Validate Referral Information*

**Description:** validates whether the input fields for referral forms are filled correctly and important

fields are not missed and input fields should be filled by appropriate input types.

**BR2:** *Check Catchment Area*

**Description:** Health centers have to check catchment area before referral was done.

**BR3:** *Redirect to Alternative Facility*

**Description:** The health centers have to check the capability of the hospitals before allocating patients.

**Related rule:** Hospitals can redirect patient to another facilities if it is beyond their capability.

**BR4:** *Settle Financial Issue*

**Description:** The patient must solve all financial related issues before getting referral service at hospitals.

**BR5:** *Alert for new information*

**Description:** Health centers have to alert hospitals for emergency case through e-mail and Hospitals have to access all referrals once a day.

**Related Rule:** Hospitals should alert health centers if the need for further information

### 4.3 Object-Oriented Analysis (OOA)

In this study, OOA phases emphasis on finding and describing the objects or concepts in the referral system. An object is a software construct that mirrors a concept in the real world. It could be a user's of the referral system, screen where the referral users used, or reports generated from referral service providers. Mainly it is concerned with creating a description of the domain from the perspective of domain objects. There is an identification of concepts, attributes, and associations that are considered noteworthy. The core tasks in OOA phase in developing WRIMSAA was defining the problem to be solved in the existing referral system and effective understanding of what to build. So the study focus on understanding the requirements document to produce a high-level specification that describes *what* the system is supposed to do. <sup>26</sup>

The OOA phase can be depicted using UI flow diagramming, System use case diagrams, Conceptual and Data modeling that give a picture of the significant domain concepts or referral objects for developing WRIMSAA.

#### **4.3.1 System use case modeling**

System use case modeling describes the system by taking into consideration hardware and software requirement that brings technological concerns into account. Sometimes known as concrete use case models or detailed use case models, model your analysis and design your behavioral requirements, describing in detail how referral users will work with your system, including references to its UI aspects. <sup>29</sup>

Use cases identified in system use case modeling were *login, refer patient, register referral result, edit referral, view referral, allocate patient* and *add user*.

As system use case includes UI concern, the following UI were identified *home screen, login screen, main menu screen, referral list screen, feedback list screen, patient allocate list Screen, add referral form, edit referral form, show referral detail screen, add feedback form, add user form* and *user list screen*.

##### **A. Use case Diagram for system use case modeling**

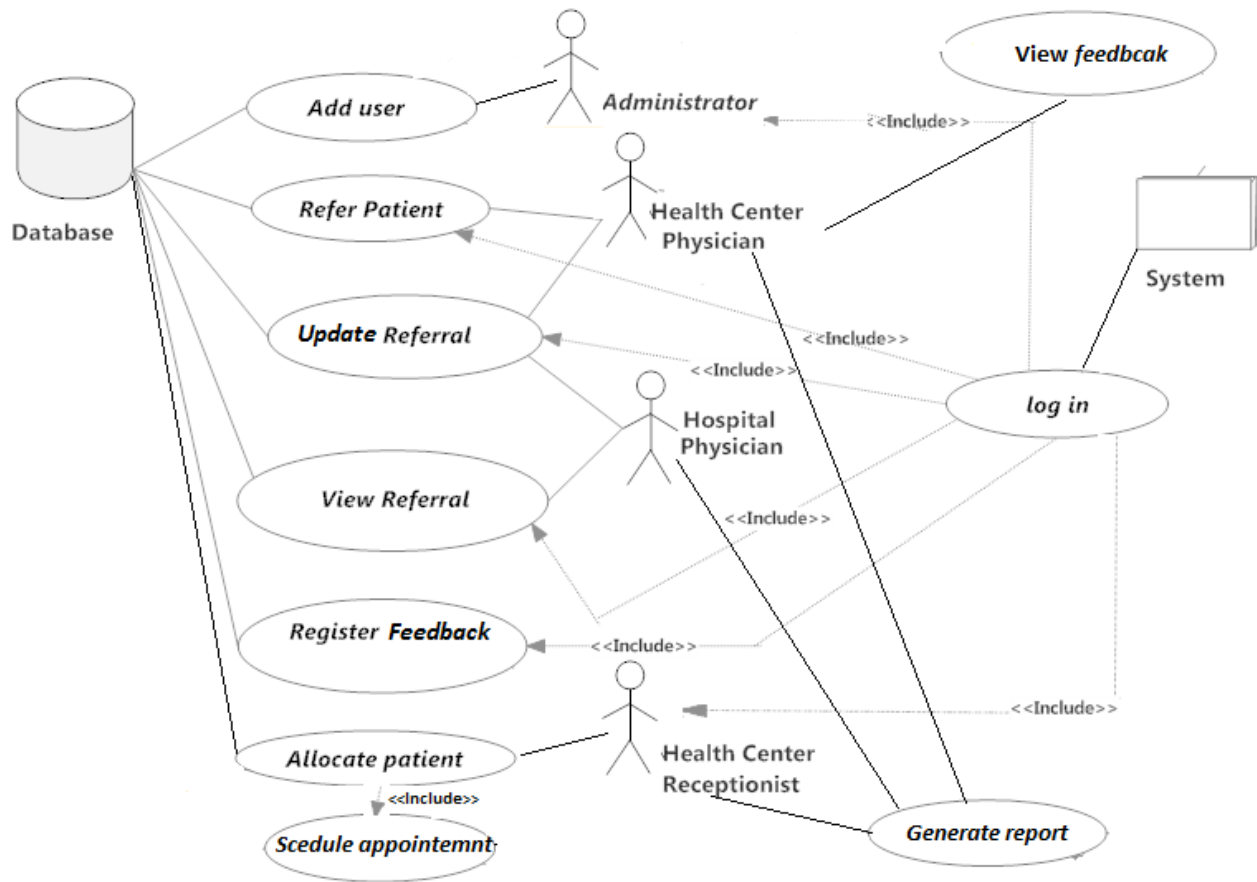


Figure 4.3: System use case diagram for WRIMSAA

## B. Use case description for System Use case modeling

**Use case UC1:** Login

**Scope:** Hospitals and health centers

**Primary actor:** User (physician and referral system administrator)

**Pre-condition:** The users need to refer a patient.

**Successes guarantee (post condition):** The users successfully logged in.

**Main Success scenario (Basic flow):**

1. The use case starts when the user opens a referral information system
2. The system displays home screen.
3. The user select proceed to system link on home screen.
4. The system displays login screen.
5. The user type username, password and select user type on login screen

6. The system verifies account information.
7. If the user account information is correct.[Alt 7.1]
8. The system displays main menu screen.
9. The use case ends.

**Alternative scenario:**

- 7.1 The system displays invalid login message.
- 7.2 The use case goes to step 4.

**Use case UC2: Refer patient**

**Include:** Login

**Scope:** Health Center

**Primary actor:** Physician

**Pre-condition:** Physician determines referral indication and login.

**Successes guarantee (post-condition):** The patient is referred successfully.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays UI3 main menu screen.
2. The user selects referral link on UI3 main menu screen.
3. The system displays UI4 referral list screen.
4. The user selects add/new referral link on UI4 referral list screen
5. The system displays the UI8 add/new referral form
6. The user enters referral information on UI5 referral form.
7. The system adds referral data on hospital database.
8. The use case ends.

**Use case UC3: Register feedback**

**Include:** Login

**Scope:** Hospital

**Primary actor:** Physician

**Pre-condition:** Patient referral sent to physician.

**Successes guarantee (post-condition):** The patient data is retrieved successfully and treatment result is registered.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects feedback link on main menu screen.
3. The system displays feedback list screen.
4. The user selects add feedback link on referral list screen
5. The system displays feedback form.
6. The user enters treatment result on feedback form.
7. The system adds feedback data to the hospital database.
8. The use case ends.

#### **Use case UC4: Update referral**

**Include:** Login

**Scope:** Health centers

**Primary actor:** Physician

**Pre-condition:** A need to change patient referral.

**Successes guarantee (post condition):** The patient data is updated accordingly.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects referral link on main menu screen.
3. The system displays referral list screen.
4. The user selects edit referral form link on referral list screen.
- 5 The system displays edit referral form.
6. The user enter data to the fields need to be edited on edit form.
7. The system adds the updated referral data to the hospital database.
8. The use case ends.

#### **Use case UC5: Show referral**

**Include:** login

**Scope:** Hospital and health center

**Primary actor:** Physician

**Pre-condition:** The patient data is available on hospital database.

**Successes guarantee (post-condition):** The patient data is viewed successfully.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects referral link on main menu screen.
3. The system displays referral list screen.
4. The user selects show referral link on referral list screen.
- 5 The system displays referral data on referral detail screen.
6. The use case ends.

### **Use case UC6: Allocate patient**

**Include:** login

**Include:** Schedule appointment

**Scope:** Health center

**Primary actor:** Physician

**Pre-condition:** New referral received.

**Success guarantee (post-condition):** Patient is allocated accordingly.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects patient allocate link on main menu screen.
3. The system displays patient allocate list screen.
4. The user selects add/new patient link on referral list screen.
- 5 The system displays add/new patient form.
6. The user enters patient data to add/new patient form.
7. The system adds the patient allocation data to the hospital database.
8. The use case ends.

### **Use case UC7: Add User**

**Include:** login

**Scope:** Hospital and health center

**Primary Actor:** Administrator

**Pre-condition:** New users identified

**Success guarantee (post-condition):** Users are added on successfully.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.

2. The user selects create/add user link on main menu screen.
3. The system displays user list screen.
4. The user selects add/new user link on user list screen.
- 5 The system displays add/new user form.
6. The user enters user data to add/new patient form.
7. The system adds user data to the hospital database.
8. The use case ends.

### **Use case UC8: Schedule appointment**

**Scope:** Health center

**Primary Actor:** Health Center Physician

**Pre-condition:** New referrals received and it is not emergency case

**Success guarantee (post-condition):** Patients appointment is scheduled successfully.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects referral link on main menu screen.
3. The system displays referral list screen.
4. The user selects schedule appointment link on referral list screen.
- 5 The system displays schedule appointment form.
6. The user enters user data and schedule on appointment form.
7. The system adds scheduled data to the hospital database.
8. The use case ends.

### **Use case UC9: Show feedback**

**Include:** login

**Scope:** Health center

**Primary Actor:** Health Center Physician

**Pre-condition:** Patient examined by hospitals and his/her feedback registered accordingly.

**Success guarantee (post-condition):** Patient feedback is accessed successfully.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects feedback link on main menu screen.

3. The system displays feedback list screen.
4. The user selects show feedback link on user list screen.
- 5 The system displays feedback details.
6. The use case ends.

**Use case UC10: Generate report**

**Include:** login

**Scope:** Hospital and health center

**Primary Actor:** Physicians and Health center receptionist

**Pre-condition:** referral tasks performed.

**Success guarantee (post-condition):** report produced and ready to be accessed.

**Main Success scenario (Basic flow):**

1. The use case starts when the system displays main menu screen.
2. The user selects referral link or patient allocate link on main menu screen.
3. The system displays referral list and patient allocate list screen.
4. The use case ends.

### **4.3.2 Conceptual/ Domain Modeling**

A domain model, also called conceptual models, domain object models, and analysis object models, is the most important and classic model in OOA. It illustrates important concepts in a real world domain of the referral system. It is a visual representation or perspective of conceptual classes or real-situation objects in a referral domain, not a software perspective and illustrated with a set of class diagrams in which no operation, method signature are defined. <sup>26</sup>

The importance of domain model in developing WRIMSAA is that it helps the study to use real world domain in creating software class and fulfills the representational gap between how stakeholders in referral system conceive the domain and its representation in software.

Domain model provides a conceptual perspective in referral systems and it may show

- Domain objects or conceptual classes
- Associations between conceptual classes
- Attributes of conceptual classes

### ***Conceptual class***

Identifying a rich set of conceptual classes is at the heart of OOA in domain modeling. There are 3 ways to find conceptual classes by reusing existing models, using a category list and through identifying noun phrases. Among the 3 mentioned above this study employed finding conceptual classes using noun phrases, to identify concepts in the domain for developing WRIMSAA.

### ***Association***

In domain modeling association is a relationship between classes that indicate some meaningful and interesting conceptual perspective in real domain. Association is defined with multiplicity that describes a numerical relationship between instances of class. For this study the association between classes is defined accordingly as shown below in fig. 4.4.

### ***Attribute***

Attribute is a logical data value of an object. It satisfies the information requirement of the current scenario under development. In developing WRIMSAA the attributes is identified and shown in table 4.1 here below.

*Table 4.1: Conceptual/domain modeling concepts, attribute and methods.*

<b><i>Concept</i></b>	<b><i>Attribute</i></b>	<b><i>Method</i></b>	<b><i>Description</i></b>
Physician	User name Password First name Last name Department email address	Login Add Feedback Edit Feedback Show feedback Show Referral Add Referral Edit Referral	The physician handles feedback and referral data. To perform this S/He has to login.
Users	Id first name last name since(refer date) user type user name Password	Login Allocate patient Add user	The receptionist at health handles patient allocation. The users have to login before using the system. System administrator with privilege to add user to the system
Feedback	Card number	Set	Feedback form

	Serial number date Referred Hospital Name First Name Last Name diagnosis Treatment Given finding Physicians Name	get	contains treatment result and processed by physician.
Referral	Card Number Serial Number date Department Referred Hospital Name Referring Health Center Name Clinical Finding diagnosis Investigation Result Treatment Given First Name Last Name dob sex Kebele House Number Occupation Sub City	Set get	Referral form contains referred patient data which is processed by physician at health center and hospitals
Patient Allocate	Card Number cases Referring Health Center Name Referred Hospital Name First Name Last Name Date	Set Get	Patient placement handled by health centers receptionist.
System(user manager)	User Type User Name Password	user	The system verifies user account Information.
Invalid User Exception		Display	A message displayed for invalid user
Referral Database Connection	Connection statement query result	isConnected write read close Set	A referral database that lets the user to insert, read data from database.

Using the identified concepts and attributes in table 1 above, the researcher develops the conceptual model for developing WRIMSAA as follows in fig. 4.4.

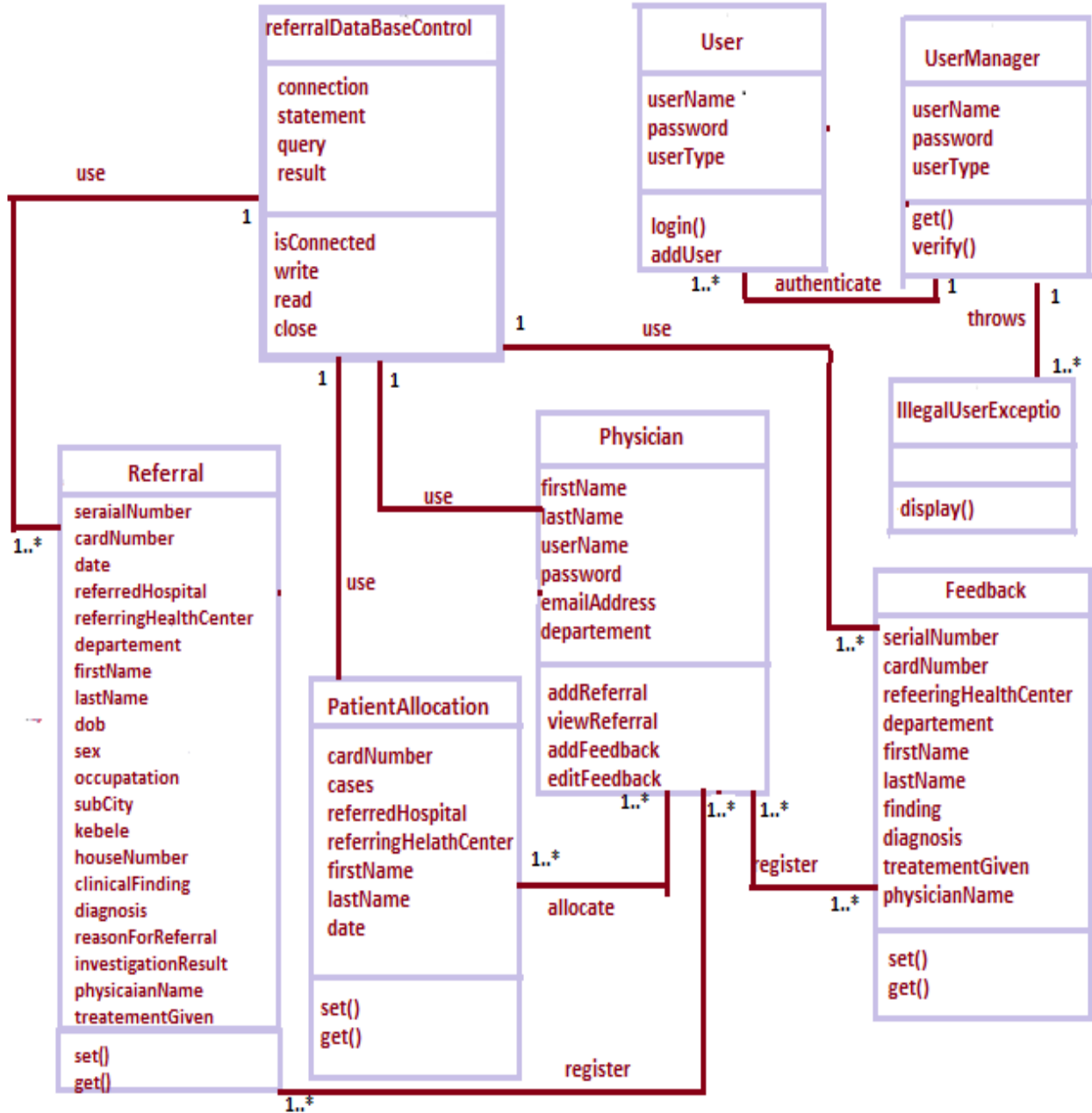


Figure 4.4: Domain/Conceptual model for WRIMSAA

### **4.3.3 Data modeling**

Data model is an integrated collection of concepts for describing and manipulating referral data, relationship between data and constraint on the data in a referral system. Database development is one component of the Web-based system, development of a data model for persistent data was important part to be included. Data modeling helps to identify the referral data objects required by WRIMSAA. This study employed data modeling through identifying the referral data which is used in providing service for referred patient from referral forms, feedback forms and other documents used by referral facilities. Data for patient allocation is captured from catchment areas documents and data attributes for users and physicians is specified by the researcher based on the important attributes that has to be included. In addition, additional contents in referral and feedback forms that is believed contribute to the new system were also included. Among different types of data modeling this study employed an object-oriented data model which is one of the common types of object-based data model in OO system.<sup>31</sup>

The data model developed provides a higher-level description of database structure which represents the basic concept and notation of referral data that will allow end-user in referral centers unambiguously and accurately communicate their understanding of the referred patient data.

#### **4.3.3.1 Object-Oriented data model**

Object-oriented data model which is used in this study extends the definition of referral entity to include not only the attributes that describe the state of the object but also the actions that are associated with the object, behavior. There are steps followed by the researcher in developing a data model for WRIMSAA.

- Entities including referral tables, feedback tables, patient allocation tables, users and physicians tables which are the significant objects of interest, referral data where this study need to store information is defined.
- The primary keys which identify uniquely each entity were identified.
- The relationship that exists between entities which is based on the primary key was defined.
- Attributes of the entity which is additional to the defined primary key were stated.
- Data normalization was performed.

Some of the entities attribute and primary key identified for this phase of the study in developing WRIMSAA is shown in table 4.2 below.

*Table 4.2: Data modeling entities, attributes and primary key for WRIMSAA*

<b><i>Entities</i></b>	<b><i>Primary Key (PK)</i></b>	<b><i>Attributes and Methods</i></b>	<b><i>Description</i></b>
Referral	Referral_ID	Shown on table 4.1	Contains referred patient data
Feedback	Feedback_ID	Shown on table 4. 1	Contains treatment result data at referred facilities (Hospitals)
Patient Allocation	Patient_Allocation_ID	Shown on table 4.1	Store patient data allocated in different hospitals
users(Referral system users)	ID	Shown on table 4.1	Store data for users who operate the system(Administrator, physician and users(Health receptionist)
Physician	Physician_ID	Shown on table 4.1	Store data for physician which helps health care staffs s to identify specialist.

Based on the entities identified in table 4.2 above the researcher derives the following data model diagram for WRIMSAA.

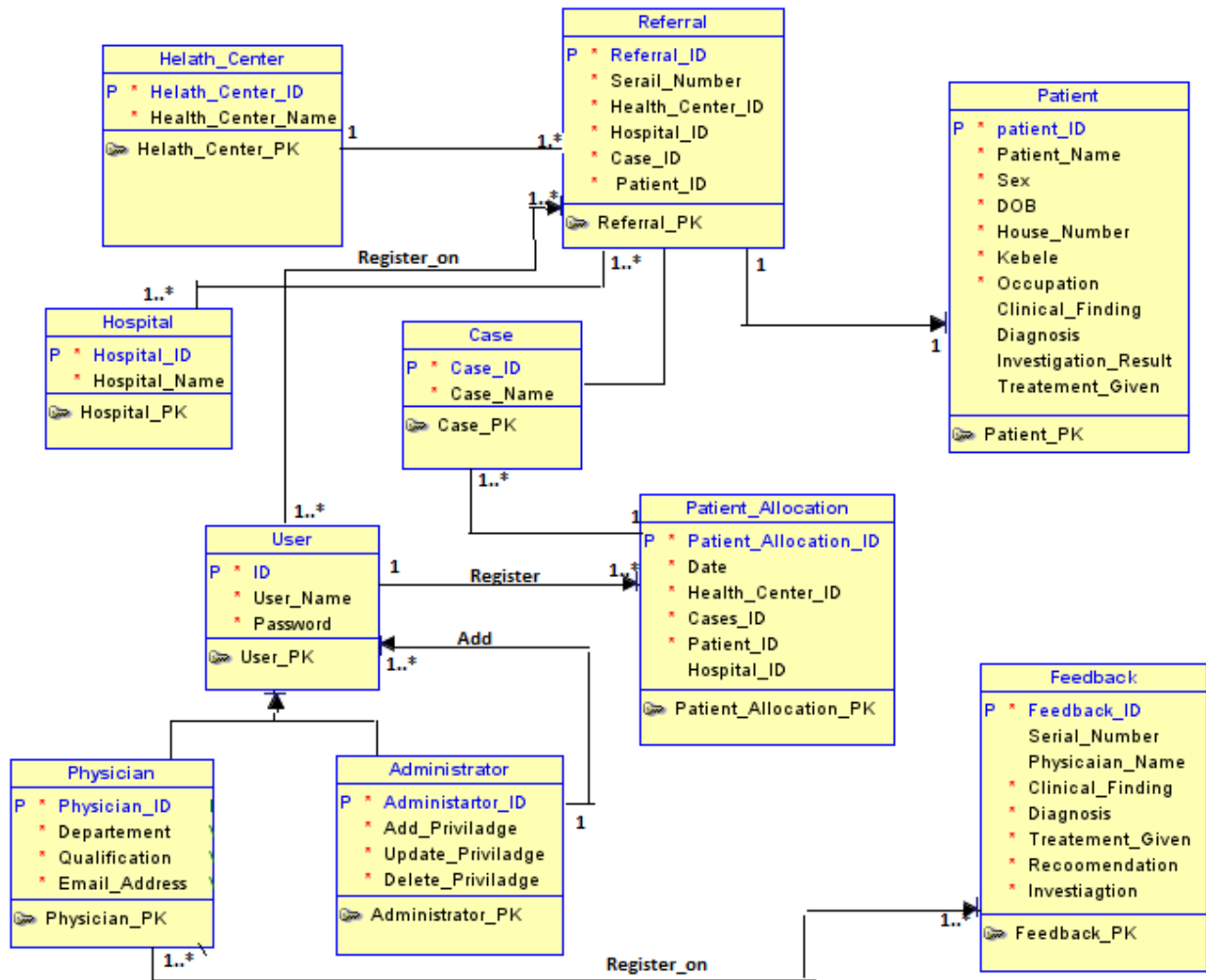


Figure 4.5: Data model for WRIMSAA

#### 4. 3.4 User-Interface Flow Diagrams

UI flow diagrams are a diagram that depicts major UI elements and how users transition between them in developed system. It also used to explore the high-level usability of a system and document the user interface of a WRIMSAA. It offers a high-level view of the interface of the system, software with which a user directly interacts. The referral user can quickly gain an understanding of how the system is expected to work using UI flow diagramming and puts the user in a position where the user can validate the overall flow of the application's user interface.<sup>30</sup>

This study employed UI flow diagrams which contain important forms and screens depicted in fig. 4. 6 that lets referral users to navigate through. The UI developed

- *Firstly*, model the interactions that users have with the software, as defined in a single use case.
- *Secondly*, enable users to gain a high-level overview of the UI for WRIMSAA. This overview is effectively the combination of all the behavioral views derived from your use cases.

## **Usability**

An important issue to be considered in developing an essential UI for Web-based system is usability. As a result this study considers all usability issues to make it user-centric and applicable to the real system. In developing WRIMSAA, the study uses an easy and structured navigation, instructions to use the system and well-organized contents. For instance the developed system is accessible by any one in referral unit of service without detailed experience of users. The developed system is easy to use and operate which further provides a mechanism for users in advancing their knowledge, and more importantly it represent the real environment for processing referral tasks which achieves usability of the system. There are five factors affecting the usability of software and also taking into consideration in developing WRIMSAA.

1. **Access.** The system being developed, WRIMSAA is a Web-based system which runs on internet that makes easily accessible system by users in referral unit of service.
2. **Efficacy.** The system did not interfere with or impede use by a skilled user who has substantial experience with the system.
3. **Progression.** The system developed facilitates continuous advancement in knowledge, skill, and facility, and accommodate progressive change in use as the user gains experience with the system.
4. **Support.** The system is designed and developed to support the real work of referral users. This is achieved through requirement gathering of the existing referral system which is used as a base for designing the new system.
5. **Context.** The developed system suited to the real conditions and actual environment of the operational context in which it will be used and deployed.

Figure 4.6 below shows the high-level overview architectural approach for system UI, the approach that enables to understand the complete UI flow diagramming for WRIMSAA.

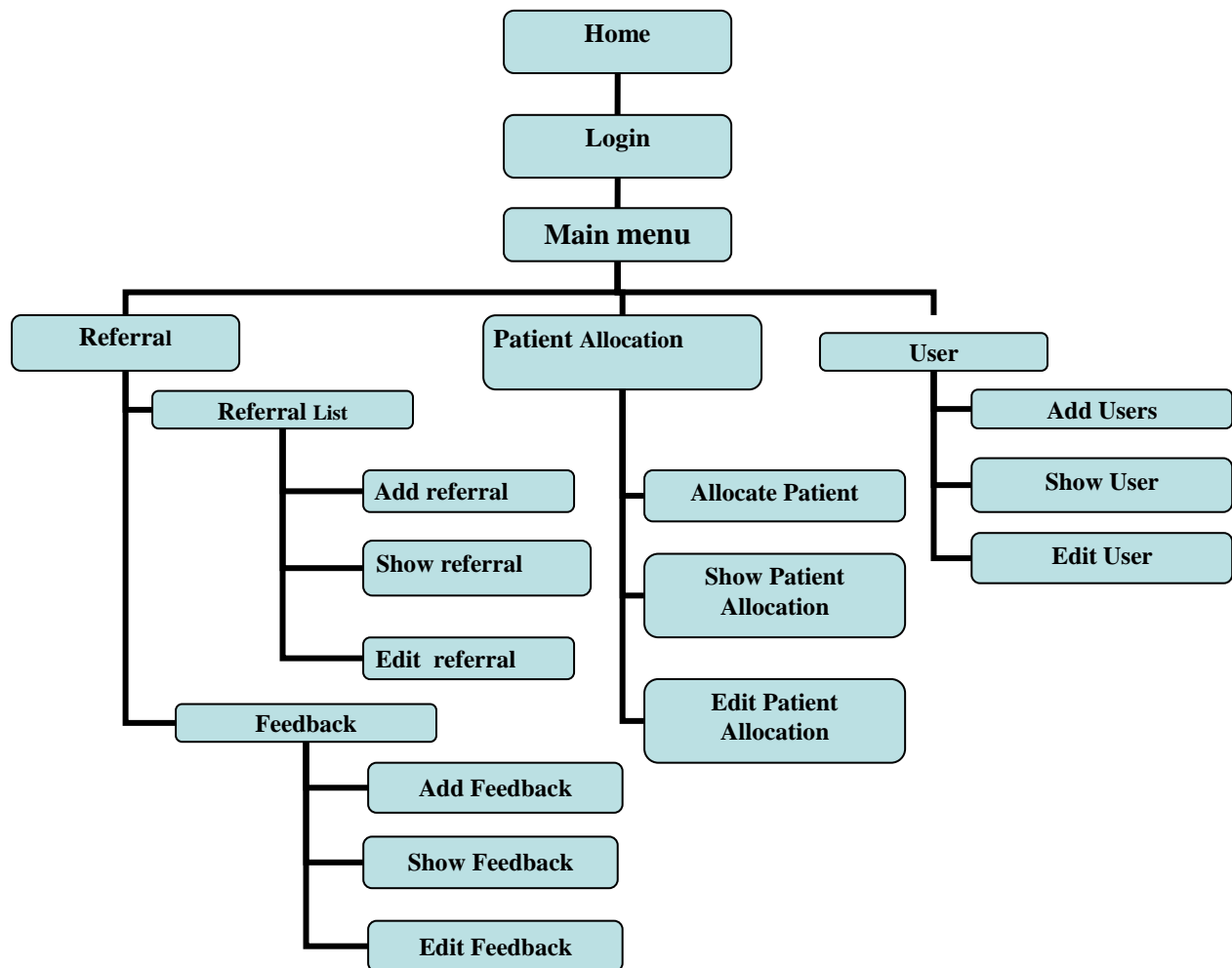


Figure 4.6: User Interface flow diagram for WRIMSAA

#### 4.4 Object-Oriented Design (OOD)

OOD is a phase which concerned with defining software objects their responsibilities and collaborations in referral system to fulfill the requirements independent of implementation details. There are a set of diagrams that are used to describe the design for software objects of referral system in object oriented design phases. This includes software class diagrams, sequence diagram also known as object interaction diagrams, package diagrams, and so forth.

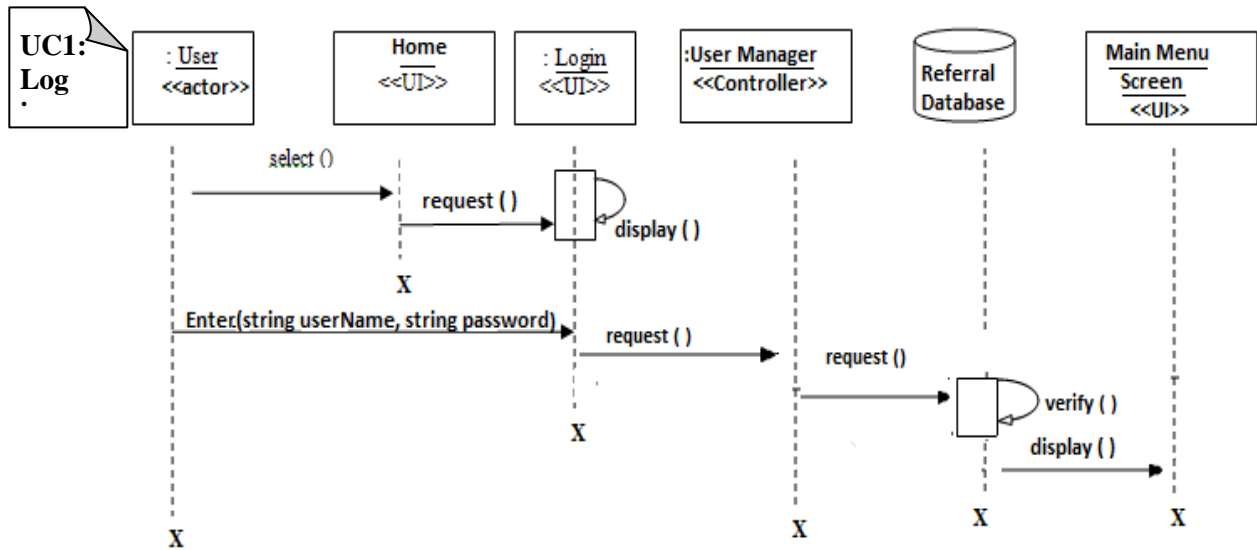
This study used UML artifacts, sequence diagrams, along with class diagrams to design the system being developed, WRIMSAA. Physical data models which is the most important design-

level models for database application development, is employed for referral database management system. Another tasks performed in this phase is designing UI which is based on referral and feedback forms and others as described in UI flow diagramming section. The output of the design stage—a *design specification*—should specify clearly *how* the system should be constructed to satisfy the specified requirements.<sup>25</sup>

#### 4.4.1 Sequence Diagram

UML sequence diagrams model the flow of logic within the referral system being developed in a visual manner, enabling both to document and validate the system logic, and are commonly used for both analysis and design purposes. It is a common notation to illustrate collaborations among referral objects and shows the flow of messages between objects, and thus the invocation of methods.<sup>30</sup>

The following section depicts system sequence diagram for designing WRIMSAA.



*Alternative scenario for login*

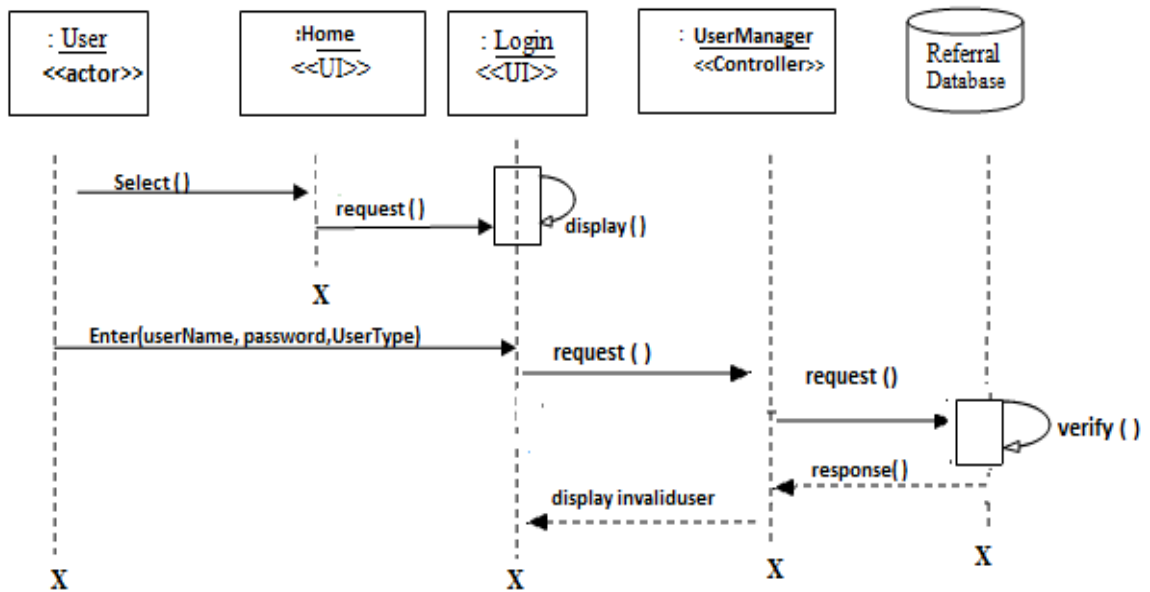


Figure 4.7: Login sequence diagram for WRIMS.

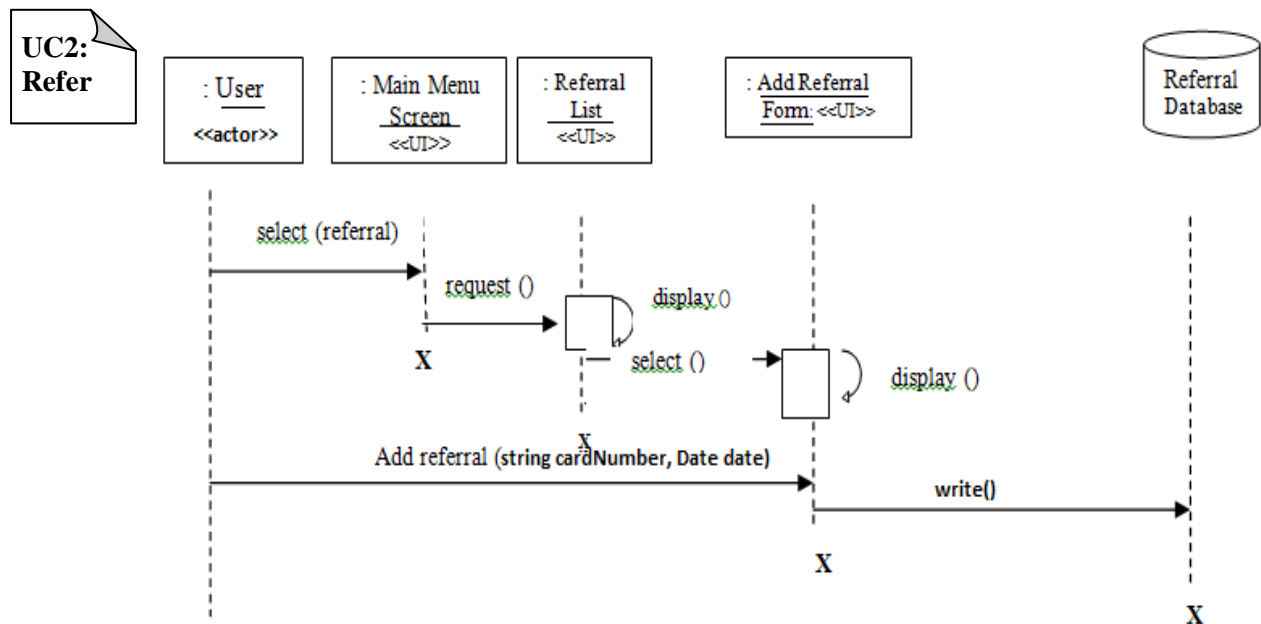


Figure 4.8: Refer patient Sequence diagram for WRIMS

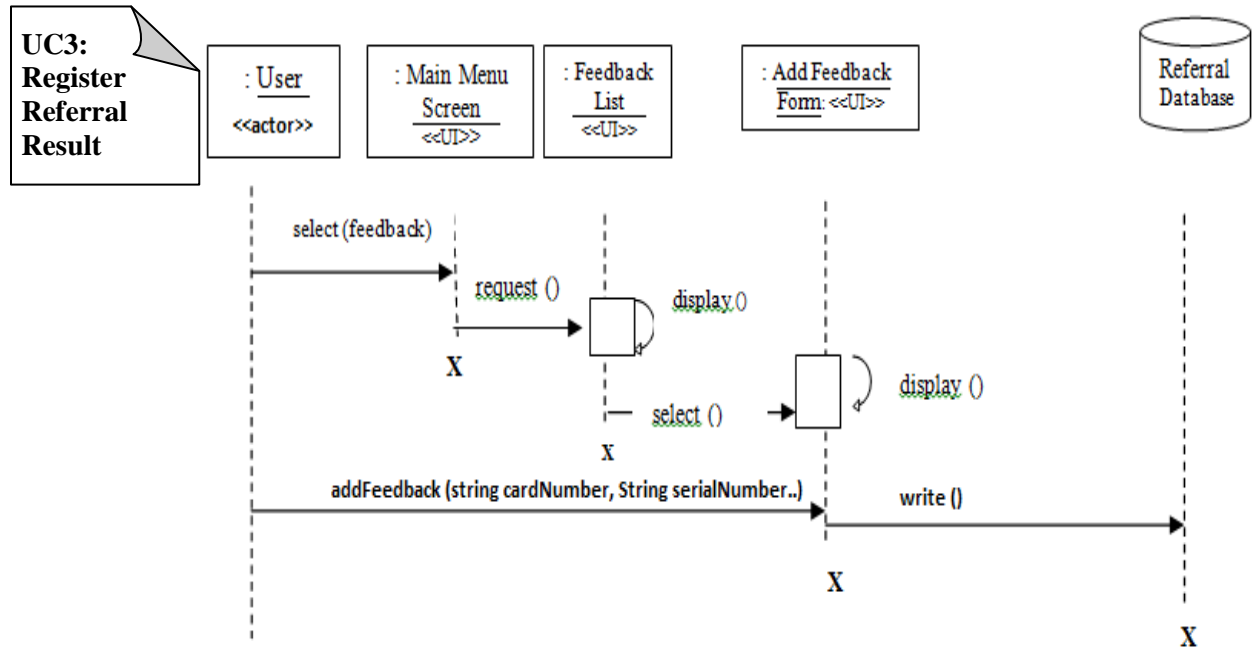


Figure 4.9: Register Referral Result sequence diagram

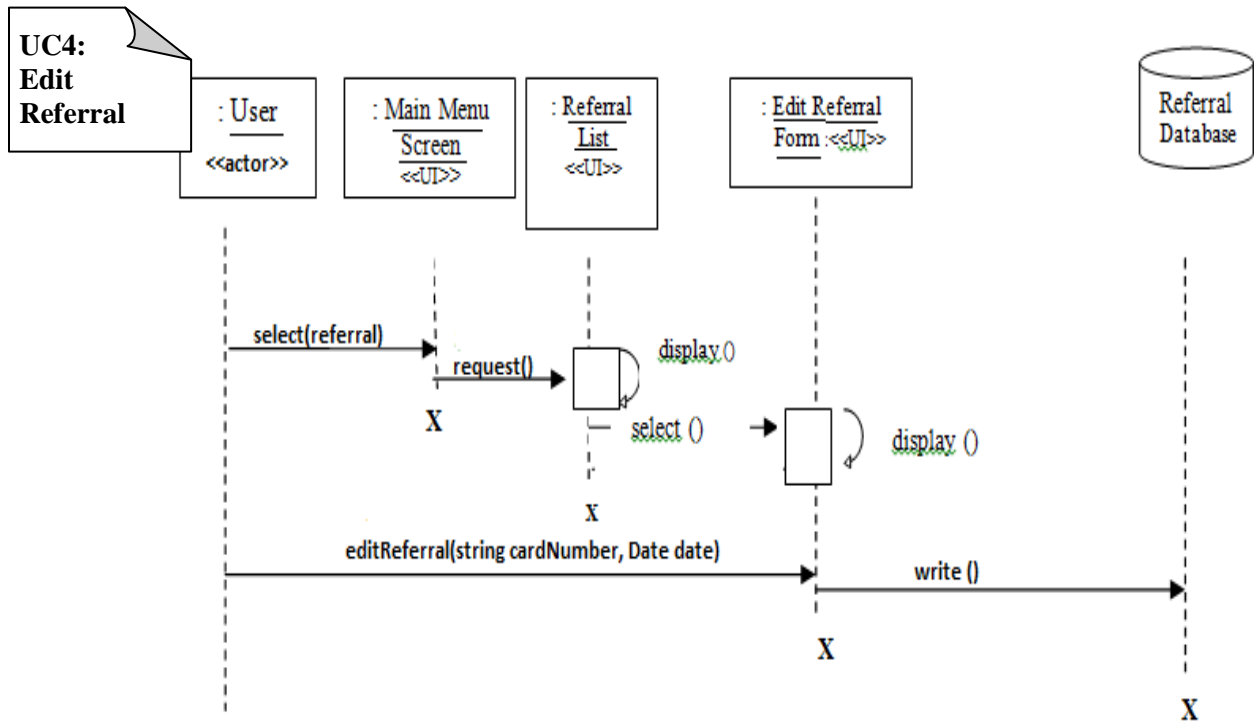


Figure 4.10: Edit Referral form System Sequence Diagram

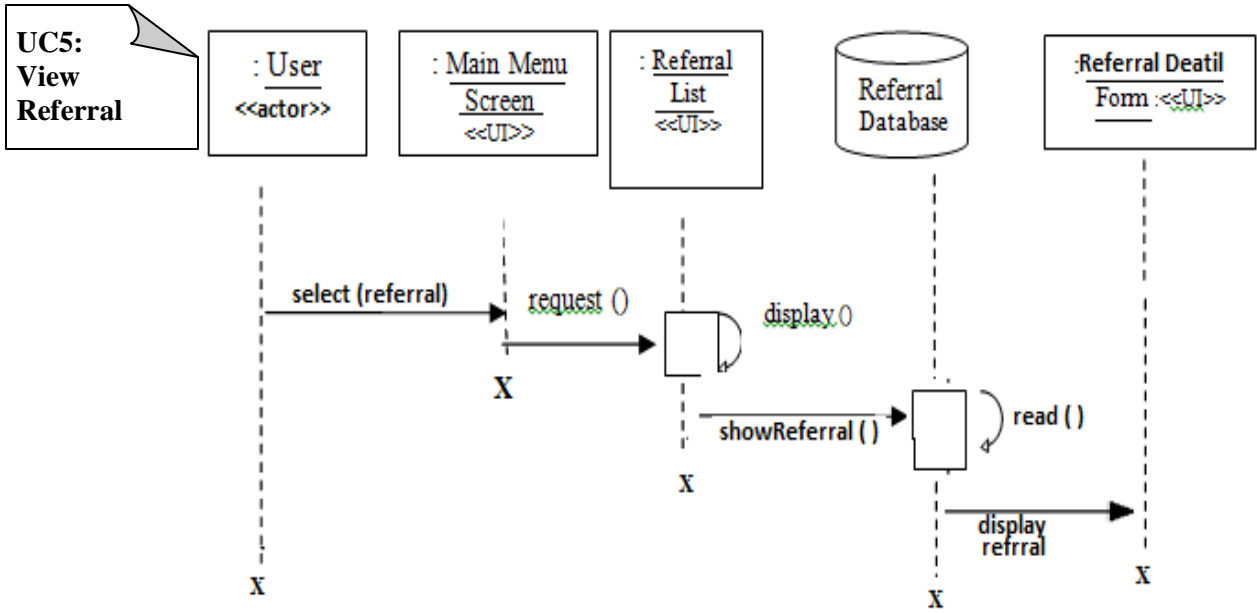


Figure 4.11: View Referral System Sequence Diagram

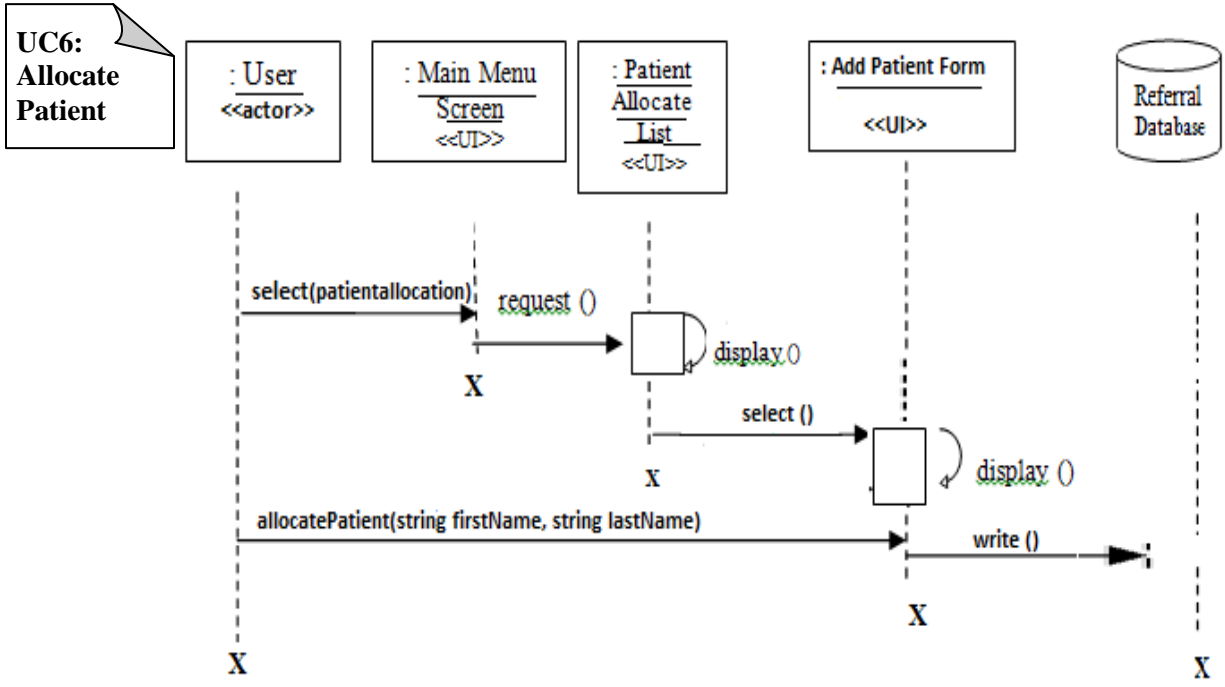


Figure 4.12: Allocate Patient System Sequence Diagram

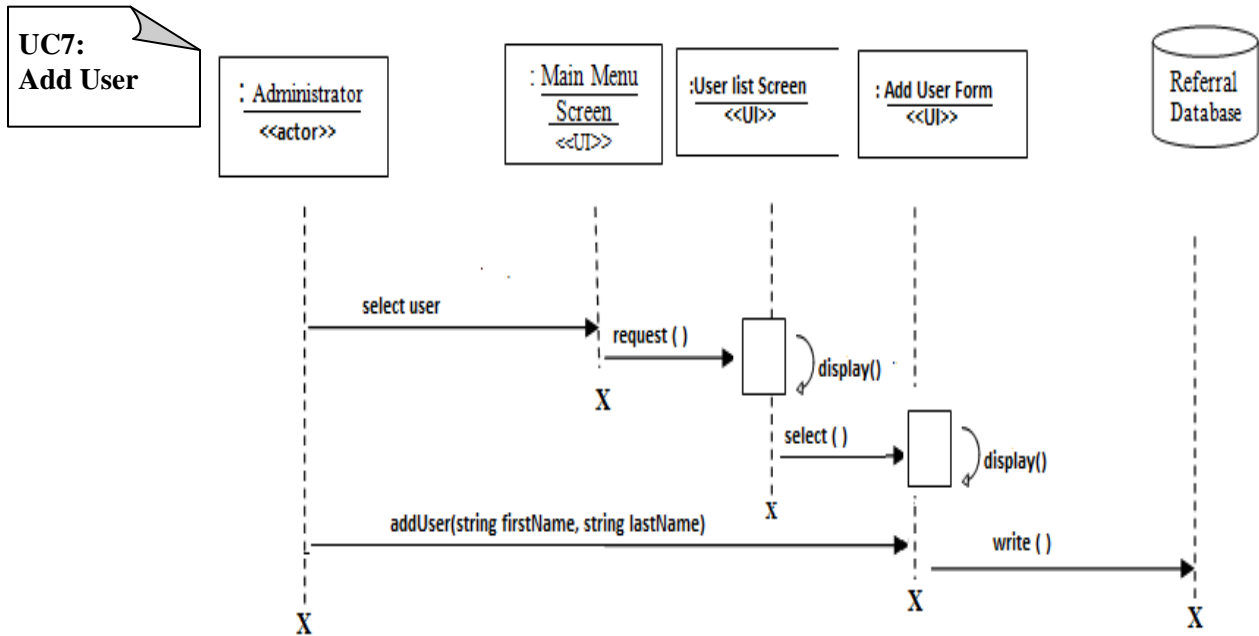


Figure 4.13: Add User System Sequence Diagram

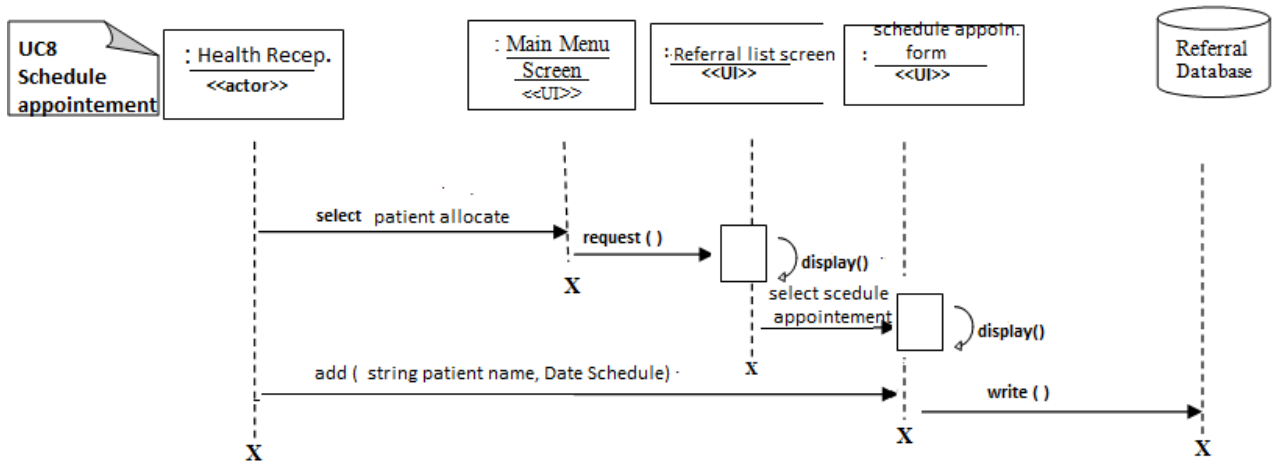


Figure 4.14: Schedule Appointment System Sequence Diagram

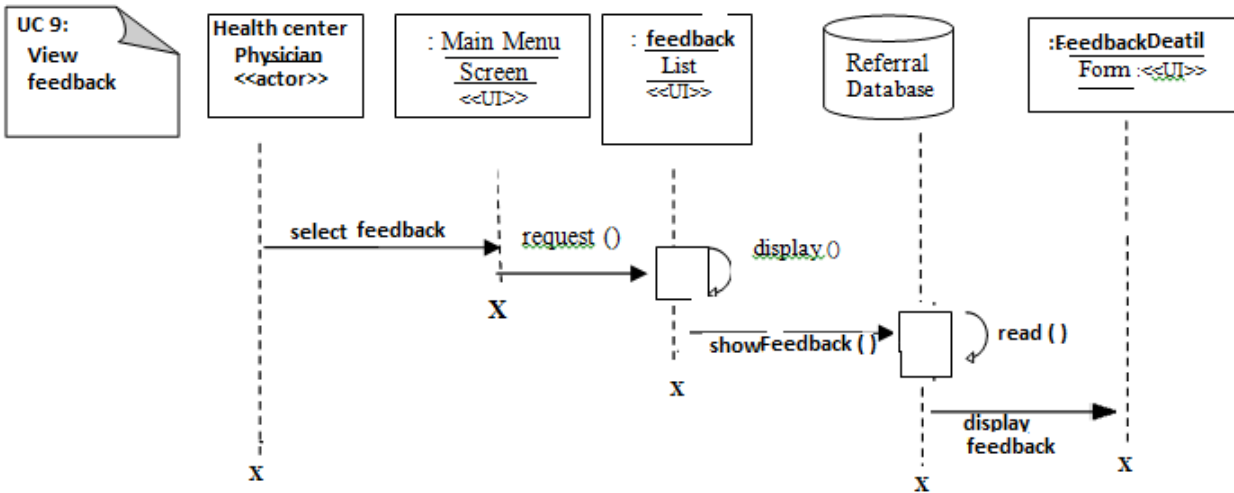


Figure 4.15: View feedback System Sequence Diagram

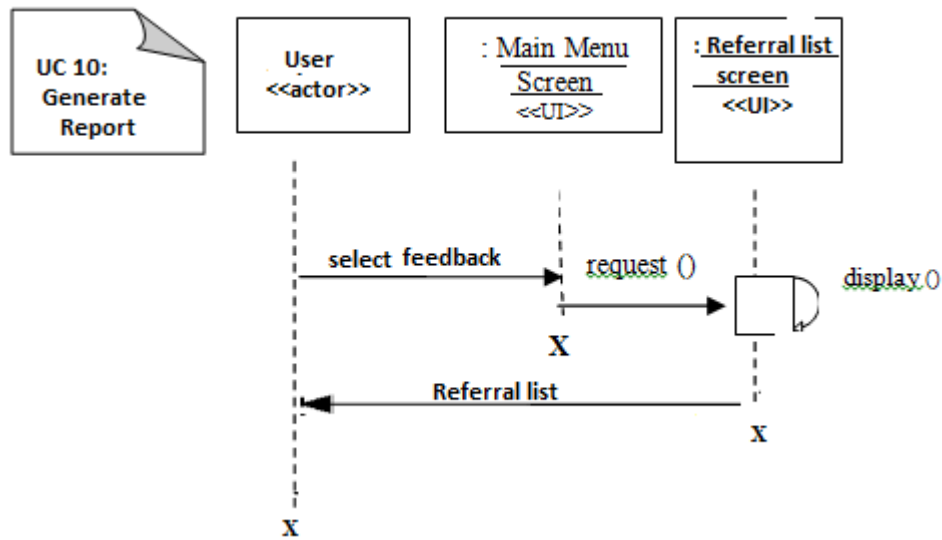


Figure 4.16: Generate Report System Sequence Diagram

The boxes across the top of the diagram represent classifiers or their instances; typically use cases, objects, classes, or actors. Objects respond to messages through the invocation of an operation and classes do so through the invocation of static operations. Objects have labels in the standard UML format *name: Class Name*, where "name" is optional and object labels are underlined as it is shown in the format. The dashed lines hanging from the boxes are called object lifelines, representing the life span of the object during the scenario being modeled. The

long, thin boxes on the lifelines are activation boxes, also called method-invocation boxes, which indicate processing is being performed by the target object/class to fulfill a message. The *X* at the bottom of an activation box, indicate an object has been removed from memory. Messages are indicated on UML sequence diagrams as labeled arrows. Stereotypes <<actor>>, <<controller>>, and <<UI>> is used to indicate an actor, a controller class, or a user-interface (UI) class, respectively.

#### 4.4.2 Class Diagram

In addition to a dynamic view of collaborating objects in WRIMSAA shown in sequence diagrams earlier, a static view of the class definitions is usefully shown with a class diagram. The class diagram shows attributes and methods of the classes identified from referral system. In contrast to the domain model that shows real-world classes, the design class diagram shows software classes. In this way, OO designs and languages can support a lower representational gap between the software components and our mental models of a domain that improves comprehension.<sup>29</sup> In class diagramming a class with its attributes and methods is clearly represented. There is level of access that external objects have to a method, on class diagrams described below.

- **Private (-)**

A private accessor method can only be invoked by other methods in the class in which it is defined, but not in the subclasses. Applicable when the method provides behavior specific to the class.

- **Public (+)**

A public accessor method can be invoked by any other method in any other object or class. Employed when the method must be accessible by objects and classes outside of the class hierarchy in which the method is defined.

- **Protected (#)**

A protected accessor method can be invoked by any method in the class in which it is defined or any subclasses of that class. Applicable when the method provides behavior needed internally within the class hierarchy, but not externally.

Figure 4.17 below shows a design class diagram for developing WRIMSAA.

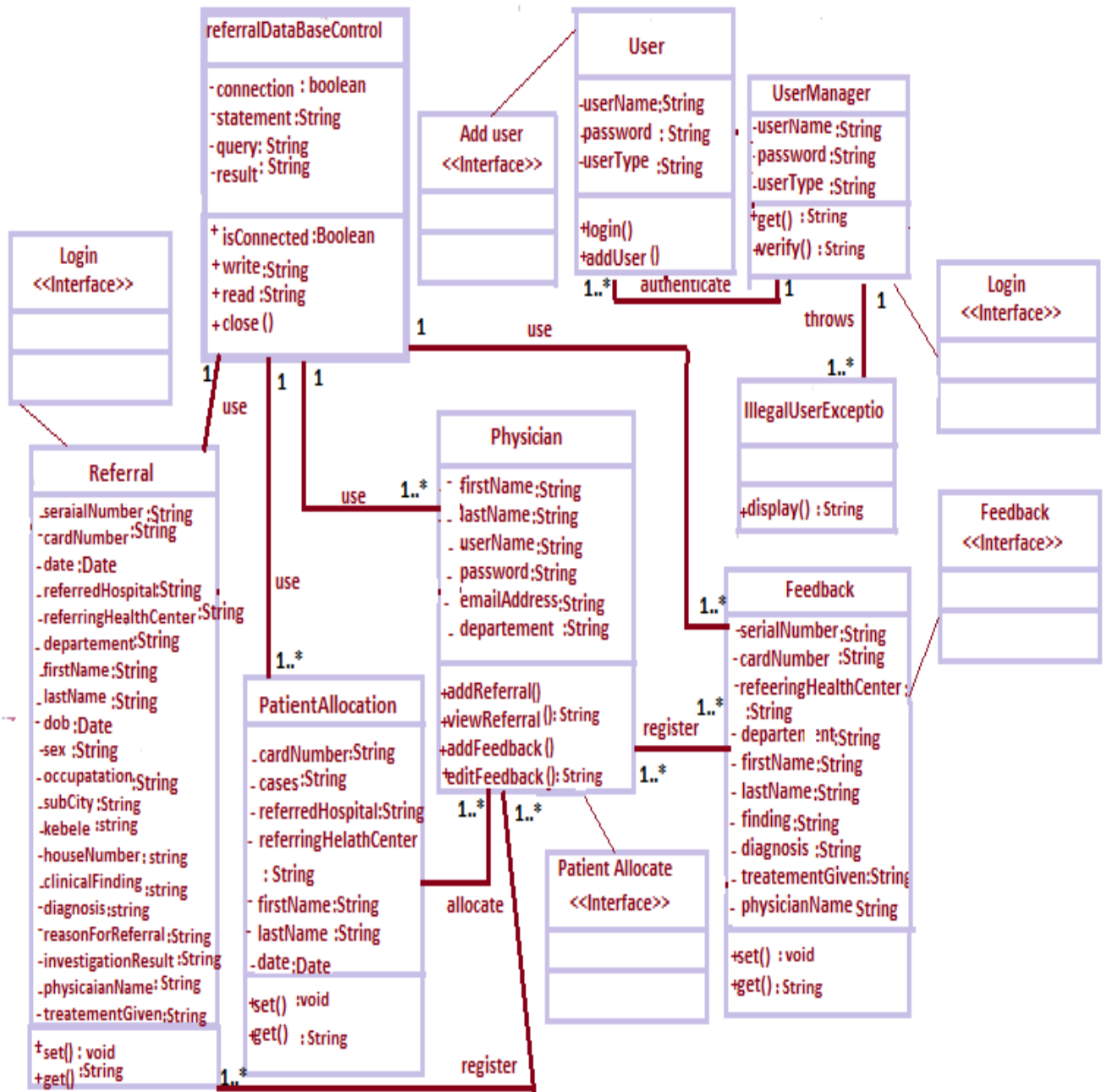


Figure 4.17: Class diagram (design) for WRIMSAA

#### 4.4.3 User Interface Designing/Prototyping

The UI is the portion of software with which a user of referral system directly interacts. It models UI requirements; requirements evolved through analysis and design to result in the final UI for the system. It introduces technology to prototyping efforts that is most likely made a design decision about the implementation technology.<sup>29</sup> The following parts shows UI prototype in developing WRIMSAA.

##### 1. Home

The first page displayed when the user need to access referral Information system. It provides a link to login page.

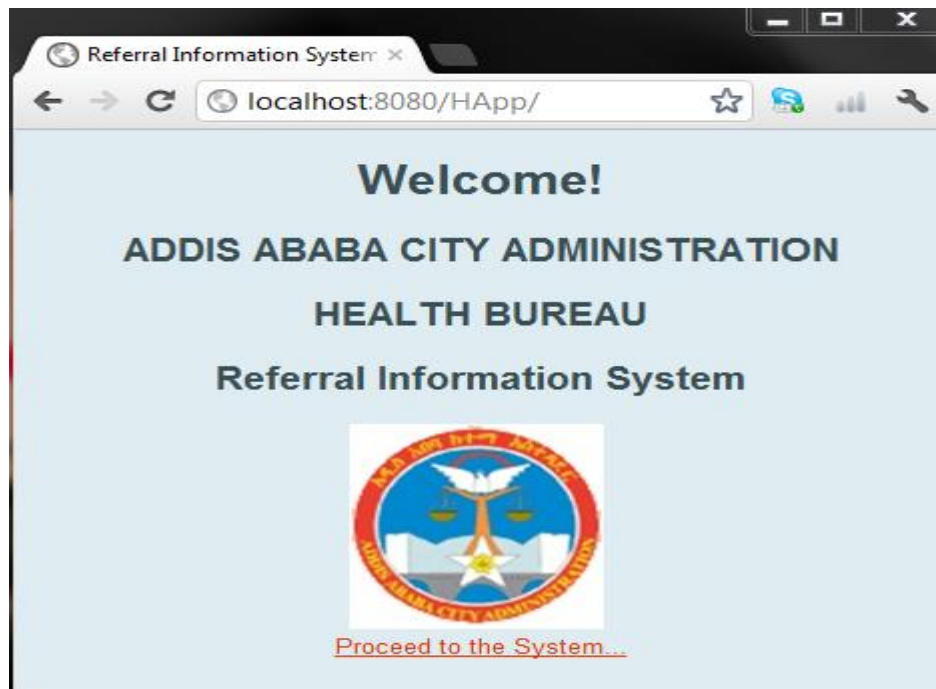


Figure 4.18: Home screen display

## 2. Login Screen

Login screen is the page that displays when the users proceed to the referral Information system. It provides interfaces to accept users' account information which is subject to verification.

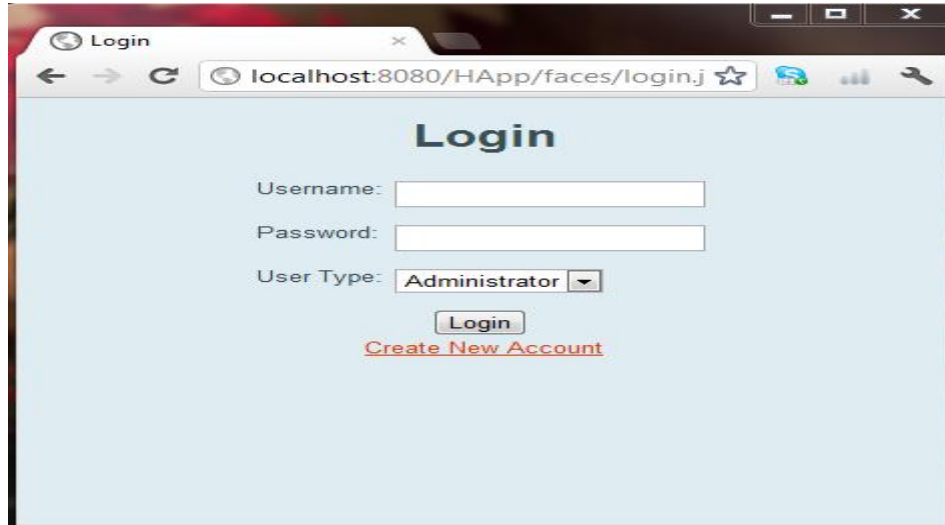


Figure 4.19: login screen display

## 3. Main menu Screen

A screen displayed when the user successfully logged in. It provides a menu to add, view and edit referral, feedback, user and patient allocation data.

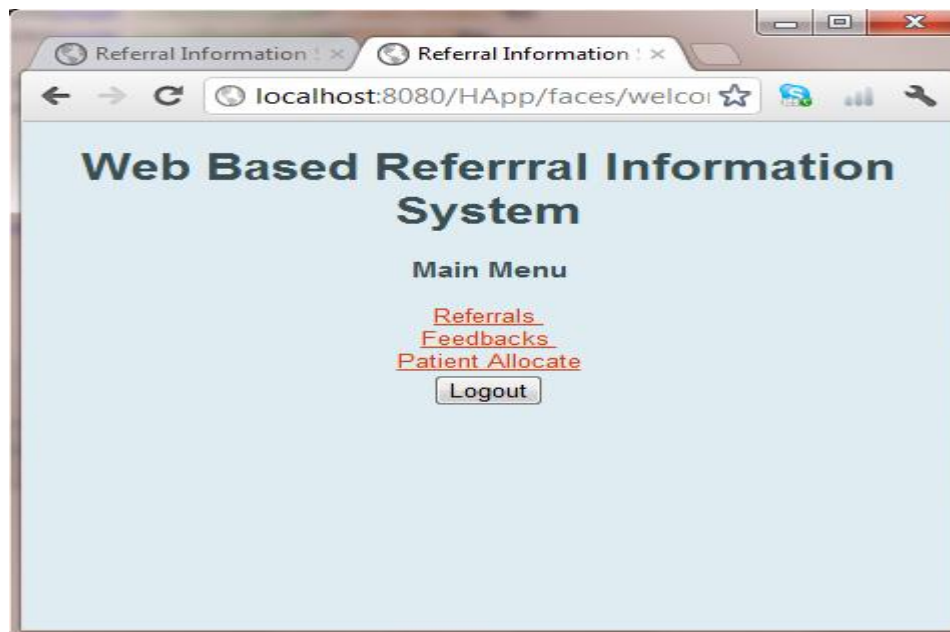


Figure 4.20: Main menu screen display

4. **Referral data list:** A screen displayed when the users need to add, view and edit referral. It also provides access to referral and proceeds to feed back.

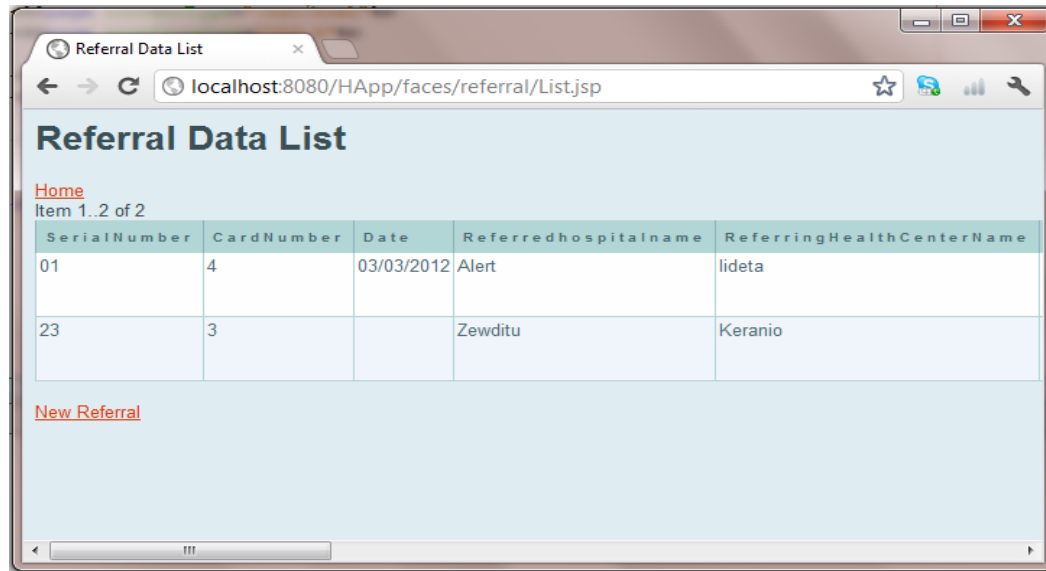


Figure 4.21: Rreferral data list screen display

### 5. Referral Detail Screen

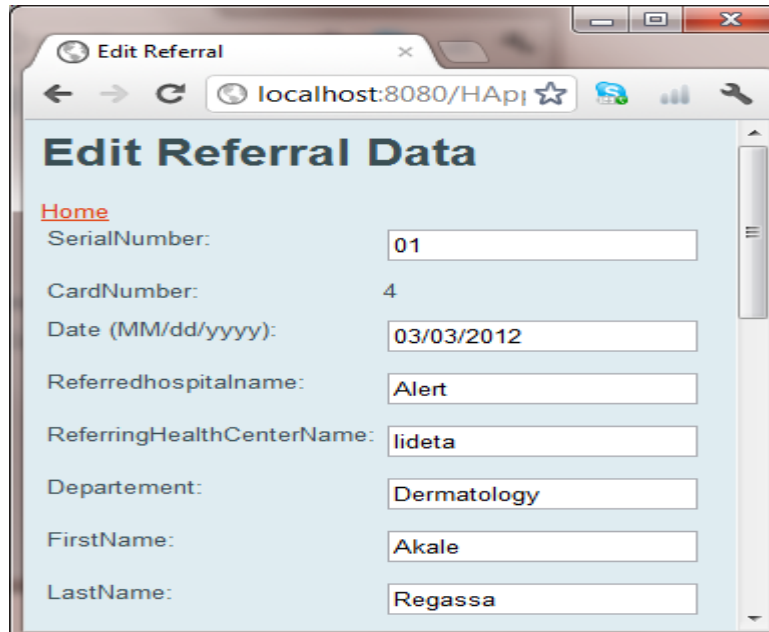
A screen displays when the user need to veiw referral detail.



Figure 4.22 Referral detail screen display

## 6. Edit Referral Screen

A screen displays when the user need to make change to the referral data.

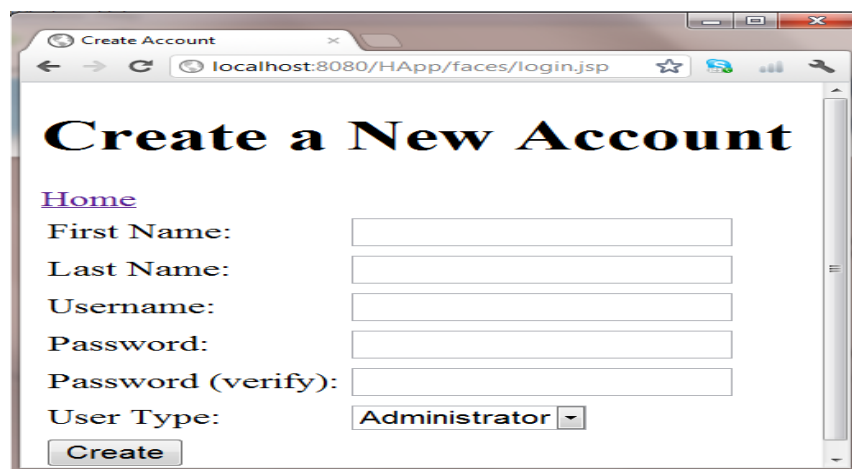


The screenshot shows a web browser window titled "Edit Referral" with the URL "localhost:8080/HApp". The page content is titled "Edit Referral Data" and includes a "Home" link. The form contains the following fields:

SerialNumber:	<input type="text" value="01"/>
CardNumber:	<input type="text" value="4"/>
Date (MM/dd/yyyy):	<input type="text" value="03/03/2012"/>
Referredhospitalname:	<input type="text" value="Alert"/>
ReferringHealthCenterName:	<input type="text" value="lideta"/>
Departement:	<input type="text" value="Dermatology"/>
FirstName:	<input type="text" value="Akale"/>
LastName:	<input type="text" value="Regassa"/>

Figure 4.23: Referral form display

7. Add user form: A form displayed when the system administrator needs to add users.



The screenshot shows a web browser window titled "Create Account" with the URL "localhost:8080/HApp/faces/login.jsp". The page content is titled "Create a New Account" and includes a "Home" link. The form contains the following fields:

First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Username:	<input type="text"/>
Password:	<input type="password"/>
Password (verify):	<input type="password"/>
User Type:	<input type="text" value="Administrator"/>

Figure 4.24: Add user form display

#### 4.4.4 Database Design

Data modeling is the process of producing a detailed data model of a referral database. It focuses on exploring data structures for identified tables from data modeling in analysis stage.<sup>31</sup>

In this phase the study employed physical data modeling which is conceptually similar to design class modeling and design the internal schema of a database including data type with memory requirement specified, depicting the data classes (tables), data attributes of those classes (fields), and the relationships between the classes in developing WRIMSAA. In designing the database the following tasks are performed.

- **Identify Classes:** Classes are the relational database equivalent of tables; data are included under classes. In developing WRIMSAA the researcher identified 6 tables as classes; such as *Referral*, *Patient allocation*, *Feedback*, *Users*, *Physician* and *Referral register*.
- **Normalize object classes (tables):** Data normalization is done to increase the cohesion of tables and to reduce the coupling between tables. The tables identified in referral database are normalized and there is no dependency anomalies observed between object tables.
- **Identify Attributes:** An attribute is the relational database equivalent of columns, and each class identified in developing WRIMSAA have one or more attributes as described in table 4.1.
- **Identify relationships.** Relationships between tables exist as it occurs in classes. In developing WRIMS the study identified and describes association between classes of tables.
- **Assign key:** A key is one or more data attributes that uniquely identify a row in a table. In this study a primary key was identified which is the preferred key for an entity type. In physical data modeling a database key would be formed of one or more attribute whose value(s) uniquely identifies a row within an object table. Primary keys are indicated using the <<PK>> stereotype and foreign keys via <<FK>>.

A physical data model for developing WRIMSAA is depicted as follows in fig. 4.25.

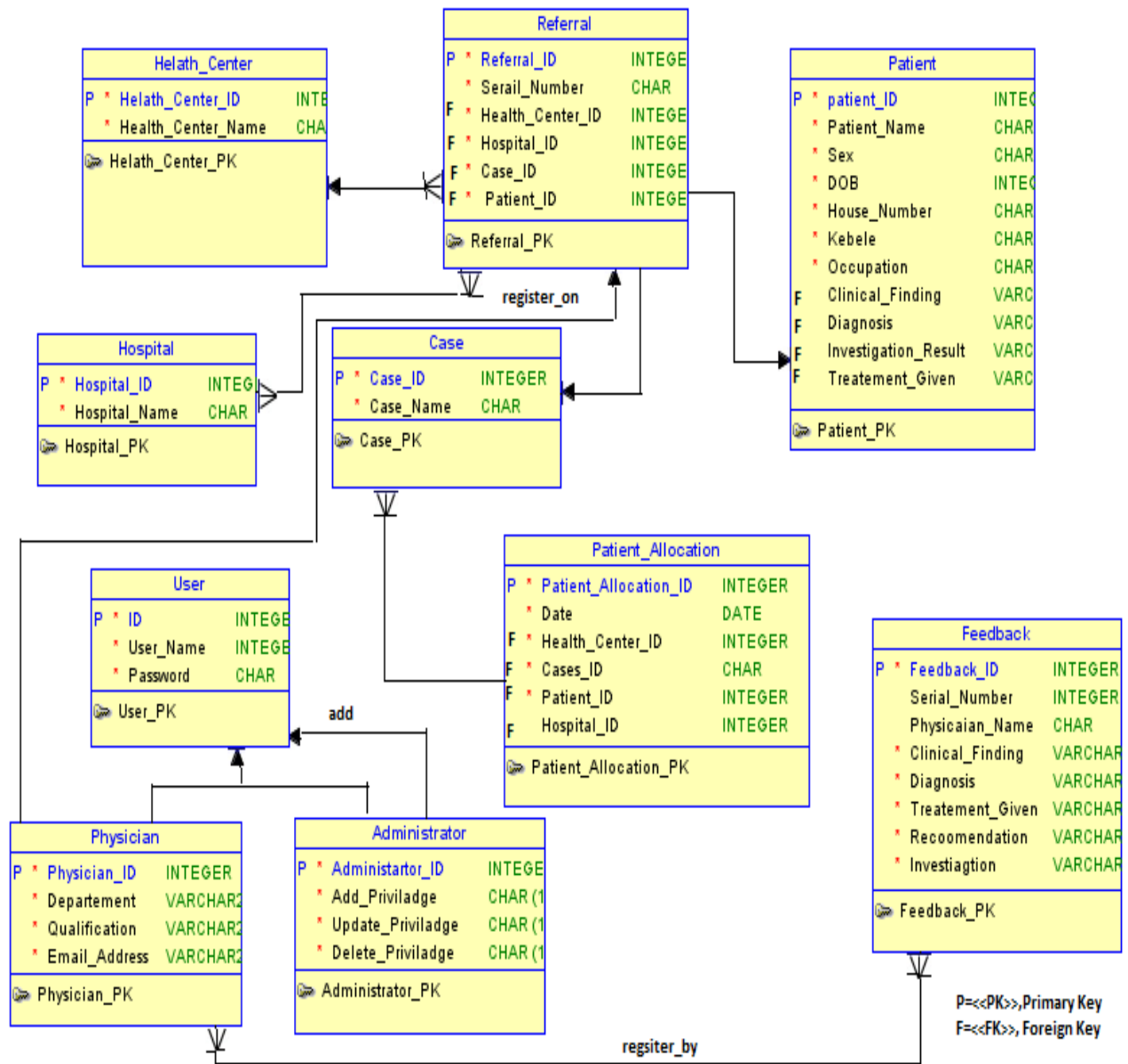


Figure 4.25: Physical Data Base Modeling for WRIMSAA

## **4.5 Implementation**

Object-Oriented implementation phases allow the researcher to build the actual system, WRIMSAA and develop the code that fulfills the system's design which is an end goal to OOA/D. Design and programming are highly interrelated and iterative and developers often move back and forth between them. Your programming efforts quickly reveal weaknesses in your design that need to be addressed, and your design efforts reveal potential strategies to code the system effectively.

Finally a component subsystem as well as a full fledge application that undergo testing, WRIMSAA is developed. The tasks in implementation phase of this study were mainly classified into two.

1. Deliverables, and
2. Things that create deliverables (such as source code and graphics) which is organized as source code in Java packages and byte code organized into JAR (Java Achieve) files.<sup>30</sup>

There are 3 main tasks performed in developing WRIMSAA.

- Implementing Classes using Java.
- Implementing Database using MySQL database and
- Implementing Web application using JSP, Java scripting and HTML language.

### **4.5.1 Class implementation**

Class implementation is one of the main tasks performed in developing WRIMSAA. Your design artifacts, such as UML class and sequence diagram, drive the development of your source code in class implementation. In this phase the researcher focuses on writing source code using an object-oriented programming language tool, Java, for identified classes, interfaces and methods. The concepts identified and modeled in analysis and design stage, are the inputs for developing class modules. All the classes are created under Java class package, source package. Attributes and methods for a class are defined for each class. Finally the written code is successfully build, compiled and run. The code for classes in developing WRIMSAA is shown in Annex 1.

#### **4.5.2 Database implementation**

The data base implementation is the physical realization of the database and application designs which is achieved using the Data Definition Language (DDL) of the selected DBMS (Data Base Management system), MySQL. The DDL statement is used to create the database structure while Data Manipulation Language (DML) is used to process data stored in a database and embedded with in a host programming language Java.

For this study the referral database is created using wamp server package which runs at local host, root@localhost, MySQL 5.1.36 community log with local host via TCP/IP protocol version 10. The database includes 5 tables as shown in Table 4.2 analysis stage of system development. The code for creating and managing referral database is shown in Annex 2.

#### **4.5.3 Web application Implementation**

Web-based system uses a web application which lets the user to access any data through their browser. The user uses queries to access data through web server from database. In developing web application for WRIMSAA the study implements files JSP and JSF scripts embedded with HTML. Glassfish server is used as application server.

JSP is a scripting language which runs on web server, with servlet engine technology, based on the java language and enables the development of dynamic websites. An Enterprise Java Beans (EJB) with JSF file is employed for developing web application. JSP and JSF source code runs on the web application server in the servlet engine and the servlet engine dynamically generates the HTML and sends the HTML output to the clients' web browser. In developing WRIMSAA a code with JSP, JSF and HTML used to implement the web application and described in Annex 3.

#### **4.5.4 Integration and Testing**

The class, and web application modules developed in the previous stage was integrated and finally tested. The study applied

- **Unit testing:** testing of each module for classes, JSF, JSP Script and HTML codes.
- **System testing:** testing the whole system, WRIMSAA application was performed.

## 4.6 Results

Web-based Referral Information Management System for Addis Ababa (WRIMSAA) was developed so that end users could manage and transfer patient referral data online. So that two-way communication between the health centers and hospitals would be possible. It also made possible the redirection of patient to alternative facility based on the information provided in advance to service delivery.

## 4.7 Evaluation of WRIMSAA

For testing, the study selected practitioners that would represent the sample in evaluation of the developed system, WRIMSAA. Thus, 3 physicians from Yekatit 12, Zewditu hospitals and Woreda 24 health centers and 2 computer technicians from hospitals selected. For sample selection the study used simple random sampling techniques. Practitioners selected randomly because tasks in referral facilities are the same and selected samples were representative in larger referral system.

This staff performed the detailed system testing in accordance with the test plan and test execution guide, as well as the integration, functional and user-acceptance testing.

**Functional testing:** refers to testing of the user functions of the software to ensure that they perform according to user specifications. This testing is usually done for each module of an application.

**Integration testing:** involves testing the complete application across multiple modules to verify that the whole application is performing according to user specifications.

**User-acceptance testing:** means having real users use the system and validate that the application is performing according to user specifications.

In addition, testing helps to

- Identify system-use barriers
- Identify areas of refinement in the system.
- Identify possible system faults

The application is installed on users' computer at referral department. The users run and operate the system and test all the functionalities, content and structure of WRIMSAA. After applying the evaluation formal interview is conducted to collect user's response.

#### **4.7.1 Evaluation Result**

Through consideration of the above issues, the following results are obtained.

- All the user response that WRIMSAA fulfills all the functionalities excepted from referral facilities except the financial issues.
- All the users' response there could be system use barriers. For instance, low internet speed, connectivity problem on both hospitals and health centers.
- Lack of IT knowledge in using the system at health centers and lack of motivation to use the system at health centers. By taking this into consideration the referral staffs suggest that there is a need for training before implementing the system.
- Lack of infrastructure such as computers, internet, human resource and finance to use the system and suggest that there is a need for government and other NGOs' involvement in implementing the system.

# **CHAPTER FIVE: CONCLUSION and RECOMMENDATION**

## **5.1 Conclusion**

Improving existing referral system through using Web-based Referral Information Management system becomes an important step that have to be taken in enhancing the quality of referral service provision in Ethiopia. It also plays a great role in reducing morbidity and mortality rate through providing on time referral service which can be achieved using an effective communication, coordination, sharing of patient medication data using Web-based system. On the course of solution developing the shortcoming,

- The study used important concepts from the literature that help the researcher to get some vital issues which can be integrated in the study for improvement of the existing referral system in the process of developing WRIMSAA.
- An approach followed for requirement gathering in developing WRIMSAA helps the researcher to rely on users requirement to develop the new system, that directly contribute to achieve usability.
- UML artifacts and programming tools used in this study makes the development phases clear, understandable and more importantly achieve the features provided by OOA/D methodology, documentation.
- At last the developed system, WRIMSAA can be used as a prototype for adopting the system at national level.

Through all this, the study addresses the research problems and made a significant contribution to the referral information system management in Addis Ababa by developing WRIMSAA which improves the current referral service.

## **5.2 Contribution of the work**

To address the shortcomings observed in the existing referral system, the study proposed and developed a Web-based Referral Information Management system and made a research contribution which deal with the following issues.

- Introduced a database system to handle patient medication data electronically at all levels, hospitals and health centers in Addis Ababa.

- Introduced a new approach and mechanism to access and transfer patient medication file from everywhere by everyone through Web- based Referral Information Management System, with a minimal time.
- Introduced a prior and pre-service provision of information among referral facilities which empowers decision making ability for referral service providers. Especially in making patient allocation flexible through redirection of patients to alternative facilities that mainly address the issues of “time factor” in referral service provision.

### **5.3 Strength and Limitation of the Study**

#### **5.3.1 Strength of the Study**

- This study provides guidance and initiation for future researcher to expand and adopt the system at country level.
- The system developed and introduced by this study plays a great role in changing the working environment for referral department and advances the knowledge of staffs in using IT.

#### **5.3.2 Limitation of the Study**

- Some of the users at health centers and hospitals were not cooperative to provide a detailed service related information which hampers developing the system which is full user-centric.
- Delay in getting response in requirement collection stage from sampled health care providers which makes the study sluggish as requirement gathering is the main tasks to be performed in developing the system.
- Lack of infrastructure such as computers, internet facilities, network and skilled human resources limit the testing tasks and manly contribute to create less motivated users towards using the new system at health centers.

## **5.4 Recommendation**

The study tried to realize the proposed Web-based Referral Information Management System under the objective of addressing the shortcomings of existing referral system. However, the researcher did not believe that the system is generic enough to incorporate potential issues in Web-Based Referral Information Management System. For instance, despite the importance of the issues, the study did not include the system for handling financial issues which is related to referral service provision.

The other line which is not considered in this study is regarding the implementation requirement and development of Web-based system at country level which depend on the health care service providers' capability to run and use the system with all the technological, financial, skill and human resource availability.

Therefore, I recommend future researchers

- To explore the resource requirement, such as financial, technical, human resource and material, for referral health care providers at country level to expand and adopt WRIMSAA.
- To expand WRIMSAA that includes external stakeholders, such as insurance companies for resolving financial issues related to referral service and
- To explore the possibility of providing direct online service to patients in need of referral service including electronic consultation and booking service through direct contact of patient with health care providers if there is no need for face to face specialty visits.

## REFERENCES

1. AMREF in Canada. Work in Ethiopia. 2011 [internet] [updated 2011; cited 2011 Dec 08]. Available from: <Http://www.working Ethiopia.htm>.
2. The earth institute of Columbia University. Center for National Health in Ethiopia. 2011. [Internet] [Cited: 2011 Dec. 10]. Available from: <Http://www.center for Health in Ethiopia.htm>.
3. Addis Ababa City Administration Health Bureau. 2007 [Internet] [Updated: 2008 May 22, Cited: 2011 Dec 05]. Available from: <Http://www. Addis Ababa City Administration Health Bureau.htm>.
4. Tadesse K. Reasons for referrals and time spent from referring sites to arrival at Tikur Anbessa Hospital in emergency obstetric: A prospective study; The Ethiopian Journal of Health Development; 2001; 15(1):17-23.
5. BMW. Referral management-frequently asked questions, guideline for GPs.2006; UK.[Internet] [Updated :2006 August 4, Cited :23 Dec 2011]. Available from: <Http:// www. Referral.html>.
6. Feng, L., Lu, H. Integrating Database and Web Technologies, International. Journal of World Wide Web; 1998;1(2): 73-86.
7. Candace I. & Chris N. Referral management, lesson for success. pdf. [Internet] [Updated: 2010 Jan 5; Cited: 12 Jan 2012]. Available from: <Http:// www.Referral management.htm>.
8. Zewdu G. A framework for multi-agent referral decision support. [Master thesis]. Addis Ababa University, 2008.
9. HIQA , Report and Recommendations on Patient Referrals from General Practice to Outpatient and Radiology Services including the National Standard for Patient Referral Information.2011; [internet] [Cited: 2012 March 13]; Available from: <http://www.hiqa.ie>.
10. L Jae O; M Won-K, K Mi N, K Won; L Young S; L Tae-H, C Han I. A Web-based Information Sharing System for Patient Referral; J Med Internet Res; 1999; 5(439): 11-36
11. Bowling A, Redfern J. The process of outpatient referral and care: the experiences and views of patients, their general practitioners, and specialists. British Journal of General Practice; 2000; 50(451): 116-20.

12. Patient Placement Systems Announces the Integrated Web-to-Referral Feature; GA (PRWEB); Atlanta, [Internet] [Updated: 2009 March 24;Cited: 2012 February 23].
13. Felt-Lisk S, McHugh M, Howell E. Monitoring local safety-net providers: do they have adequate capacity? Health Affairs (Millwood) 2002;21:277-83.
14. Dean F, Tejal K, Michael F, Masha T, Andrew J, David G. A computer-based outpatient clinical referral system. Internal Journal of Medical Informatics 1999, Aug. 3: 55(2): 149-158
15. Christina E W, Danish C, Duedal P, Henning V. The cost benefit of electronic patient referrals in Denmark report. ACCA and MedCom in collaboration with the European Commission Information Society Directorate – General;
16. Tom K.HSE, The National Cancer Control program. Dublin [Internet]; 2006. [Updated: 28 July 2010, Cited: 2012 March 13]; Available from: [http://www. Health Service Executive website /National Cancer Programme.htm](http://www.HealthServiceExecutive/NationalCancerProgramme.htm)
17. Niamh D, Chris K, Eileen R., An Electronic Patient Referral Application: A Case Study from Zambia, Research report. RTI press, March 2010.
18. Choo MC, Thennakon S, Shapey J, Toliias CM. A web-based referral system for neurosurgery-a solution to our problems? UK National Library of Medicine National Institute of Health; 2011; 25(3):384-7.
19. The Hospital for Sick Children (Sick Kids), Refer patient. [Internet][Cited: 2012 March 13]; Available from: <http://www.sickkids.ca/AmbulatoryClinics/index.html>.
20. Judy E. Kim H., Alice H, Chen M., Douglas S. Evaluating Electronic Referrals for Specialty Care at a Public Hospital. J Gen Intern Med. 2010 October; 25(10): 1123–1128.
21. North Coast Referral Network. Internet Referral Information System. [Broacher].Chicago: Humboldt-Del Norte Foundation for Medical Care; 2009.
22. Portland Interhopsital physician association. Physician-to-Physician Community Referral System. [Press release]. 2011; Portland Oregon.
23. EHealth Global Technologies. EHealth Access Record Retrieval System. 4 October 2011; ROCHESTER, New York.
24. Strata Health Solutions. Strata Pathways acts as a seamless yet essential component of a modern EHR, 14 Nov. 2012; UK.
25. Rajani S. Development of an Interactive, Web-Delivered System to Increase Provider–Patient Engagement in Smoking Cessation, J Med Internet Res; USA, 2011; 13(4):87

26. Omer I., Ahmed H, Ali M, Raviraja S. Towards a New Methodology for Developing Web-Based Systems. World Academy of Science, Engineering and Technology; 2008.
27. Craig L. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3<sup>rd</sup> Edition; Addison Wesley Professional, 2004.
28. Bernstein, P. Middleware: A Model for Distributed Services, Communications of the ACM, 1996 Feb 10; 39(2):86-87.
29. Jeremy P. Benefits of using the n-tiered approach for web applications. [Internet] [Cited 2012 Jan 15]; Available from: [Http://www.adobe.com/devnet/coldfusion/articles/ntier.html](http://www.adobe.com/devnet/coldfusion/articles/ntier.html).
30. Scott W.Amblar. The Object Primer: Agile Model-Driven Development with UML 2.0, 3<sup>rd</sup> Edition, Cambridge University Press; 2004.
31. Thomas M ,Connolly B, Carolyn E. DATABASE SYSTEMS,A Practical Approach to Design, Implementation, and Management,3<sup>rd</sup> Edition, Addison Wesley; 2001

# ANNEXES

## Annex 1: Codes for class implementation

```
// Class referral

package classes;
    import java.io.Serializable;
    import java.util.Date;
    import javax.persistence.*;
    import javax.validation.constraints.NotNull;
    import javax.validation.constraints.Size;
    import javax.xml.bind.annotation.XmlRootElement;
        // Member variables
public class Referral implements Serializable {
    private String serialNumber;
    private String cardNumber;
    private Date date;
    private String referredhospitalname;
    private String referringHealthCenterName;
    private String departement;
    private String firstName;
    private String lastName;
    private Date dob;
    private String sex;
    private String occupation;
    private String subCity;
    private String kebele;
    private String houseNumber;
    private String clinicalFinding;
    private String diagnosis;
    private String reasonForReferral;
    private String investigationResult;
    private String physicianName;
    private String treatemnetGiven;
    private String cases;
        //Constructor for class referral
    public Referral() { }
        // setter and getter methods for class referral
    public Referral(String cardNumber) {
        this.cardNumber = cardNumber; }
    public String getSerialNumber() {
        return serialNumber; }
    public void setSerialNumber(String serialNumber) {
        this.serialNumber = serialNumber;}
    public String getCardNumber() {
        return cardNumber; }
    public void setCardNumber(String cardNumber) {
        this.cardNumber = cardNumber;}
    public Date getDate() {
        return date;}
    public void setDate(Date date) {
        this.date = date;}
    public String getReferredhospitalname() {
        return referredhospitalname; }
    public void setReferredhospitalname(String referredhospitalname) {
```

```
this.referredhospitalname = referredhospitalname; }
public String getReferringHealthCenterName() {
    return referringHealthCenterName;}
public void setReferringHealthCenterName(String referringHealthCenterName) {
    this.referringHealthCenterName = referringHealthCenterName; }
public String getDepartement() {
    return departement;}
public void setDepartement(String departement) {
    this.departement = departement;}
public String getFirstName() {
    return firstName;}
public void setFirstName(String firstName) {
    this.firstName = firstName;}
public String getLastName() {
    return lastName;}
public void setLastName(String lastName) {
    this.lastName = lastName;}
public Date getDob() {
    return dob;}
public void setDob(Date dob) {
    this.dob = dob;}
public String getSex() {
    return sex;}
public void setSex(String sex) {
    this.sex = sex;}
public String getOccupation() {
    return occupation;}
public void setOccupation(String occupation) {
    this.occupation = occupation;}
public String getSubCity() {
    return subCity;}
public void setSubCity(String subCity) {
    this.subCity = subCity;}
public String getKebele() {
    return kebele; }
public void setKebele(String kebele) {
    this.kebele = kebele;}
public String getHouseNumber() {
    return houseNumber;}
public void setHouseNumber(String houseNumber) {
    this.houseNumber = houseNumber;}
public String getClinicalFinding() {
    return clinicalFinding;}
public void setClinicalFinding(String clinicalFinding) {
    this.clinicalFinding = clinicalFinding;}
public String getDiagnosis() {
    return diagnosis;}
public void setDiagnosis(String diagnosis) {
    this.diagnosis = diagnosis;}
public String getReasonForReferral() {
    return reasonForReferral;}
public void setReasonForReferral(String reasonForReferral) {
    this.reasonForReferral = reasonForReferral;}
public String getInvestigationResult() {
    return investigationResult;}
public void setInvestigationResult(String investigationResult) {
```

```

        this.investigationResult = investigationResult;}
    public String getPhysicianName() {
        return physicianName;}
    public void setPhysicianName(String physicianName) {
        this.physicianName = physicianName;}
    public String getTreatemnetGiven() {
        return treatemnetGiven;}
    public void setTreatemnetGiven(String treatemnetGiven) {
        this.treatemnetGiven = treatemnetGiven;}
    public String getCases() {
        return cases;}
    public void setCases(String cases) {
        this.cases = cases;}

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (cardNumber != null ? cardNumber.hashCode() : 0);
        return hash;}
    @Override
    public boolean equals(Object object) {
        if (!(object instanceof Referral)) {
            return false;}
        Referral other = (Referral) object;
        if ((this.cardNumber == null && other.cardNumber != null) || (this.cardNumber != null
&& !this.cardNumber.equals(other.cardNumber))) {
            return false;}
        return true;}
    @Override
    public String toString() {
        return "enterprise.jsf_jpa_war.Referral[ cardNumber=" + cardNumber + " ]"; }
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement; }

```

```

// Class Referral control
package class;
//member variables
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import javax.faces.FacesException;
import javax.annotation.Resource;
import javax.transaction.UserTransaction;
import enterprise.jsf_jpa_war.util.JsfUtil;
import enterprise.jsf_jpa_war.util.PagingInfo;
import java.util.List;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import javax.faces.model.SelectItem;
import javax.persistence.EntityManagerFactory;
import javax.persistence.PersistenceUnit;
//constructor for class referral control
public class ReferralController {
    public ReferralController() {

```

```

    pagingInfo = new PagingInfo();
    converter = new ReferralConverter(); }
private Referral referral = null;
private List<Referral> referralItems = null;
private ReferralFacade jpaController = null;
private ReferralConverter converter = null;
private PagingInfo pagingInfo = null;
@Resource
private UserTransaction utx = null;
@PersistenceUnit(unitName = "jsf-jpa-war")
private EntityManagerFactory emf = null;
public PagingInfo getPagingInfo() {
    if (pagingInfo.getItemCount() == -1) {
        pagingInfo.setItemCount(getJpaController().count()) }
    return pagingInfo;}
public ReferralFacade getJpaController() {
    if (jpaController == null) {
        FacesContext facesContext = FacesContext.getCurrentInstance();
        jpaController = emf.createEntityManager(facesContext) (ReferralFacade)
facesContext.getApplication().getELResolver().getValue(facesContext.getELContext(), null, "referralJpa");}
    return jpaController;}
public SelectItem[] getReferralItemsAvailableSelectMany() {
    return JsUtil.selectItems(getJpaController().findAll(), false);}
public SelectItem[] getReferralItemsAvailableSelectOne() {
    return JsUtil.selectItems(getJpaController().findAll(), true);}
public Referral getReferral() {
    if (referral == null) {
        referral = (Referral) JsUtil.getObjectFromRequestParameter("jsfcrud.currentReferral",
        converter, null);}
    if (referral == null) {
        referral = new Referral();}
    return referral;}
public String listSetup() {
    reset(true);
    return "referral_list";}
public String createSetup() {
    reset(false);
    referral = new Referral();
    return "referral_create";}
public String create() {
    try {
        utx.begin();
    } catch (Exception ex) {}
    try {
        Exception transactionException = null;
        getJpaController().create(referral);
        try {
            utx.commit();
        } catch (javax.transaction.RollbackException ex) {
            transactionException = ex;
        } catch (Exception ex) {}
        if (transactionException == null) {
            JsUtil.addSuccessMessage("Referral was successfully created.");
        } else {
            JsUtil.ensureAddErrorMessage(transactionException, "A persistence error occurred.");}
    } catch (Exception e) {

```

```

    try {
        utx.rollback();
    } catch (Exception ex) {}
    JsfUtil.ensureAddErrorMessage(e, "A persistence error occurred.");
    return null; }
return listSetup();}
public String detailSetup() {
    return scalarSetup("referral_detail");}
public String editSetup() {
    return scalarSetup("referral_edit");}
private String scalarSetup(String destination) {
    reset(false);
    referral = (Referral) JsfUtil.getObjectFromRequestParameter("jsfcrud.currentReferral",
    converter, null);
    if (referral == null) {
        String requestReferralString = JsfUtil.getRequestParameter("jsfcrud.currentReferral");
        JsfUtil.addErrorMessage("The referral with id " + requestReferralString + " no longer
        exists.");
        return relatedOrListOutcome(); }
    return destination;}
public String edit() {
    String referralString = converter.getAsString(FacesContext.getCurrentInstance(), null, referral);
    String currentReferralString = JsfUtil.getRequestParameter("jsfcrud.currentReferral");
    if(referralString==null||referralString.length()==0||!referralString.equals(currentReferralString)) { String outcome
= editSetup();
        if ("referral_edit".equals(outcome)) {
            JsfUtil.addErrorMessage("Could not edit referral. Try again.");}
        return outcome;}
    try {
        utx.begin();
    } catch (Exception ex) { }
    try {
        Exception transactionException = null;
        getJpaController().edit(referral);
        try {
            utx.commit();
        } catch (javax.transaction.RollbackException ex) {
            transactionException = ex;
        } catch (Exception ex) { }
        if (transactionException == null) {
            JsfUtil.addSuccessMessage("Referral was successfully updated.");
        } else {
            JsfUtil.ensureAddErrorMessage(transactionException, "A persistence error occurred.") }
    } catch (Exception e) {
        try {
            utx.rollback();
        } catch (Exception ex) {}
        JsfUtil.ensureAddErrorMessage(e, "A persistence error occurred.");
        return null; }
    return detailSetup();}
public String remove() {
    String idAsString = JsfUtil.getRequestParameter("jsfcrud.currentReferral");
    String id = idAsString;
    try {
        utx.begin();
    } catch (Exception ex) { }

```

```

try {
    Exception transactionException = null;
    getJpaController().remove(getJpaController().find(id));
    try {
        utx.commit();
    } catch (javax.transaction.RollbackException ex) {
        transactionException = ex;
    } catch (Exception ex) {}
    if (transactionException == null) {
        JsfUtil.addSuccessMessage("Referral was successfully deleted.");
    } else {
        JsfUtil.ensureAddErrorMessage(transactionException, "A persistence error occurred.");
    }
} catch (Exception e) {
    try {
        utx.rollback();
    } catch (Exception ex) {}
    JsfUtil.ensureAddErrorMessage(e, "A persistence error occurred.");
    return null;}
return relatedOrListOutcome();}
private String relatedOrListOutcome() {
    String relatedControllerOutcome = relatedControllerOutcome();
    if (relatedControllerOutcome != null) {
        return relatedControllerOutcome;}
    return listSetup();}
public List<Referral> getReferralItems() {
    if (referralItems == null) {
        getPagingInfo();
        referralItems = getJpaController().findRange(new int[]{pagingInfo.getFirstItem(), pagingInfo.getFirstItem()
+ pagingInfo.getBatchSize()});}
    return referralItems;}
public String next() {
    reset(false);
    getPagingInfo().nextPage();
    return "referral_list";}
public String prev() {
    reset(false);
    getPagingInfo().previousPage();
    return "referral_list";}
private String relatedControllerOutcome() {
    String relatedControllerString = JsfUtil.getRequestParameter("jsfcrud.relatedController");
    StringrelatedControllerTypeString=JsfUtil.getRequestParameter("jsfcrud.relatedControllerType");
    if (relatedControllerString != null && relatedControllerTypeString != null) {
        FacesContext context = FacesContext.getCurrentInstance();
        ObjectrelatedController=context.getApplication().getELResolver().getValue(context.getELContext(),null,
relatedControllerString);
        try {
            Class<?> relatedControllerType = Class.forName(relatedControllerTypeString);
            Method detailSetupMethod = relatedControllerType.getMethod("detailSetup");
            return (String) detailSetupMethod.invoke(relatedController);
        } catch (ClassNotFoundException e) {
            throw new FacesException(e);
        } catch (NoSuchMethodException e) {
            throw new FacesException(e);
        } catch (IllegalAccessException e) {
            throw new FacesException(e);

```

```

    } catch (InvocationTargetException e) {
        throw new FacesException(e);} }
    return null;}
private void reset(boolean resetFirstItem) {
    referral = null;
    referralItems = null;
    pagingInfo.setItemCount(-1);
    if (resetFirstItem) {
        pagingInfo.setFirstItem(0);} }
public void validateCreate(FacesContext facesContext, UIComponent component, Object value) {
    Referral newReferral = new Referral();
    String newReferralString = converter.getAsString(facesContext.getCurrentInstance(), null, newReferral);
    String referralString = converter.getAsString(facesContext.getCurrentInstance(), null, referral);
    if (!newReferralString.equals(referralString)) {
        createSetup();} }
public Converter getConverter() {
    return converter;} }

```

**//Class user**

```

package calss;

import java.io.Serializable;
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import static javax.persistence.GenerationType.IDENTITY;

public class user implements Serializable {
    private Integer id;
    private String firstname;
    private String lastname;
    private String username;
    private String password;
    private Date since;
    private String usertype;

    public user() { }
    public user(Integer id) {
        this.id = id; }
    public user(Integer id, String firstname, String lastname, String username, String password, String usertype) {
        this.id = id;
        this.firstname = firstname;
        this.lastname = lastname;
        this.username = username;
        this.password = password;
        this.usertype = usertype;
    }
}

```

```

// setter and getter method
public Integer getId() {
    return this.id; }
public void setId(Integer id) {
    this.id = id; }
public String getFirstname() {
    return this.firstname; }
public void setFirstname(String firstname) {
    this.firstname = firstname;}
public String getLastname() {
    return this.lastname; }
public void setLastname(String lastname) {
    this.lastname = lastname;}
public String getUsername() {
    return this.username; }
public void setUsername(String username) {
    this.username = username;}
public String getUsertype(){
    return this.usertype;}
public String getPassword() {
    return this.password;}
public void setPassword(String password) {
    this.password = password;}
public void setUsertype(String usertype){
    this.usertype = usertype;}
public Date getSince() {
    return this.since;}
public void setSince(Date since) {
    this.since = since;}

@Override
public int hashCode() {
    int hash = 0;
    hash += (this.id != null ? this.id.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (object == null || !this.getClass().equals(object.getClass())) {
        return false;}
    user other = (Wuser)object;
    if (this.id != other.id && (this.id == null || !this.id.equals(other.id))) return false;
    return true;}

@Override
public String toString() {
    //TODO change toString() implementation to return a better display name
    return "" + this.id;
} }

```

```

//Class User Control
package calss;

```

```

import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import javax.faces.FacesException;
import javax.annotation.Resource;
import javax.transaction.UserTransaction;
import enterprise.jsf_jpa_war.util.JsfUtil;
import enterprise.jsf_jpa_war.util.PagingInfo;
import java.util.List;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import javax.faces.model.SelectItem;
import javax.persistence.EntityManagerFactory;
import javax.persistence.PersistenceUnit;

public class userController {

    public userController() {
        pagingInfo = new PagingInfo();
        converter = new userConverter() }
    private user user = null;
    private List<user> userItems = null;
    private userFacade jpaController = null;
    private userConverter converter = null;
    private PagingInfo pagingInfo = null;

    public PagingInfo getPagingInfo() {
        if (pagingInfo.getItemCount() == -1) {
            pagingInfo.setItemCount(getJpaController().count());}
        return pagingInfo;}
    public userFacade getJpaController() {
        if (jpaController == null) {
            FacesContext facesContext = FacesContext.getCurrentInstance();
            jpaController = (userFacade)
facesContext.getApplication().getELResolver().getValue(facesContext.getELContext(), null, "userJpa");}
        return jpaController;}
    public SelectItem[] getWuserItemsAvailableSelectMany() {
        return JsfUtil.getSelectItems(getJpaController().findAll(), false);}
    public SelectItem[] getWuserItemsAvailableSelectOne() {
        return JsfUtil.getSelectItems(getJpaController().findAll(), true);}
    public user getWuser() {
        if (user == null) {
            user = (user) JsfUtil.getObjectFromRequestParameter("jsfcrud.currentwuser", converter, null);}
        if (user == null) {
            user = new user(); }
        return user;}
    public String listSetup() {
        reset(true);
        return "user_list";}
    public String createSetup() {
        reset(false);
        user = new user();
        return "user_create";}
    public String create() {
        try {
            utx.begin();

```

```

    } catch (Exception ex) { }
    try {
        Exception transactionException = null;
        getJpaController().create(wuser);
        try {
            utx.commit();
        } catch (javax.transaction.RollbackException ex) {
            transactionException = ex;
        } catch (Exception ex) {}
        if (transactionException == null) {
            JsfUtil.addSuccessMessage("Wuser was successfully created.");
        } else {
            JsfUtil.ensureAddErrorMessage(transactionException, "A persistence error occurred.");
        }
    } catch (Exception e) {
        try {
            utx.rollback();
        } catch (Exception ex) {}
        JsfUtil.ensureAddErrorMessage(e, "A persistence error occurred.");
        return null;
    }
    return listSetup();
}
public String detailSetup() {
    return scalarSetup("wuser_detail");
}
public String editSetup() {
    return scalarSetup("wuser_edit");
}
private String scalarSetup(String destination) {
    reset(false);
    wuser = (user) JsfUtil.getObjectFromRequestParamer("jsfcrud.currentuser", converter, null);
    if (user == null) {
        String requestuserString = JsfUtil.getRequestParameter("jsfcrud.currentuser");
        JsfUtil.addErrorMessage("The wuser with id " + requestuserString + " no longer exists.");
        return relatedOrListOutcome();
    }
    return destination;
}
public String edit() {
    String userString = converter.getAsString(FacesContext.getCurrentInstance(), null, wuser);
    String currentuserString = JsfUtil.getRequestParameter("jsfcrud.currentuser");
    if (userString == null || userString.length() == 0 || !userString.equals(currentuserString)) {
        String outcome = editSetup();
        if ("user_edit".equals(outcome)) {
            JsfUtil.addErrorMessage("Could not edit wuser. Try again.");
        }
        return outcome;
    }
    try {
        utx.begin();
    } catch (Exception ex) { }
    try {
        Exception transactionException = null;
        getJpaController().edit(wuser);
        try {
            utx.commit();
        } catch (javax.transaction.RollbackException ex) {
            transactionException = ex;
        } catch (Exception ex) {}
        if (transactionException == null) {
            JsfUtil.addSuccessMessage("Wuser was successfully updated.");
        } else {
            JsfUtil.ensureAddErrorMessage(transactionException, "A persistence error occurred.");
        }
    } catch (Exception e) {

```

```

        try {
            utx.rollback();
        } catch (Exception ex) { }
        JsfUtil.ensureAddErrorMessage(e, "A persistence error occurred.");
        return null;
    }
    return detailSetup(); }
public String remove() {
    String idAsString = JsfUtil.getRequestParameter("jsfcrud.currentWuser");
    Integer id = new Integer(idAsString);
    try {
        utx.begin();
    } catch (Exception ex) { }
    try {
        Exception transactionException = null;
        getJpaController().remove(getJpaController().find(id));
        try {
            utx.commit();
        } catch (javax.transaction.RollbackException ex) {
            transactionException = ex;
        } catch (Exception ex) {}
        if (transactionException == null) {
            JsfUtil.addSuccessMessage("Wuser was successfully deleted.");
        } else {
            JsfUtil.ensureAddErrorMessage(transactionException, "A persistence error occurred.");
        }
    } catch (Exception e) {
        try {
            utx.rollback();
        } catch (Exception ex) { }
        JsfUtil.ensureAddErrorMessage(e, "A persistence error occurred.");
        return null;
    }
    return relatedOrListOutcome();
}
private String relatedOrListOutcome() {
    String relatedControllerOutcome = relatedControllerOutcome();
    if (relatedControllerOutcome != null) {
        return relatedControllerOutcome;
    }
    return listSetup();
}
public List<user> getuserItems() {
    if (userItems == null) {
        getPagingInfo();
        userItems = getJpaController().findRange(new int[] { pagingInfo.getFirstItem(), pagingInfo.getFirstItem() +
pagingInfo.getBatchSize() });
    }
    return userItems;
}
public String next() {
    reset(false);
    getPagingInfo().nextPage();
    return "wuser_list";
}
public String prev() {
    reset(false);
    getPagingInfo().previousPage();
    return "user_list";
}
private String relatedControllerOutcome() {
    String relatedControllerString = JsfUtil.getRequestParameter("jsfcrud.relatedController");
    String relatedControllerTypeString = JsfUtil.getRequestParameter("jsfcrud.relatedControllerType");
    if (relatedControllerString != null && relatedControllerTypeString != null) {
        FacesContext context = FacesContext.getCurrentInstance();
        Object relatedController = context.getApplication().getELResolver().getValue(context.getELContext(), null,

```

```

relatedControllerString);
    try {
        Class<?> relatedControllerType = Class.forName(relatedControllerTypeString);
        Method detailSetupMethod = relatedControllerType.getMethod("detailSetup");
        return (String) detailSetupMethod.invoke(relatedController);
    } catch (ClassNotFoundException e) {
        throw new FacesException(e);
    } catch (NoSuchMethodException e) {
        throw new FacesException(e);
    } catch (IllegalAccessException e) {
        throw new FacesException(e);
    } catch (InvocationTargetException e) {
        throw new FacesException(e);
    }
}
return null;}
private void reset(boolean resetFirstItem) {
    user = null;
    userItems = null;
    pagingInfo.setItemCount(-1);
    if (resetFirstItem) {
        pagingInfo.setFirstItem(0);} }
public void validateCreate(FacesContext facesContext, UIComponent component, Object value) {
    user newUser = new user();
    String newUserString = converter.getAsString(FacesContext.getCurrentInstance(), null, newUser);
    String userString = converter.getAsString(FacesContext.getCurrentInstance(), null, user);
    if (!newUserString.equals(userString)) {
        createSetup(); } }
public Converter getConverter() {
    return converter;} }

```

### //Class User manager

```

package calss;

import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.annotation.Resource;
import javax.faces.application.FacesMessage;
import javax.faces.context.ExternalContext;
import javax.faces.context.FacesContext;
import javax.persistence.EntityManager;
import javax.persistence.NoResultException;
import javax.persistence.PersistenceContext;
import javax.servlet.http.HttpSession;
import javax.transaction.UserTransaction;

public class UserManager {
//Member variables
    public static final String USER_SESSION_KEY = "user";
    private String username;
    private String password;
    private String passwordv;
    private String fname;
    private String lname;
    private String usertype;

```

```

// setter and getter methods
public String getUertype(){
    return uertype;}
public String getUsername() {
    return username; }
public void setUsername(String username) {
    this.username = username; }
public String getPassword() {
    return password;}
public void setPassword(String password) {
    this.password = password;}
public void setUertype(String uertype){
    this.uertype = uertype;}
public String getPasswordv() {
    return passwordv;}
public void setPasswordv(String passwordv) {
    this.passwordv = passwordv; }
public String getFname() {
    return fname; }
public void setFname(String fname) {
    this.fname = fname; }
public String getLname() {
    return lname; }
public void setLname(String lname) {
    this.lname = lname;
}
}
public String validateUser() {
FacesContext context = FacesContext.getCurrentInstance();
user user = getUser();
    if (user != null) {
        if (!user.getPassword().equals(password)) {
            FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR,
                "Login Failed!",
                "The password specified is not correct.");
            context.addMessage(null, message);
            return null;}
        context.getExternalContext().getSessionMap().put(USER_SESSION_KEY, user);
        return "app-main";
    } else {
        FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR,
            "Login Failed!",
            "Username "
            + username
            + " does not exist.");
        context.addMessage(null, message);
        return null;} }
public String createUser() {
FacesContext context = FacesContext.getCurrentInstance();
user user = getUser();
if(user == null) {
    if (!password.equals(passwordv)) {
        FacesMessage message = new FacesMessage("The specified passwords do not match. Please try again");
        context.addMessage(null, message);
        return null;
    }
}
}

```

```

user = new Wuser();
user.setFirstname(fname);
user.setLastname(lname);
user.setPassword(password);
user.setUsername(username);
user.setUserType(usertype);
user.setSince(new Date());
try {
    utx.begin();
    em.persist(wuser);
    utx.commit();
    return "login";
} catch (Exception e) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "Error creating user!",
    "Unexpected error when creating your account. Please contact the system Administrator");
    context.addMessage(null, message);
    Logger.getAnonymousLogger().log(Level.SEVERE,
    "Unable to create new user", e);

    return null; }
} else {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR,
    "Username "
    + username
    + " already exists! ",
    "Please choose a different username.");
    context.addMessage(null, message);
    return null; } }
public String logout() {
    HttpSession session = (HttpSession)
    FacesContext.getCurrentInstance().getExternalContext().getSession(false);
    if (session != null) {
        session.invalidate(); }
    return "login";}
private user getUser() {
    try {
        User user = (Wuser)
        em.createNamedQuery("user.findByUsername").
        setParameter("username", username).getSingleResult();
        return user;
    } catch (NoResultException nre) {
        return null;
    } } }
} } }

```

## Annex 2: Codes for Database application

```

phpMyAdmin SQL Dump
version 3.2.0.1
http://www.phpmyadmin.net
Host: localhost
Generation Time: May 30, 2012 at 07:42 AM
Server version: 5.1.36
PHP Version: 5.3.0

```

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
Database: `hospital`
```

-----  
**Table structure for table `feedback`**

```
CREATE TABLE IF NOT EXISTS `feedback` (
  `serial_number` char(20) DEFAULT NULL,
  `cardnumber` char(20) NOT NULL DEFAULT "",
  `date` date DEFAULT NULL,
  `referred_hospital_name` char(20) DEFAULT NULL,
  `departement` char(20) DEFAULT NULL,
  `first_name` char(20) DEFAULT NULL,
  `last_name` char(20) DEFAULT NULL,
  `finding` char(20) DEFAULT NULL,
  `diagnosis` varchar(50) DEFAULT NULL,
  `treatment_given` varchar(50) DEFAULT NULL,
  `physician_name` char(20) DEFAULT NULL,
  PRIMARY KEY (`cardnumber`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Dumping data for table `feedback`

```
INSERT INTO `feedback` (`serial_number`, `cardnumber`, `date`, `referred_hospital_name`, `departement`,
`first_name`, `last_name`, `finding`, `diagnosis`, `treatment_given`, `physician_name`) VALUES
('000001', '1', '2012-05-29', 'Zewditu', 'referral', 'abebe', 'lemma', 'lung cancer', 'blood test given', 'pain killer given',
'Ermias');
```

-----  
**Table structure for table `patientallocate`**

```
CREATE TABLE IF NOT EXISTS `patientallocate` (
  `cardnumber` char(20) NOT NULL DEFAULT "",
  `referring_health_center_name` char(20) DEFAULT NULL,
  `referred_hospital_name` char(20) DEFAULT NULL,
  `cases` char(20) DEFAULT NULL,
  `first_name` char(20) DEFAULT NULL,
  `last_name` char(20) DEFAULT NULL,
  `date` date DEFAULT NULL,
  PRIMARY KEY (`cardnumber`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

-----  
**Table structure for table `physician`**

```
CREATE TABLE IF NOT EXISTS `physician` (
  `first_name` char(20) DEFAULT NULL,
  `last_name` char(20) DEFAULT NULL,
  `user_name` char(20) NOT NULL DEFAULT "",
  `qualification` char(20) DEFAULT NULL,
  `departement` char(20) DEFAULT NULL,
  `email_address` char(20) DEFAULT NULL,
  PRIMARY KEY (`user_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
INSERT INTO `physician` (`first_name`, `last_name`, `user_name`, `qualification`, `departement`, `email_address`)
VALUES
('dawit', 'regassa', 'dawit1', 'healthofficer', 'referral', 'dawit@yahoo.com'),
('yosef', 'regassa', 'yosef1', 'Doctor', 'referral', 'yos@yahoo.com');
```

-----  
**Table structure for table `referral`--**

```
CREATE TABLE IF NOT EXISTS `referral` (
  `serial_number` char(10) DEFAULT NULL,
```

```

`card_number` char(10) NOT NULL DEFAULT "",
`date` date DEFAULT NULL,
`Referred_hospital_name` char(30) DEFAULT NULL,
`referring_health_center_name` char(30) DEFAULT NULL,
`departement` char(20) DEFAULT NULL,
`first_name` char(20) DEFAULT NULL,
`last_name` char(20) DEFAULT NULL,
`dob` date DEFAULT NULL,
`sex` char(10) DEFAULT NULL,
`occupation` char(15) DEFAULT NULL,
`sub_city` char(30) DEFAULT NULL,
`kebele` char(10) DEFAULT NULL,
`house_number` char(20) DEFAULT NULL,
`clinical_finding` varchar(50) DEFAULT NULL,
`diagnosis` varchar(50) DEFAULT NULL,
`reason_for_referral` varchar(50) DEFAULT NULL,
`investigation_result` varchar(50) DEFAULT NULL,
`physician_name` varchar(50) DEFAULT NULL,
`treatemnet_given` varchar(50) DEFAULT NULL,
`cases` varchar(50) DEFAULT NULL,
PRIMARY KEY (`card_number`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
Dumping data for table `referral`
INSERT INTO `referral` (`serial_number`, `card_number`, `date`, `Referred_hospital_name`,
`referring_health_center_name`, `departement`, `first_name`, `last_name`, `dob`, `sex`, `occupation`, `sub_city`,
`kebele`, `house_number`, `clinical_finding`, `diagnosis`, `reason_for_referral`, `investigation_result`,
`physician_name`, `treatemnet_given`, `cases`) VALUES
(01, '4', '2012-03-03', 'Alert', 'lideta', 'Dermatology', 'Akale', 'Regassa', '2012-01-02', 'Female', 'Teacher', 'Lideta',
'23', '1099', 'jjjjjjjj', 'ggggg', 'hhh', 'hhhhh', 'Abebe', 'hdjsjs jjhsdkh', 'Deramtology');
-----
Table structure for table `user`
CREATE TABLE IF NOT EXISTS `wuser` (
`id` int(11) NOT NULL,
`firstname` varchar(255) DEFAULT NULL,
`lastname` varchar(255) DEFAULT NULL,
`password` varchar(255) DEFAULT NULL,
`since` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
`username` varchar(255) DEFAULT NULL,
`usertype` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Dumping data for table `user`
INSERT INTO `wuser` (`id`, `firstname`, `lastname`, `password`, `since`, `username`, `usertype`) VALUES (1,
'Akae', 'Regassa', 'akale', '2012-05-29 13:34:19', 'akale', 'Administrator');

```

### Annex 3: Code for web application

```

//Index.jsp, this is the code for home page of the referral system

<% @page contentType="text/html"% >
<% @page pageEncoding="UTF-8"% >
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Referral Information System</title>
    <link rel="stylesheet" type="text/css" href="jsfcrud.css" />
  </head>
  <body>
    <div align="center">
      <h1>Welcome!</h1>
      <h2>ADDIS ABABA CITY ADMINISTRATION </h2>
      <h2>HEALTH BUREAU</h2>
      <h2>Referral Information System</h2>
      
      <br/>
      <a href="/faces/login.jsp">Proceed to the System...</a>
    </div>
  </body>
</html>

```

---

*//login.jsp, this is a code for verifying user in using referral system*

```

<% @page contentType="text/html"%>
<% @page pageEncoding="UTF-8"%>
<% @taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<% @taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Login</title>
    <link rel="stylesheet" type="text/css" href="jsfcrud.css" />
  </head>
  <body>
    <div align="center">
      <h1>Login</h1>
      <f:view>
        <h:messages style="color: red"
          showDetail="true"/>
        <h:form id="login">
          <h:panelGrid columns="2" border="0">
            Username: <h:inputText id="username"
              value="#{userManager.username}"/>
            Password: <h:inputSecret id="password"
              value="#{userManager.password}"/>
            User Type: <h:selectOneListbox id="usertype"
              value="#{userManager.usertype}"
              size="1">
              <f:selectItem itemValue="Administrator"/>
              <f:selectItem itemValue="User"/>
              <f:selectItem itemValue="Physician"/>
            </h:selectOneListbox>
          </h:panelGrid>
          <h:commandButton id="submit"
            type="submit"
            value="Login"

```

```

        action="#{usermanager.validateUser}"/><br>
        <h:commandLink id="create"
            value="Create New Account"
            action="create"/>
    </h:form>
</f:view>
</div>
</body>
</html>

```

---

**//Home.jsp, this is a code for displaying a main menu which directs the user to different pages/screens**

```

<% @page contentType="text/html"%>
<% @page pageEncoding="UTF-8"%>
<% @taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<% @taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Referral Information System</title>
    <link rel="stylesheet" type="text/css" href="jsfcrud.css" />
</head>
<body>
    <f:view>
        <div align="center">
            <h1><h:outputText value="Web Based Referral Information System" /></h1>
            Welcome ${user.firstname} ${user.lastname}! You've been registered since ${user.since}.
            <h3> Main Menu </h3>
            <h:form>
                <a href="/referral/List.jsp">Referrals </a><br>
                <a href="/feedback/List.jsp">Feedbacks </a><br>
                <a href="/patient allocate/List.jsp">Patient Allocate</a><br>
                <h:commandButton id="logout" value="Logout" action="#{usermanager.logout}"/>
            </h:form>
        </div>
    </f:view>
</body>
</html>

```

---

**//New referral.jsp, this is a code for registering/adding a new referral.**

```

<% @page contentType="text/html"%>
<% @page pageEncoding="UTF-8"%>
<% @taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<% @taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<f:view>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>New Referral</title>
        <link rel="stylesheet" type="text/css" href="../jsfcrud.css" />
    </head>
    <body>
        <h:panelGroup id="messagePanel" layout="block">

```

```

    <h:messages errorStyle="color: red" infoStyle="color: green" layout="table"/>
</h:panelGroup>
<h1>New Referral</h1>
<a href=" ../index.jsp">Home</a>
<h:form>
<h:inputHidden id="validateCreateField" validator="#{referral.validateCreate}" value="value"/>
    <h:panelGrid columns="2">
        <h:outputText value="SerialNumber:"/>
<h:inputText id="serialNumber" value="#{referral.referral.serialNumber}" title="SerialNumber" />
        <h:outputText value="CardNumber:"/>
<h:inputText id="cardNumber" value="#{referral.referral.cardNumber}" title="CardNumber" required="true"
requiredMessage="The cardNumber field is required." />
        <h:outputText value="Date (MM/dd/yyyy):"/>
<h:inputText id="date" value="#{referral.referral.date}" title="Date" >
            <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:inputText>
        <h:outputText value="Referredhospitalname:"/>
<h:inputText id="referredhospitalname" value="#{referral.referral.referredhospitalname}" title="Referredhospitalnam
e" />
        <h:outputText value="ReferringHealthCenterName:"/>
<h:inputText id="referringHealthCenterName" value="#{referral.referral.referringHealthCenterName}"
title="ReferringHealthCenterName" />
        <h:outputText value="Departement:"/>
<h:inputText id="departement" value="#{referral.referral.departement}" title="Departement" />
        <h:outputText value="FirstName:"/>
<h:inputText id="firstName" value="#{referral.referral.firstName}" title="FirstName" />
        <h:outputText value="LastName:"/>
<h:inputText id="lastName" value="#{referral.referral.lastName}" title="LastName" />
        <h:outputText value="Dob (MM/dd/yyyy):"/>
<h:inputText id="dob" value="#{referral.referral.dob}" title="Dob" >
            <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:inputText>
        <h:outputText value="Sex:"/>
<h:inputText id="sex" value="#{referral.referral.sex}" title="Sex" />
        <h:outputText value="Occupation:"/>
<h:inputText id="occupation" value="#{referral.referral.occupation}" title="Occupation" />
        <h:outputText value="SubCity:"/>
<h:inputText id="subCity" value="#{referral.referral.subCity}" title="SubCity" />
        <h:outputText value="Kebele:"/>
<h:inputText id="kebele" value="#{referral.referral.kebele}" title="Kebele" />
        <h:outputText value="HouseNumber:"/>
<h:inputText id="houseNumber" value="#{referral.referral.houseNumber}" title="HouseNumber" />
        <h:outputText value="ClinicalFinding:"/>
<h:inputText id="clinicalFinding" value="#{referral.referral.clinicalFinding}" title="ClinicalFinding" />
        <h:outputText value="Diagnosis:"/>
<h:inputText id="diagnosis" value="#{referral.referral.diagnosis}" title="Diagnosis" />
        <h:outputText value="ReasonForReferral:"/>
<h:inputText id="reasonForReferral" value="#{referral.referral.reasonForReferral}" title="ReasonForReferral" />
        <h:outputText value="InvestigationResult:"/>
<h:inputText id="investigationResult" value="#{referral.referral.investigationResult}" title="InvestigationResult" />
        <h:outputText value="PhysicianName:"/>
<h:inputText id="physicianName" value="#{referral.referral.physicianName}" title="PhysicianName" />
        <h:outputText value="TreatemnetGiven:"/>
<h:inputText id="treatemnetGiven" value="#{referral.referral.treatemnetGiven}" title="TreatemnetGiven" />
        <h:outputText value="Cases:"/>
<h:inputText id="cases" value="#{referral.referral.cases}" title="Cases" />
    </h:panelGrid><br />

```

```
<h:commandLink action="#{referral.create}" value="Create"/><br /><br />
<h:commandLinkaction="#{referral.listSetup}"value="ShowAllReferralItems"immediate="true"/>
  <br />
</h:form>
</body>
</html>
</f:view>
```

**Annex 4: Interview guide questions for Hospitals and Health centers in Addis Ababa**

**Addis Ababa University**

**School of Information Science and School of Public Health (joint)**

**Health Informatics Program**

**Key Informant Interview (Guide Questions)**

---

*To: Referral Hospitals and Health centers managers*

*Addis Ababa*

*Date:* \_\_\_\_\_

*Respondents:* The participant in these key informant interviews are the designated staff of referral system.

***Introduction***

My name is Akale Regassa. I am a postgraduate student of Health Informatics at Addis Ababa University. I am currently doing research for my thesis entitled

**“Developing a web based referral management system for referral hospitals and health centers in Addis Ababa”.**

I am asking you esteemed organization for your invaluable support in responding to the attached Interview guide. The interview guide is designed for collecting data about the existing referral system to get information to develop web based referral system in Addis Ababa referral hospitals and health centers.

A web based referral management system is a computer based information management and exchange medium between referral hospitals and health centers. It mainly improves sharing of referred patient data online in a secure manner and improves communication among stakeholders involved in provision of referral service in a timely and less cost manner.

Developing a web based referral management system is a solution for tackling the weak referral system seen in Ethiopia.

***Interview guide questions for collection requirement data of the existing system***

**Area 1. On the current referral system**

1. How is the current referral management system providing service?
2. How the referred patient data transferred/shared among stakeholders in referral system unit?
3. How the referred patient data is stored, accessed and used before and after

service provision?

4. What are the mechanism stakeholders in referral provision unit communicate each other about the service provision status and other related referral related information?

**Area 2. On the performance of the existing referral system (as is?)**

1. What problems seen in the existing referral system?
2. What inefficiency exists in referral system on transferring referred patient data between stakeholders?
3. What communication gap exists among stakeholders of referral system?

**Area 3. On the change required for referral system (to be?)**

1. What shall be done to bring improvement in handling referred patient data on the existing referral system? Do you suggest any other mechanism that brings efficiency?
2. What has to be done to improve communication among stakeholders in referral unit for better service delivery?
3. What do plan to make referral information management system better than the existing in the future?
4. What is expected from stakeholders involved in referral unit to make a necessary change in referral information management system?

*We have finished the interview.*

*Thank you for your patience and spending your much time to conduct this interview!*

---

---

***Interview guide question for evaluation and testing of WRIMSAA***

**Area 1. On the functionality of the referral system developed, WRIMSAA**

1. Is all the function needed by the referral system included in the new system being developed?
2. Is all facilities lets the user to perform the required service?
3. What problems seen in using the new developed system?

**Area 2. On the performance of the referral system developed, WRIMSAA**

4. What makes the new developed system performance different from the previous system?
5. Is there any change in processing referral data considering time and budget it needs, seen in the new designed system?
6. What need to be refined and integrated, if there is?

**Area 3. On the barriers seen in using the referral system developed, WRIMSAA.**

5. What system use barriers you experience in using the new designed system?
6. What has to be done to solve the problems in using the new designed system?

*We have finished the interview.*

*Thank you for your patience and spending your much time to conduct this interview!*

---