A MASTERS PROJECT

ON

WEB BASED HOSPITAL MEDICATION MANAGEMENT SYSTEM IN THE CASE OF ADAMA HOSPITAL MEDICAL COLLEGE

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

M.Sc. IN TECHNICAL TEACHERS EDUCATION IN INFORMATION TECHNOLOGY MANAGEMENT

BY:

HADUSH GODEFA WELDETINSAE (GSR/5302/06)

ATSBHA ABRHA GARED (GSR/5293/06)

ABRHA ZERAY BEHONE (GSR/5287/06)

ADVISORS:

Dr. SRIDHAR UDAYAKUMAR

Mr. FANOS JEMAL

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

JUNE, 2015

ADAMA, ETHIOPIA

DECLARATION

We hereby declare that the project entitled Web-Based Hospital Medication Management System in the case of Adama Hospital Medical College is submitted in partial fulfillment of the requirement for the award of the MSc degree in Technical Teacher Education in Information Technology Management, Adama Science and Technology University, is a record of our own work carried out by us during the academic year 2015 under the supervision and guidance of Dr.Sridhar Udayakumar and co-advisor Mr.Fanos Jemal, Department of Computing, School of Electrical Engineering and computing. The extent and source of information are derived from the existing literature and have been indicated through the project at the appropriate place. The matter embodied in this work is original and has not been submitted for the award of any other degree, either in this or any other University.

Name	Date	Signature
Hadush Godefa	_____	_____
Abrha Zeray	_____	_____
Atsbha Abrha	_____	_____
Advisor		
Dr.Sridhar Udayakumar	_____	_____
Co-Advisor		
Mr.Fanos Jemal	_____	_____

June, 2015

Adama, Ethiopia

Approval sheet

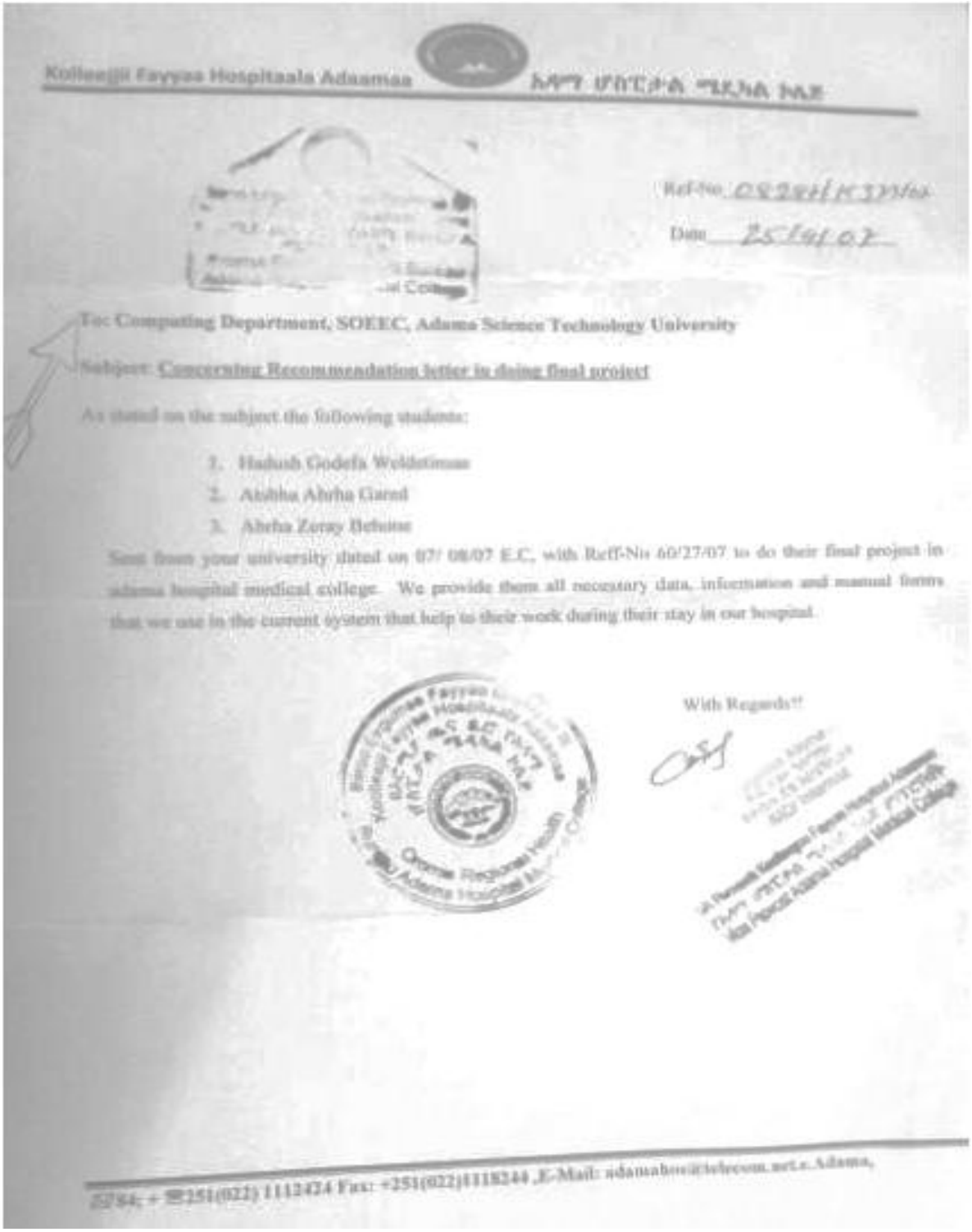
A. Candidate

Hadush Godefa	_____	_____
(Student Name)	(Signature)	(Date)
Abrha Zeray	_____	_____
(Student Name)	(Signature)	(Date)
Atsbha Abrha	_____	_____
(Student Name)	(Signature)	(Date)

B. Approved By

Dr. Sridhar Udayakumar	_____	_____
(Advisor)	(Signature)	(Date)
Mr. Fanos Jemal	_____	_____
(Co-advisor)	(Signature)	(Date)
_____	_____	_____
Internal examiner name	(Signature)	(Date)
_____	_____	_____
External Examiner name	(Signature)	(Date)
_____	_____	_____
Chairperson	(Signature)	(Date)
_____	_____	_____
Department Head	(Signature)	(Date)
_____	_____	_____
School Dean	(Signature)	(Date)

Recommendation Letter from Adama Hospital Medical College



Kolleejjii Fayyaa Hospitaala Adaamaa አዳማ ሆስፒታል ሜዲካል ኮሌጅ

Ref-No. ሰ.ጤ.ሆ/ኮ.ፕ/1796/07

Date 25/04/07

To: Computing Department, SOEEC, Adama Science Technology University

Subject: Concerning Recommendation letter to doing final project

As stated on the subject the following students:

1. Hadush Godefa Woldetinsae
2. Alubba Ahrha Garsi
3. Abeba Zeray Behutse

Sent from your university dated on 07/ 08/07 E.C, with Refl-No 60/27/07 to do their final project in adama hospital medical college. We provide them all necessary data, information and manual forms that we use in the current system that help to their work during their stay in our hospital.

With Regards!!

Oromia Regional Health Bureau

Adama Hospital Medical College

ADAMA HOSPITAL MEDICAL COLLEGE

Dr Yemesh Kebbede Fayisa Hospital Director

ዶ/ር ይመሽ ከበደ ፋይሳ የሆስፒታሉ ዳይሬክተር

ADA Hospital Adama Hospital Medical College

22764; + 251(022) 1112424 Fax: +251(022)1118244 ,E-Mail: adamahos@telecom.net.e. Adama,

ACKNOWLEDGMENT

First and foremost, we would like to give a special gratitude to the Gracious God who provided us everything to finish our courses and this project.

We gratefully acknowledge our advisor Dr. Sridhar Udayakumar and co-advisor Mr.Fanos Jemal for their commitment and patience for reading every section of the project, their valuable comments, encouragement and guidance from initial to the final level of the project that enabled us to finish the project work.

Our grateful thanks go to Dr. Prathap Mani for providing us necessary information, guidelines and project formats on time.

We grateful thank to Adama Hospital Medical College staffs specially Dr. Legesse Alemayohu medical director of the Hospital, Mr. Dejene Tolla coordinator of HMIS Department, Sister Workye, Mr. Tamru Gurmu and Mrs. Atsede Wubneh who devoted their golden time providing us all necessary information, manuals, medical forms and for their additional interviews throughout the project work.

We gratefully thank to school of Educational science and technology teacher education of Adama Science and Technology University for their cooperation in the successful completion of our project.

Finally, we would like to thank all our friends who share us constructive ideas to the successful completion of the project.

TABLE OF CONTENTS

TITLE PAGE

ACKNOWLEDGMENT I

LIST OF TABLES VIII

LIST OF FIGURES XI

LIST OF ACRONYMS XV

ABSTRACT XVII

CHAPTER ONE

INTRODUCTION

1.1 Background of the Project 1

1.2 Purpose and Description of the Project 1

1.3 Statement of the problem 2

1.4. Objectives of the Project 2

1.4.1. General objective 2

1.4.2. Specific objectives 2

1.5. Scope and Limitations of the Project 3

1.5.1. Scope of the project 3

1.5.2. Limitations of the Project 3

1.6. Significance of the Project 3

1.6.1. Beneficiaries of the project and stakeholders 4

1.7. Methodology 4

1.7.1. Data Collection method 4

1.7.2. System Analysis and Design Methodology 5

1.7.3. System Development Tools 5

1.8. Feasibility Study 6

1.8.1. Technical Feasibility 6

1.8.2. Operational Feasibility 6

1.8.3. Economic Feasibility 6

CHAPTER TWO

LITERATURE REVIEW

2.1 THEORETICAL BACKGROUND 7

2.1.1. Introduction 7

2.1.2. Characteristics of a web-based system 7

2.1.3. Goals of web-based system 8

2.1.4. The model for web-based systems 8

2.1.5. The need for web-based medication management system 9

2.1.6. Benefits of the web-based medication management system 10

2.1.6.1. Improve patient care 10

2.1.6.2. Improve cost control 10

2.1.6.3. Increased Security 10

2.2. RELATED LITRATURE 11

2.2.1. Medication management 11

2.2.2. Pharmacist's role in medication management 12

2.3. RELATED STUDIES 13

CHAPTER THREE

REQUIREMENT ANALYSIS

3.1 THE CURRENT SYSTEM 15

3.1.1 Overview of current system 15

3.2 THE PROPOSED SYSTEM.....21

3.2.1 Overview of the proposed system.....21

3.2.2 Functional requirements.....21

3.2.3 Non-functional requirements.....22

3.2.3.1. User interface and human factors.....22

3.2.3.2. Documentation.....22

3.2.3.3. Hardware consideration.....22

3.2.3.4. Performance characteristics.....23

3.2.3.5. Error handling and extreme conditions.....23

3.2.3.6. Quality issues.....23

3.2.3.7. System modifications.....23

3.2.3.8. Physical environment.....23

3.2.3.9. Security issues.....23

3.2.3.10. Resource issues.....23

3.2.3.11. Robustness.....23

3.2.3.12. Maintainability.....24

3.2.3.13. Privacy.....24

3.3. HARDWARE AND SOFTWARE REQUIREMENTS.....24

3.3.1. Hardware Requirements.....24

3.3.2. Software Requirements.....24

3.3.3. Network Requirements.....24

3.4. SOFTWARE ENGINEERING METHODS AND TECHNOLOGIES.....25

3.5. ANALYSIS MODELS.....26

3.5.1. Use case model.....27

3.5.2. Object Model.....35

3.5.2.1. Data dictionary.....35

3.5.2.2. Class Diagram 47

3.5.3. Dynamic Model 48

3.5.3.1. Sequence diagrams 48

3.5.3.2. State chart diagram 62

3.5.3.3. Activity diagrams 63

CHAPTER FOUR

SYSTEM DESIGN

4.1. INTRODUCTION 69

4.1.1 Purpose of the system 69

4.1.2 Design goals 69

4.1.3. Overview of system design 70

4. 2. CURRENT SOFTWARE ARCHITECTURE 70

4. 3. PROPOSED SOFTWARE ARCHITECTURE 71

4.3.1. Overview 71

4.3.2. Subsystem decomposition 72

4.3.3. Hardware/Software mapping 72

4.3.4. Persistent data management 73

4.3.5. Data flow diagram 75

4.3.6. Access control and security 81

4.3.7. User Interface Design 82

4.3.8. Global software control 83

4.3.9. Boundary conditions 83

4.4. SUBSYSTEM SERVICES 84

4.5. Deployment 85

CHAPTER FIVE

IMPLEMENTATION AND RESULT

5.1. ALGORITHMS USED.....87

5.2. METHODS USED.....91

5.3. SAMPLE INPUT/OUTPUT SCREEN SHOOTS 92

5.4. SAMPLE CODES.....96

5.5. RESULTS 101

CHAPTER SIX

SYSTEM TESTING

6.1. TEST PLAN 110

6.1.1. Introduction.....110

6.1.2. Test Plan Overview.....110

6.1.3. Features to be tested/not to be tested 111

6.1.4. Test approaches.....111

6.1.5. Testing materials (Hardware/Software requirements).....118

6.2. TEST CASE SPECIFICATION 118

6.2.1. Test case specification identifier.....118

6.2.2. Test items 118

6.2.3. Input specification 119

6.2.4. Output specification 119

6.2.5. Environmental needs.....119

6.2.6. Special procedural requirements.....119

6.2.7. Interface dependencies.....119

CHAPTER SEVEN

CONCLUSION AND RECOMENDATION

7.1. CONCLUSION 125

7.2. RESULTS ACHIEVED 126

7.3. FUTURE WORK 127

APPENDIX I

User manual/guide 128

APPENDIX II

Questioners 133

APPENDIX III

Glossary 134

BIBLIOGRAPHY.....138

LIST OF TABLES

Table 3.1: Login use case....28
Table 3.2: Manage staff use case 29
Table 3.3: Manage user account use case 30
Table 3.4: Manage patient detail use case 30
Table 3.5: Manage appointment use case 31
Table 3.6: Manage room use case 31
Table 3.7: Manage patient treatment use case 32
Table 3.8: Manage admission summary use case 32
Table 3.9: Manage prescription use case 33
Table 3.10: Manage drugs use case 33
Table 3.11: Manage payment use case 34
Table 3.12: Generate report use case 34
Table 3.13: Patient registration 35
Table 3.14: Appointment 36
Table 3.15: Drugs....36
Table 3.16: Admission discharge registers 37
Table 3.17: Discharge summary 38
Table 3.18: Hematology....39
Table 3.19: Physician order sheet 40
Table 3.20: Other lab forms 40
Table 3.21: Prescription 41

Table 3.22: Payment 41

Table 3.23: Staff.....42

Table 3.24: Progress note.....43

Table 3.25: Urine 43

Table 3.26: User.....44

Table 3.27: Ultra sound 44

Table 3.28: X-ray 45

Table 3.29: Lab-stool.....45

Table 3.30: Room.....46

Table 3.31: Admission summary.....46

Table 4.1: Design goal 70

Table 4.2: Access control and security 82

Table 6.1: Registration of staff unit testing 113

Table 6.2: Create user account unit testing 114

Table 6.3: Login unit testing 114

Table 6.4: Submit prescription 115

Table 6.5: View and submit prescription.....115

Table 6.6: Login test 120

Table 6.7: Manage staff test case.....120

Table 6.8: Manage user account test.....121

Table 6.9: Manage patient detail test case 121

Table 6.10: Manage appointment test case.....122

Table 6.11: Manage submit prescription test case 122

Table 6.12: Manage examination test case 123

Table 6.13: Manage Drugs test case 123

Table 6.14: Manage payment test case 124

Table 6.15: Manage generate report test case 124

LIST OF FIGURES

Figure 2.1: Local Intranet as an example of a Web-based system.....7
Figure 3.1: Patient registration cards 15
Figure 3.2: Admission discharge register form.....16
Figure 3.3: Physician order sheet.....16
Figure 3.4: Other lab request form.....17
Figure 3.5: Prescription form.....17
Figure 3.6: Progress note form.....18
Figure 3.7: Stool lab request form 18
Figure 3.8: Urine test request form 19
Figure 3.9: Hematology request form.....19
Figure 3.10: Ultra-sound request forms 20
Figure 3.11: X-ray request forms.....20
Figure 3.12: Waterfall model.....26
Figure 3.13: Use case diagram.....28
Figure 3.14: Class diagram 47
Figure 3.15: Sequence Diagram for manage staff 48
Figure 3.16: Sequence Diagram for manage user account 49
Figure 3.17: Sequence Diagram for manage patient 50
Figure 3.18: Sequence Diagram for manage room 51
Figure 3.19: Sequence Diagram for manage appointment 52
Figure 3.20: Sequence Diagram for appointment 53
Figure 3.21: Sequence Diagram for manage patient treatment 54
Figure 3.22: Sequence Diagram for admission summary 55
Figure 3.23: Sequence Diagram for discharge summary 56

Figure 3.24: Sequence Diagram for manage patient treatment 57
Figure 3.25: Sequence Diagram for prescription 58
Figure 3.26: Sequence Diagram for manage drug 59
Figure 3.27: Sequence Diagram for manage payment 60
Figure 3.28: State chart Diagrams 62
Figure 3.29: Activity Diagram of Manage Staff and Manage User Account 63
Figure 3.30: Activity Diagram for Manage Patient detail, room and Appointment 64
Figure 3.31: Activity Diagram for Manage Patient Treatment, Admission Summary, Discharge Summary and Appointment 65
Figure 3.32: Activity Diagram for Manage Patient Treatment.....66
Figure 3.33: Activity Diagram for View Prescription and Manage Drugs 67
Figure 3.34: Activity Diagram for Manage Payment 68
Figure 4.1: current software architecture 71
Figure 4.2: proposed software architecture 71
Figure 4.3: sub system decomposition 72
Figure 4.4 Hardware and software mapping 73
Figure 4.5: Persistent database relation diagram 74
Figure 4.6: Data flow context level 0.....75
Figure 4.7: Administrator Data flow Diagram level 1.....76
Figure 4.8: Receptionist data flow diagram level 1 77
Figure 4.9: Data flow Diagram of Doctor Level 1.....78
Figure 4.10: Technicians data flow diagram level 1.....79
Figure 4.11: Pharmacist data flow diagrams level 1.....80
Figure 4.12: Cashiers data flow diagrams level 1.....81
Figure 4.13: User Interface Design 83

Figure 4.14: Deployment diagram for Log in 85
Figure 4.15: User interaction deployment diagram (home page) 86
Figure 5.1: Login page 92
Figure 5.2: Admin page 93
Figure 5.3: Doctor's page 93
Figure 5.4: Receptionist page 94
Figure 5.5: Technician page.....94
Figure 5.6: Pharmacist page.....95
Figure 5.7: Cashier page 95
Figure 5.8: Add staff form 101
Figure 5.9: View staff form 101
Figure 5.10: Add user form 102
Figure 5.11: View user form 102
Figure 5.12: Add patient detail form 103
Figure 5.13: View patients list form 103
Figure 5.14: Add appointment form 104
Figure 5.15: View appointment lists form 104
Figure 5.16: Add and Update prescription form 105
Figure 5.17: View prescription form 105
Figure 5.18: Urine lab request lists form 106
Figure 5.19: Add result 106
Figure 5.20: View prescription form 107
Figure 5.21: Lists of drugs form 107
Figure 5.22: Add/update drugs form 107
Figure 5.23: View payment form 108

Figure 5.24: Add/update payment form 108
Figure 5.25: generate payment report form 109
Figure 5.26: Print payment report 109
Figure 6.1: Level of testing 112
Figure 6.2: Information needed at different levels of testing 112
Figure 6.3: System Testing 116

LIST OF ACRONYMS

ACG	American college of Gastroenterology
ACGIL	Association Cristina Gais Lesbians
ACL	Access Control List
API	Application programmer's interface
BSC	Bachelor of Science
CSS	Cascading style sheet for Dreamweaver
DFD	Data Flow Diagram
DOA	Date of admission
DOD	Date of discharge
FRD	Functional Requirements document
GIS	Geographic Information System
GUI	Graphical User Interface
HIV	Human Immunno deficiency Virus
HMIS	Hospital management information system
HMPMS	Hospital medication and payment management system
HTTP	Hypertext transfer protocol
ID	Identity
IDE	Integrated development environment
IPD	Inpatient Department
MMS	Medication Management System
MRN	Medical record number
OMT	Object modeling technique
OOP	Object oriented programming

OOP	Object oriented programming
OPD	Outpatient Department
OS	Operating system
PC	Personal Computer
PH	Patient history
PHP	Hypertext preprocessor
QTY	Quantity
RAM	Random Access Memory
RBC	Red Blood Cells
RTM	Requirements traceability matrix
SDLC	Software Development Life cycle
SID	Staff Identity
SOD	Separation of Duty
UI	User Interface
UML	Unified modeling language
WAMP	Windows for Apache, MYSQL and PHP
WBC	White Blood Cells
WBHMPMS	Web based hospital medication management system.
XAMPP	Extended for Apache, MYSQL and PHP
YYYY/DD/MM	Year, Date and Month

ABSTRACT

A web-based system enables computers to integrate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. It can also be simply defined as a collection of independent computers that appears to its users as a single coherent system. According to this to develop the Web-Based Hospital Medication Management System we studied the existing system on reception, doctors, technicians (laboratory, radiology), pharmacy and finance using interview, observation/demonstration and by collecting the necessary information we identify the problems of the existing system. In order to solve the identified problems a Web-Based Hospital Medication Management System is developed. The system allows to replace the manual system by a web based system by integrate all services. Our project provides the benefits of streamlined operations, enhanced administration and control, superior patient care, strict cost control and improved service. It is powerful, flexible, easy to use, designed and developed to deliver real conceivable benefits to the hospital. The system is accessible by each actor (administrator, receptionist, doctor, technician, pharmacist and cashier) using his/her username and password. They can view, add, modify or delete data in their own pages to and from the database. The data can be retrieved easily; the interface is user-friendly, well protected data for personal use and makes fast data processing. This project is developed based on the database, scripting language PHP. As there are many areas where we keep the records in database for which we are using MYSQL software which is one of the best and the easiest software to keep our information. In general this project uses PHP as the front-end software which is a server side scripting programming language and has connectivity with MYSQL. All the required modules and features have been particularly built to just fit in to the requirement. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. It covers all the required modules right from patient registration, up to discharge summary details.

CHAPTER ONE
INTRODUCTION

1.1. Background of the Project

In most developing countries, provision of basic preventive, promotive and curative services is a major concern of the government. With growing population and advancement in the medical technology and increasing expectation of the people especially for quality curative care, it has now become imperative to provide quality health care services through the hospitals.

However, these services have not been successful in gaining non redundant, secured, accurate and easily accessible data. The project Web-Based Hospital Medication Management System is a simple effective management structure that we introduce to Adama Hospital Medical College (Hailemariam Hospital) which is one of the public hospitals found in Adama city. This system acts for the hospitals to manage the affairs of the hospital in an automated manner. It is free to prescribe, generate and use the functioning and maintaining the quality of services.

1.2. Purpose and Description of the Project

Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities and records of its patients, doctors, and other staff personals that keep the hospital running smoothly and successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It is also very inefficient and a time-consuming process observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and also it is not economically and technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project, we have developed a web-based version of the manual system, named as “Web based hospital medication management system”. The main aim of our project is to provide a paper-less and automated hospital medication and payment management system. It also aims at providing low cost and reliable automation of the existing system. The system also provides excellent security of data at

every level of user system interaction and also provides robust and reliable storage and backup facilities.

1.3. Statement of the Problem

After careful investigation of the current method of rendering medication and allied services to the patients at the Adama Hospital Medical College (Hailemariam hospital) the following problems were discovered:

1. Non-integration of functional area involved in service delivery
2. In-accurate record keeping and difficultly to retrieve data from case files
3. Lack of prompt updating
4. Redundancy of data
5. Time-consuming service
6. Lack of security

Based on these problems we develop this web-based system which is user friendly, easily accessible, secured, fast retrieval of data, and cost effective to the hospital. It also registers and stores patient details as well as doctor and staff details and retrieves these details as and when required, manipulates these details meaningfully and stores information about the patient.

1.4. Objective of the Project

The study has the following general and specific objectives:

1.4.1. General objective

The main objective of this project work is to design an efficient Web-based system in Adama hospital medical college and to integrate the services that perform separately by the functional areas of the hospital, which will make it easy to follow up a patient's medical record from any workstation in the hospital. This will replace the method of one point of entry and retrieval that exist in non-web-based systems.

1.4.2. Specific objective

To achieve the general objective the project has the following specific objectives:

- To understand the basic concept, principle and technologies of web based system for the

advancement of medical research and analysis

- To register and store patients, doctors and other staffs detail accurately and securely
- To easily notify treatments made by the doctor to patients
- To automat patients medical examination
- To perform automated drug prescription
- To perform automated payment for prescribed medicines
- To check the lists of drugs of the hospital in an automated manner
- To retrieve and generate report of patient, doctor and other staff details as and when required
- To avoid redundancy of data

1.5. Scope and Limitations of the Project

1.5.1. Scope of the project

The focus area of web based hospital medication management system is Adama Hospital Medical College (Hailemariam Hospital). The project covers mainly to register and retrieve patients', doctors' and other staff details, doctor appointment, patient treatment, patient examination, drug prescription, payment system and report generation.

On the other hand the scope outs of our project is online registration of patient, some medical departments like surgical, psychiatric, gynecology etc are not included.

1.5.2. Limitations of the project

While developing and implementing our project, the following limitations can be observed:

- This project can be accessed only within the hospital means it isn't accessible by external bodies

1.6. Significance of the Project

The benefits that can be obtained after the implementation of this system are:

- Automating the front office management of the hospital
- Automated registration and storage of patient information, doctor and other staffs details
- Administration of doctors' appointment
- Development of software which is user friendly, simple, fast, and cost–effective
- Retrieval of patient details as and when required

- Generation of hospital reports meaningfully and securely

1.6.1. Beneficiaries of the project and stakeholders

The beneficiaries of the system are patients, doctors and hospital management of the hospital.

- The way patients can be benefited is by getting effective and efficient service within a short time accurately and this leads to save their precious life and minimize cost.
- The way hospital departments can be benefited is by integrating all functional areas and saving their time to perform end-to-end tasks.

1.7. Methodology

We tried to use the following system analysis methodology to gather information for the project work and to study the existing manual system.

1.7.1. Data collection methods

In order to collect necessary project data we used the following data collection methods

Primary data collection

- Questionnaires: are used to identify users' attitude toward the current system and also what they need on the proposed system with regard of service provided, flexibility, accessibility and time.
- Interview**:** are face to face interactions with the selected interview from the hospital to know their attitude toward the proposed system. Therefore, we made an interview with the hospital personnel about how the hierarchical structure of its human resource management is organized. We also made an interview with the medical director, HMIS Department, Admission and Discharge Register staff.
- Observation: we conduct a strict observation in all the manual activities of the hospital starting from patient admission, laboratory and radiology, pharmacy and the payment system.

Secondary data collection

To have a detailed knowledge and understanding of the current system we gathered the required documents from the hospital. We understood previous work related with the proposed system

that could help us as an input and facilitate the development of the new system. During document analysis, we gave a special consideration to those documents which can bring more features to our system.

1.7.2. System analysis and design methodology

- Class diagram
- Use case diagram
- Activity diagram
- Sequence diagram
- State chart diagram

1.7.3. System development tools

In our project we use the following tools

Frontend: PHP is a language that is specifically designed for desktop programming as front-end software.

PHP has the following Features:

- Easy to use**:** Compared to most solutions like Java, PHP doesn't need to be compiled, so it's just to write the script and then upload it to the server and then update the browser.

- Integrated database support**:** PHP has built-in support for the most popular databases like MYSQL.

Backend

MYSQL**'**s an open source relational database management system. It is based on the structured query language which is consistent, fast performance, high reliability and ease of use.

WAMP: is a freely available Web server that is distributed under an "open source" license.

Application soft wares

Microsoft Visio 2007: application software intended for visual modeling

Macromedia Dreamweaver8: for script the necessary codes to design the project (HTML, PHP, CSS, XML …)

1.8. Feasibility Study of the Project

Feasibility study is conducted once the problem is clearly understood. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility study is to determine if the problem is worth solving. The system has been tested for feasibility in the following points.

1.8.1. Technical Feasibility Study

The software used is much faster and reliable than its counterparts and its usability makes it ideal for quick access to the hospitals entire records.

It has a graphical user interface that assists employees that are not from IT background. It should be flexibility, scalability and extensibility according to the hospital requirement. The project is technically feasible because it was developed in PHP which has built-in support for the most popular databases. It provides the high level of reliability, availability and compatibility thus all makes PHP an appropriate language for this project.

1.8.2. Operational Feasibility Study

Operational feasibility aspects of the project are to be taken as an important part of the project implementation. The system is operationally feasible as it is very easy for the end users to operate and it only needs basic information about window platform.

1.8.3. Economic Feasibility Study

Economic feasibility refers to the benefits or outcomes that derived from the project compared to the total cost spending for the development of the new system. Therefore the new system is economically feasible because it can be developed through open source software. This web-based system will help in automate the selection leading the profits and details of the hospital.

CHAPTER TWO

LITERATURE REVIEW

2.1. THEORETICAL BACKGROUND

2.1.1. Introduction

A Web-based system consists of a collection of autonomous computers, connected through a network and distribution middleware, which enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. It can also be simply defined as a collection of independent computers that appears to its users as a single coherent system. [14]

2.1.2. Characteristics of a web-based system

Characteristics of a web-based system are:

- Multiple autonomous components
- Components are not shared by all users
- Resources may not be accessible
- Software runs in concurrent processes on different processors
- Multiple points of control
- Multiple points of failure

Examples of Web-based Systems

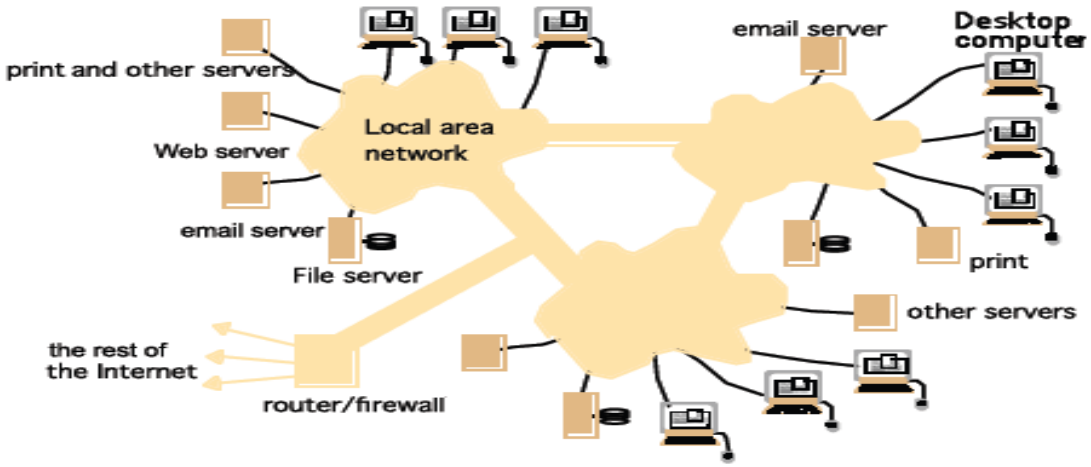


Figure 2.1: Local area networking system as an example of a Web-based system

2.1.3. Goals of Web-based system

- Making resources accessible
- Distribution transparency
- Openness
- Scalability
- Security
- System design requirements

Challenges for Making Resources Accessible

- Naming
- Access control
- Security
- Availability
- Performance
- Mutual exclusion of users, fairness
- Consistency in some cases

2.1.4. The model for web-based systems

Traditionally, having a monolithic system run across multiple computers meant splitting the system into separate client and server components. In such systems, the client component handled the user interface and the server provided back-end processing, such as database access, printing, and so on. As computers proliferated, dropped in cost, and became connected by ever-higher bandwidth networks, splitting software systems into multiple components became more convenient, with each component running on a different computer and performing a specialized function. This approach simplified development, management, administration, and often improved performance and robustness, since failure in one computer did not necessarily disable the entire system. [16]

In many cases the system appears to the client as an opaque cloud that performs the necessary operations, even though the Web-based system is composed of individual nodes.

The opacity of the cloud is maintained because computing operations are invoked on behalf of the client. As such, clients can locate a computer (a node) within the cloud and request a given operation; in performing the operation, that computer can invoke functionality on other computers within the cloud without exposing the additional steps, or the computer on which they were carried out, to the client. With this paradigm, the mechanics of a Web-based, cloud-like

system can be broken down into many individual packet exchanges, or conversations between individual nodes.

Traditional client-server systems have two nodes with fixed roles and responsibilities. Modern-Web-based systems can have more than two nodes, and their roles are often dynamic. In one conversation a node can be a client, while in another conversation the node can be the server. In many cases, the ultimate consumer of the exposed functionality is a client with a user sitting at a keyboard, watching the output. In other cases the Web-based system functions unattended, performing background operations.

The Web-based system may not have dedicated clients and servers for each particular packet exchange, but it is important to remember there is a caller, (or initiator, either of which is often referred to as the client). There is also the recipient of the call (often referred to as the server). It is not necessary to have two-way packet exchanges in the request-reply format of a Web-based system; often messages are sent only one way. [13]

2.1.5 The need for web-based medication management system

According to ACG InfoTech Limited, hospital management information systems are "designed to improve the quality and management of clinical care and hospital health care management in the areas of clinical process analysis and activity-based costing." Improved patient care, cost control and security are possible benefits of implementing such systems. However, the implementation of hospital management information systems must be accompanied by significant planning on the part of management to maximize results.

Hospital management system is been designed to streamline the business practices of hospitals and clinics. Through its electronic patient information system Hospital Medication Management ensures optimum use of the medical records of patients. Being patient-centric, the system makes it possible for all disparate files on a patient to be housed in a centrally located master file, ensuring retrieval and management of files effortless. Plus, easy integration ensures a faster start up. [15]

Hospital medication management has been so designed as to make its usage intuitive, which makes adoption of hospital medication management by the hospital personnel quick and easy. Its high level of scalability ensures quick upgrades when change in the hospital's business practices so demands. What is more, with information collection, collation and retrieval all digitized, patient care regains top of mind status with physicians and hospital personnel. Hospital

Management System provides the benefits of streamlined operations, enhanced administration and control, superior patient care, strict cost control and improved profitability. [10]

2.1.6. Benefits of the web-based medication management system

2.1.6.1. Improve Patient Care

Hospital medication management system can improve patient care by reducing delay because our project gives a faster service; properly implemented Hospital medication management can improve quality of patient care by increasing the timeliness and accuracy of records and administrative information. Hospital medication management provides simplified access to patient records and other administrative information for those who need it. It also sets a standard for patient care, making it more consistent across the board. In many cases, patients can also use the system to better educate themselves about their diseases and conditions as well as treatments and outcomes. [10]

2.1.6.2. Improve Cost Control

Cost control is another benefit; the increased standardization of hospital management information systems improves cost control by improving efficiency and productivity. Personnel costs and inventory levels can be reduced through the avoidance of duplications and repetitions. Hospital medication management system can also help eliminates confusion and delays that are caused by incomplete or missing records. These systems provide a managerial tool for decision making because they offer the functionality of assembling various types of data into usable reports. Data can be collected and categorized so that managers can analyze such information as patient lists, bed occupancy rates and statistics categorized by doctors. [10]

2.1.6.3. Increased Security

Hospital Medication Management provides increased security of patient information; hospital management information systems are secure databases of administrative and patient information that can be easily accessed only by those who need the information. Hospital medication management links departments without compromising the security of the information involved. For example, pathology, radiology and administrative offices each have access to the pieces of a patient's information that are necessary to completing their individual job functions without allowing access to information that is not specifically relevant to that function.

2.2. RELATED LITRATURE

2.2.1. Medication management

Medication management is defined as patient-centered care to optimize safe, effective and appropriate drug therapy. Care is provided through collaboration with patients and their health care teams. [12]

Medication Management System (MMS) provides the benefits of streamlined operations, enhanced administration and control, superior patient care, strict cost control and improved profitability.MMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. [6]

This MMS is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital management system that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

A hospital is a place where patients come up for general diseases and provide facilities like:

- Consultation by Doctors on Diseases
- Diagnosis for diseases
- Providing treatment facility
- Facility for admitting patients (providing beds, nursing, medicines etc)
- Immunization for Patients/Children

Various operational works that are done in a hospital are:

- Recording information about the patients
- Generating bills
- Recording information related to diagnosis given to patients
- Keeping record of the immunization provided to children/patients
- Keeping information about various diseases and medicines available to cure them

These are the various jobs that need to be done in a hospital by the operational staff and Doctors. All these works are done on papers and it is done as follows:

- Information about patients is done by just writing the patients name, age and gender. Whenever the patient comes up his information is stored freshly
- Bills are generated by recording price for each facility provided to patient on a separate sheet and at last they all are summed up
- Diagnosis information to patients is generally recorded on the document, which contains patient information. It is destroyed after some time period to decrease the paper load in the office
- Immunization records of children are maintained in pre-formatted sheets, which are kept in a file
- Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time. [6]

2.2.2. Pharmacists' role in medication management:

As medication remedy experts, pharmacists:

- Assess patients and their medication-related needs and identify actual or potential drug therapy problems
- Formulate and implement care plans to prevent and/or resolve drug therapy problems
- Recommend, adapt or initiate drug therapy where Appropriate
- Monitor, evaluate and document patients' response to therapy
- Collaborate and communicate with other health care providers, in partnership with patients. [5]

2.3. RELATED STUDIES

For the purpose of this review, potential medication errors are defined as medication incidents detected that theoretically may pose a safety concern for the patient.

Numerous studies reported on medication safety issues that represent potential medication errors, these include the co-prescribing of medications that may interact, the prescribing of medications that are not recommended for people 65 years and older, and discrepancies between medical records and patient practices. The events reported on in these categories are not necessarily due to errors; for example, the prescribing of medications that may interact or that are considered inappropriate may be the result of a well-informed decision. The medications may be prescribed when they are the best or only option, or proven not to adversely affect the particular patients. The findings of these studies are summarized in this section. [7]

The prevalence of potentially interacting drug or drug–disease combinations was reported in 27 studies. The most common methods of data collection were reviews of computerized medical records (including prescribing data and administrative database reviews) and recording of pharmacy computerized alerts at the supply stage of the medication process. The prevalence of potential drug–drug or drug–disease interaction ranged between 2.1% in the general population taking 2 or more medications and 78.8% in patients with non-malignant pain syndrome. [7]

Potentially inappropriate prescribing in people aged 65 and older was reported on in 35 studies, including 2 studies conducted in Australia. The most common method of data collection for these studies was review of computerized medical records, including prescribing information and administrative database reviews. Different criteria were used to define inappropriate prescribing; however, most used the medications outlined by Beers et al. in 1997 and revised 2002. The prevalence of potentially inappropriate prescribing ranged between 8.2% of people diagnosed with congestive heart failure admitted to home healthcare nursing services and 67.2% of patients aged 70 or older admitted to an acute medical geriatric hospital unit. In Australia, the reported prevalence ranged between 18.5% (residents of aged-care facilities)to 21.3% (veterans over the age of 70).

In addition, one study reported on the prescribing of drugs deemed inappropriate during pregnancy and one study reported on potentially inappropriate dosing in children.

These studies found 0.6% of pregnant women were prescribed potentially inappropriate medication and 14.5% of children were prescribed potentially inappropriate medication dose. [5]

Discrepancies between medical records and patient practice can be due to many different things, including errors in documentation, supply, administration and transfer of care. Nine studies reported on discrepancies between medication records and patient practice, of which 2 were conducted in Australia. The most common method of data collection for these studies was patient medication reviews. The 2 Australia studies reported 44 medication discrepancies per 100 patients in a population at high risk of medication incidence and 24.6% of prescriptions given to children at discharge from hospital not being filled. In the international studies, discrepancy rates ranged between 14 medication discrepancies per 100 patients discharged from hospital180 to 95.8% of patients having a discrepancy in medical records in a population of people 65 years or older admitted to a geriatric day hospital. [5]

CHAPTER THREE

REQUIREMENT ANALYSIS

3.1. THE CURRENT SYSTEM

3.1.1. Overview of current system

Each functional department's of the hospital are not integrated. It is very difficult to retrieve patients and doctors detail as well as to handle the whole system manually. It is less accurate to keep the data for future reference. Redundancy of data occurs and this may lead to inconsistency and time-consuming patient service. The most common existing activities include:

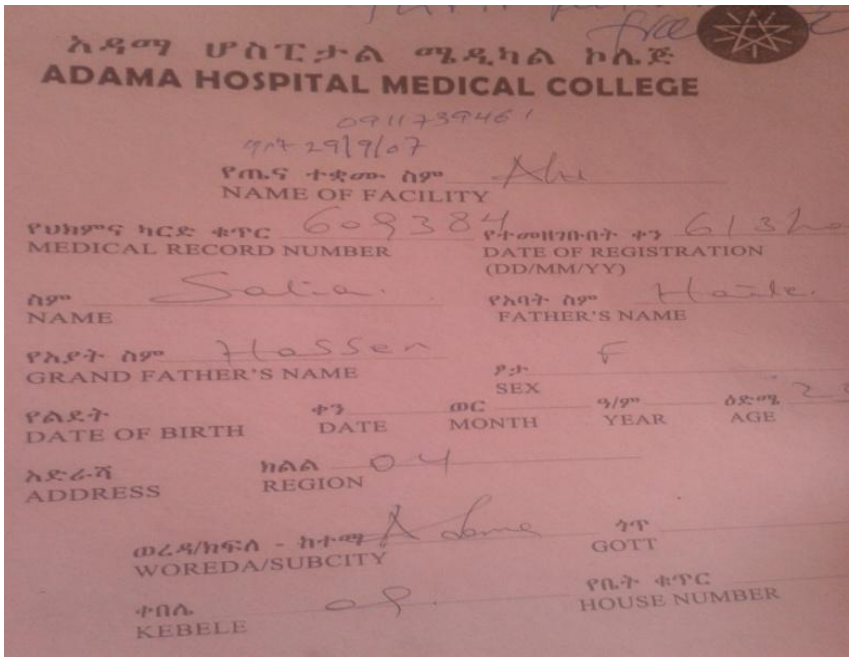


Figure 3.1: Patient registration card

Description of patient registration card: used to register for new patient's in the hospital record and which describes name of the health center, patient's MRN, patient's registration date, patient's name, patient's father's name, patient's grand father's name, patient's gender, patient's date of birth, patient's address like region, wereda, kebelle and house number in the manual system.

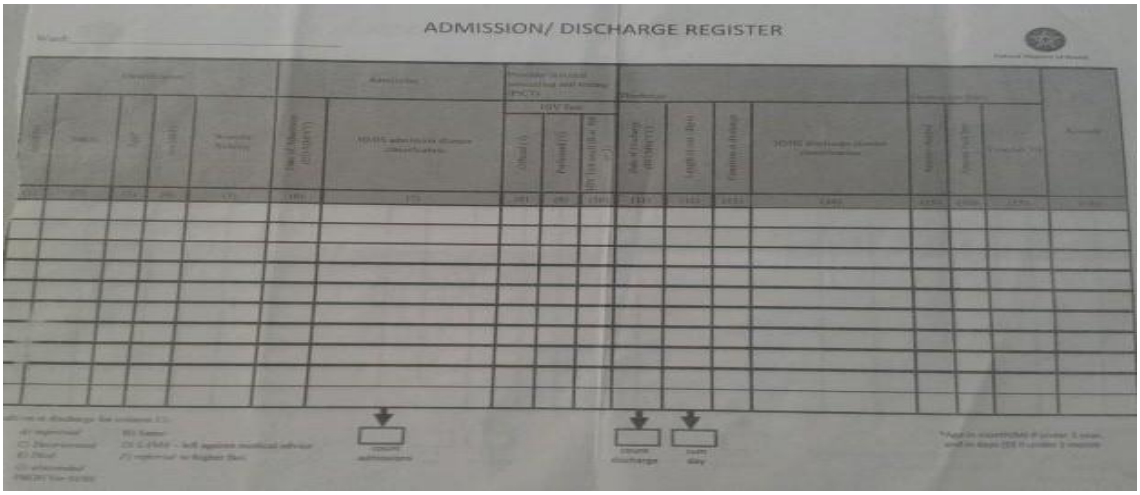


Figure 3.2: Admission discharge register form

Description of admission/discharge registers form: is a patient's registration and discharge form which describes the patient's basic identification elements like MRN, Age, sex, wereda, admission disease classification, HIV test results, patient's date of discharge clearly written by(YYYY/DD/MM), length of stay in the hospital, patient's condition at discharge time in the manual system.

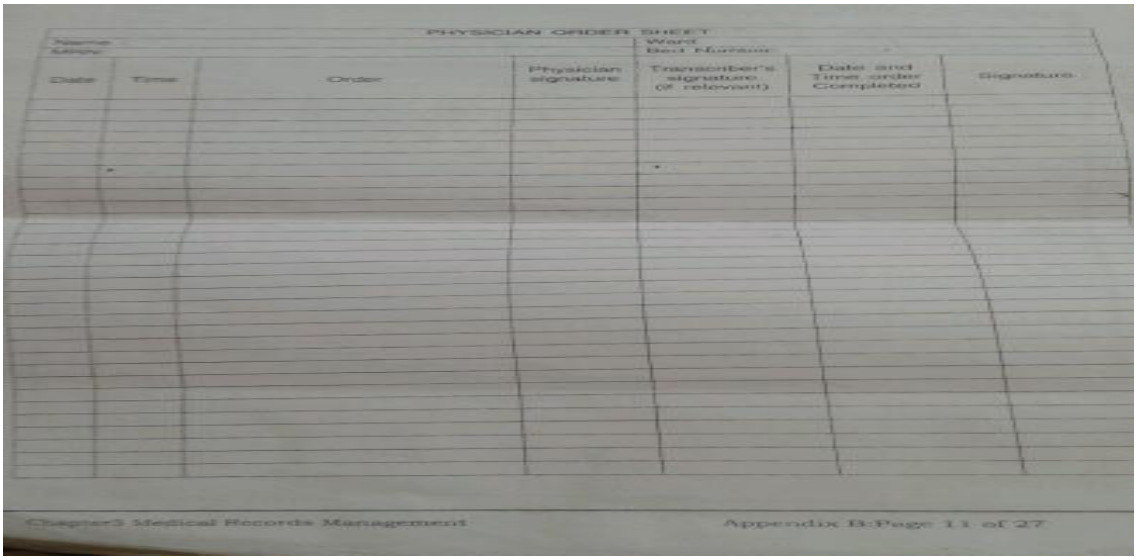
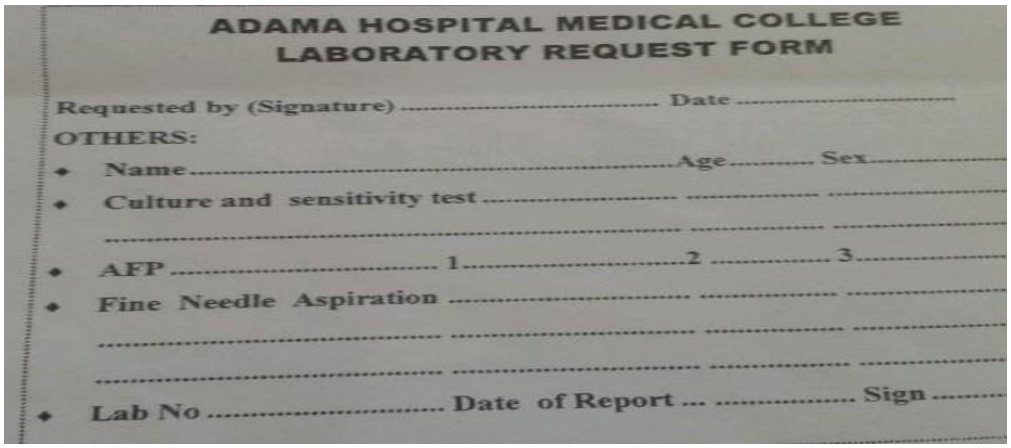


Figure 3.3: Physician order sheet form

Description: physician order sheet describes patient's name, patient's MRN, patient's bed number, ward type, ordered date, order time, physician signature, and date and time order completed in the manual system.

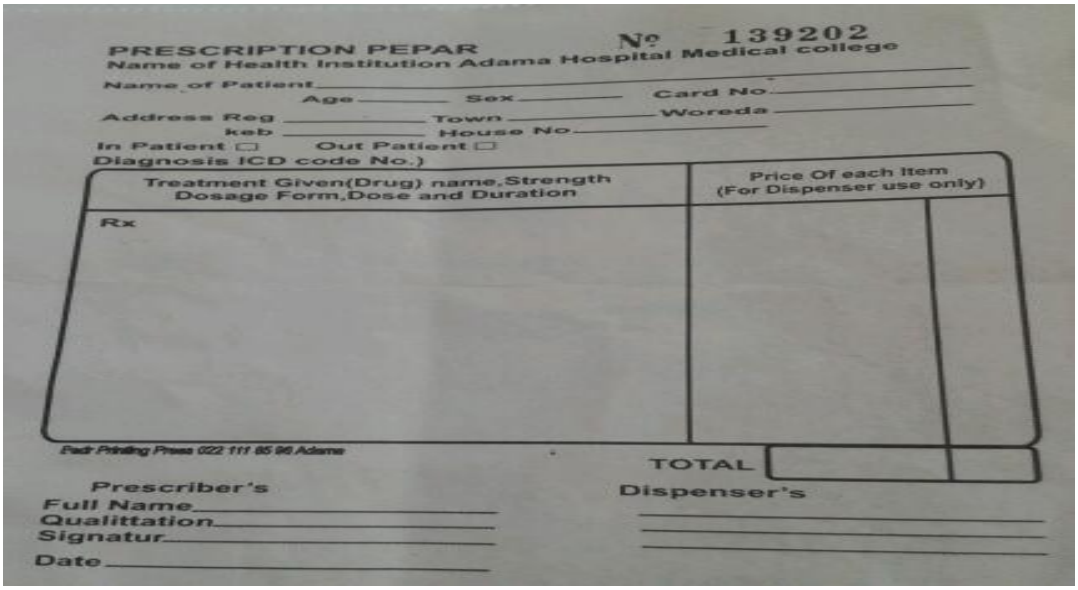


ADAMA HOSPITAL MEDICAL COLLEGE
LABORATORY REQUEST FORM

Requested by (Signature) ..... Date .....
OTHERS:
- Name ..... Age ..... Sex .....
- Culture and sensitivity test .....
- AFP ..... 1 ..... 2 ..... 3 .....
- Fine Needle Aspiration .....
- Lab No ..... Date of Report ... . Sign .....

Figure 3.4: Other lab request form

Description of other lab request form: The above laboratory patient request form contains requested Doctor name, requested date, patient's name, patent's age, sex, patient's culture and sensitivity test, fine aspiration, lab number, date of report and signature in the manual system.



PRESCRIPTION PEPAR № 139202
Name of Health Institution Adama Hospital Medical college
Name of Patient
Age Sex Card No
Address Reg Town Woreda
keb House No
In Patient ☐ Out Patient ☐
Diagnosis ICD code No.)

Treatment Given(Drug) name,Strength Dosage Form,Dose and Duration	Price Of each Item (For Dispenser use only)	
Rx		
	TOTAL	

Prescriber's
Full Name
Qualittation
Signatur
Date

Dispenser's

Figure 3.5: Prescription form

Description prescription form**:** The prescription form is describes all necessary patients' information's like name of the hospital, name of patient's, gender, patient's MRN, patient's address, region, town, wereda, IPD type, OPD type ,medicine quantity, Doctor's full name, Doctor's qualification, signature, date and dispenser's name in the manual system.

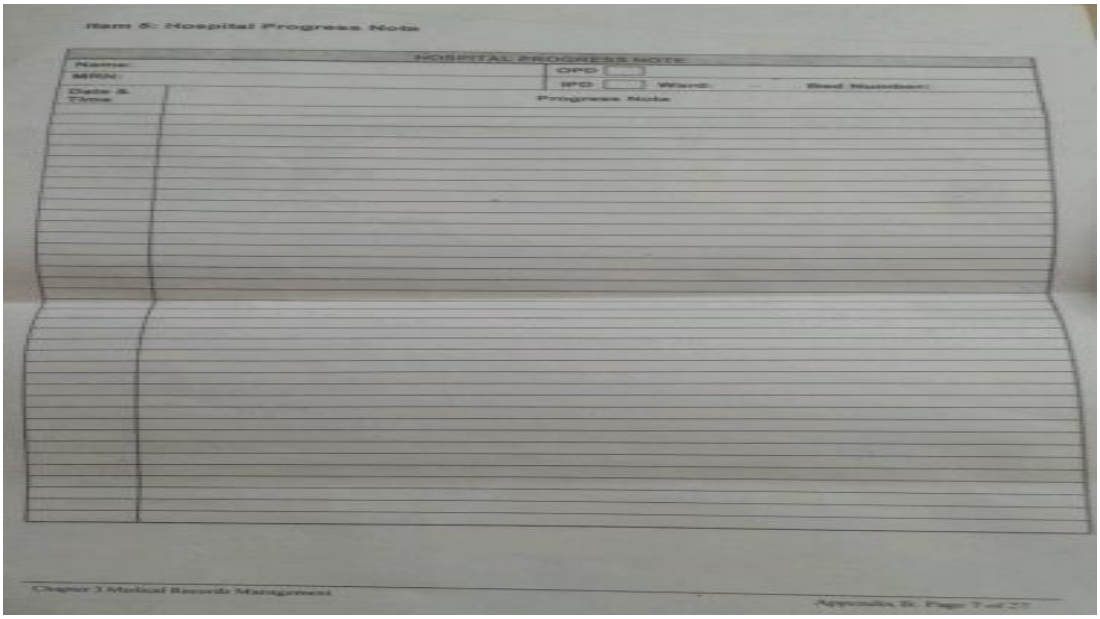


Figure 3.6: Progress note form

Description of patient's progress note: describes the patient's name, MRN, date and time, OPD type, IPD type, ward type and patient's bed number in the manual system.

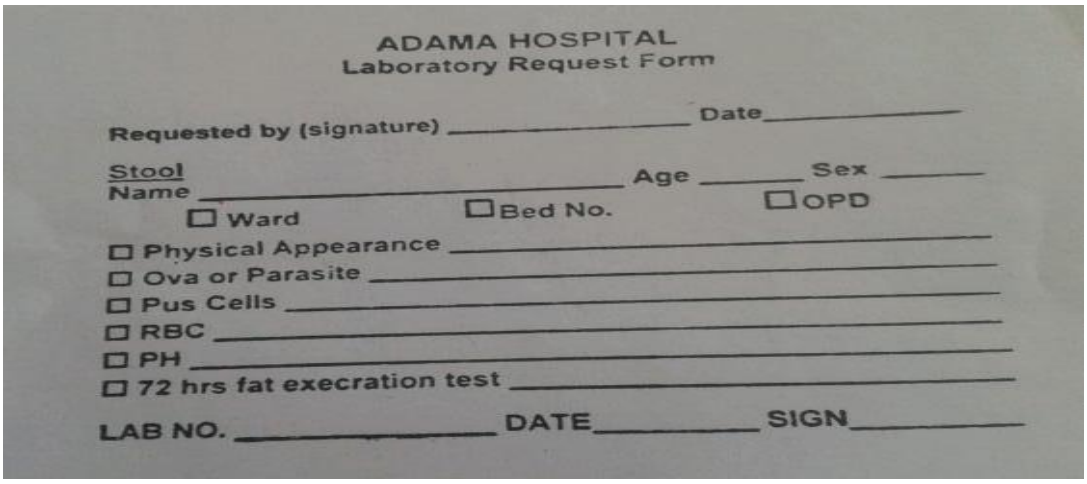
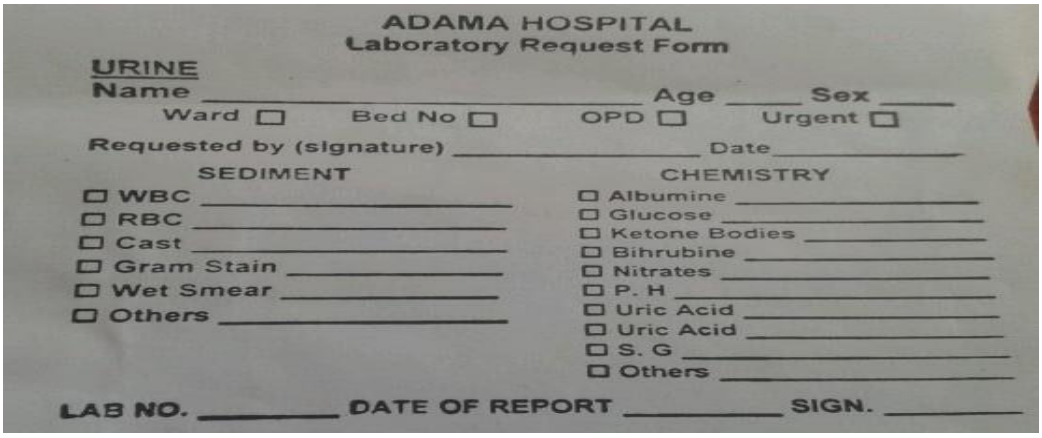


Figure 3.7: Stool lab request form

Description of stool lab request form: this type of lab request describes all necessary patient's personal information and requested Doctor's name, signature, patient's name, age, sex, patient's ward type, patient's bed number, patient's OPD, patient's physical appearance,

patient's ova or parasite, patient's pus cells, patient's RBC, patient's PH, patient's lab number and date of requested in the manual system.



ADAMA HOSPITAL
Laboratory Request Form

URINE
Name _____ Age _____ Sex _____
Ward ☐ Bed No ☐ OPD ☐ Urgent ☐
Requested by (signature) _____ Date _____

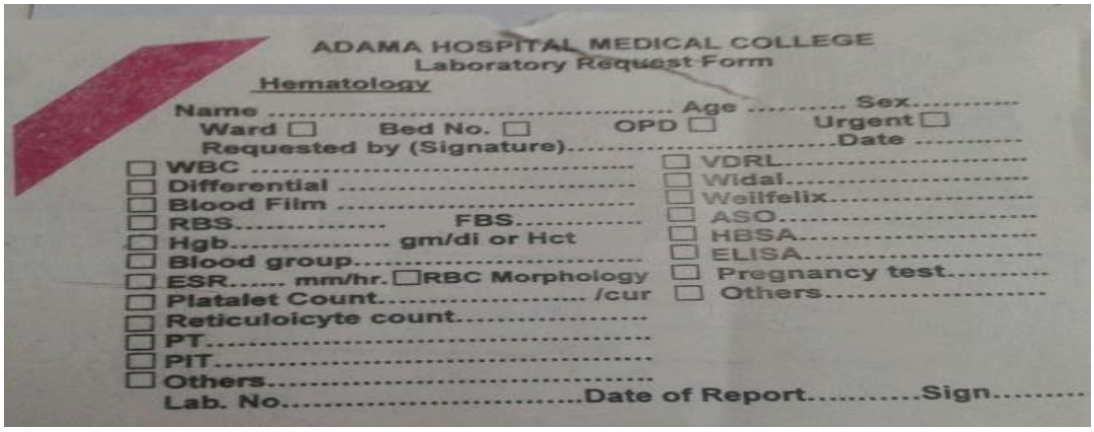
SEDIMENT
☐ WBC _____
☐ RBC _____
☐ Cast _____
☐ Gram Stain _____
☐ Wet Smear _____
☐ Others _____

CHEMISTRY
☐ Albumine _____
☐ Glucose _____
☐ Ketone Bodies _____
☐ Bihrubine _____
☐ Nitrates _____
☐ P. H _____
☐ Uric Acid _____
☐ Uric Acid _____
☐ S. G _____
☐ Others _____

LAB NO. _____ DATE OF REPORT _____ SIGN. _____

Figure 3.8: urine test request form

Description of urine test request form: This patient request form can also describe patient's name, age, sex, ward, bed number, OPD, Urgent, requested Doctor's name, request date, lab number and date of report in the manual system.



ADAMA HOSPITAL MEDICAL COLLEGE
Laboratory Request Form
Hematology

Name ..... Age ..... Sex.....
Ward ☐ Bed No. ☐ OPD ☐ Urgent ☐
Requested by (Signature).....Date .....

☐ WBC .....
☐ Differential .....
☐ Blood Film .....
☐ RBS..... FBS.....
☐ Hgb..... gm/di or Hct
☐ Blood group.....
☐ ESR..... mm/hr. ☐RBC Morphology
☐ Platelet Count..... /cur
☐ Reticuloicyte count.....
☐ PT.....
☐ PIT.....
☐ Others.....

☐ VDRL.....
☐ Widal.....
☐ Weilfelix.....
☐ ASO.....
☐ HBSA.....
☐ ELISA.....
☐ Pregnancy test.....
☐ Others.....

Lab. No.....Date of Report.....Sign.....

Figure 3.9: Hematology request form

Description of hematology lab request form: describes about patient's detail lab request form like patient's name, age, sex, ward, bed number, OPD, urgent, patient's WBC condition, patient's differential condition, patient's blood film, patient's blood group, patient's platelet count, patient's lab number, patient's date of report, requested Doctor name and signature in the manual system.

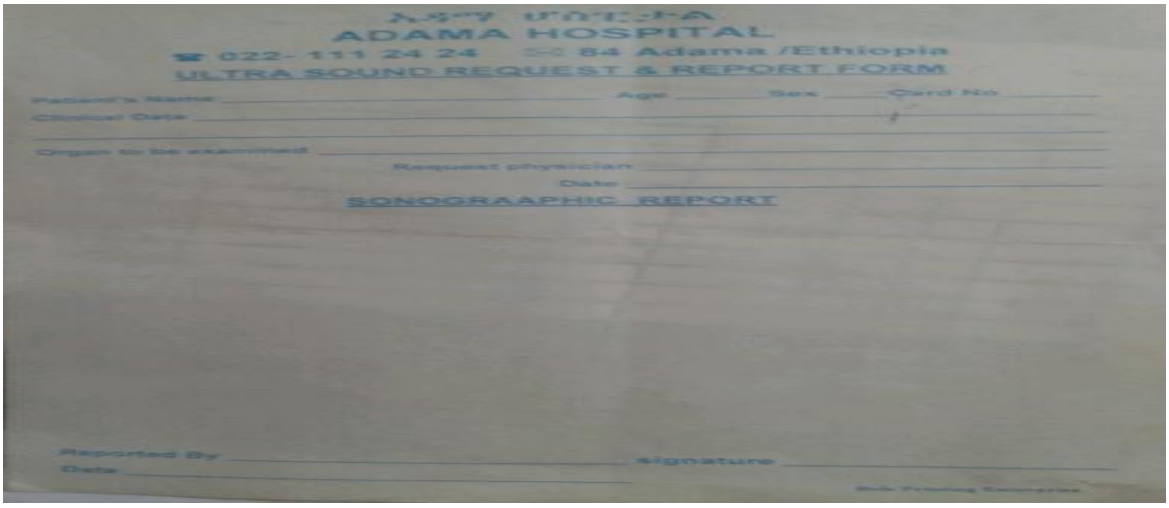


Figure 3.10: Ultra-sound request form

Description of ultra-sound lab request form: This type of lab request form describes the patient's detail like patient's name, age, sex, MRN, clinical data, organs to be needed examined, patient's son-graphic report prescribed by a radiologist, reported radiologist name, radiologist signature as well as requested doctor name and date of requested in the manual system.

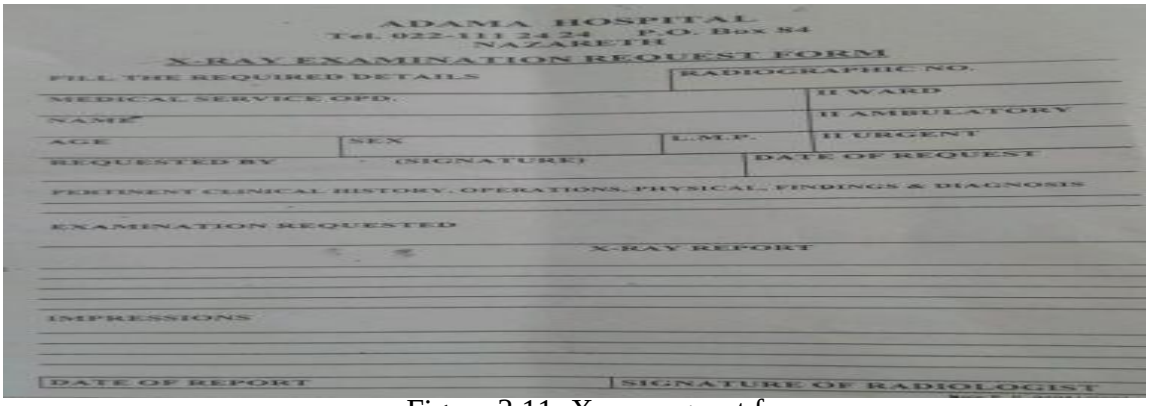


Figure 3.11: X-ray request form

Description of x-ray examination request form**:** This request form describes detail information about patient's radiographic number, patient's name, patient's age, patient's gender, requested doctor name, and doctor's signature, requested date, patient's medical services like ward, ambulatory, urgent, radiologist name and signature, impression, date of report in the manual system.

3.2. THE PROPOSED SYSTEM

3.2.1. Overview of the proposed system

The proposed system has the following advantages:

- It is very easy to operate
- Speed and accuracy are the main advantages
- The system avoids redundancy of data because the database is normalized and the ID for the staffs and patients is auto-increment
- The data are stored in the computer's secondary memories like hard disk, etc. it can be easily receive and used at any time
- The proposed system will easily handle all the data and the work done by the existing systems such that to shift from manual
- The proposed systems eliminate the drawbacks of the existing system to a great extent and it provides tight security to data

3.2.2. Functional requirements

The proposed systems will have the following functional requirements:

- The system will have the capability of authorizing users. Each actor's (administrator, reception, doctor, technician, pharmacist and cashier) logging to the system having an authorized username and password

- The system has the capability to manage staff and user accounts. Staffs will be registered and modify their detail then after user account of each user will be created to enable them access the system

- The system can give service for patients by managing their detail that is registration, updating of patients history

- Appointment management will be also made by the system. After patients are registered, receptionist will add and update an appointment of doctors to patients

- The system will be able to allot room for the IPD patients

- The system has the capability of managing medical examination. Doctors will request any type of medical examination to technicians and test result will be submitted back to doctors for further treatment
- Automated treatment will be also made by the system
- Prescription can add after treatment
- The system can perform drug management. Drugs will be added and updated.
- The system can perform automated payment management. Each patient's payment will be retrieved on the payment page using MRN and the cashier will check for each payment
- The system will generate report of each functionality

3.2.3. Non-functional requirements

Non-functional requirements describe user-visible aspects of the system that are not directly related to the functional behavior of the system. It spans a number of issues, from user interface look and feel to response time requirements to security issues.

1.2.3.1. User Interface and human factors: any number of authorized users who have fundamental knowledge and skill of computer have the privilege of using the system. Those users can easily manage and learn the system. It shows a message for an error occurred in automation of the system .In addition to the peripheral device of a computer system like keyboard, mouse, printer, system unit and monitor, users need also medical devices like x-ray and ultrasound machines.

1.2.3.2. Documentation: every aspect of the system should be well documented in a secondary storage device. The system allows user to back up the stored data to restores in case of failure.

1.2.3.3. Hardware consideration: the proposed system is developed considering hardwires with better processing speed and maximum data storing capacity which can be available in the surroundings area. The hardware requirement should be assessable by the users of the system.

1.2.3.4.Performance characteristics: During past times, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Company system, the automated system is to be undertaken. The web-based project is fully computerized and user friendly even that any of the members can see the report and status of the company at the required time.

1.2.3.5.Error handling and extreme conditions: the system shows error messages whenever error occurs. The error may be invalid inputs and immediately it is displayed in each page of the system as it is invalid input.

1.2.3.6.Quality issues: the system provide correct information to the user, it retrieve correct status of the patients and staffs. A failure may occur once per 5-8 months but it can restart in a fast manner. It is easily portable to any place by taking softcopy of it.

1.2.3.7.System modifications: system is expected to perform any modification at any time by programmers.

1.2.3.8.Physical environment: The applicability of the system should consider the physical environment. It should fit to the actual environment. So our system can't be affected in case of high temperature, humidity, vibration and magnetic data etc.

1.2.3.9.Security issues: Security is the main criteria for the proposed system. Since illegal access may corrupt the database. The system allows only those users who have correct password and user name to log in. Since, the system gives different privileges for each user; unauthorized user cannot alter unpermitted data. That means users can access only data that belongs to it. To do this we have used the MD5 hashing algorithm and an administrator has the privilege to manage users of the system.

3.2.3.10.Resource issues: The proposed system considers cost wise manner. If the resources needed are beyond the capacity of the hospital it will be difficult to implement the system.

3.2.3.11.Robustness is the ability of a computer system to cope with errors during execution. Robustness testing can be used during development testing to better test the GUI and

issues dealing with error detection and correction. Operational testers can use robustness testing concepts to guide more extensive testing of resource usage and stress testing.

3.2.3.12.Maintainability: A characteristic of design and installation, expressed as the probability that an item will be retained in or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources.

3.2.3.13.Privacy: is the ability of an individual or group to seclude themselves, or information about themselves, and thereby express them selectively.

3.3. HARDWARE AND SOFTWARE REQUIREMENTS

3.3.1. Hardware Requirements

Processor:	Intel Pentium IV 2.2 GHZ or above
Clock speed:	500 MHZ
System bus:	32 bits
RAM:	1 GB or above RAM
Hard Disk:	100 GB or higher
Monitor:	SVGA COLOR
Keyboard:	USB 108 keys
Mouse:	USB 2 button mouse

3.3.2. Software Requirements

Front-end:	PHP
Back-end:	MYSQL
Operating system:	window XP and above
Browser:	any browser
IDE:	Notepad++, Macromedia Dreamweaver

3.3.3 Network requirements

Network hardware and software devices like cables, server, network operating system,

3.4. SOFTWARE ENGINEERING METHODS AND TECHNOLOGIES

In software engineering there are two types of system development life cycle (SDLC)

1. Sequential Model
2. Incremental Model

In our project we completed each phase in a sequential way using waterfall model which is one type of sequential model.

Waterfall Model:

- It is also called as linear sequential model.
- In this model whole application is developed in a sequential approach.
- In this model each phase must be completed fully before the next phase begin.
- Provides structure to inexperienced staff.

Where to use the waterfall model

- Requirements are very well known.
- Product definition is stable.
- Technology is understood.
- New version of an existing product.

Features of Water Fall Model

1) A Water Fall Model is easy to flow
2) It can be implemented for any size of project
3) Every stage has to be done separately at the right time so you cannot jump stages
4) Documentation is produced at every stage of a waterfall model allowing people to understand what has been done
5) Testing is done at every stage

Waterfall Model Diagram

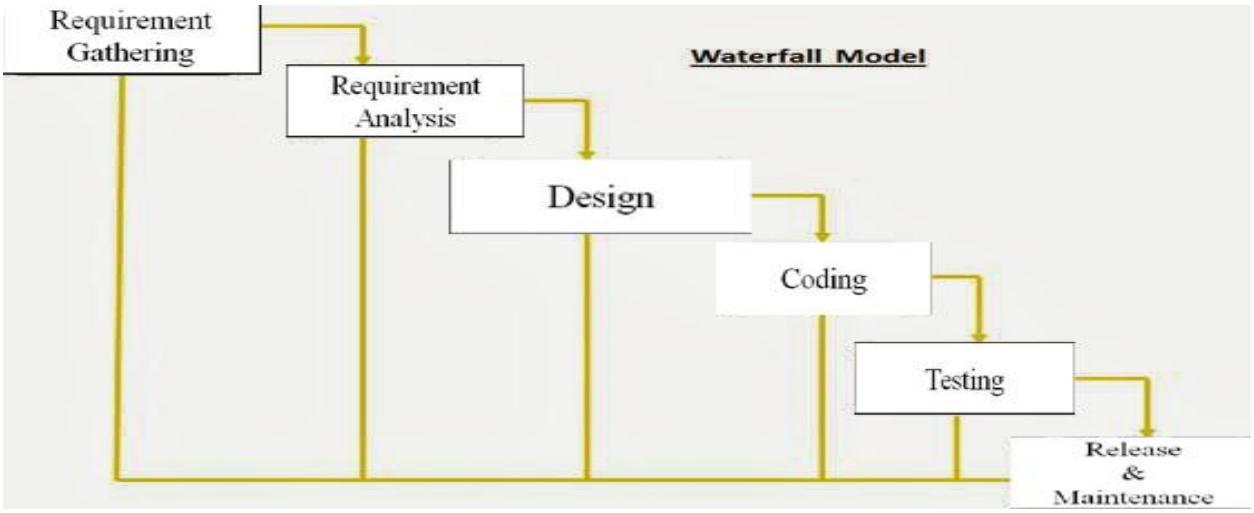


Figure 3.12: Waterfall Model

Phases of Waterfall Model:

Waterfall model has five different phases, which are following:-

1) Requirement gathering and Analysis

2) Design

3) Coding

4) Testing

5) Maintenance

3.5. ANALYSIS MODELS

UML Diagrams:

UML is a notation that resulted from the unification of OMT (Object Modeling Technique); UML has also been influenced by other object-oriented notations. UML has been designed for a broad range of applications.

- The functional model, represented in UML with use case diagrams, describes the functionality of the system from the user's point of view.
- The object model, represented in UML with class diagrams, describes the structure of a system in terms of objects, attributes, associations, and operations.

The dynamic model, represented in UML with sequence diagrams, state chart diagrams, and activity diagrams, describes the internal behavior of the system.

3.5.1. Use case model

A use-case is a model of how users interact with the system to solve a problem, and it describe the goal of the user, the interaction between user and the system and the required behavior of the system in satisfying the goal. A use case model consists of a number of model elements, the most important model elements are use-case, actors and relationship between them.

Actor Specification:

An actor represents anything that needs to interact with the system to exchange information. An actor is a user, a role, which could be an external system as well as a person. An actor initiates the system activity, a use case, for the purpose of completing the some business task. The use-case actors are listed below:

1. Administrator
2. Receptionist
3. Doctor
4. Technician
5. Pharmacist
6. cashier

Use-case diagram used to describe a set of actions on the system and show us the interaction between external users and internal of the system their privilege is varies.

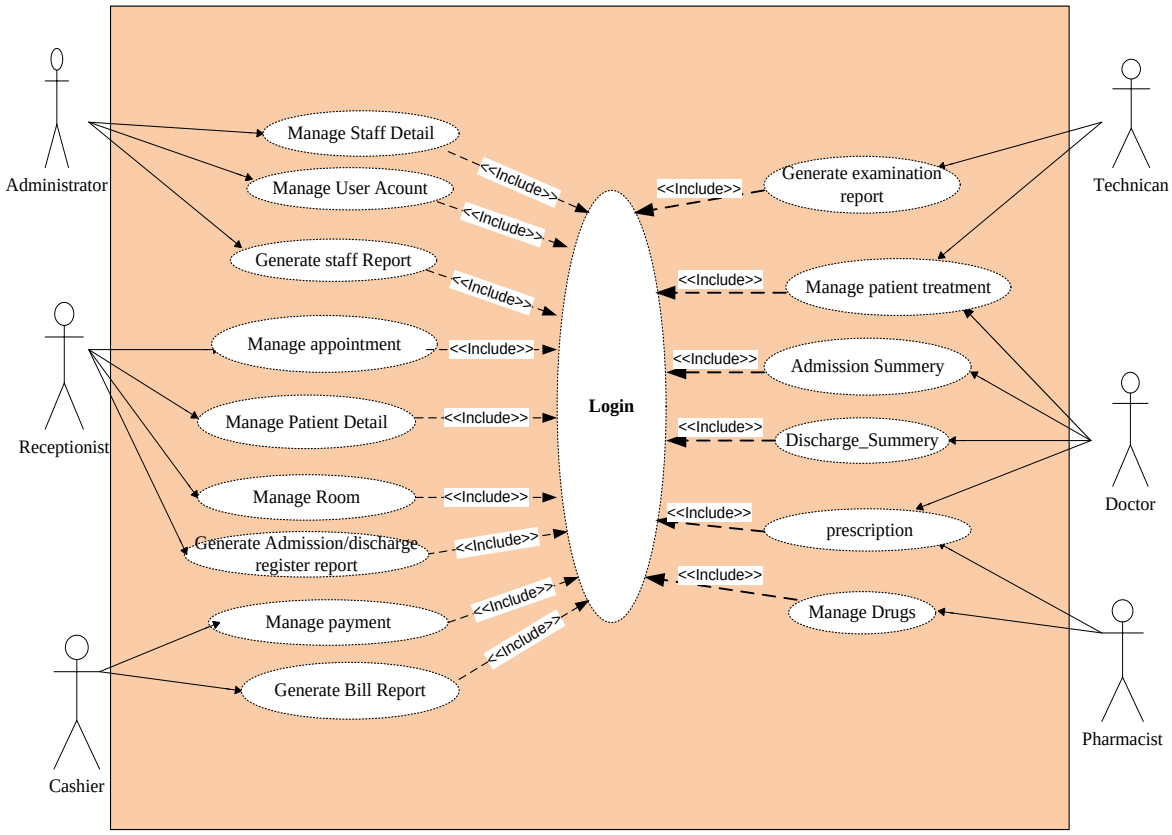


Figure 3.13: Use case diagram

Use case Description:

Table 3.1: Login use case

Use case name	Login
Participating Actors	Administrator, reception, Doctor, Technician, pharmacist and cashier
Description	It is initiated when the actors of the system wants to login to the system.
Pre-condition	The actor must have an account
Flow of events:	1. actor browse the systems page 2. System displays login window 3. actor fills user name and password 4. System authenticate the actor 5. System displays the entire page of the actor
Post condition	Actors logon to their pages

Table 3.2: Manage staff use case

Use case name	Manage staff
Participating Actor	Administrator
Description	It is initiated when the administrator wants to register, modify and delete staffs in the system.
Pre-condition	Staff name, age, sex, address, designation and specialization
Flow of events:	1. Administrator clicks staff link and then click register user account button 2. System displays user registration form 3. Administrator fills all user necessary data 4. Administrator submit registration 5. Administrator view list of registered accounts on the system 6. Administrator click modify to user account 7. System displays form to modify 8. Administrator modify users successfully 9. Administrator click Delete button to delete users accounts 10. System displays form to delete user accounts 11. Administrator delete user accounts successfully 12. Use case ends
Post condition	registered staff (have ID)

Table 3.3: Manage user account use case

Use case name	Manage user account
Participating Actor	Administrator
Description	It is initiated when the administrator of the system wants to create and modify user accounts of staffs in the system.
Pre-condition	Staff ID, username, password and role
Flow of events:	1. Administrator clicks on the manage user account link 2. Administrator click on create user account button 3. Administrator fills his/her user name and password 4. System authenticates the user 5. System displays the entire application window 6. Use case ends
Post condition	User account created, user get username and password

Table 3.4: Manage patient detail use case

Use case name	Manage patient detail
Participating Actors	Receptionist
Description	It is initiated when receptionist register, admit and discharges patients detail in the system.
Pre-condition	Patient's name, age, sex and address
Flow of events:	1. Actors click patient link 2. System displays patient page 3. Receptionist click the add button to register new patient 4. System registered patient to the database 5. Reception can view, modify and Delete patients 6. Use case ends
Post condition	Registered Patient (gets MRN)

Table 3.5: Manage appointment use case

Use case name	Manage Appointment
Participating Actors	Receptionist, Doctor
Description	It is initiated when receptionist want to appoint doctor to a patient
Pre-condition	MRN, Date, time, doctor ID
Flow of events:	1. Receptionist clicks appointment link 2. System displays appointment page 3. Receptionist clicks the add button and fills data 4. System validate the data and store in database 5. Receptionist clicks modify button to update appointment 6. System displays update appointment form 7. Doctor click appointment link 8. Doctor view list of appointed patient 9. Use case ends
Post condition	Appointed doctor

Table 3.6: Manage room use case

Use case name	Manage Room
Participating Actor	Receptionist
Description	It is initiated when receptionist want to allot room
Pre-condition	Room No, department, No. of beds
Flow of events:	1. Receptionist click room link 2. System displays room page 3. Receptionist checks the availability of beds (if bed is available) 4. Click update button and assign patient to the provided bed 5. System store the number of updated rooms in the database 6. Use case ends
Post condition	Allotted room to patient

Table 3.7: Manage Patient Treatment use case

Use case name	Manage Patient Treatment
Participating Actors	Doctor, Technician
Description	It is initiated when the Doctor requests medical examination and Technician view the request and submit the result of a patient
Pre-condition	All lab request fields
Flow of events:	1. Doctor clicks patient treatment link 2. System displays Patient treatment page 3. Doctor clicks the necessary button 4. Display page 5. Doctor ticks (✓) the required check box and Click order button 6. Technician view doctor's request 7. Technician perform the examination 8. Technician fill the fields and click submit button 9. Use case ends
Post condition	patient lab result

Table 3.8: Admission summary use case

Use case name	Admission summary
Participating Actor	Doctor
Description	It is initiated when the Doctor want to add and modify the treatment
Pre-condition	Patient name, admission date, ward, disease classification
Flow of events:	1. Doctor clicks admission summary link 2. System displays admission summary page 3. click add button 4. fill the fields and submit treatment details 5. clicks modify button to update treatment details 6. Use case ends
Post condition	Admitted patient

Table 3.9: Manage prescription use case

Use case name	Manage prescription
Participating Actors	Doctor, pharmacist
Description	It is initiated when the Doctor writes the prescription detail of the patient based on the given treatment and test result and pharmacist views prescribed medicine
Pre-condition	Patient name, drug name, dose and duration
Flow of events:	1. Doctor clicks prescription link 2. System displays the prescription page 3. Doctor fills necessary medicine and click submit button 4. Pharmacist view prescribed medicine 5. Pharmacist check the availability of medicine in the database 6. If available fill price of the prescribed medicine 7. Use case ends
Post condition	Patient having prescribed medicine

Table 3.10: Manage drugs use case

Use case name	Manage drugs
Participating Actor	Pharmacist
Description	it is initiated when the pharmacist manages the availability of drugs
Pre-condition	Drug ID, drug name, quantity and price
Flow of events:	1. Pharmacist click the drug link 2. System displays drug page 3. Pharmacist click view button to see list of available medicine 4. Pharmacist click add button to add new drugs 5. Pharmacist click modify button to update drugs 6. Use Case ends
Post condition	List of drugs

Table 3.11: Manage payment use case

Use case name	Manage payment
Participating Actor	Cashier
Description	It is activated when a cashier want to register all patient bills
Pre-condition	Patient MRN, ordered charges/fees
Flow of events:	1. Cashier clicks payment link 2. System displays the payment page 3. Cashier enter the MRN of the patient 4. System displays all charges/fees of the patient 5. Cashier ticks(✓) necessary charges/check boxes 6. Casher click the total button and collect amount 7. Cashier click print button and give the receipt to the patient 8. Use case ends
Post condition	printed receipt

Table 3.12: Generate report use case

Use case name	Generate report
Participating Actors	Administrator, reception, Doctor, Technician, pharmacist and cashier
Description	It is activated when report is necessary
Pre-condition	Organized document
Flow of events:	1. Actor open the page to be printed 2. Actor click print button 3. Use case ends
Post condition	Printed document

3.5.2. Object Model

3.5.2.1. Data dictionary

Table 3.13 Patient registration

Data (column name)	DataType	Key (allow or null)	Description
MRN	Varchar(20)	Primary	Medical record number.
FirstName	Varchar(30)	Not null	Patient's first name
LastName	Varchar(30)	Not null	Patient's Last name
GFName	Varchar(30)	Not null	patient's Garand father's name
Age	int(10)	Not null	Patient's age
Sex	Varchar(10)	Not null	Patient's Gender
DOB	Datetime	Not null	Patient's date of birth
Region	Varchar(30)	Not null	Patient's region and state.
Wereda	Varchar(30)	Not null	Patient's address and wereda.
Kebele/sub-city	Varchar(30)	Not null	Patient's kebele and sub-city.
HouseNo	Varchar(20)	Not null	Patient's house number.
PhoneNumber	Varchar(30)	Not null	Patient's phone number.
Card_Fee	Float	Not null	Patient's medical payment
Registration_Date	Datetime	Not null	Patient's admission date.

Table 3.14 Appointment

Data (column name)	Data Type	Key (Allow Nulls)	Description
Sno	Int (15)		Patient's serial number.
Date	Date	Not null	Patient registration date.
Time	Time	Not null	Patient's registration time.
MRN	Varchar(15)	Not null	Medical record number
DoctorID	Varchar(15)	Not null	Doctors ID
Designation	Varchar(50)	Not null	Position of the respective Doctors, nurses.

Table 3.15 Drugs

Data (column name)	Data Type	Key (Allow Nulls)	Description
Drug_ID	varchar (30)	Foreign key	Prescribed medicine identity
Drug_Nmae	Varchar(50)	Not null	Name of medicine
Qty_available	Int(25)	Not null	Quantity of medicine
Price	Float	Not null	Medicine price
Mfg_company	Varchar(30)	Not null	Medicine production company

Table 3.16 Admission discharge register

Data (column name)	DataType	Key (allow null)	Description
SNo	Int (25)		Serial numbers
MRN	Int (25)	Not null	patient's medical record number
Age	Int(25)	Not null	patient's age
Sex	Varchar(15)	Not null	Patient's gender.
Wereda	Varchar(30)	Not null	Patient's specific wereda
DOA	Date	Not null	Patient's date of admission
Adm_Disaese and classification	Varchar(50)	Not null	Admission and disease classification.
HIV_Offered	Varchar(10)	Not null	HIV offered or not
HIV_performed	Varchar(10)	Not null	Patient's HIV condition
HIV_Test_Result	Varchar(15)	Not null	Patient's HIV test result
DOD	Date	Not null	Date of discharge from the hospital.
Length_of_stay	Int(30)	Not null	Patient's waiting as IPD in the hospital
Condition_at_discharge	Varchar(30)	Not null	Patient's condition during discharge.
Discharge_disclassification	Varchar(50)	Not null	Classificationeither OPD or IPD
Amount_charged	Float	Not null	Medication payment
Amount_paid/Fee	Float	Not null	Patient's total amount to paid for his/her service
Voucher_No	Int(30)	Not null	Patient's payment receipt number
Remark	Varchar(100)	Not null	Describe comments of the activity

Table 3.17 Discharge summary

Data (column name)	Data Type	Key (Allow nulls)	Description
MRN	Int (15)	Foreign key	Patient's medical record number
DOA	Date	Not null	Date of Admission
DOD	Time	Not null	Date of birth
Hospital summary	Varchar(200)	Not null	Patient's detail information
Diagnosis	Varchar(200)	Not null	Prescription and identified type of illness
Diagnostic proc and Lab	Varchar(200)	Not null	Patient's diagnostic procedures &lab findings
Diet	Varchar(100)	Not null	Patient's diet
Activity	Varchar(100)	Not null	Doctor's treatment activity
Specific careNeed	Varchar(200)	Not null	Patient's specific care
Drugs	Varchar(100)	Not null	Patient's dose and durations
Condition_ at Discharge	Varchar(100)	Not null	Patient's health conditions during discharge
Discharge_Disease_Classification	Archar(200)	Not null	Discharge disease classification
App_Date	Date	Not null	Doctors appointment date
Place	Varchar(50)	Not null	address
Seen_by	Varchar(50)	Not null	patients treated by doctor
Date	Date	Not null	registration date and discharge

Table 3.18 Hematology

Data (column names)	DataType	Key (allows null)	Description
SNO	Int (20)	Primary key	serial number in the system.
MRN	Int(25)	Not null	patient's medical record number
Requested_by	Varchar(50)	Not null	Doctor request
Requested_Date	Date	Not null	Request date
WBC	Int (20)	Not null	White blood cells
Differential	Varchar(25)	Not null	Patients differential
Blood_film	Varchar(30)	Not null	Patient'sBlood film.
RBC	Int (20)	Not null	Red blood cells
FBS	Varchar(30)	Not null	Fasting blood sugar.
Hgb	Varchar(30)	Not null	Hemoglobin
Blood_Group	Varchar(30)	Not null	patient's blood group
ESR	Varchar(30)	Not null	Erythrocyte sedimentation rate
RBC_Morphology	Varchar(30)	Not null	red blood cells morphology
Platlate_count	Varchar(30)	Not null	Patient's plate late count
Reticuloicyte_count	Varchar(30)	Not null	Patient's reticulocyte count
PT	Varchar(50)	Not null	Patient's prothrombin time
PIT	Varchar(30)	Not null	Patient's Pitocin
VDRL	Varchar(30)	Not null	Patient's blood test for syphilis
Widal	Varchar(30)	Not null	Patient's widal activity
Weilfelix	Varchar(30)	Not null	Patient's weilfelix
ASO	Varchar(30)	Not null	Patient's antistreptolysin O
HBSA	Varchar(30)	Not null	Patient's HBSA
ELISA	Varchar(30)	Not null	Enzyme-linked immunosorbent assay
Pregnancy_Test	Varchar(30)	Not null	Patient's antistreptolysin O
HCV	Varchar(30)	Not null	hypertensive cardiovascular

RF	Varchar(30)	Not null	rheumatic fever
Others	Varchar(30)	Not null	other hematology results
Hematology_Fee	Float	Not null	hematology service charges
Lab_No	Varchar(30)	Not null	lab hematology result
Report_Date	Date	Not null	hematology results report

Table 3.19 Physician order sheet

Data (columns names)	DataType	Key (allow null)	Description
MRN	int(25)	Foreign key	Medical record number
Date	Date	Not null	Date
Time	Time	Not null	Time
Order	Varchar(100)	Not null	Patient's physician order type
Order_completed_Date	Date	Not null	patient's physician order completed date
DoctorID	Varchar(25)	Not null	Doctor's ID

Table 3.20 Other lab forms

Data (column names)	DataType	Key (allows null)	Description
SNo	int(30)	Primary key	Serialnumber
MRN	Int(30)	Not null	Medical record number
Requested_by	Varchar(50)	Not null	Doctor requested
Requested_Date	Date	Not null	Doctor requested date
Culture_ Sensitivity_Test	Varchar(100)	Not null	culture sensitivity test
AFP1	Varchar(50)	Not null	Alphafetoprotein1
AFP2	Varchar(50)	Not null	Alpha fetoprotein2
AFP3	Varchar(50)	Not null	Alpha fetoprotein3
AFP4	Varchar(50)	Not null	Alpha fetoprotein4
Fine_Nidle_	Varchar(100)	Not null	Fine needle aspiration

Aspiration			
Other_Lab_Fee	Float	Not null	charges fee
Lab_No	Varchar(50)	Not null	lab number
Report_Date	Date	Not null	Report date

Table 3.21 Prescription

Data (columns names)	DataType	Key (allow null)	Description
MRN	int(25)	Foreign key	Medical record number
Drug_Name	Varchar(50)	Not null	Drug name
Qty	Int(30)	Not null	Medicine quantity
Dose_and_ Duration	Datetime	Not null	Dosage of the medicine
Doctor_ID	Varchar(25)	Not null	Doctors ID
Qualification	Varchar(50)	Not null	Doctor's qualification
Date	Date	Not null	Prescription date
Drug_Fee	Float	Not null	Medication fee
Pharmacist_ID	Varchar(30)	Not null	pharmacist ID

Table 3.22 payment

Data (columns names)	DataType	Key (allow null)	Description
BillNo	int(30)	Primary key	Payment bill number.
Date	Date	Not null	Billing date
MRN	Varchar(25)	Not null	Medical record number
Card_Fee	Float	Not null	Card fee
Stool_Fee	Float	Not null	stool fee
Hematology_Fee	Float	Not null	Hematology fee
Urine_Fee	Float	Not null	Urine fee
X_Ray_Fee	Float	Not null	X-ray fee
Ultra_Sound_Fee	Float	Not null	ultra-sound fee
Treatment_Fee	Float	Not null	Treatment fee

Room_Fee	Float	Not null	Room fee for IPD
Drug_Fee	Float	Not null	Drug fee
Other_Lab_Fee	Float	Not null	Other lab fee
Sum	Float	Not null	Total payment of medication
Cashier ID	Varchar(30)	Not null	cashier's identity number

Table 3.23 staff

Data (columns names)	DataType	Key (Allow null)	Description
SID	varchar(15)	Primary k	Staff ID
FirstName	Varchar(25)	Not null	FirstName
LastName	Varchar(15)	Not null	LastName
Age	Int(10)	Not null	patient's age
Sex	Varchar(10)	Not null	Patient's gender.
Address	Varchar(25)	Not null	Patient's address
PhoneNumber	Vaechar(25)	Not null	PhoneNumber
E-mail	Varchar(50)	Not null	E-mail address
Qualification	Varchar(50)	Not null	Staff qualification
Designation	Varchar(50)	Not null	Staff specified department
Specialization	Varchar(50)	Not null	Specialization
JoinDate	Date	Not null	Staff join-date

Table 3.24 Progress note

Data (columns name)	DataType	Key (allow null)	Description
MRN	int(25)	Foreign key	Medical record number
Patient_Type	Varchar(20)	Not null	Patient type IPD or OPD
Ward	Varchar(50)	Not null	Patient's ward room
Bed_No	Varchar(30)	Not null	Patient's bed number
Date_and_Time	Datetime	Not null	examination date and time
Progress_Note	Varchar(200)	Not null	Patient's progress note
Treatment_Fee	Float	Not null	Treatment fee
Room_Fee	Float	Not null	Patient's room fee

Table 3.25 urine

Data (columns names)	DataType	Key (allow null)	Description
SNo	int(25)	Primary k	Serial number
MRN	Int(30)	Not null	Medical record number
Urgent	Varchar(10)	Not null	Patient type urgent, IPD,OPD
Requested_by	Varchar(50)	Not null	Doctor's requested
Requested_Date	Date	Not null	Doctor requested date
WBC	Varchar(30)	Not null	White blood cells
RBC	Varchar(30)	Not null	Red blood cells
Cast	Varchar(30)	Not null	cast and urine test result
Gram_stain	Varchar(30)	Not null	gram stain urine lab result
Wet_Smear	Varchar(30)	Not null	wet_smear urine result
Albumine	Varchar(30)	Not null	albmine and urine test result
Glucose	Varchar(30)	Not null	Glucose urine test result
Ketone_Bodies	Varchar(30)	Not null	Patient's ketone_results
Bihrubine	Varchar(30)	Not null	bihrubine urine test result
Nitrates	Varchar(30)	Not null	nitrates urine test result
PH	Varchar(30)	Not null	Patient's past history

Uric_Acid	Varchar(30)	Not null	uric_acid urine test result
SG	Varchar(30)	Not null	Swan Ganz urine test result
Others	Varchar(30)	Not null	others urine test result
Urine_Fee	Float	Not null	Urine test fee
Lab_No	Varchar(30)	Not null	urine test result Lab_No
Report Date	Date	Not null	urine test result report date

Table 3.26 user

Data (columnnames)	DataType	Key (allow null)	Description
SNo	int(10)	Primary key	Serial number
UserName	Varchar(25)	Not null	username
Password	Varchar(25)	Not null	password
SID	Varchar(25)	Not null	Staff ID

Table 3.27 ultra sound

Data (columns name)	DataType	Key (allow null)	Description
SNo	int(25)	Primary key	Serial number
MRN	Int(30)	Not null	Medical record number
Clinical_Data	Varchar(100)	Not null	Clinical data of an ultrasound
Organ_to_be_Examine	Varchar(50)	Not null	Organs to be examination
Requested_by	Varchar(50)	Not null	Doctors requested
Requested_Date	Date	Not null	Requested date by a Doctor
Sonographic_Report	Varchar(200)	Not null	sonographic report
UltraSound_Fee	Float	Not null	Ultra-sound bill payment.
Reported_by	Varchar(50)	Not null	Examination report date
Designation	Varchar(50)	Not null	Doctors' position and qualification
Report_Date	Date	Not null	Patient's report

Table 3.28 X-ray

Field(columns name)	DataType	Key (allownull)	Description
SNo	int(20)	Primary k	Serial number
MRN	Int(30)	Not null	Medical record number
Radiographic_No	Varchar(30)	Not null	radiographic number
Ambulatory	Varchar(30)	Not null	Patient's ambulatory
Urgent	Varchar(50)	Not null	Urgent
Requested_by	Varchar(50)	Not null	Requested Doctor
Requested_Date	Date	Not null	Patient's requested date
Clinical_Summary	Varchar(100)	Not null	clinical summary progress
Examination_Requested	Varchar(50)	Not null	Examination requested Doctor
X_Ray_Report	Varchar(200)	Not null	x-ray report
Impressions	Varchar(100)	Not null	Patient's degree of pain
X_Ray_Fee	Float	Not null	x-ray payments
Report_Date	Date	Not null	Patient's report date
Staff_ID	Varchar(20)	Not null	Staff ID

Table 3.29 Lab-stool

Data (columns names)	DataType	Key (allow null)	Description
SNO	Int (20)	Primary k	Patient's serial number
MRN	Int(25)	Not null	Medical record number
Urgent	Varchar(25)	Not null	Patient's health status
Requested_by	Varchar(50)	Not null	Requested Doctor
Requested_Date	Date	Not null	Requested date
Physical_app	Varchar(100)	Not null	patient's physical appearance
Ova_or_parasite	Varchar(50)	Not null	Patient's identified disease
Pus_cells	Varchar(50)	Not null	Patient's pus cells
RBC	Varchar(50)	Not null	Red blood cells

PH	Double	Not null	Past history
Fat_Test	Varchar(50)	Not null	Amount of fat tested
Stool_Fee	Float	Not null	Patient's stool fee
Lab_No	Varchar(30)	Not null	Patient's lab number
Seen_by	Varchar(50)	Not null	Show which Doctor is treated
Date	Date	Not null	Lab stool date

Table 3.30 Room

Data (columns names)	DataType	Key (allow null)	Description
SNo	Int(30)	Foreign k	Patient's serial number
Designation	Varchar(50)	Not null	Position of members
No_of_Rooms	Int(25)	Not null	Number of rooms
No_of_Beds	Int(25)	Not null	Number of available beds
Occupied_Beds	Int(25)	Not null	Number of occupied beds
Free_Beds	Int(25)	Not null	Free number of available beds

Table 3.31 Admission summary

Data (columnnames)	DataType	Key (allow null)	Description
MRN	Int(30)	Foreign key	Medical record number
DOA	Date	Not null	Date of admission
Adm_Disease	Int(25)	Not null	Patient's admission disease either IPD or OPD

3.5.2.2. Class Diagram

Class diagrams used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation.

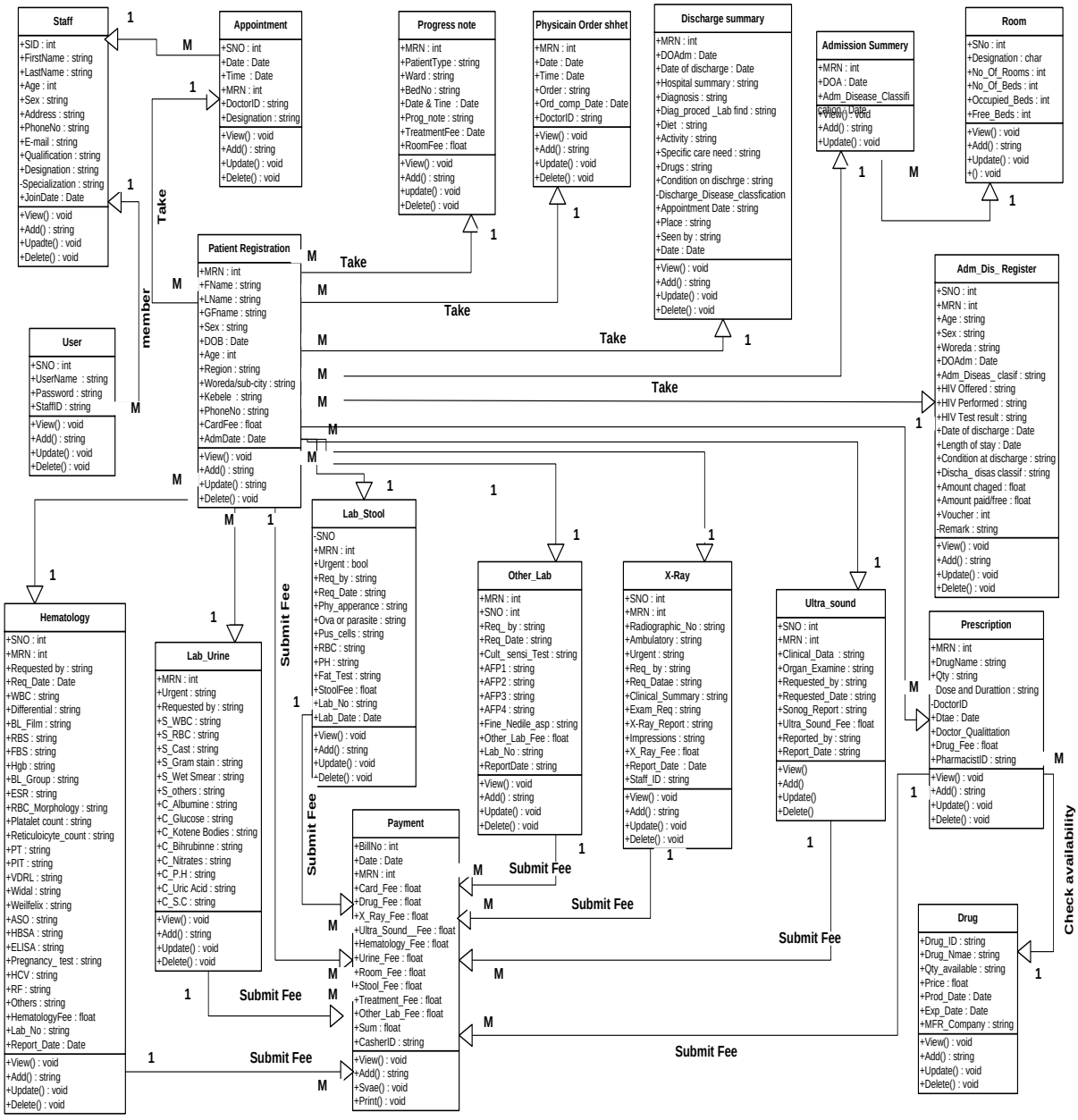


Figure 3.14: Class diagram

3.5.3. Dynamic Model

3.5.3.1 Sequence diagrams

Sequence diagrams are used to formalize the behavior of the system and to visualize the communication among objects. They are useful for identifying additional objects that Participating in the use cases. We call objects involved in a use case participating objects. A sequence diagram represents the interactions that take place among these objects.

3.5.3.1.1 Sequence Diagram for manage staff

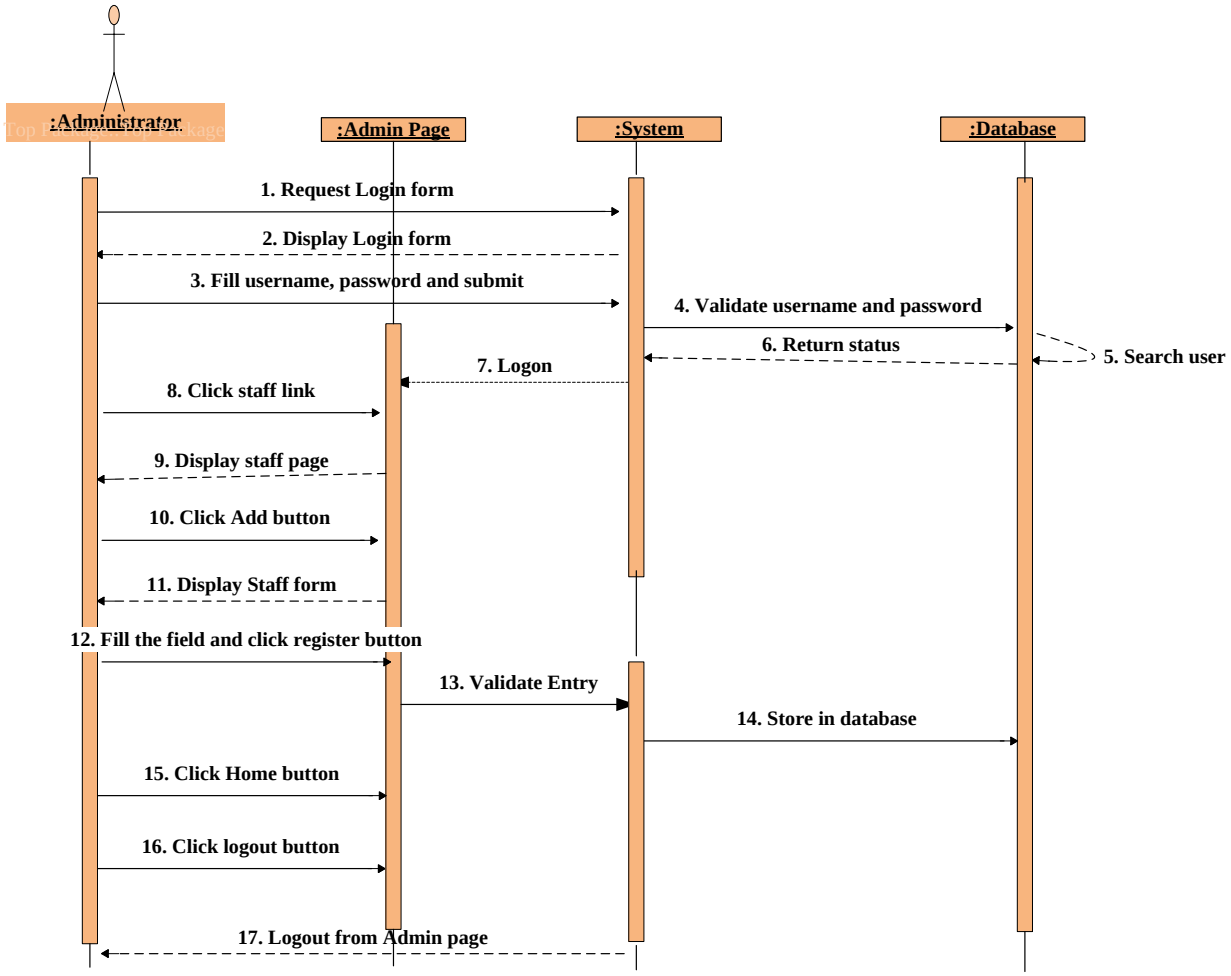


Figure 3.15: Sequence Diagram for manage staff

Description for manage staff sequence diagram

- The administrator request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.

- In database search the user name and the password.
- Retrieve the status of the user.
- If the user is authorized the administrator logon on to admin page.
- To register staff the administrator should click the manage staff link
- The system displays the staff page
- administrator should click the register button to register staffs
- The administrator should fill all necessary data in the form
- The system validate the fields
- It will be stored on the database
- Lastly the administrator click logout button.
- And finally the system logout the administrator from admin page

3.5.3.1.2 Sequence Diagram for manage user account

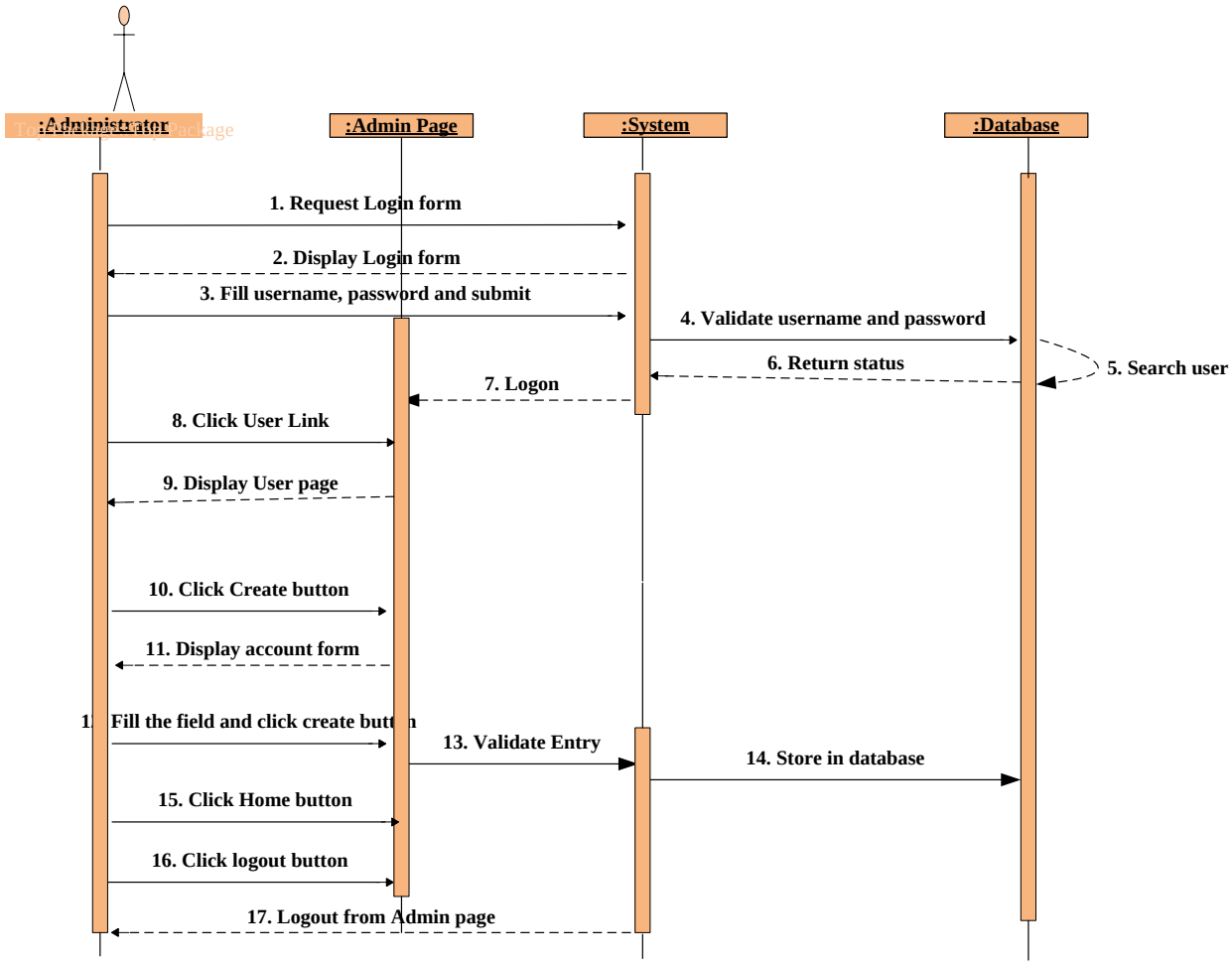


Figure 3.16: Sequence Diagram for manage user account

Description for manage user sequence diagram

- The administrator request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- In database search the user name and the password.
- Retrieve the status of the user.
- If the user is authorized the administrator logon on to admin page.
- To create user account administrator should click manage user link
- System displays user page
- To create user account administrator should click create user account button
- Display the create account form.
- Fill the field and click create button.
- The systems validate and authenticate the data and store in database.
- Lastly the administrator click logout button.
- And finally the system logout the administrator from admin page

3.5.3.1.3 Sequence Diagram for manage patient

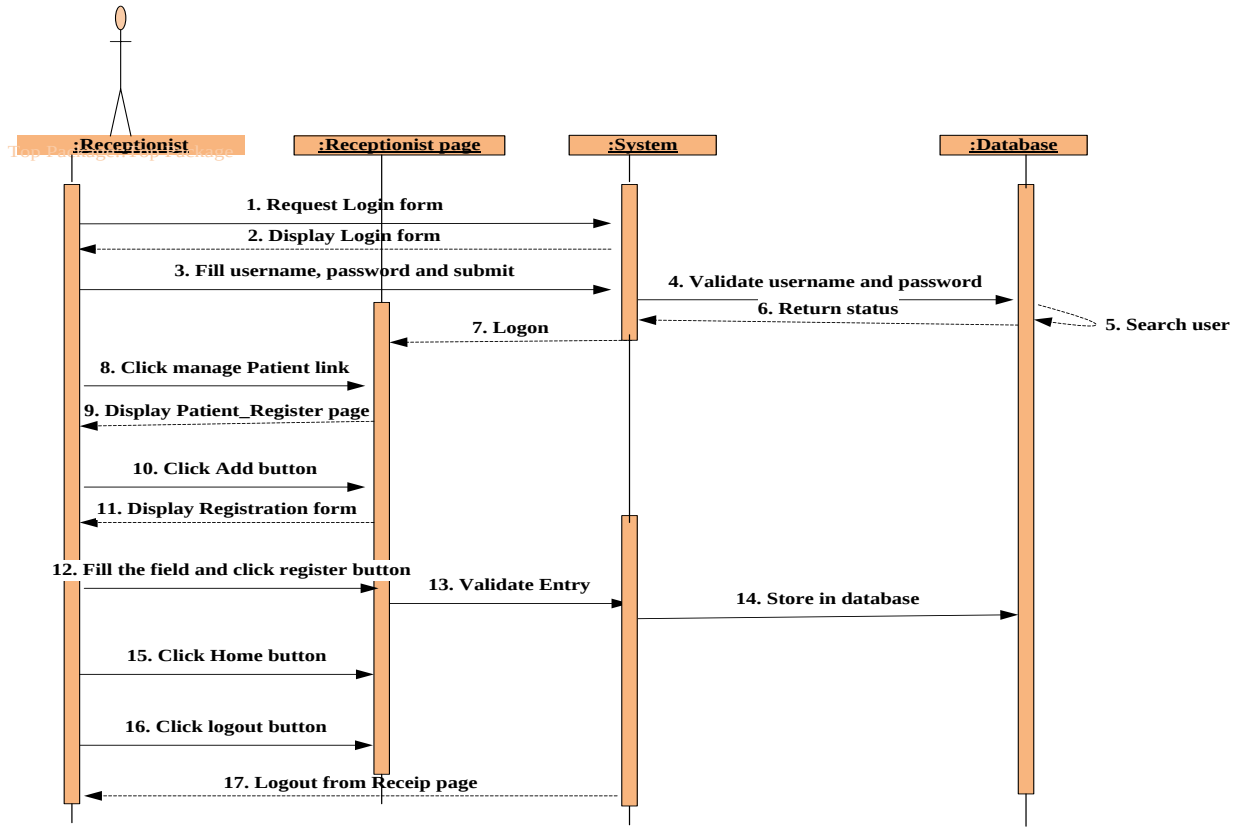


Figure 3.17: Sequence Diagram for manage patient

Description for manage patient detail sequence diagram

- The receptionist request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the receptionist logon on to receptionist page.
- To register patient the receptionist should click the patient link
- The system displays the patient page
- receptionist should click the register button to register patient
- The receptionist should fill all necessary data in the form
- The system validate the patient name
- It will be stored on the database
- Lastly the receptionist click logout button.
- And finally the system logout the receptionist from receptionist page

3.5.3.1.4 Sequence Diagram for manage room

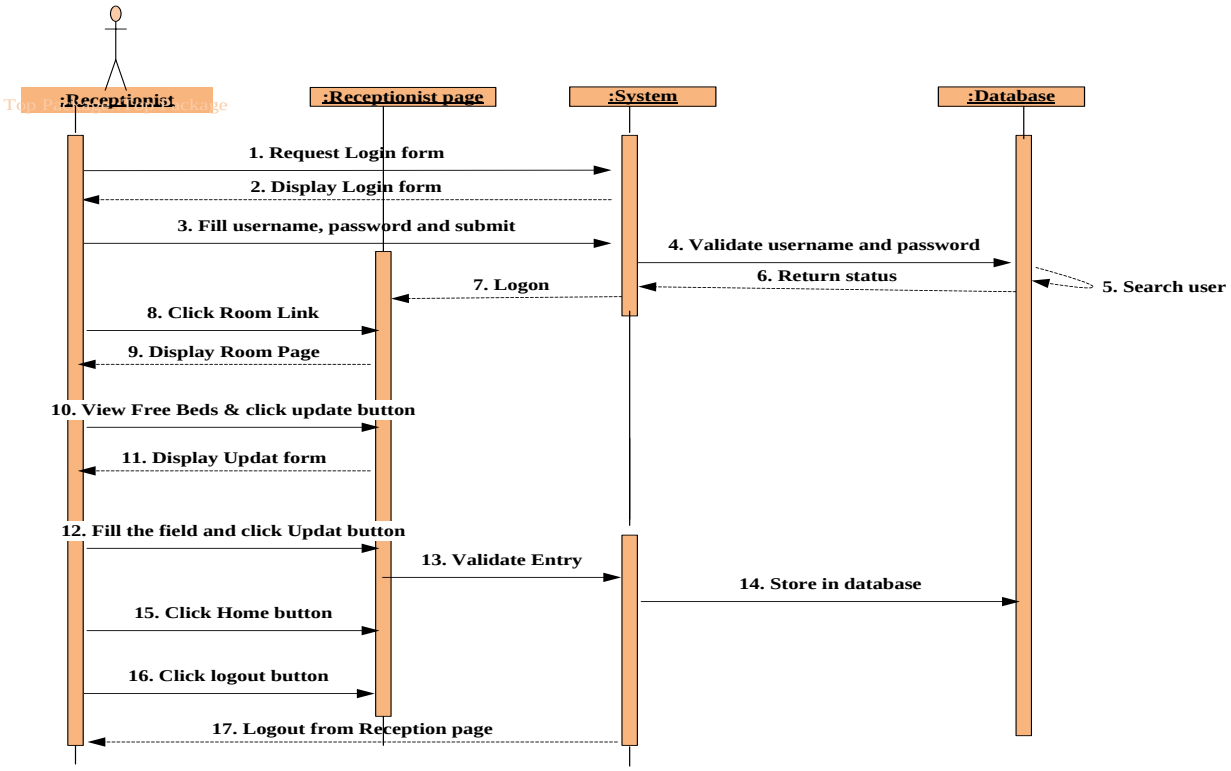


Figure 3.18: Sequence Diagram for manage room

Description for manage room sequence diagram

- The receptionist request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the receptionist logon on to receptionist page.
- To check availability of room the receptionist click room link
- Displays the lists of rooms and beds in the hospital
- If free bed is available assign the patient and click update button otherwise he/she should make some arrangement
- Lastly the receptionist click logout button.
- And finally the system logout the receptionist from receptionist page

3.5.3.1.5 Sequence Diagram for manage appointment

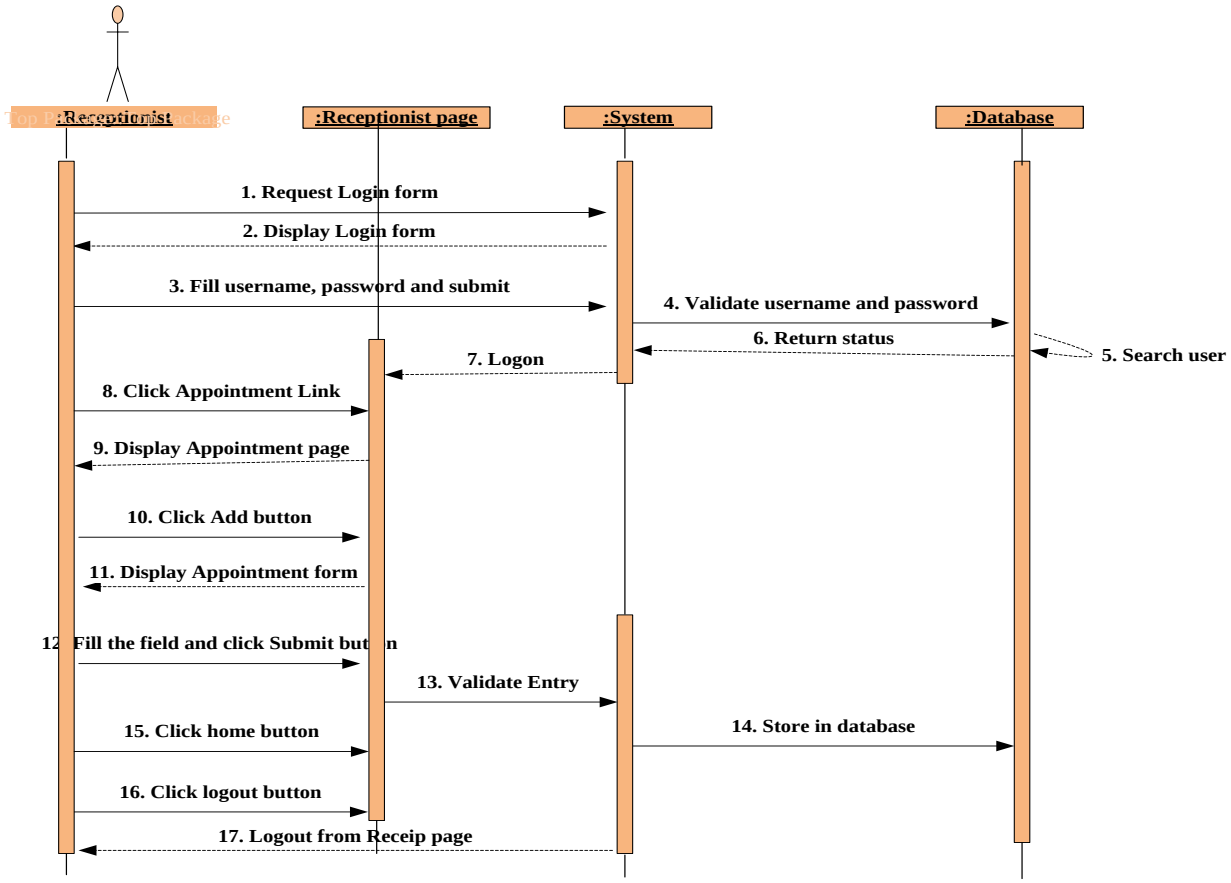


Figure 3.19: Sequence Diagram for manage appointment

Description for manage appointment sequence diagram

- The receptionist request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the receptionist logon on to receptionist page.
- To manage appointment receptionist should click appointment link
- System displays appointment page
- To create appointment receptionist should click add appointment button
- Displays add appointment button form.
- Fill the account field and click add button.
- The systems validate and authenticate the data and store in database.
- Lastly the receptionist click logout button.
- And finally the system logout the receptionist from receptionist page

3.5.3.1.6 Sequence Diagram for appointment

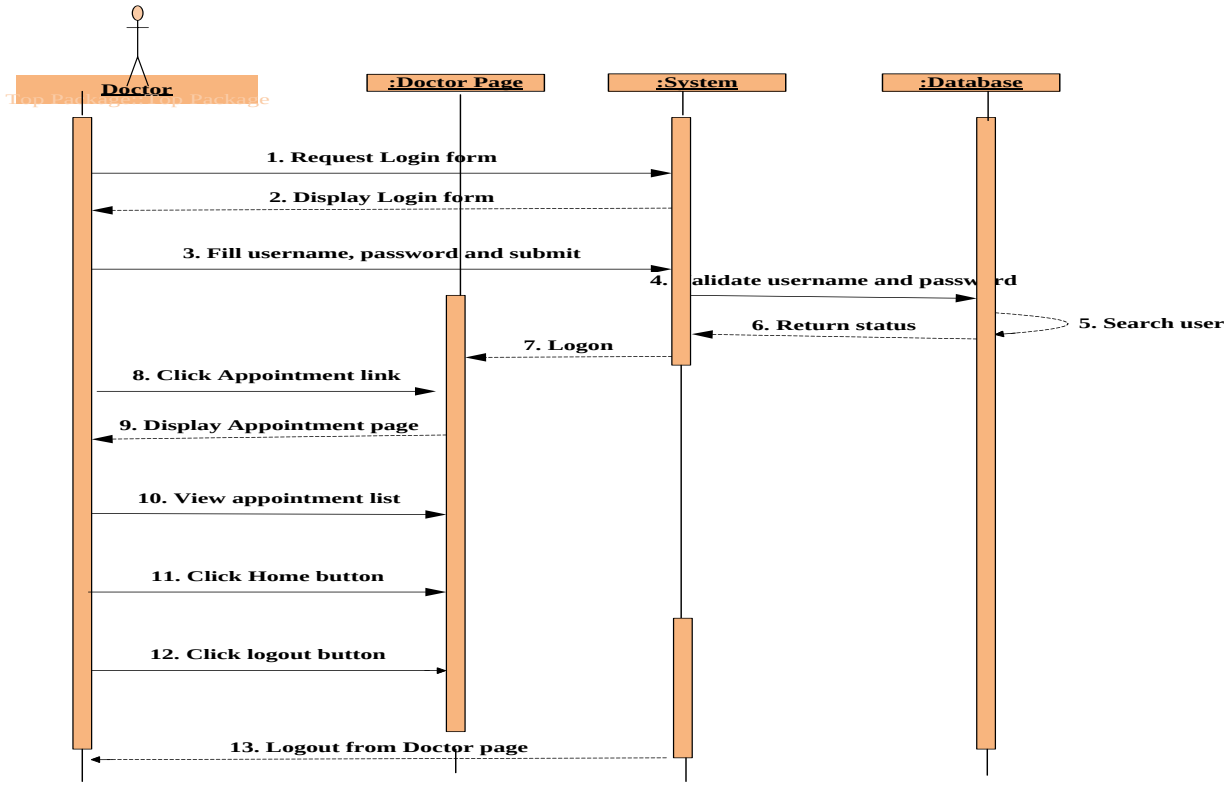


Figure 3.20: Sequence Diagram for appointment

Description for appointment Sequence diagram

- Doctor request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the doctor logon on to doctor page.
- To view appointment the doctor should click the appointment link
- The system displays the appointment page
- if there is a new appointment the doctor clicks the home button
- Lastly the doctor click logout button
- And finally the system logout the doctor from doctor page

3.5.3.1.7 Sequence Diagram for manage patient treatment

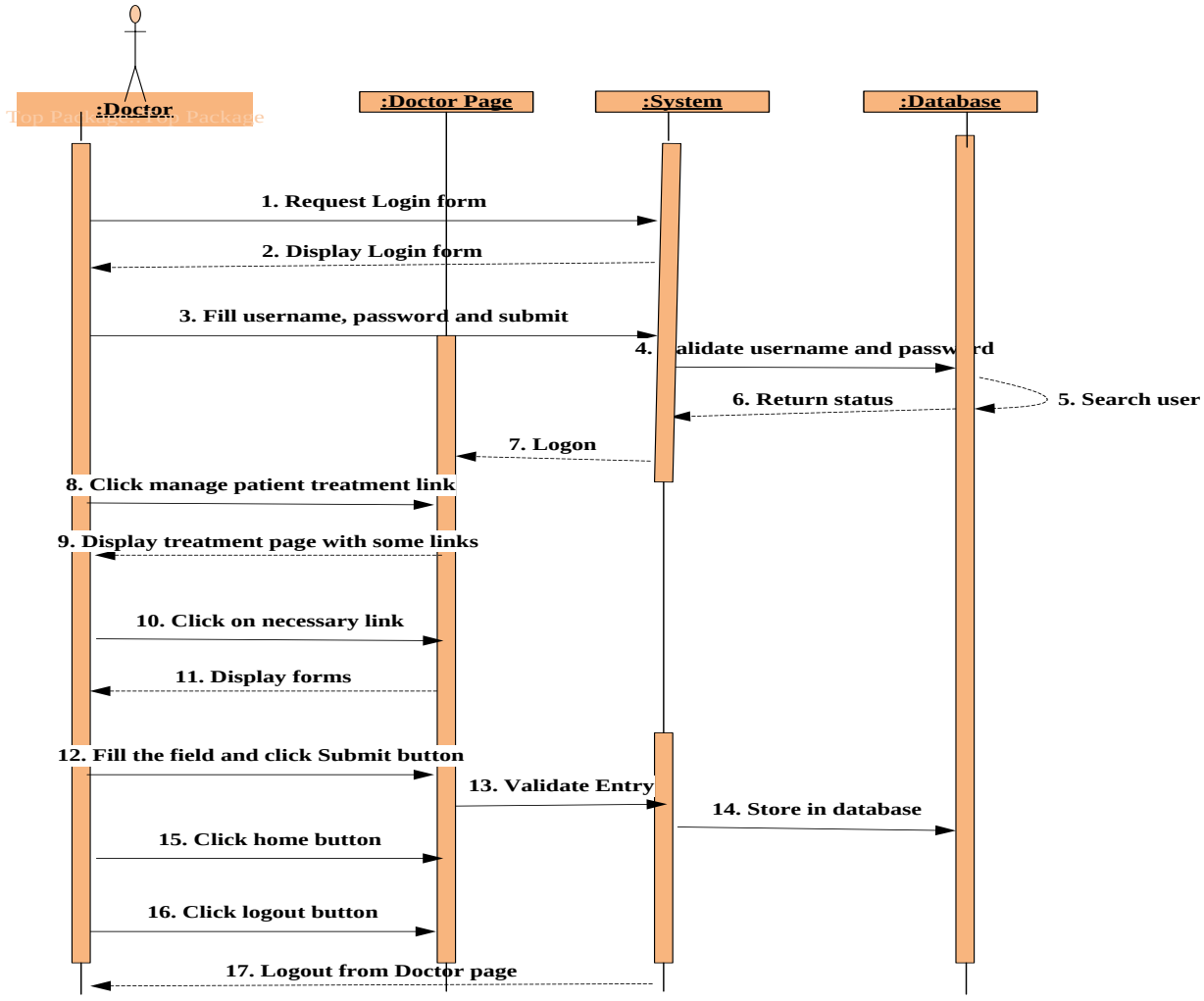


Figure 3.21: Sequence Diagram for manage patient treatment

Description for manage patient treatment Sequence diagram

- Doctor request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the doctor logon on to doctor page.
- Clicks on the manage patient treatment link
- The system displays treatment page
- Doctor fill the fields and click add button
- System validates the input and store in database
- Lastly the doctor click logout button
- And finally the system logout the doctor from doctor page

3.5.3.1.8 Sequence Diagram for admission summary

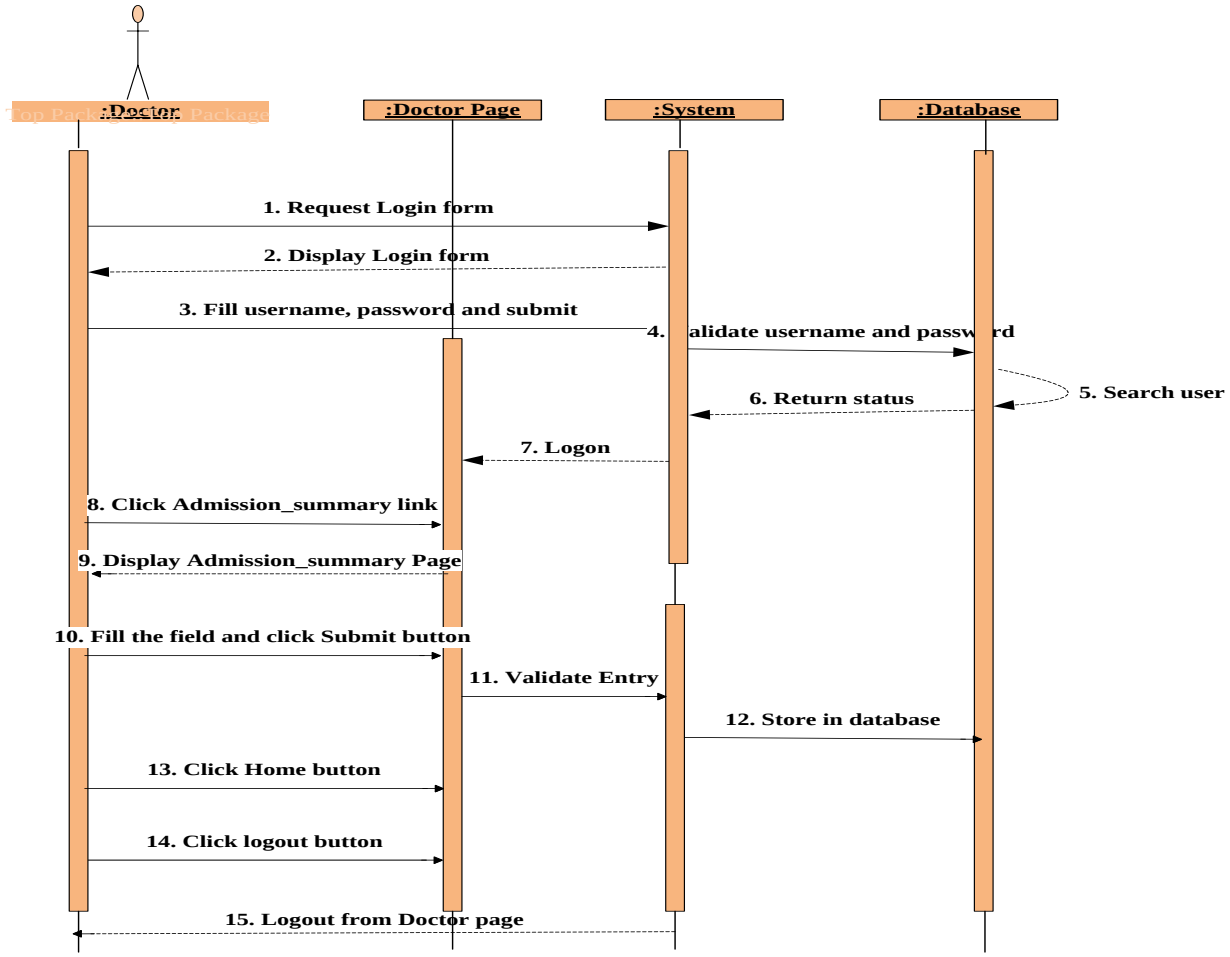


Figure 3.22: Sequence Diagram for admission summary

Description for admission summary Sequence diagram

- Doctor request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the doctor logon on to doctor page.
- Clicks on the admission summary link
- System displays admission summary page
- Fill the field and click add button
- System validates the input and store in database
- Lastly the doctor click logout button
- And finally the system logout the doctor from doctor page

3.5.3.1.9 Sequence Diagram for discharge summary

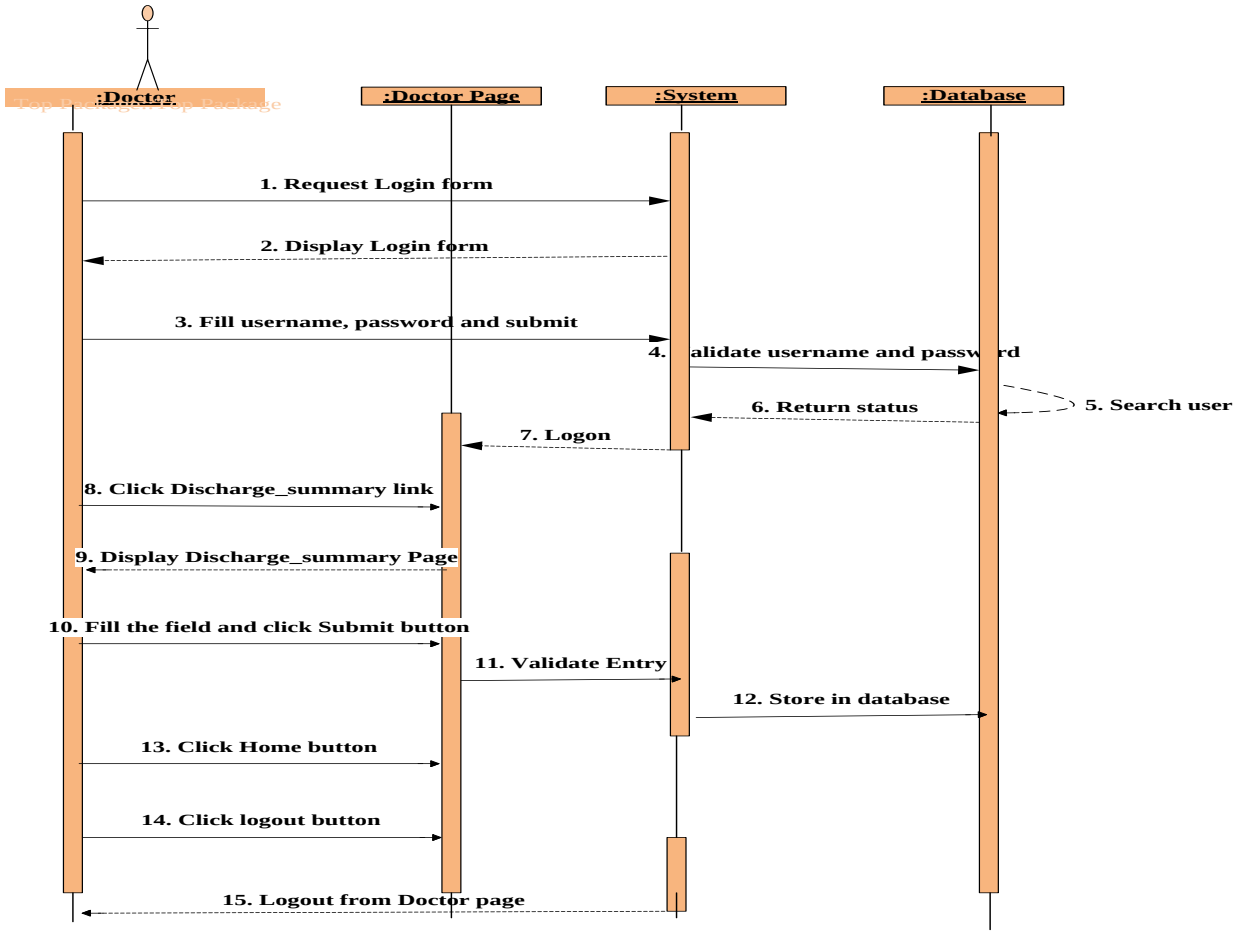


Figure 3.23: Sequence Diagram for discharge summary

Description for discharge summary Sequence diagram

- Doctor request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the doctor logon on to doctor page.
- Doctor clicks the Discharge summary link
- Discharge summary page displays
- Doctor clicks the add button
- Fill the data and click the submit button
- The system validate and authenticates the data and store in database
- Lastly the doctor click logout button
- And finally the system logout the doctor from doctor page

3.5.3.1.10 Sequence Diagram for manage patient treatment

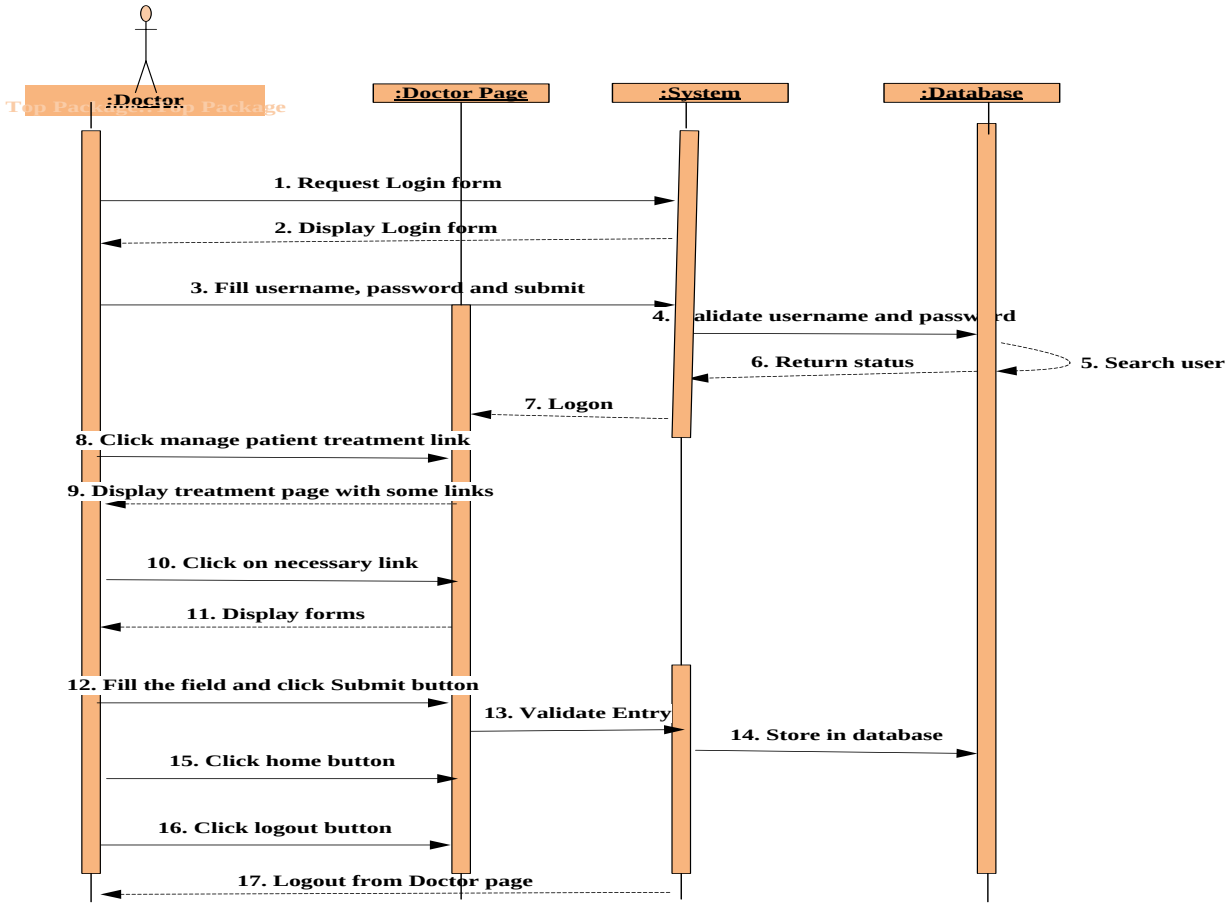


Figure 3.24: Sequence Diagram for manage patient treatment

Description for manage patient treatment sequence diagram

- The technicians request for login by entering the username and password to submit the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the technicians logon on to technicians page.
- Technician should click the lab/rad request link
- The system displays the lab/rad request page
- View doctor's request and click the add button to add lab/rad request
- Fill the data and click submit button
- The system validate and authenticate the data and store in database
- Lastly the technician click logout button
- And finally the system logout the technician from technician page

3.5.3.1.11 Sequence Diagram for prescription

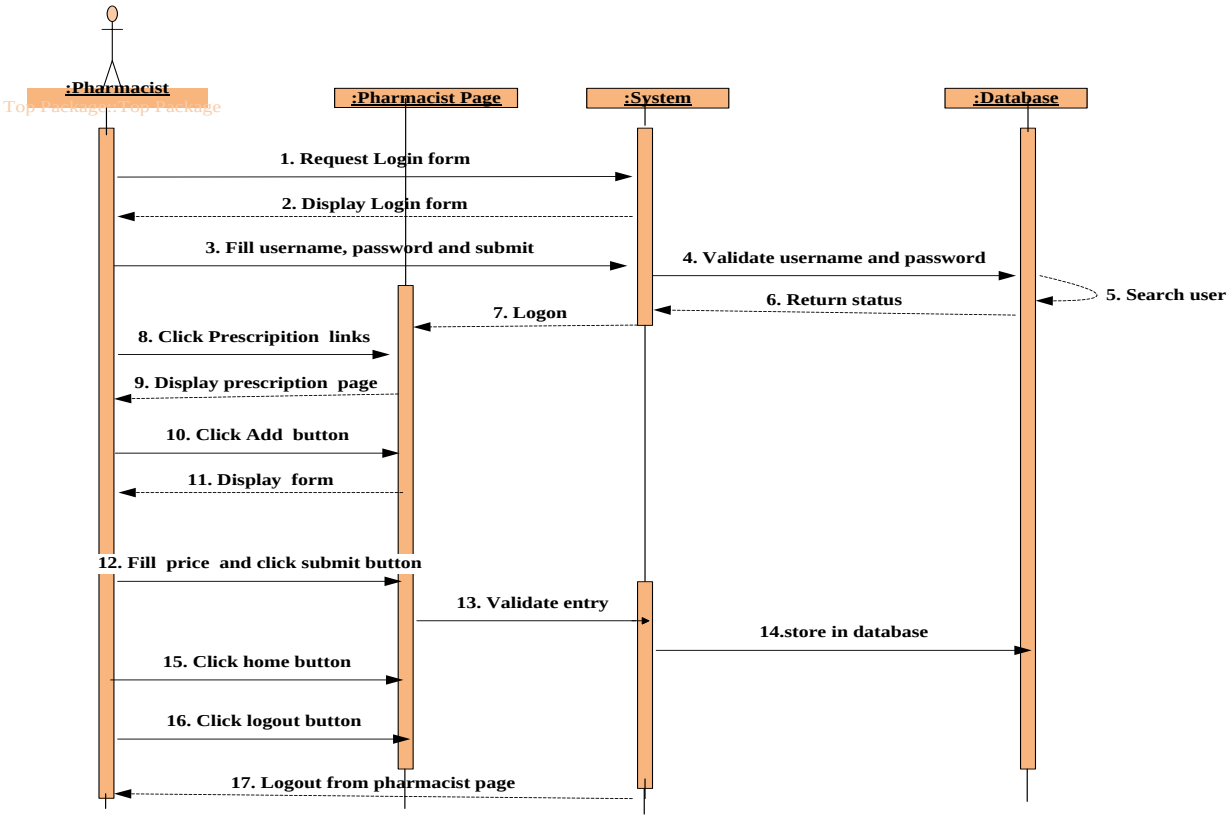


Figure 3.25: Sequence Diagram for prescription

Description for prescription sequence diagram

- The Pharmacist request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the Pharmacist logon on to Pharmacist page
- To view prescribed medicines the Pharmacist click the prescription link
- The system displays the prescription page
- Pharmacist view the medicine and click the home button
- Pharmacist fill the price of the drugs in the prescription page and click the submit button
- The system validate and authenticate the data and store in database
- Lastly the Pharmacist click logout button
- And finally the system logout the Pharmacist from Pharmacist page

3.5.3.1.12 Sequence Diagram for manage drug

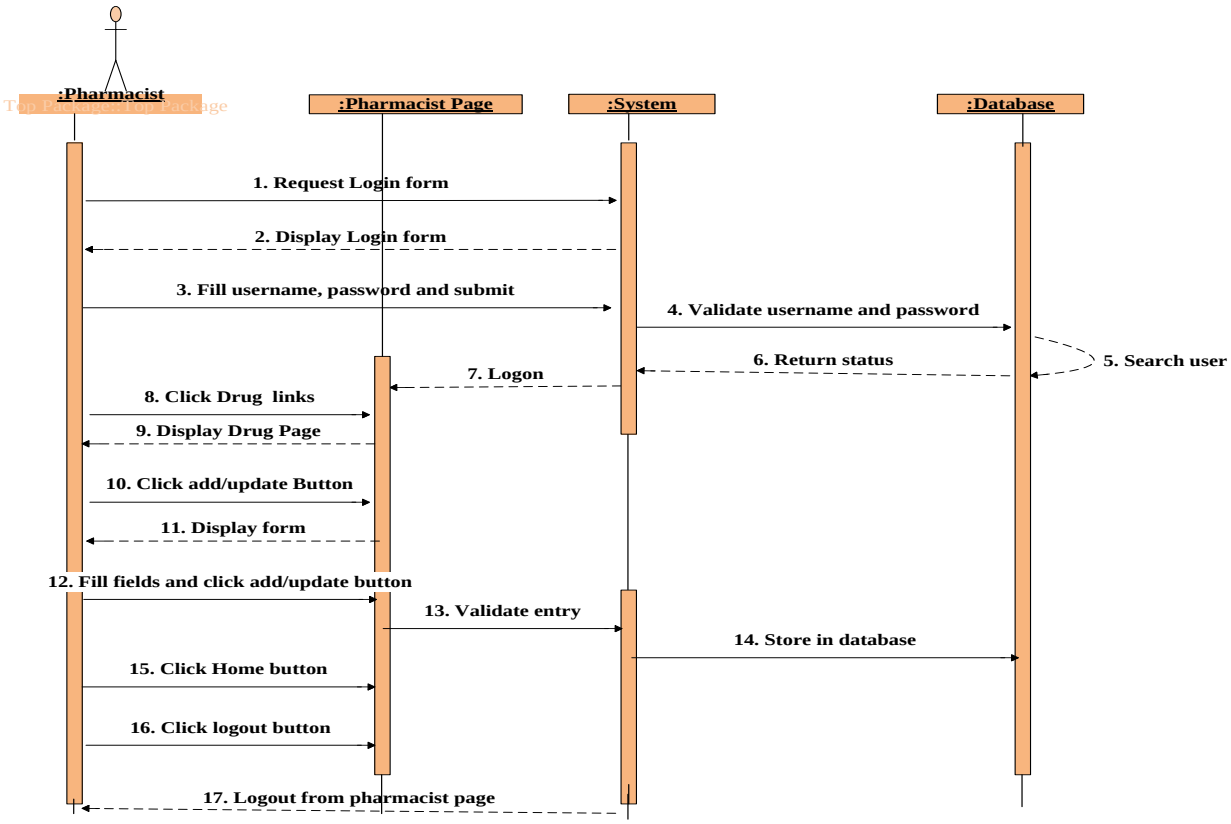


Figure 3.26: Sequence Diagram for manage drug

Description for manage drug sequence diagram

- The Pharmacist request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the Pharmacist logon on to Pharmacist page
- To check the availability of drugs the pharmacist clicks the drug link
- System displays drug list and their price
- If drug is available pharmacist checks the price of the drugs otherwise inform the patient to buy from another pharmacy outside of the hospital
- Lastly the Pharmacist click logout button
- And finally the system logout the Pharmacist from Pharmacist page

3.5.3.1.13 Sequence Diagram for manage payment

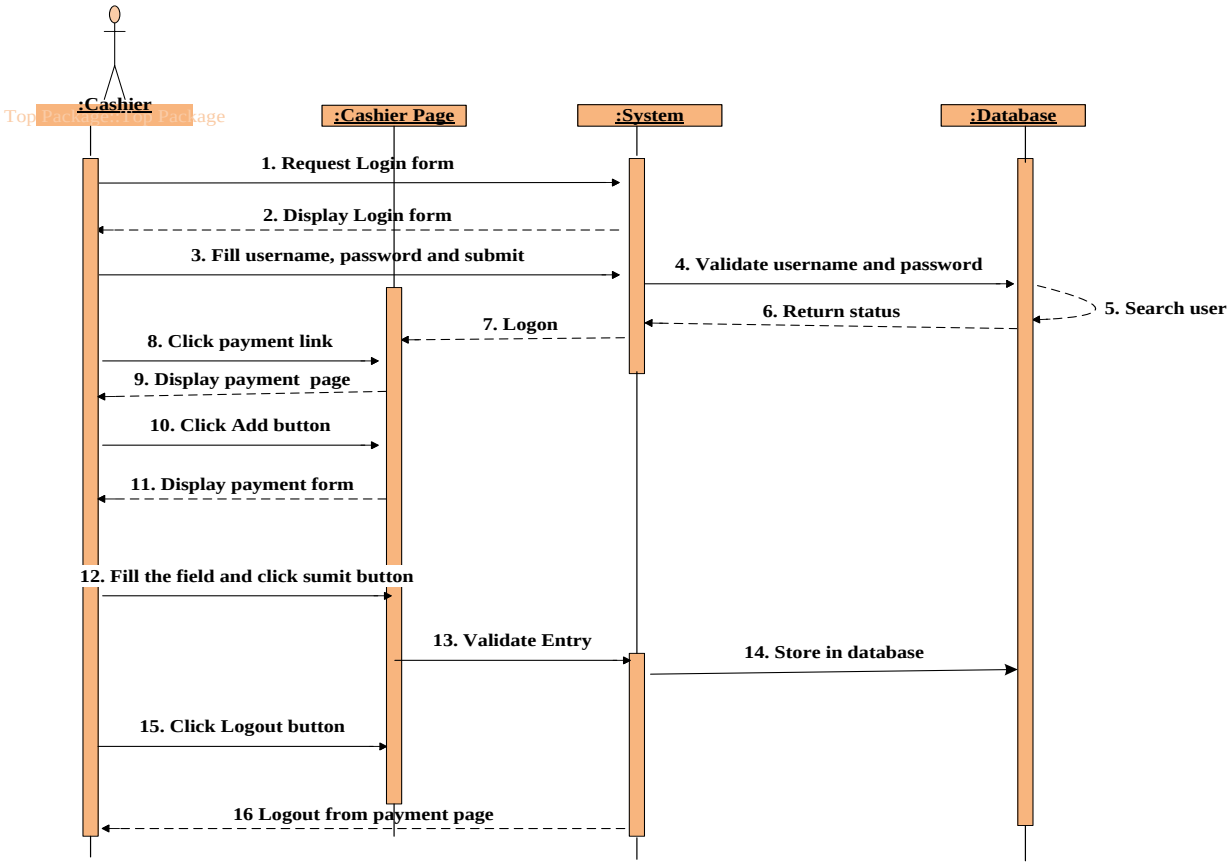


Figure 3.27: Sequence diagram for manage payment

Description for manage payment sequence diagram

- The cashier request for login by entering the username and password to submits the form.
- The system validates the username and password of the user in Database.
- Search the user name and the password in the database.
- Retrieve the status of the user.
- If the user is authorized the cashier logon on to payment page.
- To manage payment cashier should click payment link
- System displays payment page
- Cashier enter MRN of the patient
- System displays patients each fee
- cashier should click Total button
- Lastly the cashier click logout button.
- And finally the system logout the casher from payment page

3.5.3.2 State chart diagram

State diagram displays the sequences of states that an object of an interaction goes through during its life in response to received stimulus, together with its responses and actions.

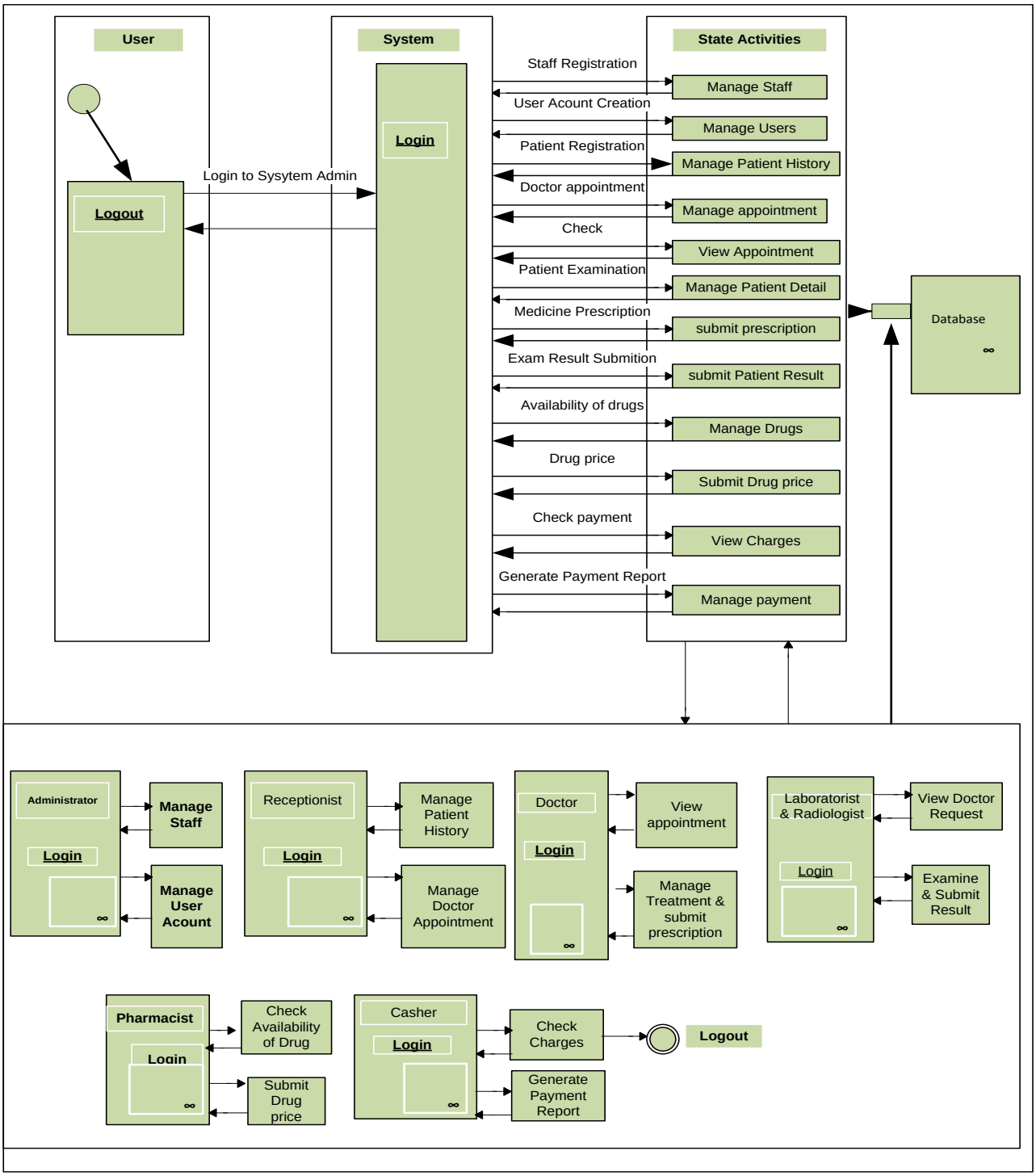


Figure 3.28: State chart Diagrams

3.5.3.3. Activity diagrams

Activity diagram displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.

3.5.3.3.1 Activity Diagram of Manage Staff and Manage User Account

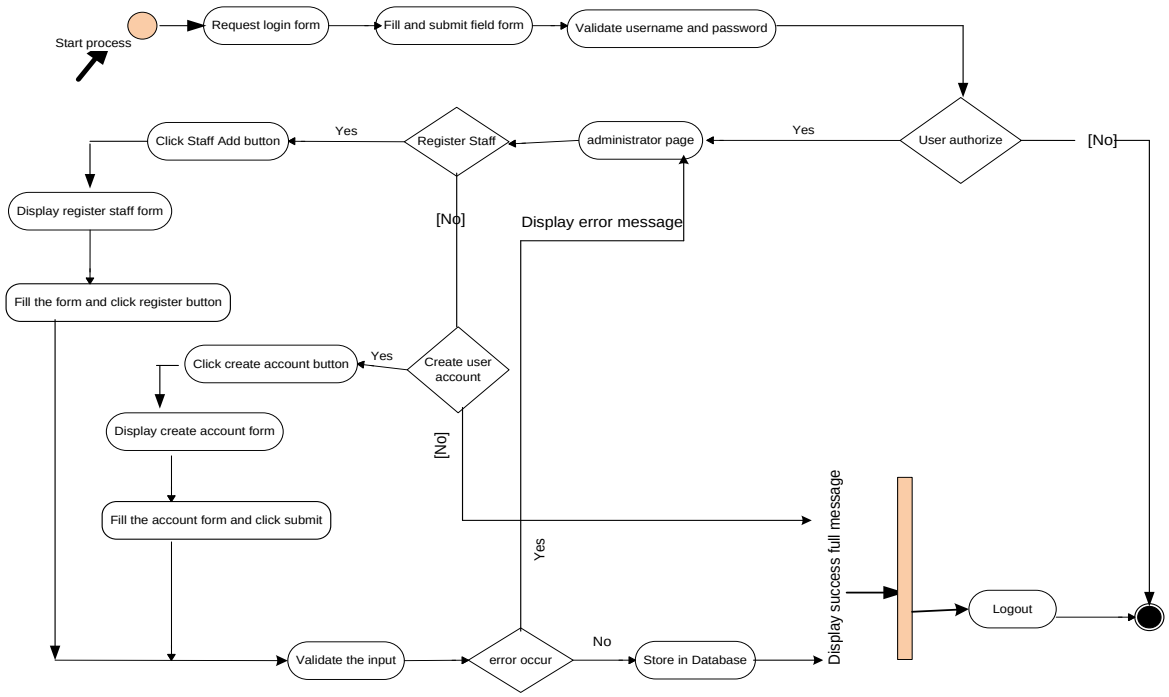


Figure 3.29: Activity Diagram of Manage Staff and Manage User Account

Description for administrator activity diagram

Staff registration and user account creation is performed by system administrator privileges. The administrator login to the system's administrator page the administrator page contains two links that is Manage staff (which includes staff registration, updating and deletion) and Manage user accounts (which includes user account creation and updating).Therefore, the administrator tries to login to its page and the system validates the username and password. After login the administrator Registers staff and create user accounts of each staff but the system validates each data entered to the database and store to the database otherwise the system displays error message.

3.5.3.3.2 Activity Diagram for Manage Patient detail, Manage room and Manage Appointment

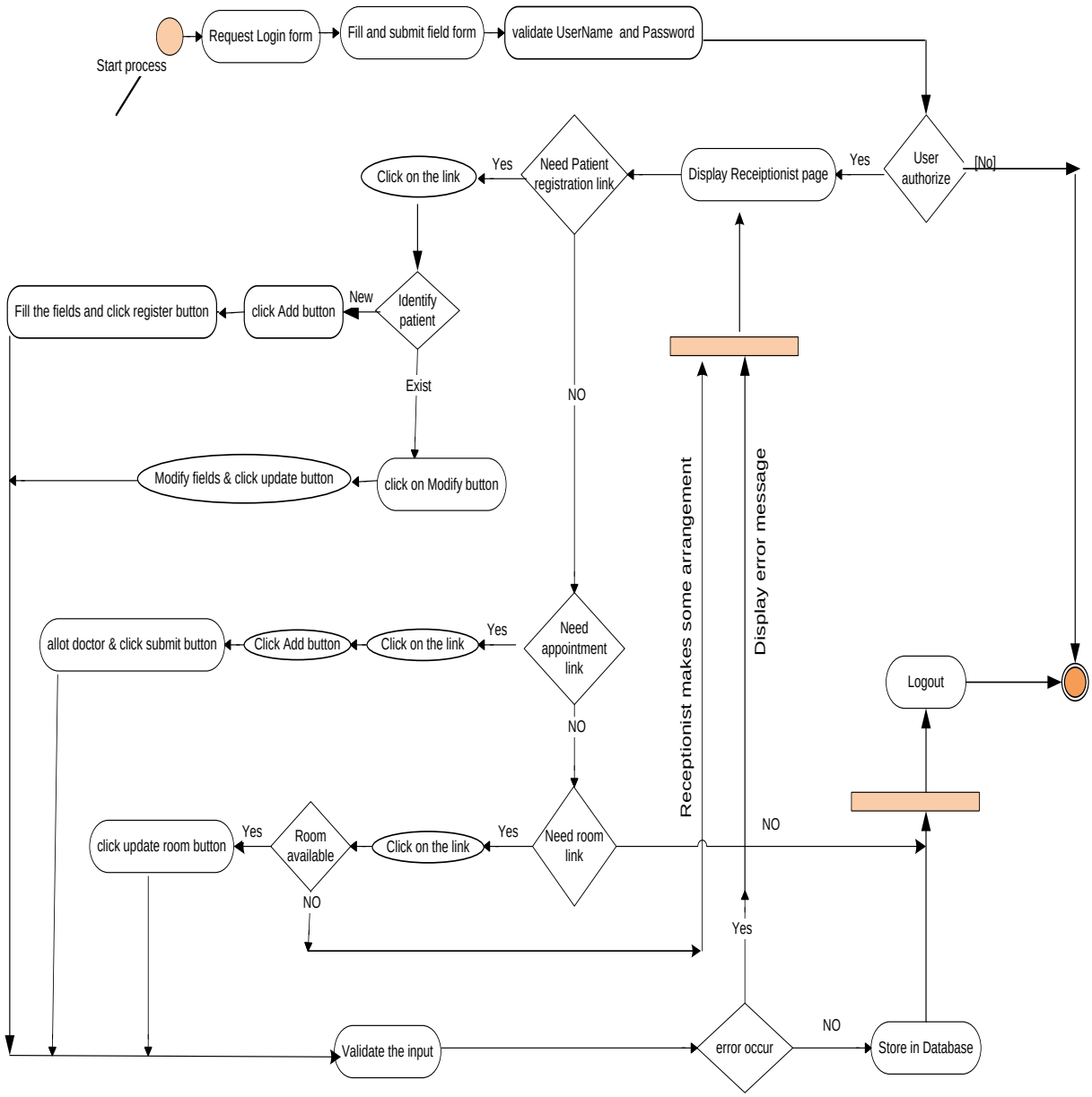


Figure 3.30: Activity Diagram for Manage Patient detail, room and Appointment

Description activity diagram for reception

Receptionist login to his/her page using his/her user name and password and the system display the receptionist page if user name and password is correct otherwise the receptionist cannot logon. The receptionist manages patient detail, manage room and doctor appointment.

3.5.3.3.3 Activity Diagram for Manage Patient Treatment, Admission Summary, Discharge Summary and Appointment

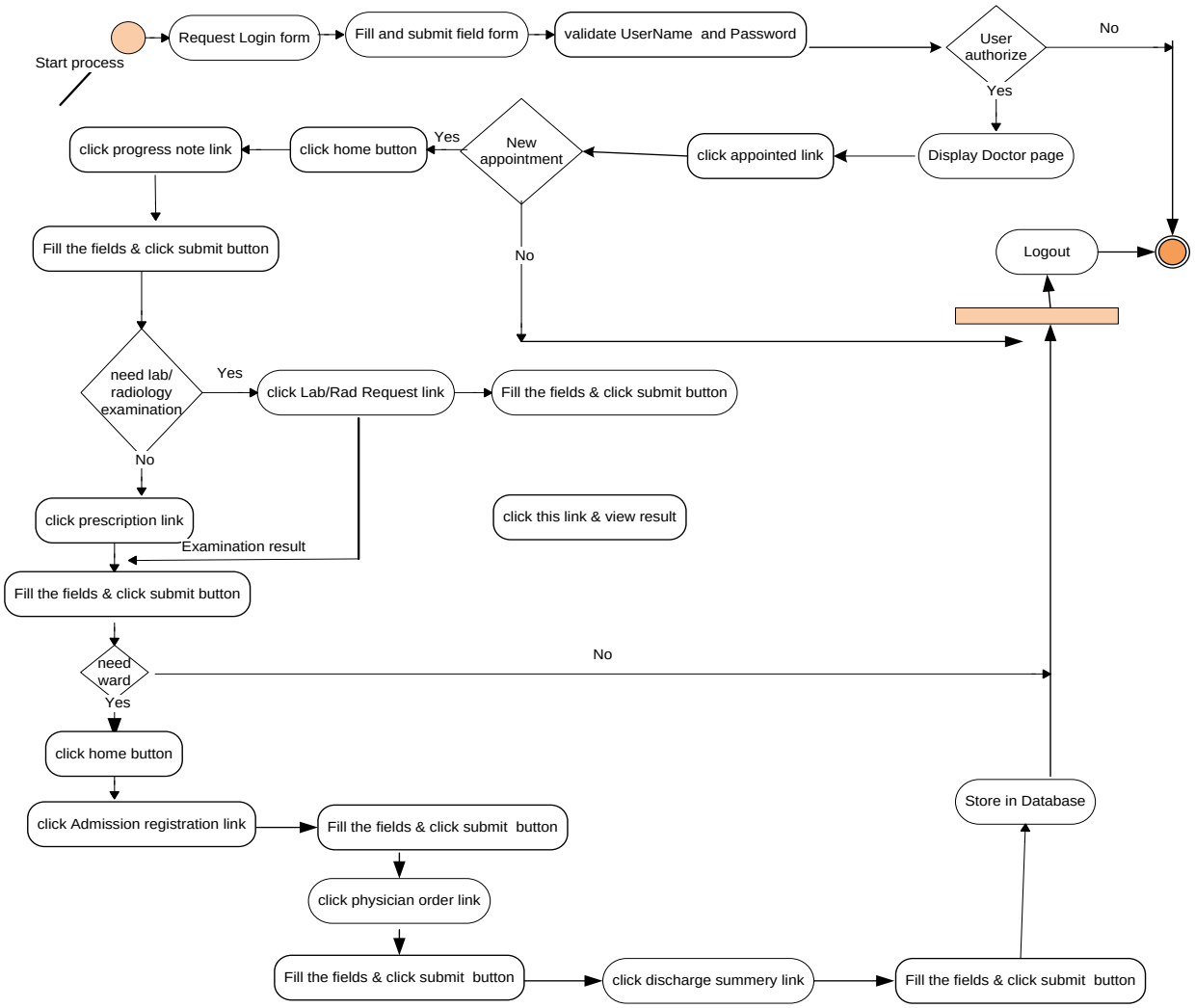


Figure 3.31: Activity Diagram for Manage Patient Treatment, Admission Summary, Discharge Summary and Appointment

Description for Doctors activity diagram

Doctor login to his/her page using his/her user name and password and the system display the doctor's page if user name and password is correct otherwise the doctors cannot logon. Doctor view appointments and manage patient treatment based on the progress note and physician order sheet and send laboratory request form to technicians. Finally he/she manages prescription.

3.5.3.3.4 Activity Diagram for Manage Patient Treatment

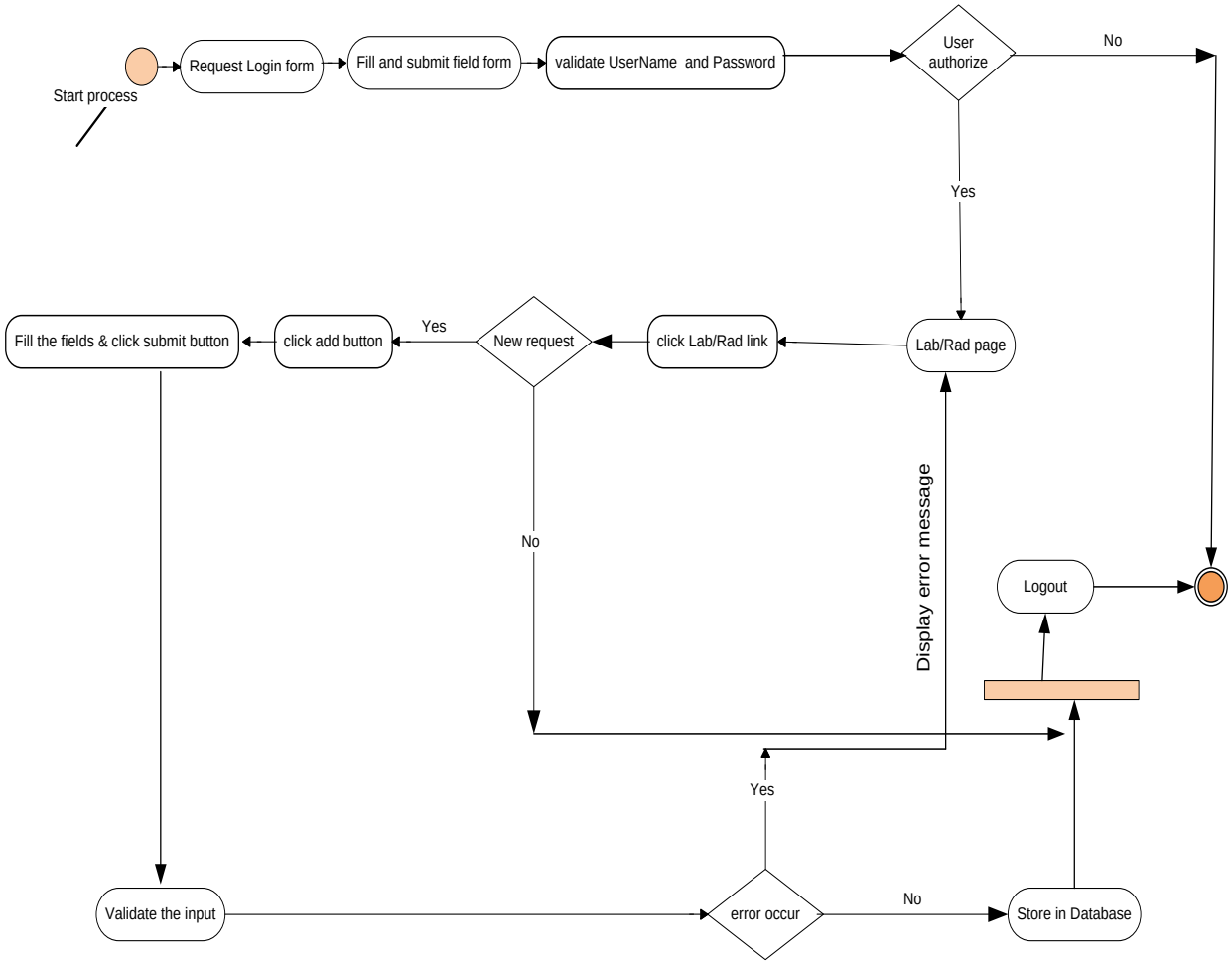


Figure 3.32: Activity Diagram for Manage Patient Treatment

Description for Laboratory and Radiology activity Diagram

Technicians login to his/her page using his/her user name and password and the system display the technician's page if user name and password is correct otherwise the technician cannot logon. She/he view laboratory request from the Doctor and undertake the patient examination and sent back the result to doctor.

3.5.3.3.5 Activity Diagram for View Prescription and Manage Drugs

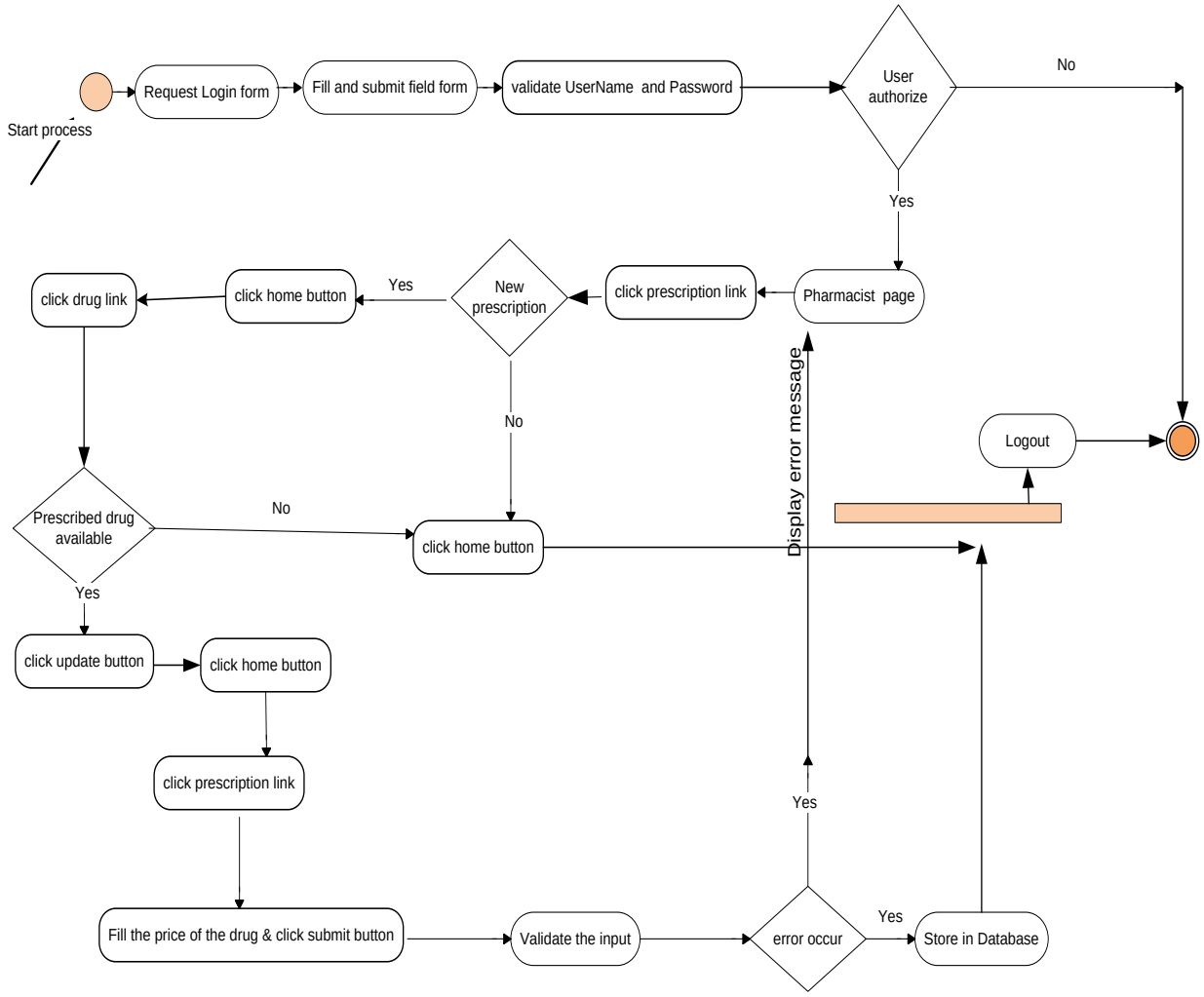


Figure 3.33: Activity Diagram for View Prescription and Manage Drugs

Description for pharmacist activity diagram

Pharmacist login to his/her page using his/her user name and password and the system display the pharmacist page if user name and password is correct otherwise the pharmacist cannot logon. S/he view prescription detail and check the availability of drugs. Finally s/he put the drug price and submits to the database.

3.5.3.3.6 Activity Diagram for Manage Payment

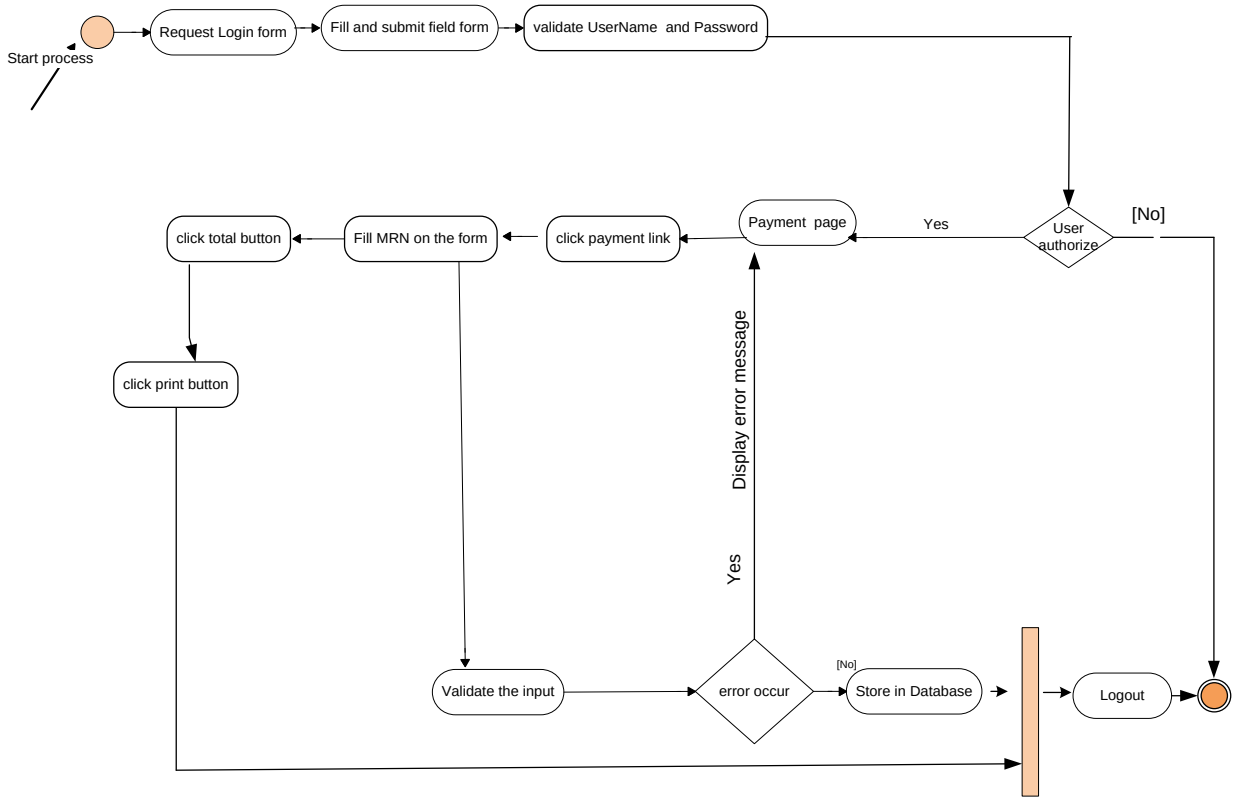


Figure 3.34: Activity Diagram for Manage Payment

Description for casher activity Diagram

Cashier login to his/her page using his/her user name and password and the system display the cashier page if user name and password is correct otherwise the cashier cannot logon. The cashier inserts patients MRN into the form and the system immediately retrieves all charges/fees of the patient and cashier calculates grand total and collects cash.

CHAPTER FOUR

SYSTEM DESIGN

4.1. INTRODUCTION

The system design describes system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, system interfaces, detailed design, processing logic, and external interfaces.

System design provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared.

4.1.1. Purpose of the system

The system design tracks the necessary information required to effectively define architecture and system design in order to give the users graphical user interface on architecture of the system to be developed system. System design are incrementally and iteratively produced during the system development life cycle, based on the particular circumstances of the information technology project and the system development methodology used for developing the web based medication and payment management system.

4.1.2. Design goals

Web-based medication and payment management system provides quality service due to time management of the patient's as well as medication practitioners and the values against which options are evaluated with the manual system practiced in the hospital before introducing the new system to expand for the following goals.

- Extension: Facilitate adding features
- Robustness: the ability to handle error occurred in the system
- Portability: it can run in any type operating system
- Readability: any user with fundamental IT skill and knowledge can read it.
- Change: Facilitate changing requirements
- Simplicity: Make easy to understand and Make easy to implement
- Efficiency:
  - Attain high speed: execution and/or compilation
  - Attain low size: runtime and/or code base

Reliability	Efficiency	User-friendliness
Modifiability	Portability	Readability
Maintainability	Fault tolerance	Ease of learning
Understandability	Backward-compatibility	Flexibility
Adaptability	Cost-effectiveness	Increased productivity
Reusability	Robustness	Minimum number of errors

Table 4.1 Design goal

4.1.3. Overview of system design

In order to fully document all the aspects of the system design, the following subsections are included:

- Description of Design goal
- Description of system decomposition
- Description of hardware software mapping
- Description of persistency data management
- Description of access control
- Description of security
- Description of software control and
- Description of boundary condition.

4. 2. CURRENT SOFTWARE ARCHITECTURE

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

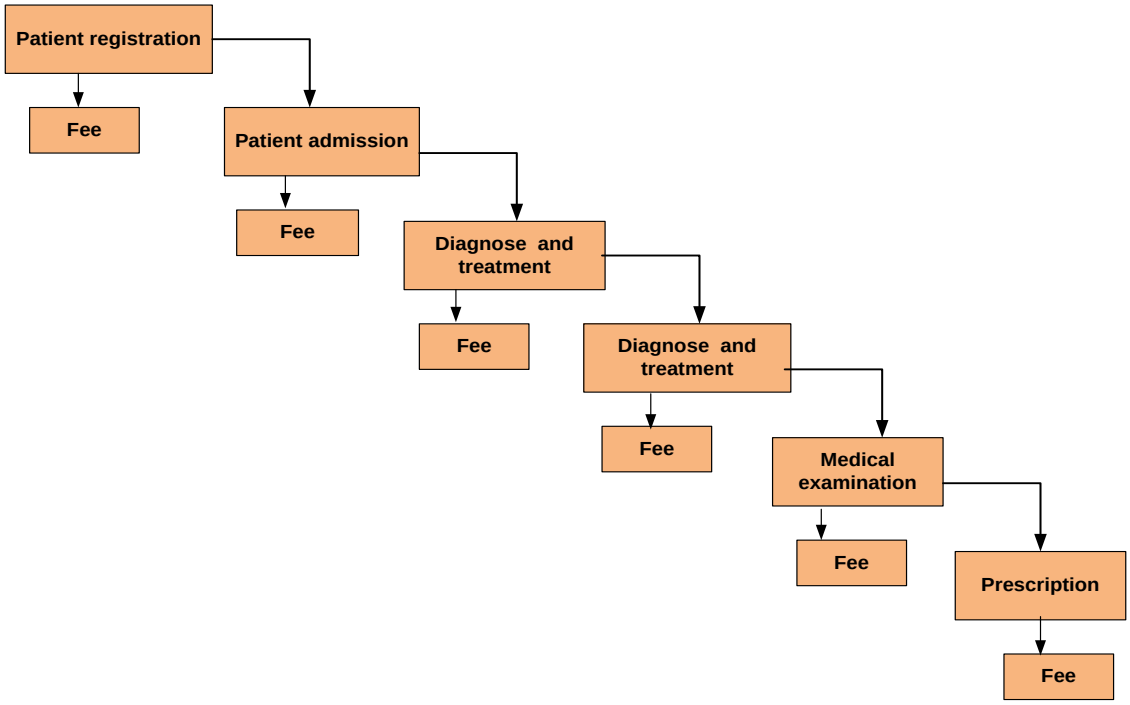


Figure 4.1: current software architecture

4. 3. PROPOSED SOFTWARE ARCHITECTURE

4.3.1. Overview

The system architecture of the project is 3-tier architecture which includes:

- The presentation tier contains the UI (User Interface) elements of the site, and includes all the logic that manages the interaction between the user
- The business tier receives requests from the presentation tier and returns a result to the presentation tier depending on the business logic it contains.
- The data tier is responsible for storing the application's data and sending it to the business tier when requested

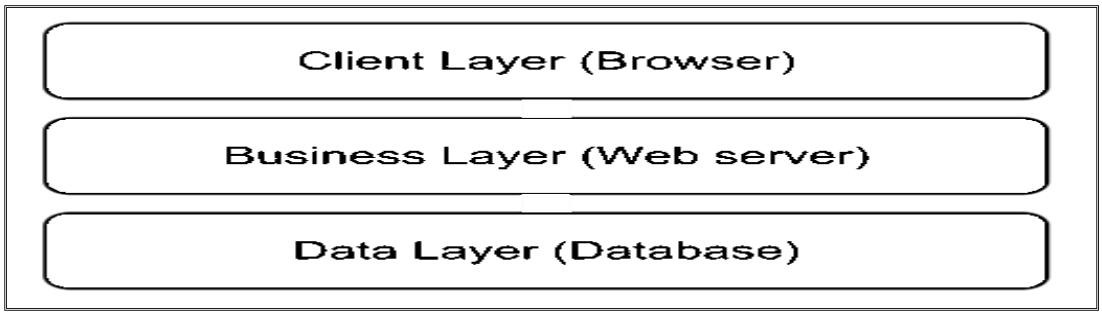


Figure 4.2: proposed software architecture

4.3.2. Subsystem decomposition

The aim of the system is to give access to medical practitioners, receptionist, administrator, Technicians and casher using patient ID or MRN to view end to end treatment needed in the system. To perform these tasks the system is divided into different subsystem.

Subsystem decomposition describes other subsystems and the responsibilities of each actor of the proposed system for patient treatment in the web-based medication and payment management system.

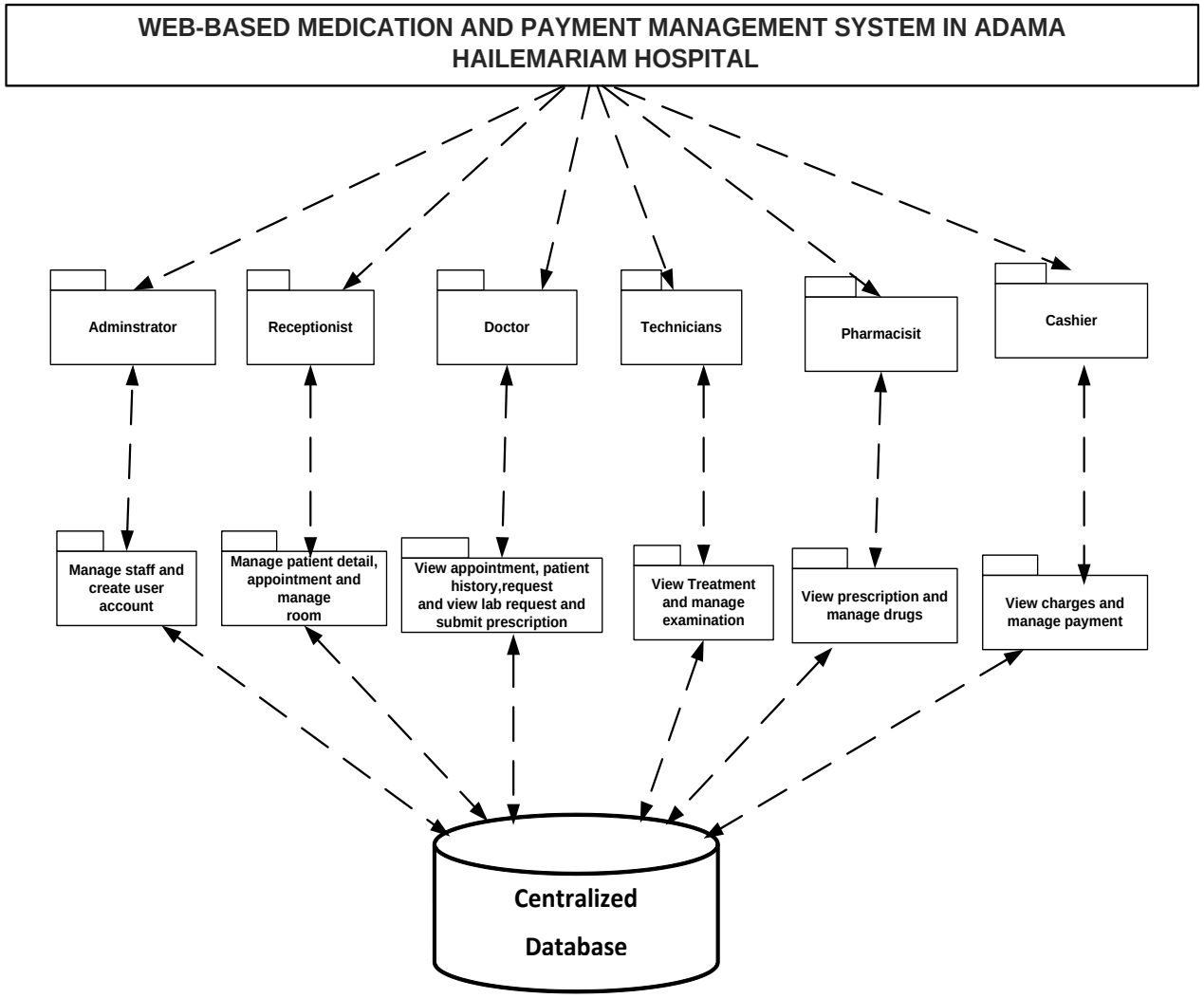


Figure 4.3: sub system decomposition

4.3.3. Hardware/Software Mapping

Hardware and software mapping provides the information needed for a system development team to build and integrate the hardware components, code and integrates the software modules, and interconnect the hardware and software segments into a functional product. It also lists the issues introduced by multiple nodes and software reuse in the systems. The medication and

payment management system includes three major components namely client, web-scripting language side PHP and database server MYSQL component. Clients communicate with the server via web browser and the server then communicates with the database via the application server. The protocol involved between the client and the server is the HTTP protocol, which allows communication between the PC's and the server.

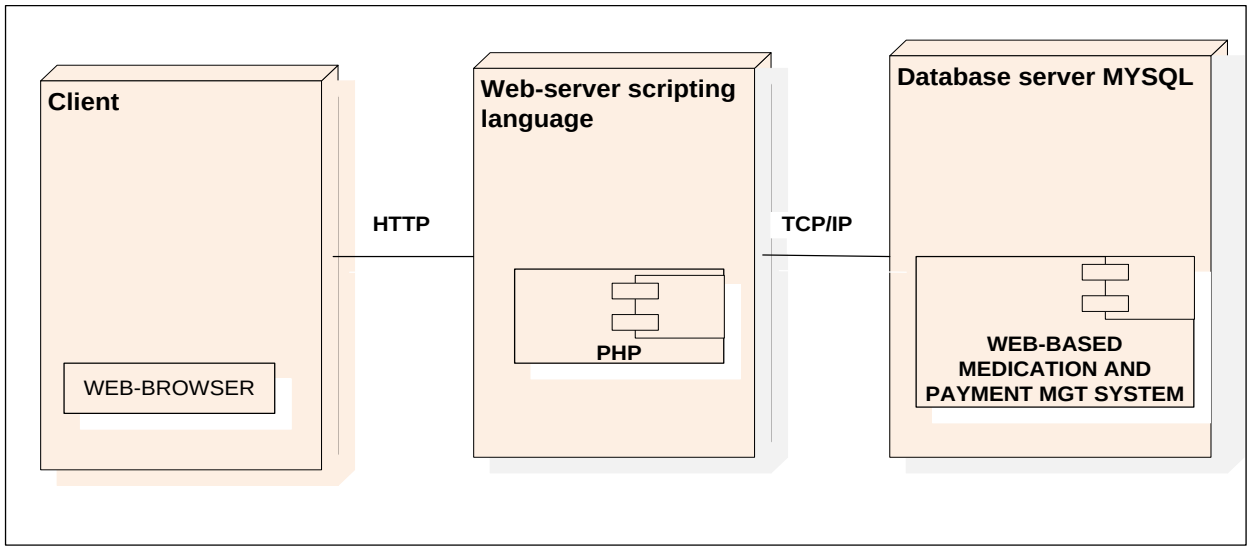


Figure 4.4 Hardware and software mapping

4.3.4. Persistent data management

Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. Typically includes the description of data schemes, the selection of a database, and the description of the encapsulation of the database.

In persistent modeling performs identifying primary key Identifies unique instance of an entity, foreign key and it helps to show the relationship between two tables or classes.

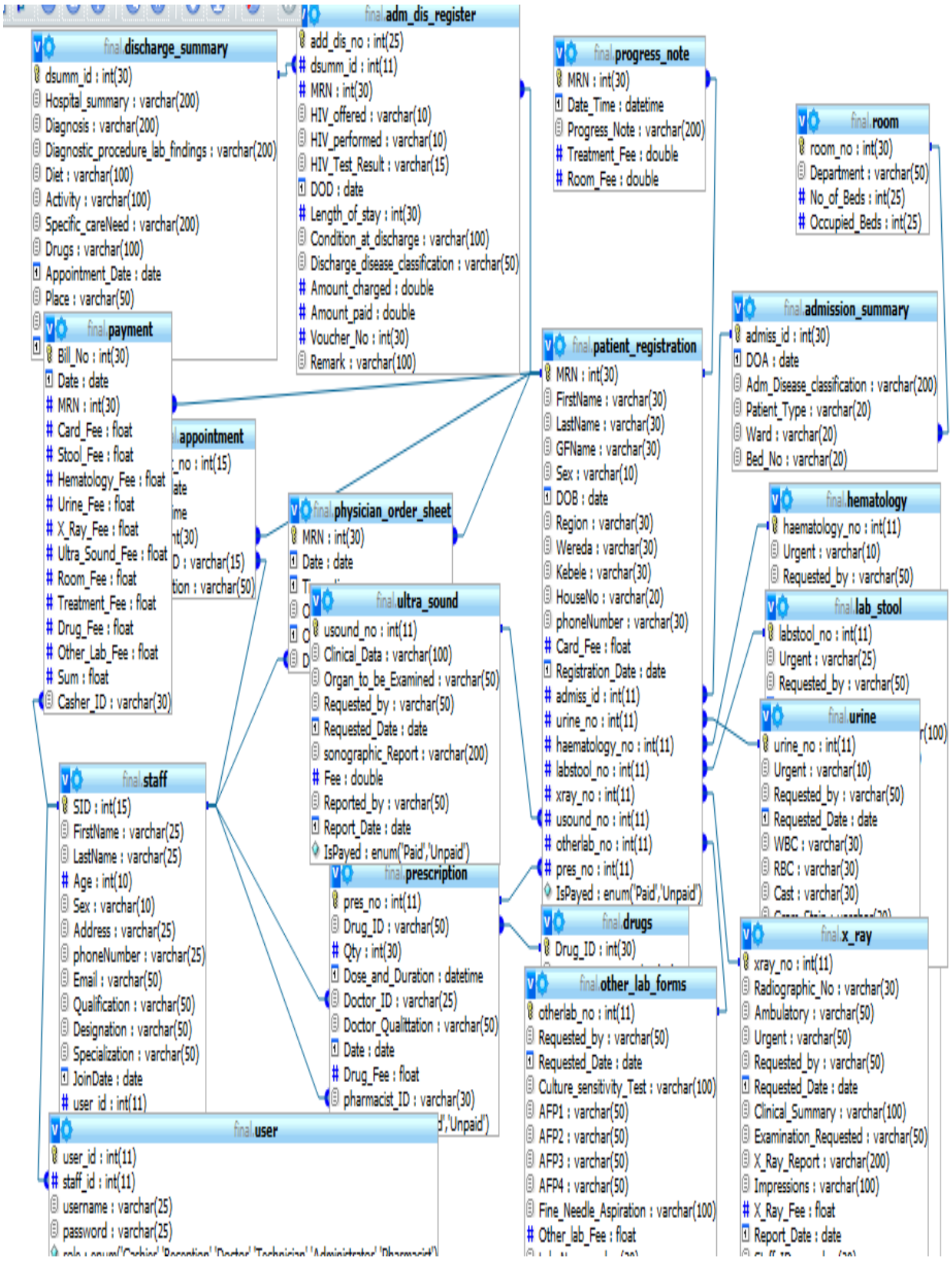


Figure 4.5: Persistent database relation diagram

4.3.5. Data flow diagrams

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

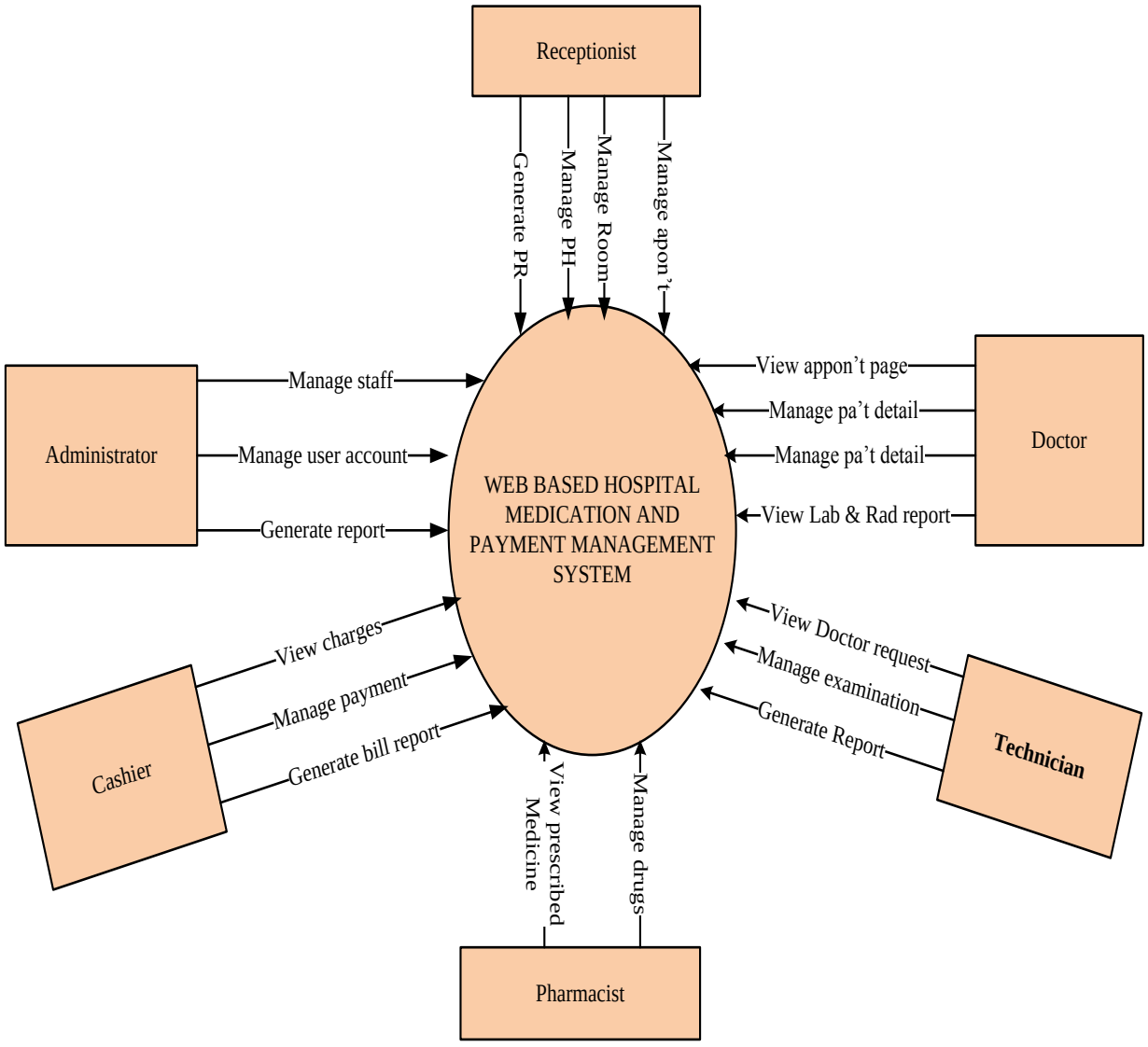


Figure 4.6: Data flow context level 0

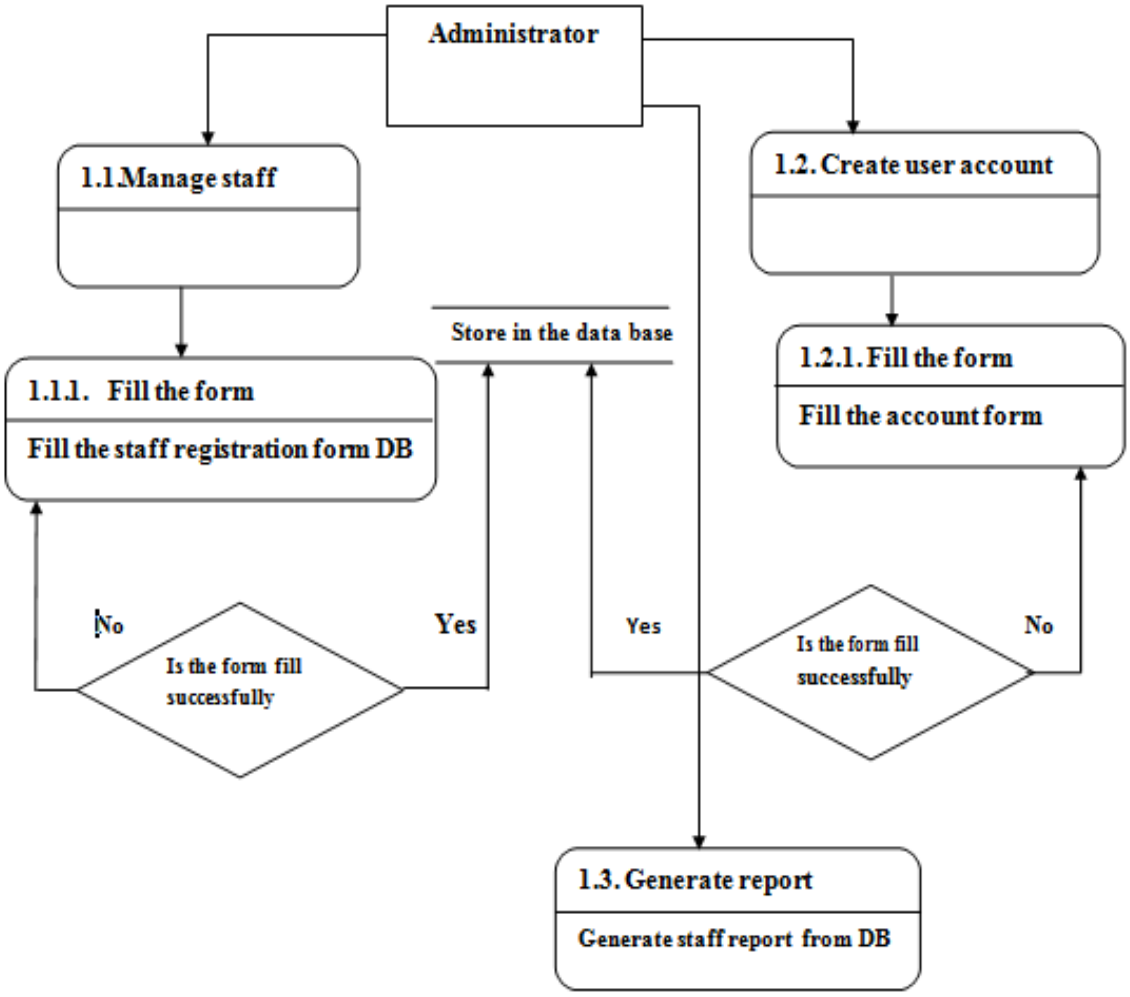


Figure 4.7: Administrator Data flow Diagram level 1

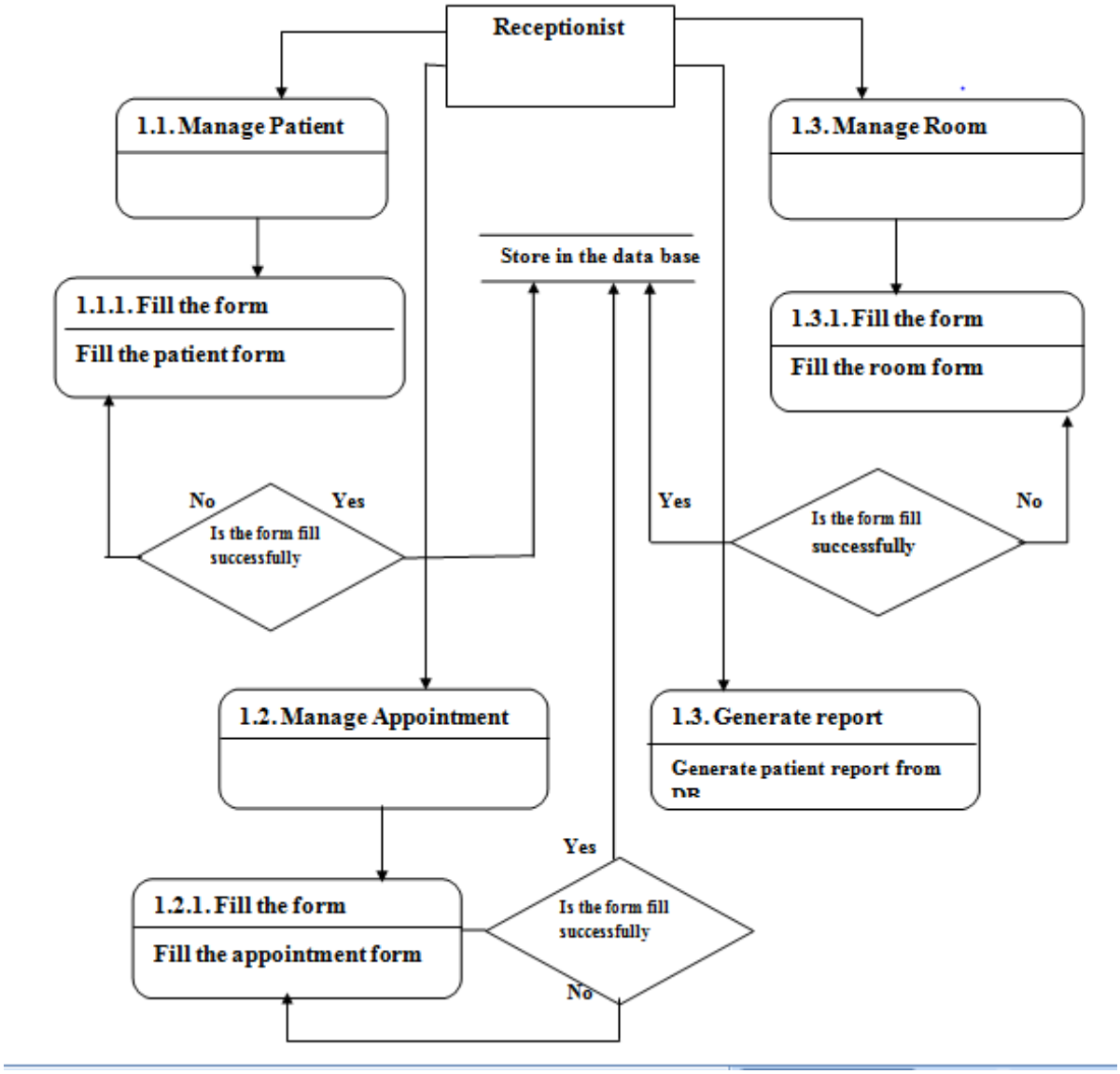


Figure 4.8: Receptionist data flow diagram level 1

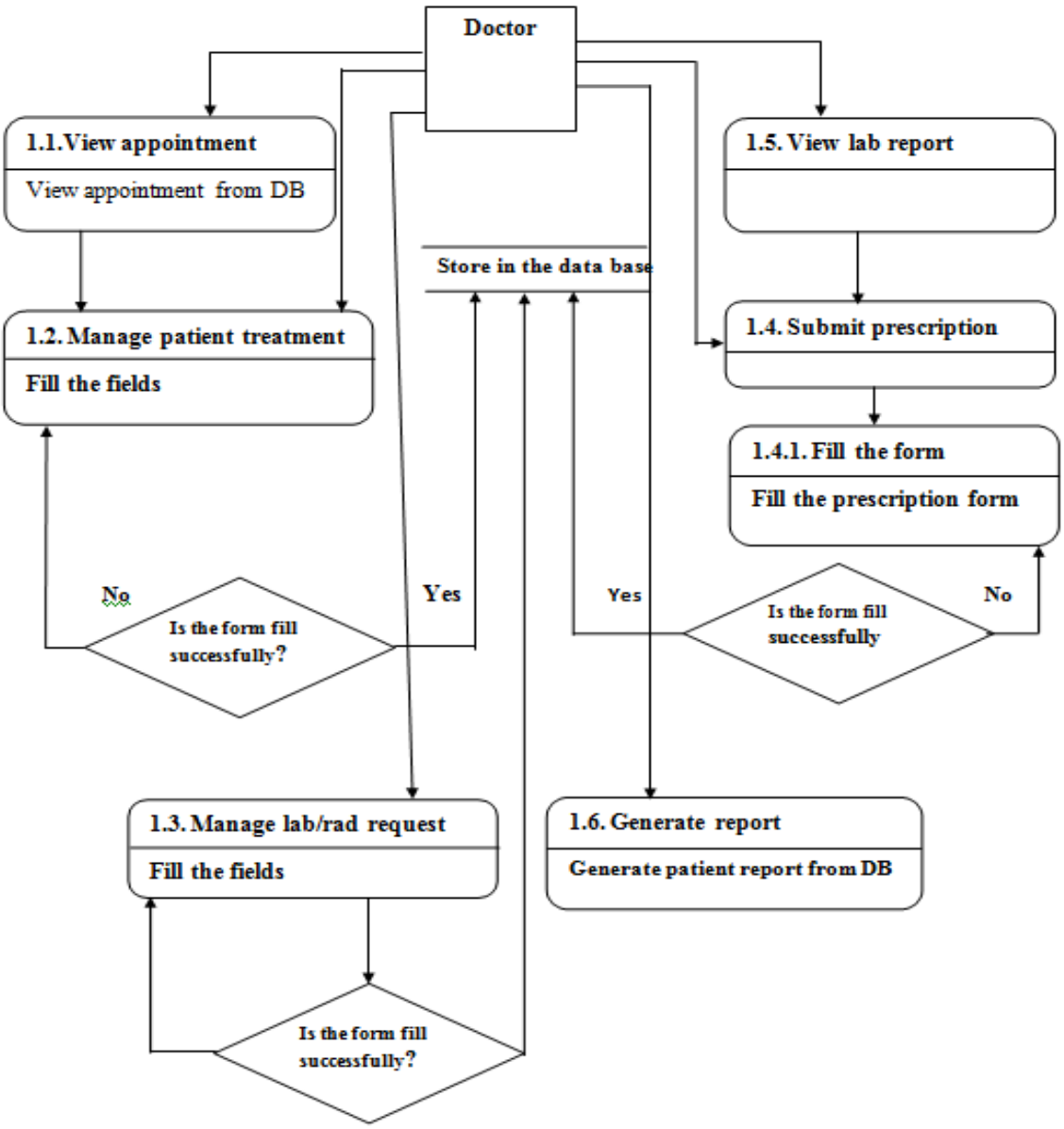


Figure 4.9: Data flow Diagram of Doctor Level 1

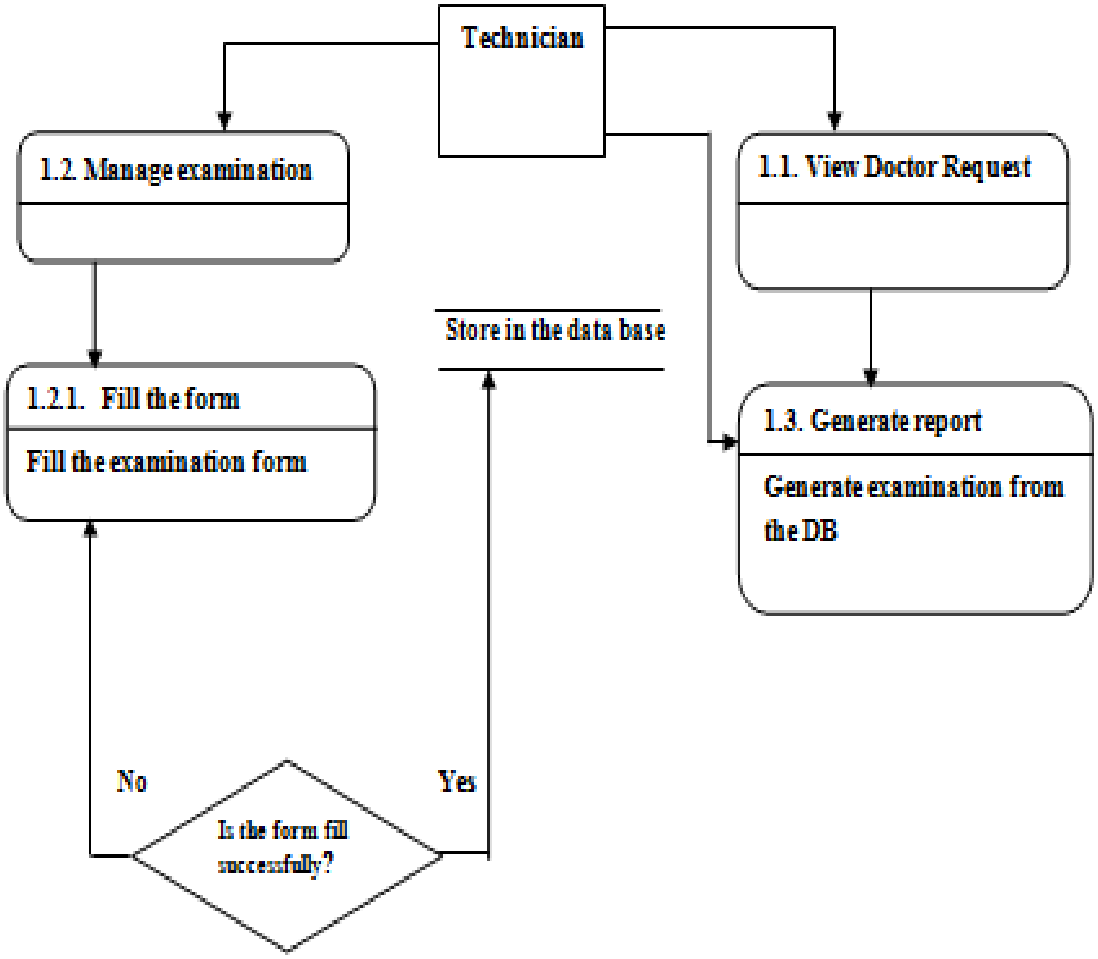


Figure 4.10: Technicians data flow diagram level 1

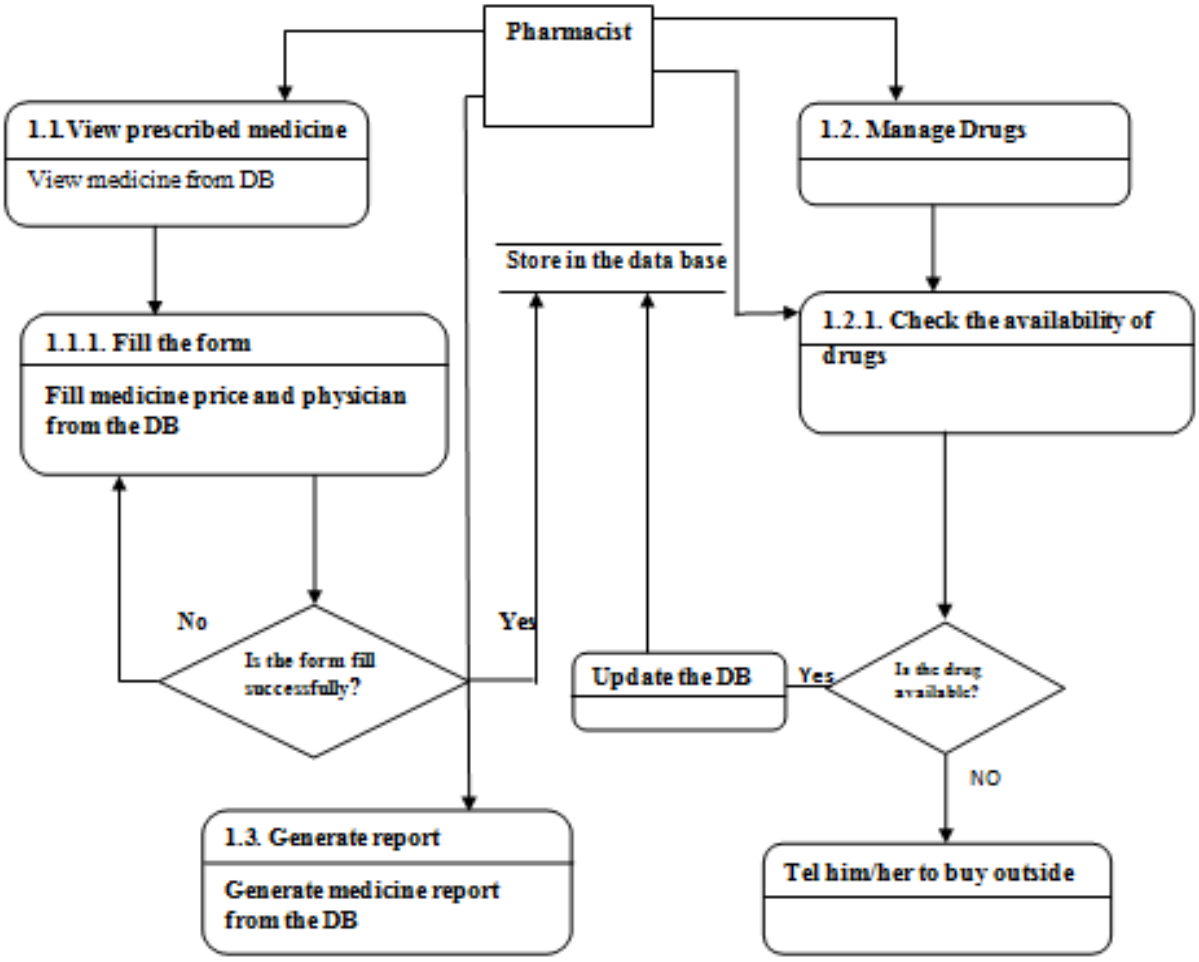


Figure 4.11: Pharmacist data flow diagrams level 1

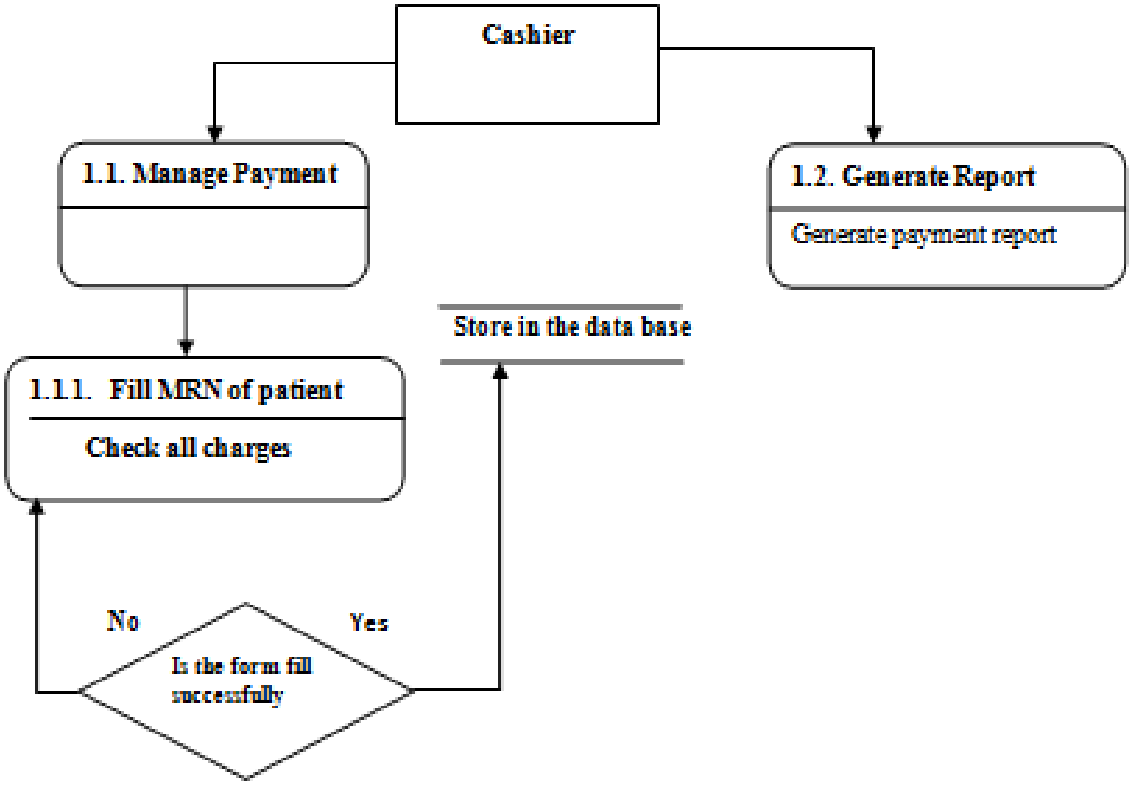


Figure 4.12: Cashiers data flow diagrams level 1

4.3.6. Access control and security

Access control and security describes the user model of the system in terms of an access matrix. Access to protected information must be restricted from users who are unauthorized to access the information. So, in this part we are going to describe who and to which object he/she has the access using access control matrix model. In this model columns represent the actors who has the access to specific object and rows represent classes to be accessed by the authorized actors.

Classes	Actors					
	Administrator	Receptionist	Doctor	Technicians	Pharmacist	Cashier
Staff	Add() view() update() delete()					
User	Add() view() update() delete()					
Patient registration		Add()update() view() delete()				
Rooms		view() update()				
Appointment		Add()view() update()	view()			
Patient treatment			Add() view()			
Lab/rad request			Add() view()	view()		
Examination result			view()	Add() view()		
Prescription			Add() view()		view()	
Drugs					view() update()	
Payment						Add() view()

Table 4.2: Access control and Security

For security of the system our system uses authentication to ensure that the user is the user that has the permission to login to the system. The system also maintains authorization to provide different level of privileges for the users of the system.

4.3.7. User Interface Design

This part describes the way that the graphical user interface is designed. The following figure shows the layout of the GUI.

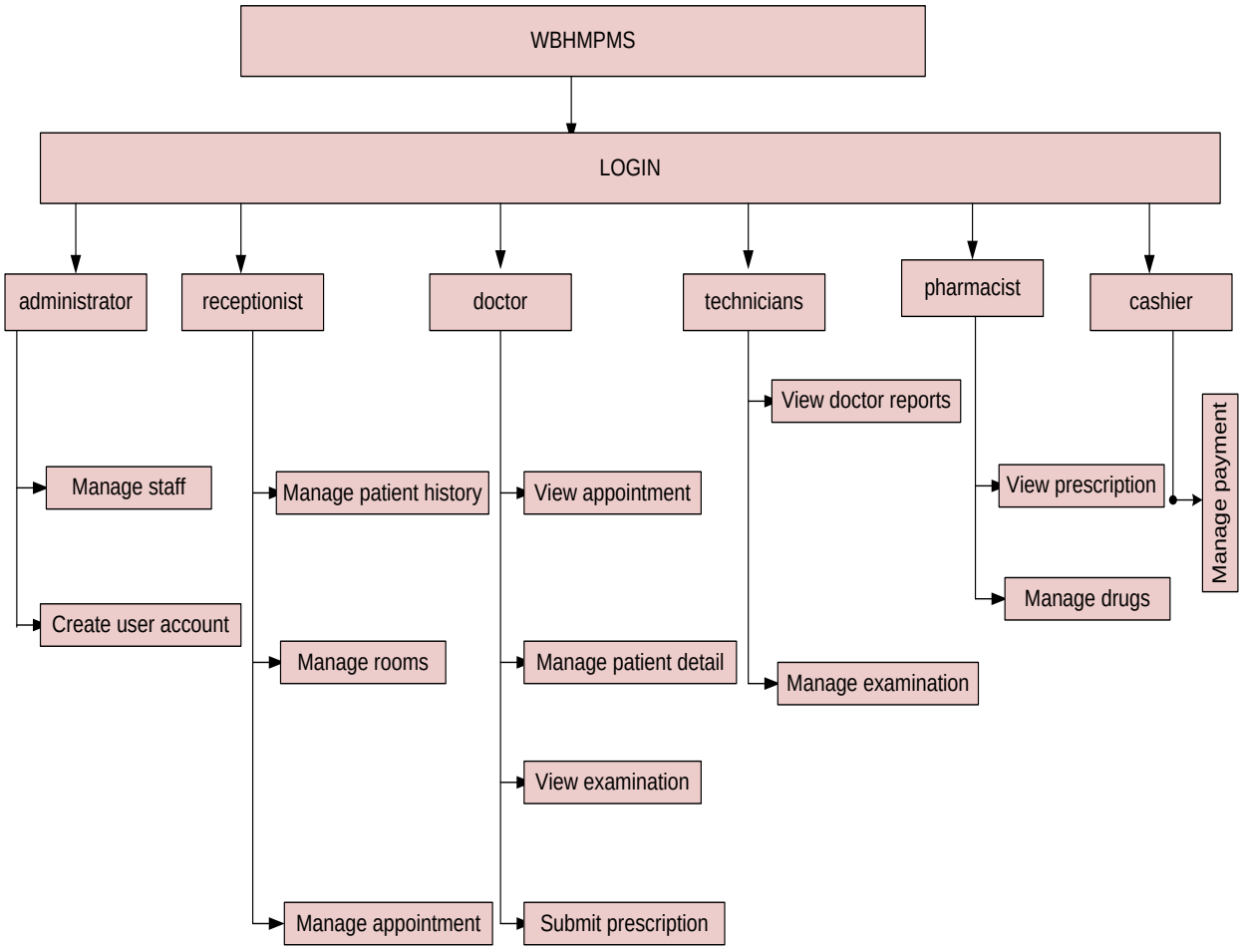


Figure 4.13: User Interface Design

4.3.8. Global software control

Global software control describes how the Web-Based Hospital Medication Management System is implemented in the hospital with a standard security of system in each medical department. In particular, this global software control should describe how requests are initiated and how subsystems synchronize within the secured concurrency issues.

4.3.9. Boundary conditions

Boundary conditions describe the start-up, shutdown, and error behavior of the whole system. Like other application soft-wares this system displays in the desktop.

To start-up the system

- Run WAMP server on the taskbar
- Click local host from the right left menu
- Write the folder name on the URL of the browser like local host/final hospital/index.php
- Insert user name and password of each role

To shutdown the system

- User should logout from the system
- Click the close button of the homepage

To handle the error behavior of the system

- Try to enter wrong input for each field like 500 for age or for email, password and all fields don't accept wrong data out of requirements. Every data entry is validated properly it shows for user status message that indicates it is wrong entry.

4.4. SUBSYSTEM SERVICES

A subsystem is a collection of units of functionalities and describes the services provided by each subsystem in the developed Web-Based Hospital Medication Management System and in each operation performs system decomposition. Sub system service is specified by subsystem interface specifies interaction and information flow from/to subsystem boundaries should be well defined and small subsystem interfaces are refined in object design often called Application Programmer's Interface (API), but this term should be used during implementation, rather than system design.

The responsibility of subsystem service is as follow:

1. Administrator: register staffs, create user accounts, and generate report.
2. Receptionist: performs patient registration, appointment, admission discharge register and manages rooms.
3. Doctors: performs view appointment, manage admission summary, progress note, physician order sheet, request and view Laboratory/radiology, prescription and manage discharge summary
4. Technician: view doctor request and manage Laboratory/radiology examination.
5. Pharmacist: View prescribed medicine and checks the availability of the prescribed drug in his or her data base (manage drugs).
6. Cashier: manage payment by checking patient charges/fees.

4.5. Deployment diagram

It describes the physical architecture of the hardware and software I the system. They depict the software components, processors and device that makeup the systems architecture.

A deployment modeling depicts a static view the run-time configuration of processing nodes and components that run on the theses nodes. It shows:

I. The hardware for the system
II. The software that is installed on the software
III. Depicts the hardware/network infrastructure of an organization
IV. Depict a major deployment configuration of a business application.

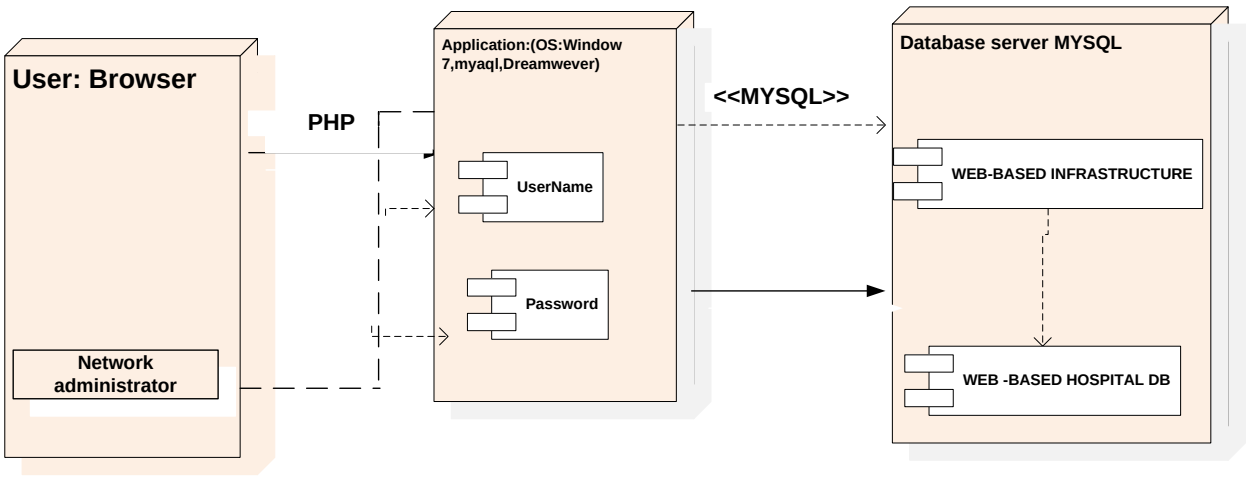


Figure 4.14: Deployment Diagram (Login page)

As can be seen on the above figure we tried to identify the hardware and software requirements of the system i.e. when the user access the system through the user interface and communicate with the system it needs a friendly user interface, for our case we use we use PHP as a front end and MYSQL as a backend and we connect them using database connection. The software for the interface side or Application program with operating system windows 7 and MYSQL, PHP and user directly interact with the system through the PHP scripting language.

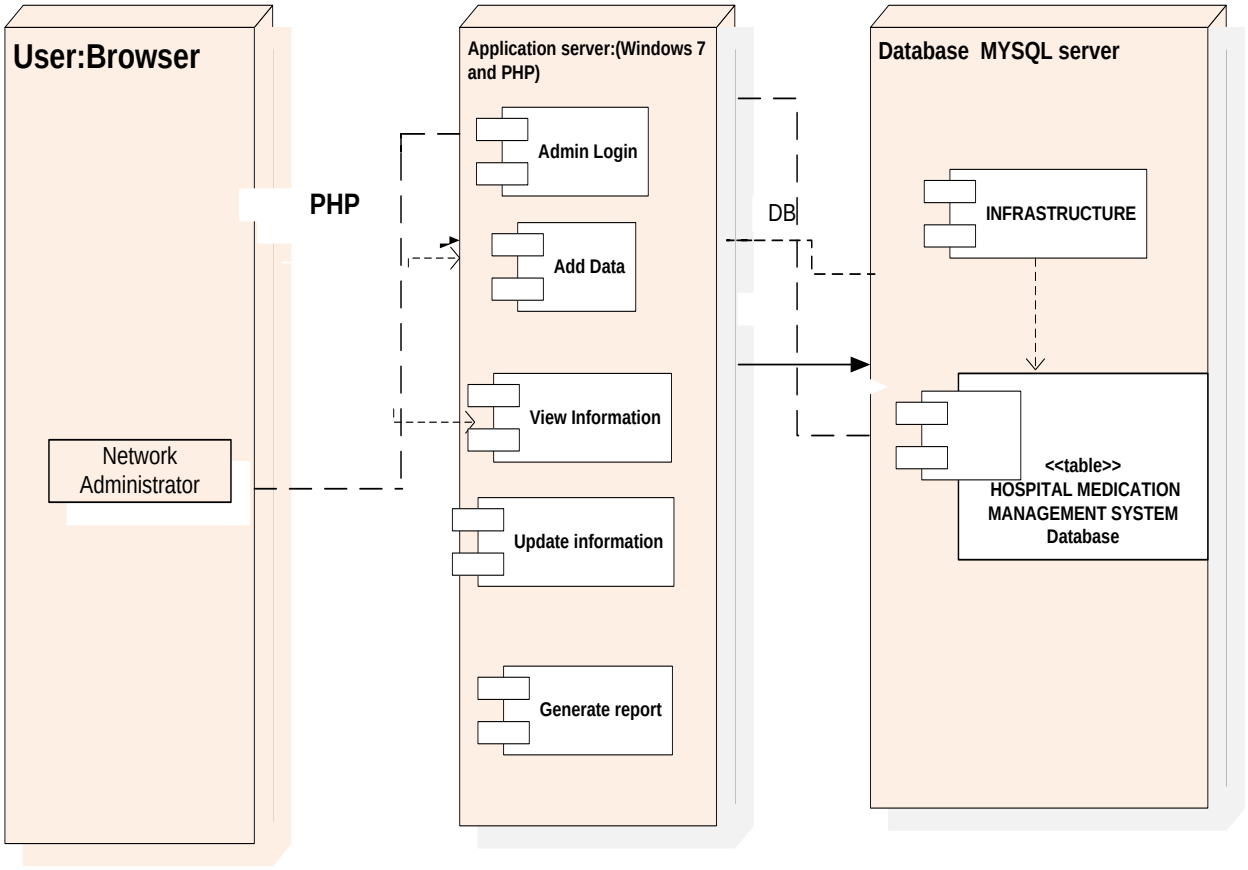


Figure 4.15: The user interaction Deployment diagram (Main menu/homepage)

CHAPTER FIVE

IMPLEMENTATION AND RESULT

After the design is completed the next stage in software development process is implementation. Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, and algorithm.

5.1. ALGORITHM USED

In its most general sense an algorithm is any set detailed instruction which results in a predictable end state from a known beginning. Algorithms are only as good as the instructions given, however, and the result will be incorrect if the algorithm is not properly defined.

We use the pseudo code algorithm for our project because pseudo code consists of short English phrases used to explain specific tasks within a program's algorithm. Pseudo code should not include keywords in any specific computer languages. It should be written as a list of consecutive phrases. Pseudo code is a combination of programming terminology and plain English to describe algorithm.

1. **Login algorithm using pseudo code**

Read username

Read password

If “Role” is equal to “Administrator” then

Display Administrator page

Endif

Else If

“Role” is equal to “receptionist” then

Display receptionist page

Endif

Else If

"Role" is equal to "doctor" then

Display doctor page

Endif

Else If

"Role" is equal to "technician" then

Display technician page

Endif

Else If

"Role" is equal to "pharmacist" then

Display pharmacist page

Endif

Else If

"Role" is equal to "cashier" then

Display cashier page

Endif

Else

Display error message "Enter correct username and password from the role given"

Endif

2. Staff registration algorithm using pseudo code

Read username, password

If “Role” is equal to “administrator” then

Read SID, Fname, Lname, Age, Sex, Address, Email, Qualification,Designation

Write SID, Fname, Lname, Age, Sex, Address, Email, Qualification,Designation

Else

Display another role page

Endif

3. Patient registration algorithm using pseudo code

Read username, password

If “Role” is equal to “receptionist” then

Read MRN, Fname, Lname, Age, Sex, wereda,kebel,HNo.

Write MRN, Fname, Lname, Age, Sex, wereda,kebel,HNo.

Else

Display another role page

Endif

4. Appointment scheduling algorithm using pseudo code

Read username, password

If “Role” is equal to “receptionist” then

Read MRN, Date. Time,DoctorID

Write MRN, Date. Time,DoctorID

Else

Display another role page

Endif

5. Prescription algorithm using pseudo code

Read username, password

If “Role” is equal to “doctor” then

Read MRN, Medicine, DoctorID

Write MRN, Medicine, DoctorID

Else

Display another role page

Endif

6. payment algorithm using pseudo code

Read username, password

If “Role” is equal to “cashier” then

Read MRN, car_Fee, Lab_fee, Rad_Fee, Treatment_Fee, medicine_Fee, room_fee

Write MRN, car_Fee, Lab_fee, Rad_Fee, Treatment_Fee, medicine_Fee, room_fee

Else

Display another role page

Endif

5.2. METHODS USED

Once a new software product has been produced it must be installed and then implemented on site. There are a number of ways to introduce a new system and each of these can suit different situations. Usually implementation will involve conversion from an old system that the new system is designed to replace, so often those methods are known as methods of conversion. The four common methods of implementation and/or conversions are:

- Direct-cut over method: this method involves the old system being completely dropped and the new system being completed implemented at the same time. The old system is no longer available. As a consequence, you must be absolutely sure that the new system is totally functional and operational. This conversion method is used when it is not feasible to continue to operating two systems together. Any data to be used in the new system must be converted and imported from the old system. In this case users must be fully trained in the operation of the system before the conversion takes place. It is a risky approach.
- Parallel method: The parallel method of implementation (or conversion) involves operating both systems together for period of time. This allows any major problems with the new system to be encountered without the loss of data. Parallel conversion also, means that users have to familiarize themselves with the new system. In essence, the old system remains operational as a backup to the new system. Once the new system is found to be meeting requirements then operation of the old system can stop. The parallel method involves double the workload for users as all functions must be performed on both the old and the new system. This strategy is less risky but more expensive because of the following:
    - ✓ Manual results can be compared with the results of the automated system
    - ✓ The operational work is doubled
    - ✓ Failure of the computerized system at the early stage does not affect to the working of the organization, because the manual system continues to work as, it used to do
- Phased methods: The phased method of implementation (or converting) from the old system to the new system involves a gradual introduction of the new system and whilst the old system is progressively discarded. This can be achieved by introducing new parts of the new product one at a time while the older parts being replaced are removed. Completed modules are released to customers as they become available. Parts of the total

system are introduced systematically across the business, each part replacing a component of the old system. Over time the complete system will be converted.

- Pilot method: In the pilot method of conversion the new system is installed for a small number of users. These users learn, use and evaluate the new system. Once the new system is deemed to be performing correctly then, the system is installed and used by all the pilot method also allows a base of users to learn the new system. These users can then assist with the training of others once the new system had been fully implemented. The pilot conversion method can be viewed as the final testing. Both the developers and the customer are able to evaluate the product in an operational environment prior to its implementation.

Therefore in our case the method that we choose to implement the system is phased method because gradual introduction of the new system can benefit the hospital and staffs whilst the old system is progressively discarded. Parts of the total system are introduced systematically across the business, each part replacing a component of the old system. Over time the complete system will be converted.

5.3. SAMPLE INPUT/OUTPUT SCREEN SHOOTS

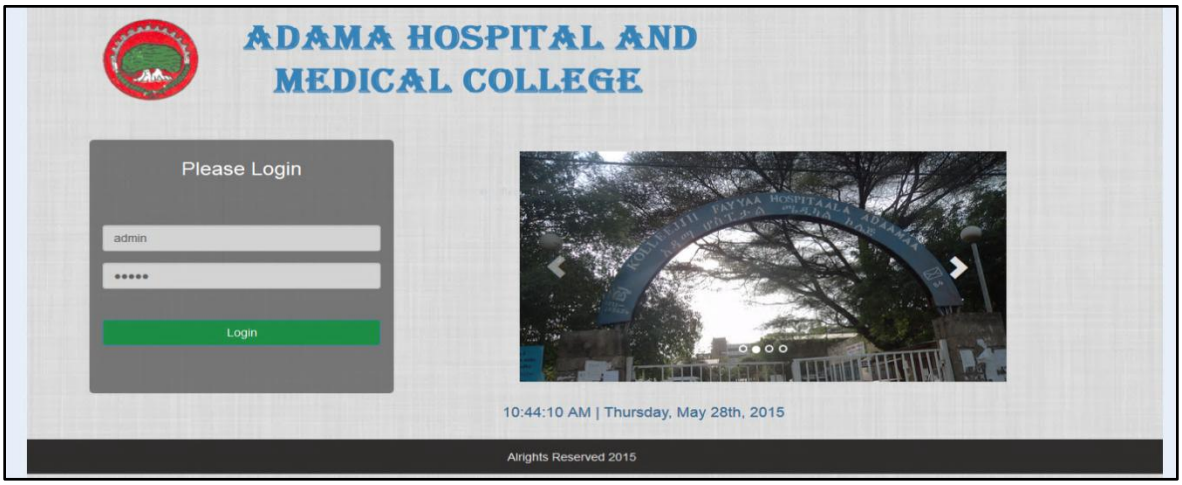


Figure 5.1: Login page

The above login form is used to insert username and password of users, when the username and password is correct the system redirect to the user page otherwise it displays an error page.

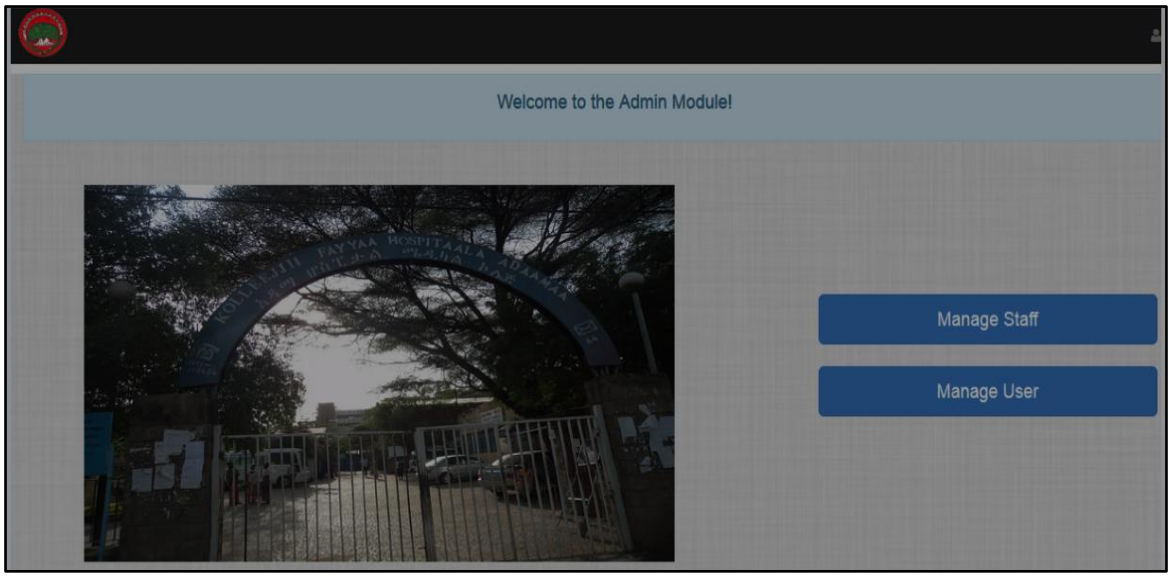


Figure 5.2: Admin page

When the administrator inserts correct username and password the system displays admin page as displayed above. The administrator page has two links manage staff and manage user accounts.

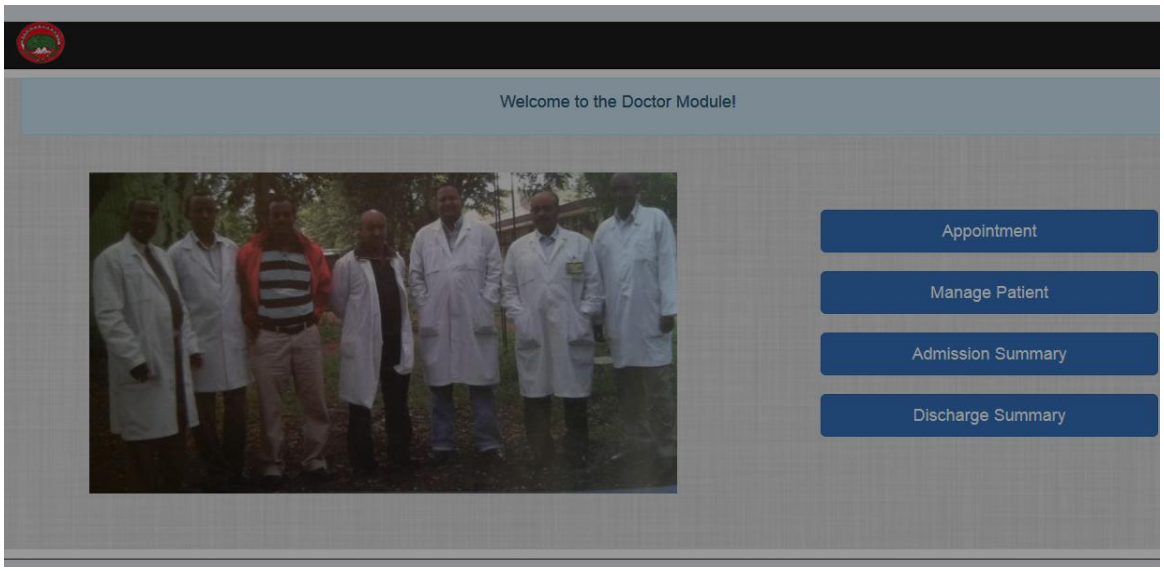


Figure 5.3: Doctor's page

After inserting the correct username and password the system displays the doctor page, which consist four links. Next the doctor clicks the appointment link to check if there is a new appointed patient if so he/she goes to the next links step by step.

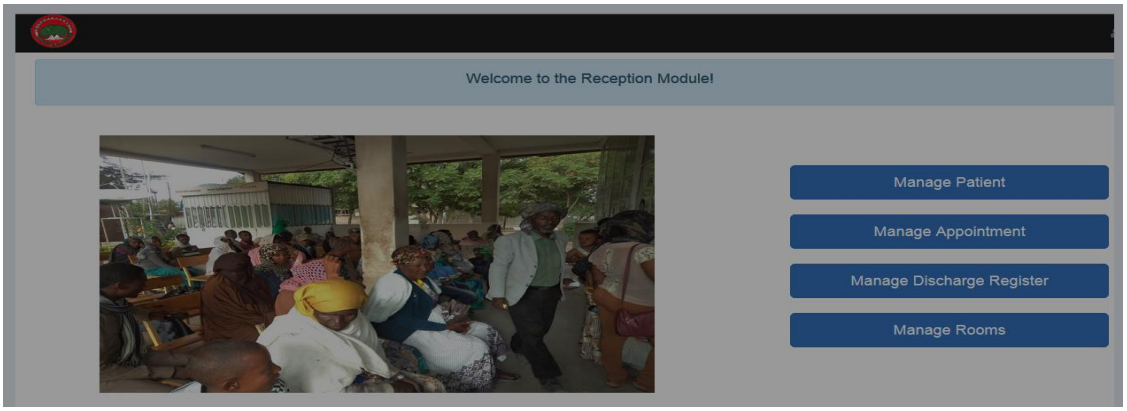


Figure 5.4: Receptionist page

After inserting the correct username and password the system displays the receptionist page, which consists four links. Then the receptionist clicks manage patient link if new patient comes to record his/her details after that the receptionist allots doctor by click appointment link. If the doctor orders the patient to take room the receptionist arranges room by clicking manage room. Finally he/she prepares a patient discharge/register report.

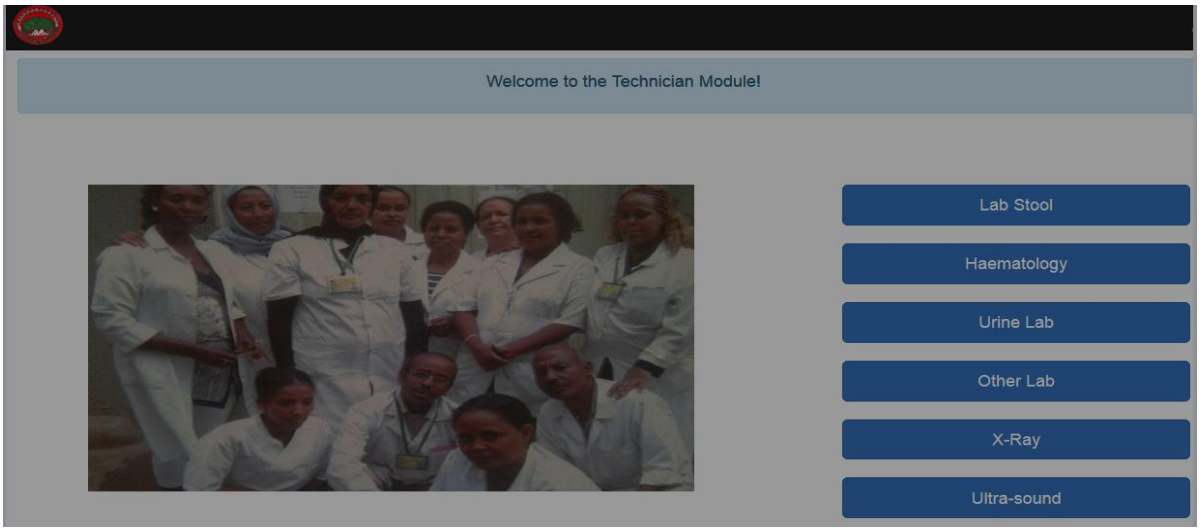


Figure 5.5: Technician page

After inserting the correct username and password the system displays the technician page. This page consist both the laboratory's (urine, hematology, stool and other links) and radiologist (x-ray and ultrasound links). Technicians see each doctor requests and put the results by examine the patient.

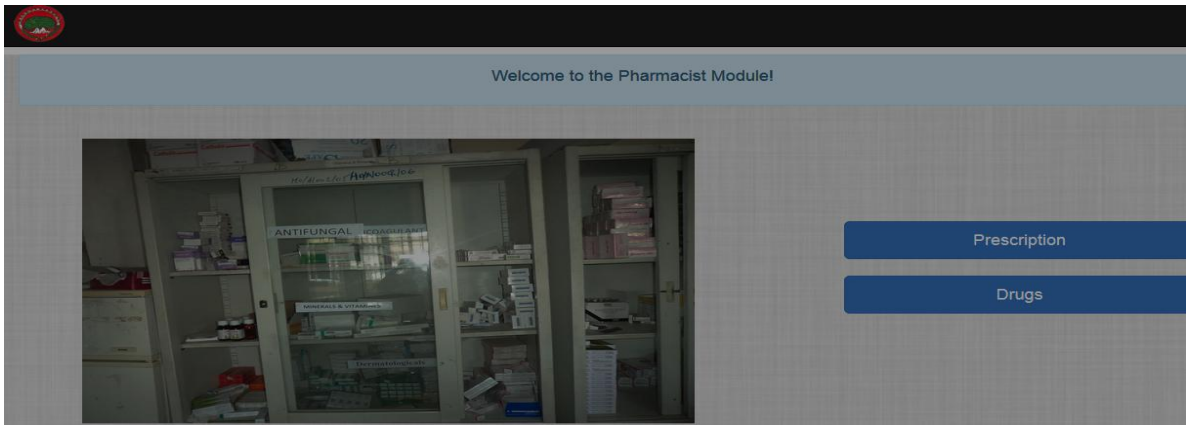


Figure 5.6: Pharmacist page

After inserting the correct username and password the system displays the pharmacist page. This page consist two links prescription and drugs. The pharmacist checks the availability of drugs in his/her database then if drug is not available he/she informs the patient to let him go to another pharmacy outside of the hospital.

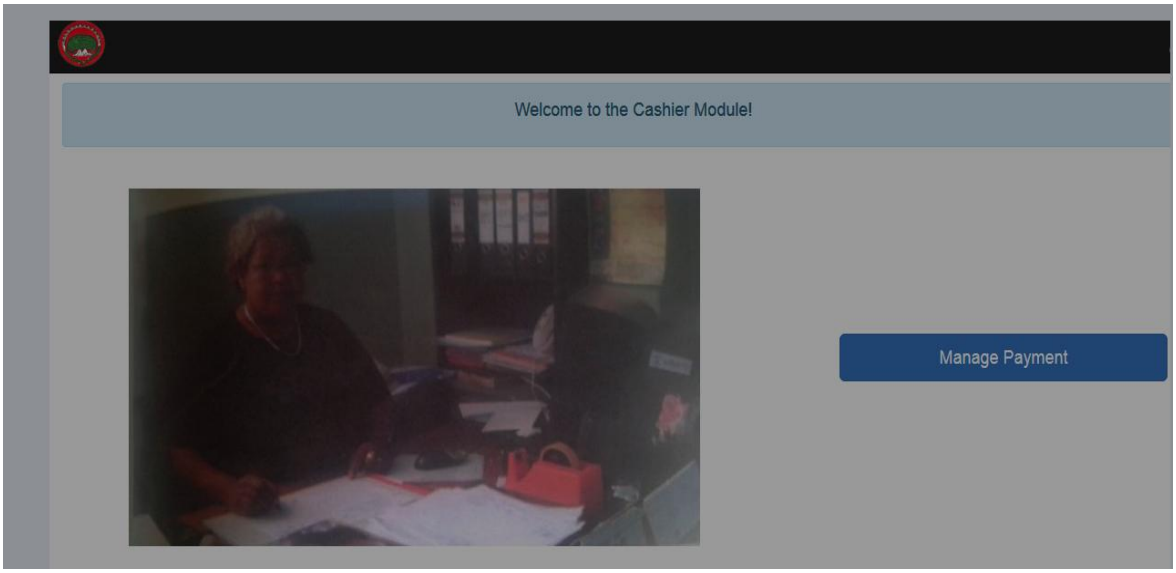


Figure 5.7: cashier page

After inserting the correct username and password the system displays the cashier page. This page has one link only which is the manage payment link.

5.4. SAMPLE CODES

Code to create a connection the front with the backend

```
$link = mysqli_connect(‘hostname', ‘username', ‘password');

mysqli_select_db($link,'hospitaldb');

mysqli_close($link);
```

Code to Retrieving Data from MySQL Database

```
$query = 'SELECT * FROM mytab';

$result = mysqli_query($link,$query)

while ($row = mysqli_fetch_array($result, MYSQL_ASSOC))

{

    echo “$result[‘NAME‘] “;

    echo “$result[‘AGE‘] “;

}
```

Codes to Insert Data to MySQL Database

```
$sql="INSERT INTO mytab
(Sno,Name,Gender,DOB,Hobbies)VALUES($sn,'$fname','$gen','$mydate','$hobi')";

$retval = mysqli_query($conn,$sql);

if(! $retval )

{

die('Could not enter data: ' . mysqli_error($conn));

}

echo “Data Entered Successfully\n";
```

Codes to Updating Data from MySQL Database

```
$sql = update mytab set name='$name', age='$age' where id='$id'"

$ retval =mysqli_query($conn,$sql);

if(! $retval )

{

die('Could not update data: ' . mysqli_error($conn));

}

echo “Data Updated Successfully\n";
```

Codes for deleting Data from MySQL Database

```
$sql = “delete from mytab where id='$id'"

$ retval =mysqli_query($conn,$sql);

if(! $retval )

{

die('Could not delete data: ' . mysqli_error($conn));

}

echo “Data Deleted Successfully\n";
```

Login session

```
<?php

//Start session

session_start();

//Function to sanitize values received from the form. Prevents SQL injection

function clean($str) {
```

```
$str = @trim($str);

If(get_magic_quotes_gpc()) {

$str = stripslashes($str);

}

returnmysql_real_escape_string($str);

            }

//Sanitize the POST values

   $username = clean($_POST['username']);

$password = clean($_POST['password']);

$fname = ($_POST['fname']);

/*$lname = ($_POST['lname']);*/

//Create query

$qry="SELECT * FROM user WHERE username='$username' AND
password='$password'";

$result=mysql_query($qry);

If($result) {

if(mysql_num_rows($result) > 0 ) {

//Login Successful

session_regenerate_id();

$member = mysql_fetch_assoc($result);

$_SESSION['username'] = $member['username'];

$_SESSION['password'] = $member['password'];
```

```
$_SESSION['fname'] = $member['fname'];

//$_SESSION['lname'] = $member['lname'];

session_write_close();

header("location: homepage.php?");

exit();

    }

else {

//Login failed

Header("location: login_error.php");

Exit();

}

 }

else

{

Die("Query failed"); }

?>
```

Check username and password

```
<?php

//$id=$_GET['id'];

$query=mysqli_query($conn, "SELECT SNo, username, password, SID FROM user")or die(mysqli_error());

While($row=mysqli_fetch_array($query)){
```

```
$id = $row['SNo'];
?>
```

Session Methods

```
<?php
session_start();
if (!isset($_SESSION['username']) || ($_SESSION['password'] == '')) {
    header("location:index.php");
    exit();
}
?>
function clean($str) {
        $str = @trim($str);
        If(get_magic_quotes_gpc()) {
            $str = stripslashes($str);
        }
        returnmysql_real_escape_string($str);
    }
```

5.5. RESULTS

In this system each actor such as (Administrator, receptionist, Doctor, technician, pharmacist and cashier) is authorized having their own username and password to login to the system otherwise they don't have a privilege of login and using the system.

5.5.1. Administrator

After administrator login to his/her page s/he will get the manage staff manage user account and generate user report functions

Mange staff

The administrator would register all staff details and can view all lists of staffs registered in the system and then after s/he can modify and delete staffs as necessary.

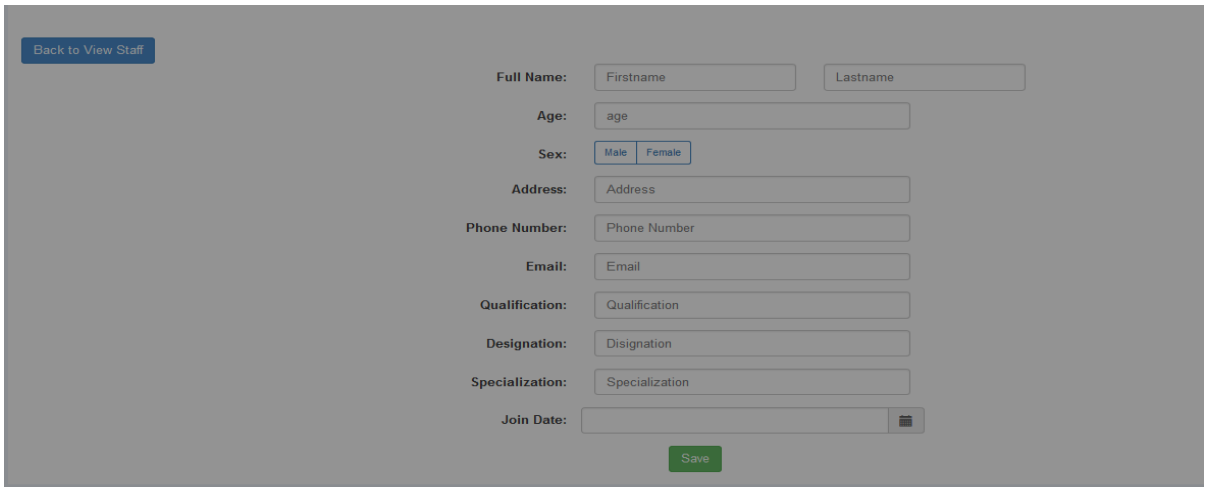


Figure 5.8 Add staff form

Figure 5.9 View staff form

Manage user account

After administrator creates the user account of each user, staffs are logging to the system

Figure 5.10: add user form

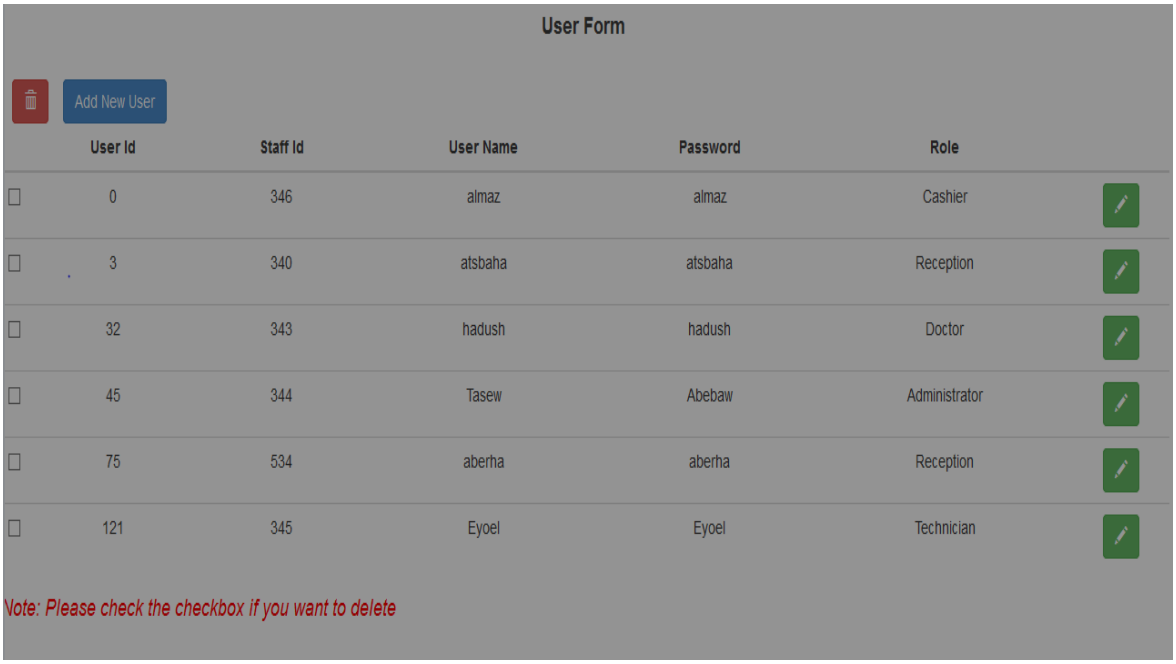


Figure 5.11: View user form

5.5.2. Receptionist

After receptionist login to his/her page s/he will get the manage patient, manage appointment and generate patient report functions

Manage patient

A result that shows the list of registered and admitted patients. It is also possible to modify and delete for staffs when necessary.

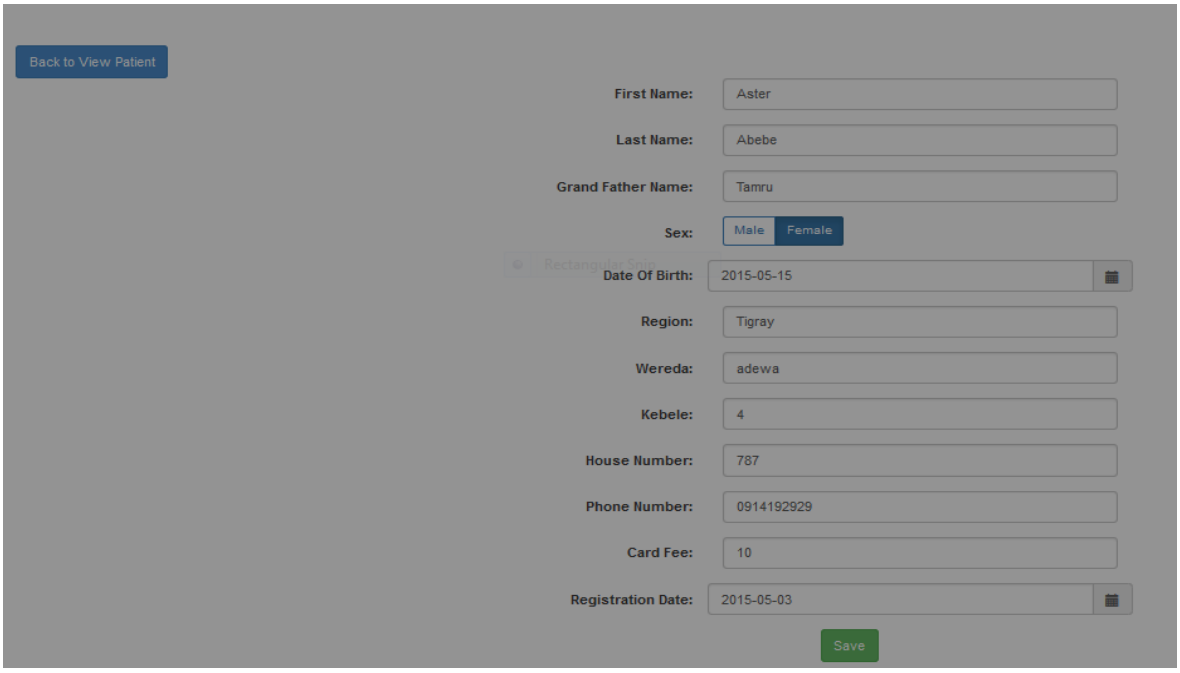


Figure 5.12: Add patient detail form

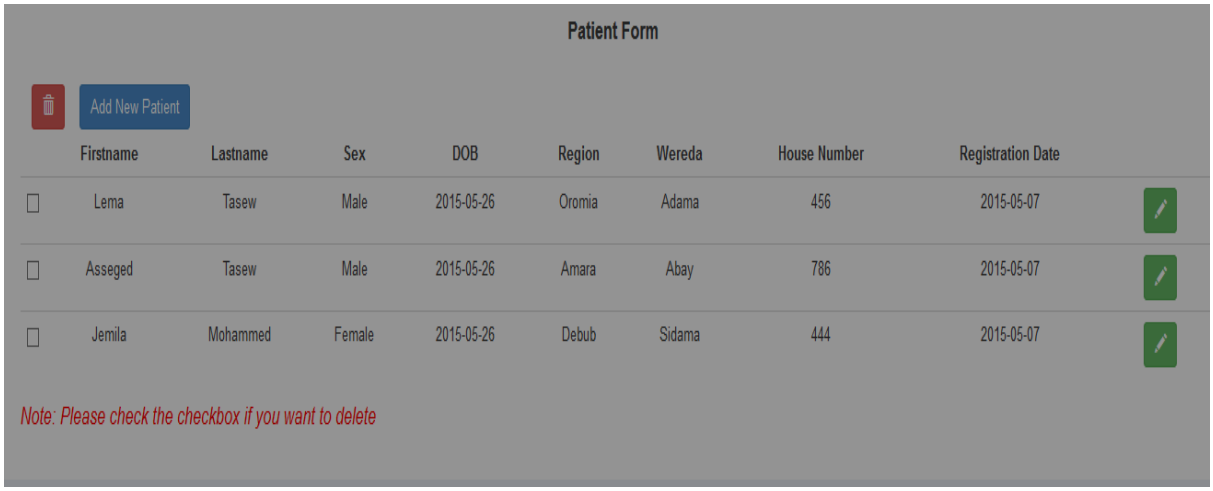


Figure 5.13: View patients list form

Manage Appointment

After receptionist fill and add appointment for doctors, list of appointments can be shown as shown below and the receptionist can modify the appointments made earlier.

Figure 5.14: add appointment form

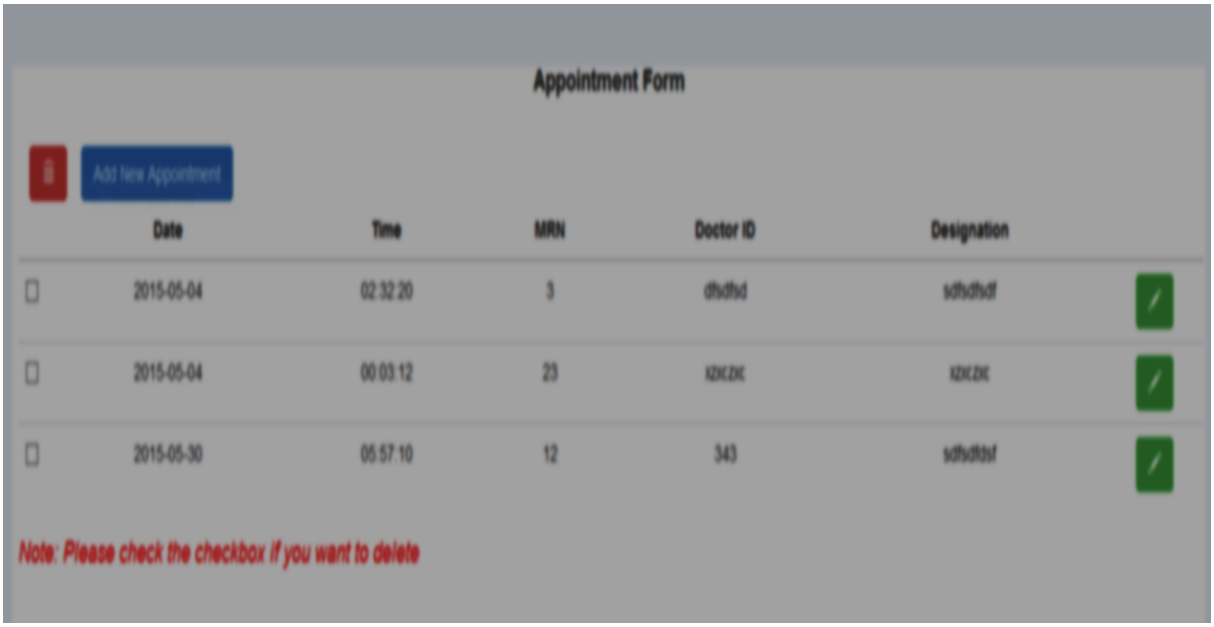


Figure 5.15: View appointment lists form

5.5.3. Doctor

After doctor login to his page, s/he can view the result of appointment, patient detail and examination result.

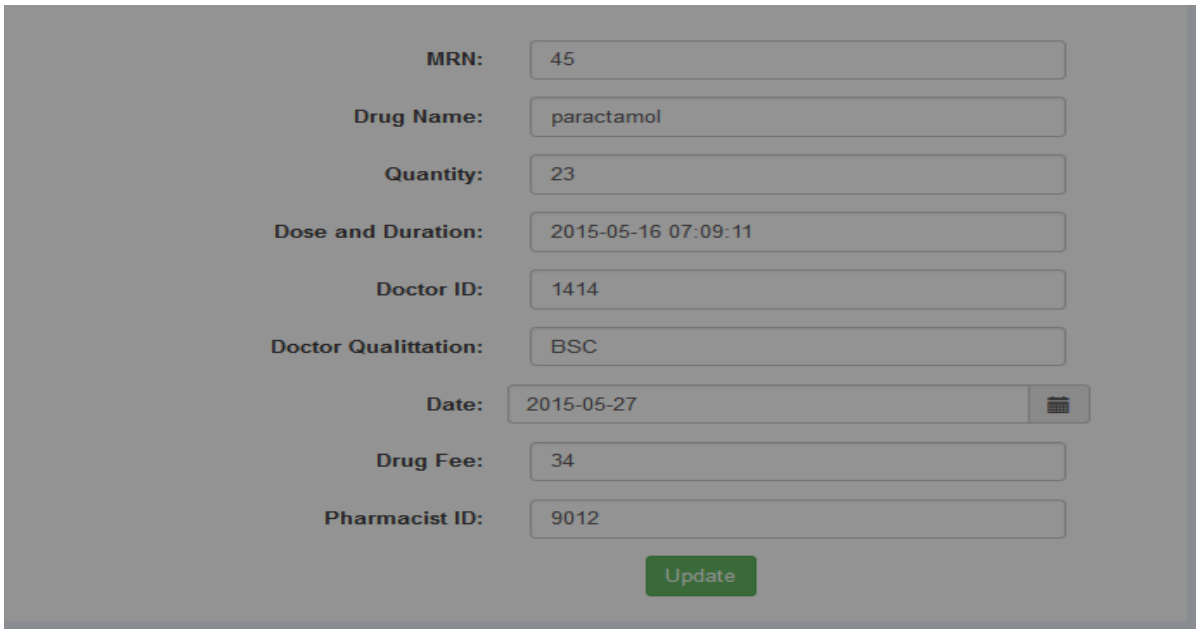


Figure 5.16 add and Update prescription form

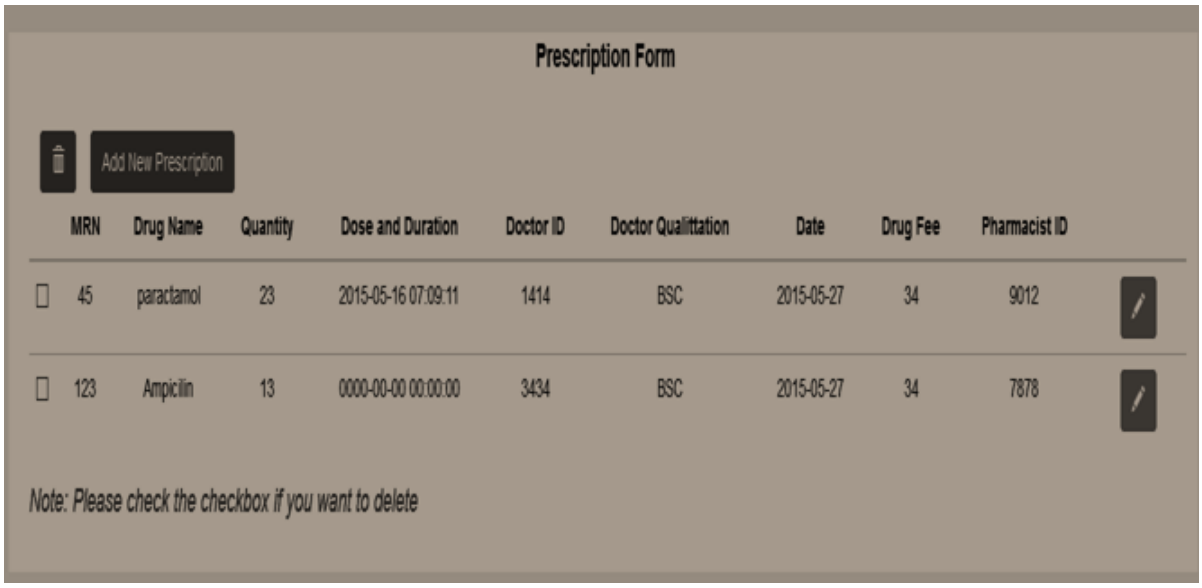


Figure 5.17 View prescription form

5.5.4. Technician

After technician login to his/her page s/he can view list of doctors lab request, manage examination, and generate report

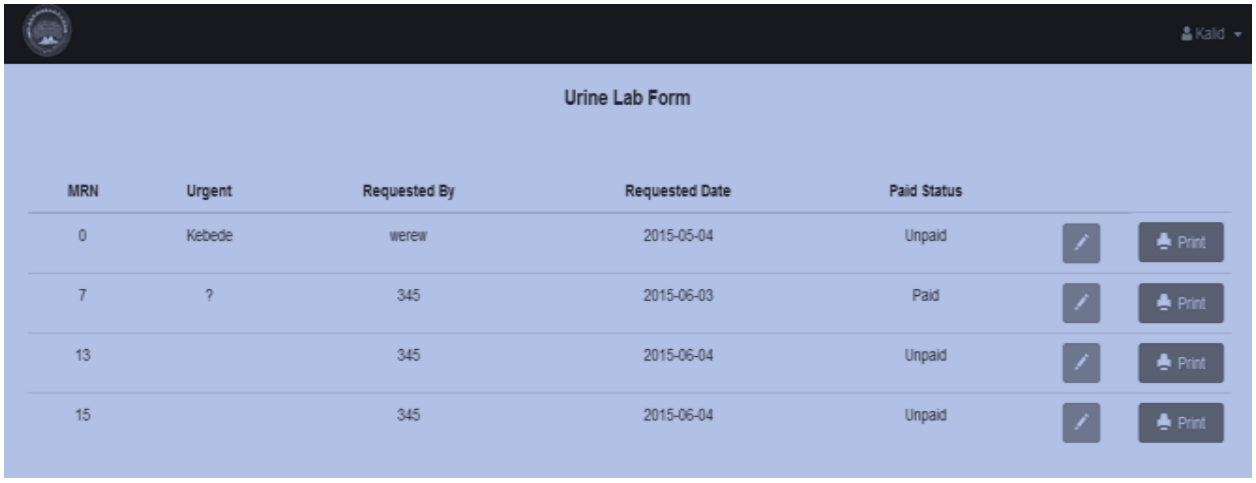


Figure 5.18: Urine lab request lists form

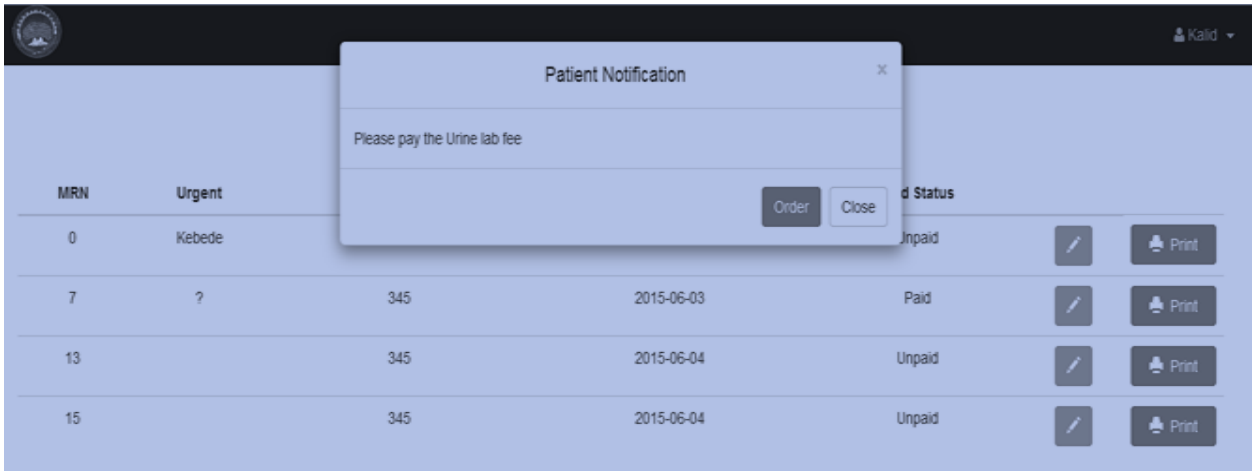


Figure 5.19: Add result

5.5.5. Pharmacist

After Pharmacist login to his/her page s/he can view list of prescribed medicine, manage drugs

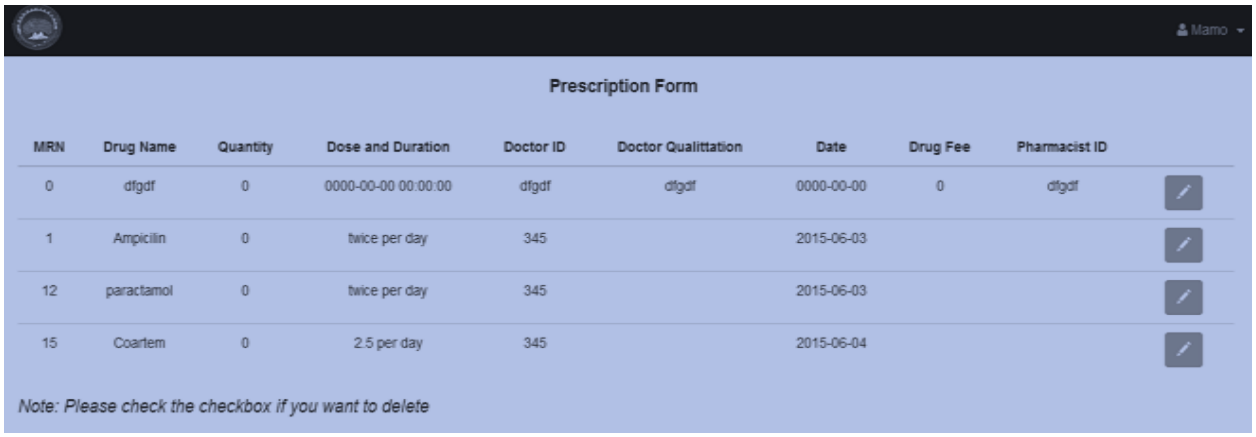


Figure 5.20: View prescription form

Figure 5.21: lists of drugs form

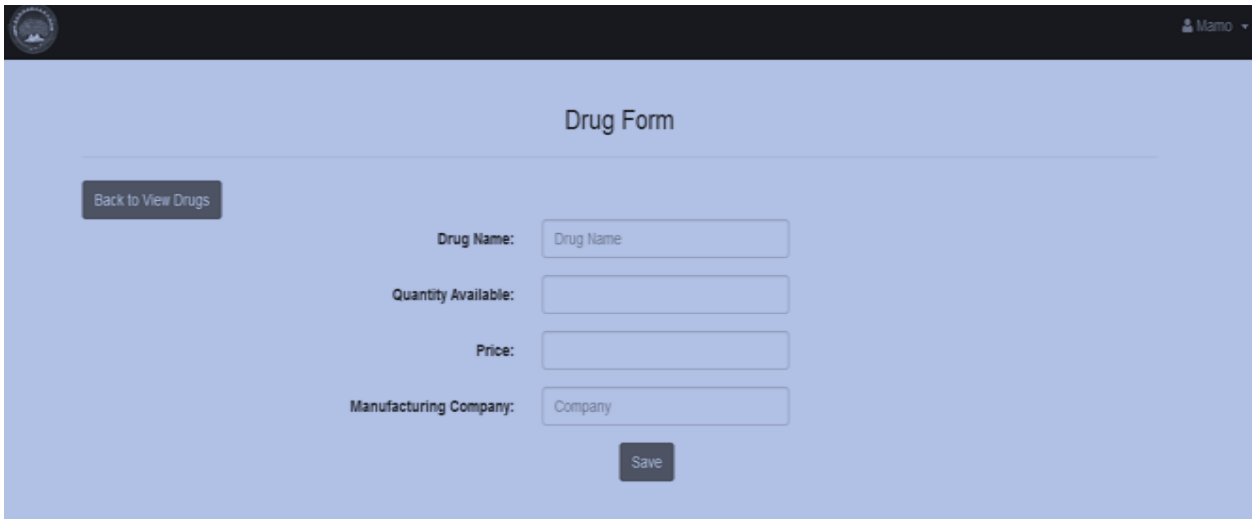


Figure 5.22: add/update drugs form

5.5.6. Cashier

After Cashier login to his/her page s/he can view list of payment

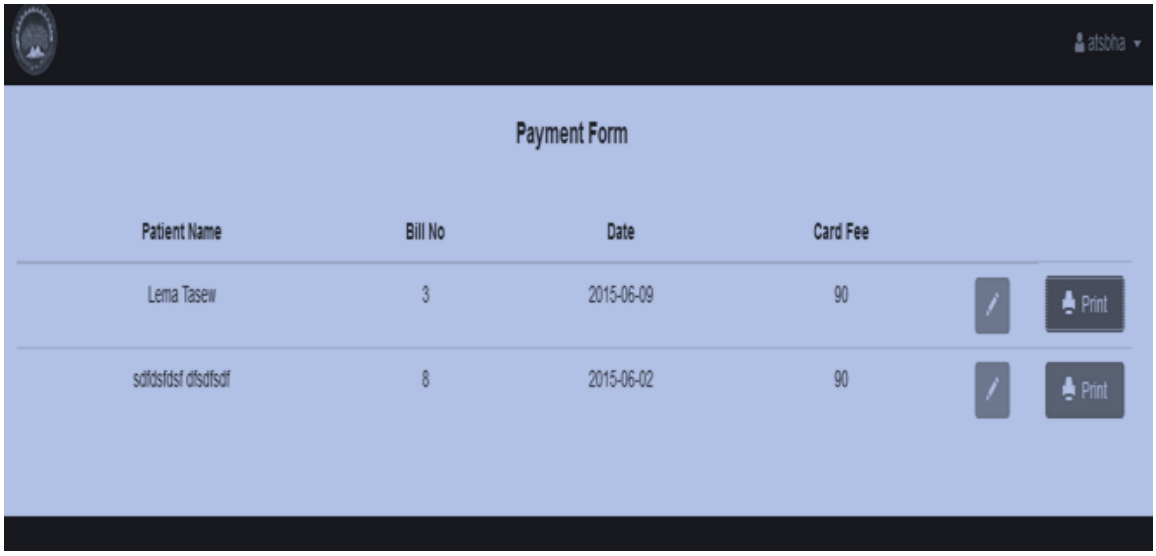


Figure 5.23: view payment form

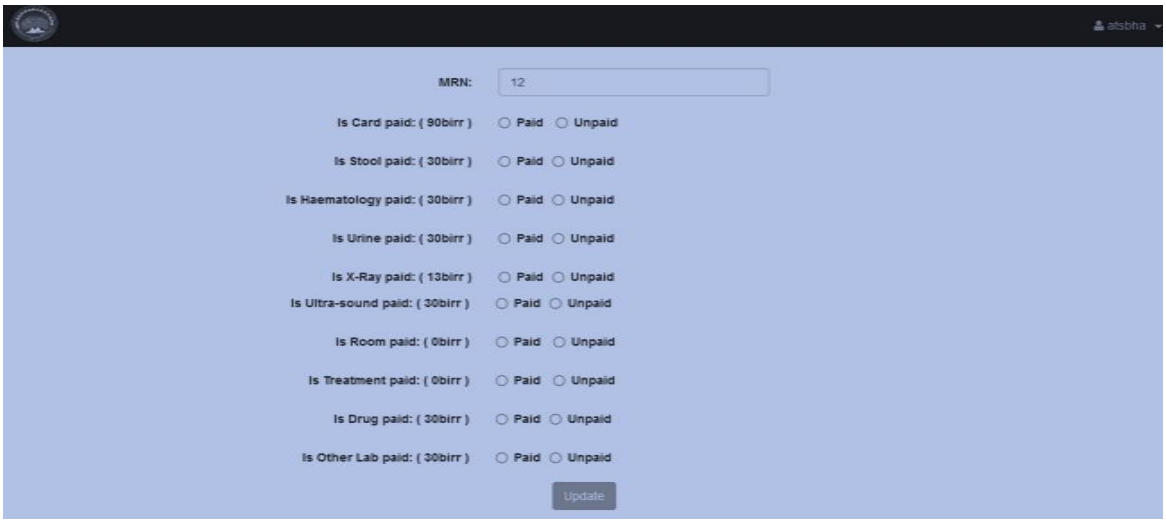


Figure 5.24: add/update payment form

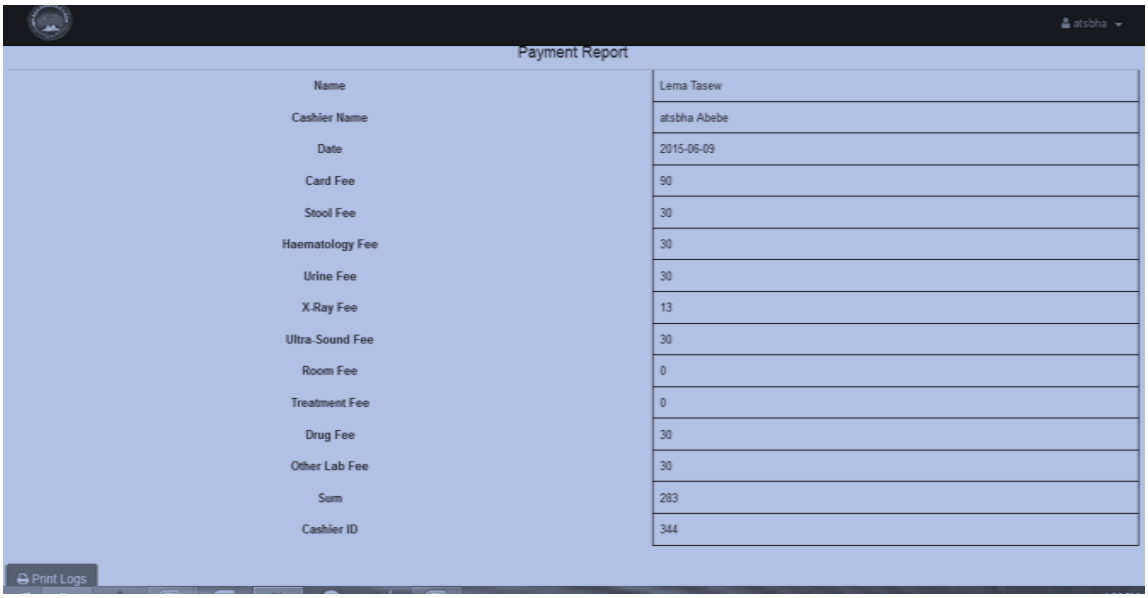


Figure 5.25: generate payment report form

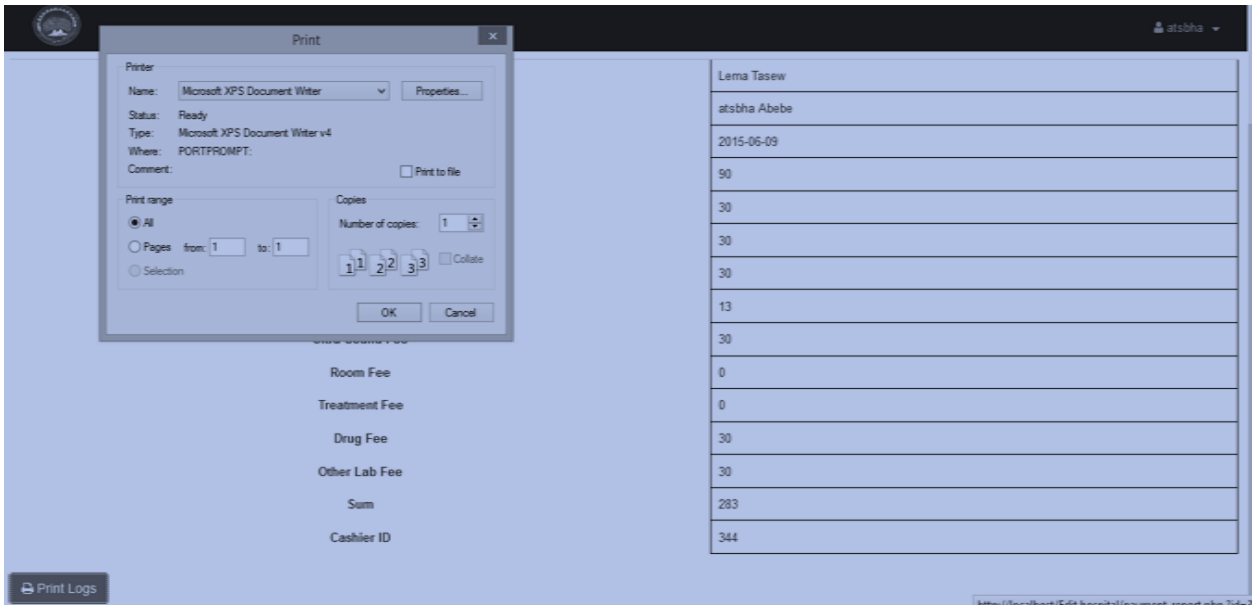


Figure 5.26: print payment report

CHAPTER SIX
SYSTEM TESTING

6.1. TEST PLAN

6.1.1. Introduction

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System testing is an important phase and it represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing. A good test case is one that has a high probability of finding an as undiscovered error.

6.1.2. Test plan overview

Careful test planning reduces the cost of testing and the elapsed time necessary for its completion. Two key elements are to start the selection of test cases early and to parallelize tests.

The functional tests that can be developed when the use cases we have selected are completed and the unit tests of the subsystems of each use case can be developed when their interfaces is defined.

The second key element in shortening testing time is to parallelize testing activities: All component tests can be conducted in parallel; double tests for components in which no faults were discovered can be initiated while other components are repaired.

Test configuration includes test plan, test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

6.1.3. Features to be tested/not to be tested

Features to be tested:

The different functions that are included in our project are to be tested separately. The features to be tested are Login, Manage staff, Manage user Account, Manage Patient Detail, Manage Appointment, Submit Prescription, Mange Examination, Manage Drugs, Manage Payment and Generate Report.

Features not to be tested:

We do not have features not to be tested

6.1.4. Test approaches

We have used two different categories of test case design techniques. These are:

1. White-box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed. It focuses on internal program structure, and discovers all internal program errors.

2. Black-box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Levels of testing

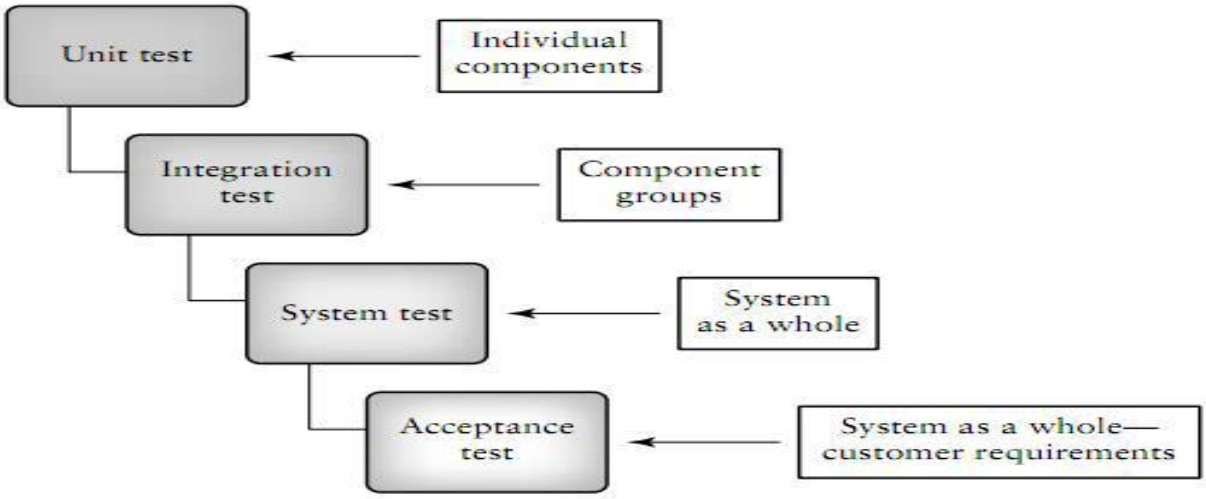


Figure 6.1 Level of testing

Information needed at different levels of testing

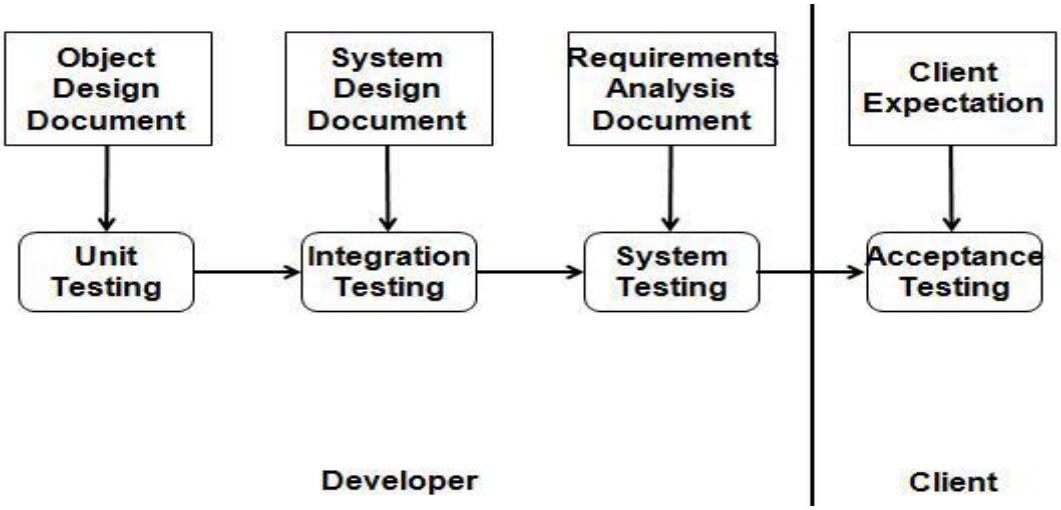


Figure 6.2 Information needed at different levels of testing

Unit testing:

A unit is the smallest testable part of software. In procedural programming a unit may be an individual program, function, procedure, etc. In an OOP, the smallest unit is a method. Unit testing is often neglected but it is, in fact, the most important level of testing. It is performed by using the method White Box testing. It is the first level of testing and is performed prior to integration testing.

This testing method is performed in certain modules by breaking them into different sub-modules. These are staff registration, create user account, login, manage treatment, manage examination, manage prescription, manage drugs and manage payment.

Test case name: Staff registration

SID	FirstName	LastName	Age	Sex	Address	Phone No.	Email	Qualification	Designation	Specialization	Join date	Validation
030	Gutta	Kumsa	35	M	K04	0912111111	guttak07@gmail.com	Diploma	Receptionist		2015-02-27	Registered
031	Amanuel	Tesfaye	40	M	K04	0912233456	Aman_07@yahoo.com	BSC	Doctor	Dental	2015-03-27	Registered
032	Asamenew	Teffera	50	M	K04	0913456780	asatefera@gmail.com	BSC	Technician		2015-04-27	Registered
033	Abebaw	Demise	25	M	K01	0912233567	Abede_7@yahoo.com	BSC	Pharmacist		2015-05-27	Registered
034	Senait	Abebe	30	F	K02	0914454540	senuabe@hotmail.com	Diploma	Cashier		2015-05-27	Registered
035	Belay	Gesesse	55	M	K05	0942117780		10+3	Guarde		2015-01-27	Registered

Table 6.1: Registration of staff unit testing

Test case name: create user account, in the table below we try to create user accounts of six different users being as an administrator.

SID	FirstName	LastName	Role	Username	password	Validness
029	Legese	Alemayo	Administrator	Legesea	Administrator	Account created
030	Gutta	Kumsa	Receptionist	Guttak	Gutta123	Account created
031	Amanuel	Tesfaye	Doctor	Amanuelt	Aman123	Account created
032	Asamenew	Teffera	Technician	Asamenewt	Asa123	Account created
098	jemal	Seid	Pharmacist	Jemals	Jemal123	Account created
099	Chaltu	Gemechu	Cashier	Chaltug	Chaltu123	Account created

Table 6.2: Create user account unit testing

Test case name: Login, to test the login use case we have prepared 3 test suits

Username	Password	Validness
Legesea	Admin	Password should not be less than 6 characters
Lemlem	Ha123@	User is not authorized
Jemals	jemal123	Password is not correct

Table 6.3: Login unit testing

Test case name: Submit prescription

MRN	Drug Name	Qty	Dose and Duration	Doctor ID	Qualification	Date	Drug Fee	Pharmacist ID	Validness
01234	Parastamol	21	3 per day	031	BSC	2015/5/31			Accepted
01245		14	3 per day	022	BSC	2015/5/31			Drug name should be filled
01236	Paractamol	21	3 per day	040	BSC				Date should be filled

Table 6.4: Submit prescription

Test case Name: View prescription and fill Drug fee by pharmacist

MRN	Drug Name	Qty	Dose and Duration	Doctor ID	Qualification	Date	Drug Fee (ETB)	Pharmacist ID	Validness
01234	Paractamol	21	3 per day	031	BSC	2015/5/31	30.00	033	Accepted
01245	Coatroom	14	3 per day	022	BSC	2015/5/31	40.00		ID should filled
01236	Ampciline	21	3 per day	040	BSC	2015/5/31	20.00	033	Accepted

Table 6.5: View and submit prescription

Integration testing:

Integration Testing is a level of the software testing process where individual units are combined and tested as a group. Integration testing tests interface between components, interaction to different parts of system. When two or more units are ready, they are assembled and Integration Testing is performed. Any of Black Boxes, White Box, grey Box Testing methods can be used. Normally, the method depends on your definition of ‘unit’.

System testing

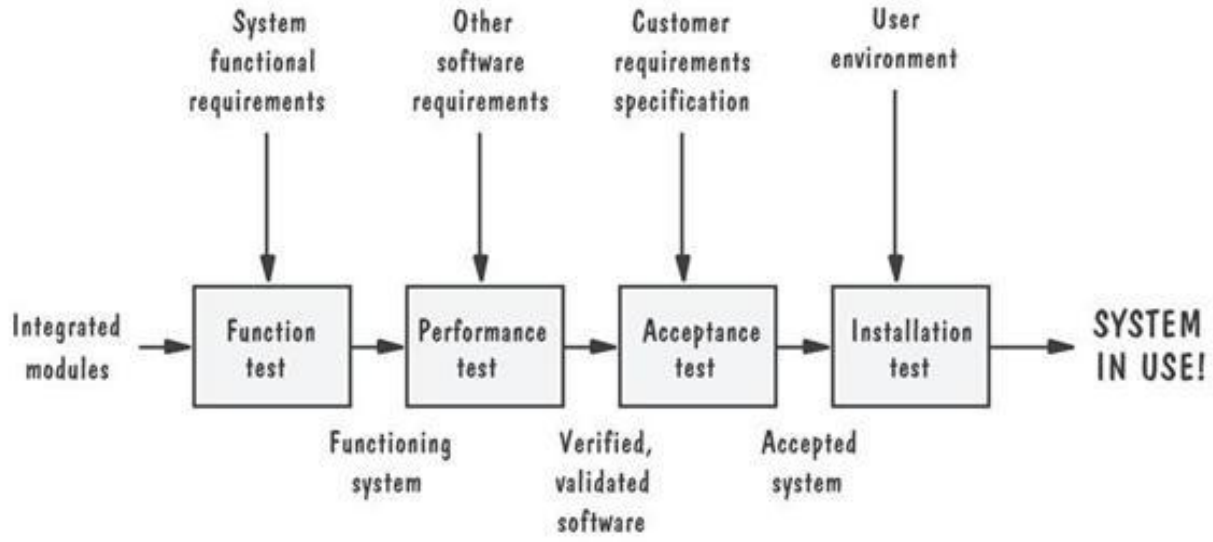


Figure 6.3 system Testing

Unit and integration testing focus on finding faults in individual components and the interfaces between the components. Once components have been integrated, system test that the complete system complies with the functional and nonfunctional requirements of the system. There are several system testing activities that are performed:

Functional Testing**:**

Goal: Test functionality of system

- Test cases are designed from the requirements analysis document (better: user manual) and centered around requirements and key functions (use cases).The system is treated as black box
- Unit test cases can be reused, but new test cases have to be developed as well.

Performance Testing:

Goal: Try to violate non-functional requirements

- Test how the system behaves when overloaded.
- Try unusual orders of execution
- Check the system's response to large volumes of data
- What is the amount of time spent in different use cases?

Acceptance Testing

Goal: Demonstrate system is ready for operational use

- Choice of tests is made by client
- Many tests can be taken from integration testing
- Acceptance test is performed by the client, not by the developer.

Alpha test:

- Client uses the software at the developer's environment.
- Software used in a controlled setting, with the developer always ready to fix bugs.

Beta test:

- Tests conducted at client's environment (developer is not present)
- Software gets a realistic workout in target environment

Installation testing

Usability, functional, and performance tests are performed by the customer in the target environment. If the system is only installed at a small selected set of customers it is called a beta test.

System testing is a black box testing, performed by testing team. The purpose of the system testing is to validate an application accuracy and completeness in performing the function of the system. In this project by integrating of all modules we test the whole system.

Test cases

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides
- Exercise all logical decisions on their true and false sides
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structure to assure their validity

The test case specification for system testing has to be submitted for review before system

testing commences.

6.1.5 Testing materials (Hardware/Software requirements)

Hard ware testing materials: Personal computer

Software testing materials: WAMP/XAMPP local server

6.2. TEST CASE SPECIFICATION

6.2.1. Test case specification Identifier

It refers to the specific identification number given for each test case items.

6.2.2. **Test Items**

User interface: execution of a set of test case is performed on all aspects of the user interface to assure correct web based medication and payment management system

User documentation: the user documentation is reviewed by the project development team and verified against the actual implementation to determine accuracy and completeness of the system. Those are the test items used in our system.

1.Login

2.Manage Staff

3.Manage User Account

4.Manage Patient Detail

5.Manage Appointment

6.Manage treatment

7.Manage Prescription

8.Mange Examination

9.Manage Drugs

10. Manage Payment

11. Generate Report

6.2.3. Input specification

Automated input controls help to ensure users to accurately input information or data, system properly record input, and systems to either reject or accept and record input errors for later review and correction.

6.2.4. Output specification

Specify all of the outputs and features (for example report) required of the test items and provide the exact value with tolerance where appropriate for each required output of the automated system.

6.2.5. Environmental needs

Hardware environmental needs: The characteristics and configurations of the hardware required to execute this test case and this may include the minimum requirements of the system with 1GB RAM, 64 bit, 2.2 GHZ processor, 500 GB hard disk space and the like.

Software environmental needs: System and application software required to execute this test case. This may include windows 7 operating system, macromedia Dreamweaver 8, XAMPP /WAMP server, Mozilla Firefox and PHP programming language.

6.2.6. Special procedural requirements

The test procedures in the system which executes test cases

6.2.7. Interface dependencies

Refers the dependency of one use case on another use case

The general test case specification in our system is prepared as follow:

Test case Identifier	Test case-01
Test Item	Login
Precondition	The user must have an authenticated account
Input specifications	User name(string) and password (string) , user name and pass must greater than 5 characters
Output specification	user name and password match, the user will enter to the page Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	The user name and password must be more than 5 characters
Inter-case dependencies	Login depends on staff registration

Table 6.6: Login test

Test case Identifier	Test case-02
Test Item	Manage staff
Precondition	Member of the hospital employee
Input specifications	Appropriate forms that accepts the necessary data and buttons to register, modify and delete the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	

Table 6.7: Manage staff test case

Test case Identifier	Test case-03
Test Item	Manage user account
Precondition	The user must be registered
Input specifications	Appropriate forms that accepts the necessary data and buttons to register, modify and delete the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	Manage user account depends on staff registration

Table 6.8: Manage user account test case

Test case Identifier	Test case-04
Test Item	Manage Patient Detail
Precondition	Check whether the patient is new or existing
Input specifications	Appropriate forms that accepts the necessary data and buttons to register, modify and delete the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	

Table 6.9: Manage patient detail test case

Test case Identifier	Test case-05
Test Item	Manage appointment
Precondition	Have registered patient
Input specifications	Appropriate forms that accepts the necessary data and buttons to register, modify and delete the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	Appointment depends on staff registration

Table 6.10: Manage appointment test case

Test case Identifier	Test case-06
Test Item	Submit Prescription
Precondition	Treated patient
Input specifications	Appropriate forms that accepts the necessary data and buttons to submit the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	Depends on treatment and lab/rad results

Table 6.11: Manage submit prescription test case

Test case Identifier	Test case-07
Test Item	Manage Examination
Precondition	Lab request by the Doctor
Input specifications	Appropriate forms that accepts the necessary data and buttons to submit the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	Depends on doctor request

Table 6.12: Manage examination test case

Test case Identifier	Test case-08
Test Item	Manage Drugs
Precondition	Prescribed medicine
Input specifications	Appropriate forms that accepts the necessary data and buttons to submit the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	Depends on doctor prescription

Table 6.13: Manage drugs test case

Test case Identifier	Test case-09
Test Item	Manage payment
Precondition	Patient fees
Input specifications	Appropriate forms that accepts the necessary data and buttons to submit the inserted data
Output specification	Correct data will be inserted to the database that match the validation rule Otherwise error message will be displayed
Environmental needs	• Ethernet Network(LAN) • Enough workstations(office space and location) to support the entire team • Tolerances for heat, humidity, motion(well ventilated and safe)
Special procedural requirements	Insert the data according to their data types and check them whether they are valid or not
Inter-case dependencies	Depends on available patient charges/fees

Table 6.14: Manage payment test case

Test case Identifier	Test case-10
Test Item	Generate report
Precondition	Medication activities
Input specifications	Appropriate forms that accepts the necessary data and buttons to view the necessary data
Output specification	Display the required report from the database
Environmental needs	• OS: windows 7, • Tools: Notepad++ or Dreamweaver, WAMP server • Browser: Mozilla Firefox
Special procedural requirements	Insert the data according to their data types and check them weather they are valid or not
Inter-case dependencies	Depends on all users

Table 6.15: Manage Generate report test case

CHAPTER SEVEN

CONCLUSION AND RECOMENDATION

7.1. CONCLUSION

This project is designed to make the manual activities of the hospital computerized/automated. In the current system, the front office management is very slow. After careful investigation of the current problem, detail analysis has been conducted based on the requirement and design of the system and implemented based on the environmental need.

Therefore, the automated system has a great improvement to speed up the process. The Web-Based Hospital Medication Management System was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of the requirements of the hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

The system registers and admits patients, doctors and other staffs detail. Receptionist makes automated appointment of doctors for the patient and doctors make automated treatment and prescription. Technicians made medical examination of patients based on the diagnosis made by the doctor for different examination types such as Hematology, Stool, urine, x-ray, ultrasound and other laboratory examinations. Patients get prescription based on the laboratory/radiology result and treatment made by the technicians and doctors. Pharmacist view doctors' prescription and give medicine to patient. Cashier collects fees and made total payment of the patient in the system.

It generates test reports and also provides the facility for searching the details of the patient. The system also provides the facility of backup as per the requirement.

As a result the proposed project integrates all the functions in the hospital made by receptionist, doctor, technician, pharmacist and cashier. It is fast, avoids data redundancy and inconsistency, easy accessibility of data, automation of the entire system improves the efficiency, provides a friendly graphical user interface which proves to be better when compared to the existing system, gives appropriate access to the authorized users depending on their permissions, it effectively overcomes the delay in communications, updating of information becomes so easier, system security, data security, integrity and reliability are the striking features. The system has adequate scope for modification in future if it is necessary.

7.2. RESULTS ACHIEVED

The project Web-Based Hospital Medication Management System integrates departments without compromising the security of the information involved. For example, doctor's, laboratory and radiology, pharmacist and finance administrative offices each have access to the pieces of a patient's information that are necessary to completing their individual job functions without allowing access to information that is not specifically relevant to that function. Therefore, the system achieves the following results:

- Improved quality of patient care by increasing the timeliness and accuracy of records and administrative information
- Provides simplified access to patient records and other administrative information for those who need it.
- Standard for patient care, making it more consistent across the hospital management rules and regulations in the system
- Increased standardization of hospital management information systems improves cost control by improving efficiency and productivity
- Personnel costs and inventory levels can be reduced through the avoidance of duplications and repetitions of tasks and similar procedures in the hospital
- Eliminate confusion and delays that are caused by incomplete or missing records to lead for decision making because web based medication and payment management system offer the functionality of assembling various types of data into usable reports
- Data can be collected and categorized so that doctors view patent's appointment arranged by receptionist can analyze such information as patient lists, bed occupancy rates and statistics categorized by the hospital responsible persons' in the system
- Increased security of patient information
- The data are stored in the computer's secondary memories like hard disk, etc. it can be easily receive and used at any time from the system database
- The proposed system will easily handle all the data and the work done by the existing systems.

- Hospital management information systems are secured databases of administrative and patient information that can be easily accessed only by those who need the information.

In general the proposed system provides quality service of the hospital to the patient's and medical practitioners.

- Immediate Access of data
- Friendly user interface
- Time saving
- Data can be easily insert/update/delete
- Saving paper work
- Give facility of different types of enquiry
- Data's are easily approachable.
- Avoids redundancy of data

7.3. FUTURE WORK

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the various services that are present in our system. Still, we found out that the project can be done in a better way.

The future work is:

- The different sub-department under medical departments will also have their own detail database.
- Anyone who wants to enhance this project will develop online services out of the campus. That means, if patient have any problem he/she can send his/her problem through online registration from his/her home, then doctor will send an appointment time/date to him/her. In this patients have some login username and password.

APPENDIX I

User manual/guide

User manual is a guide line of the project implementation in the hospital for the different medical practitioners and other staff members of the hospital community.

1. Administrator

1.1 Steps to log in to the system

1. Browse the system page
2. Login windows displays
3. Fill his/her user name and password in the provided space.
4. Click login button
5. System displays the entire application window
6. Perform the necessary tasks

1.2. Steps to manage staff (Register, modify, delete)

1. Clicks staff link on the page
2. System displays staff registration form
3. Fill data
4. Click register
5. Click view button to see list of registered staffs
6. Click modify/delete button to update/delete staff data

1.3 steps to manage user accounts (create, modify and delete)

1. Clicks user account link on the page
2. System displays user account registration form
3. Fill the user account form
4. Click create button
5. Click view button to see list of user accounts
6. Click modify/delete button to update/delete staff data
7. Logout

2. Receptionist

2.1. steps to log in to the system

1. Browse the system page
2. Login windows displays
3. Fill his/her user name and password in the provided space.
4. Click login button
5. Receptionist page displays
6. Perform the necessary tasks

2.2 Steps to manage patient's registration

1. Click patient's registration link on the page
2. System displays patient registration form
3. Fill the patient's information in the displayed form
4. Click register button
5. Click view button to see list of registered patient's
6. Click modify/delete button to update/delete patient's data

2.3 Steps to manage room

1. Click room link on the page
2. System displays room page
3. Receptionist checks the occupied and free beds in the database.
4. Assign the patient's to the available room
5. Click modify/delete button to update/delete occupied room for discharging patients in the database.

2.4 Steps to manage appointment

1. Click appointment link on the page
2. Appointment form comes
3. Fills data
4. Click add button
5. View list of appointments
6. Click modify button to update appointment
7. Click logout

3. Doctor

3.1 Steps to log in to the system

1. Browse the system page
2. Login windows displays
3. Fill his/her user name and password in the provided space.
4. Click login button
5. Doctor page displays
6. Perform the necessary tasks

3.1.Steps to view appointment

1. Click appointment link on the page
2. Appointment page Comes with lists of appointment

3.3. Manage patient detail

1. Click patient registration link on the page
2. Patient registration page displays
3. See the history of the patient

3.4 . Steps to send lab request

1. Click one of the lab request links
2. Lab request form displayed
3. Fill necessary data
4. Click submit button

3.5. Steps to View Laboratory and radiology report

1. Click lab report link on the page (e.g. urine, stool, X-ray….)
2. Lab report displayed

3.6. Steps to submit prescription

1. Click prescription link
2. Prescription page displayed
3. Fill all necessary patient data on the text fields
4. Click submit button
5. Logout

4. Laboratory and Radiology technicians

4.1. Steps to Login

1. Browse the system page
2. Login windows displays
3. Fill his/her user name and password in the provided space.
4. Click login button
5. Technician page displays
6. Perform the necessary tasks

4.2. Steps to View Doctor's lab request

1. Click any of the lab request forms that you want to view
2. Lab request displayed

4.3. Steps to manage medical examination

1. Click lab request link
2. Lab request page displayed
3. Identify type of request to be examined
4. Conduct examination
5. Put result on the forms
6. Submit the result
7. Logout

5. Pharmacist

5.1. Steps to login to the system

1. Browse the system page
2. Login windows displays
3. Fill his/her user name and password in the provided space.
4. Click login button
5. Pharmacist page displays
6. Perform the necessary tasks

5.2. Steps to View prescribed medicine

1. Click on prescription link
2. Prescription page displayed

3. View the type of prescribed medicine
4. Put the price and quantity to be taken

5.3. Steps to manage drugs

1. Click drug link
2. Page displayed
3. Check the availability of the prescribed medicine
4. Logout

6. Cashier

6.1. Steps to login

1. Browse the system page
2. Login windows displays
3. Fill his/her user name and password in the provided space.
4. Click login button
5. Cashier page displays
6. Perform the necessary tasks

6.2. Steps to view charges

1. Click payment link
2. Page displayed
3. Insert MRN of the patient
4. Charges displayed
5. Make sum
6. Click print
7. Click logout

APPENDIX II

Questioners: Write appropriate answer for each questions

Personal information

Age: ____ sex: ____ Qualification: _____Designation: _____

1. Dose the hospital have complete hierarchical structure? If so may you provide us a hard or softcopy.

   _____

   _____

2. Could you tell us the way of hiring employees? Who is responsible for that?

   _____

3. What are the functional areas and their activities under medical department? Is there any relationship between the functional areas?

   _____

   _____

4. Who is the responsible body to receive drugs to store in the pharmacy? What are the ways that he or she uses to store the drugs?

   _____

   _____

5. What are the steps performed after patient consult with doctor including treatment and prescription?

   _____

   _____

6. Who is the responsible person to check and allot the availability of rooms and what he/she does if there is no available room for the patient in some medical operations?

   _____

   _____

7. Could you tell us the way of collecting payment from patient registration up to patient discharging and way of auditing? Who is responsible for auditing?

   _____

Thank you for your cooperation!

APPENDIX III

GLOSSARY

Analysis

The part of the software development process whose primary purpose is to formulate a model of the problem domain. Analysis focuses what to do, design focuses on how to do it.

Architecture

The organizational structure of a system Architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts.

Activity diagram: diagrams showing the flow of activities and covers the dynamic view the system

Actor: a consistent entity taken by the use case user when he/she interacts with use case.

Attribute: characteristics of a named type

Association

The semantic relationship between two or more classifiers that involves connections among their instances

Attribute

A named slot within a classifier that describes a range of values that instance of the classifier may hold.

Behavior

The observable effects of an operation or event, including its results

Binary association

An association between two classes A special case of an n-ary association.

Black box testing: method of testing that focuses on determining whether or not a program does what it is supposed to do base on its functional requirements.

Class diagram

A diagram that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships

Class: a description of a group an objects which share the same attributes, operation, relationship and semantics.

Component

An executable software module with identity and a well-defined interface

Context

A view of a set of related modeling elements for a particular purpose, such as specifying an operation

Delegation

The ability of an object to issue a message to another object in response to a message Delegation can be used as an alternative to inheritance.

Dependency

A relationship between two modeling elements, in which a change to one modeling element (the independent element) will affect the other modeling element (the dependent element)

Deployment diagram

A diagram that shows the configuration of run-time processing nodes and the components, processes, and objects that live on them. Components represent run-time manifestations of code units.

Derived element

A model element that can be computed from another element, but that is shown for clarity or that is included for design purposes even though it adds no semantic information.

Extends

A relationship from one use case to another, specifying how the behavior defined for the first use case can be inserted into the behavior defined for the second use case.

Implementation

A definition of how something is constructed or computed For example, a class is an implementation of a type, a method is an implementation of an operation.

Interaction

A specification of how messages are sent between instances to perform a specific task. The interaction is defined in the context of collaboration.

Interface

A declaration of a collection of operations that may be used for defining a service offered by an instance

Message

A specification of a communication between instances that conveys information with the expectation that activity will ensue. The receipt of a message instance is normally considered an instance of an event.

Model

A semantically closed abstraction of a system

Module

A software unit of storage and manipulation Modules include source code modules, binary code modules, and executable code modules.

Object diagram

A diagram that encompasses objects and their relationships at a point in time An object diagram may be considered a special case of a class diagram or a collaboration diagram.

Object lifeline

A line in a sequence diagram that represents the existence of an object over a period of time

Operation

A service that can be requested from an object to effect behavior An operation has a signature, which may restrict the actual parameters that are possible.

Persistent data management: used to communicate the design of database, usually relational database with the user and other developers.

Relationship

A semantic connection among model elements Examples of relationships include: associations and generalizations.

Specification

A declarative description of what something is or does.

Sequence diagram: graphically depict how objects interact with each other.

Structural feature

A static feature of a model element, such as an attribute

Use case

The specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.

UML- unified modeling language that describes the behavior, the interactions and relations of various objects in the system.

Use case diagram

A diagram that shows the relationships among actors and use cases within a system

Use case instance

The performance of a sequence of actions being specified in a use case An instance of a use case

Use case model

A model that describes a system's functional requirements in terms of use cases

Uses

A relationship from a use case to another use case in which the behavior defined for the former use case employs the behavior defined for the latter.

White box testing: focuses on the internal structure of the software code.

BIBLIOGRAPHY

1. Elizabeth Nara more, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass Beginning PHP5, Apache, and MySQL® Web Development, 2005
2. Elmarsi and Navathe, Fundamentals of Database System (Third Edition), Addision Wesley.
3. Ian Somerville, Software Engineering, Third Edition, Pearson Education.
4. Ali Bahrami, Object-Oriented System Development, Third Edition, Tata McGraw Hill Edition.
5. Adams M, Koch R (2010) Pharmacology connections to nursing practice. Pearson, New Jersey
6. Anderson J, Webster C (2001) a systems approach to the reduction of medication error on the hospital ward. J AdvNurs35(1): 34-41
7. Australian Commission on Safety and Quality in Health Care (ACSQHC) (2006) National Inpatient Medication Chart.
8. Aronsky, D. (2008). An Integrated Computerized System in the Emergency Department USA: Vanderbilt University, Nashville, TN, USA.
9. Coiera E. (1993). The role of knowledge based system in clinical practice: in the EPISTOL WORKING CONFERENCE. HP company pp 1-7
10. Ja'afar, S. (2007). Development of a prototype web-based emergency triage system using fuzzy expert system Malaysia: university teknologimara.
11. Pezzè, Mauro, Young, Michal (2008) Software Testing and Analysis: Process, Principles and Techniques. NJ: Wiley pp. 375-389.
12. Blue print for Pharmacy Steering Committee, September 2012
13. Wolfgang Emmerich (1997)
14. JussiKangasharju (2008)
15. ACGIL, (2010)
16. e-How, (2010)

Additional Sites

- www.associatedcontent.com
- www.members.tripod.com
- www.phpcodesource.com
- www. Planet sourcecode.com