WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE: - STUDENTS' CLINIC MANAGEMENT SYSTEM FOR WOLKITE UNIVERSITY

BY

NO	NAME OF THE STUDENTS	ID NO
1.	ESAYAS SAMUEL	CIR/042/11
2.	GELASA JIRANE	CIR/050/11
3.	RIHOBOT GETACHEW	CIR/082/11

PROJECT ADVISOR: MR. TESFA KEBEDE

JUNE 14, 2022

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE: - STUDENTS' CLINIC MANAGEMENT SYSTEM FOR WOLKITE UNIVERSITY

SUBMITTED TO DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

BY

NO	NAME OF THE STUDENTS	ID NO
1.	ESAYAS SAMUEL	CIR/042/11
2.	GELASA JIRANE	CIR/050/11
3.	RIHOBOT GETACHEW	CIR/082/11

PROJECT ADVISOR: MR. TESFA KEBEDE

Wolkite University, Wolkite, Ethiopia

June 14, 2022

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Tesfa Kebede and having the title Students' Clinic Management System for Wolkite University is the sole contribution of:

1. Esayas Samuel
2. Gelasa Jirane
3. Rihobot Getachew

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: March 2, 2022

Group Members:

Full Name	Signature
Esayas Samuel	----------
Gelasa Jirane	----------
Rihobot Getachew	----------

APPROVAL FORM

This is to confirm that the project report entitled Students' Clinic Management System for Wolkite University submitted to **Wolkite University, College of Computing and Informatics Department of Computer Science** by: Esayas Samuel, Gelasa Jirane, and Rihobot Getachew is approved for submission.

Advisor Name	Signature	Date
Department Head Name	Signature	Date
Examiner 1 Name	Signature	Date
Examiner 2 Name	Signature	Date
Examiner 3 Name	Signature	Date

ACKNOWLEDGMENT

First of all, we praise God, the almighty and merciful, for providing us this opportunity and granting us the capability to proceed successfully. Also we want in a special way to appreciate our family for the support and sponsorship of our education from initial to preparing the first-semester final project documentation, for care, patience, and love we pray the Almighty God grants them long life on earth to enjoy the fruit of their labor. We would like to express our deep thanks for all members of clinic workers who gave us most information that we have collected to do our project. Finally we would like to express our deep thanks to our advisor Mr. Tesfa Kebede for guiding us throughout this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

BR..... Business Rule

DBMS ..... Database Management System

GUI .....Graphical User Interface

HTTP .....Hypertext Transfer Protocol

IIS .....Internet Information System

RN..... Registration Number

UC..... Use Case

UML .....Unified Modeling Language

WKU..... Wolkite University

ABSTRACT

Web based clinic management system is a system that will be used to manage the proper aspects of the clinic. The WKU students' clinic management system is developed with the objectives of making the system more reliable, easier, and informative. There are a lot of reasons for introduction to this system. In manual system, there are the number of inefficiencies that a patient faces. Large record books have to be maintained where information to be stored. But our system reduces paper work. On the other hand there are inherent problem that exist in manual system. Usually they lack efficiency. Existing system currently performs different activities manually. This approach lacks the capability to alleviate problems like susceptibility to errors, incompleteness, performance, time consumption, security, efficiency and ineffective data for analysis in the WKU students' clinic day to day activities. The best alternative solution to the existing system is to change the existing manual system in to computerized to reduce the time consumption, and human power.

CHAPTER ONE

1. INTRODUCTION

The development of different computer based system has provided tremendous advantages for many organizations and institutions. Many developed countries are using computer technology to accomplish their everyday activities effectively and efficiently. Nowadays many companies are using automated system to accomplish their task but some are still using manual system.

Clinics and Hospitals deals with the life and health of their patients. Good medical care relies on well-trained doctors, nurses, high quality facilities, and equipment's. Good medical care also relies on good record keeping. Without accurate, compressive and up to date and accessible patient notes, medical personnel may not offer the best treatment or may in fact misdiagnose the condition ,which can have a serious consequences. Good records can also ensures the health care centers administration runs smoothly; unneeded records are transformed or destroyed regularly, keeping storage areas clear accessible; and key records can be found quickly; saving time and resources. Records also provide evidences of the health centers accountability for its action and they form a key source of data for medical research, statistical reports and health information systems. However, the current system in Wolkite University students' clinic is inefficient for better performance of all tasks, for managing all medical related works to register and store patient's record [1].

1.1 Background of the Organization

Wolkite University (WKU) is one of the third generation universities in Ethiopia. The University is cited in the SNN regional state, Gurage Zone, about 150 km west of the capital. WKU is giving different services to the students and other staff members. Among those services, students' clinic is the one that provide health care service for students. This student clinic was established in 2004 E.C. together when the university is established. It is one of Academic sector that provides direct support to the teaching learning and research activities for the university. The purpose of this clinic is to give medical service for WKU students. At the beginning, the WKU Clinic gives all medication service such as, medications,

Consultation service, HIV testing, etc. There are about 34 employee that work in the Wolkite University Clinic.

The clinic is almost using manual system for managing the overall activities of the work in the clinic. The admission or registration of patients is being done using paper and pen with manpower. The circulation of information from one department to another is performed by people with paper and card.

1.2 Statement of the Problem

WKU student clinic is facing many problems with its manual (paper based) system. The number of patients is increasing from time to time, so that managing those patients by using existing system is very tedious that require more number of human power and material. The system which we want to generate is proposed to eliminate the problem of redundant, error full and incomplete data that was escalating the inefficiencies in data retrieval these limitation were mainly caused by the fact that data, under the existing manual recording system was entered into books and paper files and was later stored in overcrowded storage rooms that make retrieval of archival records close to impossible.

Some of the main problems of the existing system are listed as follows:

Large storage medium (space) is required to store medical documents, cards, reports, and patient information. Under this system patient information was entered into books and paper files and was later stored in overcrowded storage rooms. Because of this patient information takes a lot of storage space. In addition to this searching for even single data is time consuming. Since information is stored in overcrowded storage rooms it was hard to retrieve patient records at needed time.

In addition to above problem redundancy (multiple record of the same data)was occurred. Sometimes when patients may forgot their clinical code, and they told record officers that they had not any medical record before, then the record officer registered them as new, hence this leads to data redundancy. Also, this system was difficult in preparing an organized reports. When a patient needs sick leave for academic case first record officer must generate the patient medical file, since the report was not well organized it may little difficult to generate it as fast as patient needs.

Sudden damage on storage medium resulting in zero data. This is because there was no any back up of information under existing system and circulation of patient information such as lab report generated by lab technician, doctor prescription to pharmacy was too late. When record officer must directly go to OPD room to provide the information recorded from patient to the doctor. After that the patient was treated then his/her record must back to record office. This makes the circulation of the patient information too late. Also, there was loss of patient record because of patient medical files were not well organized.

Generally, to answer problems listed above we will propose web based students clinic management system.

1.3 Objectives

1.3.1. General Objective

The general objective of the project is to develop a web based students' clinic management system for Wolkite University.

1.3.2. Specific objective

The specific objective will help as to achieve the general objective that is mentioned above and the individual tasks done in the project. Specifically, this project is:

- To Study and analysis of the manual system.
- To identify functional and non-functional requirements for the new system.
- To design and develop the students' Clinic management system.
- To implement designed (proposed) system by using React JS,Node JS and Express.js.
- To test and evaluate the implemented system.
- Deploying the tested system.
- Test the performance and reliability of the new system.

1.4. Feasibility analysis

1.4.1. Technical feasibility

The project team members have learned programming languages that required for the successful completion of the project such as react JS, Node JS. We also learned the OOSAD

(Object Oriented System Analysis and Design) methodology that we followed to develop this project. Team members have the required skill to develop the system, So that the project can be said technically feasible.

1.4.2. Operational feasibility

This system will bring better achievement for the operations performed by the clinic by providing efficient registration and storage of patient information, easy updating, deletion, modification etc. This intern will increase the efficiency of work in the clinic. So that one can say that the system is operationally feasible.

1.4.3. Economic feasibility

More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. It involves the cost incurred on the system development team, estimated cost of hardware and software, cost of performing feasibility study and so on. In addition to this as the existing system is manual cost of papers and pens is very high, so once the proposed system is deployed it will reduce such costs. This makes the system to be economical feasible.

Tangible benefits

Our new system gives tangible benefits that can be estimated in terms of money which means the benefit is real or actual rather than imaginary or visionary. For example, the system provides cost reduction /avoidance such as pens, paper, and reducing materials consumption.

Intangible benefits

The new system gives intangible benefits that cannot be estimated in money such as increasing work processing efficiency, Moral satisfaction for developers, error reduction, and increased flexibility, facilitating the activities, the purchasing workers and the user.

1.5. Scope and Limitation of the project

1.5.1. Scope of the project

The system exclusively proposed for Wolkite university students' clinic. It will cover most of the functions listed below.

- **Report module:** The system generates different reports.
- **Patient management module:** The system generates sick leave, refer, and allows feedback for patients.
- **Registration module:** The system allows registration of account, employee, drugs, and patients into the system.
- **Pharmacist module:** The system checks expired date of drugs, and allows drug request.
- **Laboratory module:** The system allows the lab technician to send lab order.
- **Schedule module:** The system allows the management of schedule needed to the clinic.

1.5.2. Limitation of the project

- A blind cannot access the system.
- The system does not support any language rather than English.

1.6. Significance of the project

The significance of this project is to create efficient and effective working system for all the workers and the target users. Generally, as we are under information and technology age using computerized system is more advantageous than that of using manual system to decrease the complexity of the worker.

The system will do different activities for different units of the WKU students' clinic: -

- Students' information can be stored and searched easily.
- Reduction of paper and pen.
- Allow sharing information using the system without moving from place to place.
- It saves money for the clinic and save time for the students and also for the workers in the clinic.
- Track all the activity from one system.
- To develop the system that can easily maintain and retrieve the record of patient.
- To develop the system that can reduce the number of lost all clinic records by replacing paper by database-based system.

- To develop the system that can reduce the redundancy of patient information using database constraints.

1.7. Beneficiary of the project

The system will be so helpful and beneficiary for stakeholders in different level

WKU students' clinic: - this system provides an Easy and efficient way of handling recording system for the clinic. Therefore, the clinic will have smug working environment.

Students (Patients): - the service that is given by using this system satisfy the students by saving their time and it will exclude the exertion.

Health Care providers: - Will be able to easily access their patients information and easily prescript the medicine.

Developers: - the developers may earn experience from this project for their next carrier. Also after this project is implemented, it will be sold for many organizations in need so that the developers will be benefited by it.

1.8. Methodology of the project

1.8.1. Data collection tools/techniques

Data collection is the most important part of the project to find the main requirement of the system and to understand how the system works. We are going to use different methods to collect data. Among the methods, we use the following: -

Observation: We arrived about 3 days physically to the clinic and observe how workers carrying out their work activities in a natural setting. We observed physically at the clinic and observed that there is no any computerized system. For example when a patient arrived at the clinic he or she has to wait for a lot of minutes to get treatments. Almost all activities are being performed manually.

Document analysis: To get more information about the project we refer earlier documents that help us to develop the project. Those are documents that are done before about the WKU students' clinic and the book "Introduction to Public Health Policy and Management" [1]. In addition to this, we give a special consideration to those documents, which can bring more features to the project.

Interviews: This is the most important method that helps us to get most important and critical information about the general view of the clinic is by interviewing almost all actors of the clinic. We interview about 7 individuals. We use specific questions and ask actors of the clinic in order to gather information about the existing system, problems in the existing system and the patients need (Interview questions).

Internet: Aids us to see the available sample on the internet and to download different types of software, tutorials and documents that help us in developing the project.

1.8.2. System analysis and design

In our project, we apply the concept of OOSAD (Object Oriented System Analysis and Design) development methodology, because it is best way to construct, manage and assemble objects that we are going to implement in our system, and the composition of objects and collaboration between objects on the system. It Models the functions of the system (use case modeling), organize the objects and the relationship between them and finally model the behavior of the objects. It also shows object interactions and behaviors that support the use case and scenario, and finally update object model to reflect the implementation environment. We choose this approach because of the following advantages:

- **Increased re-usability:** object oriented system support re-usability of system.
- **Increased extensibility:** to add and change the existing module without affecting the rest of the program.
- **Improved quality:** introduces user participation, which improves quality of the project.
- **Managed complexity:** the object oriented method solves software complexity in the following way, designing the software with the expectation that it will need to be modified and being able to respond quickly when requirement or environment changed.

1.8.3. System development model

System development method is a framework that is used to structure, plan and control the process of developing a system so to develop this project we are going to use waterfall model because the implementation phase will be done after the documentation of the project is fully

finished. This model is used when the requirements are very well known, fixed and clear. In addition, the project scope stays relatively static. It is also easy to understand and use.

1.8.4. System Testing Methodology

1.8.4.1. Unit testing

We will use unit testing during the development (coding phase) of an application in order to isolate a section of code and verify it correctness. Unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules.

1.8.4.2 Integration testing

It used to verify the performance of the complete system of the software. Basically units of software are added together to create modules. Then these modules of software are integrated together into complete system and that is what we are testing.

1.8.4.3. Acceptance testing

In this testing stage, we will evaluate the systems compliance with the business requirement and verify if it has met the required criteria for delivery to end user. The users will participate on the acceptance testing of the system. According to system requirements and other resources (documentation, source code, user manual) test cases are generated to determine (validation and verification) whether the system satisfies users need and expectation to solve user's problem.

1.8.5. Development tools and technologies

- React JS is an open source JavaScript library that is used for building user interfaces.
- Express.js : express.js, or simply Express, is a back end web application framework for Node.js
- Node JS is an open source server environment.
- Database Environment: MongoDB is a document database used to build highly available and salable internet applications. With its flexible schema approach, it's popular with development teams using agile methodologies.

1.8.5.1 Front end tools

- **React JS**

1.8.5.2 Back end tools

- **Node JS**
- **Express.js**

1.9 Document organization

This sub section of our project consists of seven chapters and its descriptions shortly:

Chapter One: Introduction of the whole Project: This chapter focus on the reason for studying the problem selected should be listed. Additionally, the problem area and motivation to the need for our project work is mentioned.

Chapter Two: Description of the Existing Systems: It describes the exercising system how it works in detail wisely, to describe the users and their responsibilities in the existing system. Describe the exercising system major functions. In this chapter describe the existing system disadvantages and many more difficulties to work well. Also it studies about business rules of the existing System.

Chapter Three: Proposed System: This chapter studies about the overall description of our proposed system, functional requirements, and non-functional requirements of our project.

Chapter Four: System Analysis: It studies about use case model, object model and dynamic models of the proposed system.

Chapter Five: System Design: It describes about design goals, current and proposed software architecture, Hardware/software mapping, Persistent data management and Access control and security.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

Defining system requirements is the initial step of constructing the whole system. Without requirements, the goal is unclear and the construction may be off track (out of the path). Stating these requirements and identifying their weakness is building blocks of the new system.

The WKU students' clinic is almost using manual system for managing the overall activities of the work in the clinic. That needs intensive human labor, resource, consume time, less secure. This leads to a very inefficient and awkward way of storing records greatly slow down the flow of critical information as well as makes looking up information time-consuming.

2.2 Users of Existing System

Manager: - the clinic's manager is responsible for controlling all activities of the clinic to ensure high uptime of the clinic. That means he/she can control all the clinic's work, and generate the overall clinic report.

Doctors: - a person skilled or specializing in healing arts, the one who is trained and licensed to heal and treat the sick person. Also have the following activities in WKU students' clinic: -

- Generate refer
- Generate sick-leave
- Patient diagnosing
- View patient information
- Take physical test
- Order laboratory test
- Order medicine
- Medical advice
- Record patient history
- Inject medicines
- Write refer when the medicine is not found in the clinic

Record officer: - a person who works in the card room whose responsibilities are: -

- Registering the patient
- Store registered files across a shelf
- Search existing patient and
- Assign patient information for health care providers

Laboratory Technician: - a person who works in a laboratory room whose responsibilities are: -

- Take a sample from the patient
- Test the sample

- Return the result of the patient to the doctor

Pharmacist: - a person who works in pharmacy room he/she is responsible for the following activities: -

- View drug order
- Give drug to the patient
- Generate pharmacy reports
- Register the medical instruments
- Avoid expired drug
- Respond to the doctor when the drug is not found in the store

Patient: - a person who has some medical problems or need medical advice from the clinics. They have the following activities: -

- Get treatment
- Get laboratory services
- Get drug
- Wait for queue

2.3. Major Functions of the Existing System

Patient registration

- When the patient arrives at registration, the record officer should ask the patient student identification card (ID), if he/she has ID then the record officer ask full information of the patient such as name (first name, father's name and grandfather's name) and then look for an existing registration number in the book which holds the university's student information. This is done whether the patient reports that he/she has been to the clinic before or not.
- If the patient is already registered, the record officer should facilitate the retrieval of the existing file from the file cabinet in the record room. A runner/transistor should

retrieve the patient's file and then take the file to the area where the patient is to be treated.

- If the previous registration number cannot be found, the record officer should generate a new registration number to the patient.
- The new registration number should be issued in a straight numeric sequence, without skipping any numbers.
- Each registration number should be assigned to one and only one patient.
- All patients- regardless of which service they will access-should be registered at central registration site.

Retrieving existing medical record for a returning patient

- If the patient knows his/her registration number, then the registration number can be used to find the registration number can be used to find the patient's record from the file cabinet.
- Retrieving file by name or id number, if the patient does not remember their registration number.

Storage of medical records

- All active patient files should be filed in a single, centralized file room i.e. card room.
- The files should be filed according to the registration number.
- All patient files should be stored together.

Giving medical treatment with prescription

- If it is necessary, the doctor will order a laboratory tests to the patient or he/she will order drugs to him/her.
- The patients get medication from doctors.

Reporting laboratory results for patients

- The laboratory technician makes laboratory testing and finally reports to the doctor.

- Then he/she will resend the result to the doctor.

Deliver drugs to the patient with a prescription

- The pharmacist will accept drug orders from the doctor then will provide the ordered type of drugs to the patient.

2.4 Forms and Other Documents of the Existing Systems

In the existing system, they use different forms and reports to manipulate different records associated with different activities. Among them the followings are some of the documents used in the existing system are: -

The figure below is the from WKU students' clinic. It is the patient information form. It contains all of information about the patient like age, Name, sex, ID no, symptoms, blood pressure and so on.

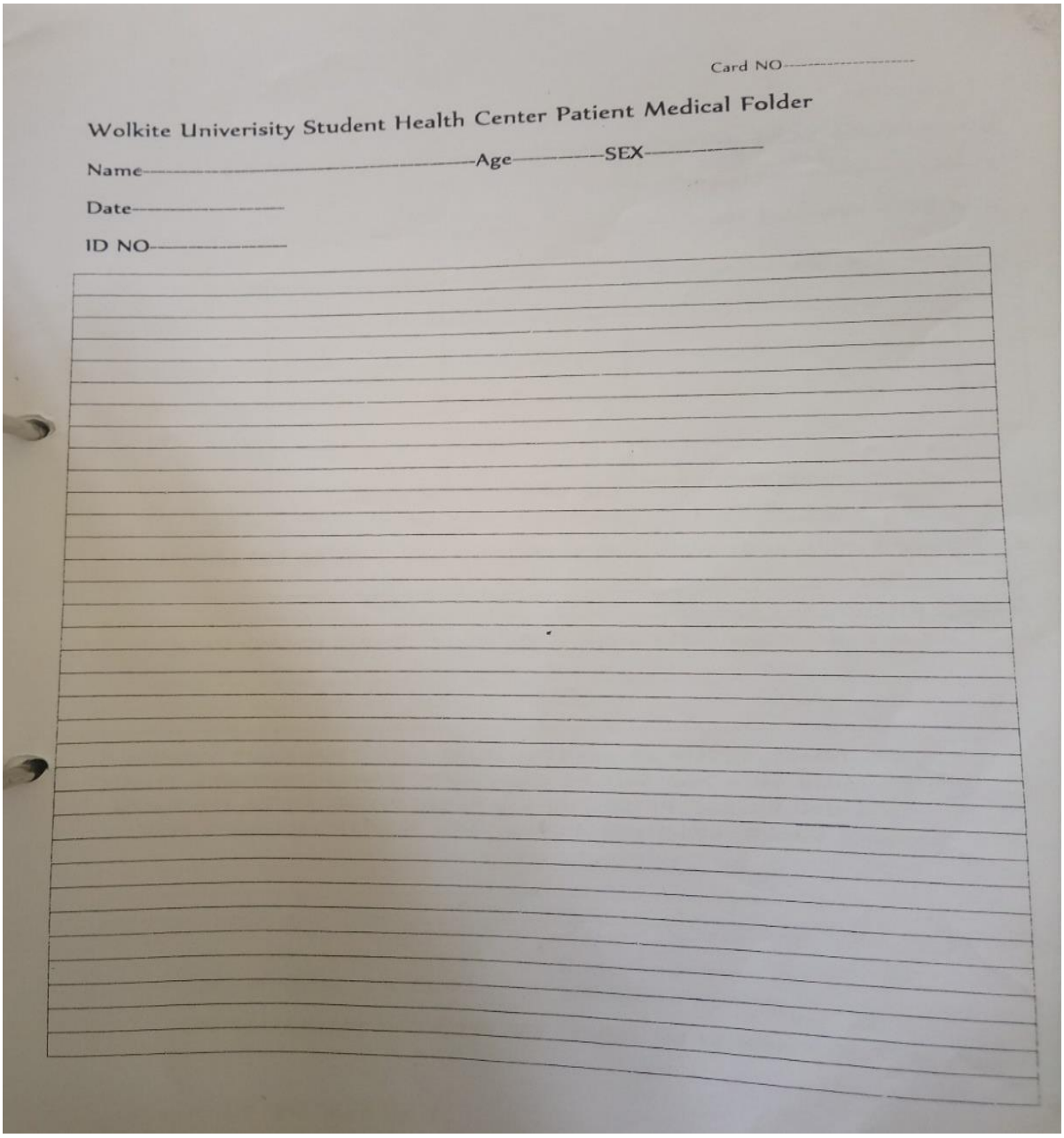


Figure 2. 1 Patients' Medical Folder

This image is from WKU students' clinic. It is the drug prescription paper, the doctor give this paper to the patient so the patient can get drug from the pharmacist.

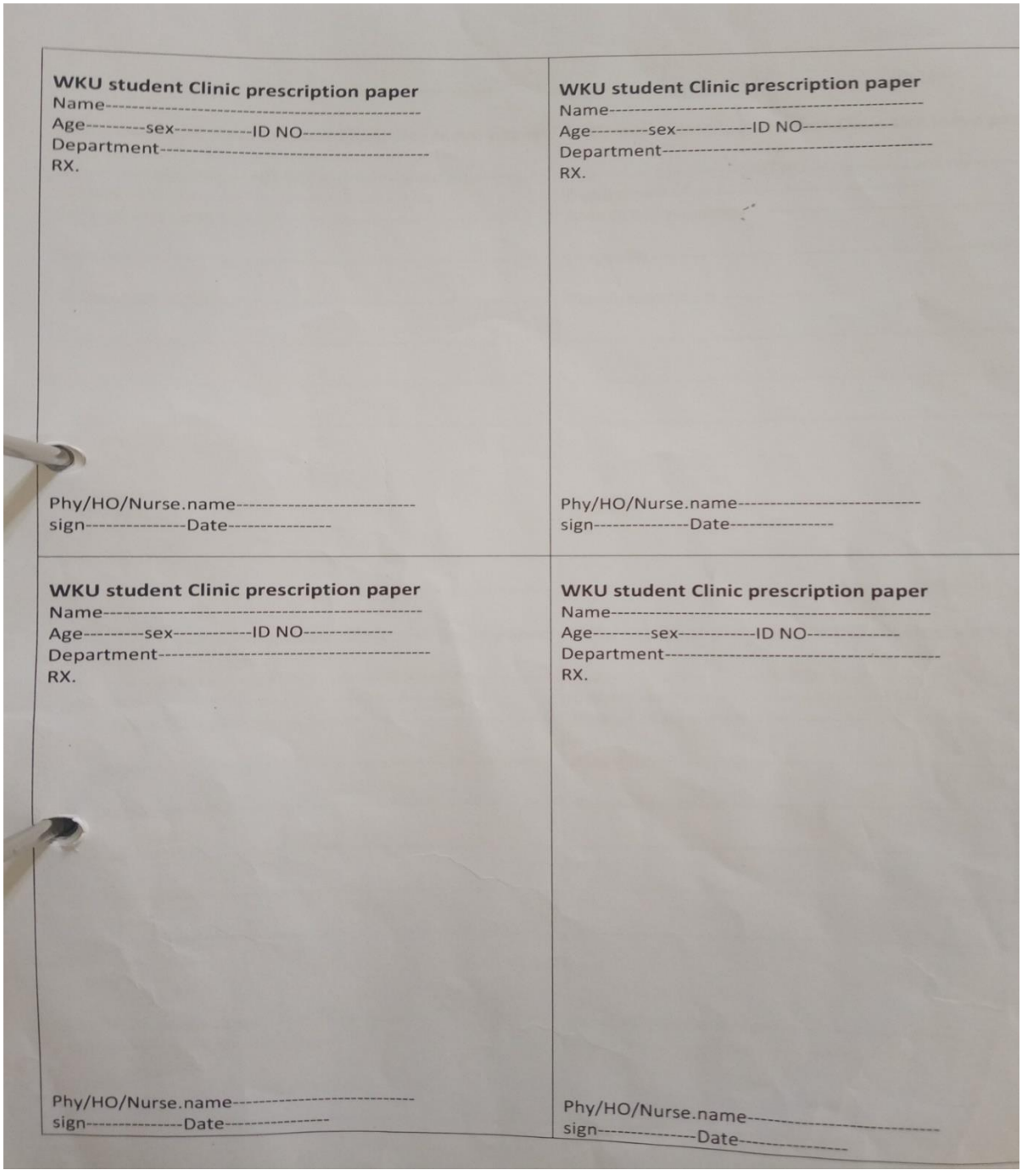
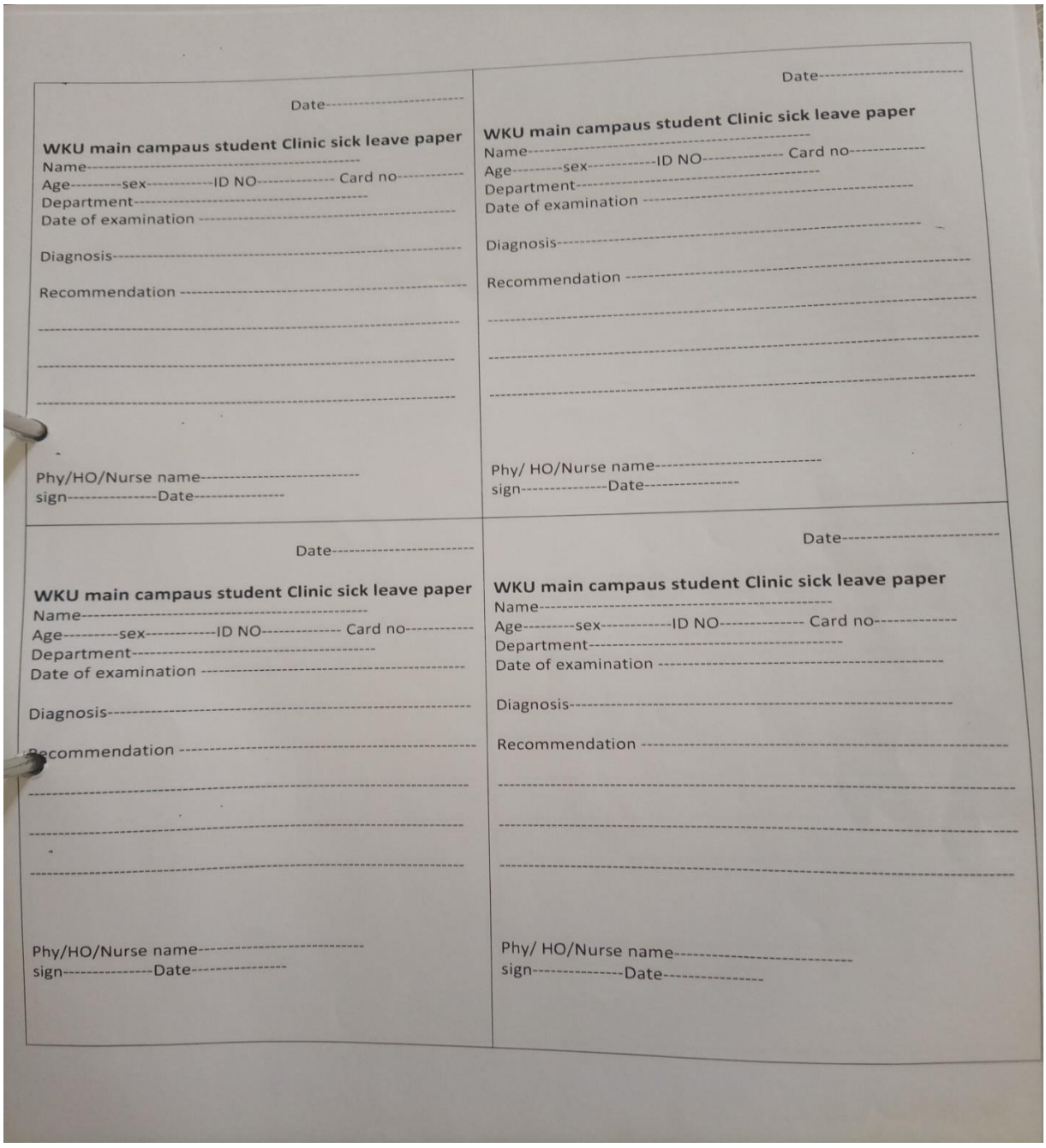


Figure 2. 2 Drug Prescription Paper

The image here is the sick leave paper for patients in case the student has any academic activity on that day. This paper will be evidence for the student.



Date----------

WKU main campaus student Clinic sick leave paper

Name----------
Age---------sex----------ID NO---------- Card no----------
Department----------
Date of examination ----------

Diagnosis----------

Recommendation ----------

Phy/HO/Nurse name----------
sign----------Date----------

Date----------

WKU main campaus student Clinic sick leave paper

Name----------
Age---------sex----------ID NO---------- Card no----------
Department----------
Date of examination ----------

Diagnosis----------

Recommendation ----------

Phy/ HO/Nurse name----------
sign----------Date----------

Date----------

WKU main campaus student Clinic sick leave paper

Name----------
Age---------sex----------ID NO---------- Card no----------
Department----------
Date of examination ----------

Diagnosis----------

Recommendation ----------

Phy/HO/Nurse name----------
sign----------Date----------

Date----------

WKU main campaus student Clinic sick leave paper

Name----------
Age---------sex----------ID NO---------- Card no----------
Department----------
Date of examination ----------

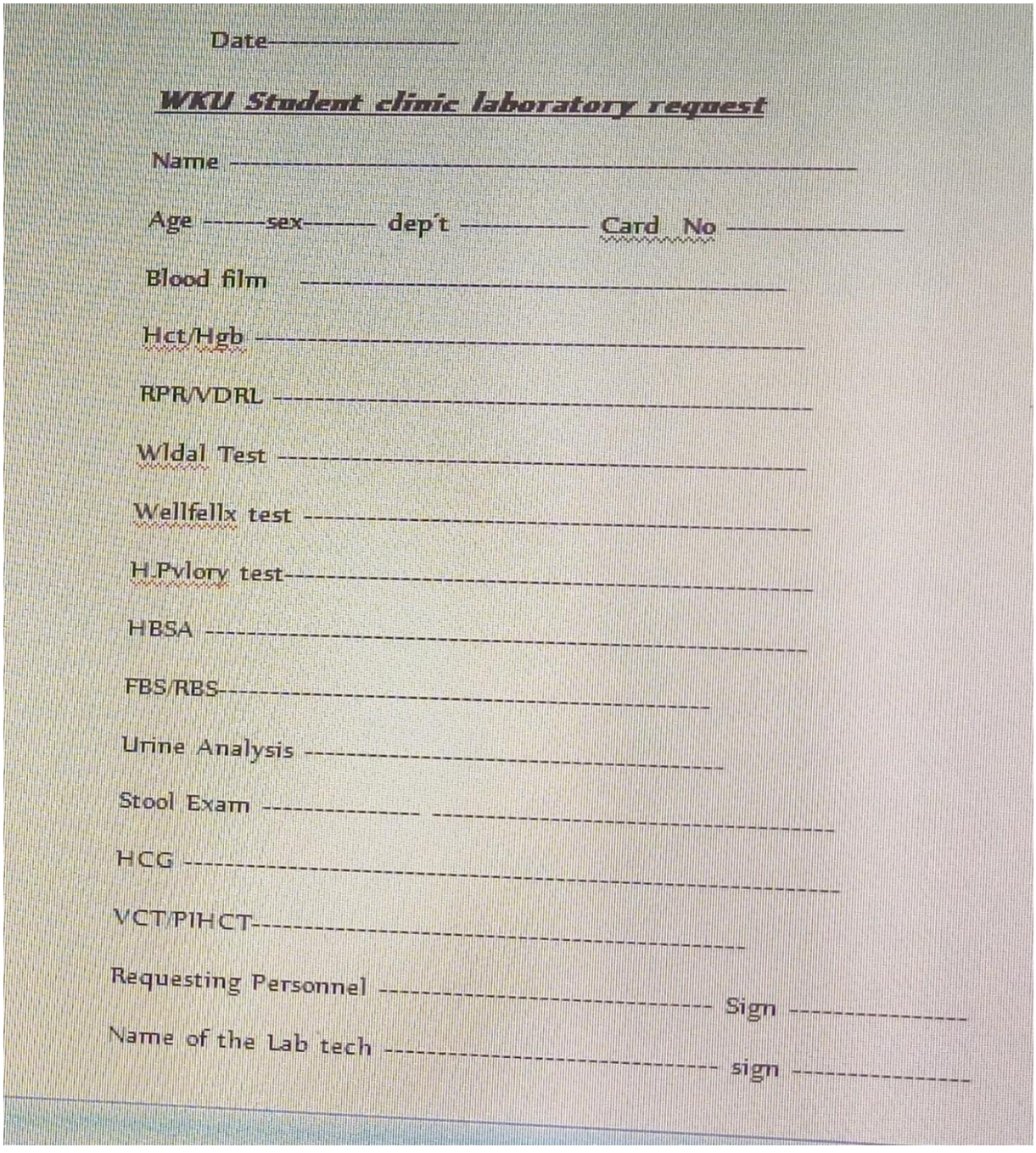
Diagnosis----------

Recommendation ----------

Phy/ HO/Nurse name----------
sign----------Date----------

Figure 2. 3 Sick Leave Paper

The figure below is laboratory request paper which the doctor gives to the patient so that the patient can get all the tests on the paper.



Date_____

WKU Student clinic laboratory request

Name _____

Age ____sex_____ dep't _____ Card No _____

Blood film _____

Hct/Hgb _____

RPR/VDRL _____

Widal Test _____

Wellfellx test _____

H.Pylory test_____

HBSA _____

FBS/RBS_____

Urine Analysis _____

Stool Exam _____ _____

HCG _____

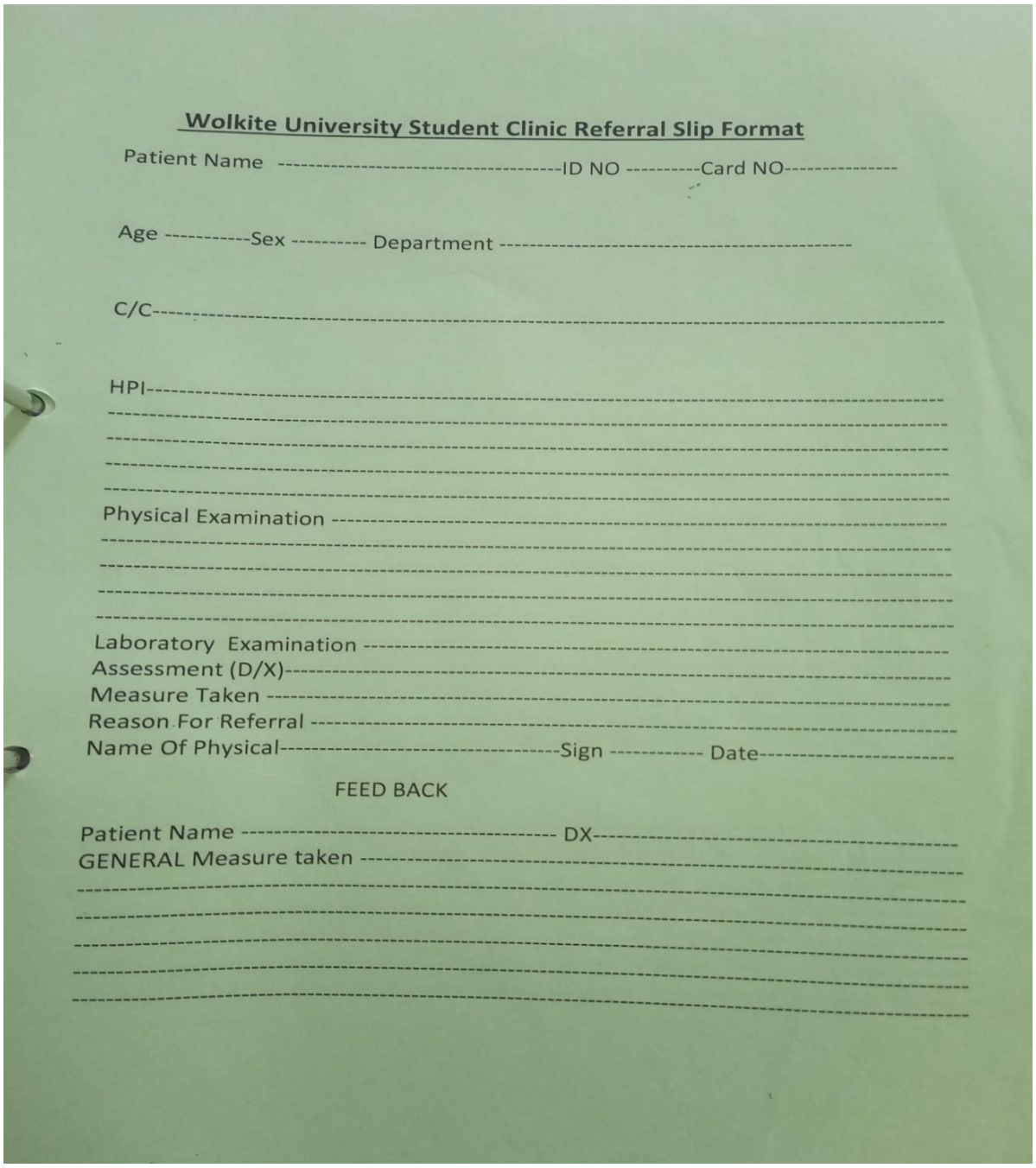
VCT/PIHCT_____

Requesting Personnel _____ Sign _____

Name of the Lab tech _____ sign _____

Figure 2. 4 Laboratory Form Paper

The figure below is the patients' full information send to referral hospital with referral paper.



Wolkite University Student Clinic Referral Slip Format

Patient Name ----------ID NO ----------Card NO----------

Age ----------Sex ---------- Department ----------

C/C----------

HPI----------

Physical Examination ----------

Laboratory Examination ----------
Assessment (D/X)----------
Measure Taken ----------
Reason For Referral ----------
Name Of Physical----------Sign ---------- Date----------

FEED BACK

Patient Name ---------- DX----------
GENERAL Measure taken ----------

Figure 2. 5 Referral Paper

2.5 Drawbacks of the Existing System

It has many problems that affect a compliment of tasks easily, efficiently and effectively, some of the drawbacks are listed as follow: -

Time consuming

The clinic records the necessary information in the paper-based form. Since the files may be misplaced and sometimes the record of the patient may be wrongly filled. This results in inconvenience and wastage of time. And also lack of immediate information storage, the information generated by various transactions takes time and effort to be stored in the right place. The patient has to wait a long time until other patients to take the service. Communication between doctors, pharmacy and record officer also take time.

It is difficult to update and modify the data

The information which is stored in the file room is not flexible for updating or modifying if any mistakes occur so this is lead to an impure collection of data.

Difficult for reporting

Since the information which is found in the file room is not well organized so to generate and prepare timely reports is much difficult.

Data redundancy

If there is no well-organized collection of data it will occur data redundancy, because of the registration room have no structured data it will be difficult to check whether the patient has registered his information or not. Especially this type of limitations occurred at the time of the patient do not remember the registration number and lie to the clerk that he/she doesn't come here previously.

Cost

The clinic records the necessary information in the paper-based form. The usage of traditional paper-based forms to record information has several drawbacks since the university will provide the papers to the clinic by demanding a high cost to it. And also, the file cabinet included this type of drawback of the existing way of giving service because the papers is need many space to be stored so additional file cabinets and space will be needed to store the information.

2.6. Business Rules of the Existing System

Business rules are rules that are used as a guide to performing all tasks according to the rules and regulations. The following are some of the business rules of the clinic in the current time.

BR1: A patient must have student identification card or ID to get service.

BR2: One patient should have one registration number (RN).

BR3: Only the concerned body can see the medical card

BR4: The drug could not be given without a prescription from the doctor

BR5: Lab tests could not be taken unless there is an order from the doctor

BR6: All staff in the clinic must be registered or members of a clinic-based perspective roles.

CHAPTER THREE

3. PROPOSED SYSTEM

Our proposed system is to develop web based clinic management system for WKU student clinic. The main aim of our proposed system is to replace the current manual system into computerized system. In general, the proposed system handles all the data and work done efficiently and effectively than the current existing system and also solve the problems that are occurred in the current existing system. After all it reduces the work load and give better service for the user.

3.1 Functional Requirements

The proposed system is expected to provide functionalities related to clinic management system. For better understanding of our system we divided the functionalities of our system by actors' role. For a better understanding of our system, we divided the functionalities by modules. The proposed system is expected to provide the following services.

1. Registration Module

Here the proposed system should allow actors of the system the following activities

- Create account
- Add, update, delete, and view (Employee, Drug, and Patient).
- Order

2. Pharmacist Module

- Request drug
- Give/delegate prescribed drug
- Check expired date

3. Lab Module

- Send lab result

4. Schedule Module

- Manage schedule

5. Patient management Module

- Generate sick leave, refer
- Set appointment
- Send feedback

6. Report Module

- Generate report
- View report

3.2 Nonfunctional Requirements

In this section, we have discussed the quality attributes in which our system should have to perform the functional requirements.

3.2.1 User Interface and Human Factors

The window format and the form prepared for the information are easy for users they can easily understand the system, shall be designed according to standard and the system shall replace the existing system. The user who don't know how to use the system they can easily access the function of the proposed system because the system will provide user documentation for its user.

3.2.2 Hardware consideration

The organization should have computers having typical storage capacity and processing speed.

The software product to be developed should run on existing standard computers. The system will be portable that can run on any type of computer and it supports any type of browsers.

3.2.3 Security Issue

The system should have a security privilege that secures the system and also, there must be a physical security that secures especially the server computer. That means the server computer is only allowed for the server admin. Security is implemented by requiring users to login before using the system. The login procedure is password protected. The system has to be well protected unauthorized access by using Bcrypt which is password hashing function for Node JS.

- All major transaction of the system must be logged to database and seen only by the administrator. If the system has login part the external intrusions or malicious users cannot get access.
- Our proposed system has session control: when the user registered once the system can save the user data.

3.2.4 Performance consideration

The end user computer should have medium processor and the server computer should have large processor.it is measured by its speed of processor. The proposed system performs its operations within a minimum amount of time and the user gets the expected result within a few seconds and the system is effective.

Response Time upon request for inquiry the system under normal condition should display results as quickly as possible.

Processing Time Since the system is developing with efficient programming language and database upon request for user's activities the system under normal condition should process the request as quickly as possible.

3.2.5 Error Handling and Validation

Our system handles error by showing the message "invalid input "when the user enters invalid input. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system and the system must handle the error.

3.2.6 Quality Issues

The system, it will be tested, verified and, validated and if the error occurs during the execution time it will give an exception. And, because the user will be involved in testing the quality of the system, it will satisfy the user's needs.

Availability: the system should have to be function at any given time. Which means our proposed system gives service 24 hours per day with maximum response time. The server should be always on to be available.

Reliability: the proposed system will minimize crash during its runtime, since more than one user could use the system simultaneously. The system should be reliable and matured enough in giving its service.

3.2.7 Backup and Recovery

The system should have backup using external hard disk. The backup is taken weekly and the system will store what filled on the form and it will help the user to get the information they filled when the connection comes back and it also helps the users those who forgets their password through sending a security number on their phone number which they used during registration.

3.2.8 Physical Environment

The system must be compatible with any environment. But for more feature we recommend that the system to deployed on the cloud that is free from any disaster.

3.2.9 Resource Issues

Our proposed system needs the resource to deploy the system. Resource like internet connection, database server, application server, and a client computer.

3.2.10 Documentation

The System has a well-defined document that helps to easily maintain the system and we will also prepare a short and precise help file on how to use the system for the system users. It will have a helping page to guide the user of the system and to show the process of how they will have to use it.

CHAPTER FOUR

4. SYSTEM ANALYSIS

In this dynamic world, the subject of System Analysis mainly deals with software development activities. We conducted it for the purpose of studying our system or its parts in order to identify its objectives. System Analysis is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. This chapter explains the System analysis by designing use case diagrams, use case description for each use case identified in the use case diagram, object model class diagram, sequence diagram, and activity diagram [2].

4.1 System Model

To produce a model of the system which is correct, complete, and consistent we need to construct the analysis model which focuses on structuring and formalizing the requirements of our system. The analysis model contains three models: functional model this model deals with the functionality of the system, what is expected from the system, the services that are given by the system. The functional model is generally described by the use case diagram. Object model this model describes the structure of the system. The object model includes attribute, object, and association. Dynamic models deal with the internal behavior of the system and also how one-use case executes, series of activities that one-use case executes as well as the state. The dynamic model is described by the sequence and activity diagrams.

4.2 Use Case Model

The Use Case Model describes the functionality of the WKU students' clinic management system. It represents a discrete unit of the interaction of a user (actors of the system) with the WKU students' clinic management system. Use Case has a description that describes the functionality that will be built in the proposed system. Use Cases are typically related to 'actors', that is a human or machine entity that interacts with the system to perform meaningful work. But in our proposed system there is no need to interact with other external systems and due to this, our actors are only humans who will use the system.

4.2.1 Use Case Diagram

We have used use case diagrams in order to capture the functional requirements of the system. It is the functionality of the system or the service provided by the system [2]. In our use case diagram, we have considered the following elements.

- **Use Case:** This is the functionality of our WKU students' clinic management system which has a direct interaction with the user.
- **Actor:** any entity that interacts with our system.
- A relationship between the use cases.

We identified both actors and use cases (functionalities) that exist in the WKU students' clinic management system as the following:

1. **System administrator**:- can
   - Create, update, and delete user account for users
   - Restore database
2. **Manager**:- can
   - Add, update, and view employee
   - Create, view, update, and delete schedule
   - View status
   - View Report
3. **Record officer: -** can
   - Add, update, and view patient
   - Assign patient to doctor

4. Doctor:- can

- Manage patient history
- Order drug to a patient
- Order lab
- Manage appointment
- Generate refer
- Generate sick leave

5. Pharmacist:- can

- Request drug

- Add drug
- Give prescribed drug for patient
- Check expired date of drugs

6. Laboratory technician:- can

- View lab order
- Send lab result

7. Patient:- can

- View queue
- Send feedback

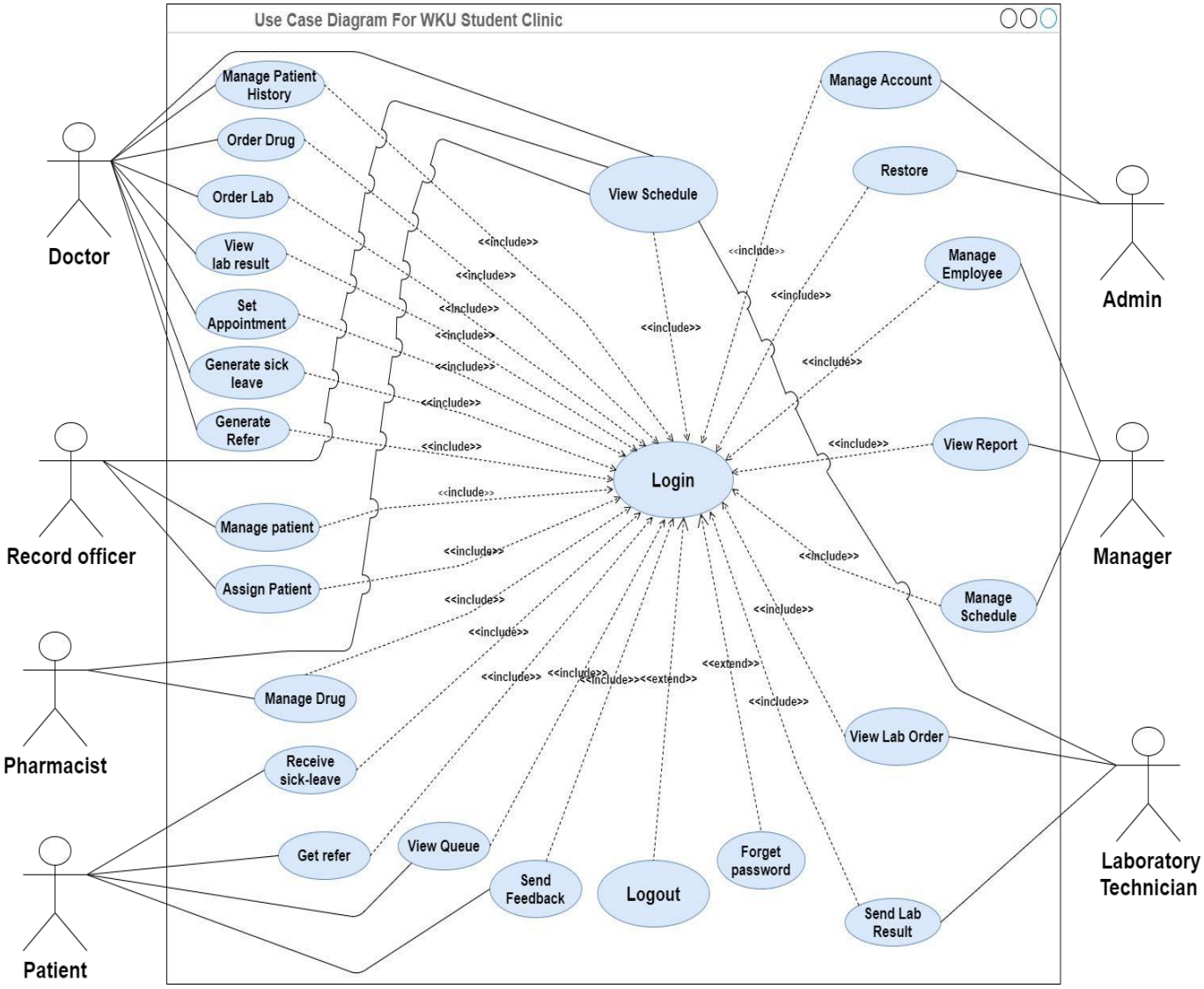


Figure 4. 1 Use case Diagram for WKU student clinic

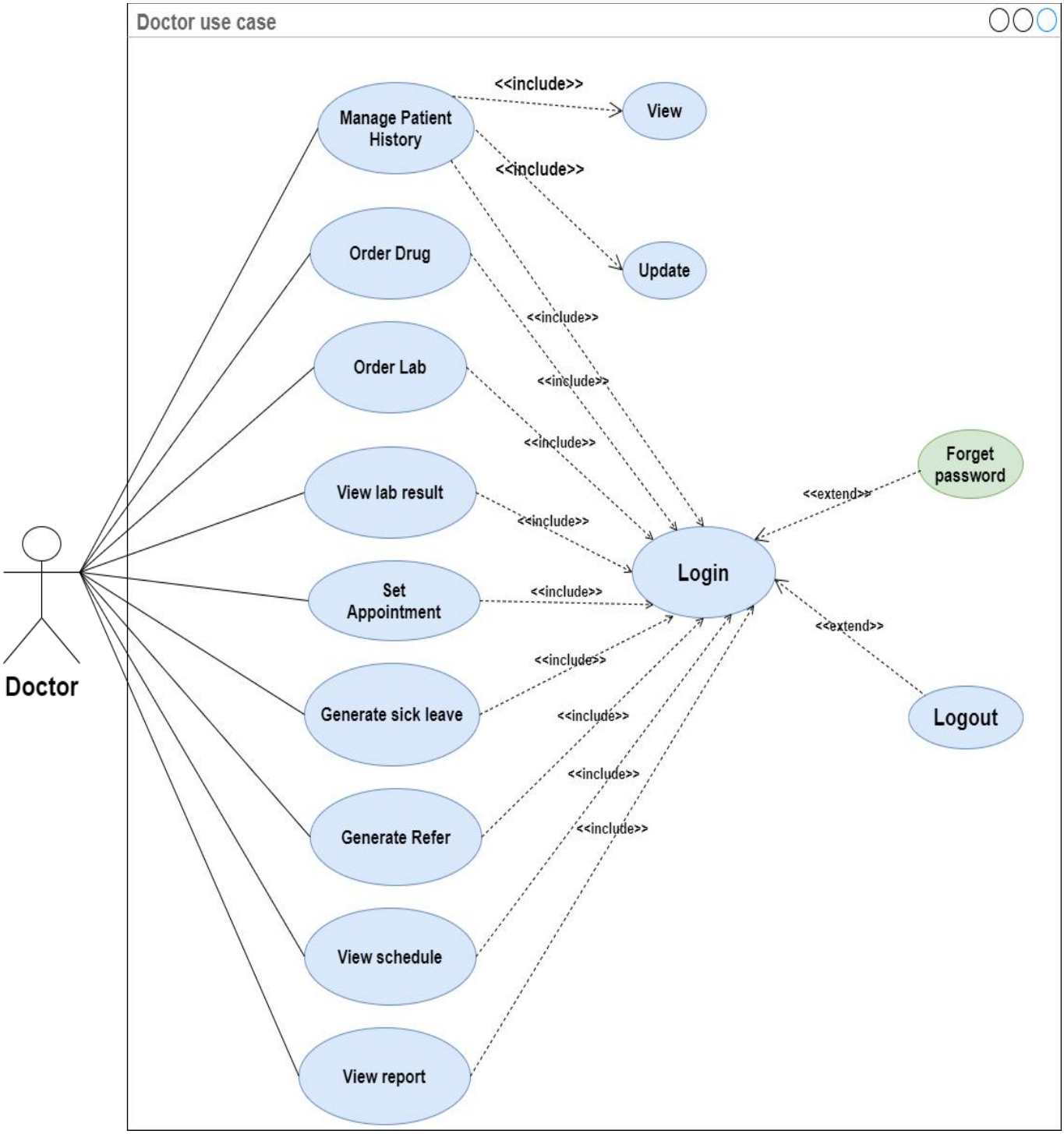


Figure 4. 2 Detail use case diagram for Doctor

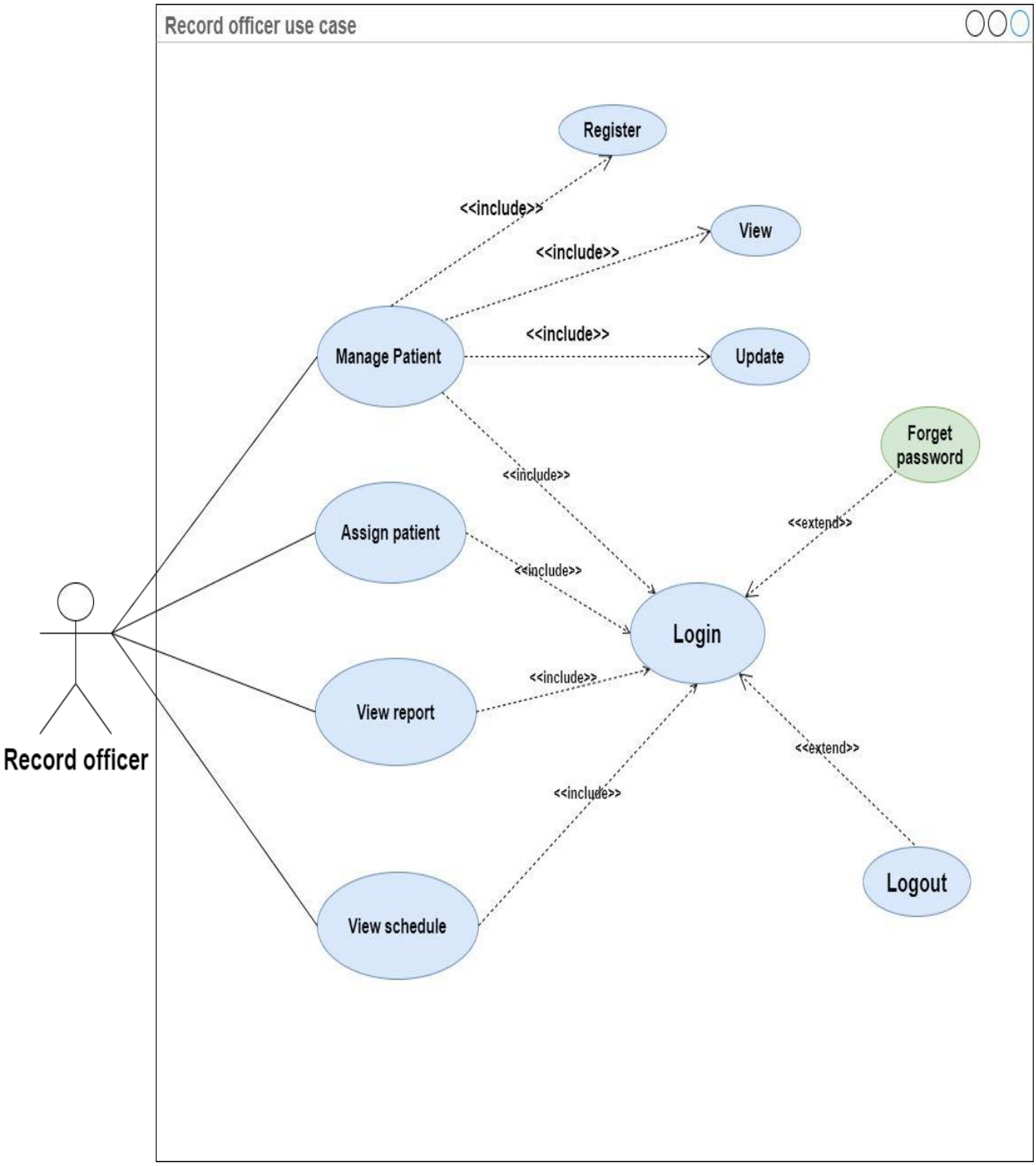


Figure 4. 3 Detail use case diagram for Record officer

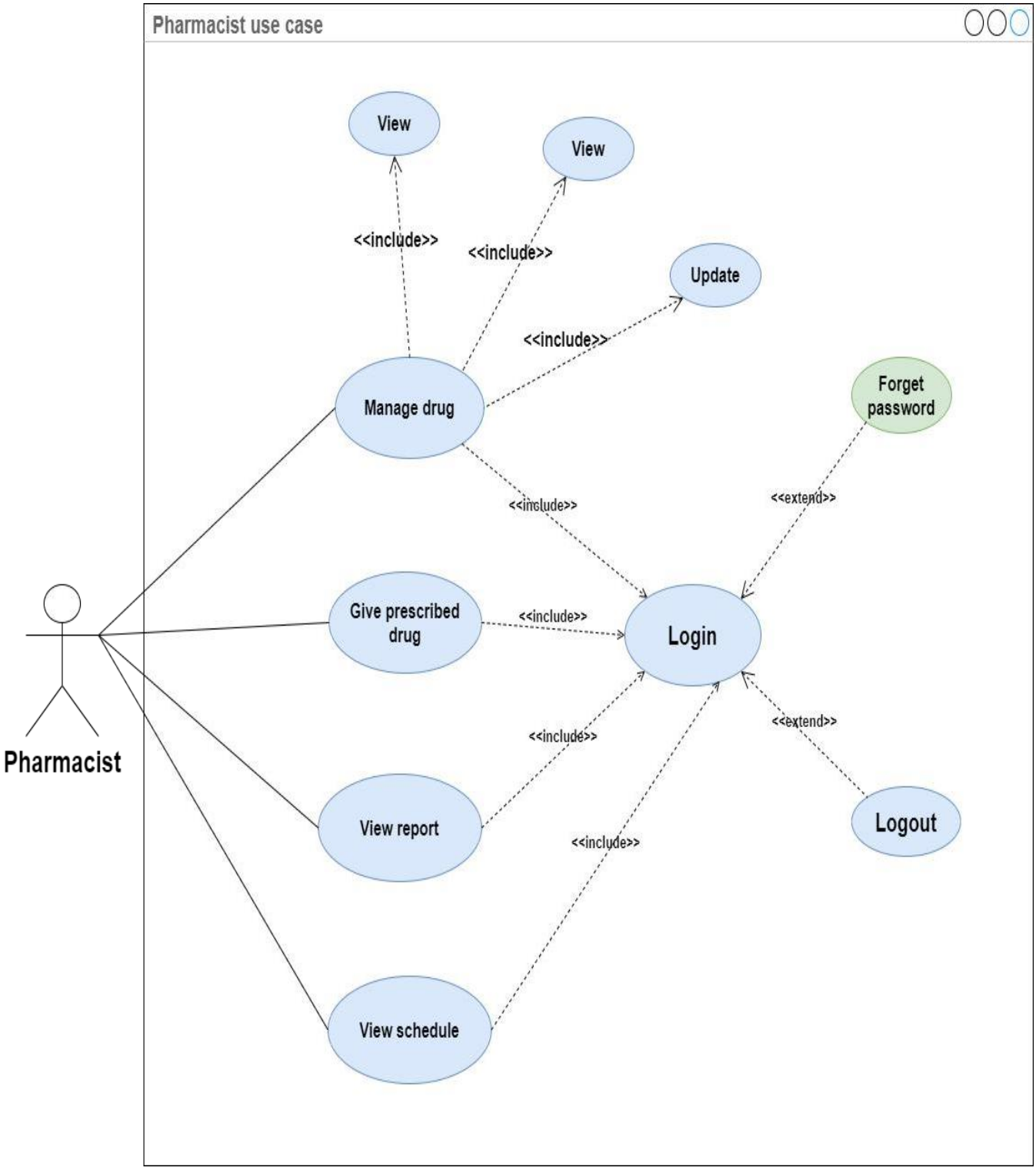


Figure 4. 4 Detail use case diagram for Pharmacist

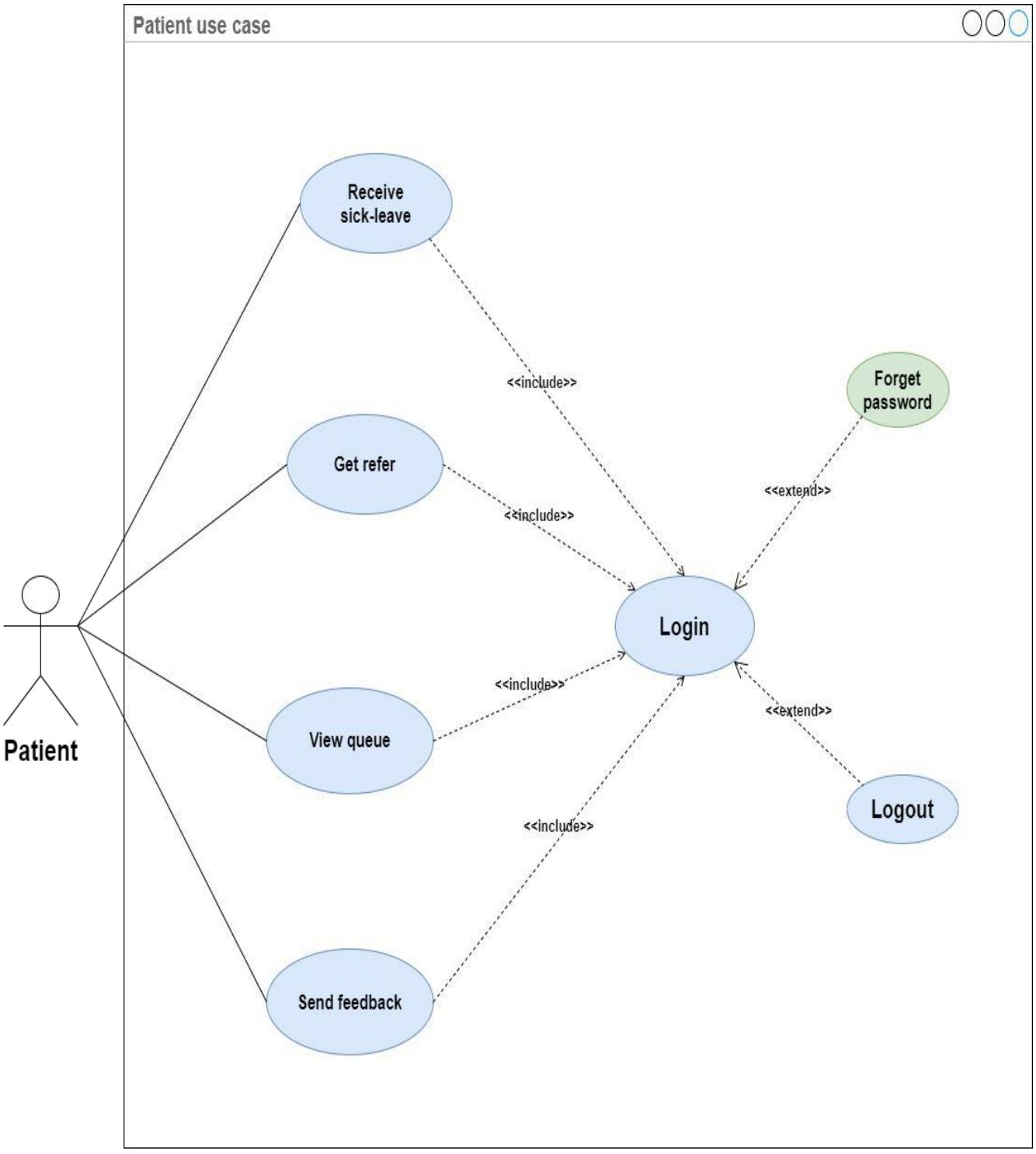


Figure 4. 5 Detail use case diagram for Patient

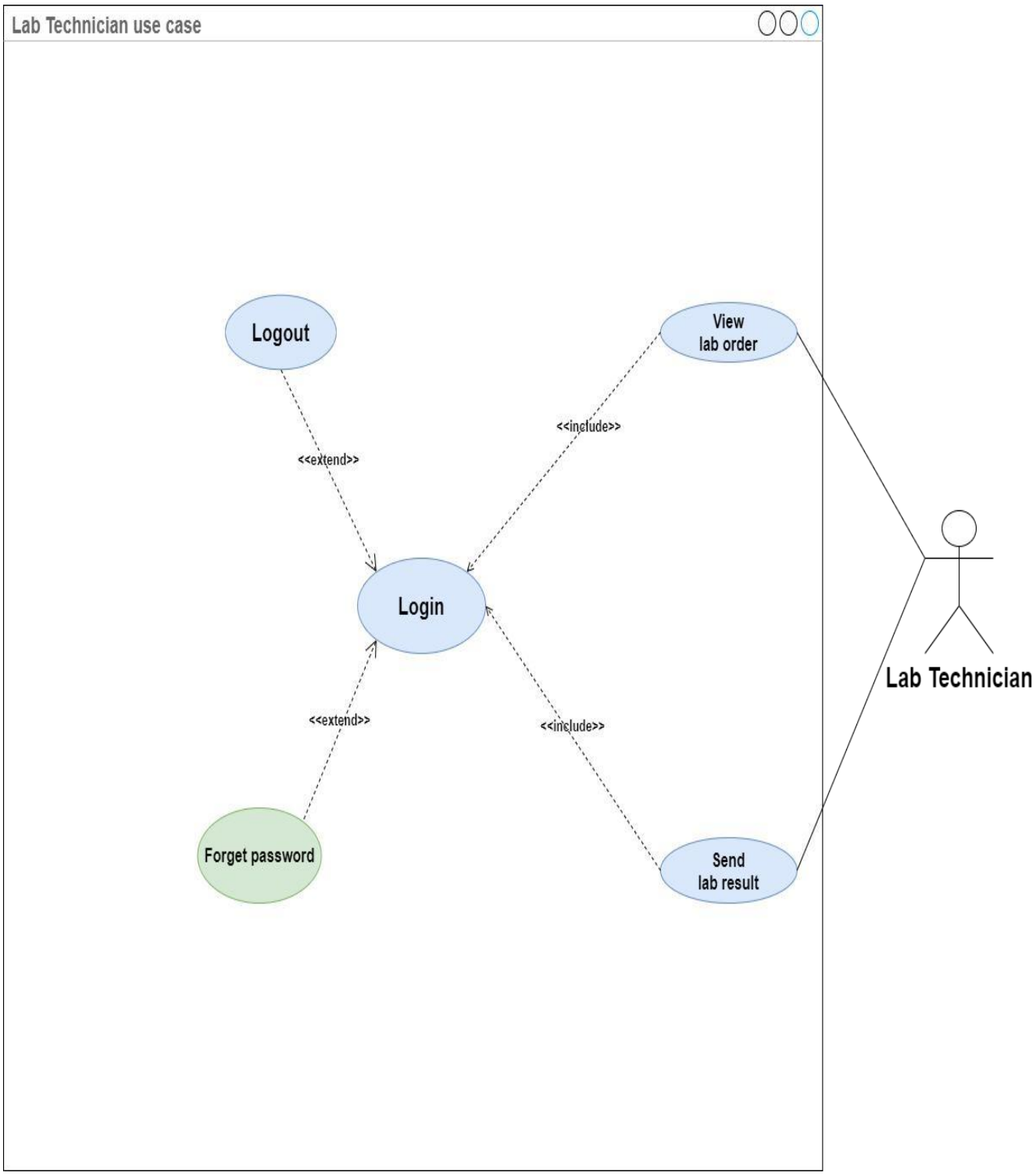


Figure 4. 6 Detail use case diagram for Lab technician

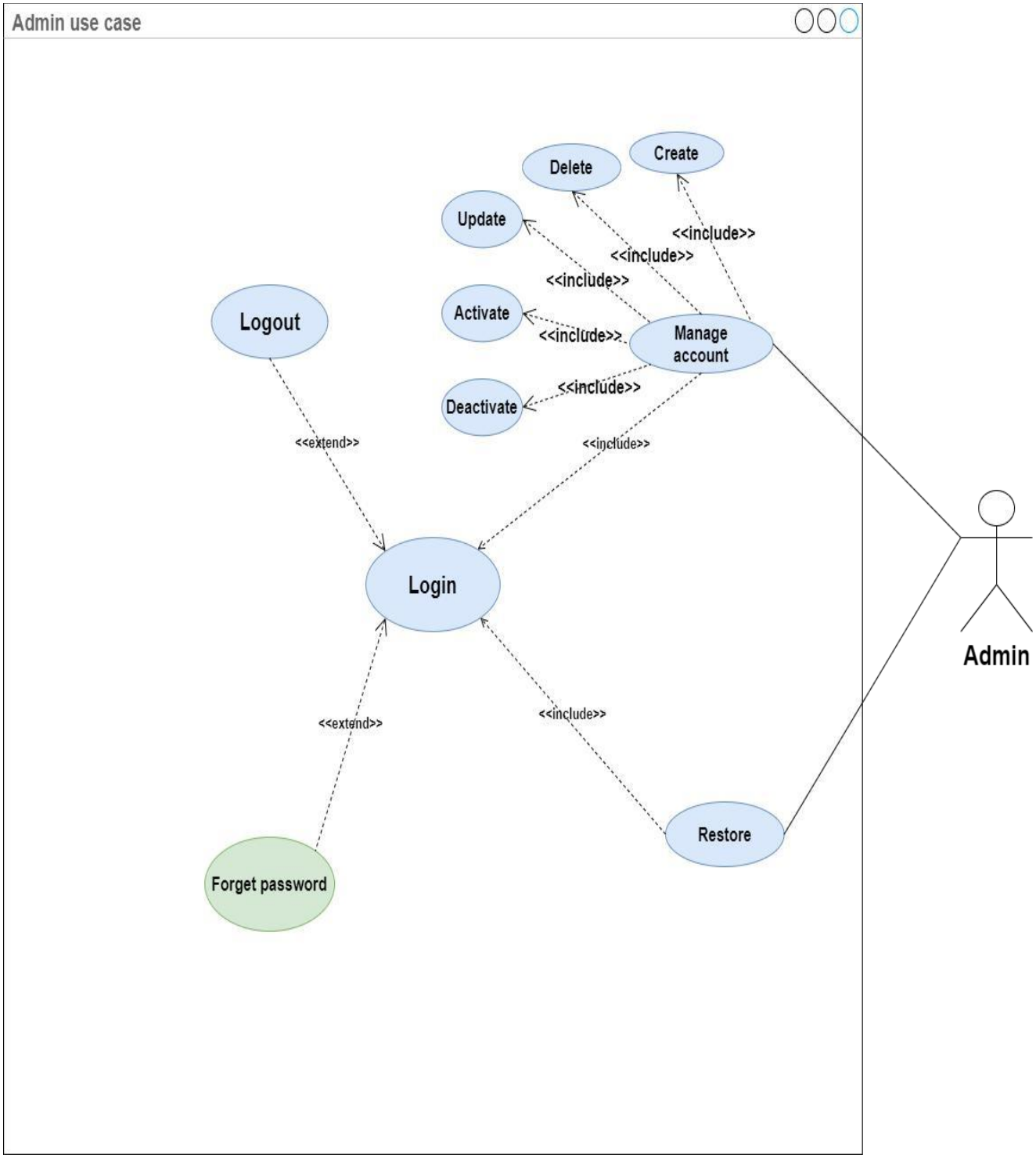


Figure 4. 7 Detail use case diagram for Admin

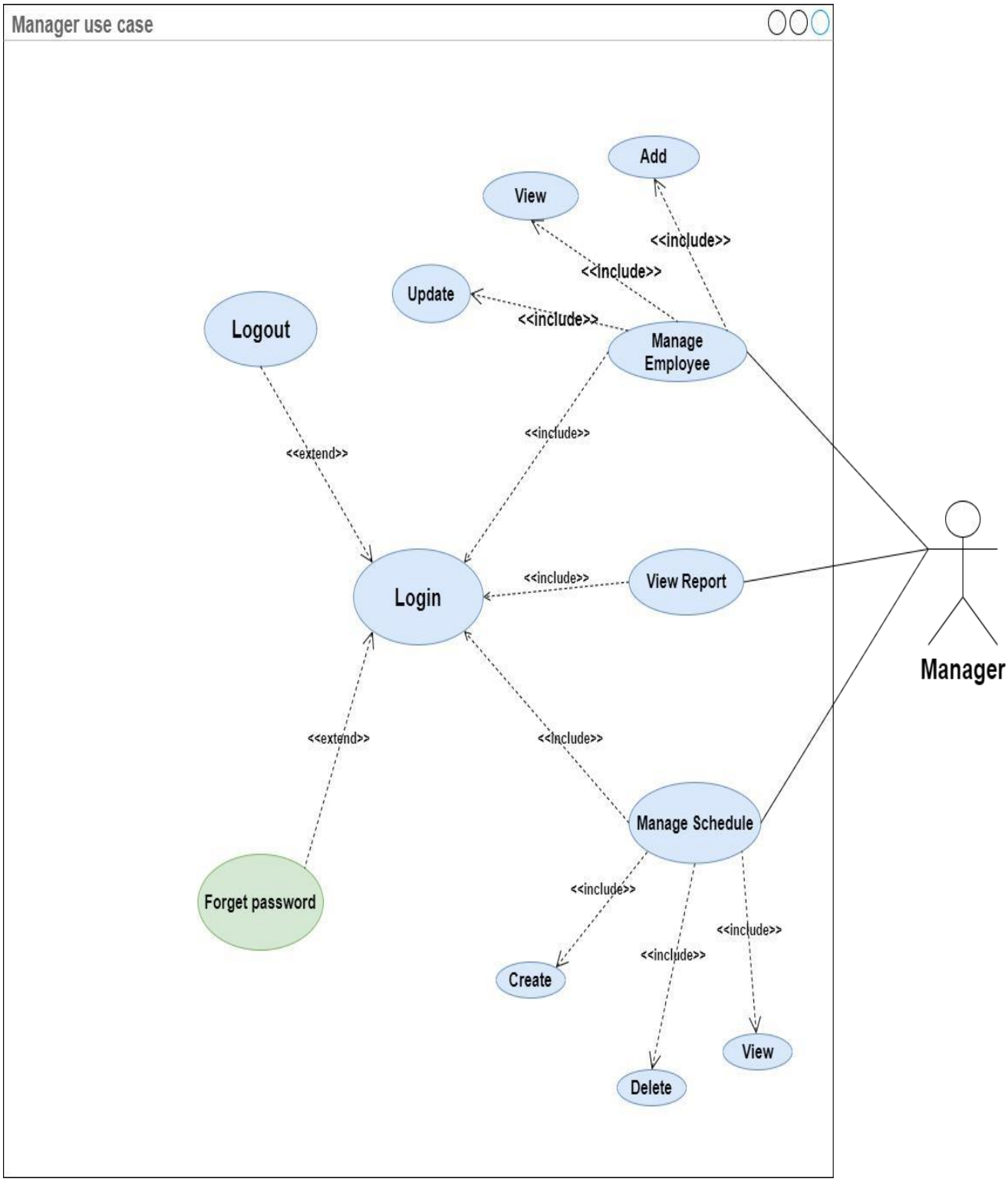


Figure 4. 8 Detail use case diagram for Manager

4.2.2 Use Case Description

A use case is a written description of how users will perform tasks on a website. It outlines from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal if fulfilled.

As use case is a list of steps, typically defining interaction between a role or actor and the system to achieve a goal some of the use cases of our system are the following:-

Table 4. 1 Login Use Case Description

Use case id	UC01	
Use case name	login	
Description	Validate the user to enter the system.	
Actor involved	All actors listed above	
Precondition	The user must have an account	
Post condition	The actor's login to their privileged page	
Basic course of action	**User action**	**Basic course of action**
	1. Users are on the homepage and press the login button.	**2**. The system displayed the login form.
	3.The user fills his or her user name and password	**4.** The system checks the validity
		5. The displays logged in page
	6.Use case ends	

The alter native course of action(if the user enters the wrong user ID and/or password)	If the user name and password is not valid at step 4 of the basic course of action	
	User action	**System action**
		4.1. The system displays an error message
	4.2. The user views the error message click the ok button	**4.3**.Go to step 2 of the basic course of action

Table 4. 2 Registers Patient Use Case Description

Use case id	UC02	
Use case name	Register patient	
Actor involved	Record officer	
Precondition	The record officer must have a username and password.	
Post condition	The user gets access to the system according to his/her predefined system and his/her medical file.	
Basic course of action	**User action**	**Basic course of action**
	1.The user click on manage patient link	**2**.The system will display a register, delete update links
	3.The user selects register button click on it	**4.**The system will display registration form
	5.The users fills the form and click on submit button	**6**.The system checks the validity of the input

		7. The adds it to the database and display successfully registered
		8.The use case ends
An alternative course of action	If the input is not valid in step 4 basic course of action	
	User action	**System action**
		2.1.The system displays an error message
	2.2. The user views the error message and click the ok button	**2.3.**Go to step 2 of the basic course of action

Table 4. 3 Create Account Use Case Description

Use case id	UC03	
Use case name	Create an account	
Actor	System administrator, Manager	
Pre-condition	User must be login to the system first	
Post condition	User account created successfully	
Basic course of action	**User action**	**Basic course of action**
	1.The user first click on add link	**2.**The system will display the create

		account page
	3.The user fills the form and click on add button	**4**. The system checks the validity of the input
		5. The system adds it to the database and display successfully created
		6. End use case
An alternative course of action (if the user enters the wrong user id and/password)	If the input data are invalid in step 4 of the basic course of action	
	User action	**System response**
		4.1 The system display error message
	4.2 The user views the error message and clicks the ok button	**4.3** Go to step 2 of basic course of action

Table 4. 4 Update Account Use Case Description

Use case id	UC04
Use case name	Update account
Actor	System administrator, manager
Description	To update stored information's
Precondition	There should be stored information

Post condition	User account updated successfully	
Basic course	**User action**	**System response**
	1.The manager has to login to the system	**2.**The system display available page
	3.The manager selects update link	**4**.The system displays the users update account form
	5.The user enters the staff id and press search button	**6.**The system checks the existence and display the information
	8.User fill the form again and click the update button	**9**.The system checks the validity of the input
		10. The system updates the information
		11.End use case
An alternative course of action(if the user enters the wrong user ID and/or password)	If the input data are invalid in step 6 of the basic course of action	
	User action	**System response**
		6.1.The system display error message
	6.2. The user views the error message and clicks the ok button	**6.3**.Go to step 3 of the basic course of action

Table 4. 5 Generate refer Use Case Description

Use case ID	UC05	
Use case name	Generate refer	
Actor	Doctor	
Description	The process begins when the system user needs to give referral	
Precondition	The actor should be properly authenticated	
Post condition	The doctor gives the refer	
Basic course	**User action**	**System response**
	1.The doctor initiates the system to generate a medical certificate	**2.**The system shows option/refer medical certificate
	3.The doctor selects refer	**4**.The system display referral form
	5.The doctor fills the form	**6.**The system validates and displays a confirmation
	7.The doctor order print	**8.**The system print filled referral form
An alternative course of action	System displays error message the request to retry	
	User action	**System response**

		6.1 The system displays an error message
	6.2 The user views the error message and clicks the ok button	**6.3** Go to step 2 of the basic course of action

Table 4. 6 Order Drug Use Case Description

Use case ID	UC06	
Use case name	Order drug	
Actor	Doctor	
Description	Informing the pharmacist to delegate appropriate drug to the patient	
Precondition	The doctor must login to the system	
Post condition	The patient receive the drug from the pharmacist	
Basic course of actin	**User action**	**System response**
	1.A doctor click on order drug link	**2.**The system display the order drug form
	3**.**The doctor fills the form and clicks the submit button	**4**.The system checks the input data
		5. The system sends it to the pharmacist
		6.End use case

An alternative course of action (if the doctor fulfills the form incorrectly)	If the input data are invalid in step 4 of the basic course of action	
	User action	**System response**
		4.1 The system display error message
	4.2 The user views the error message and clicks the ok button	**4.3** Go to step 2 of the basic course of action

Table 4. 7 Give prescribed drug Use Case Description

Use case id	UC07	
Use case name	Give prescribed drug	
Actor	Pharmacist	
Description	The user approves the requested drug order from the doctor	
Post condition	The information will be stored to the patient history	
Basic course	**User action**	**System response**
	1.User clicks on the drug order link	**2.**The system displays the orders from the doctor
	3.User see the order and click on accept request link	**4.**The system stored the accepted order to the database
		5.Use case end

Table 4. 8 View Report

Use case ID	UC08	
Use case name	View report	
Actor	Manager	
Description	The process begins when the system user needs to view reports	
Precondition	The actor should be properly authenticated	
Post condition	The manager views report	
Basic course of actin	**User action**	**System response**
	1.Manager initiates the system for view report	**2.**The system displays the report window
	3.The manager views the report and clicks on the ok button	4. Use case end

Table 4. 9 Assign Patient Use Case Description

Use case ID	UC09	
Use case name	Assign patient	
Actor	Record officer	
Precondition	The patient file must found from the system	
Post condition	The patient is assigned to a doctor	
Basic course of action	**User action**	**System response**

	1.Actor click on the patient assign button	**2**.The system will display a patient assign form
	3.The user fills out the form and click the assign button	**4.**The system checks if the form is filled out correctly
		5.The system displays an acknowledgment message
		6.The use case ends
An alternative course of action	The form is not field out correct in step 3	
	User action	**System response**
		3.1. The user inform the user that the form has not been filled out correctly
	3.2. The use case continues at step 3 of the basic flow of event	

Table 4. 10 Manage Patient History Use Case Description

Use case ID	UC10
Use case name	Manage patient medical file
Actor	Doctor
Precondition	The actor must be login to the system first

Post condition	The actor treat the patient	
Basic course of action	**User action**	**System response**
	1.The doctor selects the medical file link	**2**.The system display the available page
	3.The doctor selects the medical file link	**4.**The system display files
	5.The doctor enters the patient's id and clicks the search button	**6**.The system searches the patient's name
	7.The doctor views the file and gives the treatment	**8**.The system displays the patient medical file
	9.The doctor selects the update button	**10**.The system displays the patient file update form
	11. The doctor enters the result and clicks the submit button	**12**.The system checks validity of the input
		13. The system saves the patient history in the patient medical file
		14.Use case ends
An alternative course action(if doctor fulfil the form incorrectly)	If the input is not valid in step 11 basic course of action	
	User action	**System response**
		12.1 The system displays an error message

	12.2 The user views the error message and clicks the ok button	**12.3**.Go to step 11 of the basic course of action

4.2.3 Use Case Scenario

- Login use case scenario

Let assume Mr. Daniel is doctor, and if he wants to log in to the system the first step will be entered to the system home page then he should have login password and user name if he has he will click to the login link on the home page then after the login page will be opened for him. Then he will enter his username and password and click on login button the system will validate his authentication and will open his privileged page.

- Create account use case scenario

Let assume Mr. Amsalu is a manager of the clinic, and if he wants to register an employee and create an account for him/her he will first login to the system and then he will click on the manage account link, the system will open for him a page which has their option (create, update, search) then she will click on create account button and the system will display a form for him, then he will fill all the field and give him/her a username and password.

- Register Patient Use Case Scenario

Let assume Mr. Markos is Doctor in the clinic, and if one patient who should take a lab test comes to the clinic the doctor will send the order through the system. To do this activity the Doctor first log in to the system and the click on the lab order link, then the system will display a form to fill, the doctor fills required fields and then click on send button.

- Send Lab Result Use Case Scenario

Let assume Mr. Abera is laboratory technician of the clinic, and view the request from the doctor and tests the patient the returns back the results of testing to the doctor by clicking send lab result link.

Dr. Solomon is doctor of WKU students' clinic so he treats patients according to their illness therefore as the recording officer send the patient to the doctor. The doctor will treat the patient and send him if necessary for laboratory test then as the doctor receives laboratory result will order drug for the patient, therefore, the patient will receive the drug from pharmacist finally the Doctor will report to the Manager.

4.3 Object Model

An object model is a description of an object-oriented architecture, including the details of the object structure, interfaces between objects, and other object-oriented features and functions. In this section, we discuss the object model class diagram and data dictionary. In the class diagram, we have described diagrammatically the conceptual relationship among objects and class of the proposed system (WKU students' clinic management system).

4.3.1 Class Diagram

The class diagram represents the static view of the WKU students' clinic management system. The class diagram is not only used for visualizing, describing, and documenting different aspects of our system but also for constructing executable code of the proposed system. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object- oriented systems because they are the only UML diagrams that can be mapped directly with object-oriented languages [3].

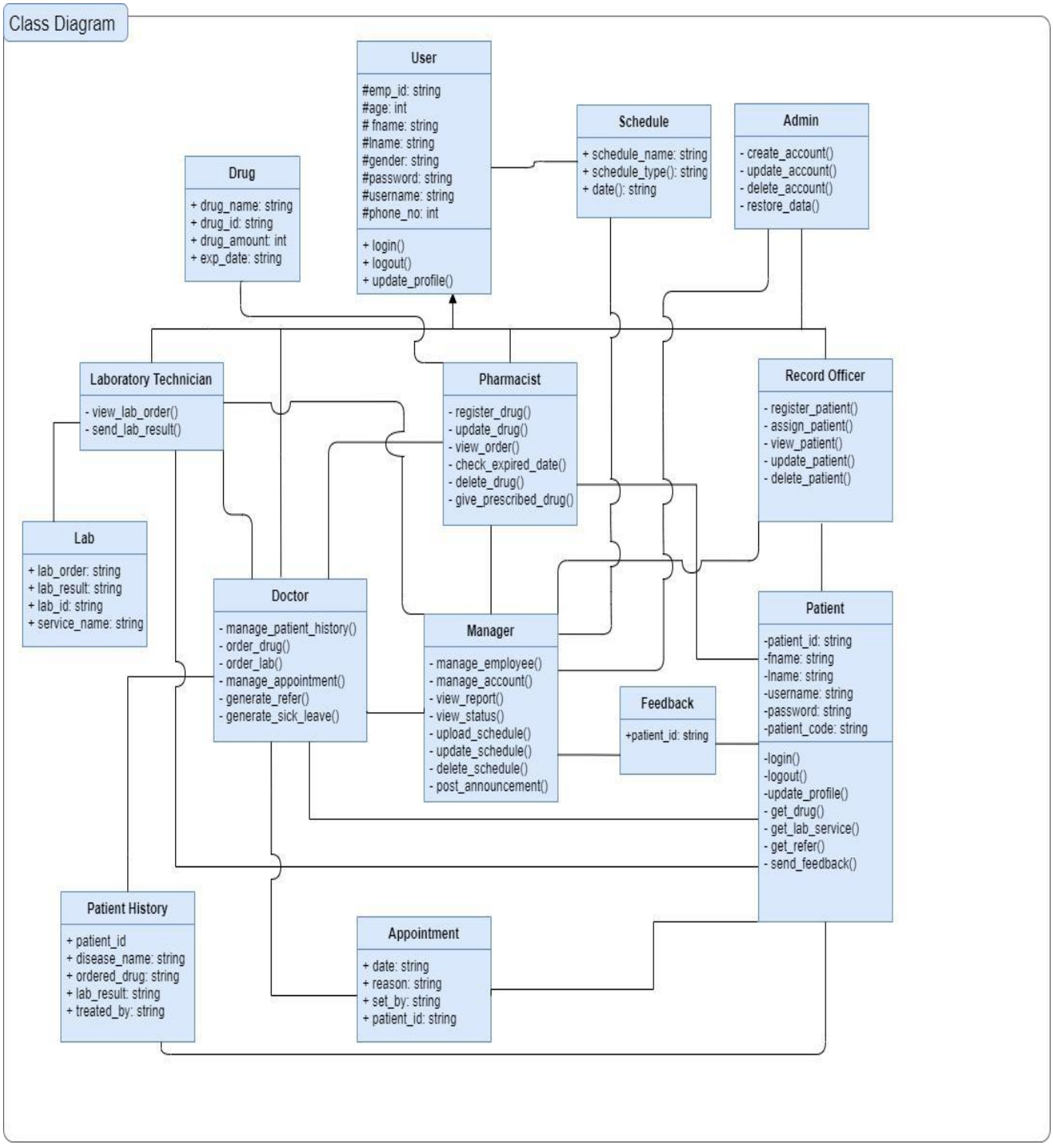


Figure 4. 9 Class diagram for WKU student clinic

4.3.2 Data Dictionary

Table 4. 11 Drug data dictionary

Attributes	Data type	Data size	Key constraint	Constraint
Drug id	String	20	Primary key	Not Null
Drug name	String	20		Not Null
Drug amount	String	20		Not Null
Expired date	String	20		Not Null

Table 4. 12 Lab data dictionary

Attributes	Data type	Data size	Key constraint	Constraint
Lab id	String	20	Primary key	Not Null
Service name	String	30		Not Null
Lab order	String	30		Not Null
Lab result	String	30		Not Null

Table 4. 13 Patient history data dictionary

Attributes	Data type	Data size	Key constraint	Constraint
Patient id	String	20	Primary key	Not Null
Disease name	String	30		Not Null

Ordered drug	String	20		Not Null
Lab result	String	30		Not Null
Treated by	String	20		Not Null

Table 4. 14 Appointment data dictionary

Attributes	**Data type**	**Data size**	**Key constraint**	**Constraint**
Patient id	String	20	Primary key	Not Null
Date	String	20		Not Null
Reason	String	30		Not Null
Set by	String	20		Not Null

Table 4. 15 Patient data dictionary

Attributes	**Data type**	**Data size**	**Key constraint**	**Constraint**
Patient id	String	20	Primary key	Not Null
First name	String	20		Not Null
Last name	String	30		Not Null
Age	Int	20		Not Null
Gender	String	7		Not Null
User name	String	20		Not Null

Password	String	15		Not Null
Phone no	Int	13		Not Null

4.4 Dynamic Model

Dynamic modelling describes those aspect of the system that are concerned with time and sequencing of the operations. It is used to specify and implement the control aspect of the system.

4.4.1 Sequence Diagram

Sequence diagrams are a type of UML diagram that show how objects in a system or classes within code interact with each other. Particularly these diagrams show interactions in the order they take place in other words, they show the sequence of events [3].

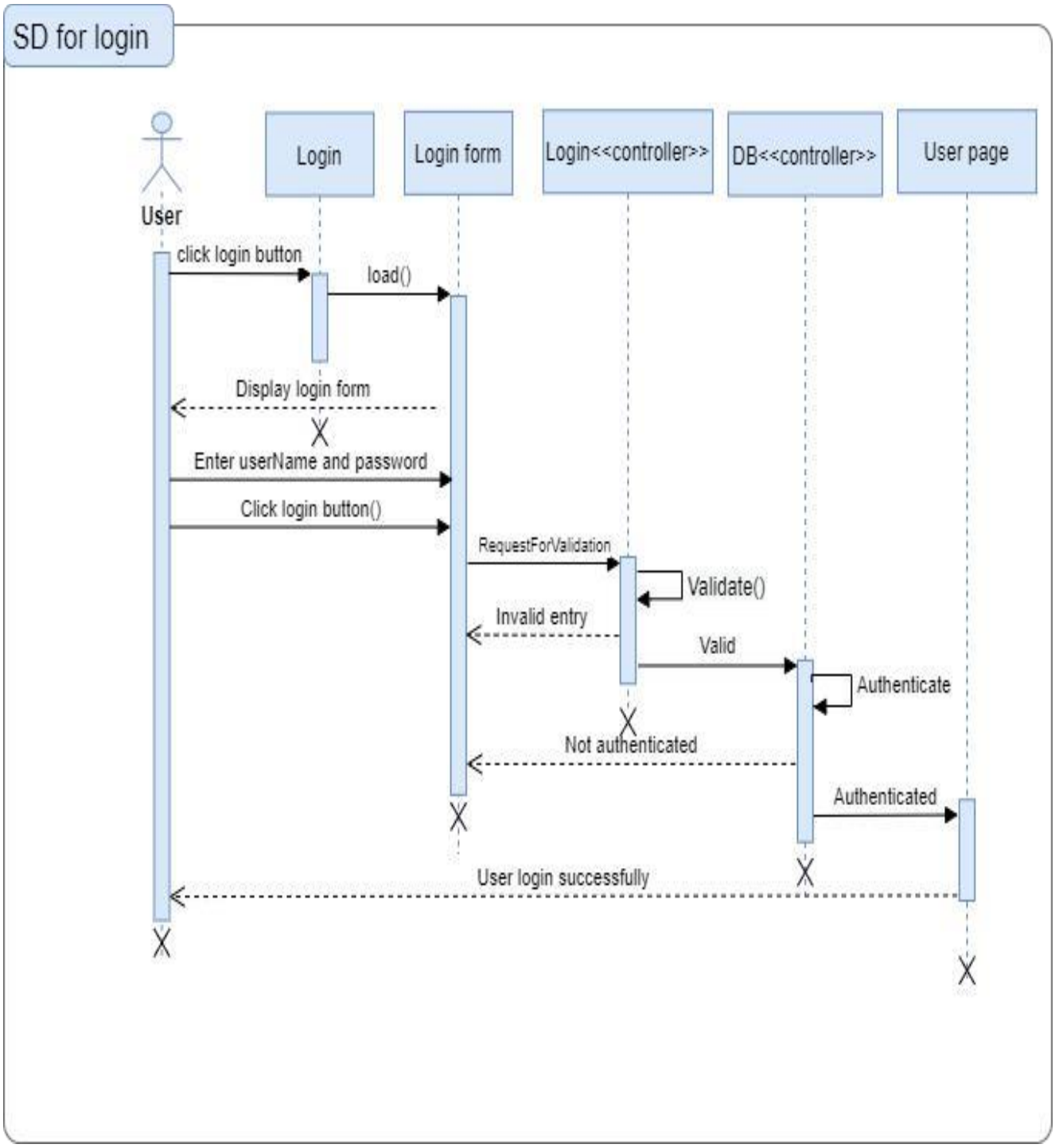


Figure 4. 10 Sequence diagram for user login

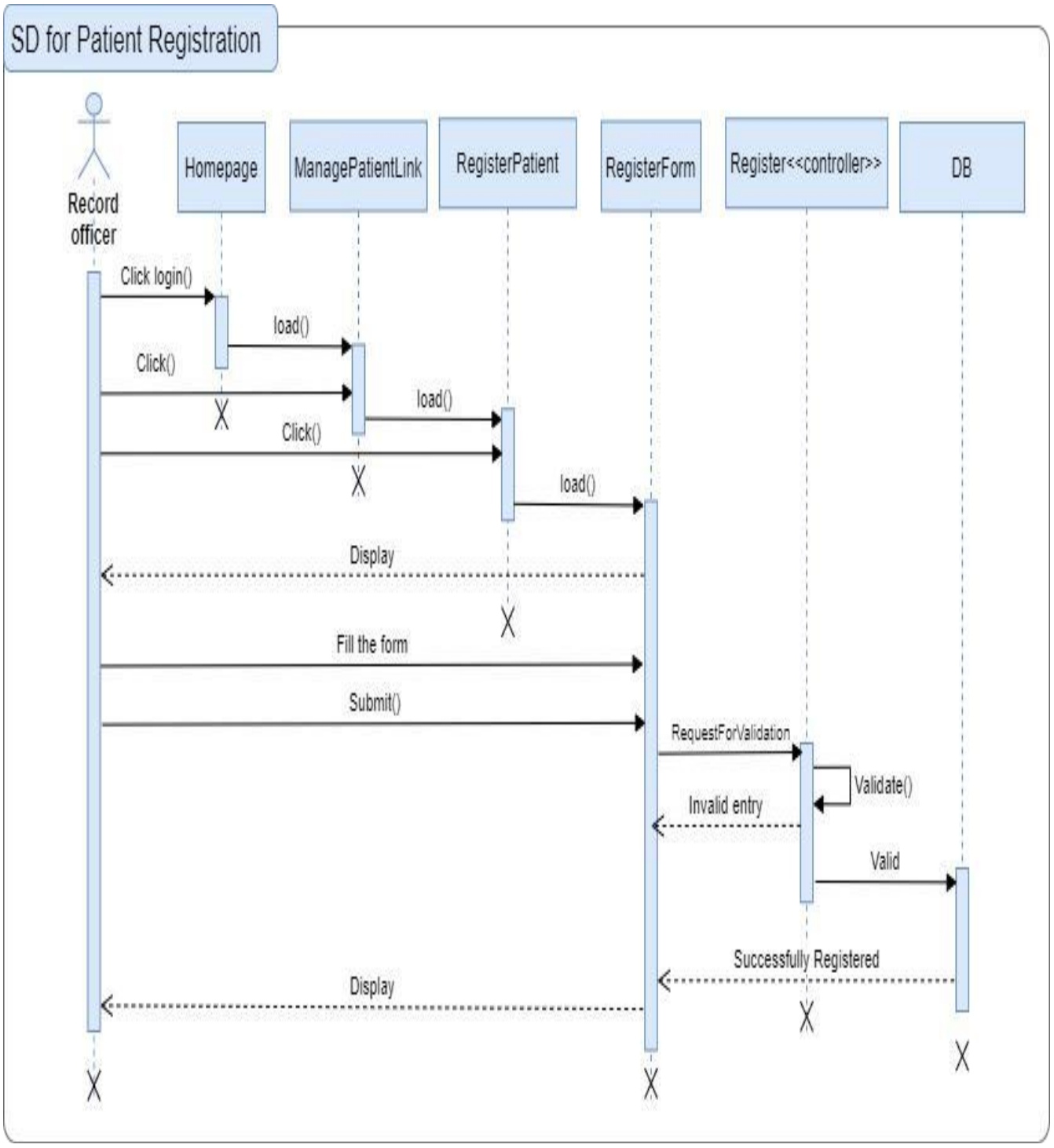


Figure 4. 11 Sequence diagram for patient registration

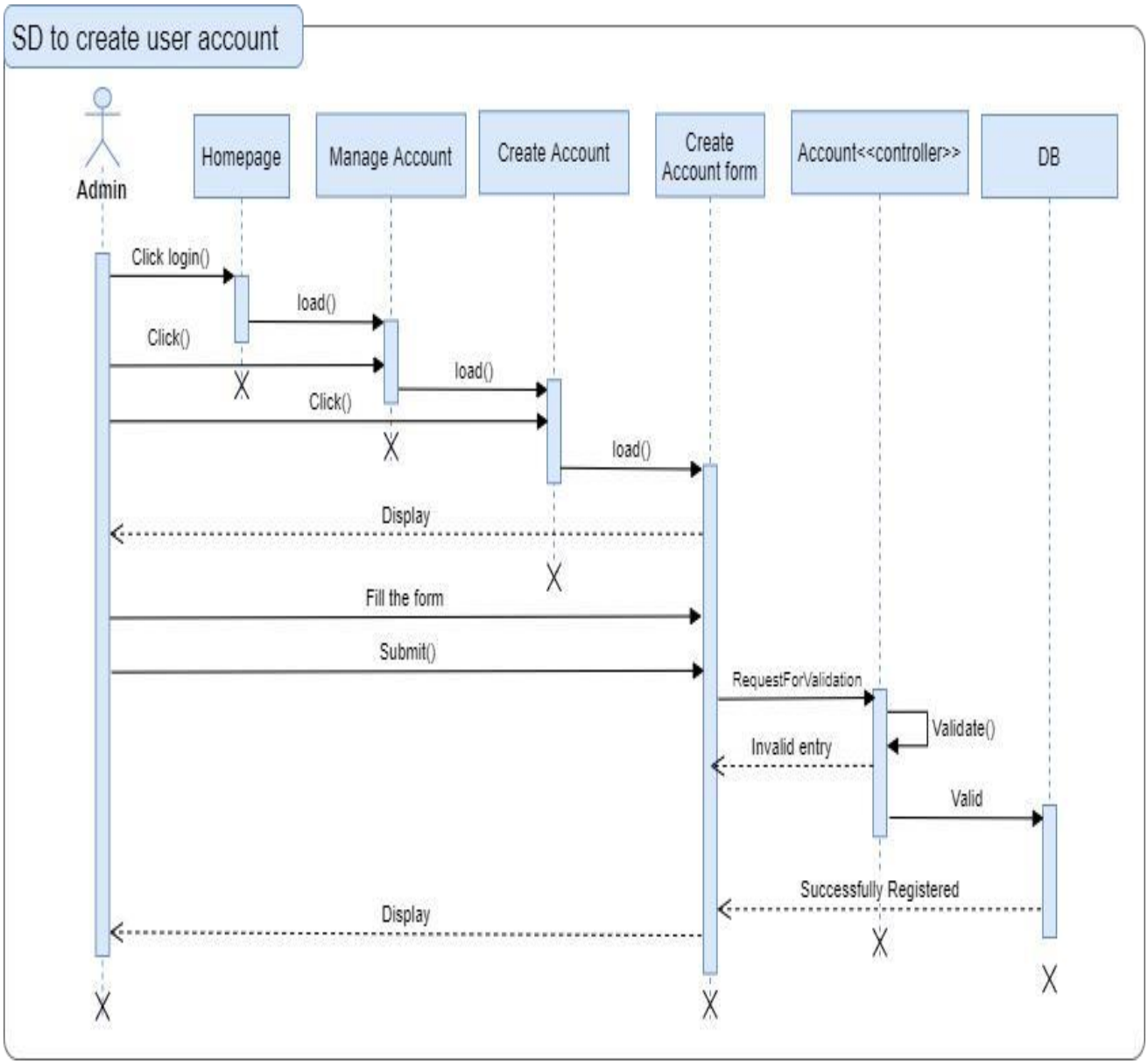


Figure 4. 12 Sequence diagram for account creation

4.4.2 Activity Diagram

It is used to understand the flow of works that an object or component performs and visualize the interaction between different use cases. The activity diagram depicts the logic for how someone to do the system. In the activity diagram the field circle indicates the starting point of the activity and the field circle with a border represents the end point of activity [3].

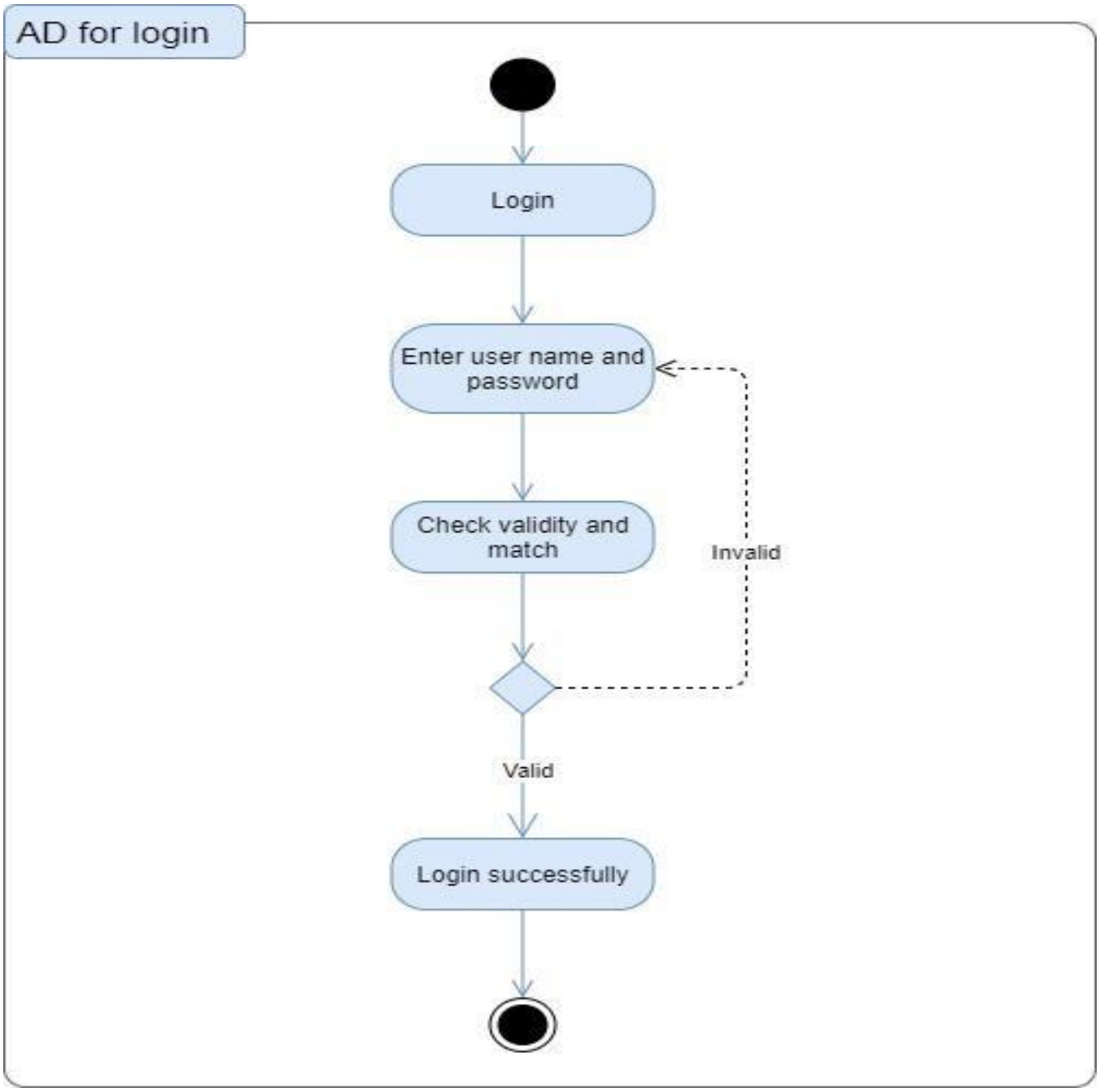


Figure 4. 13 Activity diagram for user login

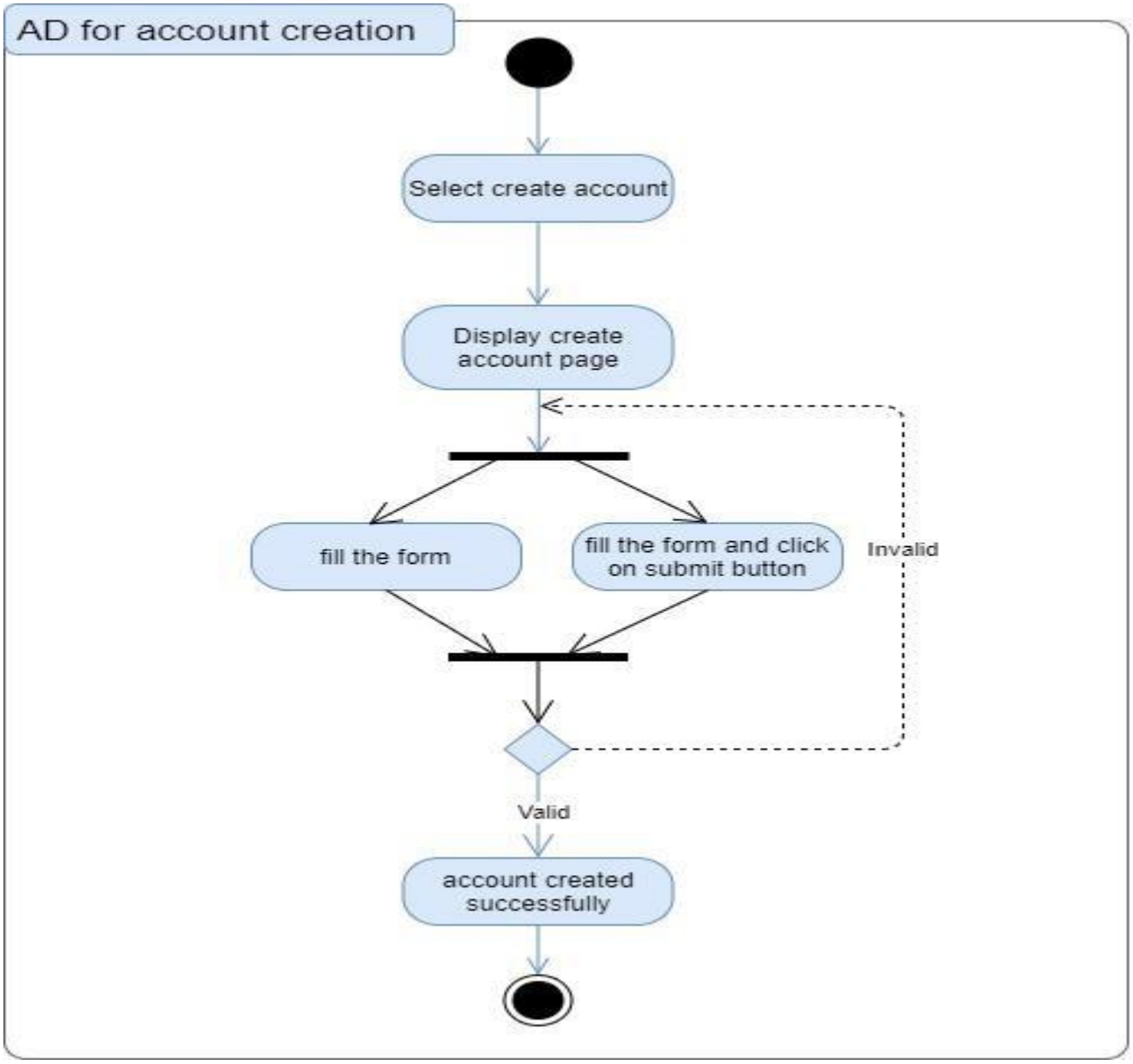


Figure 4. 14 Activity diagram for account creation

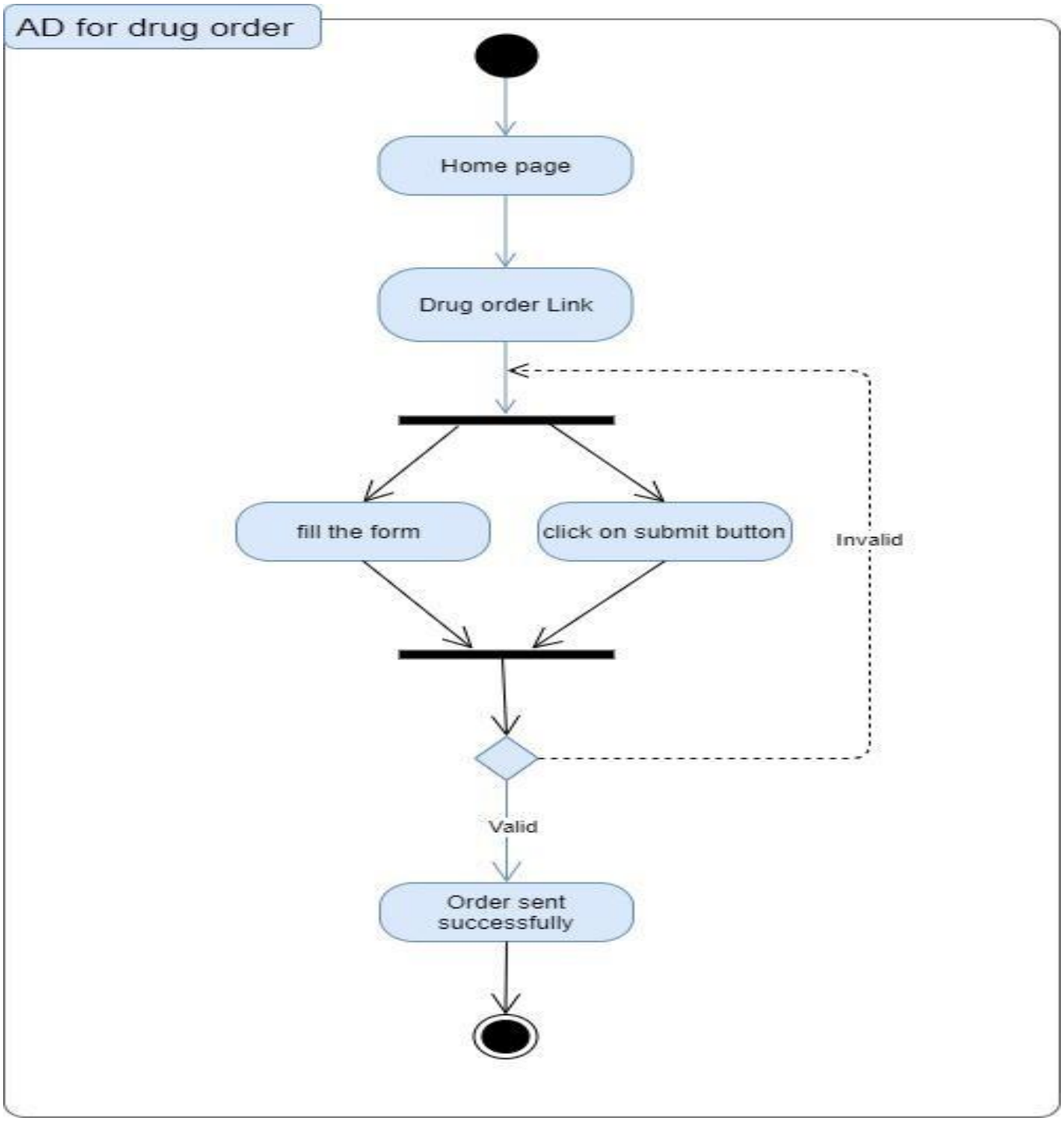


Figure 4. 15 Activity diagram for drug order

CHAPTER FIVE

5. SYSTEM DESIGN

5.1 Introduction

The Systems design of this project defines the elements of the WKU student management system like modules, architecture, components, and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of the stakeholder and also defines the coherent and well running system [4].

The system design activity starts after the system requirements analysis has been completed. The main goal of system design is to allocate the requirements of large system to hardware and software components.

5.2 Design Goals

The design goals are derived from the non-functional requirements of the system, which describes the WKU student management system that specify the qualities of the system that should be achieved and addressed during the design of the system.

The design goals of our system are:

User interface: **-** WKU student clinic management system should be easy to use and guides users how easily they use the system.

The GUI design of WKU student clinic management system shall be task-based without any redundant design. The functionalities of WKU student clinic management system also arranged in a module based on the similarities of their function that used for users to use the system easily.

Hardware Consideration: User shall be able to provide a machine either a mobile phone or a computer that is able to connect to the internet.

Security: **-** Unauthorized access to the system of WKU student clinic management system should be restricted. Hence, security will be implemented to prevent an unauthorized persons from accessing the WKU student clinic management system. There are groups with different

privilege levels and they are defined in the WKU student clinic management system. Users and the administrator belongs to a specific group will be given a password and user name to access the system at the privilege of his/her group. We will use the password encryption mechanism.

Performance: - the performance of the system is depending on the time required to perform a given task and memory required for that task so the following factors increase system performance. Avoiding redundancy of codes increases the time of operation and memory space of the system.

WKU student clinic management system should respond fast with high throughput, i.e. It should perform the task quickly because we use efficient algorithm to develop it.

Error Handling and Validation

The system is expected to handle errors encountered during run time. Errors could rise from users and the system. Errors that occurred from the wrong doing of users will be handled by appropriate exception handling mechanisms. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system.

Quality Issues: The system, it will be tested, verified and, validated and if the error occurs during the execution time it will give an exception. And, because the user will be involved in testing the quality of the system, it will satisfy the user's needs.

Requirement for reliability and it includes:-

- ✓ **User requirements**-the system should be work as the requirements of the users.
- ✓ **System portability**-the ability to work on a different personal computer and mobile phone.

Backup and Recovery: The process of backing up refer to making copies of data so that these additional copies may be used to restore the original after a data loss event. These additional copies are typically called "backups." Backups are useful primarily for two purposes. The first is to restore a state following a disaster (called disaster recovery). The second is to restore small numbers of files after they have been accidentally deleted or

corrupted. Data loss is also very common. So that our system uses back-up and Recovery in order to restore the data base after lost it.

Physical Environment

The server must be put in a place that has high security room. And the client must put in the local area it must connect with the server.

Documentation: The System has well-defined document which helps to easily maintain the system and we will also prepare a short and precise help file on how to use the system for the system users. It will have a helping page to guide the user of the system and to show the process how they will have to use it.

5.3 Current System Architecture

In this recent period Wolkite University uses manual system for WKU student clinic management system. Because of this, the university doesn't use any software architecture.

5.4 Proposed System Architecture

The proposed system is expected to replace the existing manual system by an automated system in many aspects. It is mainly based on the system analysis document. The architecture used for the WKU student clinic management system is a 3 tier client/server architecture where a client can use Internet browsers to access services provided by the system using the Internet. Three-tier architectures consist of three components distributed in 3 layers: client, application server and database.

- The client layer contains UI (User Interface) part. This layer takes input and gives Output to the user.
- The middle tier (web/application server) is between the database layer and the client layer. A web server is a program that runs on a network server (computer) to respond to HTTP requests. The most commonly used web servers are Internet Information Server (IIS) and Node JS. HTTP is used to transfer data across the Internet. It is the standard protocol for moving data across the internet.
- The database layer will be more secure and the client will not have direct access to the database.

Client layer: is user interface and topmost layer of the application. This the layer we see when we use this system. It is the interface to our system which takes information from the user. The main functionality of this layer is to communicate with the application layer. This layer passes the information which given in terms of keyboard action and mouse click to the application layer.

Application (Middle) layer: it is application layer which interacts with data layer and sends information retrieved from database toward to presentation layer.it act as the mediator between presentation layer and data layer. From above example once, the user clicks on login button application layer interact with the database and send information towards to presentation layer.

Database (Data) layer: which used to store data entered by the user? In general client of our system use browser to access the system using the internet. In this case when the user enters input and takes certain action application server process client request to interacting with the database server. Below figure is proposed system architecture

Figure 5. 1 Proposed System Architecture

5.4.1 Subsystem Description and Decomposition

A subsystem decomposition, the activity of identifying subsystems, their services, and their relationship with each other [2]. Subsystem decompositions will help reduce the complexity of the system. Subsystem decomposition for WKU student clinic management system is needed to make the system clear and understandable by dividing the whole system into

subsystems and components. In the subsystem decomposition, we try to show the relationship between each component. Subsystem decomposition of WKU student clinic management system refers to the process by which a complex system is broken down into parts that are easier to understand, program, and maintain. The major subsystems identified as following:-

Registration Subsystem

- Manage account
- Manage employee
- Manage patient
- Manage drug

Patient management Subsystem

- Send feedback
- Set appointment
- Generate refer
- Generate sick-leave

Pharmacist Subsystem

- Request drug
- Give prescribed drug
- Check expired date

Laboratory Subsystem

- Send lab result

Schedule Subsystem

- Manage schedule

Report Subsystem

- Generate report
- View report

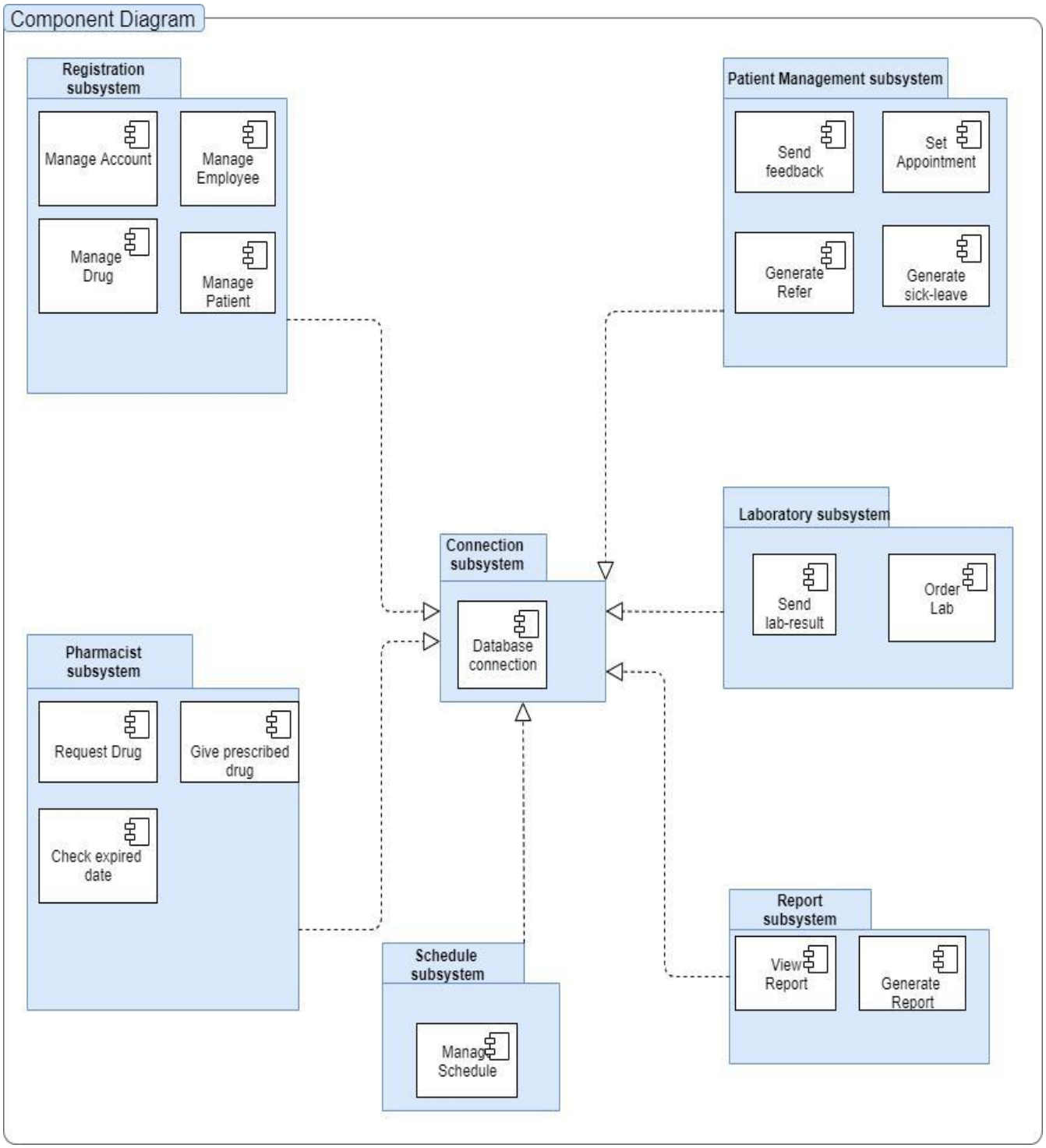


Figure 5. 2 Component diagram for WKU student clinic management system

5.4.2 Hardware/Software Mapping

The hardware/software map shows how the WKU student clinic management system and the hardware components are work together. Their interaction can be shown by using a deployment diagram which shows a static view of the run-time configuration of processing nodes and the components that run on those nodes.

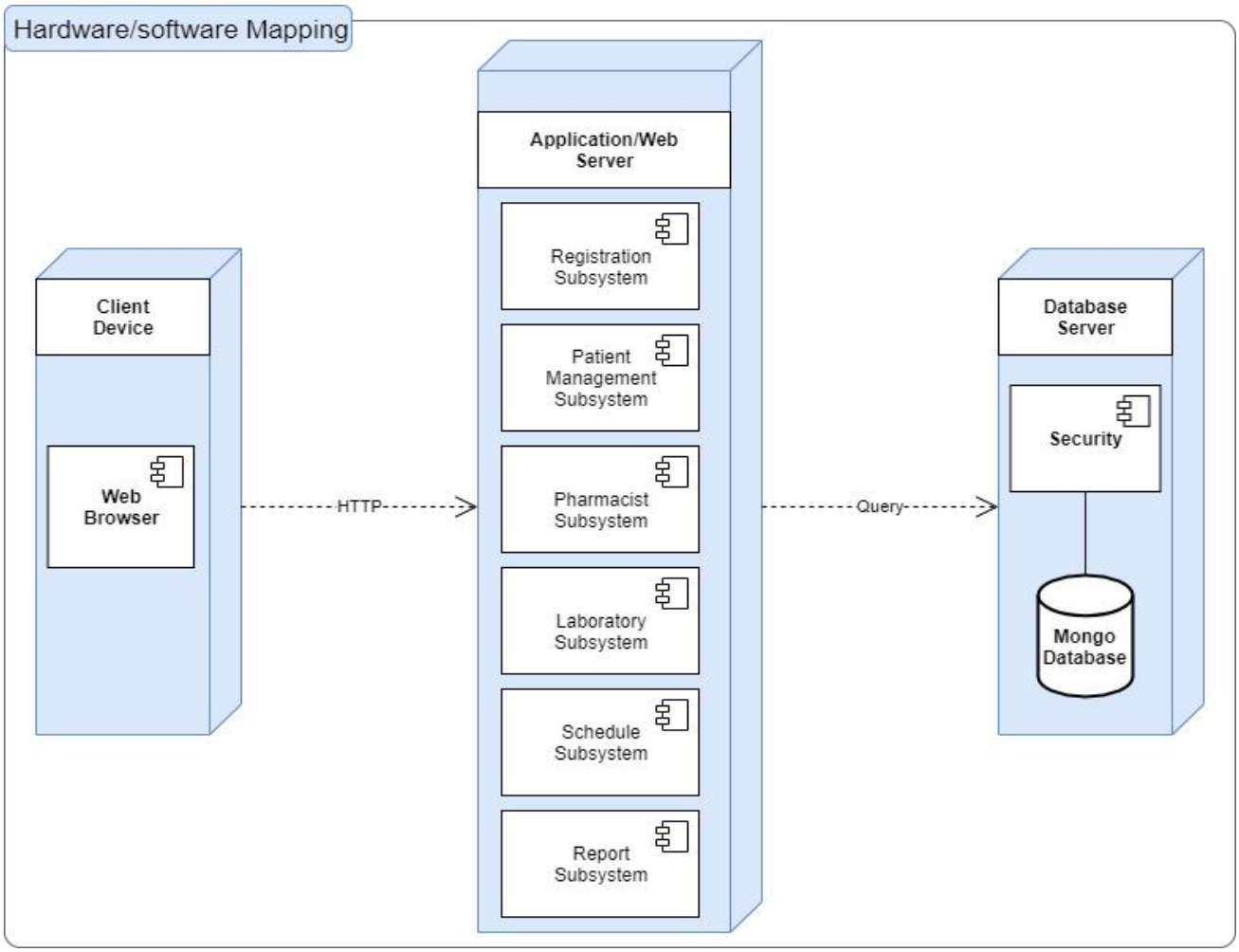


Figure 5. 3 Deployment diagram

5.4.3 Detailed Class Diagram

We use detailed class diagram to include attributes, methods, attribute data types, visibility of attributes and methods, inheritance, association, aggregation, composition, dependency, and municipality (cardinality and optimality).

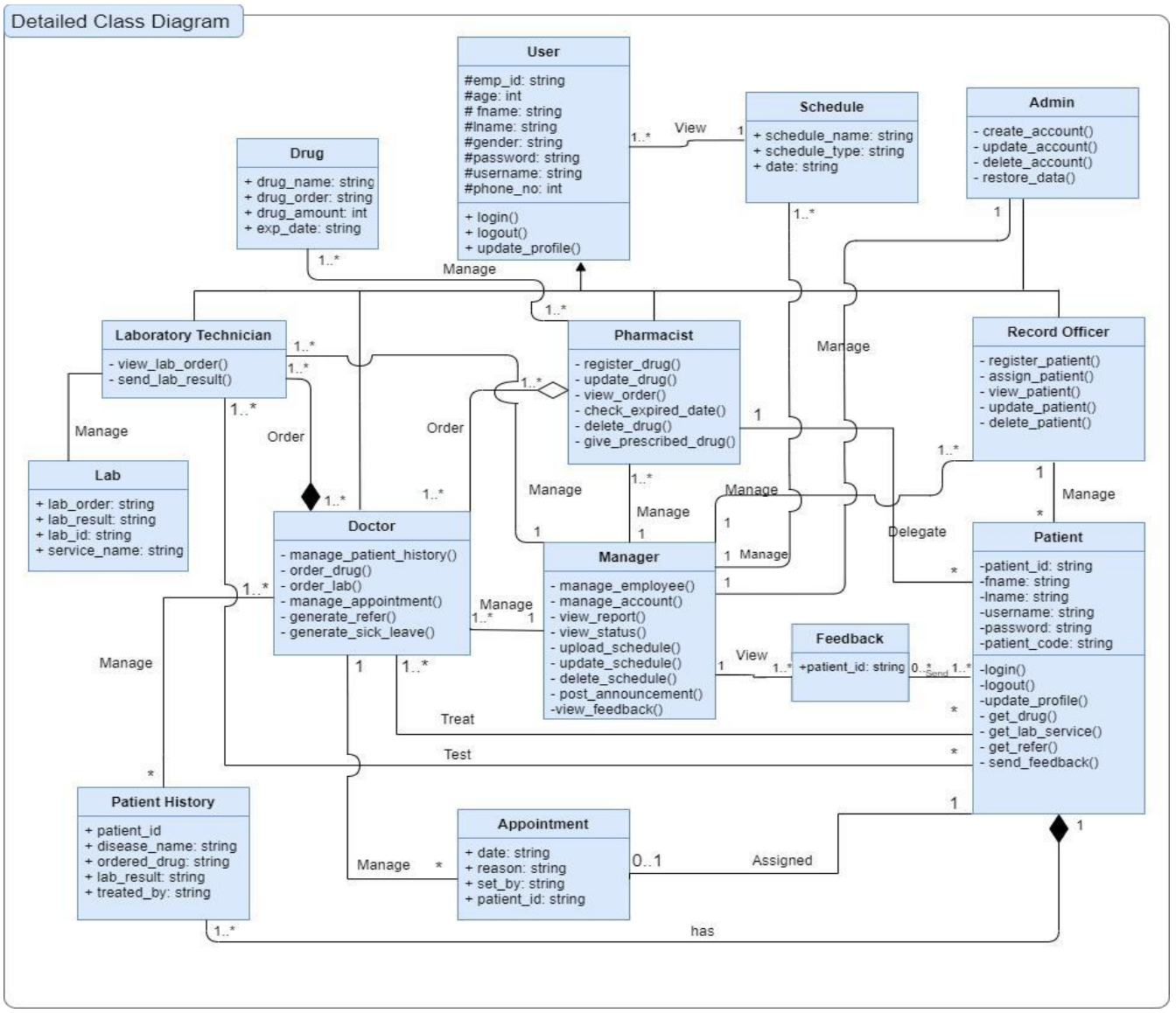


Figure 5. 4 Detailed class diagram

5.4.4 Persistent Data Management

In our system we use persistence modeling to communicate the design of the database, usually the data base to both the users and the developers.

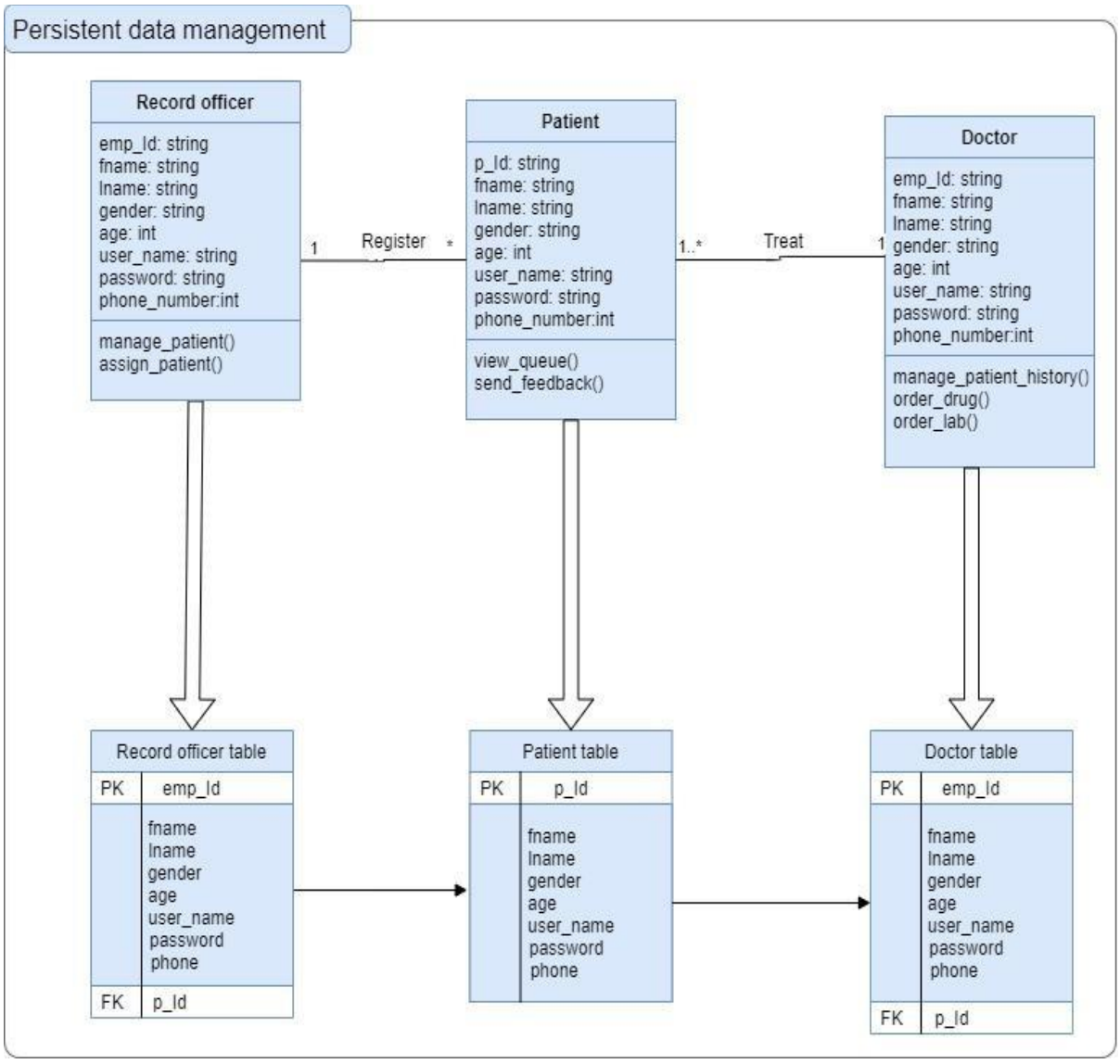


Figure 5. 5 Persistent Data Management

5.4.5 Access Control and Security

In our system, different actors have access to different information and data. Access control and security specify which users are granted access and certain privileges to systems, resources or information. This access control is verified by a username and password. The system authenticated the user. The proposed system follows a multi-user system. In a multi-user system, different actors have access to different functions and data.

Table 5. 1 Access Control and Security

Actor	Manage account	Manage patient	Set appointment	Generate sick-leave	View report	Set schedule
System Admin	CRUD	----	----	----	----	----
Manager	----	CRUD	----	----	-R--	CRUD
Doctor	----	-RD--	CRUD	CRUD	-R--	----
Pharmacist	----	-R---	----	----	-R--	----
Record officer	----	CRUD	----	----	-R--	----
Lab technician	----	-R---	----	----	-R--	----
Patient	-----	----	----	----	----	----

5.5 Packages

Package diagrams of our proposed system reflect the organization of packages and their elements and provide a visualization of their corresponding namespaces.

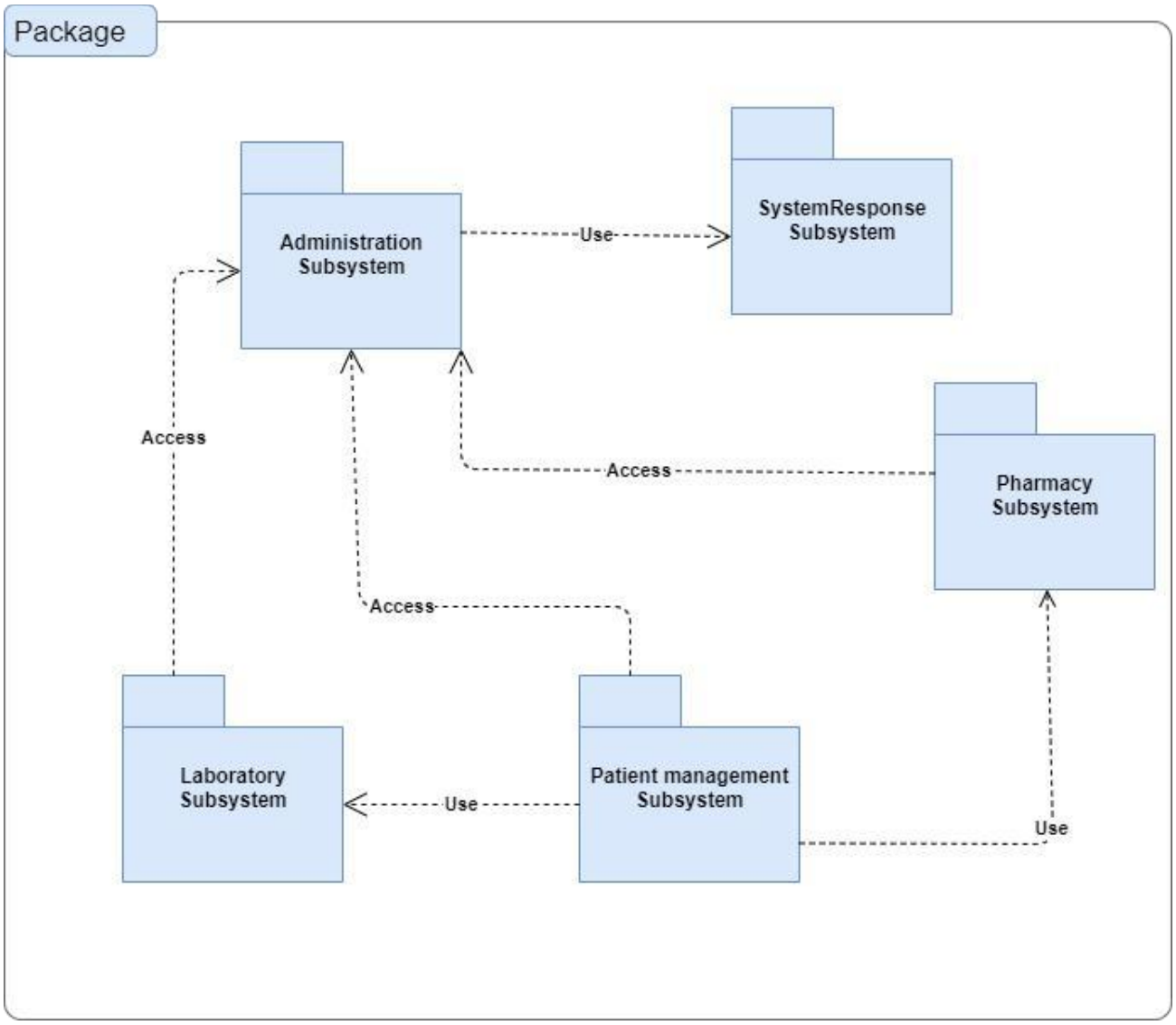


Figure 5. 6 Package diagram

5.6 Algorithm Design

In our system we have designed algorithms to show the flow of programs in the system. They are semantic driven rather than syntax driven.

Algorithm use for login

Input: Username and Password

Data Base output: login in status

Description: Enable the user login into the system

Begin

If user click login button

Login form

Fill form

If (valid)

Generate SQL select queries

Connect to database

Check query within database

If information in database

Display page

Else

Display error message

Else

Display error message (Ask user refill the form)

End if

End if

End if

Algorithm use for login

Begin

If admin or manager click create user account

Display

Then admin fill the form

If (valid)

Send information to database

```
                If information exist on database
                        Display the account already exist
                Else
                        Display message account successfully created
        Else
                Display error message
                End if
        End if
End if
```

5.7 User Interface Design

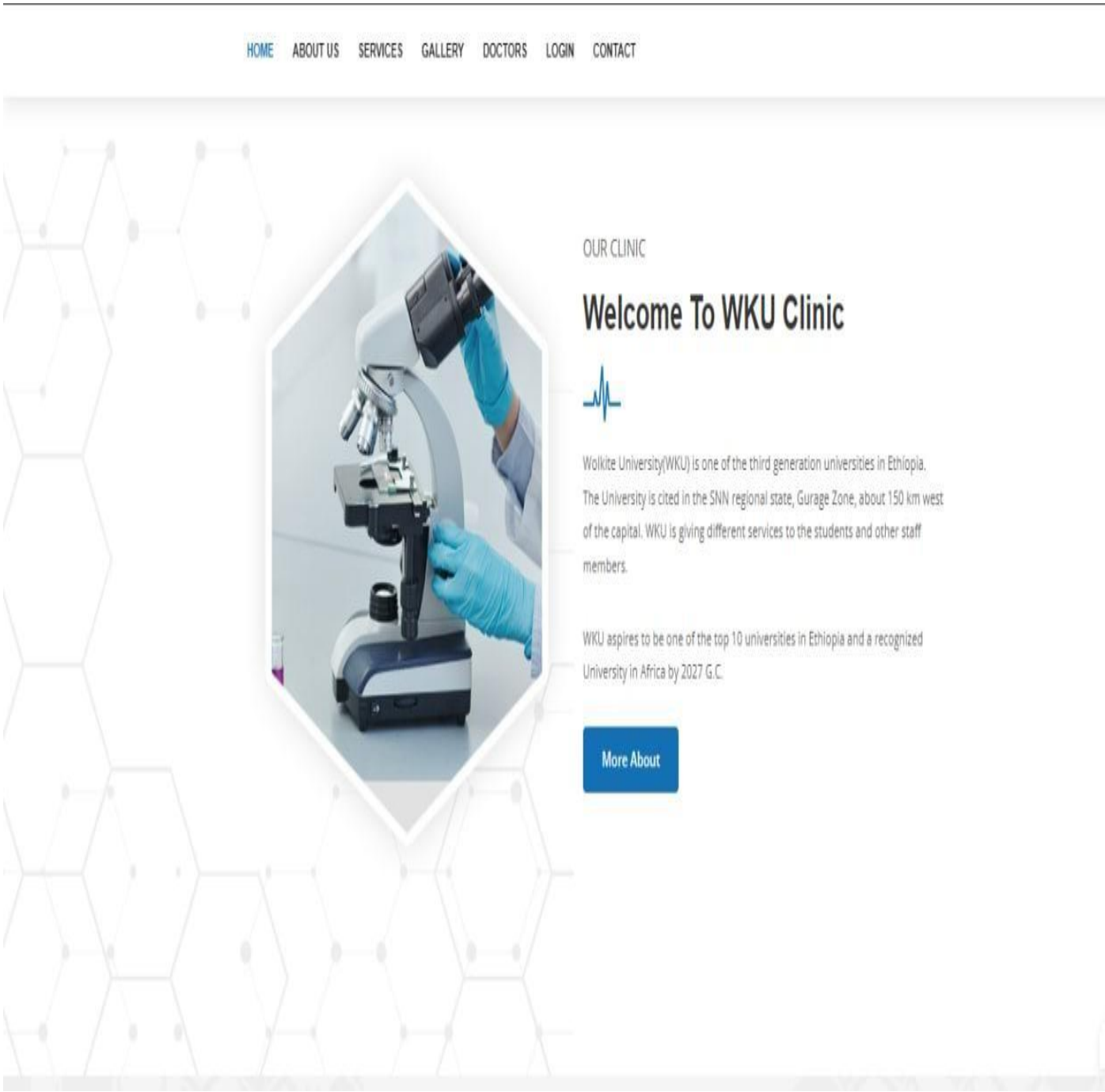


Figure 5. 7 Homepage for WKU student clinic

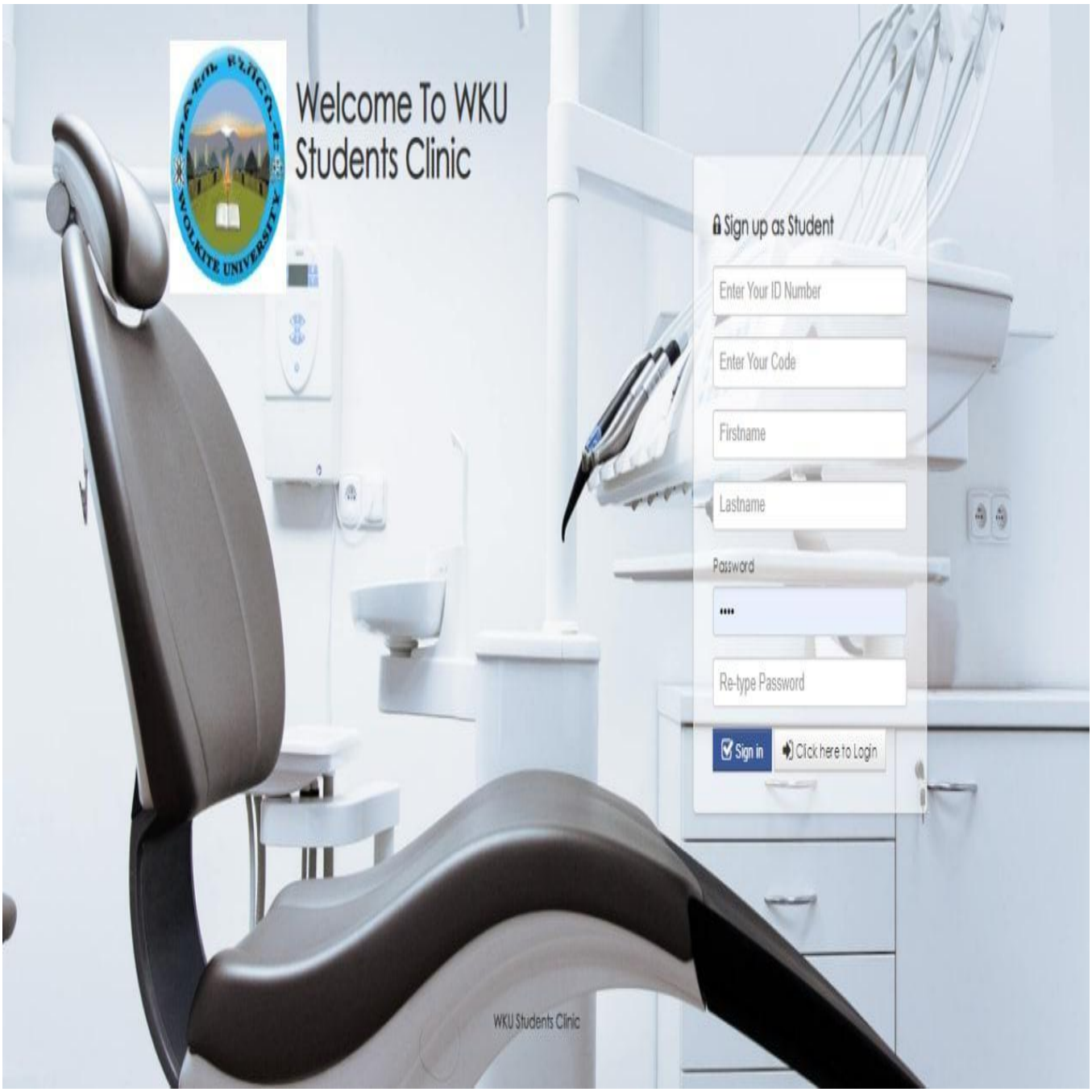


Figure 5. 8 Sign in or sign up page for WKU student clinic

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

Implementation is defined as a specified set of activities designed to put into practice an activity or program of known dimensions. It is the carrying out, execution or practice of a plan, a method, or any design, idea, model, specification, standard or policy for doing something. It is also a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programing and deployment. It is the most crucial phase in which it transform the design and analysis of the system into a tangible system by writing the code to the system to be developed and make it operational and applicable by testing and debugging the functionalities that are done. This makes the implementation stage more essential step to develop the required system.

6.1 Implementation of the Database

We use MongoDB to manage our system which assists to insert, update, delete and view data which is stored in the database.

We use MongoDB because it offers faster query processing but with an increased load and system requirements.

Database: `clinic`

Collection structure for users collection

```
const mongoose = require('mongoose')
const userSchema = new mongoose.Schema({


    username:{
       type:String,


   },
   firstname:{
       type:String,
       required: true,
```

```
    },
    lastname:{
        type:String,
        required: true,
    },
    email:{
        type:String,
        required: true,
    },
    password:{
        type:String,
        required: true,
    },
    address:{
        type:String,
        required: true,
    },
    contact:{
        type:String,
        required: true,
    },
    gender:{
        type:String,
        required: true,
    },
    image:{
        type:Object,
    }
},{
    timestamps: true
})
```

```
module.exports = mongoose.model('users', userSchema)
```

Collection structure for feedback collection

```
const mongoose = require('mongoose')
const feedbackSchema = new mongoose.Schema({

    firstname:{
        type:String,
        required: true,

    },
    lastname:{
        type:String,
        required: true,

    },
    email:{
        type:String,
        required: true,

    },
    feedback:{
        type:String,
        required: true,

    }

},{
    timestamps: true
})
module.exports = mongoose.model('feedback', feedbackSchema)
```

Inseerting data for collection `users`

```
const userController= {
  createUsers: async (req,res)=>{
     try{
      const {username, firstname, lastname, email, password,address,contact,gender,image,role}= req.body;

      if(!image) return res.status(400).json({msg: "No image Upload"})
      const user = await Users.findOne({email})
      if(user) return res.status(400).json({msg: "The email already exist"})
      if(password.length <6)
      return res.status(400).json({msg: "Password is at least 6 character"})

    const passwordHash = await bcrypt.hash(password,10)
    const newUser = new Users({
      user_id,username,firstname,lastname,address,contact,gender,role,image , email,password: passwordHash
    })

    await newUser.save()

 const accesstoken = createAccessToken({id: newUser._id})
 const refreshtoken = createRefreshToken({id: newUser._id})
    // create jsonwebtoken
    //res.json(newUser)
    // res.json({password,passwordHash })
res.cookie('refreshtoken', refreshtoken,{
   httpOnly: true,
   path: '/user/refresh_token'
})
   // res.json({accesstoken})
  //  res.json({refreshtoken})
     res.json({msg: "Register Success"})
```

```
    }
    catch(err){
      return res.status(500).json({msg: err.message})
    }
  }
```

Inseerting data for collection `feedback`

```
const mongoose = require('mongoose')
const feedbackSchema = new mongoose.Schema({

    firstname:{
        type:String,
        required: true,
    },
    lastname:{
        type:String,
        required: true,
    },
    email:{
        type:String,
        required: true,

    },
    feedback:{
        type:String,
        required: true,
    }
},{
    timestamps: true
})
module.exports = mongoose.model('feedback', feedbackSchema)
```

6.2 Implementation of Class

Model for user class

```
const mongoose = require('mongoose')
const userSchema = new mongoose.Schema({
    username:{
       type:String,

   },
   firstname:{
       type:String,
       required: true,
   },
   lastname:{
       type:String,
       required: true,

   },
   email:{
       type:String,
       required: true,

   },
   password:{
       type:String



   },
   address:{
       type:String,
       required: true,
```

```
    },
    contact:{
        type:String,
        required: true,

    },
    gender:{
        type:String,
        required: true,

    },
    image:{
        type:Object,

    }

},{
    timestamps: true
})
module.exports = mongoose.model('user', userSchema)
```

Router for user class

```
const router = require('express').Router()
const userCtrl = require('../controllers/userCtrl')

router.post('/adduser', userCtrl.createUsers)
 module.exports = router
```

Control for user class

```
  createUsers: async (req,res)=>{
     try{
      const {user_id, firstname, lastname, email,address,contact,gender, role}= req.body;
```

```
      const user = await Users.findOne({email})
     if(user) return res.status(400).json({msg: "The email already exist"})


   const newUser = new Users({
     user_id,firstname,lastname,address,contact,gender,role , email
   })


   await newUser.save()
    res.json({msg: "Register Success"})
     }
    catch(err){
      return res.status(500).json({msg: err.message})
    }
  },
```

Front end for user class

```
import { NavLink } from 'react-router-dom';
import React, { useState } from 'react';
import {  useHistory } from 'react-router'
import axios  from 'axios'
import Navbar from './Navbar';
import Sidebar from './Sidebar';
const User =()=> {

  const history = useHistory()

  const [user, setUser]=  useState({
    firstname : "",
    lastname : "",
    email : "",
```

```
    address : "",
    contact : "",
    gender : "",
    role: "",
  });

  const onChangeInput = e =>{
   const {name,value} = e.target;
   setUser({...user, [name]:value})
  }

  const registerSubmit  = async e=>{
   e.preventDefault()
   try{
     await axios.post('/user/adduser', {...user})
     localStorage.setItem('firstLogin',true)
     window.location.href ='/login';

   }
   catch(err){
     alert(err.response.data.msg)
   }
 }

    return (

      <div class="wrapper ">
      <div class="sidebar" data-color="purple" data-background-color="white" data-
image="../assets/img/sidebar-1.jpg">

        <div class="logo"><a href="" class="simple-text logo-normal">
```

```
    </a></div>
  <Sidebar></Sidebar>
</div>
  <div class="main-panel">
  <MNavbar></MNavbar>
<div class="content">
<div class="container-fluid">
  <div class="row">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header card-header-primary">
          <h4 class="card-title">User Registaration</h4>
          <p class="card-category"></p>
        </div>
        <div class="card-body">
        <form onSubmit={registerSubmit} method="POST">
           <div class="row">
            <div class="col-md-5">
            <div class="form-group">
              <label class="bmd-label-floating">User Id</label>
              <input type="text" class="form-control"
                name='user_id'
                value={user.user_id}

                onChange={onChangeInput}
              required />
            </div>
            <div class="form-group">
              <label class="bmd-label-floating">First Name</label>
              <input type="text" class="form-control"
                name='firstname'
```

```
          value={user.firstname}

          onChange={onChangeInput}
        required />
      </div>
    </div>
    <div class="col-md-3">
      <div class="form-group">
        <label class="bmd-label-floating">Last Name</label>
        <input type="text" class="form-control"
         name='lastname'
         value={user.lastname}

         onChange={onChangeInput}
        required/>
      </div>
    </div>
    <div class="col-md-4">
      <div class="form-group">
        <label class="bmd-label-floating">Email address</label>
        <input type="email"
          name='email'
          value={user.email}

          onChange={onChangeInput}

        class="form-control"/>
      </div>
    </div>
  </div>
  <div class="row">
```

```
  <div class="col-md-6">
    <div class="form-group">
      <label class="bmd-label-floating">Address</label>
      <input type="text" class="form-control"
       name='address'
       value={user.address}

       onChange={onChangeInput}

      />
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group">
      <label class="bmd-label-floating">Contact</label>
      <input type="text" class="form-control"
        name='contact'
        value={user.contact}

        onChange={onChangeInput}
      required/>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-md-12">
    <div class="form-group">
      <label class="bmd-label-floating">Role</label>
      <input type="text" class="form-control"
      name='role'
      value={user.role}
```

```
            onChange={onChangeInput}
            required />
          </div>
        </div>
      </div>
      <div class="row">

        <div class="col-md-4">
          <div class="form-group">
            <label class="bmd-label-floating">Gender</label>


            <input type="text" class="form-control"
            name='gender'
            value={user.gender}

            onChange={onChangeInput} />
          </div>
        </div>

      </div>

      <button type="submit" class="btn btn-primary pull-right">submit</button>
      <div class="clearfix"></div>
    </form>
  </div>
 </div>
</div>
<div class="col-md-4">
 <div class="card card-profile">
```

```
          </div>
        </div>
      </div>
    </div>
  </div>

  </div>

   );
}
export default User;
```

6.3 Configuration of the Application Server

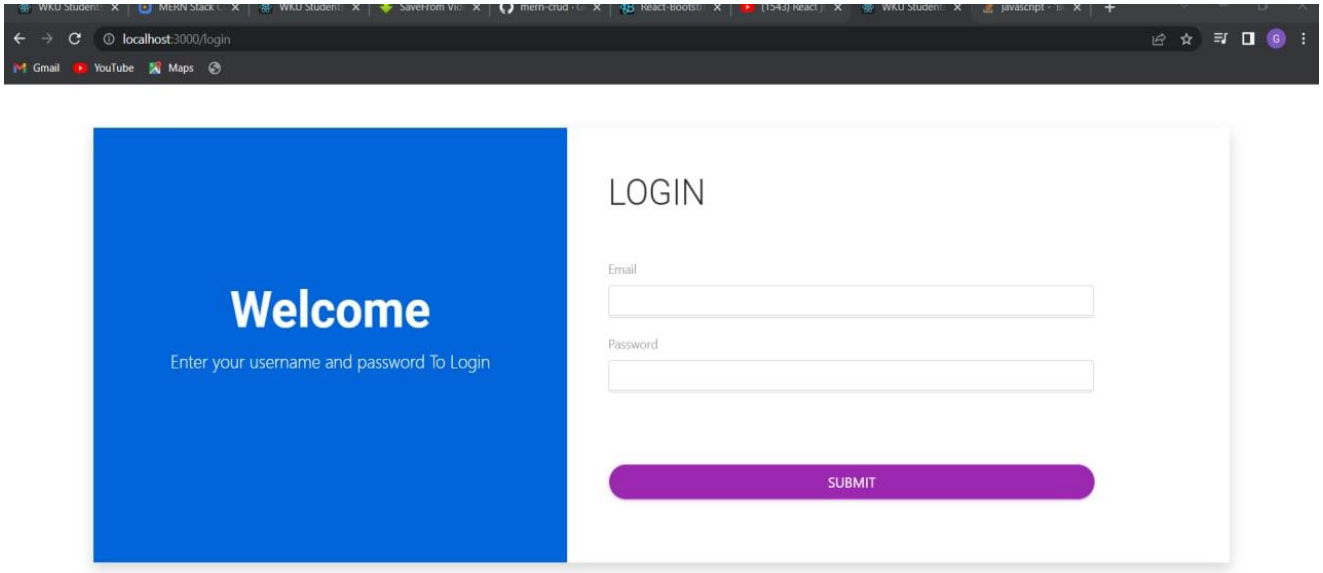
We use Mongo Community server because it offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data. Everything we needed is to set up a web server – server application (Mongo Community server), database (MongoDB), and scripting language (Node.js). Mongo works equally well on Linux, Mac, and Windows. Since it is suitable and have the function, we listed above we use Mongo application server.

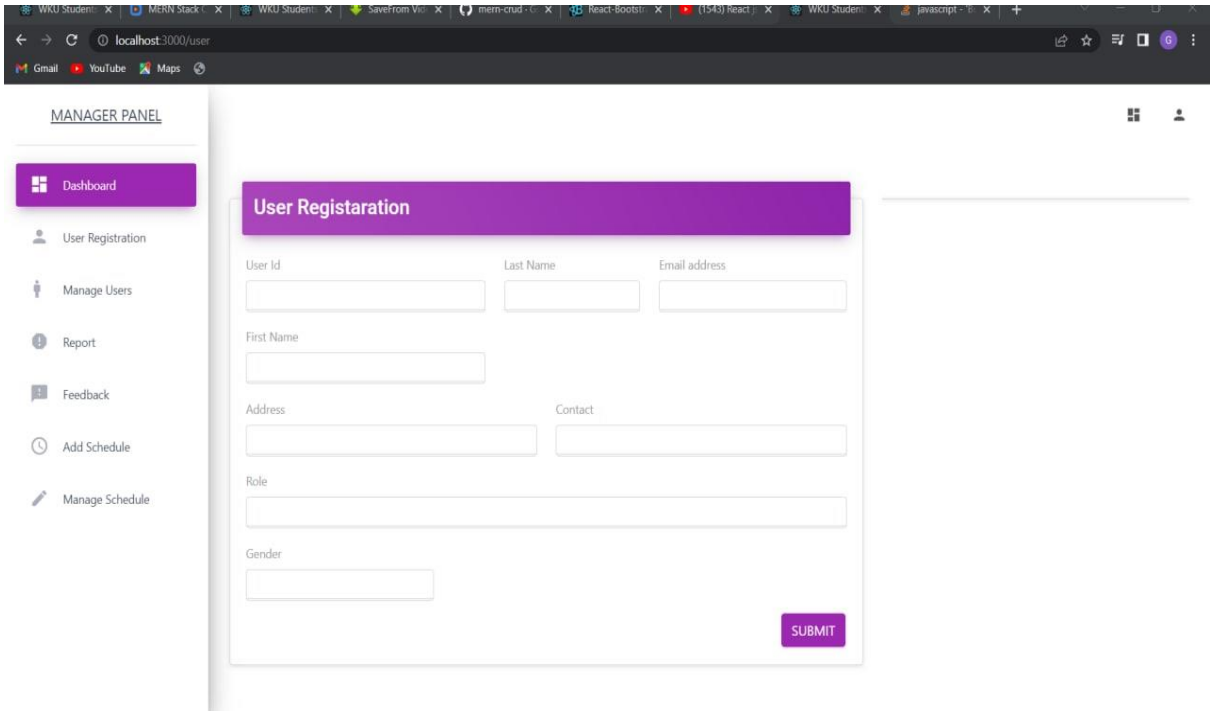
6.4 Configuration of Application Security

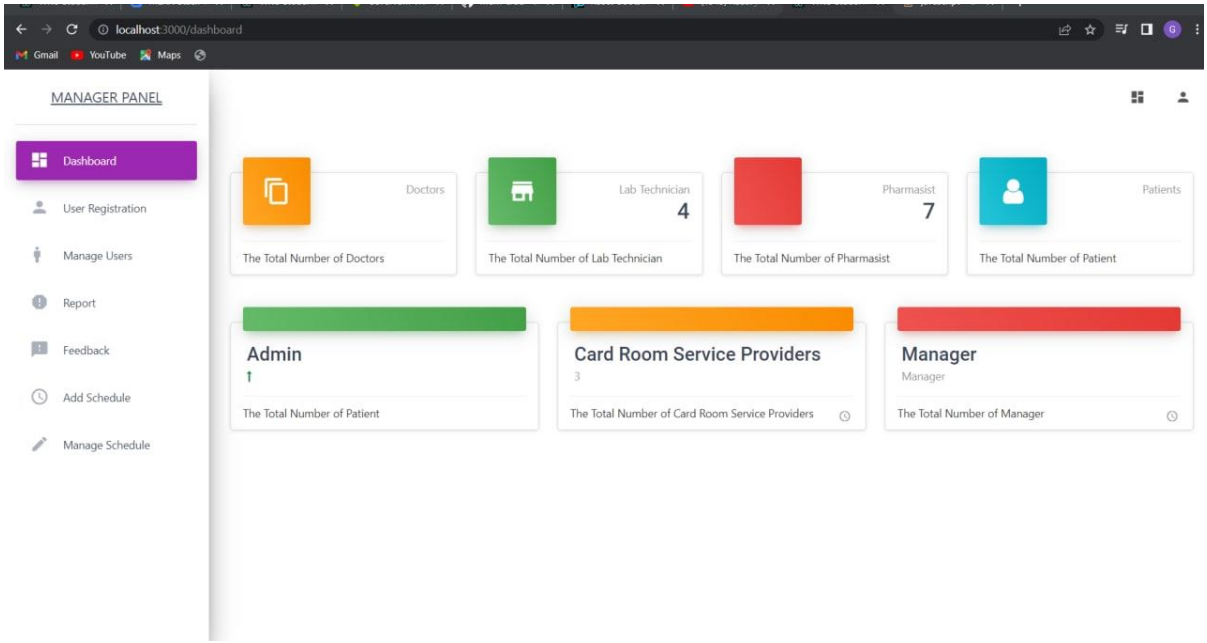
Our system validates all the input by returning error message and suggesting to try again when invalid input occur by using JavaScript in addition to this suggesting please fill out the filed when jump the fill space of the filed . We implement encryption for user password when the system Manager creates a user account or when the user changes their password the system encrypts the password by using bcrypt encryption algorithm.

6.5 Implementation of User Interface

We implement the user interface by considering the users of our system and easy for use interface of our system is developed using English language, because the users can understand simply. We implement it using React.js.


localhost:3000/login
Gmail
YouTube
Maps
Welcome
Enter your username and password To Login
LOGIN
Email
Password
SUBMIT


localhost:3000/user
Gmail
YouTube
Maps
MANAGER PANEL
Dashboard
User Registration
Manage Users
Report
Feedback
Add Schedule
Manage Schedule
User Registaration
User Id
Last Name
Email address
First Name
Address
Contact
Role
Gender
SUBMIT



6.6 Testing

6.6.1 Test case

Test case 1: Create account	
Test case objective: login into the system	
Test case description: user enters username and password, then click login button. Next to this Client side program contacts with server, server also contacts with the database and database checks for authentication and displays authorized page based on user role	
Requirements Verified: Yes	
Test Environment: Mongo Community server must be in running state, Database Should contain appropriate collection and link must be established between server and client program and Postman software is needed.	
Test Setup/Pre-Conditions: Mongo Community server should be in running state and user name and password fields should be filled correctly.	
Actions	Expected Results
The user should enter the correct user name	Displays authorized page based on user role.

or password to login.	
If user name and password are not filled correctly the system display to fill the user name and password again.	

Table 6. 1 Test Case

6.6.2 Testing Tools and Environment

Test Environment consists of elements that support test execution with software, hardware and network configured. Test environment configuration must mimic the production environment in order to uncover any environment/configuration related issues.

A typical Environmental Configuration for a web-based application is given below: For web server we use Mongo Community server, for Database MongoDB, we use widow operating system, and we use Chrome browser, and Postman software is needed.

6.6.3 Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this phase of testing we tried to eliminate faults in procedure and functions point of view by using black box and white box testing.

Tasks that we have performed under this phase

- Prepare unit test plans.
- Identify the unit test objectives.
- Prepare a test case that includes information such as set of test inputs, execution condition and expected output.
- Perform the tests according to our plan
- Analyses the test result

Tested unit	Input specification	Output specification	Environment	Result
Create account script	input is valid	Account is created	Create account form	success
	Input is invalid	Display error message	Create account form	success
Login script	Valid user name	Authorized	Login form	success

	and password	home page display		
	Invalid user name & password	Error message display	Login form	success

Table 6. 2 Unit Test

6.6.4 Integration testing

The process of bringing together for testing purposes all of the modules that a program comprises. Modules are typically integrated in a top-down, incremental fashion. If an error occurs, the process stops, the error is identified and corrected, and the test is redone. The process repeats until the entire program—all modules at all level is successfully integrated and tested with no errors. After the team tested using unit testing the next step is integrating testing. In integrated testing the team tested the system all modules that a program contains.

6.6.5 System Testing

In this level of testing process, we have examined how the whole subsystems of WKU students' clinic management work together to achieve the desired goal (user's requirements of the system). The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, it is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity of the System. But we will more focus only on function validation and performance.

Sample Tests

1. Evaluate the functionality of subsystem after combination of individual sub system weather it works correctly or not.
2. Check the coherence and coupling of each subsystem.
3. Check the overall functionality of the System that achieves the user's requirement.
4. Measure the system boundary which is beyond the goal or not.
5. Measure the weakness and the strength of the system using different metrics.
6. Check the interaction of each subsystem that performs the specified business process.
7. Verify the system completeness-based the requirement of end user.

6.6.6 Acceptance Testing

Our team members participate on the acceptance testing of our system. According to system requirements and other resources (documentation, source code, user manual) test cases are generated to determine (validation and verification) whether the system satisfies users need and expectation to maintain the reliability of our system and also meet the user's requirements.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1 Conclusion

Currently, WKU students' clinic day to day business operations such as patient registration, drug order, generating report, etc. are carried out using a traditional approach or manual based method. Due to this many problem are occurring such as information losing, data accessibility for privileged users, and also repetitive activities as recording same information, updating information, adding new patient or information is not that easy. By analyzing this problem, the group team proposed, analyze, design, and implement this computerized system for WKU students' clinic. The system that will provide more efficiency, and accuracy than the manual based system. The new web based computerized and management system perform many operations performed. It prevents the loss of paper documents, time, and effort. This system solves many problems such as lose of information form, and data duplication. But now there is a clear and base idea how the system can be developed and integrated, this project can be seen as an initiation for users who are doing project on the same field.

7.2 Recommendation

While doing this system the team has faced different types of challenges. But by the cooperation of all the group members and the adviser, the team is now able to reach to the final result. All the group members strongly fought these challenges and take the turn to the front. We would like to recommend the next fourth year student of computing and informatics to add some of the extra features add to this System which we could not do due to time limitation and unavoidable reasons and challenges including the following features.

- Video conference(To enable user communicate by video)

References

[1] D. F. A. Gorman, Introduction to Public Health Policy and Management (Graduate): 2041.

[2] "tutorialspoint,"[Online]. Available: https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm. [Accessed 2021/22].

[3] "Lucidchart," [Online]. Available: https://www.lucidchart.com/blog/types-of-UML-diagrams. [Accessed 2021/22].

[4] "TheEconomic Time," [Online]. Available: https://economictimes.indiatimes.com/definition/systems-design. [Accessed 2021/22].

[5] J. T. point, 25 01 2022. [Online]. Available: https://www.javatpoint.com/software-engineering-requirement-analysis.

Appendix

A. How record officer register a patient?
   How manager manages all employees and contact them?
   How much patients would get treatment per day?
   How patient information is stored?
   How the doctor order lab and drug?
   How can patient get sick leave when they miss assessment?
   How each activities are scheduled?
   How the expired date of drug is checked?