

**ELECTRONIC HEALTH RECORD SYSTEM**

**LAJIDE OLUWAGBOTEMI EMMANUEL**

**15010301014**

**BEING A PROJECT SUBMITTED IN THE DEPARTMENT OF COMPUTER SCIENCE  
AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE  
AWARD OF DEGREE OF BACHELOR OF SCIENCE  
MOUNTAIN TOP UNIVERSITY, IBAFO,  
OGUN STATE, NIGERIA**

**2019**

## CERTIFICATION

This project titled, **ELECTRONIC HEALTH RECORDS**, prepared and submitted by **LAJIDE OLUWAGBOTEMI EMMANUEL**, in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE (Computer Science)**, is hereby accepted.

\_\_\_\_\_  
DR. O.B. ALABA  
Supervisor

(Signature and Date)

\_\_\_\_\_  
DR. I.O. AKINYEMI  
Head of Department

(Signature and Date)

**Accepted as partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE (Computer Science)**

\_\_\_\_\_  
**PROF. A.I AKINWANDE**  
Dean, College of Basic and Applied Sciences

(Signature and Date)

## **DEDICATION**

This Project is dedicated to God Almighty

## **ACKNOWLEDGMENTS**

I owe my profound gratitude to God Almighty who gave the strength, wisdom and courage, divine help and provision to me from the beginning to the completion of this work. I express gratitude to my major supervisor, Dr. Alaba O.B., for his guidance and support in ensuring the successful completion of this research. God bless you richly sir. I sincerely appreciate the Dean, College of Basic and Applied Sciences, Dr. Akinwande A.I., for his fatherly advice, guidance and teachings. My heart-felt gratitude goes to the Head of Department, Computer Science and Mathematics – Dr. Akinyemi I.O., and all other members of staff of the department of Computer Science: Dr. Olaniyan O.O., Dr. Oyetunji M.O., Dr. (Mrs.) Kasali F.A., Mr. Falana O.J., Dr. Idowu P.A., Dr. Ojesanmi O.A., Dr. Adamu O.B., Dr. Okunoye O.B., Dr. (Mrs.) Oladeji F.A., Mr. Ebo I.O and others to mention but a few.

I acknowledge the constant support of my mentors who had contributed to my academic achievement. They include: Prof. Adedun E., Dr. Akinyugha A., Dr Akinyomi O.J., Dr. Young O., Mrs. Taiwo O., and Pastor Olumide A., I pray God would continue to increase their knowledge. I will forever be grateful to my parents Mr. and Mrs. Lajide E.O., who sacrificed wealth, time and other resources for the sake of my success; and my only sibling – Lajide Esther for her prayers and support. I also want to appreciate Mr. Bisola Ojo and family, Amasuomo D., and all Mountain Top University colleagues and friends for their prayers and support, and help in one way or the other. God bless them all greatly.

## **ABSTRACT**

Electronic Health Records (EHR) System evolved from the manual records used to store patient's data and document the activities of the patient in the hospital. In the previous system, data redundancy and loss of data files were the major problems faced along with the inability to easily book an appointment with the doctor. Therefore, the of this project is to automate the process of recording health records in medical institutions and to manage patient data properly.

In order to achieve its aim, a database for storing patients' information was created and an online interface for recording and display of information was designed. The developed web based electronic health record system was then implemented.

The EHR developed was able to store patient information, as well as handle online customer registration and booking and cancellation. It also allowed the administrator to modify the records of patients' and staff information that are within the institution.

The EHR is very useful and is a good upgrade from the previous system. It is recommended that the system is improved upon to increase the scope and productivity of the system.

Keywords: Health, Record, Clinic, Hospital, Electronic system.

## TABLE OF CONTENTS

<b>Contents</b>	<b>Page</b>
Certification	i
Dedication	ii
Acknowledgments	iii
Abstract	iv
Table of Contents	v
List of Figures	viii
<b>Chapter One: Introduction</b>	<b>1</b>
1.1 Background to the Study	1
1.2 Statement of the Problem	3
1.3 Aim and Objectives	3
1.3.1 Aim	3
1.3.2 Objectives	3
1.4 Scope of the Study	3
1.5 Methodology	4
1.6 Significance of the Study	4
1.7 Definition of terms	5
<b>Chapter 2: Literature Review</b>	<b>6</b>
2.0 Introduction	6
2.1 Conceptual Review	7

2.1.1	Electronic Health Record	7
2.1.2	Electronic Health	8
2.2	Theoretical Review	9
2.3	Review of related works	9
2.4	Summary of related works	10
2.5	Importance	11
<b>Chapter Three: Methodology</b>		<b>12</b>
3.0	Introduction	12
3.1	Flowchart of the Electronic Health Record system	13
3.2	Use Case diagram	15
3.3	Design details	22
3.4	Software Development Life Cycle	22
3.5	Method of Data Collection	23
3.5.1	Primary Source	24
3.5.2	Secondary Source	24
<b>Chapter Four: Result and Result Discussion</b>		<b>25</b>
4.0	Introduction and Documentation	25
4.1	Implementation	25
4.2	How the Electronic Health Record system works	25
4.2.1	Functional and Non-Functional Requirements	26
4.2.1.1	Functional Requirements	26

4.2.1.2 Non-Functional Requirements	26
4.3 System Requirements	26
<b>Chapter Five: Summary, Conclusion and Recommendation</b>	<b>35</b>
5.0 Summary	35
5.1 Contribution to Knowledge	35
5.2 Limitations	35
5.3 Recommendation for further Study	35
5.4 Conclusion	36
<b>References</b>	<b>37</b>
<b>Appendix</b>	<b>39</b>



## LIST OF FIGURES

<b>Figure</b>		<b>Page</b>
<b>3.1</b>	Flowchart of the EHR	<b>14</b>
<b>3.2</b>	Record Officer use case	<b>15</b>
<b>3.3</b>	Patients use case	<b>16</b>
<b>3.4</b>	Nurse use case	<b>17</b>
<b>3.5</b>	Doctor use case	<b>18</b>
<b>3.6</b>	Special Consultant case	<b>19</b>
<b>3.7</b>	Pharmacy use case	<b>20</b>
<b>3.8</b>	Admin use case	<b>21</b>
<b>3.9</b>	Iterative incremental software development model	<b>23</b>
<b>4.1</b>	Login Page	<b>28</b>
<b>4.2</b>	Admin Dashboard	<b>29</b>
<b>4.3</b>	Manage Departments	<b>30</b>
<b>4.4</b>	Admin Manage Users	<b>31</b>
<b>4.5</b>	Admin Manage Patients	<b>32</b>
<b>4.6</b>	Patient Registration	<b>33</b>
<b>4.7</b>	Doctor Manage Patient Cases	<b>34</b>

## **CHAPTER ONE**

### **INTRODUCTION**

#### **1.1 Background to the Study**

In the last fifty years, the world has experienced a huge explosion in technology, with innovation in every area making the human life more comfortable and our tasks easier, efficient and completed within a shorter period. Clinics and Hospitals are highly information-intensive organizations which spend huge amounts of money on information management and processing on inpatients and outpatients database system. It implies, of course, that techniques that enhance the collection, storage, distribution, and handling of data would be helpful instruments in hospital management. Patient base assessments of medical care are increasingly being used to measure the quality of health care (Design and Implementation of an Online Health care System Case Study of General Hospital, Minna).

Different qualitative and quantitative methods are available. Recently, patient satisfaction surveys or questionnaires are frequently used. The studies assess the patient's insight on care provided at an outpatient clinic. An outpatient is an individual who is not hospitalized for 24 hours or more but who visits a hospital, clinic or associated facility for diagnosis, enquires or treatment ("Hospital English"). An inpatient, on the other hand, is "admitted" to the hospital and stays overnight or for an indeterminate time, usually, several days or weeks (though some cases, such as coma patients, have been in the hospital for years). Treatment in this fashion is called inpatient care. The admission to the hospital or clinic involves the production of a hospital or clinic admission note (Inpatient, 2011). The leaving of the institution is officially termed discharge and involves a corresponding discharge note ("Hospital English").

In the old days, Doctors provide care in their offices or even at the patient's home. The trend reversed to where many treatments, tests, and surgeries were performed in a hospital setting. These days many treatment options exist, and your health care plan and your doctor can recommend what makes sense for you, along with a spectrum from less intensive to more intensive care. Outpatient care is any health care service provided to a patient who is not admitted to the facility. Outpatient care may be provided in a doctor's office, clinic, the patient's home or hospital outpatient department. In an urgent care facility, treatment is provided for conditions requiring prompt medical attention but that are not emergencies. Example of urgent care needs include ear infections, sprains high fevers, vomiting, and urinary tract infections. Inpatient care is care given

to a patient admitted to a hospital, extended care facility, nursing home or another facility. Long-term care is the range of services typically provided at skilled nursing, intermediate care, personal care or elder care facility. Inpatient hospitalization makes sense for major diagnostic, surgical or therapeutic services, where the patient's condition or response to medication must be closely monitored. In the case of mental health treatment, a hospital stay may make sense if the person is suicidal or self-destructive or poses a threat to others ([www.oupatientths.com](http://www.oupatientths.com)).

Efforts are continuously being made in designing and developing economically and reliable database system to satisfy an electronic record system for hospitals. This project is about the design and development of a hospital database system for medical management of the Mountain Top University and will show how records and data can be handled and stored within the system.

The system will store patients' records, drug inventory and dispensation as well as other relevant clinical information within the community. The system will be used by the staff doctors, nurses, consultants, students, and other relevant authorities as the demand may be within the clinic or the community as a whole. It will process data speedily and accurately, provide information when and where required.

The electronic health record database system will be used to store data, produce reports and handle management enquires. In clinics that consist of doctors, nurses, patient, and other personnel staff, an information system is required to keep track of the day to day transaction of the clinic and to improve the communication skills. The medical sector is a critical area and it is always growing as population increases since everyone needs access to medical services. As a result, there is a challenge in handling patient data and utilizing the data for planning and budgeting purposes. In addition, since facilities do not grow by area of coverage, they only grow by patient volume there is thus a great need for an efficient way of handling the turnout in a timely and quality adhered manner (Rao et al, 2008).

Therefore, it is hoped that the design and development of an Electronic Health Record System would provide flexible and reliable management of the University Clinic. This program would be carefully designed to work in a stand-alone system or in a multi-user environment.

## **1.2 Statement of the Problem**

It is observed that Mountain Top University has most of its record on staff, students, patients, drug, inventory, and dispensary handled manually. The problems associated with the manual method include inaccurate data, poor quality of patient care and administrative inefficiency. The task involved in this design is as follows:

1. A careful study of the manual operation of Mountain Top University Medical Centre.
2. Collection of information on general problem confronting the Medical Centre.
3. Study the identified problems, design and develop a hospital management system that will help in solving these problems.
4. Develop computer software that will help in the management in the Medical Centre.

## **1.3 Aim and Objectives**

### **1.3.1 Aim**

The aim of this project is to design and develop an electronic health record database for the Mountain Top University clinic.

### **1.3.2 Objectives**

The specific objectives of this project is to:

1. Improve efficiency and communication skill in the clinic.
2. Keep proper patient's records, drug inventory, dispensing and efficiency of drug dispensed.
3. To bring about a computer pollution-free environment that will ensure secured public health of the staff, students, and the community.
4. To acquire practical experience in the design and development of an electronic health record system and other related software.

## **1.4 Scope of the Study**

The scope of this project is mainly on the patient's record, the drug prescription, available drugs, the drug dispensed, and the efficiency of the drug dispensed. The application of query in getting the desired reports on drug inventory in the pharmaceutical unit.

## **1.5 Methodology**

The project is carried out, following the appropriate stages involved in design and development of an EHR software. The stages are briefly discussed below:

1. **Problem identification:** The activities associated with the clinic are first identified. This is done by a proper assessment of the system that is presently in place.
2. **Data collection:** The data and information required for the clinic database are acquired by conversing with staff, students, and other medical personnel.
3. **System modeling:** Using relevant information obtained from relevant sources, the clinic database system will be designed.
4. **System development and implementation:** mental designs are developed and implemented as a computer program. After the development and implementation, the system is tested to ensure that, it fulfills the objective of the project.
5. **System installation:** The developed clinic system software will be installed on the computer system and made ready for use.

## **1.6 Significance of the Study**

The large-scale sharing of medical records via network connections has the potential to bring us numerous benefits. This study would be found useful for the Mountain Top University clinic because it will help to address the problem of security; privacy and confidentiality of patient health records. It also helps to checkmate the delay, error, inconsistencies in medical records and timely access to historical records all of which had a significant impact on the quality of the clinic. It suppresses the traditional and manual method of data collection and record-keeping.

The implementation of the Electronic Health Record System is vital to the provision of quality care, as well as adequate management of scarce resources and productivity. With this in mind, the single most significant benefit of implementing an Electronic Health Record System is to institute a mechanism for reducing medical expenses and constraining the growth of these services.

## **1.7 Definition of terms**

**Admin:** This is an individual whose job is to carry out the administrative work of a business or an organization. This person is primarily responsible for the collection of data, putting it a digital (computerized) system, putting patients data therein, and ensuring that the patient data is secure at all times.

**Certification** - This relates to a national effort to "certify" various requirements for EHR software. The Certification Committee for Health Information Technology (CCHIT) is tasked with determining what basic "must-have" features EHR systems contain in order to be "certified."

**Electronic Health Record (EHR)** - This term refers to computer software that physicians use to track all aspects of patient care. Typically this broader term also encompasses the practice management functions of billing, scheduling, etc.

**Electronic Medical Record (EMR)** - This is an older term that is still widely used. It has typically come to mean the actual clinical functions of the software such as drug interaction checking, allergy checking, encounter documentation, and more.

**Integrated EHR** - This refers to an EHR that is integrated with practice management software. Typical choices include purchasing a fully integrated product which performs all the functions of practice management software, or a stand-alone EHR which is compatible with an existing practice management system.

**Structured and unstructured data entry** - There are several ways of entering data into your EHR as you practice. These include dictating straight into the software (voice recognition), templates, and writing (handwriting recognition).

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.0 Introduction**

This chapter provides a review of the literature on the factors influencing the successful implementation of HMS. The presentation of this chapter begins with the conceptual review, theoretical review, review of related literature and the summary of the literature reviewed.

The use of technology in hospital management and health care services is on the rise as the need for the focus on patient care is evolving. Electronic Health Record system is a computerized medical information system that collects, stores, and displays patient information. It deals with the drug, equipment, human resources and other relevant information in the institution. It is a means to create legible and organized patient data and to access clinical information about individual patients at any time with relative ease (Albert Boonstra & Manda Broekhuis, 2010). Electronic Health Records systems are intended to supplement existing (often paper-based) medical records which are already well known to practitioners. Patient records have been stored in paper form for centuries and, over this period; they have consumed increasing space and notably delayed access to effective medical care. In contrast, the electronic health record system would store individual patient clinical information electronically and enable instant availability of this information to all providers in the healthcare chain and so assist in providing coherent and consistent care (Albert Boonstra & Manda Broekhuis, 2010).

The advantages of EHR systems can be summarized according to (Yamamoto 2006) as "optimizing the documentation of patient encounters, improving communication of information to physicians, improving access to patient medical information, reduction of errors, optimizing billing and improving reimbursement for services, forming a data repository for research and quality improvement, and reduction of paper".

The medical record is an important primary tool in the field of medicine because Huffman (2001) stressed that the whole "idea" behind it is to provide better care of the patient through careful recording of every detail having to do with his/her case. The medical record is the, who, what, why, where, when, and how of the patient care during hospitalization. He further added that

the medical record is the only measurement of work being done by the medical staff, the only record of the progress of the patient, and it is the source of information for many purposes. Adeleke (2014) further noted that medical record is valuable to the patient because physicians see many patients a day and it is impossible for them to remember the details of each separate case at the same time. The patient of today may become ill in the future, be admitted to the same or another hospital with the same or another illness, and they may be examined by the same or a different physician.

Late Patrick Chukwunyere Eleoba (2002) in his book maintained that where computers were first introduced into hospitals some years ago that the primary use was accounting, record keeping purpose and administrative purpose, but the medical profession has fully utilized the computer to improve health records in recent times.

Electronic Health Records (EHR) are seen as a way to make simpler the management of patient information, increase productivity and reduce costs associated with medical information management. The implementation of the EHR system in healthcare facilities has not only provided potential benefits to improved quality of care, but there are significant factors to consider associated with human factors, ergonomics, workflows, and environmental conditions. These factors should be considered to improve the quality and cost of patient care, as well as employee and patient safety.

The computer has actually taken bold steps in recent times and this project work talks more on how computer and computer programs assist in the keeping of medical records but we still look forward when they will completely take over the conventional system.

## **2.1 Conceptual Review**

### **2.1.1 Electronic Health Record**

An electronic health record (EHR), or electronic medical record (EMR), is the systematized collection of patient and population electronically-stored health information in a digital format. These records can be shared across different health care settings. Records are shared through network-connected, enterprise-wide information systems or other information networks and exchanges. EHRs may include a range of data, including demographics, medical history, medication and allergies, immunization status, laboratory test results, radiology images, vital signs, personal statistics like age and weight, and billing information.



### **2.1.2 Electronic Health**

An electronic (digital) collection of medical information about a person that is stored on a computer. An electronic health record includes information about a patient's health history, such as diagnoses, medicines, tests, allergies, immunizations, and treatment plans. Electronic health records can be seen by all healthcare providers who are taking care of a patient and can be used by them to help make recommendations about the patient's care. Also called EHR and electronic medical record. A study was conducted in 2004 by Healthcare Informatics in collaboration with American Health Information Management Association (AHIMA) to measure the level of readiness of health information management (HIM) professionals and the extent of (Electronic Health Record) EHR implementation in their organization. The findings showed the industry is continuing to see more movement toward EHR. For example, when organizations were asked to describe their progress toward an EHR, 17 percent of respondents indicated they were extensively implemented; 26 percent indicated they were partially implemented; 27 percent said they were selecting, planning, or minimally implemented, and 21 percent indicated they were considering implementation and gathering information about it (Minal Thakkar and Diane .Davic August 14, 2006).

Electronic Health Record System handles all patient data, doctor and medical staff information, and hospital billing. An EHR system helps manage all sections of the hospital: reception, labs, pharmacy amongst others (Anastasia, 2019). According to Rupp (2018), an electronic health record system has several important advantages that help the day-to-day operations of a hospital or nursing home to function smoothly, such as:

- i) It costs less and makes work easier
- ii) It helps in the medical billing operations
- iii) It enables accurate capture and storage of information
- iv) It allows for better competitiveness and response

## **2.2 Theoretical Review**

Kapoor and Singh (2014) discussed the different issues faced during the implementation of an online hospital management system. The study revealed that these issues are not linked with technical problems but are linked with funds, behavior, and attitude of doctors, lack of awareness amongst others.

Srivastava (2013) assessed and evaluated the role of internet hospital management in providing health care facilities. The research findings found that the hospital management scheme may be more useful to the elderly, particularly those with disabilities and pregnant females who are unable to embark on a long-distance trip for frequent hospital check-ups. A fresh platform for dissemination of health information on precautions to be done against life-threatening illnesses such as HIV / AIDS, cancer, among others. The implementation of personal pages has appeared. The focus regions for the research were technological growth, safety and safety problems, advantages and difficulties engaged in offering e-health facilities, satellite use, and other technological facilities.

## **2.3 Review of Related Works**

The American Hospital Association (AHA) conducted a survey of all community hospitals in 2005 to measure the extent of information technology (IT) used among hospitals and better understand the barriers to further adoption. CEOs from 900 community hospitals (19.2 percent) participated in the study. The study found that 92 percent of the respondents were actively considering, testing, or using IT for clinical purposes. The remaining 8 percent that were not considering IT were primarily small, rural, nonteaching, and no system hospitals. The study reported that more than 50 percent of the respondents fully implemented the EHR functions results review—lab, order entry—lab, order entry—radiology, access to patient demographics, and results in review—radiology report. This study also reported that 50 percent of the rural hospitals specified they were just “getting started” on IT system implementation, whereas 48 percent of the urban hospitals indicated “moderate” or “high” levels of implementation of IT systems. The cost was the number one barrier to the adoption of EHR systems; 59 percent of the hospitals found that initial cost was a significant barrier; 58 percent found acceptance by clinical staff as somewhat of a barrier. Among the smaller hospitals with bed size less than 300, more than 50 percent saw cost as a significant barrier.

Ahmed and Usman (2017) focused their research on solving the amount of time patients waste before they book their appointments with the doctor. They decided to come up with a system that would manage patients' appointments with the doctor. The system was designed based on the 3-tier architectural pattern, consisting of the presentation tier (user interface or the client application), business logic tier (application server), database/storage tier (for storing patients' details). The software was developed using the waterfall model and Visual basic.NET as the development tool. With the development of the system, patients can easily book and manage their own appointment. They will also be reminded of their appointments via SMS/email which they will receive promptly before their date of appointment. The system also enabled patients to track and monitor their appointment record. However, due to technical constraints, the display of bio-data such as X-rays and laboratory results were not included in the system.

Bayanno Hospital Management System is one of the several products created by the software developers at Codecanyon. It is complete software for hospital, clinic and medical institutes. It supports a laptop, smartphone, desktop and tablet devices. It integrates and facilitates seven types of user area of a hospital, namely Administrator, Patient, Physician, Nurse, Pharmacist, Lab, and Accountant. The software also includes a security feature claimed to be invulnerable to threats such as SQL-injection, XSS attacks and CSRF (Codecanyon, n.d.). Improvements made to Bayanno Hospital Management System by the proposed system include reduced data redundancy possibilities and cost-efficiency. Due to limited time, the design of the system was not responsive on mobile devices.

#### **2.4 Summary of Related works**

Okwor Emeka Daniel (2012) worked on the Design and Implementation of Electronic Patient management system. His main objective and goal were to eradicate the improper documentation of payment records and the difficulty in retrieving patient payment records. He made use of the Rapid Application Development model and was able to build reliable storage for keeping payment records and reduce the amount of time spent during payment for services.

Ahmed and Usman (2017) analyzed and built an Outpatient Management system. Using the waterfall model, they ensured the improvement of quality of service offered by hospitals through outpatients scheduling. Patients could book and manage their own appointments with ease. They could also be reminded of their appointment dates via SMS/Email.

Ilo, Igbajar, and Acholonu (2015) implemented a web-based hospital management system for MOUAC clinic. Their aim was to increase the efficiency and interactivity in any area of specialization in the hospital. The spiral software development model was adopted and used in this system. The system was developed using C# programming language and Visual Studio 2013 as its development environment with MySQL as its database. The system is able to perform the following functions:

- i) The system can authenticate the users of the system.
- ii) Only the administrator can make changes to the database
- iii) The system was able to generate test reports, provide prescription details including various tests, and medicines prescribed to patients by doctors

Due to inadequate facilities used, the system was unable to provide an online payment functionality.

## **2.5 Importance**

The literature review has helped me to understand the nature of what people have done and encountered in the past and how they arrived at their results. It will be of great help to my work and will also make me do a project that is of a very good standard and that is backed with a new innovation.

## **CHAPTER THREE**

### **METHODOLOGY**

#### **3.0 Introduction**

This chapter provides a methodology of the EHR to be built. The presentation of this chapter begins with the flowchart, use case diagrams, design details, software development life cycle and the methods of data collection.

Research methodology is the notion within which study is scheduled (Okwor 2012). The development of a secure internet hospital management scheme is designed to enhance the performance of hospital facilities provided. With the improved changes of information and technology, the process by which automated management systems are made is changing dramatically. Using ICT in health institutions has led to the improvement in quality of hospital services. Coupled with the rapid changes of ICT evolution in the society, the health institution should be along with the changes of modern society too. Organizations have benefitted from the use of Information Technology (I.T).

The management of a clinic consists of a lot of members in various departments which include the record keepers/ receptionists, nurses, pharmacists, lab technicians, drivers, doctors, and special consultants. Each of these individuals in the positions communicates with one and other and shares some of its information requirement with one another. Each of these data is shared while some will remain private as situation demands. In a clinic, the patient's admission, treatment, and discharge will be used and updated by a variety of medical personnel who also both collaborate at various stages throughout patients stay in the clinic, their information requirement is slightly different, some vital information such as detailed medical records, drug interaction and contradiction should be available to appropriate medical staff or for clinical purposes. All these can be carried out by an electronic health record system that is designed for the clinic.

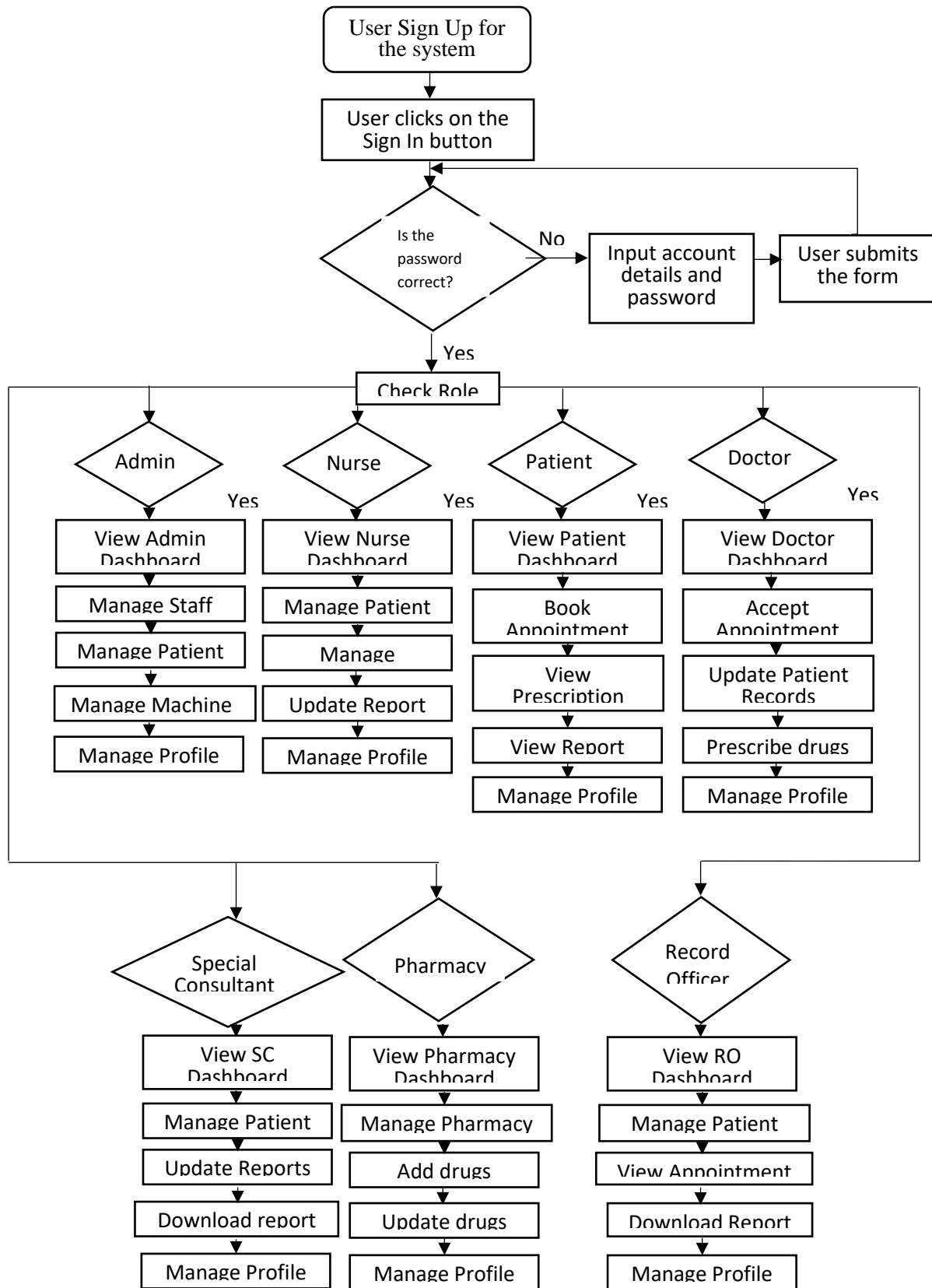
An electronic health record system is, therefore, a general-purpose software system that controls shared access to the database and provides a method that helps in screening the integrity of stored data. The requirement of this database system includes:

- Provide the means to define and design a database.
- Provide efficient and reliable shared access to the stored data.

- Support multiple views of concerned data resources similar to the needs of different functional areas and application.

### **3.1 Flowchart of the Electronic Health Record System**

A flowchart is a diagram of the sequence of movements or actions of people or things involved in a complex system or activity. It is a graphical representation of a computer program in relation to its sequence of functions (as distinct from the data it processes).

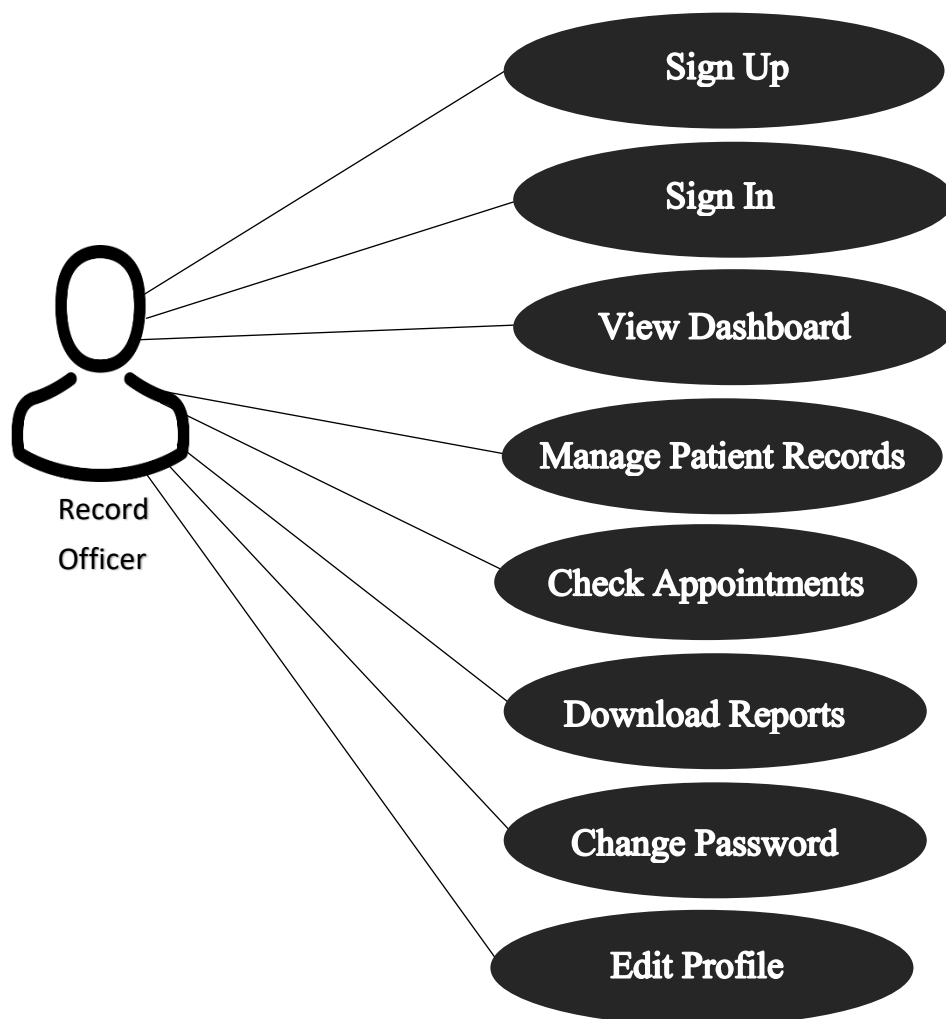


**Figure 3.1:**Flowchart of the EHR

### 3.2 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved (Wikipedia, 2019).<sup>i</sup>The following are the different use case diagrams for the system:

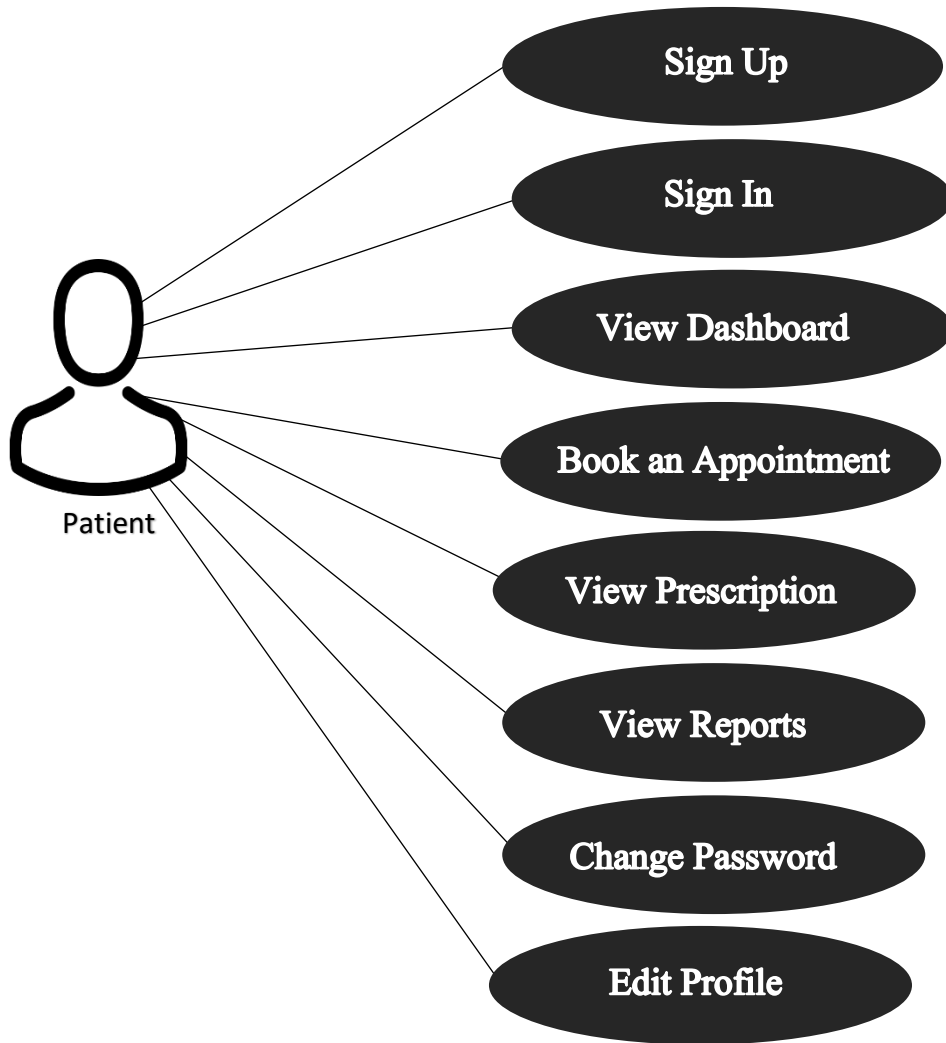
- **Record Officer Use Case:** The Record Officer shall be able to sign up for the system, then login to the system, interact with the dashboard and view what the system has to offer. Record Officers shall also be able to view patients records in the hospital, check if the patient has an appointment with the doctor and update profile.



**Figure 3.2:** Record officer use case

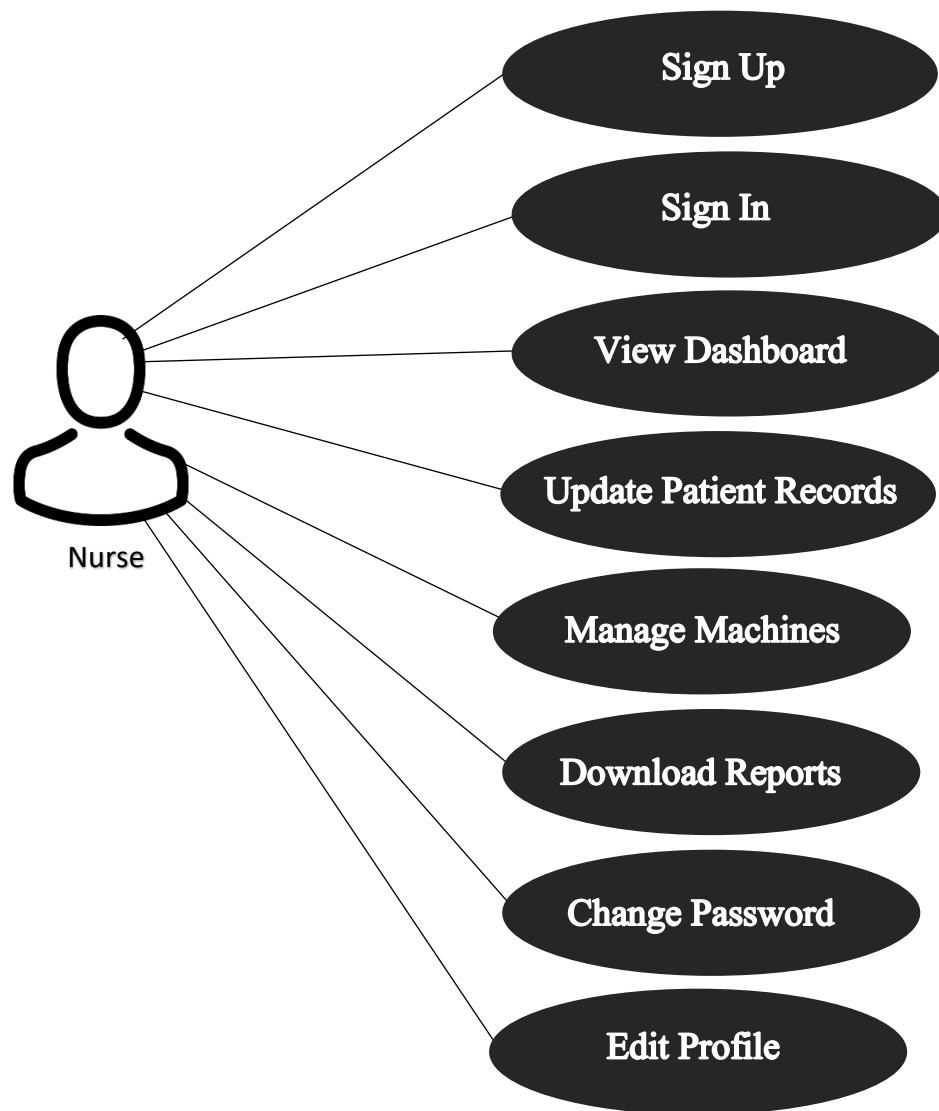


- **Patient Use Case:** The Patient shall be able to sign up for the system, then login to the system, interact with the dashboard and view what the system has to offer. Patients shall also be able to request for an appointment with the doctor, check drug prescription (if there's any), view and download medical reports, and update profile.



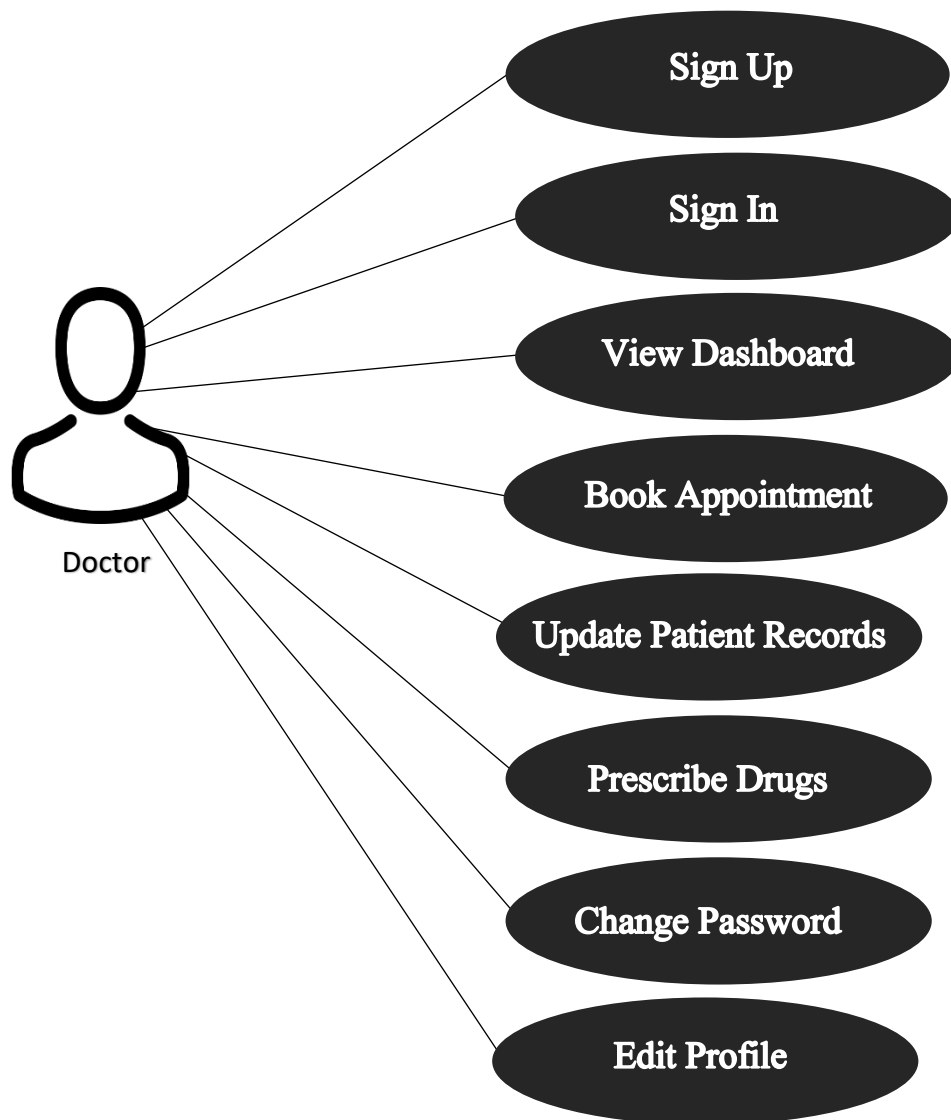
**Figure 3.3:** Patients use case

- **Nurse Use Case:** The Nurse shall be able to sign up for the system, then login to the system, interact with the dashboard and view what the system has to offer. Nurses shall also be able to check patients' previous records and update it, manage bed/ward and allocate bed to the patient (if needed), and update profile.



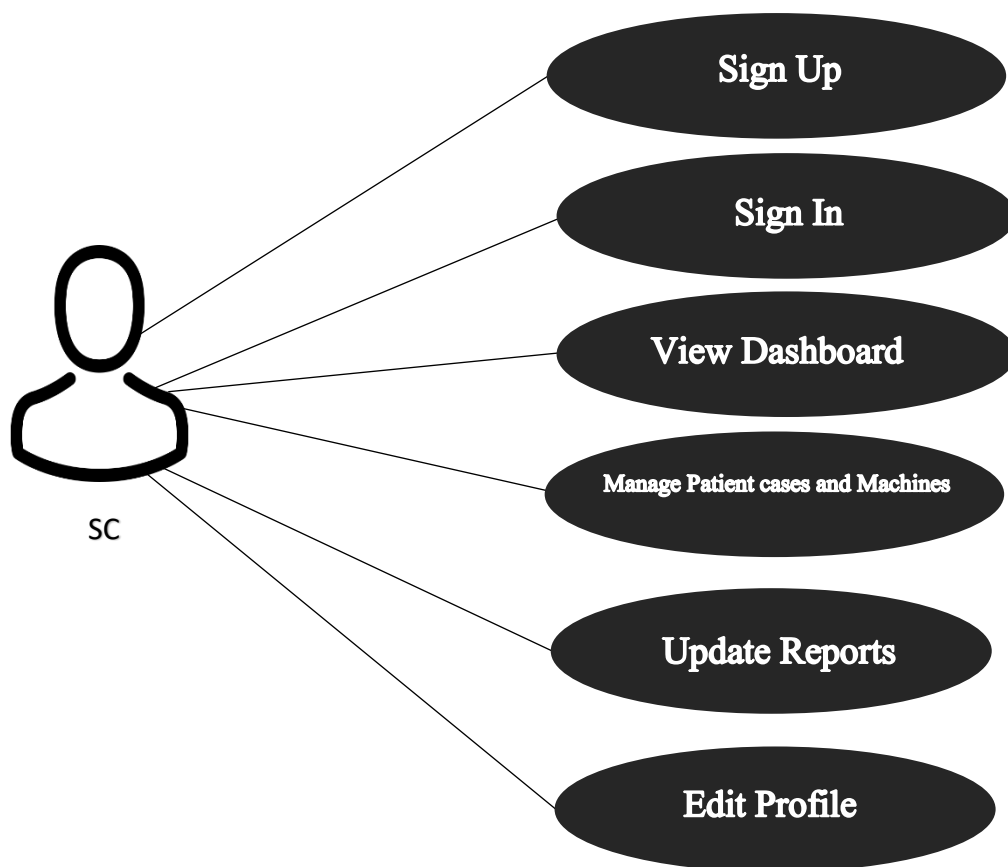
**Figure 3.4:** Nurses use case

- **Doctor Use Case:** The Doctor shall be able to sign up for the system, then login to the system, interact with the dashboard and view what the system has to offer. Doctors shall also be able to check patients' past records and update it, prescribe drugs for the patient, accept or reject patient's appointment, and update profile.



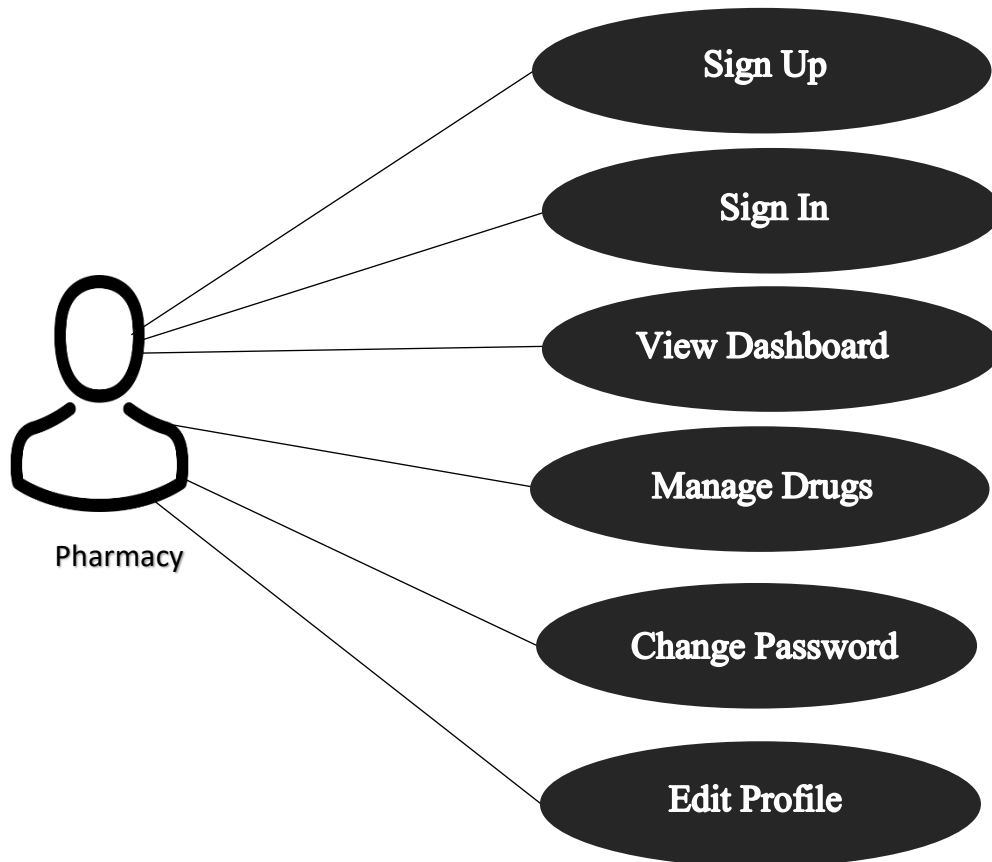
**Figure 3.5:** Doctor use case

- **Special Consultant Case:** The Special Consultants shall be able to sign up for the system, then login to the system, interact with the dashboard and view what the system has to offer. Special Consultant shall also be able to check the available machines, patients' previous records and update it with the current report, and update profile.



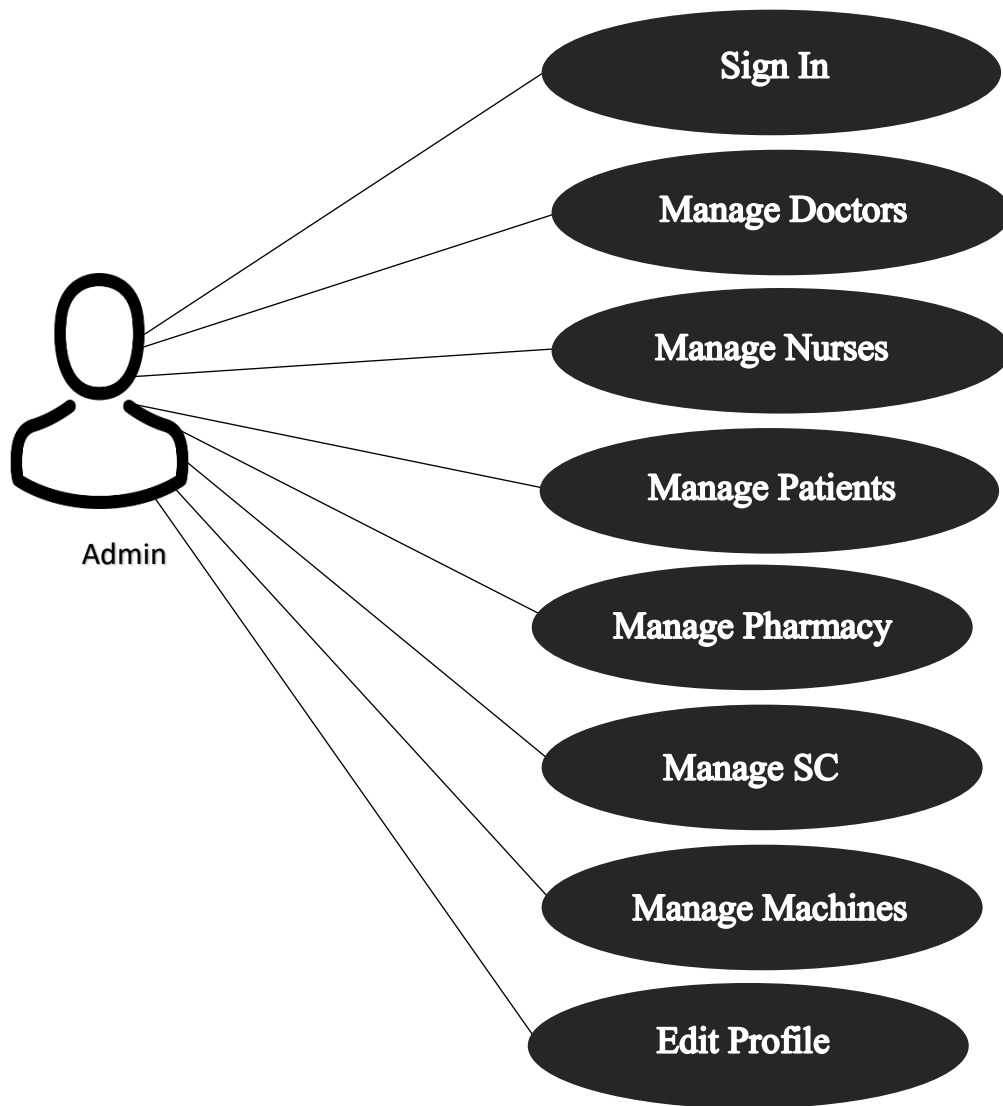
**Figure 3.6:** Special consultant case

- **Pharmacy Use Case:** The Pharmacist shall be able to sign up for the system, then login to the system, interact with the dashboard and view what the system has to offer. Pharmacist shall also be able to manage the drugs and edit profile.



**Figure 3.7:** Pharmacy use case

- **Admin Use Case:** The Admin shall be able to login to the system, interact with the dashboard and view what the system has to offer. Admins shall be able to manage doctors, manage patients, manage nurses, manage the lab, manage pharmacy, manage record officer, manage bed/ward and other departments in the hospital.



**Figure 3.8:** Admin use case

### **3.3 Design Details**

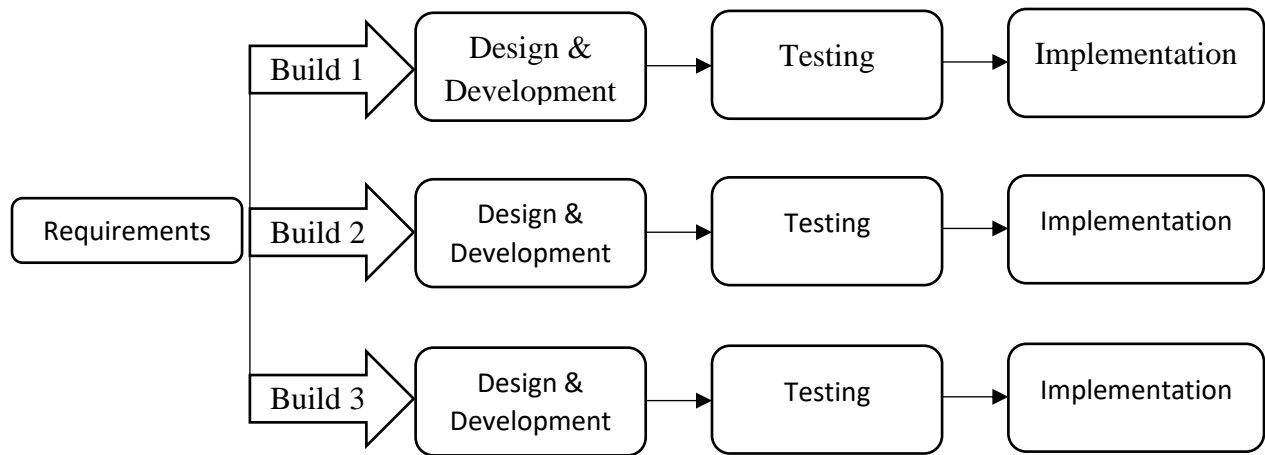
The Electronic Health Records System was developed on the Symfony Framework. Symfony is an open-source PHP web application framework, designed for developers who need a simple and elegant toolkit to create full-featured web applications. Symfony is sponsored by SensioLabs. It was developed by Fabien Potencier in 2005. Symfony is a Model-View-Controller (MVC) PHP framework used to develop web applications. The MVC is an architectural pattern that separates an application into three main logical components: The Model (data), the View (user interface), and the Controller (application logic).

This pattern helps to achieve separation of concerns. The Model part is responsible for managing the data of the application. The Controller is responsible for controlling the application logic and acts as the coordinator between the View and the Model, and the View is the component involved with the application's user interface. Using the MVC pattern for websites, requests are routed to a Controller which is responsible for working with the Model to perform actions and/or retrieve data. The Controller chooses the View to display, and provides it with the Model. The View renders the final page, based on the data in the Model. Laravel has a very rich set of features which will boost the speed of web development. (Tutorialspoint, 2019).

### **3.4 Software Development Life Cycle**

The SDLC framework adopted is the Iterative Incremental model. Initially, a temporary application of a complete scheme is built in an Iterative Incremental model so that it is in a deliverable state. The improved feature is introduced.

Defects are corrected from the previous shipment if any, and the operating item is supplied. The method will be reiterated until the full product development is finished. These procedures' repetitions are called iterations. An item increase is provided at the end of each iteration. (Tutorialspoint, 2019). The following diagram shows Iterative Incremental modeling:



**Figure 3.9:** Iterative incremental software development model

The advantages of this SDLC model include:

- i. You can develop prioritized requirements first.
- ii. Initial product delivery is faster.
- iii. Customers get important functionality early.
- iv. Lowers initial delivery cost

The demerits of this approach include the following:

- i. Requires effective planning of iterations.
- ii. Requires efficient design to ensure inclusion of the required functionality and provision for changes later.
- iii. Requires early definition of a complete and fully functional system to allow the definition of increments.
- iv. Well-defined module interfaces are required, as some are developed long before others are developed.
- v. The total cost of the complete system is not lower

### 3.5 Method of Data Collection

The method of data collection used for this project work is from both primary and secondary sources.



### **3.5.1 Primary Source**

This method of data collection was used to collect data from the doctors and nurses at the MTU Health Center. Some of the staff were interviewed on how operations are being carried out at the health center.

### **3.5.2 Secondary Source**

This includes the use of journals, internet, and books with related articles on EHR and EMR system to enlighten my understanding in developing the system.

## CHAPTER FOUR

### RESULT AND RESULT DISCUSSION

#### 4.0 Introduction and Documentation

This chapter demonstrates the information on implementing an electronic health record system. Designing and applying an EHR scheme takes into account some elements that try to provide solutions to the problems identified and identified in health care systems. It defines the instruments used to develop and implement the scheme. These instruments helped in the layout and creation of the system's main concept and feature to achieve its specified task.

It defines a range of functionalities with their inputs moving beneath each module. The whole concept is a scheme of stuff that works together as parts of a mechanism or interconnecting scheme.

collection of issues that work jointly as components of systems or as an interconnecting scheme is a collection of communicating or interdependent issues that form a collection of components to other components.

#### 4.1 Implementation

The system was built using the Symfony MVC Framework. The implementation was divided into three main parts which are frontend, backend, and database. The frontend was build using HTML, CSS, JavaScript, and jQuery. The frontend files were stored in the views folder of the MVC framework. The backend was built using Symfony PHP. The backend codes were kept in the Controller folder. The MySQL database was used for the implementation of the system in which the database codes were stored in the Model folder of the framework. Git was used in tracking changes made to the code.

#### 4.2 How the Electronic Health Record System works

The application works based on functional and non-functional requirements. This application includes the core phases of the software development lifecycle.

## **4.2.1 Functional and Non-functional Requirements**

The requirements of the system were used to design the program. These are what the system was based on during the development stages. Some of these requirements are functional while some others are non-functional in nature.

### **4.2.1.1 Functional Requirements**

An Electronic Health Records system should have the following functional requirements:

- i. A well-structured medium for information sharing between related coordinating departments.
- ii. Set up and maintain profiles for personnel in the clinic.
- iii. Pharmacy drug inventory.
- iv. Proper maintenance of medication history.
- v. Generate reports on the operations of the clinic
- vi. Comprehensive clinic follow up.

### **4.2.1.2 Non-functional Requirements**

The non-functional requirements include:

- i. Security: Access to the system is to be controlled through a unique username, password, and authentication. Modules are to be developed based on the user who is accessing the system.
- ii. Scalability
- iii. Reliability
- iv. Usability
- v. Cost-effective

## **4.3 System requirements**

Certain hardware parts and software components must be available on the scheme for effective use of the scheme. The scheme specification consists of the software and hardware components that enable the scheme to be created efficiently.

### **4.3.1 Software Requirements**

One of the main components in constructing a scheme is the compliant software part. The previous software is suggested for the effective execution of the scheme:

Front-end technologies:	HTML, CSS, Bootstrap, JavaScript, jQuery,
Backend technologies:	PHP, Symfony Framework
Database Management System:	phpMyAdmin, MySQL
Version Control Tool:	Git
Local Server:	XAMPP
IDE:	Visual Studio Code 2017
Web Browser:	Google Chrome, Mozilla Firefox, Microsoft Edge

### 4.3.2 Hardware Requirements

Hardware configuration is an important factor to put in mind when developing a system. Insufficient random-access memory may affect the speed and efficiency of the entire system. The processor should be powerful to handle the entire operations. The hard disk should also have sufficient capacity to store the file and application.

Processor:	Core i3 (Minimum)
Processor speed:	2.5GHz (Minimum)
RAM:	4GB (Minimum)
Hard disk:	500GB (Minimum)
Monitor Display:	LED
Mouse:	Touchpad with multi-touch gesture support, USB or PS/2

#### 4.4 Screenshots of Pages Implemented

The screenshots of the implemented pages show the different views of the users depending on their roles with a brief description of what it entails.

##### a. Login

This is the first page that appears when the URL of the HMS is typed in any browser. The login form is used to authenticate users' credentials before giving them access to their privileges. The login form contains a field for the email and another for the password. When the login form is submitted, the code checks that the credentials are authentic, given the user access to the restricted page. If a user is not authenticated correctly, they will not be able to proceed past the login form.

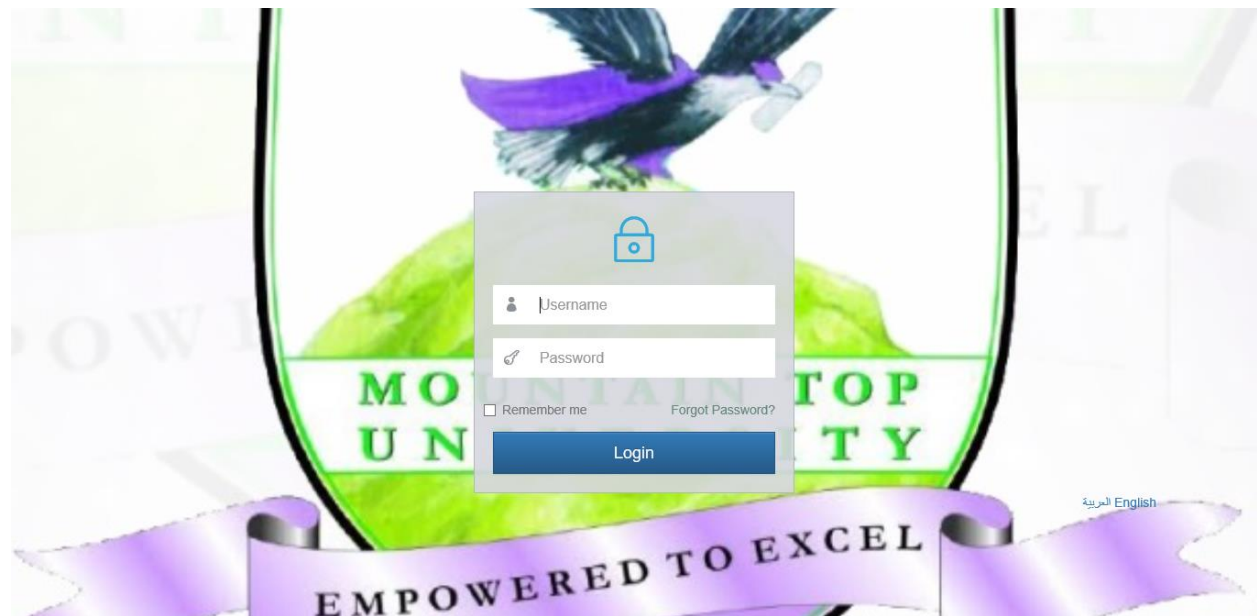
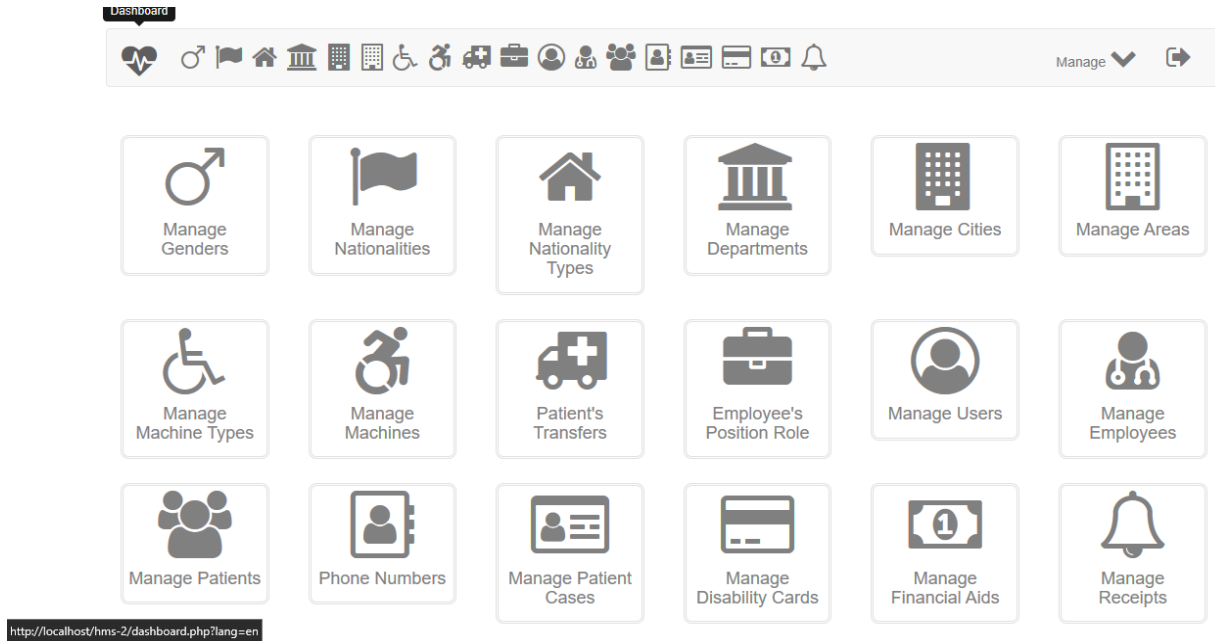


Figure 4.1: Login Page

## b. Administrator Dashboard

This interface shows the general view of the system without any form of restrictions. The administrator can add users, patients', all related entities. The administrator can also update and delete users.



**Figure 4.2:** Admin Dashboard

## c. Admin Manage Departments

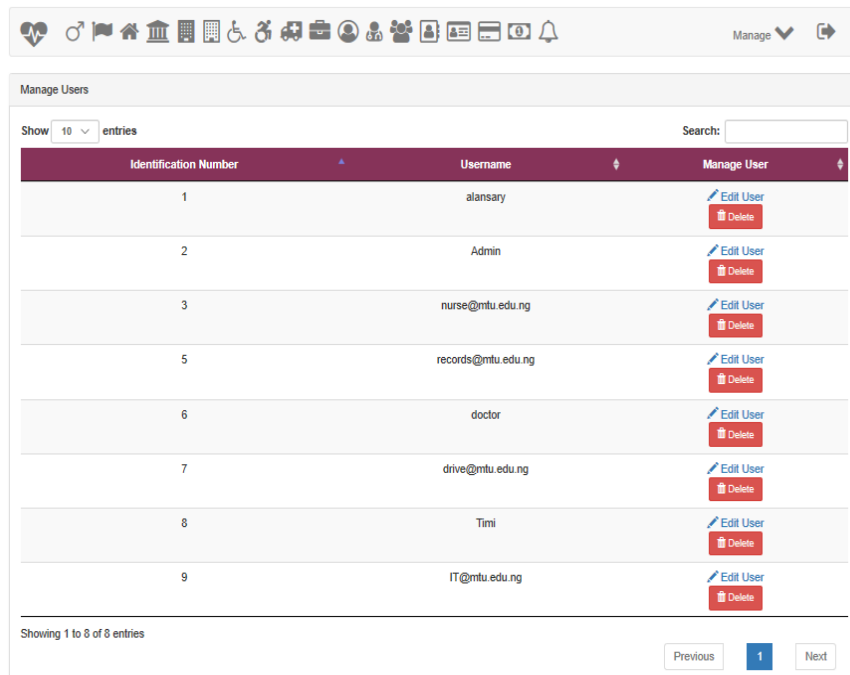
This page shows how the administrator manages the various departments that are found in the clinic.

Identification Number	Department	Manage Department
1	Receptionist	<a href="#">Edit Department</a> <a href="#">Delete</a>
2	Nursing Department	<a href="#">Edit Department</a> <a href="#">Delete</a>
3	Lab Technicians	<a href="#">Edit Department</a> <a href="#">Delete</a>
4	Pharmaceutical	<a href="#">Edit Department</a> <a href="#">Delete</a>
5	Special Consultants	<a href="#">Edit Department</a> <a href="#">Delete</a>
6	Doctor	<a href="#">Edit Department</a> <a href="#">Delete</a>
7	Drivers	<a href="#">Edit Department</a> <a href="#">Delete</a>

**Figure 4.3:** Manage Departments

#### d. Admin Manage Users

This interface shows the various users that have been given access to make efficient use of the system.



Identification Number	Username	Manage User
1	alansary	<a href="#">Edit User</a> <a href="#">Delete</a>
2	Admin	<a href="#">Edit User</a> <a href="#">Delete</a>
3	nurse@mtu.edu.ng	<a href="#">Edit User</a> <a href="#">Delete</a>
5	records@mtu.edu.ng	<a href="#">Edit User</a> <a href="#">Delete</a>
6	doctor	<a href="#">Edit User</a> <a href="#">Delete</a>
7	drive@mtu.edu.ng	<a href="#">Edit User</a> <a href="#">Delete</a>
8	Timi	<a href="#">Edit User</a> <a href="#">Delete</a>
9	IT@mtu.edu.ng	<a href="#">Edit User</a> <a href="#">Delete</a>

**Figure 4.4:** Admin Manage Users

#### e. Admin Manage Patients

This view makes the administrator view all the registered patients, the admin can update and delete the entries of the patients on the system.



Manage Patients

Show 10 entries Search:

Identification Number	First Name	Middle Name	Last Name	Civil Id	Date of Birth	Creation Time	Gender	Nationality	Manage Patient
1	Tejiri	Emmanuel	Anthony	0	1997-03-17	2019-07-24 11:29:27	Male	Nigeria	<a href="#">Patient Edit</a> <a href="#">Delete</a>
2	Doubra	Daniella	Amaosuomo	123456789123	1993-05-30	2019-07-29 12:15:38	Female	Nigeria	<a href="#">Patient Edit</a> <a href="#">Delete</a>
3	Kolawole	Smith	Daramola	123456123456	1999-07-31	2019-07-29 12:15:38	Male	Nigeria	<a href="#">Patient Edit</a> <a href="#">Delete</a>
4	Chinedu	Nicholas	Ogeleka	123123456789	1997-07-31	2019-07-29 12:18:03	Male	Nigeria	<a href="#">Patient Edit</a> <a href="#">Delete</a>
5	Olamide	Joshua	Jacob	444444444444	1996-05-08	2019-07-29 12:18:03	Male	Nigeria	<a href="#">Patient Edit</a> <a href="#">Delete</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

[+ Add New Patient](#)

**Figure 4.5:** Admin Manage Patients

## f. Patients' Registration

This interface provides the platform for a new patient to register his or her details into the system.

The screenshot displays a web interface for patient registration. At the top, there is a navigation bar with various icons and a 'Manage' dropdown menu. Below this is a yellow warning box with the text: 'Please make sure to add data in the following first:' followed by a bulleted list: 'Add the gender of the patient in the genders table if not exists', 'Add the nationality of the patient in the nationalities table if not exists', and 'Add the type of the nationality in the nationality types table if not exists'. The main form is titled 'Add New Patient' and contains the following fields: 'First Name' (text input), 'Middle Name' (text input), 'Last Name' (text input), 'Civil Id' (text input), 'Passport Number' (text input), 'Date of Birth' (date picker), 'Personal image' (with an 'Browse...' button), 'Other Notes' (text input), 'Gender' (dropdown menu with 'Male' selected), 'Nationality' (dropdown menu with 'Nigeria' selected), and 'Nationality Type' (dropdown menu with 'Citizen' selected). A blue 'Add' button is located at the bottom left of the form.

**Figure 4.6:** Patient Registration

### g. Doctor Manage Patient Cases

This view allows the doctor to view the patient cases that have been attended to and the amounts that have been made, the hospital that the patient was transferred from to the clinic, the equipment used and the department that attended to the case.

Identification Number	Amount Paid	Case Date	Patient Name	Transferred From	Machine	Department	Manage Patient Case
1	1000.00	2019-07-24	Tejiri Emmanuel Anthony	Not Transferred		Doctor	<a href="#">Edit Case</a> <a href="#">Delete</a>
2	10000.00	2019-07-29	Doubra Daniella Amaosuomo	Mecure Hospital Isolo	Saline Eyewash	Doctor	<a href="#">Edit Case</a> <a href="#">Delete</a>
3	30000.00	2019-07-29	Chinedu Nicholas Ogeleka	Mecure Hospital Isolo	Otosopes	Doctor	<a href="#">Edit Case</a> <a href="#">Delete</a>

**Figure 4.7 :** Doctor Manage Patient Cases

## **CHAPTER FIVE**

### **SUMMARY, RECOMMENDATION, AND CONCLUSION**

#### **5.0 Summary**

Having reviewed the challenges encountered through the manual use of files in keeping records in the hospital, this proposed system is believed to help eradicate the problems associated with the manual method by the introduction of an online management system which will make documentation more efficient and effective for serving the patients better and to have a more developed way of carrying out hospital operations. The electronic health record system adds amazing values to the lives of both staff and patients as it aids in the reduction of workload and reduction of data error. It can capture data, store, and view, add and delete records into the database when required. On the long run during the development of this system, some challenges were encountered, some of them includes it not being responsive on mobile and tablet devices, and the laboratory and pharmacy features were not implemented.

#### **5.1 Contribution to Knowledge**

The main contribution of knowledge was the ability to implement an electronic health record system using the Symfony Framework. With the help of Laravel, the system is protected from serious security risks like cross-site request forgery, SQL injection, and cross-site scripting. Patients can also be reminded of their appointment schedule with a doctor through their mails.

#### **5.2 Limitations**

- i. The system is not user-friendly on tablet and mobile devices
- ii. Due to limited data, the laboratory and pharmacy features of the system could not be implemented

#### **5.3 Recommendation for Further Study**

It is recommended that hospitals currently practicing the manual system should switch to the electronic system because it is more efficient and easier to use. Also, since the use of computers is

growing fast globally, introducing the electronic system will enable hospitals to fit into the current global trend. The following are also recommended.

- i. Other types of authentication methods (Single-factor, Two-factor, Multi-factor authentication) can be used for securing hospital management systems.
- ii. Other types of frameworks (AngularJS, Laravel, ASP.NET) can be used in building electronic health record systems.
- iii. Other types of database management systems can be used by electronic health record system.

#### **5.4 Conclusion**

This project takes a look at the various problems associated with the existing system which are improper documentation, loss, and mismatch of patient's data, time wastage amongst others. With all these problems being critically analyzed, a solution was embarked on, to eliminate these problems. With the design of an electronic health record system, such problems are considered to be eradicated. In conclusion, an electronic health record system not only provides an opportunity for the hospital to enhance their patient care but also can increase the profitability of the organization. Hospital administrators would be able to significantly improve the operational control and thus streamline operations. This would improve the response time to the demands of patient care because it automates the process of collecting, collating and retrieving patient information.

## REFERENCES

- Adeleke, I. T., Lawal, A. H., Adio, R. A., & Adebisi, A. A. (2015). *Information technology skills and training needs of health information management professionals in Nigeria: a nationwide study*. *Health Information Management Journal*, 44(1), 30-38.
- Adebisi, O.A, Oladosu, D.A, Busari, O.A and Oyewola Y.V. (2015). Design and Implementation of Hospital Management System. *International Journal of Engineering and Innovative Technology*, 5(1), 2277-3754.
- Agnes, N.O (2011). Automated Hospital Management System.
- Ahmed, B., and Usman, M. (2017). Design and Implementation of Outpatient Management System. *International Journal of Advanced Academic Research*, 3(6), 2488-9849.
- Boonstra, A., & Broekhuis, M. (2010). Barriers to the acceptance of electronic medical records by physicians from systematic review to taxonomy and interventions. *BMC health services research*, 10(1), 231.
- Codecanyon. (n.d.). Bayanno Hospital Management System. Retrieved from <https://codecanyon.net/item/bayanno-hospital-management-system/5814621>
- Farzadkia, M., Moradi, A., Mohammadi, M. S., & Jorfi, S. (2008). Hospital waste management status in Iran: a case study in the teaching hospitals of Iran University of Medical Sciences. *Waste Management & Research*, 27(4), 384-389.
- Healthcare Business and Technology. (2016). Management in Hospitals: A Vital Piece of the Healthcare Puzzle. Retrieved from <http://www.healthcarebusinesstech.com/hospital-management/>
- Ilo, S., Igbajar A., Acholonu, J. (2015). Designing A Web Based Hospital Management System for MOUAU Clinic. *International Journal of Trend in Research and Development*, 2(6), 2394-9333.

Kapoor, L, Singh, I. (2014). Telemedicine in India: Current Scenario and the Future. *Telemedicine and e-Health*. 15(3), 568-75.

Okwor, E.D. (2012). Design and Implementation of an Electronic Patient Management System.

Srivastava, S. (2013). Evolving Role of Hospital Management in Health Care Delivery in India. *Primary Health Care Open Access*, 07(01).

Technecon Healthcare. (2016). Importance of Hospital Management. Retrieved from <https://www.techneconhealthcare.com/blog/importance-of-hospital-management-in-healthcare-institutions.html>

(2011, December 03). Retrieved January 22, 2019, from <https://en.wikipedia.org/wiki/Inpatient>

## APPENDIX

### Source Code Database

```
CREATE DATABASE hospital;  
USE hospital;
```

```
CREATE TABLE IF NOT EXISTS genders (  
    `id` INT(11) AUTO_INCREMENT,  
    `gender` VARCHAR(6) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS nationalities (  
    `id` INT(11) AUTO_INCREMENT,  
    `nationality` VARCHAR(50) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS nationalityTypes (  
    `id` INT(11) AUTO_INCREMENT,  
    `type` VARCHAR(11) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS departments (  
    `id` INT(11) AUTO_INCREMENT,
```



```
    `name` VARCHAR(100) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS users (  
    `id` INT(11) AUTO_INCREMENT,  
    `username` VARCHAR(50) NOT NULL UNIQUE,  
    `password` CHAR(60) NOT NULL,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS auth_tokens (  
    `id` INTEGER(11) NOT NULL AUTO_INCREMENT,  
    `selector` CHAR(12),  
    `token` CHAR(64),  
    `user_id` INTEGER(11) NOT NULL,  
    `expires` DATETIME,  
    PRIMARY KEY (`id`),  
    FOREIGN KEY (`user_id`) REFERENCES users (`id`) ON DELETE CASCADE  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS cities (  
    `id` INT(11) AUTO_INCREMENT,  
    `name` VARCHAR(50) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```

CREATE TABLE IF NOT EXISTS areas (
    `id` INT(11) AUTO_INCREMENT,
    `name` VARCHAR(50) NOT NULL,
    `city_id` INT(11) NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`city_id`) REFERENCES cities (`id`) ON DELETE RESTRICT ON
    UPDATE CASCADE,
    UNIQUE(`name`, `city_id`)
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
    AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS machineTypes (
    `id` INT(11) AUTO_INCREMENT,
    `type` VARCHAR(100) NOT NULL UNIQUE,
    PRIMARY KEY (`id`)
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
    AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS machines (
    `id` INT(11) AUTO_INCREMENT,
    `arName` VARCHAR(100) NOT NULL,
    `enName` VARCHAR(100) NOT NULL,
    `description` VARCHAR(255) DEFAULT NULL,
    `type_id` INT(11) NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`type_id`) REFERENCES machineTypes (`id`) ON DELETE
    RESTRICT ON UPDATE CASCADE,
    UNIQUE(`arName`, `type_id`),
    UNIQUE(`enName`, `type_id`)
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
    AUTO_INCREMENT 1;

```

```
CREATE TABLE IF NOT EXISTS transfers (  
    `id` INT(11) AUTO_INCREMENT,  
    `name` VARCHAR(100) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS positionRoles (  
    `id` INT(11) AUTO_INCREMENT,  
    `role` VARCHAR(100) NOT NULL UNIQUE,  
    PRIMARY KEY (`id`)  
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci  
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS employees (  
    `id` INT(11) AUTO_INCREMENT,  
    `firstName` VARCHAR(30) NOT NULL,  
    `middleName` VARCHAR(30) NOT NULL,  
    `lastName` VARCHAR(30) NOT NULL,  
    `civil_id` CHAR(12) NOT NULL UNIQUE,  
    `passport_number` VARCHAR(9) DEFAULT NULL,  
    `image` VARCHAR(255) DEFAULT NULL,  
    `dob` DATE NOT NULL,  
    `hireDate` DATE NOT NULL,  
    `creationTime` DATETIME DEFAULT CURRENT_TIMESTAMP,  
    `modificationTime` DATETIME ON UPDATE CURRENT_TIMESTAMP,  
    `department_id` INT(11) NOT NULL,  
    `gender_id` INT(11) NOT NULL,  
    `nationality_id` INT(11) NOT NULL,  
    `nationalityType_id` INT(11) NOT NULL,
```

```

`positionRole_id` INT(11) NOT NULL,
PRIMARY KEY (`id`),
FOREIGN KEY (`department_id`) REFERENCES departments (`id`) ON DELETE
RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (`gender_id`) REFERENCES genders (`id`) ON DELETE RESTRICT
ON UPDATE CASCADE,
FOREIGN KEY (`nationality_id`) REFERENCES nationalities (`id`) ON DELETE
RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (`nationalityType_id`) REFERENCES nationalityTypes (`id`) ON
DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (`positionRole_id`) REFERENCES positionRoles (`id`) ON DELETE
RESTRICT ON UPDATE CASCADE
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS patients (
`id` INT(11) AUTO_INCREMENT,
`firstName` VARCHAR(30) NOT NULL,
`middleName` VARCHAR(30) NOT NULL,
`lastName` VARCHAR(30) NOT NULL,
`civil_id` CHAR(12) NOT NULL UNIQUE,
`passport_number` VARCHAR(9) DEFAULT NULL,
`dob` DATE NOT NULL,
`creationTime` DATETIME DEFAULT CURRENT_TIMESTAMP,
`modificationTime` DATETIME ON UPDATE CURRENT_TIMESTAMP,
`image` VARCHAR(255) DEFAULT NULL,
`notes` VARCHAR(255) DEFAULT NULL,
`gender_id` INT(11) NOT NULL,
`nationality_id` INT(11) NOT NULL,
`nationalityType_id` INT(11) NOT NULL,
PRIMARY KEY (`id`),

```

```

FOREIGN KEY (`gender_id`) REFERENCES genders (`id`) ON DELETE RESTRICT
ON UPDATE CASCADE,
FOREIGN KEY (`nationality_id`) REFERENCES nationalities (`id`) ON DELETE
RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (`nationalityType_id`) REFERENCES nationalityTypes (`id`) ON
DELETE RESTRICT ON UPDATE CASCADE
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS phoneNumbers (
    `id` INT(11) AUTO_INCREMENT,
    `number` VARCHAR(11) NOT NULL,
    `typeDiscriminator` VARCHAR(255) NOT NULL,
    `phoneOf_id` INT(11) NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE (`number`, `typeDiscriminator`, `phoneOf_id`)
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS articles (
    `id` INT(11) AUTO_INCREMENT,
    `price` NUMERIC(8, 2),
    `date` DATE NOT NULL,
    `patient_id` INT(11) NOT NULL,
    `transfer_id` INT(11) DEFAULT NULL,
    `machine_id` INT(11) DEFAULT NULL,
    `department_id` INT(11) NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`patient_id`) REFERENCES patients (`id`) ON DELETE RESTRICT
ON UPDATE CASCADE,

```

```

FOREIGN KEY (`transfer_id`) REFERENCES transfers (`id`) ON DELETE RESTRICT
ON UPDATE CASCADE,
FOREIGN KEY (`machine_id`) REFERENCES machines (`id`) ON DELETE RESTRICT
ON UPDATE CASCADE,
FOREIGN KEY (`department_id`) REFERENCES departments (`id`) ON DELETE
RESTRICT ON UPDATE CASCADE
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS disabilities (
    `id` INT(11) AUTO_INCREMENT,
    `description` VARCHAR(255) DEFAULT NULL,
    `amount` NUMERIC(8, 2) NOT NULL,
    `article_id` INT(11) NOT NULL,
    `date` DATE NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`article_id`) REFERENCES articles (`id`) ON DELETE RESTRICT ON
UPDATE CASCADE
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
AUTO_INCREMENT 1;

```

```

CREATE TABLE IF NOT EXISTS financialAids (
    `id` INT(11) AUTO_INCREMENT,
    `description` VARCHAR(255) DEFAULT NULL,
    `amount` NUMERIC(8, 2) NOT NULL,
    `article_id` INT(11) NOT NULL,
    `date` DATE NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`article_id`) REFERENCES articles (`id`) ON DELETE RESTRICT ON
UPDATE CASCADE

```

```
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
    AUTO_INCREMENT 1;
```

```
CREATE TABLE IF NOT EXISTS receipts (
    `id` INT(11) AUTO_INCREMENT,
    `receiptDate` DATE NOT NULL,
    `deliveryDate` DATE NOT NULL,
    `article_id` INT(11) NOT NULL,
    `machine_id` INT(11) NOT NULL,
    `employee_id` INT(11) NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`article_id`) REFERENCES articles (`id`) ON DELETE RESTRICT ON
    UPDATE CASCADE,
    FOREIGN KEY (`machine_id`) REFERENCES machines (`id`) ON DELETE RESTRICT
    ON UPDATE CASCADE,
    FOREIGN KEY (`employee_id`) REFERENCES employees (`id`) ON DELETE
    RESTRICT ON UPDATE CASCADE
)ENGINE InnoDB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci
    AUTO_INCREMENT 1;
```

```
INSERT INTO `users` (`id`, `username`, `password`) VALUES
(1, 'gbotemi', '$2y$11$bVW9W6I7Jbuy0l9GsX4RMuAPS2kI33mICBkeXlmDjfG6DwWd6ar/6');
```

## DEPARTMENTS

```
<?php require_once "layouts/header.php"; ?>
<?php if (isset($_SESSION['error'])) { ?>
    <div class="alert alert-danger">
        <?php
            echo $_SESSION['error'];
            unset($_SESSION['error']);
        ?>
    ?>
```

```

    </div>
<?php } ?>
<?php if (isset($_SESSION['success'])) { ?>
    <div class="alert alert-success">
        <?php
            echo $_SESSION['success'];
            unset($_SESSION['success']);
        ?>
    </div>
<?php } ?>

<?php

if (isset($_GET['manage']) && $_GET['manage'] == 'view') {

/*****
* VIEW DEPARTMENTS
*****/

?>

<div class="panel panel-default">
    <div class="panel-heading">
        <?php echo language("departments-heading", $_SESSION['lang']); ?>
    </div>
    <div class="panel-body">
        <table class="table table-striped table-responsive table-hover text-center">
            <thead>
                <tr>
                    <th><?php echo language('id', $_SESSION['lang']); ?></th>
                    <th><?php echo language('departments-dept', $_SESSION['lang']); ?></th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td><?php echo language('id', $_SESSION['lang']); ?></td>
                    <td><?php echo language('departments-dept', $_SESSION['lang']); ?></td>
                </tr>
            </tbody>
        </table>
    </div>
</div>

```



```

        <th><?php echo language('departments-manage', $_SESSION['lang']); ?></th>
    </tr>
</thead>
<tbody>
    <?php
        $query = "SELECT * FROM departments";
        $stmt = Connection::conn()->prepare($query);
        $stmt->execute();
        while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            extract($row);
            ?>
            <tr>
                <td><?php echo $id; ?></td>
                <td><?php echo $name; ?></td>
                <td>
                    <a href="departments.php?lang=<?php echo $selectedLang;
?>&manage=edit&id=<?php echo $id; ?>">
                        <span class="glyphicon glyphicon-pencil" aria-hidden="true"></span>
                        <?php echo language('departments-edit', $_SESSION['lang']); ?>
                    </a>
                    <form action="departments.php?lang=<?php echo $selectedLang;
?>&manage=delete" method="POST">
                        <input type="hidden" name="id" value="<?php echo $id; ?>">
                        <button type="submit" class="btn btn-danger btn-sm">
                            <span class="glyphicon glyphicon-trash"></span> <?php echo
language("delete", $_SESSION['lang']); ?>
                        </button>
                    </form>
                </td>
            </tr>
        <?php } ?>

```

```

        </tbody>
    </table>
</div>
<div class="panel-footer">
    <a href="departments.php?lang=<?php echo $selectedLang; ?>&manage=add" class="btn
        btn-primary btn-lg">
        <span class="glyphicon glyphicon-plus" aria-hidden="true"></span>
        <?php echo language('departments-add', $_SESSION['lang']); ?>
    </a>
</div>
</div>

<?php } elseif (isset($_GET['manage']) && $_GET['manage'] == 'add') {

/*****
* ADD DEPARTMENT
*****/

?>
<!-- ADDING DEPARTMENT -->
<div class="panel panel-primary">
    <div class="panel-heading">
        <?php echo language('departments-add', $_SESSION['lang']); ?>
    </div>
    <div class="panel-body">
        <form action="departments.php?manage=store" method="POST" data-parsley-
            validate="">
            <div class="form-group">
                <label for="department"><?php echo language('departments-dept',
                    $_SESSION['lang']); ?></label>

```

```

        <input type="text" placeholder="<?php echo language('departments-dept',
$_SESSION['lang']); ?>"
            value="<?php echo isset($_SESSION['department']) ? $_SESSION['department'] :
"; ?>" required=""
            class="form-control" name="department" data-parsley-required="true" data-
parsley-length="[1, 100]"
        />
    </div>
</div>
<div class="panel-footer">
    <input type="submit" value="<?php echo language('add', $_SESSION['lang']); ?>"
        class="btn btn-primary btn-lg"/>
</form>
</div>
</div>
<?php } elseif (isset($_GET['manage']) && $_GET['manage'] == 'store') {

/*****
* STORE DEPARTMENT
*****/

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['department'])) {
        $department = filter_var(testInput($_POST['department']),
FILTER_SANITIZE_STRING);
        // if (strlen($department) > 100 || strlen($department) < 1) {
        //     $_SESSION['error'] = language('departments-check-error', $_SESSION['lang']);
        //     header('Location: departments.php?manage=add&lang='.$selectedLang);
        //     die();
        // }
        try {

```

```

$query = "INSERT INTO departments (name) VALUES (:name)";
$stmt = Connection::conn()->prepare($query);
$stmt->bindParam(':name', $department, PDO::PARAM_STR);
$stmt->execute();
} catch(PDOException $e) {
    dbError($stmt, 'departments.php?manage=add&lang='.$selectedLang);
    die();
}
$_SESSION['success'] = language('departments-add-success', $_SESSION['lang']);
header('Location: departments.php?manage=view&lang='.$selectedLang);
die();
}
else {
    $_SESSION['error'] = language('departments-required', $_SESSION['lang']);
    header('Location: departments.php?manage=add&lang='.$selectedLang);
    die();
}
}
else {
    $_SESSION['error'] = language('method_not_allowed_error', $_SESSION['lang']);
    header('Location: departments.php?manage=add&lang='.$selectedLang);
    die();
}
} elseif (isset($_GET['manage']) && $_GET['manage'] == 'edit') {

/*****
* EDIT DEPARTMENT
*****/

?>

```

```

<?php
if (isset($_GET['id'])) {
    $id = filter_var(testInput($_GET['id']), FILTER_VALIDATE_INT);
    $query = "SELECT * FROM departments WHERE id = :id";
    $stmt = Connection::conn()->prepare($query);
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);
    $stmt->execute();
    if ($stmt->rowCount()) {
        extract($stmt->fetch(PDO::FETCH_ASSOC));
    }
}

<div class="panel panel-primary">
    <div class="panel-heading">
        <?php echo language('departments-edit', $_SESSION['lang']); ?>
    </div>
    <div class="panel-body">
        <form action="departments.php?manage=update" method="POST" data-parsley-
validate="">
            <div class="form-group">
                <label for="department"><?php echo language('departments-dept',
$_SESSION['lang']); ?></label>
                <input type="text" placeholder="<?php echo language('departments-dept',
$_SESSION['lang']); ?>"
                value="<?php echo $name; ?>" name="department" required="" class="form-
control"
                data-parsley-required="true" data-parsley-length="[1, 100]"
                />
            </div>
        </div>
    </div>
    <div class="panel-footer">
        <input type="hidden" name="id" value="<?php echo $id; ?>" />
        <input type="submit" value="<?php echo language('edit', $_SESSION['lang']); ?>"

```

```

        class="btn btn-primary btn-lg">
    </form>
</div>
</div>
<?php } else {
    $_SESSION['error'] = language('id-not-found', $_SESSION['lang']);
    header('departments.php?lang='.$selectedLang);
    die();
}
}
else {
    $_SESSION['error'] = language('id-required', $_SESSION['lang']);
    header('Location: departments.php?lang='.$selectedLang);
    die();
}

} elseif (isset($_GET['manage']) && $_GET['manage'] == 'update') {

/*****
* UPDATE DEPARTMENT
*****/

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['department'])) {
        $department = filter_var(testInput($_POST['department']),
        FILTER_SANITIZE_STRING);
        // if (strlen($department) > 100 || strlen($department) < 1) {
        //     $_SESSION['error'] = language('departments-check-error', $_SESSION['lang']);
        //     header('Location:
        departments.php?manage=edit&id='.$_POST['id'].'&lang='.$selectedLang);
        //     die();
    }
}

```

```

// }
try {
    $query = "UPDATE departments SET name = :name WHERE id = :id";
    $stmt = Connection::conn()->prepare($query);
    $stmt->bindParam(':name', $department, PDO::PARAM_STR);
    $stmt->bindParam(':id', $_POST['id'], PDO::PARAM_INT);
    $stmt->execute();
} catch(PDOException $e) {
    dbError($stmt,
'departments.php?manage=edit&id='.$_POST['id'].'&lang='.$selectedLang);
    die();
}
$_SESSION['success'] = language('departments-update-success', $_SESSION['lang']);
header('Location: departments.php?manage=view&lang='.$selectedLang);
die();
}
else {
    $_SESSION['error'] = language('departments-required', $_SESSION['lang']);
    header('Location:
departments.php?manage=edit&id='.$_POST['id'].'&lang='.$selectedLang);
    die();
}
}
else {
    $_SESSION['error'] = language('method_not_allowed_error', $_SESSION['lang']);
    header('Location: departments.php?manage=view&lang='.$selectedLang);
    die();
}
} elseif (isset($_GET['manage']) && $_GET['manage'] == 'delete') {

/*****

```

## \* DELETE DEPARTMENT

\*\*\*\*\*/

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_POST['id'])) {
        $id = filter_var(testInput($_POST['id']), FILTER_VALIDATE_INT);
        try {
            $query = "DELETE FROM departments WHERE id = :id";
            $stmt = Connection::conn()->prepare($query);
            $stmt->bindParam(':id', $id, PDO::PARAM_INT);
            $stmt->execute();
            $_SESSION['success'] = language("departments-delete-success", $_SESSION['lang']);
            header('Location: departments.php?manage=view&lang='.$selectedLang);
            die();
        } catch (PDOException $e) {
            dbError($stmt, 'departments.php?manage=view&lang='.$selectedLang);
            die();
        }
    }
}
else {
    $_SESSION['error'] = language('id-required', $_SESSION['lang']);
    header('Location: departments.php?manage=view&lang='.$selectedLang);
    die();
}
}
else {
    $_SESSION['error'] = language('method_not_allowed_error', $_SESSION['lang']);
    header('Location: departments.php?manage=view&lang='.$selectedLang);
    die();
}
```



```
} else {  
    $_SESSION['error'] = language('page-not-found', $_SESSION['lang']);  
    header('Location: departments.php?manage=view&lang='.$selectedLang);  
} ?>
```

```
<?php require_once "layouts/footer.php"; ?>
```

---