Table of Contents

[CHAPTER ONE 4](#_Toc492291588)

[INTRODUCTION 4](#_Toc492291589)

[1.1 Background of the study 4](#_Toc492291590)

[1.2 Statement of problem 5](#_Toc492291591)

[1.3 Motivation 5](#_Toc492291592)

[1.4 Aims and objectives 5](#_Toc492291593)

[1.5 Purpose of the study 6](#_Toc492291594)

[1.6 Significance of the study 6](#_Toc492291595)

[1.7 Organization of the work 7](#_Toc492291596)

[1.8 Definition of terms 7](#_Toc492291597)

[CHAPTER TWO 8](#_Toc492291598)

[LITERATURE REVIEW 8](#_Toc492291599)

[2.1 Introduction 8](#_Toc492291600)

[2.2 Records 9](#_Toc492291601)

[2.3 Patient 9](#_Toc492291602)

[2.4 Existing System 9](#_Toc492291603)

[2.5 Computer use in the medical field 10](#_Toc492291604)

[2.6 Improving Patient Records 10](#_Toc492291605)

[2.7 The Computer-Based Patient Record System 11](#_Toc492291606)

[2.8 Electronic health records 11](#_Toc492291607)

[2.9 How developments in technology and data in the NHS are improving outcomes for patients 12](#_Toc492291608)

[2.10 Federal Health Architecture 13](#_Toc492291609)

[2.11 Patient Engagement within the Patient-Centered Medical Home Model 13](#_Toc492291610)

[2.12 All Systems Go: The Voice of Nurse Practitioners Using Electronic Patient Records 14](#_Toc492291611)

[CHAPTER THREE 14](#_Toc492291612)

[SYSTEM DESIGN AND METHODOLOGY 14](#_Toc492291613)

[3.1 Introduction 15](#_Toc492291614)

[3.2 Software Development Cycle 15](#_Toc492291615)

[3.3 Design phase using unified modelling language 20](#_Toc492291616)

[3.4 Flowcharts 20](#_Toc492291617)

[3.5 Use case diagram 21](#_Toc492291618)

[3.6 Entity Relationship Diagram (ERD) 22](#_Toc492291619)

[3.7 Data Flow Diagrams (DFD) 22](#_Toc492291620)

[3.8 Level 0 DFD 23](#_Toc492291621)

[3.9 Non-Functional Requirement 24](#_Toc492291622)

[3.10 Functional Requirements 24](#_Toc492291623)

[CHAPTER 4 26](#_Toc492291624)

[IMPLEMENTATION AND EVALUATION 26](#_Toc492291625)

[4.1 Introduction 26](#_Toc492291626)

[4.2 Hardware requirement 26](#_Toc492291627)

[4.3 Software requirement 26](#_Toc492291628)

[4.4 Programming platform 27](#_Toc492291629)

[4.5 Choice of programming language and framework 27](#_Toc492291630)

[4.6 Testing 29](#_Toc492291631)

[4.7 Test Results 30](#_Toc492291632)

[4.8 Implementation 34](#_Toc492291633)

[4.9 Data Source 41](#_Toc492291634)

[4.10 Evaluation 42](#_Toc492291635)

[CHAPTER FIVE 43](#_Toc492291636)

[DISCUSSION AND CONCLUSION 43](#_Toc492291637)

[5.1 Summary 43](#_Toc492291638)

[5.2 Problems encountered 44](#_Toc492291639)

[5.3 Suggestions for further improvements 44](#_Toc492291640)

[5.4 Recommendation 44](#_Toc492291641)

[5.5 Conclusion 45](#_Toc492291642)

[References 46](#_Toc492291643)

[Appendix 48](#_Toc492291644)

# 

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background of the study

Hospitals are essential part of our lives, providing us with medical facilities. It is necessary for the hospitals to keep track of its daily activities and records of its patients, doctors, nurses, and other staff so that there will be smooth running of the institution. Everyone knows that keeping records using traditional methods can be very challenging, cumbersome, time consuming, prone to errors and inefficient. Especially, when considering the fact the population is increasing and you have so much to keep track of. For this reason, there has to be a better way to handle this problem.

The following challenges that are facing our current hospitals highlighted above prompted this research. Although other researchers have addressed some of the problems mentioned above, this very one addresses a few issues that have not been done much justice to. Hence, this research is titled “Patient Record Management System: Using Bingham University as Case Study”.

## 1.2 Statement of problem

In Bingham University, on several occasions, patients (students) are not being attended to because they have forgotten their ‘clinic card number’. The patient would be forced to manually go through the registration book until he/she finds their record. You can imagine an ailing person being subjected to such a kind of stress before being given attention. It has happened to me sometimes back and sadly, has claimed the life of a friend of mine. On the other hand, the workers cannot help the situation at that moment because according to the rules, they need the card to trace the patient’s folder before administering medication to the patient. Throwing all critical thinking skills out of the window, it is very clear that this is posing a big threat to the students, placing their lives on the line and must be addressed to.

## 1.3 Motivation

With the above problem stated above, the reason for this research is highly justified and worth being given due consideration. We are talking about the lives of the potential leaders of tomorrow being at stake and should not be toyed with. Going outside Bingham University where the population is high, this problem can be by definition, very dangerous to the society. This stands to be the motivation.

## 1.4 Aims and objectives

The aim of this research is to have a working system that will automate almost up to 80% of the manual record keeping techniques that is currently in use in the clinic. This is of course the bottom-line for the whole project, and will greatly reduce the stress the staff undergo while searching for folders and likewise the trauma patients face going through the registration book to get their records.

## 1.5 Purpose of the study

The purpose of this research is summarized below:

1. Automate as many existing manual systems as possible
2. Having a paperless record keeping system
3. Have a reliable database backup in case of natural disasters
4. Ease the stress patients face when they forget their card numbers

## 1.6 Significance of the study

This study is of highly significant to Bingham University and the entire society. Considering the problems it addresses, there is surely no doubt that all my time, resources, energy and effort committed to this work is worth it. The study will contribute to human knowledge in many aspects; one of them is harnessing the power of technology to solve the issue of misplaced clinic cards, which end patients in distress and trauma.

The research is an evidence that what is being taught in class not in vain, because this is an applied knowledge used in solving real life problem. For anybody that values and cherishes humanity, admittedly, it is undoubtedly true that human life comes first in everything. As the slogan says, “health is wealth”. The possibilities of this software is limitless as long as time and space are unbound, because it is going to be constantly improved over time and not be dumped upon completion.

With that, the effort put to this research is very much rewarding, and somebody will one day add more to it. The Bingham community is going to be greatly imparted and of course the society.

## 1.7 Organization of the work

Chapter 1: This contains the an introduction of the whole work. Things like significance of the study and motivation of the research are listed in this section.

Chapter 2: This section contains literature review of relevant work relating to this research topic.

Chapter 3: It contains the system design and methodology used in carrying out this research.

Chapter 4: This section contains the system implementation. It also includes the choice of programming language and reasons for choosing the languages.

Chapter 5: This is the last chapter, It briefly summarizes the whole research, states recommendations and suggested further improvements to this system.

## 1.8 Definition of terms

CFD: – Context Flow Diagram

DFD: – Data Flow Diagram

IDE: – Integrated Development Environment

SRS: – Software Requirement Specification.

GUI: - Graphical User Interface

JavaScript – A web programming language

Node js - A JavaScript environment that allows JavaScript to be run on the server

Express js – A Node js web framework

MongoDB - A No-SQL database

JSON – JavaScript Object Notation

Sublime Text – A text editor

RESTful - Representational State Transfer

API - Application Programming Interface

MVC – Model View Controller

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction

In order to understand the concepts associated with records management and or computer based records management systems, it is imperative to examine and analyze published material from experts in this field. The purpose of this review is to analyze, examine, and obtain experience as regards the creation and archival processing of electronic records. The review is based on an exhaustive assessment of the literature on computerized electronic management and electronic records, and contains an overview of the main concepts associated with the creation of an electronic patient records management system from the perspective of published works by experts. The use of computers in our medical centers today is very crucial. As highlighted earlier, computers greatly reduce stress and cost; at the same time improve performance.

Before then, let us briefly look at how these two words “record and patient” relate because they are going to be used throughout this section

## 2.2 Records

A record is recorded information produced or received in the initiation, conduct or completion of an institutional or individual activity and that comprises content, context and structure sufficient to provide evidence of the activity regardless of the form or medium.

According to the National Archives and Records Administration (NARA) (2011), records include “…all books, papers, maps, photographs, machine-readable materials, or other documentary materials, regardless of physical form or characteristics, made or received ...or in connection with the transaction of public business and preserved or appropriate for preservation by that agency or its legitimate successor as evidence of the organization, functions, policies, decisions, procedures, operations, or other activities of the Government or because of the informational value of the data in them”.

## 2.3 Patient

According to Merriam Webster dictionary, a patient is “an individual awaiting or under medical care and treatment”. From that definition, we can easily correlate between patient and record. Just like any management system, the patient must also provide a set of data in order to keep track of his/her.

## 2.4 Existing System

Many hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information (on forms) is incomplete, or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

Below, we will study the work of some industry experts on the existing system and see some their strengths and weaknesses as well.

## 2.5 Computer use in the medical field

This research clearly states that a patient’s prescribed medications (past and present) can also be stored in a computer system in a hospital. This makes it much easier to transfer any prescriptions and data that a patient needs to local or national drug stores or hospitals. Past hospital visits and billing information can be stored and kept for further use or future hospital experience. For example, doctors keep a computer handy anytime they prescribe a medication. They can use the computer to find out what medicines the patient may be allergic to or what medicines may interfere with one another. Important information is at the tip of their fingers and this can be very helpful. (Amber & Morgan, 2014). We can see how computer based record systems simplify some of the manual tasks being done.

## 2.6 Improving Patient Records

Computer-based patient records and the systems in which they function are becoming an essential technology for health care in part because the information management challenges faced by health care professionals are increasing daily. (Yan, H., Gardner, R., & Baier, R., 2012).

Technological progress makes it possible for hospital to provide total, cost-effective access to more complete, accurate patient care data and to offer improved performance and enhanced functions that can be used to meet those information management challenges. Hospitals can play an important role in improving the quality of patient care and strengthening the scientific basis of clinical practice; they can also contribute to the management and moderation of health care costs. (Don & Elaine , 2011). This research is focused towards data accuracy and minimizing cost and that is a great advantage to the any health organization.

## 2.7 The Computer-Based Patient Record System

This research buttresses the fact that computer-based patient records influence the medical sector in four ways. According to Press National Academy, first, they offer a means of improving both the quality of and access to patient care data. Second, they allow providers to integrate information about patients over time and between settings of care. Third, they make medical knowledge more accessible for use by practitioners when needed. Fourth, they provide decision support to practitioners.

The author also claim that Research efforts could also benefit from computer-based patient record keeping in two key ways. First, improved data and access to those data would be available to researchers. Second, research findings could be communicated to practitioners through computer-based patient record systems (Press, 2012). The research is related to Bingham University because all the points listed above are directly applicable to any health care system.

## 2.8 Electronic health records

According to a study published in the U.S. journal Milbank Quarterly, researchers at the University College of London (UCL) said they identified fundamental and often overlooked tensions in the design and implementation of EHRs (Electronic Health Records). The study was based on findings from hundreds of previous studies from all over the world. They said that their findings have implications for President Barack Obama’s election promise to establish electronic health records for every American by 2014, and for other large-scale EHR initiatives around the world.

Professor Trish Greenhalgh, lead author of UCL’s Department of Open Learning, said that EHRs are often depicted as the cornerstone of a modern healthcare, capable of making care better, safer and cheaper. Yet, clinicians and managers the world over struggle to implement EHRs. (Manos, 2009). Manos also claim that between 50 and 80 per cent of electronic health record projects fail – and the larger the project, the more likely it is to fail. Taking a closer look at the challenges such systems face, especially large systems, we can see that implementing a standard HER requires much effort which in turn seems to be one of the limitations of EHRs.

## 2.9 How developments in technology and data in the NHS are improving outcomes for patients

Technology has drastically improved the outcomes for patients. This technology includes both record systems and sophisticated treatment techniques. “In other parts of our lives we take technology and its benefits for granted. We are able to book our holidays and manage our finances online. But although 61% of UK adults have a smartphone and 76% of us access the internet every day, only 2% of the population have any kind of digital interaction with the NHS. This needs to change, not only for patient care, but also to improve the efficiency of the NHS, and in order to support the integration with social care services” (Sir Jeremy, 2014).

It is very important that organizations across health and social care have made further commitments in the area of data and technology, with the publication of Personalized Health and Care 2020. This sets out a framework for how the health and care sector will use data and technology to transform outcomes for patients and citizens (Marcus, 2011).

## 2.10 Federal Health Architecture

According to the Federal Health Architecture, CONNECT open source software enables health IT systems to securely communicate via nationally recognized standards. With CONNECT, information can be shared securely via the Internet among doctor’s offices, federal agencies, state agencies, disability organizations, public health organizations, pharmacies and other health stakeholders. This enables health professionals to request, send and receive medical records so critical information can follow patients as they navigate through the health care system.

CONNECT was developed by more than 20 federal agencies working together through FHA. Rather than each federal agency independently building its own separate Nationwide Health Information Network compliant gateway, they decided to collaborate and create a single platform to enable the secure exchange of health information (pub, 2013). This system enhances security and reliability in a higher degree. It also makes it easier to maintain because it has a central data source.

## 2.11 Patient Engagement within the Patient-Centered Medical Home Model

This research focused on what best practices might exist among patient-centered primary care delivery to highly-activate their patients. The problem that was studied was what specific aspects of primary care delivery, such as physician leadership, staffing models, staff training and development, and electronic tools and resources, support or undermines the successful high-level activation of patients. The research method used was a qualitative study using the Delphi Method of collecting subject matter expert (SME) opinions from among four classifications of primary care practice team members as study participants (Constantz, 2017).

## 2.12 All Systems Go: The Voice of Nurse Practitioners Using Electronic Patient Records

Electronic patient record systems have become an integral component of the Canadian healthcare system, introduced to improve the delivery, coordination, and quality of patient care. This vital technology is a cost-effective order entry method for healthcare providers designed to decrease redundancy in the system, facilitate workflow, improve clinical decision-making, and decrease medication errors

The research also indicates that electronic patient record system, referred to as HUGO (Healthcare Under Going Optimization), was implemented throughout hospitals in Southwestern Ontario in 2014. HUGO was designed with the traditional hierarchical approach to care in which the physician enters an order for other care providers to follow. Several aspects of the traditional approach of HUGO that make it valuable for physicians, pose challenges for NPs as autonomous prescribing providers (Amy , Ashley , & Christina , 2016).

# CHAPTER THREE

# SYSTEM DESIGN AND METHODOLOGY

## 3.1 Introduction

The system developed is going to accommodate changes along the way. Improvements are going to be constantly made. Out of the numerous software development methodologies, the researcher chose to use agile methodology.

Waterfall model

Iterative model

Agile model

## 3.2 Software Development Cycle

What is Waterfall model?

The Waterfall Model was the first model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is used for projects that are small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model, software-testing starts only after the development is complete also, phases do not overlap (ISTQBExamCertification, 2015).

When to use the waterfall model

This model is used only when the requirements are very well known, clear and fixed.

Product definition is stable.

Technology is understood.

There are no ambiguous requirements

Ample resources with required expertise are available freely

The project is short.

Advantages of waterfall model

This model is simple and easy to understand and use.

It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.

In this model, phases are processed and completed one at a time. Phases do not overlap.

Waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages of waterfall model

Once an application is in the testing stage, it is very difficult to go back and change something that was not well thought out in the concept stage.

No working software is produced until late during the life cycle.

High amounts of risk and uncertainty.

Not a good model for complex and object-oriented projects.

Poor model for long and ongoing projects.

Not suitable for the projects where requirements are at a moderate to high risk of changing.

(ISTQBExamCertification, 2015)

Iterative model

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. When discussing the iterative method, the concept of incremental development can also be used liberally and interchangeably, which describes the incremental alterations made during the design and implementation of each new iteration (Powell-Morse, 2015).

Advantages of the Iterative Model

**Inherent Versioning**

**Rapid Turnaround**

**Suited for Agile Organizations**

**Easy Adaptability**

Disadvantages of Iterative Model

**Costly Late-Stage Issues**

**Increased Pressure on User Engagement**

**Feature Creep**

(ISTQBExamCertification, 2015)

What is agile model?

Agile software development methodology is a process for developing software (like other software development methodologies – Waterfall model, V-Model, Iterative model etc.). In English, Agile means ‘ability to move quickly and easily’ and responding swiftly to change – this is a key aspect of agile software development as well (Wikipedia, 2017).

Brief overview of Agile Methodology

In traditional software development methodologies like Waterfall model, a project can take several months or years to complete and the customer may not get to see the product until the completion of the project (Johnny Moore, 2014).

At a high level, non-Agile projects allocate extensive periods for Requirements gathering, design, development, testing and User Acceptance Testing, before finally deploying the project.

In contrast to this, agile projects have Sprints or iterations, which are shorter in duration (Sprints/iterations can vary from 2 weeks to 2 months) during which pre-determined features are developed and delivered.

Agile projects can have one or more iterations and deliver the complete product at the end of the final iteration.

Characteristics of agile development

Modularity: When writing the code, the design pattern used is called MVC; this pattern makes it possible to write modular. It solves separating presentation layer and business logic.

Iterative: The code underwent constant iterations in order to make changes. These iterations included refactoring and optimizations.

Time-bound: Following a deadline for the submission, the project can be said to be time-bound. It was structured into milestones.

Incremental: During the iterative stage, more features were added to make the software better.

Collaborative: When developing this software, this characteristic was not included because the researcher did not collaborate with anybody. Everything was designed from ground up.

Advantages of agile methodology

In agile methodology, the delivery of software is unremitting.

The customers are satisfied because after every Sprint working feature of the software is delivered to them.

Customers can have a look of the working feature, which fulfilled their expectations.

If the customers has any feedback or any change in the feature then it can be accommodated in the current release of the product.

The daily interactions are required between the business people and the developers.

In this methodology, attention is paid to the good design of the product.

Changes in the requirements are accepted even in the later stages of the development.

Disadvantages of agile methodology

In agile methodology, the documentation is less.

Sometimes in agile methodology, the requirement is not very clear hence, it is difficult to predict the expected result.

In few of the projects at the starting of the software development life cycle, it is difficult to estimate the actual effort required.

The projects following the agile methodology may have to face some unknown risks, which can affect the development of the project (Powell-Morse, 2015).

The Agile Process Flow

Concept – At this stage, the researcher envisioned and prioritized the whole idea of the patient record system based on some things he observed.

Inception – The researcher identified and figured out the initial environments and requirements of the system. This is a very important stage because wrong requirements

Iteration/Construction – In this stage, the researcher constructed the software based on the laid requirements. Iterations were carried out to ensure that code becomes fully optimized.

Release - The researcher carried out QA (Quality Assurance) testing, internal and external testing, documentation development, and final release of the iteration into production.

Production – The software has not yet been given out into production. It will be done when the clinic

Retirement – This is supposed to be the end of activities of the whole system but it has not been done because the software itself has not been deployed into production.

Why Agile?

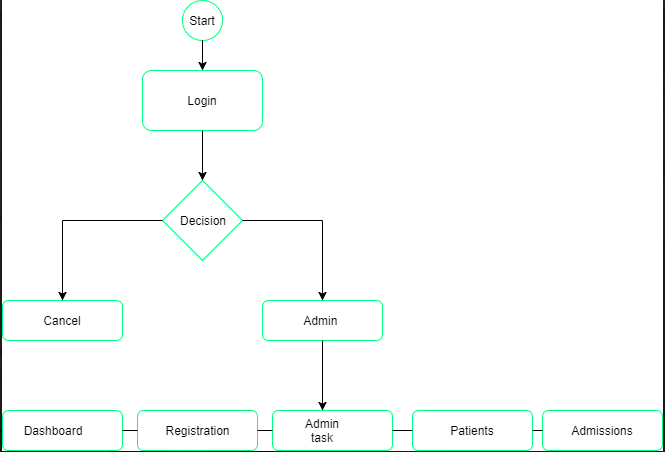
Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” As earlier stated, this software is going to encounter a lot of iterations and changes and hence agile is best suited for our need (The Agile Movement, 2008).

## 3.3 Design phase using unified modelling language

Modeling is the designing of software applications before coding. Modeling is an essential part of large software projects, and helpful to medium and even small projects as well. A model plays the analogous role in software development that blueprints and other plans (site maps, elevations, physical models) play in the building of a skyscraper. Using a model, those responsible for a software development project's success can assure themselves that business functionality is complete and correct, end-user needs are met, and program design supports requirements for scalability, robustness, security, extendibility, and other characteristics, before implementation in code renders changes difficult and expensive to make. The researcher made models that served as guide throughout the development process.

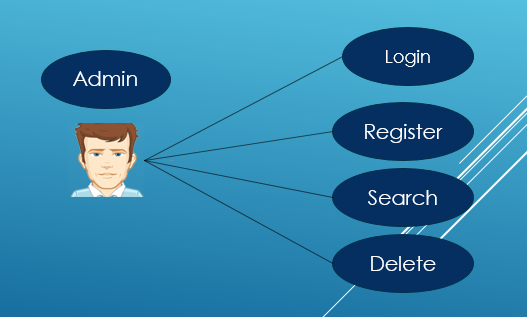
## 3.4 Flowcharts

This is the diagrammatic representation of the application’s functionalities from start to finish. It indicates actions to be taken based on some predefined conditions in the system.



*Fig 3.1* System Flowchart

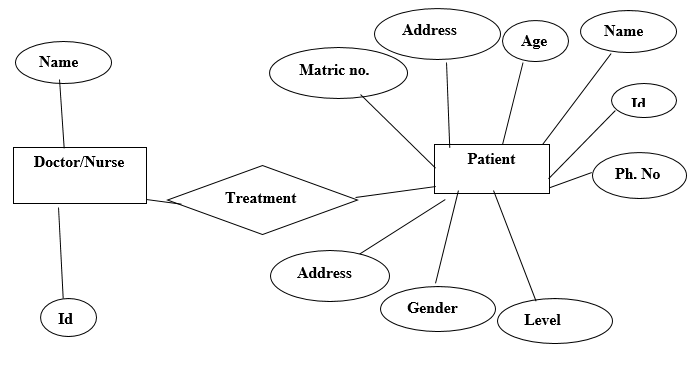
## 3.5 Use case diagram



*Fig 3.2* Use case diagram

## 3.6 Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. (Smartdraw, 2014).



*Fig 3.3* ER diagram

## 3.7 Data Flow Diagrams (DFD)

The Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system. Data flow diagrams are used by systems analysts to design information-processing systems but also as a way to model whole organizations. The main merit of DFD is that it can provide an overview of what data a system would processes, what transformations of data are done, what data are stored and which stored data is used, and where the result is flow.

## 3.8 Level 0 DFD

A context diagram is a top level (also known as Level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities. In level 0 DFD, system is shown as one process.

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system (Smartdraw, 2014).

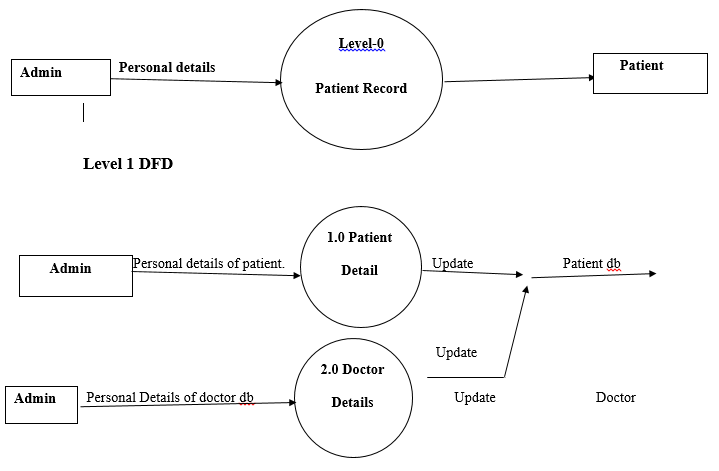


Fig 3.4 Data flow diagram

## 3.9 Non-Functional Requirement

Reliability

This application is a reliable product that produces fast and verified output of all its processes.

Availability

This application will be available to use and help them to carry out their operations conveniently.

Security

The application will be password protected. User will have to enter correct username, password and role in order to access the application.

Maintainability

The application is designed in such a way that it can easily be maintained. It will be easy to incorporate new requirements and featured in the individual modules.

Portability

The application will be easily portable on device that has a web browser installed on it. It is a cross-platform application.

## 3.10 Functional Requirements

The admin can login to the system

The admin can register patients

The admin can delete patients

The admin can search for patients using their names, age, level and id

The admin can modify the patients’ records

# CHAPTER 4

# IMPLEMENTATION AND EVALUATION

## 4.1 Introduction

After observing this critical need of a patient record in Bingham University, the researcher came up with the idea of a system that will stand in between this gap. This involves making the system user friendly, hence easy to use. The system was designed to make it possible for easy scaling as patients increase in the clinic. The system will also provide room for modifications as requirements change. This was achieved because of the right choice of programming platform and languages. Different frameworks and languages were used to achieve this.

## 4.2 Hardware requirement

The system requires the following set of hardware to run as expected:

Processor: Intel Dual CoreTM or above

Processor Speed: 2.0 GHz or above

RAM: 2 GB or above

Hard Disk: 50 GB or above

## 4.3 Software requirement

Windows 7 or above and MacOS Mavericks or above for OSX users

Node js server

MongoDB database

Web browser (preferably Google Chrome browser)

## 4.4 Programming platform

The application was built using Node js. It is an event-driven, asynchronous, non-blocking server side runtime sitting on top of Google Chrome V8 runtime. It is known to be one of the fastest server side environments. A web server was created using Node js and it was blazing fast. The researcher also used Sublime Text, a text editor for coding. RESTful API service was implemented for communicating between the web server and the client. All the communications between the server and the client were done in form of JSON. The researcher also used MongoDB for the database. Then finally, used HTML, CSS and JavaScript for the rendering on the client side and other logic. This is obvious because they are the backbone of any web application.

## 4.5 Choice of programming language and framework

JAVASCRIPT

Whenever you are talking about the web, it is not possible not mention JavaScript. This programming language has eaten the web in all aspects ranging from small, medium, and enterprise web applications. Every web browser has JavaScript language installed on it. Coupled to the fact that the language is easy to start working with and its ubiquity, the researcher quickly jumped into it and started developing.

Back into history, Brendan Eich developed the language in 10 days during a hackathon at Netscape in 1995. Since then, many programmers because of its quirks had rejected the language.

In 2009, those programmers did not have option than to embrace it. The same language was made to be run on the server, serve as the server itself and handle many concurrencies. This has solved many issues faced by the traditional web servers.

Another reason the application was written in JavaScript is that the developer will have to use only one language across the stack for development. It is serving as the client side language, server side language and database as well. There is no need of changing context when building the application.

The developer used Express js, an MVC JavaScript server side framework and Angular js, a client side MVC framework. The combination of this stack is coined as “MEAN” stack (MongoDB, Express js Angular js, Node js). . Everything is JavaScript. There is a slogan for the language, “One language rule them all”.

MONGODB

A NoSQL (originally referring to "non SQL" or "non-relational") database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases (Wikipedia, 2017).

NoSQL databases are built to allow the insertion of data without a predefined schema. That makes it easy to make significant application changes in real-time, without worrying about service interruptions – which means development is faster, code integration is more reliable, and less database administrator time is needed. (MongoDB, 2017).

Considering the kind of database being used, there is no predefined schema for this application; hence, the interactions between all the data entities is highly flexible.

The Benefits of NoSQL

When compared to relational databases, NoSQL databases are more scalable and provide superior performance, and their data model addresses several issues that the relational model is not designed to address:

Large volumes of rapidly changing structured, semi-structured, and unstructured data

Agile sprints, quick schema iteration, and frequent code pushes

Object-oriented programming that is easy to use and flexible

Geographically distributed scale-out architecture instead of expensive, monolithic architecture

Other features of the NoSQL databases include auto-sharding, load-balancing, replication, integrated caching and indexing. (MongoDB, 2017)

Haven listed all these features of NoSQL databases; also, being written in C/C++/JavaScript, it is obvious that it is not a bad choice.

## 4.6 Testing

Software testing is a process of executing a program or application with the intent of finding the software bugs. Two kinds of tests were carried out namely development and user testing.

Development test

Development testing: this was carried out during the development of the software. It is done so as to make sure that there are no bugs before deployment into production environment. The researcher tested for many things in the application. Below are the different things tested during the development process.

The researcher tested for the database connectivity before the application even starts running so that it can keep track of all “CRUD” operations in the system.

All RESTful API endpoints were tested to make sure that they are serving the required resources.

Broken links were tested also

Environment variables were tested to make sure they are properly loaded.

The resaercher tested for available ports on the machine

User test

User testing is the process of giving the application to the user of the software to check for any possible bugs. In this case, the researcher did not give the clinic officials but instead gave it to some people around him to use the application and give their feedback, which they did. Below are the different things they tested for.

They tested for user UI/UX (User Interface/ User Experience).

Simple navigation

Speed of loading pages

Security

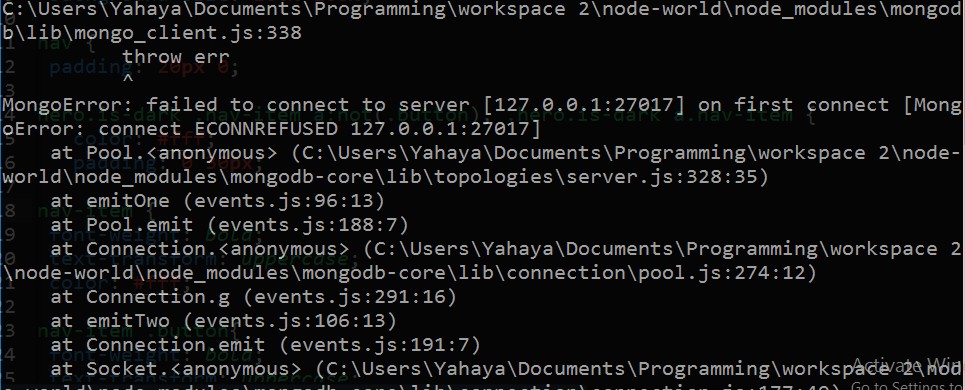
Hardware requirement

## 4.7 Test Results

The researcher got some output after the test processes. Some parts of the application were working as intended but some were not. The results from these tests made it possible to improve some of the faulty functionalities of the application. Though it took the researcher couple of times to fix those issues, the bugs were finally addressed to, to be honest, not all. The user interface needs more design.

Development test result

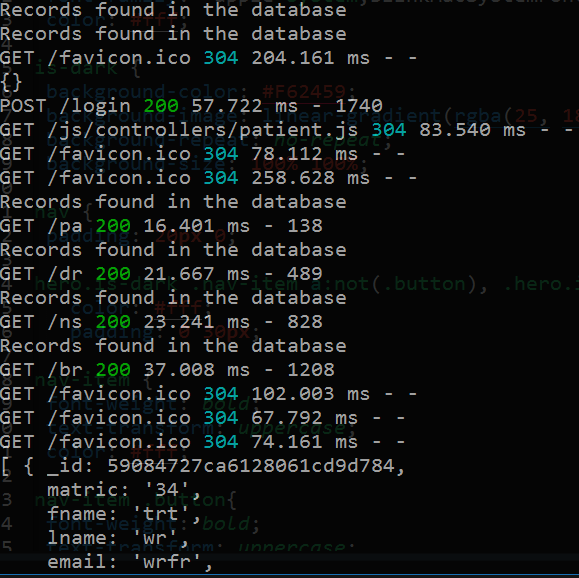
1. There was a persisting issue of not connecting to the database when the application starts.



*Fig 4.1*  Database error

Solution: An ORM (Object Relational Mapper) called MongoJS was used to handle the connection logic.

1. There were no API endpoint errors because the browser was served with status code of 200, meaning all endpoints were working correctly.

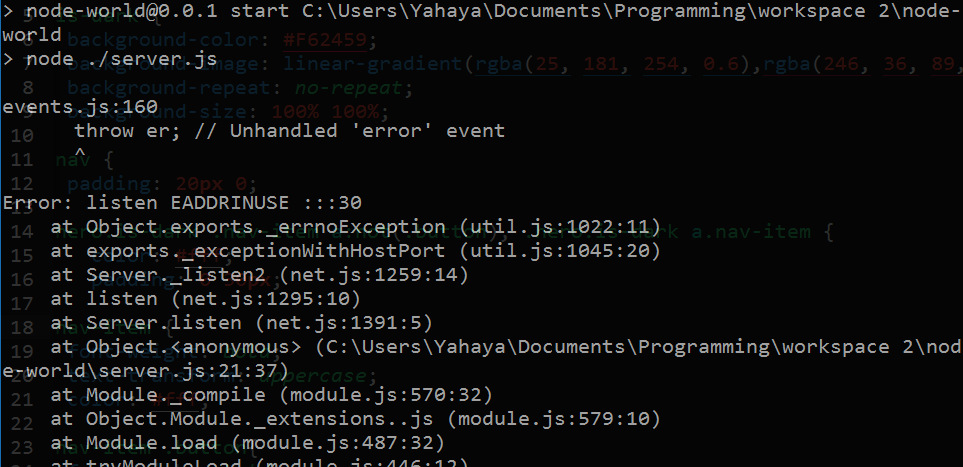


*Fig. 4.2* Web server 200 status success response code

1. The researcher did not encounter any broken links throughout the application development.
2. There was an error, which made the browser not to load all the necessary environment variables, examples is the session secret. There was no way to access the machine’s runtime.

Solution: The researcher downloaded a module from npm (Node Package Manager) called ‘dotenv’ and it solved that issue.

1. The last error encountered in development test was using a port more than once to start the server.



*Fig. 4.3*  Server port error

Solution: Only one server terminal window was open and in use to avoid the error.

User test result

1. After testing for the UX/UI, they really complained that the interface was not looking nice and that it should be improved a bit.

Solution: A CSS framework called Bulma to enhance the design, but there is more work to be done to improve the design.

1. The users suggested that the navigation used was too complex; they recommended making it simpler.

Solution: The navigation was simplified; it also made the application to be an SPA (Single Page Application). Other parts require the same feature, but could not be implemented.

1. The speed of page loads and resource serving was awesome according to them, they did not spot any bug with that.
2. Security features have not been fully implemented since login credentials are stored in session memory. However, the credentials entered must match with the ones stored on the server before anyone access the application.
3. The hardware requirement specified was okay and did not show any sign of lag or slow performance, so it the application has passed the hardware test.

## 4.8 Implementation

This is the stage where the actual application comes into life. After using several platforms and technologies, the researcher ended up with the workable version of the application. The key part players of this application are:

Jade: A HTML preprocessor that compiles to HTML. It is easy to work with because of its clear syntax. A functional templating language that uses mixins, embedded JavaScript and so many things.

CSS: As earlier mentioned, this is the heart of any web application design. It gives a visual sense of appeal to the user. It is responsible for beautifying the whole application and makes everything looks attractive and presentable.

JavaScript (Client side): The heartbeat of the whole web application. It is responsible for the application’s logic and makes everything happen.

Node js (Server side): An event-driven, asynchronous, non-blocking server side runtime built on top of Google Chrome V8 engine. This is the server itself. It is so powerful that you write the code to spin up your own webserver. It is really low level and flexible.

Express js: It is a web framework sitting on top of Node js to handle all the native Node API interactions on your webserver. It makes things such as routing and authentication a bit easier when compared to the native Node APIs.

MongoDB: The application has a sort of persistence to data storage. MongoDB handles that aspect. It stores data in a BSON format. It makes it easy to interact with all the API endpoints because you are not parsing any data when you receive it. You just store it directly in the database.

Login

The application has a single point of access. As stated earlier, there is no much security implementation to this system since the credentials are all stored in-memory storage. That is one of the limitations of the system that be addressed in the next version because the project is open source and a work in progress. The admin enters a username and a password, the password is sent to the server in JSON format, the compared to the login credentials stored on the server and then grant access to the user if they match up and returns a 403 unauthorized status code if they do not. I used the simplest login implementation in the application.

Dashboard

On successful login, the admin is taken to the admin dashboard. The dashboard contains summary of the whole application data. Things such as the number of patients in the clinic, number of staff, number of death and birth recorded. It also has a graphical representation of all these information for easy understanding and to track all the metrics.

Admin tasks

The tasks is where all the entries regarding the admitted patients, birth rate and death rates are being entered. In provides all the input fields that are required to make these entries and sent to the server for data persistence.

Registration

This component is where the admin registers the patient into the system. Information such as the patient’s name, age, blood group, state, address, etc. are all collected in the provided input fields. Upon successful registration, the patient’s information can then be accessed in the patients’ page.

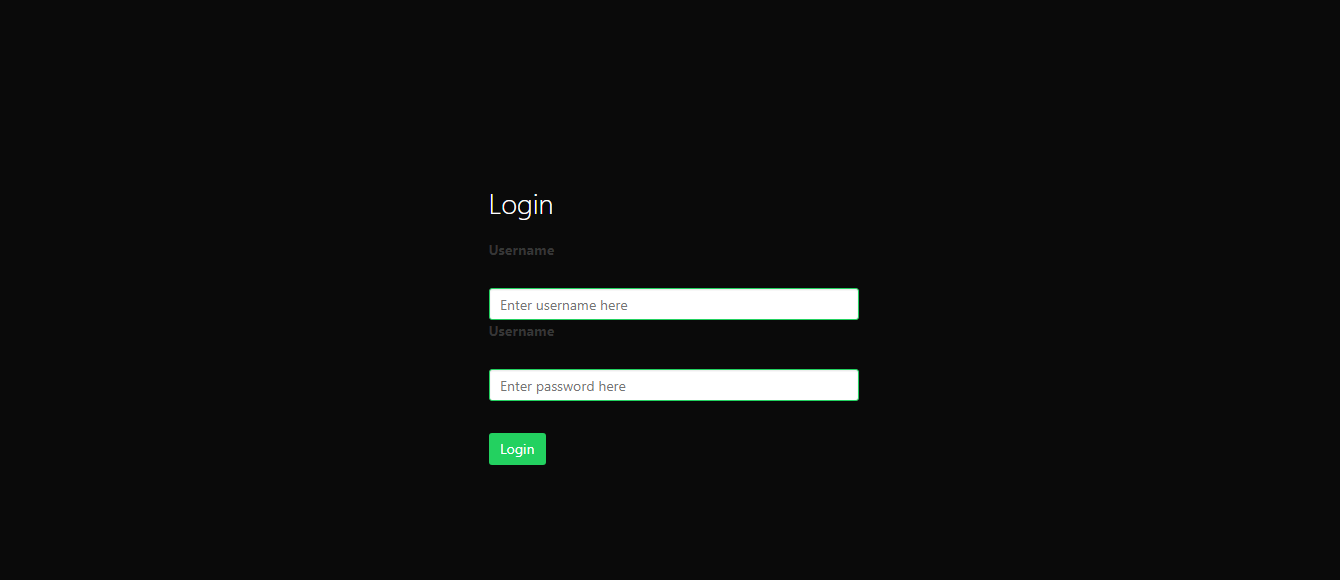
Patients

This section contains all the registered patients in the system. The admin can view all the registered patients and can delete them. There is a search area where the admin can search for a patient. The search query can accept any input and then returns a result of the patient(s) matching that query. The search algorithm uses a filter technique to sort the result based on the input. The admin can then select the patient and view the patient’s information. The admin can also add treatment details administered to the patient and it keeps track of the admin that administered the treatment and the date and time of the treatment.

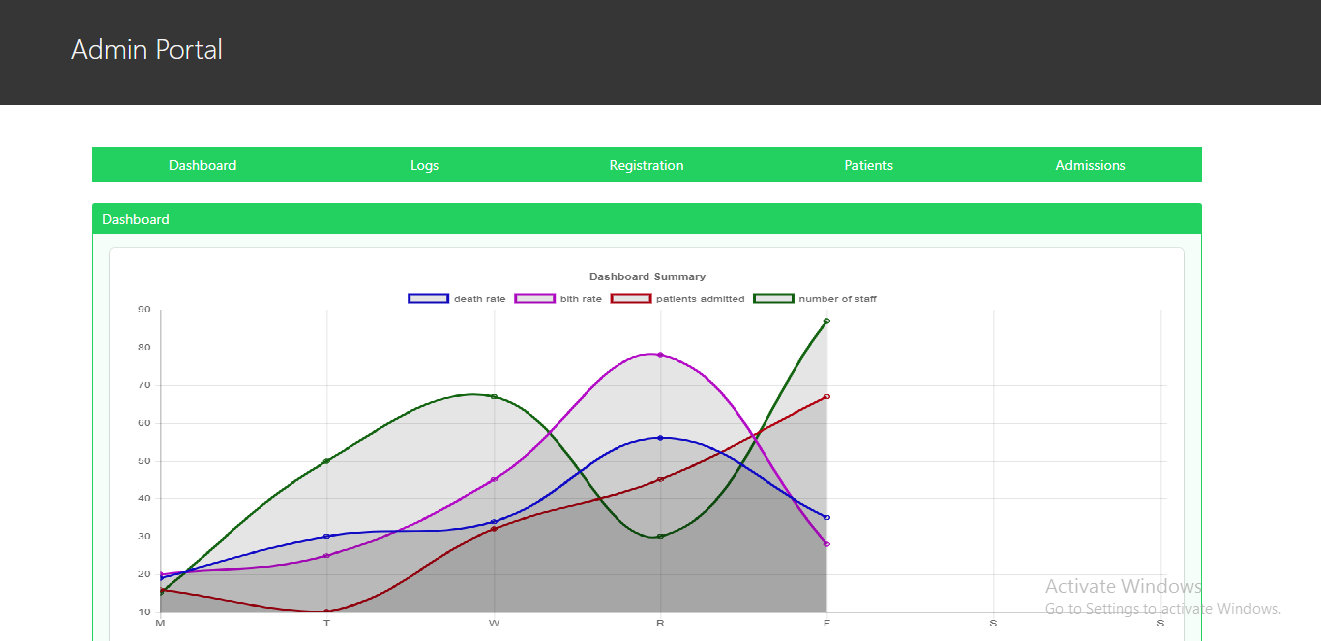
Admissions

This is the part of the application is responsible for keeping track of all admitted patients in the clinic. It displays all the list of admitted patients and gives the option to discharge the patient whenever their treatment is over.

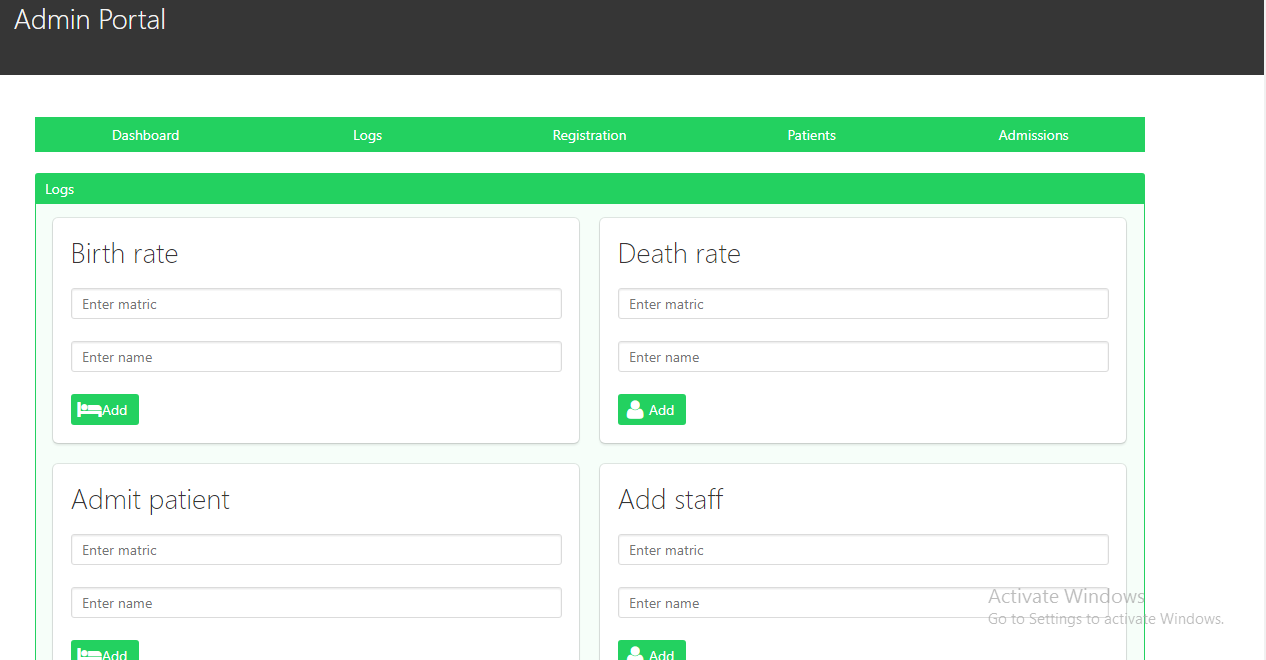
Screenshots of the application



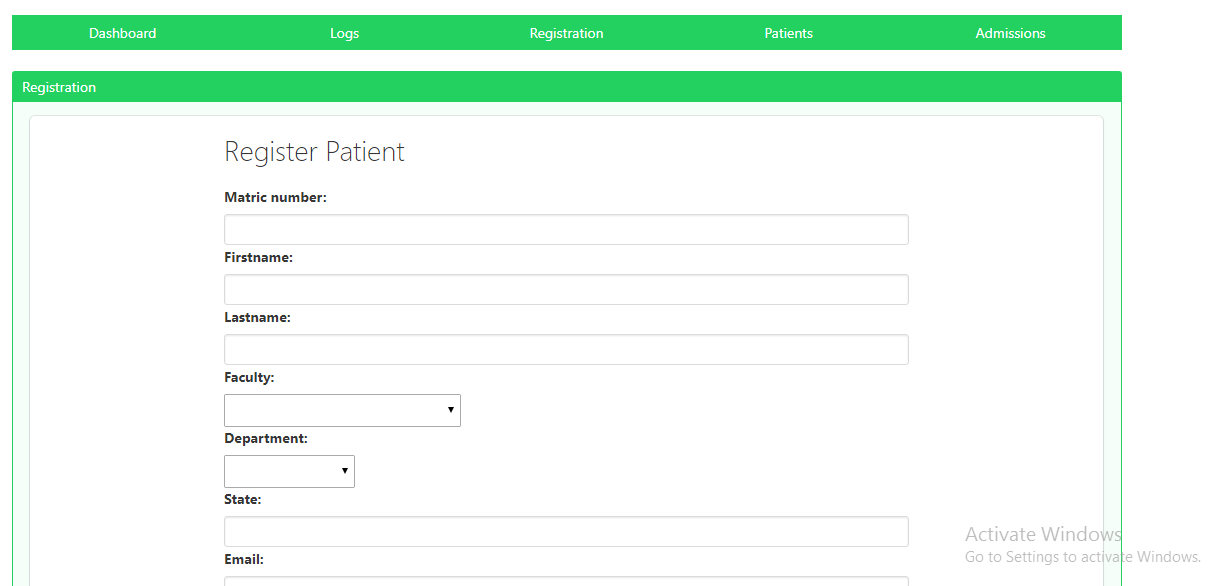
*Fig 4.4* Login screen



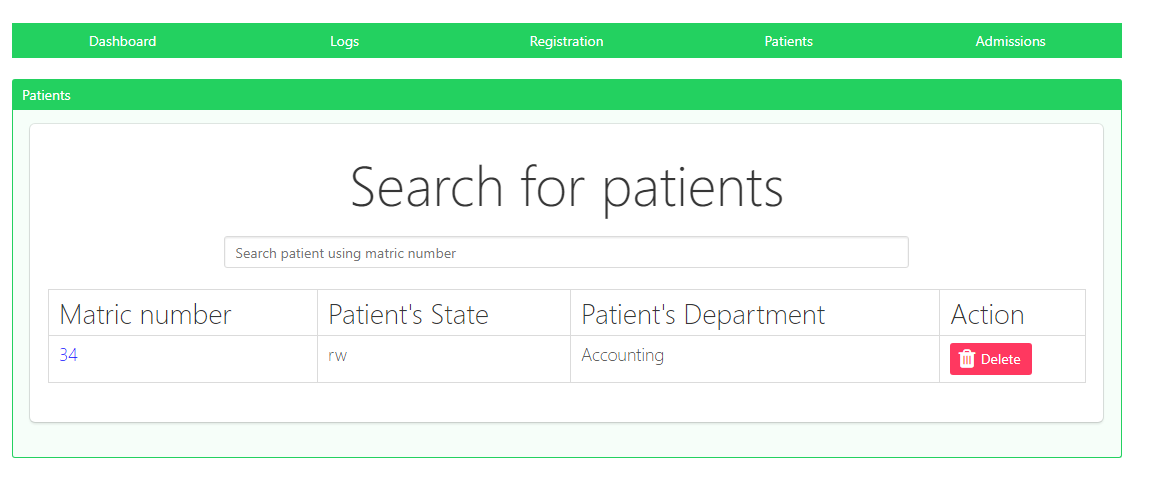
*Fig. 4.5* Admin dashboard



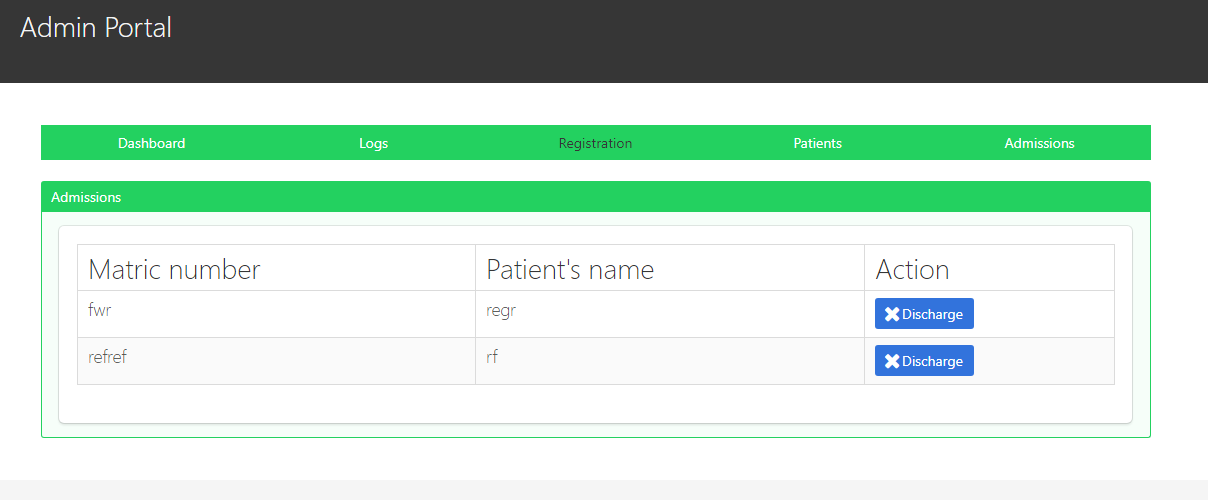
*Fig 4.6* Admin tasks*.*



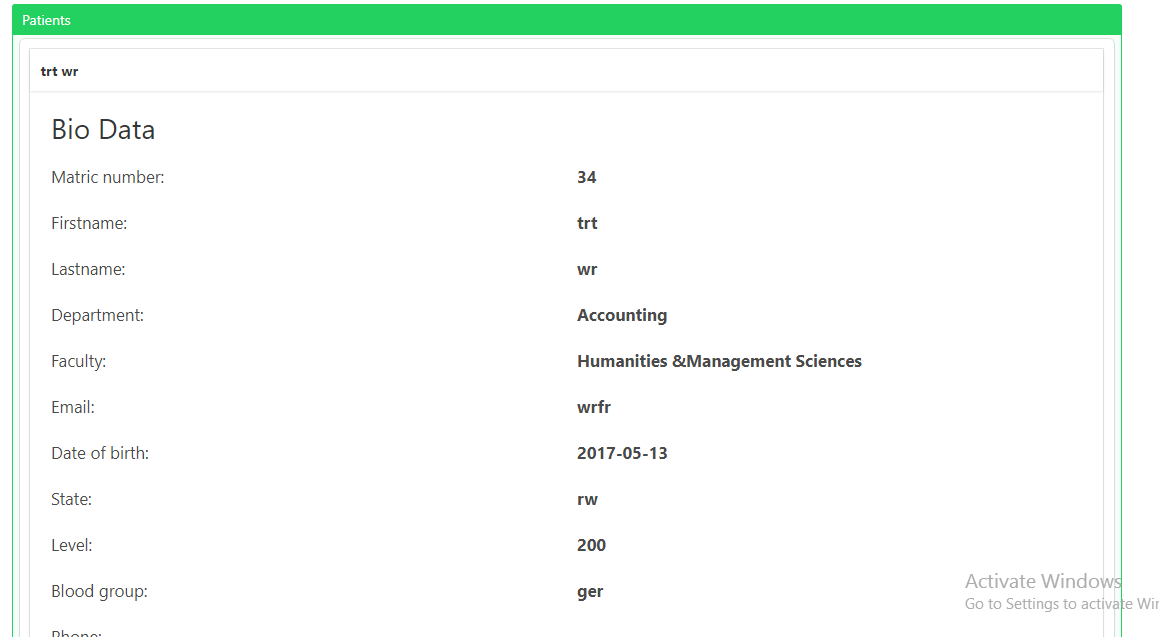
*Fig. 4.7* Patient registration



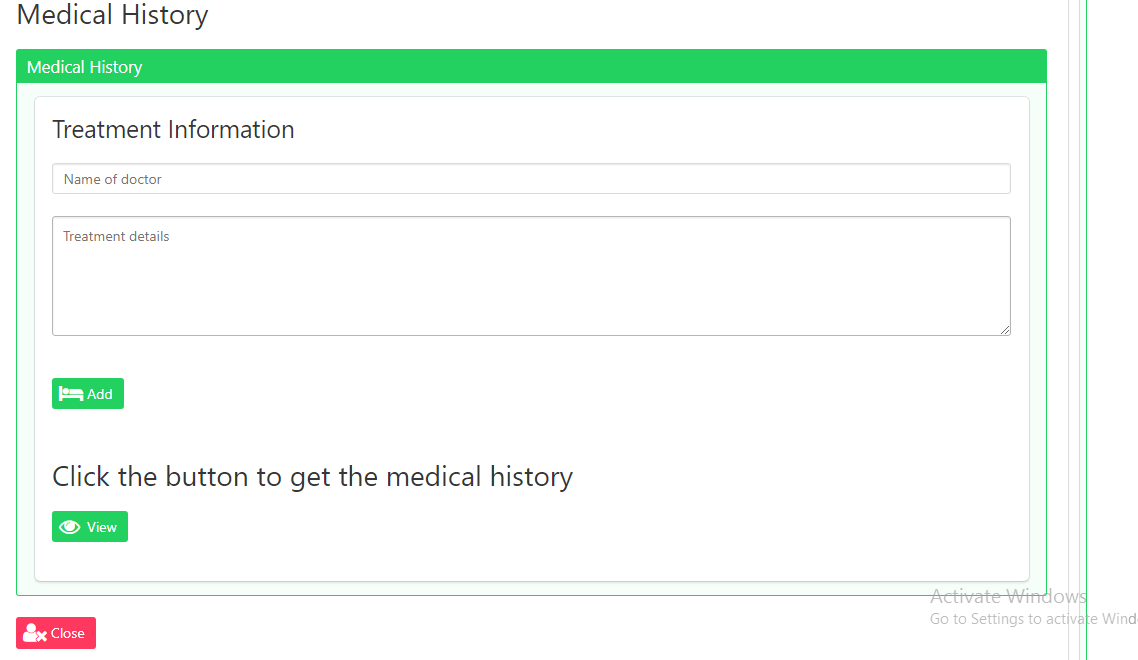
*Fig. 4.8* List of patients



*Fig. 4.9* Admitted patients



*Fig. 4.10* Patient profile



*Fig. 4.11* Patient medical history

## 4.9 Data Source

The data used for this research was obtained from oral interview and observation. Below are the questions asked by the researcher to obtain the data from one of the clinic staff.

Question: How do you keep patient records?

Answer: We manually record the data of every patient in our register.

Question: How efficient is that method?

Answer: Well, it is not the best method but it has been a norm and tradition for us here.

Question: Do you think that computers can help in some aspects such as recording patients’ records?

Answer: Of course, as I said, our manual system is not the best method and hence computer can solve some of the challenges with the manual method.

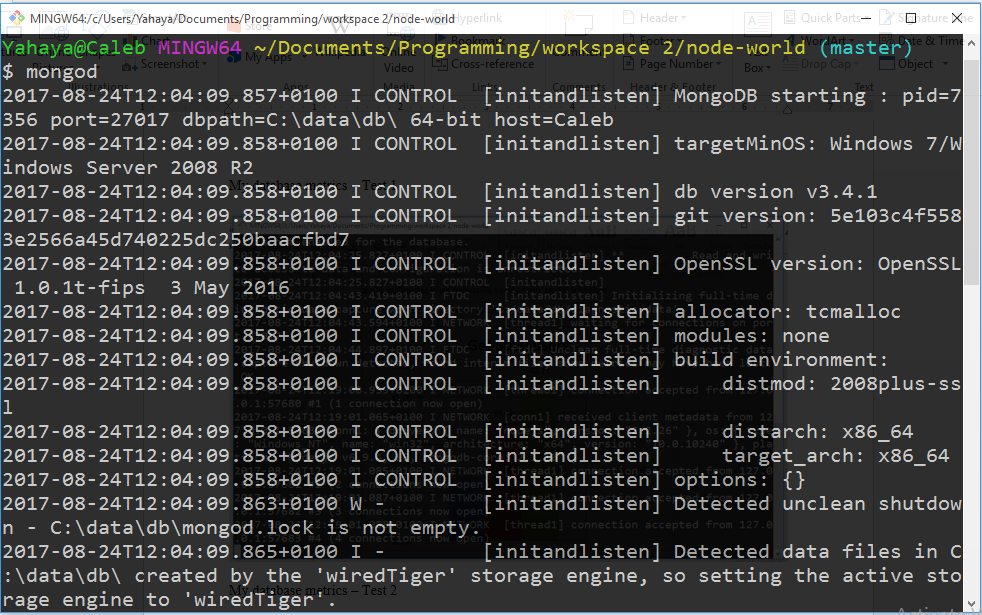
Question: You just mentioned challenges, what are the challenges you face with the current system with respect to the patients’ record?

Answer: You see, often, students insult us that we do not attend to a sick person when the person is without clinic card. There is no way you can treat a person without a card number, so this is a challenge to us. The other issues is the fact that when the registers starts wearing, we can hardly see the complete patients’ record when the need arises.

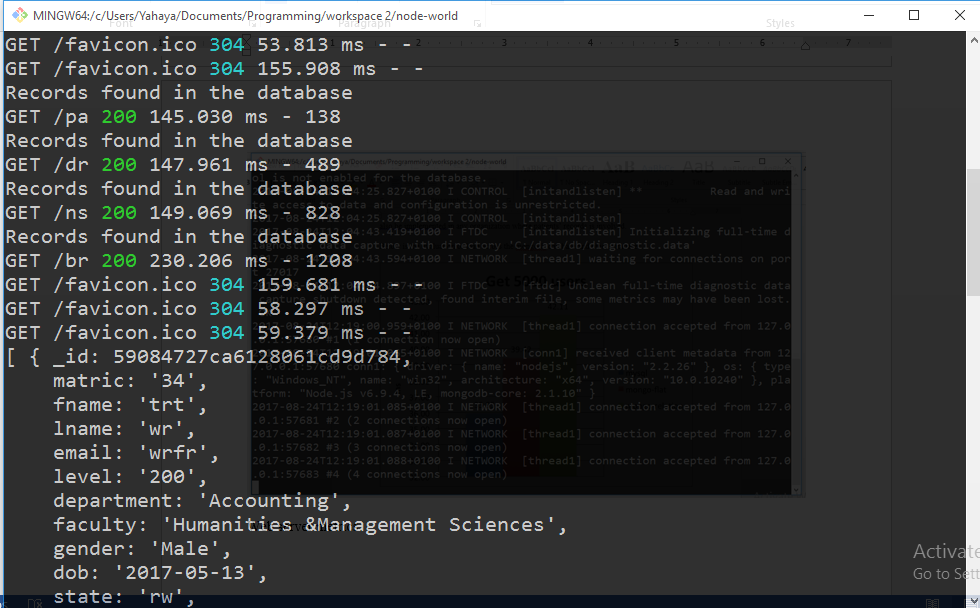
The results obtained from this interview and observations made add up to give sufficient data to carry out this research.

## 4.10 Evaluation

Here, the researcher evaluated his own work and then compared it with other similar stacks e.g the LAMP stack (Linux Apache MySQL PHP). The performance metrics on the Node server and the MongoDB server is by far better than those ones. Below is the evaluation of the other stacks.



*Fig. 4.12* Database metrics – Test



*Fig. 4.13* Web server metrics – Test

It is obvious that the maximum time to serve any given resource is less than 60 millisecond and the database is still running on one thread. This is extremely fast. Although it can be improved over time when there is much more load or concurrencies.

# CHAPTER FIVE

# DISCUSSION AND CONCLUSION

## 5.1 Summary

When it comes to the issue of scaling, this system can scale very well, in the sense that whenever there is an increase in the number of users in the application, it can accommodate such load to a reasonable level. The system was designed to help people to reduce manual labor and help them automate the manual processes in the clinic. This will go a long way in improving our modern hospitals. It can also be implemented in any organization where patients’ records are involved.

## 5.2 Problems encountered

The major problem faced while carrying out this project was combining learning how to code and immediately applying the concepts to the application. This is indeed a challenge the researcher faced. It actually slowed down the whole process and reduced the net productivity.

## 5.3 Suggestions for further improvements

This system is still in its beta stage and it is open source. I highly recommend going through the source code and improve this piece of software. Below is the repository of the project on GitHub. Feel free to clone the software and make your own changes to it.

https://github.com/microworlds/node-world

The aspect of hospital management system is very broad; the researcher solely focused on the patient record aspect, which includes search indexing, admin tasks and basic CRUD operations. My recommendation to anybody improving on this system is to focus on handling referring patients to wards, checking for space availability and booking an appointment with the doctor, and then each user should access his or her own profile on a remote server. When all these are completed and fully integrated into the system, then the application is going to be super functional.

## 5.4 Recommendation

The researcher highly recommends that the school should put more effort in making sure that students learn programming languages that would be applied in their daily lives to solve real life problems and not to only focus on the course outlines alone. Programming events such as meetups and hackathons should be organized to broaden the horizons of students and change their mindsets towards programming and real life problem solving techniques.

## 5.5 Conclusion

Lessons have been learnt while making this system come to reality. This has given me a subtle knowledge of how the medical record systems work and have given me more ideas on how to improve these problems with my new found coding skills and not just writing a 1000 page essay trying to deliver a speech of how to solve them. The system will really help automate daily manual tasks in the hospital and hence reducing the rate at which patients wait to search for their id number or clinic card numbers before being attended to. It will also provide a backup copy for all the data in the system in case of any natural disaster or hazard. For sure, this is not the end of it because the researcher already more zealous than ever to venture into the medical field and continue solving real life problems.

# References

Abele, D., Vincenzo, G., & Simona, P., Paola (2008). *The impact of a clinical information system in an intensive care unit*. Journal of Clinical Monitoring and Computing 22(1), 31-36.

Amber Weber and Morgan Swink, (2014). *Computer use in the medical field*. Amber Weber

Bosch, J. (2014). *Electronic medical records*: My journey. The Journal for Nurse

Don & Elaine, (1997). *Improving Patients Record.*

Kijsanayotina, B., Pannarunothai, S., & Speedie, S. M. (2009). *Factors in?uencing health information technology adoption in Thailand’s community health centers:Applying the UTAUT model.* International journal of medical informatics 78, 404-416.

Kloss Linda (2012) “Information Management and Governance: *Essential Health Data Integrity Practices.”*, Chicago, IL.

Narcisse, M., Kippenbrock, T. A., Odell, E., & Buron, B. (2013). *Advance practice nurses’ meaningful use of electronic health records*. Applied Nursing Research, 26, 127-132. doi:10.1016/j.apnr.2013.02.003

Thurston, J. (2014). Meaningful use of electronic health records. The Journal or Nurse Practitioners, 10, 510-513. doi:10.1016/j.nurpra.2014.05.012

Victoria, H., & Kannan, S. (Eds.). (2008) *Healthcare Information Systems* (Vols. 1). United States of America: Information Science Reference.

Victoria, H., & Kannan, S. (Eds.). (2008) *Healthcare Information Systems (Vols. 1)*. United States of America: Information Science Reference.

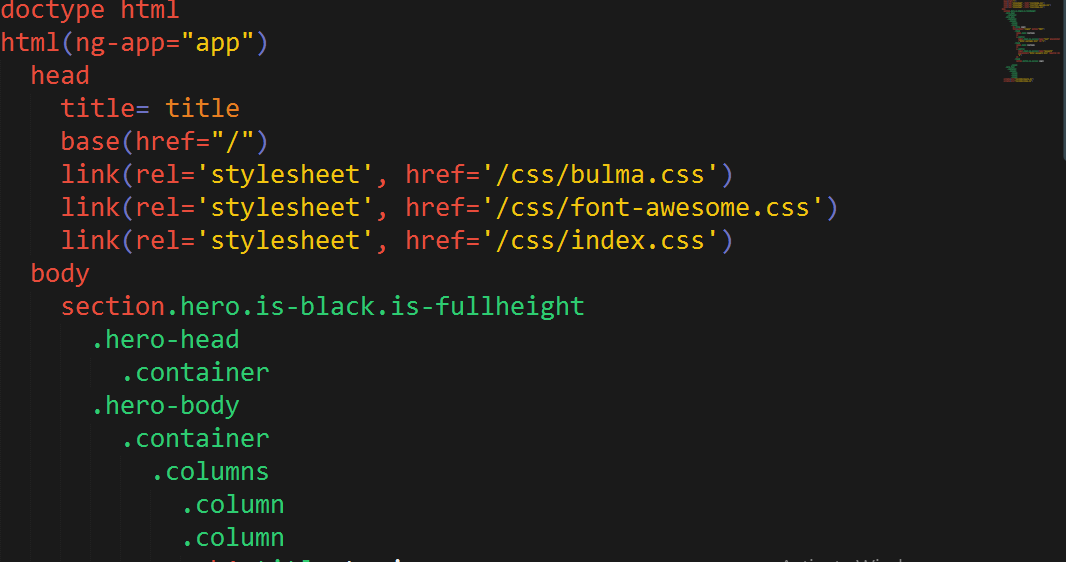
Yan, H., Gardner, R., & Baier, R. (2012). *Beyond the focus group*: Understanding physicians’ barriers to electronic medical records. The Joint Commission Journal on Quality and Patient Safety, 38, 184-191.

Yong, W., Huadeng, W., & Xiliang, T. (2009). *On the Reliability of Information*. Paper presented at the Control and Decision Conference.

# 

# Appendix

Source code for homepage



Source code for home client route



Source code for app middlewares



Source code for database congfiguration



This is a work on progress; check this url to download the complete source code for the software. It will always be available there.

https://github.com/microworlds/node-world