

**TOPIC: - E-HEALTH RECORD MANAGEMENT SYSTEM**

**{EHRMS}**

**SCHOOL: - SCHOOL OF INFORMATION SCIENCE & TECHNOLOGY**

**DEPARTMENT: - COMPUTING SCIENCES**

**COURSE: - SOFTWARE ENGINEERING/COMPUTER SCIENCE**

**NAME REG NO CONTACT**

1. **LENAH MADONNA IN16/20408/17 0704446045**
2. **PETER KAMAU WANJIRU IN13/20506/17 0742567868**

A Research Project submitted in partial fulfillment of the requirement for Degree of **SOFTWARE ENGINEERING** and COMPUTERSCIENCE of Kisii University.

**©2021**

DECLARATIONThe attachment report is our original work prepared with no other than indicated source support and has not been presented anywhere else for degree or any other award.

**Signature…………………………… Date…………………………………**

**Name:** LENAH MADONNA **Registration Number:** IN16/20408/17

**Signature……………………………** **Date……………………………........**

**Name:** PETER KAMAU WANJIRU **Registration Number:** IN13/20506/17

# SUPERVISOR

I hereby affirm that this is original work of these students under my supervision and submitted to partial fulfillment of the requirement for Degree of **SOFTWARE ENGINEERING** and COMPUTERSCIENCE of Kisii University.

**Signature…………………………… Date……………………………………**

**Name:** MR.KENGERE KIBWAGE

# DEDICATION

We dedicate this thesis to both our lovely moms’ Jerusha Pamela & Paulina Wanjiru for their support and inspiration throughout our career.

# ACKNOWLEDGEMENT

We thank the Almighty God for the gracious protection in our academics.

We give special gratitude to our supervisor Mr. Kengere Kibwage.

TABLE OF CONTENTS

[DECLARATION i](#_Toc72147112)

[SUPERVISOR i](#_Toc72147113)

[DEDICATION i](#_Toc72147114)

[ACKNOWLEDGEMENT i](#_Toc72147115)

[LIST OF FIGURES iv](#_Toc72147116)

[LIST OF TABLES v](#_Toc72147117)

[LIST OF ACRONYMS AND ABBREVIATIONS v](#_Toc72147118)

[ABSTRACT vi](#_Toc72147119)

[CHAPTER ONE 1](#_Toc72147120)

[1.0 INTRODUCTION 1](#_Toc72147121)

[1.1 Background information 1](#_Toc72147122)

[1.2 Problem Statement 1](#_Toc72147123)

[1.3 Objectives 2](#_Toc72147124)

[1.3.1 General objective 2](#_Toc72147125)

[1.3.2 Specific objectives 2](#_Toc72147126)

[1.4 Scope and Boundary 2](#_Toc72147127)

[1.5 Justification of the System 2](#_Toc72147128)

[CHAPTER TWO 3](#_Toc72147129)

[2.0 LITERATURE REVIEW 3](#_Toc72147130)

[2.1 Blockchain Overview 3](#_Toc72147131)

[2.1.1 Blockchain evolution 4](#_Toc72147132)

[2.1.2 Blockchain Ownership 4](#_Toc72147133)

[2.1.3 Blockchain Architecture 4](#_Toc72147134)

[2.1.4 Features of Blockchain Technology 5](#_Toc72147135)

[2.2 Use of Blockchain in Healthcare and Related Work 7](#_Toc72147136)

[2.3 Proposed system 7](#_Toc72147137)

[2.4 Summary 8](#_Toc72147138)

[CHAPTER THREE 9](#_Toc72147139)

[3.0 METHODOLOGY 9](#_Toc72147140)

[3.1 System Development Methodology 9](#_Toc72147141)

[3.2 System Analysis and Design Tools 11](#_Toc72147142)

[3.2.1 Flow chart diagram 11](#_Toc72147143)

[3.2.2 Data Flow Diagrams 11](#_Toc72147144)

[3.2.3 Use case diagram 11](#_Toc72147145)

[3.2.4 Activity diagrams 11](#_Toc72147146)

[3.3 System Implementation Tools and Techniques 11](#_Toc72147147)

[3.3.1 Environment development tools 12](#_Toc72147148)

[3.3.2 Fabric runtime application 12](#_Toc72147149)

[3.4 System Testing and Validation 12](#_Toc72147150)

[3.4.1 Integration Testing 12](#_Toc72147151)

[3.4.2 Functionality Testing 12](#_Toc72147152)

[3.4.3 Performance Testing 12](#_Toc72147153)

[CHAPTER FOUR 13](#_Toc72147154)

[4.0 SYSTEM ANALYSIS AND DESIGN 13](#_Toc72147155)

[4.1 System Analysis 13](#_Toc72147156)

[4.2 Requirements Specification 13](#_Toc72147157)

[4.2.1 Functional Requirements 13](#_Toc72147158)

[4.2.2 Non-Functional Requirements 13](#_Toc72147159)

[4.2.3 Hardware and Software Requirements 14](#_Toc72147160)

[4.3 System Design 15](#_Toc72147161)

[4.3.1 System Architecture 15](#_Toc72147162)

[4.3.2 System Design Phase 16](#_Toc72147163)

[4.3.2.1 Use Case Diagram 16](#_Toc72147164)

[4.3.2.2 Flow Chart Diagram 19](#_Toc72147165)

[4.3.2.3 Entity Relationship Diagram 20](#_Toc72147166)

[CHAPTER FIVE 21](#_Toc72147167)

[5.0 SYSTEM IMPLEMENTATION, TESTING AND VALIDATION 21](#_Toc72147168)

[5.1 System Implementation 21](#_Toc72147169)

[5.2 Systems Components 22](#_Toc72147170)

[5.2.1 Define Tab 22](#_Toc72147171)

[5.2.1.1 Model File 22](#_Toc72147172)

[5.2.1.2 Script File 23](#_Toc72147173)

[5.2.1.3 Access Control File 23](#_Toc72147174)

[5.2.2 Test Tab 24](#_Toc72147175)

[5.2.2.1 Doctors’ Node 24](#_Toc72147176)

[5.2.2.2 Patients’ Node 24](#_Toc72147177)

[5.2.2.3 All Transactions 25](#_Toc72147178)

[5.2.3 Hyperledger Composer REST Server 25](#_Toc72147179)

[5.3 Testing Phase 26](#_Toc72147180)

[5.4 System Evaluation 27](#_Toc72147181)

[5.5 Deployment 27](#_Toc72147182)

[5.5.1 Phased Approach 27](#_Toc72147183)

[5.6 Conclusions 28](#_Toc72147184)

[CHAPTER SIX 29](#_Toc72147185)

[6.0 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS 29](#_Toc72147186)

[6.1 Summary 29](#_Toc72147187)

[6.2 Conclusions 30](#_Toc72147188)

[6.2.1 Advantages of the Blockchain-Based Electronic Health Record Management System 30](#_Toc72147189)

[6.2.2 Limitations of the Blockchain-Based Electronic Health Record Management System 30](#_Toc72147190)

[6.3 Recommendations 30](#_Toc72147191)

[REFERENCES 31](#_Toc72147192)

[APPENDICES 33](#_Toc72147193)

[APPENDIX A: Schedule 33](#_Toc72147194)

[APPENDIX B: Gantt chart of the Electronic Health Records Management System 33](#_Toc72147195)

[APPENDIX C: Budget 34](#_Toc72147196)

# LIST OF FIGURES

**FIG. 2.1** Centralized vs. Distributed Ledger

**FIG. 2.1.3** Blockchain Transaction process

**FIG. 3.1** Different phases of the waterfall model

**FIG. 4.3.1** Architecture of the system

**FIG. 4.3.2.1** Use Case Diagram

**FIG. 4.3.2.2** Flow Chart Diagram

**FIG 4.3.2.3** Entity Relationship Diagram

**FIG 5.2.1.1** Model File

**FIG 5.2.1.2** Script File

**FIG 5.2.1.3** Access Control File

**FIG 5.2.2.1** Doctor Nodes created and added to the network

**FIG 5.2.2.2** Patient Nodes created and added to the network

**FIG 5.2.2.3** a record of all transactions that executed in the network

**FIG 5.2.3** EHRMS REST Server

**FIG 5.4** Access Control Rules

# LIST OF TABLES

**Table 4.3.2.1 Use Case Description**

**Table 5.3 AddParticipant Test Case**

**Table 5.3 Submit Transaction Test Case**

**Table 5.3 Proof of Existence Test Case**

**Table 6.1: Schedule of the Electronic Health Records Management system**

**Table 6.2: Budget of the Electronic Health Records Management system**

# LIST OF ACRONYMS AND ABBREVIATIONS

EHRMS - Electronic Health Record Management System

EHRs - Electronic Health Records

EMR - Electronic Medical Record

MOH - Ministry Of Health

REST - Representational State Transfer

CRUD - Create Read Update Delete

CLI - Command Line Client

NPM- Node Package Manager

HIPAA- Health Insurance Portability and Accountability Act

# ABSTRACT

The Blockchain technology was first introduced through Bitcoin, extending its usage to non-financial applications, such as managing electronic medical records, is an attractive mission for recent research to balance the needs for increasing data privacy and the regular interaction among patients and health providers. Various systems that adopt the Blockchain in managing medical records have been proposed. However, there is a need for more work to better characterize, understand and evaluate the employment of Blockchain technology in the healthcare industry. Our project is a design of Blockchain based system, namely electronic health record management system (**EHRMS)**, for managing medical records. **EHRMS** is designed to improve the current systems as it provides interoperable, secure, and effective access for medical records by patients, healthcare providers, while keeping the patients’ privacy. **EHRMS** employs timed-based smart contracts for governing transactions and controlling accesses to electronic medical records. It adopts advanced encryption techniques for providing further security. Our project proposes a new incentive mechanism that leverages the degree of health providers regarding their efforts on maintaining medical records and creating new blocks. Our mission is to place the patient at the Centre of the digital transformation of healthcare so that their data will always be secure and under supervision.

Blockchain Technology facilitates a shared, immutable and history of all the transactions creating software of trust, responsibility and transparency. Electronic Health Records (EHRs) are both crucial and sensitive as they contain essential information and are frequently shared among different parties including hospitals, pharmacies or private clinics. This information must remain correct, up to date, private, and accessible only to the authorized people. Securing data, storage, transaction, and managing their smooth integration are immensely valuable to any data-driven organization, especially in healthcare where Blockchain technology has the potential to resolve these critical issues in a robust and effective way.

# CHAPTER ONE

## 1.0 INTRODUCTION

### 1.1 Background information

In this era, the common thing that roams around is data. Users expect a rapid and smooth flow of data. Unfortunately, the healthcare industry has lagged behind. Medical data contained in older systems is soiled and difficult to share with others because of varying formats and standards. As a result of this, stakeholders are incentivized to preserve their own records, and no single version of the truth exists. Nowadays, patients have their healthcare data distributed over various systems, hospitals and networks. Each medical institution has their own electronic health records (EHRs) for the patients. Hence the medical data can be altered by the organization to benefit their own needs. So it is important to develop a medical chain which can provide an electronic health record (EHR) for each user.Blockchain based EHRs provide the user with complete access and control over their own data. Moreover, the Blockchain based EHRs platform will provide the users with a complete record of who is authorized to access their records, the time of access and the extent to which the data can be accessed. Our main goal is to place the patient at the core of the digital transformation of healthcare. Many have tried to overcome this issue, and it is prioritized on the agenda of governments and a cause of frustration for both doctors and patients. Public hospitals in Kenya are increasingly purchasing systems to support administrative functions and this study highlights challenges faced by hospital users and vendors. Significant work is required to ensure interoperability of systems within hospitals and with other government services. Additional studies on clinical usability and the workflow fit of digital health systems are required to ensure efficient system implementation. However, this requires support from key stakeholders including the government, international donors and regional health informatics organizations. Despite electronic medical record (EMR) systems being in existence since 1972, it’s only recently that governments worldwide have begun to encourage digitization of medical records. In Kenya, the Ministry of Health (MOH), i.e., the Ministry of Medical Services and the Ministry of Public Health and Sanitation, are actively promoting the standard implementation of EMR systems with the aim of improving health care delivery, health systems management and patient health outcomes.

### 1.2 Problem Statement

Over decades, medical facilities have evolved elegantly. Still most of us are the witness of the fact that whenever we see a doctor, we need to put forward our medical file in front of him/her. We will be using Blockchain technology to store the patient records. This will ensure that the information remains secure while being decentralized across different peers.

### 1.3 Objectives

### ****1.3.1 General objective****

* To develop a Blockchain based electronic health record management system.

### ****1.3.2 Specific objectives****

1. **To design and develop a patient-centered system.**
2. **To preserve privacy of patient’s electronic medical records.**

### 1.4 Scope and Boundary

The target users of this system are the patients, medical and scientific community. The Blockchain-based Electronic Health Record Management System ensures ready access to a lot of all-inclusive, updated patient info, quick, reliable and secure information. The system goes beyond Streamlining of clinical data and powerful tools that take care, supporting multi-disciplinary team operations. Online call support tools, access to top care and quicker access to consultant doctor opinions and identification is also achieved through the Blockchain-based Electronic Health Record Management System.

(IJITEE et al., 2019)

### 1.5 Justification of the System

Blockchain-based Electronic Health Record Management System is meant to solve the existing electronic health records problems.

The system solves the existing system problems by ensuring that the patient has full control of who can access his/her medical records. By the help of Blockchain technology and smart contracts our system enables employment of a set of connectivity and timing functions to provide reasonable period of time for performing transactions and thus ensuring an authorized transaction is intended and also proof of existence of a given medical record. (Medchain et al., 2019).

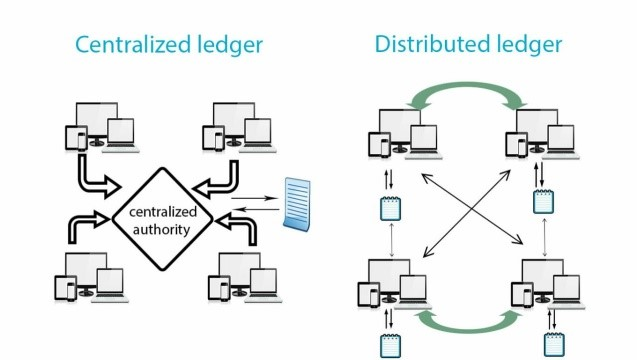
# CHAPTER TWO

## 2.0 LITERATURE REVIEW

This section of the study will give an in-depth description of the Blockchain-based Electronic Health Record Management System. This section also provides an overview of fundamental concepts related to the Blockchain technology and applications of the Blockchain technology in the area of healthcare.

### 2.1 Blockchain Overview

Blockchain, a distributed ledger, is a chain of time-stamped blocks containing a specific number of validated transactions. Blocks are linked cryptographically using the hash value of the previous block. Each transaction generated by a user or node is digitally signed using a private key and broadcasted to the network. A validation/mining node takes up that transaction and encloses it into a block then block is broadcasted to the network. Each node of the network checks the validation of the block by implementing the consensus protocol. The validated block is appended to the chain then updated ledger is replicated throughout the permissioned nodes of the network. Consensus protocol replaces the trusted third party or the central authority. Fig. 2.1 illustrates the difference between centralized and distributed ledger. The ledger provides security, auditability and anonymity-based transparency.

 **FIG. 2.1** Centralized vs. Distributed Ledger.

### 2.1.1 Blockchain evolution

Blockchain technology evolution is categorized into three phases: Blockchain 1.0, 2.0, and 3.0. Blockchain 1.0 is for the decentralization of money or known as “Internet of Money”. This first application established a peer to peer digital payment system without reliance on a third party. This tier of technology implements Proof of work, the consensus protocol for validating a block to embed in the ledger. A digital reward is given to the successful miner for mining the block, for his contribution to the ledger. The second generation is Blockchain 2.0 which is the application of decentralization of smart property and smart contracts, came in 2014. It aims to transfer any unit of value using the concepts of smart contracts for automated administration and supervision. The smart contract is a script which triggers after meeting the conditions encoded within it. Ethereum and Eris Blockchain come under this category. The third Blockchain technology refers to Blockchain 3.0 which targets the welfare of society and is particularly recommended to register and transfer public records in the areas of government, health, science, literacy, and art. Examples are health coin, learning coin and grid coin. Blockchain 2.0 and 3.0 are also known as non-financial applications. Alternative consensus protocols and alternative crypto-currencies have also been introduced. (Melanie Swan et al.).

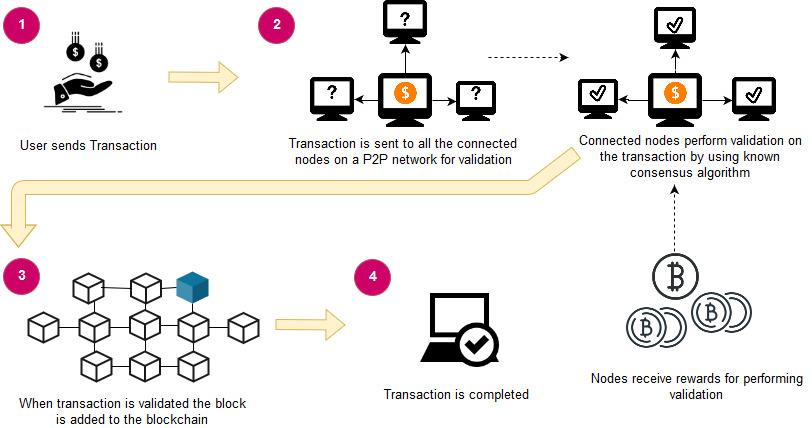
### 2.1.2 Blockchain Ownership

There are two basic types of Blockchain i.e. permissioned Blockchain and Permissionless Blockchain. Permissioned Blockchain is a custom-built setup by a single authority or a consortium. The verification process can be done by a central authority or a set of trusted preselected parties (consortium). This private setup restricts data access to the group of users or a set of groups that controls the Blockchain. A smaller number of participants provide efficiency and scalability. These Blockchain ultimately have a central authority. This centralization of the setup can pave way for tampering as the 51% majority is required to get the consensus and it can be done easily in this controlled setup. Eris, Ripple, and Hyper ledger are examples. Permissionless Blockchain are fully decentralized to a large number of nodes and low in efficiency. These Blockchain require no prior authorization of participants for mining the transaction blocks. Anyone can contribute his/her computational power for network tasks and can get a monetary reward in return. This Blockchain gives the public access to read and write transactions to the Blockchain which is visible to everyone so also known as Public Blockchain. Examples of permissionless Blockchain include Bitcoin and Ethereum.

### 2.1.3 Blockchain Architecture

To understand the Blockchain architecture let us use the following figure 2.1.3 that explains the whole process of a transaction being sent from a user on the Blockchain network.

1. A new transaction being sent by a user on the Blockchain network suggests that a new block is created. A block in the Blockchain is used for keeping transactions in them and these blocks are distributed to all of the connected nodes in the network. That transaction placed inside a block is broadcasted to all of the nodes in the network. All the nodes in the network have a copy of the complete Blockchain that helps them in verification process. When a block containing the user transaction is broadcasted to all of the connected nodes, they verify that the block is not tampered by any means. If this verification results in success then the nodes add that block in their own copy of Blockchain.
2. This whole process of the block being added on the Blockchain is done by the nodes reaching upon a consensus where they decide which blocks are valid to be added on the Blockchain and which are not. This validation is performed by the connected nodes using some known algorithms to verify the transaction and to ensure that sender is an authenticated part of the network. When a node succeeds in performing the validation that node is rewarded with crypto-currency. This process of validating the transaction is known as mining and the node performing this validation is known as miner.
3. After validation is done that block is added to the Blockchain.
4. After the whole process of validation is performed the transaction is completed.



**FIG. 2.1.3** Blockchain Transaction process.

### 2.1.4 Features of Blockchain Technology

1. **Decentralization:** The Blockchain is a distributed digital ledger composed of a chain of blocks containing transactions. The decentralized database is shared and opens to all parties throughout the nodes of the network. Blockchain-based networks provide fault-tolerant architecture as end-to-end replications remove the reliance on a single point of failure.
2. **Consensus mechanism:** Blockchain is a peer-to-peer distributed network without any intermediary. Each digitally signed block is sent to the mining pool where it is taken over by network nodes called miners and verified using the consensus algorithm. The winner from the miners broadcast the block to all other nodes which confirm and validate the block with consensus and append the block in their ledger. The winner also gets a financial reward for its work. Many alternative consensus protocols have also been proposed, such as proof of-stake, proof-of-burn, and proof-of elapsed-time. Data integrity is maintained by computing these consensus algorithms as a substitute to third trusted party.
3. **Immutability:** Blockchain is immutable and tamper-proof thus ultimately provides security. The hash function makes the Blockchain as a tamper-resistant ledger. A hash value is calculated by implementing some hashing algorithms (SHA-256, RSA, RIPEMD-160, etc.) on a block of transactions. The hash value is further used to create a chain of blocks. In this way Blockchain provides robustness. If someone intends to alter previous transactions, then it will require a change in the hash value which further needs the consensus of network and high computational power which is an unrealistic approach in this computational model. The hash value is also used to represent a user concealing real identity which is used for privacy purposes.
4. **Traceability**: Blockchain is a digital ledger consisting of continuously growing sequence of blocks. A block is comprised of a complete list of transaction records. In this chain of blocks, every block has a parent block. The first block in the chain is known as the genesis block. Hash code of genesis block is added to the header of the second block then hash code for the second block is computed over the hash of genesis block and transactions of the block jointly. Hash of the second block becomes the block header of the third block and so on. In this way, the blocks are linked with each other having a time stamp as well. This link can be chased back to the origin or genesis block. This feature of Blockchain provides data provenance to keep chronical track of activities and may also help to investigate backward throughout the chain.
5. **Smart Contract:** A smart contract is a computerized computational logic or terms of the contract. It automatically triggers transactions between parties after fulfilling encoded logic. This implementation makes the Blockchain flexible and programmable. Smart contracts are programmed for management and administration purposes. The smart contract can be implemented in supply chains, claim insurance and clinical trials. Clinical trials usually encompass a sequence of dependent phases to get specific outcomes. Each phase can be encoded in a smart contract which will be triggered after the consensus of network nodes. So, smart contracts can enforce traceability and transparency with full control over associated processes.

### 2.2 Use of Blockchain in Healthcare and Related Work

Blockchain technology was designed by **Satoshi Nakamoto**; the basic idea was to have a cryptographically secured and a decentralized currency that would be helpful for financial transactions. Eventually, this idea of Blockchain was being used in various other fields of life; healthcare sector also being one of them intends to use it.

Healthcare sector always remains one of the most popular areas of research from the last few decades; we keep on finding innovative and more reliable ways to help the community and healthcare industry. Different stakeholders (practitioners, medical specialists, hospitals, therapist and patients) need to organize access and share health records without any modification in a secure and interoperable way. Data provenance is also essential to prove the authenticity of records. Blockchain technology is being implemented in different scenarios and has the potentials to address the key issues of the healthcare sector. However, it needs more research to be focused to deploy real-time applications of this technology. Following are some applications of this technology in the healthcare sector: Firstly, MedRec is a platform that provides decentralized record management, authorization and data sharing among healthcare stakeholders. Patients can save their data and can also grant and withdraw permissions to their records. This framework provides full confidentiality as the records are not stored on Blockchain instead pointers to the data storage locations, logs and permissions are only stored in this Blockchain. Secondly, Gem in collaboration with Philips Blockchain Lab has been introduced Gem Health Network using Ethereum Blockchain. This framework is established to address operational costs. This shared infrastructure provides interoperability among various providers accessing the same information to boost better patient care. Guard time healthcare platform creates a no intermediated relationship between patient and provider in Estonia. Guard time Blockchain enables transparent information sharing among the patient, provider, and payer which promises secure, reliable and auditable records. Patient’s health data is being required by research organizations. In this context, Health bank has been providing a platform for patients to save and share their health data with research organizations which can be used for academic research and pharmaceuticals. Owners of data can access their EMRs from a shared data pool. This secure and scalable system identifies, authenticates and authorizes users using cryptographic keys and digital signatures acquiring an edge over HDG (Healthcare Data Gateways) which is a smartphone application built over Blockchain cloud. Fast Healthcare Interoperability Resources: FHIR chain was developed by the Health Level Seven International (HL7) organization for exchanging clinical data. FHIR increases efficiency and interoperability.

### 2.3 Proposed system

In the field of healthcare, Blockchain can bring a great difference. The data will be distributed among the nodes. Therefore each node will be responsible for handling its own data. This gives the user a control over their data. Organizations can store the medical information’s in a distributed ledger. As the data is decentralized, it will be easy to create trust among various participants.

### 2.4 Summary

Blockchain is a decentralized network technology. Blockchain consists of a number of blocks. The blocks are connected with each other through a chain, hence the name Blockchain. A block consists of a number of transactions. The links between the blocks are made up of hash values. The hash values are calculated using the transactions in a block and the hash value of the previous block. Healthcare is one of the biggest industries. It also remains as an industry which lacks transparency. At crucial situations the patients’ medical reports are not readily available. Blockchain is a technology which can provide transparency to its participants. Combining Blockchain with healthcare can bring a huge change in the healthcare domain. By including frameworks like Hyperledger, we can provide an industrial standard to healthcare industry along with transparency and patient consent. All the medical data can be stored in a distributed ledger, which can be used at critical periods for examining report details. Blockchain also provides high security to the data. The data in the Blockchain will remain tamper proof.

# CHAPTER THREE

## 3.0 METHODOLOGY

This chapter gives an in-depth discussion of the software methodology that was used in the development process of the system. This study used waterfall methodology as a framework to plan, design, develop and evaluate the system. This section explains the waterfall methodology and its application in the development of the system (Pekka et al., 2012).

### 3.1 System Development Methodology

The system development methodology used in development of the Electronic Health Records management system is waterfall methodology. It was the first methodology to be introduced in software development. It is also referred to as linear sequential life cycle model. It is very simple to use and understand. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

In waterfall approach the whole process of software development is divided into separate phases. The output of one phase acts as input of the next phase sequentially (Luhmann et al., 1990).

Following is the diagrammatic representation of the different phases of the waterfall model.

**FIG. 3.1** Different phases of the waterfall model

The sequential phases of waterfall model are:

* **Requirement gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document
* **System design:** The requirement specifications from the first phase are studied in this phase and system design prepared. System design helps in specifying hardware and system requirements and also helps in defining the overall system architecture.
* **Implementation:** With inputs from system design, the system is first developed in small programs called units which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as unit testing.
* **Integration and testing:** All the units developed under the implementation phase are integrated into a system after testing of each unit. In the post integration the entire system is tested for any failure and faults.
* **Deployment of the system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released to the market.
* **Maintenance:** There are some issues that come up in the client environment to fix these issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after a designed set of goals are achieved from the previous phase and it is signed off, so the name “waterfall model”. In this model phases do not overlap.

**Waterfall mode application**

Every software developed is different and requires a suitable system development life cycle (SDLC) approach to be followed based on internal and external factors. Some situations where use of waterfall model is suitable are:

* When requirements are very well documented, clear and fixed.
* When product definition is stable
* When the technology is well understood and not dynamic
* When there are no ambiguous requirements
* When ample resources with required expertise are available to support the product

**Advantages of waterfall model**

1. Simple and easy to understand and use
2. Easy to manage due to the rigidity of the model, each phase has specific deliverables and a review process.
3. Phases are proposed then completed one at a time.
4. Works well for a small project where the requirements are well understood.
5. Clearly defined stages.
6. Well understood milestones.
7. Easy to arrange tasks.
8. Process and results are well documented.

### 3.2 System Analysis and Design Tools

### 3.2.1 Flow chart diagram

Flow charts are used to show how information will be flowing within the Electronic Health Record Management System and how the user goes through the processes to attain the use goal.

### 3.2.2 Data Flow Diagrams

Data Flow diagrams graphically characterize data processes and flows in a business system. In their original state data flow diagrams depict the broadest possible overview of system inputs, processes and outputs, which correspond to those of the general systems model. A series of layered data flow diagrams may also be used to represent and analyze detailed procedures within a larger system. The data flow approach emphasizes the logic underlying the system. By using combinations of only four symbols, the systems analyst can create a pictorial depiction of processes that will eventually provide solid system foundation.

### 3.2.3 Use case diagram

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. A use case can be described as a collection of related success and failure scenarios, describing actors using the system to support a goal. An actor is something with a behavior or a role for example a person, another system or another organization. The actors of this system are administrators and the users. A scenario is a specific sequence of actions and interactions between actors and the system. A goal is the end result of most use case.

### 3.2.4 Activity diagrams

An activity diagram is used to show the flow from one activity to another activity. It captures dynamic behavior of a system. So the control flow was drawn from one operation to another. The only missing thing in an activity diagram is the message part. It does not show any message flow from one activity to another.

### 3.3 System Implementation Tools and Techniques

System implementation workflow includes tasks to set up the environment for implementation. All the medical data can be stored on a permissioned Blockchain. Our Blockchain is built using the Hyperledger Fabric framework. We use a tool called Hyperledger composer to implement the fabric system. The workspace environment is set up to work with the Hyperledger fabric and Hyperledger composer.

### 3.3.1 Environment development tools

The following tools are used to build an application using Hyperledger fabric: -

* Composer-cli- The composer command client interface provides various commands to execute the composer application.
* Composer REST server- It is used to generate a REST server which consists of methods such as get, put, post, delete.
* Composer playground- It is web user interface where the composer applications can be deployed and tested.
* Yo generator- It is used to generate the composer template and to create an Angular application.

### 3.3.2 Fabric runtime application

A folder is created (fabric-dev-servers) for the development setup in the home directory. The required packages are downloaded. The packages are then extracted. The version of the fabric is exported. The hlfv12 specifies the version of the Fabric to be installed. Bash scripts are executed to setup all the development environments. After the creation and usage of Fabric environment, we need to stop all the instances and remove all the instances from the peer.

### 3.4 System Testing and Validation

Testing the system is an integral part of the system development process. Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction. Testing helps in verifying the correctness, functional behavior, and usability of the system before it is released publicly. The following testing methodologies were used to test the Electronic Health Record Management System:

### 3.4.1 Integration Testing

Integration testing was done to ensure that all system components that share data or depend on other components work together properly. Workflows reflected actual new processes.

### 3.4.2 Functionality Testing

This was done to test the system’s functional and non-functional requirements. Entries entered were displayed correctly. Fields edits (for example; valid values, options, defaults) function as expected.

### 3.4.3 Performance Testing

Performance testing was done to measure the response times for key transactions or interactions with the system, and ensured they are within acceptable limits which were defined in the contract.

# CHAPTER FOUR

## 4.0 SYSTEM ANALYSIS AND DESIGN

### 4.1 System Analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Electronic Health Record Management System was analyzed using use case diagrams, data flow diagrams and flow charts. This was able to disclose the system components and interpret the facts about the system (Szalvay, 2014).

### 4.2 Requirements Specification

### 4.2.1 Functional Requirements

These requirements define the capabilities and functions that the implemented system must have in order to achieve its intended purpose. It includes a set of inputs, behavior and outputs in line with the objectives of the study. They include the following transactions executed in the system:-

1. AddParticipant - Whenever a new node will be added to the system this transaction will get executed.
2. UpdateParticipant - It occurs when a modification is made in the data inside the participant’s node.
3. CreateMedicalRecord - This transaction would create records in the network. It contains fields like recordID, owner, and list of authorized patients. It contains fields that store medical information of the patients like medical history, last consultation with which doctor, the date of consultation, allergies, any harmful habits, etc. The ID generated for the record is unique to the record and is used to identify that specific record in the collection.
4. GrantAccess - To manipulate the records the doctor would need to have access to the record, only the authorized doctor would have the right to access and read or write the medical record. This access is granted using this transaction.
5. UpdateAsset - It occurs when we alter the details of the medical records.

### 4.2.2 Non-Functional Requirements

1. **Usability**Usability defines how difficult it will be for a user to learn and operate the system. Usability can be assessed from different points of view:

* **Efficiency of use:** the average time it takes to accomplish a user’s goals, how many tasks a user can complete without any help and the number of transactions completed without errors.
* **Intuitiveness:** how simple it is to understand the interface, buttons, headings, etc.
* **Low perceived workload:** how many attempts are needed by users to accomplish a particular task?

1. **Security**Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users’ roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information. Security also includes protection against viruses and malware attacks.
2. **Reliability**Reliability defines how likely it is for the software to work without failure for a given period of time. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components. To measure software reliability, you can count the percentage of operations that are completed correctly or track the average period of time the system runs before failing.
3. **Performance**Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes system safety when it’s is overloaded.
4. **Availability**

Availability is gauged by the period of time that the system’s functionality and services are available for use with all operations. So, scheduled maintenance periods directly influence this parameter. And it’s important to define how the impact of maintenance can be minimized. When writing the availability requirements, the team has to define the most critical components of the system that must be available at all time. You should also prepare user notifications in case the system or one of its parts becomes unavailable.

1. **Scalability**Scalability requirements describe how the system must grow without negative influence on its performance. This means serving more users, processing more data, and doing more transactions. Scalability has both hardware and software implications. For instance, you can increase scalability by adding memory, servers, or disk space. On the other hand, you can compress data, use optimizing algorithms, etc.

### 4.2.3 Hardware and Software Requirements

The software and hardware requirements are highlighted below:

1. Computer - Linux operating system; Ubuntu 16.04 LTS with 2.50GHz speed, 8GB RAM, 7.7GiB memory, 64-bit operating system and 154.4GB ROM.
2. Docker Engine - It is a container technology used to build applications. It is a light weight tool.

* Docker version 20.10.5, build 55c4c88

1. Docker Compose - It is used in order to define and run multi container Docker applications.

* Docker-compose version 1.13.0, build 1719ceb

1. Node - Constitutes communication entities of the Blockchain.

* Version 8.17.0

1. Npm - Is a package manager for JavaScript programming language.

* Version 6.13.4

1. GIT- It is a version control system. It is used to keep a track about the changes made to an application.

* Git version 2.31.0

1. Python - It is an interpreted high level language.

* Python 2.7.12

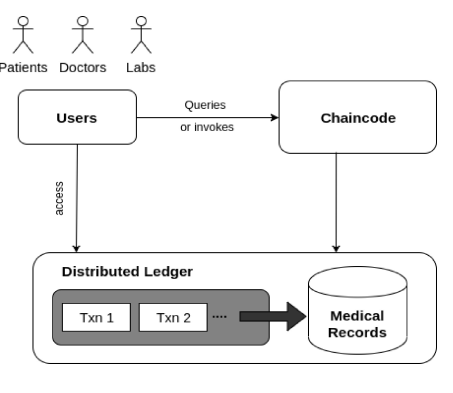
1. Visual studio code - code editor.

### 4.3 System Design

This section focuses on the design structure of the solution the researcher has developed. It shows how various components of the system work together to achieve the desired goal and their relationships. Design includes the system architecture, a use case, class diagram, entity relation diagram, system sequence diagram and the database schema.

### 4.3.1 System Architecture

In this system, medical records are the asset of the network. Each medical record is owned by some patient who is registered on the network. Whenever a transaction is executed the value of the asset changes. Changes are like updates in the records if the patient is diagnosed with some new disease, modifications in medications and test results. The transactions are actions performed mostly on the asset in the network like adding a participant in the network, creating a medical record, retrieving specific information from the network, updates in the participant’s information, giving access to clinician and revoking access from them. For the execution of some of these transactions, there is a need to have a relationship between the two participating nodes. For example, to give access to a patient’s medical information to a clinician the patent’s ID must be in the list of that clinician’s patients. In simple words, the person whose medical records are to be accessed must be a patient of the clinician who wants to get access to the medical record of that patient. The permission rules are also defined in this system. These rules of control which a participant is granted defines what kind of access the participant has and to what resources. This helps in restricting access to all the resources of the system. Only authorized users get to manipulate or read specific records only.



**FIG. 4.3.1** Architecture of the system

### 4.3.2 System Design Phase

The system requirements specified under requirement specification were combined with the developer’s ideas to develop a system design that fulfills the functional, non-functional requirements and the research objectives. The following section discusses the design diagram that depicts the actual implementation of the system.

### 4.3.2.1 Use Case Diagram

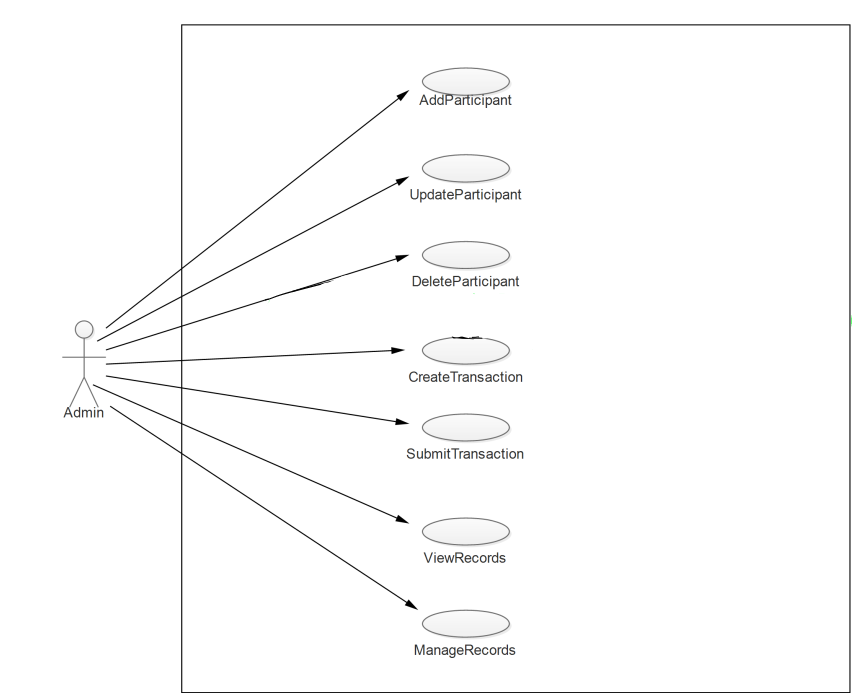
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. The purpose of use case diagram is to capture the dynamic aspect of the system.

The system functionality is further described in detail using the use case diagram.

**Actor**

Below is the actor who will interact with the system:

Administrator - this is the individual in charge of records management.



**FIG. 4.3.2.1** Use Case Diagram

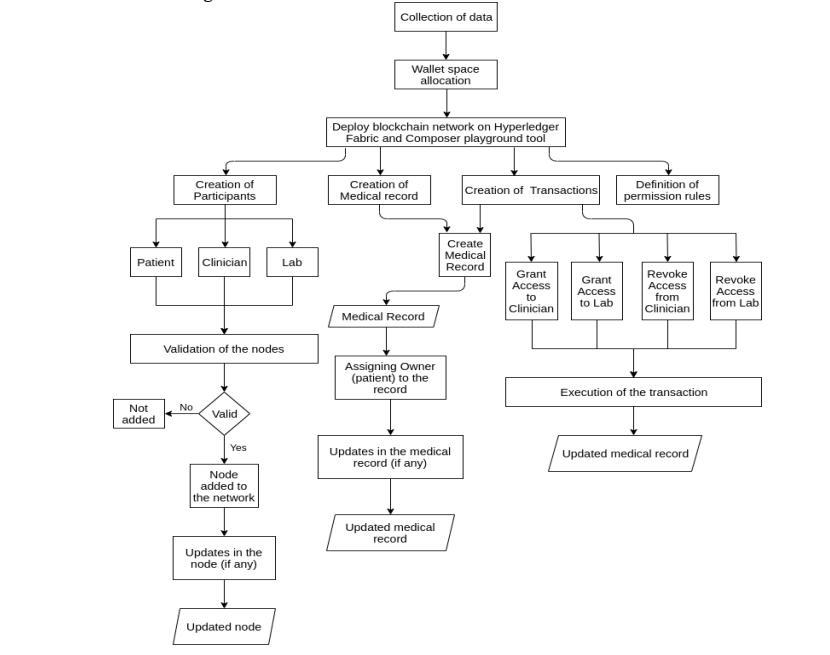
**Use case Description**

Discussed below is the use case description.

**Table 4.3.2.1 Use Case Description**

|  |  |
| --- | --- |
| **Use case:** Transaction Flow | |
| **Use case Description:** Start - Finish of Medical Record creation | |
| **Primary Actor:**  Admin | |
| **Condition:**  The Network Admin | |
| **Typical Case of Events** | |
| **Actor Response**   1. CRUD participant(Patient/Doctor) 2. View Existing Medical Records | **System Response**   1. Perform Executed Transactions 2. Retrieve Records |
| **Alternative Flow**  Participant cannot be added if he/she exists in the system {EHRMS}.   * The participant already exists. | |

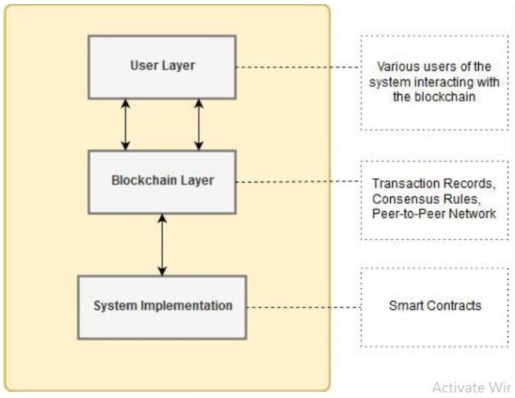
### 4.3.2.2 Flow Chart Diagram



**FIG. 4.3.2.2** Flow Chart Diagram

### 4.3.2.3 Entity Relationship Diagram

The entity relationship diagram as shown in FIG 4.3.2.3 illustrates the conceptual view of the entire system. The conceptual system consists of participants who are the patients, doctors and the administrator.



**FIG 4.3.2.3** Entity Relationship Diagram

**User layer**

A user of a system is as an individual who makes effective use of the system and its resources. The users of this system are the patients, doctors and administrator. The main task of the administrator is to interact with the system and perform basic tasks such as create, read, update and delete the medical records.

**Blockchain layer**

This layer contains the code or mechanism for interaction of user with the system which is functioning on the Blockchain. The system includes the transaction such as Add record, Update record, View record and Delete record.

**System implementation**

The system is implemented using Hyperledger composer.

# CHAPTER FIVE

## 5.0 SYSTEM IMPLEMENTATION, TESTING AND VALIDATION

This chapter explains how implementation, testing and validation phases of the system were carried out. How the major functionalities of the system have been implemented and the tests carried out on them, this aids in understanding how the application meets the set research objectives as stated in Chapter 1.

### 5.1 System Implementation

The system was implemented successfully upon completion of the coding process. To implement this system, we used a Blockchain-based framework Hyperledger Fabric and Composer tool to implement the project.

**Implementation Environment**

1. **Hyperledger Fabric**

It is a Blockchain framework implementation founded by Linux foundation and one of the Hyperledger projects. We used this framework as it allows components, such as consensus and membership services, to be plug-and-play. It helps container technology to host smart contracts, called “chaincode”, that comprise the logic of the system.

1. **Hyperledger Composer**

It is an open source tools for building a Blockchain business network. The tool helps to create multiple smart contracts and Blockchain application. When a medical record of a patient is generated and tested, it can be added on to the Blockchain network, which offers patients with the perfect and assurance that the record cannot be altered. These personalized health records could be encrypted and kept on the Blockchain network with a private key, which allows only verified users to access the health records in crucial time, thereby ensuring the privacy of the patient.

**Implementation Details**

To implement this Blockchain-based EHRMS network we followed these steps:

1. Collection of data: Patient’s personal information and medical data like vital signs, allergies, harmful habits, medical history, test results, medications and data generated through clinical diagnosis by the doctor.
2. Wallet allocation: It is a space allocated to deploy the Blockchain network. It is the place where all the transactions are recorded.
3. Deploying a Blockchain network using Hyperledger Fabric and Composer: On composer playground, after wallet allocation, we deploy our business network and start our Blockchain network.
4. Creation of different nodes in the system: We create a model of our system that had the template design of different participants/nodes (like Patients and Doctors) in our Blockchain network.
5. Creation of medical records: We also create a template for storing medical records owned by patients.
6. Creation of transactions: We create the transactions that have to be executed as per the need.
7. Addition of node to the system: Creation of an instance of the Patient node, Clinician/Doctor node and medical record node owned by some patient is done using sample data collected. The nodes are then validated by other registered nodes in the network and a public identifier is generated before adding them to the network.
8. Specification of various permissions granted to the user: Here we specify what system resources (medical records) can be accessed by which participants. Only the participant with certain permission (like Read-only, Write or All) is allowed to access certain medical records data only.
9. Execution of transactions: Various transactions are executed according to the need of the user and records can also be retrieved from the stored collection, if needed. After the execution, an updated medical record is generated.

### 5.2 Systems Components

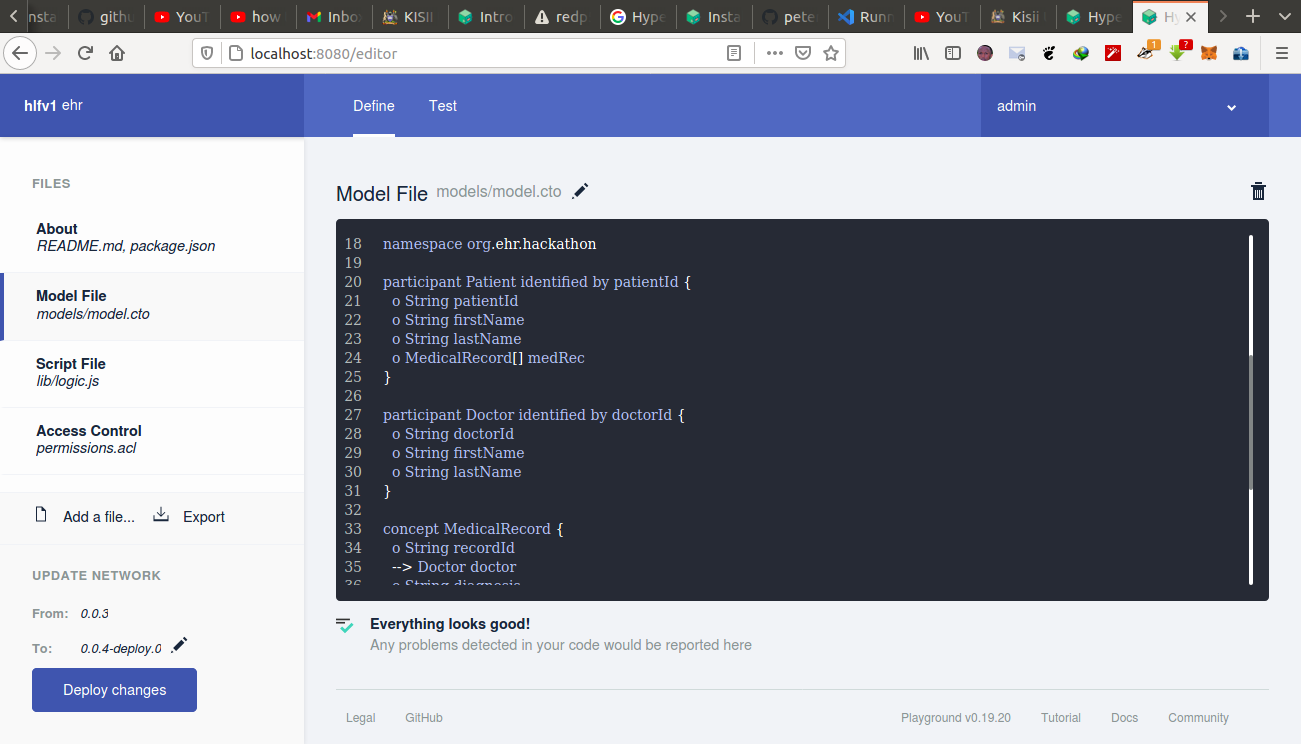
Transaction on the Blockchain network is invoked only by the hospital Admin. The hospital Admin in the EHRMS performs various transactions like adding a participant in the network, creating a medical record, retrieving specific information from the network and updating the participant’s information.

### 5.2.1 Define Tab

The Define Tab shows a list of all the files in the current business network definition. To examine the contents of a file, click on it, and it will appear in the editor view.

### 5.2.1.1 Model File

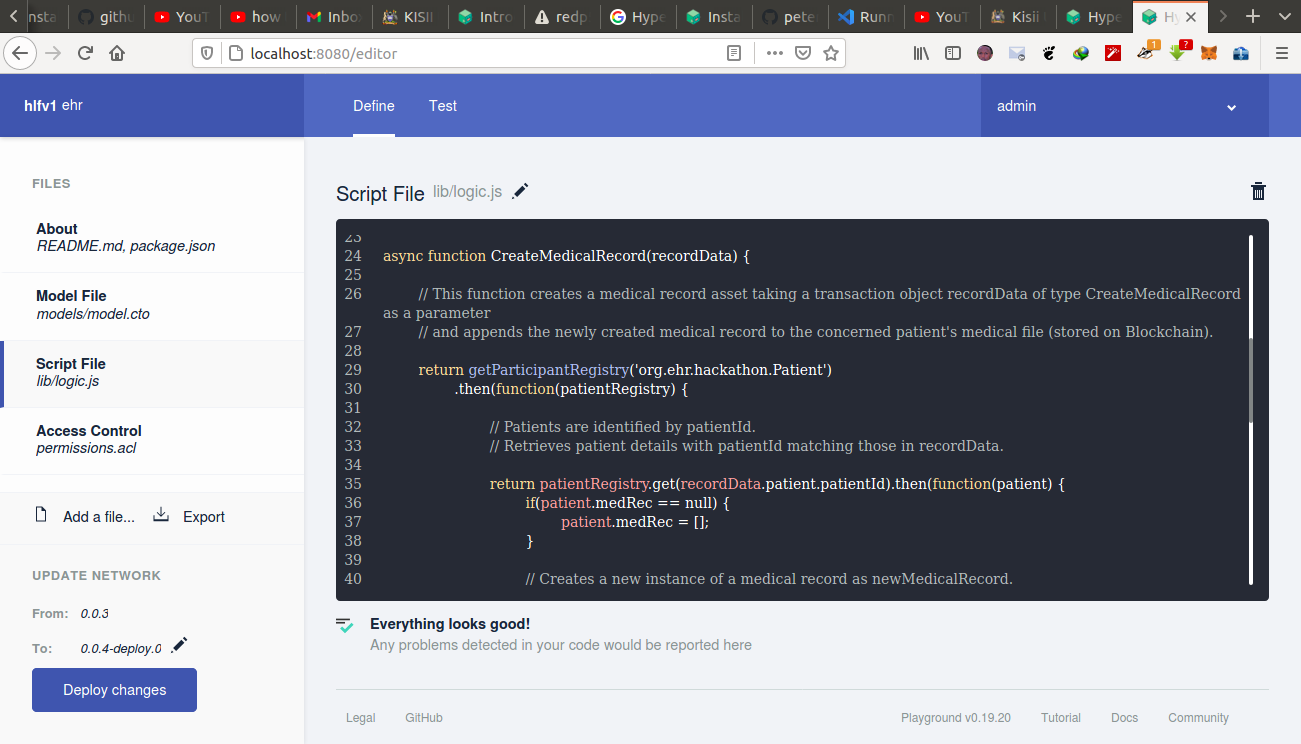
Model file represents the database schema. Object-oriented modeling language is used to define domain model of the business network.



**FIG 5.2.1.1** Model File

### 5.2.1.2 Script File

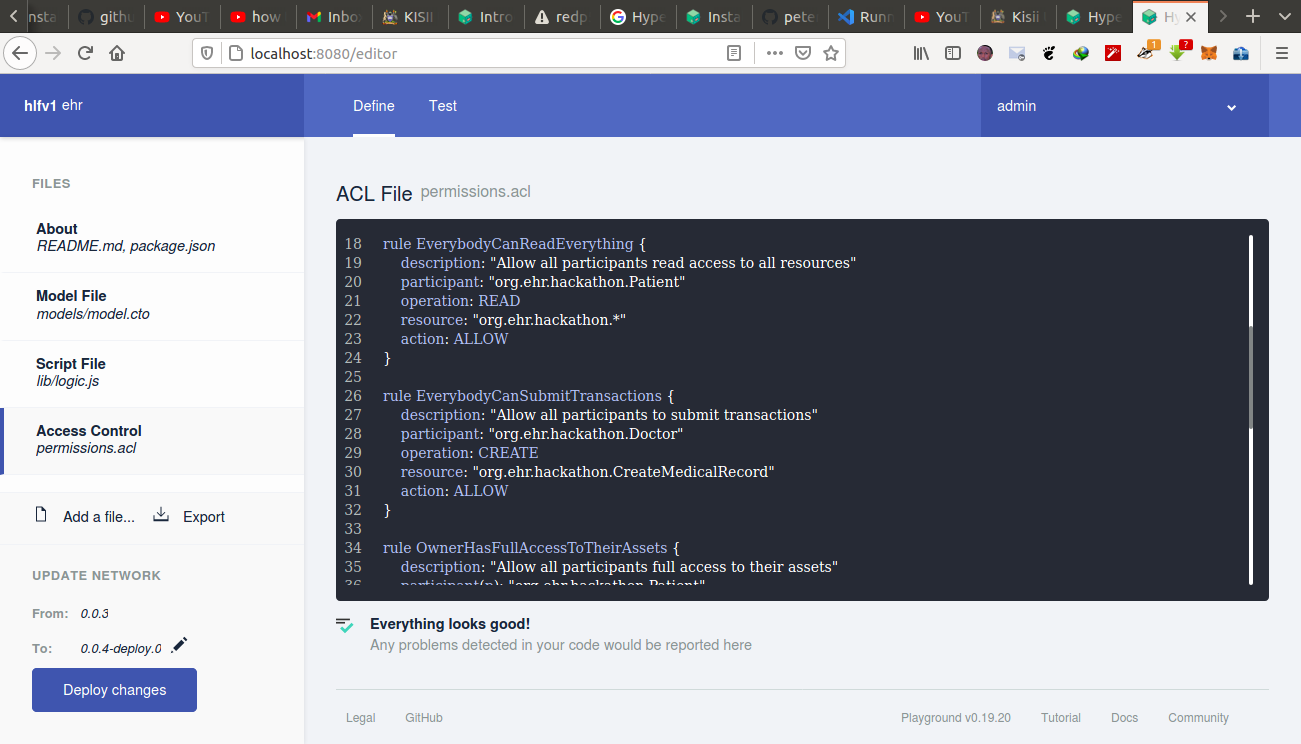
Uses JavaScript which is “smart contract” language used to define the business logic of the system.



**FIG 5.2.1.2** Script File

### 5.2.1.3 Access Control File

Set of rules to govern the access of the system.



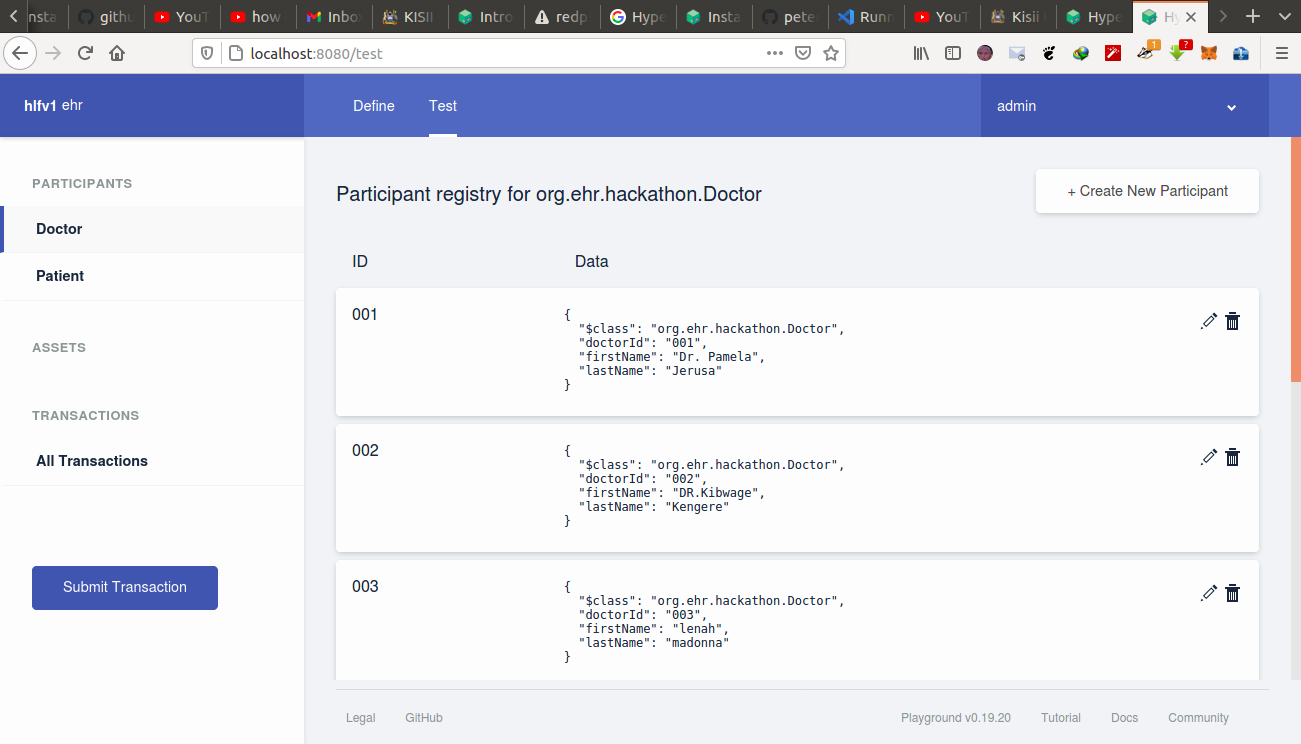
**FIG 5.2.1.3** Access Control File

### 5.2.2 Test Tab

We used the Test tab to test the deployed business network by using the asset types, participant types, and transactions which we defined in the Define tab.

### 5.2.2.1 Doctors’ Node

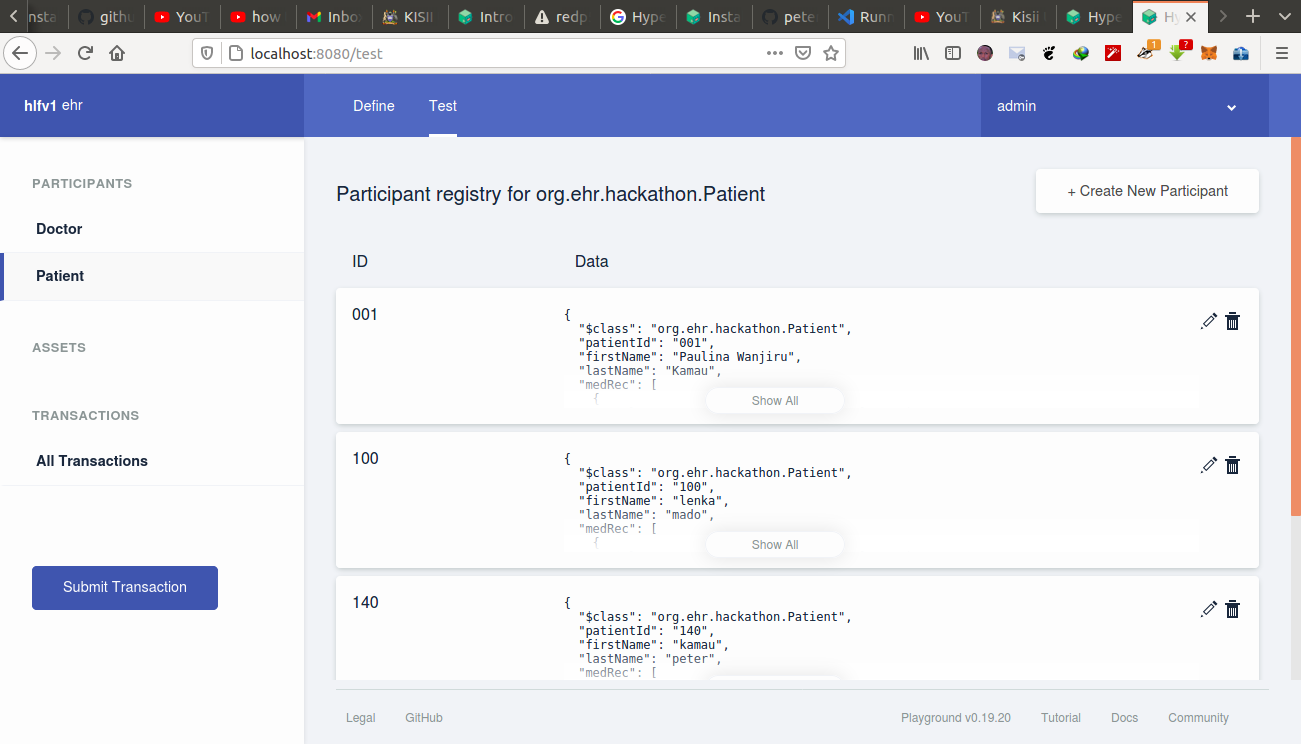
List of Doctors created in the system.



**FIG 5.2.2.1** Doctor Nodes created and added to the network

### 5.2.2.2 Patients’ Node

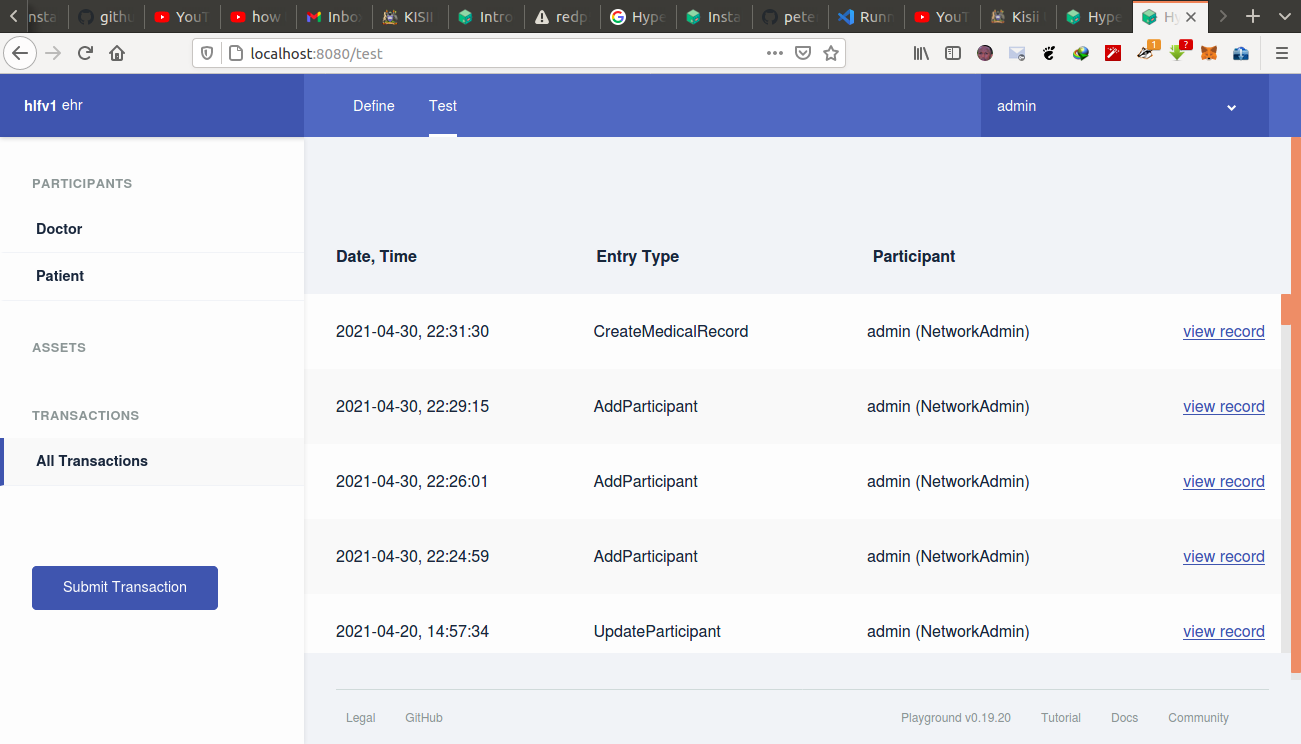
List of Patients created in the system.



**FIG 5.2.2.2** Patient Nodes created and added to the network

### 5.2.2.3 All Transactions

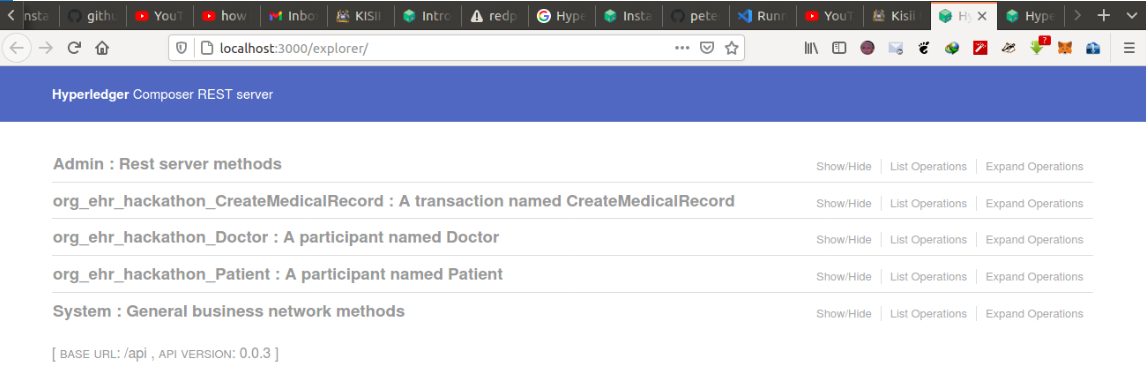
In the All transactions registry, also known as the Historian, there is a record of each transaction that has taken place in the business network, including some transactions which are caused by system events, such as creating participants or assets. In the transaction registry, transactions are submitted and we then check that their effects have occurred by checking the resources that were changed.



**FIG 5.2.2.3** A record of all transactions that executed in the network

### 5.2.3 Hyperledger Composer REST Server

The REST server provides functions such as get, put, post, delete.



**FIG 5.2.3** EHRMS REST Server

### 5.3 Testing Phase

**System Testing**

This section details the test carried out on the EHRMS with the aim of observing whether the set functional and non-functional requirements have been met. It also gives explanations on the test used on the developed application. The test included:

**Functional Testing**

Functional testing is used to test whether the functions of the system are working as specified in the system requirement. Testing was applied on the several use cases to see if they performed as expected or not.

**Table 5.3 AddParticipant Test Case**

|  |  |
| --- | --- |
| Identifier | 001 |
| Test Case | Add new participant, either Doctor or Patient |
| Description | The administrator adds new participant into the system. The participant can be either Doctor or Patient. |
| Utilized Use Case | Add Participant |
| Results | Thenew Participant added to the network. |
| Pass/Fail | Pass |

**Table 5.3 Submit Transaction Test Case**

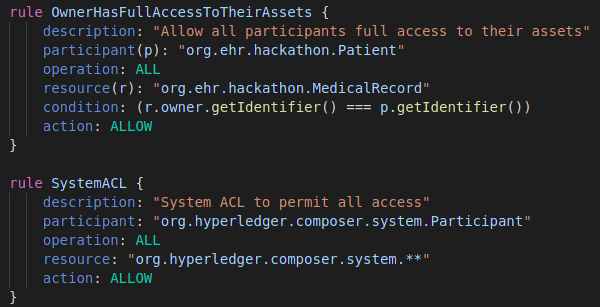
|  |  |
| --- | --- |
| Identifier | 001 |
| Test Case | Create Patient Medical Record. |
| Description | The administrator creates a new medical record using both the Doctor’s ID and Patient’s ID. |
| Utilized Use Case | Submit Transaction |
| Results | Thenew Medical Record created successfully. |
| Pass/Fail | Pass |

**Table 5.3 Proof of Existence Test Case**

|  |  |
| --- | --- |
| Identifier | 2eb0c6f6f9ac1ab2b6cb4ce0c9ae962acaa66d5430a9a1620284e6dd0312f883 |
| Test Case | Check the Existence of the added participant and created Medical Record. |
| Description | The administrator can search the Existence of the Medical Record using the Transaction ID. |
| Utilized Use Case | Search Medical Record, through the GET function. |
| Results | The Response body is displayed showing the created Medical Record. |
| Pass/Fail | Pass |

### 5.4 System Evaluation

This was done in order to ascertain whether the Electronic Health Record Management system preserves the privacy of the patient’s electronic medical records and also that the system is patient-centered. Administrator has administrative rights on all the peers for effective management of the system. The figure below shows the permissions granted to the Administrator for ease of operation performance throughout the system: -



**FIG 5.4** Access Control Rules

### 5.5 Deployment

Generally, there are four approaches for the implementation of the system in an organization. These are: Direct cutover, phased approach, pilot approach and the parallel approach. We greatly analyzed the four approaches to the system implementation and chose the phased operation:

### 5.5.1 Phased Approach

Phased operation works in different stages. It normally entails the implementation of the new system in modules. It is also a combination of the cutover and the parallel approach. We implemented EHRMS this way due to the fact that the system is new and it was therefore essential to implement it module by module till the last module of the system proved to be effective and well operational as required. Risk of errors or failures in this system may also have prompted me to use this method. Risks did not affect the entire system but to the single module or the several modules implemented so far. The third reason for its use is the cost involved in its implementation may be relatively lower compared to other approaches such as the direct approach which entails the overall implementation of the system at once.

Phased operation works in different phases or stages and it’s also a combination of direct cutover and parallel similar to pilot operation. But in this approach the entire system was not provided to everyone. (Morgan S, 2009).

### 5.6 Conclusions

The system requirement analysis stage clarified what functionalities were preferable in EHRMS. This in turn translated to development of the system that met the set objectives and the system requirements also. Hence, the set objectives were met with the implementation of the system. This was made possible by the methodology used, that is, the waterfall methodology, as it facilitated constant inflow of information from stakeholders that helped in coming up with a good design for the system.

# CHAPTER SIX

## 6.0 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

The research was done with the aim of solving the existing electronic health records problems by ensuring that the patient has full control of who can access his/her medical records. By the help of Blockchain technology and smart contracts, our system enables employment of a set of connectivity and timing functions to provide reasonable period of time for performing transactions and thus ensuring an authorized transaction is intended and also proof of existence of a given medical record hence provide a digital solution to this problem so that next time when patient visits a doctor there is no need of carrying medical file. We used Blockchain technology to store the patient records. This would ensure that the information remains secure while being decentralized across different peers. This chapter seeks to find if the set objectives were achieved and provide a review of the developed system in relation the current system, its advantages, benefits and some of its limitations.

### 6.1 Summary

From the literature review it can be clearly derived that the existing system uses Ethereum which is Blockchain-based framework, which is a public Blockchain, in Ethereum there is a poor scalability and data verification is not efficient. Without coordinated data management the health records are only distributed instead of being also cohesive; MedRec and Gem health network use Ethereum which is Blockchain-based framework. In our system we used Hyperledger fabric framework to improve efficiency in verification of data and to enhance scalability.

In order to maintain data privacy and security, various Blockchain participants have different  
role-based access rights to patients’ health records. The primary care health provider should a have full access to all patient’s health history, including patient’s identification. This provides the primary care with a cohesive view of the patient’s health records for a personalized care and artificial intelligence-based prognosis/diagnosis. For biomedical research purposes, another level of accessibility to patients’ data is defined. Biomedical researchers have access to anonymous data; anonymity is reinforced via consent rules executed by the Blockchain smart contracts. Implementation of smart contracts in healthcare can simplify things even better. Where invoking, record creation and validation is done on Blockchain.

Storing data to the Blockchain we add some value to enhance the quality of the data. Patient’s medical records are accurate, understandable and structured because unstructured data leads to inconsistencies and delays in the treatment process. Blockchain does not just help in decentralizing the data; it also gives the real-time data access, keeps the data confidential, handles high volumes of data efficiently, and also authenticate and authorize the data.

The constitution of Kenya 2010, under Article 31 recognizes the right to privacy. The Data Protection Act and the Human Rights Act provide a framework that governs a confidential usage and sharing of patients’ health records. These privacy and confidentiality laws can be reinforced using smart contracts and access control mechanism. Furthermore, according to the HIPAA security rule the integrity of health records should be ensured by employing proper encryption and authentication methods. In Blockchain, security is established by signing every health transaction digitally using encryption mechanisms and harsh-chaining of transactions to reinforce data integrity.

Our EHR management system on Blockchain is not meant to replace current systems completely, but to integrate and provide the features that are missing where all medical records can be stored on a permissioned Blockchain. Therefore we have built the EHRMS using the Hyperledger fabric framework and implemented it with Hyperledger composer tool, by integrating Hyperledger fabric we were able to record all medical transactions and access entire record of an individual using the Hyperledger composer REST server.

### 6.2 Conclusions

We were able to deploy a Blockchain-based EHRMS network and implement basic functionalities in the network. We successfully achieved the main objective of this research of creating a patient-centered EHRMS and protecting the privacy of the patients using the primary features of Blockchain that is cryptography/hashing and decentralization. We conclude that Blockchain technology is an innovative technology for implementing EHRMS and also it has the potential to help in the research and progress of healthcare in the near future. Hyperledger Fabric architecture allows varying access levels; users control who can view their records, how much they see and for what length of time. By endowing users, the future of healthcare can be built together. It will be a platform for various health applications to develop on; users will be able to admit for these applications and services which are fuelled by their medical data and secured by smart contracts.

### 6.2.1 Advantages of the Blockchain-Based Electronic Health Record Management System

1. Providing accurate, up-to-date and complete information about patients at the point of care.
2. Enabling quick access to patient records for more coordinated and efficient care.
3. Securely sharing electronic information with patients and other clinicians.
4. Helping providers more effectively diagnose patients, reduce medical errors and provide safer care.
5. Improving patient and provider interaction and communication, as well as health care convenience.
6. Enabling safer, more reliable prescribing.

### 6.2.2 Limitations of the Blockchain-Based Electronic Health Record Management System

1. While the Blockchain characteristics are suitable for implementing a healthcare system,   
   these mechanisms are still costly considering execution time and amount of data transferred for ledger update.

### 6.3 Recommendations

The idea and implementation can be further extended in the future by implementing various smart contracts to handle the advanced functionality of the EHRM system. Various sectors like billing, transportation, etc. can be added to the network to implement a full-fledged healthcare management system. To make it interactive it can be integrated with a web application. EHRMS can be made helpful for pharmacists in monitoring medical sales by adding them to the system as another participant.

# REFERENCES

**Ayesha Shahnaz, Dr. Usman Qamar and Dr. Ayesha Khalid**. (2019). “Using Blockchain for Electronic Health Records," IEEE Access. National University of Science and Technology (NUST), Islamabad, Pakistan.

**B. Narendra Kumar Rao, B. Bhaskar Kumar Rao and Vellingiri J.** (2019). “Blockchain-Based Implementation of Electronic Medical Health Record,” IJITEE. Department of IT, India, Volume-8.

**Sobia Yaqoob, Muhammad Murad Khan, Ramzan Talib, Arslan Dawood Butt, Sohaib Saleem, Fatima Arif and Amna Nadeem.** (2019). “Use of Blockchain in Healthcare,” IJACSA. Department of Computer Science, Government College University Faisalabad, Faisalabad, Pakistan, Vol. 10, No. 5.

**Eman-Yasser Daraghmi, Yousef-Awwad Daraghmi and Shyan Ming Yuan.** (2019). “Medchain: A Design of Blockchain-based System for Medical Records Access and Permissions Management,” IEEE Access. Department of Computer Systems Engineering, Palestine Technical University, Tulkarm, Palestine.

**Luhmann, R**. (1990). ResearchMethodology. Methods and Techniques (Second Revised).

**V.Vijayakumar, K.M.Sabarivelan, J.Tamizhselvan, B.Ranjith, B.Varunkumar.** (2019). “Utilization of Blockchain in Medical Healthcare Record using Hyperledger Fabric,” E-ISSN. International Journal of Research in Advent Technology, Vol.7, No.4.

**Fan K., Wang S., Ren Y, Li H, Yang Y.** (2018). “Efficient and secure medical data sharing via Blockchain”, Medblock.  
**P. Taylor.** (2016). Applying Blockchain Technology to Medicine Traceability.

**Harshini V M, Shreevani Danai, Usha H R, Manjunath R Kounte.** (2019). “Health Record Management through Blockchain Technology,” REVA University, Bangalore, India.

**Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang.** (2018). “Blochie: A Blockchain-based platform for healthcare information exchange”.

**Guy Zyskind, Oz Nathan, Alex sandy pentland.** (2015). “Decentralizing privacy: using Blockchain to protect personal data”.

**Dylan Yaga, Nik Roby, Karen Scarfone.** (2018). “Blockchain Technology Overview,” National Institute of Standards and Technology.

**D. S. Raymond Cheng.** (2018). “Smart Contract”.

**Pekka A., Outi, S., Jussi, R.** (2012). Waterfall software development methods: Review and Analysis. Vorimiehentie 5: VTT Technical Research Centre of Finland.

**W. J. Gordon and C. Catalini.** (2018) “Blockchain Technology for Healthcare : Facilitating the Transition to Patient-Driven Interoperability,” Comput*.* Struct*.* Biotechnol*.* J., vol. 16.

# APPENDICES

## APPENDIX A: Schedule

**Table 6.1: Schedule of the Electronic health Management Records system**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Expected starting date** | **Expected ending date** |
| Project title | 26th Nov 2020 | 2nd Dec 2020 |
| Project proposal | 4th Dec 2020 | 15th Dec 2020 |
| Introduction |
| Literature Review | 15th Dec 2020 | 19th Dec 2020 |
| Methodology | 20th Dec 2020 | 24th Dec 2020 |
| Coding | 26th Dec 2020 | 16th March 2021 |
| Testing | 18th March 2021 | 12th April 2021 |
| Implementation | 15th April 2021 | 2nd May 2021 |
| Documentation | 3rd May 2021 | 10th May 2021 |

## APPENDIX B: Gantt chart of the Electronic Health Records Management System

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *WEEKS*  *ACTIVITIES* | **Nov/2020** | | | | **Dec/2020** | | | | **Jan-Mar/2021** | | | | **Apr/2021** | | | | **May/2021** | | | | |
|  |  |  | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** |  |
| Proposal Writing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Feasibility study |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requirement analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Environment Setup |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation |  | | | |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |

## APPENDIX C: Budget

The system was developed in budget ranges which significantly improved its responses. Every project has a range of options and degree of customization that significantly impacts the final price. By leaving the budget open. The goal in this project is that to figure the best work within a certain budget range.

**Table 6.2: Budget of the Electronic Health Records Management system**

|  |  |  |
| --- | --- | --- |
| **ITEM** | **DESCRIPTION** | **ESTIMATED COST** |
| LAPTOP(HP)  {DUAL BOOT} | Intel(R) Core(TM) i5-2520M CPU @2.50GHz,8.00GB RAM,500GB HARD DISK CAPACITY | KSH 35,000.00 |
| SECONDARY STORAGE DEVICE | 500GB RE MOVABLE DISK | KSH 6,000.00 |
| RESEARCH | TRAVELLING | KSH 2,500.00 |
| STATIONERY | full scarps, pens, | KSH 400.00 |
| CONECTIVITY | Internet(Data Bundles) | KSH 7,000.00 |
| TOTAL |  | KSH 50,900.00 |