VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"Securing EHRs Using Blockchain Technology"

Submitted in Partial fulfilment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

AARATHI NAIR (1CR16CS002)

S.M. DIVYAVANI (1CR16CS138)

SABITA G. (1CR16CS141)

SOPHIA MARIA AUGUSTINE (1CR16CS164)

Under the Guidance of,

Prof. SAGARIKA BEHERA

Assoc. Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Securing EHRs Using Blockchain Technology"** carried out by **Ms. AARATHI NAIR** , USN **1CR16CS002**, **Ms. S.M. DIVYAVANI**, USN **1CR16CS138**, **Ms. SABITA G**, USN **1CR16CS141, Ms. SOPHIA MARIA AUGUSTINE**, USN **1CR16CS164,** bonafide students of CMR Institute of Technology, in partial fulfilment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraya Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Prof. Sagarika Behera	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Associate Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the examiners	Signature with date
1.	_____
2.	_____

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**Securing EHRs Using Blockchain Technology**" has been successfully completed under the guidance of **Prof. Sagarika Behera**, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university

Place:

Date:

Team members:

AARATHI NAIR (1CR16CS002) _____

S. M. DIVYAVANI (1CR16CS138) _____

SABITA G. (1CR16CS141) _____

SOPHIA MARIA AUGUSTINE (1CR16CS164) _____

ABSTRACT

At present, Electronic Health Records are stored on centralized databases that make the medical data remain largely non-portable. Database storage can expose the data to integrity issues and accuracy problems, especially in the medical field. Blockchain which is also a popular topic nowadays, can ensure data security while using the distributed and decentralized topology. The data is also protected by only allowing the permissioned personals to whom access was granted by the Blockchain Technology. The main issue in the medical sector is the synchronization of patients' data. Each medical institution has their own individual databases, but the patients may not go to a particular clinic all the time. Therefore, it is tedious to get the updated record of the patient to provide accurate diagnosis.

In this system, patients themselves retain control over 'who can access their data'. The basic architecture includes creating a network of trusted data repositories, the access to which are determined by a set of smart contracts. These contracts are stored on a distributed ledger. It increases the patient's control while retaining a measure of the privacy of both the data content and the source.

Utilizing the work of this proof of concept, the framework of Blockchain 4.0 Hyperledger Fabric will be implemented to solve the above stated problem. A Hyperledger Fabric Composer will be used to set up the development environment, running environment and playground for running the Blockchain. Besides these, Channel and Chaincode will also be used in this system. At the end of the project, a system will be designed for usage by the medical sector.

ACKNOWLEDGEMENT

Success is the result of hard work, determination, proper guidance and help. With such determination and guidance, we have been able to complete this project on time and in an effective manner. We take this opportunity to thank, all those who extended a helping hand, whenever we needed one.

We take this opportunity to express our sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing us a platform to pursue our studies and carry out our final year project.

We have a great pleasure in expressing our deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

We would like to thank **Dr. Prem Kumar Ramesh,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

We consider it a privilege and honour to express our sincere gratitude to our guide **Prof. Sagarika Behera, Associate Professor,** Department of Computer Science and Engineering, for her valuable guidance throughout the tenure of this review.

We would also like to express our sincere gratitude to Dr**. Jhansi Rani P, Professor,** Department of Computer Science and Engineering, for her valuable insights and critical suggestions throughout the tenure of this phase.

We also extend our thanks to all the faculty of Computer Science and Engineering Department who directly or indirectly encouraged us.

We are also grateful to our parents and friends for all their moral support they have given us during the completion of this work.

Last but not the least, we thank the almighty God for his abundant blessings and guidance on us throughout this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

BCT	**-**	**Block Chain Technology**
EHRs	**-**	**Electronic Health Records**
EMRs	**-**	**Electronic Medical Records**
MSP	**-**	**Membership Service Provider**

CHAPTER 1

INTRODUCTION

Electronic Health Records or EHRs contain health data of patients which is sensitive in nature, and it is utilized by many authorized users within and across medical institutions. This can lead to integrity issues and accuracy problems.

Hence, the use of blockchain as a technology for secure retrievals and transactions ensures privacy and security of the patient's health data during several different types of clinical practices. We utilize the Hyperledger Fabric, which leverages container technology to host smart contracts called 'Chaincode' for applying specified code when certain conditions are fulfilled

1.1 Relevance

Blockchain Technology provides a distinctive opportunity to support the healthcare sector. Our EHR solution enables the healthcare professionals to access an entire medical history of the patient in order to provide accurate medication and treatment. This solution then records interactions with this data in an auditable, transparent, and secure way on the system's distributed ledger. This solution is a platform for individuals to use that complements and improves the user's overall experience. It also improves the care for patients by placing them at the centre of the digital transformation of healthcare.

A Hyperledger Fabric is used that is based on a permission network, which indicates that all the participants are required to be authenticated in order to participate and transact on the blockchain. Here, all the members are enrolled into the Membership Service Provider (MSP) to create a permissioned network. MSP is able to manage the role and permission of each user in order to use the application. Therefore, all the users are authenticated and identified by the system while using it. Hence, the users who request and operate into the medical records must be granted permission in order to access certain information.

1.2 Problem Statement

The purpose of this project is to secure the electronic medical records with the help of Blockchain Technology. This system is based on a blockchain that contains smart contracts (chaincode) for defining user identity and distribution specifics. We create a network of trusted data repositories, the access to which are determined by a set of smart contracts that are stored on a distributed ledger.

The Hyperledger blockchain network is permission-based and requires users to check in to use it. The access control rules in the Hyperledger modelling defines the various permissions. All the members are enrolled into the Membership Service Provider(MSP) to create a permissioned network. MSP manages the roles and permissions of each user in order to use the application.

Medical information is highly sensitive in nature in a social as well as legal sense, Hence, a permissioned blockchain like Hyperledger Fabric helps to maintain privacy which is highly required for such an application. The Hyperledger Fabric is a framework for blockchain infrastructure development for distributed ledger solutions which provides confidentiality, resilience, flexibility and scalability. This is a better solution for managing access to health records, as it accommodates multiple layers of permission, which means that the owner of a set of data has control over which parts of their data is accessed.

1.3 Objectives

- To develop a Blockchain Network to securely store medical records such the data become more useful for research institutions: The traditional method of storing medical records is to keep the data in a database server. Sometimes the data will be inaccessible if the database server is down or the data will be exposed to attackers if the database server is vulnerable. The feature of Blockchain will assure that the blocks in ledger are not modifiable
- To synchronize the records between each system member. Each medical institution will keep their own copy in an independent database, so it is difficult

to trace the latest update of the records. Blockchain is a decentralized system, it stores data distributed on multiple machines.

- Secure, immutable and decentralized EHR database with patient as data owner of EHR data
- Can act as a foundation for the DISHA(Digital Information Security in Healthcare, Act) bill in India
- To provide a single point of access for the full medical history of a patient
- Easier prescription verification for pharmacists can be enabled
- Transparency can be increased
- Insurance Frauds can be avoided.

1.4 Scope of the Project

This project is to develop a Blockchain by implementing Hyperledger Fabric for protecting medical records of patients from security and integrity issues. Here, we focus on the confidentiality and integrity of data, whereby all medical records must be up to date and synchronized. In the Hyperledger Fabric, all the members are enrolled into the Membership Service Provider to create a permissioned network. The MSP is able to manage the role and permission of each user in order to use the EHR system. Therefore, all users are authenticated and identified by the system while using it.

- Ready access to a lot of all-inclusive, updated patient info, quick reliable and secure information.
- 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.
- Reduced patient info management, paperwork, duplication work and other forms-less time spent finding identified notes, X-rays, admission or discharge information.

1.5 Methodology

- In the approach to securing EHRs, blockchain technology is pivotal in maintaining confidentiality, integrity, and easy sharing of health data. It is based on the decentralized nature for the records management system, and provides a detailed transaction history which is not modifiable. This move is to ensure better portability of the sensitive medical data across several treatment and research institutions.
- The blockchain network contains not the actual health data but instead the details of the block that correspond to the particular health information. This gives better portability of the sensitive medical data across several treatment and research institutions due to this synchronised implementation of blockchain.
- Since the patient is the data owner, he has the authority to authorize the use of their personal data as required. This transfer of control from the medical institution to the patient is possible due to the implementation of blockchain.

1.6 Evolution of Blockchain Technology

The evolution of blockchain technology across its generations is discussed in detail below.

Blockchain First Generation: Crypto-Currency. The first generation blockchain deals with the implementation of distributed ledger technology (DLT) in the form of its first application as cryptocurrencies. Financial transactions that were performed with Bitcoin is one such example for cryptocurrency. It was primarily used as the currency for internet based financial transactions, which can be considered equivalent to internet money.

Blockchain Second Generation: Smart Contracts. 2nd Gen Blockchain Technology gave rise to smart contracts. Smart contracts are program files that specify the conditions for verifications and authorizations. They are pre-defined in nature and it is impossible to tamper with its contents. This allows for smart contracts to ensure

security and fraud prevention, which can act as a result from its transparent nature. Ethereum Technology utilises such smart contracts under the blockchain technology.

Blockchain Third Generation: DApps. DApps refers to decentralized applications that do not have a centralized infrastructure. The backend structure of DApp is decentralized in nature wrt storage and communication with front end infrastructure is also achieved if used technologies like Ethereum. Decentralized nature of the application allows for the generation of a peer to peer network which is the core concept of the blockchain technology

Blockchain Fourth Generation: Meeting Current Industry demands. Blockchain technology in fourth gen has been made usable to meet business demands. The present industry deals with many business cases of ERP, automation, legacy systems etc. However, there is an emerging demand for an increasing degree of trust, privacy and protection in the present industry. When integrating blockchain with various IT systems, a business integration, the added security layer implemented by blockchain will be the answer to the previously mentioned demands. Supply chain management, financial transactions, IoT data collection, health data management and confidential asset management are just a few examples of domain areas that are trusted applications developed utilising blockchain.4th Gen Blockchain is basically the previous generation utilised in real-life business scenarios.

1.7 Overview of Blockchain Technology

Blockchain structure: A blockchain is a growing list of records, called “blocks” that are generally linked to each other using cryptography. Blocks are data structures with the intent of bundling transaction sets and being spread to all network nodes. These blocks are established by miners. Fig 1.1 describes the structure of a block in a blockchain network. Each of these blocks contains a cryptographic hash of the previous block, a timestamp, a nonce and transaction data which is generally represented as a “Merkle tree”. The hash of a previous block (Prev_hash) in a sequence is a tamper-proof sequence because as a function of the design, a hash is very sensitive. So, to change any variable of any one of the hashes in a given block would cause a domino effect, altering all of the previous transactions in the block.

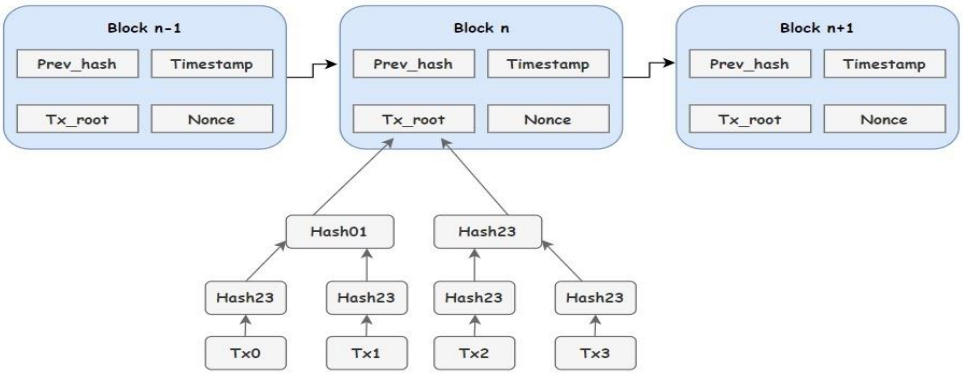

Fig 1.1: Structure of Blockchain Network

A nonce is a "number only used once." For a Bitcoin block, a nonce is a 32-bit (4-byte) numerical string. This is generally called a "target hash" that miners solve for. They adjust the value of the nonce so that the hash of the block is less than or equal to the current target of the network. The presentation of the block with the correct nonce value constitutes a proof-of-work, as this iterative calculation requires time and resources. A timestamp is the time of the creation of a particular block.

A Merkle tree lists all transactions in a block by generating a digital fingerprint of the entire collection of transactions, enabling a consumer to verify if a transaction is included in a block or not. These are generated by hashing pairs of nodes continuously, until just one hash remains. (this is called the Root Hash, or the Merkle Root). They are designed from the ground up, from individual transaction hashes (known as the Transaction IDs). Each leaf node is a hash of transactional data, and each non-leaf node is a hash of preceding hashes. Merkle trees are binary, and they need an equivalent number of leaf nodes. If the number of transactions is not even, repeat the last hash once to establish an even number of leaf nodes.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the process in which a complete and comprehensive review is conducted encompassing both the published and unpublished work from other alternative sources of information. This review is conducted in the domains of specific interest to the person or researcher. Further, the results of this process are documented.

This entire process comes in aid of the researcher to address the important and relevant aspects of the research that had not been addressed prior to the conduction of this research. Therefore, it can be understood that the conduction of literature survey is necessary for the process of gathering secondary data for the research which might prove to be extremely helpful in the research and also designing the architecture of the project. There can be multiple reasons behind the purpose of conducting literature survey.

2.1 PAPER 1:

Title: [2] Dinh C. Nguyen, Pubudu N. Pathirana, Aruna Seneviratne **"**Blockchain for Secure EHRs Sharing of Mobile Cloud based E-health Systems" *School of Engineering, Deakin University, Waurn Ponds, VIC 3216, Australia.*

Context:

This paper presents a EHRs sharing framework that combines blockchain and the decentralized Interplanetary File System (IPFS) on a mobile cloud platform to design a trustworthy access control mechanism using smart contracts to achieve secure EHRs sharing among different patients and other medical providers. The system model uses a private Ethereum blockchain deployed on AWS.

It also presents a prototype implementation using the Ethereum blockchain in a real data sharing scenario on a mobile app with Amazon cloud computing. The Empirical results show that this proposal provides an effective solution for reliable data exchanges on mobile clouds while preserving sensitive health information of the patients against potential threats. This system evaluation and security analysis also demonstrates the

performance improvements in lightweight access control design, minimum network latency with high security and data privacy levels, compared to existing data sharing models.

2.2 PAPER 2:

Title: [3] Haider Dhia Zubaydi, Yung-Wey Chong, Kwangman Ko, Sabri M. Hanshi and Shankar Karuppayah "A Review on the Role of Blockchain Technology in the Healthcare Domain" *School of Computer and Information Engineering, Sangji University, Gangwon 220-702, Korea, 15 June 2019.*

Context:

This paper explains the basic concepts of blockchain technology and its hurdles in their adoption in the healthcare domain.

The concept of centralized, decentralized and distributed architectures is briefly explained with the help of relevant examples and important applications. It gives an overview of how the consensus algorithms are used to secure blockchains.

It also gives a brief review on the latest implementations of blockchain technology in healthcare as patient information is very case-sensitive and must be stored and shared in a secure and confidential manner. Finally, a new case study of a blockchain-based healthcare platform is presented addressing the drawbacks of current designs, followed by recommendations for future blockchain researchers and developers.

2.3 PAPER 3:

Title: [4] Sara Rouhani, et al. MediChain: A Secure Decentralized Medical Data Asset Management System

Context:

Inability to access medical data (produced by a physician) in a timely and efficient manner is a serious problem in the delivery of health care worldwide. Hyperledger Fabric is an approved Blockchain platform (as in, it is a permissioned network); all nodes in the network have an identity. A Membership Services Provider (MSP) uses a

public key to issue each participant a cryptographic certificate. The immutable chaincode smart contract helps with privacy and security. Hence this allows trusted transactions between the known participants. The most important characteristics of the Chaincode smart contracts are immutability, durability, and reliability. Ethereum, an open source, blockchain-based distributed computing platform is used to create this permissioned network.

2.4 PAPER 4:

Title: [5] Guang Yang, et al. "A Blockchain-based Architecture for securing electronic health record system", 13 July 2019

Context:

The Blockchain consists of distributed ledger which is completely immutable. A ledger is nothing but a chain of blocks which is time-stamped. The ledger is basically a chain of blocks linked by cryptographic hash functions, where each block has the hash value of the previous block. A new block can be added to the existing blockchain. Every time a new block has to be added the block can be appended to the existing chain. Due to the sensitivity of the data, the access control is of utmost importance. Due to the inherent characteristics of the blockchain, it has a great potential in the field of managing Electronic Health records.

2.5 PAPER 5:

Title: [6] Asad Ali Siyal, et al.Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives, 2 January 2019

Context:

Some of the key functionalities of a Blockchain Technology are that they are decentralized, Transparent, Immutable, Autonomy, Open Source and Anonymity. Blockchain Technology has played a vital role in biomedical research and clinical domains. The Blockchain Technology has potentially been used in the fields of Neuroscience, Biomedical, Pharmaceuticals, Clinical and Genomics Medicine.

Blockchain brings in a lot of honesty to the doctors in the patient health history point of view, as it is immutable. Hence the data remains authentic and quality source-documented in real time, which in turn reduces errors. The opportunities of Blockchain in Healthcare are Transparency, Reduced Transaction Time, Security and Cost Efficiency. The framework helps engaging individuals and hence ultimately improving the quality of life.

2.6 PAPER 6:

Title: [8] Asaph Azaria, et al. MedRec: Using Blockchain for Medical Data Access and Permission Management, 2016 2nd International Conference on Open and Big Data

Context: In this project, block chain technology is used to implement a decentralized system for the maintenance of electronic medical records (EMRs). It is suggested that the blockchain could be considered as a set of decentralized compute resources, each such resource being an individual entity which could transcend between two states through cryptographic transactions. These entities were associated with a logic that would distinguish a transaction from being valid or not. For every valid transaction, the blocks would undergo a transformation, such a transformation is known as 'smart-contracts'. Ethereum blockchain was employed to implement smart-contracts, thereby developing a record management system which could furnish medical information to the patients as well as all medical service providers. Blockchain properties were used to design the system in such a manner that the data would remain confidential, tamper-proof and easily accessible. This design facilitated data miners with large pools of anonymous medical data, which could be used to interpret the wide array of medical treatment patterns and socio-economic demand patterns.

2.7 PAPER 7:

Title: [7] Nchinda Nchinda, et al. MedRec: A Network for Personal Information Distribution, 2019 International Conference on Computing, Networking and Communications(ICNC): Internet Services and Applications

Context:

MedRec is a distributed system which gives an individual user control over his/her identity and the distribution of their information. Fig 2.1 shows the basic system architecture of the MedRec system. Blockchain is used to manage a set of smart contracts that define who can access the records. Contrary to the web, the servers here are linked and maintain a common blockchain. These servers are the managerial members of the network. A 'proof of work' algorithm is used to maintain the integrity of the blockchain. Here, all the transactions on the public Ethereum network are stored by the full nodes present on the network.

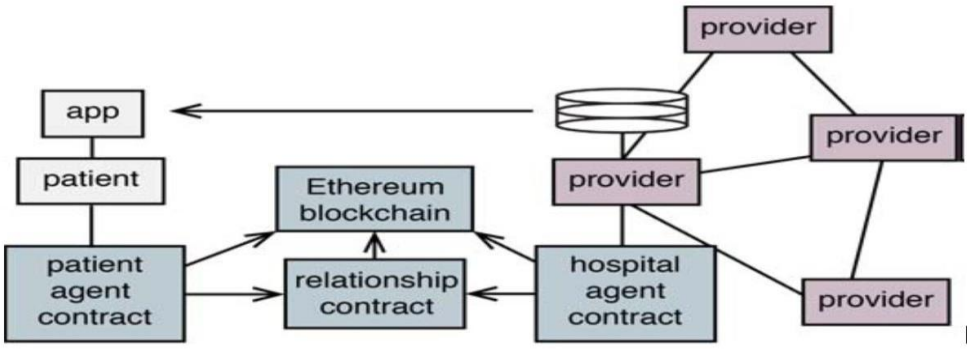

Fig 2.1 Architecture of MedRec

This system is like a wallet that contains the Ethereum private key. This key is stored in a vault that is password protected and it is unlocked when a user logs in. This system is designed to support multiple accounts and multiple users can use the same hardware without compromising their individual security. As soon as the patient logs in, data is retrieved from each provider that they have a relationship with and is visualized in the interface.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

This chapter describes the requirements for the project. The Software Requirement Specification (SRS) is explained in detail, which includes overview of dissertation as well as the functional and non-functional requirements of this dissertation. An SRS document describes all data, functional and behavioural requirements of the software under production or development.

SRS is a fundamental document, which forms the foundation of the software development process. It is the complete description of the behaviour of a system to be developed. Requirement Analysis discusses the conditions to be met for a new or altered product. Requirement Analysis is critical to the successful development of a project. Requirement must be documented, measurable, testable, related to in identified business needs or opportunities, and defined to a level of detail sufficient for system design.

The SRS functions as a blueprint for completing a project. The goal of preparing the SRS document is to:

- Facilitate communication between the customer, analyst, system developers, maintainers.
- To form a foundation for the design phase.
- Support system testing facilities.
- Controlling the evolution of the system.

3.1 Functional Requirements

- System role and actions: There are 3 different roles designed in our system, each role has different permission to perform different tasks. The first role is patient, only allowed to view the medical records, medical history and modify their profile. The second role is for researcher, they are able to view and request for medical records. The next role is doctor, allowed to view, request, and update the patient's medical record.

- Update medical records to Blockchain: The medical records are added into the blockchain as a transaction. Only the peers with authorized permission within the organization can perform the records update action. For example, the doctor can update the patient's medical profile after consultation.
- Retrieve medical records to database server: Retrieve the medical records from blockchain and API calls to display the records result to the patient, doctors or researchers.
- Design chaincode to allow data retrieval from blockchain network: Chaincode acts as an intermediary between Blockchain and web server. All the requests that come from users are needed to pass through the chaincode. Therefore, chaincode is able to verify the request. If the requirements stated in chaincode are fulfilled by the requester, then the requester can receive the medical records successfully from the Blockchain network.

3.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties. Non-functional requirements for this system are specified as follows:

- Responsiveness of the system needs to be appropriate since timely retrieval of sensitive health data is essential.
- The software utilised should be portable so that medical institutions can easily expand to their inter-connected hospitals, spread across locations.
- Privacy of sensitive data should always be maintained and must not be misused in any manner.
- Blockchain is a technology that relies on the growing network of peers. Systems should be developed in a manner to handle such a dynamic requirement.

- Researchers requesting for data must be from authorized sources. Proof of consent is a responsibility that is borne by the medical institution where the health data is generated.

3.3 Hardware Requirements

- Server-Side Requirements:
- Processor: Core 2 duo or above
- Hard Disk: 5GB
- RAM: 2GB
- Client-Side Requirements:
- Processor: Dual Core
- Hard Disk: 2GB
- RAM: 2GB

3.4 Software Requirements

3.4.1 Ubuntu OS Version:18.04

Ubuntu is a free and open-source Linux distribution based on Debian. Ubuntu is officially released in three editions: Desktop, Server, and Core for the internet of things devices and robots. All the editions can run on the computer alone, or in a virtual machine. Hyperledger projects can be executed only using ubuntu as the operating system.

3.4.2 Hyperledger Fabric

Hyperledger Fabric is a blockchain framework implementation and one of the Hyperledger projects hosted by The Linux Foundation. We are using it as the foundation for developing our application with a modular architecture, Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Hyperledger Fabric leverages container technology to host smart contracts called “chaincode” that comprise the application logic of the system.

3.4.3 Hyperledger Composer

Hyperledger Composer is a set of collaboration tools for building blockchain business networks that make it simple and fast for business owners and developers to create smart contracts and blockchain applications to solve business problems.

Built with JavaScript, leveraging modern tools including node.js, npm, CLI and popular editors, Composer offers business-centric abstractions as well as sample apps with easy to test DevOps processes to create robust blockchain solutions that drive alignment across business requirements with technical development. We used it to interact with our underlying blockchain.

3.4.4 JavaScript

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded. We used it to write our chaincode for the blockchain.

3.4.5 node.js

node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Node.js enabled us to use JavaScript for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

3.4.6 npm

npm is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. We used it to get packages for JavaScript.

3.4.7 HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. We used it to create our user interface.

3.4.8 jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. We used it to send POST and GET ajax API calls to our back end.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

This chapter gives an overview of the proposed system along with defining the system architecture and UML diagrams such as use-case and class diagrams. A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

4.1 Overview of System Design

Hyperledger Fabric is the chosen technology for this project since it has enhanced features in terms of security by providing a provision for permissioned participation. Moreover, it is feasible to develop industry-oriented applications by utilising Hyperledger as compared to the Blockchain 2.0. Besides that, SHA-256 is used to hash the medical records and RSA is used to encrypt the hashed value before attached to the Blockchain.

Different components of the blockchain network use certificates to identify themselves to each other as being from a organization. The mapping of certificates to member organizations is achieved via a structure called a Membership Services Provider (MSP). There's a resource, the network N, accessed by a set of users defined by a Certificate Authority (CA), who have a set of rights over the resources in the network (N) as described by policies contained inside the network configuration. All of this is made real when we configure and start the ordering service node (O).

Hyperledger Fabric users often use the terms smart contract and chaincode interchangeably. In general, a smart contract defines the transaction logic that controls the lifecycle of a business object contained in the world state. It is then packaged into a Chaincode which is then deployed to a blockchain network. While smart contracts are for governing transactions, chaincode governs how smart contracts are packaged for deployment. Golang or Java programming languages are used to program the

chaincode. The Chaincode will be used to verify the request to allow retrieval of the medical records from the blockchain network.

Mining is the process that let the members of the network arrive at a consensus on the contents of the Blockchain. The orderer determines which transactions to add to the Blockchain and in what order. All members of the network rely on what the orderer says.

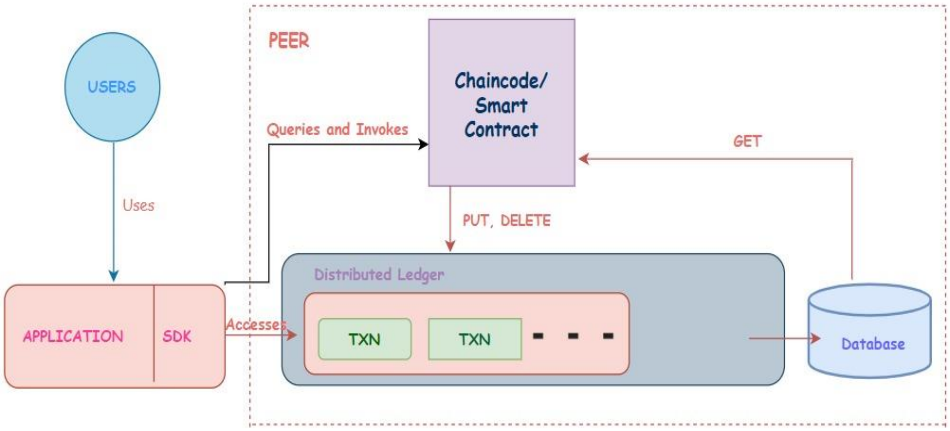

Fig 4.1: Generalized Blockchain Application diagram

Utilising the Hyperledger Fabric Framework by the Linux foundation, we can easily develop industry oriented blockchain applications. The general flow of blockchain infrastructure applications created using blockchain is discussed in this section. The fig 4.1. describes the structure and flow of a blockchain network having a business application. The set of users would interact with the main user interface and doing so will end up invoking some type of query to the database. The SDK or Software Development Kit will verify the global state of the blockchain, and query will be submitted to the blockchain through RESTful service-based APIs. The blockchain network will inform other peers in the network for consensus. After the successful consensus, a transaction will be submitted to the blockchain in the distributed ledger that contains multiple other transactions executed , following which a subsequent key-value pair is generated or modified depending on the type of the request.

Whenever a new user is created, a 128-bit token is generated, and this token is also stored in the database in the backend. Whenever the user logs in the application, the credentials are verified, and its corresponding token is fetched from the backend via RESTful API. Now with every query, the application passes the users tokens and, in the backend, utilising the RESTful API, the token is verified, and the identity of the user is known. This acts as the first layer of security.

When the actual data is being pulled from the blockchain, the user identity is verified with the certificate that was issued when the user was created in the blockchain. This acts as the second layer of security.

4.2 System Architecture

In general, a blockchain application will have different types of participants. Any of these users will interact with the main application user interface (UI). Next, the user will invoke some query. SDK will verify the global state of the blockchain, and query will be submitted to the blockchain through RESTful service-based API. Blockchain will send the request to other peers for consensus. After the successful consensus, a transaction will be submitted to the blockchain and the subsequent key-value pair will be created or modified according to the request.

The fig 4.2 describes the architecture of the proposed system. The application built considers three types of users: Doctors, Patients and Researchers. The users will use the web UI application to perform view, request or update to the data. Once the web server receives the request from users, then it will go for verification from Chaincode. The web server is only allowed to access the Blockchain after it is verified by Chaincode. After that, the Blockchain will check the participant role and permission that will be assigned by the Membership Service Provider (MSP). All the members in the Blockchain must be identified in order to perform any actions. If the user permission is allowed, then the Blockchain will perform retrieve or update to the medical record according to the user's request received by Blockchain.

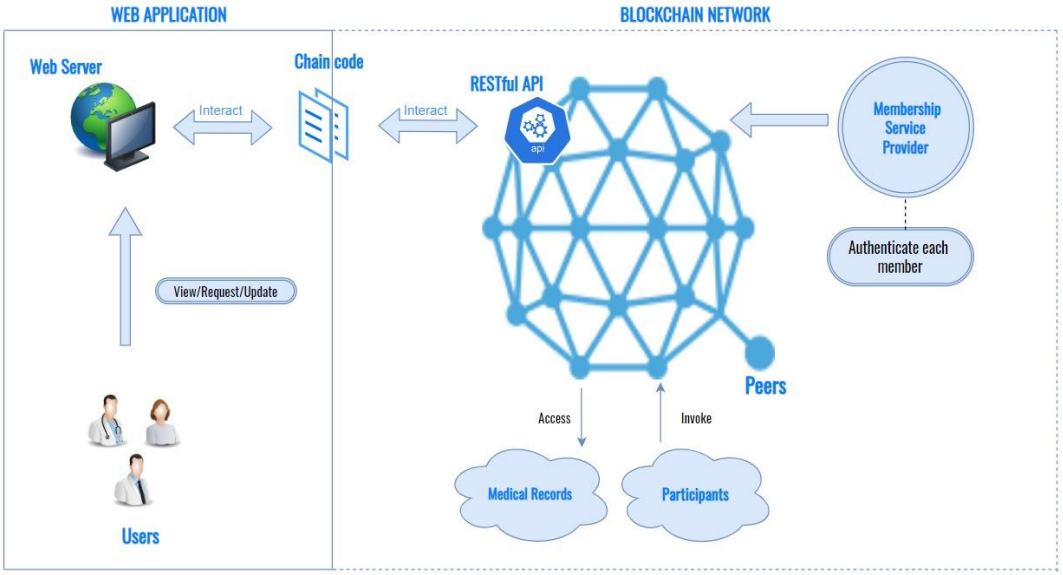

Fig 4.2 System Architecture

This proposed architecture is a proof of concept for the healthcare industry, where multiple hospitals with already existing large centralized databases can expand their current systems by incorporating such high level of security technology pertaining to transactions of assets like sensitive health data.

4.3 Use-Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour. A use case diagram is usually simple. It does not show the detail of the use cases:

- ❖ It only summarizes some of the relationships between use cases, actors, and systems.

- It does not show the order in which steps are performed to achieve the goals of each use case.

The use-case diagram corresponding to the project is depicted in Fig 4.3

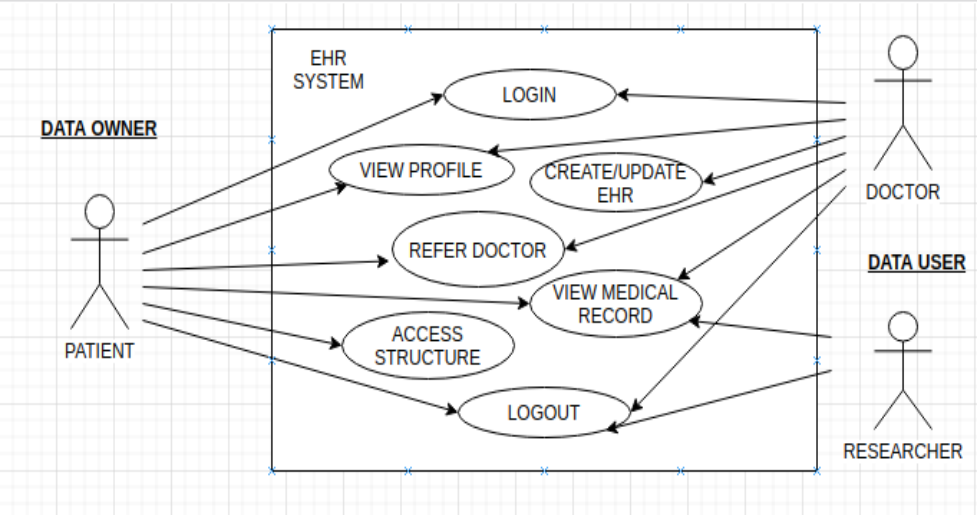

Fig 4.3 Use-Case Diagram

4.4 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Purpose of Class Diagrams is as follows :

- Shows static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribed by UML
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business perspective.

A UML class diagram is made up of:

- A set of classes and
- A set of relationships between classes

The class diagram corresponding to the project is depicted in Fig 4.4.

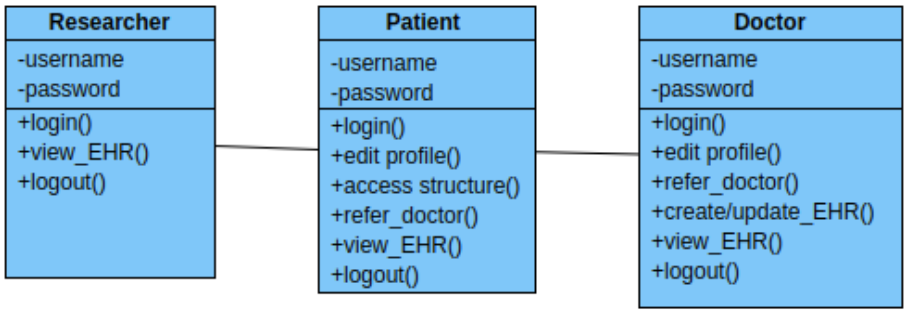

Fig 4.4 Class Diagram

CHAPTER 5

IMPLEMENTATION

This chapter covers the stages of implementation for the proposed system to successfully deploy the blockchain network using Hyperledger Fabric Composer which is under the umbrella of Hyperledger project

5.1 Blockchain Setup

Hyperledger Fabric is under the umbrella of Hyperledger project, which supports collaborative development of distributed ledgers. A Distributed ledger is a peer-to-peer network, where each node maintains a copy of a ledger state. Ledger state is a storage of all transactions that took place in the network. A transaction cannot be modified later, therefore there is immutability of the distributed ledger. For consistency support (each node in the network must store the same transaction in the same order) we need a special mechanism called consensus. It keeps transactions synchronous in the network. Membership services combine elements of Public Key Infrastructure (PKI) and decentralization/consensus to transform a non-permissioned blockchain into a permissioned blockchain.

A Hyperledger Fabric architecture also offers the security of channels, meaning that a different ledger can for different channels, which can store different types of records. This framework can be tuned to business requirements and does not need to establish a cryptocurrency. The HLF ledger has the following core components, Certificate authority (MSP), Chaincode Containers, nodes/peers, Ordering service, Channels, and Shared Ledger. When a user enrols into the network through MSP (Membership service provider) and submits a proposed transaction to the endorsing peer, the peer executes the chain code (smart contract), endorses it and returns the transaction to the user/client. The user then submits the proposal to the orderer. Orderer is the automated service that bundles in all transactions. The orderer service verifies the received transaction and adds this transaction along with other transactions, sorts the transactions, and creates a block of transactions. Then the peers validate the transaction in the returned block and commit the ordered block to the ledger. Membership service provider (MSP) is a Certificate authority, which provides

authentication, registration, and certificate generation services to its users. A Membership service provider also generates a public key/Private key pair for its members. Each member of the network should present a digital certificate to join the HLF network. MSP will create these certificates for all its members in the network.

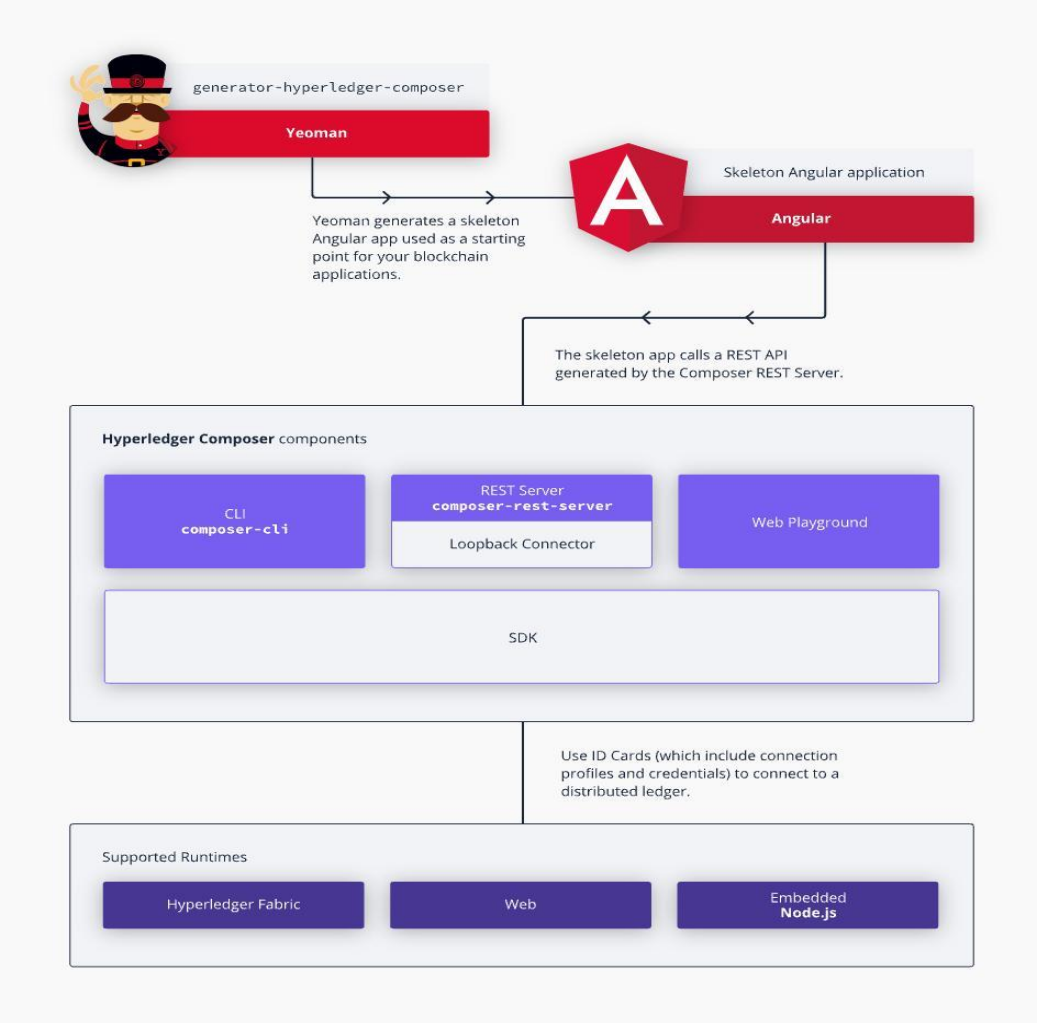

Fig 5.1 Typical Solution Architecture

To build the blockchain application utilising Hyperledger Fabric, all the prerequisites and the development environment must be installed. The steps for the same can be referred to the official Hyperledger documentation.

A typical Hyperledger based solution architecture is shown in Fig 5.1 is from the official Hyperledger documentation - “Hyperledger Composer enables architects and developers to quickly create "full-stack" blockchain solutions. I.e. business logic that runs on the blockchain, REST APIs that expose the blockchain logic to web or mobile applications, as well as integrating the blockchain with existing enterprise systems of record.”

5.2 Building Blockchain Web Application

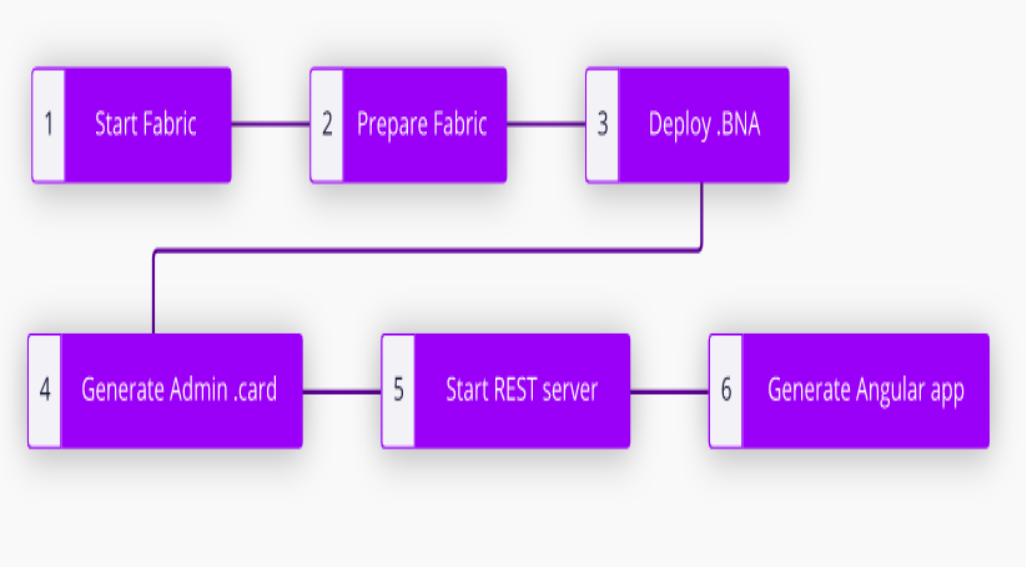

Fig 5.2 The flow of building blockchain web-apps

Fig 5.2 describes the steps (1-6) to build an angular based blockchain web application which is provided by the Official Hyperledger Documentation. In order to build and deploy the Business Network Archive file (.bna), the development environment must be set up which a can be found here - Developer Tutorial

After setting up the development environment, the following steps can be followed to build and deploy the .bna file, to have a functioning Blockchain Application.

5.3 Key Concepts in Hyperledger Composer

5.3.1 Blockchain State Storage:

All transactions submitted through a business network are stored on the blockchain ledger, and the current state of assets and participants are stored in the blockchain state database. The blockchain distributes the ledger and the state database across a set of peers and ensures that updates to the ledger and state database are consistent across all peers using a consensus algorithm.

5.3.2 Connection Profiles

Hyperledger Composer uses Connection Profiles to define the system to connect to. A connection profile is a JSON document the is part of a business network card. These profiles are usually provided by the creator of the system they refer to and should be used to create business network cards in order to be able to connect to that system.

5.3.3 Assets

Assets are tangible or intangible goods, services, or property, and are stored in registries. Assets can represent almost anything in a business network, for example, a house for sale, the sale listing, the land registry certificate for that house, and the insurance documents for that house may all be assets in one or more business networks. Assets must have a unique identifier, but other than that, they can contain whatever properties you define. Assets may be related to other assets or participants.

5.3.4 Participants

Participants are members of a business network. They may own assets and submit transactions. Participant types are modelled, and like assets, must have an identifier and can have any other properties as required. A participant can be mapped to one or multiple identities.

5.3.5 Identities

An identity is a digital certificate and private key. Identities are used to transact on a business network and must be mapped to a participant in the business network. A single identity is stored in a business network card and if that identity has been mapped to a participant, it allows the user of that business network card to transact on a business network as that participant.

5.3.6 Business Network Cards

Business network cards are a combination of an identity, a connection profile, and metadata, the metadata optionally containing the name of the business network to connect to. Business network cards simplify the process of connecting to a business network and extend the concept of an identity outside the business network to a 'wallet' of identities, each associated with a specific business network and connection profile.

5.3.7 Transactions

Transactions are the mechanism by which participants interact with assets. This could be as simple as a participant placing a bid on a asset in an auction, or an auctioneer marking an auction closed, automatically transferring ownership of the asset to the highest bidder.

5.3.8 Queries

Queries are used to return data about the blockchain world-state. Queries are defined within a business network and can include variable parameters for simple customization. By using queries, data can be easily extracted from your blockchain network. Queries are sent by using the Hyperledger Composer API.

5.3.9 Events

Events are defined in the business network definition in the same way as assets or participants. Once events have been defined, they can be emitted by transaction processor functions to indicate to external systems that something of importance has

happened to the ledger. Applications can subscribe to emitted events through the composer-client API.

5.3.10 Access Control

Business networks may contain a set of access control rules. Access control rules allow fine-grained control over what participants have access to what assets in the business network and under what conditions. The access control language is rich enough to capture sophisticated conditions declaratively, such as "only the owner of a vehicle can transfer ownership of the vehicle". Externalizing access control from transaction processor function logic makes it easier to inspect, debug, develop and maintain.

5.3.11 Historian registry

The historian is a specialised registry which records successful transactions, including the participants and identities that submitted them. The historian stores transactions as Historian Record assets, which are defined in the Hyperledger Composer system namespace

5.4 Business Network Archive

Fig 5.3 describes the structure of a .bna file. The model, script, access and query files together comprise the business network archive.

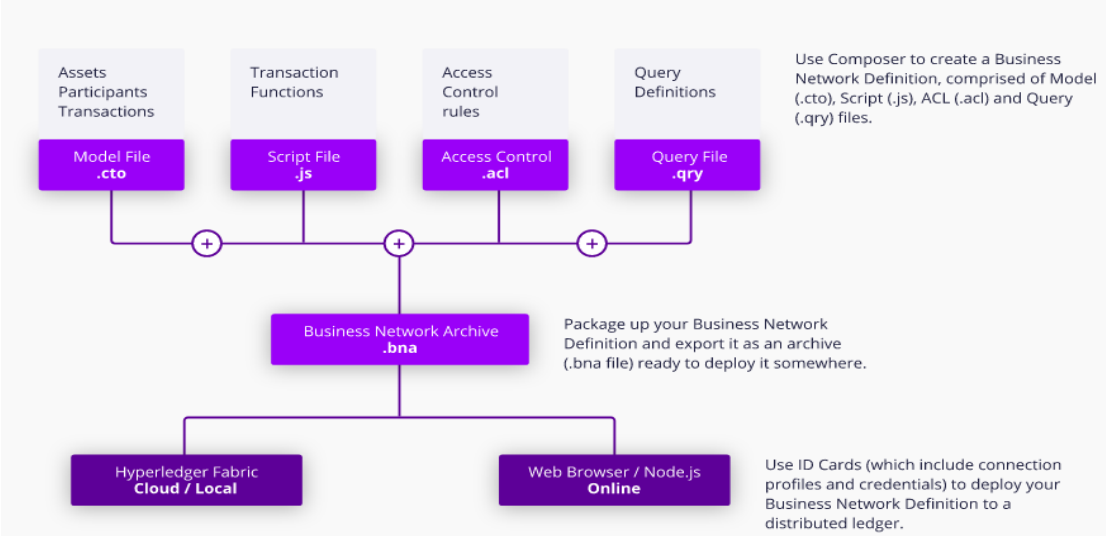

Fig 5.3 Structure of a .bna file

Step 1: Creating a business network structure

Use the following command to generate a business network structure.

```
yo hyperledger-composer:businessnetwork
```

Step 2: Defining a business network

In this step, we need to define our model file, script file, access control file and the query file.

Modelling assets, participants, and transactions (model.cto): The model file contains the definitions of each class of asset, transaction, participant, and event. It implicitly extends the Hyperledger Composer System Model described in the modelling language documentation. The code for our model.cto file is as follows

code:

```
// EHR model (.cto file)
namespace org.acme.biznet
//Namespace to identify the various entities in the model file.
asset Patient_profile  identified by profile_id{
o String profile_id
   --> Patient Patient
   o String firstName
   o String lastName
   o String EmailAddress
   o Integer Dob
   o Address address
}
asset Doctor_profile identified by profile_id{
        // defined an asset called doctor profile identified by an id
```

```
}
participant Patient identified by PatientId {
                // defined a participant patient identified by an id
}
participant Doctor identified by DoctorId {
                //defined a participant doctor identified by an id
}
asset Medical_Record identified by record_id{
                //defined an asset medical record identified by an id
}
asset Encounter identified by encounter_id{
                //defined an asset encounter defined by an id
  }
transaction UpdateRecord {
                //defined a transaction UpdateRecord
}
transaction AllowOtherDoctorsRead{
          //doctor1 will allow doctor2 to read the medical record
}
transaction AllowAdoctorWrite{
                //will allow doctor to write the medical record
}
concept Address {
                // a concept added to enhance the features of the asset medical record.
}
```

Adding JavaScript transaction logic (logic.js file): The transaction processor function file contains the JavaScript logic to execute the transactions defined in the model file.

code:

```
'use strict';

/**

 * Create and store an EHR

 * @param {org.acme.biznet.AllowAdoctorWrite} allowDoctor

 * * @return {Promise} Asset Registry Promise

 * @transaction

 */

function AuthorizeDoctors(allowDoctor){

                return
getAssetRegistry('org.acme.biznet.Patient').then(function(assetRegistery){

                        allowDoctor.patient.authorized.push(allowDoctor.DoctorId);

                        return assetRegistery.update(allowDoctor.patient);

      });

                throw "Too big";

}


/**

 * Create and store an EHR

 * @param {org.acme.biznet.UpdateRecord} updaterecord

 * * @return {Promise} Asset Registry Promise

 * @transaction

 */

function onUpdateRecord(updaterecord) {


        // defines the function for the transaction to update a medical record.


}
```

```
/**
 * Allow a doctor to access a record
 * @param {org.acme.biznet.AllowOtherDoctorsRead} allowAccess
 * * @return {Promise} Asset Registry Promise
 * @transaction
 */
function allowDoctor(allowAccess){
        //defines the function for the transaction that allows doctors read write capabilities.
}
```

Adding access control (permissions.acl) : We need to specify the various access control definitions for the participants of the network.

code:

```
/**
 * Access control rules for EHR-network
*/
rule DoctorCanReadPatient{
        description: "Allow doctor read access to all granted patients"
   participant(p): "org.acme.biznet.Doctor"
   operation: READ
   resource(r): "org.acme.biznet.Patient"
   condition: (r.authorized && r.authorized.indexOf(r.getIdentifier())>-1)
        action: ALLOW
}
rule DoctorCanUpdateEHR{
        //define permission rules for doctor to update HER
        }
rule DoctorCanReadEHR{
```

```
                    //define permission rules for doctor to update EHR
}
rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}
rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
rule SystemACL {
  description:  "System ACL to permit all access"
  participant: "ANY"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
```

Adding Queries File(queries.qry): Each query entry defines the resources and criteria against which the query is executed.

code:

```
/**
 * New query file
 */
query selectPatients {
        //define query to select all patients
}
query selectPatientById {
        // define query to select the patient based on their id
}
query selectDoctors {
        //define query to select all doctors
 }
query selectDoctorById {
        // define query to select the doctor based on their id
}
query selectMedicalRecordByDoctorId {
        // define query to select the EHR based on doctor id
}
query selectMedicalRecordByPatientId {
        // define query to select the EHR based on patient id
}
query selectMedicalRecordByDoctorAndPatientId {
   // define query to select the EHR based on both patient and doctor id
}
```

Step 3: Generate a business network archive : Run the following command to generate a .bna file also known as the business network archive.

```
composer archive create -t dir -n .
```

This archive file will later be deployed to test your blockchain business network created.

5.5 Transactions Structure

Any interactions with health records are recorded as transactions on the network. Transactions are viewable only to the participants associated with the transaction. Here are examples of how transactions take place in the application.

Patient Granting Write Access:

• Patient A grants write access to medical record to Doctor A

• Doctor A's ID is added to Patient A's authorized asset on the ledger

Patient Granting Read Access:

• Patient A grants read access to medical record to Doctor A

• Doctor A's ID is added to medical record's authorized asset on the ledger

Doctor Referring Patient:

• Doctor A updates the permissions to allow Doctor B to read the Patient's medical record

• Chaincode will check that Doctor A has permission to write on the medical record

• Doctor B's ID is added to medical record's authorized asset

• Patient A has to add Doctor B explicitly to his authorized list for write permission

5.6 Data Structures

Table 5.1 Doctor

Variable	Variable Type	Description
Doctor ID	String	A unique doctor ID
Doctor Profile	String	Reference to the doctor profile ID containing demographic data

Table 5.1 describes the basic data structure of the Participant Doctor that is utilised in our business network logic framework of the Hyperledger based blockchain network.

Table 5.2 Encounter

Variable	Variable Type	Description
Encounter ID	String	A unique encounter ID mapped to a unique record ID
Description	String	Description of the diagnosis
Prescription	String	Medical prescription

Table 5.2 describes the basic data structure of the Asset Encounter that is utilised in our business network logic framework of the Hyperledger based blockchain network.

Table 5.3 Medical Record

Variable	Variable Type	Description
Record ID	String	A unique record ID
Doctor ID	String	Reference to a doctor
Patient ID	String	Reference to a patient
Encounter ID	String	Pointer to the encounter containing description and prescription
Location	String	Hospital where medical record is generated
Authorized	String Array	List of read permitted doctors
Encounter Time	Date Time	Time of the event

Table 5.3 describes the basic data structure of the Asset Medical Record that is utilised in our business network logic framework of the Hyperledger based blockchain network.

Table 5.4 Patient

Variable	Variable Type	Description
Patient ID	String	A unique patient ID
Array Authorized	String	List of write permitted doctors
Patient Profile	String	Reference to the patient profile ID containing demographic data

Table 5.4 describes the basic data structure of the Participant Patient that is utilised in our business network logic framework of the Hyperledger based blockchain network.

CHAPTER 6

RESULTS AND DISCUSSION

After installing the pre-requisites and the development environment, the .bna archive file is created as discussed in the previous chapters, must now be deployed and the blockchain business network must be tested. Following which, the REST API is developed and a responsive web interface for the blockchain application is developed. All this is discussed in the following sections of this chapter along with appropriate results and discussions.

6.1 Starting Fabric

The first time you start up a new runtime, you'll need to run the start script, then generate a PeerAdmin card, run the following commands:

```
cd ~/fabric-dev-servers
export FABRIC_VERSION=hlfv11
./startFabric.sh
./createPeerAdminCard.sh
```

You can start and stop your runtime using ~/fabric-dev-servers/stopFabric.sh, and start it again with ~/fabric-dev-servers/startFabric.sh. Fig 6.1,6.2,6.3 depicts the starting of fabric and creation of a peer admin

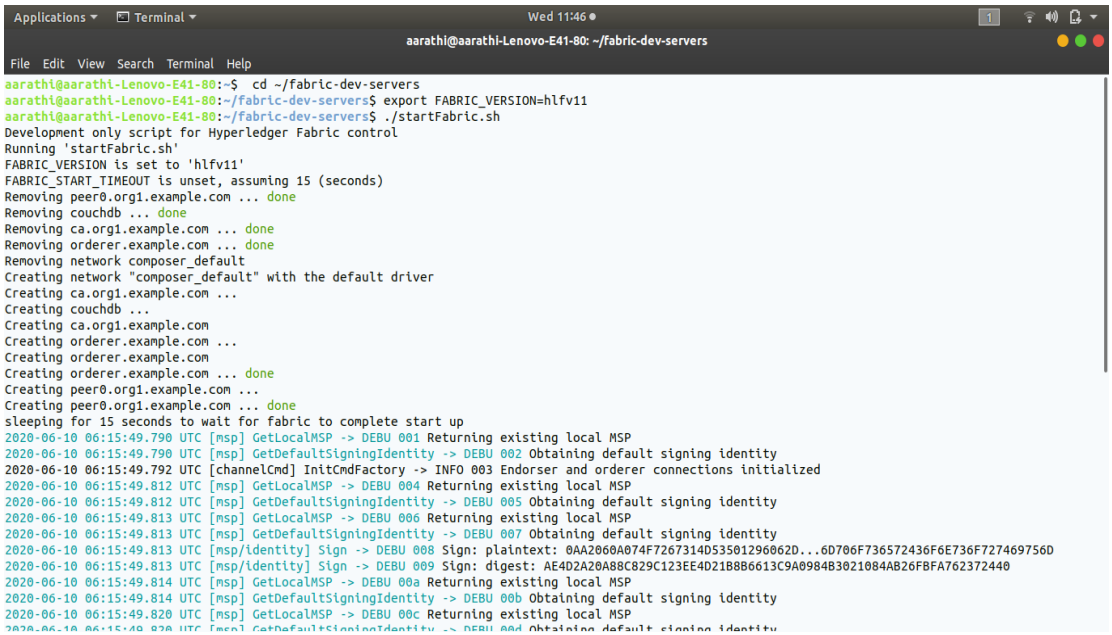

Fig 6.1 : Start Fabric

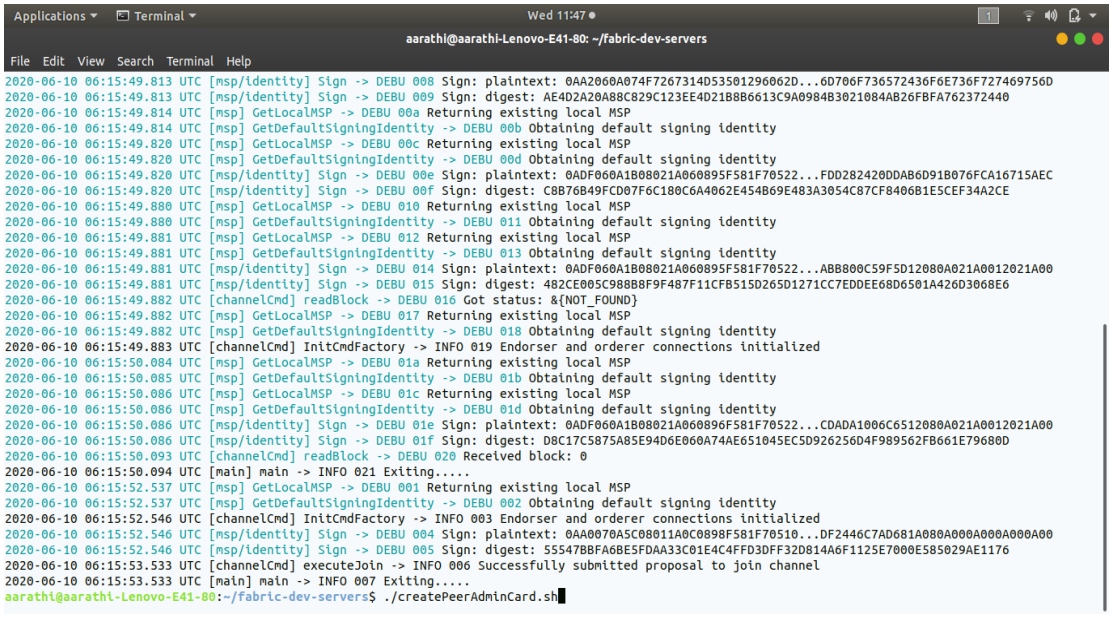

Fig 6.2 : Fabric Started

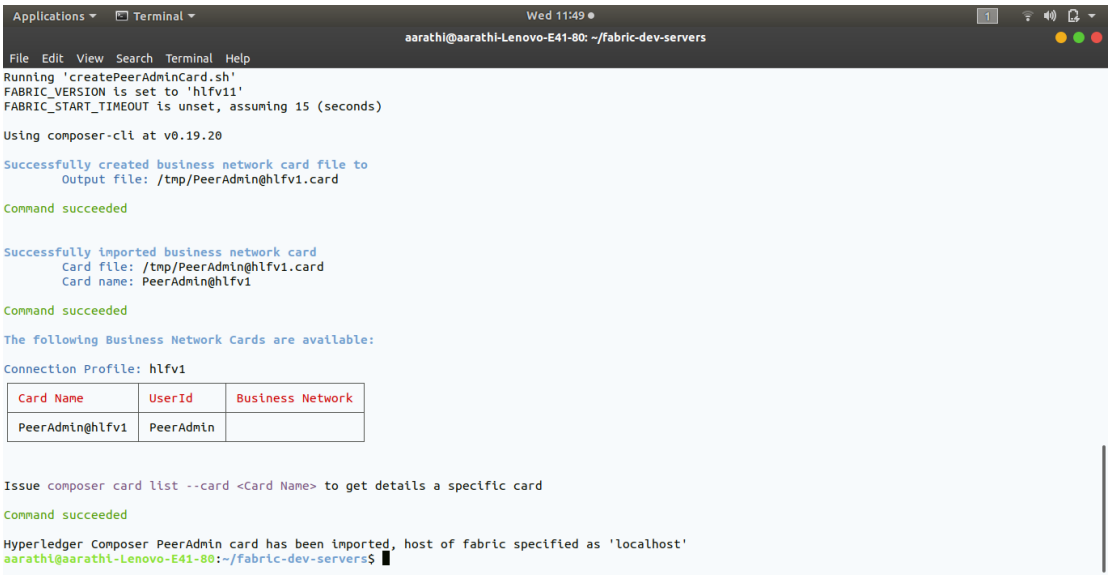

Fig 6.3 : PeerAdmin Created

6.2 Deploying the Business Network

After creating the .bna file, the business network can be deployed to the instance of Hyperledger Fabric. Normally, information from the Fabric administrator is required to create a PeerAdmin identity, with privileges to install chaincode to the peer as well as start chaincode on the composer channel. However, as part of the development environment installation, a PeerAdmin identity has been created already.

After the business network has been installed, the network can be started. For best practice, a new identity should be created to administer the business network after deployment. This identity is referred to as a network admin. Deploying a business network to the Hyperledger Fabric requires the Hyperledger Composer business network to be installed on the peer, then the business network can be started, and a new participant, identity, and associated card must be created to be the network administrator. Finally, the network administrator business network card must be imported for use, and the network can then be pinged to check it is responding.Fig 6.4 and Fig 6.5 depicts this process.

step 1:To install the business network, from the tutorial-network directory, run the following command:

```
composer network install --card PeerAdmin@hlfv1 --archiveFile tutorial-network@0.0.1.bna
```

The composer network install command requires a PeerAdmin business network card (in this case one has been created and imported in advance), and the the file path of the .bna which defines the business network.

step 2:To start the business network, run the following command:

```
composer network start --networkName tutorial-network --networkVersion 0.0.1 --networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card
```

The composer network start command requires a business network card, as well as the name of the admin identity for the business network, the name and version of the business network and the name of the file to be created ready to import as a business network card.

step 3:To import the network administrator identity as a usable business network card, run the following command:

```
composer card import --file networkadmin.card
```

The composer card import command requires the filename specified in composer network start to create a card.

step 4: To check that the business network has been deployed successfully, run the following command to ping the network:

```
composer network ping --card admin@tutorial-network
```

The composer network ping command requires a business network card to identify the network to ping.

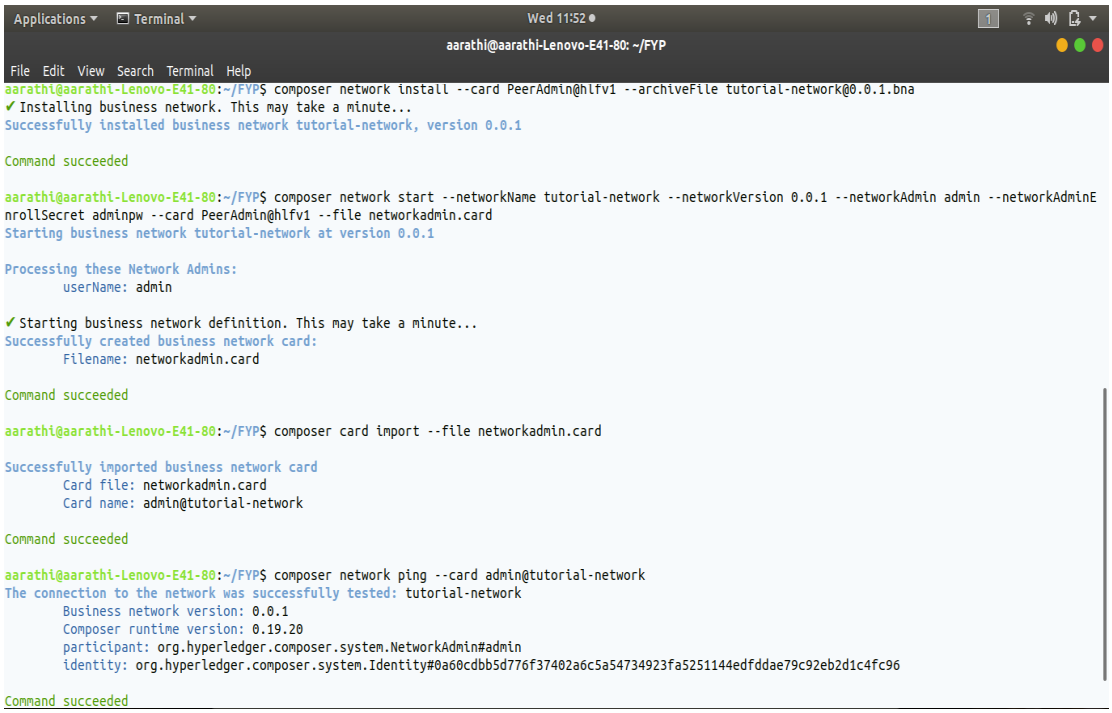

Fig 6.4 : Deploying .bna Archive File

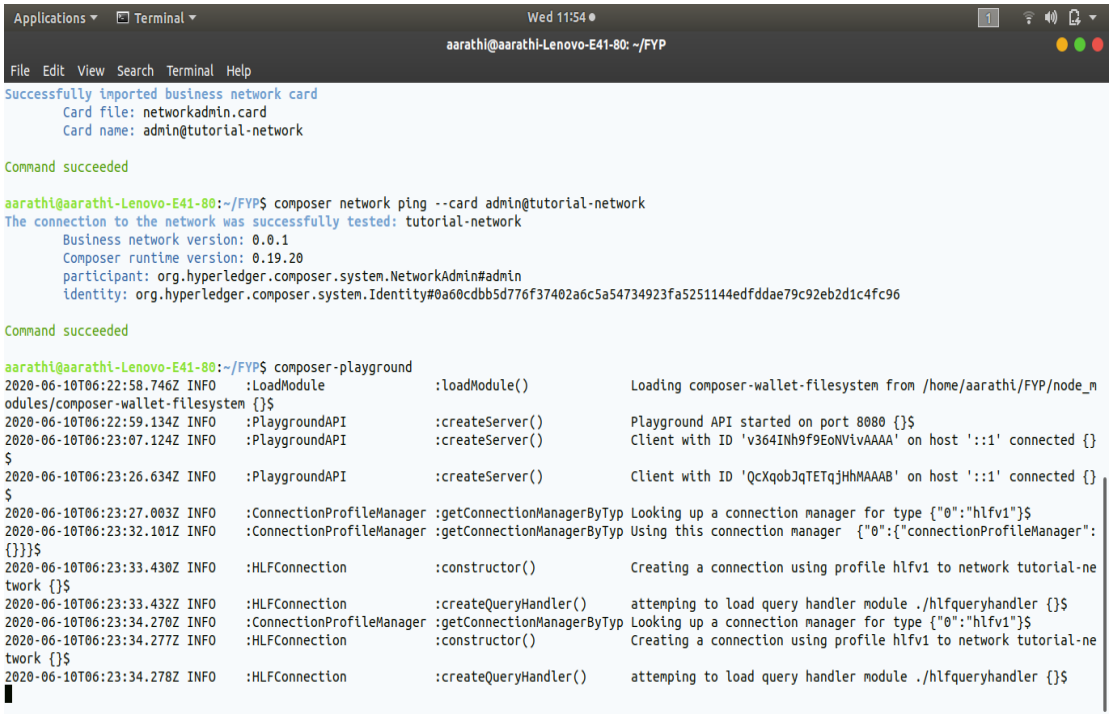

Fig 6.5 : Run Composer Playground

6.3 Testing the Blockchain Network

After successfully deploying the business network, the blockchain network can be tested using the composer playground, a local IDE for testing blockchain applications. To test run the command, composer-playground

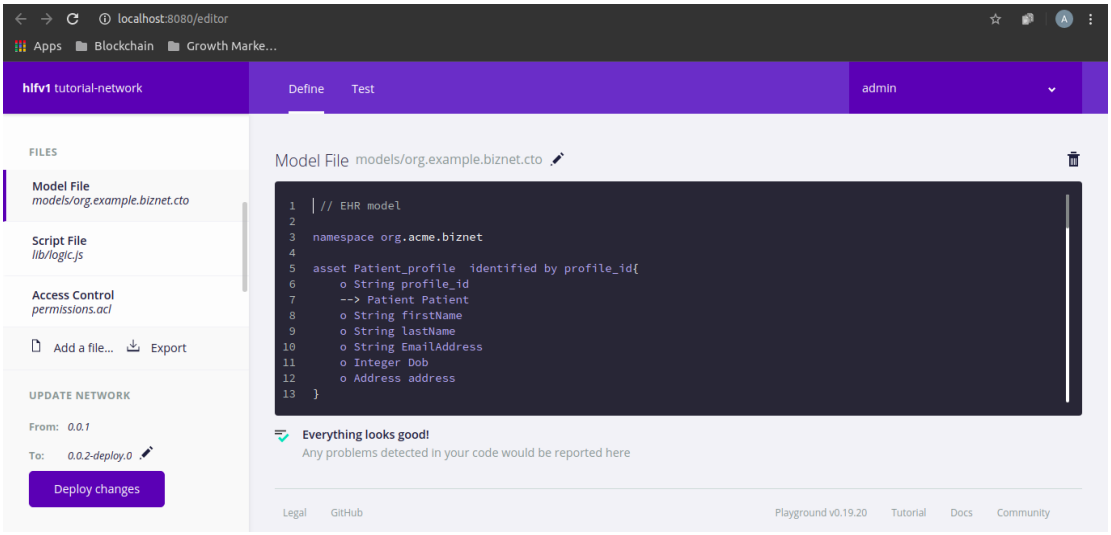

Fig 6.6 : Defining the .cto, .acl, .js files

Fig 6.6 shows how we can build and define our blockchain network on the composer playground IDE. The modelling, logic and the access control files can be defined and deployed.

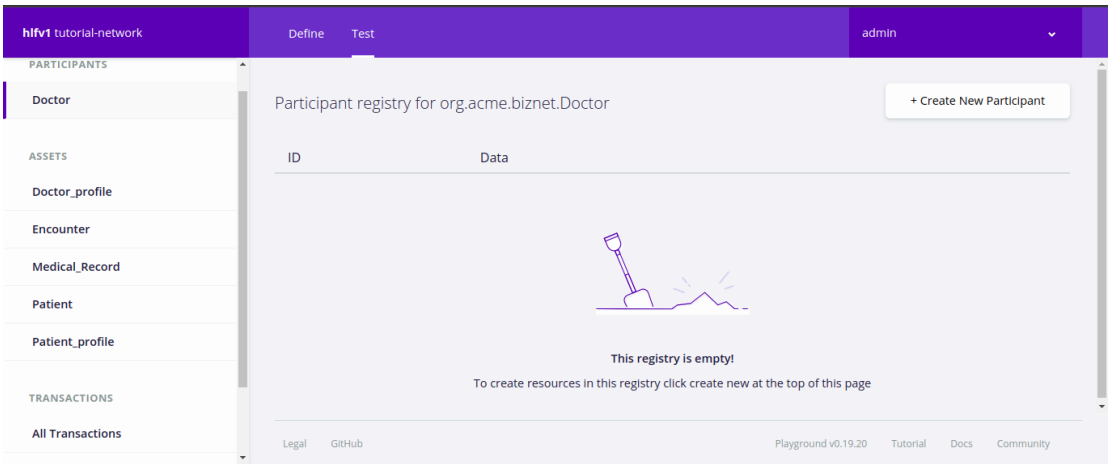

Fig 6.7 : Testing the deployed .bna file

Fig 6.7 shows how we can test the various participants, assets, transactions created and test out the blockchain network.

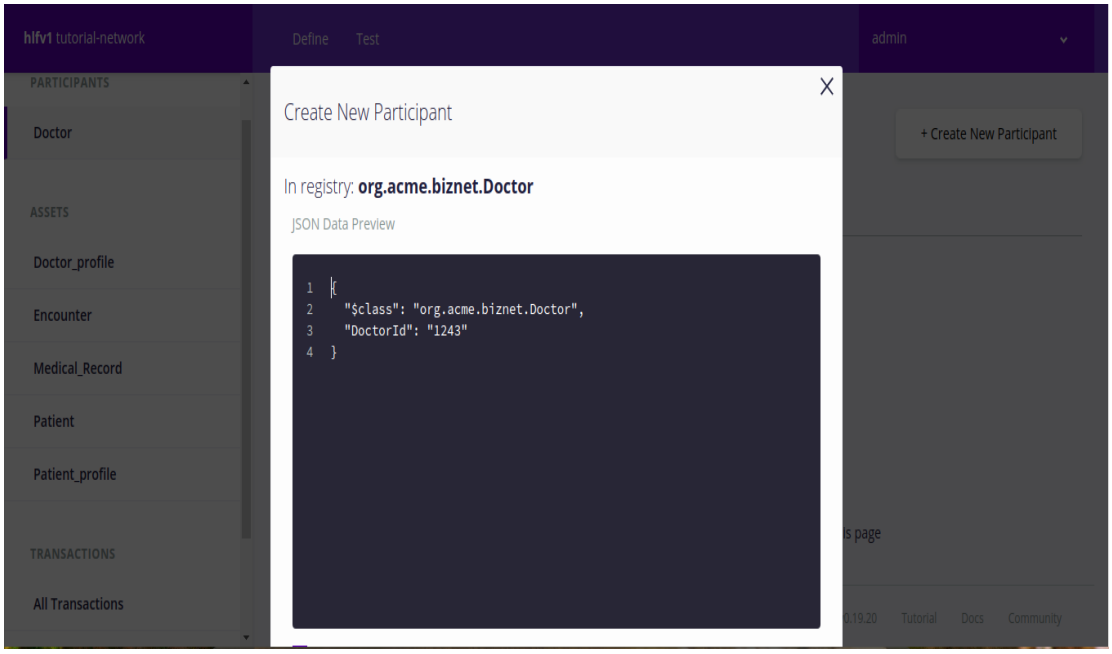

Fig 6.8 : Create a new participant

Fig 6.8 shows how we can create a new participant by filling in the required details. Each entity in the business network is defined using a namespace.

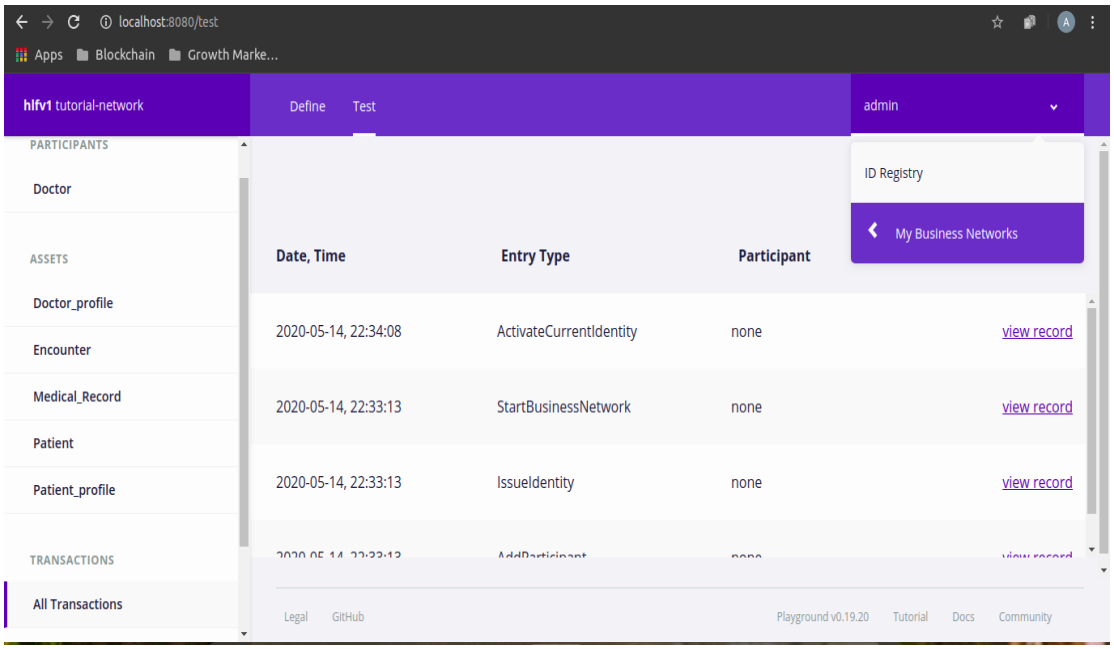

Fig 6.9 : Transaction History

Fig 6.9 shows the transaction historian. All the transactions such as creation, deletion, updations, peer rights, etc. will be recorded. This is the main feature of a blockchain network. It displays all the transactions and this data cannot be modified. On the top right, we can even check out the ID registry or even change the business network.

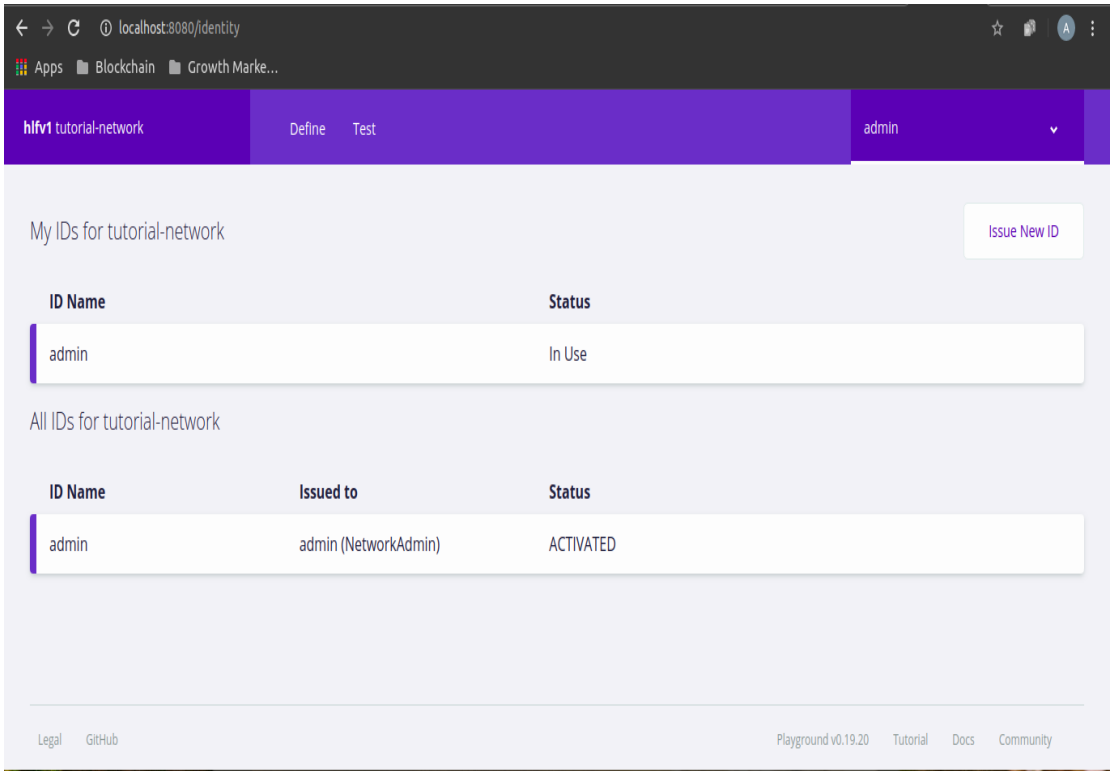

Fig 6.10 : ID registry

Fig 6.10 shows the ID registry and the ID in use. This space is for the creation of new IDs or peers and can be given admin rights to create new IDs depending on the rights.

6.4 REST API

Generating a REST server. Hyperledger Composer can generate a bespoke REST API based on a business network. For developing a web application, the REST API provides a useful layer of language-neutral abstraction.

```
composer-rest-server
```

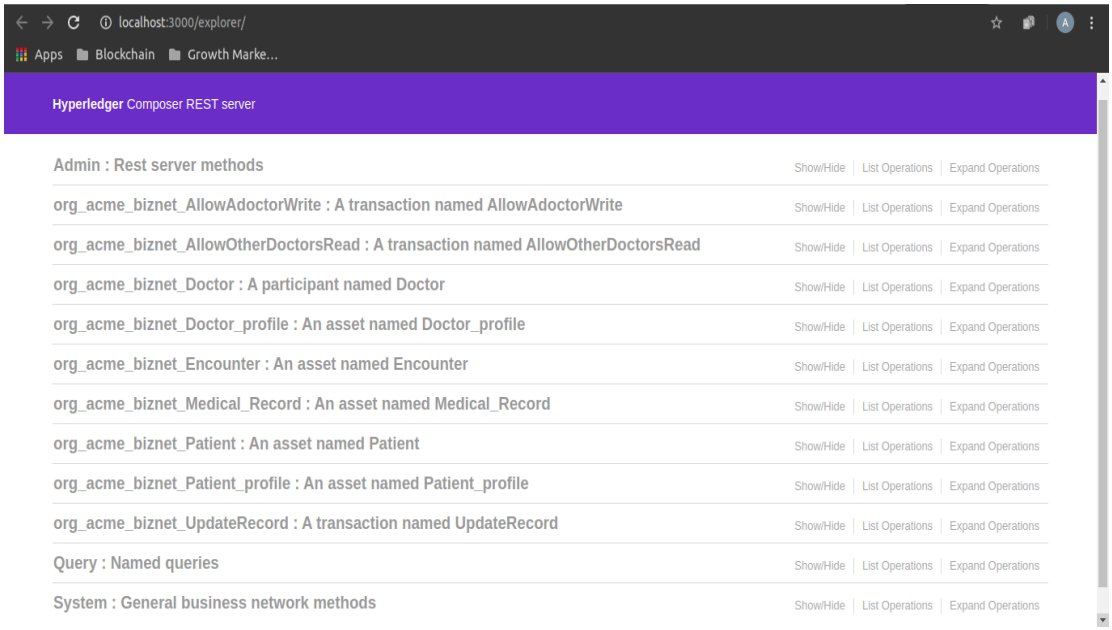

Fig 6.11 : REST API interface

Fig 6.11 shows us the RESTful Api interface that acts as a layer between the web application and the blockchain network.

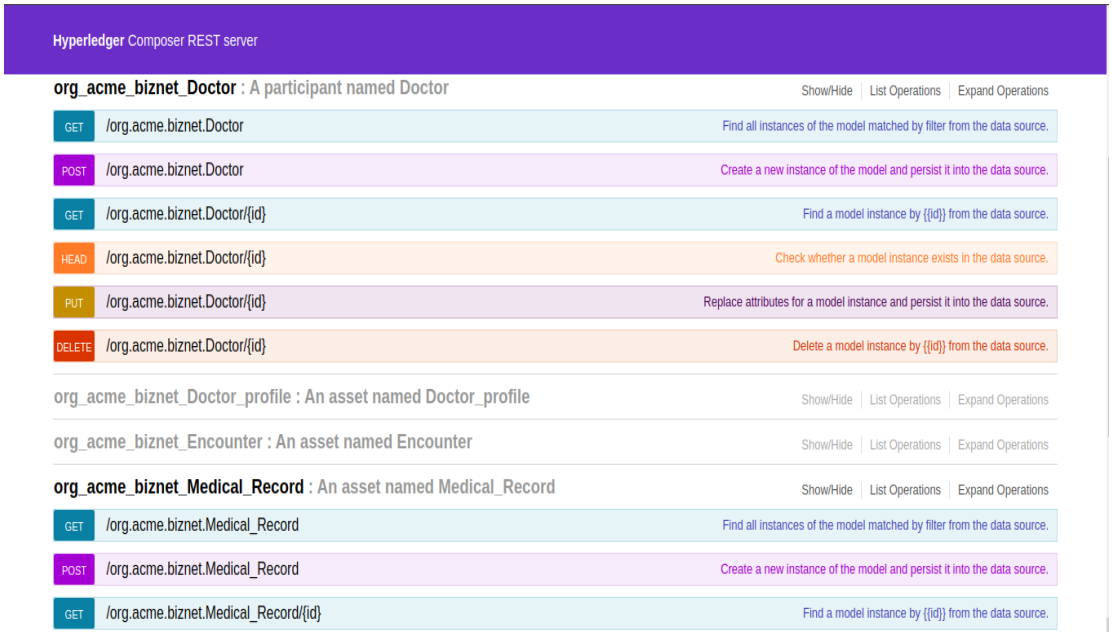

Fig 6.12 : REST Api calls

Fig 6.12 shows the various http protocols that the rest api utilises. The api calls corresponds to the CRUD(Creation- POST, Retrieve/Select- GET, Update- PUT, Delete)

6.5 Web Application User interface

Depending on the business requirements and technical feasibility, the front-end User Interface can be developed. The proposed system acts as a proof of concept for the healthcare industry that have varying business requirements and accordingly appropriate tech stack can be used to develop the web application such as Angular, Reactjs etc.

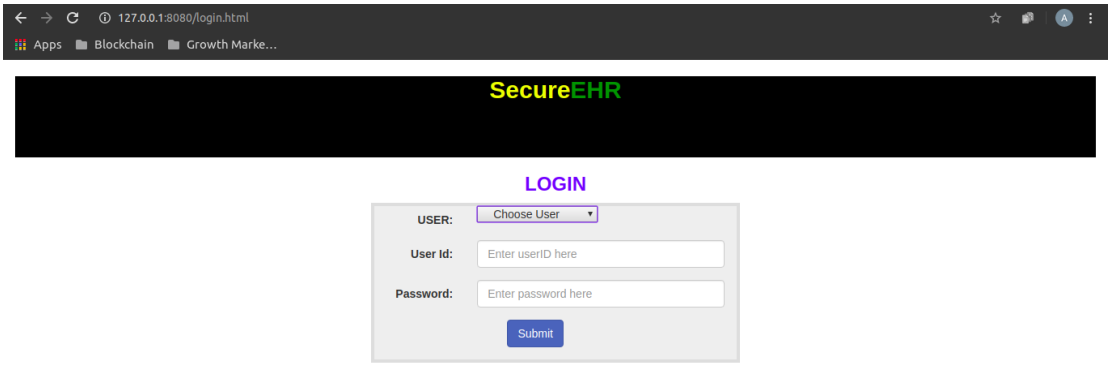

Fig 6.13 : Web-App

CHAPTER 7

TESTING

Software Testing is the process to evaluate the functionality of the software application with an intent to find whether the software developed meets the specified requirements and to identify defects to ensure to that standards of quality and performance are maintained. This chapter gives an overview of the various types of testing incorporated during the entire duration of the project.

7.1 Unit Testing

Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses. We have tested the individual units of our business logic such as the model, script and permission files.

7.2 Integration Testing

Testing of all integrated modules to verify the combined functionality after integration is termed as Integration Testing. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems. Here, we performed testing for the Blockchain network which is achieved using Hyperledger composer playground. Here we can do trial and error method to test out our business logic that allows for a functioning blockchain network

7.3 Interface Testing

The objective of this Interface Testing is to validate the interface as per the business requirement. The expected interface of the application is mentioned in the detailed design document and interface mock-up screens. Checks if the application correctly connects to the server.

7.4 System Testing

Under System Testing technique, the entire system is tested as per the requirements. It is a Black-box type Testing that is based on overall requirement specifications and covers all the combined parts of a system. Here we tested if the application is fully functional as a whole.

7.5 Compatibility Testing

Compatibility Testing checks whether the application is compatible with the specified software and hardware requirements and functions efficiently as expected.

CHAPTER 8

CONCLUSION & FUTURE SCOPE

This chapter gives an overview of the conclusion, our contribution to the project, the future scope of the proposed system and our learnings incurred during the entire duration of the project.

8.1 Conclusion

The confidentiality and integrity is of high importance in the healthcare domain. It may directly or indirectly affect the success rate of surgery and the outcome of research organized by institutions. The Blockchain 4.0 is implemented using Hyperledger Fabric framework provides a secure environment for storing the medical records. The peers involved in the Blockchain are only allowed to access the specified data which are defined based on their permission and the organization they participated in.

8.2 Contribution

The proposed system provides control to patients, as they are the data owners w.r.t to the asset i.e. EHRs. This allows patients control over their own records and data users need to request for accessing data. Besides that, the medical records of patients are unified and stored distributed on Blockchain, so the doctors can retrieve it within a second in order to make any medical decision based on the medical record. On the other hand, the medical research institutions do not need to worry about the quality and quantity of data samples anymore as the EHRs that they are allowed to access will act as useful datasets to carry out various researches which will benefit the medical industry

The Blockchain infrastructure and environment was developed by using Hyperledger Fabric and Composer. A responsive web application was designed for users to review on their own medical records. In addition, the web application allows doctors within the inter-hospital network and research institution to query data. A public ledger was also created to store the sensitive data from medical institutions.

The Blockchain infrastructure application and a fully developed web application was designed for secure medical record storing for an inter-hospital network. This project makes the personal health data of patients become more secure and protected by applying the latest blockchain technology. The current centralized method for storing medical records can be replaced by applying Blockchain technology. The patients' data can be retrieved more efficiently anywhere and anytime. Thus, the project undertaken for this proof of concept will act as a solution to today's healthcare industry by allowing for more secure resource sharing for business as well as research.

8.3 Future Scope

The proposed system is a proof of concept for the medical domain. Real-time application can be expanded to further functionalities to cater to needs of the healthcare industry.

- Fingerprint or SMS authentication can be implemented.
- Besides that, a mobile application can be more convenient for patients to access their own medical information.
- If due to unforeseeable circumstances the patient is unavailable to authorize access to the doctor, the patient can assign a trusted proxy to authorize the access.
- EHR structure can be modelled to suit specific diseases such as Cancer, Diabetes ,COVID-19 etc.
- Pharmacists can be added to monitor medical sales
- Insurance companies can be given limited access for fairer claims.

8.4 Learnings

1) Blockchain:

A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block typically contains a cryptographic hash of the previous block, a timestamp and transaction data. By design, a blockchain

is inherently resistant to modification of the data. It is ”an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way”. For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol or inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without the alteration of all subsequent blocks, which requires collusion of the network majority.

2) Importance of blockchain:

Blockchain technology has a large potential to transform business operating models in the long term. Blockchain distributed ledger technology is more a foundational technology with the potential to create new foundations for global economic and social systems than a disruptive technology, which typically ”attack a traditional business model with a lower-cost solution and overtake incumbent firms quickly”. The use of blockchains promises to bring significant efficiencies to global supply chains, financial transactions, asset ledgers and decentralized social networking.

3) Difference between public and private blockchain:

The sole distinction between public and private blockchain is related to who is allowed to participate in the network, execute the consensus protocol and maintain the shared ledger. A public blockchain network is completely open and anyone can join and participate in the network. The network typically has an incentivizing mechanism to encourage more participants to join the network. Bitcoin is one of the largest public blockchain networks in production today. One of the drawbacks of a public blockchain is the substantial amount of computational power that is necessary to maintain a distributed ledger at a large scale. More specifically, to achieve consensus, each node in a network must solve a complex, resource-intensive cryptographic problem called a proof of work to ensure all are in sync.

4) Importance of Electronic Health Records:

EHRs and the ability to exchange health information electronically can help you provide higher quality and safer care for patients while creating tangible enhancements

for your organization. EHRs help provide accurate, up-to-date, and complete information about patients at the point of care. It helps securely sharing electronic information with patients and other clinicians. It promotes legible, complete documentation and accurate, streamlined coding and billing. It enhances privacy and security of patient data. It reduces costs through decreased paperwork, improved safety, reduced duplication of testing, and improved health.

REFERENCES

[1] https://hyperledger.github.io/composer/v0.19/introduction/introduction

[2] Dinh C. Nguyen, et al. **"**Blockchain for Secure EHRs Sharing of Mobile Cloud based E-health Systems" *School of Engineering, Deakin University, Waurn Ponds, VIC 3216, Australia*

[3] Haider Dhia Zubaydi, et al. "A Review on the Role of Blockchain Technology in the Healthcare Domain" *School of Computer and Information Engineering, Sangji University, Gangwon 220-702, Korea, 15 June 2019*

[4] Sara Rouhani, et al. MediChain: A Secure Decentralized Medical Data Asset Management System

[5] Guang Yang, et al. "A Blockchain-based Architecturefor securing electronic health record system" , *13 July 2019*

[6] Asad Ali Siyal, et al.Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives, *2 January* 2019

[7] Nchinda Nchinda, et al. MedRec: A Network for Personal Information Distribution, *2019 International Conference on Computing, Networking and Communications(ICNC): Internet Services and Applications*

[8] Asaph Azaria, et al. Medrec: Using Blockchain for Medical Data Access and Permission Management, *2016 2nd International Conference on Open and Big Data*