BLOCKCHAIN IN HEALTHCARE

Project report submitted in fulfilment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Ayush Saini (161300)

Under the supervision of

Dr. Geetanjali Rathee

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan - 173234, Himachal Pradesh

CERTIFICATE

Candidate's Declaration

This is to certify that the work which is being presented in the report entitled "**Blockchain in Healthcare**" in partial fulfilment of the requirements for the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science and Engineering, Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2019 to December 2019 under the supervision of **Dr. Geetanjali Rathee** (Assistant Professor, Senior Grade, Computer Science & Engineering Department).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ayush

Ayush Saini (161300)

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Geetanjali

Dr. Geetanjali Rathee

Assistant Professor (Senior Grade)

Department of Computer Science

Dated:

ACKNOWLEDGEMENT

I am greatly indebted to my project supervisor **Dr. Geetanjali Rathee** for her guidance and constant supervision as well as providing us with all the information regarding the project titled - **Blockchain in Healthcare.**

I am also thankful to **Jaypee University of Information Technology** for providing me with all the latest technologies, thus, comforting me with the project.

Again, I owe my profound gratitude to my project guide, for not only helping me with the project but also developing a keen interest in the same during its progress.

Geetanjali

Date: 3 June 2020

Dr. Geetanjali Rathee

Assistant Professor

CONTENTS

Chapter 1

Introduction 1

1.1 What is blockchain? 1

1.1.1 Characteristics of blockchain 1

1.1.2 Types of blockchain… 3

1.1.3 Working of a blockchain 5

1.1.3.1 Data elements inside a block 6

1.1.3.2 Concepts used in a blockchain… 8

1.1.4 Traditional electronic health record systems 11

1.2 Problem statement 13

1.3 Objectives 14

Chapter 2

Literature survey

2.1 TITLE: A Case Study for Blockchain in Healthcare: "MedRec" Prototype for Electronic Health Records and Medical Research Data 15

2.1.1 EHR issues 16

2.1.2 MedRec as a solution 16

2.1.3 System implementation 17

2.1.4 Smart contract structure 17

2.1.5 System node description 19

2.2 TITLE: A Systematic Review of the Use of Blockchain in Healthcare 20

2.2.1 Distributed consensus protocol 21

2.2.2 Research question 22

2.2.3 Data sources 22

2.2.4 Discussion on questions 23

2.3 TITLE: The Potential of Blockchain Technology for Health Information Exchange: Experimental Study from Patients' Perspectives 25
2.3.1 Role of patients in healthcare information Exchange 26
2.3.2 Issues with the current system 27
2.3.3 Experiment design 28
2.4 TITLE: Applications of Blockchain in Healthcare: Current Landscape & Challenges 31
2.4.1 Areas of application 31
2.4.2 Analytics 32
2.4.3 Challenges to blockchain… 33
2.5 TITLE: Blockchain-based Electronic Patient Records for Regulated Circular Healthcare Jurisdictions 35
2.5.1 Working Context 36
2.5.2 Participation considerations 37
2.5.3 Smart contracts 37
2.5.4 Data model 38
2.5.5 Operative Cases 39

Chapter – 3

System Development 41
3.1 Proposed Model 41
3.1.1 High Level Use Case 41
3.1.2 Uploading of Files by Patient 42
3.1.3 Accessing of Files by Doctor 43
3.2 Model Development 43
3.2.1 Hardware Used 43
3.2.2 Software Used 44
3.3 Implementation of Model 44
3.3.1 Flow of execution of the project 44

3.3.2 Generation of Public & Private Keys.....47
3.3.3 Registering entities and creating a list of miners and doctors….48
3.3.4 Blockchain class and genesis block creation……...49
3.3.5 Creation of a block & mining it 50
3.3.6 Transaction class..... 51
3.3.7 Main driver code to create chain and call other functions……..52
3.3.8 Checking if a doctor is authentic and registered…….....53
3.3.9 Selecting a miner……...….53
3.3.10 Adding new diagnosis onto the blockchain……54
3.3.11 Function to add transaction……..55
3.3.12 Checking if the transaction is valid……..56
3.3.13 Signing a transaction……56
3.3.14 Mining the pending transaction and checking if a miner is fake .57
3.3.15 Printing all diagnosis of a patient……58
3.3.16 Getting balance of a miner……...……59
3.3.17 Checking if the chain is valid……...……59

Chapter – 4

Performance Analysis 51
4.1 Time take to generate key pairs for various entities 61
4.2 Time Taken to Mine a Block with Different Difficulties By a Miner..... 62
4.3 Graph of variation of time taken to mine a block with difficulty level……..63
4.4 Output analysis……...64

Chapter - 5

Conclusion …….……65

Chapter - 6

Future work……..65

References……66

List of Tables

S.NO	Title	Page No.
1	**Table 4.1 – Time taken to generate key pairs for each entity**	**61**
2	**Table 4.2 – Time taken to mine blocks with different difficulty**	**62**

List of Figures

S.NO	Title	Page
1	**Figure 1.1 Pivotal characteristics of a blockchain**	**3**
2	**Figure 1.2 Types of blockchain**	**5**
3	**Figure 1.3 Example of hash input and output**	**7**
4	**Figure 1.4 Linking of blocks using hash of previous block**	**7**
5	**Figure 1.5 Mining process and working of proof of work**	**10**
6	**Figure 1.6 Proof of work vs proof of stake**	**11**
7	**Figure 1.7 Problems with EHR systems**	**13**
8	**Figure 2.1 Data content of MedRec contracts**	**18**
9	**Figure 2.2 Provider adding patient records.**	**19**
10	**Figure 2.3 Sixteen scenarios of experiment**	**29**
11	**Figure 2.4 Experiment design diagram**	**30**
12	**Figure 2.5 Data model**	**38**
13	**Figure 2.6 Provider requesting EPR access agreement**	**39**
14	**Figure 2.6 Provider accessing an EPR**	**40**
15	**Figure 3.1 – High Level Flow Diagram**	**41**
16	**Figure 3.2 – Flow Diagram for Uploading of files**	**42**
17	**Figure 3.3 - Flow Diagram for Accessing of files**	**43**
18	**Figure 3.4 – Flow of execution of the project**	**44**
19	**Figure 3.5 – Module to generate public and private keys and display time taken to generate keys.**	**47**
20	**Figure 3.6 – registering entities onto the system and creating a list of miners and doctors**	**48**
21	**Figure 3.7 blockchain class and genesis block creation**	**49**
22	**Figure 3.8 – Class of a block and its functions**	**50**
23	**Figure 3.9 A transaction class**	**51**
24	**Figure 3. 10 – Main driver code to create chain and call other functions**	**52**
25	**Figure 3.11 – Checking if a doctor is authentic and registered**	**53**

26	**Figure 3.12 - selecting a miner to mine a block**	**53**
27	**Figure 3.13- adding new diagnosis onto the blockchain**	**54**
28	**Figure 3.14 The add transaction function**	**55**
29	**Figure 3.15 –Checking if the transaction is valid**	**56**
30	**Figure 3.16 Signing a transaction**	**56**
31	**Figure 3.17 mining the pending transaction and checking if the miner is fake.**	**57**
32	**Figure 3.18 displaying all diagnosis of a patient**	**58**
33	**Figure 3.19 getting balance of a miner**	**59**
34	**Figure 3.20 Checking if the chain is valid**	**59**
35	**Figure 4.3 – Graph showing variation of time taken to mine block with difficulty level**	**63**
36	**Figure 4.4 – Final Output Snippet**	**64**

ASBTRACT

Medical data is the most key resource of the modern healthcare but the traditional electronic health record system is incapable of providing a safe, efficient and consistent database of patient medical record. The current system also lack interoperability, patient centralism and secure access. Blockchain as an emerging technology solves the above mentioned problems and provides a distributed, decentralized and immutable ledger of patient. A permissioned blockchain is implied on the use case and uses various fundamentals of blockchain such as public-private key cryptography, interlinking of hashes, distributed consensus algorithm and proof of work.

CHAPTER - 1

INTRODUCTION

This chapter explains the cornerstone ideas and fundamental concepts used in blockchain. The characteristics, types, working, data elements and working concepts are thoroughly described. This chapter also puts forward the problem statement and objectives of the project.

1.1 What is Blockchain?

Blockchain is a decentralized, distributed and immutable ledger which can store records in a growing fashion. Blockchain makes the storage of records and data secure by making them nearly impossible to change and tamper with. It creates a set of cryptographically hashed records linked with each other using cryptographic keys.[1]

1.1.1 Characteristics of blockchain

Blockchain comprises of three characteristic points in its definition, which form the cornerstone of the technology:-

- **Decentralized**

Decentralization in blockchain means that there is no central point of control and all the transactions happen in a peer to peer way. In the extreme cases i.e in case of a

private and consortium blockchain, apart from registering users on the blockchain there is no role of any central authority in the transactions that happen in the blockchain. This is also means that there is no administrator in blockchain unlike traditional database systems where a central administrator controls the database. Validation of the transactions is done by the nodes which participate in the blockchain. This decentralized property of blockchain creates a peer to peer trust in a trustless environment.

- **Distributed**

Distribution means that each and every user registered on a blockchain has a separate and independent copy of the blockchain residing with him/her. This makes tampering records a really arduous task as in order to successfully hack into the blockchain record system the hacker will have to change the records stored in all the nodes of blockchain as change in one node's data will not reflect in other node's data. This is one of the property that makes blockchain impossible to tamper with and marks it as the most secure record storing technology.

- **Immutable**

It means that once records are added onto a blockchain it is impossible to remove them or alter them in any way. In fact it is this immutable property of blockchain that makes it secure and thus lays the foundation of its use cases as records are

immutable and so can neither be tampered with and can neither be deleted.

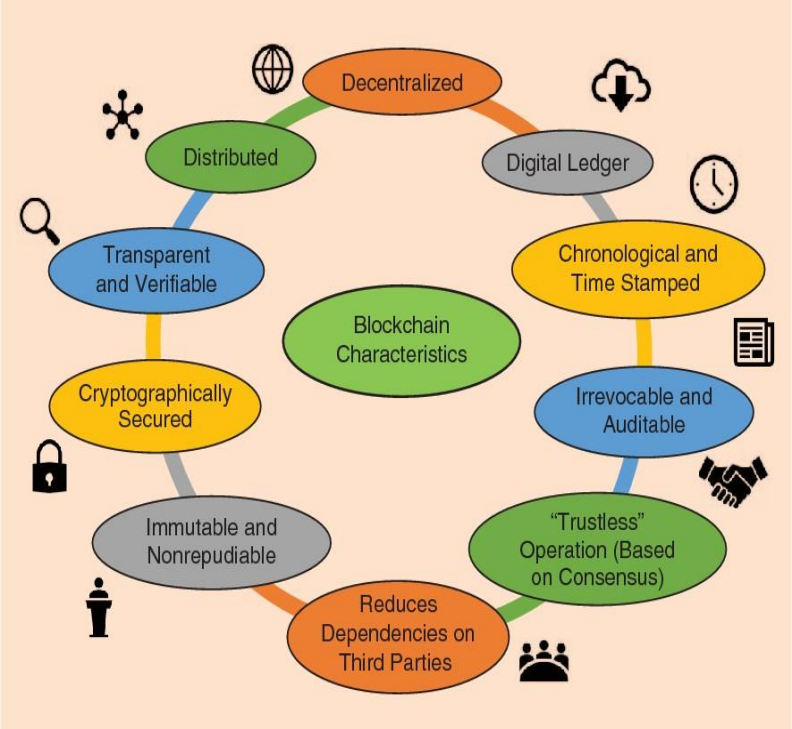


Figure 1.1 Pivotal characteristics of a blockchain

1.1.2 Types of blockchain[2]

Blockchain has three primary types :-

- Private blockchain
- Public blockchain
- Hybrid blockchain or Consortium blockchain

1.1.2.1 Public blockchain

Public blockchain is fully decentralized because authority is equally distributed among all the nodes (peers). In this anyone cane be a validator and has no restriction

on who the participants of the blockchain can be. The transactions that take place in an open blockchain are completely transparent which means that transaction details can be viewed by anyone.

1.1.2.2 Private blockchain

This type of blockchain has restrictions on participation, transitioning and validation. A few pre-chosen entities can access a private blockchain. The access privileges are given by developers while building the blockchain and the network administrator. Due to these properties a private blockchain is also called a permissioned blockchain. Private blockchains can be used within an organization to store and record its sensitive data and transactions that should be visible to only a certain people within organization.

1.1.2.3 Consortium blockchain

This, basically, means a hybrid blockchain with the properties of both private as well as public blockchain.

It consists of two types of users : who have control over who can access the blockchain and users who access the blockchain. The main difference between hybrid and private blockchain is that in private blockchain the administrative rights lie with more than one entity (people/organizations) whereas in private blockchain administrative rights reside with a single entity (person/ a single organization).

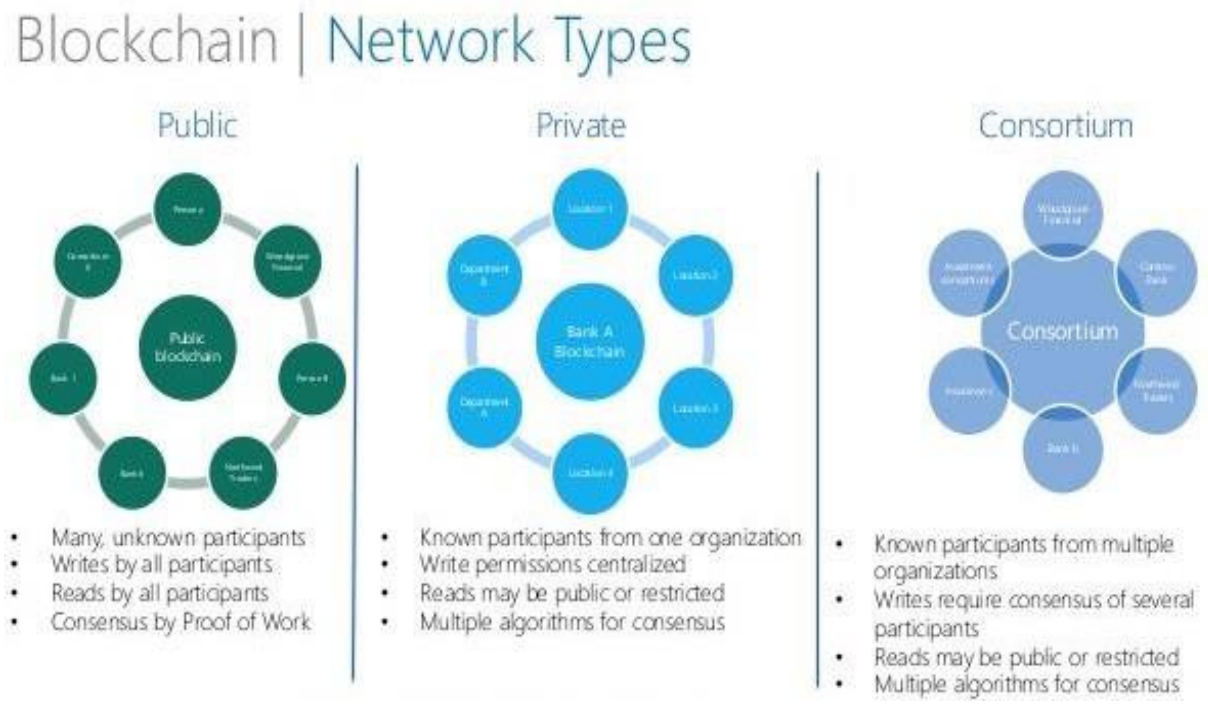


Figure 1.2 types of blockchain

1.1.3 Working of blockchain

A blockchain is basically a number of blocks which are linked to each other using cryptographic hashes. Every block has a hash to the previous block so that if any block gets tampered with then all the other blocks will get to know about it and so, to hack into the blockchain a hacker will have to change the hash of all the blocks.

1.1.3.1 Data elements inside a block

A block consists of the following components:-

- Hash of current block
- Hash of previous block
- Nonce
- Data
- Timestamp
- Index

- **Hash of current block**

Each block is identified uniquely using a hash that is stored in the block. The hash is calculated using a cryptographic function which takes as input all the components that are in a block and returns a unique hash value which is unique to every block. The hash function returns a fixed length of output no matter what length of the input is given to it.

Input	Output (Hash)
Cat	93g56gtf229hbno00r45sktrpbs59so9r3t7saer
A white cat is outside	js03bbstgo94r6s1z8mg05fgt3sba9tob32bsap7
A white cat is inside	bbr19007go2tsi52bsi50o21nmiseas45on23mjn
A whiet cat is inside	339n5sbk249nb9530gjdl04h92jg02jg9sm93hpz
A white cat is insid	4bbj390osoh9djm395bksh94gf03sg034dfjh31x

Figure 1.3 Example of hash input and output

- **Hash of previous block**

Each block stores hash of the preceding block and it is in fact this component of the block which ensures that the blockchain is safe and tamper proof. Each block links to the previous one so that tampering with any block leads to inconsistency of data in the next block. So, this creates a chained inconsistency that helps to prevent any data hack.

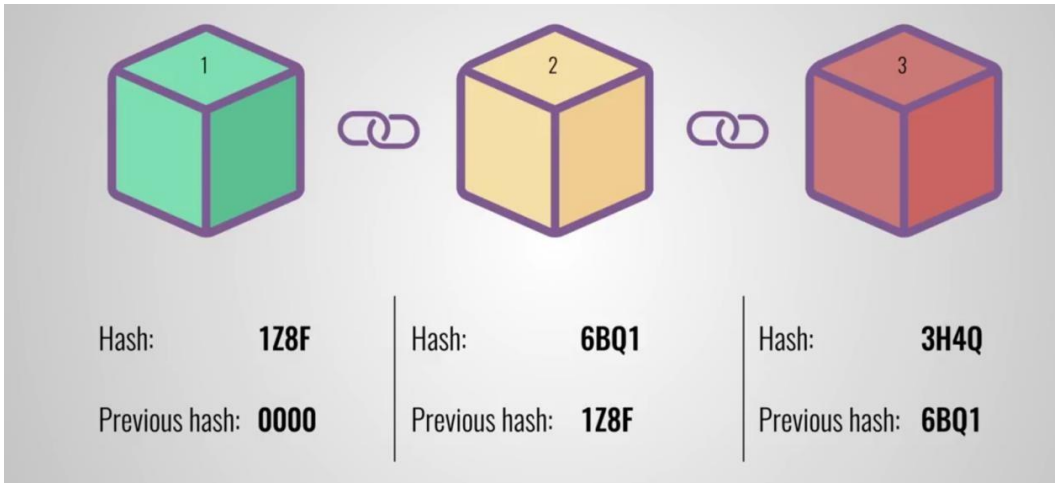


Figure 1.4 linking of blocks using hash of previous block

- **Data**

This contains the actual information which is to be stored in the blockchain.

- **Timestamp**

This refers to the date and time of creation of the block.

- **Index**

The index stores the serial number of the block created.

- **Nonce**

Nonce is used to calculate the proof of work. Nonce helps to change hashes while calculating hash during proof of work as the hash function takes all of the components of the block as input and out of all those components we cannot change any of them. So, we increment nonce each time so that the hash value changes with each iteration to get the desired proof of work.

1.1.3.2 Concepts used in a blockchain[3]

Following is the list of some general terms and concepts which must be understood first in order to understand blockchain :-

- **Genesis block**

Genesis block is the first block of any blockchain and is also referred to as the zero block. The block next to zero block is indexed 1.

- **Block difficulty**

Block difficulty represents how hard it will be for the miner to find the value of the nonce that will calculate the required proof of work to validate transactions. In a general blockchain scenario it represents the length of string of zeroes that should be prefix to the hash of the block.

- **Miners**

Miners are the peers that participate and access the blockchain network. These miners validate new transactions and record them on the ledger. These miners receive a reward when they successfully mine a block by calculating it's proof of work.

- **Validators**

Validators are used in proof of stake and are also known as forgers, they compete to get the proof of stake in the block transaction validation and hence so forth achieve a consensus which is primarily distributed in nature.

- **Proof of work[4]**

Proof of work algorithm confirms blocks before their addition onto the blockchain and consequently results in putting new blocks onto the blockchain. All competing miners have to solve an intricate mathematical problem to validate a about to be added block and this process is referred to as mining and the complex mathematical problem that must be solved is termed as proof of work. This makes hacking impossible as the time and resources required by the hacker to calculate proof of work pertaining to each block of the blockchain supersedes the benefit he would derive from getting the tampering of data done. The addition of a block in a typical blockchain usually takes 10 minutes due to the time required to calculate proof of work for any incoming block.

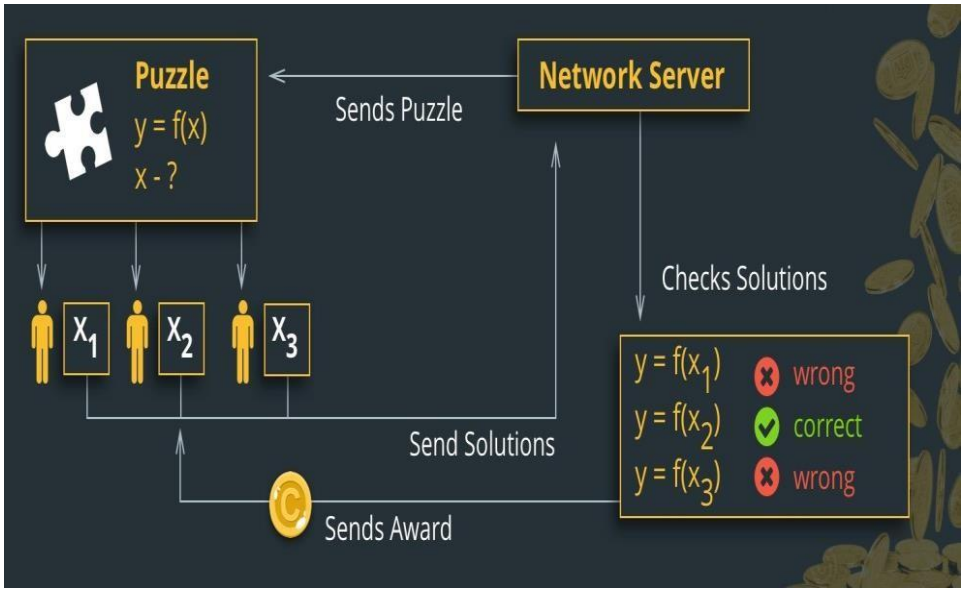


Figure 1.5 Working of proof of work and mining process

- **Proof of stake[5]**

Proof of stake has validators instead of miners. The validators put some of their ether on stake in the blockchain network and then bet on the block they want to add into the ledger. When the block gets added they get a percentage of reward on their stake. The validators also known as forgers are selected on two basis: the number one is the value of stake. Higher the value of stake of a forger the higher are his chances to select him as a forger. And the second element is random selection. This is done to avoid the wealthiest person to be always selected as the forger. The value of stake increases the probability of forger being selected but does not ensure his selection due to the randomization algorithm.

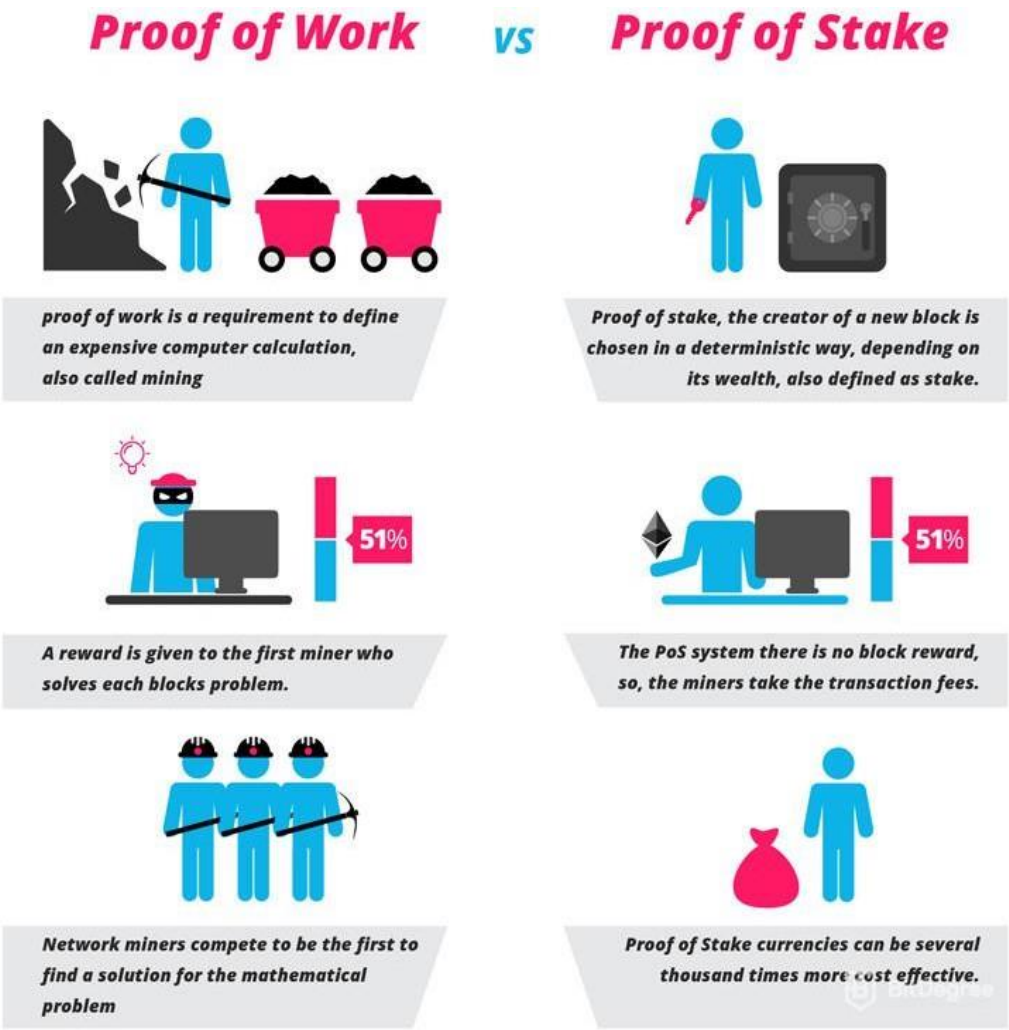


Figure 1.6 proof of work vs proof of stake

1.1.4 Traditional electronic health record systems[6]

Traditionally patient records and data are stored in database systems of individual hospitals and healthcare providers and are called electronic health record systems (EHR). This storage technique faces a lot of issues and problems. Primarily, patient has no control over his data and all the authority and administrative rights of the data lie with the respective hospital. This leads to lack of transparency as the patient cannot see who is getting access to his records and history. Moreover, the patient

cannot even access his own records whenever he wishes to do so as all the administrative privileges lie with the

healthcare provider. This problem of administrative privileges also leads into another issue of lack of interoperability. Lack of interoperability arises due to the fact that healthcare providers, hospitals and doctors are not willing to share the data with other hospitals, healthcare providers and doctors due to the fear of loosing the patient to some other rival provider. Due to this patients face a lot of inconvenience as they have to get the same tests done over and over again each time they visit a new hospital. Since, sharing of data among hospitals is something that never happens, the most valuable asset of healthcare system intelligence i.e patient records are fragmented over different provider's database and no concrete and single record history can be extracted. The electronic health record systems also lack security and data privacy. These systems are prone to data leaks and security hacks. Patient records get tampered with and stolen for various malicious reasons. Another common issue with EHR systems is patient record mismatch which has caused a lot of patients to suffer serious implications. Patient records are often interchanged with each other and this swapping of data among patients leads to inconsistency of the patient records and causes impedance in proper treatment of the patients. Since there is no consensus based validation of data, the records can be duplicated as well as can be erroneous and fallible. This leads to overloading of data as well as causalities in the hospitals. The electronic health record systems being stored on individual databases of the hospitals leads to increase in cost of storage incurred by the hospitals. With the increase in patients treated the database capacity has to be increased by the hospital administrations which can be a costly affair to the hospitals. Hospitals ,therefore, are not willing to invest in database infrastructure and so adopting the EHR systems becomes a big problem.

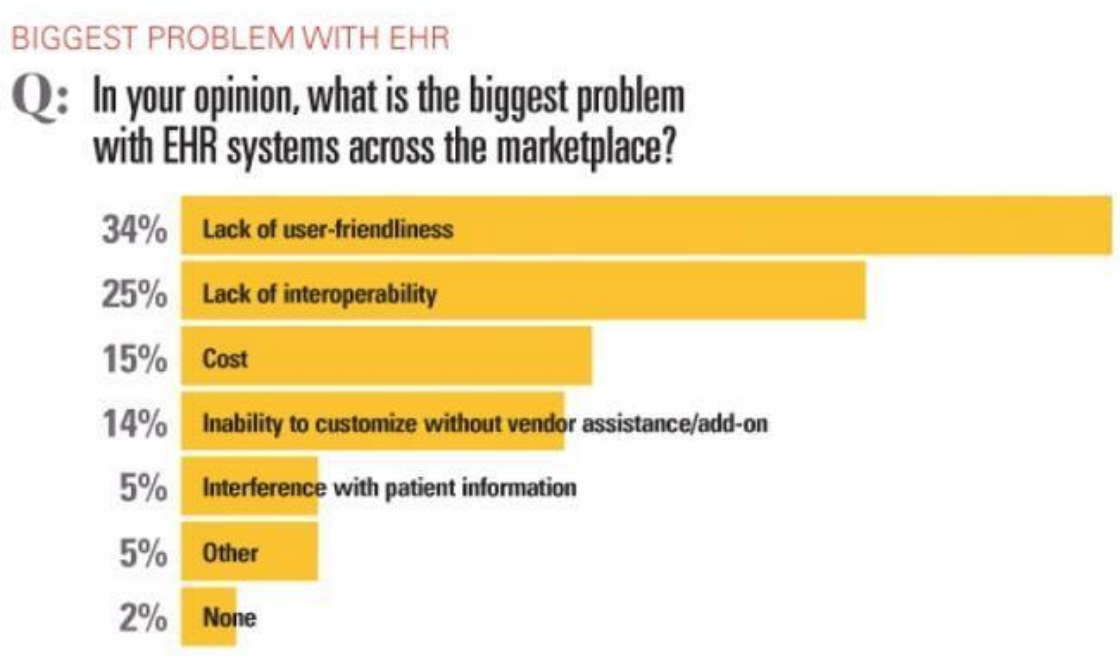


Figure 1.7 Problems with EHR systems

1.2 Problem statement

Lack of a secure, transparent, interoperable and efficient way to store patient data and records digitally.

Traditional EHRs were created to save the data digitally on databases of individual hospitals, doctors, and healthcare providers, hence, shifting the paradigm from paper based record storage to digital storage. But, with the EHRs facing monumental problems such as record mismatch, security, interoperatibility, cost and patient centricism it became necessary to develop a more secure, transparent and efficient way to store patient records digitally. EHRs are prone to security attacks, lack interoperatibility between hospitals, are cost ineffective and do not provide administrative privileges to patients to view, access and share their own data. To overcome the aforementioned problems a blockchain based patient record storage needs to be developed.

1.3 Objectives

Creating blockchain for efficient, secure and transparent storage of patient data in healthcare.

The objective of this project is to overcome all the problems that the traditional electronic health record systems pose by using the concepts of blockchain and applying them in storage of patients health records such that the system has the subsequent advantages. The system created will be patient centric in the sense that patient will have the access to his records and will decide who can access the data. Since, the storage technique is based on blockchain it will have the key property of blockchain i.e being secure and immutable which means that once the records are added into the ledger they cannot be altered. This provides an essential feature that the database will be free from manipulations and data leaks. Being a consensus based patient record storage system the data entered will be verified as well as accurate. Patient being the center of his own record also means that he can present his records to any healthcare provider he wants to and hence introduce interoperatibility between healthcare providers. This will also lead to transparency in the storage system with the patient being the focal point of all operations.

CHAPTER - 2

LITERATURE SURVEY

This unit describes the literature surveys and papers which are put forward by researchers and scholars. Various prototypes and methodologies that have been proposed by them and the challenges faced are also described.

2.1 " A Case Study for Blockchain in Healthcare: "MedRec" Prototype for Electronic Health Records and Medical Research Data"[7]

This literature survey pitches a need for innovation in EHRs & furthermore states that data personal customization is prompting patients to become more engaged in their medical records and re-establish control over their own personal heathcare data. The paper proposes MedRec which is system developed on the foundation of blockchain techniques and practices and inherits all key features of the technology. The proposed solution provides patients with an easily accessible, comprehensible and immutable storage of their data across various medical providers. It makes use of and leverages blockchain technology and it's unique features and therefore MedRec manages accountability, privacy, authentication and data sharing considerations while dealing with sensitive information and data. Medical stakeholders are incentivized to become a participant of the network as a blockchain miner. This helps in sustaining proof of work by them getting access to data whenever they act like a miner and do the proof of work. The motive here is to expose a working model through which the potential to blockchain in healthcare and IT can be discussed and analysed.

2.1.1 EHR issues

EHRs were not developed to work and operate with multi-institutional and long life medical records. Patients leave their data distributed along various providers as they have to move from one organization onto other. This makes patients to loose accessibility to their medical data because the stewardship lies with the provider. Patients can thus , interact with their data and access them in a fractured and fragmented way. An additional barrier to efficient data sharing is interoperatibility challenge. Medical data is also a necessary component of research work in the medical field but availability of an integrated and consistent database hinders the research work .

2.1.2 MedRec as a solution

MedRec is a solution to EHRs and is constructed on the foundations of blockchain technology. The technology uses public key cryptographic hashes to produce an immutable, timestamped and appending exclusive chain of data blocks. The copy of the data blockchain is distributed among each node of the proposed system. Proof of work helps to safeguard the contents and make them tamper proof. The MedRec model addresses four key problems: fragmentation of information, interoperatibility, time-consuming accessing of records and patient agencies. Integration of MedRec and currently already existing provider databases is achieved with the help of modular APIs. The technology and methodology is presented as a attempt to tap into the medical space to illustrate innovative EHR solutions rather than as a panacea for medical record management.

2.1.3 System Implementation

The content of a block, for MedRec, signifies information record ownership and viewing permissions which are shared in a private, safe and secure peer to peer network. This aids the implementation of smart contracts which helps to track and automate certain system changes. These smart contracts store and record the relationship of the provider and his patient which cause the association of medical records with viewing and data retrieving commands for processing on external databases. A hash of every data record is maintained to make sure data integrity is there. Patients do authorization of record sharing amongst various mainstays and the providers can add records. The entity which receives new data gets a notification and can authenticate the incoming record prior to it's addition onto the ledger. A singular reference point for checking of updates in the medical history is provided by providing a dedicated smart contract that sums up references to every provider and patient relationship. Confirming of the identity is done by public key cryptography and mapping an existing form of ID to a entity's address on the network.

2.1.4 Smart contract structures

2.1.4.1 Registrar contract

It is a contract that plots identificating string of participant to their network address. The purpose of using string rather than key format is due to the requirement of compatibility and integration with existing ID forms. The coding of a policy with the contract helps in regulating the registration of new entities or making changes in the ones that are already existing. The registrar contract furthermore relates a identity strings to addresses on a system network.

2.1.4.2 Patient provider relationship contract

It is created between two system nodes where one is responsible for storing and managing records for second node. Patient-provider relationship contract elucidates a pool of data pointers and corresponding permission accesses. Every pointer holds a query string which on execution on a provider's database, returns back a subset of patient health information record. Hash of this data set is affixed with query string to guarantee integrity of the data and that it is not tempered with. A hash table maps addresses of the viewers to an additional query string so that patients are enabled to share records with others.

2.1.4.3 Summary contract

It contains a reference list to patient-provider relationships signifying previous and current engagement of participant with different entities of the system. Example:- the patients will possess reference list to all providers they are engaging with in their summary contract. Patients can rejoin the network for any number of times and for any chosen period and can again get their history by downloading latest version of blockchain through the system. Till, there are participating nodes, blockchain log is maintained. User notification functionality is also maintained by this smart contract.

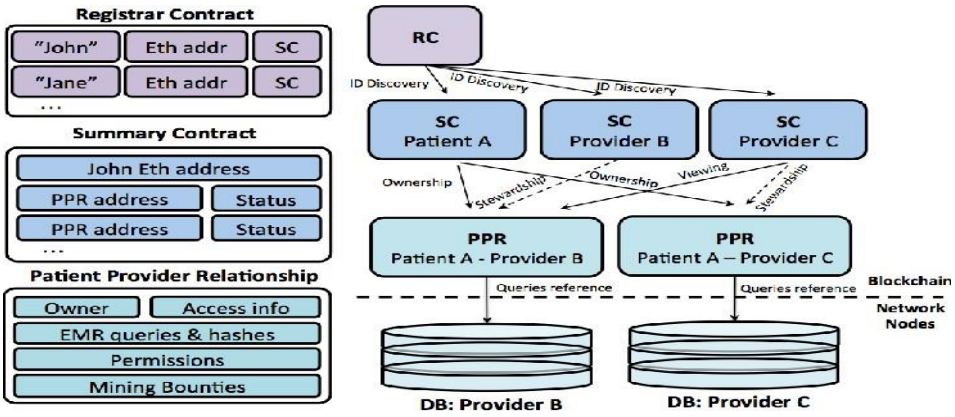


Figure 2.1 data content of MedRec contracts and sample relationship graph.

2.1.5 System node description

The designing of the various system components is such that they can be integrated with existing EHR infrastructure. There are 4 integrants in the network system: EHR manager, Ethereum client, backend-library and database gatekeeper.

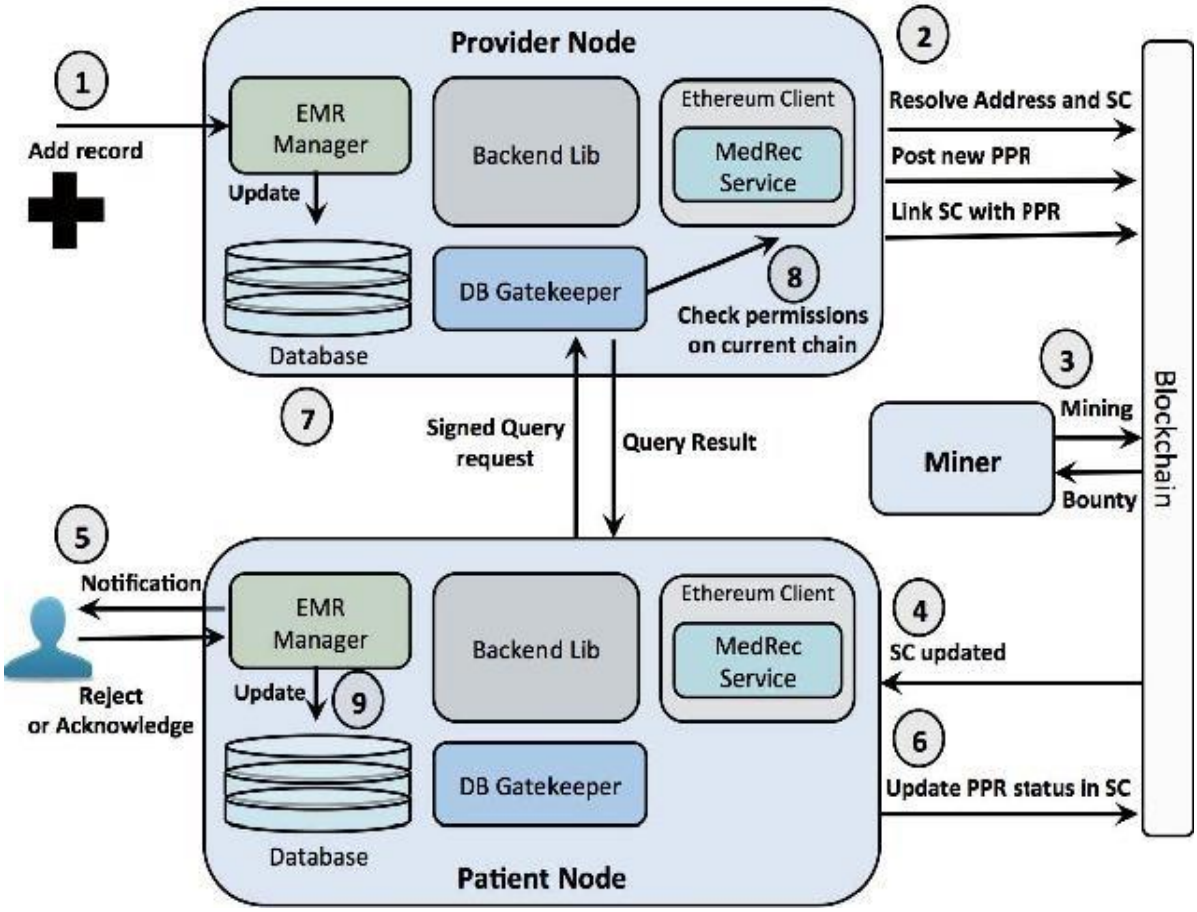


Figure 2.2 record addition by provider.

2.1.5.1 Backend API library

Numerous utilities are provided in a backend library to support operations in the model. Exporting a function call and communication with the blockchain is abstracted by the library. The library handles, automatically, the uncertainty when the mining of

transactions is done and deals with discardation cases. The library links and communicates with the etherium client to handle and do the parsing of ethereum protocol and low level formatting.

2.1.5.2 Ethereum client

It is responsible for the implementation of complete functionality which is needed for participation and joining in the network. Tasks such as peer-peer networking connection, sending & encoding of a transaction and storing a local and verifiable duplicate of blockchain.

2.1.5.3 Database gatekeeper

It is responsible for implementation of a off the chains, interface for accessing a native node database which is controlled and supervised by different permissions saved onto the blockchain. It deploys a server for listening of query requests from clients. Issuer cryptographically signs a requisition request and this helps gatekeeper to confirm identities. The gatekeeper then checks blockchain contracts to check if access can be granted

2.1.5.4 EHR manager

All the previously mentioned components are tied together using EHR manager. Data from local SQL databases is rendered for viewing and notifications are also provided with data sharing and data retrieval options.

2.2 "A Systematic Review of the Use of Blockchain in Healthcare"[8]

This systematic review aims at exploring the possible applications of blockchain techniques and features and to showcase the possible direction & challenges of the technology in the health sector.

2.2.1 Distributed consensus protocols

Peers of a blockchain network must have a consensus on a particular state in which the distributed system is and on record packing techniques and strategies. It is essential for any blockchain to be functional. These agreements are called distributed consensus protocols and they validate the chronological order of transactions which are generated. It also makes sure that quorum of peers of blockchain network have an agreement on the system's current scenario and condition and therefore also of the sequence of addition of records and data.

A consensus protocol elucidates how a participating node seals & prepares the latest block which contains non formatted and unconfirmed information. The concept behind above mentioned protocols is that a peer contributes something valuable to the network system and in return gets awarded for the contribution. This promotes a competition in the blockchain network in which the competitors keep a check on each other and thus help in achieving a consensus. The proof of work protocol considers the computations strength as a determinant for choosing the network participant which becomes the miner and wins the competition. The proof of stake protocol on the other hand is based on assets that a peer possess. Proof of stake reduces chances of a validator attacking the blockchain as it would mean that he is attacking his own assets. Another version of proof of stake algorithms is the distributed proof of stake in which stakeholders vote to decide the validator.

2.2.3 Research questions

The review paper was developed to gain insights on various three primary questions. The questions were raised in relation to blockchain technology and its consequent use case in the healthcare. The three primary questions which were raised and aimed to solve and answer are:-

- What is the present scenario and condition of blockchain in the health sector and how has the technology in itself and it's implications changed over time in the past few years?
- What all are the various explorations and researching patterns and course of movement in the technology and techniques of blockchain and it's use case in the health-sector?
- Which components of blockchain are implied in the publishing of health-sector?

2.2.4 Data sources

Information was collected from numerous sources and was used to prepare the review and it's conclusions. The various sources of the information used in this literature review are:-

- Sci hub
- Springer SpringerLink

- Elsevier ScienceDirect
- ACM Digital Library
- Cochrane Central
- Research gate
- Google Scholar
- Clarivate Analytics—Web of Science (WoS)
- IEEE Xplore
- PubMed
- Medline EBSCO

Data was gathered and analyzed from above mentioned sources and answers to questions asked in the beginning were formulated.

2.2.4 Discussions on questions

- What is the present scenario and condition of blockchain in the health sector and how has the technology in itself and it's implications changed over time in the past few years?

It was found out that blockchain and it's use case in the health-sector is a fresh and rather undiscovered concept but is increasing in usage with time since 2007.

- What all are the various explorations and researching patterns and course of movement in the technology and techniques of blockchain and it's use case in the health-sector?

It was found that most publications in this technology are intended for information sharing, access control and health-records. Furthermore, the technology is barely used for audit trail management, managing of supply-chain and some other use cases such as drug prescription management and auditing. It was thus concluded that blockchain is reasonably used in sector of healthcare.

- Which components of blockchain are implied in the publishing of health-sector?

This review found that ethereum and hyper ledger are most preferred frameworks and platforms for development of blockchain. For healthcare, a private or consortium blockchain was preferred as access control in healthcare is a major issue and data modifications and additions done by anyone is not desirable. Proof of work turned out to be the most popular consensus algorithm.

2.3 "The Potential of Blockchain Technology for Health Information Exchange: Experimental Study From Patients' Perspectives"[9]

This review paper takes into consideration the patient's point of view towards blockchain based healthcare record systems. Little is known about what patients think of blockchain based EHR system networks. The paper was aimed to explore the considerations that patients have for the system. Various sharing methods and techniques are in use from public as well as private healthcare providers to support the initiatives of information exchange. Various surveys in healthcare information exchange reveal that there are currently three exchange models which are applied by healthcare providers to transmit patient data electronically: direct model, query model and centric to patient.

- **Direct model**

This model has a working mechanism where encrypted patient records can be shared by a healthcare provider with a pre-identified receiver. This provides and supports point to point communication where recipient's identity becomes known to sender and medical records of patients can be exchanged directly via adopted mail protocols from one healthcare organization to other. Direct model, has the mutual understanding and cooperation between various organizations as it's cornerstone and incorporates information transactions on the receiver's EHR system in a confidential and safe network which is supervised by healthcare providers. This system can make communication and coordination between organizations effective as it safely interchanges information pertaining to patients.

- **Query based model**

It gives healthcare mainstays the capability of requesting & finding healthcare data records of a user from different mainstays. Centralized data warehouse has a very important part to play in this technique as aggregated EHRs will be stored from EHR systems of various organizations and into a central hub. This facilitates the query to be processed from a central data storage pool. This approach is incepted for production of an effective tool to give aggregate, multi-provider patient information for the standard and performance metric report generation & registries pathology research.

- **Patient centred exchange**

In this model patient information pertaining to different transactions having diagnosis are transferred from the mainstays to users. For example, the users can lookup to their lab results, radiology reports, and medications after each visit which are stored on user warehouses and then transacts the information and medical data with the mainstays. This engages patients and initiates participation in their entire medical procedure , handle data they have ownership of and act as a principal entity of information sharing procedures by considerating a way of mediation for the users.

2.3.1 Role of patients in healthcare information exchange

With heavy reliance on electronic methods of storing data a risk arises as patients may be concerned that more than required data might be shared on the network and information can be erroneous which will be then fallaciously shared between providers. Patients are key stakeholders in the Healthcare information systems systems as user permission and acceptibility is required in all data storage and

sharing procedures. Patients are a significant producer of healthcare information and their perception and attitude towards HIE refrains them form sharing information. If users are not willing to make their data accessible to different mainstays then it can result to outdated, incomplete and even inaccurate information. This will obstruct healthcare providers from providing a useful, reliable and integrated healthcare information. Support of public is essential to the success of healthcare information exchange initiatives. Previous studies lay down the fact that people are not actually fully cognizant of the data and record sharing process & it's application in the current transfer protocols. Positive attitude and perception of HIE will be created in patients by the means where user information is gathered, deposited and shared in a confidential way. If the security expectations of patients are not met then he will hide future details of his healthcare data.

2.3.2 Primary issues with current systems

- **Trust issues**

Whenever a gap exists among the suppliers and providers, like in a web based exchanges, trust plays a pivotal role. The core in HIE settings is patient's trust in the system as security and privacy issues may entail. Patients need to be made assured that their healthcare information will not be misused or leaked in any way and so trust makes up an important factor of how patients react to the HIE systems. Thus, a confidence must be established in the user's mind for the HIE systems prior to them taking any step towards committing to an opting of the system or revealing their confidential and important health-data. Belief and faith into HIE performance, reliability and efficiency elucidates how much will users depend on the efficiency and functioning of healthcare information exchange to efficiently distribute data among an assortment of various conglomerates and providers. Moreover the trust is

pertaining to a credence of accord between patients and providers is dependable, trustworthy and genuine. Distance between healthcare providers and patients and the lack of communication and interaction between them is the primary cause of lack of trust in HIE systems

- **Privacy concern and privacy policy**

Whenever a networking-oriented technology (like the one we are referring to in our use case) work with confidential and private data, the chances are very high of it violating privacy of the users.

The competence of patient databases and HIE's is influenced by information privacy and this may lead into inaccurate treatment, increased mortality, wasteful investment and erroneous care of patients. Preceding surveys and reviews strongly bring out the fact that users remain strongly worried of loosing command and authority on the process in which their information is handled and exchanged on the HIE systems. This concern arises primarily due to absence of transparency and privacy policies and confidentiality protocols.

2.3.3 Experiment design

Sixteen scenarios were designed to analyse perception of healthcare consumers regarding all risks & potentials that come with the implementation of the different prototypes of HIE that are built on various architectures. The architectures of four HIE models vary according to two factors: transparency in the privacy and confidentiality policy and sensitivity of healthcare data. The models were subdivided into 16 scenarios where each scenario pertains to a different experiment. Between subject experiments were also used in which participants are made to participate in one experiment each in an ad-hoc way.

Respondents were randomly exposed to different exchange models and were explained their features and privacy policies.

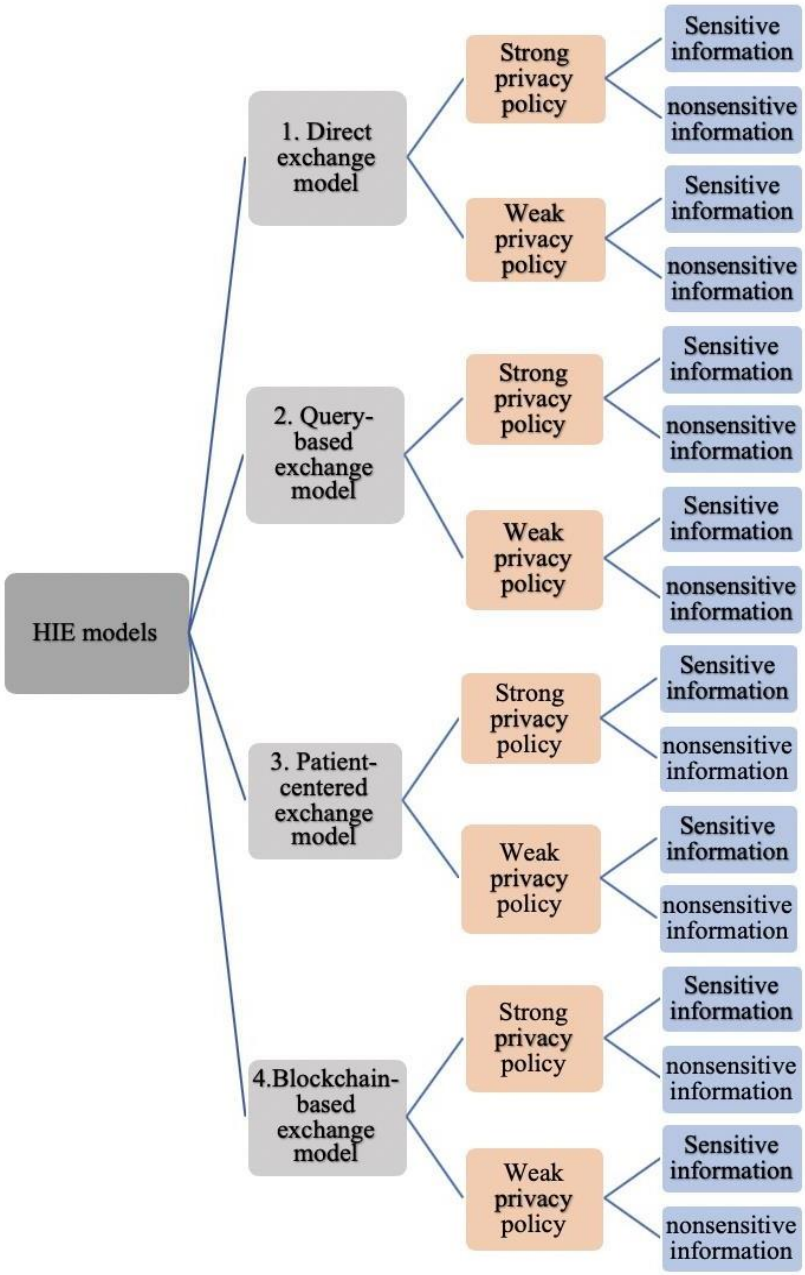


Figure 2.3 sixteen scenarios of experiment

HIE[a] model	HIE architecture							
	Strong policy/sensitive information		Strong policy/nonsensitive information		Weak policy/sensitive information		Weak policy/nonsensitive information	
	Experiment #	n	Experiment #	n	Experiment #	n	Experiment #	n
Direct exchange model	1	128	5	123	9	128	13	132
Query-based exchange model	2	128	6	131	10	124	14	126
Patient-centered exchange model	3	122	7	126	11	127	15	125
Blockchain-based exchange model	4	120	8	125	12	126	16	122

[a]HIE: health information exchange.

Figure 2.4 Experiment design diagram where n is number of respondents

2.4 "Applications of Blockchain in Healthcare: Current Landscape & Challenges"[10]

Consuming more than ten percent of GDP in a majority of the countries, health-sector is one of the globe's most biggest sectors. Since it is an intricate mechanism of interdependent elements which face strong scrutiny and supervision from regulatory bodies, user information is majorly sparsely distributed and tariff of health-care service is rapidly increasing because of some loopholes in the mechanism and heavy reliance on many mediators. Moreover, transparency in system is also lacking and control for the patient is yet to be established. Patients have often raised concerns that information is getting exploited for profit organizations and elements. For assisting solving the complex healthcare data problems, revolutionary technology of blockchain comes to rescue introducing decentralization and transparency. Scattering of data over various organizations (data silos) and a multi level permission control system required for the access of data has made it difficult to access important data at the time of urgency. Blockchain can create a access point for all healthcare record pertaining to any patient. Users, thus, become the primary administrators of their data and can control flow of information sharing as well. Immutability of blockchain provides information authenticity & security which can be used to construct efficient diagnosis for putting a halt to drug counterfeiting and medical frauds. Personalized regulations, restrictions and rules are laid down in a blockchain using smart contracts. These smart contracts play a crucial role in laying down concurrence and accords between users, health-care service suppliers and other stakeholders like insurance companies & drug organization.

2.4.1 Areas of application

- Drug discovery and pharmaceutical research
- Genomics
- Imaging
- Electronic health records
- Supply chain and counterfeit drug detection
- Prescription management
- Billing claims management

2.4.2 Analytics

Blockchain is competent of creating and leaving a humongous constructive footmark on the 3 crucial support stands of analytics: data, model, and computation.

- **Data**

Key for a better analytics is access to right and accurate information. Using blockchain model based HIE, organization gain the capability and the tool to acquire the data needed and required for analytics. If data is stored on a blockchain model then it is easier to trust it's integrity and provenance.

- **Model**

Blockchain is capable of grouping multiple model predictions together for creation of a complete sturdy final model. A collective model is always better than individual predictions.

.

- **Computation**

The increase in model complexity and data quantity has increased the demand for computing resources. Blockchain has now introduced a new platform and concept of computation known as decentralized computing. Blockchain provides a cheaper computational platform as compared to traditional cloud based computations.

2.4.3 Challenges to blockchain

Despite the huge potential that blockchain possesses, the current state of blockchain has it's own limitations. The biggest problem with blockchain is that the proof of work algorithm makes addition of new records and transactions onto the ledger very slow. Typically, a new block takes 10 minutes to get added on the blockchain. This speed is highly inefficient to deal with current world scenario where transaction frequency reaches thousand per second. Moreover with addition of more blocks the space taken by blockchain becomes exponentially large and this makes it really hard for the stakeholders of the network to retain a complete duplicate of the chain. The three pillars of blockchain : decentralization, scalability and consistency have a trade off amongst them. Blockchain can only provide two of them at a time. Another challenge is the challenge to integrate the blockchain HIE with currently present EHR systems. All the healthcare providers and stakeholders must be motivated to join the blockchain network and adopt with the changes. Since healthcare providers currently enjoy a monopoly over patient data and records, it is not easy to persuade them to adopt the blockchain model of healthcare information exchange. To get hospitals and providers to coordinate in the initiative, they must be given some incentives. So a new business model must be developed to attract the stakeholders to adopt the blockchain model. Another major

challenge is how blockchain is going to comply with current regulations on healthcare. Since, healthcare is a serious industry with lives at stake, the regulations and standards are also strict and stern. The basic premise of blockchain is to provide patients access of their data and create a decentralized database of records. But in some models of blockchain it is really important and necessary for someone take up the responsibility of governing the blockchain network and it's operations. This is sometimes necessary to meet the regulatory standards.

2.5 "Blockchain-based Electronic Patient Records for Regulated Circular Healthcare Jurisdictions"[11]

A circular data driven healthcare system works like a great efficient prototype and mechanism that offers a low-price and efficient health-care offerings to patients for the coming times. The concept of circular economy in the recent years has evolved to an ambitious undertaking from a niche idea. This model refers to a continuous developmental cycle program revolutionizing present mechanism of economy . It efficiently manages finite stocks and renewable flows and thus minimizing system risks. The rise in costs of healthcare causes depreciation in working capability & has also led to overcrowding. To solve the new world healthcare management issues with adherence to circular economy principles, researchers have proposed a horizontal cross functional and service oriented structure rather than traditional vertical solutions. The aim is to create medium which will integrate multifarious services & products to give a complete service portfolio to patients. This enables further opportunities such as personalized healthcare, sensor based tele health and ambient assisted living while simultaneously ensuring a sustainable healthcare environment. Non sharing of data by providers has following reasons :-

- Business logic

Every provider has it's own flow of logic regarding updating, reading and sharing of records

- Trust

Providers should be able to trust each other to keep an up to date and authoritative record base

- Privacy

What can be shared with whom and how patient privacy can be handled.

- Security attacks

Making a single source of truth for all medical data can make it seriously vulnerable to security attacks.

2.5.1 Working context

Despite having huge potential blockchain cannot solve all of the issues in the current healthcare industry and so we need to understand the working context of the application. The scope of this work is constrained to the requirement of regulated healtchare jurisdiction which is accountable to government bodies and authorities that have a crucial responsibility and job to deal with a ledger of users & beneficiaries and mainstays. Scenario examined is where authorities become the regulators of medical services rather than becoming a provider of such services.

2.5.2 Participation considerations

Since in healthcare, identification of all involved parties is necessary therefore, we use a permissioned blockchain for the application. Since it is a permissioned blockchain, the authorities will be able to decide who can run the system and validate the transactions. The blockchain transactions can be validated by all the participants of the network such as authorities, patients and providers who are then in turn incentivized for their contribution. Consumer right organizations, non government organizations and privacy watchdogs can also become validators in the network.

2.5.3 Smart contracts

These can be considered to be cornerstones for all the records and entities which are stored programs on blockchain which are triggered when transactions are done on the system. These on a system can be distinctively identifiable, has the ability to store and save state & do pre-defined execution. Smart contract of this system has following key components: -

- **Identity registry**

This is a contract that contains addresses pertaining to all the entities which work in the system & relates those entities to real world identities. It is going to be maintained by the regulatory authorities.

- **Patient record**

The actual medical records of the patients are stored in this medical record. The beneficiary himself owns this smart contract.

- **EPR access agreement**

It's an accord between patient and the providers which specifies which part of the patient record can be accessed by the provider. The access scope and access conditions are defined in this contract. This contract is owned and even created by the beneficiary i.e. patient.

- **Agreement list**

This contract has a map of owner EPR agreements which are made with different stakeholders.

2.5.4 Data model

The data model is implemented in solidity using mapping constructs for key value pairs. Locations of csontracts can be referenced using the address type. Agreements and metadata are augmented together to denote validity of data and contracts.

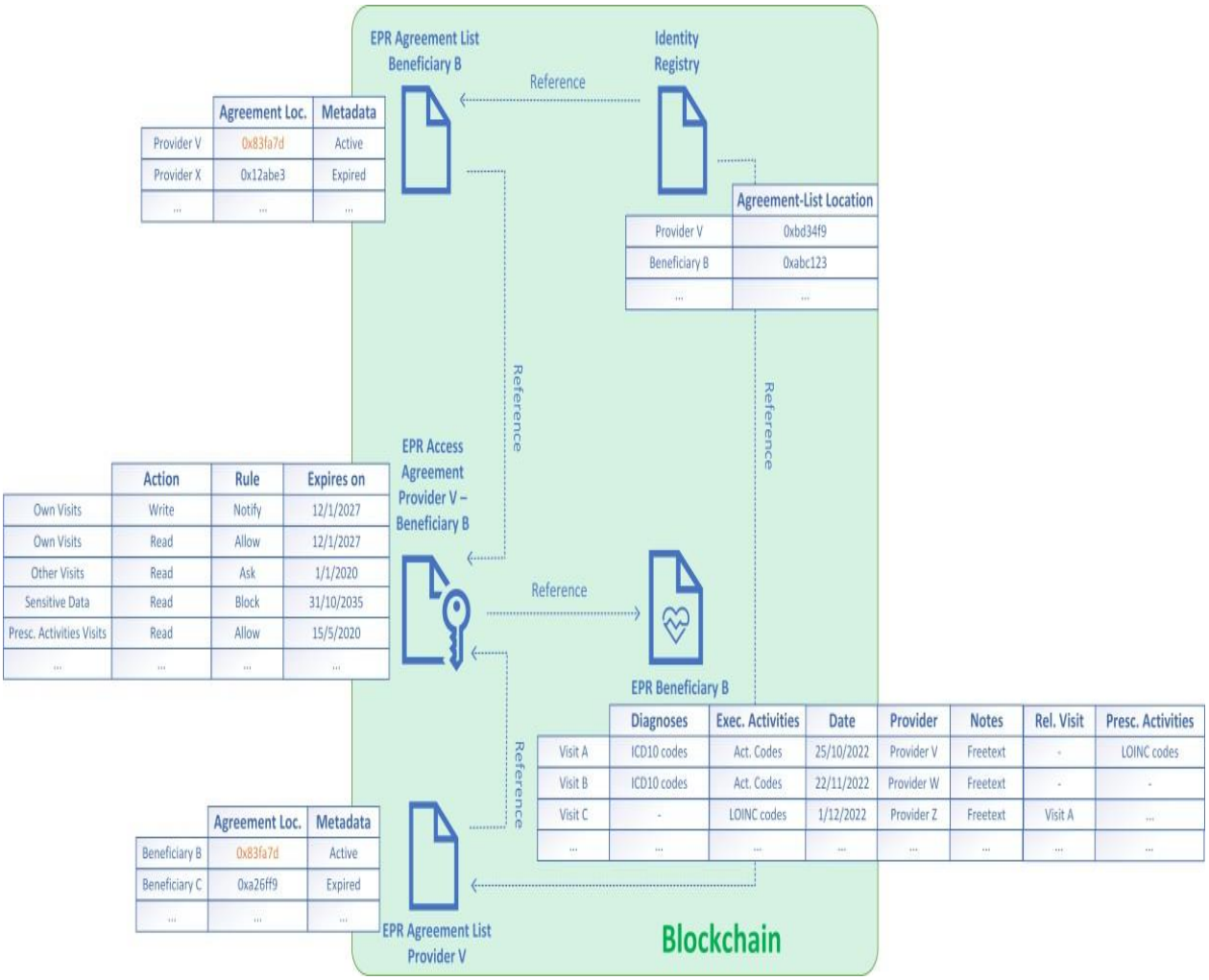


Figure 2.5 data model

2.5.5 Operative cases

- Generating Beneficiary-Provider accord

In it the beneficiary must be mapped to a doctor of first resort and so this resembles the concept of personal doctor. The personal doctor can have more data access privileges than any other provider. In order to create an agreement a provider must loop up to the patient's agreement list via identity registry contract and request access. If this request is granted then the EPR access agreement is created.

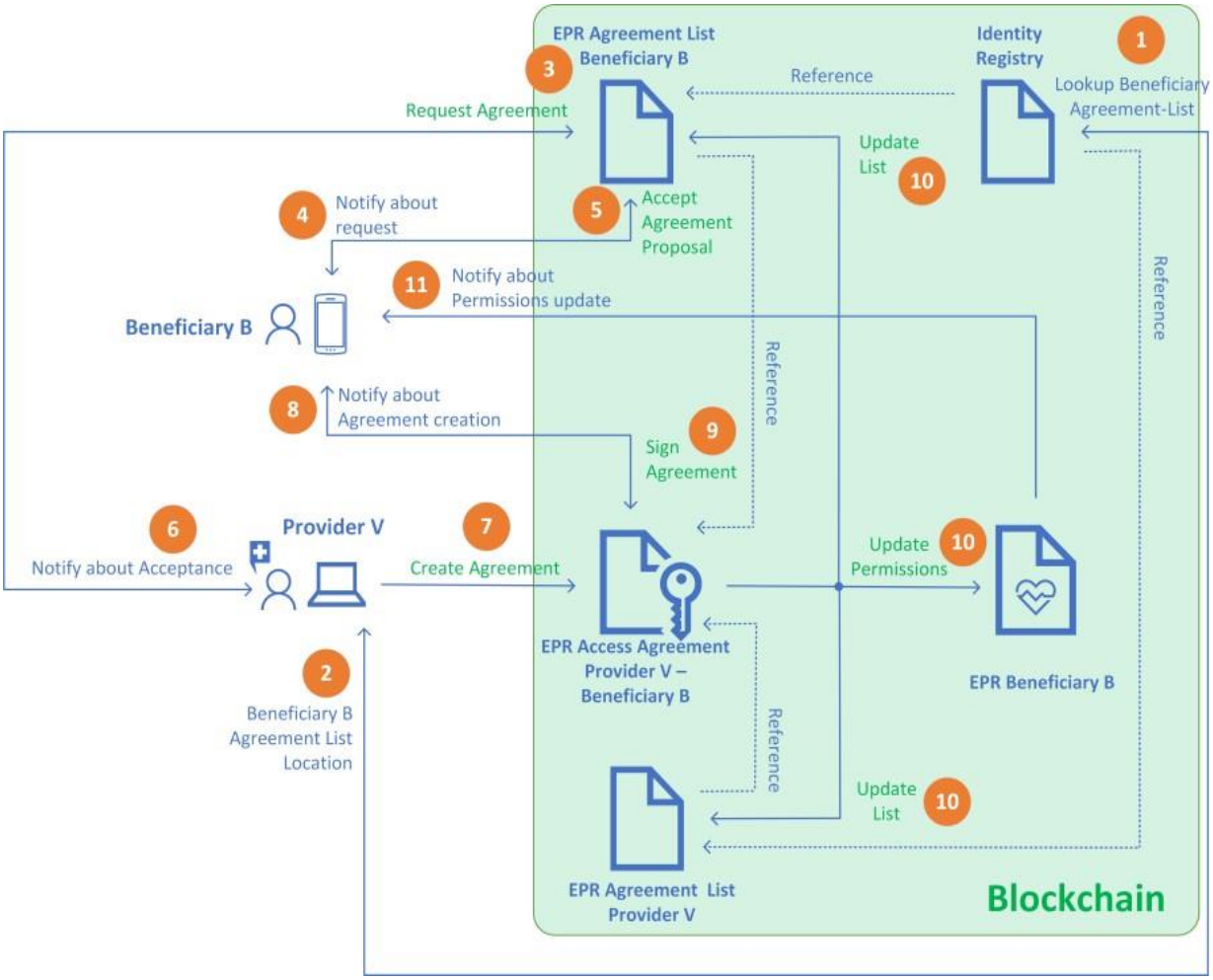


Figure 2.6 Provider requesting EPR access agreement

- Allowing provider access to EPR

A contract's access rules may be organized and handled directly using an off chain application. Access to patient data contract by mainstays can occur exclusively through a EPR access agreement which then returns requested data.

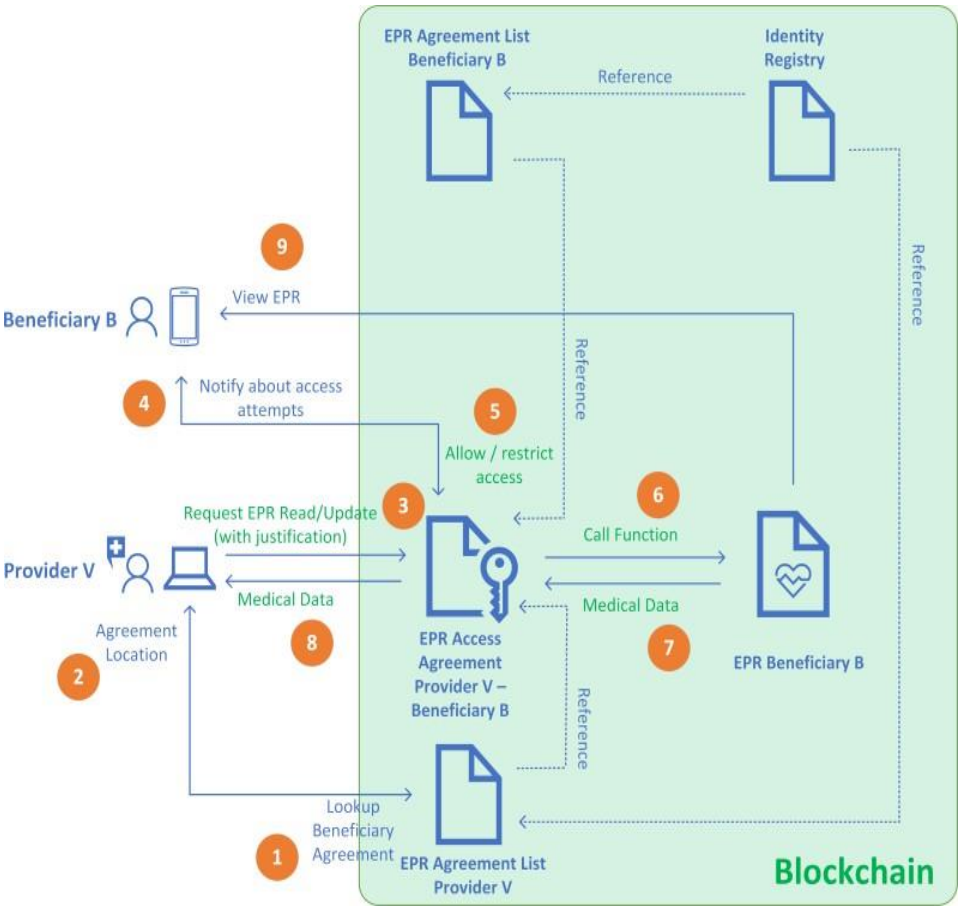


Figure 2.6 Provider accessing an EPR

CHAPTER – 3

SYSTEM DEVELOPMENT

This chapter describes the proposed model, the working and building of the basic prototype. The requirements of the hardware and software have also been described.

3.1 PROPOSED MODEL

- **High Level Use Case**

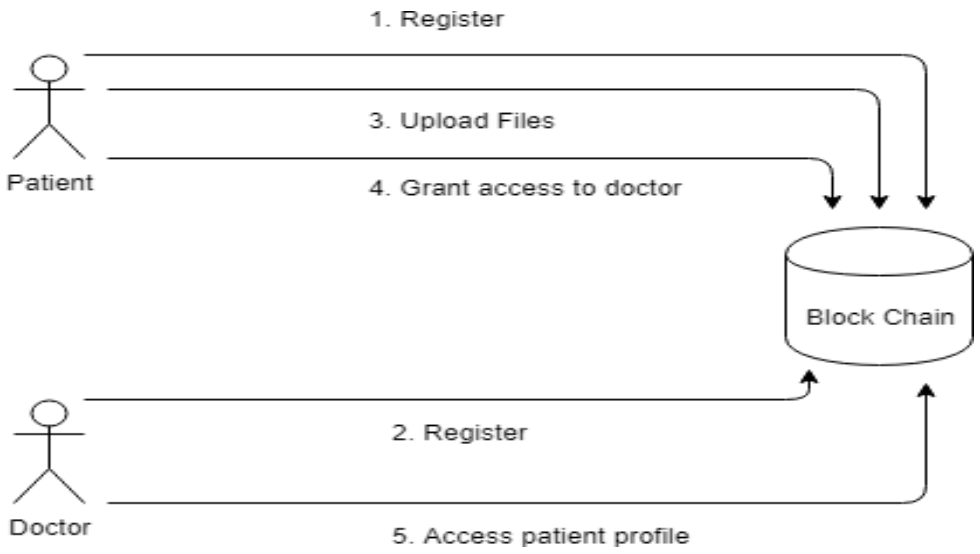


Figure 3.1 – High Level Flow Diagram

Overview of this project can be broken in three parts – patient side, doctor side & the actual blockchain.

The patient registers on to the site and uploads his/her basic data like age, weight, height, etc. on the server. This is saved as the genesis block of the patient's blockchain.

The doctor too registers but gets access of patient's files only after the patient allows it. The doctor gets a copy of blockchain and adds a block at the end of it which has the current diagnosis.

The blockchain is maintained by verifying every block by validating the data and signing it. Each block is mined by miners so it is immutable and safe. Below are two figures explaining the patient and doctor's side of things:-

- **Uploading of Files by Patient**

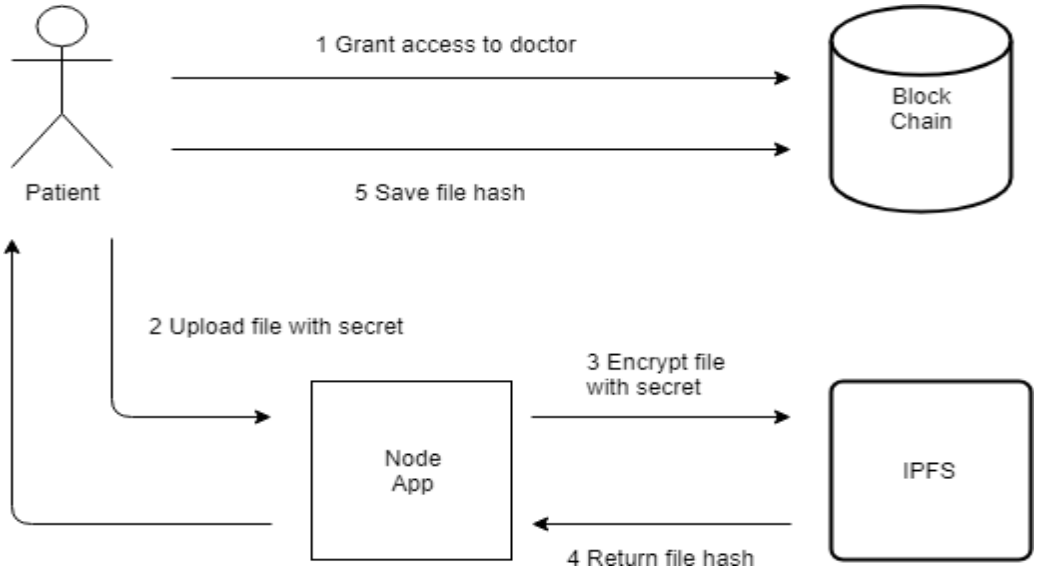


Figure 3.2 – Flow Diagram for Uploading of files

- **Accessing of Files by Doctor**

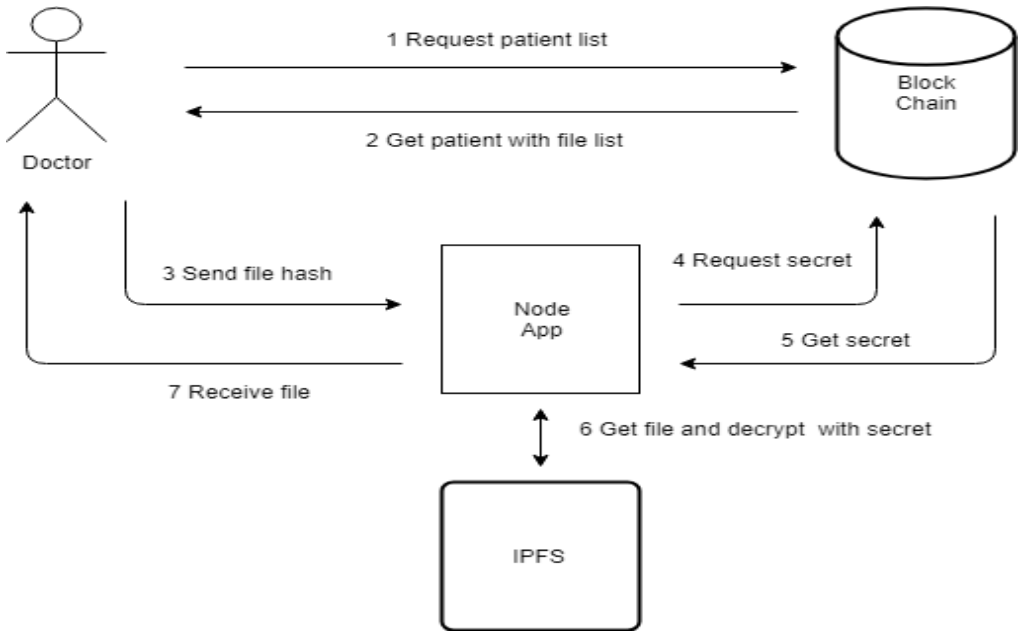


Figure 3.3 - Flow Diagram for Accessing of files

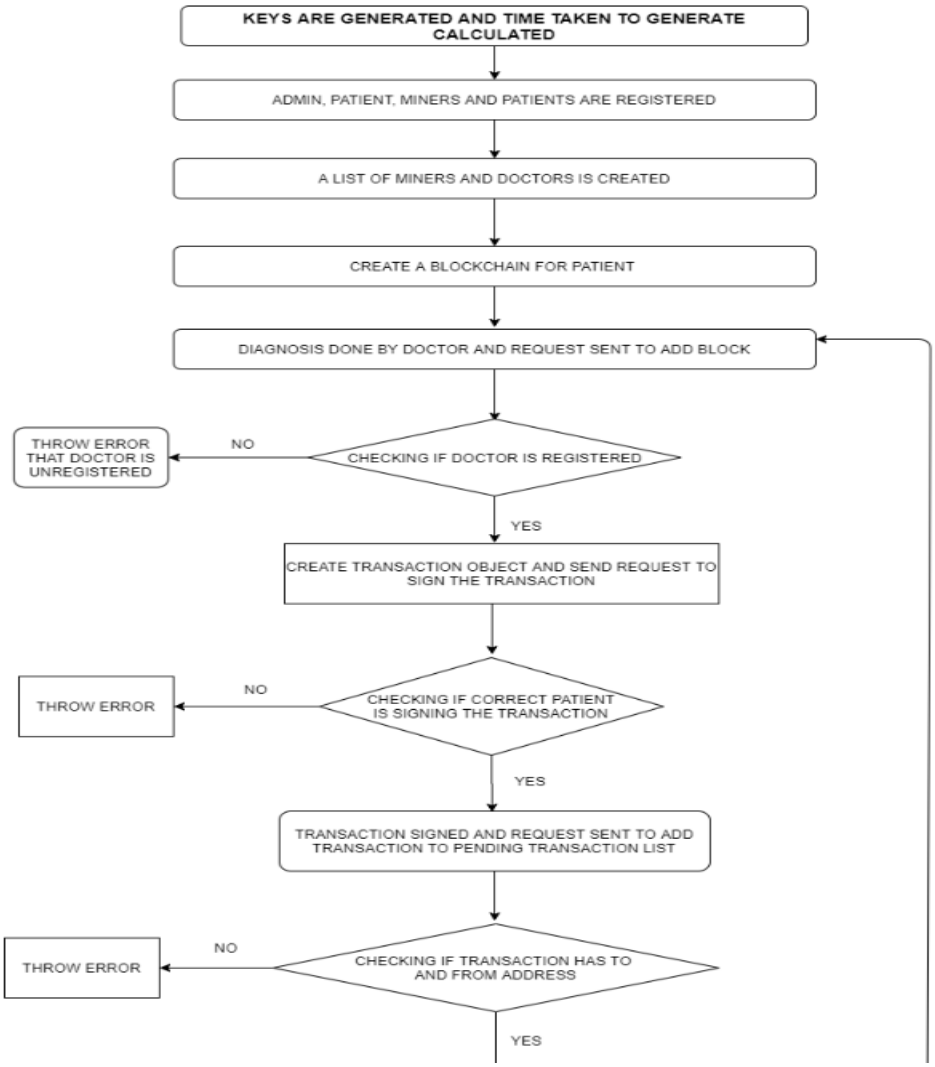
3.2 MODEL DEVELOPMENT

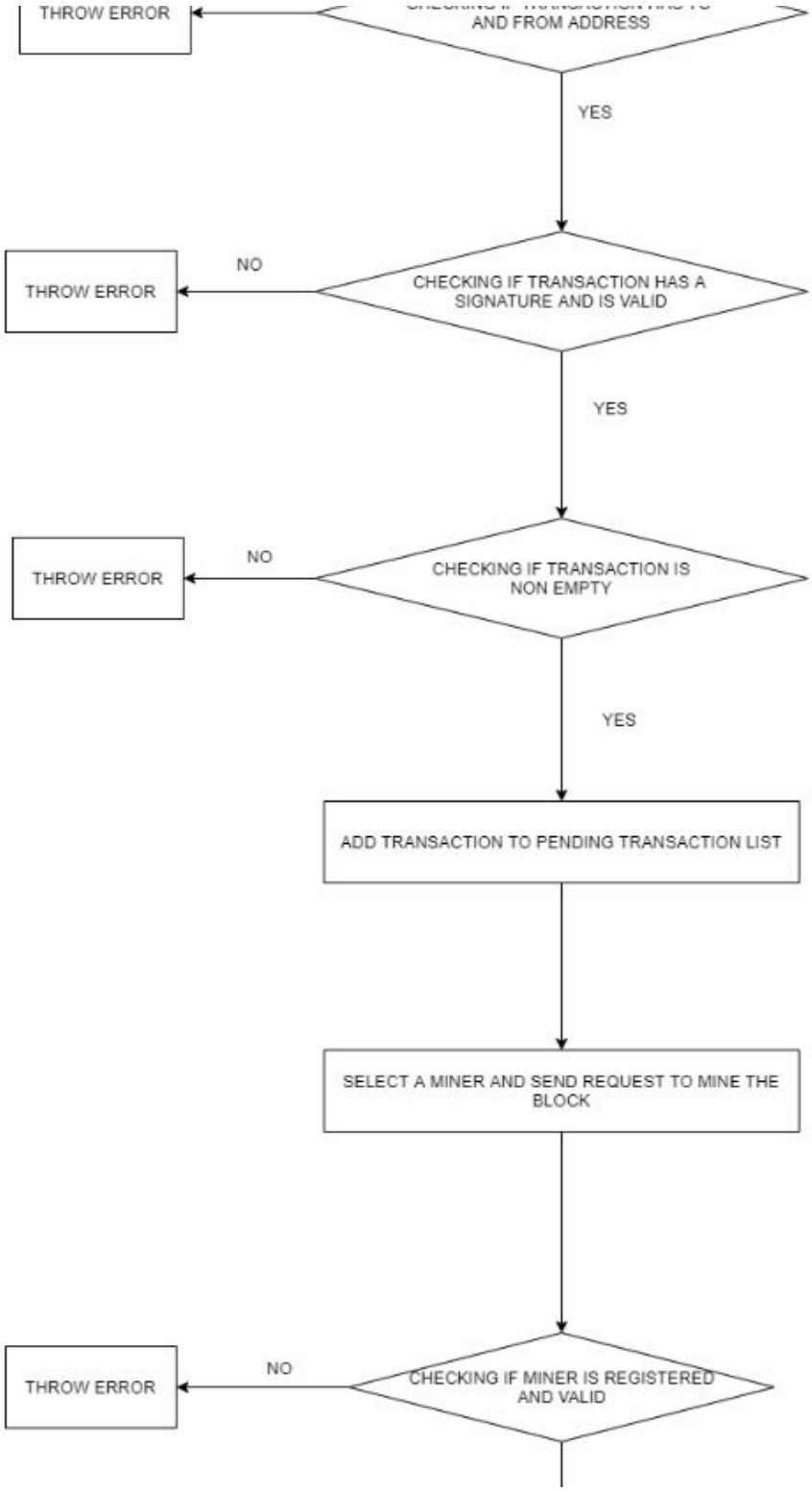
- **Hardware Used**
    - 2.2 GHz Intel Core i5 5th generation CPU
    - 8GB DDR3 RAM
    - NVidia GeForce 940M GPU

- **Software Used**
    - Microsoft Visual Studio Code
    - JavaScript
    - Node Package Manager
        - Elliptic.js Module
        - Crypto.js Module
        - Secp256k1.js Module

3.3 IMPLEMENTATION OF MODEL

3.3.1 Flow of execution of the project




THROW ERROR
AND FROM ADDRESS
YES
THROW ERROR
NO
CHECKING IF TRANSACTION HAS A SIGNATURE AND IS VALID
YES
THROW ERROR
NO
CHECKING IF TRANSACTION IS NON EMPTY
YES
ADD TRANSACTION TO PENDING TRANSACTION LIST
SELECT A MINER AND SEND REQUEST TO MINE THE BLOCK
THROW ERROR
NO
CHECKING IF MINER IS REGISTERED AND VALID

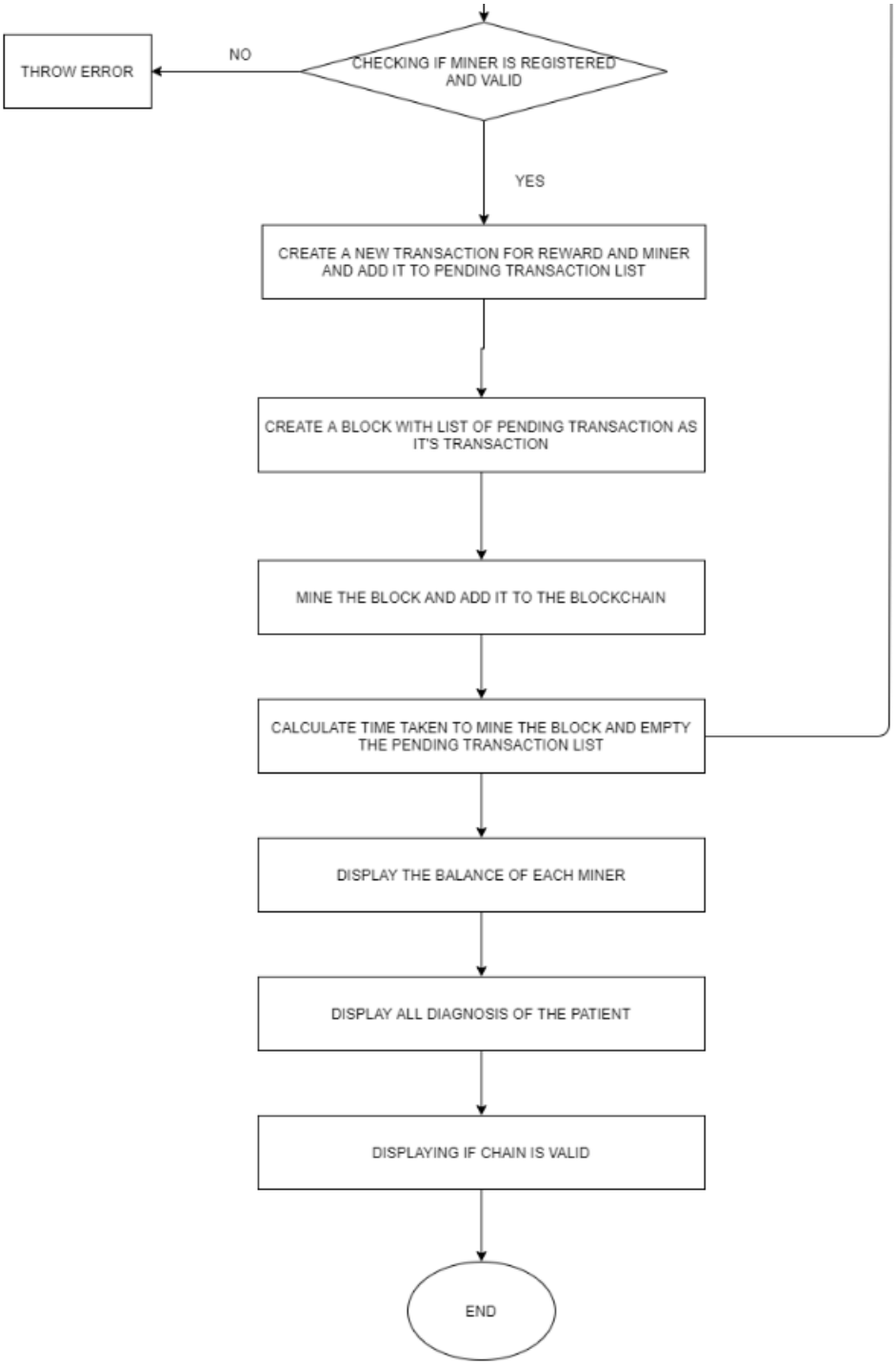


Figure 3.4 Flow of execution of the project

3.3.2 Generation of Public & Private Keys

```
pTime=Date.now();
EC = require('elliptic').ec;
ec = new EC('secp256k1');

key = ec.genKeyPair();
publicKey = key.getPublic('hex');
privateKey = key.getPrivate('hex');
cTime=Date.now();
console.log();
console.log('Time to generate the key set for current node is',(cTime-pTime)/1000);
console.log();
console.log('Your public key\n', publicKey);
console.log();
console.log('Your private key\n', privateKey);
```

Figure 3.5 – Module to generate public and private keys and display time taken to generate keys

The elliptic module is used to generate a new key pair having two 256 bit keys which are encrypted to hex format to generate the public and private key which can be used to give access and signing/verifying of account. The time taken to generate each key pair is also calculated and displayed. These keys are used to register the patient onto the proposed healthcare system. Patients, doctors, miners and the administrator can use these keys to authenticate themselves while performing various actions in the system. They key pairs are unique for each entity of the system and a total of 2^256 unique keys can be generated using this module which can then be allocated to any desired user.

3.3.3 Registering entities and creating a list of miners and doctors

```
const  {Blockchain}  = require('./blockchain');
const { Transaction }  = require('./transaction');
EC = require('elliptic').ec;
ec = new EC('secp256k1');
var miner_list=[];
var doc_list=[];
adminKey = ec.keyFromPrivate('cb3b5da42ade054461a2afbaa8d50a0235b69b48a6c4db91c29c4f8303b8274c');
adminAddress = adminKey.getPublic('hex');
myKey = ec.keyFromPrivate('ca2345d89b7b83993cbbc983cf03a3bd76485c5034fcadaea134dff4db93fb26');
myWalletAddress = myKey.getPublic('hex');

docKey = ec.keyFromPrivate('939ced0952d13aa161af79e28588f7ca3cd35c3e16bdb26c68e323eac6cd7522');
docAdrress = docKey.getPublic('hex');
doc2Key=ec.keyFromPrivate('548c36d539d6558592f2cb49f7af97de8b4912c4be6cda69fbbafc57050f83cf');
doc2Adrress = doc2Key.getPublic('hex');
doc3Key=ec.keyFromPrivate('4a8465da3b84e41012ad5df891a20c0d0f5a4c272875c5d8fc19ed81a400f209');
doc3Adrress = doc3Key.getPublic('hex');

doc_list.push(docAdrress);
doc_list.push(doc2Adrress);
doc_list.push(doc3Adrress);

mineKey = ec.keyFromPrivate('820e6ad697a8c8ea4b8a5d8140187eab978f2efdbeb1def0baa1abde58f4adb9');
mine2Key=ec.keyFromPrivate('d5b71e92c592be6f94136b3d6dee4733d89d0dcf2d482f344cda6c5191e4045b');
mine3Key=ec.keyFromPrivate('ff7208c723096ceb74c9702bda6f3fa390a433b986e6313c9deae56afa727e8a');
mine4Key=ec.keyFromPrivate('f3ae219298a840dc85b0d2e7391252421fbc18c38080bce35bbc377dba82dffb');
mine5Key=ec.keyFromPrivate('30ae967fb0e141b863945c8f8e52620d77c05571231a699e56f32f0d9a30c98e');

mineAddress = mineKey.getPublic('hex');
mine2Address=mine2Key.getPublic('hex');
mine3Address=mine3Key.getPublic('hex');
mine4Address=mine4Key.getPublic('hex');
mine5Address=mine5Key.getPublic('hex');

miner_list.push(mineAddress);
miner_list.push(mine2Address);
miner_list.push(mine3Address);
miner_list.push(mine4Address);
```

Figure 3.6 – registering entities onto the system and creating a list of miners and doctors

The miners, doctors, patients and the administrator are registered onto the system. Also a list of doctors and miners is created which will be used to validate authenticity of doctors while adding transactions and miners while mining a block. This list creation will then save the system from the attack of fake miners and non-registered doctors. This will also help in patients being assured that their doctor is an authentic doctor.

3.3.4 Blockchain class and genesis block creation

```
const crypto = require('crypto');
const EC = require('elliptic').ec;
const ec = new EC('secp256k1');
const { Transaction } = require('./transaction')
const { Block } = require('./block');
class Blockchain {
  constructor(myWalletAddress) {
    this.chain = [this.createGenesisBlock(myWalletAddress)];
    this.difficulty = 4;
    this.pendingTransactions = [];
    this.miningReward = 100;
  }
show_chain_owner()
{
  return this.chain[0].transactions;
}
createGenesisBlock(myWalletAddress) {
  return new Block(Date.parse('01-01-1997'), myWalletAddress, '');
  }
getLatestBlock() {
  return this.chain[this.chain.length - 1];
  }
```

Figure 3.7 blockchain class and genesis block creation

This code snippet shows the blockchain class which is invoked every time a blockchain is made for a patient. The blockchain stores the properties such as the chain of blocks which is the actual chain where records are stored. The difficulty of the blockchain which means number of zeroes that must be appended prior to the hash of a block. The difficulty of the chain determines how much time will be taken to mine a block.

Also, the mining reward which will be awarded to a miner for each successful mining of the block is stored. A list of pending transactions gets stored in the blockchain class. The first block of a blockchain is termed genesis block. The genesis block will store timestamp of creation of the blockchain and also will store the public id of the patient to which the blockchain belongs.

3.3.5 Creation of A Block & Mining It

```
const crypto = require('crypto');
const EC = require('elliptic').ec;
const ec = new EC('secp256k1');
class Block {

    constructor(timestamp, transactions, previousHash = '') {
        this.previousHash = previousHash;
        this.timestamp = timestamp;
        this.transactions = transactions;
        this.nonce = 0;
        this.hash = this.calculateHash();
    }

    calculateHash() {
        return crypto.createHash('sha256').update(this.previousHash + this.timestamp + JSON.stringify(this.transactions) + this.nonce).digest('hex');
    }

    mineBlock(difficulty) {
        while (this.hash.substring(0, difficulty) !== Array(difficulty + 1).join('0')) {
            this.nonce++;
            this.hash = this.calculateHash();
        }


    }

    hasValidTransactions() {
        for (const tx of this.transactions) {
            if (!tx.isValid()) {
                return false;
            }
        }
        return true;
    }
}

module.exports.Block = Block;
```

Figure 3.8 – Class of a block and its functions

We now code a module of a block containing a class and it's properties & functions. It has the previous hash, timestamp, transactions, nonce value and hash of current block. The nonce is initialized to 0 and is used to calculate the proof of work. This nonce is incremented each time so that hash of the block changes with each attempt while calculating the proof of work for a block. While calculating hash of a block we send as arguments all the values

of the blocks and the SHA256 encryption function then returns a 256 bit long hash value in the hexadecimal format. The hasValidTransactions checks whether the transactions stored onto the blockchain are valid or not.

3.3.6 Transaction class

```
const crypto = require('crypto');
const EC = require('elliptic').ec;
const ec = new EC('secp256k1');

class Transaction {

    constructor(fromAddress, toAddress, diagnosis) {


        this.fromAddress = fromAddress;
        this.toAddress = toAddress;
        this.diagnosis = diagnosis;
        this.timestamp = Date.now();
    }

```

Figure 3.9 A transaction class

The diagnosis of a patient is stored in the transaction class. The class has a fromAddress, toAddress, diagnosis and a timestamp. The diagnosis holds the actual diagnosis and prescription given by the doctor. The timestamp holds the time of creation of the transaction. The fromAdress holds the patient's address and the toAddress holds the public key of the doctor doing the diagnosis of the pertaining patient.

3.3.7 Main driver code to create chain and call other functions

```
newChain = new Blockchain(myWalletAddress);

console.log("Owner of the blockchain is" ,newChain.show_chain_owner()+"\n");


invoke_to_create_new_diagnosis(myKey,myWalletAddress,docAdrress,'he has TB', doc_list);
invoke_to_create_new_diagnosis(myKey,myWalletAddress,docAdrress,'he has cancer', doc_list);


console.log();
for (let i of miner_list)
{
    console.log('Balance of miner with id  '+ i+ '  is   '+ newChain.getBalanceOfAddress(i));

}
console.log('\n \n \n');
console.log(`All Diagnosis of patient is:\n ${newChain.getAllDiagnosisOfAddress(myWalletAddress)}`);
//newChain.chain[1].transactions[0].diagnosis = 'he is fine';
console.log('\n \n \n');
console.log('Blockchain valid?', newChain.isChainValid(myWalletAddress) ? 'Yes' : 'No');
```

Figure 3. 10 – Main driver code to create chain and call other functions

For each patient a blockchain is created. The public key of the owner of the blockchain is then printed. Whenever a new diagnosis is done by a doctor he calls invoke_to_create_new_diagnosis() function to add the diagnosis on the blockchain of the patient. The exact working of that function is discussed later in this section. The balance of each miner along with their public keys is then printed. This driver code then shows all the diagnosis of the patient to which the particular blockchain belongs. In the end the code sends a request to verify the integrity of the blockchain and tells whether or not the blockchain is still valid or there is some tampering in the blockchain.

3.3.8 Checking if a doctor is authentic and registered

```
function check_doc(docAdrress,doc_list)
{
    var flag=0;
    for (var i=0;i<doc_list.length;i++)
    {
        if (doc_list[i]==docAdrress)
        {
            flag=1;
        }
    }
    if (flag==0)
    {
        return false;
    }
    else{
        return true;
    }
}
```

Figure 3.11 – Checking if a doctor is authentic and registered

Whenever a doctor sends request to add a diagnosis to the blockchain of a patient this method checks whether the doctor is a registered doctor of the system or not. It throws an error if the doctor is not registered.

3.3.9 Selecting a miner

```
function select_miner()
{
    ind= Math.floor(Math.random()*miner_list.length);
    return miner_list[ind];
}
```

Figure 3.12 - selecting a miner to mine a block

This function selects a random miner from the list of miners to mine a block.

3.3.10 Adding new diagnosis onto the blockchain

```
function invoke_to_create_new_diagnosis(myKey,myWalletAddress, docAdrress, diagnosis,doc_list)
{
    if (!check_doc(docAdrress,doc_list))
    {
        throw new Error("doctor not registered");
        return -1;
    }


    tx1 = new Transaction(myWalletAddress, docAdrress, diagnosis);
    tx1.signTransaction(myKey);
    newChain.addTransaction(tx1);
    selected_miner=select_miner();
    newChain.minePendingTransactions(adminAddress, selected_miner ,miner_list);
}
```

Figure 3.13- adding new diagnosis onto the blockchain

This method is invoked whenever the doctor wants to add a new diagnosis onto the blockchain of the patient. First it checks if doctor is registered or not. Then it creates a new transaction of the diagnosis. Then the transaction is signed by the patient. It will be a digital signature and can be sent in the form of a notification to the mobile device of the patient where he will be prompted if he/she wants to add the block on his blockchain. Once the patient gives the permission and signs the transaction then the addTransaction method is invoked. A miner is then selected by the program using the select_miner function. In the end the pending transaction is mined by the miner. When a block is mined successfully, it gets added onto blockchain of the miner.

3.3.11 Function to add transaction

```
addTransaction(transaction) {
  if (!transaction.fromAddress || !transaction.toAddress) {
    throw new Error('Transaction must include from and to address');
    }
  if (!transaction.isValid()) {
    throw new Error('Cannot add invalid transaction to chain');
    }
  if (transaction.diagnosis ===null) {
    throw new Error('Diagnosis cannot be left empty');
    }
  this.pendingTransactions.push(transaction);

  }
```

Figure 3.14 The add transaction function

This method adds a transaction onto the pending transaction list of the blockchain. It fist checks a few things to ensure the authenticity of the transaction. It first checks if the transaction has a to and from address. It throws an error if it doesn't. It then invokes the isValid function to see whether the transaction is valid. It then checks if the diagnosis is empty or not. This ensures that an empty transaction is not added to the blockchain. After all of the above criteria is met, the transaction is added to the pending transaction list of the blockchain.

3.3.12 Checking if the transaction is valid

```
    isValid() {

        const adminKey = ec.keyFromPrivate('cb3b5da42ade054461a2afbaa8d50a0235b69b48a6c4db91c29c4f8303b8274c');
        const adminAddress = adminKey.getPublic('hex');
        if (this.fromAddress === adminAddress)
        {

            return true;
        }

        if (!this.signature || this.signature.length === 0) {
            console.log('ccvnvbmnvm');
            throw new Error('No signature in this transaction');
        }

        const publicKey = ec.keyFromPublic(this.fromAddress, 'hex');
        return publicKey.verify(this.calculateHash(), this.signature);
    }
}

module.exports.Transaction = Transaction;
```

Figure 3.15 –Checking if the transaction is valid

This method ensures that a transaction is valid prior to it's addition onto the blockchain. For that it has some criteria that a transaction must pass. If a transaction is coming from an administrator the transaction is accepted. If the transaction has no signature then an error is raised. In the end it verifies the transaction using the verify function.

3.3.13 Signing a transaction

```
    calculateHash() {
        return crypto.createHash('sha256').update(this.fromAddress + this.toAddress + this.diagnosis + this.timestamp).digest('hex');
    }


    signTransaction(signingKey) {

        if (signingKey.getPublic('hex') !== this.fromAddress) {
            throw new Error('You cannot sign transaction for other accounts!');
        }



        const hashTx = this.calculateHash();
        const sig = signingKey.sign(hashTx, 'base64');

        this.signature = sig.toDER('hex');
```

Figure 3.16 Signing a transaction

This method signs a transaction. First, it ensures that the correct patient is requesting to sign the transaction. Then it signs the transaction.

3.3.14 Mining the pending transactions and checking if a miner is fake

```
detect_miner(mine_list,mine_tofind)
{

for(var i=0;i<mine_list.length;i++)
{
  if (mine_list[i]==mine_tofind)
  {
    return true;
  }

}
throw new Error('miner is not registered and cannot mine the transaction');
    return false;
}

minePendingTransactions(rewarderAddress, miningRewardAddress, mine_list) {
  const ret_status=this.detect_miner(mine_list,miningRewardAddress);
  if (ret_status==false)
  {

    this.pendingTransactions=[];
    return -1;
  }

 const previousTime=Date.now();
  const rewardTx = new Transaction(rewarderAddress, miningRewardAddress, this.miningReward);
  this.pendingTransactions.push(rewardTx);
  const block = new Block(Date.now(), this.pendingTransactions, this.getLatestBlock().hash);
  block.mineBlock(this.difficulty);
  this.chain.push(block);
  const currentTime=Date.now();
  console.log('block mined successfuly and the miner is '+miningRewardAddress);
  console.log("Time to mine the current block is",(currentTime-previousTime)/1000,"secs with difficulty ",this.difficulty+'\n \n');
  this.pendingTransactions = [];
  }
```

Figure 3.17 mining the pending transaction and checking if the miner is fake

This snippet shows two methods. The detect_miner method checks if the miner who is about to mine the block is a registered miner or not. The minePendingTransactions

method helps to add the transaction onto the blockchain. First a new transaction gets created which stores rewarder's address, the miners address and the mining reward. This transaction is pushed to the pending transactions list. A new block is created with list of pending transactions as the transaction of the block. This block is then mined and pushed onto the blockchain. The function then displays the time taken to mine the block and miner's address. Pending transaction list of the blockchain then gets emptied.

3.3.15 Printing all diagnosis of a patient

```
getAllDiagnosisOfAddress(address) {
  var totalDiagnosis = [];

  for (const block of this.chain)
  {
    for (const trans of block.transactions)
    {
      if (trans.fromAddress === address || trans.toAddress === address)
      {
        totalDiagnosis.push(trans.diagnosis);
        totalDiagnosis.push('\n');
        }
      }
    }
    return totalDiagnosis;
  }
```

Figure 3.18 displaying all diagnosis of a patient

This method return all the diagnosis of a particular patient. For doing this it checks the first transaction in the transaction list of each block and if the address matches then it considers it to be a transaction of the patient.

3.3.16 Getting balance of a miner

```
getBalanceOfAddress(address) {
  let balance = 0;

  for (let block of this.chain) {
      if (block.transactions[1].toAddress === address) {
        balance += this.miningReward;
      }
    }
  return balance;
}
```

Figure 3.19 getting balance of a miner

This function calculates the reward amount of a miner and returns it. For doing it, it checks the second transaction of transaction list of each block of a blockchain.

3.3.17 Checking if the chain is valid

```
  isChainValid(myWalletAddress) {

    const realGenesis = JSON.stringify(this.createGenesisBlock(myWalletAddress));

    if (realGenesis !== JSON.stringify(this.chain[0])) {
      return false;
    }


    for (let i = 1; i < this.chain.length; i++) {
      const currentBlock = this.chain[i];
      const previousBlock = this.chain[i-1];
      if (!currentBlock.hasValidTransactions()) {
        // console.log('dsafads');
        return false;
      }

      if (currentBlock.hash !== currentBlock.calculateHash()) {
        // console.log('fghfgh');
        return false;
      }

      if(previousBlock.hash !== currentBlock.previousHash)
      {
        return false;
      }
    }

    return true;
  }
}
```

Figure 3.20 Checking if the chain is valid

This function tells whether the chain is valid or not i.e. whether chain has been tampered with or not. For this it first checks whether the genesis block has been changed. This can tell whether there is a patient record mismatch. Then it calculates hash of every block of the chain & equates it with stored hash of that block. If both are equal then data of that block has not been tampered with. Then it equates the previous hash of current block with hash of previous block. This tells whether previous block has been tampered with.

CHAPTER – 4

PERFORMANCE ANALYSIS

This chapter puts forward the analysis and statistics observed in the implementation of the proposed prototype of the model.

4.1 TIME TAKEN TO GENERATE PUBLIC-PRIVATE KEY PAIRS

Sr. No.	ENTITY	TIME (in seconds)
1	Administrator	0.364
2	Patient	0.284
3	Doctor	0.254
4	Miner 1	0.274
5	Miner 2	0.316
6	Miner 3	0.256
7	Miner 4	0.281
8	Miner 5	0.278

Table 4.1 – Time taken to generate key pairs for each entity

The above table shows the time taken (in seconds) to generate public-private key pairs for each of the participant. The mean of above 8 outcomes is **0.288375 seconds.**

4.2 TIME TAKEN TO MINE A BLOCK WITH DIFFERENT DIFFICULTIES BY A MINER

Difficulty	Time (in seconds)						Mean (in sec)
	Attempt 1		Attempt 2		Attempt 3		
	Block 1	Block 2	Block 1	Block 2	Block 1	Block 2	
1	0	0.002	0.001	0.001	0.003	0.004	0.00183
2	0.022	0.017	0.003	0.018	0.006	0.007	0.01216
3	0.065	0.159	0.012	0.052	0.067	0.007	0.06033
4	1.524	1.174	0.418	0.783	1.537	2.664	1.35
5	20.261	22.351	90.019	17.376	6.195	4.546	26.79133
6	414.496	504.767	81.835	60.802	86.513	363.054	251.9111

Table 4.2 – Time taken to mine blocks with various difficulty

The above table shows us an exponential trend of how much time our designated computer system takes in order to mine the block with 6 difficulties. The time was calculated by taking 3 attempts for each difficulty level and in each attempt, time was recorded which was taken by the program to mine 2 blocks successfully. The mean of all the observations was calculated, then, to find out the mean time taken by the system to mine block with each difficulty level. These times were then used to plot a graph and show effect of difficulty level on mining time of blocks. The graph is shown in the next page.

4.3 GRAPH OF VARIATION OF TIME TAKEN TO MINE A BLOCK WITH THE DIFFICULTY LEVEL

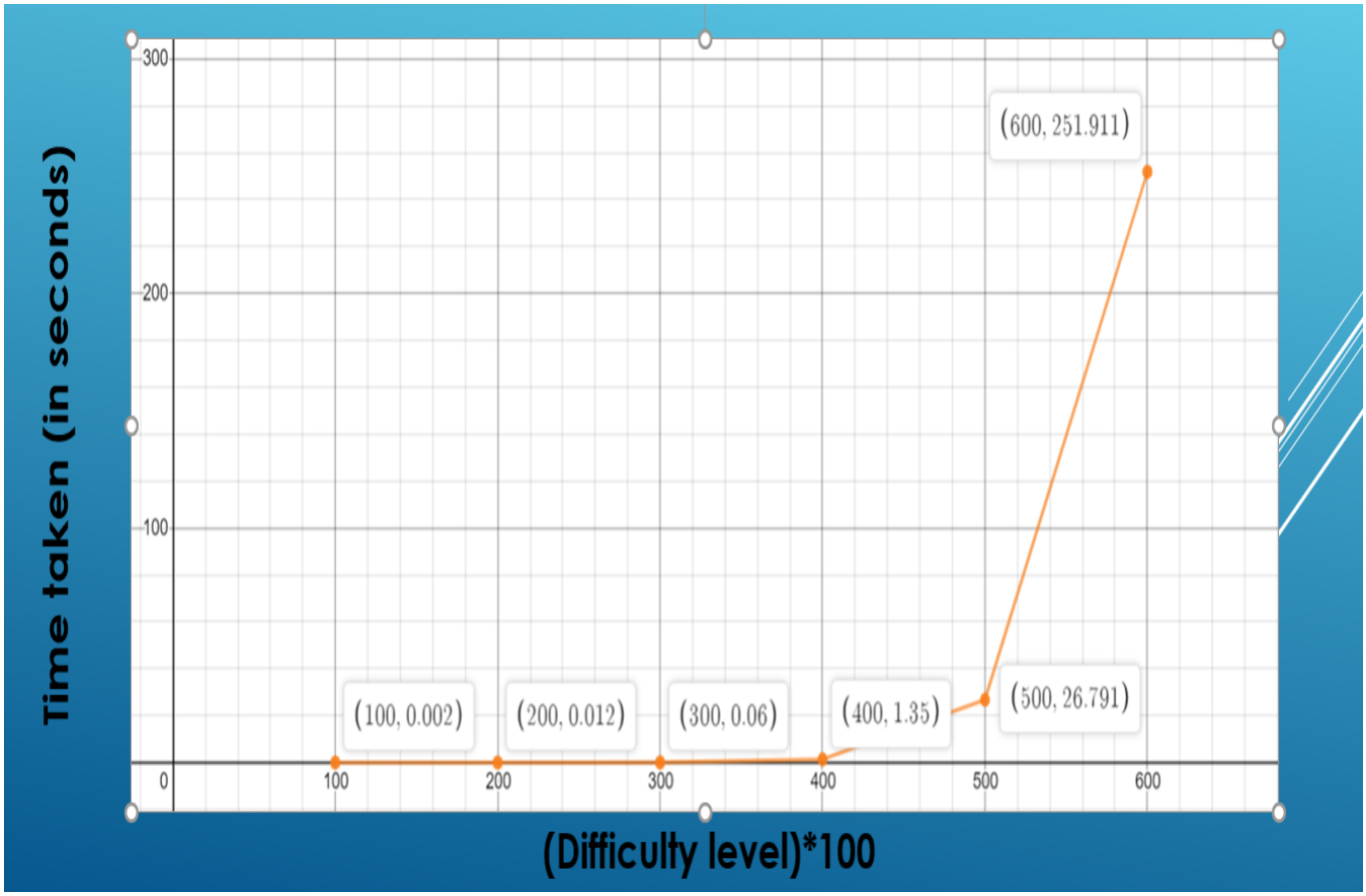


Figure 4.3 – Graph of variation of time taken to mine block with difficulty level

In the above graph we can observe that the time taken to mine the block increases exponentially with increase in difficulty level. Difficulty level on x -axis of the graph is multiplied by 100 for better readability of the graph. This graph can be used to conclude that in order to make our blockchain more secure we can just increase the difficulty of the blockchain. This will make it difficult for the hacker to change the blocks.

4. 4 OUTPUT ANALYSIS

```
[Running] node "c:\Users\acer\Desktop\project screenshotw\final year project\blockchain\blockchaininhealthcare\src\main.js"
Owner of the blockchain is
04ee7b55c9c1cba9923eb7e7a6673155a5ff78e5660e42c32a7fb0deed5db734ea55b78e99df82a07476d5dc1ded1ab46db7c3a1d177430a3f198d5e996d466e99

block mined successfuly and the miner is
04ee07e224fa1671cd87dce803a1a2efaec0f98a16f8521eca44791c9208bbed6526e759e6c23eb04b644b3742026a68eef0b7c0ef4852a307edd79d198f427431
Time to mine the current block is 0.763 secs with difficulty  4


block mined successfuly and the miner is
04c243b1ce589016b8ef337320f5586fc94785c00fc3caad466937fee6dc42552a197b1619fe7db2da2ba17e1e6aacd3e20e76ae922a26ebf8b1c19b31f5f1521d
Time to mine the current block is 0.116 secs with difficulty  4



Balance of miner with id
04c243b1ce589016b8ef337320f5586fc94785c00fc3caad466937fee6dc42552a197b1619fe7db2da2ba17e1e6aacd3e20e76ae922a26ebf8b1c19b31f5f1521d  is   100
Balance of miner with id
04ee07e224fa1671cd87dce803a1a2efaec0f98a16f8521eca44791c9208bbed6526e759e6c23eb04b644b3742026a68eef0b7c0ef4852a307edd79d198f427431  is   100
Balance of miner with id
04a650ef623f67e2064ed87524a891e0ce9a7cf35a3ca0bbd16980ca9b28fa0fbc052e3fc96c57eb052bf51618df86f33b057a5ee2d4d9e643ae37b132d31f2e87  is   0
Balance of miner with id
04a03c48416b2f84a4b1fdb875cb7e8c060874e970927ba0e5a30a5047c6c0dc98e6ac597d1b5e5883cc8e0c515c78347cf67cdb835b231b765e3df629fbd6f34e  is   0
Balance of miner with id
0441886722380fc9100b15a86b5aa66183b3124bc693898457c795ca1d115d370ffbb10400696e2a30d6059e3f0ce12e3bc4873eb291b40a114b15dd15a7af43f4  is   0




All Diagnosis of patient is:
 he has TB,
,he has cancer,

Blockchain valid? Yes

[Done] exited with code=0 in 1.218 seconds
```

Figure 4.4 – Final Output Snippet

The output of our project displays the public key of the owner of the blockchain. It also shows the status of mining of a block (success or failure) along with time taken to mine the block, public key of the miner mining the block and the difficulty level. The balance of each miner is displayed then along with their public keys. Then all diagnosis of the patient is shown. In the end the project tells whether the blockchain is valid or not. The output also throws error whenever there is any threat to the integrity of the blockchain.

CHAPTER – 5

CONCLUSIONS

In this project, we have presented a framework based on blockchain for Electronic Health Records. This project mainly focusses on the backend aspect of the implementation. This has provided us with great outcomes and in depth knowledge on various uses of blockchain in the healthcare sector.

CHAPTER 6

FUTURE WORK

In the future we can deploy the project onto the etherium hyperledger where the blockchain can be distributed to the entities. We can also build smart contracts to incorporate automatic health insurance billing. We can also try and implement the system with proof of stake rather than proof of work and then analyze the difference between the two approaches. We will also incorporate a NoSql database to store our data. A front end for the system will also be developed.

REFERENCES

1. Mettler, M. (2016, September). Blockchain technology in healthcare: The revolution starts here. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1-3, IEEE.
2. Zhang, P., Walker, M. A., White, J., Schmidt, D. C., & Lenz, G. (2017, October). Metrics for assessing blockchain-based healthcare decentralized apps. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1-4. IEEE.
3. Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, *40*(10), 218.
4. Witchey, N. J. (2019). *U.S. Patent No. 10,340,038*. Washington, DC: U.S. Patent and Trademark Office.
5. Rabah, K. (2017). Challenges & opportunities for blockchain powered healthcare systems: A review. *Mara Research Journal of Medicine and Health Sciences*, *1*(1), 45-52.
6. Gordon, W. J., & Catalini, C. (2018). Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. *Computational and structural biotechnology journal*, *16*, 224-230.
7. Ekblaw, A., Azaria, A., Halamka, J. D., & Lippman, A. (2016, August). A Case Study for Blockchain in Healthcare:"MedRec" prototype for electronic health records and medical research data. In *Proceedings of IEEE open & big data conference* , Vol. 13, p. 13.
8. Hölbl, M., Kompara, M., Kamišalić, A., & Nemec Zlatolas, L. (2018). A systematic review of the use of blockchain in healthcare. *Symmetry*, *10*(10), 470.

9. Esmaeilzadeh, P., & Mirzaei, T. (2019). The Potential of Blockchain Technology for Health Information Exchange: Experimental Study From Patients' Perspectives. *Journal of Medical Internet Research*, *21*(6), e14184.
10. Alexaki, S., Alexandris, G., Katos, V., & Petroulakis, E. N. (2018, September). Blockchain-based Electronic Patient Records for Regulated Circular Healthcare Jurisdictions. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1-6. IEEE.
11. Katuwal, G. J., Pandey, S., Hennessey, M., & Lamichhane, B. (2018). Applications of Blockchain in Healthcare: Current Landscape & Challenges. *arXiv preprint arXiv:1812.02776*.
12. github.com/pramodramdas/digital_healthcare

PLAGIARISM CERTIFICATE

AS

ORIGINALITY REPORT

7%	3%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	2%
2	Submitted to University of Warwick Student Paper	<1%
3	Submitted to Victoria University Student Paper	<1%
4	www.jmir.org Internet Source	<1%
5	Submitted to De Montfort University Student Paper	<1%
6	Patrick Ocheja, Brendan Flanagan, Hiroshi Ueda, Hiroaki Ogata. "Managing lifelong learning records through blockchain", Research and Practice in Technology Enhanced Learning, 2019 Publication	<1%
7	ethesis.nitrkl.ac.in Internet Source	<1%