Efficient Contact Tracing for Pandemic Using Blockchain

Author

Nida Bari

MS-17(CSE) 00000204571

Supervisor

Dr. Usman Qamar

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

SEPTEMBER, 2021

Efficient Contact Tracing for Pandemic Using Blockchain

Author

Nida Bari

MS-17(CSE) 00000204571

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Software Engineering

Thesis Supervisor:

Dr. Usman Qamar

Thesis Supervisor's Signature: _____

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

SEPTEMBER, 2021

DECLARATION

I certify that this research work titled *"Efficient Contact Tracing for Pandemic Using Blockchain"* is my own work under the supervision of Dr. Usman Qamar. This work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

_____

Signature of Student

Nida Bari

FALL 2017-MS-17(CSE) 00000204571

LANGUAGE CORRECTNESS CERTIFICATE

This thesis is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university for MS thesis work.

_____

Signature of Student

Nida Bari

FALL 2017-MS-17(CSE) 00000204571

_____

Signature of Supervisor

Dr. Usman Qamar

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express my gratitude to my supervisor **Dr. Usman Qamar** for their constant motivation and help throughout this thesis. It was a great privilege and honor to work and study under his guidance.

I would also like to thank my Guidance Committee Members **Dr. Wasi Haider** and **Mr. Jahan Zeb** for being on my thesis guidance and evaluation committee. I would like to pay special thanks to **Dr. Ayesha Khalid** for her incredible cooperation. Her recommendations are very valued for improvement of the work. I appreciate her guidance throughout the whole thesis.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and brothers whose tremendous support and cooperation led me to this wonderful accomplishment

ABSTRACT

Blockchain technologies have been benefiting many industries by being decentralized, secure and confidential. They offer great potential in pandemic impacted scenarios as well. Contact tracing helps to mitigate the transmission of disease by alerting people who may have exposed so they can act on time to protect themselves. Contact tracing systems face some challenges related to issues of medical privacy, data security and transparency. Multiple research show concern that contacts tracing discourages people to seek medication because of the fear of loss of data, subsequent stigma, discrimination, or abuse. In this paper, we discuss how contact tracing can be improved using blockchain technology and could be able to solve these issues. The aim of our proposed system would be to reduce the impact of pandemic, to implement blockchain for contact tracing and to ensure user privacy and avoid data misuse by incorporating a symmetric key cryptographic mechanism. To tackle the scalability related issues of Blockchain this framework uses IPFS, a distributed file storage system also known as Interplanetary File System. This blockchain based solution system will enhance contact tracing by making it more stable, secure, performant, highly usable and above all effective in the fight against any pandemic.

Keywords: Contact tracing, COVID-19, Interplanetary File System, Blockchain, Distributed system, and Pandemic.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

CHAPTER 1: INTRODUCTION

The Coronavirus 2019 (COVID-19) is rapidly spreading infectious disease caused by novel severe acute respiratory syndrome coronavirus SARS-COV-2 [1]. Throughout the globe, millions of people were sent to quarantine and lockdown to mitigate the wide spread of this disease. WHO reported 27,486,960 confirmed cases and this deadly disease took 894,983 precious lives across worldwide, as of 9th September 2020 [2]. There is currently no vaccine available for COVID-19 but around 169 COVID-19 vaccine candidates are under development [3]. For decreasing the rate of COVID-19 transmission various countries have implemented non-pharmaceutical interventions (NPIs) by reducing the contact rate in general public [4]. Strict measures were adopted by shutting down crowded areas and avoided gathering in large groups such as schools, airlines, workplaces etc. Maintaining social distancing of 6 feet was advised to be strictly followed all around the world.

Strict measures have successfully reduced the number of new cases in cities that implemented NPIs timely. In the absence of vaccine, NPIs have been a great success in the H1NI influenza pandemic (1918-1919) [5], the last pandemic that matched up to COVID-19 scale. Along with the serious threat to human lives, infectious diseases also bring huge economic losses. This pandemic COVID-19 has caused biggest setback to US economy. Statistically, GDP collapsed at 32.9% annualized rate known as the deepest decline since records begin back in 1947 [6]. To cope up with the damage various countries are developing balanced strategies. Contact Tracing had worked significantly to control infectious diseases for decades and shows impeccable capacity in controlling COVID-19. With no vaccine [3], the strategies of easing up lockdown and social distancing restrictions of most countries focused more on tracking approach. This approach will help in bringing normality into the society as well as saving lives and in the long run recusing economy.

Our proposed approach will combine the best of both contact tracing and blockchain technology and overcome any challenges that are faced by existing contact tracing solutions. Details discussion on contact tracing and blockchain technology will be found later in this paper.

This chapter offers a detailed introduction of the research. Section 1.1 discusses the background study; Section 1.1.1 presents the details about contact tracing system, Section 1.1.2 gives an overlook of the history and evolution of contact tracing system. Section 1.1.3 discusses

the gaps of current contact tracing system, research contribution is detailed in Section 1.1.4 and Section 1.1.5 contains thesis organization.

1.1. Background and Motivation

The purpose of providing the background study is to introduce the main concepts used in this research. The concepts involved are: 1) Contact Tracing Systems and 2) Blockchain Technology. The details of the following are given in subsequent sections.

Before the advancement of modern technology, contact tracing system used handwritten paper-based mechanism. They used to store important and critical medical record of patients in the form of paper. This paper-based mechanism was unorganized, not secure, was inefficient and was not tamper proof. Data duplication was another major issue related to paper-based mechanism. Patient had various copies of their own medical record from different institutions, thus creating the problem of data duplication. Patients were never the owner of their own medical data, but the providers were.

1.1.1. Introduction to Contact Tracing

Contact Tracing is a process of identifying, assessing, and managing people who are in close contact with confirmed index case. These contacts are then tracked and informed about their risk to get care and treatment beforehand to prevent further transmission of the virus [7]. Hence this method has successfully reduced case numbers and large-scale outbreak for decades. Before the advent of modern technology, contact tracing relied heavily on paper-based system to store list of potential virus carriers that have been in close contact with confirmed cases over the defined time period [17]. Calls, letters or direct meetings with potential contacts were some methods used to inform who might be contacted. Using these traditional contact tracing approaches had limited the tracing efficiency and delayed the information process. This paper-based tracing system was alterable, incomplete, unorganized, inaccurate and insecure. It was also ineffective due to problem of data redundancy.

Digitalized Contact Tracing was designed to overcome all these above challenges and issues to develop a system that is more reliable in all the above aspects [18]. For all the confirmed cases,

it is of high importance to conduct contact tracing to get all the related information to minimize the outbreak. Figure 1 depicts the series of steps in undertaking contact tracing effectively.

1.1.1.1. Defining Contacts

For A contact is someone who has been in close contact with a COVID-19 case and has following exposures, within the period of 2 days before to 14 days afterward the case develops signs of infection [12].

- Being in a range of 1 metre of a COVID-19 case for more than 15 minutes.
- Physical contact with patient of COVID-19 disease.
- Providing direct care to a confirmed COVID-19 case without using PPE (proper personal protective equipment).

1.1.1.2. Identifying Contacts

Identifying contacts are critical; it requires detailed case information that can be obtained by thorough case investigation and extensive interviews with the confirmed case of COVID-19 patient or their carer. Public health officials need to detect the contacts with respect to the local area context and cultural measures [12].

1.1.1.3. Informing Contacts

The contact tracing workforce should then develop a list of individuals who had been in contact with the patient of COVID-19. Each contact that made it to the list should be informed and thus require monitoring. Context of informing the contact include information about the process of contact tracing, information on quarantine, during self-monitoring what symptoms to look out for, what to do if they become unwell or any specific query a patient might have.

1.1.1.4. Managing and Monitoring Contacts Daily

There are multiple ways to manage and monitor the contacts, i.e., quarantine with the objective of monitoring their symptoms, direct monitoring by the contact tracing team or self-reporting that is determined according to each case specification. A contact with symptoms should

self-isolate [12] and interact with medical provider that is suggested through referral pathway if symptoms become severe or worsen.

1.1.1.5. Data Processes and Analysis

All the data that's been collected throughout the whole process of contact tracing should be saved in somewhere safe, tamper-proof for further descriptive analysis and for compiling performance indicators.

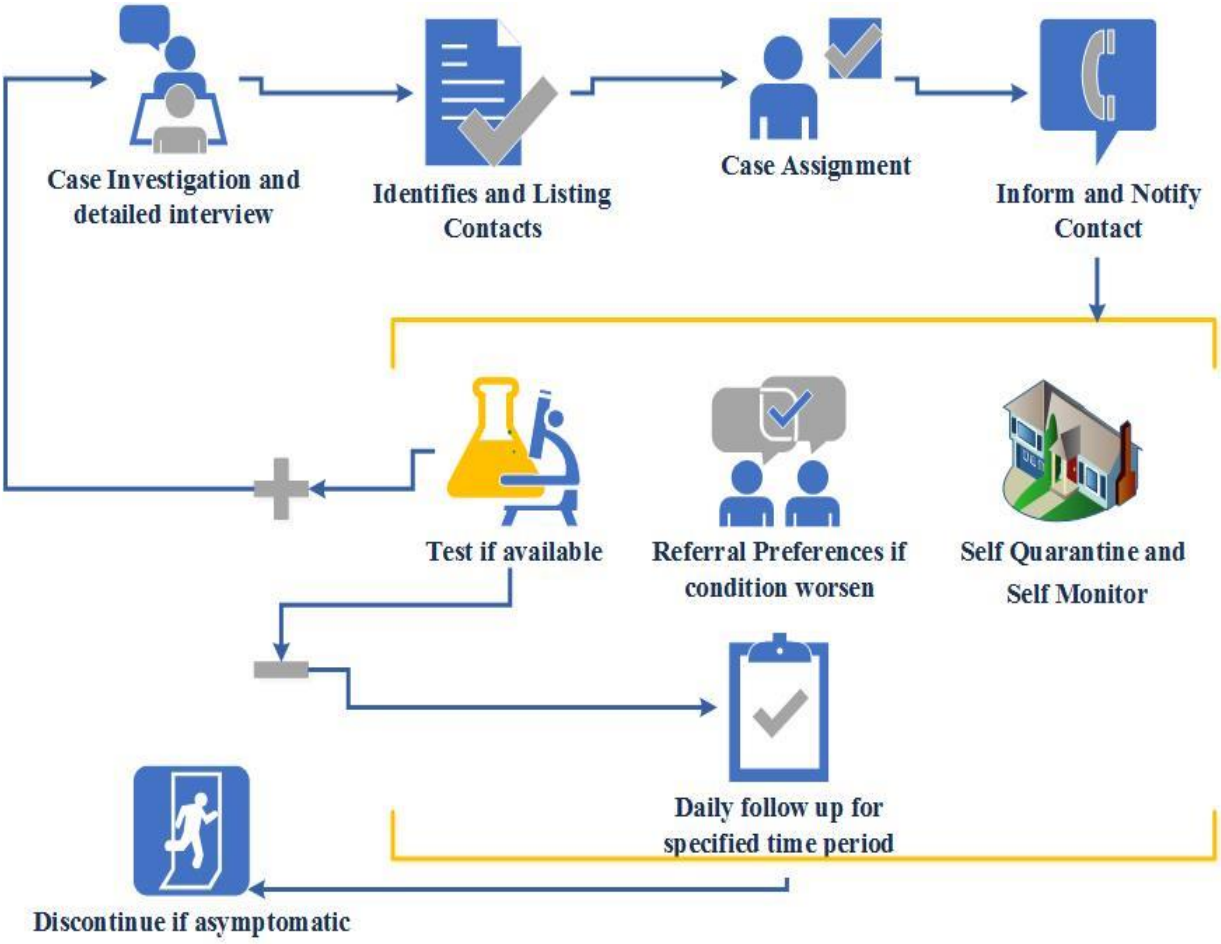

Figure 1: An overview of contact tracing process.

1.1.2. History and Evolution of Contact Tracing System

We evaluate some most recently proposed contact tracing approaches that made significance effort namely, TraceTogether from Singapore [27], NHS COVID-19 App [26], Google/Apple joint contact tracing project [28], and China Health code system [29]. We estimate these

approaches on the basis of level of privacy preservation, security of the technology, power usage, and scalability.

Bluetrace [27] is an open-source application protocol that powers the contact tracing for the TraceTogether app developed by Singapore government digital services. BlueTrace is a privacy-preserving protocol for community-driven contact tracing using Bluetooth devices, that allows for global inter-operability. For making this app work, the user needs to be in the range of active broadcasting stations and keep the device in active state, hence the battery would be drained. As Bluetooth technology has a wireless interface, that leads to have security concerns, some risks including bugging, sniffing, and jamming. All Bluetooth based contact tracing solutions are exposed to these risks.

Replay attacks to the tracing network and user security are the main concern as Bluetooth protocol is prone to the risks because it does not conceal the identification of the hardware used, that will eventually reveal the physical hardware, which may later cause a massive scale of panic to the public. Tracetogether is a centralized service thus user privacy is not known to the third party but the authority. User privacy cannot be guaranteed if the malicious activity is by the central service provider.

Another app named Google Apple contact tracing [28] applies the same approach as Bluetooth LE. This approach is more secure in terms of user privacy preserving because service provider does not get hold of the user's real identity hence it becomes more privacy servings than TraceTogether. However, for further steps of contact tracing such as contact matching and notification require to use the central server that's where user privacy is compromised. Similarly, the NHS COVID-19 App [26] faces these potential risks of user privacy.

An app named health code system [29] is quite different from the upper mentioned apps that does not require a Bluetooth nor proximity detection. This app uses the QR code to be scanned which is associated with the user. This is also a centralized application that does not respect the identity of the user and it is not hidden to the authority as well. However, this app only needs to be scanned while passing through the checkpoints that means there would be less drainage of phone battery and consumption of less data than the other apps. The coverage can be easily extendable because of its highly centralized hierarchy.

Many other applications and approaches are emerging to deal with this pandemic through contact tracing such as Aarogya Setu, COVIDSafe, Decentralized Privacy-Preserving Proximity Tracing, Pan-European Privacy-Preserving Proximity Tracing [38] – [41], etc. All of these approaches are quite similar with some modification in certain features but there remained some challenges that needs to be overcome.

1.1.3. Current Gaps in Contact Tracing System

Many countries have taken measures to implement the Contact Tracing systems in order to bring an end to this pandemic. The basic focus is to provide prompt and quick information to contacts for their safety. So, the relevant record must be kept secure, temper-proof and immutable. But there are also some problems associated with these systems which are being discussed in the following subsections.

User Privacy:

User privacy and security is crucial to contact tracing system, hence these features must be respected in the solution framework design. It would create panic to the general public if not handled properly and it is the most concerning factors in all contact tracing proposals. As we have seen, existing contact tracing systems are centralized which brings the big risk of manipulation and corruption. Meanwhile, privacy should never be compromised while using these systems. Another big challenge is secure data sharing that needs to be resolved. One of the features of the blockchain is that it removes the identity from the beginning, and it offers an ultimate confidence in privacy.

Compromise on Tracing Performance:

Another challenge is to maintain the performance of the tracing network that includes the level of technology and the coverage of network. Existing solutions do not have an impact on wider range of users so to enable it globally we need a lot richer technology than Bluetooth interactions. Our goal is to use blockchain technology by connecting all the users through chain while respecting their privacy.

Misinformation:

Misinformation can cause panic to general public and can be very harmful in prevention of pandemic. Information inaccuracy and information transparency are the main two categories that lead misinformation. The health agencies can involve trusted third parties in the process in order to reduce misinformation. Although, their involvement would compromise the privacy and can be harmful from user privacy perspective. Meanwhile, that is not the case for blockchain technology because of its privacy preserving ability. Thus, this technology would reduce the panic in general public. There is a possibility that the trusted third parties have motivation to hold back some information or provide false statistics. Blockchain technology has a feature of transparency that enables the easy verifiable trusted tracing information by the public.

Security:

Security is also a concerning factor of contact tracing solutions. Privacy should be valued from the start till its end, hence there is a need of system that provide protection throughout its life cycle. Blockchain platform is capable of providing the user confidentiality, privacy with cryptography. Therefore, the privacy of user would be protected throughout the contact tracing system.

Table 1: Current Gaps in Contact Tracing System

Current Gaps in Contact Tracing System	
User Privacy	User privacy would create panic to the general public if not handled properly and it is the most concerning factors in all contact tracing proposals.
Compromise on Tracing Performance	Existing solutions do not have an impact on wider range of users so does not maintain the tracing performance.
Misinformation	Information inaccuracy and information transparency are the main two categories that lead misinformation. Involving the third party would compromise the privacy.
Security	. Privacy should be valued from the start till its end, hence there is a need of system that provide protection throughout its life cycle.

1.1.4. Objective and Contribution

The major contributions of our work are as follows:

- We propose the design of contract tracing application to fight pandemics that is backed up by the blockchain-enabled contact tracing framework. Such an endeavor has not been undertaken till date to the best of our knowledge. This combines the best of two worlds since Blockchain technology's benefits can be a game changer for existing contact tracing schemes. The benefits include Improved Traceability, better Transparency, immutability of records.
- The privacy and the confidentiality of each data record is ensured. Using the Role-based access mechanism, users get a granular access to the system according to their assigned role ensuring that the security of the patient's personal medical data is not compromised, and the access is provided to only the authorized users of the system. For secure data sharing, symmetric key mechanism (encryption/decryption) is used.
- In order to solve the problem of blockchain's scalability, our proposed framework uses off-chain scaling mechanism of Interplanetary File System (IPFS). IPFS is also a favorable choice for storing critical and sensitive data due to its cryptographic capabilities.
- We perform real-case scenario of various users performing different functions on the framework to assess the performance of our framework. We conducted performance evaluation using Apache JMeter version 5.3 and Apache Version 2.0. Apache JMeter is a desktop performance testing tool which is used for analysis and testing of applications.

The entire research is done in a systematic way. Flow of the research is shown in **Figure 2.** First of all, we identify the problem, then we propose a solution to the identified problem. Then, we carry out a comprehensive systematic literature review which becomes the foundation of the proposed solution. Research related to the proposed solution are analyzed and compared.

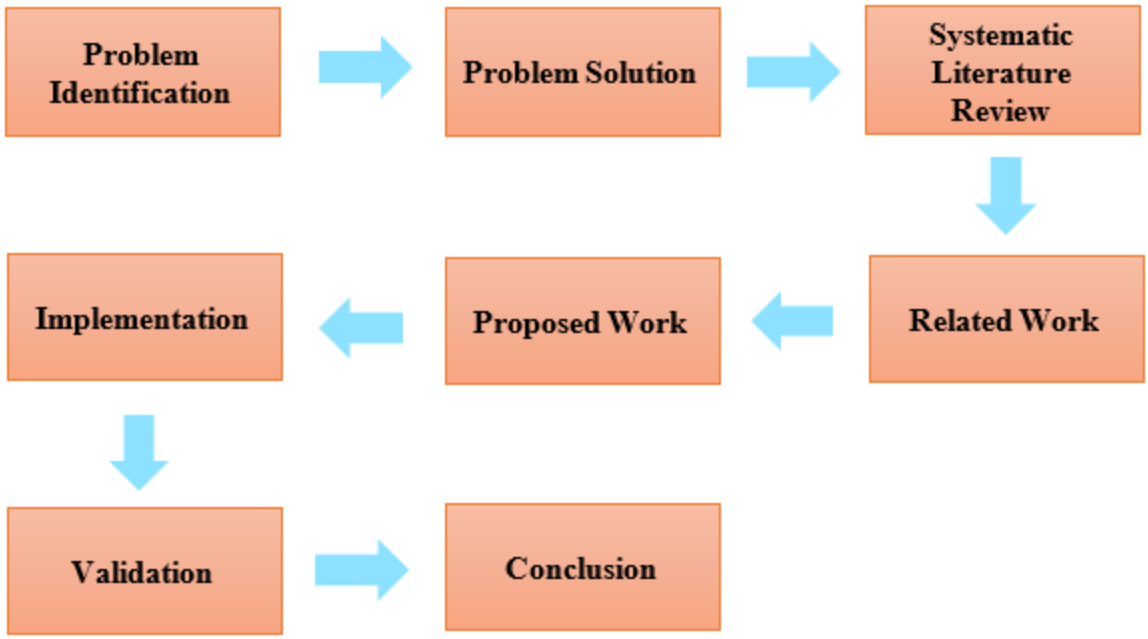

Figure 2: Research Flow

1.1.5. Thesis Organization

Organization of the thesis is represented in Figure 3**. CHAPTER 1**: INTRODUCTION offers a brief introduction containing the background study, current gaps in contact tracing system, research contribution and thesis organization. **CHAPTER 2**: contains the introduction to blockchain technology. **CHAPTER 3**: provides the detailed literature review highlighting the work done in the domain of contact tracing system application and blockchain technology and comparisons to the previous work or studies conducted in this domain. **CHAPTER 4**: presents the detailed implementation regarding the proposed system and its design along-with its architecture.

CHAPTER 5: provides the validation performed for our proposed methodology and its performance in real world scenarios. **CHAPTER 6:** concludes the research and recommends a future work for the research.

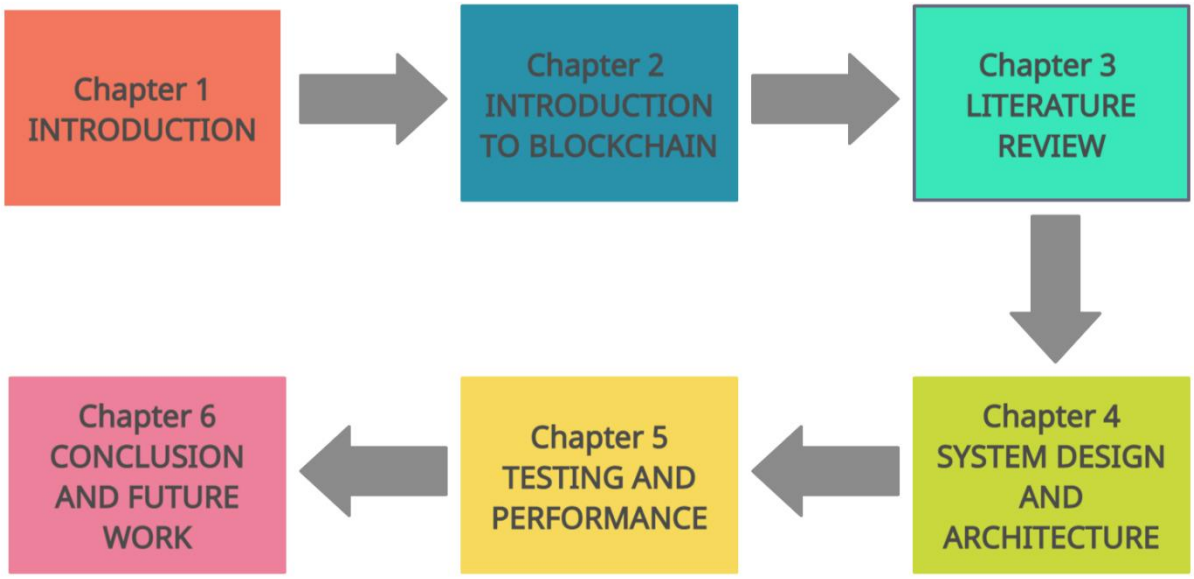

Figure 3: Thesis Outline

Chapter 2

Introduction to Blockchain Technology

CHAPTER 2: INTRODUCTION TO BLOCKCHAIN TECHNOLOGY

2.1. Introduction

Distributed ledger technology called blockchain was invented by Nakamoto in 2008 [8] to solve the double-spending problem of first digital currency, i.e., bitcoin. Blockchain emerged as a digital currency, Bitcoin, it possesses high potential to change IT in the same way as open-source software did a quarter century ago.

Immutability, a unique property of blockchain that ensures no third-party interference to this continuously growing chained data. All the data introduced in blockchain is stored using blocks that are connected and secured using cryptography. Each block contains hash of that block, transaction data, cryptographic hash of previous block and a timestamp. Based on a P2P (peer-to-peer) topology, blockchain is a decentralized ledger that is not owned by any single entity. With no third-party intervention, blockchain is more secure, irreversible, transparent and more performant.

Blockchain technology has exposed great abilities in various fields and can be integrated with contact tracing application to solve problems in existing schemes such as security, trust, transparency, and privacy [19].

When someone joins the network, it gets the full copy of the blockchain for the nodes to verify that everything is in order. The new block is sent to everyone in the network, each node than verify the block to make sure they haven't been tampered with. After successful verification each node adds this block in to their blockchain. All the nodes in this network create consensus. They agreed on which blocks are valid, and which aren't. This validation is performed by the nodes which are connected in the network using specific consensus mechanism to ensure the new member is authentic. This process of validating the transaction is commonly known as mining and miner are the nodes performing this validation. The other nodes will reject such blocks that are tampered with in the network.

Blocks that are tampered with will be rejected by other nodes in the network. So, to successfully tamper with the blockchain one needs to manipulate with all the blocks of the chain, do the proof of work for each block again and take control over 50% of the P2P network. Only then your tamper will be accepted by all others. So, this is almost impossible to do. After validation that block is added to the blockchain and the transaction is completed.

2.1.1. A World without Middleman

The basic benefits of blockchain technology in general are explained in the above discussion but the most important benefit provided by the blockchain technology is the end of having an intermediary between transactions of two parties. This is known as removing the middleman between your agreements and transactions. Blockchain technology provides such a platform that would allow your agreements to not have a middle party controlling all of the transactions, validating these transactions. This middle party is responsible for any sort of services needed and required by its clients.

The process followed by the applications not using the blockchain technology is that of client server where the central party is in this use case is being referred to as the middleman who is controlling this whole process. The middleman has the authority over all the connected nodes in a computing network.

If considered in general terms, let us consider an example where two parties are signing an agreement with the help of a middleman. In this scenario the important task would be the agreement of the parties to the terms and conditions of the agreement. This agreement would be handled by the middleman who would initiate the agreement process, make the two parties understand the terms and conditions of the agreement and validate the agreement. The middleman would also be responsible to get the two parties sign the agreement. This middleman could be an agent, or any other individual responsible for performing this important task. In the case of banking scenario this middleman would be the bank whose resources are being used for transfer of money from one party to another.

This mechanism of using a middleman for such an important task is not supported due to the obvious reasons the authority becomes central and if this middleman is vulnerable to any attack or worse if it is exposed to any attack that would corrupt the whole process of transaction or agreement occurring between parties or entities. Using a process that has no middleman would ensure an effective and efficient environment. As it would help the whole process to be done in a better way with effective costs associated with the whole process.

Blockchain technology uses a peer-to-peer network that does not allow any middleman to act as the authority in the network. There are defined consensus algorithms used in this technology that would help with the authentic way this technology operates the transactions or any other task done on them.

2.1.2. Blockchain Architecture

The basic architecture of blockchain can be understood as that it is a sequence of blocks connected together to form a system that is used to store transactions just like a traditional ledger [48]. Once some data is stored inside the blockchain it becomes difficult to change it as blockchain uses peer-to-peer network, and it consists of many computers that are responsible for the transactions (data) on the blockchain. All of this is managed by a consensus between all the involved parties on the blockchain. There are many consensus algorithms that are used in various kind of blockchain but the most common of all is proof of work (PoW) consensus algorithm.

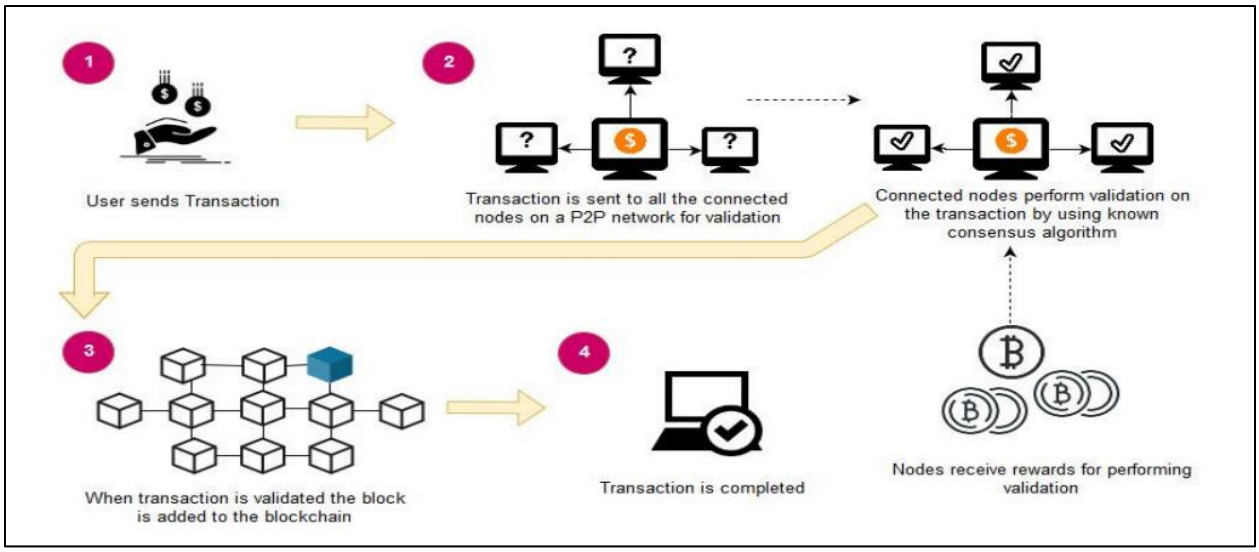

Figure 4: Blockchain Architecture

The blockchain architecture can be more easily understood by a simple scenario where a user on the blockchain network sends a transaction. The transaction of the user is broadcasted to all of the connected nodes in the peer-to-peer network of blockchain. The purpose of this broadcasting of transaction to all the nodes is to validate that transaction. This validation is performed by the connected nodes using some known algorithms to verify the transaction and to ensure that sender is an authenticated part of the network. When a node succeeds in performing the validation that node is rewarded with bitcoin. This process of validating the transaction is known as mining and the node performing this validation is known as miner. This concept would be explained in later sections of this chapter. After, the transaction is validated that block is added

to the blockchain and transaction gets completed. This whole process is defined in **Figure 4** above. Some basic concepts of blockchain technology can be understood in the following descriptions.

2.1.3. Peer to Peer Network

As mentioned in the architecture of blockchain, this technology uses the peer-to-peer (P2P) network of computers for validation and transformation of transactions being sent on it. As the idea was to have a distributed technology that would not be controlled by any central entity due to this the blockchain makes use of P2P network where the connected nodes in a network act as peers to each other. No node connected in this network has the authority to lead or control the whole network but in actual all of them have the same level of control in the network.

These peers connected in a network enjoy the same control or in more specific terms equally privileged [49]. These nodes also share the same resources i.e., these nodes that are acting as peers to each other in a network make a part of their resources available to the other nodes in this network due to which they can have access to them without the need of a central authoritative party [49]. By resources here we refer to processing power and disk storage etc. which is made available for other nodes in this network.

The other model used in contrast of this P2P network model is client-server model where a server is responsible for managing and defining the network rules, whereas the client requests server for using any resource or performing any task. In this network the server has the control over network and fulfills the client's requests. If the server becomes busy or gets crashed the whole network gets affected by this, although there are many methods and mechanisms to control the server from getting damaged but still this networking model could not be used in blockchain technology which is being built on the idea of decentralization and no entity acting as the controller in any scenario. Moreover, in a client-server networking model the resources are divided among the nodes connected in it, which gave the nodes complete control over them and these cannot be shared easily with the other connected nodes.

The following figure 5 represents the way these networks have nodes connected inside them. The P2P network has nodes connected in a manner where every node is connected to each other and there is no central authority controlling the whole network. In contrast to P2P network the client-server network has a server that exists at the center of the network and is controlling all the client nodes connected in the network. The server node has the authority over the whole

network which does not allow decentralization in the network. The client-server network model supports the idea of centralization whereas the P2P network model supports the idea of decentralization.

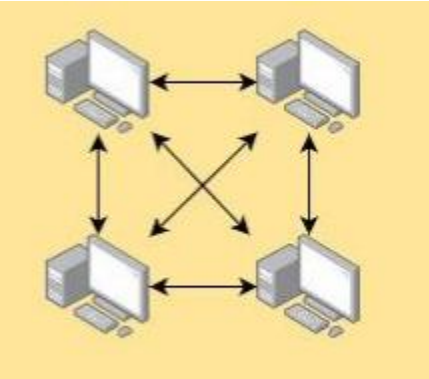

Figure 5: Model of P2P Network

Before moving on to the details of P2P network model let us first understand the centralization and decentralization, their differences and benefits offered by them and also the possible applications of them in various domains.

Centralization:

The concept of centralization could be understood as the scenario where there is a central authority controlling all of the resources and important decisions of that network. In terms of an organization, centralization can be taken as when the higher-level stakeholders have the right of making decisions. In case of computing, centralization occurs when one node has higher level of control then the other nodes connected in the network. This node having the control is known as the central node and it owns all the resources of the computer network. This whole concept is known as centralization. The client-server network uses the centralization concept where a server is responsible for making any control decisions in the networks and the client nodes request for the services and resources from the server node. The problem faced by the centralization is that the other nodes in the network are dependent on the central node. The reliance of the connected nodes on the central node could cause the nodes to lose resources and services in case of central node being non-functional due to any reason. In simple terms, if the central node is attacked by any third party the connected nodes would also become vulnerable to the attack caused by third party.

Decentralization:

With decentralization there is no single node connected in the network that can control the whole network and its resources. In simpler terms there is not point of control which is central in the network or in any other scenario. It is the exact opposite of centralization where a node exists at the center and is responsible for all of client nodes connected in the network. But all of the nodes present have their own roles instead with no central node having the complete authority over the network and its resources. Each node in the network has equal number of resources available to their workstations. The P2P network follows the same model of decentralization where all peers have equally allocated resources to them. There is no authority existing that would control these peers and their services.

2.1.4. Block

As explained earlier blockchain are formed together by a number of blocks connected together in a peer-to-peer network thus making a decentralized application. These blocks contain the hashes in their header of the previous blocks on the chain. These blocks contain three things in them:

- Data
- Hash of current block
- Hash of previous block

The data could be anything as it depends on the type of blockchain. As in case of bitcoin, the data consists of coins that are actually electronic cash [8]. The hash that is stored in these blocks contains a SHA 256 cryptographic algorithm. This hash is used for unique identification of a block on the chain and due to this reason, they are stored on the previous block so they can be connected to each other.

2.1.5. Consensus Algorithm

Each block that is added on the chain would need to follow some consensus rules for it to be added on the blockchain. For this purpose, blockchain technology uses consensus algorithms. The most common consensus algorithm used is Proof of Work (PoW) algorithm and it was used by Nakamoto [8], in bitcoin network. The basic working of this algorithm is that there are number of nodes or participants on a blockchain network so when a transaction is requested to be added

on the network by any participating node it needs to be calculated. This process is called mining and the nodes that are performing these calculations are miners [50]. There are number of other consensus algorithms being used in other platforms for blockchain technology but the aim of these is same that they need to protect the blockchain network from being compromised and keep it transparent and secure.

The process of mining in blockchain is the way this technology ensures a decentralized secure network of blockchain. As mentioned before, mining is the process of validating transactions on the blockchain using consensus algorithm to ensure security of peer-to-peer (P2P) network. This process not only validates the transaction but also prevents a user on blockchain from double spending [51]. The nodes performing this validation are known as miners. Miners perform validation on the transactions and these transactions stored in a block are added on the global blockchain. The process of mining is performed by miners solving difficult mathematical problems. The most commonly used consensus algorithms Proof of Work (PoW) contains the mining process as miners solving a cryptographic algorithm using SHA-256 hash algorithm to find the target hash of the block [52]. When a miner confirms a transaction and adds it to the blockchain the miner receives a reward. This reward could be of two forms i.e., bitcoins or transaction fees for the mining of blocks.

There are various types of consensus algorithms used by vendors of blockchain technology, but we would discuss PoW.

Proof of Work The most common consensus algorithm used by the blockchain technology is Proof of Work (PoW) algorithm and it was first used by Nakamoto [8], in bitcoin network. It is considered as the original algorithm of blockchain technology because it was said to work for blockchain in its true sense.

The basic working of this algorithm is that there are number of nodes or participants on a blockchain network so when a transaction is requested to be added on the network by any participating node it needs to be calculated. This process is called mining and the nodes that are performing these calculations are miners [50]. This could also be understood that this algorithm is used for securing the transactions and blockchain network electronically [46]. In PoW algorithm as mentioned earlier the miners compete with each other to earn the rewards for adding the

transactions on the block. This ensures that miners would not cheat while mining the transactions onto the network because they would not get any reward in case of them cheating.

The reason due to which the miners would not cheat is because they need to solve a complex mathematical calculation for adding the transaction onto the block. To solve this problem, they would also need computational power which explains the fact that this algorithm electronically secures the network. The complex mathematical calculation when solved by the miner would result in the form of hash which is actually the target hash of the block. This hash is an important value that must be found by the miners in order to claim the reward for mining.

Proof of work is such a piece of data which is difficult and is also costly to be solved as it needs a lot of computational power and time to solve it [53]. The main target is for the network participants to accept block in it and that should cover all of the data of the block. To stop from many numbers of blocks to be simultaneously being added on the network the difficulty level should be set. This difficulty level would help in limiting the blocks to be added on the network. Also, this would also ensure that it is not able to predict which miner has added the block [53].

The difficulty level should be set of the kind that it is not too easy to solve the mathematical problem and it should also not be too difficult to solve it. If it is too easy it becomes vulnerable to any attacks on the network [54]. And if takes too much time to solve it i.e., it is too difficult the problem the executions would be stuck [54]. Also, the hash generated by the miners if not similar should be less than it. The blocks in the blockchain as defined before are connected in a chain and each block has the hash of the previous block in its header.

So, if any attacker wants to change any block on the blockchain by tampering with any block it would require a lot of work as the blocks can only be tampered or changed by performing work on them which is essence of PoW algorithm. This mechanism helps the blockchain technology to be protected from any possible tampering and reduces its vulnerabilities.

This consensus algorithm is used by many crypto currencies such as Bitcoin, Ethereum etc. These are popular crypto-currencies and usage of PoW in these depict the importance and reliability of this algorithm. The main advantages of this algorithm are as follows,

- It protects the system from any attacks caused by the third party who is trying to add a block to the chain.

- It helps the network from not being spammed by the attacker as the difficulty level is set adequately.

This algorithm also has some problems or flaws in it which are mentioned as follows,

- The PoW algorithm consumes too much energy which is mainly used for performing extra calculations. This is the problem or flaw of Bitcoin which was criticized the most and is considered to be an issue with PoW algorithm. This problem basically occurred because miners were making use of more and more of electricity power to make their high processing systems to compute the complex mathematical problem and gather more rewards by adding the block to the main chain.
- An important issue with PoW algorithm is that it does not provide security in smaller networks of nodes. Because if there is a huge network with a number of miner nodes it becomes difficult for an attacker to compromise the system and cause its connected nodes to lose their resources. In a smaller network the possibility of getting an attacker to have an entry point in the system is higher thus it is not a feasible choice for applications having smaller network of nodes as it would not provide security.

Various types of consensus algorithms used by vendors of blockchain technology are given in the following table.

Table 2: Consensus Algorithms and platforms

Algorithm	**Applications/Platforms**
Proof of Work	Bitcoin, Ethereum
Proof of Stake	Ethereum
Practical Byzantine Fault Tolerance	Zilliqa, Hyperledger
Federated Byzantine Fault Tolerance	Stellar
Delegated Byzantine Fault Tolerance	NEO

2.1.6. Types of Blockchain

This technology also offers different kinds of blockchains which could be used by the developers for developing the application that could benefit the society. These different types of blockchains have different applications and advantages provided by them. They are explained as follows:

Public Blockchain:

As its name suggests this blockchain is public which means that it could be accessed by anyone existing on the blockchain network. By accessing here, it means that anyone on it can send transactions and receive transactions from this kind of blockchain [47]. Also, the data stored on it would be visible to anyone connected to it from any part of the world. Moreover, the consensus on this kind of blockchain is also using these publicly connected users to have a right of decision making. These blockchains do not require for any permission from the already connected nodes in the network anyone can connect them without any issue and can perform their desired task on them. So, following are the main points of these types of blockchains [46],

- Anyone can connect to this blockchain and send transactions on it and would also expect these transactions to be added on the block (if they are valid).
- Anyone can participate in the consensus while validating transactions and adding them on to the chain.
- The transactions being performed on the blockchain would be visible to everyone without the need of any permission.

Federated Blockchain:

Unlike the public blockchains these operate under an authoritative group of nodes or in simple terms under a federation. This federation would not allow any other entity to join this blockchain. These types of blockchain are also referred to as Consortium Blockchain as they select a number of nodes which are responsible for performing and managing the consensus process being done in the blockchain network [47]. These blockchains could be customized to allow a node to have access to read the transactions being done on the blockchain network or to make these transactions to be public i.e., visible by everyone. One of the greatest advantages of these types of blockchains is providing privacy to the transactions being conducted on it. This was not possible with the public blockchains because they would never be private and were accessible by everyone.

Private Blockchain:

As understandable by its name these blockchains are fully private with no access provided to the any party existing outside of the network. The permissions to write in these types of blockchains is central to one party and the read permission is also assigned to the authorized parties

existing on the blockchain network [46]. Their greatest advantage is considered to be the security provided by the tight permission-based model it uses in its network. The transactions being done are secure and are not visible or accessible to any third party. They are mostly desirable for use in small companies which do not need their data to exist outside their company's network.

2.1.7. Key Features of Blockchain Technology

The key features or the benefits offered by blockchain are discussed in this section. They are as follows:

Decentralization: In a decentralized blockchain system, the information is distributed across the network and not stored by one single entity. Decentralization solves trust issues as users don't have to trust a central authority and it authorize multiple participants to manage a network. It gains user confidence because of its lower risk of systematic failure.

Data transparency and confidentiality: Transparency add much needed, level of accountability that is required by many businesses. All transactions are traceable and searchable in open blockchain. This feature may contradict with blockchain's privacy and security, but they can work together. Blockchain can provide granular access to some authorized users while protecting your data from unauthorized users. The ownership of data is now shared, and this makes it to be transparent and confidentially secure from any third-party intervention.

Security and Privacy: Blockchain technology uses cryptographic functions to provide security to the nodes connected on its network. It uses SHA-256 cryptographic algorithm on the hashes that are stored on the blocks. SHA stands for Secure Hashing Algorithm; these hashes provide security to the blockchain as data integrity is ensured by them. A hash can be generated by any data being stored digitally but data cannot be extracted from that hash [42]. This makes blockchains to be secure. And as the blockchain technology is decentralized and secured by the cryptographic approaches this makes it to be a good option for privacy protection of certain applications.

Immutability: This immutability property of blockchain is game changer for many industries. Once data is entered in the blockchain, it cannot be tampered with. The reason for this blockchain's property is because of cryptographic hash function. Each block has unique identification through cryptographic algorithm.

2.1.8. Challenges Faced by Blockchain Technology

Scalability and storage capacity: The main concerns related to blockchain is lack of scalability and lack of interoperability. Storage capacity of blockchain is quite limited thus introducing voluminous data of contacts would highly affect the efficiency of blockchain [20]. Because blockchain can store data on it but its protocol was not designed for this purpose. The data on the blockchain is visible to everyone that is present on the chain this makes the data vulnerable and is not a desired outcome for a decentralized platform. As, the purpose behind using blockchain was to get data security.

Lack of social skills: Technology has certain advantages and limitations to certain groups of people, mostly elders and minors. That may affect the rolling out of transactions of any system, but there are ways to incorporate them. Many advanced technologies such as wireless IoT and wearable gadgets can enable them for using any system.

Lack of universally defined standards: As this technology is still in the initial phases and is constantly evolving so there is no defined standard for it. Due to this the implementation of this technology in healthcare sector would also take some time and effort. As it would require certified standards from international authorities that overlook the standardization process of any technology [43]. These universal standards would benefit in deciding upon the data size, data format and type of data that could be stored on the blockchain. Also, the adaptation of this technology would become easier due to the defined standards, as they could be easily enforced in the organizations.

Resistance in using the Technology: It's a fact that centralized systems are more efficient in terms of economics than decentralized systems. On the other side, the decentralized blockchain system can preserve privacy a lot better than a centralized system that can effectively diminish the resistance of using the technology because of the fear of human rights violation. This will increase the uptake of our proposed system among the citizens and eventually be of supreme importance in saving billion lives each day.

Table 3: Barriers of Blockchain Technology

Barriers of Blockchain Technology	
Scalability and Storage Capacity	Storage capacity of blockchain is quite limited thus introducing voluminous data of contacts would cause scalability problems.
Lack of Social Skills	With evolving technology, it is quite challenging to shift from old technologies and understanding new techniques.
Lack of universally defined standards	There are no defined standards and principles for blockchain technology that are universally applied which makes it difficult to enforce it throughout a specific domain.
Resistance in using Technology	Users are resistance in using technology because of the fear of human rights violation.

2.1.9. Solutions to the Challenges Faced by the Blockchain Technology

The solution to one of the most commonly faced challenge i.e., scalability and data storage is discussed in this section. As scalability is concerned storage of data on-chain is not a preferred solution as it would not only affect the confidentiality of data but also increases data size on the chain. So, the solution to this problem could be store the data off the blockchain. Using off-chain storage mechanism would effectively minimize the storage burden off from the blockchain. Also, the stored data should be encrypted to ensure security of data. One such solution was provided by Sahoo and Baruah [44], they proposed storing the data on a database system and using it as an off-chain storage system. They proposed to use the scalability provided by the underlying Hadoop database along with the decentralization provided by the blockchain technology. They used the method to store blocks on the Hadoop database, the blockchain on top this framework include all of the needed dependencies of blockchain but the blocks are stored on Hadoop database to improve scalability of the blockchain technology. It was a good solution but the problem with it was again that data was being stored on a centralized storage which kills the purpose of blockchain i.e., decentralization. So, we needed such a platform that would store data in a decentralized manner off-chain. One such platform is InterPlanetary File System (IPFS); it is a protocol that uses peer-to-peer solution of storing data [45].

IPFS protocol uses a cryptographic mechanism to protect the data stored on it from being tempered or altered. The data that is stored on IPFS enjoys following benefits:

- Assigning unique cryptographic hash to the stored files
- File duplication not allowed
- Storage of files by decentralized naming system i.e., IPNS

These benefits make it a feasible solution for data storage and scalability issue faced by the blockchain technology.

Chapter 3

Literature Review

CHAPTER 3: LITERATURE REVIEW

This chapter presents the literature review carried out for the research. Section 3.1 discusses the research based on Blockchain implementation and prototype, Section 3.2 presents the implementation of existing contact tracing system and Section 3.3 highlights the comparison of proposed framework with related work.

3.1. Prototype / Implementation Blockchain-Based Research

Distributed ledger technology called blockchain was invented by Nakamoto in 2008 [8] to solve the double-spending problem of first digital currency, i.e., bitcoin. Blockchain emerged as a digital currency, Bitcoin, it possesses high potential to change IT in the same way as open-source software did a quarter century ago.

Vujičić et al. [9], presented the framework of Ethereum, the mechanism of blockchain technology and bitcoin. IT is evolving to its full swing, many novel technologies like blockchain is providing benefits to the information system. Bitcoin is defined as distributed P2P (peer to peer) network that performs transaction of bitcoin. A detailed concept of mining of blockchain and well-known consensus algorithm PoW was elaborated. A comparison conducted between bitcoin and Ethereum blockchain in this paper and it also described Ethereum's dependencies. There are a number of challenges posed by this novel technology, the authors highlighted its scalability issue and tried to mitigate this by proposing solutions such as Bitcoin Gold and Bitcoin Cash, Lightning and SegWit.

The authors in [10], discussed elaborately about all the aspects of smart contact, i.e., its architecture, the framework it follows, the upcoming challenges related to it and its vast range of applications. Smart Contract is where all the functionality of the blockchain is defined and programmers can easily manipulate them according to their needs. The main reason for using blockchain is because of the programming language being decentralized. Through this paper, the authors aware the readers with basics of smart contract and the system functioning due to combined efforts of various layers of blockchain. Parallel blockchain, might be a future trend, was also discussed by the authors that is aiming to enhance two different but significant components.

Scalability is a major challenge faced by blockchain and to solve it efficiently various techniques are proposed by authors in [11]. This study identified such projects that are used to solve this problem. Blockchain is based on peer-to-peer mechanism that is an intelligent combination of both economical and computational concepts. This paper discussed two ways to store records such as on-chain and off-chain mechanism and which data should be stored on which mechanism. The basic concept and implementation of five different off-chain patterns is presented, along with explanation of which data type that is to be stored using other storage medium. For reducing further inconvenience, any data that requires transactions should be stored on-chain and all the remaining should be stored off-chain.

In [35] the authors proposed a solution for the scalability issue of blockchain using Hadoop database. They combined the decentralized property of blockchain technology with more scalability provided by the Hadoop database. This framework included all the necessary dependencies of blockchain but stored the blocks on the Hadoop database to make it more scalable. This technique mainly focused on solving the scalability problem of the blockchain instead of focusing on implementing it for any particular sector. Apache Hadoop Databases system was used for this approach. Hadoop database systems are known for their big data stores that will improve scalability and can also perform well on distributed system. In this proposed system multiple nodes are connected in a Federation that forms a blockchain. Every node has some accessibility privileges. The Federation nodes are responsible for managing any transaction that occurs on blockchain. For security, a validation system named SHA3-256 is used to validate and encrypt every transaction. Java is the programming language used to develop this approach. This study was beneficial for understanding that other platforms can be combined with blockchain to make it more scalable or solve this scalability issue.

This study [36], proposed a scalable solution for clinical records using blockchain. This approach was designed for health care sector that conforms the requirements of the office of national coordinator of health information technology (ONC). They highlighted multiple barriers of this technology that cause hindrance in effectiveness of this technology. Firstly, it faces concerns related to privacy and security of blockchain, sharing the data related to health care is crucial so it needs a mechanism to protect it from hijacking, storing huge volume of data sets is also a problem and there is no universal standard enforced for the data being exchanged on

blockchain. This approach includes possible solution for all these barriers. They used a DApp application designed according to the ONC requirements.

One of the benefits of this proposed approach is modularization of data being stored on the blockchain. Other benefits are faster and granular access controls to enhance the trust factor among stakeholders, scalable solution with data integrity. For further research this study would be very helpful because it gives in depth understanding of blockchain platform and also discuss about its challenges, limitations and barriers and provide solution to all these problems under conventional healthcare sector.

Kim et al. [37], presented a system for management of medical questionnaires using blockchain technology. Main purpose of this system was to share the medical data through blockchain technology. For storing this type of information, they used EMR systems. EMR systems faces many challenges related to no defined standards for terminologies and security related issues that associated with this technology. So, they used blockchain technology for this proposed framework. Main functionality of this proposed framework was to create, store and share the data of the questionnaires. If a third-party request to access the questionnaires data, patient's permission is required to let the third-party view that data. In previous studies data was not accessible to the patients and they never asked for their consent before using their data. The authors of this study make sure the fact that no data would be used without patient's permission or consent and to take certain security measures for data security.

3.2. Prototype / Implementation of Contact Tracing for COVID-19 Research

A Most recent releases of contact tracing solution are NHS COVID-19 App [26], TraceTogether from Singapore [27], Google/Apple joint contact tracing project [28], and China Health code system [29]. These new launches are premature at the moment and results have also not been published except the German "corona warn app" [76], so it's hard to tell the success rate. Every scheme is facing some challenges that need to be addressed.

Health Code System uses QR code associated with each user for relational crossmatch. This centralized approach does not preserve privacy and user identity. As for power usage, this approach scans only at the time of passing checkpoints that reduces the consumption of battery and data [29]. The network coverage can be easily stretched due to its central hierarchy.

Contact tracing developed by Google Apple protects user identity and thus this approach becomes privacy preserving. But still, due to the use of central server for searching of contacts and notification, privacy can come under attack and intruder can access user credentials from the server [28].

In TraceTogether, BLE (Bluetooth low energy) is used to find and locate potential virus carriers. A drawback of this scheme is that it requires the user device to be on active mode all the time, hence it consume a huge amount of device power. Bluetooth contact tracing solutions are not secure, thus have a risk of data misuse that creates huge panic among general public. This being a centralized service thus inherits all the related drawbacks making privacy a major concern [27].

Similar to all the other technique NHS COVID-19 App is also struggling with user privacy, concern of misuse of user information. This app also uses BLE that eventually leads to the battery drainage of user device. Long term impact of this technology over health care is still questionable. Lacking infrastructure, limited interoperability and underinvestment are some of the underlying challenges of NHS that needs to be addressed [30].

Authors in [34] proposed a blockchain based solution for digital health passports that will eventually provide imunity certificates. They focused on reducing the response time, providing imutable logs, decentralized solution and re-encryption proxies. This approach addresses many challenges and developed a solution that provides accurate and timely response to reduce the impact of this contagious disease COVID-19.

In [12], the authors provide guidance on how to control COVID-19 by developing contact tracing capacity. All the investigation and suggestion conducted in this study related to COVID-19 is carried according to the consideration of WHO. This document provides details of steps in undertaking contact tracing such as identifying contacts, informing contacts, managing contacts and analysis. This study highlighted Go Data software application and other tools for recording the symptoms submitted by contacts, as well as proximity applications that indicate potential clusters of COVID-19.

Authors in [7] gave insight on the requirements for case isolation and contact tracing. They analyzed significant parameters of COVID-19 spread to estimate the involvement of different transmission routes. Further, they described the limitations of using manual contact tracing, that it

is slower and less effective in containing this rapid viral spread. A fast and effective system which covers a large scale is required to control this outbreak. This system also needs to notify contacts of surrounding positive cases immediately and has the capability to control this pandemic if used by enough people.

In [13], the authors focused on evaluating how smartphone contact tracing technology can influence the control and spread of infectious diseases. Astochastic model is introduced that further transformed into deterministic model, while taking in account of quarantine measures and the effects of contact tracing. They evaluated various scenarios while using these models and concluded that quarantining restrictions can be minimized, and new developed cases can be controlled with the usage of highly accurate tracing technologies and can overcome this crisis.

In [14], the authors conducted an investigation against this viral spread. They tried to track down the route of this spread to achieve epidemic control by extensive investigating, classifying, tracking, and managing contacts. Previous activities can suffer drawback due to substitute interviews with the patient. Moreover, they presented multiple methods such as card transaction, medical record facility and more, to reduce the queries regarding patient's claims.

In [15], the authors proposed that for epidemic control, methods such as isolation of cases and contact tracing can perform wonders to mitigate any pandemic. They used a mathematical model to measure whether these methods can be useful for COVID-19 as well.

Privacy and data security being the major concern can affect contact tracing mechanism. Authors in [16], discussed best possible ways to protect privacy and user's data and develop contact tracing technologies accordingly. They conducted risk analysis of various techniques according to users and societies. This study gathered all the advantages of previously developed contact tracing technologies and combined them to form a system for limiting concerned challenges. Precautions, prevention and upcoming strategies are also discussed.

Considering all the challenges in the existing contact tracing solution, we propose a system that combines the contact tracing approach with the Blockchain technology [32]. With this combination all the benefits of blockchain technology leads the contact tracing approach to be more secure, information to be immutable and more efficient.

3.3. Comparison of Proposed Framework with Related Work

The following table 4 compares our proposed framework benefits and features with that of the related work [26] [27] [28] [29]. The above defined features offered by our proposed framework are blockchain-based; privacy preserving, security of technology and scalability are included in this comparison. These features are then compared and observed that whether they exist in the related work under consideration or not.

Most recent releases of contact tracing solution are NHS COVID-19 App [26], TraceTogether from Singapore [27], Google/Apple joint contact tracing project [28], and China Health code system [29]. These new launches are premature at the moment and results have also not been published, so it's hard to tell the success rate. Every scheme is facing some challenges that need to be addressed.

Health Code System uses QR code associated with each user for relational crossmatch. This centralized approach does not preserve privacy and user identity. As for power usage, this approach scans only at the time of passing checkpoints that reduces the consumption of battery and data [29]. The network coverage can be easily stretched due to its central hierarchy.

Contact tracing developed by Google Apple protects user identity and thus this approach becomes privacy preserving. But still, due to the use of central server for searching of contacts and notification, privacy can come under attack and intruder can access user credentials from the server [28].

In TraceTogether, BLE (Bluetooth low energy) is used to find and locate potential virus carriers. A drawback of this scheme is that it requires the user device to be on active mode all the time, hence it consume a huge amount of device power. Bluetooth contact tracing solutions are not secure, thus have a risk of data misuse that creates huge panic among general public. This being a centralized service thus inherits all the related drawbacks making privacy a major concern [27].

Similar to all the other technique NHS COVID-19 App is also struggling with user privacy, concern of misuse of user information. This app also uses BLE that eventually leads to the battery drainage of user device. Long term impact of this technology over health care is still questionable.

Lacking infrastructure, limited interoperability and underinvestment are some of the underlying challenges of NHS that needs to be addressed [30].

Table 4: Comparison with related work

	[26]	**[27]**	**[28]**	**[29]**	**Our Proposed System**
Blockchain Based	N	N	N	N	Y
Privacy Preserving	Yes, Partially	N	Yes, Partially	N	Y
Security of Technology	Low	Low	Low	Medium	High
Scalability	N	N	N	N	Y

Chapter 4

System Design and Architecture

CHAPTER 4: SYSTEM DESIGN AND ARCHITECTURE

In this section, we describe the framework that we propose for implementing contact tracing system using blockchain.

4.1. Preliminaries

In this section, we formally describe the preliminaries used in proposed framework. The software platform used to develop the framework. Also, the advantages of using these and how they benefit our framework.

4.1.1. Ethereum

Just like the popular crypto currency Bitcoin [8], Ethereum is a distributed blockchain network. It was formally introduced in year 2015, by the founders Vitalik Buterin, Gavin Wood and Jeffrey Wilcke who began their work on this groundbreaking technology in year 2014 [55]. The idea behind Ethereum was to create a trustless smart contract platform that would be open-source and would also hold the feature of programmable blockchain. This technology also shares the peer-to-peer networking that makes it distributed. This platform also makes use of its own crypto currency known as Ethers [55] [56]. This crypto currency can be used for sharing it between accounts connected on Ethereum blockchain [57]. Ethereum also provides the programmers a language in which they can customize their own blockchain, this language is known as Solidity. It was developed for smart contracts that are the main feature of Ethereum.

The Ethereum blockchain comprises of following components:

- Accounts
- Transaction
- Gas
- Blocks
- EVM
- Smart Contracts
- Consensus Algorithm
- Merkle Patricia Trie

4.1.2. Accounts

The interaction between various users existing on the Ethereum blockchain is actually between accounts existing on it. Every account on the Ethereum has a public address and private key associated with it. The public address is visible to everyone and is actually used for identification of a user on the blockchain; the public address is 20 bytes address. Ethereum has two types of accounts,

Contract Accounts: These accounts are governed by the code existing inside them. The contract code when gets executed it performs the operations and this execution is triggered by internal transactions and message calls. It contains an ether balance and smart contract code inside it.

Externally Owned Accounts: This account is used for sending transactions from one account to another account. This account also holds an ether balance, but it does not have a contract code residing inside it. The control is of private keys in externally owned accounts (EOA).

The working of Ethereum blockchain by using these accounts can be understood as that this scenario starts when a transaction is fired an EOA. This transaction is received by contract account which contains the code that gets executed by EVM [58]. The main difference between EOA and contract accounts is that EOA can sign a transaction by its private key and can send this transaction to another EOA or contract accounts. Whereas contract accounts are not able to initiate transaction by themselves [59]. The contract account's function when they are triggered by a transaction being sent from an EOA.

These both types of accounts hold an account state that comprises of four components. These states are [59]:

Table 5: Components of an Account State

Four Components of Account State	
nonce	EOA: Number of transactions that are sent from account address Contract Account: Number of contracts created
balance	Wei owned by this account address
storageRoot	Merkle tree's hash of the root node
codeHash	Contract Account: the code inside them is hashed and stored as codeHash EOA: Empty string is hashed and stored here

4.1.3. Transactions

In Ethereum, transaction is the way external entity would interact with Ethereum. It can be used by external user to update the state of the record or information stored on the Ethereum blockchain network. A transaction is a piece of information that contains cryptographic signatures. This transaction is generated by externally owned accounts (EOA) and then they are submitted to the Ethereum blockchain [59]. This signed transaction can be send from one account to another on the blockchain [58]. Transactions primarily have two types:

- Contract Creation: Every transaction contains smart contract that contain the code for performing various functions on the blockchain. This type of transactions is performed when we want to create Ethereum contract.
- Message Calls: Ethereum contracts have the ability to send message calls to another account on the blockchain. These message calls are not from externally owned account (EOA) but from inside the contract.

An Ethereum transaction regardless of types contains following elements [58][59]:

- From – sender of the message using the blockchain to the recipient. Both sender and recipient have a 20-bytes address.
- To – message recipient, also having a 20-bytes address.
- Value - the fund amount (wei) transferred from sender to recipient
- Data (optional) – contains the message that is being sent to the recipient
- Gas – For every transaction on the Ethereum blockchain the sender needs to pay some fees for performing that operation this fee is known as Gas. Every transaction contains the gas limit and gas price in it. This process is secure because the gas not consumed in a transaction is refund to the sender. And sender is charged for the only gas he consumes during transaction [58].
    - Gas Price: that fee the transaction sender is willing to pay for gas
    - Gas Limit: maximum gas that could be paid for this transaction
- v, r, s – used for creation of the signature that is used for identification of transaction sender
- nonce – keeps the count of number of sent transaction
- init – used for contract initialization for the first type of transaction i.e. Contract Creation

4.1.4. Block

As explained earlier blockchain are formed together by a number of blocks connected together in a peer-to-peer network thus making a decentralized application. These blocks contain the hashes in their header of the previous blocks on the chain. These blocks contain three things in them:

- Data
- Hash of current block
- Hash of previous block

The data could be anything as it depends on the type of blockchain. As in case of bitcoin, the data consists of coins that are actually electronic cash [8]. The hash that is stored in these blocks contains a SHA 256 cryptographic algorithm. This hash is used for unique identification of a block on the chain and due to this reason they are stored on the previous block so they can be connected to each other. In Ethereum blockchain the block has following three elements in it:

- Header of block
- Transactions information
- Header of current block‘s ommers

The blocks in Ethereum have a parent block and the blocks that have an ‘uncle’ relationship with the parent block this block is referred as ommer block [59]. Before understanding this let us first know about different forms a block can take in Ethereum blockchain. These are:

- Parent blocks: acts as parent of the immediate next block in the chain.
- Orphan blocks: blocks having no parent blocks attached with them.
- Ommers blocks: these blocks are linked to the chain of blocks but are not included in the final chain of blocks and ate stale blocks.

4.1.5. Smart Contract

Smart contract are known as the piece of code that is used to perform any task on the blockchain. That task could be to exchange money, or any other valuable piece of work on the blockchain [60]. They run on the blockchain directly thus making themselves secure from any kind of tampering and alterations. Smart contract commonly use solidity language and they can be used

to program any kind of operation that a programmer wants to do on the blockchain. After programming the required operations the programmers can compile them by using EVM bytecode that would be explained in next section. And after compiling them it could be executed and deployed on the Ethereum blockchain [61]. The programming language of JavaScript and Python are encapsulated with the Solidity language provided by Ethereum to write code in smart contracts.

The notion of Smart Contracts was given by Nick Szabo, in 1994. He gave the idea that self-executable contracts can be converted into codes; these digital contracts were used from the distributed ledger technology. This idea was used as smart contracts being used in blockchain technology for defining various functionalities that an application deployed on the blockchain should perform.

Smart Contracts was a unique innovation in the sense because it ended the role of third party existing between two entities performing a transaction. Instead of using a third party for performing the function of signing and allowing the transaction to be done. This was replaced by the two entities agreeing upon the smart contract being used as defining the terms and conditions for a function to be done.

The benefits provided by the smart contracts are discussed in detail in the section below,

Secure

For any component of a system to be secure the system should be able to protect itself from any potential threats or attacks from an external entity. The system that holds the property of being secure should be able to protect all of the information stored inside it from being accessed or tempered with by any external entity.

Just like a great feature of blockchain technology was it being secure, smart contracts also enjoy this benefit. As explained earlier, smart contracts contain the code that is executed for performing required functionality of the system in form of transactions. Considering the importance of this contract code it should be secure and temper-proof at all cost. As, blockchain technology is cryptographically encrypted by SHA- 256 algorithm. The smart contracts are also encrypted by this algorithm and this makes the transaction being executed because of the smart contract more efficient. As the danger of a transaction failing to occur would lessen.

Moreover, smart contracts are automated to perform the functionalities written in its code. The user does not need to process the critical steps to perform those functionalities but instead he would only need to send transaction to perform them.

State of Trust

The main aim of smart contract was to act as an automated contract that would need no middle party for helping two entities signs a contract. This middle party was replaced by smart contracts that were automated to perform these operations. The rules and regulations that were at first written legally and were followed by manual mechanism are now written in smart contracts. The blockchain technology i.e., in our case Ethereum blockchain ensures that two parties holding smart contract fulfill their tasks while following the defined rules and regulations in smart contracts.

Also the smart contracts being executed and used for performing transactions while making use of the triggers make it impossible for any third party to intervene and change the conditions of the contract. The Ethereum blockchain under discussion, also treats smart contract like an actual paper based contract, it has certain libraries through which we can define the ownership rights, access rights, roles and responsibilities of an entity using the system. This automation allows for creating a state of trust in the users of the blockchain technology in whole. As they are sure that their work is being hosted or done on a trusted platform backed by trusted smart contracts.

Transparency

In perspective of blockchain technology, achieving data transparency in any technology is to have a trust based relationship between entities. The data or record at stake should be secured and temper proof. Any data being stored on the blockchain is not concentrated at one place and is not controlled by one node but is instead distributed across the network. The ownership of data is now shared and this makes it to be transparent and confidentially secure from any third party intervention.

For smart contracts, transparency is the rules and regulations defined by using the contract code. The entities associated with the contract are all agreed on these rules so they ensure their transparency during the transactions performed. The terms and conditions of a contract are not

compromised during the transaction process it remain same this also ensures the transparency property of smart contracts.

Time-efficient

Another important benefit offered by smart contracts is time-efficiency i.e., the ability to perform the desired task in minimum amount of time. The world has transformed into a global village and time is the most important entity of any process being done in this global village. With the paper based contract system the participating parties needed to wait for the contracts to be processed and this system was not time-efficient.

In order to cope with the changing needs, smart contracts are the best choice for two parties signing a contract. As smart contracts are run and executed on the internet and they are governed by blockchain technology. So, they can be used for eliminating the delay in processing of contract‘s operation caused by the manual mechanism.

Table 6: Benefits of Smart Contract

Benefits of Smart Contract	
Secure	Smart contracts are automated and encrypted piece of code that is executed for performing various functions of a system. These properties of being automated and encrypted make it a reasonable choice for using it for systems having the requirement of security.
State of Trust	Also the smart contracts being executed and used for performing transactions while making use of the triggers make it impossible for any third party to intervene and change the conditions of the contract this ensure the state of trust among various entities associated with them.
Transparency	For smart contracts, transparency is the rules and regulations defined by using the contract code. The entities associated with the contract are all agreed on these rules so they ensure their transparency during the transactions performed.
Time-efficient	Another important benefit offered by smart contracts is time-efficiency i.e. the ability to perform the desired task in minimum amount of time. As smart contracts are run and executed on the internet and they are governed by blockchain technology. So, they can be used for eliminating the delay in processing of contract‘s operation caused by the manual mechanism.

4.1.6. EVM Architecture

The Ethereum Virtual Machine (EVM) follows the stack-based architecture i.e., it stores the data from top-down and pushes down the older stored data when new data is added on the stack. The EVM architecture has three states in it [62]:

- Immutable: An immutable object holds the property of being unchangeable i.e. once created its state cannot be modified after that [63] [64]. Immutable data can help in development of simple applications and also in detecting the change that occurs using the simple logic [64]. In EVM architecture, this state holds the EVM Code which is a vital part of the EVM.
    - EVM Code: This is the smart contract code that could be executed by Ethereum virtual machine. The human-readable code written in Solidity programming language needs to be translated into machine-readable code. This is machine-readable code is the EVM code.
- Volatile: The machine-readable code i.e., EVM code is used by this state. The term "volatile" refers that when a machine's power is switched off it would lose all the contents that it has stored in its memory. This state of EVM holds following important components in it:
    - Program Counter: Commonly referred as PC, this component is used for storage of address of the current program that is to be executed by the compiler.
    - Gas: For every transaction on the Ethereum blockchain the sender needs to pay some fees for performing that operation this fee is known as Gas. Here in EVM it stores the amount of gas available in this state.
    - Stack: The memory of stack is 256bits and it can store 1024 elements inside it. It uses the mechanism of stack data to store elements inside it i.e. LIFO (Last In First Out).
    - Memory: The EVM memory is linear and this stores the items inside it. Memory component exists inside the volatile state, thus it loses its contents if the machine is powered off.

- Persistent: Persistency is the state in which the data is not changed on every alteration but instead it yields the data i.e., non-volatile. This state contains the storage component inside it.
    - Storage: The EVM storage is a store that contains key-value pairs. The mapping is 256-bit words to 256-bit words.

The code that is written in Solidity programming language that code is actually a high level language code which needs to be converted to EVM bytecode.

4.1.7. IPFS

IPFS is a protocol that uses peer-to-peer network for data storage. It provides secure data storage as data stored on IPFS is protected from any alteration. It uses a cryptographic identifier that protects the data from alteration as any attempt to make change on the data stored on IPFS could only be done by changing the identifier. All the data files stored on IPFS contain a hash value that is generated cryptographically. It is unique and is used for identification of stored data file on the IPFS [65].

This secure storage strategy of IPFS protocol makes it a favorable choice for storing critical and sensitive data. The cryptographic hash that is generated could be stored on the decentralized application to reduce the exhaustive computational operations over the blockchain. IPFS protocol works using a peer-to-peer (P2P) network, this network contains a data structure known as IPFS object that contains data and link in it. Data is unstructured binary data and link consists of an array. The IPFS protocol works in the following way [45]:

- Files stored on IPFS are assigned a unique cryptographic hash.
- Duplicate files are not allowed to exist on the IPFS network.
- A node on the network stores content and index information of the node.

IPFS follows a P2P network which is distributed due to which it has no single failure and there is also trustless state between connected nodes [66]. By design, it has following sub-protocols that help in functioning of the IPFS protocol. These sub-protocols are defined briefly as follows [66]:

- Identity/ies – Each node on the network is identified by NodeID, public key and cryptographic hash. These elements are used for identity management and verification of connected nodes.
- Network – A number of nodes are connected on the IPFS network they all enjoy some benefits such as reliability, integrity and authenticity etc.
- Routing – This is used for finding the connected peers network addresses and the peers who can serve as particular objects.
- Block Exchange – IPFS uses BitSwap protocol for distribution of data by exchange of blocks with peers. It functions like BitTorrent and have a new set of blocks ready for exchange with a list of blocks. This exchange is used for block distribution.
- Objects – IPFS builds a directed acyclic graph DAG or precisely Merkle DAG that has links contains hashes of the targets in the source. It is used for addressing of contents, resistance for content temper and to avoid duplication of objects on the IPFS.
- File – On top of Merkle DAG, IPFS defines object for modeling a file system.
- Naming – A naming system for IPFS file objects, this naming would be self-certified and human friendly.

4.2. Truffle

Truffle can be defined as a development environment and testing framework that is used along with EVM of the Ethereum blockchain for developing DApps [67]. Truffle provides a number of features that help the developers while creating their DApps. They are [67],

- Compile, link, deploy and manage the built-in smart contract
- Test the smart contracts
- Framework for deployment and migrations
- Network management
- Package management
- Contract communication (through console)

Truffle environment has a unique way of providing various frameworks for helping developer to develop their DApps i.e. it contains various functionalities in forms of boxes. These boxes are unboxed for using those libraries in their projects by developers. These boxes contain the

boilerplates that could be used by developers to customize them for their unique decentralized applications [68]. These boxes are pre-loaded with the smart contracts, migration files, front-end libraries and other dependencies along with the testing code. Some of the famous boxes of truffle framework are Drizzle, React, Metacoin and Webpack etc. For our proposed system we are using Truffle React Box.

4.2.1. Truffle React Box

React is a JavaScript library which is used to build user interface of applications. The benefits that this library offers are that it‘s declarative, component-based and encapsulation. It makes use of components that are encapsulated and that manage their own states throughout the life of the components. The data inside the components is of two types: props and state. Props can be understood as the function parameters, they are passed to the components instead of function in react. State contains the input data that would change as the data changes on an input form. It can be understood as an object that is stored inside a component class as its property. The building blocks of react are considered to be elements and components that combine together to keep the react applications to be functioning. Their brief definition is given below:

- Elements are the smallest building blocks of react application and it holds the property of immutability [69].
- Components are used for splitting the UI into separate and independent pieces and these pieces also hold the property of reusability [69].
- An event in simple computing terms is an action that results by a user triggering any activity that activates the event.
- React uses the controlled components mechanism for handling the form elements and the input entered in the form elements by the user. These form elements maintain their own state which is updated when an event is triggered and updated by using JavaScript functions. This whole process is known as controlled components.

4.2.2. Truffle Configuration

This is a configuration file written using JavaScript language and is located in your root folder of project. The file is named as truffle.config and is used for mainly configuration purposes of the project.

```
const path = require ("path");

module.exports = {
  // See <http://truffleframework.com/docs/advanced/configuration>
  // to customize your Truffle configuration!
  contracts_build_directory: path.join (__dirname,
"client/src/contracts"),
  networks:{
      development: {
            host: "127.0.0.1",
            port: 7545,
            network_id: "*"    //Match any network id
      }
  }
};
```

The code snippet shown above contains the default setting for configuration of the truffle project. The commands for running the project in test environment are used alongside with this configuration file. This command is:

```
truffle migrate
```

The network configuration explained in the above code is used for deployment of contracts on the MetaMask. If this configuration of network is not defined the contract would not be deployed by truffle. While the migration process is initiated the specified network is used for deployment of the contract. The network has sub requirements which are development and live. The development has host, port and network id. The host would be 127.0.0.1 which is the localhost of the computer and the port is 7545 or 8545 which is the default port used by MetaMask. The live option has some configuration values which include gas, gas price, from, provider etc. It also includes the other configuration values of host, port and network id.

- Gas: For every transaction on the Ethereum blockchain the sender needs to pay some fees for performing that operation this fee is known as Gas. This includes the gas limit for deployment of the contract.
- Gas Price: The actual price for deployment is included in this portion.
- From: This is the address of the account which is used for migration and deployment. It is the first account which is used by the client for deployment.

4.3. Ganache

For developing decentralized applications on Ethereum, Ganache is used as a personal blockchain. It can be used for development of DApps, deployment of smart contracts and running tests [70]. It is also known as a virtual blockchain that can be used by developers as it provides 10 Ethereum addresses for development purposes. These addresses contain 100 ethers pre-loaded in them. These addresses can be used for development of applications and running tests on the smart contracts or overall application's functionality. As specified in the official documentation Ganache offers two flavors i.e., Ganache CLI (Command Line Interface) and Ganache UI (User Interface). It runs on the localhost of the developer's system i.e., at 127.0.0.1 and port 7545 for UI and port 8545 for CLI.

The addresses on the Ganache can be considered as nodes running on Ethereum blockchain and the decentralized applications can be run on them. It looks like a virtual network of nodes but acts like a real-world application of Ethereum having nodes connected on its network. The following steps define how a user can start using the Ganache blockchain for development and testing purposes on their systems.

1. **Installation:** The user can download the version of Ganache according to his operating system. After download ends, user can initiate the process of installation by clicking on the Ganache setup downloaded. The download wizard would guide user to specify various requirements. As installation process ends, user is provided with a mnemonic seed that must be kept secure for using it in case of account being lost.
2. **Creating Workspace:** As installation process is completed, user can run the Ganache blockchain. Depending on the type installed by the user i.e., Ganache CLI or Ganache GUI,

both of these are used blockchain development. On first setup, user is prompted to select to create a workspace for using Ganache. It provides two options i.e.,

- **Quickstart workspace** – Selecting this option would provide the user with a new blockchain having the genesis block every time you open the Ganache on your system. This workspace would provide four panels:
    i. Account: The 10 accounts provided by default for development purposes. The user can view account address, balance, Tx count and private key in the account panel.
    ii. Blocks: The blocks mined are visible on this panel. It also includes the block number, data & time when block was mined, gas used, and transaction done on the account.
    iii. Transactions: All of the transactions that are running on the transactions are presented in the form of list.
    iv. Logs: Contains the log of the server.
- **New workspace** – This option can be selected for the scenario where a user wants to customize the workspace according to his own project requirements. The user can save the Quickstart workspace and create a new workspace as well [71].

4.4. MetaMask

MetaMask is another tool that helps in the development and testing of Ethereum blockchain development. The Ethereum addresses provided by Ganache can be used to deploy and run the decentralized application on the MetaMask. It acts as a bridge that allows the developers to view the working of the decentralized applications on the web browser. It also benefits for allowing the developer's browser to act as an Ethereum blockchain without running complete node of the blockchain [72]. The workflow of MetaMask is explained in the following figure 6,

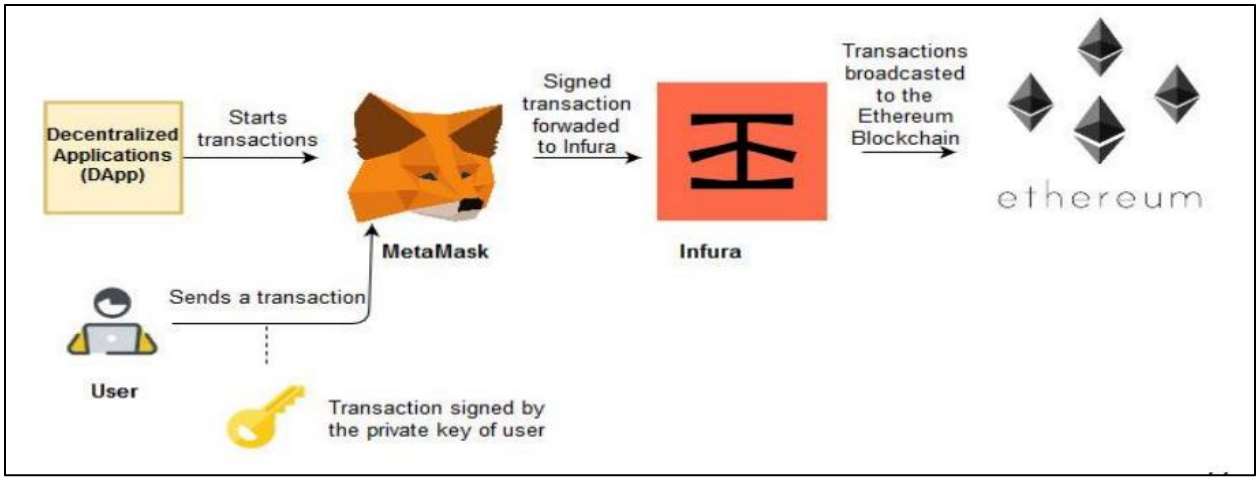

Figure 6: Workflow of MetaMask

4.4.1. MetaMask Connection:

The following code could be used for connecting the MetaMask and Ganache, the two tools that we have explained above.

```
if (typeof web3! == 'undefined') {
  App.web3Provider = web3.currentProvider;
  web3 = new Web3 (web3.currentProvider);
} else {
  // If no injected web3 instance is detected, fallback to Ganache.
  App.web3Provider = new web3.providers.HttpProvider('http://127.0.0.1:7545');
        web3 = new Web3 (App.web3Provider);
      }
```

The above code is in JavaScript language the “App” is the class name of the JavaScript class. This component of class is used for injecting web3 for connecting MetaMask and Ganache. The Http Provider differs for Ganache UI and Ganache CLI. As shown in above code snippet for UI it is http://127.0.0.1:7545 and for CLI the http provider is http://127.0.0.1:8545.

The following code could be used for connecting the MetaMask and Truffle, the two tools that we have explained above.

```
if (typeof web3! == 'undefined') {
  App.web3Provider = web3.currentProvider;
  web3 = new Web3 (web3.currentProvider);
} else {
  // If no injected web3 instance is detected, fallback to Truffle Develop.
  App.web3Provider = new web3.providers.HttpProvider('http://127.0.0.1:9545');
        web3 = new Web3 (App.web3Provider);
      }
```

The above code is in JavaScript language the "App" is the class name of the JavaScript class. This component of class is used for injecting web3 for connecting MetaMask and Truffle.

4.5. System Design

System design is the most important and vital part of any framework as it is used for the development of the system from its theory. This section includes the modules, architecture and various elements that are combined together to form the whole system's framework. As defined earlier the purpose behind this proposed framework is to create such a decentralized system that is temper-proof, secure and confidential blockchain-based system for contact tracing which provide quit responses to the potential contacts.

Every system has some entities that are functioning together to keep the system functioning. The entities of this proposed framework along with the system design are explained in next section. As visible in below figure 7, the proposed framework or system has three entities or modules. These modules when combined together would keep our system working. These entities or modules have further concepts that need to be understood they are explained as follows.

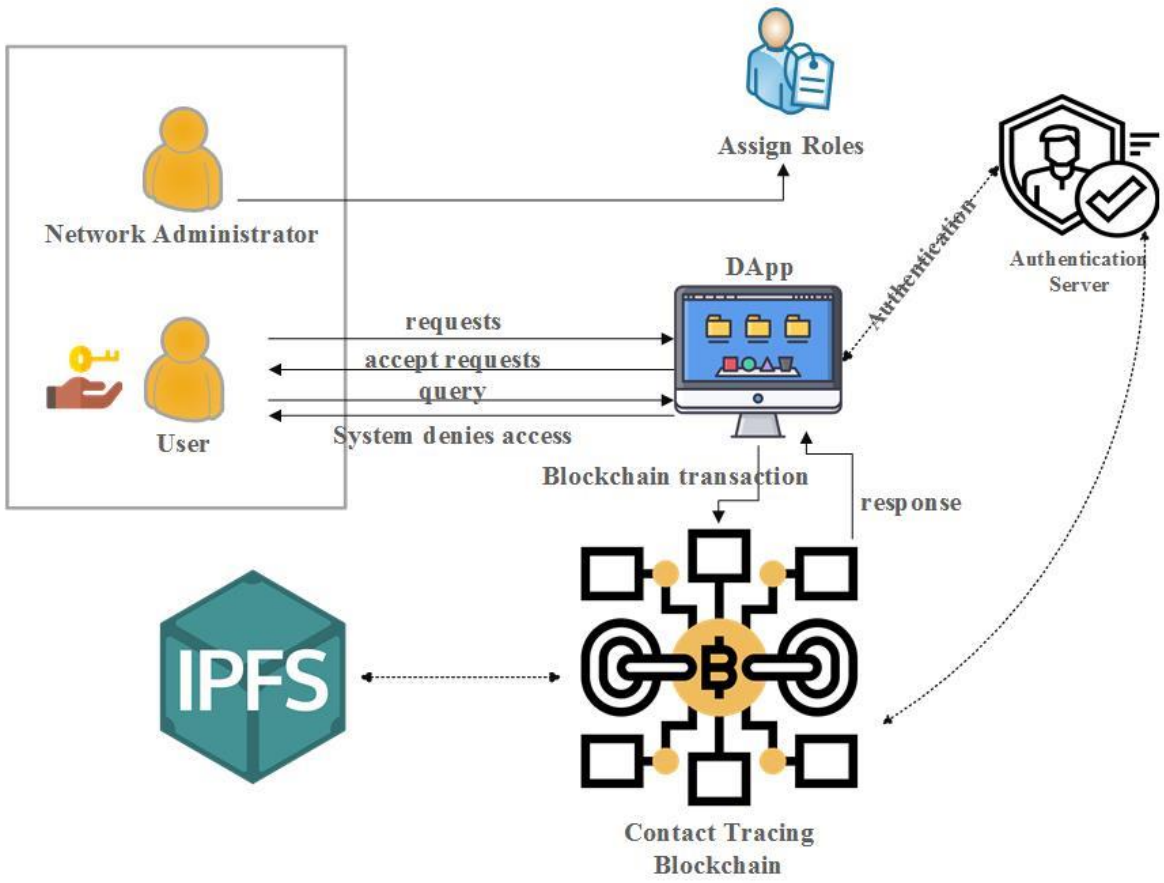

Figure 7: System Design of Proposed framework

4.5.1. User Layer

A user of a system is defined as an individual who makes effective use of the system and its resources. A user can have various features these can be a role name, an account on the system and some identification, authentication system [73]. These features are used to let a user be identifiable on the system.

Just like any user attached with a system, blockchain-based decentralized system also has users associated with it. Their basic task is to be able to use the proposed system and its resources. The users could belong to hospitals including patients, contact tracers, and administrative staff etc. The main task of these users could be to interact with the system and perform tasks related to tracing the potential contacts and providing them with preventive instructions according to their medical records.

The users using this system would be accessing the system's functionality by a browser which in technical terms we refer as DApp browser, as it is containing the GUI (Graphical User Interface) of the DApp i.e., our proposed system framework. The GUI contains all the functions that could be accessed by a particular user. The user according to the assigned role could use this GUI for interacting with the other layer of the system i.e., blockchain layer.

4.5.2. Blockchain Layer

The next layer on the system is the blockchain layer; this layer contains the code or mechanism for interaction of user with the DApp which is functioning on the blockchain. This layer contains three elements inside it. They are:

Blockchain Assets: In Ethereum blockchain, transaction is the way external entity would interact with Ethereum. It can be used by external user to update the state of the record or information stored on the Ethereum blockchain network. This layer stores the record of transactions that are done on it. These transactions are treated as assets by the Ethereum blockchain as they are piece of information or data that user intends to send to another user on the chain or to simply store it for using it later.

Governance Rules: As explained in previous sections, blockchain technology in general follows consensus rules for transactions to be done and computed. For this they need some consensus algorithms to keep the blockchain temper-proof and secure. Also, usage of these consensus algorithms ensures that blockchain technology remains decentralized, distributed, transparent and confidential. Ethereum blockchain uses Proof of Work (PoW) consensus algorithm that is explained in previous sections in detail. The reason behind using a consensus algorithm is also for ensuring that governance of blockchain is maintained in a trusted manner which is through consent from all the trusted nodes attached to the blockchain network.

Network: Ethereum blockchain uses the peer-to-peer network. In this network all the nodes are connected as peers. With no node acting as the central node controlling all the functions of the network. The reason behind using this network was because the idea was to create a distributed platform not a centralized. So, using a network where all the connected nodes have equal status and right was the best choice this technology could have done.

The above-mentioned components were explained in an abstract for getting an insight as to why these elements are important to the DApp. As mentioned before, our proposed framework consists of users that could be patients, contact tracers, nursing staff, and administrative staff. They are given granular access as they should have varying level of authority on the system. The following section explains how these access rights and basic functions are being performed in the system.

4.5.3. Transactions

The system includes following transactions:

- Add records would create patient's medical symptoms records in the DApp. It contains the fields of ID, name, symptoms, blood group, and IPFS hash. The patient's basic symptom records are stored along with the IPFS hash that contains the file uploaded containing the lab results or other medical records of patient.
- User login would allow only the authentic users to access the system while protecting it from unauthorized users.
- View records would let the user view the medical records of a patient stored in DApp.
- Send message would transmit quick response to any patient.
- Get message would deliver the message stored in DApp to particular patient.
- Grant access for each of the above-mentioned transactions, user would need certain role to have access to them i.e., contact tracer would monitor and observe the condition of the patient and provide quick response to the patient. Contact tracer would also track down potential contacts and would send them quick instruction related to preventing the spread of the disease. A patient can add his symptoms on daily basis to get observed by the contact tracer and receive instruction on what to do when condition worsen.

4.5.4. System Implementation

As already explained in the previous sections, the system would be implemented by using the Ethereum and its dependencies. The users need to have a wallet or a personal account address on the blockchain for the system to be fully functional. The system implementation is explained in the following section.

4.5.4.1. Smart Contract

As explained earlier, smart contracts are an important part of DApps as they are used for performing basic operations. Patient Records is the contract that is included in this framework. This contract is used for giving access to the users on the DApp and performing Contact Tracing operations on the records of patient. The Patient Records smart contract is made purely for implementing the functionality of the proposed framework. It performs the multiple operations along with the defining roles for access of these functions.

The algorithm for defining the Patient Records smart contract is given below. It defines all the operations that are being performed in it and various conditions that are associated with them. It also explains how the roles are being maintained for granting access to a particular functionality.

Algorithm 1 Smart Contract for Contact Tracing

```
Assign Roles:
 function Assign Roles (New Role, New Account)
        add new role and account in roles mapping
end function
Add Data:
function Add Patient Record (contains variables to add data)
        if (msg.sender = = patient ) then
            add data to particular patient's record
        else Abort session
        end if
end function
Retrieve Data:
function View Patient Record (patient id)
        if (msg.sender == contact tracer || patient) then
            if (patient id) == true then
                retrieve data from specified patient ( id )
                return (patient record) to the account requested the retrieve operation
            else Abort session
            end if
        end if
end function
Add Test Records:
function Add Patient Test Record (contains variables to add data)
        if (msg.sender == patient ) then
                Sends the information to IPFS
                IPFS stores the information in distributed hash table
                IPFS assigns hash to the information file stored
                Sends the hash information to blockchain
        else Abort session
        end if
```

```
end function
Send Message:
function Send Message (patient id)
        if (msg.sender == contact tracer) then
                send message to particular patient
        else Abort session
        end if
end function
Get Message:
function Get Message (patient id)
        if (msg.sender == patient) then
            if (patient id) == true then
                return (message) to the account that requested the get message operation
            else Abort session
            end if
        end if
end function
```

4.5.5. Illustrative Use case Scenarios

The basic working of this decentralized application (DApp) could be understood by following through the process by which user would interacts with it. As depicted in figure 8 below, a user interacts with the User Interface (UI) of the DApp. The UI is used for interaction with the DApp deployed on the Ethereum blockchain. The user is only shown the front end of the application with no source code or working of the Ethereum blockchain visible in the browser.

At the back end, the Ethereum blockchain has some components or packages that combine together to keep it functioning. The DApp contains three main packages that are used for running and deploying the application on the Ethereum client. These packages or libraries are ReactJS, Web3JS, and Smart contracts; they are explained in below,

ReactJS:

React is a JavaScript library which is used to build user interface of applications. The benefits that this library offers are that it's declarative, component-based and encapsulation. It makes use of components that are encapsulated and that manage their own states throughout the life of the components. The data inside the components is of two types: props and state. Props can be understood as the function parameters, they are passed to the components instead of function in react. State contains the input data that would change as the data changes on an input form. It can be understood as an object that is stored inside a component class as its property.

Web3JS

A collection of different libraries bundled together for interaction of developer's system to the Ethereum node that can be remote or local [74]. The main purpose of web3js library is to help in developing such client application that can interact with the Ethereum blockchain. The various functionalities that it offers are creating smart contract, read & write operations using smart contracts and sending ether (Ethereum crypto currency) from one account to another on Ethereum blockchain. For interaction with the Ethereum, web3js uses the JSON-RPC (remote procedure calls) for its interaction with Ethereum blockchain.

Web3JS has a number of sub-dependencies, an important one of them is explained below,

Infura

For a DApp to function it needs to interact with Ethereum blockchain the above-mentioned libraries are used for this purpose but to have access to the Ethereum blockchain we need to form some sort of connection between them, this is provided by the Infura RPC URL. Its main features are that it is blockchain-based service, reliable and a secure distributed storage system [75]. It can be used for providing developers a free service to connect to the Ethereum node. Infura can be used as a tool that is an alternative for Geth and Parity that are used for running your own Ethereum node.

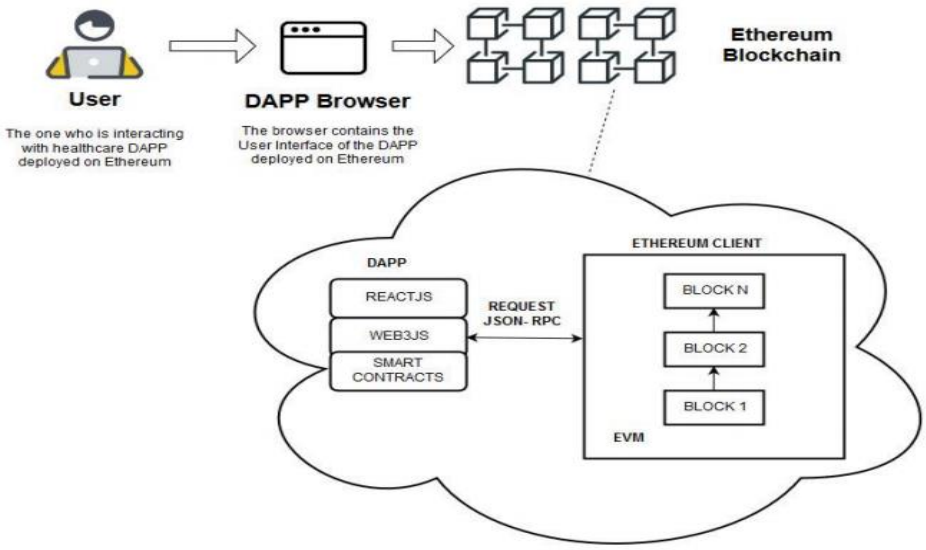

Figure 8: User Interaction with DApp

In technical terms, the working of system could be defined as that a user be able to perform multiple operations. The access to these functions depends upon the role of the user i.e., a Contact Tracer can monitor and observe all the records patient submit and can generate messages compromising of preventive measures. And a patient would be able to add his biodata along with the symptoms he is observing in his body. Patient has to submit symptoms on the daily basis for a required period (varies for each patient) according to his health condition. After submitting daily reports, Patient would receive messages from the assigned Contact Tracer corresponding his condition. As record of patient would be of huge volume so we also provide on off-chain storage solution of using Inter Planetary File System (IPFS). The patient record being stored would contain basic patient information along with IPFS hash. The IPFS hash could contain the lab tests or other information that are to be stored with patient's medical records.

Usage Scenario 1 – Activity Flow

Access Granted: Let us now understand the flow of activities for a user i.e., patient who intends to add his records and daily observations on the system. The figure 9 below depicts the whole process of this scenario's activity flow. This whole process starts from the Administrator, who is some trusted individual inside the hospital or healthcare organization. The administrator is responsible for assigning the Roles to various users of the system. This individual would obviously have some technical skills and experience as well for understanding and using the system. So, the first activity would be that administrator assigns roles and this would include Role Name and Account Address of the user who is being assigned that role. Every user of this proposed system would have a role name and account address for using the system. So, after administrator assigns this user some role, that role name and account address is stored in a roles list for validation purpose required in later steps.

After roles are assigned, now when a patient wants to perform some operations on the proposed system, he would at first request to perform them. The system would verify the patient's role name and account address from the Roles List and allows the patient to perform those functions after validation returns success. The patient would perform the desired functions and the system would store the information on the Blockchain that would perform transactions for that information. Once the transaction is confirmed the system receives the message of success from the blockchain layer that patient can view on the DApp browser on which the whole proposed system is being visible.

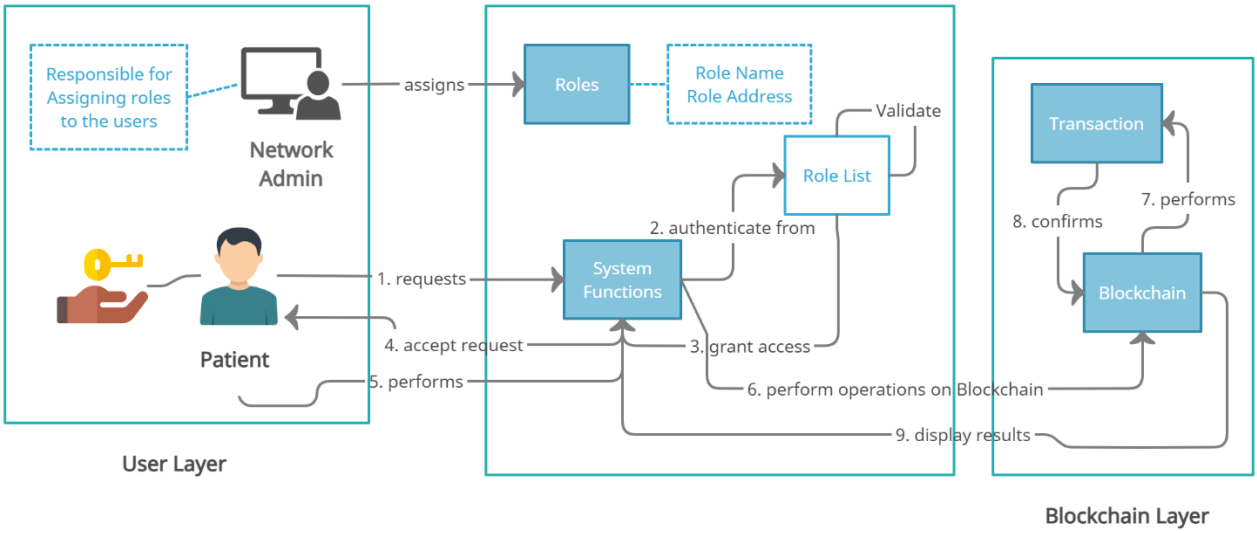

Figure 9: Usage Scenario Patient (Access Granted)

Access Denied: Let us now understand the scenario when a user who is not assigned a role by the administrator or is not using the account address that is assigned the roles would try to access the system functions. As seen in figure 10 below in such a scenario system would deny access to the user requesting to use to the function. In this process when a user requests to access dashboard of the system, the system checks Roles List to determine that should the user be allowed or not to use the various functions of the system. If the user is not assigned a role for performing these functions the system deny access to the user and indicating the user failed to authorize.

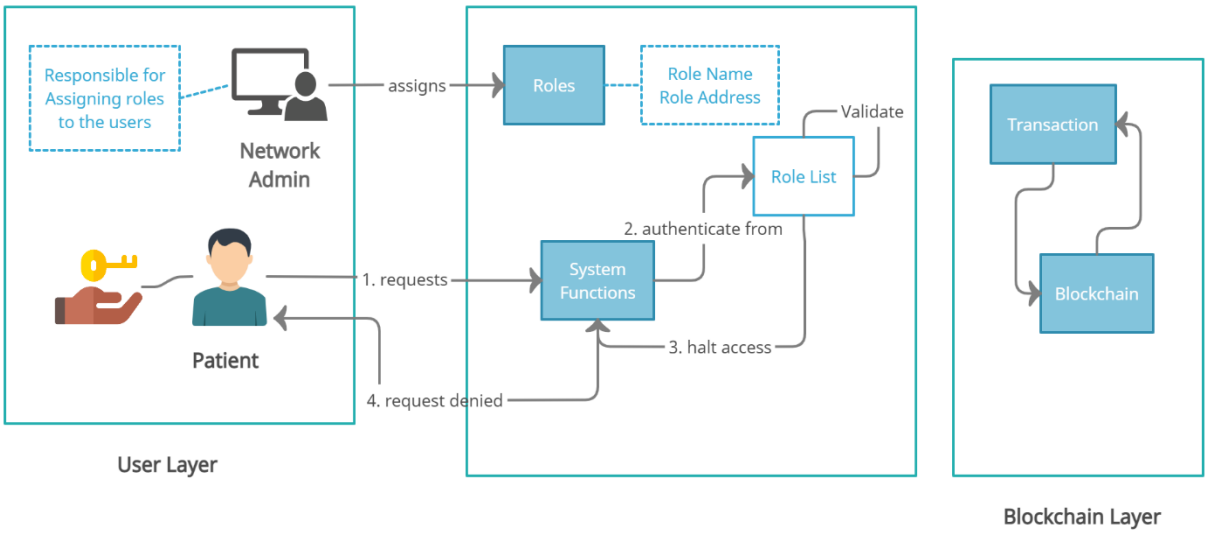

Figure 10: Usage Scenario Patient (Access Denied)

Usage Scenario 2 – Activity Flow

The second scenario for this system would be when a Contact Tracing wants to observe the condition of all the patients and after observing giving his feedback and prompting all the potential contact to quarantine themselves as quickly as possible. The system's administrator would assign a role to the Contact Tracer, containing the role name and an account address. The contact tracer using that account requests to use the system function of View Records.

When the contact tracer requests to use the view function the system verifies from the Role List and after validation is done and it results in success the system allows the contact tracer to view all the medical records. The system fetches the information from the Blockchain that would perform transactions for that information. Once the transaction is confirmed the contact tracer can view this information on the DApp browser on which the whole proposed system is being visible. Contact tracer can then send message to any patients that need assistance or guidance related to their health. Contact tracer can also send quick response to all the potential contacts to provide them with preventive measures. The figure 11 below depicts this whole scenario.

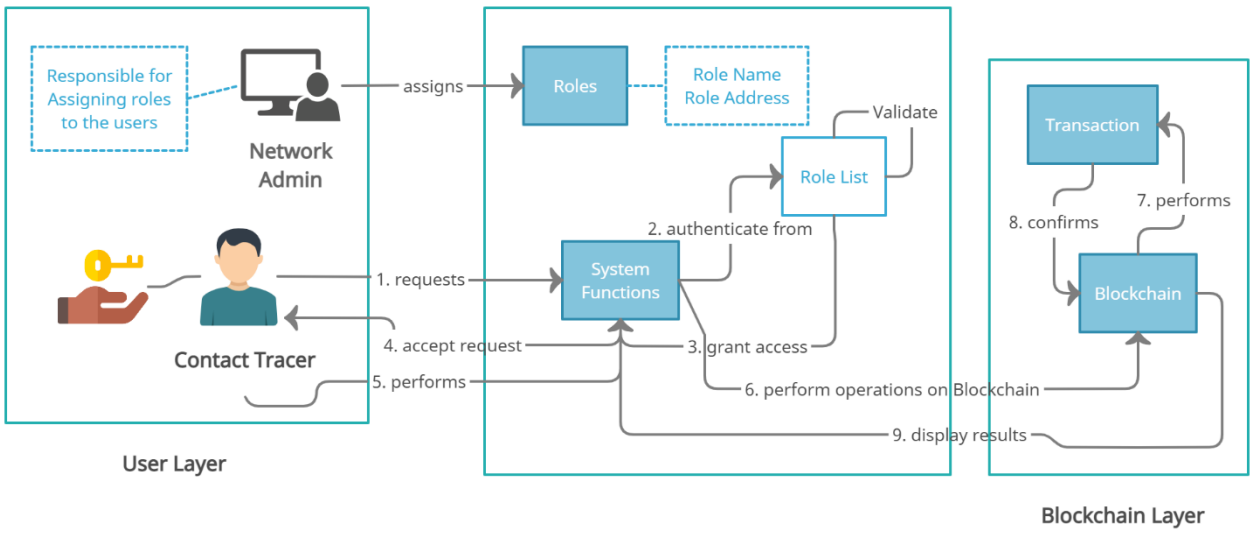

Figure 11: Usage Scenario Contact Tracer

The above section explains the usage process of various scenarios that arises during the usage of the decentralized applications. The next section explains the decentralized system's modular state.

In short, the functionalities provided by the system how these are incorporated while the implementation of the system is explained in this section.

A. Assign Roles:

This phase of the DApp is handled by an administrative entity, which is trusted by the hospital and is assigned with the job of defining the roles of a user on the system. The Assign Roles sends (Role Name, Roll Address) to the Blockchain. These both act as the unique identification of a user on the system. Just like the system has two different users on the system, the roles would also be of two kinds i.e., Assign Roles might assign roles to Contact tracer or Assign Roles could also assign roles to Patient. The Assign Roles sends (Role Name, Role Address) to Blockchain that is stored in Role List for Patient and Contact Tracer that are separate to both of these entities.

B. User Login:

For using the main functionality of the proposed system, users need to be authenticated first. The login would ask for the Role Address and the Role name that is assigned by the administrator entity. This will send a request to Blockchain for verification of the user recorded in the Role List. If they match, user would be logged in to the system to perform multiple operations according to their role. If not, error of unauthorized entity will be displayed.

C. Add Patient Records:

At this stage, Patient is assigned the role to perform various functions on DApp. When requesting the Blockchain to add records related to biodata on the Blockchain, the patient sends record to be stored on Blockchain.

D. View Patient Records:

The patient uses Role Address assigned to him to perform the view function on his medical records. The system would follow the steps below to perform this function,

1. After logging in to the system, the system now matches the Patient's address to whom its being logged in with the record stored in Blockchain.
2. After validation at step 1 is done, the record R is available on the user's system.

E. Add Patient Symptoms and Test Record:

Patient will monitor his own symptoms and daily submit it to the system for contact tracer to view and monitor his health. The system would follow the steps below to perform this function,

1. After authentication, the patient can add the symptoms and test reports to an off chain named, IPFS.
2. IPFS stores the information in distributed hash table.
3. IPFS assigns hash to the information file stored.
4. Sends the hash information to blockchain.

F. Get Message:

Patient can Get Message from the contact tracer related to the preventive measures or the emergency information incase things go wrong. After patient being logged in, system will automatically generate the message first up the screen from the contact tracer.

G. Add Contacts:

Patient can add contacts information if he test positive of the disease. Whosoever he's in close contact with, approximately within the range of 1 metre, for more than 15 mins in last 3-4 days or 1 week, he will provide information of them all. So that contact tracer can track contacts as soon as possible to halt the spread of this disease.

H. View All Records (For Contact Tracer):

This function is for Contact Tracer it was mentioned in our proposed framework that Contact Tracer would be allowed to view all the records of the patients to help them be aware of their health situation and if condition of patients worsen then, Contact Tracer would provide the emergency information related to hospitals available and their details etc. The system would follow the steps below to perform this function,

1. The system first would verify and validate the Contact Tracer's Role Name, Role Address from Role list that are stored on Blockchain. After ensuring that sender is the Contact Tracer who is allowed to perform the View/ Retrieve operation.

2. Contact Tracer can view single Patient's record as well as the whole record of all patients. For viewing single patient, the system now matches the Patient Address with the record stored in Blockchain.
3. For viewing record of all patient, the system would return all record stored in Blockchain.

I. Send Message:

After evaluating the symptoms of a particular patient, Contact Tracer would be able to send quick response to the patient regarding their health, preventive measures or emergency information. The system would follow the steps below to perform this function,

1. Contact Tracer would specify the Patient's address to whom he wants to send the message as well as the message.
2. This message would store on the Blockchain and deliver to particular patient immediately.

Chapter 5

Testing and Performance

CHAPTER 5: TESTING AND PERFORMANCE

In this chapter, the applicability and validity of our proposed approach is presented with the help of the following step by step guide for successful running of the implementation code. Details about testing environment and guide are given in Section 5.1 and performance of the proposed system is discussed in Section 5.2.

5.1. Testing

Step 1:

For testing the proposed system on a particular device, a variety of dependencies are required to be installed first on the system to run this smoothly. A package called node.js must be installed on the device. For running the proposed system on a local host, node package manager, npm is required to be installed. List of dependencies and essential packages are as following for properly building the testing environment.

- Node.js
- NPM
- Truffle
- Ganache
- Git
- MetaMask

Step 2:

After installing all the required dependencies next step would be to compile the proposed system code. For this purpose, enter the following command in Git console that is opened through the root folder, where all files of implementation rest.

```
truffle compile
```

This command will compile all the code and look for any errors that the code contains related to syntax etc. This is a vital step to be done before further processes because it has a potential to create problem later on.

Step 3:

Next step is to migrate the smart contract to the Ethereum blockchain using Ganache. Ganache is a platform used for testing and deployment of DApps by the developers of the blockchain. The command used for that purpose is as follows:

```
truffle migrate -reset
```

For deployment of every smart contract there's a file that our project must contain that is named migration. The compile command also checks for the existence of this file. The reset flag would reset all the past commands and push that contract to migrate for testing. A receipt would be generated on the Git console that includes accounts used for deployment, the gas price used for migration, the account address, gas limit and the total number of blocks added to Ganache.

Step 4:

Next step is to check whether the blocks have been added to the Ganache after migration. If blocks are not added to the Ganache after successful migration you need to look into the truffle.config file for further errors in setting up the configuration.

Step 5:

Now for actual running of the proposed system on the browser, type the following command on Git console in the 'client' folder of the project.

```
npm start
```

This command would open up the localhost server on the browser and display the UI of the project. All the UI source files are contained in a client folder that defines the system functionality.

Step 6:

Next step is to open up the MetaMask extension on browser for the connectivity of the accounts of user or the tester of MetaMask with the project.

Step 7:

Project would most probably be functional by now so we would test the functionality or behavior of the proposed system. First step is to assign roles for each account that needs to be signed in. For this proposed project the roles are patient and contact tracer. Administrator would be the one to assign these roles to different accounts.

Step 8:

After testing the whole project, one can stop the project by typing Control + C command on the Git console to terminate the running project.

5.2. Performance

In this section, the performance of the proposed solution is discussed. Comparison with related frameworks and challenges derived with this novel technology is also presented in this section. **Section 5.2.1** covers the system specification required for testing. **Section** Error! R eference source not found. presents the performance assessment of the proposed framework. Lastly, the comparison of proposed framework with related work has been provided in **Section** Error! Reference source not found..

5.2.1. System Specification

The specifications of system used for network simulations are:

- Intel Core (TM) i5-8250U CPU @ 1.60 GHz processor
- And 8.00 GB of memory with Windows 10 Home (64-bit)

Solidity is the programming language used for development of this proposed framework. JavaScript and Python are encapsulated in the Solidity language which is provided by the Ethereum to write code in smart contracts.

5.2.2. Performance Assessment

In this subsection, we will discuss how our proposed system would perform different functions by various users in real case scenarios. For performance evaluation, we used Apache JMeter version 5.3, which is popular in testing environment. It is an open-source testing software for analysis and measuring performance of various services provided by the applications [24].

Vulnerability Analysis:

Safeguarding the crucial and valuable data is vital part of any business. The code needs to be free of risks of security breaches and any vulnerability that may lead to future losses [77]. The smart contract defines all of the functionality of our proposed system, to make it all rounded and free of security vulnerabilities, the code needs to be thoroughly tested. There are many security analysis tools that provide this facility for checking code. Oyente, a smart contract auto-auditing tool, analyze smart contracts and returns possible bug attacks on it including the famous DAO attack [78]. This was developed by researchers from National University of Singapore in Jan 2016.

It provides analysis report with various attributes that verifies the availability of security threats. The analysis report for our smart contract is shown in Figure 12. All the vulnerabilities are checked false that means there are no security bugs in our smart contract.

```
INFO:root:contract CTracing.sol
INFO:symExec: ============ Results ===========
INFO:symExec: EVM Code Coverage:  9.5%
INFO:symExec: Integer Underflow: False
INFO:symExec: Integer Overflow: False
INFO:symExec: Parity Multisig Bug 2: False
INFO:symExec: Callstack Depth Attack Vulnerability: False
INFO:symExec: Transaction-Ordering Dependence (TOD): False
INFO:symExec: Timestamp Dependency: False
INFO:symExec: Re-Entrancy Vulnerability: False
INFO:symExec: ====== Analysis Completed ======
```

Figure 12: Security vulnerability report by Oyente tool

Cost Analysis:

We can calculate the transaction sizes of various functions of our proposed framework by using the data payload. Keeping in mind that transaction sizes are specifically calculated for various functions of Algorithm 1 through data payload point of view. All these transactions cost some gas or fee, that can also be calculated in 'ETH' with units wei or gwei. The transaction fee for a transaction is the product of gas consumed and gas price. It could be represented as follows,

```
(gas limit x gas price) = transaction fee
```

Table 7: Function Caller, and Gas of proposed framework

Function Caller	Function Name	Size	Fee[Gas]
Admin	registerPatient	132 bytes	0.001321 ETH
Admin	registerContactTracer	132 bytes	0.001324 ETH
Patient	addPatientRecord	580 bytes	0.008994 ETH
Patient	addTestResult	356 bytes	0.004498 ETH
Patient	viewRecord	112 bytes	0.003124 ETH
Patient	getMessage	36 bytes	0.002976 ETH
Contact	sendMessage	228 bytes	0.003785 ETH
Contact	viewPatientRecord	420 bytes	0.004125 ETH

We can calculate the transaction fee by using the recommended figure for gas consumed which is 66199 and is 20 Gwei for gas price. So,

$$\text{Transaction Fee} = 66199 \times 20 = 1323980 \text{ Gwei}$$

And to calculate the transaction fee of 1ether, we know each gwei is equal to 0.000000001 ETH (10-9 ETH).

Transaction Fee for

$$1 \text{ Ether} = 1323980/1000{,}000{,}000 \text{ Gwei} = 0.0013 \text{ Gwei}$$

The transaction fees for various functions of Algorithm 1 are presented in table 7.

Average Execution Time:

Time requires to completely and successfully executes a specific function by a user, is known as execution time. With the increment in number of transactions, the time to execute them would also increase. The functions against we evaluated execution time in this subsection are defined in Algorithm 1 of section. For single user, our system performed the functions Assign Roles, Add Patient Records, View Patient Records, Send Message and receive message for about 10.7 sec, 1 min 9 sec, 50 sec, 9 sec and 4 sec respectively. This time would increase as the number of users using the system increases simultaneously.

Throughput:

In Figure 13, the throughput of different segments of Contact Tracing smart contract are analyzed, in terms of number of users ranging from 100 to 500 that are performing different functions explained in Algorithm 1. The unit for Throughput is in Data/time, i.e., KB/sec in JMeter.

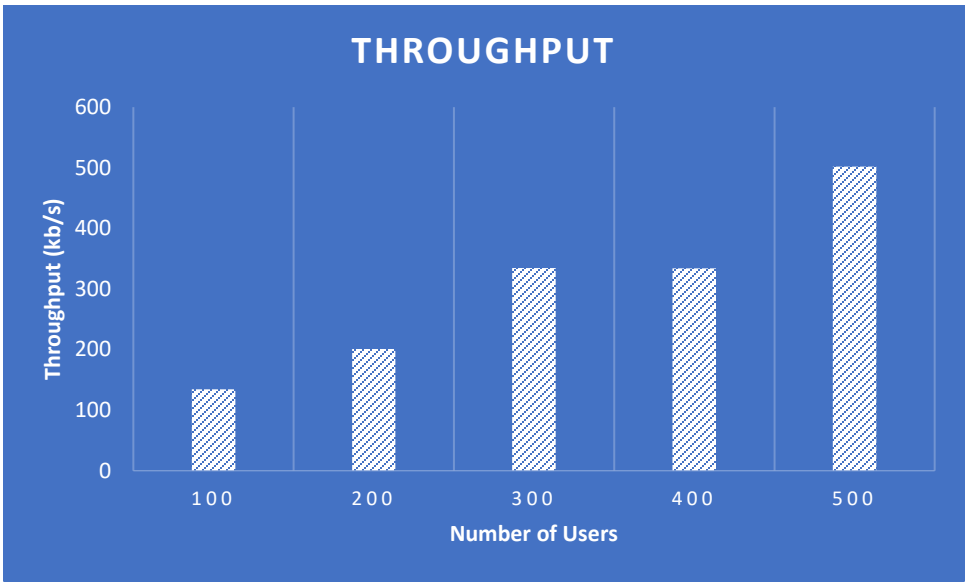

Figure 13: Throughput of the proposed system

A number of experimented performed on our proposed framework to evaluate the performance of the system. From this, we conclude that there is a smooth linear graph formed between number of users and overall throughput indicating the efficiency of the proposed framework.

Average Latency:

In Figure 14, the average latency of our proposed solution is presented in terms of throughput. The latency is the time between the request sent to a system and the response received by the system. JMeter is used to evaluate latency for this proposed framework. Milliseconds is the unit for latency in JMeter. In this experiment, 13ms is recorded as the highest latency.

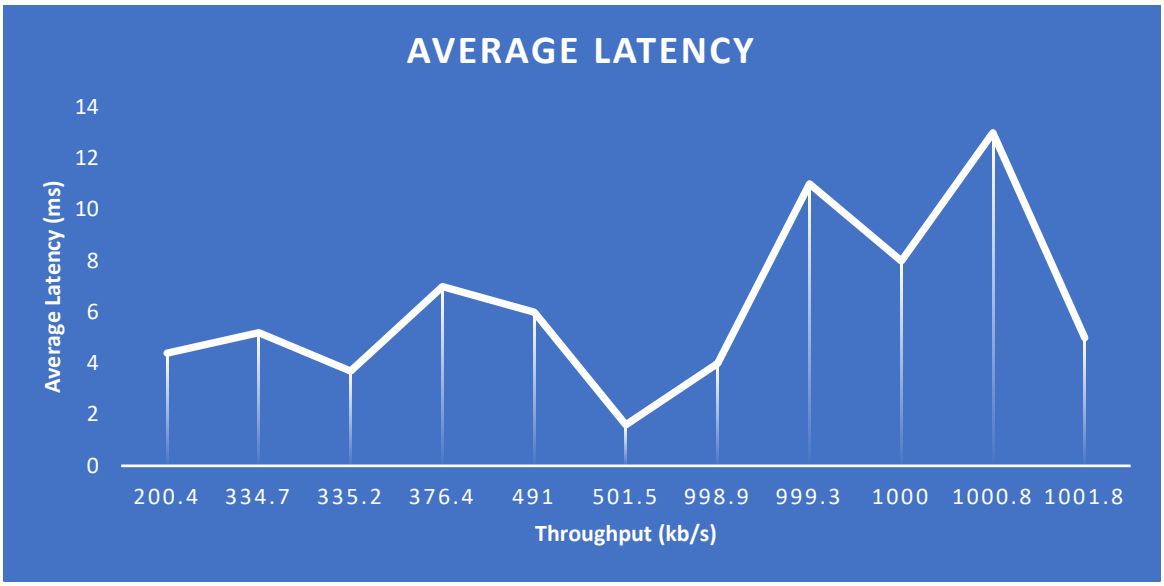

Figure 14: Average Latency of the proposed system

The lower the latency, quicker would the app respond. According to NHS the number of New COVID cases reported on the 19th of Nov were 30 in the city of ABERDEEN in UK with population of 189,120 [33]. This means that with the increase in users our system would support the upcoming users with minimum latency.

5.2.3. Comparison of Proposed Framework with Related Work

In this section we discuss some parameters that made our proposed contact tracing system better in comparison with other systems. To cope up with the challenges, we present our blockchain-built contact tracing that fulfils both privacy and performance requirements. For post pandemic contact tracing all the below aspects are required for better capability to fight against this novel disease. These parameters included:

- Enhanced privacy as the main focus
- Access control
- Scalability
- Trustworthy Information
- Fault tolerance
- Availability

ENHANCED PRIVACY AS THE MAIN FOCUS:

Privacy is the most significant and alarming factor of contact tracing as information has to be collected, traced and distributed. Protecting the identity of users is also a major concern. Meanwhile, Information handling in centralized system has a risk of manipulation and corruption. Nevertheless, it is not a trouble for blockchain, it has the capability to bridge out these differences and create a distributed environment for the tracing participants. Rather than relying on rules and regulation in centralized system, Blockchain offers privacy preserving technical solution, where the identity is removed at the beginning and it can further combine with encryption technologies to provide ultimate confidence in privacy.

ACCESS CONTROL:

By controlling the access of the users, this framework ensures that no unauthorized party would be able to access the system. Using blockchain technology ensures no third party intrusion, that makes our proposed solution more secure. Confidentiality ensures that the communication made between any two parties cannot be interpreted by anyone else. Therefore, unauthorized access to the data cannot take place. Confidentiality can be achieved through encryption. Therefore, in our solution, message encryption and decryption is achieved through systematic key encryption that ensures the authentication of the user. Every authorized user has a defined role and according to that specified role, user have granular access to the system ensuring security of personal records of the patients.

SCALABILITY:

One challenges that blockchain is facing is scalability that requires a solution. For developing a robust framework, we used Interplanetary File System (IPFS) that is off-chain storage mechanism [25]. The data is encrypted using symmetric key encryption. For data decryption a unique private key is required, that provide protection against malicious attacks because it’s not easily guessable, therefore security of the framework is not compromised. The IPFS hash is then stored in the blockchain, which has a trait of being tamperproof, so it is difficult to perform any malicious activity. Hence, the data size of records stored in blockchain has now decreased and system performs adequately.

TRUSTWORTHY INFORMATION:

Misinformation can cause panic to the general public and leads to inaccuracy of data; hence it slows the process of pandemic prevention. Information inaccuracy and information transparency

are the main reasons for the misinformation. The higher authorities can have a motive to not disclose the accurate information or provide false records according to their indulgence in centralized mechanism. But that's not the case with blockchain. Blockchain's transparency feature comes handy for preventing data misinformation normally caused by third parties and authorities by providing verifiable trusted tracing information.

FAULT TOLERANCE:

For overcoming the problem of single point of failure, we proposed to incorporate the benefits of IPFS in our system. This storage system is distributed that acts as a P2P network. To store, each file is broken down into different portions and then assign them a specific node on the network.

AVAILABILITY:

Interplanetary File System is used for data storage in our proposed framework. This distributed storage system generates hashes against each data storage. A request is sent to IPFS whenever the data is required, and it will direct to that specific node where the data are kept. This distributed storage mechanism would ensure the availability of data while maintaining high throughput of the system.

Evaluation:

The following table 8 compares our proposed framework benefits and features with that of the related work [26] [27] [28] [29]. The above defined features offered by our proposed framework are blockchain-based; privacy preserving, security of technology and scalability are included in this comparison. These features are then compared and observed that whether they exist in the related work under consideration or not.

A Most recent releases of contact tracing solution are NHS COVID-19 App [26], TraceTogether from Singapore [27], Google/Apple joint contact tracing project [28], and China Health code system [29]. These new launches are premature at the moment and results have also not been published, so it's hard to tell the success rate. Every scheme is facing some challenges that need to be addressed.

Health Code System uses QR code associated with each user for relational crossmatch. This centralized approach does not preserve privacy and user identity. As for power usage, this

approach scans only at the time of passing checkpoints that reduces the consumption of battery and data [29]. The network coverage can be easily stretched due to its central hierarchy.

Contact tracing developed by Google Apple protects user identity and thus this approach becomes privacy preserving. But still, due to the use of central server for searching of contacts and notification, privacy can come under attack and intruder can access user credentials from the server [28].

In TraceTogether, BLE (Bluetooth low energy) is used to find and locate potential virus carriers. A drawback of this scheme is that it requires the user device to be on active mode all the time, hence it consume a huge amount of device power. Bluetooth contact tracing solutions are not secure, thus have a risk of data misuse that creates huge panic among general public. This being a centralized service, thus inherit all the related drawbacks making privacy a major concern [27].

Similar to all the other technique NHS COVID-19 App is also struggling with user privacy, concern of misuse of user information. This app also uses BLE that eventually leads to the battery drainage of user device. Long term impact of this technology over health care is still questionable. Lacking infrastructure, limited interoperability and underinvestment are some of the underlying challenges of NHS that needs to be addressed [30].

Table 8: Comparison of proposed framework with related work

	[26]	**[27]**	**[28]**	**[29]**	**Our Proposed System**
Blockchain Based	N	N	N	N	Y
Privacy Preserving	Yes, Partially	N	Yes, Partially	N	Y
Security of Technology	Low	Low	Low	Medium	High
Scalability	N	N	N	N	Y

Chapter 6

Conclusion and Future Work

CHAPTER 6: CONCLUSION & FUTURE WORK

6.1. Overview

COVID-19 a deadly virus that took over the world like a storm, has affected life on daily life of citizens and slowed down the global economy. This pandemic has drastic effect on human health and has led to loss of numerous human lives worldwide. This infectious disease has multiple symptoms like fever, cold, shortening of breath, nasal congestion and pneumonia that gets severe with time. This new-found virus has affected humans for the first time, so its vaccine is still under development. With no available vaccine, the most appropriate remedy would be following non-pharmaceutical interventions (NPIs) strictly. The goal of NPIs is to reduce the transmission of the virus and has been proved very effective in previously occurred pandemic of same scale.

NPIs targets on social distancing by defining a certain distance between individuals and by avoiding large gathering. Closing of crowded areas like school, restaurants and social events. Enforcing strict quarantine to reduce physical interaction and widespread of this communicable disease. A number of countries that implemented or strictly imposed these NPIs have got fruitful outcomes that is the reduction in the number of new COVID-19 cases. But these measures and social disruption has created a havoc in business industry, many have lost their jobs, and many are on the verge of falling into extreme poverty.

To get a hold on the conditions, multiple researchers and developers have created different approaches to reduce COVID-19 impact with the help of technology. The advancement in technology has benefited many businesses and has provided new vision to the lacking ones that hasn't even imagined before. Therefore, getting help from the technology in this pandemic will save lives and overall well-being of humans. Although, these proposed approaches have solved some challenges of COVID-19 but still exits many challenges.

Many developers have developed Contact tracing apps to control this infectious virus. Contact tracing is a method to trace and identify all the possible virus carrier that have been in close contact with COVID-19 patient. After identification, monitoring begins where patient or care giver will observe the symptoms and report on daily basis and can transfer to a hospital if condition worsen.

These approaches are good enough in terms of controlling the pandemic, but there also arises concerns of data privacy, user ownership of data, integrity and security that are critical to patients. To solve this blockchain technology could be used as it offers multiple benefit such as immutable data, secure transactions, ownership and many more.

Satoshi Nakamoto introduced blockchain technology called bitcoin [8], a digital cryptocurrency. This technology consists of blocks that are interconnected and keep on increasing as the records increases. Blocks are used for storage of records. Blockchain technology is a decentralized platform where information is stored in a distributed or shared manner.

Combining blockchain technology with contact tracing would solve many challenges that are critical. Blockchain technology discards the third-party intervention, that makes it an excellent choice for storing privacy critical patient's data. It offers data immutability, security, data integrity and privacy of data. This technology will benefit the patient and authorities in terms of data accessing across different platforms and encryption of patient's data.

Blockchain technology has a capability to address various challenges faced by contact tracing approaches. It being decentralized won't allow only one medium to control the entire network. The data stored in blockchain would be encrypted and it won't entertain third party intervention thus ensuring security. This technology ensures to provide granular access for all the users such as patients, contact tracer, nurses, doctors and managing authorities of the system.

The Aim of the Thesis was to build a system that uses benefits of blockchain technology in the field of contact tracing and providing quick responses to the patients so that many lives can be saved in this pandemic. We intended to build a decentralized system that could store the patient's critical data in encrypted form and give access to authorized and concerned individuals only. According to the data provided by patient or the care giver, our system will be able to track possible virus carriers and provide assistance to the patients corresponding to their symptoms. We also proposed to solve the scalability challenge of blockchain by using off chain scaling method. For this, IPFS a decentralized storage system is used. Initially blockchain is not designed to store huge amount of data, so we are storing all the data on IPFS using peer to peer network.

6.2. Conclusion

In this research work we discussed different challenges of the existing contact tracing system and how can it be better with the use of blockchain technology. This naive technology has been profiting many industries from the time it discovered. Detailed discussion on the challenges of contact tracing has also been covered to give deep insight of the reason behind the usage of blockchain technology.

Our proposed system is implemented on Ethereum platform and its dependencies. Ethereum is an open source distributed blockchain network used to create smart contacts according to programmers need for research.

Our proposed framework is a secure and resource effective blockchain-based data storage mechanism with role-based access for Contact Tracing is proposed. Only authenticated and trusted individuals can access the proposed framework by this novel technology, i.e., blockchain. Challenges of centralized data storage are known worldwide and to tackle those IPFS, a distributed file storage system is considered. Blockchain technology in vision new advanced capabilities in contact tracing field and addressed some issues that faced by existing contact tracing solutions.

This system offers benefits such as information confidentiality, access control, guarding privacy, scalability, quick response. These advantages will definitely leave a benchmark for further research.

6.3. Future Work

Our approach to trace each individual that has a potential of carrying the virus and informing them on time is efficient and scalable. Users can granularly access the system according to their role defined by the administrator that will ensure user privacy. All the data is being encrypted first and then stored on the off-chain storage to make our system more scalable and to overcome the challenge of scalability in blockchain technology.

In future, we plan to implement the existing framework more general basically for all expected pandemics. We would develop a module where patients can get assistance online from doctor, merging contact tracing with healthcare sector. Present decentralized frameworks are restricted to a local network, hence using third party suppliers would boost the coverage globally

without much effort. Payment module for online assistance would also be developed in future research works. Future versions would also include Blockchain interoperability and cross-chain functionalities that would widens the adaptability of blockchain [79]. We also aim to modify our current system with additional vaccine module in our future releases.

APPENDIX A

Patient Records Smart Contract Code

```
// SPDX-License-Identifier: MIT
pragma solidity 0.5.0;

contract PatientRecords {

 struct Role {
     string role;
   }

  struct Message {
     string msg;
   }

   struct Record {
     uint id;
     string name;
     string age;
     address patient;
     string phone;
     string addres;
     string symptom;
   }

   struct Test {
     string result;
     string detail;
     string ipfs;
     address patient;
   }

    struct Contact {
     string name;
     string phone;
     string name2;
     string phone2;
     address patient;
   }

   struct Chain {
     uint id;
     string na;
     address name;
      }
```

```
  mapping (address => Message) public messages;
  mapping(address => Contact) public contacts;
  mapping(address => Test) public tests;
  mapping(address => Record) public records;
  uint public count;

  mapping (address => Role) public roles;
  mapping(uint => Chain) public chains;

  function addRole(address _recipient, string memory _role) public {
    roles[_recipient] = Role(_role);
  }

  function readRole(address v) public view returns(string memory) {
    return(roles[v].role);
  }

  function addPatientRecord(string memory _name, string memory _age, address _patient, string memory
_phone,string memory _addres,string memory _symptom, address _recipient)
      public {
      count++;
      chains[count] = Chain(count, _name, _patient);
      records[_recipient] = Record(count, _name, _age, _patient, _phone, _addres, _symptom);
      }

     function viewPatientRecord(address v)
      public view returns(uint, string memory, string memory,address ,string memory, string memory,string
memory) {
      return(records[v].id,records[v].name, records[v].age, records[v].patient,records[v].phone,
records[v].addres, records[v].symptom);
      }

      function addPatientTest(string memory _result, string memory _detail, string memory _ipfs, address
_recipient)
      public {
      tests[_recipient] = Test(_result, _detail, _ipfs, _recipient);
      }

      function viewTest(address v)
      public view returns(string memory, string memory, string memory) {
      return(tests[v].result, tests[v].detail ,tests[v].ipfs);
      }

      function addContactInfo(string memory _name, string memory _phone,string memory _name2, string
memory _phone2, address _recipient)
      public {
      contacts[_recipient] = Contact(_name, _phone, _name2, _phone2, _recipient);
      }

      function viewContactInfo(address v)
      public view returns(string memory, string memory,string memory, string memory){
```

```
    return(contacts[v].name, contacts[v].phone, contacts[v].name2, contacts[v].phone2);
    }

    function sendMessage(address _recipient, string memory _message) public {
    messages[_recipient] = Message(_message);
    }

    function readMessage(address v) public view returns(string memory) {
    return(messages[v].msg);
    }

    function deleteMessage(address v) public {
    delete messages[v];
    }
}
```

Migration Code:

```
var PatientRecords = artifacts.require("./PatientRecords.sol");

module.exports = function(deployer) {
  deployer.deploy(SimpleStorage);
};
```

JavaScript Code:

App.js Code

```
import React, { Component } from "react";
import SimpleStorageContract from "./contracts/SimpleStorage.json";
import getWeb3 from "./getWeb3";
import { Select, MenuItem, TextField, Button, Switch, Table, TableBody, TableCell, TableHead,
TableRow } from '@material-ui/core'
import "./App.css";
import ipfs from "./ipfs"

class App extends Component {
  constructor(props){
    super(props)
     this.state = { web3: null, accounts: null, contract: null, showSignup: false, showSignin: false,
showPatientForm1: false, showNav: false, showData: false, showTest: false, showContact: false,
disabled: false, showContactNav: false, showPatientRecord: false, showSendMessage: false,

      selectrole: '', address: '', irole: '', pname: '', paddress: '', page: '', pphone: '', psymp: '', ptest: '', pdetail: '',
cname: '',cphone: '',ccname: '',ccphone: '', chains : [], ppaddr: '', result: [], testrecord: [], contactrecord: [],
cmsg: '', cpaddr: '', message: '', noValue: '', ipfsHash: ''};

    this.handleSubmit = this.handleSubmit.bind(this);
    this.handleSubmit1 = this.handleSubmit1.bind(this);
```

```
    this.handleSubmit2 = this.handleSubmit2.bind(this);
    this.handleSubmit3 = this.handleSubmit3.bind(this);
    this.handleSubmit4 = this.handleSubmit4.bind(this);
    this.handleSubmit5 = this.handleSubmit5.bind(this);
    this.handleSubmit6 = this.handleSubmit6.bind(this);

    this.operation = this.operation.bind(this);
    this.operation1 = this.operation1.bind(this);
    this.operation2 = this.operation2.bind(this);
    this.operation3 = this.operation3.bind(this);
    this.operation4 = this.operation4.bind(this);
    this.operation5 = this.operation5.bind(this);
}
 componentDidMount = async () => {
  try {
    // Get network provider and web3 instance.
    const web3 = await getWeb3();
    // Use web3 to get the user's accounts.
    const accounts = await web3.eth.getAccounts();
    // Get the contract instance.
    const networkId = await web3.eth.net.getId();
    const deployedNetwork = SimpleStorageContract.networks[networkId];
    const instance = new web3.eth.Contract(
      SimpleStorageContract.abi,
      deployedNetwork && deployedNetwork.address,
    );
    // Set web3, accounts, and contract to the state, and then proceed with an
    // example of interacting with the contract's methods.
    this.setState({ web3, accounts, contract: instance });
  } catch (error) {
    // Catch any errors for any of the above operations.
    alert(
      `Failed to load web3, accounts, or contract. Check console for details.`,
    );
    console.error(error);
  }
 };

 captureFile = (event) => {
   event.stopPropagation();
   event.preventDefault();
   const file = event.target.files[0];
   let reader = new window.FileReader();
   reader.readAsArrayBuffer(file);
   reader.onloadend = () => this.convertToBuffer(reader);
   };

   convertToBuffer = async(reader) => {
   const buffer = await Buffer.from(reader.result);
   this.setState({buffer});
   //console.log('buffer', this.state.buffer)
```

```
    }

     //function for sending the buffer to ipfs node
     //and shows the ipfs hash onto the UI
      onIPFSSubmit = async(event) => {
      event.preventDefault();
      await ipfs.add(this.state.buffer, (err,ipfsHash) => {
      console.log(err,ipfsHash);
      this.setState({ipfsHash:ipfsHash[0].hash})
      //console.log('IPFS hashhhhhhhhhhhh', this.state.ipfsHash);
      })
     }

updateSelectVal= (e) =>{
   this.setState({selectrole : e.target.value})
   //console.log(this.state.selectrole)
  }

operation(){
   this.setState({
     showSignup:!this.state.showSignup
   })
  }

  operation1(){
   this.setState({
     showSignin:!this.state.showSignin
   })
  }

  async operation2(){
   this.setState({
     showData:!this.state.showData
   })
   const { contract } = this.state;
   const response = await contract.methods.viewPatientRecord(this.state.accounts[0]).call()
     this.setState({ response })
  }

  operation3(){
   this.setState({
     showTest:!this.state.showTest
   })
  }

 async operation4(){
   this.setState({
     showPatientRecord:!this.state.showPatientRecord
   })
   const { contract } = this.state;
```

```
  const recordCount = await contract.methods.count().call();
  this.setState({ recordCount });
  for (var i = 1; i <= recordCount; i++) {
    const record = await contract.methods.chains(i).call()
    this.setState({
      chains: [...this.state.chains, record]
    })
   }
}

operation5(){
  this.setState({
    showSendMessage:!this.state.showSendMessage
  })
}

handleChange(event,key){
  this.setState({[key]: event.target.value });
}

async handleSubmit(event){
  event.preventDefault()
  const { accounts, contract} = this.state;
  const response = await contract.methods.readRole(this.state.address).call();
  this.setState({ response });

  if (this.state.accounts[0] === '0x8cFdd9F729375C6AA5A3e123ec49Ee86d4619b16')
  {
    if (this.state.response === '')
    {

      let t0= performance.now(); //start time
      await contract.methods.addRole(this.state.address, this.state.selectrole).send({from: accounts[0]})
      let t1= performance.now(); //end time

      console.log('Time taken to execute role function:'+ (t1-t0) +' milliseconds');
    }
    else
    {
      alert(' This ' + this.state.address + ' already has a role ' + this.state.response )
    }

  }
  else{
    alert(' ***YOU ARE NOT ADMIN*** Choose Admin Account from Metamask ' )
  }
  console.log("Assigning roles done", this.state.selectrole)
}

async handleSubmit1(event){
  event.preventDefault()
```

```
const { accounts, contract} = this.state;

const response = await contract.methods.readRole(this.state.accounts[0]).call();
this.setState({ response });
if (this.state.irole === this.state.response)
{
  this.state.showSignin = false
  alert(' *Authentication Verified* ')

  if (this.state.irole === 'Patient')
  {
    let t0= performance.now(); //start time
    const response = await contract.methods.viewPatientRecord(this.state.accounts[0]).call()
    this.setState({ response })
    let t1= performance.now(); //end time

    console.log('Time taken to execute View function:'+ (t1-t0) +' milliseconds');

    if (this.state.response[1] === '')
    {
      this.state.showPatientForm1 = true
    }
    else
    {
      else
      {
        this.state.disabled = true
        // console.log(this.state.disabled)
      }
      this.state.showNav = true
      let t0= performance.now(); //start time
      const message = await contract.methods.readMessage(this.state.accounts[0]).call();
      this.setState({ message });
      let t1= performance.now(); //end time

      console.log('Time taken to execute recieve message function:'+ (t1-t0) +' milliseconds');

        if (message === '')
        {
        }
        else{
          alert('You recieved a message from our Contact Tracer '+ this.state.message )
          await contract.methods.deleteMessage(this.state.accounts[0]).send({from: accounts[0]})
        }
    }
  }
  else
  {
    this.state.showContactNav = true
  }
}
```

```
    else
    {
      alert(' *Authentication Failed* ')
    }
  }

 async handleSubmit3(event){
     event.preventDefault()
    const { accounts, contract} = this.state

    await contract.methods.addPatientTest(this.state.ptest , this.state.pdetail, this.state.ipfsHash,
this.state.accounts[0]).send({from: accounts[0]})
     if (this.state.ptest === 'Tested Positive')
     {
      alert(' Its OK that you tested Positive, you will recover soon. Self isolate and monitor yourself if
condition worsen contact Blue Hospital ')
      this.state.showContact = true
      this.state.showTest = false
     }
     else
     {
      alert(' Its Great news that you tested Negative but do follow SOPs and social distancing measures. ')
      this.state.showTest = false
     }
    this.state.disabled = true
  }

  async handleSubmit4(event){
     event.preventDefault()
    const { accounts, contract} = this.state
    await contract.methods.addContactInfo(this.state.cname , this.state.cphone, this.state.ccname ,
this.state.ccphone, this.state.accounts[0]).send({from: accounts[0]})
     alert('Done Thank You')
     this.state.showContact = false
  }

 async handleSubmit5(event){
     event.preventDefault()
    const { contract } = this.state
    const result = await contract.methods.viewPatientRecord(this.state.ppaddr).call();
    this.setState({ result })
    const testrecord = await contract.methods.viewTest(this.state.ppaddr).call();
    this.setState({ testrecord })
    //console.log(this.state.testrecord)
    const contactrecord = await contract.methods.viewContactInfo(this.state.ppaddr).call();
    this.setState({ contactrecord })
  }

  async handleSubmit6(event){
     event.preventDefault()
    const { accounts, contract} = this.state
```

```
  let t0= performance.now(); //start time
  await contract.methods.sendMessage(this.state.cpaddr, this.state.cmsg).send({from: accounts[0]})
  let t1= performance.now(); //end time

  console.log('Time taken to execute send message function:'+ (t1-t0) +' milliseconds');
   alert('Your Message has been sent!')
 }

 getvalue = (e,val)=>{
 //console.log(val)
 if (val === true)
 {
  this.state.ptest = 'Tested Positive'
 }
 else
 {
  this.state.ptest = 'Tested Negative'
 }
}

 render() {
  //if (!this.state.web3) {
    //return <div>Loading Web3, accounts, and contract...</div>; }
    const { showSignup, showSignin, showPatientForm1, showNav, showData, showTest, showContact,
showContactNav, showPatientRecord, showSendMessage } = this.state;

  return (

    <div className="App">
      <nav>
      <h1>Contract Tracing</h1>
      <Button type="submit" variant="contained" color="primary" onClick={()=> this.operation()}>
Assign Roles </Button>
      <Button type="submit" variant="contained" color="primary" onClick={()=> this.operation1()}>
Sign In </Button>
      </nav>

      {showSignup?
        <div>
        <h1> Sign up Form </h1>
        <br/>
        <form onSubmit= {this.handleSubmit}>
        <div className="dibox">
          <Select
            value={this.state.selectrole}
            onChange={this.updateSelectVal}
            displayEmpty
            variant="outlined"
          >
            <MenuItem value="" disabled>Select Role</MenuItem>
            <MenuItem value="Patient">Patient</MenuItem>
```

```
    <MenuItem value="Contact Tracer">Contact Tracer</MenuItem>
   </Select>
   </div>
   <br/>
   <TextField
    placeholder="Enter Address"
    variant="outlined"
    label="Enter Address"
    value = {this.state.address}
    onChange={event => this.handleChange(event,'address')}
   />
   <br/><br/>
   <Button type="submit" variant="contained" color="primary"> Submit </Button>
  </form>
  </div>
  :null
}

{showSignin?
  <div>
  <h1> Sign In Form </h1>
  <form onSubmit= {this.handleSubmit1}>
   <TextField
    placeholder="Enter Role"
    variant="outlined"
    label="Enter your Role"
    value = {this.state.irole}
    onChange={event => this.handleChange(event,'irole')}
   />
   <br/><br/>
   <Button type="submit" variant="contained" color="primary"> Verify </Button>
  </form>
  </div>
  :null
}

{showPatientForm1?
  <div >
  <h1> Patient Personal Information </h1>
  <form onSubmit= {this.handleSubmit2}>
   <TextField
    placeholder="Enter Name"
    variant="outlined"
    label="Full Name"
    value = {this.state.pname}
    onChange={event => this.handleChange(event,'pname')}
   />
   <br/><br/>
   <TextField
    placeholder="Enter your Age"
    variant="outlined"
```

```
            label="Age"
            value = {this.state.page}
            onChange={event => this.handleChange(event,'page')}
          />
          <br/><br/>
          <TextField
            placeholder="Enter your Phone Number"
            variant="outlined"
            label="Phone Number"
            value = {this.state.pphone}
            onChange={event => this.handleChange(event,'pphone')}
          />
          <br/><br/>
          <TextField
            placeholder="Enter your Address"
            variant="outlined"
            label="Full Address"
            value = {this.state.paddress}
            onChange={event => this.handleChange(event,'paddress')}
          />
          <br/><br/>
          <h2> Symptoms Details </h2>
          <label>
                        
          Please state your symptoms in detail. Do you have fever ? or any other signs and symptoms:
          sore throat, cough, runny nose or <br/> nasal congestion, shortness of breath or difficulty
breathing,
           muscle pain, loss of smell or taste, or diarrhoea.
          <br/><br/>
          <TextField
            className="textarea"
            label="Symptoms"
            multiline rows={5}
            placeholder="Enter Your Symptoms"
            variant="outlined"
            value = {this.state.psymp}
            onChange={event => this.handleChange(event,'psymp')}
          />
          </label>
          <br/><br/>
          <Button type="submit" variant="contained" color="primary"> Submit </Button>
        </form>
        </div>
        :null
      }

      {showNav?
        <div className="container">
          <Button type="submit" variant="contained" color="primary" onClick={()=> this.operation2()}>
View your Details </Button>
          <br/>
```

```
          <Button type="submit" variant="contained" color="primary" disabled={this.state.disabled}
onClick={()=> this.operation3()}> Enter Test Details </Button>
        </div>
        :null
      }

      {showData?
        <div className="yourdata">
          <div className="data"><h5>ID:</h5>     {this.state.response[0]}</div>
          <div className="data"><h5>Name:</h5>     {this.state.response[1]}</div>
          <div className="data"><h5>Age:</h5>     {this.state.response[2]}</div>
          <div className="data"><h5>Phone Number:</h5>    
{this.state.response[4]}</div>
          <div className="data"><h5>Address:</h5>     {this.state.response[5]}</div>
          <div className="data"><h5>Symptoms:</h5>     {this.state.response[6]}</div>
        </div>
        :null
      }

      {showTest?
        <div>
        <h1>Test Details</h1>
        <div className="data">
          <h3>Have you tested Positive? </h3>          No
          <Switch
            color="primary"
            onChange={this.getvalue}
          /> Yes
        </div>
          <br/>
          <h3>Any Further details related to symptoms or test:</h3>
          <TextField
            className="textarea"
            label="More Detail"
            multiline rows={5}
            placeholder="Enter"
            variant="outlined"
            value = {this.state.pdetail}
            onChange={event => this.handleChange(event,'pdetail')}
          />
          <br/><br/>
          <h3>Add a file to IPFS here </h3>
            <form onSubmit= {this.onIPFSSubmit}>
               <input type='file' onChange={this.captureFile} />
               <input type='submit' /> Send it
            </form>
            <p>The IPFS hash is: {this.state.ipfsHash}</p>
            <br/><br/>
          <Button type="submit" variant="contained" color="primary" onClick = {this.handleSubmit3}>
Submit </Button>
        </div>
```

```
        :null
       }

       {showContact?
        <div>
        <h1>Contacts Information</h1>
        <div className="label">
        <h4>***Read the Following Carefully***</h4>
        Have you been close to any person within 1 metre for >15 minutes in passed week
        or had Direct physical contact or had somebody cared for you without using proper
         personal protective equipment ? <br/><br/> Do mention their names and phone numbers to safe
the lives of others.
        </div>
        <form onSubmit= {this.handleSubmit4}>
        <h3>Please Enter the Details of Contact 1:</h3>
        <br/>
        <div className="data">
          <TextField
            placeholder="Enter Contact Name"
            variant="outlined"
            label="Contact Name"
            value = {this.state.cname}
            onChange={event => this.handleChange(event,'cname')}

          />

                      
          <TextField
            placeholder="Enter Contact Phone"
            variant="outlined"
            label="Contact Phone"
            value = {this.state.cphone}
            onChange={event => this.handleChange(event,'cphone')}
          />
        </div>
          <br/><br/>
          <h3>Please Enter the Details of Contact 2:</h3>
          <br/>
          <div className="data">
          <TextField
            placeholder="Enter Contact Name"
            variant="outlined"
            label="Contact Name"
            value = {this.state.ccname}
            onChange={event => this.handleChange(event,'ccname')}
          />

                      
          <TextField
```

```
            placeholder="Enter Contact Phone"
            variant="outlined"
            label="Contact Phone"
            value = {this.state.ccphone}
            onChange={event => this.handleChange(event,'ccphone')}
           />
           </div>
           <br/><br/>
           <Button type="submit" variant="contained" color="primary"> Submit </Button>
          </form>
          </div>
          :null
         }

         {showContactNav?
          <div className="contactcontainer">
          <br/><br/><br/>
            <Button type="submit" variant="contained" color="primary" onClick={()=> this.operation4()}>
View Patient Record </Button>

              &n
bsp; 
            <Button type="submit" variant="contained" color="primary" onClick={()=> this.operation5()}>
Send Message </Button>
          </div>
          :null
        }

        {showPatientRecord?
          <div>
          <h1>Registered Patient's Information</h1>
          <br/>
          <div className="tablecontainer">
          <Table>
            <TableHead>
            <TableRow>
            <TableCell align = "right">Registration ID</TableCell>
            <TableCell align = "right">Patient Name</TableCell>
            <TableCell align = "right">Patient Public Address</TableCell>
            </TableRow>
            </TableHead>
            { this.state.chains.map((record, key) => {
              return(
                   <TableBody key={key}>
                   <TableRow>
                   <TableCell align = "right">{record.id}</TableCell>
                   <TableCell align = "right">{record.na}</TableCell>
                   <TableCell align = "right">{record.name}</TableCell>
                   </TableRow>
                   </TableBody>
              )
```

```
  })
  }
</Table>
</div>
<br/>
<h3>Search for Specific Patient</h3>
<br/>
<form onSubmit= {this.handleSubmit5}>
<TextField
   placeholder="Enter Patient's Public Address"
   variant="outlined"
   label="Patient's Public Address"
   value = {this.state.ppaddr}
   onChange={event => this.handleChange(event,'ppaddr')}
 />
<br/><br/>
<Button type="submit" variant="contained" color="primary"> Find Record </Button>
<br/>
</form>
<h3>Patient's Personal Record</h3>
<br/>
<Table>
  <TableHead>
  <TableRow>
  <TableCell align = "right"><b>Registration ID</b></TableCell>
  <TableCell align = "right"><b>Name</b></TableCell>
  <TableCell align = "right"><b>Age ID</b></TableCell>
  <TableCell align = "right"><b>Public Address</b></TableCell>
  <TableCell align = "right"><b>Phone Number</b></TableCell>
  <TableCell align = "right"><b>Residential Address</b></TableCell>
  <TableCell align = "right"><b>Symptoms</b></TableCell>
  </TableRow>
  </TableHead>
        <TableBody>
        <TableRow>
        <TableCell align = "right">{this.state.result[0]}</TableCell>
        <TableCell align = "right">{this.state.result[1]}</TableCell>
        <TableCell align = "right">{this.state.result[2]}</TableCell>
        <TableCell align = "right">{this.state.result[3]}</TableCell>
        <TableCell align = "right">{this.state.result[4]}</TableCell>
        <TableCell align = "right">{this.state.result[5]}</TableCell>
        <TableCell align = "right">{this.state.result[6]}</TableCell>
        </TableRow>
        </TableBody>
</Table>
<br/>
<h3>Patient's Test Record</h3>
<br/>
<Table>
  <TableHead>
  <TableRow>
```

```
          <TableCell align = "right"><b>Test Result</b></TableCell>
          <TableCell align = "right"><b>More Details</b></TableCell>
          <TableCell align = "right"><b>IPFS Record</b></TableCell>
          </TableRow>
          </TableHead>
                <TableBody>
                <TableRow>
                <TableCell align = "right">{this.state.testrecord[0]}</TableCell>
                <TableCell align = "right">{this.state.testrecord[1]}</TableCell>
                {console.log}
                <TableCell align = "right"> <a href={`https://ipfs.io/ipfs/${this.state.testrecord[2]}`}> text
</a></TableCell>
                </TableRow>
                </TableBody>
        </Table>
        <br/>
        <h3>Patient's Contact Record</h3>
        <br/>
        <Table>
          <TableHead>
          <TableRow>
          <TableCell align = "right"><b>First Contact Name </b></TableCell>
          <TableCell align = "right"><b>First Contact Phone</b></TableCell>
          <TableCell align = "right"><b>Second Contact Name</b></TableCell>
          <TableCell align = "right"><b>Second Contact Phone</b></TableCell>
          </TableRow>
          </TableHead>
                <TableBody>
                <TableRow>
                <TableCell align = "right">{this.state.contactrecord[0]}</TableCell>
                <TableCell align = "right">{this.state.contactrecord[1]}</TableCell>
                <TableCell align = "right">{this.state.contactrecord[2]}</TableCell>
                <TableCell align = "right">{this.state.contactrecord[3]}</TableCell>
                </TableRow>
                </TableBody>
        </Table>
        </div>
        :null
       }

       {showSendMessage?
        <div>
          <h1>Send Message to Patient</h1>
          <form onSubmit= {this.handleSubmit6}>
          <TextField
            className="textarea"
            label="Write your Message"
            multiline rows={5}
            placeholder="Enter"
            variant="outlined"
            value = {this.state.cmsg}
```

```
            onChange={event => this.handleChange(event,'cmsg')}
           />
           <br/><br/>
           <TextField
            placeholder="Enter Public Address"
            variant="outlined"
            label="Public Address"
            value = {this.state.cpaddr}
            onChange={event => this.handleChange(event,'cpaddr')}
           />
           <br/><br/>
           <Button type="submit" variant="contained" color="primary"> Send </Button>
          </form>
          </div>
          :null
         }

    </div>
    );
  }
}

export default App;
```

REFERENCES

[1] E Gorbalenya et al., “The species Severe acute respiratory syndrome related coronavirus: classifying 2019-nCoV and naming it SARS-CoV2,” Nat Microbiol, vol. 5, pp. 536–544, 2020.

[2] “Weekly Operational Update on COVID-19 9 September 2020.” [Online]. Available: https://www.who.int/docs/default-source/coronaviruse/weekly-updates/wou-9-september-2020-cleared-14092020.pdf?sfvrsn=68120013_2&download=true

[3] “The push for a COVID-19 vaccine.” [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/covid-19-vaccines

[4] Neil. Ferguson et al., “Report 9: Impact of nonpharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand,” London: Imperial College London, Tech. Rep., 2020. [Online]. Available: https://www.imperial.ac.uk/media/imperial-college/medicine/sph/ide/gida-fellowships/Imperial-College-COVID19-NPI-modelling-16-03-2020.pdf

[5] M. C. J. Bootsma and N. M. Ferguson, “The effect of public health measures on the 1918 influenza pandemic in US cities,” vol. 104, pp. 7588–7593, 2007.

[6] “COVID-19: These countries are most at risk from falling tourism.” [Online]. Available: https://www.weforum.org/agenda/2020/07/covid-19-coronavirus-usa-united-states-econamy-gdp-decline/

[7] L. Ferretti, C. Wymant, M. Kendall, L. Zhao, A. Nurtay, L. AbelerDorner, M. Parker, D. Bonsall, and C. Fraser, “Quantifying sars- ¨ cov-2 transmission suggests epidemic control with digital contact tracing,” Science, vol. 368, no. 6491, 2020.

[8] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” pp. 1–9, 2008.

[9] D. Vujičić, D. Jagodić, and S. Randić, ‘‘Blockchain technology, bitcoin, and Ethereum: A brief overview,’’ in Proc. 17th Int. Symp. INFOTEHJAHORINA (INFOTEH), pp. 1–6, Mar. 2018.

[10] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, ‘‘An overview of smart contract: Architecture, applications, and future trends,’’ in Proc. IEEE Intell. Vehicles Symp. (IV), pp. 108–113, Jun. 2018.

[11] J. Eberhardt and S. Tai, ‘‘On or off the blockchain? Insights on offchaining computation and data,’’ in Proc. Eur. Conf. Service-Oriented Cloud Comput., pp. 11–45, Oct. 2014.

[12] WHO., “Contact tracing in the context of COVID-19. Interim guidance,” Pediatria i Medycyna Rodzinna, vol. 16, pp. 33–39, 2020.

[13] E. H. Orallo, P. Manzoni, C. T. Calafate, J. C. Cano, “Evaluating How Smartphone Contact Tracing Technology Can Reduce the Spread of Infectious Diseases: The Case of COVID-19,” IEEE Access, vol. 8, pp. 99083-99097, 2020.

[14] COVID-19 National Emergency Response Center, Epidemiology & Case Management Team, Korea Centers for Disease Control & Prevention, “Contact Transmission of COVID-

19 in South Korea: Novel Investigation Techniques for Tracing Contacts," Osong Public Health Res Perspect, vol. 11, pp. 60-63, 2020.

[15] J. Hellewell, S. Abbott, A. Gimma, N. I. Bosse, C. I. Jarvis, T. W. Russell, J. D. Munday, A. J. Kucharski, W. J. Edmunds, S. Funk, R. M. Eggo, Centre for the Mathematical Modelling of Infectious Diseases COVID-19 Working Group, "Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts," Lancet Glob Health, vol. 8, pp. e488-e496, 2020.

[16] R. Raskar, I. Schunemann, R. Barbar, K. Vilcans, J. Gray, P. Vepakomma, S. Kapa, A. Nuzzo, R. Gupta, A. Berke, D. Greenwood, C. Keegan, S. Kanaparti, R. Beaudry, D. Stansbury, B. B. Arcila, R. Kanaparti, V. Pamplona, F. M. Benedetti, A. Clough, R. Das, K. Jain, K. Louisy, G. Nadeau, V. Pamplona, S. Penrod, Y. Rajaee, A. Singh, G. Storm, J. Werner, "Apps Gone Rogue: Maintaining Personal Privacy in an Epidemic," ResearchGate , 2020.

[17] D. Klinkenberg, C. Fraser, H. Heesterbeek, "The Effectiveness of Contact Tracing in Emerging Epidemics," PLoS ONE, vol. 1, 2006.

[18] Braithwaite, T. Callender, M. Bullock, R. W. Aldridge, "Automated and partly automated contact tracing: a systematic review to inform the control of COVID-19," The Lancet Digital Health, vol. 2, pp. e607-e621, 2020.

[19] T. A. Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, T. Alghamdi, "A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations," IEEE Access, vol. 7, pp. 176838-176869, 2019.

[20] F. Casino, T. K. Dasaklis, C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," Telematics and Informatics, vol. 36, pp. 55-81, 2019.

[21] Dannen, "Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners," first ed., Apress, Berkely, CA, USA, 2017.

[22] Ganache, 2019. [Online]. Available: https://truffleframework.com/ganache. [Accessed: 04-Mar-2019].

[23] MetaMask, 2019. [Online]. Available: https://metamask.io/. [Accessed: 04-Mar-2019].

[24] M. Niranjanamurthy, K. Kumar S, A. Saha, and D. D. Chahar, ''Comparative study on performance testing with jmeter,'' Int. J. Adv. Res. Comput. Commun. Eng., vol. 5, no. 2, pp. 70–76, 2016.

[25] InterPlanatery File System (IPFS). Accessed: Feb. 4, 2019. [Online]. Available: https://ipfs.io/.

[26] Levy, "The security behind the NHS contact tracing app," pp. 1–14, 2020. [Online]. Available: https://www.ncsc.gov.uk/blog-post/security-behind-nhs-contact-tracing-app

[27] J. Bay, J. Kek, A. Tan, C. S. Hau, L. Yongquan, J. Tan, and T. A. Quy, "BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders," p. 9, 2020.

[28] Apple Inc. and Google LLC., "Exposure Notification," May 2020.

[29] P. Mozur, R. Zhong, and A. Krolik, “In Coronavirus Fight, China Gives Citizens a Color Code, With Red Flags,” New York, NY, 2020. [Online]. Available: https://www.nytimes.com/2020/03/01/business/chi na-coronavirus-surveillance.html
[30] R. Hutchings “The impact of Covid-19 on the use of digital technology in the NHS”, 2020. [Online]. Available: https://www.nuffieldtrust.org.uk/files/2020-08/the-impact-of-covid-19-on-the-use-of-digital-technology-in-the-nhs-web-2.pdf
[31] Shahnaz, U. Qamar and A. Khalid, “Using blockchain for electronic health records,” IEEE Access, vol. 7, pp. 147782-147795, 2019.
[32] M. Arifeen, A. Mamun, M. Kaiser, M. Mahmud, “Blockchain-enable Contact Tracing for Preserving User Privacy During COVID-19 Outbreak,” Preprints 2020, 2020070502, DOI: 10.20944/preprints202007. 0502.v1.
[33] “List of cities in the United Kingdom.” [Online]. Available: https://en.wikipedia.org/wiki/List_of_cities_in_the_United_Kingdom#List_of_cities
[34] H.R. Hasan, K. Salah, R. Jayaraman, J. Arshad, I. Yaqoob, M. Omar, S. Ellahham, “Blockchain-Based Solution for COVID-19 Digital Medical Passports and Immunity Certificates,” IEEE Access, vol. 8, pp. 222093 - 222108, 2020.
[35] M. S. Sahoo and P. K. Baruah, “HBasechainDB -- A Scalable Blockchain Framework on Hadoop Ecosystem,” in Supercomputing Frontiers, 2018, pp. 18–29.
[36] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, “FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data,” Comput. Struct. Biotechnol. J., vol. 16, pp. 267–278, 2018.
[37] M. G. Kim, A. R. Lee, H. J. Kwon, J. W. Kim, and I. K. Kim, “Sharing Medical Questionnaries based on Blockchain,” Proc. - 2018 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2018, pp. 2767–2769, 2019.
[38] India Goverment, “Aarogya Setu Mobile App,” 2020. [Online]. Available: https://www.mygov.in/aarogya-setu-app/
[39] Department of Health Australia, “The COVIDSafe Application,” 2020. [Online]. Available: https: //www.health.gov.au/sites/default/files/documents/2020/04/covidsaf e-application-privacy-impact-assessment-agency-response.pdf
[40] D.-T. task group, “Decentralized Privacy-Preserving Proximity Tracing,” no. April, p. 33, 2020.
[41] P.-P. e.V. i. Gr, “Pan-European Privacy-Preserving Proximity Tracing,” 2020. [Online]. Available: https://www.pepp-pt.org/content
[42] “SHA-256,” Bitcoin Wiki, 2018. [Online]. Available: https://en.bitcoinwiki.org/wiki/SHA-256.
[43] A. A. Siyal, A. Z. Junejo, M. Zawish, K. Ahmed, A. Khalil, and G. Soursou, “Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives,” Cryptography, vol. 3, no. 1, p. 3, 2019.
[44] M. S. Sahoo and P. K. Baruah, “HBasechainDB -- A Scalable Blockchain Framework on Hadoop Ecosystem,” in Supercomputing Frontiers, 2018, pp. 18–29.

[45] “InterPlanetary File System,” Wikipedia, 2019. [Online]. Available: https://en.wikipedia.org/wiki/InterPlanetary_File_System. [Accessed: 03-Dec-2019].
[46] S. Voshmgir and V. Kalinov, “Blockchain A Beginners guide,” 2017, p. 57.
[47] V. Buterin, “On Public and Private Blockchains,” Ethereum Blogs, 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.
[48] D. Lee Kuo Chen, Handbook of digital currency, 1st Editio. Elsevier, 2015.
[49] “Peer to Peer,” Wikipedia. Wikimedia Inc, 2019.
[50] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017, no. June, pp. 557–564, 2017.
[51] “Mining,” 2018. [Online]. Available: https://en.bitcoin.it/wiki/Mining. [Accessed: 16-Jan-2021].
[52] T. Xue, Y. Yuan, Z. Ahmed, K. Moniz, G. Cao, and C. Wang, “Proof of Contribution : A Modification of Proof of Work to Increase Mining Efficiency,” 2018 IEEE 42nd Annu. Comput. Softw. Appl. Conf., pp. 636–644, 2018.
[53] “Proof of work,” Bitcoin Wiki. 2019.
[54] A. Tar, “Proof of Work, Explained.” [Online]. Available: https://cointelegraph.com/explained/proof-of-work-explained.
[55] “What is Ethereum?,” 2018. [Online]. Available: http://www.ethdocs.org/en/latest/introduction/what-is-ethereum.html#a-next-generation-blockchain.
[56] “Ethereum,” 2020. [Online]. Available: https://www.investopedia.com/terms/e/ethereum.asp.
[57] U. W. Chohan, “Cryptocurrencies : A Brief Thematic Review,” SSRN Electron. J., 2017.
[58] “Account Types, Gas, and Transactions,” 2018. [Online]. Available: http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html#what-is-a-transaction.
[59] P. Kasireddy, “How does Ethereum work, anyway?,” Medium Corporation, 2017. [Online]. Available: https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369.
[60] “What is Ethereum? The Most Comprehensive Gudie Ever,” 2018. [Online]. Available: https://blockgeeks.com/guides/ethereum/.
[61] “Ethereum,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Ethereum.
[62] T. Takenobu, “Ethereum EVM illustrated,” 2018.
[63] “Immutable Object,” Wikipedia, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Immutable_object.
[64] “Immutable collections for JavaScript,” npm, 2021. [Online]. Available: https://www.npmjs.com/package/immutable.
[65] T. Dey, S. Jaiswal, S. Sunderkrishnan, and N. Katre, “A Medical Use Case of Internet of Things and Blockchain,” 2017 Int. Conf. Intell. Sustain. Syst., no. Iciss, pp. 486–491, 2017.

[66] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” no. Draft 3, 2014
[67] “Truffle,” 2021. [Online]. Available: https://www.truffleframework.com/docs/truffle/overview.
[68] “Truffle Boxes,” 2021. [Online]. Available: https://truffleframework.com/boxes.
[69] “React Main Concepts,” Facebook Inc., 2021. [Online]. Available: https://reactjs.org/docs/hello-world.html.
[70] “Ganache,” 2021. [Online]. Available: https://truffleframework.com/ganache. [Accessed: 16-Jan-2021].
[71] “Creating Workspaces,” 2021. [Online]. Available: https://truffleframework.com/docs/ganache/workspaces/creating-workspaces.
[72] “MetaMask,” 2021. [Online]. Available: https://metamask.io/. [Accessed: 16-Jan-2021].
[73] “User (Computing),” Wikipedia, 2021. [Online]. Available: https://en.wikipedia.org/wiki/User_(computing).
[74] “web3.js - Ethereum JavaScript API,” 2021. [Online]. Available: https://web3js.readthedocs.io/en/1.0/. [Accessed: 16-Jan-2021].
[75] “Infura,” 2021. [Online]. Available: https://infura.io/. [Accessed: 16-Jan-2021].
[76] J. Hoerdt, “Current facts and figures about the Corona-Warn-App” , 2021. [Online]. Available: https://www.coronawarn.app/en/blog/2021-06-25-facts-and-figures/
[77] H.R. Hasan and K. Salah, “Proof of Delivery of Digital Assets Using Blockchain and Smart Contracts,” IEEE Access, vol. 6, pp. 65439 - 65448, 2018.
[78] The Story of the DAO—Its History and Consequences. [Online]. Available: https://medium.com/swlh/the-story-of-the-dao-its-history-andconsequences-71e6a8a551ee
[79] M. Madine, K. Salah, R. Jayaraman, Y. Al-Hammadi, J. Arshad, and I. Yaqoob, “appXchain: Application-Level Interoperability for Blockchain Networks,” IEEE Access, vol. 9, pp. 87777 - 87791, 2021.