PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

UNIVERSITÉ KASDI MERBAH OUARGLA

Thesis Submitted in
partial fulfillment of the requirements for the degree of

ACADEMIC MASTER

Title

Integrate and manage data using Blockchain in electronic health records

Examination Committee :

Dr. Djedia . H	**Chair**
Dr. Hamioud . S	**Examinator**
Dr. Akram Boukhamla	**Supervisor**

Presented by:

Marouan Okba

2021 – 2022

Acknowledgement

The first thanks is to Allah Almighty, then to my parents for all their efforts since my birth until these moments. You are everything I love you most in Allah. I am also pleased to extend my thanks to everyone who advised me, guided me, directed me, or contributed to the preparation of this research by connecting me to the required references and sources at any stage of it, and I especially thank my honorable professor, Dr. My thanks are also directed to the administration of the Faculty for their good provision and facilitation of services to students and their assistance in all matters that would give them a comfortable space to study and seek knowledge in safety and order.

Okba Marouan

Abstract

Blockchain have been an interesting research area for a long time and the benefits it provides have been used by a number of various industries Similarly, because to security, privacy, secrecy, and decentralization, blockchain technology has a lot of potential in the healthcare sector.

Nonetheless, EHR systems have issues with data security, integrity, and administration.We examine how blockchain technology may be utilized to alter EHR systems and could be a solution to these problems in this thesis. We provide a methodology for implementing blockchain technology in the healthcare industry for electronic health records.

The goal of our proposed framework is to first integrate blockchain technology for EHR and then to enable safe storage of electronic data for users of the framework by setting granular access controls. This framework provides the EHR system with the advantages of a blockchain-based solution that is scalable, secure, and integrated.

Keywords : Electronic Health Record (EHR), Privacy preservation, Privacy issues of Blockchain-based EHR, Blockchain.

الملخص

لطالما كانت البلوكشين (Blockchain) مجالًا بحثيًا مثيرًا للاهتمام ، وقد استخدم عدد من الصناعات المختلفة الفوائد التي توفرها بالمثل ، نظرًا للأمان والخصوصية والسرية واللامركزية ، تتمتع تقنية البلوكشين (Blockchain) بالكثير من الإمكانات في قطاع الرعاية الصحية.

ومع ذلك ، فإن أنظمة السجلات الصحية الإلكترونية(EHR) لديها مشكلات تتعلق بأمن البيانات ، وسلامتها ، وإدارتها ، ونحن ندرس كيفية استخدام تقنية البلوكشين (Blockchain) لتغيير أنظمة السجلات الصحية الإلكترونية ويمكن أن تكون حلاً لهذه المشكلات في هذه الأطروحة. نحن نقدم منهجية لتطبيق تقنية البلوكشين (Blockchain) في صناعة الرعاية الصحية للسجلات الصحية الإلكترونية.

الهدف من إطارنا المقترح هو دمج تقنية البلوكشين (Blockchain) في السجلات الصحية الإلكترونية (EHR)أولاً ثم تمكين التخزين الآمن للبيانات الإلكترونية لمستخدمي الإطار من خلال وضع ضوابط وصول دقيقة. يوفر هذا الإطار لنظام السجلات الصحية الإلكترونية مزايا الحل القائم على البلوكشين (Blockchain) القابل للتطوير والآمن والمتكامل.

الكلمات المفتاحية : السجلات الصحية الالكترونية (EHR) ,تقنيات حماية الخصوصية , البلوكشين (Blockchain) , مشاكل حماية البيانات في السجلات الصحية إلكترونية .

Summary

Acknowledgment	
Abstract	
List of figures	
الملخص	
Chapiter I : STATE OF THE ART	
1. Introduction	**02**
2. Electronic Health Record Systems	**02**
2.1. Features & Functionalities of EHR Systems	**03**
2.2. Cloud-Based EHR Systems	**03**
2.3. Workflow of Cloud-Based EHR Systems	**04**
2.4. Security Risks in Cloud-Based EHR Systems	**04**
2.5. Security Requirement for EHR Systems	**05**
3. Data Preservation Strategies	**05**
3.1. Encryption	**06**
3.2. Hashing	**07**
3.3. Anonymization	**07**
3.4. Differential Privacy	**08**
4. Blockchain Technology	**08**
4.1. Types of Blockchain	**09**
4.2. Working Phenomenal Of Blockchain	**10**
4.3. Characteristics of Blockchain	**11**
5. Related Work	**12**
6. Conclusion	**14**
Chapiter II : Architecture & operation of the proposed mode	
1. Introduction	**16**

Summary

2. Data Preservation Mechanisms	**16**
2.1. Ethereum	**16**
2.2. Information Transaction	**17**
2.3. Smart contract	**17**
2.4. Ethereum Virtual Machine (EVM)	**18**
2.5. Interplanatery File System (IPFS)	**18**
3. System Design and Architecture	**19**
4. Explain how the algorithm works	**25**
5. Consultation process	**26**
6. Conceptual study of our application	**27**
6.1. Case Diagram	**28**
6.2. Sequence diagrams	**29**
7. Conclusion	**30**
Chapiter III : IMPLEMENTAION & RESULTS	
1. Introduction	**33**
2. Practical components	**33**
2.1. Remix IDE	**33**
2.2. Visual Studio Code	**34**
2.3. Truffle	**34**
2.4. Ganache	**35**
2.5. Node.js	**36**
2.6. React	**37**
2.7. MetaMask	**38**
3. Ethereum Blockchains	**39**
4. Working example for proposed framework	**40**
5. Creation of smart contracts	**41**
6. System interfaces	**42**
6.1. Home page	**42**

Summary

6.2. Admin Page	**42**
6.3. Doctor Page	**44**
6.4. Patient Page	**44**
7. Conclusion	**46**
8. Future Works	**47**
Bibliographical references	**48**

List of figures :

Number	Title	Page
1	Architecture of Cloud-Based Electronic Health Record System	03
2	Process of encryption and decryption using Symmetric and Asymmetric cryptography	07
3	Simplified structure of blocks	10
4	System design of proposed framework	20
5	Folder move scheme	27
6	General Use Case Diagram	28
7	Encryption and storage sequence diagrams of a tracking sheet	29
8	Sequence diagrams downloading and decrypting a medical file	30
9	Remix IDE and Solidity logo	33
10	Visual Studio Code logo	34
11	Truffle logo	34
12	Ganache logo	35
13	The Ganache home page	36
14	Node js logo	36
15	Tool versions	37
16	React Logo	37
17	MetaMask logo	38
18	MetaMask interface	38
19	Ethereum Virtual machine	39
20	Work Flow of DApp	40
21	Define the users in the function	41
22	Add Medical record to block chain	41
23	Home Page	42

List of figures

24	Admin Page	42
25	Add Doctor	43
26	Doctor try to access Admin Function	43
27	Doctor can View Patient Record	44
28	Doctor Can Update Patient Record	45
29	Patient Can View medical info	46

CHAPTER I :

STATE OF THE ART

1. Introduction :

With the advancement of IoT technology and data preservation mechanisms, as well as the introduction of blockchain technology, new doors of improvement for Electronic Health Record (EHR) systems have opened, providing an amazing chance to successfully adapt e-Health systems in many circumstances.

The healthcare sector faced a trend shift towards HER systems that were designed to combine paper-based and electronic medical records (EMR). These systems were used to store clinical notes and laboratory results in its multiple

components [1]. They were proposed to enhance the safety aspect of the patients by preventing errors and increasing information access [2]. The goal of EHR systems was to solve the problems faced by the paper-based healthcare records and to provide an efficient system that would transform the stateof healthcare sector [3].

The EHR systems have been implemented in a number of hospitals around the world due the benefits it provides, mainly the improvement in security and its cost-effectiveness. They are considered a vital part of healthcare sector as it provides much functionality to the healthcare [4]. These functionalities are electronic storage of medical records, patients' appointment management, billing and accounts, and lab tests. They are available in many of the EHR system being used in

the healthcare sector. The basic focus is to provide secure, temper-proof, and shareable medical records across different platforms. Despite the fact that notion behind usage of HER systems in the hospitals or healthcare was to improve the

quality of healthcare, these systems faced certain problemsand didn't meet the expectations associated with them [3].

A study was conducted in Finland to find the experiences of nursing staff with the EHR, it was concluded that HER systems faced the problems related to them being unreliable and having a poor state of user-friendliness [5].

New privacy threats and hacking tactics, on the other hand, have emerged. To safeguard patient information within the EHR System and create a new danger to patient privacy and e-Health systems, innovative development and integrated design are required. We described some of the features, functions, and security needs of EHR Systems in this chapter. We also discussed some of the most often utilized data preservation techniques.

We investigated blockchain technology in depth, identifying features, difficulties, and solutions to enhance EHR systems.

2. Electronic Health Record Systems :

An electronic health record is a digital copy of a patient's data and information (EHR). EHR is a critical part of the medical sector since it provides a real-time database of patient records and makes them easily and rapidly accessible to authorized users such as physicians and administrators. EHR Systems (Electronic Health Record Systems) are another term for systems that go beyond just documenting a patient's medical information and treatment history. EHR Systems may provide a wide range of services and features to users by using various approaches and functionalities on patient data.

2.1. Features & Functionalities of EHR Systems :

- The workflow of providers can be automated and optimized.
- Share health information with approved parties, such as health organizations or labs, for a variety of medical purposes.
- Provide a safe system that prevents the loss or manipulation of patient data and is resistant to all forms of attacks.
- The medical history, diagnosis, prescriptions, treatment plans, vaccination dates, allergies, radiological pictures, and laboratory and test results of a patient are all contained in one file.

2.2. Cloud-Based EHR Systems :

Patient data is managed through a variety of electronic health record systems; cloud-based EHR systems, which use Cloud Computing Technology to manage patient data, are one of the most trustworthy methods to handle sensitive data. Figure 1 depicts the architecture of a cloud-based electronic health record system. Cloud-based EHR solutions provide a number of advantages and features to help with patient data management.

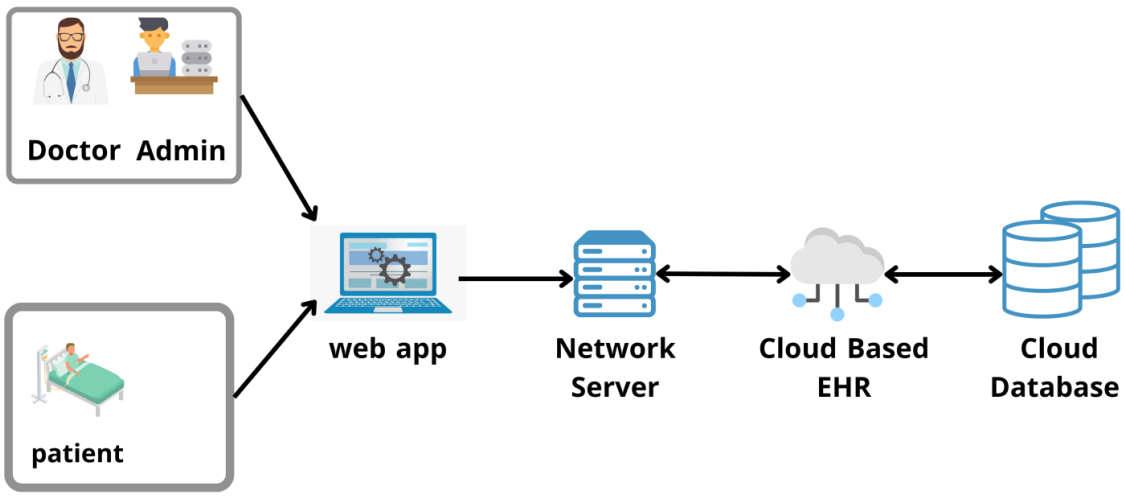

Figure 1 : Architecture of Cloud-Based Electronic Health Record System.

2.3. Workflow of Cloud-Based EHR Systems:

Patient records in cloud-based EHR systems follow a specified path via the network from the patient's body sensors or the user's client computers (doctors, administrators) back to the cloud-based EHR's database. To begin, there are two primary methods for committing patient information to the cloud database: users can input data from their client computers, or data can be acquired via the patient's body sensors. The recordings are then transferred to a network server, which organizes them into datasets and uses specialized privacy preservation measures to assist safeguard the patient's information.

Data leaking is prevented by storing information inside the dataset records. Finally, the network server transfers the data to the cloud-based EHR, which saves it in its database. The patient records in the dataset are structured and anonymised, which offers an infrastructure for numerous medical research activities and aids researchers in conducting studies and analyzing the patient record.

2.4. Security Risks in Cloud-Based EHR Systems:

Despite the numerous advantages of cloud-based EHR management solutions, security remains a major concern. Abuse, leakage, loss, or theft of EHR in cloud-based management systems are all possibilities. Intruders can, for example, erase or tamper with EHRs to interfere with treatments that provide insurance companies benefits or hide medical malpractices. Health insurance and electronic health records are inextricably linked. Dishonest health insurance companies may engage hackers to destroy or tamper with patients' electronic health records (EHRs) in order to show the existence of pre-existing diseases. Medical

malpractice cases are common for a variety of reasons, including misdiagnosis and delayed diagnosis. Due of the aforementioned difficulties, patients are frequently unable to show medical negligence. Patients, on the other hand, alter medical records to obtain financial benefits despite having pre-existing medical issues. Several cryptographic countermeasures are proposed to ensure the security of EHRs. Unfortunately, due to the centralized nature of cloud-based systems, security vulnerabilities remain a major concern.

2.5. Security Requirement for EHR Systems:

- **Data privacy :** The security of patient data is a major concern. As a result, the EHR System should make a concerted effort to safeguard these bits of data from assaults, which may be accomplished by employing various types of privacy preservation methods.
- **Immutability :** The EHR System should prohibit any sort of health information tampering and assure patient data integrity. As a result, health information that is transmitted should be a true depiction of the original data, with no changes or additions.
- **Anonymity :** Patient genuine identity must be properly secured from the public owing to the intimate and private nature of this information, hence patient anonymity should be a top concern for the EHR System.
- **Control of access :** Only authorized users should have access to patient data, which should be maintained away from those who should not have access. As a result, the EHR System must manage who has access to patient data in order to prevent unauthorized individuals from doing so.

3. Data Preservation Strategies :

As discussed in the preceding section, sensitive data and information must be protected from a variety of threats. Any leak of this type of information may be extremely damaging to the person who owns it. As a result, any data management system should have some sort of privacy preservation mechanism to secure data from intruders. In this part, we'll go through some of the most often used data management systems' privacy preservation features.

3.1. Encryption :

Encryption is one of the most widely utilized methods for providing data privacy and security while also hiding it from outsiders. There are several forms of encryption, one of which is Symmetric Encryption, in which all communication partners use the same secret key for data encryption and decryption. Asymmetric encryption, often known as public key encryption, is another kind of encryption (PKE), There are two sorts of keys in it: one is a public key, which is used to encrypt data before transmitting it and should be available to anybody, and the other is a private key, which is used to decode received data and should be kept private. Figure 2 is an example of the Symmetric and Asymmetric Encryption method. It's worth noting that information before encryption is referred to as PlainText, while information after encryption is referred to as CypherText.

The incredible power of encryption and decryption protects and secures data . However, many additional sophisticated encryption methods are being considered by researchers in order to give data with greater privacy and security.

When estimating the efficacy of an encryption scheme, three things should be considered: The amount of processing power and resources placed into the process of running the encryption method is known as computational load. Key size (public/private): The amount of bits needed to produce the encryption keys. The

size of the band corresponds to the amount of bits necessary to send a message once it has been encoded or signed.

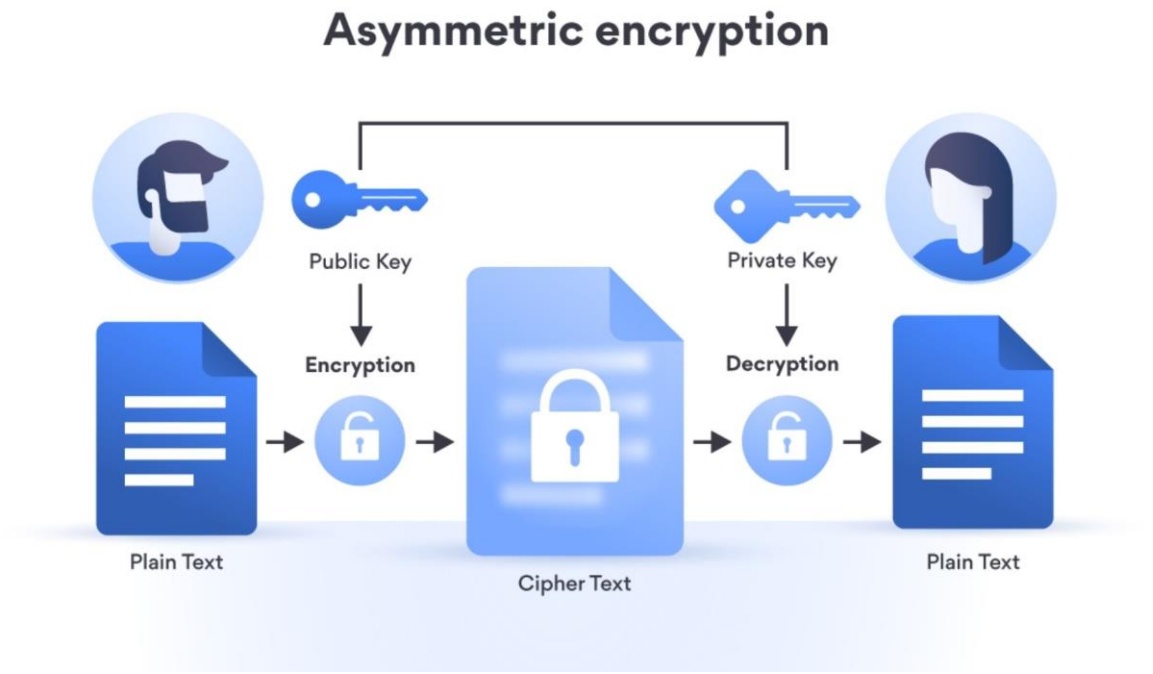

Figure 2 : Process of encryption and decryption using Symmetric and Asymmetric cryptography. [24]

3.2. Hashing :

Hashing is the process of converting a string of characters into a generally fixed-length value or key that functions as a unique signature for that string. When a hashing function is given a document or a file as an input, it produces a hash output that reflects the original. The input is the message to be hashed, the hash function is the algorithm that conducts the hashing, and the output is the hash value. Although several formulae can be used to hash a message, a cryptographic hash function must possess specific characteristics in order to be considered useful.

3.3. Anonymization :

Anonymization is a well-known approach for ensuring the anonymity of the person's genuine identity behind the data by deleting personally identifiable information from the data. Researchers are debating a variety of sophisticated ways for providing secure levels of anonymization [6]**.**

One is k-anonymity, which can be defined as a guarantee of "hiding in the crowd." If each person is a member of a bigger group, any of the entries in that group might belong to the same person. To achieve k-anonymity, the dataset must contain at least k people who share the set of qualities that may be used to identify each individual.

3.4. Differential Privacy :

C. Dwork was the first to propose differential privacy, which entails introducing noise to query evaluation in order to safeguard the database. Differential privacy is a valuable privacy preservation strategy that maintains data's utility while preserving its private. As a result, researchers are developing more sophisticated differential privacy strategies to safeguard data from leaking, as well as discussing the application of this technology in other areas of our lives, such as healthcare, where differential privacy can be an effective means of protection [11] .

4. Blockchain Technology :

Blockchain is a distributed digital record of transactions that is shared across all computers in a network. It is made up of a series of blocks that are linked together in a sequential sequence by a hashing algorithm. Blockchain was initially presented in 2008, with the launch of the world's first cryptocurrency, BitCoin [7]. Blockchain is a decentralized peer-to-peer network that performs various

transaction exchange processes without the involvement of a centralized third party. The huge success of BitCoin boosted the popularity of the technology that underpins it, prompting experts to dig further into the many functionality and components of Blockchain.

4.1. Types of Blockchain:

Blockchain can be split up into different types based on its operating system and its way of implementation in the network [6].

4.1.1. Private Blockchain :

A private or acceptable blockchain is a form of blockchain designed to govern the activities of a single business or group of people. A private blockchain network's membership is limited and regulated. Any new user cannot join the network unless he or she has been granted permission to do so. The consensus protocol's execution and the shared ledger's maintenance.

Mining is also managed by preset nodes from the network because there is often no native token or incentives to incentivize people to join and undertake mining. These qualities of private blockchain make it safe to use in some industries, such as the financial industry and healthcare [8]**.**

4.1.2. Public Blockchain :

A public blockchain is a transparent, decentralized peer-to-peer blockchain that allows anybody to join the network. Nodes can have a copy of the distributed ledger in public blockchains to examine and validate all transactions. They may also participate in the mining process by gathering transaction data and utilizing a consensus method to verify its correctness, producing blocks, and adding these new blocks to the blockchain by completing complex mathematical calculations. Nodes

are rewarded when a block they created is successfully uploaded to the blockchain. Because anybody may participate in the mining mechanism, and miners are able to construct and broadcast blocks, the blockchain network must devise a means of verifying the blocks' integrity. Proof-of-Work (PoW) is the most popular consensus mechanism in the public Blockchain. This mechanism works on the concept of mining power, which requires the blockchain to trust whichever block has the most computational work put into it. This mechanism requires miners to compete on who will create blocks first, which makes it more difficult for intruders to create new blocks on the blockchain because they must have 51 percent mining power to control the blockchase.

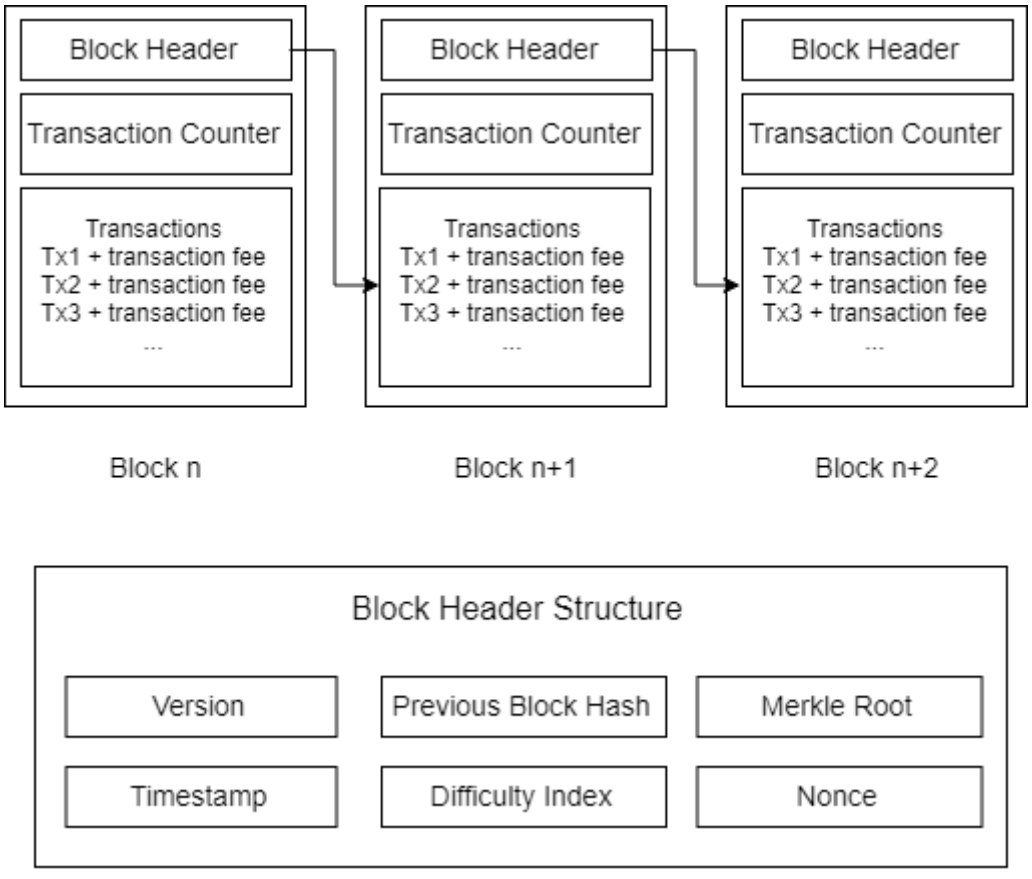

Figure 3 : Simplified structure of blocks.

4.2. Working Phenomenal Of Blockchain :

Blockchain is a peer-to-peer network in which each node may exchange data using a broadcasting protocol that distributes the data throughout the network. Miners monitor the network for transactions, then generate blocks and broadcast them. To be added to the blockchain, a block must have computing power invested in it. Each block in the blockchain includes transactional information as well as its previous hash and other information such as creation time; once confirmed and added to the blockchain, a block cannot be edited or erased. The genesis block is the first block ever produced on the blockchain.

Blocks are linked in chronological order via a hash technique and are visible and transparent to all network users.

Instead than involving a third party to supervise transaction activities, blockchain relies on a consensus process to validate transactions. There are other consensus techniques available, including Proof of Stake (PoS), Proof of Importance (PoI), and Measure of Trust (MoT), but the most common is Proof of Work (PoW), which is the first consensus mechanism in Blockchain history[6].

4.3. Characteristics of Blockchain :

Blockchain offers a number of benefits and qualities that distinguish it as a valuable technology for a variety of businesses and as a solution to a variety of issues. We'll go through some of the finest features of blockchain in this part.

- **Identity backtracking:** Data identification in blockchain may be traced back to its source using a unique identifier, which can be useful, especially when identifying the source of information.
- **Security:** The data is organized into blocks in the blockchain. To ensure security, each block contains a transaction or a bundle of transactions, which can be encrypted using various layers of data preservation strategies.

- **Decentralized nature:** Blockchain's decentralized structure eliminates the idea of network trust, thus no one has to know or trust anybody else. In the form of a distributed ledger, each member of the network owns a copy of the same data. If a member's ledger is tampered with or corrupted in any way, the majority of the network's members will reject it. This approach eliminates the need for a centralized third-party intervention, which is beneficial in many industries where trust is a major concern.
- **Immutability:** While anybody in the blockchain may examine and verify all transactions, no one can edit or remove any transaction or block, fostering trust in the data contained in the blockchain and making it a trustworthy information ledger.

5. Related Work :

We will explore several EHR applications provided in previous work in this section. Showcase and assess the methods and tactics employed in these applications; in this part, we'll present background information on several existing approaches to EHR system security.

A few programs that are used to give security and protection to EHR systems have influenced our work. Despite current solutions, privacy concerns are a fundamental roadblock to wider adoption of public clouds throughout the world. Because the information must be distributed to a large and possibly anonymous group of recipients, and sensitive data is dangerous to outsource to the cloud, there is a growing need to research data anonymization strategies used in this sector. The

[21] authors introduce the BSPP protocol to safeguard data in a consortium blockchain-based e-health system in The authors stated that patient identity leaking might have negative consequences and so stressed the need of maintaining one's original identity.

The authors employed the technique of anonymization to shield their identities by adopting the notion of pseudo identities as evidence for compliance verification. Only the appointed doctor may check pseudo identities of patients, and an adversary cannot track the genuine identities, according to this technique. Furthermore, the provided technique provides searchable confidentiality, allowing hospital employees to search patient data without inferring personal information. Another noteworthy feature of the article is its unique consensus technique, known as "proof-of-conformance," in which patients register for a certain doctor in the hospital, and the doctor produces patient PHI records, encrypts keys, and protects them with pseudo identities.

Another anonymization system is provided in [22], in which the authors employed common secret keys in conjunction with a PoW consensus process to produce anonymous ID and protect blockchain data from Sybil assaults. To safeguard the identity and whereabouts of patients, the authors adopted the notion of bilinear pairing. They went one step farther by preventing location leakage during patient communication and message transfer. The suggested model improved message secrecy and safeguarded e-health hospital data from tracking threats.

Moving on to the following scenario, the authors in [23] offer an architecture for a tamper-proof electronic health record (EHR) management system based on blockchain technology. They Introduce the notion of a blockchain handshaker, which acts as a wrapper layer integrated mechanism that manages communication between an existing cloud-based EHR management system and the public blockchain network. The authors also employed a public-key cryptography-based anonymization process to construct an anonymized transaction from the user-submitted medical data (doctor, administrator). This suggested protocol protects the

system from severe assaults while also improving the confidentiality of patient information contained inside the transaction.

On the other hand, this suggested system is not flawless and contains flaws. One of these flaws is that the proposed design does not prioritize the security of patient information, instead focusing on the system's structure,which might lead to data leakage. Furthermore, the authors failed to solve the issue of data retrieval, leaving the design unfinished.

6. Conclusion :

This chapter discussed the many features and components of EHR systems before finishing with the privacy and security requirements for their deployment. We also spoke about the various technologies, methodologies, and tactics that may aid us in laying a solid basis for an integrated EHR system.

CHAPTER II : IMPLEMENTAION & RESULTS

CHPITER II

1. Introduction :

We now have a theoretical basis that might assist us propose our integrated system after going through introductions and definitions of the various technologies and methodologies that we will require in our study. This chapter begins by outlining the key principles that will be required in our integrated system. Following that, we explain each of the components of our proposed Blockchain integrated cloud-based EHR System in depth. In addition, we cover the workflow of our integrated system, as well as the many components engaged in the process. We'll go through Data Preservation Ethereum and IPFS being the most prominent and important for implementation of this framework are also discussed in the following section .

2. Data Preservation Mechanisms :

This section formally describes the preliminaries used in proposed framework. It describes the software platform used for development of this framework and its advantages.

Ethereum and IPFS being the most prominent and important for implementation of this framework are also discussed in the following section.

2.1. Ethereum :

Ethereum is a distributed blockchain network that uses the idea of blockchain that was previously used in the popular crypto currency Bitcoin [7]. Ethereum was formally introduced in year 2015 and 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 [9].This crypto currency can be used for sharing it between accounts connected on Ethereum blockchain [10]. 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.

2.2. Information Transaction :

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 block chain network. An Ethereum transaction contains following elements [11] :

- **From _** message sender, having a 20-bytes address.
- **To _** message recipient, also having a 20-bytes address.
- **Value** - the fund amount (wei) transferred from senderto recipient.
- **Data (optional) _** contains the message that is being sent to the recipient.
- **Gas** _ For every transaction on the Ethereum block chain 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.

- **Gas Price:** that fee the transaction sender is willing to pay for gas
- **Gas Limit:** maximum gas that could be paid for this transaction.

2.3. Smart contract :

Smart contract is a piece of code that may be used to carry out any action on the block chain. When users send transactions [7], this piece of code is run. They operate directly on the block chain, making them impenetrable to modification.

Smart contracts are written in the Solidity programming language and may be used to implement any operation on the block chain that a programmer desires. After implementing the necessary procedures, programmers can compile them using EVM byte code, which will be discussed in the next section.

And after compiling them it could be executed and deployed on the Ethereum block chain [8]. The programming language of JavaScript and Python are encapsulated with the Solidity language provided by Ethereum to write code in smart contracts.

2.4. Ethereum Virtual Machine (EVM) :

The programmable blockchain is one of the most important features of the Ethereum platform. It gives users the option of creating their own Ethereum-based applications. Distributed Apps are the applications created with this platform (DApps). They are made up of a variety of protocols that have been packed together to form a DApp platform. These DApps have smart contracts, which contain code specified by the user to fulfill a specific application job. The Ethereum Virtual Machine (EVM) is used to deploy and run that code [11], [14]. As a result, the smart contract-based apps are really running on the EVM.

2.5. Interplanatery File System (IPFS) :

IPFS is a data-storage technology that leverages a peer-to-peer network. It enables safe data storage since data stored on IPFS is encrypted and shielded from tampering. It employs a cryptographic identification to safeguard data from tampering, as any effort to alter data saved on IPFS would require altering the identifier. Every file on IPFS has a cryptographically generated hash value. It's one-of-a-kind, and it's used to identify IPFS data files [10].

IPFS protocol is a good option for storing vital and sensitive data because of its safe storage technique. The resulting cryptographic hash might be saved on the decentralized application, reducing the time spent on the block chain computing.

The IPFS protocol uses a peer-to-peer (P2P) network, which comprises a data structure called an IPFS object, which contains data and links. The data is unstructured binary data, and the connection is an array. The IPFS protocol functions as follows [16]:

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

3. System Design and Architecture:

System design is the most significant and critical component of any framework since it is used to construct the system from its theoretical foundation. This section contains the modules, architecture, and numerous parts that come together to form the structure of the entire system. As previously stated, the goal of this suggested architecture is to develop a decentralized system for electronic health data that is temper-proof, safe, and confidential.

The suggested framework or system is made up of three elements or modules, as shown in Figure 5. Our system would continue to function if these modules were integrated. These things or modules have additional notions to grasp, which are described below.

Patients, doctors, administrative employees, and nursing staff are all possible users in the proposed framework. They were granted granular access to the system since they should have varied levels of authority on it.

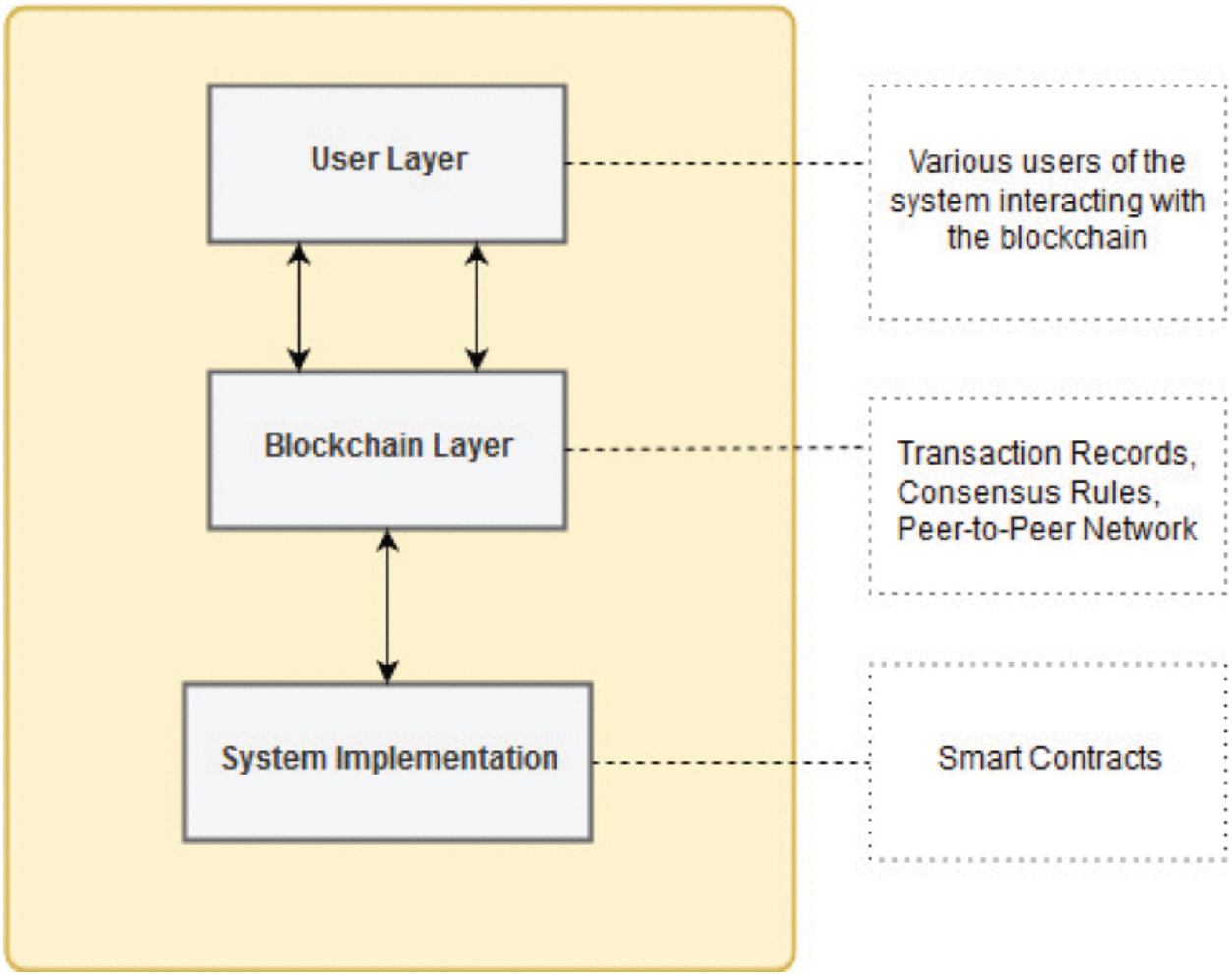

Figure 4 : System design of proposed framework**.**

1) User Layer :

A system user is defined as someone who makes good use of the system's resources. On the system, a user can be identified by his numerous responsibilities and attributes.

The users of this system could be patients, doctors and administrative staff etc. These users' main job would be to interface with the system and conduct simple operations including creating, reading, updating, and deleting medical records. Users would utilize a browser to access the system's capabilities, which we refer

to as a DApp browser since it contains the DApp's GUI (Graphical User Interface), i.e., our proposed system framework.

The graphical user interface (GUI) contains all of the functionality that a given user can utilize. This GUI might be used by the user to communicate with the other layer of the system, the blockchain layer, depending on their job.

2) Blockchain Layer :

The blockchain layer is the next tier in the system; this layer contains the code or method for user interaction with the DApp that is running on the blockchain. There are three items in this layer. They are as follows:

- **Blockchain Assets:**

The method by which an external user can edit the state of a record or information kept on the Ethereum blockchain network is referred to as a transaction on the Ethereum blockchain. The Ethereum blockchain treats these transactions as assets since they are bits of data that users may distribute to other users or save for later use.

- **Governance Rules:**

For transactions to be completed and calculated, blockchain technology follows specific consensus norms. To do this, various consensus techniques are required to keep the blockchain temper-proof and safe.

The Proof of Work (PoW) consensus mechanism is used on the Ethereum blockchain to ensure that the governance of the blockchain is maintained in a trustworthy way, which is done through permission from all of the trusted nodes connected to the blockchain network.

- **Network:**

The peer-to-peer network is used by the Ethereum blockchain. All of the nodes in this network are connected as peers. No node acts as the network's central node, managing all of the network's operations. This network was chosen since the goal was to develop a distributed platform rather than a centralized one. As a result, the greatest decision this technology could have made was to use a network in which all linked nodes had equal status and rights.

3) System Implementation:

As previously stated in the preceding sections, the system was built utilizing Ethereum and its dependencies.

This section delves further into system implementation to have a better understanding of the system's many functions.

4) Smart Contracts:

As explained earlier, smart contracts are an important partof DApps as they are used for performing basic operations. Following contracts are included in this framework:

- Patient Records
- Roles

These contracts are used to grant users access to the DApp and to conduct CRUD operations on patient records. The Patient Recordssmart contract is just for the purpose of implementing the recommended frameworks' capabilities.

It executes CRUD activities as well as denying roles for these functions' access.

The above-mentioned second contract, Roles, is a pre-existing smart contract from the OpenZeppelin smart contract library. This library offers a number of smart contracts that perform a variety of functions and may be used to create your own smart contracts. The benefit of adopting this library was to take use of the features it offers, such as tested and community approved code. The Asset library, which is a sub-component of the OpenZeppelin library, contains the Rolessmart contract. Although the asset library has different contracts for denying access rules, the roles library provides a granular role denition method, which was the primary reason for choosing this smart contract.

The following is the algorithm for rejecting the Patient Records smart contract. It specifies all of the actions that are carried out within it, as well as the numerous circumstances that are linked with them. It also illustrates how the roles for allowing access to a certain functionality are maintained.

Algorithm Smart Contract for Patient Records

Assign Roles :

function Define 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 == doctor) **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 == doctor || patient) **then**

if (patient id) == true **then**

retrieve data from specified patient (id)

return (patient record)

to the account that requested the retrieve operation

else Abort session

end if

end if

end function

Update Data :

function Update Patient Record (*contains variables to update data*)

if (msg.sender == doctor) **then**

if(id == patient id && name == patient name) **then**

update data to particular

patient's record

return success

else return fail

end if

else Abort session

end if

end function

Delete Data:

```
function Delete Patient Record ( patient id )
    if (msg.sender == doctor ) then
        if ( id == patient id ) then
            delete particular patient's record
            return success
        else return fail
        end if
    else Abort session
    end if
end function
```

4. Explain how the algorithm works :

The Algorithm describes how the smart contract for patient records works. This algorithm includes various functions for denying roles, adding, viewing, updating, and deleting records.

The administrator and other users of the system employ these functions. The administrator performs the first function of Algorithm define roles, which comprises two variables: new role and new account, which are used to add new roles and accounts to the role mapping list. Later on, this list would be used to access the roles of the system's users.

The second function is to add a patient record, which is done by the doctor once the administrator has assigned them this job via the dene roles function. This function also ensures that the job is being carried out by the doctor's account's verified public address and not by a third party.

They do this by using the'msg.sender' keyword, which is used in Ethereum's programming language, Solidity, to identify the user's address. After the doctor has

completed the validity check, he or she can add the patient's records and then save the function. View patient records is the third function, and it requires the patient id to be given as a variable. The system would use this id to seek for the patient's data and then return those records to the account that had requested them. Validation for the designated roles of patient or doctor is also included in this function.

Only the patient and the doctor would have access to the records. The fourth function is update patient records, which is used to make any modifications to the patient's stored records. To guarantee that only authenticated users have access to this function, the validation procedure is repeated. The last function of Algorithm is to remove patient records, which, as the name implies, is used to erase a specific patient's information.

This method takes the patient's unique id as input and deletes those records after confirming that the doctor is the one doing the function. This role-based access would ensure that no unauthorized users have access to these functions, and that only the system's authenticated users have access to them.

5. Consultation process :

In Figure 7, we will explain how data flows through our application when requesting to read the folder

1. User requests access to a folder

2. The application asks a blockchain node for user attributes and folder access policy as well as the location of the encrypted folder stored in the cloud.

3. The node responds to the request from the application

4. The application sends the user attributes to generate the secret key

5. The trusted authority sends the secret key

6. Application requests the folder from the cloud

7. The Cloud responds with the requested folder which is encrypted.

8. Application decrypts folder with secret key and send to user. Note here that the decryption succeeds only if the user has the right to access the folder, otherwise an error will be displayed.

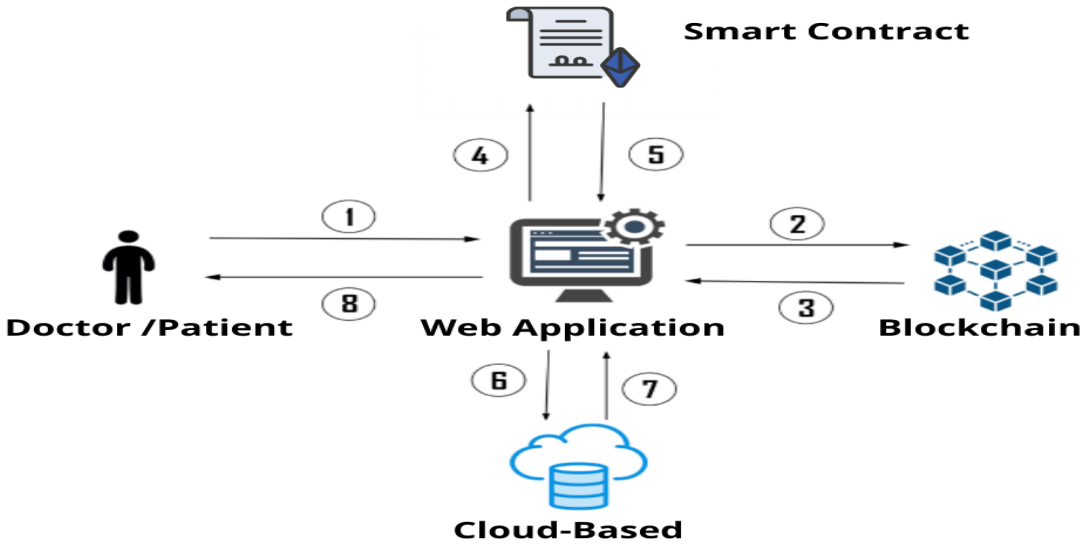

Figure 5 : Folder move scheme.

6. Conceptual study of our application :

To implement our solution, we will develop an application that must meet the following functional needs:

- Data attribute-based encryption/decryption.

- Data storage in the cloud.

- Download data from the cloud.

- Data sharing between users

6.1. Case Diagram :

Case diagram is used to represent the needs of the users in relation to the system used. In our application, we have the following actors:

- Doctor: the individual who can access the medical records of patients.
- Patient: the individual who can access their medical file.
- Admin: the entity responsible for managing user attributes and the keys.

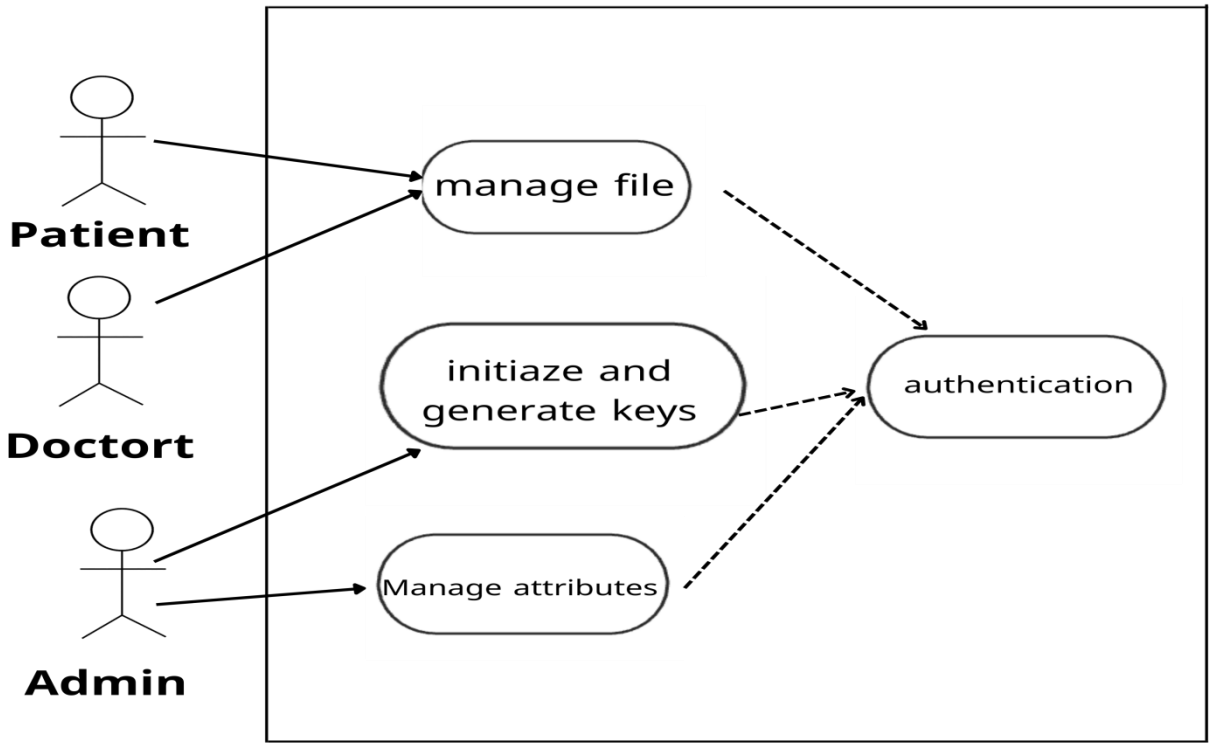

Figure 6 : General Use Case Diagram.

6.2. Sequence diagrams :

In this section, we will try to explain the operation of some main functions of our application: Management of user attributes, Encryption and Storage and Download and decryption.

6.2.1. Encryption and Storage:

For security reasons, medical data must be encrypted using the encryption technique. The steps required to achieve this are described in the following sequence diagram:

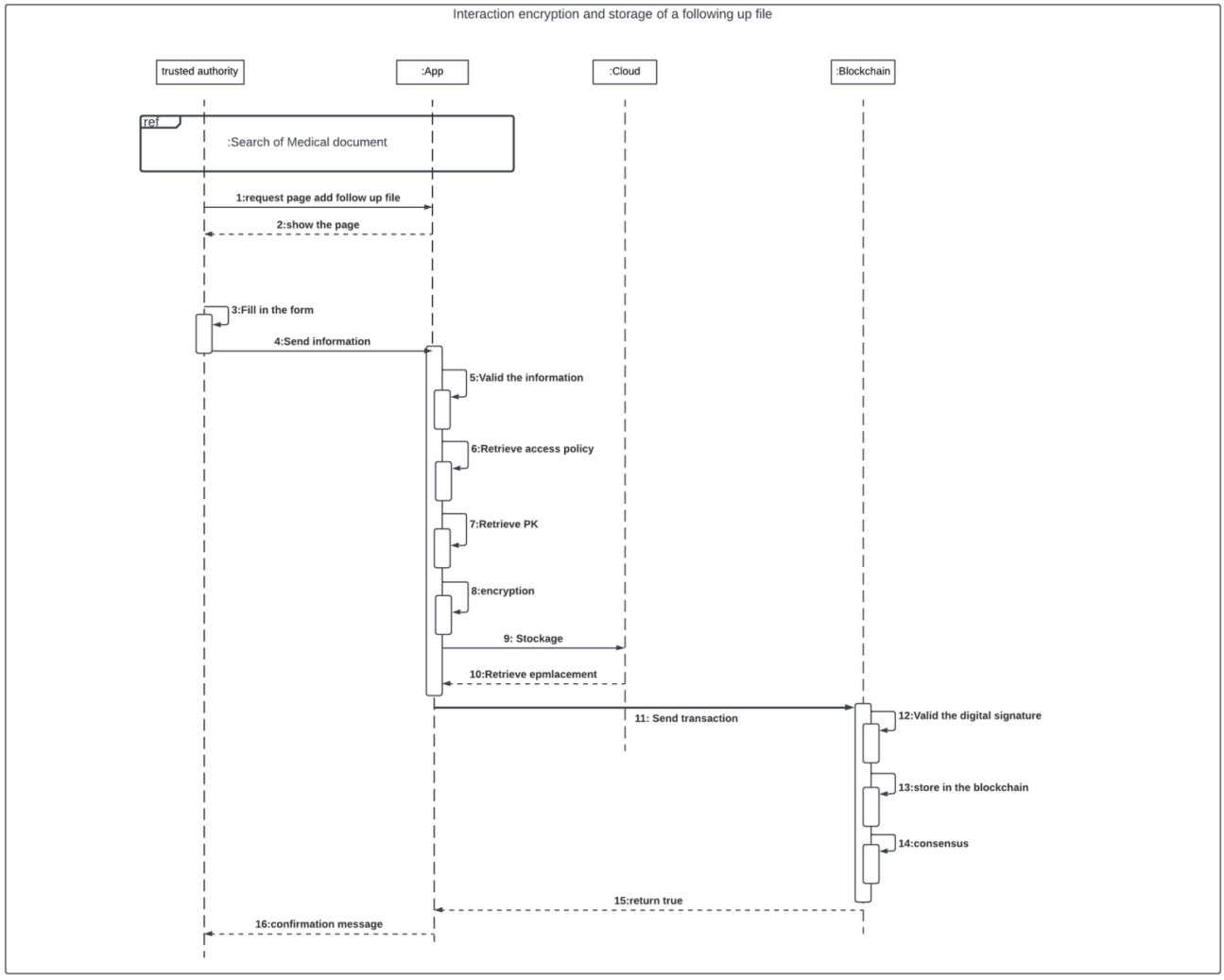

Figure 7 : Encryption and storage sequence diagrams of a tracking sheet.

6.2.2. Download and Decryption :

As explained above, medical data is encrypted and then stored. So to be able to recover these in clear, we must find the folder and have the corresponding attributes to decrypt and download it. The steps required to achieve this are described in the following sequence diagram:

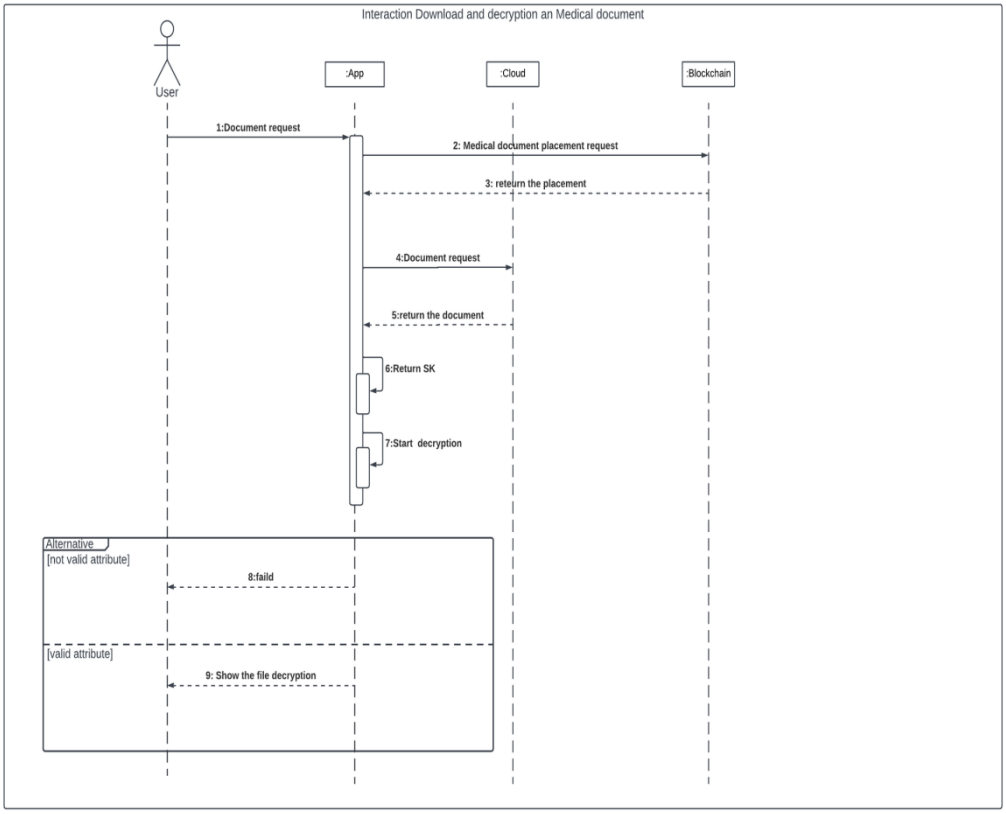

Figure 8 : Sequence diagrams downloading and decrypting a medical file.

7. Conclusion :

In this chapter, we proposed our integrated system architecture of Cloud-based Electronic Health Record, describing each component in detail and going through the different concepts and techniques used in our proposed system. Furthermore, we discuss our system's workflow and go over the lifecycle of patient records within the system, which gave us a precise and accurate understanding of the

different parts of our system. This understanding makes it easy for us to build our integrated system.

CHAPTER III :

IMPLEMENTAION & RESULTS

1. Introduction :

We are getting closer to achieving our ultimate aim, which is to turn our integrated EHR management system into an implemented prototype, after proposing our integrated EHR management system and emphasizing and identifying each component's function in the system in the previous chapter. We'll show you our working prototype of a cloud-based EHR administration system in this chapter. First, we go over each of our prototype's possible implementation possibilities, detailing each one in depth. Then, using images, we display our prototype component and describe the purpose of each component in our system.

2. Practical components :

2.1. Remix IDE :

Is a web application that can be used to write, debug and deploy Ethereum smart contracts. (https://remix.ethereum.org). To write code in Solidity and then deploy it to a blockchain.

Figure 9 : Remix IDE and Solidity logo.

2.2. Visual Studio Code :

Is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Node.js, and C++. It is cross-platform, open source and free.

Figure 10 : Visual Studio Code logo.

2.3. Truffle :

Truffle framework [18] which provides a suite of tools for developing Ethereum smart contacts with the Solidity [19] programming language. We return to this tool in the next section.

Figure 11 : Truffle logo

- **Pour installer truffle sous windows :**

```
npm install -g truffle
```

2.4. Ganache :

Is a personal blockchain for the rapid development of Ethereum and Corda distributed applications. You can use Ganache throughout the development cycle; allowing you to develop, deploy and test your distributed applications in a safe and deterministic environment.

Ganache UI is a desktop application supporting Ethereum and Corda technologies.

Figure 12: Ganache logo.

After installation you should see the Ganache home page :

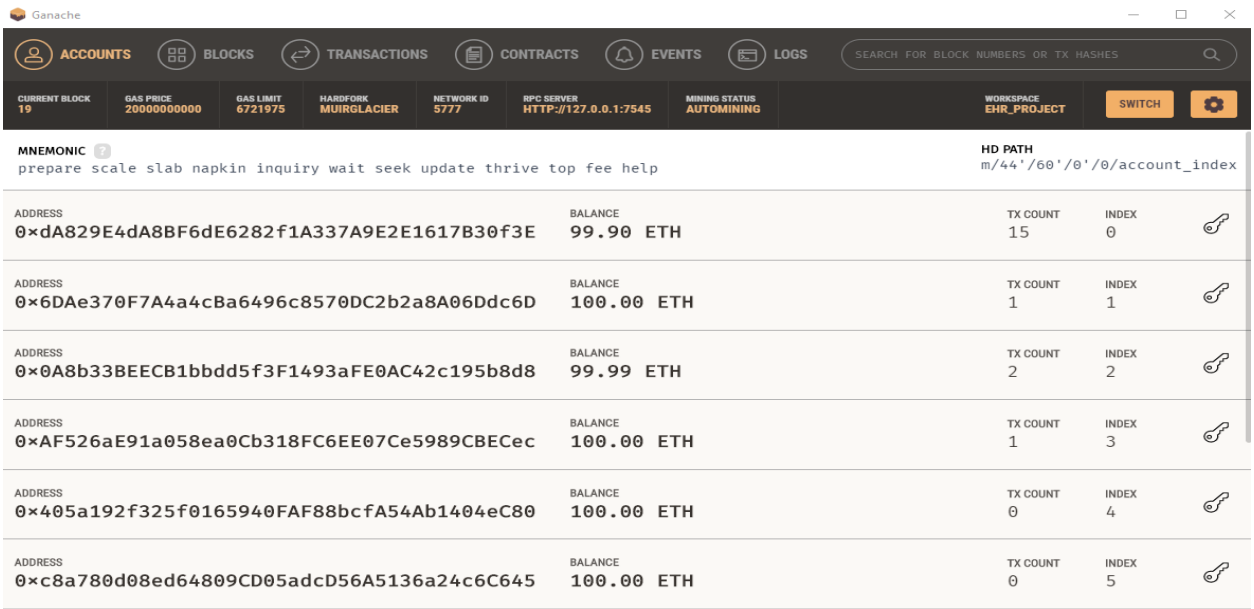

Figure 13 : The Ganache home page.

2.5. Node.js :

Node JS [17] which is the Javascript framework that we will use to develop our application on the server, responsible for executing Web3.JS, Truflle and CPABE commands and to host our application on the nodes [18].

Figure 14 : Node js logo.

The first dependency we will need is Node Package Manager (NPM) and Interplanetary File System (IPFS), which comes with Node.js.

NPM is the package manager for the Node JavaScript platform. It puts modules in place for the node to find and intelligently handles dependency conflicts.

IPFS is a distributed system for storing and accessing files, websites, applications, and data.

After installing nodejs, npm and ipfs we use command prompt for, or even the versions.

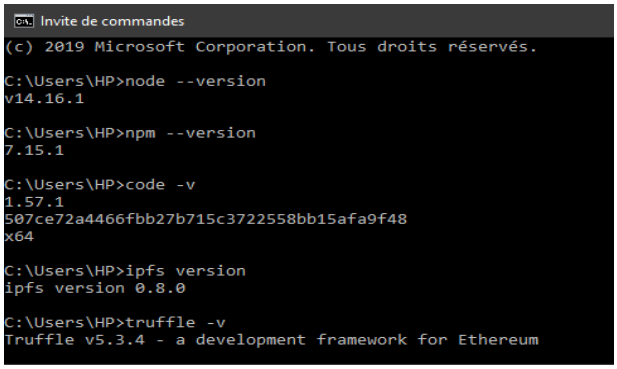

Figure 15 : Tool versions.

2.6. React :

Is a JavaScript library for creating interactive user interfaces.

It helps developers define interfaces such as functions and procedures.

Figure 16 : React Logo.

2.7. MetaMask :

Is an extension for accessing Ethereum-enabled distributed applications, or "DApps" in your browser.

Figure 17 : MetaMask logo.

After creating an account, you should end up on the main page of Meta Mask.

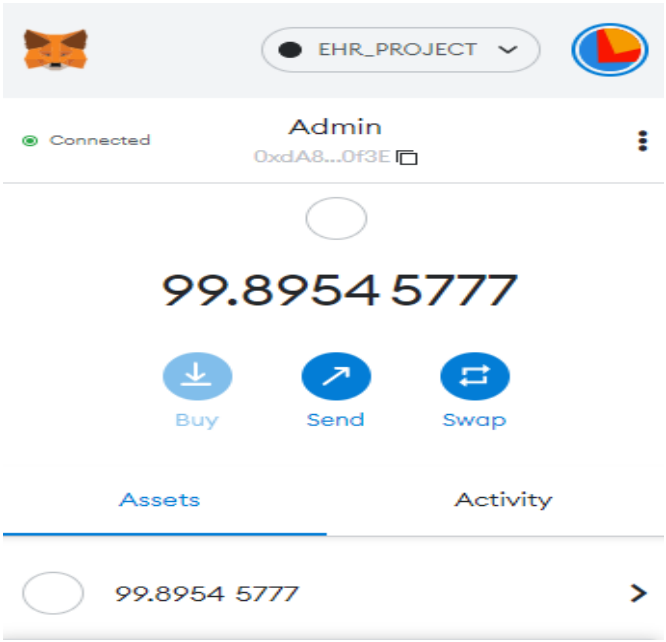

Figure 18 : MetaMask interface.

3 Ethereum Blockchains:

In the Ethereum universe, there is a single canonical computer (called the Ethereum Virtual Machine, or EVM) whose status everyone on the Ethereum network agrees on Figure 19. Whoever participates in the Ethereum network, each Ethereum node maintains a copy of that computer's state. Additionally, any participant can broadcast a request for that computer to perform arbitrary calculations. Each time such a request is broadcast, other participants on the network verify, validate and perform ("run") the calculation. This causes a state change in the EVM, which is committed and propagated throughout the network. [19]

Calculation requests are called transaction requests; the record of all transactions as well as the current state of the EVM is stored in the blockchain, which in turn is stored and agreed to by all nodes. [19]

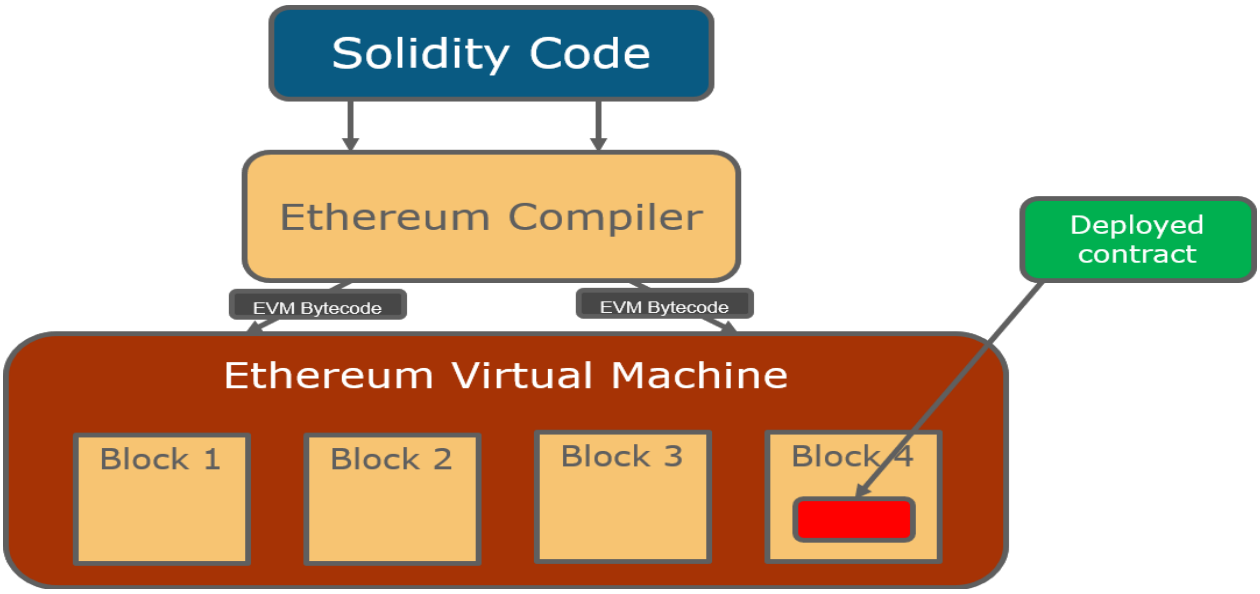

Figure 19: Ethereum Virtual machine

4 Working example for proposed framework :

The system mainly has two entities, i.e., Administrator and User. Users are further divided into two categories for our proposed framework they are doctor and patient. These users are assigned roles by the administrator of the system who is someone belonging to the hospital's administrative staff. Here administrator is assigned the task of the defining the granular access to two main users of our system, i.e., doctor and patient .

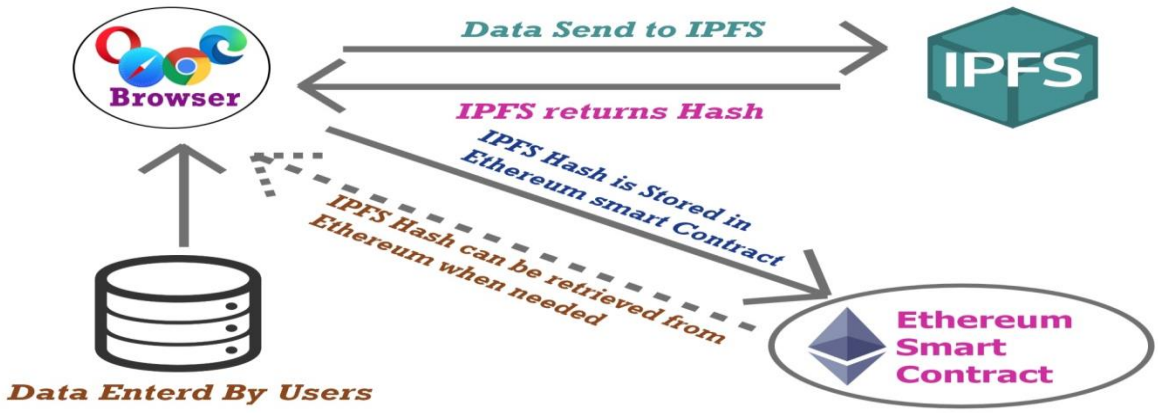

Figure 20 : Work Flow of DApp.

5 Creation of smart contracts:

In this section, we will present some parts of the smart contracts that we have created and used in our application:

```
import './Roles.sol';
contract Contract{
    using Roles for Roles.Role;
    Roles.Role private admin;
    Roles.Role private doctor;
    Roles.Role private patient;
    struct Doctor{
        string drHash;
    }
    struct Patient{
        string patHash;
    }
    struct MedRec{
        string RecordHash;
    }
    mapping(address => Doctor) Doctors;
    mapping(address => Patient) Patients;
    mapping(address => MedRec) Records;
    address[] public Dr_ids;
    address[] public Patient_ids;
    string[] public RecordHashes;

    address accountId;
    address admin_id;
    address get_patient_id;
    address get_dr_id;
```

Figure 21 : **Define the users in the function**

- Add Medical record to block chain

```
    // Add Medical record to block chain

    function addMedRecord(string memory _recHash, address _pat_id) public{
        require(doctor.has(msg.sender) == true, 'Only Doctor Can Do That');

        MedRec storage record = Records[_pat_id];
        record.RecordHash = _recHash;
        RecordHashes.push(_recHash);
    }
```

Figure 22 : Add Medical record to block chain

6 System interfaces :

In this part, we will present some interfaces, starting with the home.

6.1 Home page :

Represents the main page of our site, it offers the possibility for the user (Admin, Patient, Doctor) to access each of their spaces to enter information and to go to the registration or authentication pages.

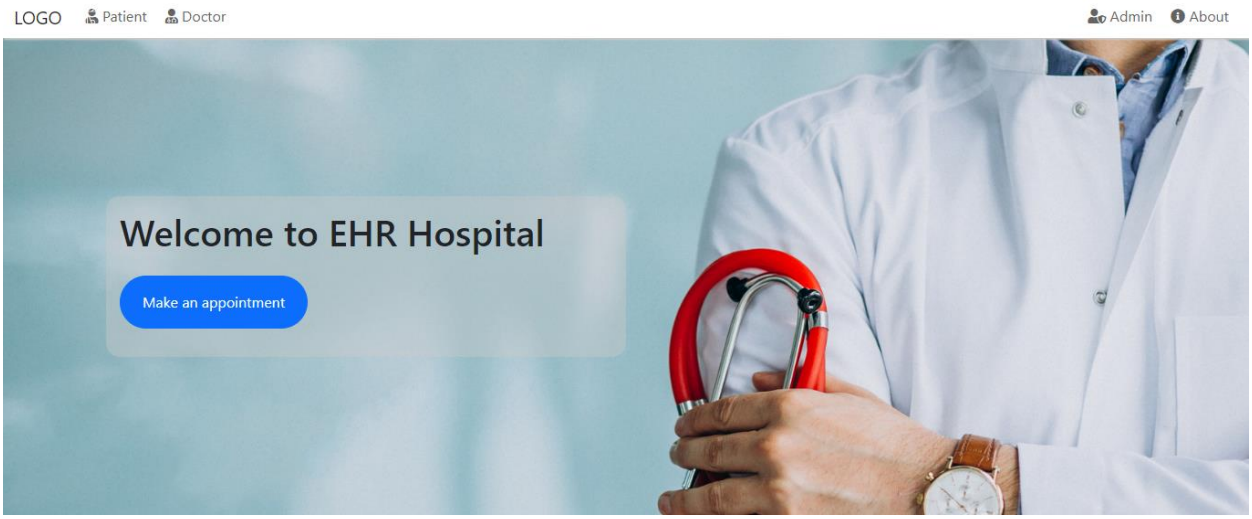

Figure 23: Home Page.

6.2 Admin Page :

If you are an administrator, you can have the list of access requests to our blockchain. And you can access your personal information to modify it if you want.

Admin Can :

- Add Doctor to the Network
- Add Patient to the Network
- Delete User(Doctor/Patient)

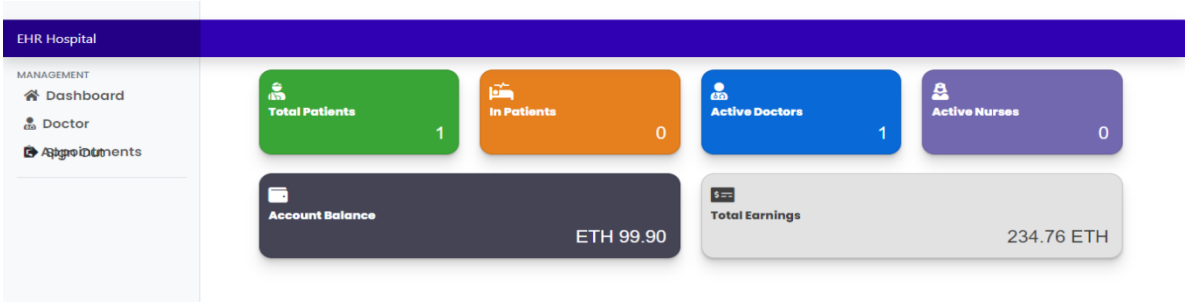

Figure 24 : Admin Page.

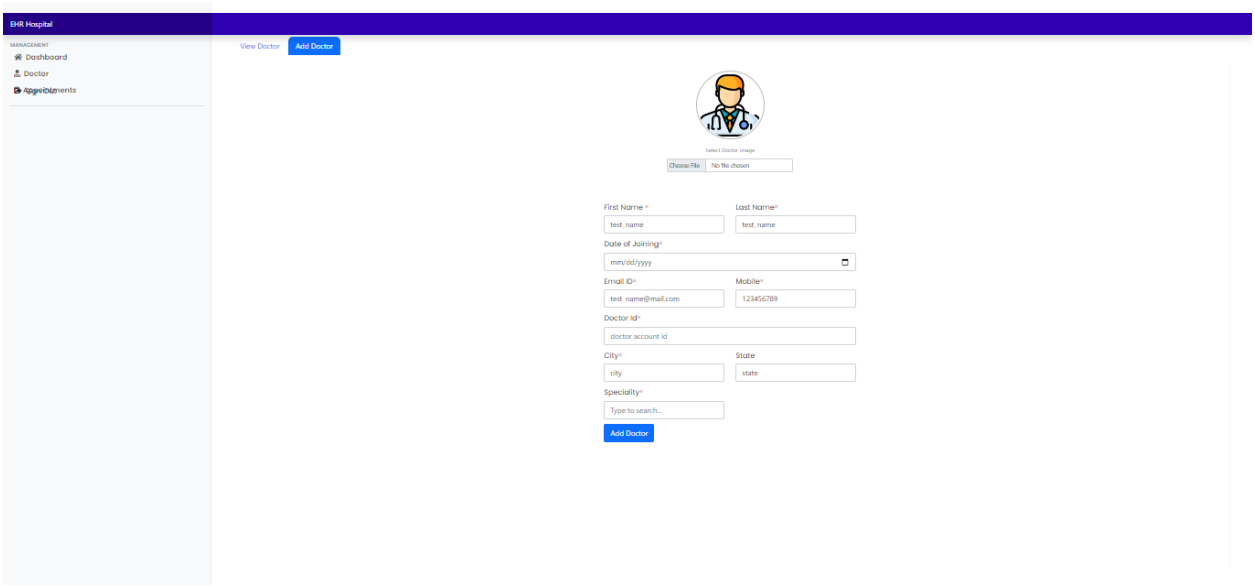

Figure 25: Add Doctor.

The Admin function can only be accessed by System Administrator if Any others try to Access the function it throws a Error.!!

Message Doctor try to access Admin Function Figure Given →

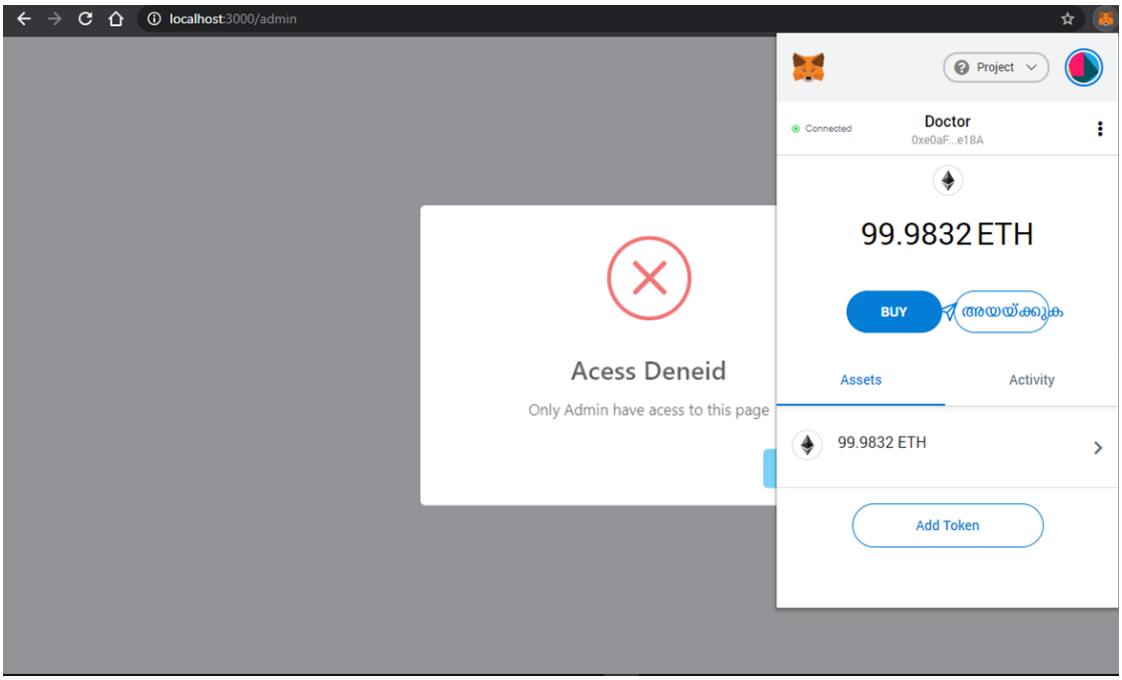

Figure 26 : Doctor try to access Admin Function.

6.3 Doctor Page :

Doctor Can :

- Add Patient Record
- View Patient Record
- Update Patient Record
- Delete Patient Record
- Edit Doctors Info

Doctor Can View Patient Record by Entering the Patents Account ID

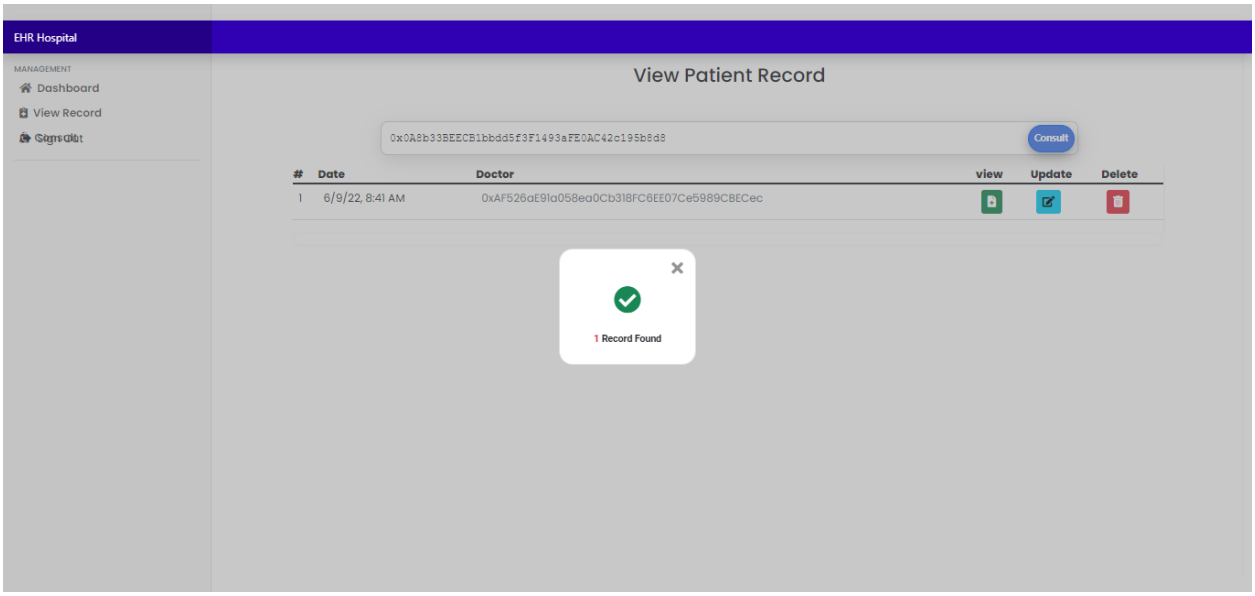

Figure 27: Doctor can View Patient Record.

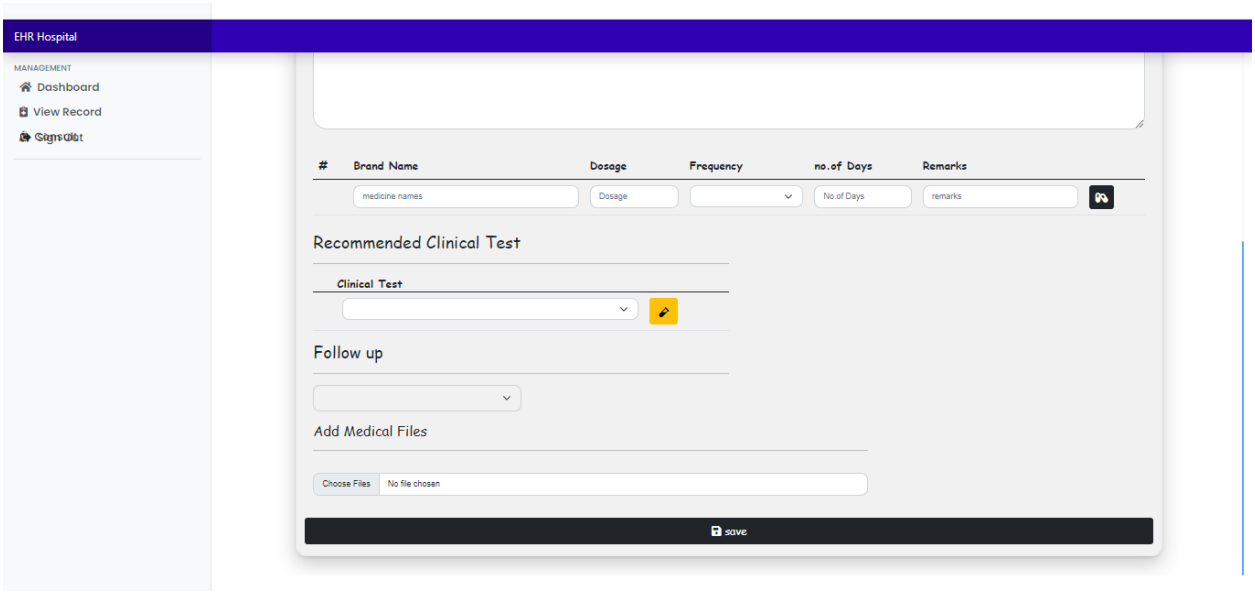

Figure 28 : Doctor Can Update Patient Record.

6.4 Patient Page :

Patient Can :

- View medical info
- Edit About info

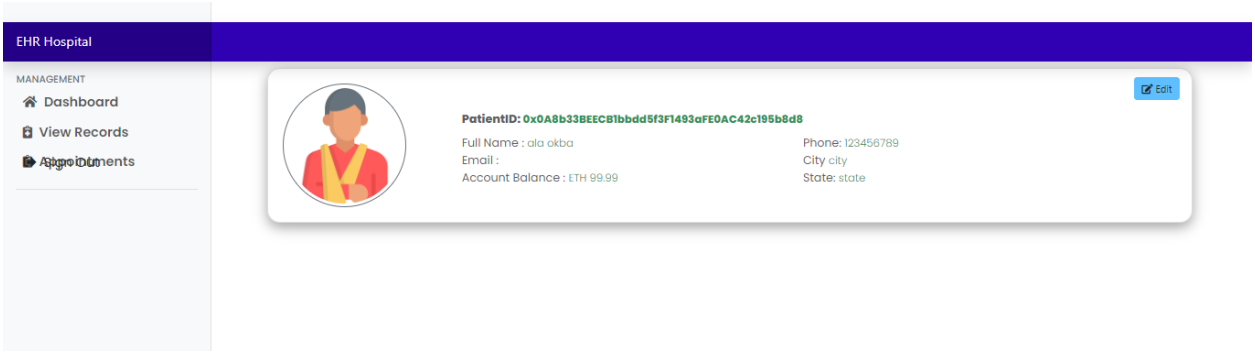

Figure 29: Patient Can View medical info.

7 Conclusion :

In this project we discussed how blockchain technology can be useful for healthcare sector and how can it be used for electronic health records. Despite the advancement in healthcare sector and technological innovation in EHR systems they still faced some issues that were addressed by this novel technology, i.e., blockchain. Our proposed framework is a combination of secure record storage along with the granular access rules for those records. It creates such a system that is easier for the users to use and understand. Also, the framework proposes measures to ensure the system tackles the problem of data storage as it utilizes the off-chain storage mechanism of IPFS. And the role-based access also benefits the

system as the medical records are only available to the trusted and related individuals. This also solves the problem of information asymmetry of EHR system.

8 Future Works :

We plan to implement the payment module in the existing framework. For this we need to have certain considerations as we need to decide how much a patient would pay for consultation by the doctor on this decentralized system functioning on the blockchain. We would also need to implement medical insurance module for patients and also a Appointment module , there the patients can take appointments to see the doctors.

References :

[1] G. Jetley and H. Zhang, ``Electronic health records in IS research: Quality issues, essential thresholds and remedial actions," Decis. Support Syst., vol. 126, pp. 113_137, Nov. 2019.
[2] K. Wisner, A. Lyndon, and C. A. Chesla, ``The electronic health record's impact on nurses' cognitive work: An integrative review," Int. J. Nursing Stud., vol. 94, pp. 74_84, Jun. 2019.
[3] M. Hochman, ``Electronic health records: A ``Quadruple win," a ``quadruple failure," or simply time for a reboot?" J. Gen. Int. Med., vol. 33, no. 4, pp. 397_399, Apr. 2018.
[4] Q. Gan and Q. Cao, ``Adoption of electronic health record system: Multiple theoretical perspectives," in Proc. 47th Hawaii Int. Conf. Syst. Sci., Jan. 2014, pp. 2716_2724.
[5] T. Vehko, H. Hyppönen, S. Puttonen, S. Kujala, E. Ketola, J. Tuukkanen, A. M. Aalto, and T. Heponiemi, ``Experienced time pressure and stress: Electronic health records usability and information technology competence play a role," BMC Med. Inform. Decis. Making, vol. 19, no. 1, p. 160,Aug. 2019.

[6] Hassan, M. U., Rehmani, M. H., & Chen, J. (2019). *Privacy preservation* in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions.
[7] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008.
[8] Hyperledger, 2021, (online), Available: https://www.hyperledger.org.
[9] S. Gupta and M. Sadoghi, ``Blockchain transaction processing," in *Encyclopedia of Big Data Technologies*. 2019, pp. 366_376.
[10] U. W. Chohan, ``Cryptocurrencies: A brief thematic review," *SSRN Electron. J.*, 2017.
[11] G. Wood, ``Ethereum: A Secure Decentralised generalised transaction ledger. EIP-150 revision," Tech. Rep., Aug. 2017, p. 33.
[12] N. Atzei, M. Bartoletti, T. Cimoli, S. Lande, and R. Zunino, ``SoK: Unraveling bitcoin smart contracts," in *Proc. Int. Conf. Princ. Secur. Trust*, Thessaloniki, Greece, 2018, pp. 217_242.
[13] D. Vuji£i¢, D. Jagodi¢, and S. Randi¢, ``Blockchain technology, bitcoin, and Ethereum: A brief overview," in *Proc. 17th Int. Symp. INFOTEH- JAHORINA*

(INFOTEH), Mar. 2018, pp. 1_6.
[14] I. Grishchenko, M. Maffei, and C. Schneidewind, ``A semantic framework for the security analysis of ethereum smart contracts," in *Principles of Security and Trust*. 2018, pp. 243_269.
[15] T. Dey, S. Jaiswal, S. Sunderkrishnan, and N. Katre, ``HealthSense: A medical use case of Internet of Things and blockchain," in *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2017, pp. 486_491.
[16] *InterPlanatery File System (IPFS)*. Accessed: Feb. 4, 2019. [Online]. Available: https://ipfs.io/.

[17]NodeJs , https://nodejs.org/en/

[18] Gregory. How to build Blockchain App-Ethereum

https://www.dappuniversity.com/articles/blockchain-app-tutorial,

consulter le 20/06/2021

[19] Truffle.Js , https://www.trufflesuite.com/

[20]Ethereum Documentation,https://ethereum.org/en/developers/docs/

[21] A. Zhang, X. Lin, Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain, J. Med. Syst. (2018).

[22] Rahman, M. S., Khalil, I., Mahawaga Arachchige, P. C., Bouras, A., & Yi, X. (2019). A Novel Architecture for Tamper Proof Electronic Health Record Management System using Blockchain Wrapper.

[23] Kanwal, T., Anjum, A., & Khan, A. (2020). Privacy preservation in e-health cloud: taxonomy, privacy requirements, feasibility analysis, and opportunities. Cluster Computing.

[24] https://nordvpn.com/blog/what-is-asymmetric-cryptography/.