A major project report on

Using Blockchain for Electronic Health Records

submitted in partial fulfillment of the requirements for the award of

degree of Bachelor of Technology

Submitted by

Manish Kanuri

(20eg105633)

Akshay Reddy

(20eg105637)

Eswarnadh Reddy

(20eg105658)

Under the guidance of

Dr A Jyothi

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANURAG UNIVERSITY

VENKATAPUR – 500088

TELANGANA

YEAR 2023 - 2024

DECLARATION

I hereby declare that the Report entitled "**Using Blockchain for Electronic Health Records**" submitted for the award of Bachelor of technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

Date:16-04-2024

Manish Kanuri
(20eg105633)

Akshay Reddy
(20eg105637)

Eswarnadh Reddy
(20eg105658)

CERTIFICATE

This is to certify that the Report entitled with "**Using Blockchain for Electronic Health Records"**.that is being submitted by **Manish Kanuri** bearing roll number **20eg105633**, **Akshay Reddy** bearing roll number **20eg105637** and **Eswarnadh Reddy** bearing roll number **20eg105658** in partial fulfillment for the award of B.Tech. in to the Anurag University is a record of bonafide work carried out by them under our guidance and supervision.

The results embodied in this Report have not been submitted to any other University or Institute for the award of any degree or diploma.

Dean, CSE

Signature of Supervisor
Dr A Jyothi
Assistant Professor ,CSE

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Dr A Jyothi** for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. Her patience, guidance and encouragement made this project possible.

We would like to express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering , Anurag University, for their encouragement and timely support in our B.Tech program. We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy** , Dean, Dept. of CSE ,Anurag University. We also express my deep sense of gratitude to **Dr. V V S S S Balaram** , Acadamic co- ordinator ,**Dr .Pallam Ravi** , Project in-Charge , **Dr A Jyothi** .Project Co-ordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage our project work.

ABSTRACT

Blockchain has been an interesting research area for a long time and the benefits it provides have been used by a number of various industries. Similarly, the healthcare sector stands to benefit immensely from blockchain technology due to security, privacy, confidentiality, and decentralization. Nevertheless, Electronic Health Record (EHR) systems face problems regarding data security, integrity, and management. In this paper, we discuss how blockchain technology can be used to transform EHR systems and could be a solution to these issues. We present a framework that could be used for the implementation of blockchain technology in the healthcare sector for EHR. The aim of our proposed framework is firstly to implement blockchain technology for EHR and secondly to provide secure storage of electronic records by defining granular access rules for the users of the proposed framework. Moreover, this framework also discusses the scalability problem faced by blockchain technology in general via the use of off-chain storage of records. This framework provides the EHR system with the benefits of having a scalable, secure, and integral blockchain-based solution.

TABLE OF CONTENTS

S.NO	CONTENT	PAGE NO.
1.	**Introduction** 1.1 Problem Statement 1.2 Objective	1 2 2
2.	**Literature Survey**	3-5
3.	**Proposed Methods** 3.1 Existing System 3.2 Disadvantages of existing system 3.3 Proposed System 3.4 Advantages of proposed system	6 6 6 6 7

4.	**Implementation**	8
	4.1 Modules	8
	4.1.1 Patient module	8
	4.1.2 Doctor module	8
	4.1.3 Admin module	8
	4.2 Technologies Used	8-17
	4.3. System Requirements	18
	4.4 Sample Code	18-20
5.	**System Design**	21
	5.1 System Architectue	21
	5.2 UML Diagrams	22-24
	5.3 Flow of Events	24-25
	5.3.1 Construction of Usecase diagrams	25-26
	5.3.2 Sequence Diagram	26-27
	5.3.3 Class Diagram	27-28
6.	**Experimental Results**	29
	6.1 Introduction	29
	6.2 Levels of Testing	29-32
7.	**Discussion of Results**	33
	7.1 Test cases	35

	7.2 Performance Assessment 7.3 Result Screen shots	36 37-40
8.	Conclusion	41
9.	Future Scope	42
10.	References	43-44

LIST OF FIGURES

Figure. No.	Figure. Name	Page No.
Figure 4.2.1.1	Illustration of how program works	9
Figure 4.2.1.2	Working of VM	10
Figure 4.2.1.3	Programming on Java platform	11
Figure 4.2.1.4	Blockchain	17
Figure 5.1.1	Architecture design	21
Figure 5.2.1	Graphical representation	22
Figure 5.3.1.1	Use case Diagram	26
Figure 5.3.2.1	Activity Diagram	27
Figure 5.3.3.1	Class Diagram	28
Figure 7.3.1	Patient Login page	37
Figure 7.3.2	Patient Registration page	37
Figure 7.3.3	Patient Book Appointment page	38
Figure 7.3.4	Upload documents	38
Figure 7.3.5	Admin login page	38
Figure 7.3.6	Admin page	39
Figure 7.3.7	Doctor registration page	39
Figure 7.3.8	Document Requested	39
Figure 7.3.9	Keys sent to user	40
Figure	User verification	40

LIST OF TABLES

SNO	TABLES	PAGE NO
1	Table 7.1.1 Test Case 1	33
2	Table 7.1.1 Test Case 2	34
3	Table 7.1.1 Test Case 3	34-35
4	Table 7.1.1 Test Case 4	35

LIST OF GRAPHS

SNO	GRAPHS	Page No.
1	Graph 7.2.1 Performance	36
2	Graph 7.2.2 Latency	36

1.INTRODUCTION

The recent advent in technology is affecting all parts of human life and is changing the way we use and perceive things previously. Just like the changes technology has offered in various other sectors of life, it is also finding new ways for improvement in the healthcare sector. The main benefits that advancement in technology is offering are to improve security, user experience and other aspects of healthcare sector. These benefits were offered by Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems. However, they still face some issues regarding the security of medical records, user ownership of data, data integrity etc. The solution to these issues could be the use of a novel technology, i.e., Blockchain. This technology offers to provide a secure, tamper-proof platform for storing medical records and other healthcare related information.

Before the advent of modern technology, the healthcare sector used a paper-based system to store the medical records, i.e., using handwritten mechanism. This paper-based medical record system was inefficient, insecure, unorganized and was not tamper-proof. It also faced the issue of data-duplication and redundancy as all the institutions that the patient visited had various copies of the patient's medical records. The healthcare sector faced a trend shift towards EHR 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. They were proposed to enhance the safety aspect of the patients by preventing errors and increasing information access. 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 state of the healthcare sector.

The EHR systems have been implemented in a number of hospitals around the world due to the benefits it provides, mainly the improvement in security and its cost-effectiveness. They are considered a vital part of the healthcare sector as it provides much functionality to healthcare.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 systems being used in the healthcare sector. The basic focus is to provide secure, tamper-proof, and shareable medical records across different platforms. Despite the fact that the notion behind the usage of EHR systems in the hospitals or healthcare was to improve the quality of healthcare, these systems faced certain problems and didn't meet the expectations associated with them.

1.1 PROBLEM STATEMENT

The problem statement revolves around the vulnerability of medical data to theft, tampering, or deletion, jeopardizing patient safety and treatment efficiency. A secure data preservation system is essential, ensuring data integrity, privacy, and accessibility amid these challenges. Advanced encryption techniques must safeguard sensitive information, while strict access controls prevent unauthorized entry. Redundancy measures, like regular backups, mitigate data loss risks. By prioritizing security and reliability, healthcare providers can maintain patient trust and streamline treatment processes while safeguarding critical medical information.

1.2 OBJECTIVE

Traditional cloud storage methods can be susceptible to breaches and unauthorized access. This project aims to significantly improve security by implementing a two-layer defense system. By combining RSA encryption for data confidentiality and a blockchain system for decentralized access control, the project strives to create a robust shield against data breaches. Ultimately, this enhanced security aims to address the concerns around data exchange in healthcare, fostering trust and enabling secure collaboration and research.

2. LITERATURE SURVEY

Blockchain technology, originally designed by Nakamoto [13], the basic idea was to have a cryptographically secured and a decentralized currency that would be helpful for financial transactions. Eventually, this idea of blockchain was being used in various other fields of life; healthcare sector also being one of them intends to use it. A number of researchers have carried out the research on this area, these research works focus on the fact that whether the idea of using blockchain for healthcare sector is feasible or not. They also identify the advantages, threats, problems or challenges associated by the usage of this technology. Some researchers also discussed the challenges that would be faced while actually implementing this on a larger scale.

Gordon and Catalini [14], conducted a study that focused on the methods by which blockchain technology would facilitate the healthcare sector. They identified, that healthcare sector is controlled by hospitals, pharmaceutical companies and other involved third parties. They specified data sharing as the key reason why blockchains should be used in healthcare. This study also identified four factors or approaches due to which healthcare sector needs to transform for usage of blockchain technology. These include way for dealing of digital access rights, data availability, and faster access to clinical records and patient identity. It also discusses the on-chain and off-chain storage of data. The study also included the challenges or barriers faced by usage of blockchain technology these were huge volume of clinical records, security and privacy, patient engagement.

Eberhardt and Tai [20], conducted a study to understand possible approaches to solve the scalability problem of blockchain and also to identify such projects that intend to solve this problem. They define blockchain as composition of various computational and economical concepts based on peer-to-peer system. The aim of this study was to find which data should be stored on-chain and what could be stored off-chain. This study presented five patterns for off-chain storage of data and also includes the basic ideas and implementation framework of these patterns. The authors explain on-chain data is any data that is stored on the blockchain by performing transactions on it. While off-chain data storage is to place data elsewhere on any other storage medium but not

on-chain and it also would not include any transactions.

Vujičić et al [21], presented an overview of blockchain technology, bitcoin and Ethereum. The authors define that information technology landscape is constantly changing and blockchain technology is benefiting the information systems. They explained bitcoin as a peer-to-peer distributed network used for performing bitcoin transactions. They also defined that proof-of-work consensus algorithm along with the mining of blockchain concept. The authors emphasize on the fact that scalability is a severe problem faced by blockchain and that certain solutions are proposed for solution of scalability problem these include SegWit and Lightning, Bitcoin Cash and Bitcoin Gold. The paper also explained Ethereum and its dependencies and it also differentiates Ethereum blockchain from bitcoins' blockchain.

Wang et al [22], conducted a study that focused on smart contracts and its application in blockchain technology. They first introduce the smart contracts, their working framework, operating systems and other important concepts attached with them. The authors also discuss that how could smart contracts be used for the new concept of parallel blockchains. They identify that reason of using smart contracts in blockchain is due to the decentralization that is offered through the programming language code written in them. After introducing the basics of smart contract the author explained the various layers of blockchain that combine together to keep system functioning. These layers are data, network, consensus, incentive, contract, and application layer. The paper not only discusses the architecture and framework followed by smart contracts but it also gives an insight on its applications and challenges. The paper also discusses an important future trend of parallel blockchain that intends to create such blockchain that can optimize two different but important modules.

Kuo et al [23], conducted a review that discussed several applications of blockchain in biomedical and healthcare sector. The authors identified that using blockchains for this domain offers many advantages and some of these are decentralization, persistence of clinical or medical records, data pedigree, and continuous accessibility to data and lastly secure information being accessible to biomedical or healthcare stakeholders. The limitations of blockchain technology were identified to be, confidentiality, speed, scalability and threat of malicious attack, i.e., 51% attack. The authors identified these limitations to be critical for healthcare or biomedical sector as they are being used to

store sensitive medical or clinical records.

Sahoo and Baruah [24], proposed a scalable framework of blockchain using Hadoop database. In order to solve the scalability problem of blockchain, 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 of this framework includes all of the needed dependencies of blockchain but the blocks are stored on Hadoop database to improve scalability of the blockchain technology. To tackle the scalability problem of blockchain platform this study offers to use Hadoop database system, along with SHA3-256 for hashing used for transactions and blocks. The programming language used for this architecture was Java. This study, was helpful in understanding that blockchain can be used with other platforms that are scalable to improve or solve the scalability of this platform.

Zhang et al [25], proposed a scalable solution to the blockchain for clinical records. The basic aim of this study was to design such an architecture that complies with the Office of National Coordinator for Health Information Technology (ONC) requirements. This study identified the barriers that this technology faces mainly include concerns related to privacy, security of blockchain, and scalability problems related to huge volume of datasets being transmitted on this platform, and lastly there is no universal standard enforced for data being exchanged on blockchain. This study also include a demonstration of a decentralized application (DAPP) based on the design formulated on the ONC requirements as mentioned before. They also included the lessons learnt and how can FHIR chain be improved.

Kim et al [26], proposed a system for management of medical questionnaires and the aim of this system is data sharing through blockchain technology. The authors explain that selection of data storage and sharing of medical questionnaire is to use this data for further medical and clinical research purposes. The questionnaires that are added on this system are first validated to be correct specified format and then are parsed to differentiate the personal data and specific data related to questionnaire results. The authors also address the scenario when a third party requests to access this questionnaire data, this would need the patients' permission that is asked by the doctor to let third party view that data.

3.PROPOSED METHODS

3.1 EXISTING SYSTEM

The existing healthcare record systems, primarily Electronic Health Records (EHR) and Electronic Medical Records (EMR), have significantly advanced healthcare management but are not without their drawbacks. These systems often face challenges related to data security vulnerabilities, user data ownership ambiguity, and data integrity issues. Security vulnerabilities expose these systems to risks of data breaches and cyber-attacks, compromising patient confidentiality and the integrity of medical records. Moreover, the ambiguity surrounding user data ownership raises concerns about privacy and inhibits effective data management. Additionally, centralized systems may encounter problems such as data duplication and inconsistency, leading to inaccuracies and inefficiencies in healthcare data management. These limitations underscore the need for innovative solutions to address the shortcomings of existing healthcare record systems.

3.2 DISADVANTAGES OF EXISTING SYSTEM:

- Security concerns: Existing systems face challenges related to the security of medical records, as they are vulnerable to data breaches and cyber-attacks.
- Lack of data ownership: Patients often do not have full ownership and control over their medical data, leading to privacy concerns and potential misuse.
- Data integrity issues: The centralized nature of existing systems can lead to data duplication and inconsistency, affecting the accuracy and reliability of medical records.

3.3 PROPOSED SYSTEM

Proposed blockchain-based systems offer promising solutions to address the shortcomings of existing healthcare record systems. These systems leverage the inherent security and decentralization of blockchain technology to enhance the management, security, and accessibility of medical data. For example, researchers have introduced scalable frameworks that integrate blockchain with existing databases, such as Hadoop, to improve the scalability of healthcare data storage and management. Additionally, blockchain-based solutions ensure data integrity and immutability, minimizing the risk of unauthorized access, tampering, and

data breaches. Furthermore, these systems prioritize patient data ownership and control, providing patients with greater autonomy over their medical records and fostering trust between patients and healthcare providers. By streamlining data sharing and access, blockchain-based systems facilitate efficient communication and collaboration among healthcare stakeholders, ultimately improving the quality of patient care and healthcare outcomes.

3.4 ADVANTAGES PROPOSED SYSTEM

- Enhanced security: Blockchain technology offers a decentralized and tamper-proof platform, ensuring the security and integrity of medical records.
- Improved data ownership: Patients have greater control and ownership over their medical data, enhancing privacy and confidentiality.
- Efficient data management: Blockchain-based systems can streamline data sharing and access, reducing duplication and improving the overall efficiency of healthcare data management.

4 IMPLEMENTATION

4.1 MODULES:

4.1.1 Patient model:

Patients register on the application, providing necessary details for login. They upload files and share them with registered doctors. Patients can also view their uploaded files and requests for secret keys and blockchain keys from doctors. Responses, along with the keys, are sent to the doctor via email. With the provided keys, doctors can download and view the files securely, ensuring seamless access to medical information while maintaining privacy and security measures.

4.1.2 Doctor Module:

Doctors can register on the application, obtaining unique usernames and passwords. They gain access to view encrypted files uploaded by all users. To download data, they must request approval from the respective user. Upon approval, blockchain hashes and security keys are shared with the owner via email for download. This process ensures secure data access and maintains privacy while enabling efficient sharing between doctors and users

4.1.3 Admin Module:

The admin can register within the application and oversee user data, including uploaders and requesters. Details maintained by the admin are then securely stored in the database, ensuring efficient management of user information.

4.2 TECHNOLOGIES USED

4.1.4 Java Technology

Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

In the Java programming language, a program undergoes both compilation and interpretation. Initially, the program is compiled into an intermediate language known as Java bytecodes, which are platform-independent and interpreted by the Java platform's interpreter. This interpreter then parses and executes each Java bytecode instruction on the computer. The compilation process occurs only once, while interpretation happens each time the program is run. This unique approach allows Java programs to be portable across different platforms, as the bytecode interpretation adapts to the underlying system architecture.

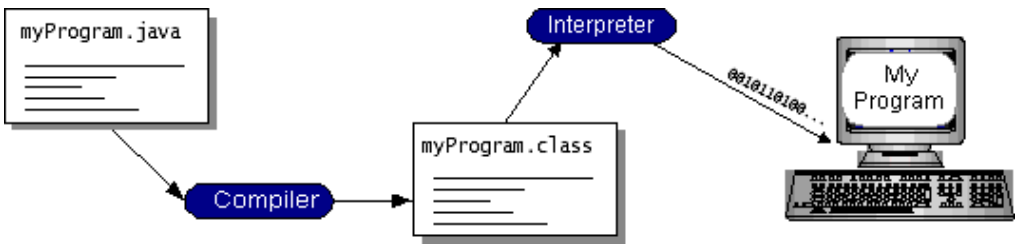


Fig 4.3.1.1 Illustration of how program works

Java byte codes serve as the machine code instructions for the Java Virtual Machine

(Java VM). Each Java interpreter, whether it's a development tool or a web browser capable of running applets, functions as an implementation of the Java VM. By compiling programs into byte codes, the "write once, run anywhere" paradigm becomes achievable. With a Java compiler present, programs can be compiled into byte codes on any platform. These byte codes are then executable on any Java VM implementation.

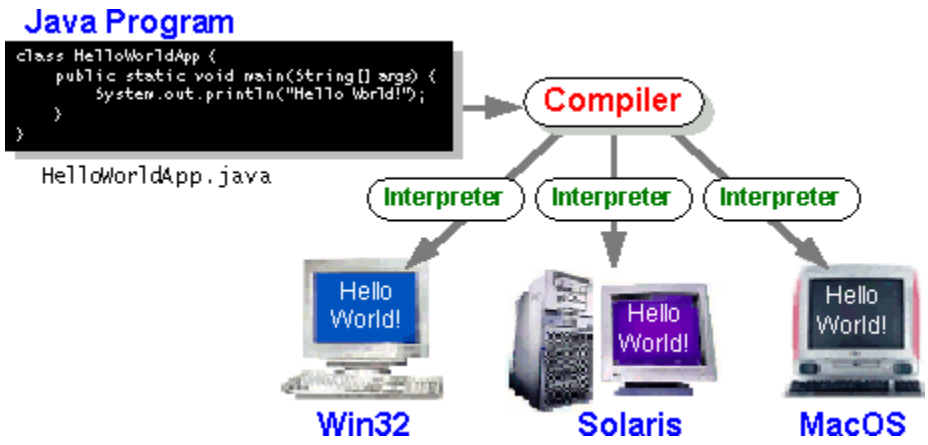


Fig 4.3.1.2 Working Java VM

The Java Platform

A platform is the hardware or software environment in which a program runs.We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

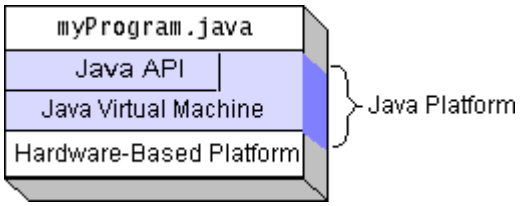


Fig 4.3.1.3 A program running on the Java platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door

will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer. .

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As

you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

2. SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. JDBC must be implemental on top of common database interfaces

The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. Provide a Java interface that is consistent with the rest of the Java system

Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6.Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

Networking

- **TCP/IP stack**

The TCP/IP stack is shorter than the OSI one

TCP is a connection-oriented protocol; UDP (User Datag is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call `socket`. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with `Read File` and `Write File` functions.

```
#include <sys/types.h>

#include <sys/socket.h>

int socket(int family, int type, int protocol);
```

Here "family" will be `AF_INET` for IP communications, `protocol` will be zero, and `type` will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

BLOCKCHAIN SYSTEM

Blockchain technology is a database mechanism that stores data in blocks linked together in a chain. Each block contains a timestamp, transaction data, and a cryptographic hash of the previous block. The blocks are linked together to form a chain that is chronologically consistent because the chain cannot be modified or deleted without consensus from the network.

Blocks store the data in terms of transaction, date, time, amount etc. This feature contributes to the data integrity principle of security.

It also stores the participating entities of the transaction that are involved. This maintains the authenticity of the security principal.

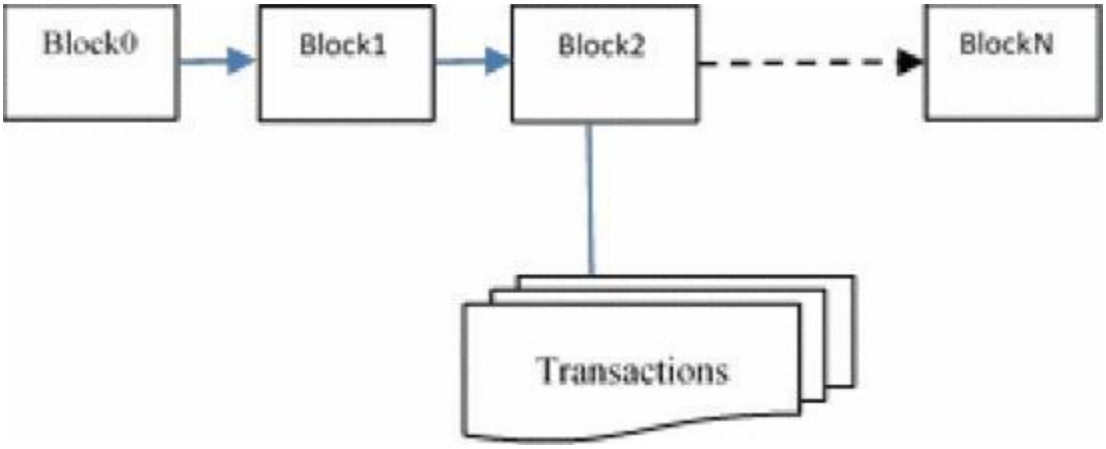


figure 4.2.1.4

The blocks generated are unique by calculating its hash. Hash is one time function that is irreversible. This feature ensures that each record is different, consisting of different transactions and identities. This maintains the integrity of the record.Once the records are stored they cannot be altered and these blocks are available for access by all the entities participating in the network and transactions. Thus maintains the non-repudiation and confidentiality in the transactions.All these features of Blockchain have made its usage in the wide area of applications for providing security. Figure shows the basic structure of blockchain. It shows that the blocks are interconnected with the property that change in one block leads to changes in the subsequent blocks. This property made it difficult or nearly impossible for a malicious entity to enter and update in-between transactions.

4.3 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 200 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Ram : 1 GB.

SOFTWARE REQUIREMENTS:

- Operating system : Windows XP/7/10.
- Coding Language : Java
- Tool : Netbeans
- Database : MYSQL
-

4.4 SAMPLE CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>Cloud Dispersion</title>
  <meta content="" name="description">
  <meta content="" name="keywords">
  <!-- Favicons -->
```

```
<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
  <!-- Template Main CSS File -->
  <link href="assets/css/style.css" rel="stylesheet">
 </head>
<body>
  <!-- ======= Header ======= -->
  <header id="header" class="fixed-top d-flex align-items-center">
    <div class="container d-flex justify-content-between">
      <div class="logo">
        <h1><a href="index.html">Cloud Dispersion</a></h1>
        <!-- Uncomment below if you prefer to use an image logo -->
        <!-- <a href="index.html"><img src="assets/img/logo.png" alt="" class="img- fluid"></a>-->
      </div>
      <nav id="navbar" class="navbar">
        <ul>
          <li><a class="nav-link scrollto " href="index.html">Home</a></li>
          <li><a class="nav-link scrollto " href="owner.jsp">Owner</a></li>
          <li><a class="nav-link scrollto active" href="cloud.jsp">Cloud</a></li>
          <li><a class="nav-link scrollto " href="user.jsp">User</a></li>
        </ul>
```

```
    <input type="password" name="password" id="form2Example2"
placeholder="password" required="" class="form-control" />

  </div>

  <!-- 2 column grid layout for inline styling -->

  <!-- Submit button -->

 <button type="submit" class="btn btn-primary">Submit</button>

  <br>

</form>

                        </center>

   </div>

  </section><!-- End Hero -->

  <main id="main">

  </main><!-- End #main -->

  <!-- ======= Footer ======= --

  <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i
class="bi bi-arrow-up-short"></i></a>

  <!-- Vendor JS Files -->

  <script src="assets/vendor/aos/aos.js"></script>

  <script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

  <script src="assets/vendor/glightbox/js/glightbox.min.js"></script>

  <script src="assets/vendor/isotope-layout/isotope.pkgd.min.js"></script>

  <script src="assets/vendor/swiper/swiper-bundle.min.js"></script>

  <script src=“assets/vendor/php-email-form/validate.js"></script>

……..

……..
```

5 SYSTEM DESIGN

System Design Introduction:

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

5.1 SYSTEM ARCHITECTURE

Architecture Flow:

Below architecture diagram represents mainly flow of request from the users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business layer, data link layer. This project was developed using 3-tier architecture.

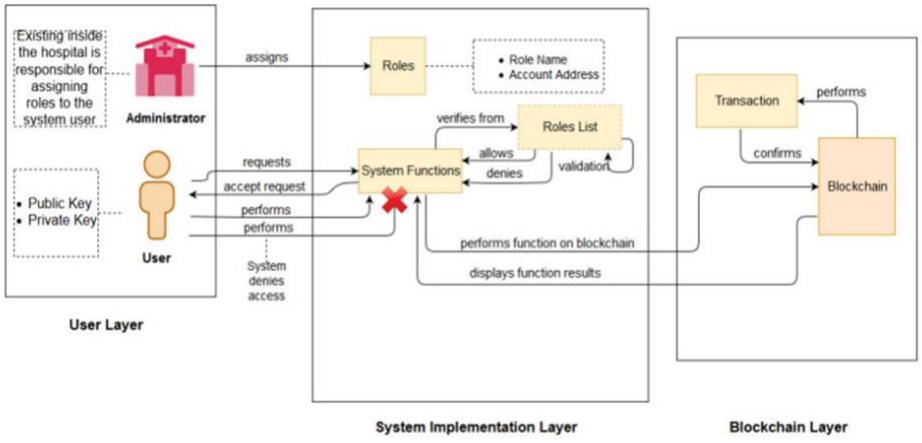


Figure 5.1.1: Architecture diagram

3-Tier Architecture:

The three-tier software architecture (a three layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process

management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems

.

Advantages of Three-Tier:

- Separates functionality from presentation.
- Clear separation – better understanding.
- Changes limited to well define components.
- Can be running on WWW.
- Effective network performance.

5.2 UML DIAGRAMS

Global Use Case Diagrams:

Identification of actors:

Actor: Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

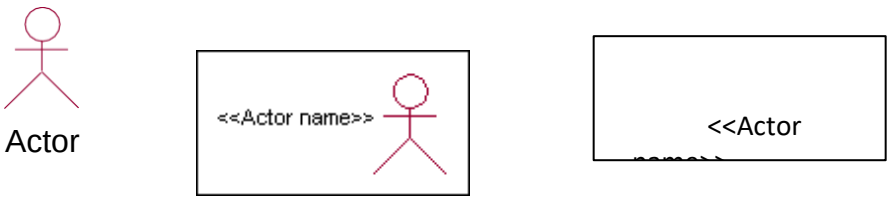


Fig 5.2.1 Graphical representation:

An actor is someone or something that:

Interacts with or uses the system.

- Provides input to and receives information from the system.
- Is external to the system and has no control over the use

cases. Actors are discovered by examining:

- Who directly uses the system?
- Who is responsible for maintaining the system?
- External hardware used by the system.
- Other systems that need to interact with the

system. Questions to identify actors:

- Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?
- Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.
- Which external hardware or systems (if any) use the system to perform tasks?
- What problems does this application solve (that is, for whom)?
- And, finally, how do users use the system (use case)? What are they doing with the system?

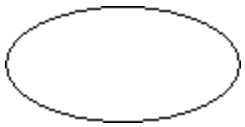
The actors identified in this system are:

a. System Administrator
b. Customer
c. Customer Care

Identification of usecases:

Usecase: A use case can be described as a specific way of using the system from a user's (actor's) perspective.

Graphical representation



A more detailed description might characterize a use case as:

- Pattern of behavior the system exhibits
- A sequence of related transactions performed by an actor and the system
- Delivering something of value to

the actor Use cases provide a means to:

- capture system requirements
- communicate with the end users and domain experts
- test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guide lines for identifying use cases:

- For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
- Name the use cases.
- Describe the use cases briefly by applying terms with which the user is familiar.

5.3 Flow of Events

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

- When and how the use case starts and ends
- Use case/actor interactions
- Data needed by the use case
- Normal sequence of events for the use case
- Alternate or exceptional flows

5.3.1 Construction of Usecase diagrams:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

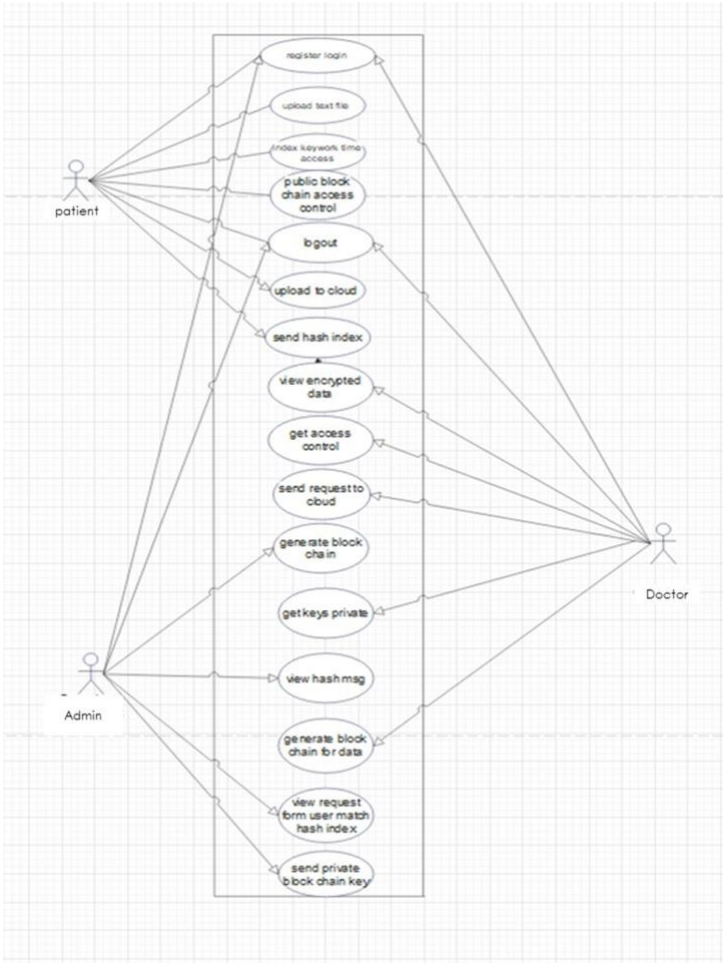


Figure 5.3.1.1 Use Case Diagram

5.3.2 SEQUENCE DIAGRAMS:

Activity diagrams are graphical representstions of workflows of stepwise activities and actions with support of choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business amd operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

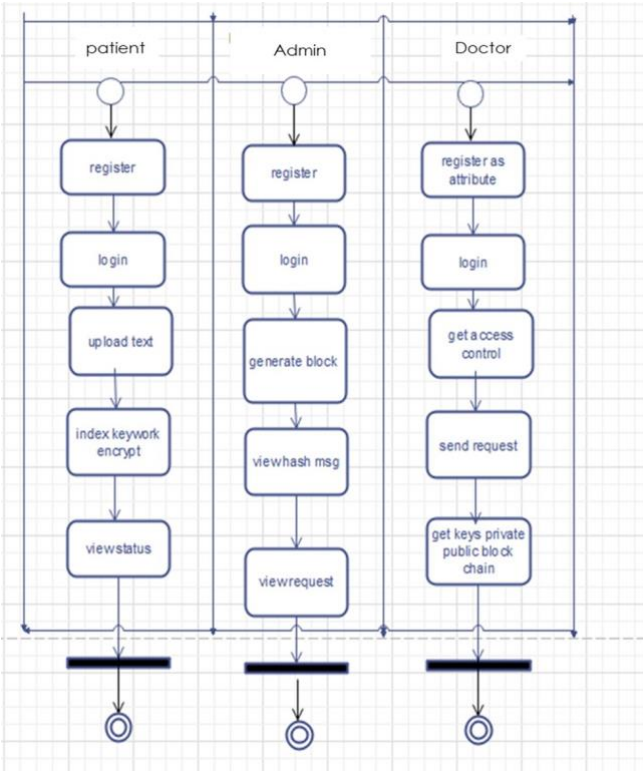


Figure 5.3.2.1 Activity diagram

5.3.3. CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

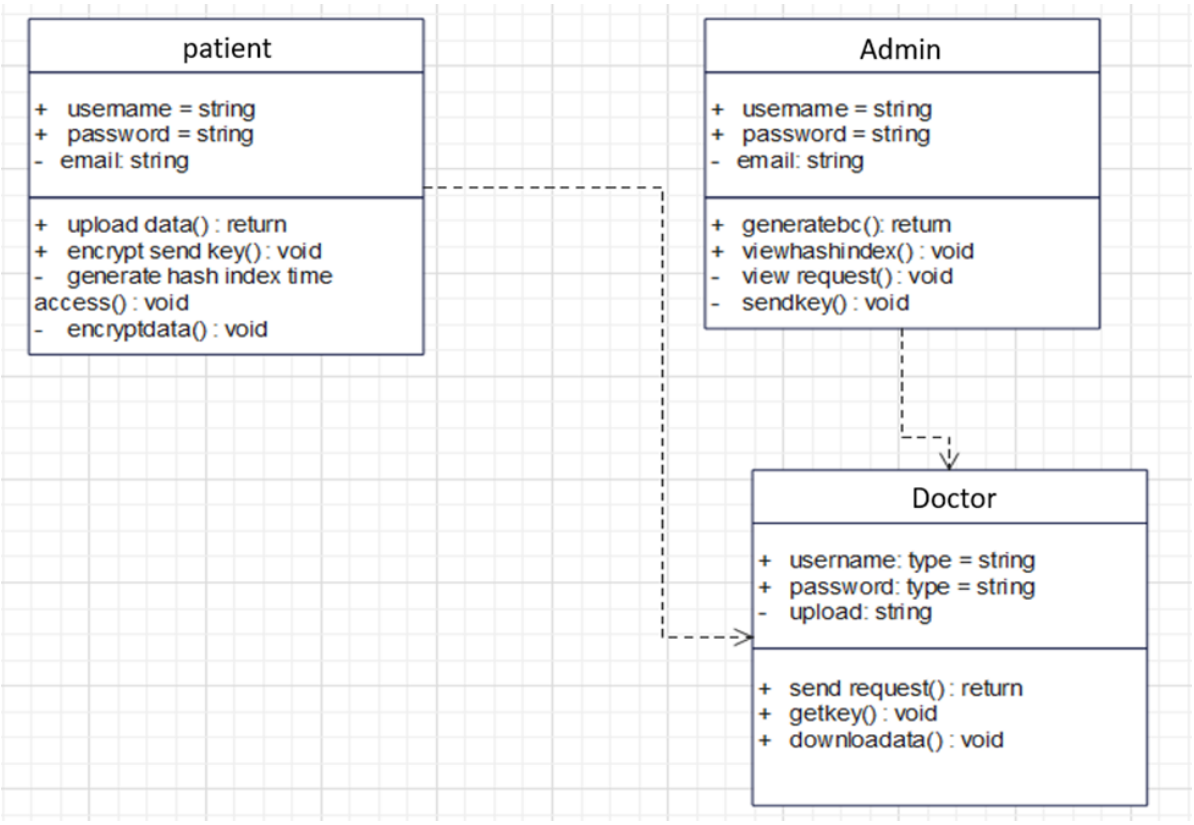


Figure 5.3.3.1: Class Diagram

6. EXPERIMENT RESULTS

6.1 INTRODUCTION:

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

1. A successful test is one that uncovers an as yet undiscovered error.
2. A good test case is one that has probability of finding an error, if it exists.
3. The test is inadequate to detect possibly present errors.
4. The software more or less confirms to the quality and reliable standards.

6.2. Levels of Testing:

Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

Specification Testing:

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

BLACK BOX TESTING

What is Black Box Testing?

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

The Black Box can be any software system you want to test. For example : an operating system like Windows, a website like Google ,a database like Oracle or even your own custom application. Under Black Box Testing , you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Black box testing - Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

WHITE BOX TESTING

White Box Testing is the testing of a software solution's internal coding and infrastructure.It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.White box testing is also known as clear, open, structural, and glass box testing.

It is one of two parts of the "box testing" approach of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

What do you verify in White Box Testing ?

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code

- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

How do you perform White Box Testing?

To give you a simplified explanation of white box testing, we have divided it into two basic steps. This is what testers do when testing an application using the white box testing technique:

STEP 1) UNDERSTAND THE SOURCE CODE

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing.. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

STEP 2) CREATE TEST CASES AND EXECUTE

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

7. DISCUSSION OF RESULTS

7.1 Test Cases

Below stable shows the test case for the admin. The details if admin are stored in the database such as admin name, admin name etc. Admin is responsible for handling the transactions. The admin sets password and user name. Weather all the passwords and usernames are correct not is checked. Because by using these details the user will login in into the cloud and can view the details of file stored.

Test case 1

The test case 1 tests the login of admin's. The passwords and username is given if the correct password and usernames are entered the login will be successful. If any wrongin passwords and username the login will be denied.

Test Case : -	UTC-1
Name of Test: -	**Patient login**
Items being tested: -	**Validation for Patient login**
Sample Input: -	**Fill form**
Expected output: -	**Details stored in database if wrong details are given check validation**
Actual output: -	**Validation verified**
Remarks: -	Pass.

Table 7.1.1 Test Case1

Test case 2

The test case 2 is for checking the users. Again for the doctor password and user name is given based on only the correct input the users will be able to login

Sl # Test Case : -	UTC-2
Name of Test: -	**upload data**
Items being tested: -	**Data encrypted or not**
Sample Input: -	**Text file**
Expected output: -	**encryption key generated and stored**
Actual output: -	**encryption can be viewed by server**
Remarks: -	sucess.

Table 7.1.2 Test Case2

Test case 3

The test case 3 shows for the file upload and it is being stored in different cloud which is called block generation and storing and hence this file is divided and stored .If the storage and block generation is not successful than we can store the file.

Sl # Test Case : -	ITC-1
Name of Test: -	**File request**
Item being tested: -	**Request sent to patient**

Sample Input: -	**Encrypted Text data**
Expected output: -	**Data stored in server, and database with confirmation**
Actual output: -	**Request sent success.**
Remarks: -	Pass.

Table 7.1.3 Test Case 3

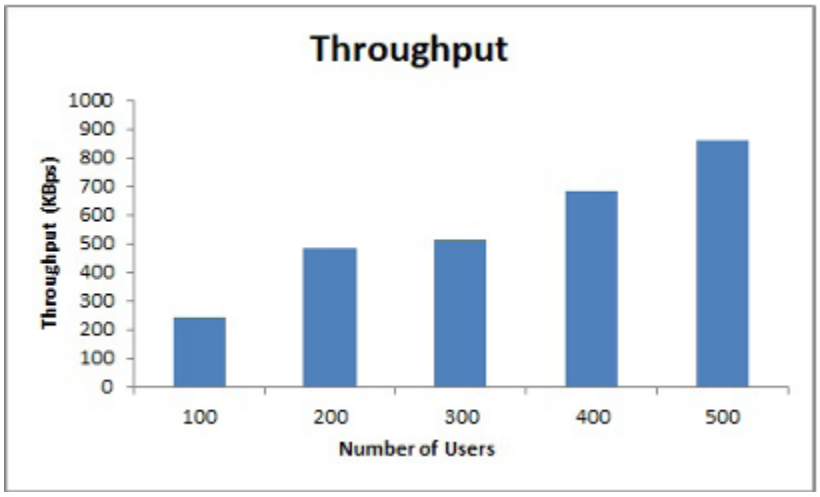
Sl # Test Case : -	STC-1
Name of Test: -	**doctor completed download**
Item being tested: -	**Check tasks and complete download**
Sample Input: -	**Encrypted data**
Expected output: -	**Data decrypted send to doctor**
Actual output: -	**doctor can download data**
Remarks: -	Pass

Table 7.1.4 Test Case 4

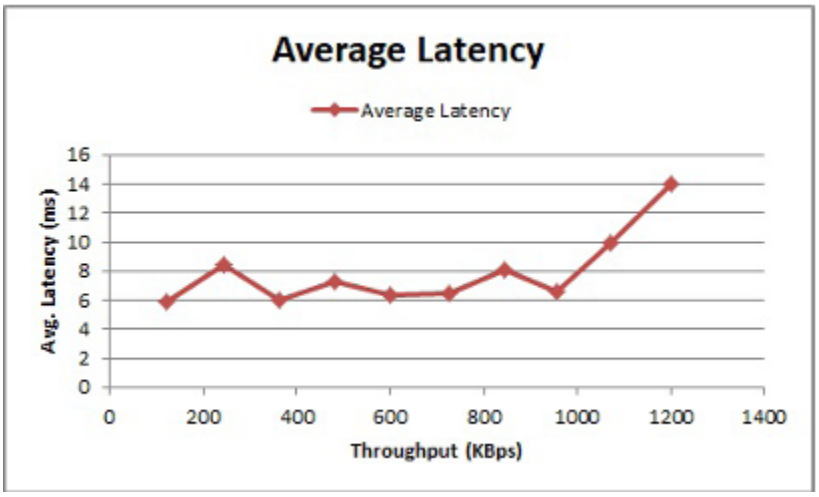
Test case 4

Test case 4 is for file upload where the uploaded is downloaded. During downloadif any one of the cloud fails the recovery should start automatic. If the recovery is successful all the blocks will be generated and merged together.

7.2 PERFORMANCE ASSESMENT



Graph 7.2.1 Performance



Graph 7.2.2 Latenc

7.3 RESULT SCREEN SHOTS

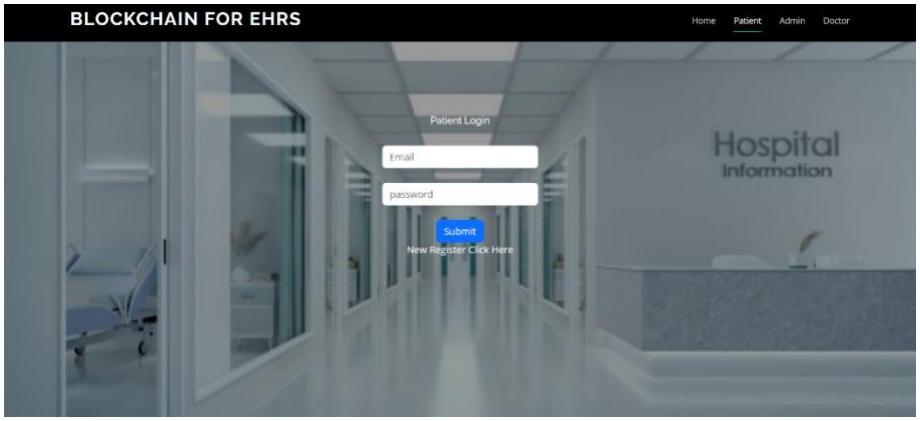


Fig 7.3.1 Patient Login page

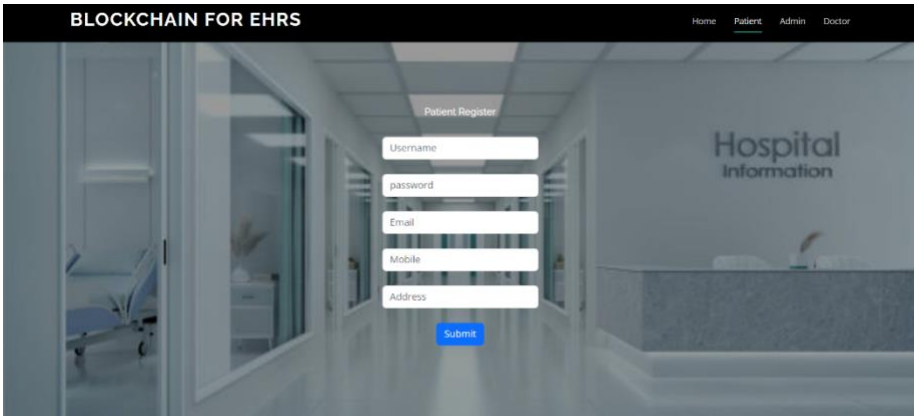


Fig 7.3.2 Patient Registration page

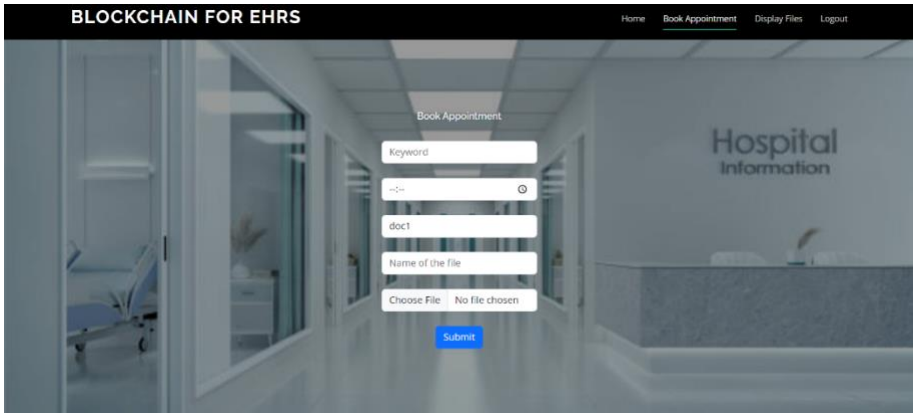


Fig 7.3.3 Patient Book Appointment page

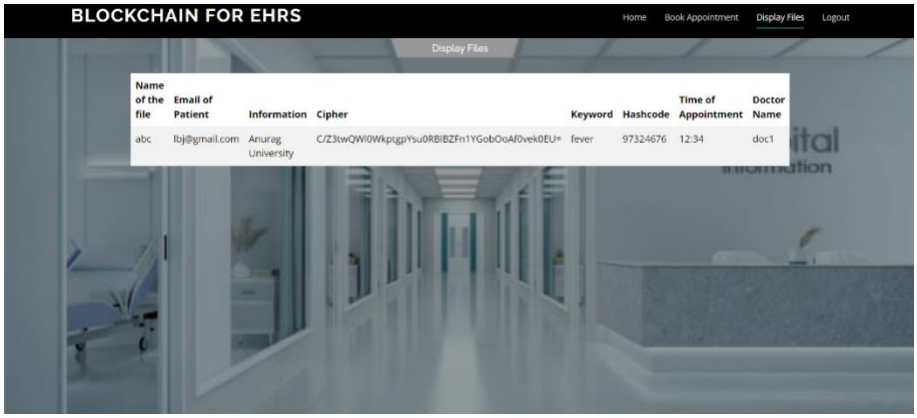


Fig 7.3.4 All uploaded documents

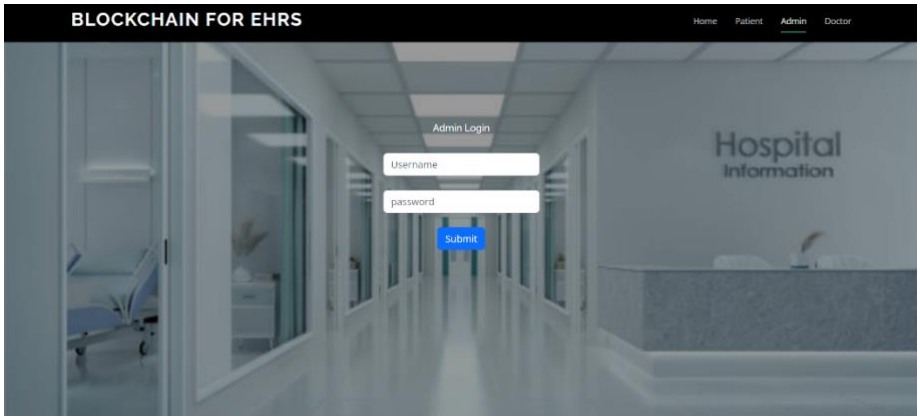


Fig 7.3.5 Admin login page

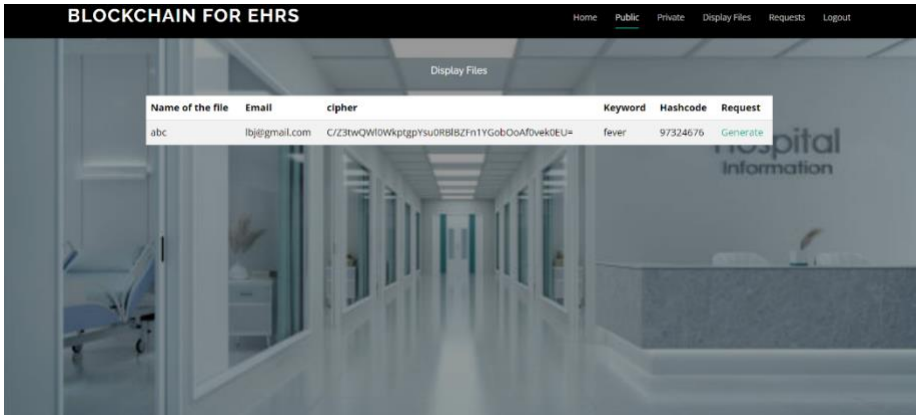


Fig 7.3.6 Admin page

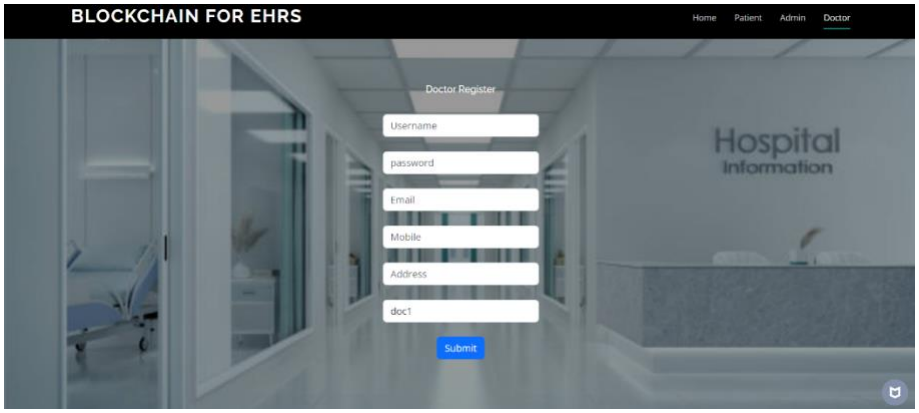


Fig 7.3.7 Doctor registration page

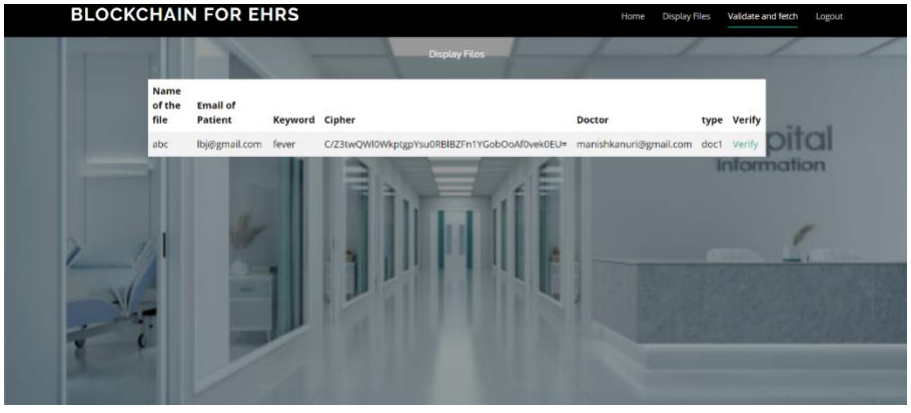


Fig 7.3.8 Document Requested

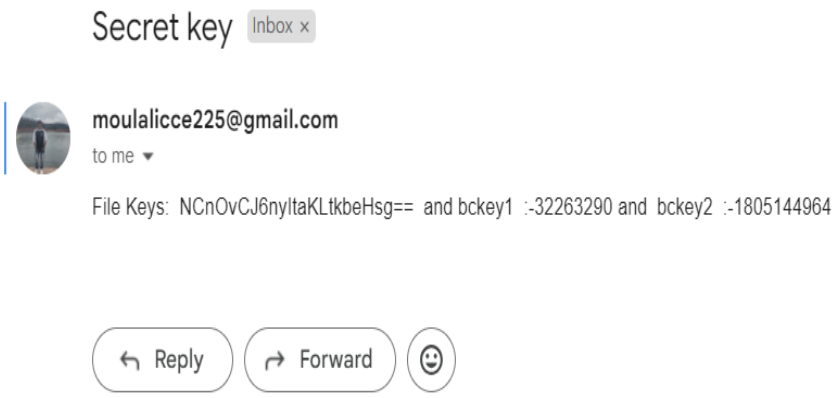


Fig 7.3.9 Block keys and secret key sent to user

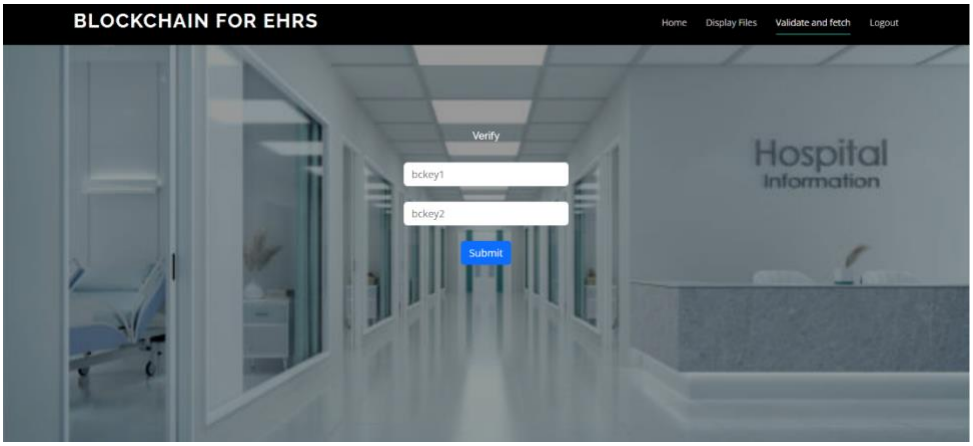


Fig 7.3.10 user verification level 1(block keys)

7. CONCLUSION

In this project, we discussed how blockchain technology can be useful for the healthcare sector and how it can be used for electronic health records. Despite the advancement in the 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.

For the future, 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 define certain policies and rules that comply with the principles of the healthcare sector.

9. FUTURE SCOPE

The future scope of integrating blockchain technology into Electronic Health Record (EHR) systems is promising, offering substantial improvements in data security, integrity, and management within the healthcare sector. As blockchain continues to mature and gain widespread adoption, its application in healthcare is poised for significant advancements.

One key aspect of the future scope is the enhanced security and privacy that blockchain brings to EHR systems. By leveraging cryptographic techniques and decentralized consensus mechanisms, blockchain can ensure data confidentiality and integrity, reducing the risk of unauthorized access and tampering.

Furthermore, the adoption of blockchain in EHR systems paves the way for interoperability among healthcare providers and systems. This interoperability facilitates seamless data exchange while maintaining data security standards, enabling better coordination of patient care and improving overall healthcare outcomes.

Another area of growth is the integration of smart contracts within blockchain-based EHR systems. Smart contracts automate and enforce predefined rules and agreements, streamlining processes such as patient consent management, billing, and insurance claims processing, leading to increased efficiency and reduced administrative overhead.

Additionally, advancements in blockchain scalability solutions, such as off-chain storage techniques and layer-two protocols, address the scalability challenges typically associated with blockchain technology. This scalability enhancement ensures that blockchain-based EHR systems can handle large volumes of data and transaction throughput, meeting the demands of modern healthcare environments.

In conclusion, the future scope of blockchain in EHR systems includes heightened security, improved interoperability, streamlined processes through smart contracts, and scalability enhancements, all contributing to a more efficient, secure, and patient-centric healthcare ecosystem.

10.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] M. Reisman, ‘‘EHRs: The challenge of making electronic data usable and interoperable.,’’ PT, vol. 42, no. 9, pp. 572–575, Sep. 2017.

[7] W. W. Koczkodaj, M. Mazurek, D. Strzałka, A. Wolny-Dominiak, and M. Woodbury-Smith, ‘‘Electronic health record breaches as social indicators,’’ Social Indicators Res., vol. 141, no. 2, pp. 861–871, Jan. 2019.

[8] S. T. Argaw, N. E. Bempong, B. Eshaya-Chauvin, and A. Flahault, ‘‘The state of research on cyberattacks against hospitals and available best practice recommendations: A scoping review,’’ BMC Med. Inform. Decis. Making, vol. 19, no. 1, p. 10, Dec. 2019.

[9] A. McLeod and D. Dolezel, ‘‘Cyber-analytics: Modeling factors associated with healthcare data breaches,’’ Decis. Support Syst., vol. 108, pp. 57–68, Apr. 2018.

[10] L. Coventry and D. Branley, ‘‘Cybersecurity in healthcare: A narrative review of trends, threats and ways forward,’’ Maturitas, vol. 113, pp. 48–52, Jul. 2018.

[11] ‘‘The future of health care cybersecurity,’’ J. Nursing Regulation, vol. 8, no. 4, pp. S29–S31, 2018.

[12] D. Spatar, O. Kok, N. Basoglu, and T. Daim, ‘‘Adoption factors of electronic health record systems,’’ Technol. Soc., vol. 58, Aug. 2019, Art. no. 101144.

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

[14] W. J. Gordon and C. Catalini, ‘‘Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability,’’ Comput. Struct. Biotechnol. J., vol. 16, pp. 224–230, Jan. 2018.

[15] A. Boonstra, A. Versluis, and J. F. J. Vos, ‘‘Implementing electronic health records in hospitals: A systematic literature review,’’ BMC Health Services Res., vol. 14, no. 1, Sep. 2014, Art. no. 370.

[16] T. D. Gunter and N. P. Terry, ‘‘The emergence of national electronic health record architectures in the United States and Australia: Models, costs, and questions,’’ J. Med. Internet Res., vol. 7, no. 1, p. e3, Jan./Mar. 2005.

[17] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, ‘‘An overview of blockchain technology: Architecture, consensus, and future trends,’’ in Proc. IEEE Int. Congr. Big Data (BigData Congr.), Jun. 2017, pp. 557–564.

[18] C. Pirtle and J. Ehrenfeld, ‘‘Blockchain for healthcare: The next generation of medical records?’’ J. Med. Syst., vol. 42, no. 9, p. 172, Sep. 2018.

[19] 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, Jan. 2019.

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

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

[22] 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), Jun. 2018, pp. 108–113.

[23] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, ‘‘Blockchain distributed ledger technologies for biomedical and health care applications,’’ J. Amer. Med. Inform. Assoc., vol. 24, no. 6, pp. 1211–1220, 2017.

[24] M. S. Sahoo and P. K. Baruah, ‘‘HBasechainDB—A scalable blockchain framework on Hadoop ecosystem,’’ in Supercomputing Frontiers. 201