

SECURE CLOUD BASED HEALTH SHARING SYSTEM USING BLOCKCHAIN

Submitted in partial fulfillment of the requirements for the award
of Bachelor of science in Computer Science

By

HARI KUMARAN P (Reg.No.38290022)

AND

ARAVINDHAN D (Reg.No.38290006)



DEPARTMENT OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600119

MARCH 2021.



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

B ONAFID E CERTIF ICATE

This is to certify that this Project Report is the bonafide work HARIKUMARAN P (38290022), ARAVINDHAN D (38290006) who carried out the project entitled SECURE CLOUD BASED HEALTH RECORD SHARING SYSTEM USING BLOCKCHAIN under my / our supervision From November 2020 to April 2021.

Internal Guide

Dr. G. NAGARAJAN., M.E., Ph.D.

Head of the Department

DR.L.LAKSHMANAN M.E.,PH.D

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, HARIKUMARAN P (38290022) hereby declare that the Project Report entitled **SECURE CLOUD BASED HEALTH RECORD SHARING SYSTEM USING BLOCKCHAIN** done by me and my Partner ARAVINDHAN D (38290006) under the guidance of **Dr. G. NAGARAJAN, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of **Bachelor of SCIENCE IN COMPUTER SCIENCE**

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D. Professor & Dean**, School of Computing, **Dr. S. Vigneshwari M.E., Ph.D. and Dr. L. Lakshmanan M.E., Ph.D., Professors & Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G. NAGARAJAN M.E., Ph.D.** Associate Professor, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Bachelor of science in Computer Science** who were helpful in many ways for the completion of the project.

ABSTRACT

Recent years we are witnessing a various technology innovation in Electronic Health Records (EHRs) where mobile devices are integrated with cloud computing to facilitate medical data exchanges among patients and healthcare providers. This advanced model enables healthcare services with low operational cost, high flexibility, security and EHRs availability. In existing system, the patient health information's (PHI) are stored in the hospital database, hence always there is a risk of intrusion i.e., illegal file access, information getting modified by attackers. To overcome this problem, we proposed blockchain technology based EHR, thus providing decentralized system. We design a trustworthy access control mechanism using smart contracts to achieve secure EHRs sharing among different patients and medical providers. In the proposed system, we also integrate crypto currency i.e, ethereum preserving sensitive health information against potential threats. Our proposed system invokes doctor, patient and pharmacist authentications. Patients can register and unique block with hash value would be created. Once the patient conveys the disease to the doctor, the doctor can able to view the respective patient disease and suggest prescribed medicines. The prescribed medicines would be updated to the pharmacist .

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	xi
1	INTRODUCTION	1
2	LITERATURE SURVEY	
	2.1 LITERATURE REVIEW	3
3	METHODOLOGIES AND ALGORITHM USED	
	2.1 EXISTING SYSTEM	6
	2.2 PROPOSED SYSTEM	6
	2.3 SCOPE	6
	2.4 MODULES	7
	2.4.1 REGISTRATION	7
	2.4.2 SMART CONTRACT	7
	CREATION	8
	2.4.3 HASH VALUE CREATION	8
	2.4.4 BLOCK CREATION	9
	2.4.5 PRIVACY PRESERVING	9
	2.4.6 MIDDLEWARE	9
	2.4.7 CRYPTOCURRENCY	9

	2.5 SOFTWARE ENVIRONMENT	18
	2.5.1 JAVA	18
	2.5.2 APPLICATION APPLET	18
	2.5.3 JAVA ARCHITECTURE	19
	2.5.4 COMPILATION OF CODE	19
	2.6 JAVA VIRTUAL MACHINE	20
	2.6.1 SIMPLE	21
	2.6.2 OBJECT ORIENTED	21
	2.6.3 ROBUST	21
	2.7 JAVA SERVER PAGES	21
	2.8 CSS	22
	2.9 SERVLETS	23
	2.9.1 SERVLET ARCHITECTURE	24
	2.10 XML	25
	2.11 IDE	26
	2.12 NETBEANS	27
	2.13 MYSQL	27
	2.13.1 SQL YOG	27
	2.14 GANACHE	28
	2.14.1 REMIX	
4	RESULTS AND DISCUSSION	29
3	CONCLUSION	31
	REFERENCES	32
	APPENDICES	33
	A. CODE IMPLEMENTATION	33
	B. SCREENSHOTS	39

LIST OF FIGURES

FIGURE NO.	NAME OF THE FIGURE	PAGE NO.
314	System Architecture	10
315	Data Flow Diagram – Level 0	11
3.3	Data Flow Diagram – Level 1	11
3.4	Data Flow Diagram – Level 2	12
3.5	Data Flow Diagram – Level 3	12
3.6	Data Flow Diagram – Level 4	13
3.7	Usecase Diagram	14
3.8	Activity Diagram	15
3.9	Sequence Diagram	16
3.10	Class Diagram	17
3.11	Collaboration Diagram	17
3.12	Java Architecture	19
3.13	Servlet Architecture	24
4.1	Doctor seeking permission from user	29
4.2	Doctor accessing data after permission	30

LIST OF ARBBREVATION

EHR	- Electronic Health Records
EVM	- Ethereum Virtual Machine
CGI	- Common Gateway Interface
ABI	- Applications Binary Interface

CHAPTER 1

INTRODUCTION

Recently, there has been a growing interest in employing the blockchain technology to promote medical and e-health services. Blockchain with its decentralized and trustworthy nature has demonstrated immense potentials in various e-health sectors such as secure sharing of Electronic Health Records (EHRs) and data access management among multiple medical entities. Therefore, the adoption of blockchain can provide promising solutions to facilitate healthcare delivery and thus revolutionize the healthcare industry.

With the emergence of innovative technologies, including Mobile Cloud Computing (MCC) and Internet of Medical Things (IoMT), the healthcare industry has witnessed significant changes in e-health operations. Patients now can collect their personal health information at home based on mobile devices (such as smartphones and wearable sensors) and share on cloud environments where healthcare providers can access instantly to analyze medical records and provide timely medical supports. This smart e-health service allows healthcare providers remotely monitor patients and offer ambulatory care at home, which not only facilitates healthcare delivery but also brings economic benefits to patients. Further, the availability of complete EHRs on clouds also helps healthcare providers track patient health and offers proper medical services during diagnosis and treatment processes. Besides all these great advantages, however, the trend of EHRs storage on clouds also poses security challenges which hinder the deployment of e-health applications on clouds. Among such security issues is secure EHRs sharing between patients and healthcare providers on mobile cloud environments. Unauthorized entities may gain malicious access to EHRs without consent of patients, which has detrimental impacts on data integrity, privacy and security of cloud e-health systems. Moreover, patients may find it difficult to track and manage their health records shared among healthcare providers on clouds. It therefore is necessary to propose efficient access control solutions for mobile cloud EHRs sharing systems. Traditional access control approaches for EHRs sharing assume that cloud servers are fully trusted by data owners and enable the servers to perform all access control and authentication

rights on data usage. However, this assumption no longer holds in mobile clouds since the cloud server is honest but curious. The cloud server will honestly perform the data requests, but meanwhile will obtain personal information without consent of users, which leads to serious information leakage issues and network security, accordingly. More importantly, conventional access control systems mainly rely on a predefined point of access, i.e. a centralized cloud server, and this can lead to the central point of failure for e-health networks.

Meanwhile, blockchain-based access control provides various new security features for e-health with great advantages over conventional access control solutions.

CHAPTER 2 LITERATURE

SURVEY

2.1 LITERATURE REVIEW

Blockchain-Based Implementation for Financial Product Management Bihuan

Chen, Zhixiong Tan, Wei Fang

IEEE 2018

In this project, they proposed a blockchain-based financial product management platform. It constructs a distributed network architecture for financial product information management which possess a tamper-resistant, information transparency and secure information sharing environment. The management platform in our paper, using Hyperledger Fabric as underlying architecture, realized the fundamental financial product manage operations. Such as routine maintenance of financial product, multi-function data inquiry and traceability of financial product. As last, considering the characteristics of financial product management, we put forward a follow-up prospect for improving the weaknesses of Hyperledger Fabric to perfect our management platform.

E-Voting using a Decentralized Ethereum Application

V. Arun, Aditya Dutta, Sourav Rajeev, Rohan Varghese Mathew

IJEAT, 2019

As technology progresses with each passing day, its impacts only become more positive. One such outcome is blockchain. It has the potential to revolutionize the process of voting due to its decentralized nature and property of immutability. Voting in most places is a very non-transparent process and is known to be rife with corruption. With the introduction of blockchain as a service in the voting sphere, a potential protocol can be created such that the process of voting is completely open, fair and independently verifiable by anyone. Moreover, this paper elucidates the

potential of a distributed ledger using a case study and aims to highlight the pros/cons of using blockchain architecture as a practical application in the process of voting.

Blockchain Technologies for the Internet of Things: Research Issues and Challenges

Mohamed Amine Ferrag, Makhlof Derdour, Mithun Mukherjee IEEE

2019

In this paper presents a comprehensive survey of the existing blockchain protocols for the Internet of Things (IoT) networks. We start by describing the blockchains and summarizing the existing surveys that deal with blockchain technologies. Then, we provide an overview of the application domains of blockchain technologies in IoT, e.g, Internet of Vehicles, Internet of Energy, Internet of Cloud, Edge computing, etc. Moreover, we provide a classification of threat models, which are considered by blockchain protocols in IoT networks, into five main categories, namely, identity- based attacks, manipulation based attacks, cryptanalytic attacks, reputation-based attacks, and service-based attacks.

A Small Java Application for Learning Blockchain Xing Liu

IEEE 2018

This paper introduces a small Java application named ChainTutor for learning basic Blockchain concepts. Although the term Blockchain is widely known and Blockchain technologies are finding applications in various areas such as banking, healthcare and Internet of Things, some concepts of Blockchain are not easy for beginners to understand. Fully textbased tutorials are often difficult to follow. General picture of Blockchain operations gets lost in lengthy textual descriptions. With the Java application introduced in this paper, users can experiment with key Blockchain

concepts through a graphical user interface. They can generate keys, hashes, transactions, blocks and wallets. They can see the low level details of a blockchain such as encryption keys and hashes. They can see how mining works and how blocks are added to a blockchain.

Blockchain Technology Applied in Health Andrei

Cirstea, Nicu Bizon, Cosmin Stirbu ECAI, 2018

This paper will introduce a minimal introduction to blockchain technology, followed by a medical application (MedBlocks). The application is based on this technology, with the purpose to make medical field more efficient. Blockchain technology has the potential to revolutionize both medicine and any other field for system decentralization. This can change the whole world through efficiency and secure way to act. The objective of this paper is to show the extraordinary potential of this technology and how it will fundamentally change all aspects of receiving, transmitting and securing of information.

CHAPTER 3

METHODOLOGIES AND ALGORITHM USED

41 EXISTING SYSTEM:

The existing system is a centralized distribution system, hence medical data intrusion is always possible. In existing system, there is no data privacy, lack of network security, very less reliability of electronic health record sharing among cloud servers. Also, in the existing system, single point of failure can happen any time thus raising data non-availability. The current healthcare system faces high storage issue and thus leads to much time in data retrieval performance.

42 PROPOSED SYSTEM:

We proposed blockchain technology based EHR, thus providing decentralized system. We design a trustworthy access control mechanism using smart contracts to achieve secure EHRs sharing among different patients and medical providers involving hospitals and pharmacist as well. In the proposed system the patient can register and provide the patient health information which would be converted into a single hash value using SHA 256 algorithm and embed into a QR code. The doctor, hospital can view the patient permitted information using the hash value. The doctor can prescribe the medicine which would be converted into an another block and this block can be viewed by the pharmacist and automatically respective invoice would be prepared.

43 SCOPE:

The use of Blockchain in healthcare will ensure, first, the reliability of the stored data. Reliability is determined by the fact that each record is confirmed from several sources. Secondly, the data will remain unchanged over time. No one will be able to make changes to the record without agreeing with other sources, let alone deleting the data. And thirdly, the proper data security will be ensured, since no one can access them without the consent of the data source. Such qualities make prospects for using Blockchain significant in medicine

3.4 MODULES:

- 1. Registration**
- 2. Smart contract creation**
- 3. Hash value creation**
- 4. Block creation**
- 5. Privacy Preserving**
- 6. Middleware**
- 7. Cryptocurrency**

431 REGISTRATION:

Users of the system (doctors and patients) register their details using a registration module. Registration module collects details of the user. Doctor need to submit the name, mobile number, email id, date of birth, address, and registration number. Every doctor in the system belongs to one of the associated healthcare providers. Patient needs to submit the name, mobile number, email id, date of birth, address and health insurance details. Besides, doctors and patients need to provide a valid password.

432 SMART CONTRACT CREATION:

A smart contract is a kind of self-operating computer program, which can be executed automatically when specific conditions are met. In the Ethereum blockchain, a smart contract is a special account, which contains data and code with multiple programmable functions. Users can use their Ethereum account to interact with smart contracts via application binary interfaces (ABI). Functions defined in smart contracts can be triggered by a new transaction sent from an account. This property allows entities to implement their job functionalities such as data transmission, request handling or access management.

433 HASH VALUE CREATION:

A cryptographic hash is an algorithm that takes an input and turns it into an output of a fixed size. It looks like a mix up of numbers and letters. There are many types of cryptographic hashes. Bitcoin, for example, uses a hashing algorithm called SHA-

256. A hashing algorithm is a computational function that condenses input data into a fixed size, the result of which is the output called a *hash* or a *hash value*. Hashes are used to identify, compare or run calculations against files and strings of data. Typically, the program first computes a hash and then compares the values to the original files.

A basic example of hashing is used to digitally sign a piece of software so that it is available for download. To do this you need a hash of the script of the program you want to download. You also need a digital signature, which gets hashed as well.

Once the input data has been hashed the software is encrypted and it can be downloaded. So, when someone downloads software, the browser needs to decrypt the file and check the two unique hash values. The browser then runs the same hash function, using the same algorithm and hashes both the file and the signature again. If the browser successfully produces the same hash value, it can confirm that both the signature and the file are authentic and that have not been altered.

434 BLOCK CREATION:

Patient has to approve the EHR notification received. EHR verification by the patient avoids double-spending types of attacks. After the successful verification of EHR, the Block Generator module generates a block containing the EHR details. Each block in a blockchain contains the hash of the block data. The EHR is hashed using the key provided by the Key Generator Module.

435 PRIVACY PRESERVING

The solution for data privacy and security could and should very well be blockchain technology. The blocks relate to each other in the form of a chain. The first block of the chain is known as Genesis. Each block consists of a Block Header, Transaction Counter and Transaction. Also the patient data privacy is preserved using patient public address. The patient can share the public address rather than patient details to other members like doctors, hospitals etc.

436 MIDDLEWARE

This dependency is a personal blockchain, which is a local development blockchain that can be used to act as a public blockchain. Ganache is used to deploy smart contracts and for running tests. Here you have a personal blockchain network running. Ganache provides 10 accounts with 100Ether to test our smart contracts on local blockchain bases.

437 CRYPTOCURRENCY

A cryptocurrency is a digital asset that can be used as a medium of exchange. The blockchain technology provides cryptocurrencies with the required level of security needed to withstand attacks on the state of the system and from preventing double spending. When a user sends cryptocurrency, there is no physical exchange of coins. Instead what happens is a signing off of ownership in the sender's address to the receiver's address. In order to be able to access the coins being sent the receiver must hold the private key which unlocks the sender's public key. If the keys match, the transaction is recorded on the blockchain and simultaneously the balance is altered in the sender's and receiver's address. Through cryptocurrency the patient can pay the bills, pharmacy bills, appointment fee, doctor fee etc.

10

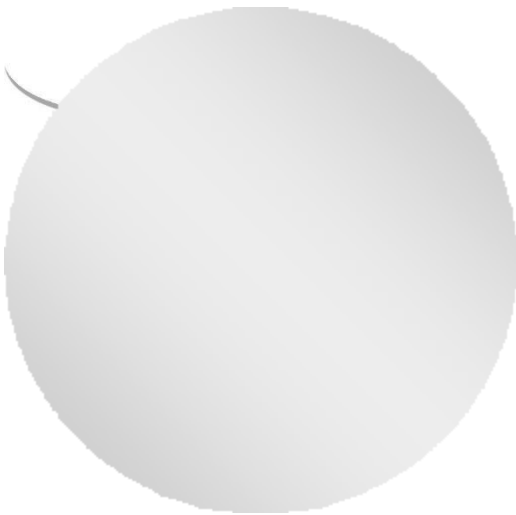
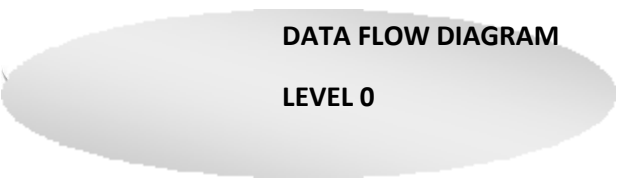
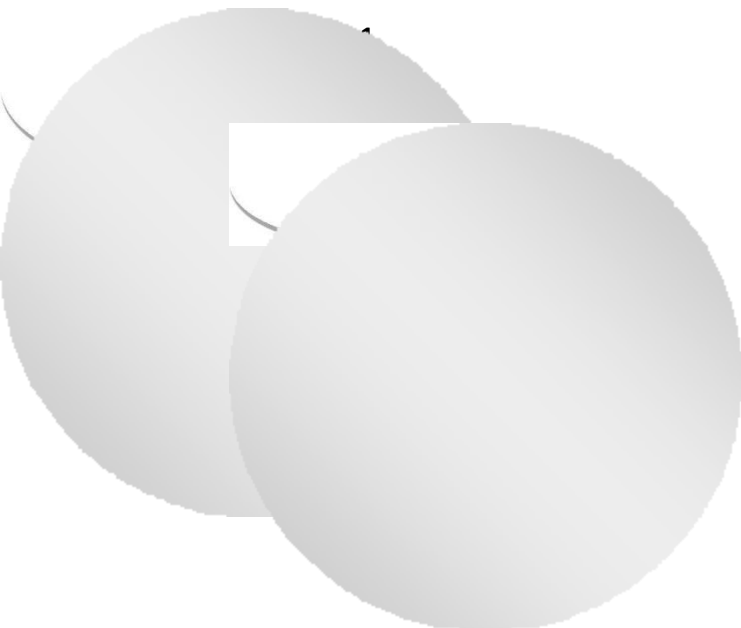


Fig 3.2



g 3.3
 F_i

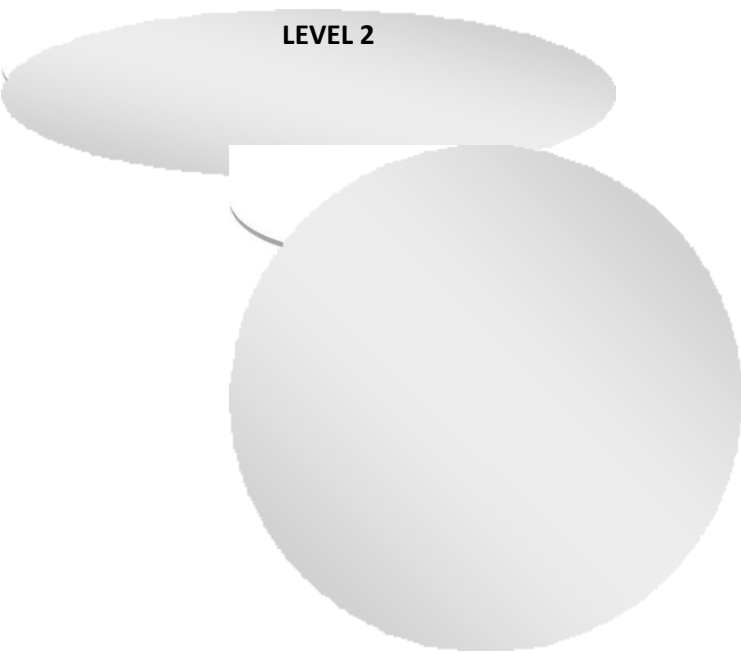


Fig 3.4

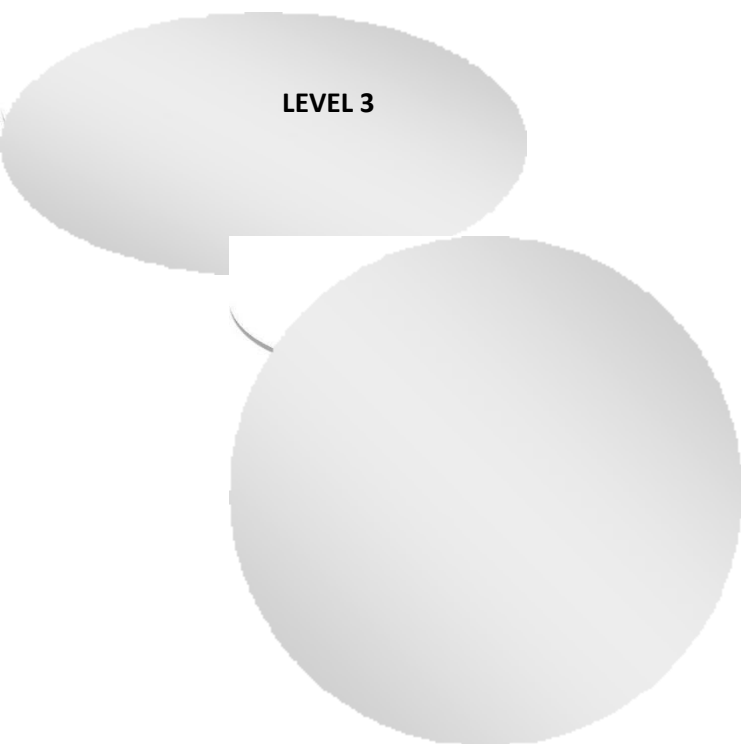




Fig 3.5

LEVEL 4

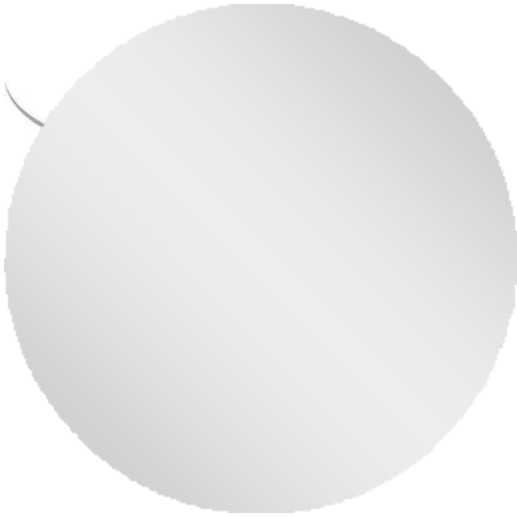


Fig 3.6

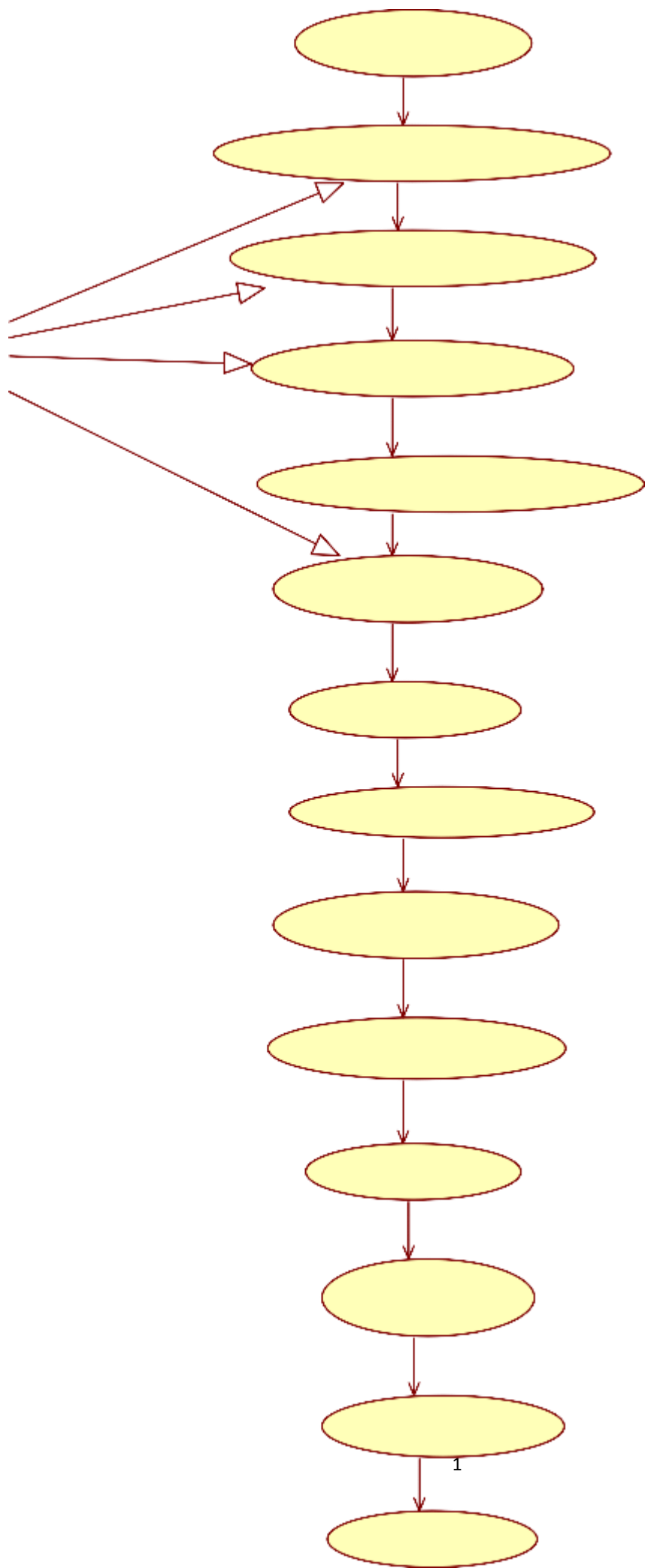


Fig 3.7 USECASE DIAGRAM

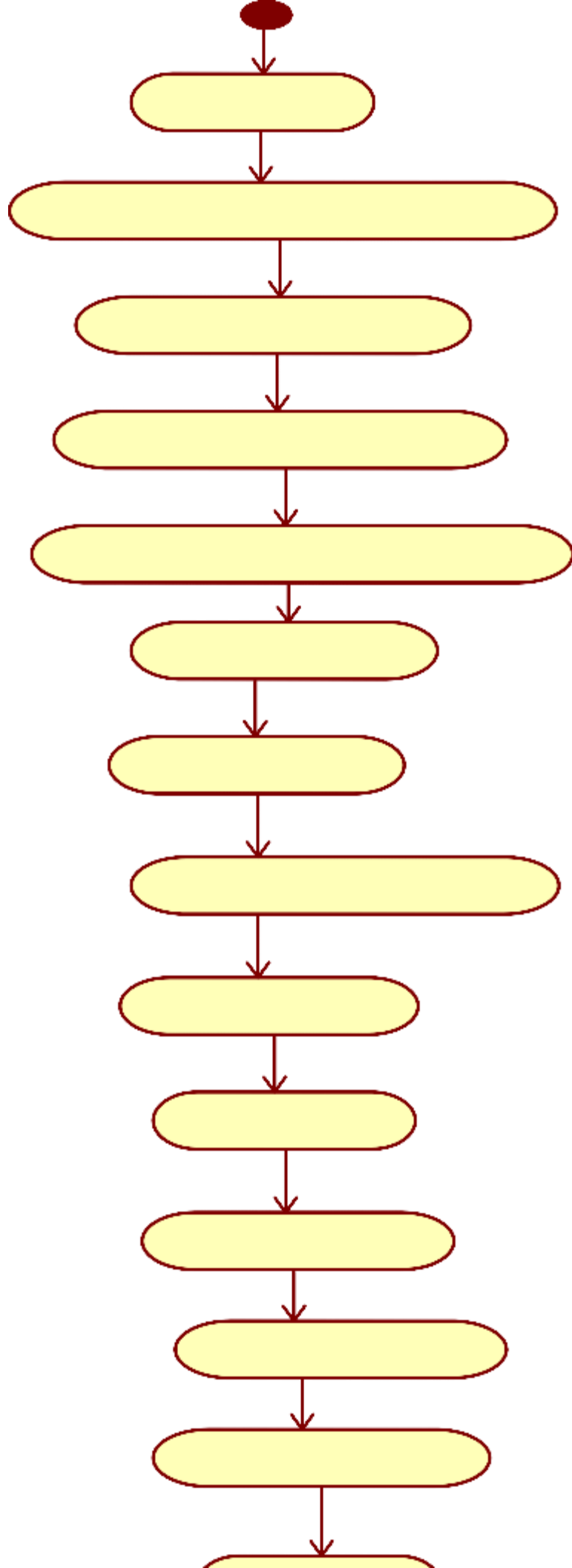


Fig 3.8 ACTIVITY DIAGRAM

SEQUENCE DIAGRAM

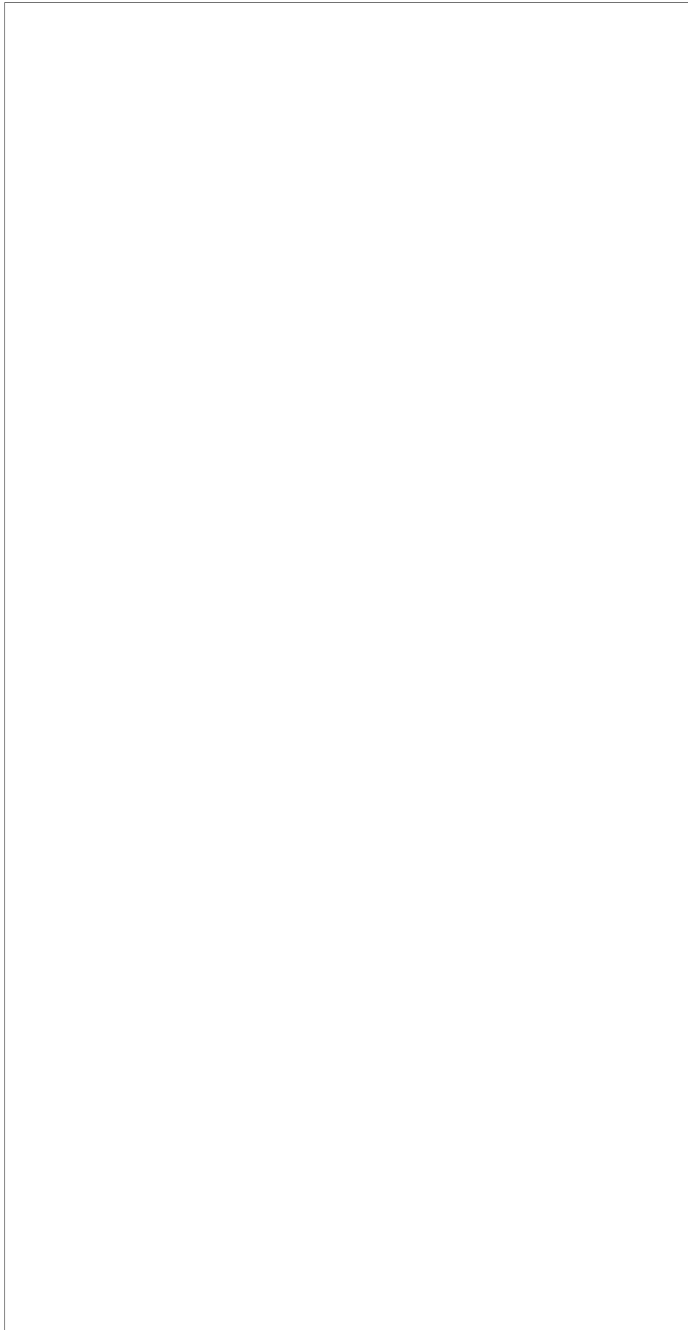


Fig 3.9 SEQUENCE DIAGRAM

CLASS DIAGRAM

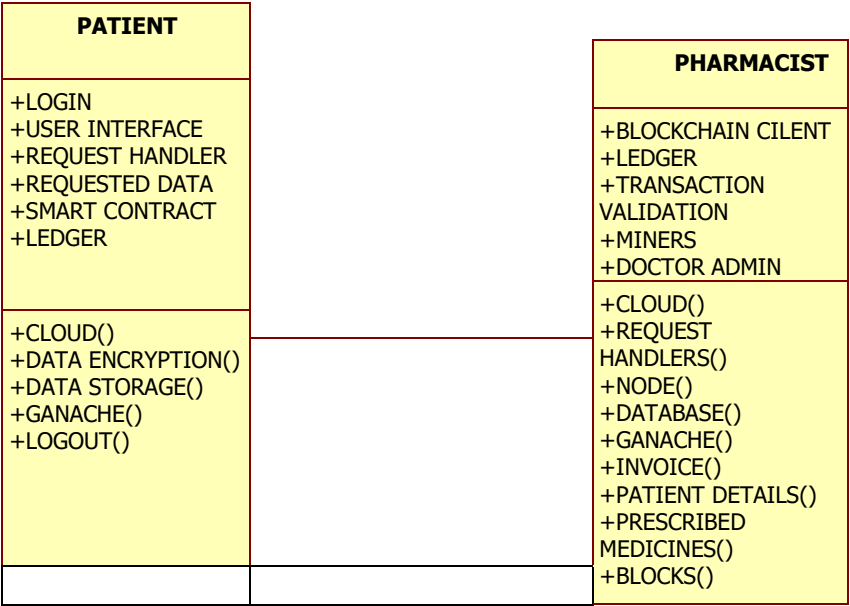


Fig 3.10 CLASS DIAGRAM

COLLABORATION

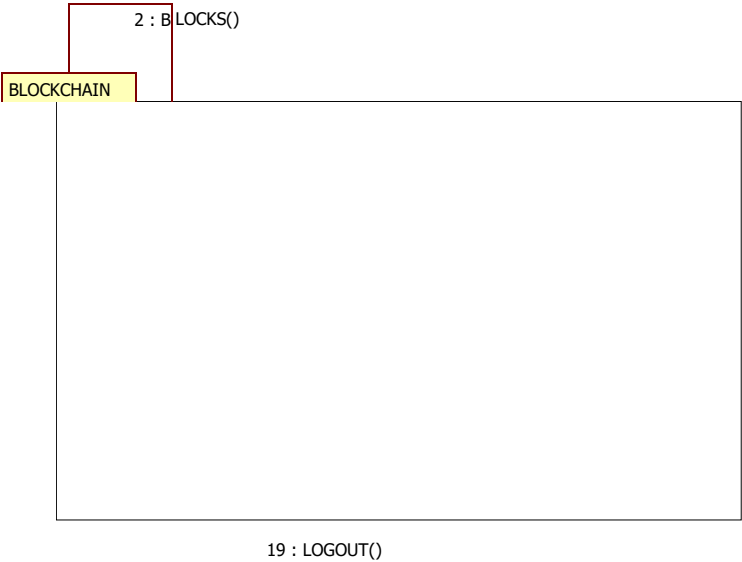


Fig 3.11 COLLABORATION DIAGRAM

44 SOFTWARE ENVIRONMENT:

JAVA:

Initially the language was called as “oak” but it was renamed as “java” in 1995. The primary motivation of this language was the need for a platform- independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

➤ Java is a programmer’s language ➤

Java is cohesive and consistent

➤ Except for those constraints imposed by the Internet environment, Java gives the programmer, full control

Finally Java is to Internet Programming where C was to System Programming.

44.1 Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. In the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

44.2 Applications and applets. An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one created using C or C++. Java’s ability to create Applets makes it important. An Applet is an application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image.

44.3 Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

44.4 Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine(JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability.

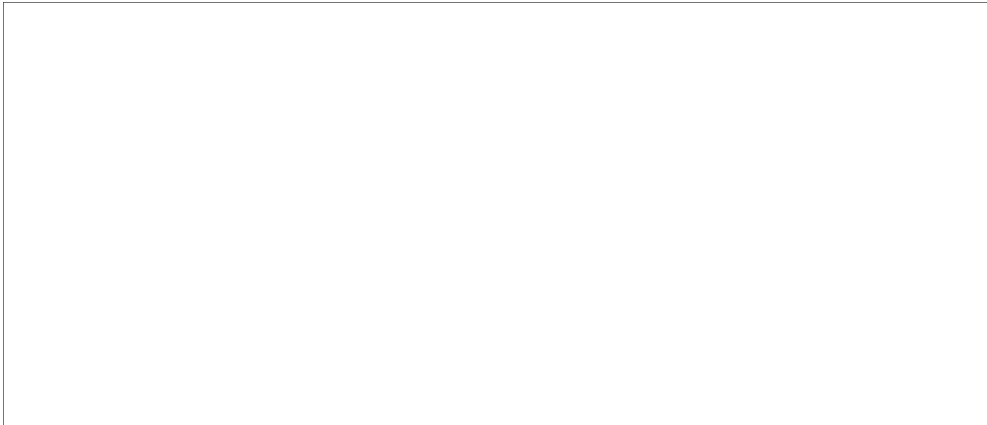


Fig 3.12 Java Architecture

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

46 JVM:

The Java Virtual Machine is the cornerstone of the Java platform. It is the component of the technology responsible for its hardware- and operating system independence, the small size of its compiled code, and its ability to protect users from malicious programs. The Java Virtual Machine is an abstract computing machine. Like a real computing machine, it has an instruction set and manipulates various memory areas at run time. It is reasonably common to implement a programming language using a virtual machine; the best-known virtual machine may be the P-Code machine of UCSD Pascal. The first prototype implementation of the Java Virtual Machine, done at Sun Microsystems, Inc., emulated the Java Virtual Machine instruction set in software

Hosted by a handheld device that resembled a contemporary Personal Digital Assistant (PDA). Oracle's current implementations emulate the Java Virtual Machine on mobile, desktop and server devices, but the Java Virtual Machine does not assume any particular implementation technology, host hardware, or host operating system. It is not inherently interpreted, but can just as well be implemented by compiling its instruction set to that of a silicon CPU. It may also be implemented in microcode or directly in silicon. The Java Virtual Machine knows nothing of the Java programming language, only of a particular binary format, the class file format. A class file contains Java Virtual Machine instructions (or *byte codes*) and a symbol table, as well as other Ancillary information. For the sake of security, the Java Virtual Machine imposes strong syntactic and structural constraints on the code in a class file. However, any language with Functionality that can be expressed in terms of a valid class file can be hosted by the Java Virtual Machine. Attracted by a generally available, machine-independent platform, implementers of other languages can turn to the Java Virtual Machine as a delivery vehicle for their languages. The Java Virtual Machine specified here is compatible with the Java SE 7 platform, and supports the Java programming language specified in *The Java Language Specification, Java SE 7 Edition*.

3.6 .1 Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++ . Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

3.6 .2 Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high- performance non-objects.

3.6 .3 Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

47 JSP:

The most significant of the many good reasons for this is that it is amazingly easy to develop sophisticated Web sites with JSPs. Anyone who can write HTML can quickly create rich, dynamic, and responsive Web sites that enable users to get the most out of their online time. Through a mechanism called JavaBeans, JSPs have made it possible for large teams or individuals working on complex projects to divide the work in such a way as to make each piece simple and manageable, without sacrificing any power. JSPs also provide a great deal of flexibility when generating HTML, through the ability to create HTML-like custom tags.

In addition to this fundamental ease of development, high-quality JSP tools are readily available and easy to use. Developers do not need to buy expensive software or commit to a particular operating system in order to use JSPs. The CD-ROM accompanying this book contains everything a JSP author needs to get started, and the tools are powerful enough to serve even a mid-sized Web site without problems. These free, open-source tools are stable and secure and run on nearly every platform. Of course, high-quality commercial JSP tools are available as well, suitable for serving even the most complex and high-traffic Web sites. Although JSPs have been useful and powerful since the beginning, this is an especially exciting time to be a JSP developer. The recently released version 2.0 of the JSP Specification provides even more features that simplify the process of creating Web sites. In addition, a standard tag library that provides many JSP tags that solve a wide range of common problems has been released. Finally, in the time since they were released, a number of best practices for using JSPs have emerged. This book covers all the topics: the basic powerful features of the JSP specification, the improvements introduced with version 2.0, as well as the new standard tag library and all the things it does. In addition, this book discusses how best to use these tools, based on real-world experiences. However, before we get into all the fun, let's take a look back at how the Web has evolved. This will highlight the kinds of problems that Web authors have faced since the 12 beginning. Once this is understood, it will be clear how JSPs solve these problems and make page creation so easy.

3.8 CSS:

While (X)HTML is used to describe the content in a web page, it is Cascading Style Sheets (CSS) that describe how you want that content to *look*. In the web design biz, the way the page looks is known as its presentation. CSS is now the official and standard mechanism for formatting text and page layouts. CSS also provides methods for controlling how documents will be presented in media other than the traditional browser on a screen, such as in print and on handheld devices. It also has rules for specifying the non-visual presentation of documents, such as how they will sound when read by a screen reader. Style sheets are also a great

tool for automating production, because you can make changes to all the pages in your site by editing a single style sheet document. Style sheets are supported to some degree by all modern browsers.

Although it is possible to publish web pages using (X)HTML alone, you'll probably want to take on style sheets so you're not stuck with the browser's default styles. If you're looking into designing web sites professionally, proficiency at style sheets is mandatory.

3.9 SERVLETS:

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI. Performance is significantly better. Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request. Servlets are platform-independent because they are written in Java. Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted. The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

3.9 .1 SER VLET A RCH IT ECTU RE :

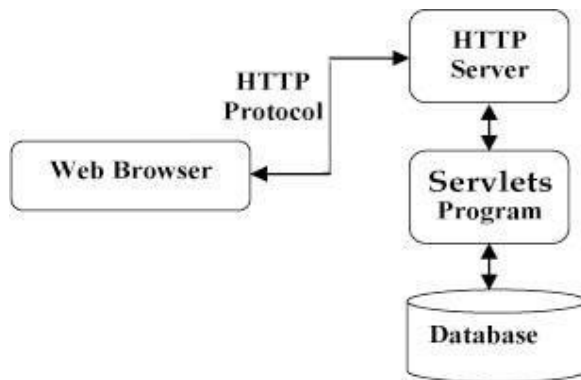


Fig 3.13 Servlet Architecture

Servlets perform the following major tasks:

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
- Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.

- Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

410 XML:

XML stands for **Extensible Markup Language**. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions:

- **XML is extensible:** XML allows you to create your own self-descriptive tags, or language, that suits your application.
- **XML carries the data, does not present it:** XML allows you to store the data irrespective of how it will be presented.
- **XML is a public standard:** XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

A short list of XML usage says it all:

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.

- XML can easily be merged with style sheets to create almost any desired output.

411 IDE:

An **integrated development environment (IDE)** or **interactive development environment** is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs offer intelligent code completion feature. Some IDEs contain a compiler, interpreter, or both, such as Net Beans and Eclipse; others do not, such as Sharp Develop and Lazarus. The boundary between an integrated development environment and other parts of the broader *software development environment* is not well-defined. Sometimes a version control system and various tools are integrated to simplify the construction of a Graphical User Interface (GUI). Many modern IDEs also have a class browser, an object browser, and class diagram, for use in object-oriented software development.

412 NET BEANS:

Net Beans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyzers and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, Net Beans IDE sets the standard for developing with cutting edge technologies out of the box.

413 MY SQL:

MySQL is (as of March 2014) the world's second most widely used open- source relational database management system (RDBMS). It is named after co- founder Michael Widenius's daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project have made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, Word Press, phpBB, MyBB, Drupal and other software. MySQL is also used in many high- profile, large-scale websites, including Google (though not for searches), Face book, Twitter, Flickr, and YouTube.

3.13.1 SQL YOG:

SQLyog is a GUI tool for the RDBMS MySQL. It is developed by Webyog, Inc. based out of Bangalore, India and Santa Clara, California. SQLyog is being used by more than 30,000 customers worldwide and has been downloaded more than 2,000,000 times.

3.14 GANACHE:

Ganache, previously Testrpc, is a virtual blockchain which sets up 10 default Etheruem addresses, complete with private keys and all, and pre-loads them with 100 simulated Ether each. There is no "mining" per-se with Ganache - instead, it immediately confirms any transaction coming its way. This makes iterative development possible - you can write unit tests for your code which execute on

this simulated blockchain, deploy smart contracts, play around, call functions, and then tear it all down for further simulation or new tests, returning all addresses to their initial state of 100 Ether.

3.14.1 Remix:

The browser based IDE for writing and deploying Solidity code for the Ethereum blockchain - is fine and all, but it can be slow to develop on. It is a browser based system and as such, tooling is limited.

When working with EVM (Ethereum Virtual Machine) code that's going to end up in production, you're going to need inheritance, tests, auto-completion, helper tools like linters and validators and much more.

CHAPTER 4 RESULTS

AND DISCUSSION

To implement our EHRs sharing framework, we first deployed a private Ethereum blockchain. Data access and transactions are recorded and shown on the web interface. Based on blockchain settings, we deployed smart contracts, IPFS storage, established network entities and connected with mobile applications to build our e-health framework. With these settings, we operated the EHRs sharing system and evaluated the efficiency of our design through two main performance metrics: access control and network overheads. We present two use cases with authorized and unauthorized access to evaluate the performance of our EHRs sharing model with a designed access control. The objective of our framework is to allow authorized entities (such as healthcare providers) to retrieve effectively EHRs on cloud, while being able to prevent unauthorized access to our EHRs resources. A mobile user such as a doctor, who wants to access EHRs of his patient on cloud, can use our mobile application with a mobile user interface to create an Ethereum account and register user information for interacting with the blockchain. After his request is verified by the cloud EHRs manager, he now starts to make a transaction to access EHRs by providing the address of his patient.

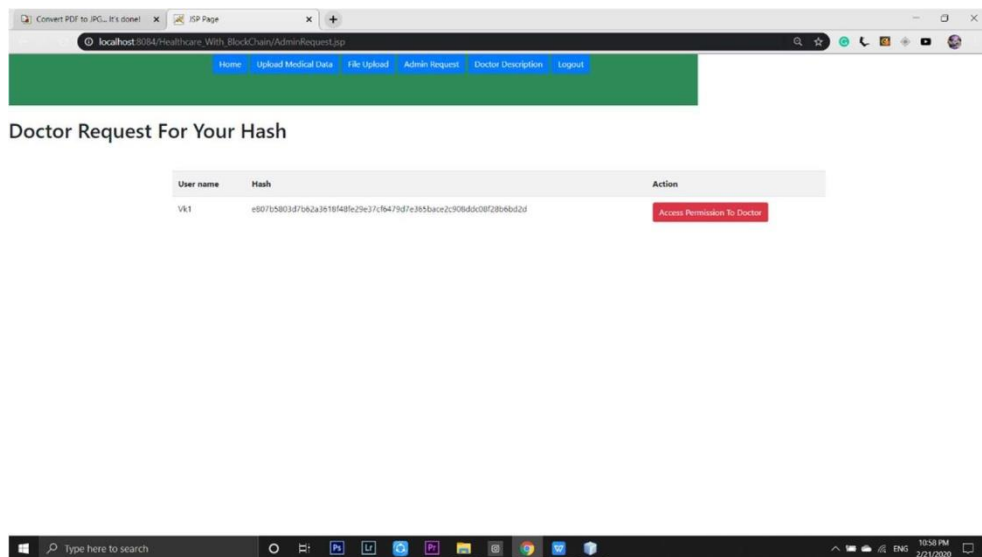


Fig 4.1 Doctor Requesting Patient to view Medical Data

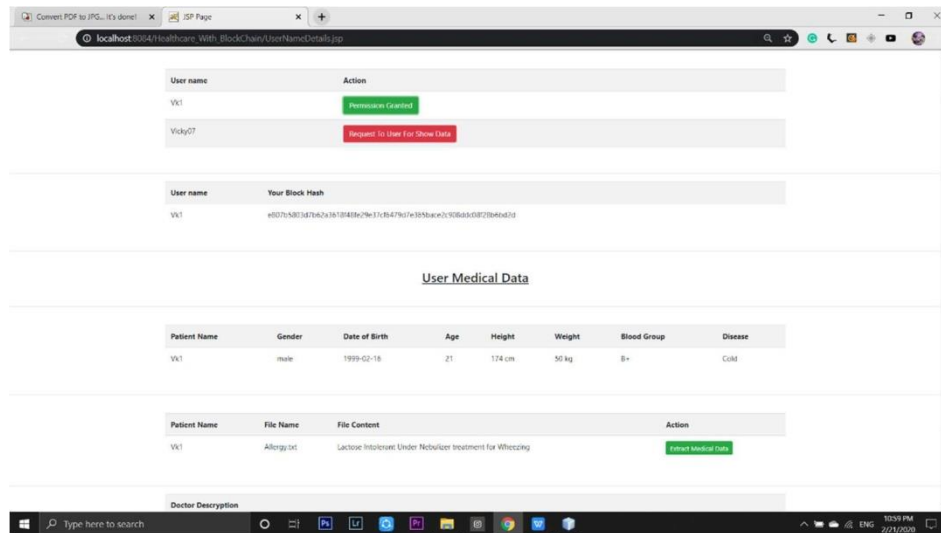


Fig 4.2 Doctor accessing data after permission granted

Our EHRs system will then return data access results which are also updated on his mobile interface. Thus, the doctor can acquire all medical records of his patient to analyse and provide healthcare supports accordingly. Now the EHRs access process is finished and transaction is appended to blockchain by the cloud miner and broadcast to all entities in the network. Therefore, a patient can keep track of the exchange of their EHRs and know exactly who uses their data, thus ensuring data ownership of individuals and network trustworthiness.

CHAPTER 5

CONCLUSION:

In this project, we identify critical challenges of current EHRs sharing systems and propose efficient solutions to address these issues through a real prototype implementation. In this work, our focus is on designing a trustworthy access control mechanism based on a single smart contract to manage user access for ensuring efficient and secure EHRs sharing. Our access control can identify and prevent effectively unauthorized access to the e-health system, aiming for achieving a desired level of patient privacy and network security.

REFERENCE:

- [1] Kuo TT, Kim HE, and Ohno-Machado L, "Blockchain distributed ledger technologies for biomedical and health care applications," *Ame. Medi. Infor. Assoc. J.*, vol. 6, pp. 1211-1220, 2017.
- [2] Matthias Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. 18th IEEE Int. Conf e-Health Net., Appli. and Ser*, Sept 2016.
- [3] Gordon W and Catalini, "Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability," *Comput Struct Biotechnol J.*, pp. 224- 230, 2018.
- [4] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, and Fusheng Wang "Secure and Trustable Electronic Medical Records SharingusingBlockchain,"in*Proc.AMIAAnnuSymp*,pp.650-659,2017.
- [5] Marko Holbl et al., "A Systematic Review of the Use of Blockchain in Healthcare," *Symmetry*, 2018.
- [6] Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang, Mingyu Ma,Jianfei He, "BloCHIE: A BLOCKchain-Based Platform for Healthcare Information Exchange," in *Proc. IEEE on Smart Compu. (SMARTCOMP)*, 2018.
- [7] Lo.ai A. Tawalbeh, Rashid Mehmood, Elhadj Benkhelifa, and Houbing Song,"*MobileCloudComputingModelandBigDataAnalysisforHealthcare Applications*," *IEEE Access*, vol. 4, pp. 6171-6180, 2016.
- [8] S. M. Riazul Islam, Daehan Kwak, MD. Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678-708, 2015.
- [9] Bahga A, and Madiseti VK "A Cloud-based Approach for Interoperable Electronic Health Records (EHRs)," *IEEE J Biomed Health Inform.*, pp. 894-906, 2013.
- [10] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, "Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability," *Comput Struct Biotechnol J.*, 2010.

APPENDICES

CODE IMPLEMENTATIONI:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

<link href="css/bootstrap.min.css" rel="stylesheet" type="text/css"/>

<script src="js/bootstrap.min.js" type="text/javascript"></script>

<script src="js/bootstrap.bundle.min.js" type="text/javascript"></script>

<script src="public/js/analytics.js"></script>

<script src="public/js/jquery-3.2.1.min.js"></script>

<script src="public/js/truffle-contract.js"></script>

<script src="public/js/web3.min.js"></script>

<script src="public/js/truffle-config.js"></script>

<script src="public/js/app.js" type="text/javascript"></script>

<script src="public/js/truffle-config.js" type="text/javascript"></script>

</head>

<body>

<div style="width: 1350px;height: 100px;background-color: lightblue">

<div style="margin-left: 500px;">

<a href="index.html"><button class="btn btn-secondary">Home</button></a>

<a href="SuperAdminLogin.jsp"><button class="btn btn-secondary">Super Admin</button></a>

<a class="active" href="AdminLogin.jsp"><button class="btn btnsecondary">Doctor</button></a>

<a href="UserLogin.jsp"><button class="btn btn-secondary">Patient</button></a>

</div>

</div>
```

```

<br>

<!--      <h3>You have <span class="black"><span id="balance"></span> META</span></h3>-->

<h4>Super Admin Login Page</h4>

<center>

<form action="SuperAdminLogin" method="post">

<table class="table table-hover col-6">

<tr><td>User name</td><td><input class="form-control" type="text" name="n"></td></tr>

<tr><td>Password</td><td><input class="form-control" type="password" name="p"></td></tr>

<!--      <tr><td></td><td><input class="btn btn-success" type="submit"
value="Login"></td></tr>-->

<tr><td></td><td><input id="send" class="btn btn-success" type="submit" value="Login"></td></tr>

</table>

</form>

</center>

</body>

</html>

```

```

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

```

```

/**
 *
 *
 */

public class AdminLogin extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            String n = request.getParameter("n");

            String p = request.getParameter("p");

            Connection con = null;

            PreparedStatement ps = null;

            String has = null;

            String hash = "hash";

```

```

try {

    if ("Admin".equals(n) && "Admin".equals(p)) {

        Class.forName("com.mysql.jdbc.Driver").newInstance();

        con = (Connection)
        DriverManager.getConnection("jdbc:mysql://localhost:3306/healthcareb_ lockchain" , "root", "root");

        ps = con.prepareStatement("select * from admin where name=? ");

        ps.setString(1, n);

        ResultSet rs = ps.executeQuery();

        while (rs.next()) {

            System.out.println(" rs.getString " + rs.getString("hashdata")); has

            = rs.getString("hashdata");

        }

        if (has !=null && has.contains(hash)) {

            request.getRequestDispatcher("AdminHome.jsp").forward(request, response);

        } else {

            System.out.println("has---> " + has);

            NoobChain block = new NoobChain(); String

            x = block.block(n); System.out.println("Admin

            Block----> " + x); HttpSession s =

            request.getSession(); s.setAttribute("admin",

            x);

            ps = con.prepareStatement("insert into admin (name,hashdata) values (?,?)");

            ps.setString(1, n);

            ps.setString(2, x);

            int ii = ps.executeUpdate();

            request.getRequestDispatcher("AdminHome.jsp").forward(request, response);

        }

    } else {

```

```

        request.getRequestDispatcher("AdminLogin.jsp").forward(request, response);
    }

    } catch (Exception e) {
        System.out.println(e);
    }

}

}

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *

```

```

* @param request servlet request

* @param response servlet response

* @throws ServletException if a servlet-specific error occurs

* @throws IOException if an I/O error occurs

*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

    throws ServletException, IOException {

    processRequest(request, response);

}

/**

* Returns a short description of the servlet.

*

* @return a String containing servlet description

*/

@Override

public String getServletInfo() { return

    "Short description";

} // </editor-fold>

```

B.SCREENSHOTS:

The first screenshot shows a web application interface with two main sections. The top section, titled 'User name' and 'Action', displays a table with two rows: 'Vk1' with a green 'Permission Granted' button, and 'Vicky07' with a red 'Request To User For Show Data' button. The bottom section, titled 'User name' and 'Your Block Hash', displays a table with one row: 'Vk1' with a long hexadecimal hash. Below this is a section titled 'User Medical Data' containing two tables. The first table has columns: Patient Name, Gender, Date of Birth, Age, Height, Weight, Blood Group, and Disease. The second table has columns: Patient Name, File Name, File Content, and Action. The bottom section is titled 'Doctor Description' and contains a text area with a green 'Send Description' button.

User name	Action
Vk1	Permission Granted
Vicky07	Request To User For Show Data

User name	Your Block Hash
Vk1	e807b5803d7b62a3618f48fc29e37c1f5479d7e365bacc2c908dcd08f28b6bd2d

User Medical Data

Patient Name	Gender	Date of Birth	Age	Height	Weight	Blood Group	Disease
Vk1	male	1999-02-16	21	174 cm	50 kg	B+	Cold

Patient Name	File Name	File Content	Action
Vk1	Allergy.txt	Lactose Intolerant Under Nebulizer treatment for Wheezing	Extract Medical Data

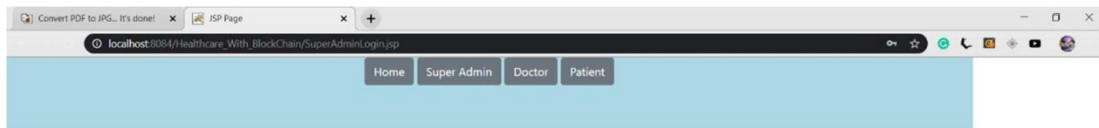
Doctor Description

Deryphilin 150 mg - 4 Nos. (1-0-1)
Montec LC 150 mg - 2 Noa. (0-0-1)

Send Description

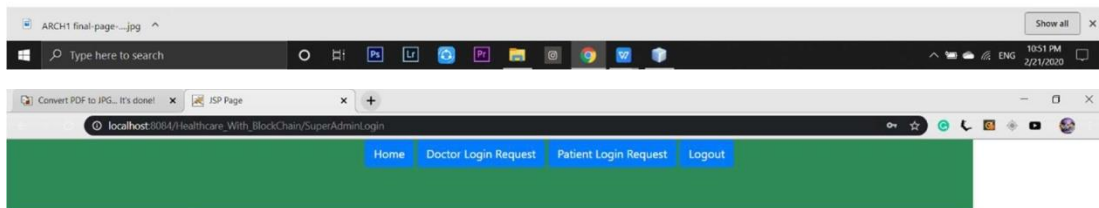
The second screenshot is identical to the first, but the 'Doctor Description' section is expanded, showing a text area with the following text: 'Deryphilin 150 mg - 4 Nos. (1-0-1)' and 'Montec LC 150 mg - 2 Noa. (0-0-1)'. The 'Send Description' button is still visible below the text area.

Accessing / Viewing Patient data



Super Admin Login Page

User name	Admin
Password	*****
Login	



Admin Home Page

User name	Hash	Previous Hash
Admin	4d30dfb5573795a16dfd27570d4f98f2a363b8647182eb2b24f0e172ddc66814	0

You have META



Admin login and Homepage



Doctor Login Form

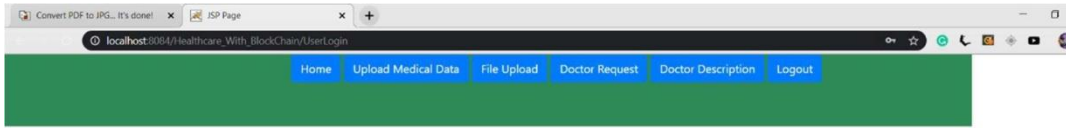
User name	
Password	
DoctorLogin Create New Account	

You have META

Doctor Login Form

UserName:	
Password:	
Doctor's Name:	
Hospital Name:	
National Registration No:	
Date Of Birth:	mm/dd/yyyy
Gender:	--select--
Nationality:	
Address:	
Contact No:	
Email Id:	
Specification:	
Year Of Experience:	
Sign Up	

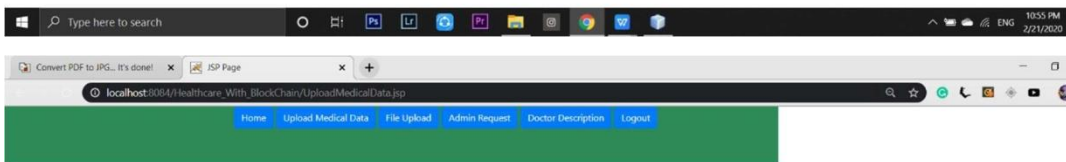
Doctor Login Pages



User Home Page

You have META

User name	Hash	Previous Hash
Vk1	e807b5803d7b62a3618f48fe29e37cf6479d7e365bace2c908ddc08f28b6bd2d	4d30dfb5573795a16dfd27570d4f98f2a363b8647182eb2b24f0e172ddc66814



Upload Medical Data

Your Block Hash e807b5803d7b62a3618f48fe29e37cf6479d7e365bace2c908ddc08f28b6bd2d

User Name	Vk1
Gender	Male
Date of Birth	02/16/1999
Age	21
Height	174 cm
Weight	50 kg
Blood Group	B+
Disease	Cold
Upload	



Fig 5.7 & Fig 5.8 User Login and Data Uploader pages

