

MULTI SECURED MEDICAL DATA USING BLOCKCHAIN WITH CRYPTO AND STEGANO SYSTEM

Submitted in partial fulfilment of the required for the award of

Bachelor of Computer Science

By

J. AUGUSTO SMARTSON (40290016)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

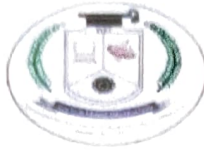
INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC|
Approved by AICTE

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

MAY 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AUGUSTO SMARTSON J (40290016)** who carried out the project entitled **"MULTI SECURED MEDICAL DATA USING BLOCKCHAIN WITH CRYPTO AND STEGANO SYSTEM"** under our supervision from JANUARY 2023 TO MAY 2023.

S. Rajeshwary.
Internal Guide

Ms. RAJESHWARY.S M.Sc., Assistant professor

Rekha Chakravarthi
DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on 08-05-2023

6/8/2023
Internal Examiner

B. Vishwajay
External Examiner

DECLARATION

I am, **AUGUSTO SMARTSON J (40290016)** here by declare that the Project Report entitled **“MULTI SECURED MEDICAL DATA USING BLOCKCHAIN WITH CRYPTO AND STEGANO SYSTEM”** done by me under the guidance of **Ms. RAJESHWARY.S , M.Sc., Assistant Professor,** Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY,** Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.



DATE:

AUGUSTO SMARTSON J

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

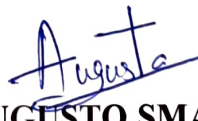
ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** 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 **MS. RAJESHWARY.S M.Sc., Assistant Professor** Department of Computer Science for her 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 member of the Department of Computer Science who were helpful in many ways for the completion of the project.



AUGUSTO SMARTSON J

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	1
1	INTRODUCTION	2
	1.1 DOMAIN INTRODUCTION	2
	1.2 PROJECT INTRODUCTION	5
2	LITERATURE SURVEY	6
3	OBJECTIVE	10
	3.1 EXISTING SYSTEM	11
	3.2 PROPOSE SYSTEM	11
4	PROJECT DESCRIPTION	12
	4.1 ARCHITECTURE DIAGRAM	12
	4.2 USE CASE DIAGRAMS	13
	4.3 SEQUENCE DIAGRAM	14
5	SYSTEM SPECIFICATION	15
	5.1 HARDWARE ENVIRONMENT	15
	5.2 SOFTWARE ENVIRONMENT	15
	5.3 N-TIER ARCHITECTURE	19
6	MODULES	23
7	SYSTEM TESTING	25
	7.1 TAXONOMY OF TESTING	25
	7.2 TESTING IN PARTICULAR	26
8	APPENDIX	27
	8.1 CODING	27
	8.2 OUTPUT SCREENSHOT	49
9	CONCLUSION	58
	REFERENCES	59

ABSTRACT

In the existing system, hospital details are maintained as a hardcopy like report. In most of hospital records are maintained as a hard copy due to huge volume and complexity, it is difficult to manage those data sets using traditional software and hardware . In this paper we propose a efficient storage system for patient information.

Through this system we providing a security system for patient and hospital details. To secure the record we are implementing an efficient system by crypto stegno system. Patient details are stored in cloud for memory management and it will easy way access the data. We are providing permission key to access the patient details. No one can access the information without the knowledge of doctor. So through this a security level is increased by storing data using crypto stenography.

CHAPTER 1

INTRODUCTION

1.1 DOMAIN INTRODUCTION:

Cloud computing is a fast-growing technology that has established itself in the next generation of IT industry and business. Cloud computing promises reliable software, hardware, and IaaS delivered over the Internet and remote data centers. Cloud services have become a powerful architecture to perform complex large-scale computing tasks and span a range of IT functions from storage and computation to database and application services. The need to store, process, and analyze large amounts of datasets has driven many organizations and individuals to adopt cloud computing. A large number of scientific applications for extensive experiments are currently deployed in the cloud and may continue to increase because of the lack of available computing facilities in local servers, reduced capital costs, and increasing volume of data produced and consumed by the experiments.

Cloud computing “is a model for allowing ubiquitous, convenient, and on-demand network access to a number of configured computing resources (e.g., networks, server, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. Cloud computing has a number of favorable aspects to address the rapid growth of economies and technological barriers. Cloud computing provides total cost of ownership and allows organizations to focus on the core business without worrying about issues, such as infrastructure, flexibility, and availability of resources. Moreover, combining the cloud computing utility model and a rich set of computations, infrastructures, and storage cloud services offers a highly attractive environment where scientists can perform their experiments. Cloud service models typically consist of PaaS, SaaS, and IaaS.

PaaS, such as Google's Apps Engine, Salesforce.com, Force platform, and Microsoft Azure, refers to different resources operating on a cloud to provide platform computing for end users.

SaaS, such as Google Docs, Gmail, Salesforce.com, and Online Payroll, refers to applications operating on a remote cloud infrastructure offered by the cloud provider as services that can be accessed through the Internet.

IaaS, such as Flexiscale and Amazon's EC2, refers to hardware equipment operating on a cloud provided by service providers and used by end users upon demand.

Cloud Setup

Cloud Service Provider will contain the large amount of data in their Data Storage. Also the Cloud Service provider will maintain the all the User information to authenticate the User when are login into their account. The User information will be stored in the Database of the Cloud Service Provider. Also the Cloud Server will redirect the User requested job to the Resource Assigning Module to process the User requested Job. The Request of all the Users will process by the Resource Assigning Module. To communicate with the Client and the with the other modules of the Cloud Network, the Cloud Server will establish connection between them. For this Purpose we are going to create an User Interface Frame. Also the Cloud Service Provider will send the User Job request to the Resource Assign Module in First in First out (FIFO) manner.

Private Cloud

Private clouds are dedicated to one organization and do not share physical resources. The resource can be provided in-house or externally. A typical underlying requirement of private cloud deployments are security requirements and regulations that need a strict separation of an organization's data storage and processing from accidental or malicious access through shared resources. Private cloud setups are challenging since the economical advantages of scale are usually not achievable within most projects and organizations despite the utilization of industry standards.

Public Cloud

Public clouds share physical resources for data transfers, storage, and processing. However, customers have private visualized computing environments and isolated storage. Security concerns, which entice a few to adopt private clouds or custom deployments, are for the vast majority of customers and projects irrelevant. Visualization makes access to other customers' data extremely difficult.

Hybrid Cloud

The hybrid cloud architecture merges private and public cloud deployments. This is often an attempt to achieve security and elasticity, or provide cheaper base load and burst capabilities. Some organizations experience short periods of extremely high loads, e.g. as a result of seasonality like black Friday for retail, or marketing events like sponsoring a popular TV event. These events can have huge economic impact to organizations if they are serviced poorly. The hybrid cloud provides the opportunity to serve the base load with in-house services and rent for a short period a multiple of the resources to service the extreme demand. This requires a great deal of operational ability in the organization to seamlessly scale between the private and public cloud.

BLOCK CHAIN INTRODUCTION:

BLOCK CHAIN:

The block chain is an undeniably ingenious invention – the brainchild of a person or group of people known by the pseudonym, Satoshi Nakamoto. But since then, it has evolved into something greater, and the main question every single person is asking is: What is Block chain?

By allowing digital information to be distributed but not copied, block chain technology created the backbone of a new type of internet. Originally devised for the digital currency, Bitcoin, (Buy Bitcoin) the tech community is now finding other potential uses for the technology.

Bitcoin has been called “digital gold,” and for a good reason. To date, the total value of the currency is close to \$112 billion US. And block chains can make other types of digital value. Like the internet (or your car), you don’t need to know how the block chain works to use it. However, having a basic knowledge of this new technology shows why it’s considered revolutionary. So, we hope you enjoy this, What Is Block chain Guide. And if you already know what block chain is and want to become a block chain developer.

1.2 PROJECT INTRODUCTION:

The development of computer and communication technology, EHR has become an indispensable tool for medical services. The system utilizes some electronic devices such as the computer to deal with digital medical records, and it has the advantages of easy to use, stronger timeliness, and low cost. EHR not only provides the most useful data for diagnosis and scientific research but also it gives one kind of judgment basis for handling medical disputes. So, it has attracted a wide range of attention including the government, the medical community, cybersecurity department, and so on. Because the medical data is crucial for the diagnosis, and it is personal and sensitive for patients. Thus, data sharing and privacy preservation issues are critical in EHR. The medical data should be stored, managed, and accessed securely. Notably, the doctor usually needs to know the medical history of the patient when he/she makes the diagnosis or treatment. However, the patient can not professionally describe his/her medical history, which will affect the latest treatment. Thus, in EHR, historical medical data generated by different doctors in different hospitals should be capable of being securely and timely queried by a legitimate doctor with the patient's consent.

In recent years, the EHR system is markedly developed with the rise of cloud computing. Authors first expounded the security requirements of the EHR system based on cloud computing. Also, some suggestions are suggested to ensure the security of medical data in the cloud. The attribute-based encryption is utilized to protect the data in the cloud, and then the proposed EHR system is implemented in an android phone. Xhafa et al. proposed an attribute-based EHR with privacy awareness in cloud computing. However these cloud-based schemes have some flaws. For example, they have a dependency on the cloud provider. If some targeted attacks to cloud provider are carried out, then the information leakage is likely to occur. Additionally, the server may suddenly stop if the cloud providers would go bankrupt or be swallowed up by the larger companies. That is, the security of EHR will be threatened. In 2008, the blockchain structure was proposed. It can be viewed as a distributed database and satisfies the features of decentralization, tamper resistance, and asymmetric encryption. This technology can provide a reliable way to manage and store data. So it may be a promising solution for EHR. At present, the blockchain based researches for EHR have already started attracting attention from medicine.

CHAPTER 2

LITERATURE SURVEY

Paper 1: Robust watermarking method in DFT domain for effective management of medical imaging

Author: Manuel Cedillo-Hernandez · Francisco Garcia-Ugalde , Mariko Nakano-Miyatake · Hector Perez-Meana

Abstract:

In general, management of medical data is achieved by several issues of medical information such as authentication, security, integrity, privacy, among others. Because medical images and their related electronic patient record (EPR) data are stored separately; the probability of corruption of this information or their detachment from the corresponding EPR data could be very high. Losing data from the corresponding medical image may lead to a wrong diagnostic. Digital watermarking has recently emerged as a suitable solution to solve some of the problems associated with the management of medical images. This paper proposes a robust watermarking method for medical images to avoid their detachment from the corresponding EPR data in which the watermark is embedded using the digital imaging and communications in medicine standard metadata together with cryptographic techniques. In order to provide a high robustness of the watermark while preserving at the same time a high quality of the watermarked images, the generated watermark is embedded into the magnitude of the middle frequencies of the discrete Fourier transform of the original medical image. During the detection process, the watermark data bits are recovered and detected using the bit correct rate criterion. Extensive experiments were carried out, and the performance of the proposed method is evaluated in terms of imperceptibility, payload, robustness and detachment detection. Quantitative evaluation of the watermarked images is performed by using three of the more common metrics: the peak signal-to-noise ratio, structural similarity index and visual information fidelity. Experimental results show the watermark robustness against several of the more aggressive geometric and signal processing distortions. The receiver operating characteristics curves also show the desirable detachment detection performance of the proposed method. A comparison with the previously reported methods with similar purposes respect to the proposed method is also provided.

Paper 2: Effective Management of Medical Information Through A Novel Blind Water marking Technique

Authors : Sudeb Das · Malay Kumar Kundu

Abstract:

Medical Data Management (MDM) domain consists of various issues of medical information like authentication, security, privacy, retrieval and storage etc. Medical Image Watermarking (MIW) techniques have recently emerged as a leading technology to solve the problems associated with MDM. This paper proposes a blind, Contourlet Transform (CNT) based MIW scheme, robust to high JPEG and JPEG2000 compression and simultaneously capable of addressing a range of MDM issues like medical information security, content authentication, safe archiving and controlled access retrieval etc. It also provides a way for effective data communication along with automated medical personnel teaching. The original medical image is first decomposed by CNT. The Low pass subband is used to embed the watermark in such a way that enables the proposed method to extract the embedded watermark in a blind manner. Inverse CNT is then applied to get the watermarked image. Extensive experiments were carried out and the performance of the proposed scheme is evaluated through both subjective and quantitative measures. The experimental results and comparisons, confirm the effectiveness and efficiency of the proposed technique in the MDM paradigm.

Paper 3: Information Hiding—A Survey

Authors : Fabien a. P. Petitcolas, ross j. Anderson, and markus g. Kuhn

Abstract:

Information-hiding techniques have recently become important in a number of application areas. Digital audio, video, and pictures are increasingly furnished with distinguishing but imperceptible marks, which may contain a hidden copyright notice or serial number or even help to prevent unauthorized copying directly. Military communications systems make increasing use of traffic security techniques which, rather than merely concealing the content of a message using encryption, seek to conceal its sender, its receiver, or its very existence. Similar techniques are used in some mobile phone systems and schemes proposed for digital elections. Criminals try to use whatever traffic security properties are provided intentionally or otherwise in the available communications systems, and police forces try to restrict their use. However, many of the techniques proposed in this young and rapidly evolving field

can trace their history back to antiquity, and many of them are surprisingly easy to circumvent. In this article, we try to give an overview of the field, of what we know, what works, what does not, and what are the interesting topics for research.

Paper 4: Robust and distortion control dual watermarking in LWT domain using DCT and error correction code for color medical image

Author : A. K. Singh

Abstract:

This paper presents lifting wavelet transform (LWT) and discrete cosine transform (DCT) based robust watermarking approach for tele-health applications. For identity authentication, 'signature watermark' of size 64×64 and 'patient report' of size 80 characters are hiding into the host medical image. Further, the signature watermark is encrypted by message-digest (MD5) and 'patient report' is encoded by BCH error correcting code before embedding into the host image. Experimental demonstrations indicate that the method provides sufficient robustness and security against various attacks without significant distortion between cover and watermarked image. Further, our results proved that the method offer NC value more than 0.9214 for most of the considered attacks. Furthermore, it is evident from results our method shows the improvement in robustness to previously reported techniques under consideration while providing low computational complexity.

Paper 9: A Multimedia Healthcare Data Sharing Approach Through Cloud-based Body Area Network

Author: Mohammad MehediHassana, Kai Linb, XuejunYuec, Jiafu Wand

Abstract:

Wireless Body Area Network (WBAN), as a dramatic platform for pervasive computing and communication, has been widely applied in healthcare domains. Since the patient-related data in the form of text, image, voice, etc. is significant in the process of healthcare services, efficiently managing these media data from various WBAN is vital for various applications. Recently, Cloud-assisted WBAN has become popular that can supply massive computing, flexible storage and various software services to WBAN. Still, there are some challenging issues exist in this platform to deliver and share the huge media healthcare data to remote terminals timely with guaranteed QoS support. In the paper, we propose an efficient network model that combines WBAN and Cloud for valid data sharing. The proposed network architecture is designed as four layers: perception layer, network layer, cloud computing layer, and application layer. In the network, the integration of TCP/IP and Zigbee

in the coordinator devices is utilized. Consequently, WBAN coordinators can compatibility inter-operate with various local networks such as WiFi and LTE network to support high mobility of users. Besides, we integrate Content Centric Networking (CCN) with our proposed architecture to improve the ability of the WBAN coordinator. Thus, it can support uninterrupted media healthcare content delivery. In addition, adaptive streaming technique was also utilized to reduce packet loss. Various simulations were conducted using OPNET simulator to show the feasibility of the proposed architecture in terms of transmitting a huge amount of media healthcare data in real-time under traditional IP-based network.

CHAPTER 3

Objective

The main objective of the Project is to store the medical records safely in the Cloud environment. As for as Security is concerned, we deploy two main methodologies like Steganography and cryptography. We also deploy Block chain concept for the secured data storage and accessibility. For the hackers our system will identify the hackers and will provide dummy medical records called Decoy File.

Problem statement:

As for as problem statement, lots of security breach is happening in cloud environment. Data thread and hacking of any data is a common usability in cloud environment. That's why cloud was emerged 15 Years before till it is testing state. Security is the main issue in the Cloud environment.

Problem analysis:

After complete understanding of the security breaches happening in cloud environment, We deploy two main technological implementations like General Data Security (Steganography & Cryptography) and High level security like Block chain. Medical data is splitted into Images and Text. Images are converted into binary and made cryptography. Text content are hidden in an Image made Steganography is achieved. Our overall platform is built on Block chain Technology which ensures security features towards any data transfer.

3.1 EXISTING SYSTEM:

System, Electronic health record (EHR) has recorded the process of occurrence, development, and treatment of diseases. So it has high medical value. Owing to the private and sensitive nature of medical data for patients, the data sharing and privacy preservation are critical issues in EHR.

DISADVANTAGES:

- There is a chance to miss the record
- Hardcopy of record is not safe
- Record may be hacked by someone.

3.2 PROPOSE SYSTEM:

We propose a efficient storage system for patient information. Through this system we providing a security system for patient and hospital details. To secure the record we are implementing an efficient system by crypto stegno/ Watermarking system. Patient details are stored in cloud for memory management and it will easy way access the data. We are providing permission key to access the patient details. No one can access the information without the knowledge of doctor. So through this a security level is increased by storing data uploading crypto stegnography / Watermarking.

ADVANTAGES:

- More security on storage
- Long time storage on cloud
- User can view at any time.

CHAPTER 4

PROJECT DESCRIPTION

Aim of the project :

The aim of the project is to secure the data using steganography and cryptography.

4.1 ARCHITECTURE DIAGRAM:

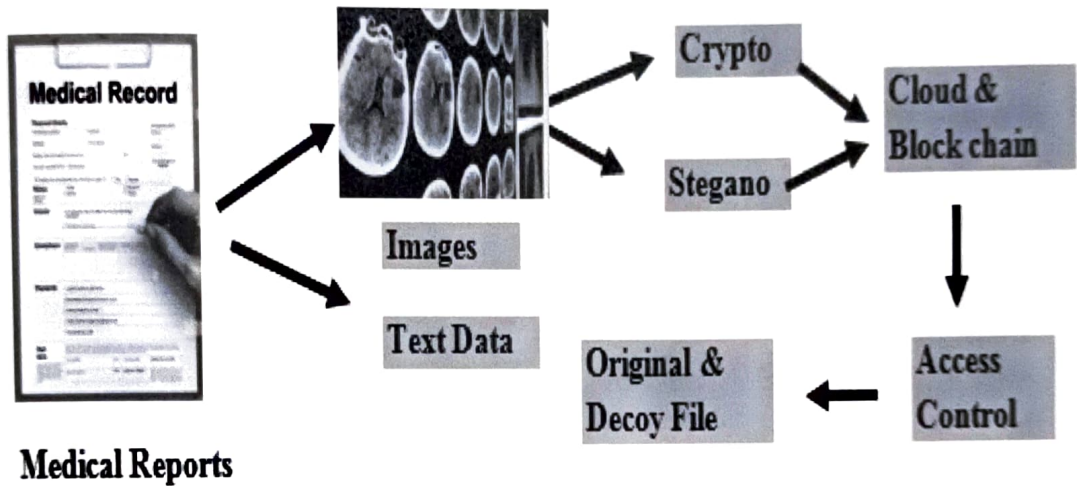


Fig:4.1

4.2 USE CASE DIAGRAMS:

A use case is a list of steps that define interaction between an actor (a human who interacts with the system or an external system) and the system itself. Use case diagrams depict the specifications of a use case and model the functional units of a system. These diagrams help development teams understand the requirements of their system, including the role of human interaction therein and the differences between various use cases. A use case diagram might display all use cases of the system, or just one group of use cases with similar functionality.

1. To begin a use case diagram, add an oval shape to the center of the drawing.
2. Type the name of the use case inside the oval.
3. Represent actors with a stick figure near the diagram, then use lines to model relationships between actors and use cases.

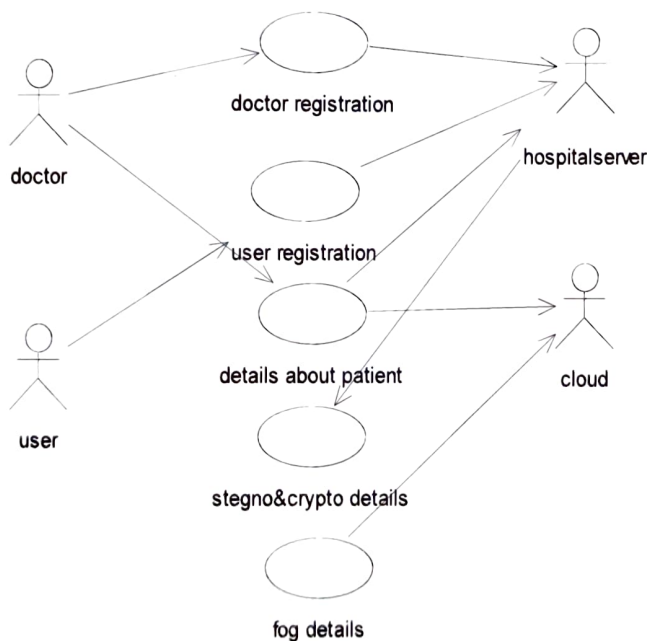


Fig:4.2

4.3 SEQUENCE DIAGRAM :

Sequence diagrams, also known as event diagrams or event scenarios, illustrate how processes interact with each other by showing calls between different objects in a sequence. These diagrams have two dimensions: vertical and horizontal. The vertical lines show the sequence of messages and calls in chronological order, and the horizontal elements show object instances where the messages are relayed.

1. To create a sequence diagram, write the class instance name and class name in a rectangular box.
2. Draw lines between class instances to represent the sender and receiver of messages.
3. Use solid arrowheads to symbolize synchronous messages, open arrowheads for asynchronous messages, and dashed lines for reply messages.

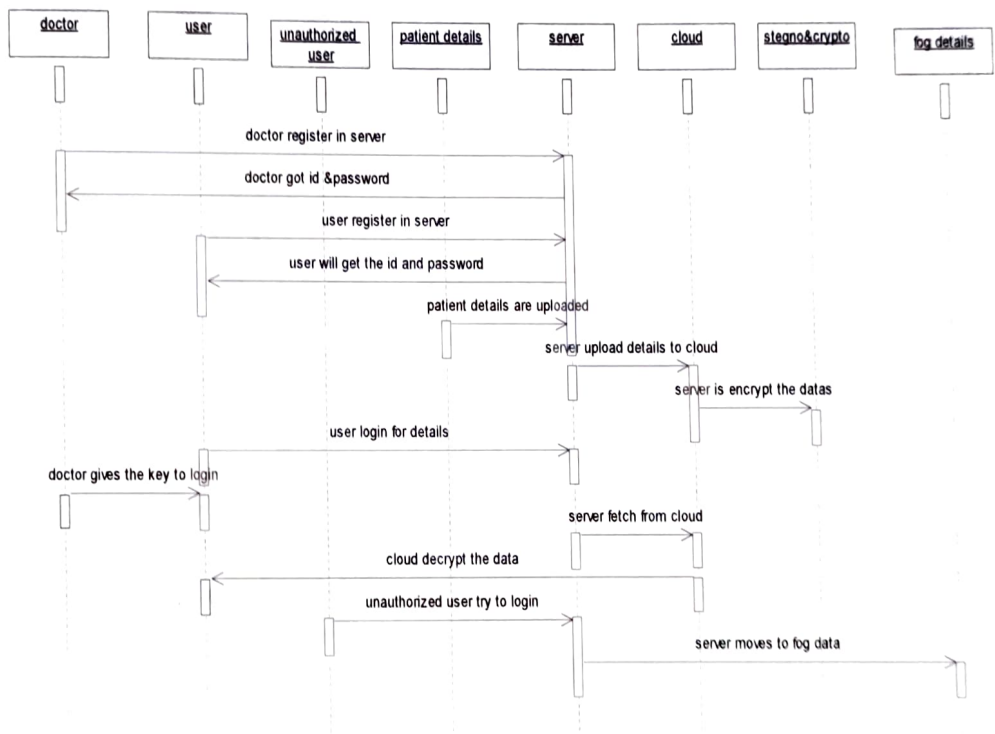


Fig:4.3

CHAPTER 5

SYSTEM SPECIFICATION

Requirement analysis

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

5.1 HARDWARE ENVIRONMENT

- Hard disk : 500GB
- Ram : 2 - 4 GB
- Processor : Core i5

5.2 SOFTWARE ENVIRONMENT

- Operating system : Windows 7/ 8
- Languages : Java 8
- Data Base : Mysql
- IDE : Net Beans 8.2

Overview of the mysql database management system

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded in 1995 by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

- The MySQL software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL is a registered trademark of MySQL AB.

- The MySQL software is Dual Licensed. Users can choose to use the MySQL software as an Open Source product under the terms of the GNU General Public License or can purchase a standard commercial license from MySQL AB.

MYSQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MYSQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard.

MYSQL overview

MySQL is a database management system.

A database is a structured collection of data. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.

MYSQL is a relational database management system.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.”

MYSQL software is open source.

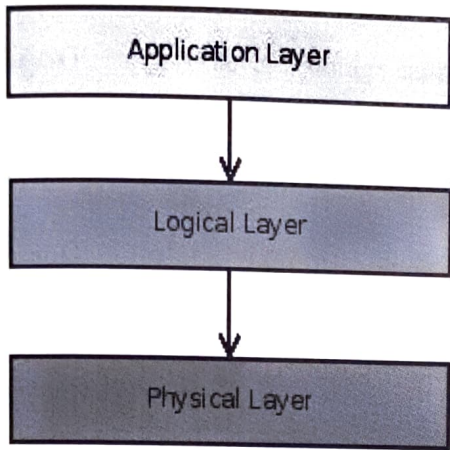
Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. If you want to redistribute altered versions MySQL, other licences are available

MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

MySQL Architecture.

Three layer model:



- The Application layer contains common network services for connection handling, authentication and security. This layer is where different clients interact with MySQL these clients can be written in different API's: .NET, Java, C, C++, PHP, Python, etc...
- The Logical Layer is where the MySQL intelligence resides, it includes functionality for query parsing, analysis, caching and all built-in functions (math, date...). This layer also provides functionality common across the storage engines.
- The Physical Layer is responsible for storing and retrieving all data stored in "MySQL". Associated with this layer are the storage engines, which MySQL interacts with very basic standard API's. Each storage engine has its strengths and weaknesses, some of these engines are MyISAM, InnoDB, CSV, NDB Cluster, Falcon, etc...

Some internal components:

Each client connection gets its own thread within the server process.

When clients (applications) connect to the MySQL server, the server needs to authenticate them.

Before even parsing the query, though, the server consults the query cache, which only stores SELECT statements, along with their result sets.

The storage engine does affect how the server optimizes query.

Storage engines:

MySQL supports several storage engines that act as handlers for different table types. MySQL storage engines include both those that handle transaction-safe tables and those that handle non-transaction-safe tables.

MySQL Cluster Overview:

MySQL Cluster is a technology that enables clustering of in-memory databases in a shared-nothing system. The shared-nothing architecture enables the system to work with very inexpensive hardware, and with a minimum of specific requirements for hardware or software.

MySQL Cluster is designed not to have any single point of failure. In a shared-nothing system, each component is expected to have its own memory and disk, and the use of shared storage mechanisms such as network shares, network file systems, and SANs is not recommended or supported.

MySQL Cluster integrates the standard MySQL server with an in-memory clustered storage engine called NDB. In our documentation, the term NDB refers to the part of the setup that is specific to the storage engine, whereas “MySQL Cluster” refers to the combination of one or more MySQL servers with the NDB storage engine. A MySQL Cluster consists of a set of computers, known as hosts, each running one or more processes. These processes, known as nodes, may include MySQL servers, data nodes (for storage of the data), one or more management servers, and possibly other specialized data access programs.

5.3 N-TIER ARCHITECTURE:

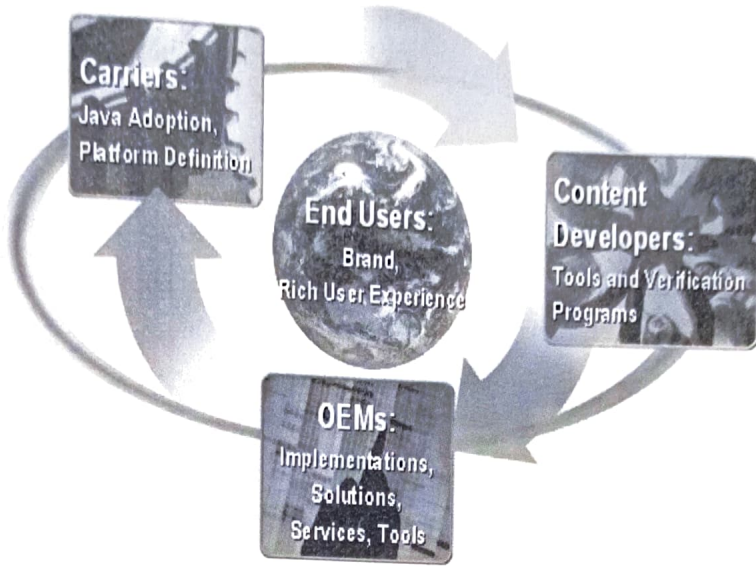


Fig:5.1 N-Tier Architecture

5.3.1 Java:

The Java platform is the ideal platform for network computing. Running across all platforms -- from servers to cell phones to smart cards -- Java technology unifies business infrastructure to create a seamless, secure, networked platform for your business. The Java platform benefits from a massive community of developers and supporters that actively work on delivering Java technology-based products and services as well as evolving the platform through an open, community-based, standards organization known as the Java Community Process program. You can find Java technology in cell phones, on laptop computers, on the Web, and even trackside at Formula One Grand Prix races. The fact is today, you can find Java technology just about everywhere!

5.3.2 Features of java:

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code(class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and

object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is considered by many as one of the most influential programming languages of the 20th century, and widely used from application software to web application.

5.3.3 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
- Portal
- Distributed
- High performance

Each of the preceding buzzwords is explained in The Java Language Environment, a white paper written by James Gosling and Henry Chilton. In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains byte codes the machine language of the Java Virtual Machine. The java launcher tool then runs your application with an instance of the Java Virtual Machine.

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java Hotspot virtual machine, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code.

5.3.4 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 Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying 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
- The Java Application Programming Interface (API)

5.3.5 Java virtual machine:

A Java virtual machine (JVM) is a virtual machine that can execute Java byte code. It is the code execution component of the Java software platform. A Java virtual machine is a program which executes certain other programs, namely those containing Java byte code instructions. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware. A JVM provides an environment in which Java byte code can be executed, enabling such features as automated exception handling, which provides root-cause debugging information for every software error (exception), independent of the source code.

A JVM is distributed along with a set of standard class libraries that implement the Java application programming interface (API). These libraries, bundled together with the JVM, form the Java Runtime Environment (JRE). JVMs are available for many hardware and software platforms.

5.3.6 Java api:

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as packages.

CHAPTER 6

MODULES

MODULES:

- A modular design reduces complexity, facilitates change, and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified.
- Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity, Modular Protection.
- The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

1. User registration
2. Hospital server
3. Stegno analysis
4. Image cryptography
5. Fog computing
6. Data access
7. Block chain implementation

USER REGISTRATION:

In this module user/ patient have to register their personal information like name, address, mail ID, mobile number, address. And those details will be stored on database. After registration user will get user ID and password to access the application. This is an application to view their hospital report from cloud. to access the hospital records, we are creating user Id and password for authentication.

HOSPITAL SERVER:

Server is the main process for evry application because it is the only way for communication it will establish the communication between client and corresponding website. In this module we are implementing hospital server to maintain both patient information , doctor information and other hospital details. All doctors have to register their designation and other details same like that other hospital have to register their details on this server. Because patient may change their treatment from one hospital to another that is why hospital will also register their

information. server will maintain all the details and provide details whenever user request for the query.

STEGNO ANALYSIS:

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. We are hiding the patient's personal information and their report about disease. Every data will be stored as a stegno format. when we stored the record as a stegno it will not be hacked or theft by any one.

IMAGE CRYPTOGRAPHY:

In our system we are storing the patient scanning report like their x-ray, ECG and other images in encrypted form using ECC algorithm. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. Using ECC algorithm we are encrypting the image file and store it in cloud server.

FOG COMPUTING:

Fog computing is a term created by Cisco that refers to extending cloud computing to the edge of an enterprise's network. Fog computing facilitates the operation of compute, storage and networking services between end devices and cloud computing data centers. It is an separate to maintain the decoy information. Here we are decoy the patient information in secure way. Decoy is the process of creating a fake report of patient.

DATA ACCESS:

In this module we are implementing accessing process for user as well as hospital. In this initially a patient detail will be maintained by a doctor who is the person patient met at first time. He had only the permission to access the patient information. If it any case patient change the hospital or he/she want to view his/her details they need some key to access the data which is stored in cloud. if the patient their hospital a new doctor have to give key to access the patient details. For both people they need one key to access the file. That key will be provided by a doctor from hospital 1. Bothe patient and doctor have user id and password to access the file. System will check their ID and password if it match with previous data base they will get key from doctor and view the details. if it will not match with previous database server will provide decoy file and alert will be send to the patient.

BLOCK CHAIN IMPLEMENTATION:

A block is a container data structure. The average size of a block seems to be 1MB. Here every certificates number will be created as a block. For every block an hash code will generate for security. Patient records are created as a block chain.

CHAPTER 7

SYSTEM TESTING

Generally testing process is carried out to ensure that the system has been developed according to the required specifications and the expected output is properly obtained. There are two main categories of testing namely, the White Box Testing and the Black Box Testing. Each of this testing in turn consists of many types of testing.

7.1 TAXONOMY OF TESTING:

7.1.1 White box testing:

White Box testing is also known as glass box testing. This type of testing, tests the internal structure of the program. This can be applied at the unit, integration and system levels of testing. Mostly, it is used in the unit level of the software testing process. Sometimes it may not reveal defects in areas which have not been implemented. It has its own advantages and its own disadvantages. The advantage is that knowing the programming language code and familiarizing with them may prove vital and help in identifying the errors quickly and at times may help in avoiding them at the earliest.

7.1.2 Black box testing:

It is amongst the two methods of mostly used testing methods. This tests the main functionality of the program. It can be applied to every level of testing such as Unit, Integration, System and Acceptance levels of testing. Exhaustive input testing is required to find all errors. For doing this type of testing knowing the internal code and how it works is not needed but what it is supposed to do is known by the person who is performing the test. The test cases are developed based on the specific requirements according to the goals. There are Boundary Value Analysis, Class Partitioning, Cause Effect Graph etc.,

7.1.3 Unit testing:

Unit testing is also known as Module Testing which focuses on verification efforts on the module. The module is tested separately and this is carried out at the programming stage itself. Unit test comprises of the set of tests performed by an individual programmer before integration of the unit into the system. This will help to test each and every single part or we can say as each and every module completely.

7.1.4 Functional testing:

Functional testing is mainly used as a Quality Assurance process. This is a very simple process where each function is provided with an appropriate input and is verified against an expected output and with boundary values. This would help in ensuring that the output is as acquired according to the expectations and would help assuring the quality. The various functions developed using Java are separately tested for their proper working by executing them as separate files.

Input Data: Text File , Image

Output Data: Cryptography, Steganography

7.2 TESTING IN PARTICULAR

7.2.1 Unit testing:

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this project, all statements are executed properly. All units of program programs are tested in different computer. And the result of the project is same in all system.

7.2.2 Integration testing:

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

CHAPTER 8

APPENDIX

8.1 CODING:

Registration:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButton3ActionPerformed

    // TODO add your handling code here:

    String patid=jTextField4.getText().toString().trim();
    String Name=jTextField5.getText().toString().trim();
    String Email=jTextField6.getText().toString().trim();
    String Password=jTextField7.getText().toString().trim();
    String ContactNo=jTextField8.getText().toString().trim();
    String Location=jTextField9.getText().toString().trim();
    String Hospital=jComboBox1.getSelectedItem().toString();
    com.databaseoperation.AddDataoperation ad=new AddDataoperation();
    int i=ad.InsertPat(patid,Name,Email, Password, ContactNo, Location, Hospital);

    if(i>0){

        jTextField4.setText("");
        jTextField5.setText("");
        jTextField6.setText("");
        jTextField7.setText("");
        jTextField8.setText("");
        jTextField9.setText("");

        JOptionPane.showMessageDialog(rootPane, "Details saved Successfully");
    }

    else {
```

```

JOptionPane.showMessageDialog(rootPane, "Connection Error");

}

} //GEN-LAST:event_jButton3ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButton1ActionPerformed

    // TODO add your handling code here:

    this.dispose();

    new AdminMenu().setVisible(true);

} //GEN-LAST:event_jButton1ActionPerformed

try {

    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {

        if ("Nimbus".equals(info.getName())) {

            javax.swing.UIManager.setLookAndFeel(info.getClassName());

            break;

        }

    }

} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(Userregister.class.getName()).log(java.util.logging.Level.
        SEVERE, null, ex);

} catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(Userregister.class.getName()).log(java.util.logging.Level.
        SEVERE, null, ex);

} catch (IllegalAccessException ex) {

```

```
java.util.logging.Logger.getLogger(Userregister.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(Userregister.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
//</editor-fold>
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
    public void run() {
```

```
        new Userregister().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

Doctor Registration:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jButton1ActionPerformed
```

```
    // TODO add your handling code here:
```

```
String name= jTextField1.getText().toString().trim();
```

```
String unname=jTextField2.getText().toString().trim();
```

```
String pass=jTextField3.getText().toString().trim();
```

```
String email=jTextField4.getText().toString().trim();
```

```
String cno=jTextField5.getText().toString().trim();
```

```
String hospital=jComboBox1.getSelectedItem().toString();
```

```
AddDataoperation _uSrdetails=new AddDataoperation();
```

```
int i=_uSrdetails.InsertDoc(name, unname, pass, email, cno, hospital);
```

```

if(i>0){
    JOptionPane.showMessageDialog(rootPane, "Details Saved");
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
}

```

```

} //GEN-LAST:event_jButton1ActionPerformed

```

```

try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(DoctorRegister.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(DoctorRegister.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(DoctorRegister.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(DoctorRegister.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

```

/ Create and display the form */*

```

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new DoctorRegister().setVisible(true);

    }

});

}

```

Upload Page:

```

private void uploadActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_uploadActionPerformed

```

```

    // TODO add your handling code here:

```

```

    log.addLog("Upload button is pressed");

```

```

    exit:

```

```

    if (true) {

```

```

        log.addLog("Validating whether index is added or not");

```

```

        FileInputStream fis = null;

```

```

        byte[] txtByte = null;

```

```

while(st.hasMoreTokens()){
    StringBuilder sb=new StringBuilder();
    String data = (String)st.nextToken();
    sb.append(data);
    System.out.println("Method"+data);
    modell.addElement(data);
}

```

Encrypt:

```

public class Encrypt {

```

```

    public static byte[] getFile(String path) {

```

```

        File f = new File("/home/bridgeit/Desktop/Olympics.jpg");

```

```

        InputStream is = null;

```

```

        try {

```

```

            is = new FileInputStream(f);

```

```

        } catch (FileNotFoundException e2) {

```

```

            // TODO Auto-generated catch block

```

```

            e2.printStackTrace();

```

```

        }

```

```

        byte[] content = null;

```

```

        try {

```

```

            content = new byte[is.available()];

```

```

        } catch (IOException e1) {

```

```

        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    try {
        is.read(content);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return content;
}

public static byte[] encryptPdfFile(SecretKey secretKey, byte[] content) {
    Cipher cipher;
    byte[] encrypted = null;
    try {
        cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");

        cipher.init(Cipher.ENCRYPT_MODE, secretKey);

        encrypted = Base64.encodeBase64(cipher.doFinal(content));

    } catch (Exception e) {

        System.out.println("Error while encrypting: " + e.toString());
    }
}

```

```
}  
  
public static String main(String path)  
    throws NoSuchAlgorithmException, InstantiationException, IllegalAccessException,  
    IOException {
```

```
    SecretKeySpec secretKey;
```

```
    byte[] key;
```

```
    String dummy="dd";
```

```
    String myKey = "ThisIsAStrongPasswordForEncryptionAndDecryption";
```

```
    MessageDigest sha = null;
```

```
    key = myKey.getBytes("UTF-8");
```

```
    System.out.println(key.length);
```

```
    sha = MessageDigest.getInstance("SHA-1");
```

```
    key = sha.digest(key);
```

```
    key = Arrays.copyOf(key, 16); // use only first 128 bit
```

```
    System.out.println(key.length);
```

```
    System.out.println(new String(key, "UTF-8"));
```

```
    secretKey = new SecretKeySpec(key, "AES");
```

```
    byte[] content = getFile(path);
```

```
    System.out.println(content);
```

```
    byte[] encrypted = encryptPdfFile(secretKey, content);
```

```
    System.out.println(encrypted);
```

```

// byte[] decrypted = decryptPdfFile(secretKey, encrypted);
// System.out.println(decrypted);

//saveFile(decrypted);
System.out.println("Done");

return dummy;

}

}

```

Decrypt:

```

public class DecryptImage {

    public static byte[] getFile(String path) {

        File f = new File(path);

        InputStream is = null;

        try {

            is = new FileInputStream(f);

        } catch (FileNotFoundException e2) {

            // TODO Auto-generated catch block

            e2.printStackTrace();

        }

        byte[] content = null;

        try {

            content = new byte[is.available()];

        } catch (IOException e1) {

```

```

// TODO Auto-generated catch block
e1.printStackTrace();

}

try {
    is.read(content);
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

return content;
}

public static byte[] encryptPdfFile(SecretKey secretKey, byte[] content) {
    Cipher cipher;
    byte[] encrypted = null;
    try {
        cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");

        cipher.init(Cipher.ENCRYPT_MODE, secretKey);

        encrypted = Base64.encodeBase64(cipher.doFinal(content));

    } catch (Exception e) {

        System.out.println("Error while encrypting: " + e.toString());
    }
}

```

```

String dropfilename=RandomInteger.main(4);
byte[] encrypted = encryptPdfFile(secretKey, content);
System.out.println(encrypted);
System.out.println("KEY IS"+secretKey);
FileOutputStream fout=new
FileOutputStream(constants.Constants.IMAGEPATHOUT+dropfilename+filename+".txt");
fout.write(encrypted);
fout.close();

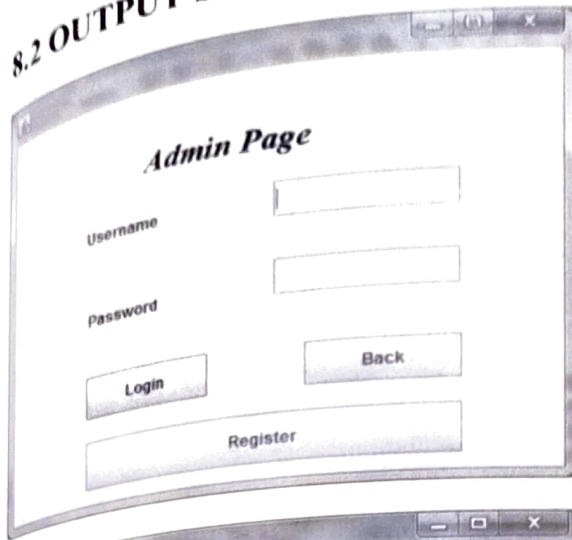
Filepath=constants.Constants.IMAGEPATHOUT+"/"+dropfilename+filename+".txt";
com.entity.BankEntity.setFilename(dropfilename+filename+".txt");
//byte[] decrypted = decryptPdfFile(secretKey, encrypted);
//System.out.println(decrypted);

//saveFile(decrypted);
// System.out.println("Done");

return Filepath;
}
}

```

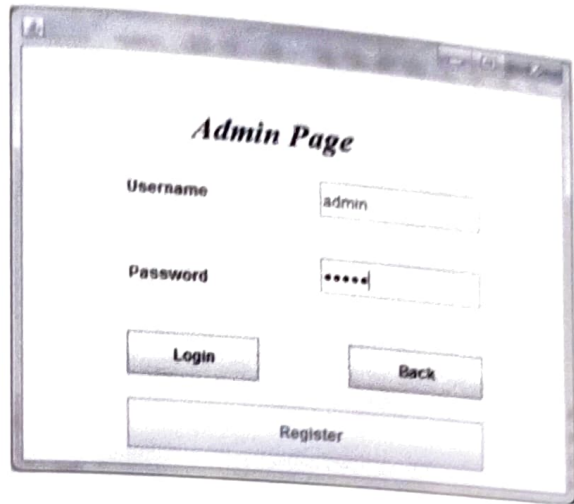
8.2 OUTPUT SCREENSHOT:



Admin Page

Username

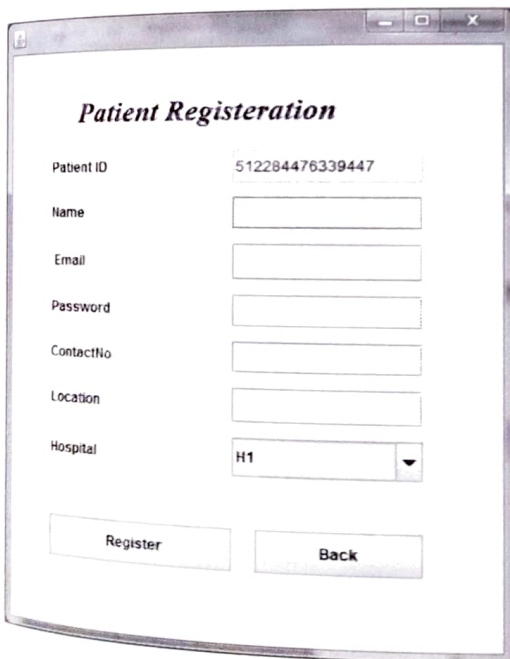
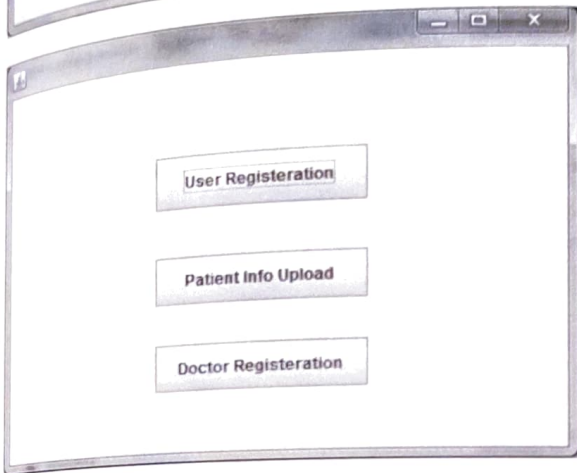
Password



Admin Page

Username

Password



Patient Registration

Patient ID

Name

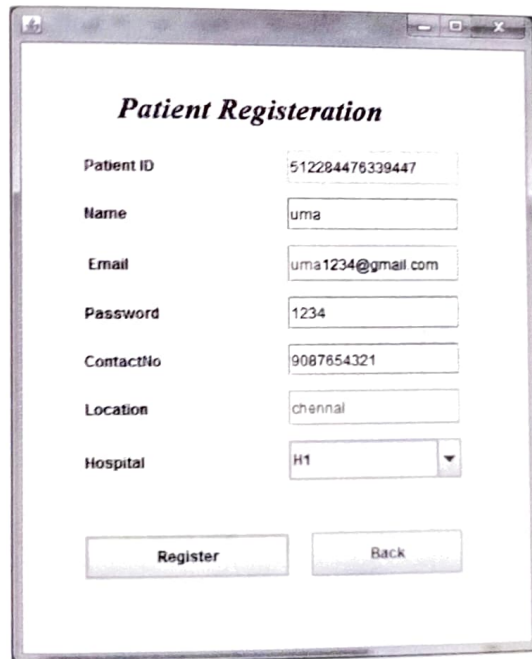
Email

Password

ContactNo

Location

Hospital



Patient Registration

Patient ID

Name

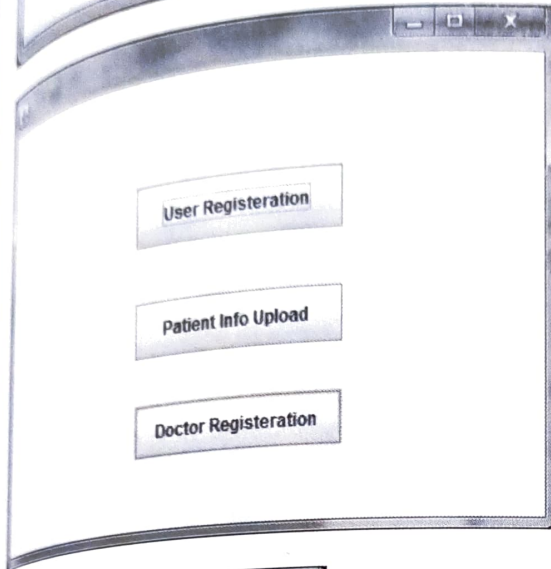
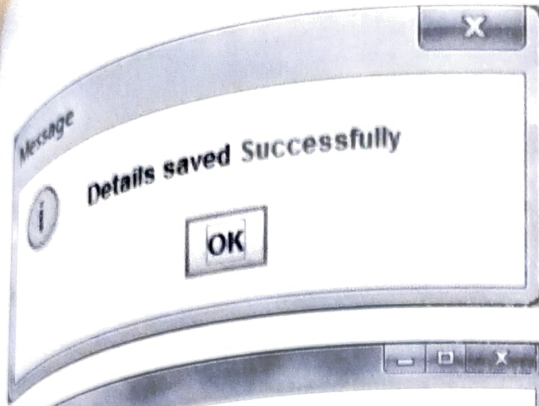
Email

Password

ContactNo

Location

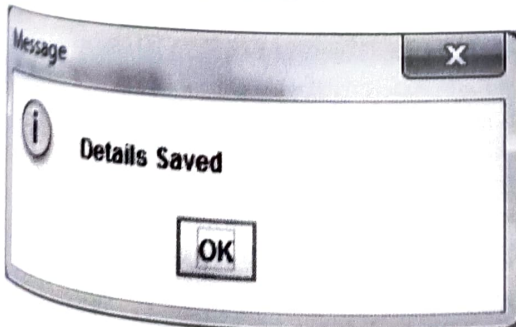
Hospital



A screenshot of the "Doctor Registration" form. The title bar says "Doctor Registration". The form contains the following fields and labels:

- NAME:
- UNAME:
- PASSWORD:
- EMAIL:
- CONTACT NO:
- HOSPITAL:

At the bottom of the form are two buttons: "Register" and "Back".



Doctor Registration

NAME

U NAME

PASSWORD

EMAIL

CONTACT NO

HOSPITAL

Message

 **Details Saved**

Patient Info

Patient ID

Disease

Dosage

Doctor Name

Hospital

Report

Image File:

Content File:

Patient Info

Patient ID: 512284476338447

Disease: cancer

Dosage: 500ml

Doctor Name: doc1

Hospital: H1

Report:

Image File:

Content File:

Patient Info

Patient ID: 512284476338447

Disease: cancer

Dosage: 500ml

Doctor Name: doc1

Hospital: H1

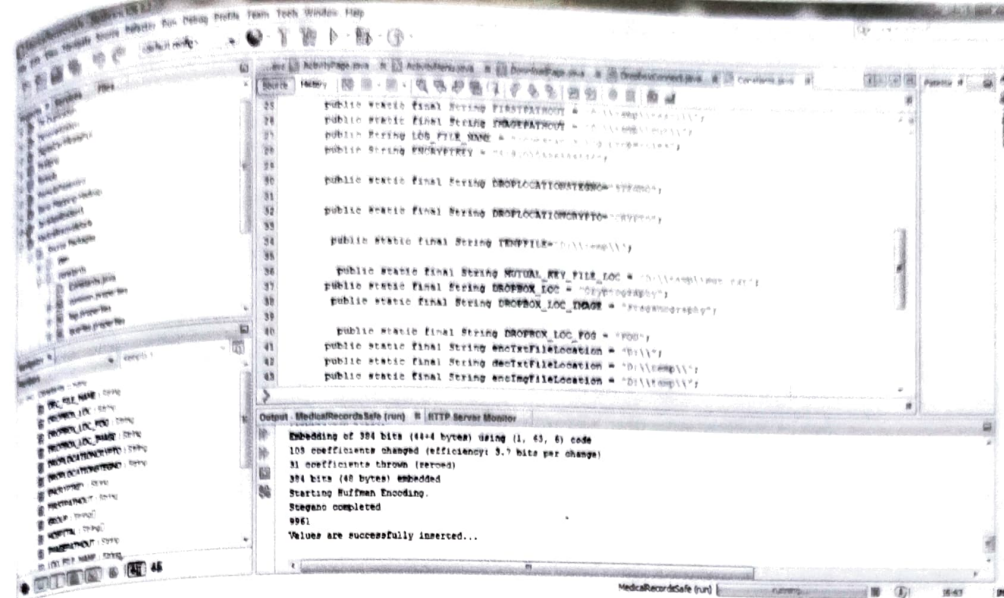
Report: ReC:\Users\Satish\digital\Pictures\download.jpg

Image File: download.jpg

Content File: cert.txt

Message

 File Uploaded



Patient Info

Patient ID:

Disease:

Dosage:

Doctor Name:

Hospital:

Report:

Image File:

Content File:

Patient Info

Patient ID: 512284478339447

Disease:

Dosage:

Doctor Name: pradhap

Hospital: H1

Report: **Browse**

Image File: **Browse**

Content File: **Browse**

Upload **Close**

Stegno **Download**

View Request

Message

i Click to view the content

OK



Admin Page

Username

Password

Patient Info

Patient ID

Disease

Dosage

Doctor Name

Hospital

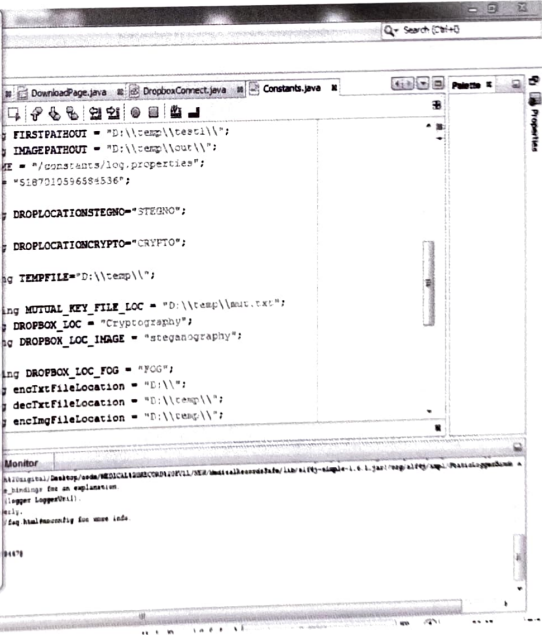
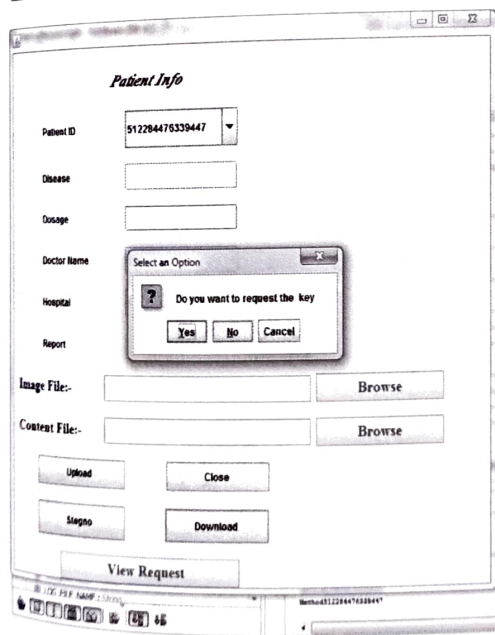
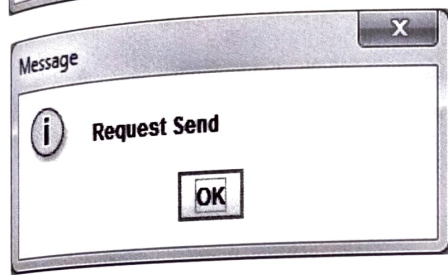
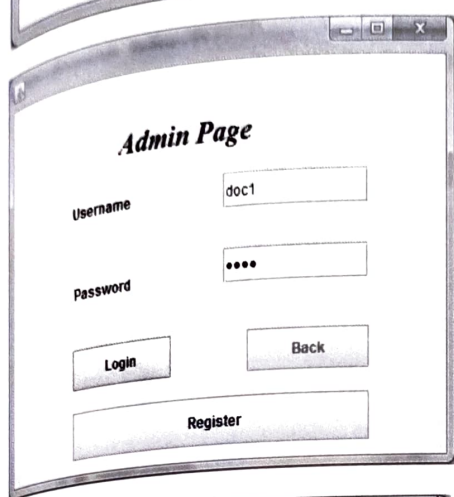
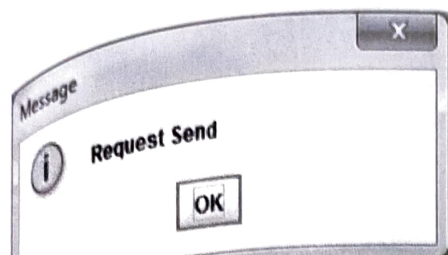
Report

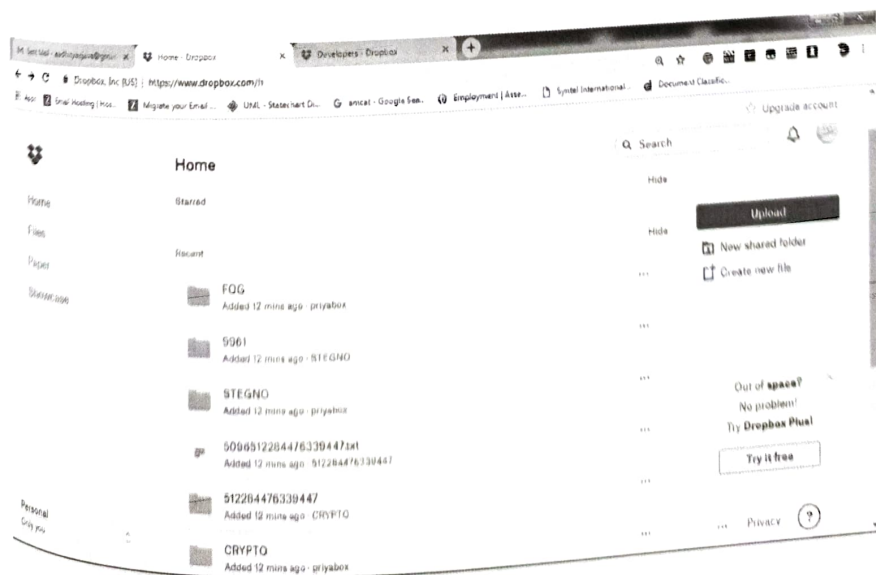
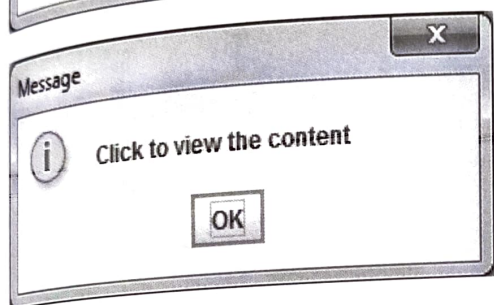
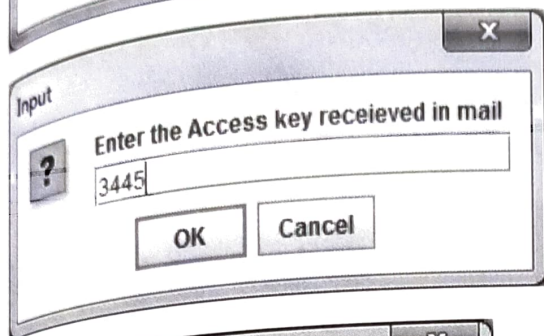
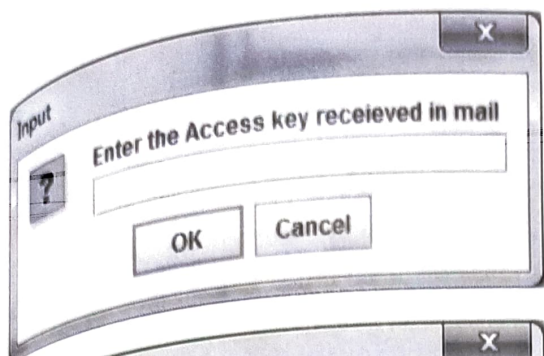
Image File:-

Content File:-

Select an Option

Do you want to request the key





CHAPTER 9

CONCLUSION:

In this paper we infer that crypto stegno technique will be more secure for Patient information .for that ECC algorithm is used. They can view their information anywhere but by providing to access the data. So security level increased by comparing with existing work.

FUTURE ENHANCEMENT:

The future work of this paper is we can secure the content in two different cloud server and also we can divert the cloud server while third party try to accessing our patients information. Try to implement this system in mobile application for user feasible.