



Blockchain-Based Public Uprightness Check For Distributed Storage Against Hesitating Inspectors

A PROJECT REPORT

Submitted by
Anbarazsi.T
Harithaa.S
Lavanya.T

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

PANIMALAR ENGINEERING COLLEGE

ANNA UNIVERSITY : CHENNAI 600 025

March 2021



Blockchain-Based Public Uprightness Check for Distributed Storage Against Hesitating Inspectors

A PROJECT REPORT

Submitted by

ANBARAZSI.T [REGISTER NO:211417104015]

HARITHAA.S [REGISTER NO:211417104081]

LAVANYA.T [REGISTER NO:21147104131]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2020

BONAFIDE CERTIFICATE

Certified that this project report **“Blockchain-based public uprightness check for distributed storage against hesitating inspectors”** is the bonafide work of **“Anbarazsi.T (211417104015), Harithaa.S (211417104081), Lavanya.T (211417104131)”** who carried out the project work under my supervision
Mrs.Jackulin.

SIGNATURE

**Dr.S.MURUGAVALLI,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT**

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

**Mrs.C.Jackulin
SUPERVISOR
Assistant Professor**

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123

Certified that the above candidate(s) was/ were examined in the Anna University
Project Viva-Voce Examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI., Dr.C.SAKTHIKUMAR M.E.,Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Mrs.C.Jackulin** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NAME OF THE STUDENTS

Harithaa.S
Anbarazsi.T
Lavanya.T

ABSTRACT

Since network storage services achieve widespread adoption, security and performance issues are becoming primary concerns, affecting the scalability of storage systems. Countermeasures like data auditing mechanisms and deduplication techniques are widely studied. However, the existing data auditing mechanism with deduplication cannot solve the problems such as high cost and reliance on trusted third parties in traditional approaches, and it also faces the problem of repeated auditing of data shared by multiple-tenant. This paper proposes a blockchain-based deduplicatable data auditing mechanism. We first design a client-side data deduplication scheme based on bilinear-pair techniques to reduce the burden on users and service providers. On this basis, we achieve a trustworthy and efficient data auditing mechanism that helps to check data integrity by using both the blockchain technique and bilinear pairing cryptosystem. The blockchain system is used to record the behaviors of entities in both data outsourcing and auditing processes so that the corresponding immutable records can be used to not only ensure the credibility of audit results but also help to monitor unreliable third-party auditors. Finally, theoretical analysis and experiments reveal the effectiveness and performance of our scheme.

TABLE OF CONTENTS

CHAPTERNO.	TITLE	PAGE NO.
	ABSTRACT	5
	LIST OF TABLES	8
	LIST OF FIGURES	9
	LIST OF SYMBOLS, ABBREVIATIONS	12
1.	INTRODUCTION	13
	1.1 Overview	13
	1.2 Problem Definition	14
2.	LITERATURE SURVEY	15
3.	SYSTEM ANALYSIS	18
	3.1 Existing System	18
	3.2 Proposed system	18
	3.3 Requirement Analysis and Specification	19
	3.3.1 Hardware requirement	
	3.3.1.1 input requirements	19
	3.3.1.2 output requirements	
	3.3.2 Software specification	

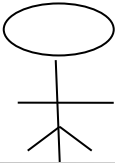
CHAPTERNO.	TITLE	PAGE NO.
4.	SYSTEM DESIGN	
	4.1. ER diagram	24
	4.2 Uml diagrams	25
	4.2.1 use case diagram	26
	4.2.2 class Diagram	27
	4.2.3 object diagram	27
	4.2.4 sequence diagram	28
5.	SYSTEM ARCHITECTURE	29
	5.1 Architecture Overview	30
	5.1 Module Design Specification	31
	5.2 Algorithm	31
6.	SYSTEM IMPLEMENTATION	33
	6.1 Client-side coding	33
	6.2 Server-side coding	42
7.	SYSTEM TESTING	52
	7.1 Unit Testing	53
	7.2 Integration Testing	53

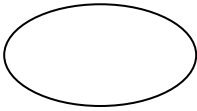


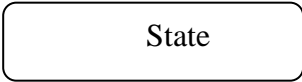
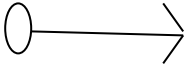
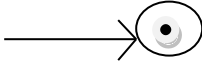
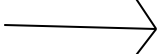
CHAPTERNO.	TITLE	PAGE NO.
	7.3 Test Cases & Reports / Performance Analysis	59
8.	CONCLUSION	60
	8.1 Conclusion and Future Enhancements	60
	APPENDICES	61
	A.1 Sample Screens	61
	A.2 Publications	65
	REFERENCES	67

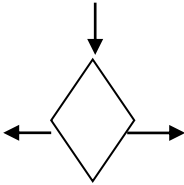
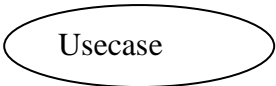
List Of Figures

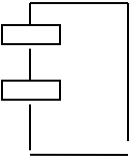
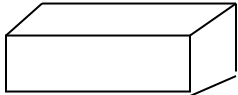
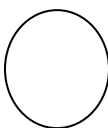
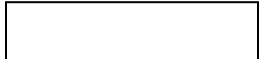
FIGURE NO	NAME OF THE FIGURE	PAGE NO.
1	E-R Diagram	24
2	Use case Diagram	25
3	Class diagram	26
4	Object diagram	27
5	Sequence diagram	28
6	Architecture Diagram	29
7	Homo morphic Algorithm	31



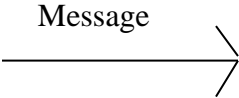
LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <i>+ public</i> <i>- private</i> <i># protected</i> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <i>Class Name</i> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <i>-attribute</i> <i>-attribute</i> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <i>+operation</i> <i>+operation</i> </div> </div>	Represents a collection of similar entities grouped together.
2.	Association	<div style="display: flex; justify-content: center; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin: 0 10px;">Class A</div> <div style="border-bottom: 1px solid black; width: 50px; margin: 0 5px;"></div> <div style="border: 1px solid black; padding: 5px; margin: 0 10px;">Class B</div> </div> <div style="display: flex; justify-content: center; align-items: center;"> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 10px;"></div> <div style="border-bottom: 1px solid black; width: 50px; margin: 0 5px;"></div> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 10px;"></div> </div>	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Class A</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Class B</div> <div style="font-size: 20px;">↑</div> </div> <div style="text-align: center;"> <div style="border: 1px solid black; width: 100px; height: 20px; margin-bottom: 10px;"></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Class B</div> <div style="font-size: 20px;">↑</div> </div> </div>	Interaction between the system and external environment

5.	Relation (uses)	<p>Uses</p> 	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processs.
9.	Initial State		Initial state of the object
10.	Final state		F inal state of the object
11.	Control flow		Represents various control flow between the states.

12.	Decision box		Represents decision making process from a constraint
13.	Usecase		Interact ion between the system and external environment.

14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.

18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	DB	Database
2.	JVM	Java Virtual Machine
3.	JSP	Java Server Page
4.	CB	Collective Behavior
5.	SD	Social Dimension
6.	JRE	Java Runtime Environment
7.	SSD	Sparse Social Dimension
8.	LGP	Line Graph Partition

CHAPTER 1

INTRODUCTION

In the era of network computing , network storage services achieve widespread adoption and benefit countless users worldwide due to the superior capability of providing low cost and highly scalable storage anytime, anywhere. According to a recent report, by 2025, approximately 50% of data will be stored in network storage devices with an incredible market value of more than \$100 billion globally . With the great success of network storage technology, its security and performance issues are increasingly becoming significant concerns. This is due to the fact that network storage service providers may maliciously or accidentally corrupt the user's data. Remote data auditing mechanism is one of the simplest but most effective security solutions that can help to check the integrity of outsourced data has attracted much attention. Meanwhile, because almost 75% of outsourced data are duplicate copies, the deduplication technique that eliminates duplicate data is widely adopted in the commercial settings for reducing storage costs and for improving the scalability of the system

1.1 OVERVIEW

In order to solve this problem, we propose a double encryption concept. The proposed framework can both take full advantage of cloud storage and protect the privacy of data. Besides, Hash-Solomon code algorithm is designed to divide data into different parts. Then, we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.

1.2 PROBLEM DEFINITION

Since network stockpiling administrations accomplish boundless appropriation, security and execution issues are getting essential concerns, influencing the versatility of capacity frameworks. Countermeasures like information evaluating instruments and deduplication procedures are generally contemplated. In any case, the current information examining instrument with deduplication can't take care of the issues, for example, significant expense and dependence on confided in outsiders in customary methodologies, and it likewise faces the issue of continued examining of information shared by different occupant

CHAPTER -2

LITERATURE SURVEY

Security and privacy of big data in a cloud environment.

In smart meters, data of the consumers must be protected else private information can be leaked. Similarly, due to the cost-efficiency, reduced overhead management and dynamic resource needs, content owners are outsourcing their data to the cloud who can act as a service provider on their behalf. However, by outsourcing their data to the cloud, the owners may lose access control and privacy of data as cloud becomes a third party. By using these data storage services, the data owners can relieve the burden of local data storage and maintenance. Since data owners and the cloud servers are not in the same trusted domain, the outsourced data may be at risk as the cloud server may no longer be fully trusted. Therefore, data integrity is of critical importance. Cloud should let the owners or a trusted third party to check for the integrity of their data storage without demanding a local copy of the data.[1]

Survey on secret sharing scheme with deduplication in cloud computing.

Data deduplication is one of the techniques used for eliminating duplicate copies of data which is widely used in cloud to reduce storage space and increase bandwidth. Convergent encryption has been extensively adopted for secure deduplication, in order to use efficiently and reliably manage a huge number of convergent keys. A baseline approach named as Dekey is used to distribute the convergent key which would be shared across multiple servers. A heavy computational cost is required to make n shares and recover the secret as a solution to this problem. Hence a new (k, L, n) -threshold ramp scheme is proposed which is perfect, idle and faster secret sharing scheme, every combination of k or more participants can recover the secret, but every group of less than k participants cannot obtain any information about the secret.[2]

Cooperative and Distributed Computation Offloading for Blockchain-Empowered Industrial Internet of Things.

A multihop cooperative and distributed computation offloading algorithm that considers the data processing tasks and the mining tasks together for blockchain-empowered IIoT.

1. We study the multihop computation offloading problem for both the data processing tasks and the mining tasks to minimize the economic cost of IIoT devices.
2. We formulate the offloading problem as a potential game in which the IIoT devices can make their decisions autonomously and prove the existence of Nash equilibrium (NE) for the game.
3. We design an efficient distributed algorithm based on exchanging messages between IIoT devices to achieve the NE with low computational complexity. Our experimental results demonstrate that our distributed algorithm scales as well as the number of IIoT devices increases and has the minimum system cost compared with other approaches.

The computing capabilities in IIoT are usually limited, whereas the blockchain mining tasks are computationally intensive. Most of the existing solutions for offloading assume that all IIoT devices can directly connect to the ESs or cloud data centers. [3]

DeyPoS: Deduplicatable Dynamic Proof of Storage for Multi-User Environments.

Dynamic Proof of Storage (PoS) is a useful cryptographic primitive that enables a user to check the integrity of outsourced files and to efficiently update the files in a cloud server. we introduce the concept of deduplicatable dynamic proof of storage and propose an efficient construction called DeyPoS, to achieve dynamic PoS and secure cross-user deduplication. Considering the challenges of structure diversity and private tag generation, we exploit a novel tool called Homomorphic Authenticated Tree (HAT). A practical multi-user cloud storage system needs the secure client-side cross-user deduplication technique, which allows a user to skip the uploading process and obtain the ownership of the files immediately, when other owners of the same files have uploaded them to the cloud server. [4]

Secure verifiable database supporting efficient dynamic operations in cloud computing

in this paper, we propose a secure verifiable database scheme that is based on the polynomial commitment for cloud computing, which can realize the verifiability of database records in the cloud. moreover, the proposed scheme can support public verifiability in that all clients in the system can verify the database. in addition, we use the bls signature and the index-hash table to construct dynamic operations for the database. security analysis shows that our scheme can achieve real-world security requirements. the simulation results show that our scheme is more efficient than similar schemes. The security analysis shows that our scheme can achieve the properties of security, correctness, verifiability and accountability. In the performance analysis, we compare our scheme with two similar schemes. The comparison results and simulation results indicate that our scheme is more efficient than similar schemes, which demonstrates that our scheme can be well used in secure verifiability for databases in cloud computing.[5]

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

We present a (probabilistic) public key encryption (PKE) scheme such that when being implemented in a bilinear group, anyone is able to check whether two ciphertexts are encryptions of the same message. Interestingly, bilinear map operations are not required in key generation, encryption or decryption procedures of the PKE scheme, but is only required when people want to do an equality test (on the encrypted messages) between two cipher texts that may be generated using different public keys. We show that our PKE scheme can be used in different applications such as searchable encryption and partitioning encrypted data. Moreover, we show that when being implemented in a non-bilinear group, the security of our PKE scheme can be strengthened from One-Way CCA to a weak form of IND-CCA.

3.2 PROPOSED SYSTEM

This paper proposes a blockchain-based deduplicatable information examining instrument. We first plan a customer side information deduplication conspire dependent on bilinear-pair procedures to diminish the weight on clients and specialist organizations. On this premise, we accomplish a reliable and effective information evaluating system that assists with checking information uprightness by utilizing both the blockchain procedure .

3.3 REQUIREMENT ANALYSIS

3.3.1 Input Requirements

PROCESSOR	:	INTEL CORE I9-9980XE
RAM	:	4GB DD RAM
MONITOR	:	15" COLOR
HARD DISK	:	250 GB

3.3.2 Output requirements

FRONT END	:	J2EE (JSP, SERVLETS)
BACK END	:	MY SQL 5.5
OPERATING SYSTEM	:	WINDOWS 07
IDE	:	ECLIPSE

3.4 Software Environment

THE JAVA FRAMEWORK

Java is a programming language originally developed by James Gosling at Sun Microsystems 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 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 is widely used from application software to web applications. The Java framework is a new platform independent that simplifies application development. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Why Software Developers Choose Java

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform

- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

Some Ways Software Developers Learn Java

- Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

OBJECT ORIENTED

To be an Object Oriented language, any language must follow at least the four characteristics.

1. Inheritance : It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding additional features as needed.
2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.
3. Polymorphism: As the name suggests one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
4. Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

JAVASERVER PAGES - AN OVERVIEW

Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provide excellent server side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy. JSP pages are efficient, it loads into the web servers memory on receiving the request very first time and the subsequent calls are served within a very short period of time. In today's environment most web sites servers dynamic pages based on user request. Database is very convenient way to store the data of users and other things. JDBC provide excellent database connectivity in heterogeneous database environment. Using JSP and JDBC its very creasy to develop database driven web application.

Java is known for its characteristic of "write once, run anywhere." JSP pages are platf
Java Server Pages Java Server Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well-defined way. The JavaServer Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates, and Java code, which is secure, fast, and independent of server platforms. JSP has been built on top of the Servlet API and utilizes Servlet semantics. JSP has become the preferred request handler and response mechanism. Although JSP technology is going to be a powerful successor to basic Servlets, they have an evolutionary relationship and can be used in a cooperative and complementary manner.

Servlets are powerful and sometimes they are a bit cumbersome when it comes to generating complex HTML. Most servlets contain a little code that handles application logic and a lot more code that handles output formatting. This can make it difficult to separate and reuse portions of the code when a different output format is needed. For these reasons, web application developers turn towards JSP as their preferred servlet environment.

EVOLUTION OF WEB APPLICATIONS

Over the last few years, web server applications have evolved from static to dynamic applications. This evolution became necessary due to some deficiencies in earlier web site design. For example, to put more of business processes on the web, whether in business-to-consumer (B2C) or business-to-business (B2B) markets, conventional web site design technologies are not enough. The main issues, every developer faces when developing web applications, are:

1. Scalability - a successful site will have more users and as the number of users is increasing fastly, the web applications have to scale correspondingly.
 2. Integration of data and business logic - the web is just another way to conduct business, and so it should be able to use the same middle-tier and data-access code.
 3. Manageability - web sites just keep getting bigger and we need some viable mechanism to manage the ever-increasing content and its interaction with business systems.
 4. Personalization - adding a personal touch to the web page becomes an essential factor to keep our customer coming back again. Knowing their preferences, allowing them to configure the information they view, remembering their past transactions or frequent search keywords are all important in providing feedback and interaction from what is otherwise a fairly one-sided conversation.
-
1. Serve HTML and XML, and stream data to the web client
 2. Separate presentation, logic and data
 3. Interface to databases, other Java applications, CORBA, directory and mail services
 4. Make use of application server middleware to provide transactional support.
 5. Track client sessions.

SERVLETS

Earlier in client- server computing, each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that is mostly supported by all the browsers and web pages are displayed to the client as static documents. A web page can merely displays static content and it also lets the user navigate through the content, but a web application provides a more interactive experience.

Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing and unloading the Servlets and JSP. While servlets can be used to extend the functionality of any Java- enabled server. They are mostly used to extend web servers, and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server side dynamic content solutions, so before going forward it is very important to know the difference between CGI and the Servlets.

JAVA SERVLETS

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable. Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servlets can be handled by separate threads within the same process. Servlets are also portable and platform independent.

A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the web pages to the client and to an application by using the web browser and the HTTP protocols respectively. The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

Chapter 4

SYSTEM DESIGN

4.1 ER diagram

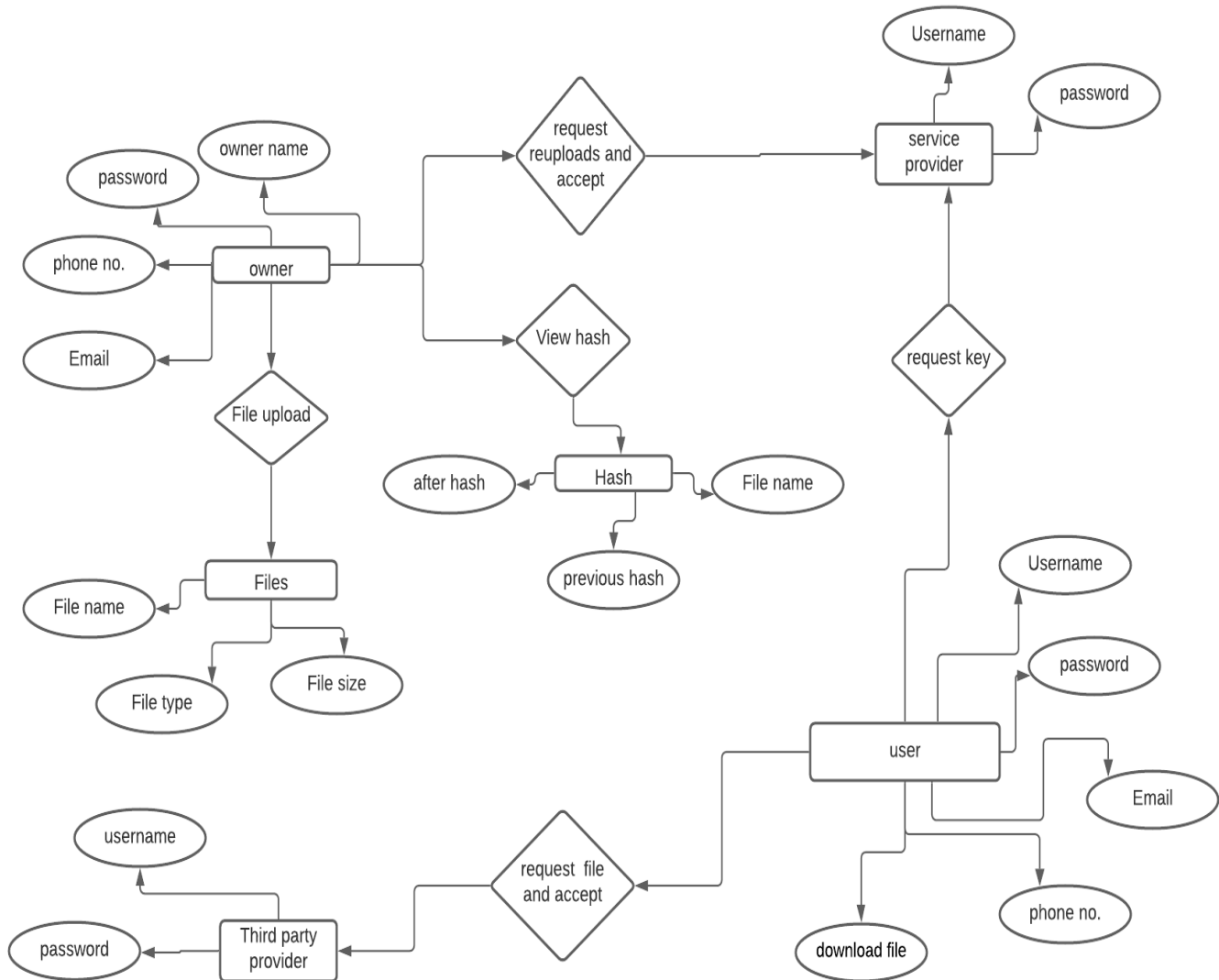


Fig 1: ER diagram describes interrelated things of interest in specific domain of knowledge

Uml diagrams

Use Case Diagram:

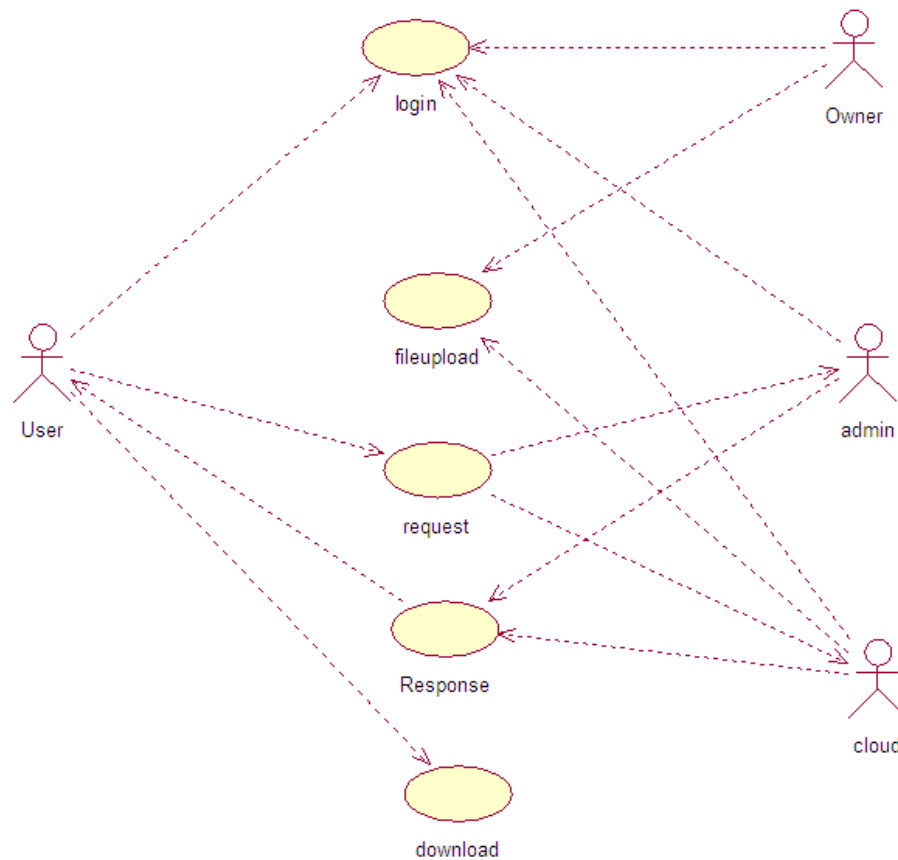


Fig2 : A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

EXPLANATION:

In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

CLASS DIAGRAM:

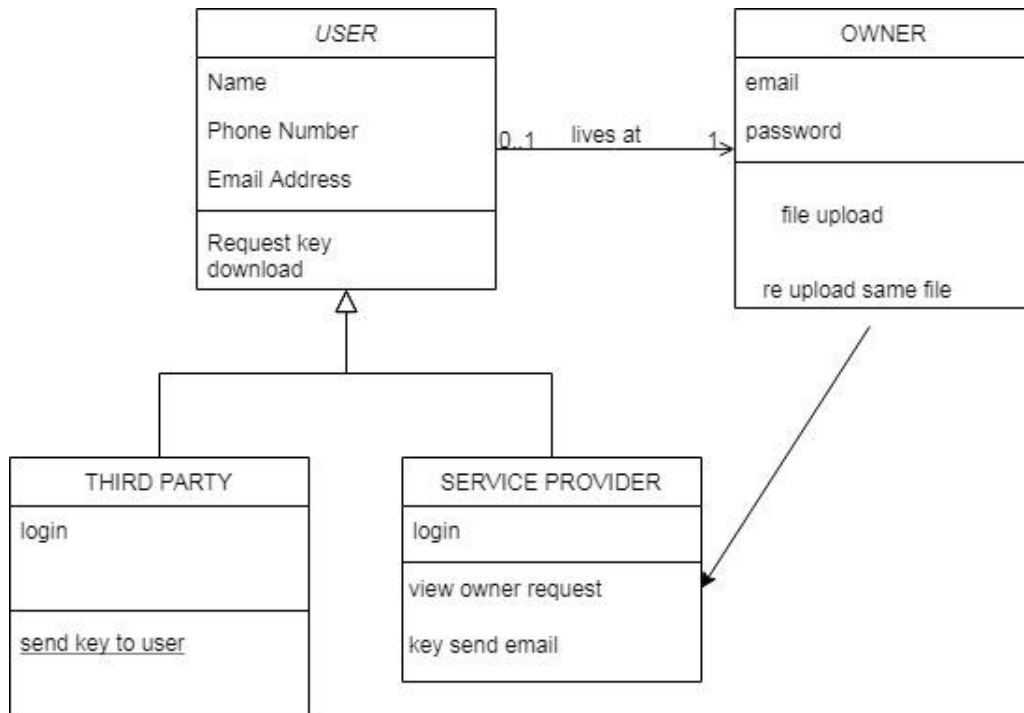


Fig 3: A class diagram is an illustration of the relationships and source code dependencies among classes

EXPLANATION:

in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

Object Diagram:

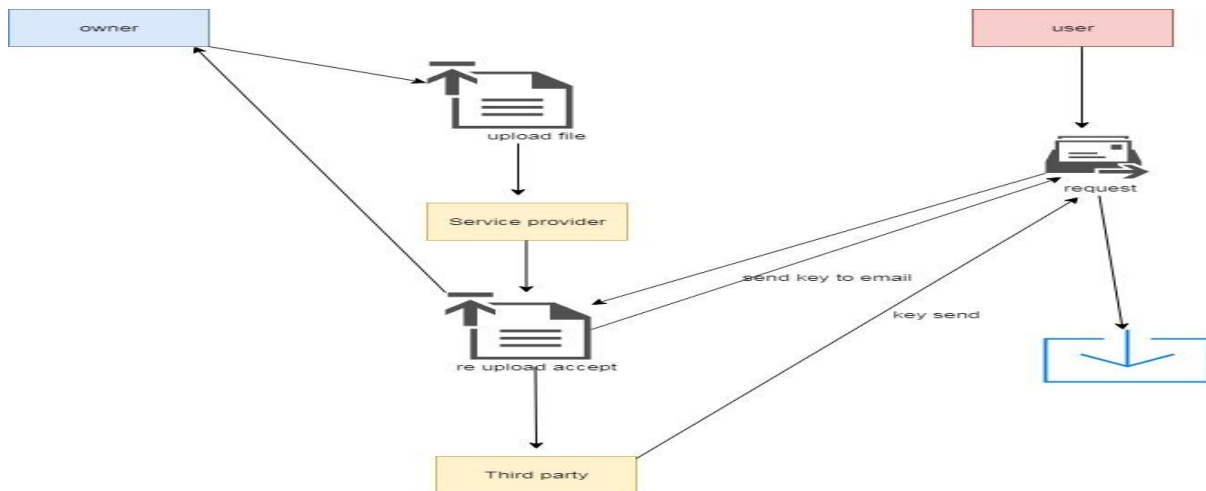


Fig4: Object is an instance of a particular moment in runtime, including objects and data values.

EXPLANATION:

A static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships at a point in time. It may be considered a special case of a class diagram or a communication diagram.

Sequence Diagram:

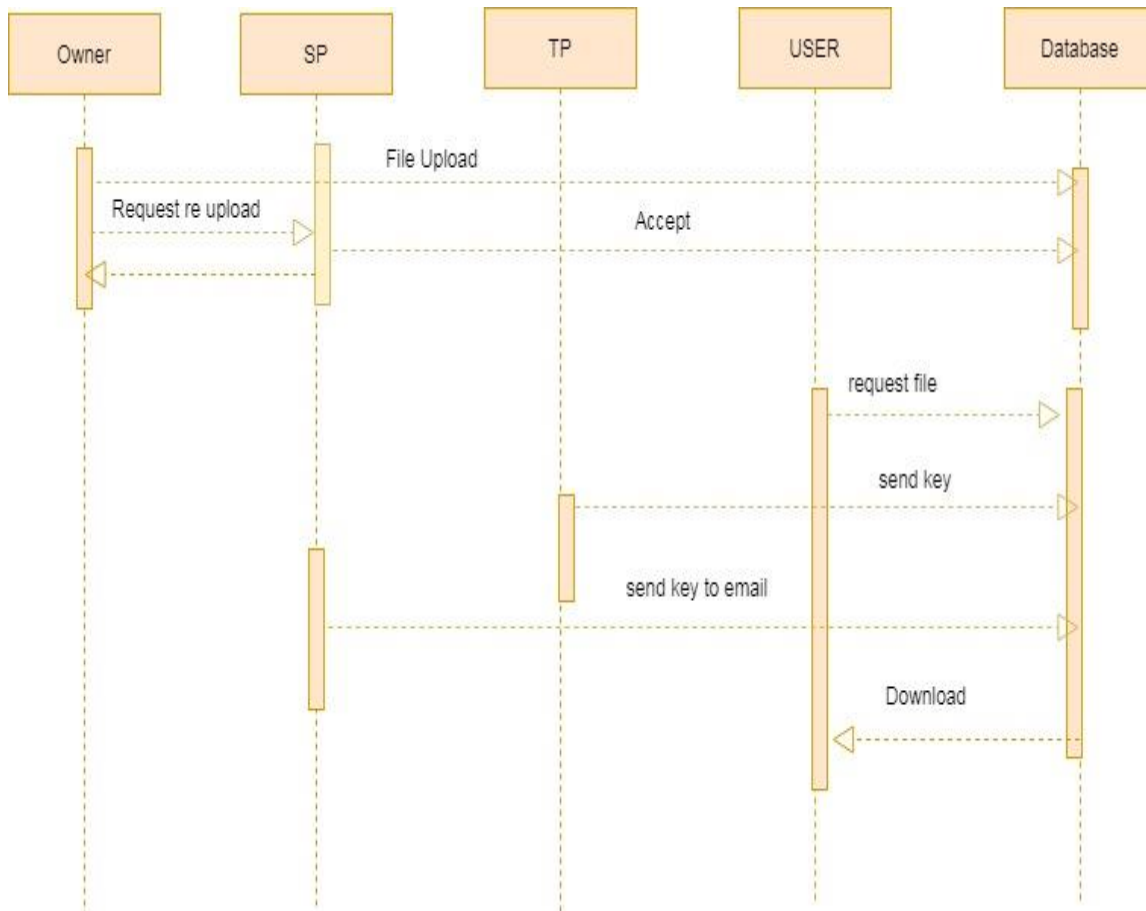


Fig5: A sequence diagram shows object interactions arranged in time sequence.

EXPLANATION:

It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

CHAPTER -5

System Architecture

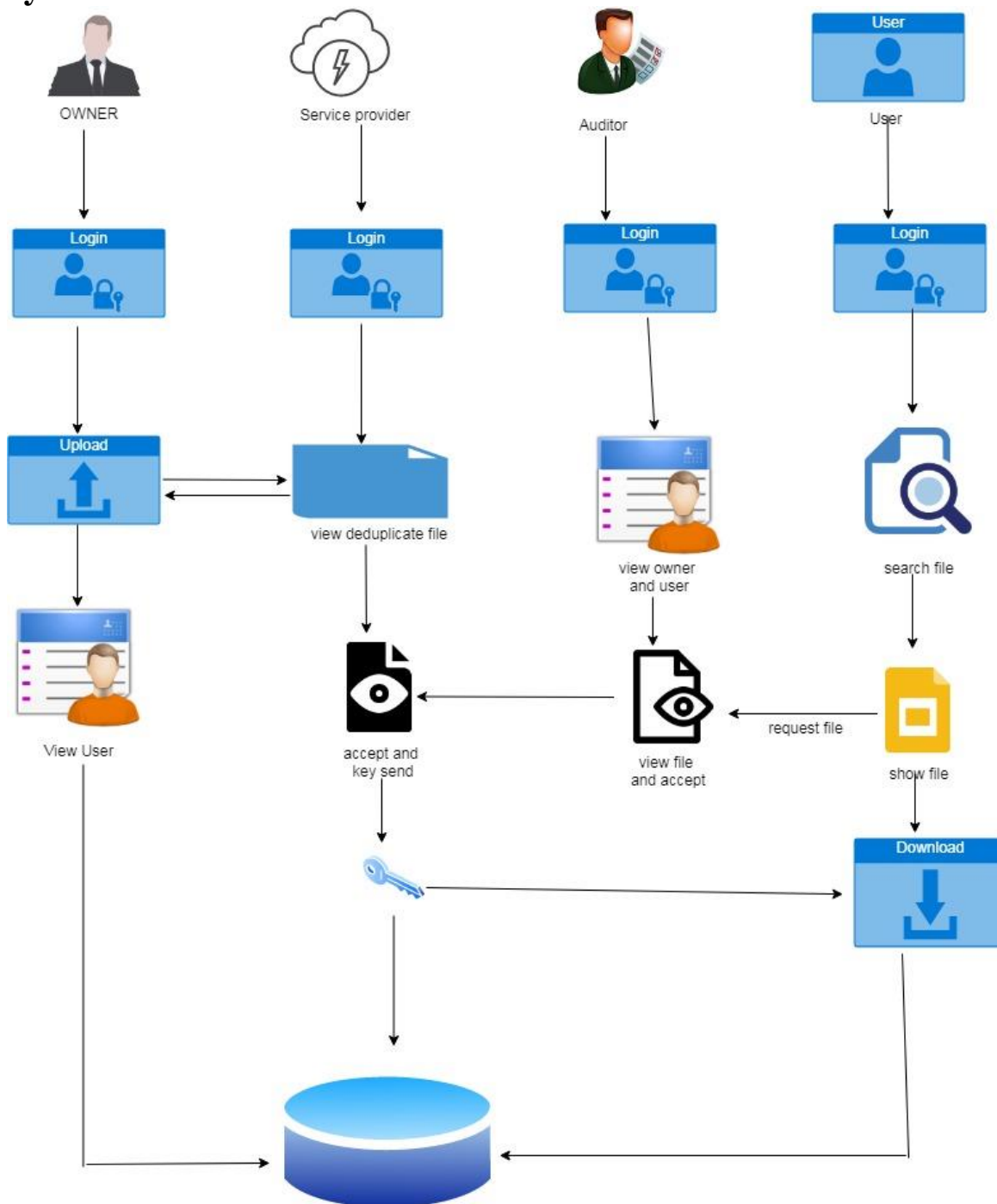


Fig6: System architecture is the conceptual model that defines the structure, behavior, of a system.

An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages.

5.1 Modules design specification

1. INITIALIZE AND EXCHANGE OF FILES

This is the primary module of our undertaking. The significant job for the client is to move login window to client window. This module has made for the security reason. In this login page we need to enter login client id and secret phrase. It will check username and secret word is coordinate or not (substantial client id and legitimate secret phrase). On the off chance that we enter any invalid username or secret word, we can't go into login window to client window it will shows mistake message. In this way, we are keeping from unapproved client going into the login window to client window. It will give a decent security to our venture. In this way, worker contain client id and secret word worker likewise check the verification of the client. It well improves the security and keeping from unapproved client goes into the organization. In our venture we are utilizing JSP for making plan. Here we approve the login client and worker validation.

2. ANALYZE

In this module, after login the proprietor will transfer utilizing Blockchain (hash works) the document subtleties and it will be put away in the data set. offered subtleties to login and see what are generally the documents transferred by proprietor.

3.FORECAST:

In this module, owner solicitation a record to specialist co-op which document need to re transferring give acknowledge and proprietor reuploading an equivalent document name and put away an information base. the client will send the record solicitation to the outsider for which documents, the client needs the entrance. without the consent structure the outsider and specialist co-op, the client can't ready to download the document.

4.COMPUTE

In this module, The Third party send key and specialist co-op send key through client mail will be giving the acknowledgment to the client for which document needs the entrance. After the acknowledgment, the record key will be shipped off the user.after getting the key from the outsider and specialist co-op, the client can download the document utilizing the key given by the specialist co-op.

5.2 Algorithm

Homomorphic encryption is an encryption algorithm that is also a homomorphism. It allows the recipient of encrypted data to encrypt the result of some computation without knowing the inputs. The most popular example for the use of homomorphic encryption is where a data owner wants to send data up to the cloud for processing, but does not trust a service provider with their data. Using a homomorphic encryption scheme, the data owner encrypts their data and sends it to the server. The server performs the relevant computations on the data without ever decrypting it and sends the encrypted results to the data owner. The data owner is the only one able to decrypt the results, since they alone have the secret key. A much stronger and secure encryption than private and public key encryption developed is Homomorphic encryption. Homomorphic encryption is a technique of encrypting the plaintext and performing computations on the encrypted text without disclosing the plaintext i.e. without decrypting it. Homomorphic Encryption can be called building blocks of modern day cryptography as it is used in many tools of cryptography.

The existing HE constructions are homomorphic with respect to two basic operations: some kind of addition and some kind of multiplication (e.g. $++$ and \times over the integers or the binary operations $\{XOR\}$ XOR and $\{AND\}$ AND, etc.). What we mean is that the scheme allows the efficient computation of $c_{\{add\}}$ from the individual ciphertexts $c_1 = \text{Enc}(pk, m_1)$ and $c_2 = \text{Enc}(pk, m_2)$ such that the decryption of $c_{\{add\}}$ yields $m_1 + m_2$.

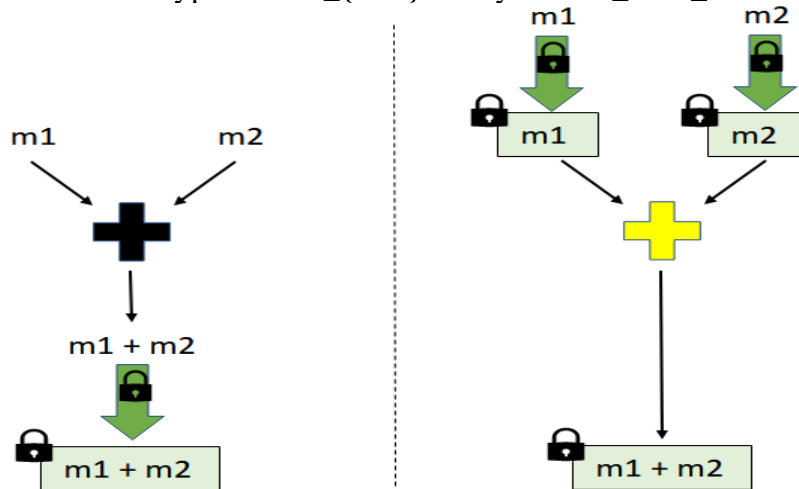


Fig 7:

Analogously, the cipher text $c_{\{mul\}}$ corresponding to multiplication, that decrypts to $m_1 \times m_2$, is efficiently computable from the individual ciphertexts $c_1 = \text{Enc}(pk, m_1)$ and $c_2 = \text{Enc}(pk, m_2)$, respectively.

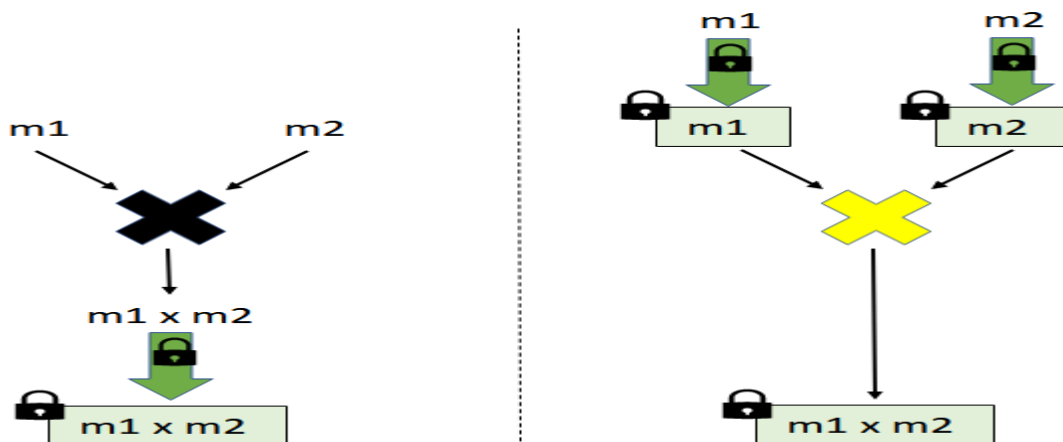


Fig:8

CHAPTER 6

SYSTEM IMPLEMENTATION

OWNER REGISTRATION:

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta charset="utf-8">
<title>RegistrationForm_v1 by Colorlib</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="fonts1/material-design-iconic-font/css/material-design-
iconic-font.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<link rel="stylesheet" href="css1/style.css">
<script>
$("#formCheckPassword").validate({
  rules: {
    password: {
      required: true,
      minlength: 6,
```

maxlength: 10, } ,

cfmPassword: {

equalTo: "#pass"

minlength: 6,

maxlength: 10

} },

messages:{

password: {

required:"the password is required" } }));

</script></head><body>

<div class="wrapper" style="background-image: url('images/GatorEyes.jpg');">

<div class="inner"><div class="image-holder">

</div>

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

<h3>Registration Form</h3>

<div class="form-wrapper">

<input type="text" placeholder="Username" name="Username" class="form-control">

<i class="zmdi zmdi-account"></i>

</div>

<div class="form-wrapper">

<input type="text" placeholder="Email Address" class="form-control" name="mail">

<i class="zmdi zmdi-email"></i></div><div class="form-wrapper">

```
<select name="gen" id="" class="form-control">
<option value="" disabled selected>Gender</option>
<option >Male</option>
<option >Female</option>
<option >Other</option>
</select>
<i class="zmdi zmdi-caret-down" style="font-size: 17px"></i>
</div>
<div class="form-wrapper">
<input type="tel" name="phno" placeholder="Phone Number" class="form-control"
minlength="10" maxlength="10" required>
<i class="zmdi zmdi-phone"></i>
</div>
<div class="form-wrapper">
<input type="password" placeholder="Password" name="pass" class="form-control">
<i class="zmdi zmdi-lock"></i>
</div>
<div class="form-wrapper">
<input type="password" placeholder="Confirm Password" name="cpass"
class="form-control">
<i class="zmdi zmdi-lock"></i>
</div>
<button>Register
<i class="zmdi zmdi-arrow-right"></i>
```

```

/button>
<div class="form-wrapper">
<h2 class="signup" style="font-size:20px;font-family:Times New Roman">Already
have account yet? <a href="Ownerlogin.jsp" class="signuplink"
style="color:#cc0000">LOGIN</a></h2>

</div></form></div></div>

</body>
</html>

```

OWNERVIEW:

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
<meta name="description" content="">
<meta name="keywords" content="">
<meta name="author" content="">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">

```

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

```
<link rel="stylesheet" href="css/font-awesome.min.css">
```

```
<link rel="stylesheet" href="css/aos.css">
```

```
<!-- MAIN CSS -->
```

```
<link rel="stylesheet" href="css1/tooplate-gymso-style.css">
```

```
<!--
```

```
Tooplate 2119 Gymso Fitness
```

```
https://www.tooplate.com/view/2119-gymso-fitness
```

```
-->
```

```
</head>
```

```
<body data-spy="scroll" data-target="#navbarNav" data-offset="50">
```

```
<%Stringemail=request.getSession().getAttribute("email").toString();
```

```
System.out.println("email-----"+email);    %>
```

```
<!-- MENU BAR -->
```

```
<nav class="navbar navbar-expand-lg fixed-top">
```

```
<div class="container">
```

```
<a class="navbar-brand" href="index.html"> Public-Key Encryption</a>
```

```
<button    class="navbar-toggler"    type="button"    data-toggle="collapse"    data-  
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"  
    aria-label="Toggle navigation">
```


</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ml-lg-auto">

<li class="nav-item">

Home

<li class="nav-item">

FILE UPLOAD

<li class="nav-item">

VIEW FILES

<li class="nav-item">

VIEW USER

<li class="nav-item">

LOGOUT

</div></div></nav>

```
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/aos.js"></script>
<script src="js/smoothscroll.js"></script>
<script src="js/custom.js"></script>
</body>
</html>
```

SERVICELOGIN:

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"% >
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="UTF-8" />
<meta name="keywords"
content="Desk Login form Responsive web template, Bootstrap Web Templates, Flat
Web Templates, Android Compatible web template, Smartphone Compatible web
template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web
design" />
```

```
<!-- //Meta tag Keywords -->
<link href="//fonts.googleapis.com/css?family=Mukta:200,300,400,500,600,700,800"
rel="stylesheet">
<!--/Style-CSS -->
<link rel="stylesheet" href="css4/style.css" type="text/css" media="all" />
<!--/Style-CSS -->
<style>
#grad1 {
height: 100px;
background: -webkit-linear-gradient(left, red , blue);
background: -o-linear-gradient(right, red, blue);
background: -moz-linear-gradient(right, red, blue);
background: linear-gradient(to right, red , blue);
}
</style></head>
<body>
<div id = "grad1">
<section class="w3l-forms-main-61">
<div class="form-inner">
<div class="wrapper">
<div class="d-grid top-form">
<div class="logo">
<a class="brand-logo" href="index.html"><span><span class="fa fa-viadeo"
aria-hidden="true"></span> Service Provider Login form</span></a>
```



```
<!-- if logo is image enable this
<a class="brand-logo" href="#index.html">

</a> -->
</div>
</div>
<div class="form-bg-blur">
<div class="form-61">
<h4 class="form-head">User Login</h4>
<form action="Serviceprovider" method="post">
<div class="">
<p class="text-head">Username</p>
<input type="text" name="username" class="input" placeholder="username" required
/>
</div>
<div class="">
<p class="text-head">Password</p>
<input type="password" name="password" class="input" placeholder="password"
required />
</div>
<label class="remember">
<input type="checkbox">
<span class="checkmark"></span>Keep me logged in
</label>
```

```

<button type="submit" class="signinbutton btn">LOGIN</button>
<p class="signup">Forgot password?<a href="#forgot" class="signuplink">Click
here</a></p>
</form></div></div></div>
</div></section></div>
</body>
</html>

```

VIEW DOWNLOAD:

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @page import="Database.database"%>
<% @page import="java.sql.ResultSet"%>
<% @page import="java.sql.PreparedStatement" %>
<% @page import="java.sql.*" %>
<% @page import="java.util.*" %>
<% @ page import="java.util.*" %>
<% @ page import="java.util.Random" %>
<% @ page import="javax.servlet.http.HttpSession" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<meta charset="utf-8">

```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/bootstrap.min.css">
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
<title>QR-Code Generator</title>
<style>
*{margin:0px; padding:0px;font-family: Helvetica, Arial, sans-serif;}
h1 { text-align: center; text-shadow: 2px 2px 0px rgba(255,255,255,.7), 5px 7px 0px
rgba(0, 0, 0, 0.1); font-size:50px; margin-top:40px; color:#fff; }
input[type=text]{
    width: 90%;
    padding: 12px 20px;
    margin: 8px 26px;
    display: inline-block;
    border: 1px solid #ccc;
    box-sizing: border-box;
    font-size:16px;
}
button {
    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
    margin: 8px 26px;
    border: none;
```

```

    cursor: pointer;
    width: 90%;
    font-size: 20px;
}
button:hover {
    opacity: 0.8;
}
</style>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head>
<% String email=request.getSession().getAttribute("email").toString();
    System.out.println("email-----"+email);    %>
<body background="../background1.png">
<div class="container-fluid" style="background-color: silver">
<div class="container" style="margin-top: 20px;">
<h1 style="color: royalblue;font-size:50px"> PKE-MET: PUBLIC-KEY
ENCRYPTION WITH MULTI-CIPHERTEXT EQUALITY TEST IN CLOUD
COMPUTING</h1>
<nav class="navbar navbar-inverse">
<div class="container-fluid">
<div class="navbar-header"></div>
</div></nav>
<div class="col-sm-2"></div>

```

```
<div class="col-sm-8">
```

```
<center>
```

```
<table class="table table-sm" style="margin-top: 100px">
```

```
<tr>
```

```
<th>S.no</th>
```

```
<th>Filename</th>
```

```
<th>User Email</th>
```

```
<th>File Key</th>
```

```
<th>Csp Key</th>
```

```
<th>Download</th>
```

```
</tr>
```

```
<% String sno=request.getParameter("id");
```

```
    System.out.println("sno:"+sno);
```

```
String fname=request.getParameter("fname");
```

```
System.out.println("fname:"+fname);
```

```
String ckey=request.getParameter("ckey");
```

```
System.out.println("ckey:"+ckey);
```

```
String fkey=request.getParameter("fkey");
```

```
System.out.println("fkey:"+fkey);
```

```
String file="";
```

```
%>
```

```
<%-- Connection con = database.create();
```

```
PreparedStatement p = con.prepareStatement("SELECT * FROM  
enabling.cloudrequest where email='"+name+"'");
```

```

ResultSet rs=p.executeQuery();
while(rs.next())
{
sno=rs.getString(1);
owner=rs.getString(2);
mail=rs.getString(3);
key=rs.getString(4);
tra=rs.getString(5);
file=rs.getString(6);
%> --%>

<tr>

<td><%=sno%></td>

<td><%=fname%></td>

<td><%=email%></td>

<td>*****</td>

<td>*****</td>

<td><a
href="Download?fname=<%=fname%>&&ckey=<%=ckey%>&&fkey=<%=fkey%>
"><button class="btn btn-success btn-xs" style="font-size:20px; padding:5px;">
Download</button></a></td>

</tr>

<%-- <% } %> --%>

</table>

</center>

```

```
</div>
<div class="col-sm-2"></div>
</center>
</div>
<h1>QR-Code Generator</h1>
<div id="form-wrapper" style="width:46%; float:left; border:5px solid
rgba(255,255,255,0.6); margin-top:20px; padding:10px">
<form id="generator">
<label for="codeSize" style="font-size:20px; margin-right:20px; color:#fff;">Select
QR Code Size:</label>
<select id="codeSize" name="codeSize" style="width:260px; height:40px; ">
<option value="75">XSmall</option>
<option value="155">Small</option>
<option value="186">Medium</option>
<option value="248" selected="selected">Large</option>
<option value="300">XLarge</option>
<option value="450">XXLarge</option>
</select>
<input type="hidden" onclick="myFunction()" id="codeData" name="codeData"
size="50" value="<%=fkey %>" placeholder="Enter a url or text" style="margin-
top:20px" autocomplete="off"/ >
<br>
<button id="generate">generate</button>
</form>
```

```

<div id="alert" style="height:20px; text-align:center; margin:10px auto"></div>
</div>
<div style="float:right;">
<div id="image" style="margin:auto"></div>
<div id="link" style="margin-top:10px; text-align:center"></div>
</div>
<div id="code" style="float:left; width:100%; height:20px; text-align:center; margin-
top:10px"></div>
<script>
function myFunction() {
    document.getElementById("alert").innerHTML = "";
    }

$("#generate").on("click", function () {
var data = $("#codeData").val();
var size = $("#codeSize").val();
if(data == "") {
    $("#alert").append("<p style='color:#fff;font-size:20px'>Please Enter A Url Or
Text</p>"); // If Input Is Blank
    return false;
} else {
    if( $("#image").is(':empty'))
    {
        //QR Code Image
        $("#image").append("<img src='http://chart.apis.google.com/chart?cht=qr&chl="
+ data + "&chs=" + size + "' alt='qr' />"); //This Provide An Image Download Link
    }
}
}

```



```

    $("#link").append("<a style='color:#fff;'
href='http://chart.apis.google.com/chart?cht=qr&chl=" + data + "&chs=" + size +
">Download QR Code</a>");

//This Provide the Image Link Path In Text
    //    $("#code").append("<p style='color:#fff;'><strong>Image    Link:</strong>
http://chart.apis.google.com/chart?cht=qr&chl=" + data + "&chs=" + size + "</p>");
    return false;
} else {
    $("#image").html("");
    $("#link").html("");
    $("#code").html("");
    //QR Code Image
    $("#image").append("<img src='http://chart.apis.google.com/chart?cht=qr&chl="
+ data + "&chs=" + size + "' alt='qr' />");
    //This Provide An Image Download Link
    $("#link").append("<a style='color:#fff;'
href='http://chart.apis.google.com/chart?cht=qr&chl=" + data + "&chs=" + size +
">Download QR Code</a>");
    //This Provide the Image Link Path In Text
    $("#code").append("<p style='color:#fff;'><strong>Image    Link:</strong>
http://chart.apis.google.com/chart?cht=qr&chl=" + data + "&chs=" + size + "</p>");
    return false;    }
} });
</script>

```

</body></html>

LOGOUT:

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @page import="java.io.PrintWriter"%>
<% @page import="Database.database"%>
<% @page import="java.sql.ResultSet"%>
<% @page import="java.sql.PreparedStatement" %>
<% @page import="java.sql.*" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>FILE UPLOAD</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="utf-8">
<meta http-equiv="refresh" content="3; URL=index.jsp">
</head>
<%-- <%
response.setContentType("text/html");
```

```
PrintWriter out=response.getWriter();
```

```
    request.getRequestDispatcher("link.html").include(request, response);
```

```
    HttpSession session=request.getSession();
```

```
    session.invalidate();
```

```
    %>    --%>
```

```
<% session.invalidate();
```

```
response.sendRedirect("Index.jsp"); %>
```

```
<body bgcolor="#99ffff">
```

```
<div>
```

```
<center>
```

```

```

```
<h1 align="center">Logout Successfully</h1>
```

```
</center>
```

```
</div></body></html>
```

CHAPTER 7

SOFTWARE TESTING

7.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

7.3 TYPES OF TESTS

7.3.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit

tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.3.2 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

7.3.3 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.3.4 PERFORMANCE TEST

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

7.3.5 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

7.3.6 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

ACCEPTANCE TESTING FOR DATA SYNCHRONIZATION:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updating process

7.4 BUILD THE TEST PLAN

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identify the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Enter valid name, email, phno, password, confirm password	Name: xxx Email: xxx@gmail.com Phno:9423539453 Password:*** Confirm password:***	xxx xxx@gmail.com 9423539453 *** ***	xxx xxx@gmail.com 9423539453 *** ***	PASSED
2.	Enter valid email and password	Email: xxx@gmail.com Password: ***	xxx@gmail.com ***	xxx@gmail.com ***	PASSED

3.	Compare email and password with registered field	Email: xxx@gmail.com Password: ***	Owner Page	Owner Page	PASSED
----	--	---	------------	------------	--------

S. NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Enter valid username and password	Username: Service Password: ***	Service ***	Service ***	PASSED
2.	Compare username and password with registered field	Username: Service Password: ***	Service Provider Page	Service Provider Page	PASSED

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Enter valid username, email, password, confirm password and phno	Username: xxx Email: xxx@gmail.com Password: *** Confirm password:*** Phno:9453623491	xxx xxx@gmail.com *** *** 9453623491	xxx xxx@gmail.com *** *** 9453623491	PASSED
2.	Enter valid email and password	Email: xxx@gmail.com Password: ***	xxx@gmail.com ***	xxx@gmail.com ***	PASSED
3.	Compare email and password with registered field	Email: xxx@gmail.com Password: ***	User Page	User Page	PASSED

S.NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Enter valid username and password	Username: Admin Password:***	Admin ***	Admin ***	PASSED
2.	Compare username and password with registered field	Username: Admin Password:***	Third Party Page	Third Party Page	PASSED

S. NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	To upload a file in owner page	Choose file: xxx.pdf	File uploaded successfully	File uploaded successfully	PASSED
2.	To request a file in user page	Request the file which uploaded	Request sent successfully	Request sent successfully	PASSED
3.	To accept the user request in third party page	Accept the user request	Request accepted successfully	Request accepted successfully	PASSED
4.	To send key to the user in third party page	Send the key to user	Key sent successfully	Key sent successfully	PASSED
5.	To download the file in user page	Filekey: xxx Cspkey: xxx	File downloaded	File downloaded	PASSED

S. NO	ACTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	To request a file in owner page	Request the file which uploaded. Message: xxx	Request sent successfully	Request sent successfully	PASSED
2.	To accept the owner request in service provider page	Accept the owner request	Request accepted successfully	Request accepted successfully	PASSED
3.	To upload a new file in owner page	Choose file: yyy.pdf	New file uploaded successfully	New file uploaded successfully	PASSED
4.	To download a new file in user page	Filekey: xxx Cspkey: xxx	File downloaded	File downloaded	PASSED

CHAPTER 8

8.1 Conclusion

This paper proposes a blockchain-enabled deduplicatable data auditing mechanism to improve the efficiency of the network storage service and protect the users' data. With the assistance of the deduplication technology, the network storage service provider can remove the duplicate data that the user outsourced and save only one copy, thereby reducing the storage burden. On this basis, we designed a blockchain-based data audit mechanism to ensure the integrity of outsourced data and avoid repeated audits by multiple tenants. The blockchain technique is introduced to record the data auditing log, thereby monitoring untrusted TPA during the data auditing process.

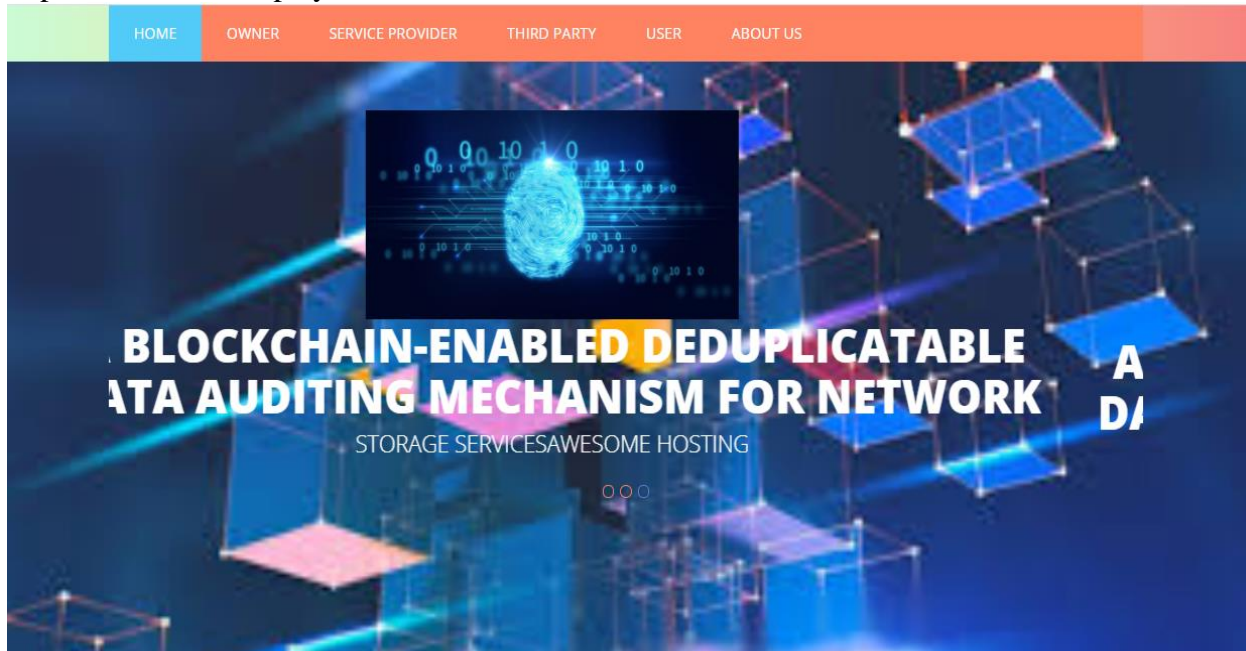
8.3 Future Enhancement

We think about utilizing a savvy agreement to actualize a blossom channel and an irregular number generator on the blockchain, to accomplish a programmed client-side information deduplication without the requirement for network capacity specialist co-op inclusion. Also, we tend to utilize the blockchain innovation to execute a totally decentralized information reviewing component without the requirement for a confided in TPA.

APPENDICES

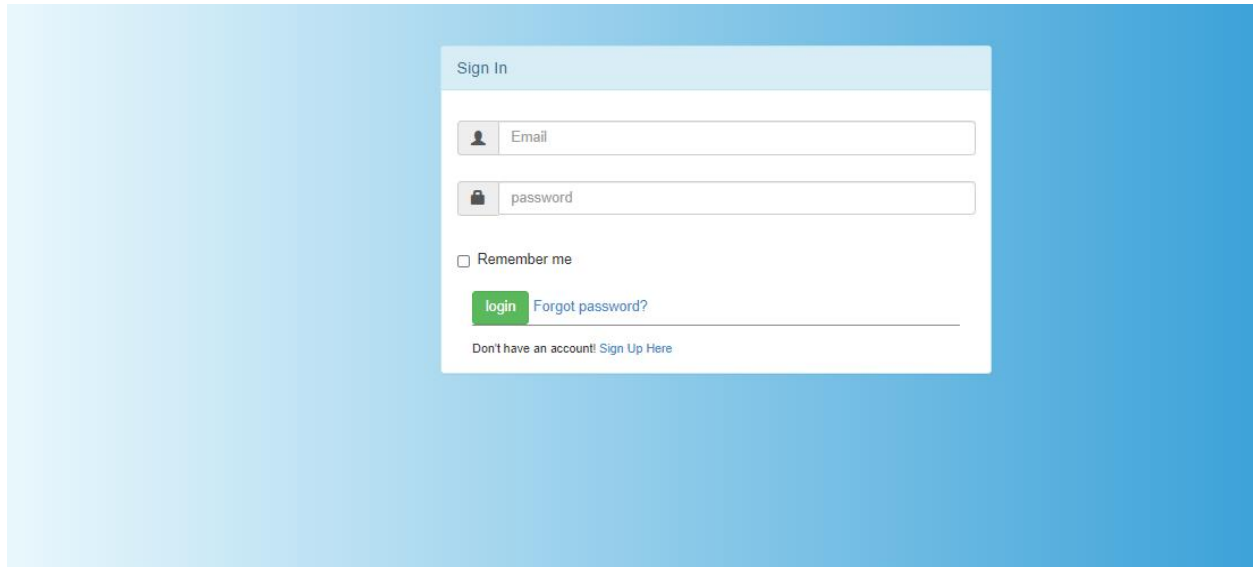
A.1 SAMPLE SCREENS

Home page of block chain based public uprightness check for distributed storage against hesitating inspectors will be displayed.



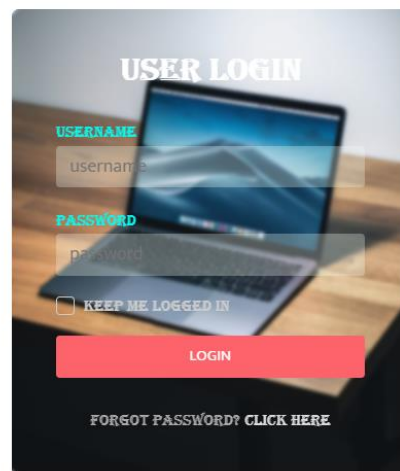
To achieve deduplicatable part, Owner and service provider are involved. Now, click on the owner tab, it will show tabs namely, view files to view the files which he has uploaded already.

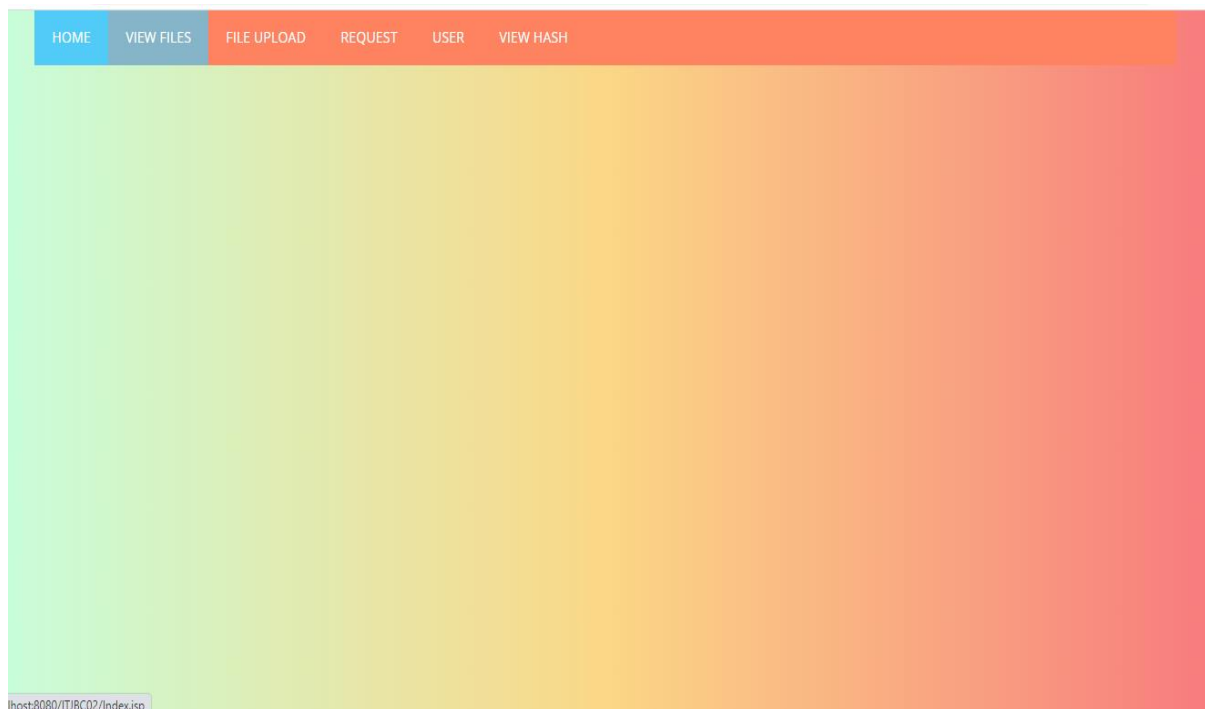
The screenshot shows a "Custom Validation Form" on a light gray background. On the left, there is a smartphone displaying a list of items. On the right, there is a black coffee cup and some papers. The form itself is a teal-colored box with the following fields: "NAME" with a red asterisk, "E-MAIL" with a red asterisk, "PHONE NUMBER" with a red asterisk, "PASSWORD" with a red asterisk, and "CONFIRM PASSWORD" with a red asterisk. Each field has a corresponding input box. At the bottom of the form is a dark blue "Submit" button.



Service provider login page

SERVICE PROVIDER LOGIN FORM





A Blockchain-enabled Deduplicatable Data Auditing Mechanism for Network Storage Services

[HOME](#) [UPLOAD](#) [LOGOUT](#)

57012

242CB

No file chosen

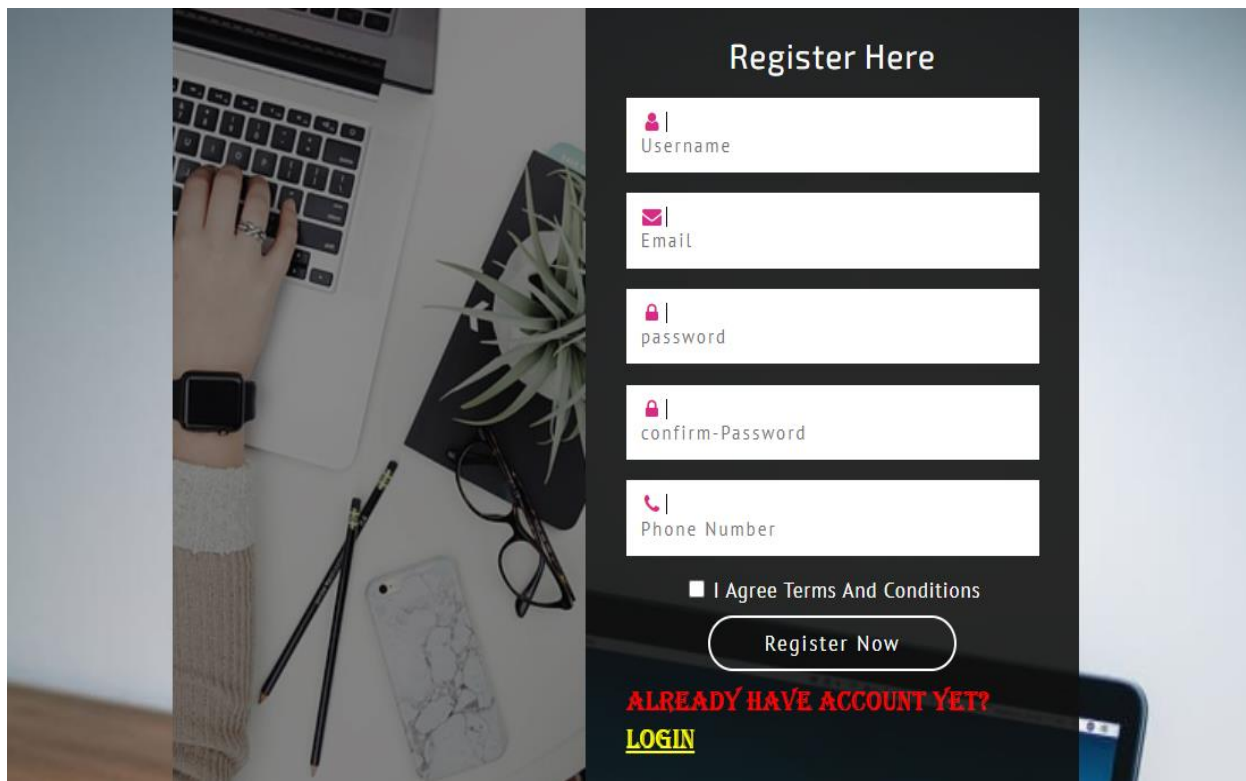
BLOCK CHAIN

FILENAME	USERNAME	FILESIZE	MESSAGE	STATUS	STATUS
NEW2019.PDF	SUBASH1996@GMAIL.COM	1035243	HF6CYHGUFJYY	<button>Accept</button>	<button>Delete</button>
NEW2019.PDF	SUBASH1996@GMAIL.COM	1035243	TYG HFTGUJYI	<button>Accept</button>	<button>Delete</button>
NEW2019.PDF	SUBASH1996@GMAIL.COM	1035243	TYHTUJTYUJ	<button>Accept</button>	<button>Delete</button>
NEW2019.PDF	SUBASH1996@GMAIL.COM	1035243	DFHGHYFGUJHTYUJ	<button>Accept</button>	<button>Delete</button>

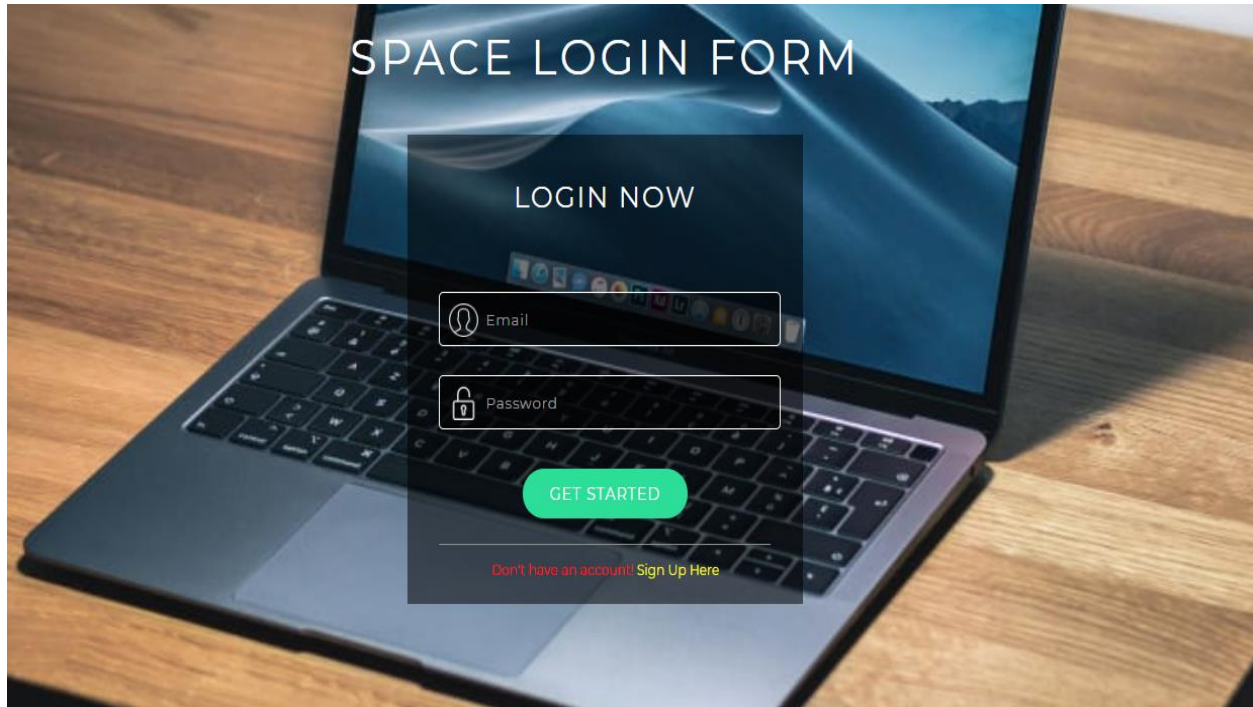
BLOCK CHAIN

FILENAME	USERNAME	OWNERNAME	FILEKEY	CSPKEY	STATUS	STATUS
NEW2019.PDF	SUBASHANBU1996@GMAIL.COM	SUBASH1996@GMAIL.COM	41195	3A3A1	ACTIVATED	<button>SEND</button>

Click on the owner request tab, there he can see the request of file reupload sent by the owner with the options accept or decline. The service provider can authenticate the owner can accept if he is verified.



Provide the user credentials, now user can see view files tab where all the files are available .If user needs particular file ,click on the request button



Now login the user login page with registered credentials.

A.2 PUBLICATIONS

“Blockchain-Based Public Uprightness Check For Distributed Storage Against Hesitating Inspectors”, **Institute for Technology and Research** ,(www.ijar.in)
ISBN: 978-93-90150-34-2 , Edn: 41 , May 2021 ,Available :
https://www.digitalxplore.org/up_proc/pdf/903-162087828917-23.pdf

REFERENCES:

- [1] D. Reinsel, J. Gantz, and J. Rydning, “Data age 2025: The digitization of the world: from edge to core,” Int. Data Corporation(IDC), pp. 1–28, 2018.
- [2] F. Zafar, A. Khan, S. U. R. Malik, M. Ahmed, A. Anjum, M. I. Khan, N. Javed, M. Alam, and F. Jamil, “A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends,” *Computers & Security*, vol. 65, pp. 29–49, 2017.
- [3] L. Zhou, A. Fu, S. Yu, M. Su, and B. Kuang, “Data integrity verification of the outsourced big data in the cloud environment: A survey,” *J. Netw. Comput. Appl.*, vol. 122, pp. 1–15, 2018.
- [4] J. Gratz and D. Reinsel, “The digital universe decade-are you ready?” IDC White Paper, pp. 1–16, 2010.
- [5] Y. Shin, D. Koo, and J. Hur, “A survey of secure data deduplication schemes for cloud storage systems,” *ACM Comput. Surveys (CSUR)*, vol. 49, no. 4, pp. 74–112, 2017.
- [6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 598–609.
- [7] H. Shacham and B. Waters, “Compact proofs of retrievability,” in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, 2008, pp. 90–107.
- [8] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, “Cooperative provable data possession for integrity verification in multicloud storage,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, 2012.
- [9] Y. Yu, Y. Li, B. Yang, W. Susilo, G. Yang, and J. Bai, “Attribute-based cloud data integrity auditing for secure outsourced storage,” *IEEE Trans. Emerg. Topics Comput.*, vol. 14, no. 8, pp. 1–13, 2017.

- [10] J. Shen, D. Liu, M. Z. A. Bhuiyan, J. Shen, X. Sun, and A. Castiglione, "Secure verifiable database supporting efficient dynamic operations in cloud computing," *IEEE Trans. Emerg. Topics Comput.*, pp. 1–11, 2017, doi: 10.1109/TETC.2017.2776402.
- [11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proc. 18th ACM Conf. Comput. Commun. Security*, 2011, pp. 491–500.
- [12] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "Cloudedup: Secure " deduplication with encrypted data for cloud storage," in *Proc. 5th Int. Conf. Cloud Comput. Technol. Sci.*, vol. 1, 2013, pp. 363–370.
- [13] S. Li, C. Xu, and Y. Zhang, "CSED: Client-side encrypted deduplication scheme based on proofs of ownership for cloud storage," *J. Inf. Security Appl.*, vol. 46, pp. 250–258, 2019.
- [14] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in *Proc. 2nd ACM Conf. Data Appl. Security Privacy*, 2012, pp. 1–12.
- [15] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2013, pp. 145–153.