# Signature-Based Public Sensitive Data Sharing for Cloud Storage

**A PROJECT REPORT**

***Submitted by***

**SRIRAM R [REGISTER NO: 211419104266]**

**RITHIK M R [REGISTER NO: 211419104223]**

**VIJAYVIKRAMAN V [REGISTER NO: 211419104304]**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**


**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**APRIL 2023**

# BONAFIDE CERTIFICATE

Certified that this project report **"Signature-Based Public Sensitive Data Sharing for Cloud Storage"** is the bonafide work of **"SRIRAM R (211419104266) , RITHIK M R (211419104223) , VIJAYVIKRAMAN V (211419104304)"**who carried out the project work under supervision.

.

**SIGNATURE**

**Dr.L.JABASHEELA,M.E.,Ph.D.,**
**HEAD OF THE DEPARTMENT**

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

**SIGNATURE**

**Mrs.A.KANCHANA,M.E.**
**SUPERVISOR**
**ASSISTANT PROFFESOR**

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**

## DECLARATION BY THE STUDENT

We **SRIRAM R (211419104266)** , **RITHIK M R (211419104223)** , **VIJAYVIKRAMAN V (211419104304)** hereby declare that this project report titled “Signature-Based Public Sensitive Data Sharing for Cloud Storage” , under the guidance of **Mrs.A.KANCHANA,M.E.** is the orginial work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**SRIRAM R**

**RITHIK M R**

**VIJAYVIKRAMAN V**

# ACKNOWLEDGEMENT

We would like to 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 express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Mrs.A.KANCHANA , M.E.** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**SRIRAM R (211419104266)**

**RITHIK M R (211419104223)**

**VIJAYVIKRAMAN V (211419104304)**

## ABSTRACT

To ensure information security, the information proprietor necessities to check the respectability of information put away somewhat in the cloud server with the public examining method. In reality the staff needs to enlist here and it will tell to the group chief who needs to support to their worker. In the future who can login in this application in light of group like A, B, C, D. The group can share the venture subtleties to a workers. In the event that the subtleties added by Group A representative's, it will tell to Group A worker's not to other people. Before that the group chief added by the administration and keep up with the group chief's subtleties. The staff could ready to see the encoded subtleties just not the first happy it will be changed over completely to cryptography design. Cryptography is procedure of getting data and correspondences through utilization of codes with the goal that main those individual for whom the data is planned can grasp it and interaction it. The solicitation of staffs move to the group chief who needs to support it then, at that point, moved to the administration who will check staff subtleties and group pioneers. After the endorsement the staff can recover the information utilizing the QR. A QR code is a sort of standardized identification that can be perused effectively by a computerized gadget and which stores data as a progression of pixels in a square-formed matrix.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVATIONS

| | |
|---|---|
| GB | Gigabyte |
| ERD | Entity Relationship Diagram |
| API | Application programming interface |
| URL | Uniform Resource Locater |
| *str* | *String* |
| func | Function |
| HTML | HyperText Markup Language |
| Css | Cascading Styling Sheets |
| FP | False Positives |
| FN | False Negatives |
| TP | True Positives |
| TN | True Negatives |

# 1 . INTRODUCTION

## 1.1 OVERVIEW

This application presents an execution for data splitting between applications considering hilter kilter encryption computations, high level mark. It in like manner gives a momentous idea of data splitting between applications. Today, dispersed capacity becomes one of the fundamental organizations, since clients can without a doubt change and proposition data with others in cloud. Nevertheless, the genuineness of shared cloud data is helpless against unpreventable gear imperfections, programming dissatisfactions or human bumbles. To ensure the uprightness of the normal data, a couple of plans have been expected to allow public verifiers (i.e., untouchable commentators) to survey data decency without recuperating the entire clients' data from cloud beneficially. Unfortunately, public looking at on the reliability of shared data could reveal data owners' fragile information to the pariah evaluator. In this paper, we propose one more insurance careful public surveying part for shared cloud data by building a homomorphic verifiable get-together imprint.) the result of secure computation prohibits 0. Inconsistent numbers revamped by each server are fixed and each server holds unpredictable numbers dark to the adversary and holds parts of sporadic numbers that make up the unpredictable numbers dark to the foe. Secret sharing is a critical means to achieve characterization and data security. Secret contribution game plans to separating a limited information with various players. The target of the secret sharing is security of secret, insurance and disguising information. This application will stay aware of the strong data move with individual social affair bunch trailblazers and delegates then organization will support for data sharing. This application will run in various administrative groups.

## Database

In computing, a database is an organized collection of data stored and accessed electronically. Small databases can be stored on a file system, while large databases are hosted on computer clusters or cloud storage. The design of databases spans formal techniques and practical considerations, including data modeling, efficient data representation and storage, query languages, security and privacy of sensitive data, and distributed computing issues, including supporting concurrent access and fault tolerance.

A database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a database system. Often the term "database" is also used loosely to refer to any of the DBMS, the database system or an application associated with the database.

Computer scientists may classify database management systems according to the database models that they support. Relational databases became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, collectively referred to as NoSQL, because they use different query languages.

## AES Algorithm

The Advanced Encryption Standard (AES), also known by its original name Rijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

The AES algorithm is a symmetrical block cipher algorithm that takes plain text in blocks of 128 bits and converts them to ciphertext using keys of 128, 192, and 256 bits. Since the AES algorithm is considered secure, it is in the worldwide standard.

The AES algorithm uses a substitution-permutation, or SP network, with multiple rounds to produce ciphertext. The number of rounds depends on the key size being used. A 128-bit key size dictates ten rounds, a 192-bit key size dictates 12 rounds, and a 256-bit key size has 14 rounds. Each of these rounds requires a round key, but since only one key is inputted into the algorithm, this key needs to be expanded to get keys for each round, including round 0.

NIST specified the new AES algorithm must be a block cipher capable of handling 128-bit blocks, using keys sized at 128, 192 and 256 bits.

- **Security.** Competing algorithms were to be judged on their ability to resist attack as compared to other submitted ciphers. Security strength was to be considered the most important factor in the competition.

- **Cost.** Intended to be released on a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.
- **Implementation.** Factors to be considered included the algorithm's flexibility, suitability for hardware or software implementation, and overall simplicity.

## Encryption

In cryptography, encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext. Ideally, only authorized parties can decipher a ciphertext back to plaintext and access the original information. Encryption does not itself prevent interference but denies the intelligible content to a would-be interceptor.

For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is possible to decrypt the message without possessing the key but, for a well-designed encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

## Decryption

The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the

encrypted information so that an authorized user can only decrypt the data because decryption requires a secret key or password.

# 2 . LITERATURE SURVEY

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

**1.TITLE:** Research of Data Sharing between Applications Base on User Request

**AUTHOR:** Xie Suping, Chen Huaichu, Luo Nianlong, Zhang Huilin

**YEAR:** 2015

**PAPER EXPLANATION**

This paper presents an implementation for data sharing between applications based on asymmetric encryption algorithms, digital signature. It also provides a new idea of data sharing between applications. It can be a supplementary program for data sharing. It can be an effective solution to solve the problem of sensitive personal data grant between applications data and can be used for proof of personal income data, statistical data sharing and the like.

**2.TITLE:** A New Privacy-Aware Public Auditing Scheme for Cloud Data Sharing with Group Users

**AUTHOR:** Anmin Fu; Shui Yu; Yuqing Zhang; Huaqun Wang; Chanying Huang

**YEAR:** 2017

**PAPER EXPLANATION**

Today, cloud storage becomes one of the critical services, because users can easily modify and share data with others in cloud. However, the integrity of shared cloud data is vulnerable to inevitable hardware faults, software failures or human errors. To ensure the integrity of the shared data, some schemes have been designed to allow public verifiers (i.e., third party auditors) to efficiently audit data integrity without retrieving the entire users' data from cloud. Unfortunately, public auditing on the integrity of shared data may reveal data owners' sensitive information to the third party auditor. In this paper, we propose a new privacy-aware public auditing mechanism for shared cloud data by constructing a homomorphic verifiable group signature. Unlike the existing solutions, our scheme requires at least t group managers to recover a trace key cooperatively, which eliminates the abuse of single-authority power and provides non-frameability. Moreover, our scheme ensures that group users can trace data changes through designated binary tree; and can recover the latest correct data block when the current data block is damaged. In addition, the formal security analysis and experimental results indicate that our scheme is provably secure and efficient.

**3.TITLE:** Secure Computation by Secret Sharing using Input Encrypted with Random Number.

**AUTHOR:** Keiic Keiichi Iwamura and Ahmad Akmal Aminuddin Mohd Kamalhi Iwamura and Ahmad Akmal Aminuddin Mohd Kamal

**YEAR:** 2019

**PAPER EXPLANATION**

Typically, unconditionally secure computation using a $(k, n)$ threshold secret sharing is considered impossible when $n < 2k - 1$. Therefore, in our previous work, we first took the approach of finding the conditions required for secure computation under the setting of $n < 2k - 1$ and showed that secure computation using a $(k, n)$ threshold secret sharing can be realized with a semi-honest adversary under the following three preconditions: (1) the result of secure computation does not include 0; (2) random numbers reconstructed by each server are fixed; and (3) each server holds random numbers unknown to the adversary and holds shares of random numbers that make up the random numbers unknown to the adversary. In this paper, we show that by leaving condition (3), secure computation with information-theoretic security against a semi-honest adversary is possible with $k \leq n < 2k - 1$. In addition, we clarify the advantage of using secret information that has been encrypted with a random number as input to secure computation. One of the advantages is the acceleration of the computation time. Namely, we divide the computation process into a preprocessing phase and an online phase and shift the cost of communication to the preprocessing phase. Thus, for computations such as inner product operations, we realize a faster online phase, compared with conventional methods.

**4.TITLE:** Secure Secret Sharing Using Homomorphic Encryption

**AUTHOR:** Nileshkumar Kakade; Utpalkumar Patel

**YEAR:** 2020

**PAPER EXPLANATION**

Secret sharing is an important means to achieve confidentiality and data privacy. Secret sharing deals with splitting a secret information with various players. The goal of the secret sharing is security of secret, privacy and hiding information. There are numerous techniques available for secret sharing e.g. polynomial, Chinese remainder theorem, vector space, matrix projection. Techniques have characteristics like threshold, proactive, verifiable. Proactive secret sharing scheme allow user to change share in case of doubt of theft. In this work we propose the proactive secret sharing scheme based on homomorphic techniques. Our scheme consists of three phases of share construction, share renewal, share reconstruction. Central authority splits an encrypted secret with each parties using homomorphic property of paillier encryption i.e. subtraction. In renewal process two or more parties relate share with each other for to generated renewed share. In reconstruction process all parties share will be add to central authority then encrypted secret will be generated. Central authority will decrypt encrypted secret using secret key then original secret will be generated. Our schemes unique features is share can be renewed any time, Each party can choose secret of their own choice, If any two parties have same content share then also encrypted share will be different due to non-deterministic property of paillier encryption.

**5.TITLE:** Secure multi-party computation in differential private data with Data Integrity Protection

**AUTHOR:** Sundari S, Ananthi M

**YEAR:** 2015

**PAPER EXPLANATION**

Secure multiparty computation (SMC) is needed now-a-days in which data are distributed between different parties. Moreover, organizations are wished to collaborate with other parties who conduct same business, for their mutual benefits. SMC provides users to gain much information from the larger data without disclosing the data. This project combines the technique secure multi-party computation and the differential privacy for vertically partitioned data between parties. To achieve this, a multi-party protocol has been proposed for the exponential mechanism. Reliable access to data is must for most computer applications and data servers. Some factors causes unauthorized access to stored data. Two Phase Validation (2PV) provides the authentication for the users, while integrating the data in multiparty computation. Data can get corrupted due to some malfunctions. Disk errors are common today but the storage technologies are not designed to handle such kind of errors. A simple integrity violation is detected by the higher level software which causes further loss of data. The proposed system is to verify the integrity of random subsets of data against general or malicious corruptions through Distributed Data Integrity (DDI) Protection.

# 3 . SYSTEM ANALYSIS

## 3.1 Existing System

A public auditing scheme that supports sensitive data sharing for the cloud environment. In our scheme, the data owner's privacy can be protected while sharing data. The integrity checking of the remotely stored data in the cloud server can also be executed efficiently. The security analysis shows that our proposed scheme is more secure compared to that of related works, and several experiments show that our scheme achieves a desirable efficiency.

**Technique:**Bilinear pairing and cryptographic difficult problems

**Disadvantages:**It takes a long time to process function methods.

## 3.2 Proposed system

We propose another plan in light of the redactable signature. In our proposed plot, the cloud server can change the mark straightforwardly without the extra sanitizer while sharing delicate information.

**Technique:** RSA algorithm. SQL Operation

**Advantages:** It gives a standard and valid solution to process the data with has to function.

## 3.3 Functional requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

### Non-Functional Requirements

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

# 3.4 Hardware Environment

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

| | | |
|---|---|---|
| PROCESSOR | : | PENTIUM IV 2.6 GHz, Intel Core 2 Duo. |
| RAM | : | 4GB DD RAM |
| MONITOR | : | 15” COLOR |
| HARD DISK | : | 40 GB |

## 3.5 Software Environment

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s and tracking the team’s progress throughout the development activity.

| | | |
|---|---|---|
| Front End | : | J2EE (JSP, SERVLETS) JAVASCRIPT |
| Back End | : | MY SQL 5.5 |
| Operating System | : | Windows 07 |
| IDE | : | Eclipse |

# 4 . SYSTEM DESIGN

## 4.1. ER diagram

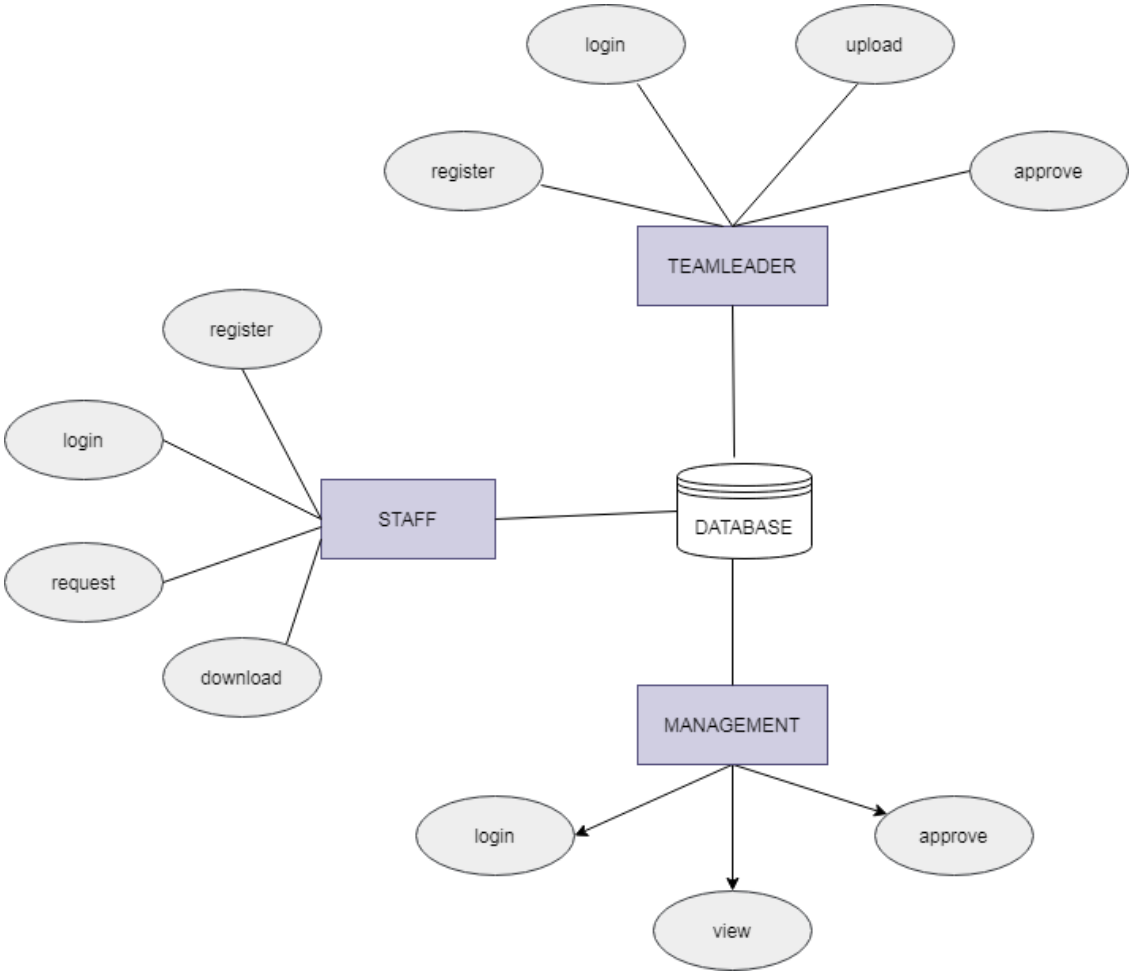


**Fig 4.1 ER Diagram**

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a

referral point, should any debugging or business process re-engineering be needed later.

## 4.2 Data flow diagram

### LEVEL 1:

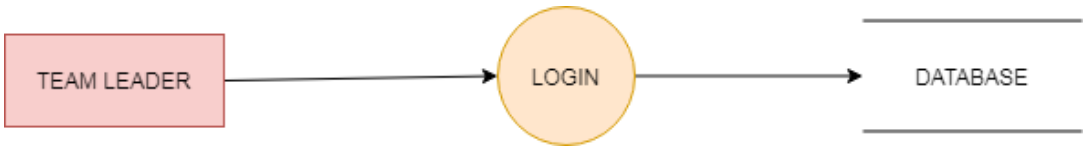


**Fig 4.2.1 Data Work Flow Diagram Level 1**

### LEVEL 2:

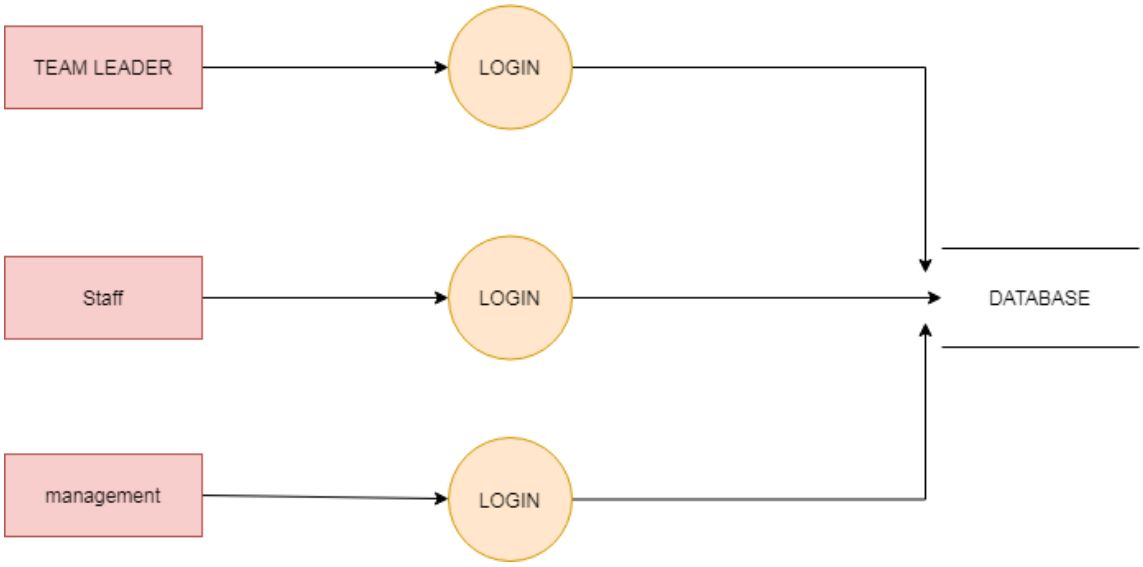


**Fig 4.2.2 Data Work Flow Diagram Level 2**

### LEVEL 3:

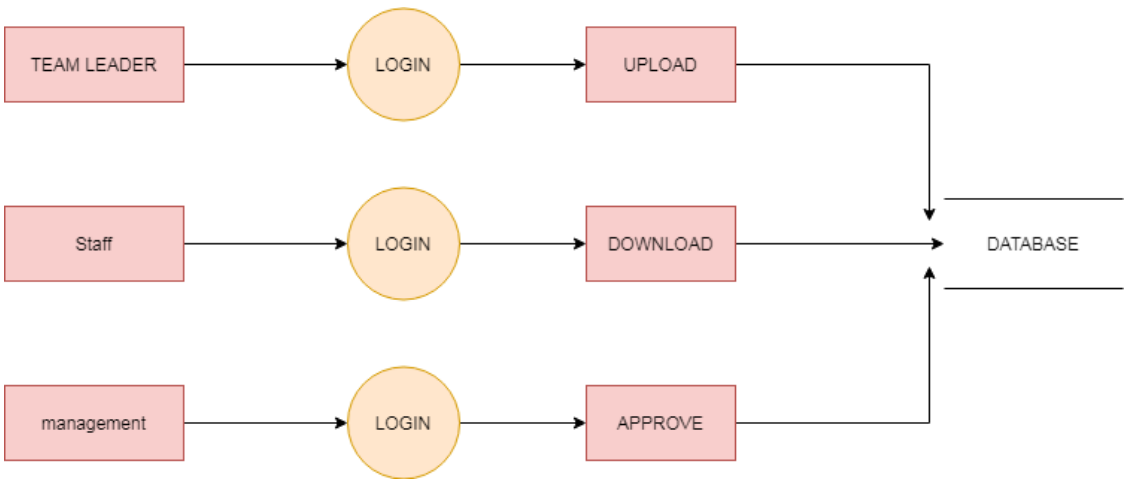

**Fig 4.2.3 Data Work Flow Diagram Level 3**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

## 4.3 USE CASE DIAGRAM

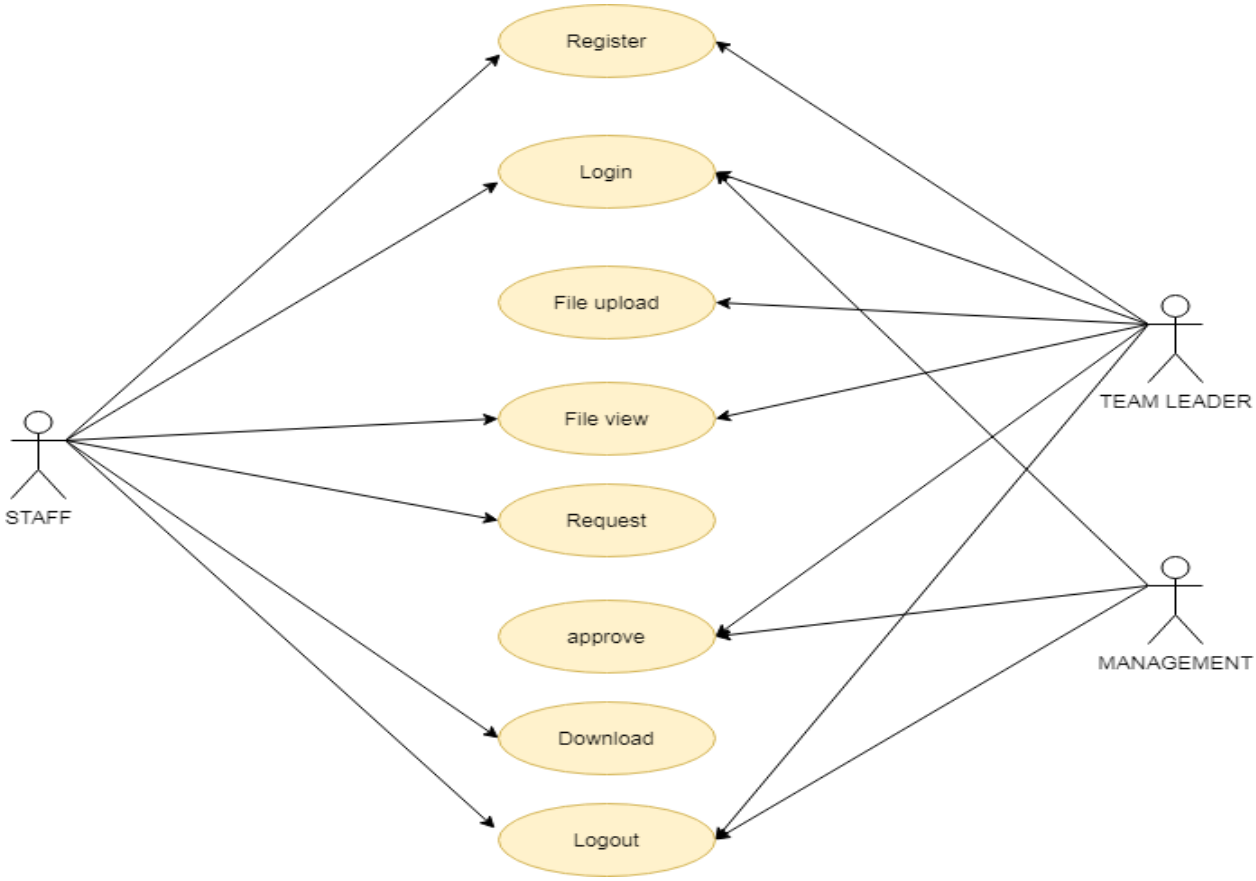


**Fig 4.3 : Use Case Diagram**

The use case diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models

into programming code. For this in our component diagram first propose a data In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

## 4.4 CLASS DIAGRAM

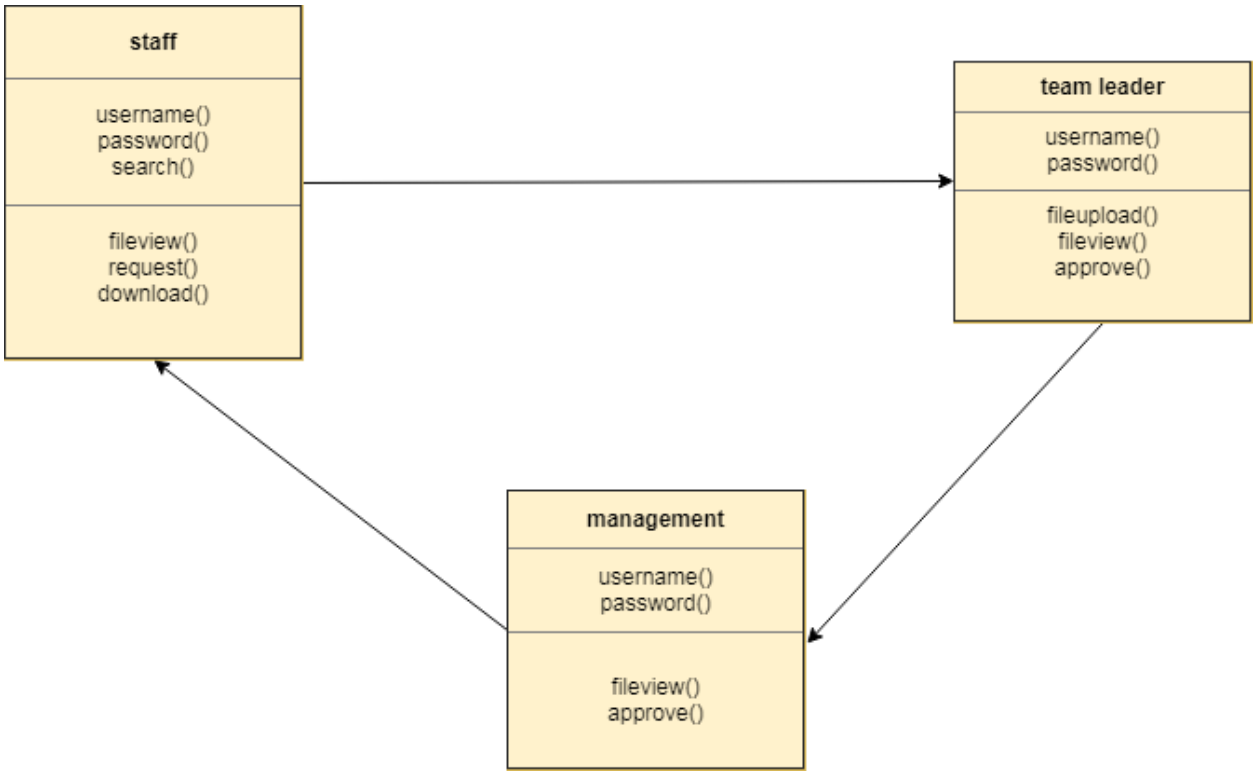


**Fig 4.4 : Class Diagram**

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The classes in a class diagram represent both the main objects and or interactions in the application and the objects.

## 4.5 ACTIVITY DIAGRAM

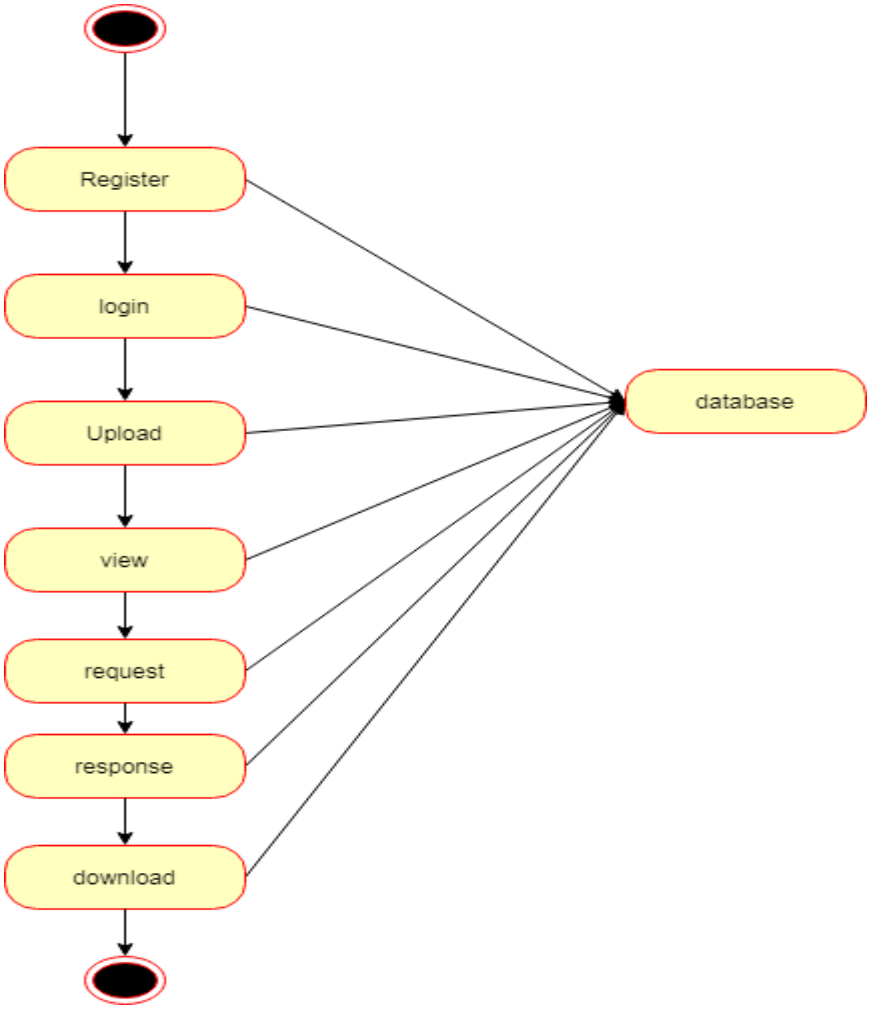


**Fig 4.5 : Activity Diagram**

Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML

activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs)from structural development.

## 4.6 SEQUENCE DIAGRAM

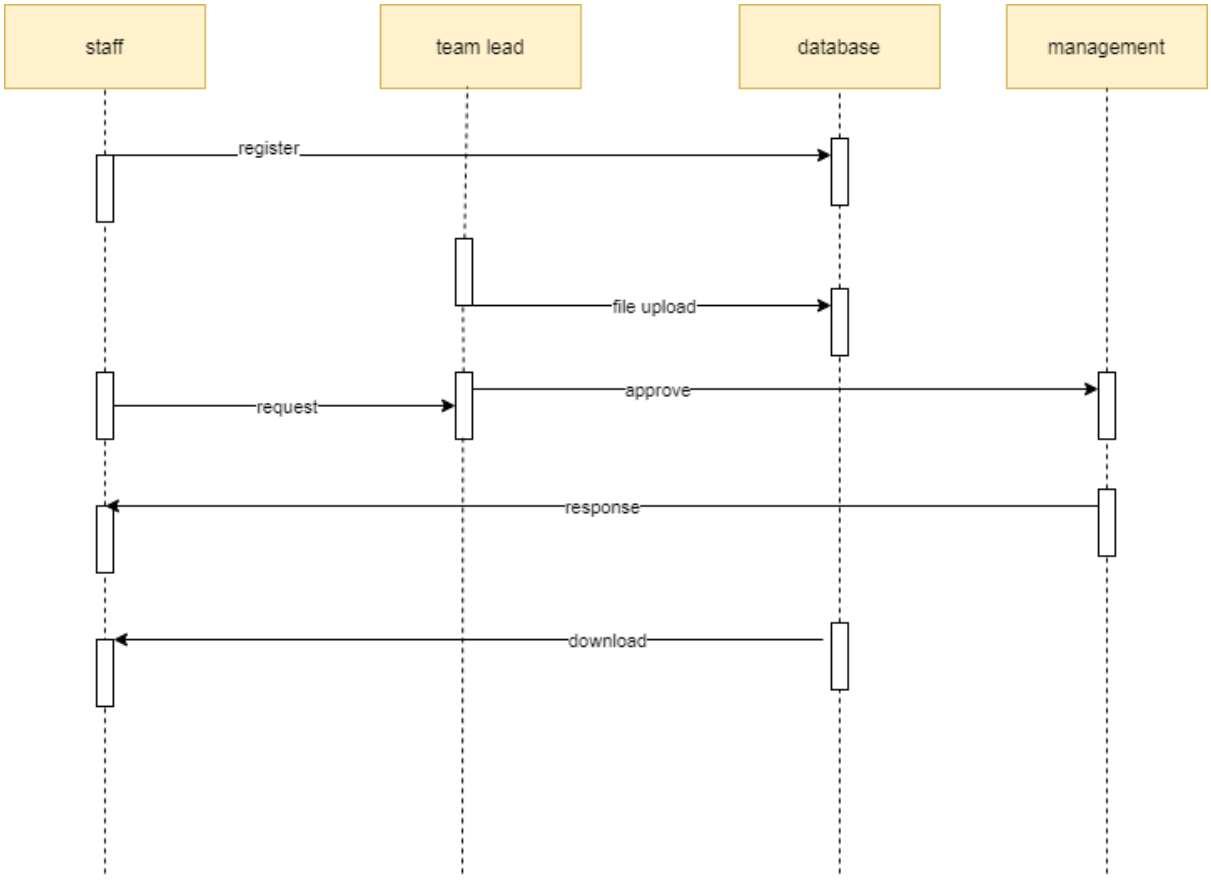


**Fig 4.6 Sequence Diagram**

In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram first propose for this in our component diagram first propose a data in this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

## 4.7 COLLABORATION DIAGRAM

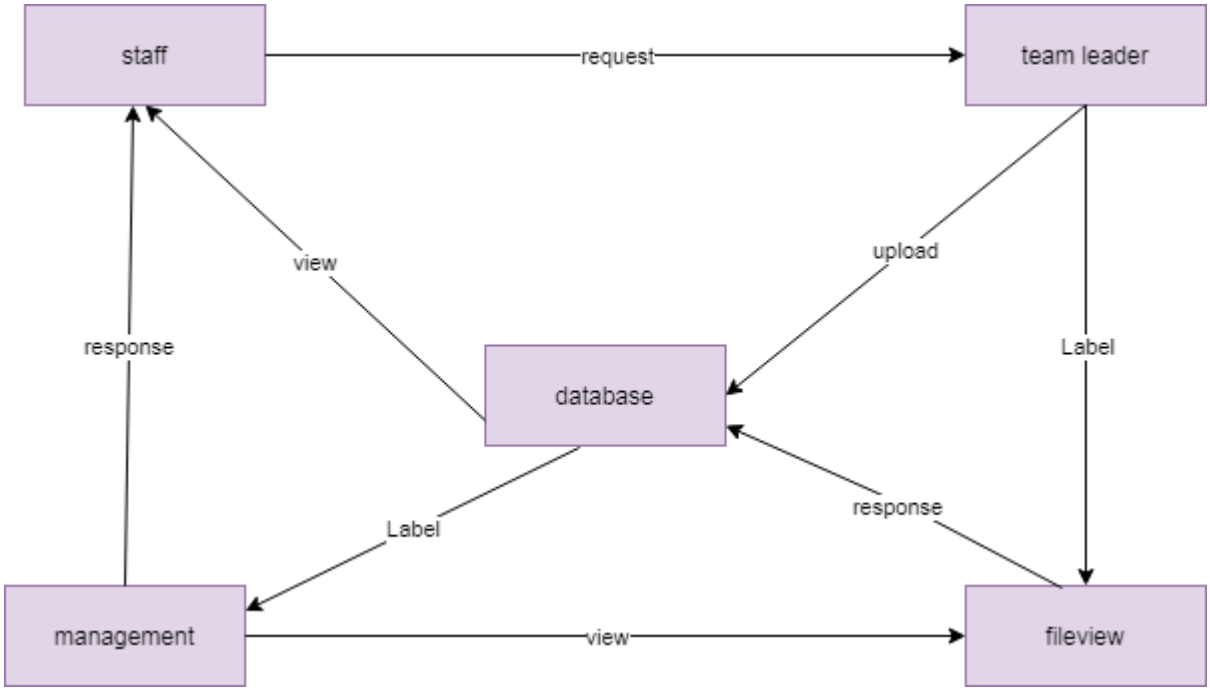


**Fig 4.7 Collaboration Diagram**

## 4.8 STATE DIAGRAM:

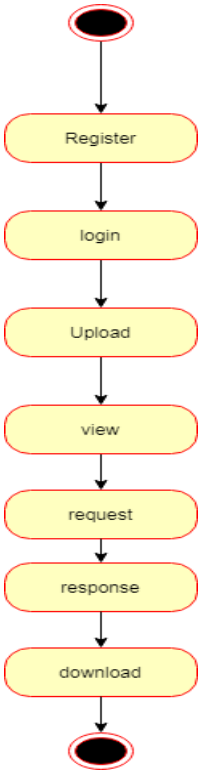


**Fig 4.8 State Diagram**

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. In our state diagram first propose a . For this in our component diagram first propose a data In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

# 5 . SYSTEM ARCHITECTURE

## 5.1 MODULE DESCRIPTION

### STAFF REGISTER:

The register module provides a conceptual framework for entering data on those staff in a way that: eases data entry & accuracy by matching the staff entry to the data source (usually paper files created at point of care), ties easily back to individual staff records to connect registers to staff data, and collects data elements to enable better supervision of donation programs.


Fig 5.1 Module Diagram I

### STAFF LOGIN:

In this module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.

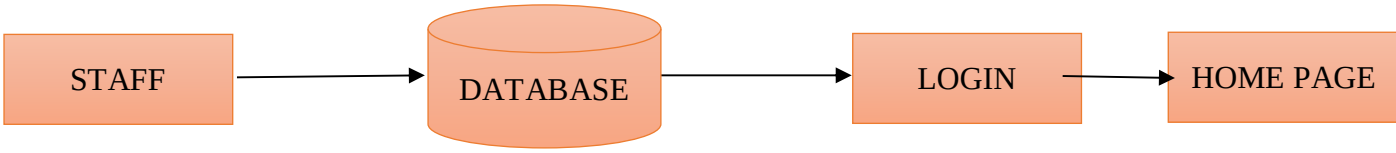


Fig 5.2 Module Diagram II

**STAFF FILE VIEW:**

In this module the staff will also view the team leader added file. And analysis the details will be responsible for your file stored in database.

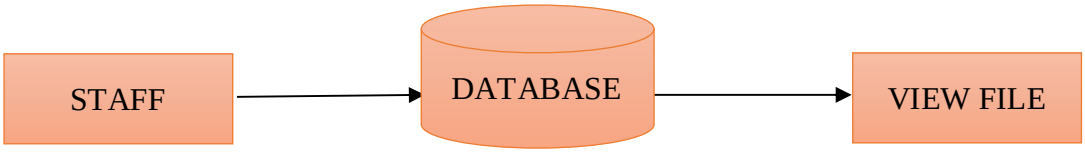


Fig 5.3 Module Diagram III

**STAFF FILE REQUEST:**

In this module is used to help to the staff to Request for download file with the land longitude and the user will update the report along with their opinion and the will be stored the database

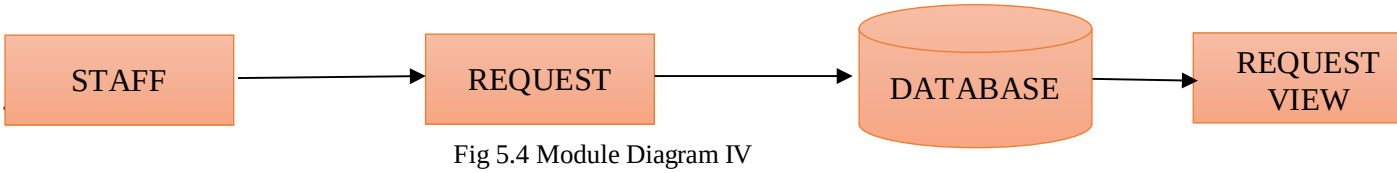


Fig 5.4 Module Diagram IV

**STAFF FILE DOWNLOAD:**

In this module the staff download the file after management accept the request. It will be stored on local storage.

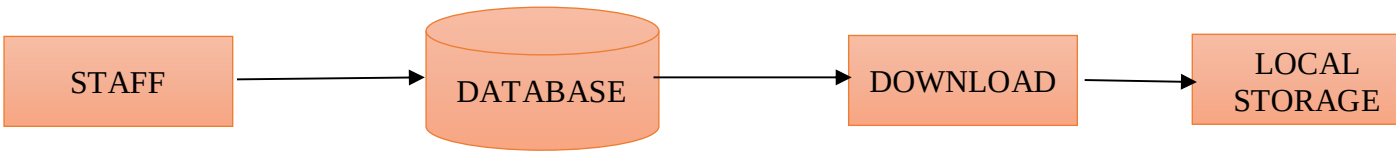


Fig 5.5 Module Diagram V

## TEAM LEADER LOGIN:

In this module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.

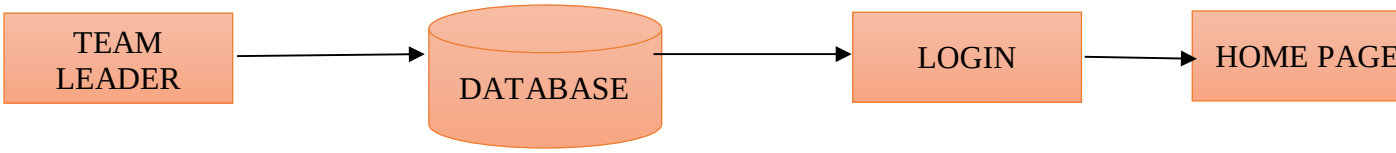


Fig 5.6 Module Diagram VI

## TEAM LEADER FILE UPLOAD:

The team leader can then select a file from their computer and click the Upload button to submit the file to the server. The Java file upload Servlet will then capture that file and persist. It will be stored in database.


Fig 5.7 Module Diagram VII

## TEAM LEADER FILE VIEW:

This module to help us the staff add the file to the staffs. The data directly stored in database. Then staff will view the uploaded file.

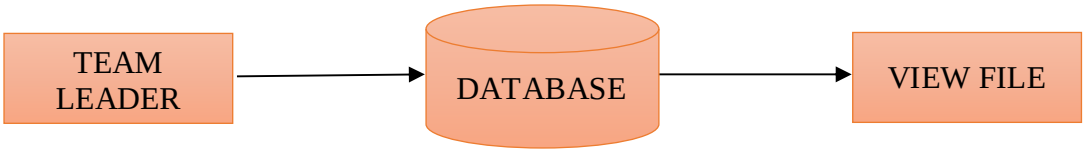


Fig 5.8 Module Diagram VIII

## MANAGEMENT LOGIN:

In this module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.

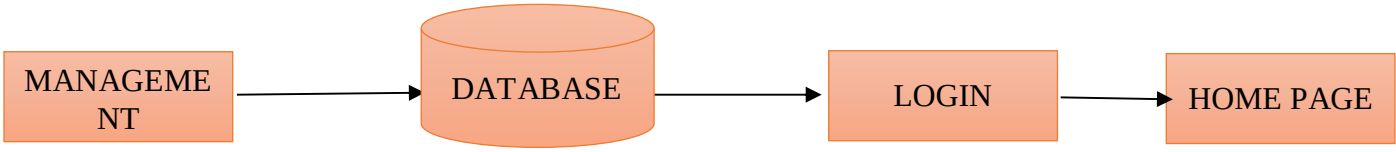


Fig 5.9 Module Diagram IX

## MANAGEMENT TEAM LEADER REGISTRATION:

The register module provides a conceptual framework for entering data on those team leader in a way that: eases data entry & accuracy by matching the team

leader entry to the data source (usually paper files created at point of care), ties easily back to individual team leader records to connect registers to team leader data, and collects data elements to enable better supervision of team programs.

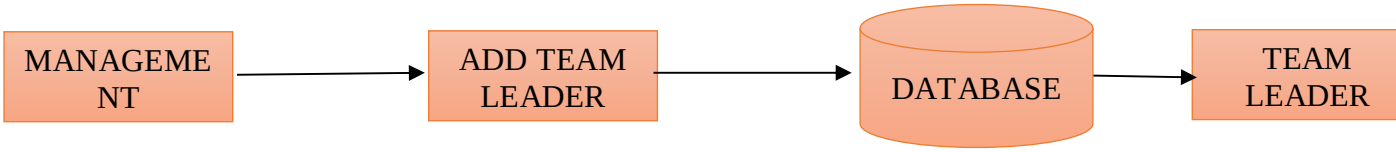


Fig 5.10 Module Diagram X

**MANAGEMENT GENERATE KEY:**

In this module the management generate key for the staff request. Because the key for the security purpose. After get the key from management the staff will download the file with key.


Fig 5.11 Module Diagram XI

**MANAGEMENT RESPONSE:**

In this module the bank will response the data file fully analyzed data in category wise view Bank will be responsible for your file stored in database.

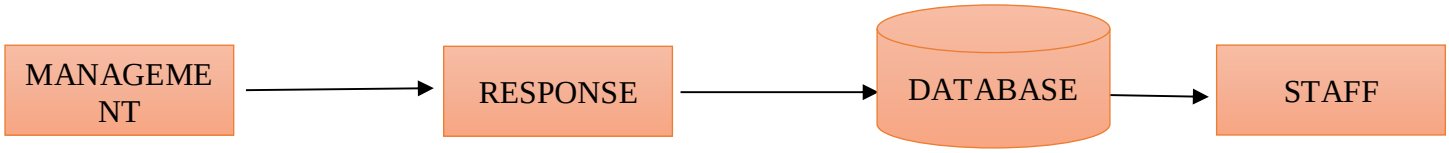


Fig 5.12 Module Diagram XII

**THE JAVA System:**

Java is a programming language initially created by James Gosling at Sun Microsystems and delivered in 1995 as a center part of Sun Microsystems' Java stage. The language determines quite a bit of its linguistic structure from C and C++ yet has a more straightforward item model and less low-level offices. Java applications are ordinarily incorporated to byte code that can run on any Java Virtual Machine (JVM) paying little mind to PC engineering. Java is broadly useful, simultaneous, class-based, and object-arranged, and is explicitly intended to have as scarcely any execution conditions as could be expected. Letting application designers "compose once, run anywhere is planned".

Java is viewed as by a lot of people as one of the most powerful programming dialects of the twentieth 100 years, and is generally utilized from application programming to web applications The java system is another stage free that works on application improvement web. Java innovation's flexibility, productivity, stage movability, and security make it the ideal innovation for network registering. From PCs to datacenters, game control center to logical supercomputers, PDAs to the Web, Java is all over the place!

**Goals OF JAVA:**

To see spots of Java in real life in our day to day existence, investigate java.com.

WHY Programming Designers Pick JAVA:

Java has been tried, refined, expanded, and demonstrated by a committed local area. Also, numbering more than 6.5 million engineers, it's the biggest and

most dynamic on earth. With its adaptability, productivity, and movability, Java has become priceless to engineers by empowering them to:

- Compose programming on one stage and run it on essentially some other stage
- Make projects to run inside an Internet browser and Web administrations
- Foster server-side applications for online discussions, stores, surveys, HTML structures handling, and then some
- Join applications or administrations utilizing the Java language to make exceptionally tweaked applications or administrations
- Compose strong and proficient applications for cell phones, distant processors, minimal expense buyer items, and essentially some other gadget with a computerized heartbeat

A few Different ways Programming Designers Learn Java:

Today, numerous schools and colleges offer courses in programming for the Java stage. What's more, designers can likewise improve their Java programming abilities by perusing Sun's java.sun.com Site, buying into Java innovation centered bulletins, utilizing the Java Instructional exercise and the New to Java Programming Center, and pursuing Web, virtual, or teacher drove courses.

**Object Situated:**

To be an Item Arranged language, any language should follow essentially the four attributes.

1. Legacy: It is the most common way of making the new classes and utilizing the way of behaving of the current classes by extending them just to reuse the current code and adding option a highlights on a case by case basis.

2. Exemplification: It is the component of consolidating the data and giving the reflection.

3. Polymorphism: As the name propose one name various structure, Polymorphism is the approach to giving the different usefulness by the capabilities having a similar name in light of the marks of the techniques.

4. Dynamic restricting: Now and again we don't have the information on objects about their particular kinds while composing our code. It is the approach to giving the greatest usefulness to a program about the particular sort at runtime

**Java Server Pages - An Outline:**

Java Server Pages or JSP for short is Sun's answer for creating dynamic sites. JSP give superb server side prearranging support for making information base driven web applications. JSP empower the engineers to straightforwardly embed java code into jsp record, this makes the advancement cycle extremely basic and its upkeep additionally turns out to be exceptionally simple.

JSP pages are effective, it loads into the web servers memory on getting the solicitation absolute first time and the resulting calls are served inside an extremely brief timeframe.

In the present climate most sites servers dynamic pages in view of client demand. Data set is extremely advantageous method for putting away the information of clients and different things. JDBC give astounding data set availability in heterogeneous data set climate. Utilizing JSP and JDBC its exceptionally cc simple to foster information base driven web application.

Java is known for its quality of "compose once, run anyplace." JSP pages are stage Java Server Pages

Java Waiter Pages (JSP) innovation is the Java stage innovation for conveying dynamic substance to web clients in a compact, secure and distinct way. The Java Server Pages particular stretches out the Java Servlet Programming interface to furnish web application engineers with a vigorous structure for making dynamic web content on the server utilizing HTML, and XML layouts, and Java code, which is secure, quick, and free of server stages.

JSP has been based on top of the Servlet Programming interface and uses Servlet semantics. JSP has turned into the favored solicitation controller and reaction instrument. Despite the fact that JSP innovation will be a strong replacement to essential Servlets, they have a transformative relationship and can be utilized in a helpful and corresponding way.

Servlets are strong and in some cases they are a piece bulky with regards to producing complex HTML. Most servlets contain a little code that handles application rationale and much more code that handles yield designing. This can make it challenging to isolate and reuse bits of the code when an alternate result design is required. Thus, web application engineers turn towards JSP as their favored servlet climate.

**Advancement of Web Applications:**

Throughout the course of recent years, web server applications have advanced from static to dynamic applications. This development became important because of certain lacks in prior web composition. For instance, to put a greater amount of business processes on the web, whether in business-to-buyer

(B2C) or business-to-business (B2B) markets, customary web composition advancements are sufficiently not. The main pressing concerns, each engineer faces while creating web applications, are:

1. Versatility - an effective website will have more clients and as the quantity of clients is expanding fastly, the web applications need to correspondingly scale.

2. Incorporation of information and business rationale - the web is simply one more method for leading business, thus it ought to have the option to utilize similar center level and information access code.

3. Sensibility - sites simply continue to get greater and we really want a suitable component to deal with the steadily expanding content and its cooperation with business frameworks.

4. Personalization - adding an individual touch to the site page turns into a fundamental component to keep our client returning once more. Knowing their inclinations, permitting them to design the data they view, recalling their previous exchanges or successive pursuit watchwords are extremely significant in giving criticism and cooperation based on what is generally a genuinely uneven discussion.

Aside from these general requirements for a business-situated site, the need for new innovations to make powerful, dynamic and minimized server-side web applications has been understood. The principal attributes of the present powerful web server applications are as per the following:

1. Serve HTML and XML, and stream information to the web client

2. Separate show, rationale and information

3. Point of interaction to data sets, other Java applications, CORBA, index and mail administrations

4. Utilize application server middleware to give conditional help.

## Track client meetings

**Advantages of JSP:**

One of the fundamental justifications for why the Java Server Pages innovation has advanced into what it is today and it is as yet developing is the staggering specialized need to improve on application configuration by isolating powerful happy from static layout show information. One more advantage of using JSP is that it permits to additional neatly discrete the jobs of web application/HTML fashioner from a product designer. The JSP innovation is honored with various energizing advantages, which are chronicled as follows:

1. The JSP innovation is stage free, in its dynamic pages, its web servers, and its fundamental server parts. That is, JSP pages perform impeccably with practically no problem on any stage, run on any web server, and web-empowered application server. The JSP pages can be gotten to from any web server.

2. The JSP innovation underscores the utilization of reusable parts. These parts can be joined or controlled towards growing more intentional parts and page plan. This certainly diminishes improvement time separated from the At advancement time, JSPs are totally different from Servlets, in any case, they are precompiled into Servlets at run time and executed by a JSP motor which is introduced on an Internet empowered application server like BEA Web Rationale and IBM Web Circle.

**Servlets:**

Prior in client-server registering, every application had its own client program and it filled in as a UI and should be introduced on every client's PC. Most web applications use HTML/XHTML that are for the most part upheld by all the temple

## 5.2 Algorithms

The **RSA algorithm** is an asymmetric cryptography algorithm; this means that it uses a *public* key and a *private* key (i.e two different, mathematically linked keys). As their names suggest, a public key is shared publicly, while a private key is secret and must not be shared with anyone.

The RSA algorithm is named after those who invented it in 1978: Ron Rivest, Adi Shamir, and Leonard Adleman.

The following illustration highlights how asymmetric cryptography works:

How it works

The RSA algorithm ensures that the keys, in the above illustration, are as secure as possible. The following steps highlight how it works:

1. Generating the keys

1. Select two large prime numbers, x*x* and y*y*. The prime numbers need to be large so that they will be difficult for someone to figure out.
2. Calculate n = x*n*=*x* x y*y*.
3. Calculate the ***totient*** function; \phi(n) = (x-1)(y-1)$\phi(n)$=($x$−1)($y$−1).

4. Select an integer e*e*, such that e*e* is ***co-prime*** to \phi(n)$\phi(n)$ and 1 < e < \phi(n)1<$e$<$\phi(n)$. The pair of numbers (n,e)($n$,$e$) makes up the public key.

**Note:** Two integers are co-prime if the only positive integer that divides them is 1.

5. Calculate d*d* such that e.d = 1*e*.*d*=1 mod*mod* \phi(n)$\phi(n)$.

d*d* can be found using the ***extended euclidean algorithm***. The pair (n,d)($n$,$d$) makes up the private key.

2. Encryption

Given a plaintext P*P*, represented as a number, the ciphertext C*C* is calculated as:

C = P^{e}*C*=*Pe* mod*mod* n*n*.

3. Decryption

Using the private key (n,d)($n$,$d$), the plaintext can be found using:

P = C^{d}*P*=*Cd* mod*mod* n*n*.

Pseudocode

Consider an example of the RSA algorithm through the following pseudocode:

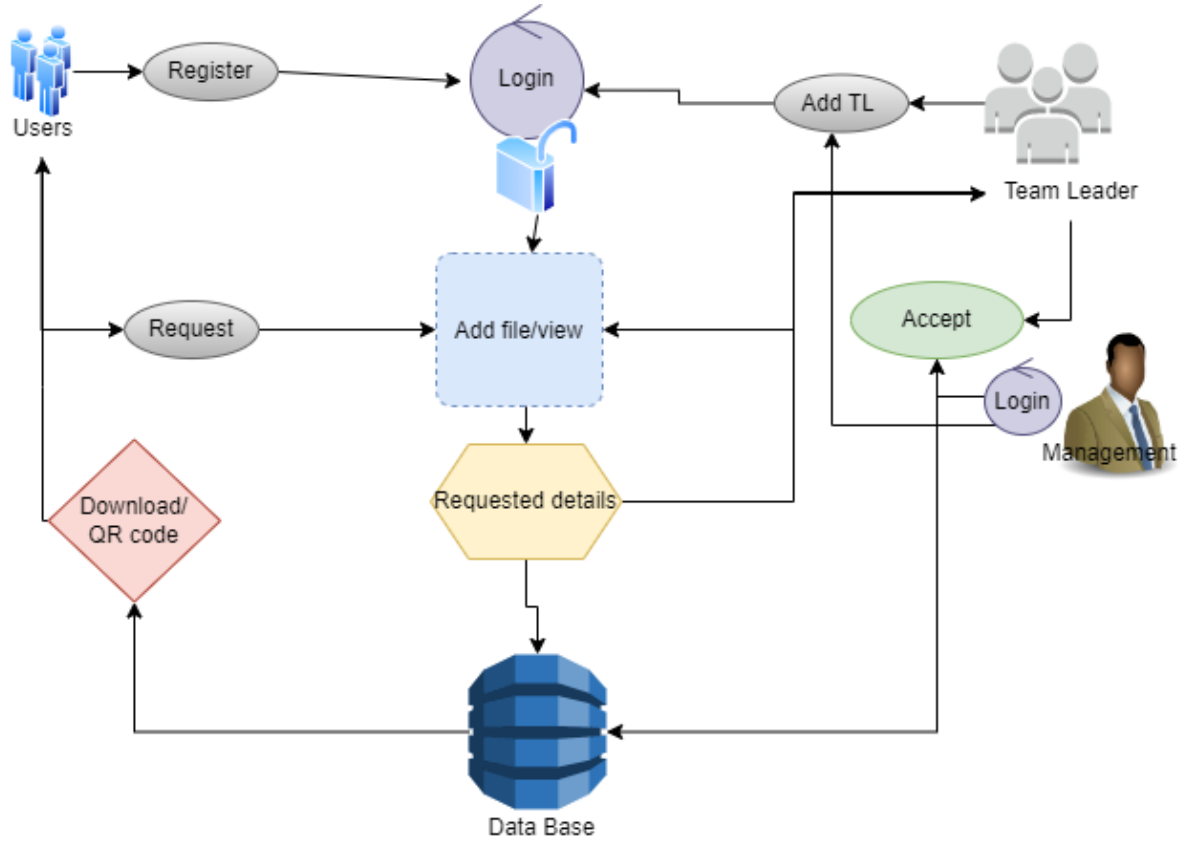


Fig 5.13 System Architecture Diagram

The systems architect establishes the basic structure of the system, we propose a Hash code Solomon algorithm and 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

# 6 . SYSTEM IMPLEMENTATION

## SYSTEM IMPLEMENTATION

This chapter describes the implementation of searched based application. It deals with the source code for main viewpoint for Anonymous Database Management.

**Homepage.jsp**

```
<%@ 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>HOME PAGE</title>

</head>

 <style>ul {

  list-style-type: none;

  margin: 0;
```

```
  padding: 10px;
  overflow: hidden;
  background-color:LightGray;
}
li {
  float: RIGHT;
}
li a {
  display: block;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  }
li a:hover:not(.active) {
  background-color: #6dd5ed;
.active {
  background-color: #4CAF50;
```

```
}
body {
  background-image:url("image/cc5-01.jpg") ;
}
h2{
  text-shadow: 2px 2px 5px green;
  font-style:bold;
  font-family:Sans-serif;
  color:white;
   font-size: 50px;
}
/* img{
padding-right:20%;
} */
 </style>
<body>
<ul>
  <li><b><a href="ceologin.jsp">CEO</a></b></li>
```

```
 <li><b><a href="reginalmanlogin.jsp">Regional Manager</a></b></li>

 <li><b><a href="bmanagerlogin.jsp">Branch manager</a></b></li>

 <li><b><a href="homepage.jsp">Home</a></b></li>

 </ul>

<center><h2>Business</h2></center>

<!-- <center><img alt="welcomeimage" src="image/2.jpg" width=350
height=350>

</center> -->

</body>

</html>
```

**Bmanagerview.jsp**

```
<%@ 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>Login Page</title>
```

```
<style>
.myDiv {
  border: 5px outset #77c732;
  background-color: lightblue;
  border-radius: 10px;
  width:500px;
  height:260px;
  margin: auto;
  padding-top:30px;
 /*  box-shadow: 25px 20px 20px #888888; */

}
.myDiv2 {
  font-size:25px;
 font-family:Times New Roman;
font-weight: bold;
color: white;
```

```
}
.myDiv3 {
  font-size:25px;

font-weight: bold;
 font-family:Times New Roman;
color: #bd0d91;
}

body
{

background: url(image/CC1.jpg)no-repeat 0px 0px;
          background-size: 100% 100%;
    min-height: 610px;
         position:relative;

}
```

```
</style>

</head>

<body>

<br><br><br>

<center>

<div class="myDiv2">

Branch Manager Login

</div>

</center>

<br><br>

<div class="myDiv">
```

```
<center>

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

  <input type="text" name="username" placeholder="Username" required
style="width:280px;height:40px;border-radius: 10px;text-
align:center;"><br><br>

  <input type="text" name="password" placeholder="Password" required
style="width:280px;height:40px;border-radius: 10px;text-
align:center;"><br><br>

  <input type="submit" value="Submit"
style="width:100px;height:40px;border-radius: 10px;"><br><br>




   <div class="myDiv3">

 New User <a href="bmanagerregister.jsp"> <br> Signup</a>

</div>



  </form>

  </center>

</div>

</body>
```

```
</html>
```

**Bmanagerregister.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

    <%-- <%@page import="dbconn.Dbconn"%> --%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement" %>
<%@page import="java.sql.*" %>
<%@page import="java.util.*" %>

<!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>
```

```
<style>
.myDiv {
  border: 1px outset #77c732;
  background-image: linear-gradient(to right top, #d16ba5, #c777b9, #ba83ca, #aa8fd8, #9a9ae1, #8aa7ec, #79b3f4, #69bff8, #52cffe, #41dfff, #46eefa, #5ffbf1);
  border-radius: 20px;
  width:600px;
  height:450px;
  margin: auto;
  padding-top:30px;
  box-shadow: 25px 20px 20px #888888;

}
.myDiv2 {
  font-size:25px;
  font-style: bold;
  font-family:TIMES NEW ROMEN;
font-weight: bold;
```

```
color: #77c732;
}

</style>

</head>
<body>

<br><br><br>
<center>
<div class="myDiv2">
BRANCH REGISTER
</div>
```

```
</center>

<br><br>


<div class="myDiv">

<center>

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

  <input type="text" name="username" placeholder="Username"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"
required><br><br>

  <input type="text" name="phonenumber" placeholder="Phonenumber"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"
required><br><br>

  <input type="text" name="email" placeholder="Emaiid"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"
required><br><br>

  <select name="company" id="company"
style="width:280px;height:40px;border-radius: 10px;text-align:center;">

  <option value="Company A">COMPANY A</option>

  <option value="Company A">COMPANY B</option>

  <option value="Company A">COMPANY C</option>

  </select><br><br>
```

```
  <input type="text" name="password" placeholder="Password"
id="password1" style="width:280px;height:40px;border-radius: 10px;text-
align:center;" required><br><br>

   <input type="text" name="confpass" placeholder="Re-Enter Password"
id="password2" style="width:280px;height:40px;border-radius: 10px;text-
align:center;" required><br><br>

  <input type="submit" value="REGISTER"
style="width:100px;height:40px;border-radius: 10px;"><br><br>



  </form>

  </center>

</div>

 <script>

                        window.onload = function () {

        document.getElementById("password1").onchange = validatePassword;

        document.getElementById("password2").onchange = validatePassword;

                        }

                        function validatePassword() {
```

```
                                    var pass2 =
document.getElementById("password2").value;

                                    var pass1 =
document.getElementById("password1").value;

                                    if (pass1 != pass2)


        document.getElementById("password2").setCustomValidity("Password
Doesn't Match");

                                    else


        document.getElementById("password2").setCustomValidity('');

                                    //empty string means no validation error

                                }

                            </script>




</body>

</html>
```

**Bmanagermainpage.jsp**

```
<%@ 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>Branch Manager</title>


<style>


ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color:black;
}
```

```
li {
  float:left;
  padding-right:100px;
}

li a {
  display: block;
  color: white;
  text-align: center;
  text-style:bold;
  padding: 14px 16px;
  text-decoration: none;
}

.myDiv2 {
  font-size:25px;
  font-style: italic;
font-weight: bold;
```

```
color:grey;
}

body {
    background-image: url("image/2.jpg");
     background-size: cover;

  /*background: linear-gradient(to bottom, #996633 0%, #ff66cc 100%);
   background-repeat: no-repeat;
   background-size: cover;  */
  /* background-image: linear-gradient(to right, red , yellow);*/
}
.dropbtn {
  background-color: black;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
```

```
}

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
```

```
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}



</style>



</head>
<body>

<ul>

      <li><a href="homepage.jsp">Home</a></li>
       <li><a href="bmrequest.jsp">Request</a></li>
       <li><a href="bmanagerdownload.jsp">View</a></li>
       <li><a href="report.jsp">Report</a></li>
```

```
    <!--      <li><a href="report22.jsp">Report2</a></li> -->
        <li><a href="homepage.jsp">Logout</a></li>
</ul>
<br><br>
<center>
<div class="myDiv2">


</div></center>


</body>
</html>
```

**Regionalmainpage.jsp**

```
<%@ 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>Regional Manager</title>

<style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color:black;
}

li {
  float:left;
  padding-right:100px;
}
```

```
li a {
  display: block;
  color: white;
  text-align: center;
  text-style:bold;
  padding: 14px 16px;
  text-decoration: none;
}

.myDiv2 {
  font-size:25px;
  font-style: italic;
font-weight: bold;
color:grey;
}

body {
```

```
    background-image: url("image/6.jpg");
  /*background: linear-gradient(to bottom, #996633 0%, #ff66cc 100%);
   background-repeat: no-repeat;
   background-size: cover;  */
  /* background-image: linear-gradient(to right, red , yellow);*/
}




</style>



</head>
<body>


<ul>


        <li><a href="homepage.jsp">Home</a></li>
```

```
        <li><a href="Generalreportregionalview.jsp">General Report
view</a></li>

        <li><a href="branchview.jsp">Tech Report view</a></li>

        <li><a href="regionaladdfile.jsp">Add</a></li>

        <li><a href="regmanfileview.jsp">View</a></li>

        <li><a href="homepage.jsp">Logout</a></li>

</ul>

<br><br>

<center>

<div class="myDiv2">



</div></center>



</body>

</html>
```

**Ceomainpage.jsp**

```
<%@ 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>CEO</title>


<style>


ul {

  list-style-type: none;

  margin: 0;

  padding: 0;

  overflow: hidden;

  background-color:#00bfff       ;

}


li {

  float: right;
```

```
  padding-right:10px;
}

li a {
  display: block;
  color: White;
  text-align: center;
  text-style:bold;
  padding: 14px 16px;

  text-decoration: none;
}

body {
  background-image: url("image/bg-6.jpg");
    background-size: 100% 100%;
    min-height: 610px;
          position:relative;
```

```
}


</style>




</head>

<body>


<ul>

<li><a href="homepage.jsp">Logout</a></li>

<li><a href="ceoresponse.jsp">Response</a></li>

<li><a href="ceoreportview.jsp">Tech Report view</a></li>

<li><a href="generalreportceoview.jsp">General Report view</a></li>

<li><a href="homepage.jsp">Home</a></li>

</ul>


</body>

</html>
```

**Regionallogin.jsp:**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

    <%@page import="dbconn.Dbconn"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement" %>
<%@page import="java.sql.*" %>
<%@page import="java.util.*" %>

<!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>View Page</title>

<style>
```

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #0000ff;
}

li {
  float: right;
  padding-right:95px;
}

li a {
  display: block;
  color: white;
  text-align: center;
  text-style:bold;
```

```
  padding: 14px 16px;
  text-decoration: none;
}

.backtag{
float:right;
margin-right:50px;
font-size:30px;
}

.myDiv2 {
  font-size:25px;
  font-style: italic;
font-weight: bold;
color:blue;
}

table,td,th {
```

```
  border: 2px solid black;


}
table {
  border-collapse: collapse;
  width: 90%;
}

td {
  text-align: center;
 height: 40px;
}
th{
 height: 30px;
 color: blue;
}
```

```
</style>


</head>

<body>


<!-- <ul>


<li><a href="ownermainpage.jsp">Logout</a></li>


</ul>

 -->

<div class="backtag">

<a href="reginalmanmainpage.jsp">Back</a>

</div>


<br>

<center>

<div class="myDiv2">
```

RManager  Viewlist

<br><br><br>

<table>

<!-- <th>Pharmacyname</th> -->

<th>Filename</th>

<!-- <th>Filekey</th> -->

<th>Encryption</th>

<th>Manageremail</th>

<%-- <%

String idlist="";

```
String fnamelist="";

String fkeylist="";

String encryptlist="";

String maillist="";




%> --%>

<%String email=session.getAttribute("emailkey").toString(); %>


<%

        Connection con = Dbconn.create();

        PreparedStatement p = con.prepareStatement("SELECT * FROM
`businesstwodb`.`upload` where manageremail="'+email+'"");

       ResultSet rp = p.executeQuery();


            while (rp.next()){
```

```
String          idlist=rp.getString(1);
String fnamelist=rp.getString(2);
String fkeylist=rp.getString(3);
String encryptlist=rp.getString(4);
String maillist=rp.getString(5);



%>


 <tr>
<td><%=rp.getString(2)  %></td>
<%--  <td><%=rp.getString(4)  %></td> --%>
<td><%=rp.getString(6) %> </td>
<td><%=rp.getString(9) %> </td>



 </tr>
```

```
        <%

        }

        %>


</table>


</center>


</body></html>
```

# 7.CONCLUSION

## 7.1 SCREENSHOT:


Fig 7.1 INDEX

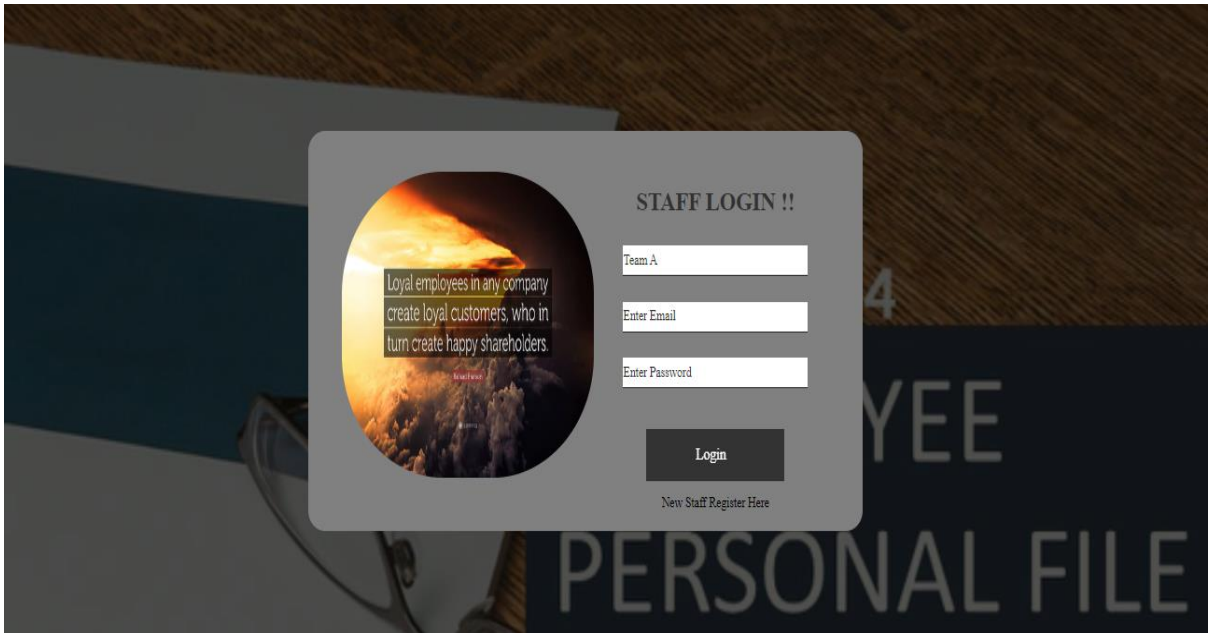


Fig 7.2 STAFFLOG

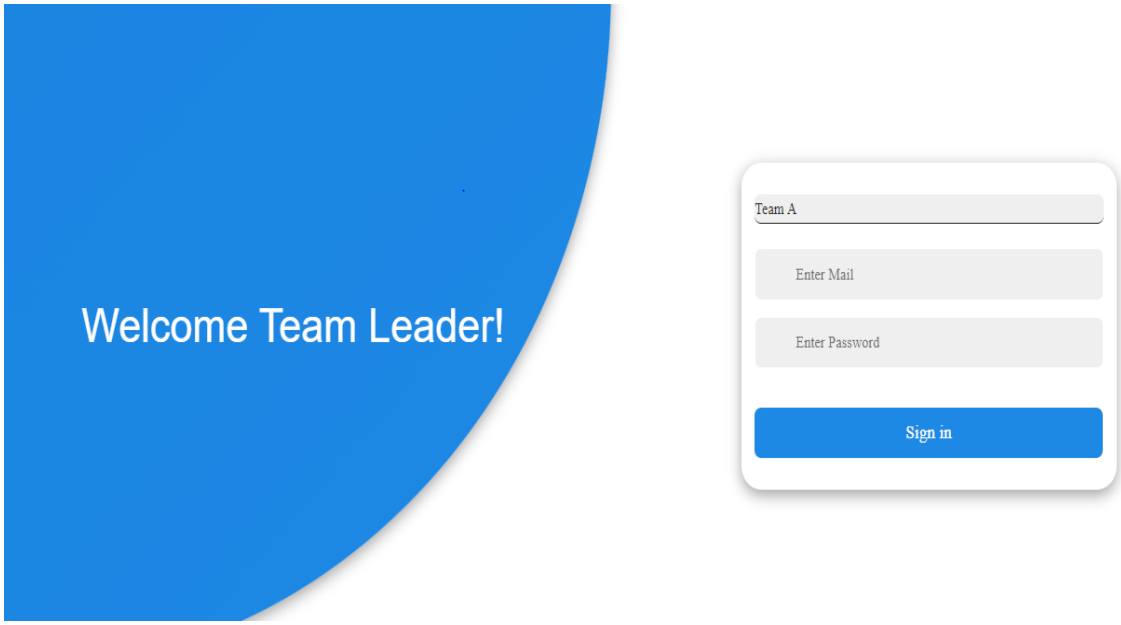


Fig 7.3 TEAMLEADER LOG

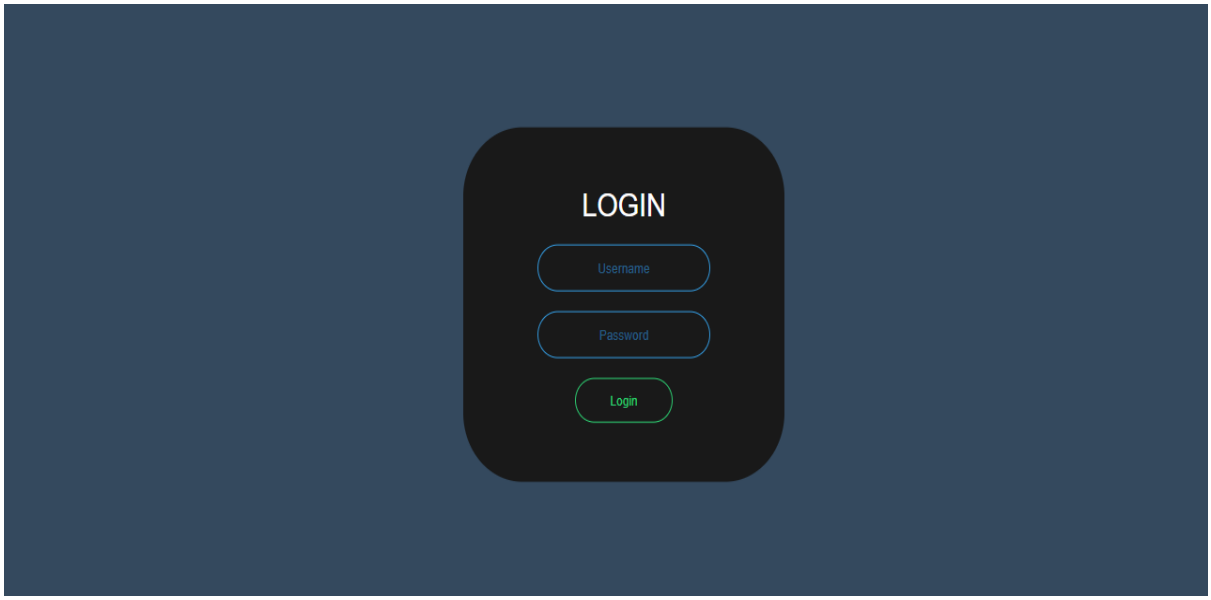


Fig 7.4 MANAGEMENT LOG

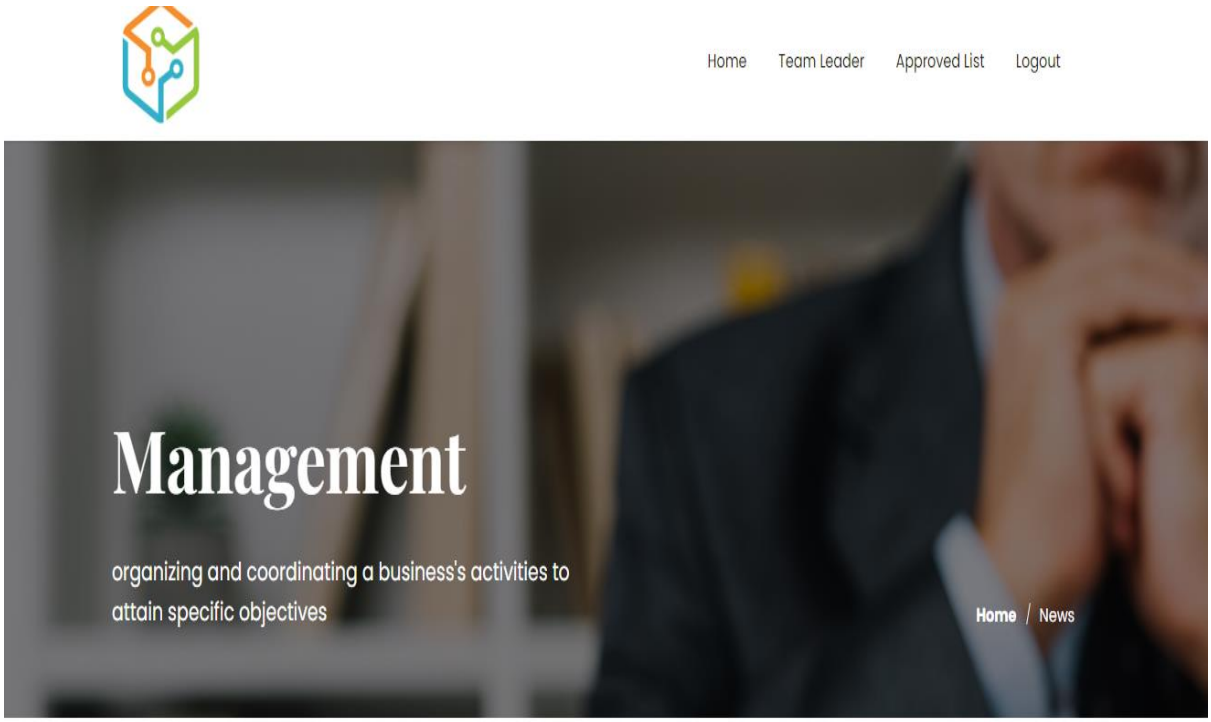


Fig 7.5 MANAGEMENTHOME

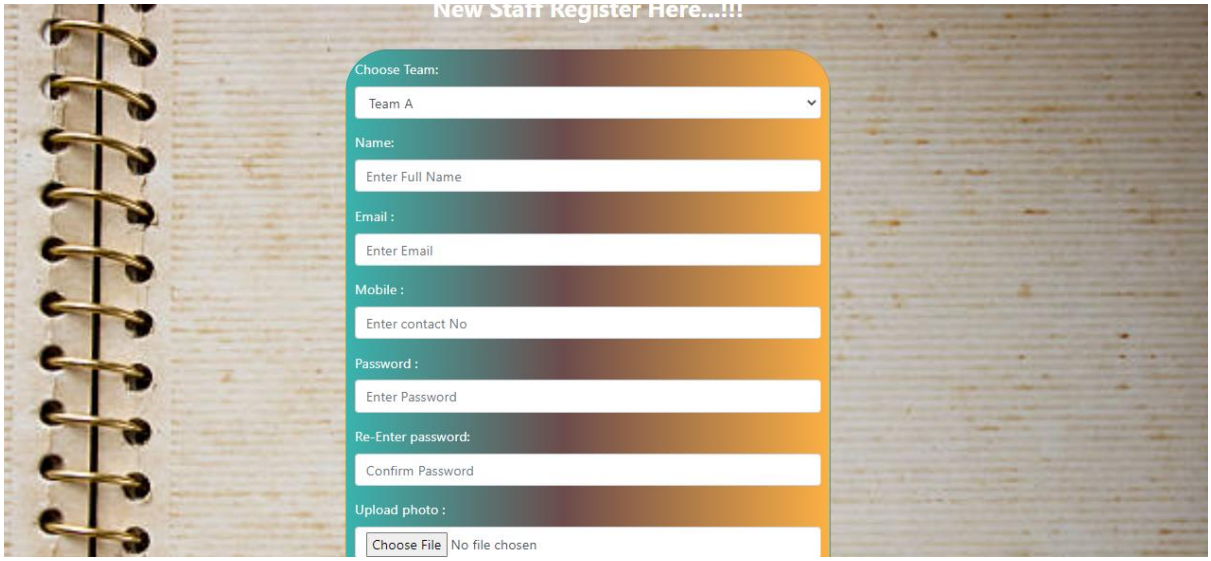


Fig 7.6 STAFFREG

## 7.2 FUTIRE ENHANCEMENT

1. Implementing a real-world anonymous database system.
2. Improving the efficiency of protocols, in terms of number of messages exchanged and in terms of their sizes, as well.
3. Implement using two are more algorithms

## 7.3CONCLUSION

Data sensitivity concerns information that should be protected from unauthorized access or disclosure due to its sensitive nature. For some, that might be Team leader, Staff details records. Sensitive data is confidential information that must be kept safe and out of reach from all outsiders unless they have permission to access it. Access to sensitive data should be limited through sufficient data security and information security practices designed to prevent.

# REFERENCES

[1] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147-156, 2020.

[2] D. Liu and J. Lee, "CNN based Malicious Website Detection by Invalidating Multiple Web Spams," *IEEE Access*, vol. 8, no. 1, pp. 97258-97266, 2020.

[3] W. Martin, V. Friedhelm, and K. Axel, "Tracing manufacturing processes using blockchain-based token compositions," *Digital Communications and Networks*, vol. 6, no 2, pp. 167-176, 2019.

[4] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos and G. Das, "Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6-14, 2018.

[5] L. Peng, W. Feng, and Z. Yan. (2020). Privacy preservation in permissionless blockchain: A survey. *Digital Communications and Networks*. [Online]. Available: https://doi.org/10.1016/j.dcan.2020.05.008.

[6] N. Kakade and U. Patel, "Secure Secret Sharing Using Homomorphic Encryption," in *Proc. 2020 11th International Conference on Computing, Communication and Networking Technologies*, 2020, pp. 1-7.

[7] S. Sundari and M. Ananthi, "Secure multi-party computation in differential private data with Data Integrity Protection," in *Proc. 2015 International Conference on Computing and Communications Technologies*, 2015, pp. 180-184.

[8] S. Jiao, T. Lei, Y. Gao, Z. Xie and X. Yuan, "Known-Plaintext Attack and Ciphertext-Only Attack for Encrypted Single-Pixel Imaging," *IEEE Access*, vol. 7, no.2, pp. 119557-119565, 2019.

[9] S. Kaushik, and S. Puri, "Online transaction processing using enhanced sensitive data transfer security model," in *Proc. 2012 Students Conference on Engineering and Systems*, 2012, pp. 1-4.

[10] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li and R. Chen, "NutBaaS: A Blockchain-as-a-Service Platform," *IEEE Access*, vol. 7, pp. 134422-134433, 2019.

[11] F. Casino and C. Patsakis, “An Efficient Blockchain-Based Privacy-Preserving Collaborative Filtering Architecture,” *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1501-1513, Nov. 2020.

[12] D. Chkliaev, J. Hooman and P. van der Stok, “Mechanical verification of transaction processing systems,” in *Proc. ICFEM 2000. Third IEEE International Conference on Formal Engineering Methods*, 2000, pp. 89-97.

[13] S. Zhang, and J H. Lee. “Mitigations on Sybil-based Double-spend Attacks in Bitcoin,” *IEEE Consumer Electronics Magazine*, vol.7, no. 2, pp. 1-1, 2020.

[14] X. Wang, Q. Feng and J. Chai, “The Research of Consortium Block chain Dynamic Consensus Based on Data Transaction Evaluation,” in *Proc. 2018 11th International Symposium on Computational Intelligence and Design*, 2018, pp. 214-217.

[15] S. Zhang, and J. H. Lee, “A group signature and authentication scheme for blockchain-based mobile-edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 5, 4557-4565, 2019.