



SENSITIVE DATA TRANSACTIONS USING BLOCKCHAIN METHODS

A PROJECT REPORT

Submitted by

ARIKRISHNA.A(311518104007)

MUNNA MANISH BALAJI(311518104025)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE,

KODAMBAKKAM, CHENNAI-24

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**SENSITIVE DATA TRANSACTIONS USING BLOCKCHAIN METHODS**” is the bonafide work of “**ARIKRISHNAN.A (311517104007), MUNNA MANISH BALAJI (311517104025)**” who carried out the project work under my supervision.

SIGNATURE

Dr.B.MonicaJenefer,M.E,Ph.D

HEAD OF THE DEPARTMENT

Computer Science and Engineering
Meenakshi Sundararajan Engineering
College,
No.363, Arcot Road, Kodambakkam,
Chennai -600 024.

SIGNATURE

Mrs.MR.Nithya,M.E

SUPERVISOR

ASSISTANT PROFESSOR

Computer Science and Engineering
Meenakshi Sundararajan Engineering
College,
No.363, Arcot Road, Kodambakkam,
Chennai -600 024.

Submitted for the project viva voce of Bachelor of Engineering in Computer Science and Engineering held on_____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

A block chain-based processing framework for sensitive data is proposed. The underlying block chain module provides technical support, such as virtual machines, consensus algorithms, transaction verification mechanisms, and accounting mechanisms. The E-contract layer module provides a distributed application service and uses the block chain technology to support it. The proposed smart system is used by each party involved in the production of sensitive data. The final sensitive data are produced by the final data generator, and other modules involved in the process of data production are unaware of the final data. This approach prevents the leakage of sensitive data into the circulation. Sensitive data transaction is confidential information that must be kept safe and out of reach from all outsiders unless they have permission to access it. Encryption is a very generic term and there are many ways to encrypt data. Companies need to implement and manage encryption correctly.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	LIST OF FIGURES	vii
	LIST OF TABLES	viii
1	INTRODUCTION	
	1.1 ABOUT THE PROJECT	
	1.2 DOMAIN OVERVIEW	
	1.3 EXISTING SYSTEM	
	1.4 PROBLEM STATEMENT	
	1.5 CHAPTER OVERVIEW	
2	LITERATURE SURVEY	
	CNN BASED MALICIOUS WEBSITE DETECTION BY INVALIDATING 2.1 MULTIPLE WEB SPAMS	
	A SYSTEMATIC LITERATURE REVIEW OF BLOCKCHAIN CYBERSECURITY 2.2	

2.3 SECURE COMPUTATION BY SECRET
SHARING USING INPUT ENCRYPTED
WITH RANDOM NUMBER

2.4 SECURE SECRET SHARING USING
HOMOMORPHIC ENCRYPTION

2.5 SECURE MULTIPARTY COMPUTATION
IN DIFFERENTIAL PRIVATE DATA
WITH DATA INTEGRITY PROTECTION

3 SYSTEM ARCHITECTURE

3.1 PROJECT ARCHITECTURE

3.2 SYSTEM ARCHITECTURE

3.3 HARDWARE REQUIREMENTS

3.4 SOFTWARE REQUIREMENTS

4 SYSTEM MODELLING

4.1 UNIFIED MODELLING LANGUAGE

4.2 USE CASE DIAGRAM

4.3 CLASS DIAGRAM

4.4 SEQUENCE DIAGRAM

4.5 COLLABORATION DIAGRAM

- 4.6 ACTIVITY DIAGRAM
- 4.7 STATECHART DIAGRAM
- 4.8 DATA FLOW DIAGRAM
- 4.9 ER DIAGRAM

5 SYSTEM IMPLEMENTATION

- 5.1 PROPOSED SYSTEM
- 5.2 MODULES
- 5.3 MODULE DESCRIPTION
- 5.4 ALGORITHM USED

6 SOFTWARE TESTING

- 6.1 FEASIBILITY STUDY
- 6.2 SYSTEM TESTING
- 6.3 TESTING TYPES
- 6.4 TEST RESULTS

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

7.2 FUTURE ENHANCEMENT

APPENDIX SCREENSHOT REFERENCES

LIST OF FIGURES

FIGURE NO.	NAME OF THE FIGURE	PAGE NO .
3.2	SYSTEM ARCHITECTURE	
4.1	USE CASE DIAGRAM	
4.2	CLASS DIAGRAM	
4.3	SEQUENCE DIAGRAM	
4.4	COLLABORATION DIAGRAM	
4.5	ACTIVITY DIAGRAM	
4.6	STATE CHART DIAGRAM	
4.7	COMPONENT DIAGRAM	
4.8	DATA FLOW DIAGRAM	
4.9	ER DIAGRAM	

LIST OF TABLES

TABLE NO.	NAME OF THE TABLE	PAGE NO.
3.3	HARDWARE REQUIREMENTS	
3.4	SOFTWARE REQUIREMENTS	

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT:

Sensitive data transaction 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 data leaks and data breaches. Sensitive data can be any sort of information that needs to be protected from unauthorised access to safeguard the privacy or security of an individual or organisation. It can include any information pertaining to: Passwords. Encryption keys. Collecting staff information and maps their relationships for a complete picture of user account organisation. Encryption is a very generic term and there are many ways to encrypt data. Companies need to implement and manage encryption correctly. The key to a good encryption strategy is using strong encryption and proper key management. Encrypt sensitive data before it is shared over untrusted networks (ex. Encrypted file storage).

1.2 DOMAIN OVERVIEW:

A blockchain is a distributed database that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Blockchains are best known for their crucial role in cryptocurrency systems, such as Bitcoin, for maintaining a secure and decentralised record of transactions. The innovation with a blockchain is that it guarantees the fidelity and security of a record of data and generates trust without the need for a trusted third party.

One key difference between a typical database and a blockchain is how the data is structured. A blockchain collects information together in groups, known as blocks, that hold sets of information. Blocks have certain storage capacities and, when filled, are closed and linked to the previously filled block, forming a chain of data known as the blockchain. All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled.

A database usually structures its data into tables, whereas a blockchain, like its name implies, structures its data into chunks (blocks) that are strung together. This data structure inherently makes an irreversible timeline of data when implemented in a decentralised nature. When a block is filled, it is set in stone and becomes a part of this timeline. Each block in the chain is given an exact time stamp when it is added to the chain.

1.3 EXISTING SYSTEM

Concept:

Most of the existing homomorphic secret sharing and secure multi-party computing technologies have the problems of massive communication rounds and too much traffic load.

Technique:

Different Hash Function Model.

Disadvantage:

It takes a long time to process various hash function methods.

1.4 PROBLEM STATEMENT:

Some of the fundamental problems in data transactions have not been addressed effectively. But,Blockchain methods of sensitive data transaction has received wide attention due to its potential approach of activating data security while transaction. A robust and efficient blockchain algorithm is proposed for sensitive data transactions. The Proposed algorithm is used by each party involved in the production of sensitive data.

1.5 CHAPTER OVERVIEW

The project report is organised with various chapters that denote the various functionalities and aspects of the system being developed.

Chapter 1 gives a general description about the project. It represents the basic idea of the project and introduces the topics of the existing system and proposed system.

Chapter 2 deals with the related works of the project. A literature review for each related work is explained in detail.

Chapter 3 presents the system architecture and requirements. It specifies the hardware and software components that are required. It also lists the technologies used in the implementation of the project.

Chapter 4 explains the system design with the use of UML diagrams and the data flow diagrams.

Chapter 5 contributes a detailed description of different modules that are there in the design and how they are implemented.

Chapter 6 gives a detailed description of the different test cases that were performed on the system.

Chapter 7 provides the conclusion. It also elucidates how the project can be further enhanced.

CHAPTER 2

LITERATURE SURVEY

2.1 CNN BASED MALICIOUS WEBSITE DETECTION BY INVALIDATING MULTIPLE WEB SPAMS.

Although a variety of techniques to detect malicious websites have been proposed, it becomes more and more difficult for those methods to provide a satisfying result nowadays. Many malicious websites can still escape detection with various Web spam techniques. In this paper, we summarise three types of Web spam techniques used by malicious websites, such as redirection spam, hidden IFrame spam, and content hiding spam. We then present a new detection method that adopts the perspective of users and takes screenshots of malicious webpages to invalidate Web spams.

Disadvantages:

- Can improve accuracy using more algorithms.

2.2 A SYSTEMATIC LITERATURE REVIEW OF BLOCKCHAIN CYBERSECURITY

Since the publication of Satoshi Nakamoto's white paper on Bitcoin in 2008, blockchain has (slowly) become one of the most frequently discussed methods for securing data storage and transfer through decentralised, trustless, peer-to-peer systems. This research identifies peer-reviewed literature that seeks to utilise blockchain for cyber security purposes and presents a systematic analysis of the most frequently adopted blockchain security applications. Our

findings show that the Internet of Things (IoT) lends itself well to novel blockchain applications, as

do networks and machine visualisation, public key cryptography, web applications, certification schemes and the secure storage of Personally Identifiable Information (PII). This timely systematic review also sheds light on future directions of research, education and practices in the blockchain and cyber security space, such as security of blockchain in IoT, security of blockchain for AI data, and sidechain security, etc.

Disadvantages:

- It lacks accuracy.

2.3 SECURE COMPUTATION BY SECRET SHARING USING INPUT ENCRYPTED WITH RANDOM NUMBER

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 realised 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 realise a faster online phase, compared with conventional methods.

Disadvantages:

- Can improve using better techniques.

2.4 SECURE SECRET SHARING USING HOMOMORPHIC ENCRYPTION

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 schemes allow users to change shares 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 party using the homomorphic property of paillier encryption i.e. subtraction. In the renewal process two or more parties relate share with each other to generate renewed

share. In the reconstruction process all parties share will be added 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 scheme's unique feature is that shares can be renewed any time, Each party can choose a secret of their own choice, If any two parties have the same content share then also encrypted share will be different due to non-deterministic property of paillier encryption.

Disadvantages:

- Can improve accuracy using better encryption technique.

2.5 SECURE MULTIPARTY COMPUTATION IN DIFFERENTIAL PRIVATE DATA WITH DATA INTEGRITY PROTECTION

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.

Disadvantages:

- Can improve accuracy using different integrity.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 PROJECT ARCHITECTURE

System Architecture defines a comprehensive solution based on principles, concepts, and properties logically related and consistent with each other. The architecture has features, properties, and characteristics satisfying, as far as possible, the problem or opportunity expressed by a set of system requirements and life cycle concepts (e.g., operational, support) and is implementable through technologies (e.g., software, services, procedures, human activity).

The Architecture explains how the patient's details are taken and based on the data collected it can be predicted if the patient has heart disease or not.

3.2 SYSTEM ARCHITECTURE

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.

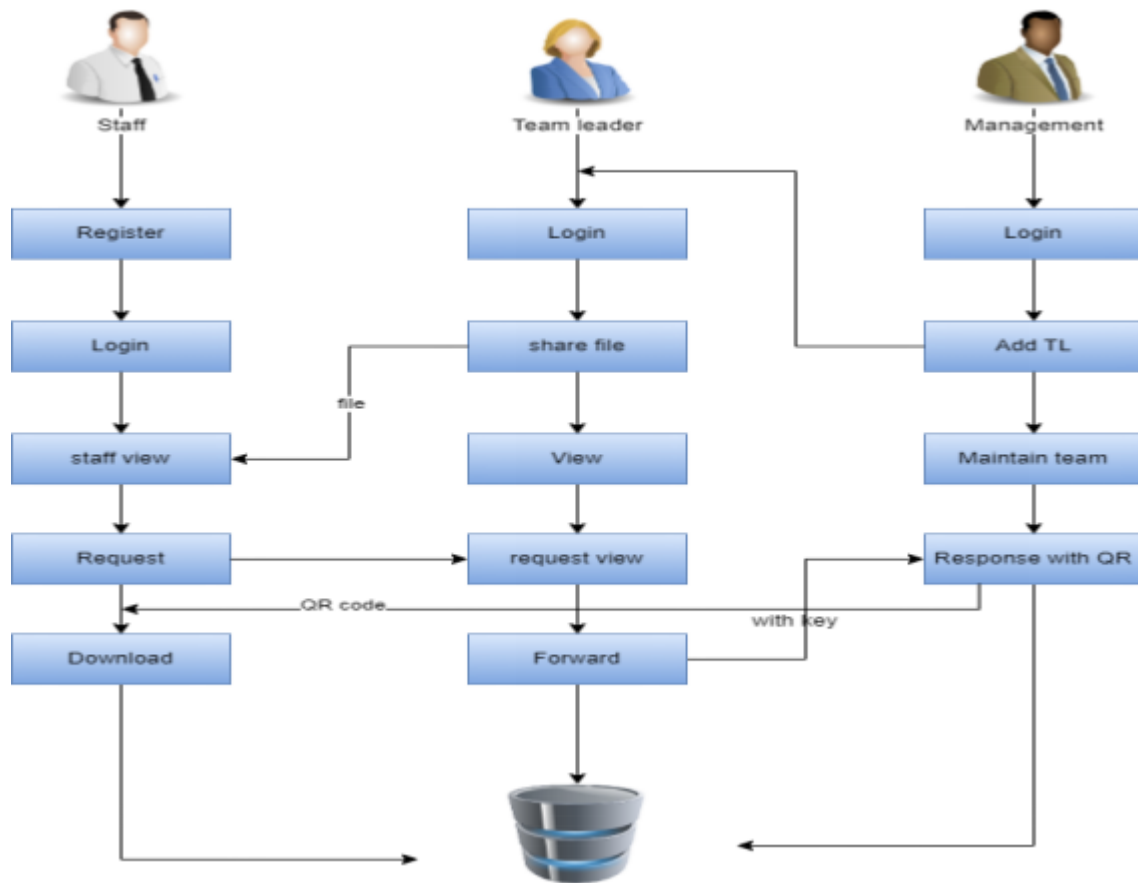


Figure 3.2 System Architecture

3.3 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. A HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

S.No	REQUIREMENTS	RECOMMENDED	MINIMUM REQUIREMENTS
1	Operating System	Windows 10	Windows 8
2	RAM	4 GB	2 GB
3	Hard Disk	40 GB	20 GB
4	Processor	Intel Quad Core PENTIUM IV	Intel Dual Core PENTIUM III
5	Monitor	15 inch	12 inch

TABLE 3.3 Hardware Requirements

3.4 SOFTWARE REQUIREMENTS

Requirements	Specification
TOOL	Eclipse, MYSQL.
CODING LANGUAGE	J2EE(JSP,SERVLETS) JAVASCRIPT, HTML,CSS.

Table 3.4 Software Requirements

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 bytecode 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”.

JSP:

Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provides excellent server side scripting support for creating database driven web applications. JSP enables the developers to directly insert java code into a 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 server's 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 serve dynamic pages based on user request. Databases are a very convenient way to store the data of users and other things.

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.

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 .

ECLIPSE FRAMEWORK:

Eclipse is a free, Java-based development platform known for its plugins that allow developers to develop and test code written in other programming languages. Eclipse is released under the terms of the Eclipse Public Licence.

The Eclipse Foundation is an independent, nonprofit corporation based in Canada that shepherds the open source Eclipse software development community and includes the legal jurisdiction of the European Union.

Eclipse is supported by over 320 members, 1,750 committers and more than 332 million lines of code. The foundation's goal is to create both a community and an ecosystem of complementary products and services.

MYSQL DATABASE:

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

CHAPTER 4

SYSTEM MODELLING

4.1 UNIFIED MODELLING LANGUAGE (UML)

Unified Modelling Language is a standardised modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualising, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

The primary goals in the design of the UML as follows:

- Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
- Provide extensibility and specialisation mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of the OO tools market.

- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

4.2 USE CASE DIAGRAM

The use case diagram is used to define the core elements and processes that make up a system. The key elements are termed as “actors” and the processes are called “use cases”. The use case diagram shows which actors interact with each use case. This definition defines what a use case diagram is primarily made up of – actors and use cases.

In software and system engineering, a use case is a list of steps, typically defining interactions between a role (known in UML as an “actor”) and a system, to achieve a goal. The actor can be a human or an external system. In system engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.

The purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Showing the interacting among the requirements are actors.

Use cases help in identifying the operations that can be performed by an actor. It gives a list of the various applications that can be utilised by the system. The actor can be a real time human or a system. It helps in identifying the various

modules present in the system. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.

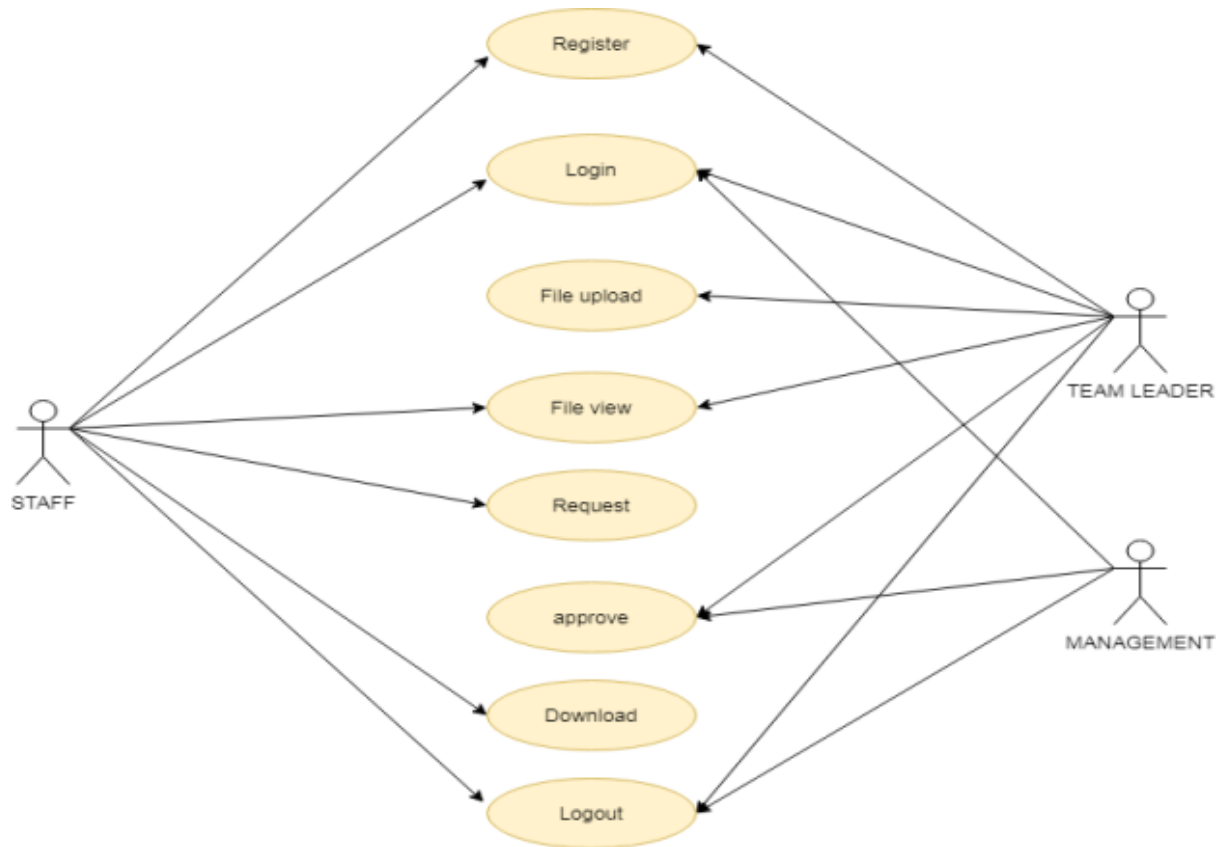


Figure 4.1 Use Case Diagram

4.3 CLASS DIAGRAM

Class diagram is a static diagram. It is the building block of every object-oriented system and helps in visualising and describing the system. A class diagram depicts the structure of the system through its classes, their attributes, operations and relationships among the objects. A class is a blueprint that defines the variables and methods common to all objects of a certain kind.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. The characteristics of Class Diagram are:

1. Each class is represented by a rectangle having a subdivision of three compartments - name, attributes and operations.
2. There are three types of modifiers which are used to decide the visibility of attributes and operations : + is used for public visibility, # is used for protected visibility, – is used for private visibility.

In the diagram, classes are represented with boxes that contain three compartments. The top compartment contains the name of the class. It is printed in bold and centred, and the first letter is capitalised. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

The main modules that are involved in this system are patient, and the database. The patient is the end user of the system who uploads the medical documents. The system stores the uploaded documents and the result is displayed to the patient.

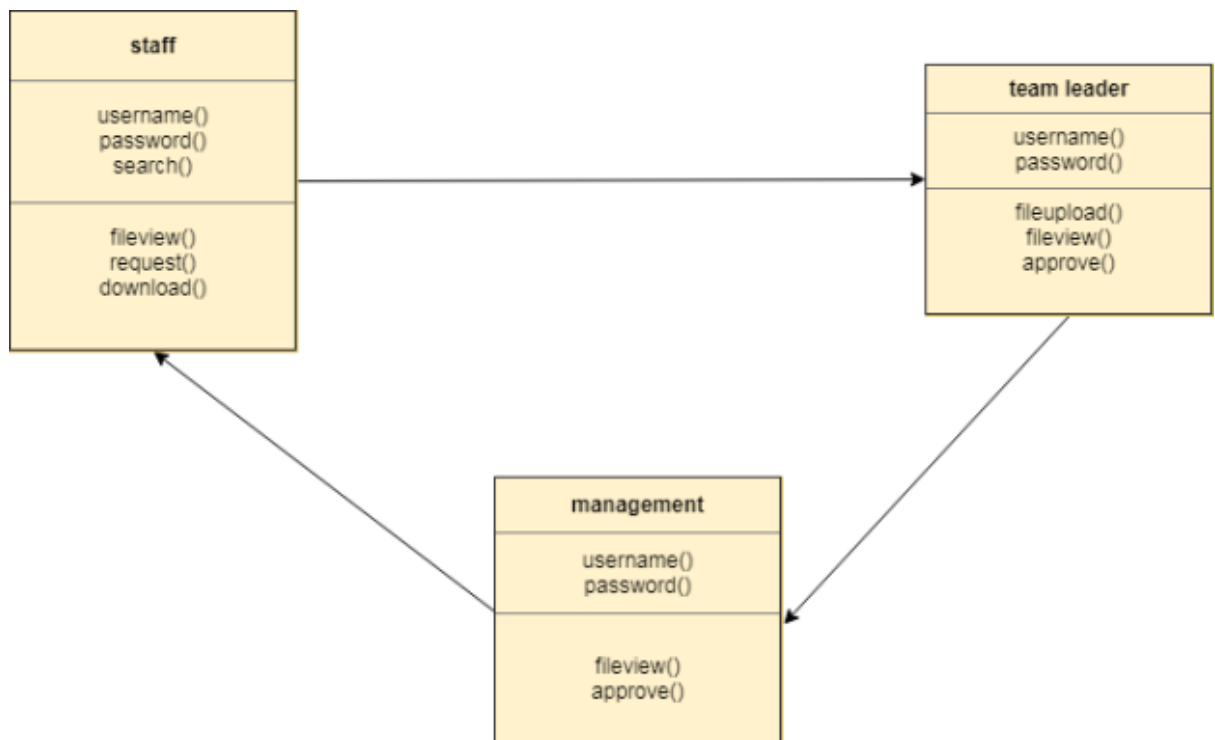


Figure 4.2 Class Diagram

4.4 SEQUENCE DIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in which order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

Sequence diagrams are a popular dynamic modelling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

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.

A sequence diagram shows different processes or objects that live simultaneously as parallel vertical lines (lifelines) and the messages exchanged between them and the order in which they occur as horizontal arrows.

The main purpose of the Sequence diagram is

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organisation of the objects.
- To describe the interaction among objects.

Sequence diagrams can be used

- To model the flow of control by time sequence.
- To model the flow of control by structural organisations.
- For forward engineering
- For reverse engineering

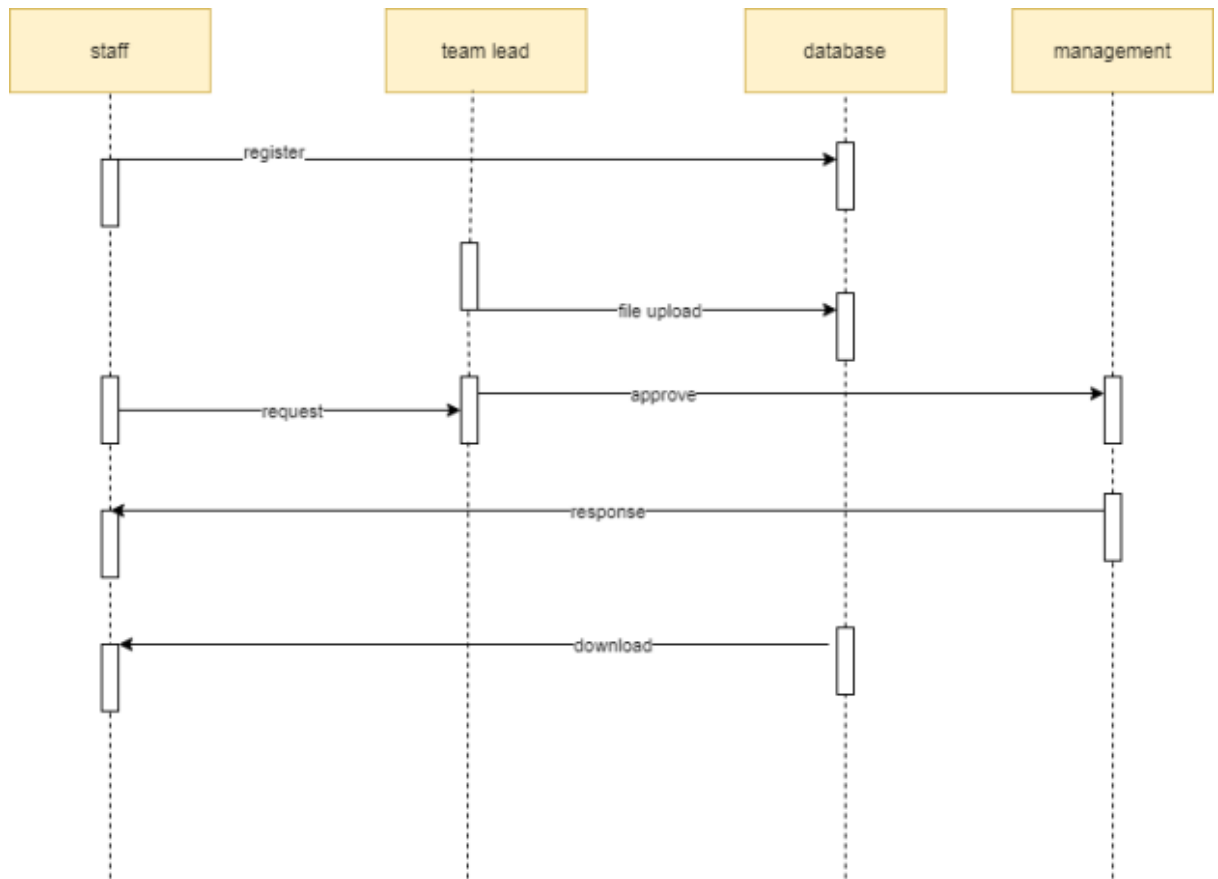


Figure 4.3 Sequence Diagram

4.5 COLLABORATION DIAGRAM

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among objects in the Unified Modelling Language (UML).

Collaboration diagrams convey the same information as sequence diagrams, but focus on object roles instead of the timings of messages. It illustrates messages being sent between classes and objects (instances).

Collaboration diagrams represent a combination of information taken from class, sequence and use case diagrams describing both the static structure and dynamic behaviour of a system. The collaboration diagram describes the messages or roles sent between objects.

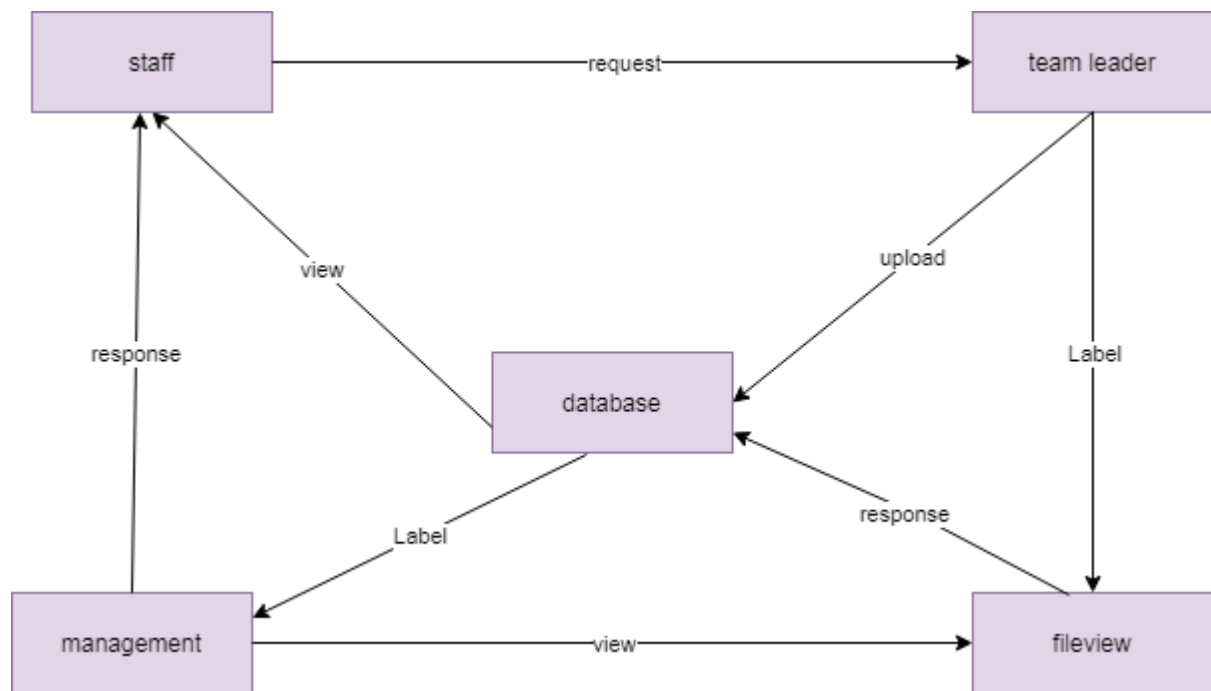


Figure 4.4 Collaboration Diagram

4.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organisational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc. Activity diagrams are constructed from a limited number of shapes, connected with arrows.

The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial node) of the workflow;
- an encircled black circle represents the end (final node).

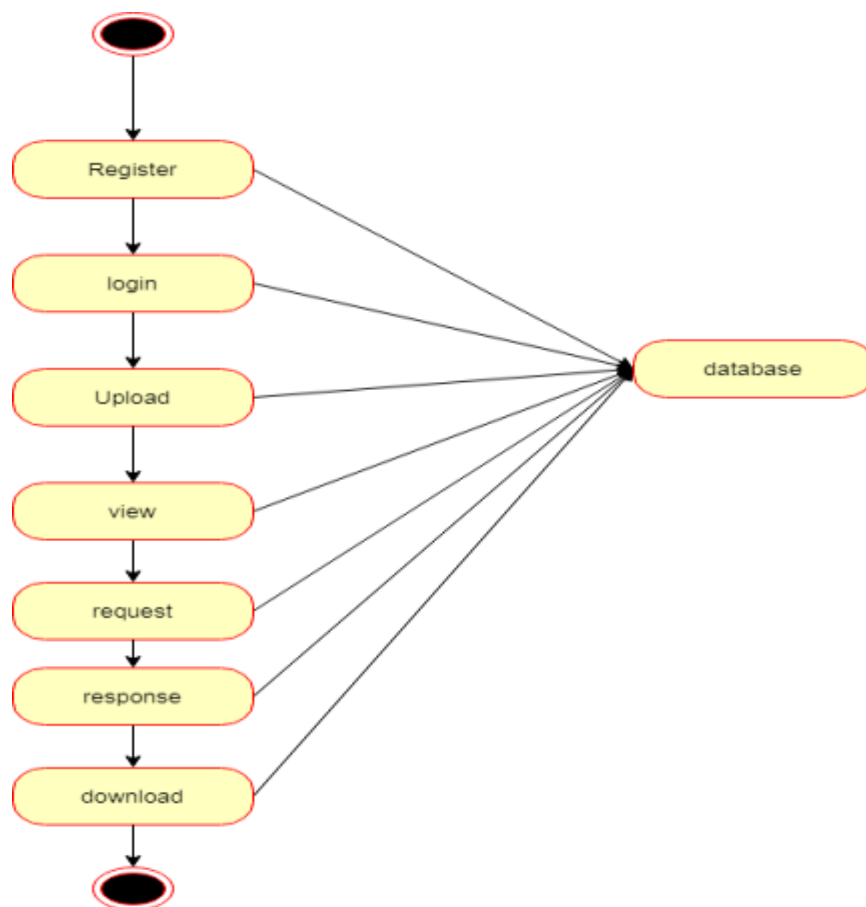


Figure 4.5 Activity Diagram

4.7 STATE CHART DIAGRAM

A State chart diagram describes a state machine. A state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. It describes different states of a component in a system. The states are specific to a component/object of a system. State chart diagrams are used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagrams is to model the lifetime of an object from creation to termination.

The main purposes of using State chart diagrams

- To model the dynamic aspect of a system.
- To model the lifetime of a reactive system.
- To describe different states of an object during its life time
- an encircled black circle represents the end (final node).

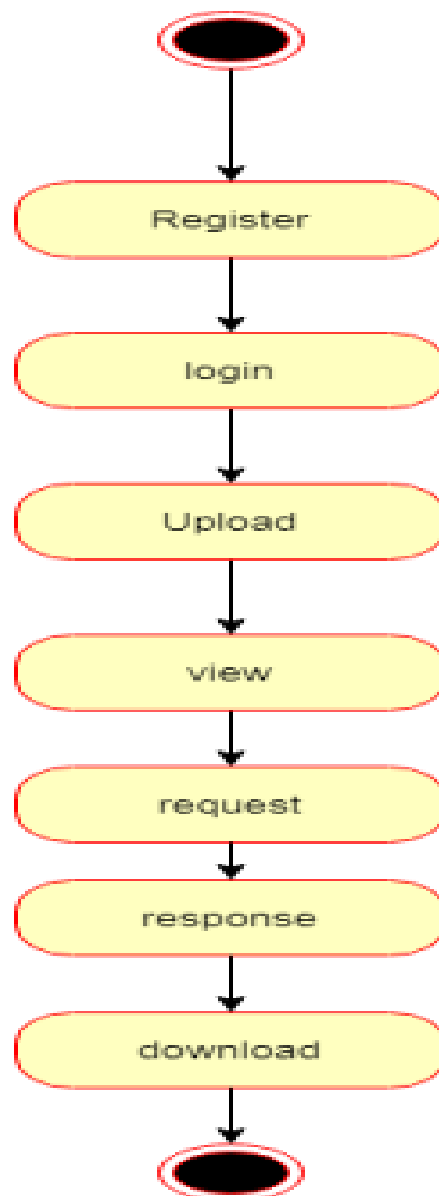


Figure 4.6 State Chart Diagram

4.8 DATA FLOW DIAGRAM

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 visualisation 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.

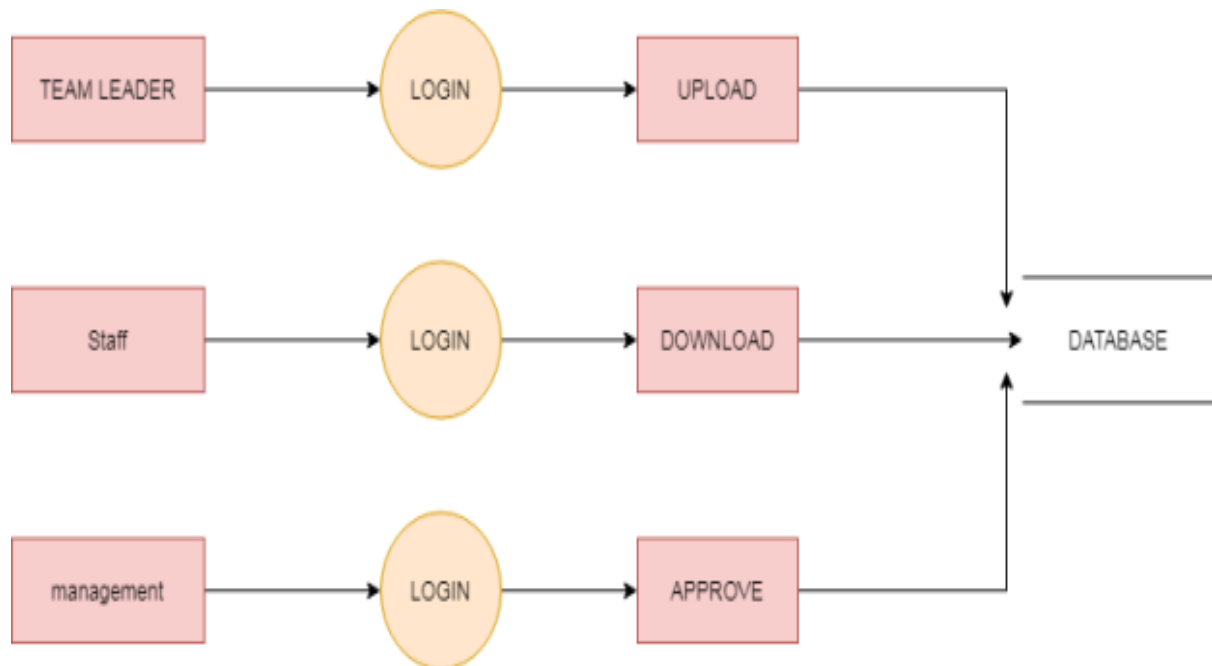


Figure 4.7 Data Flow Diagram

4.9 ER DIAGRAM

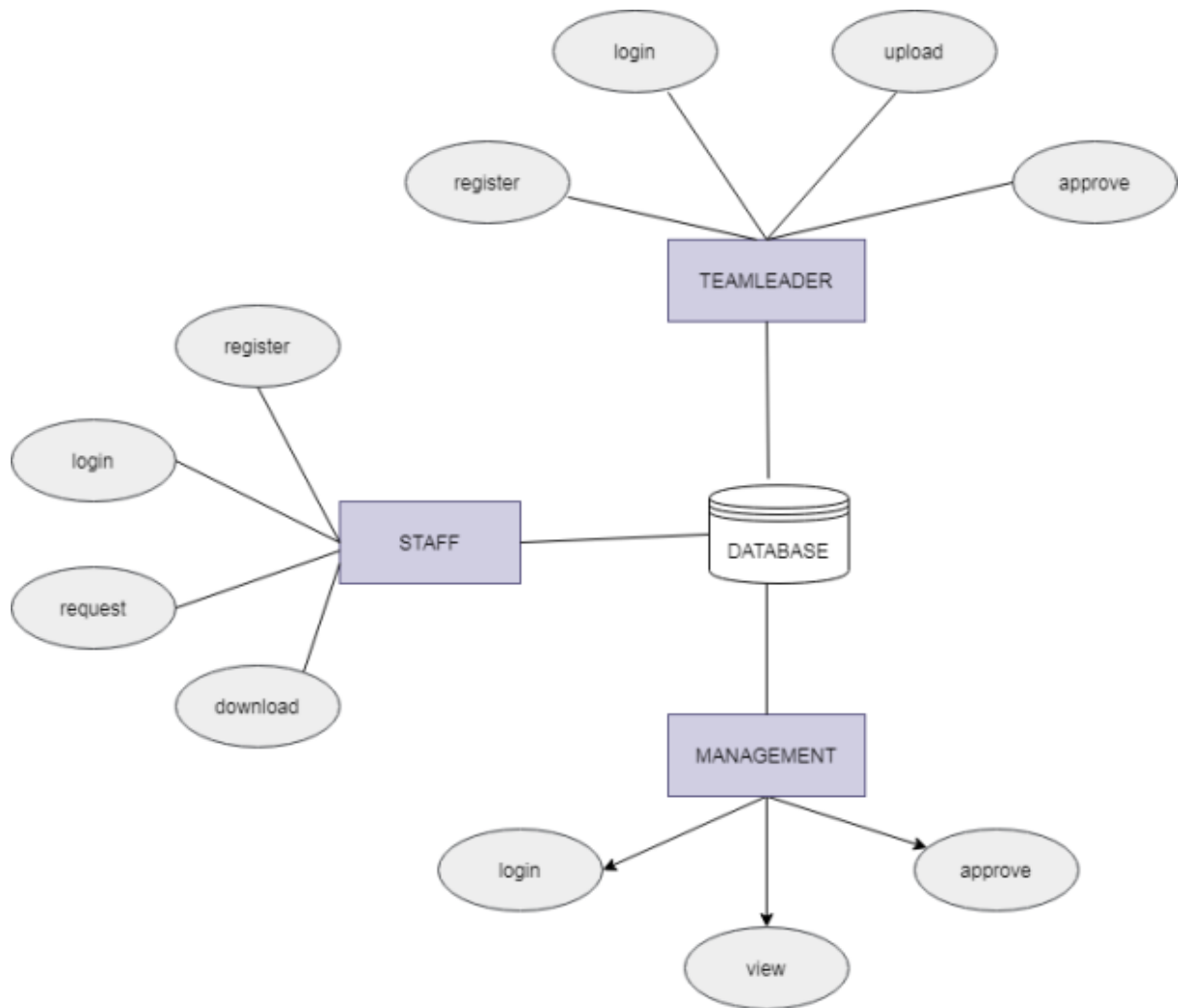


Figure 4.8 ER Diagram

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 PROPOSED SYSTEM

Concept:

The proposed smart system is used by the admin or authority involved in the production of sensitive data. The final sensitive data are produced by the final data administrator or authority person, and other modules involved in the process of data aggregation are unaware of the final data.

Technique:

SHA algorithm

Advantage:

It gives a standard and valid solution to process the data with a function.

5.2 MODULES :

1. STAFF REGISTER
2. STAFF LOGIN
3. STAFF FILE VIEW
4. STAFF FILE REQUEST
5. STAFF FILE DOWNLOAD
6. TEAM LEADER LOGIN
7. TEAM LEADER FILE UPLOAD
8. TEAM LEADER FILE VIEW

- 9. MANAGEMENT LOGIN
- 10.MANAGEMENT TEAM LEADER REGISTRATION
- 11.MANAGEMENT GENERATE KEY
- 12.MANAGEMENT RESPONSE

5.3 MODULE DESCRIPTION:

1. STAFF REGISTER:

The register module provides a conceptual framework for entering data on those staff in a way that: ease us 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.

2. STAFF LOGIN:

In this module in our project, here symbolises 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 the provider.

3. STAFF FILE VIEW:

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

4. STAFF FILE REQUEST:

This module is used to help the staff to request for download file with the land longitude and the user will update the report along with their opinion and it will be stored in the database.

5. STAFF FILE DOWNLOAD:

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

6. TEAM LEADER LOGIN:

In this module in our project, here symbolises 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 the provider.

7. 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 a database.

8. TEAM LEADER FILE VIEW:

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

9. MANAGEMENT LOGIN:

In this module in our project, here symbolises 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 users will transfer the amount to the provider.

10. MANAGEMENT TEAM LEADER REGISTRATION:

The register module provides a conceptual framework for entering data on those team leader in a way that: ease us 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.

11. MANAGEMENT GENERATE KEY:

In this module the management generates a key for the staff request. Because the key is for the security purpose. After getting the key from management the staff will download the file with the key.

12. MANAGEMENT RESPONSE:

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

5.4. ALGORITHM USED:

SHA ALGORITHM:

In the field of cryptography and crypt analytics, the SHA-1 algorithm is a crypt-formatted hash function that is used to take a smaller input and produces a string that is 160 bits, also known as 20-byte hash value long. The hash value therefore generated, is known as a message digest which is typically rendered and produced as a hexadecimal number which is specifically 40 digits long.

Characteristics:

1. The cryptographic hash functions are utilised and used to keep and store the secured form of data by providing three different kinds of characteristics such as pre-image resistance, which is also known as the first level of image resistance, the second level of pre-image resistance and collision resistance.
2. The cornerstone lies in the fact that the preimage crypt resistance technique makes it hard and more time consuming for the hacker or the attacker to find the original intended message by providing the respective hash value.
3. The security, therefore, is provided by the nature of a one way that has a function that is mostly the key component of the SHA algorithm. The pre-image resistance is important to clear off brute force attacks from a set of huge and powerful machines.
4. Similarly, the second resistance technique is applied where the attacker has to go through a hard time decoding the next error message even when the first level of the message has been decrypted. The last and most difficult to crack is the collision resistance, making it extremely hard for the attacker to find two completely different messages which hash to the same hash value.
5. Therefore, the ratio to the number of inputs and the outputs should be similar in fashion to comply with the pigeonhole principle. The collision resistance implies that finding two different sets of inputs that hash to the same hash is extremely difficult and therefore marks its safety.

Uses of SHA Algorithm:

These SHA algorithms are widely used in security protocols and applications, including the ones such as TLS, PGP, SSL, IPsec, and S/MiME. These also find their place in all the majority of cryptanalytic techniques and coding standards

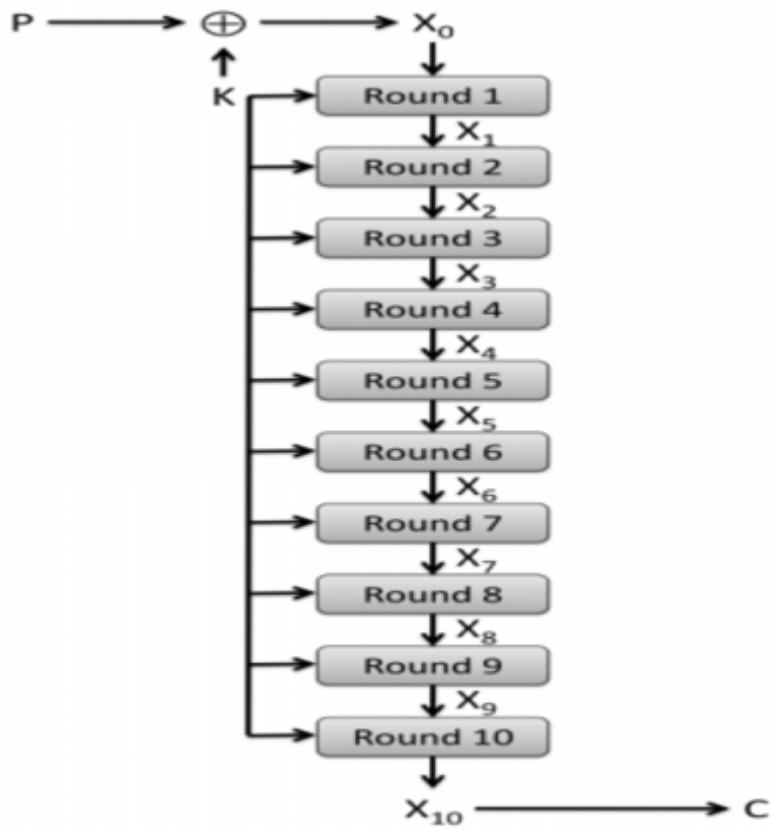
which is mainly aimed to see the functioning and working of majorly all governmental as well as private organisations and institutions. Major giants today such as Google, Microsoft, or Mozilla have started to recommend the use of SHA-3 and stop the usage of the SHA-1 algorithm.

AES ALGORITHM:

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

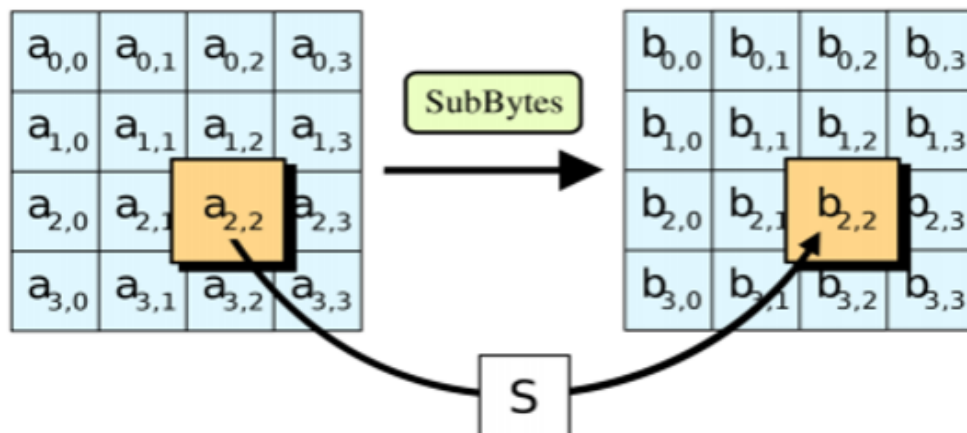
AES Working:

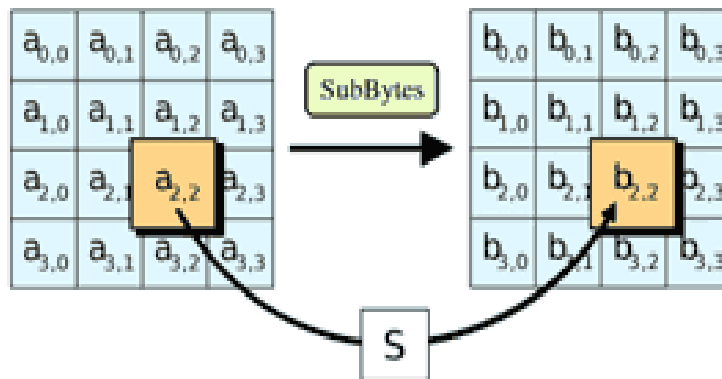
The AES algorithm uses a substitution-permutation, or SP network, with multiple rounds to produce ciphertext. The number of rounds depends on the key size being used. A 128-bit key size dictates ten rounds, a 192-bit key size dictates 12 rounds, and a 256-bit key size has 14 rounds. Each of these rounds 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.



Substitution of the bytes:

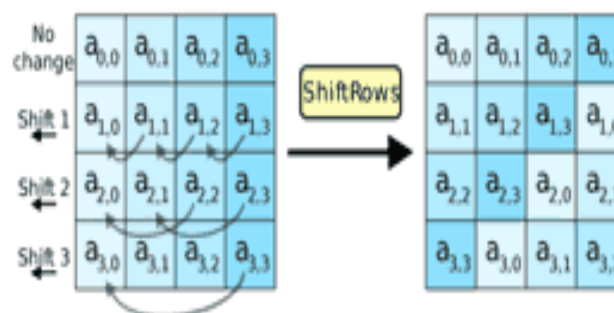
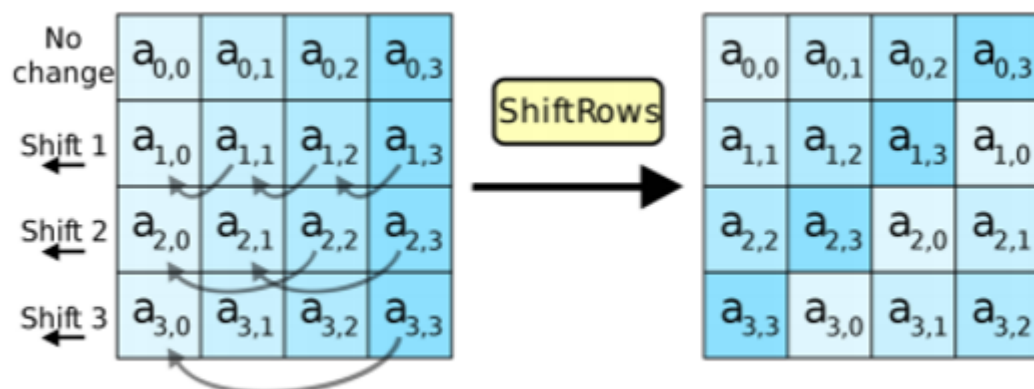
In the first step, the bytes of the block text are substituted based on rules dictated by predefined S-boxes (short for substitution boxes).





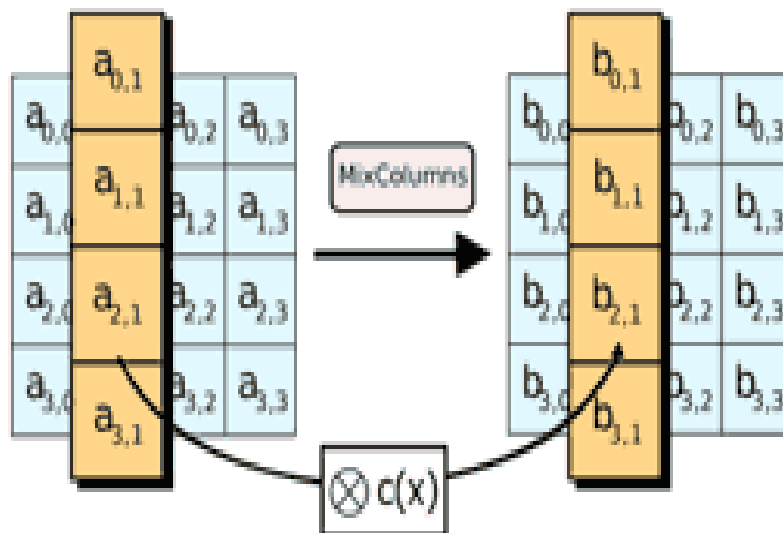
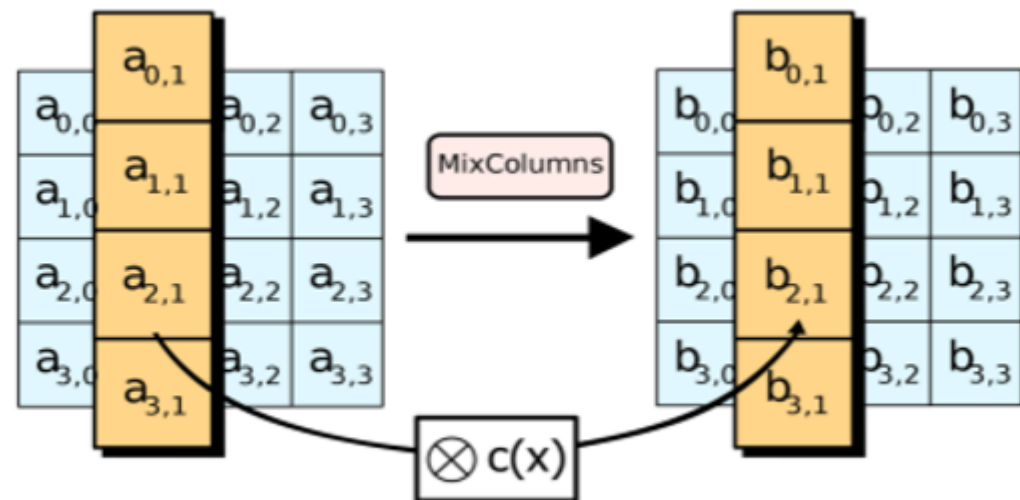
Shifting the rows

Next comes the permutation step. In this step, all rows except the first are shifted by one, as shown below.



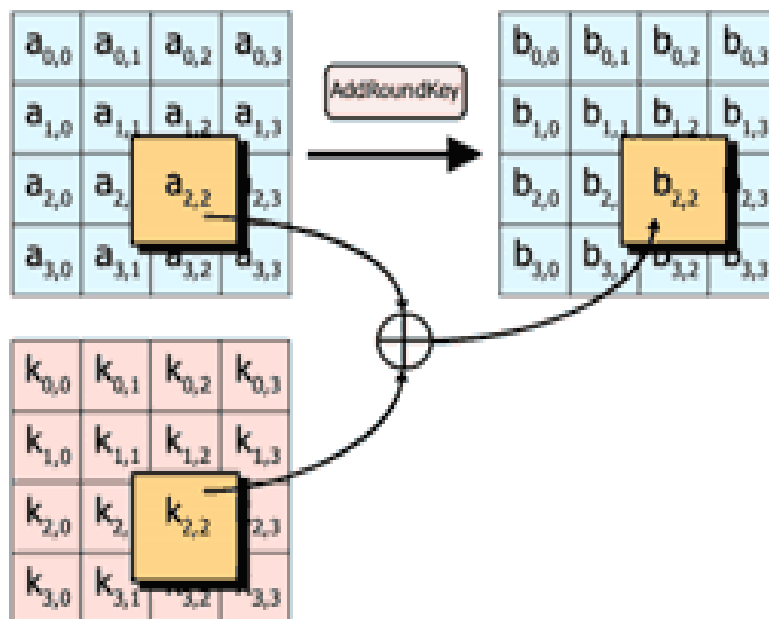
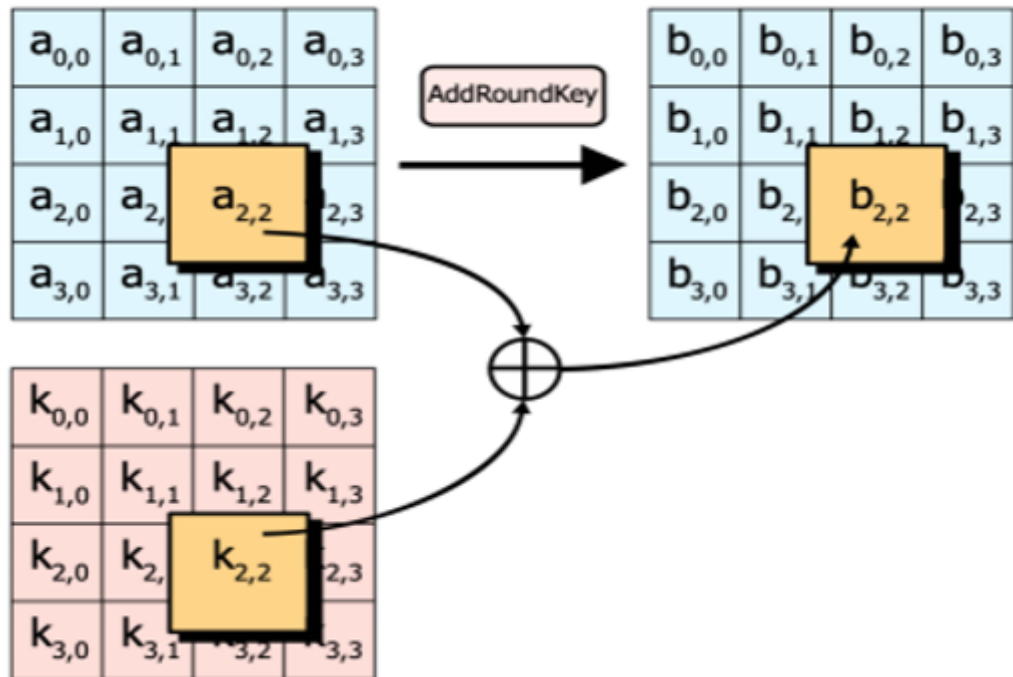
Mixing the columns

In the third step, the Hill cipher is used to jumble up the message more by mixing the block's columns.



Adding the round key

In the final step, the message is XORed with the respective round key.



CHAPTER 6

SYSTEM TESTING

6.1 FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

They are 3 types of Feasibility

1. Economical feasibility
2. Technical feasibility
3. Operational feasibility

6.1.1. ECONOMICAL FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not.

6.1.2. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

6.1.3. OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

6.2. SYSTEM TESTING

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases and makes it shine.

6.3. VARIOUS LEVELS OF TESTING

1. White Box Testing.
2. Black Box Testing.
3. Unit Testing.
4. Functional Testing.
5. Performance Testing.
6. Integration Testing.
7. Validation Testing.
8. System Testing.
9. Output Testing.
10. User Acceptance Testing.

6.3.1.1. WHITE BOX TESTING

White-box testing, sometimes called glass-box, is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that

6.3.1.2. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated as

a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

In this testing by knowing the internal operation of a product, a test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

6.3.1.3. UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each class is tested to be working satisfactorily.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs 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.

6.3.1.4. FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does. Functional testing differs from system testing in that functional testing "verifies a program

by checking it against ... design document(s) or specification(s)", while system testing "validate a program by checking it against the published user or system requirements.

6.3.1.5. PERFORMANCE TESTING:

In general testing is performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system.

6.3.1.6. INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when put together. The problem of course, is "putting them together"- interfacing. There may be the chances of data lost across on another's sub functions, when combined may not produce the desired major function; individually acceptable impressions may be magnified to unacceptable levels; global data structures can present problems.

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

6.3.1.7. VALIDATION TESTING

Verification and Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfils its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded with "Independent" (or

IV&V), indicating that the verification and validation is to be performed by a disinterested third party.

6.3.1.8. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

6.3.1.9. OUTPUT TESTING

After performing the validation testing, next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in two ways. One is on screen and another is printed format. The output comes as the specified requirements by the user. Hence output testing does not result in any correction in the system.

6.3.1.10. USER ACCEPTANCE TESTING

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

6.4. TEST RESULTS

```
INFO:symExec: ===== Results =====
INFO:symExec:   EVM Code Coverage:          50.3%
INFO:symExec:   Integer Underflow:                False
INFO:symExec:   Integer Overflow:                   True
INFO:symExec:   Parity Multisig Bug 2:              False
```

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

Data sensitivity concerns information that should be protected from unauthorised 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.

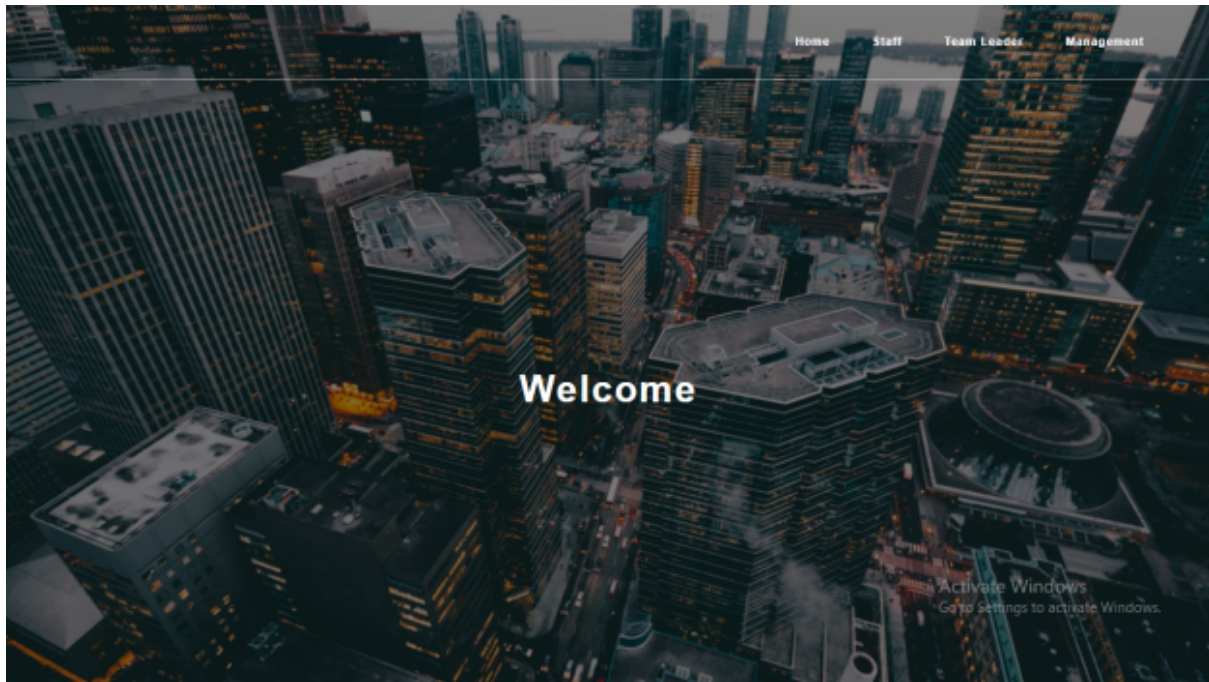
7.2 FUTURE ENHANCEMENT

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 or more algorithms.

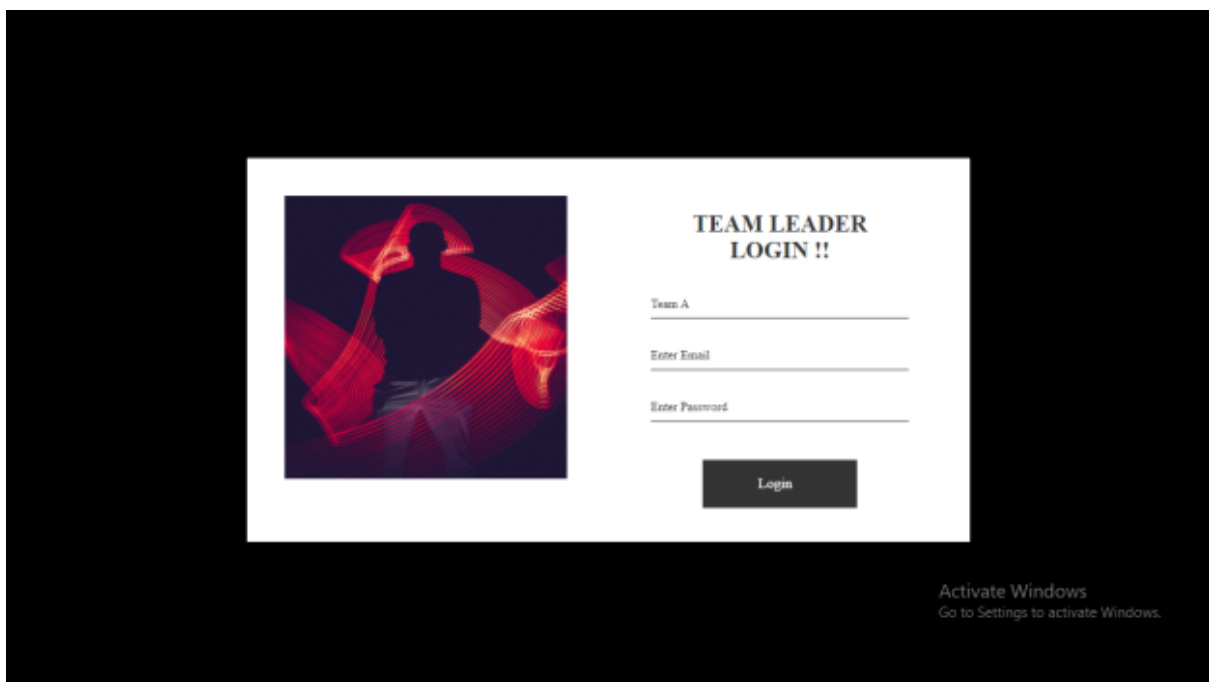
APPENDIX-I

SCREENSHOTS

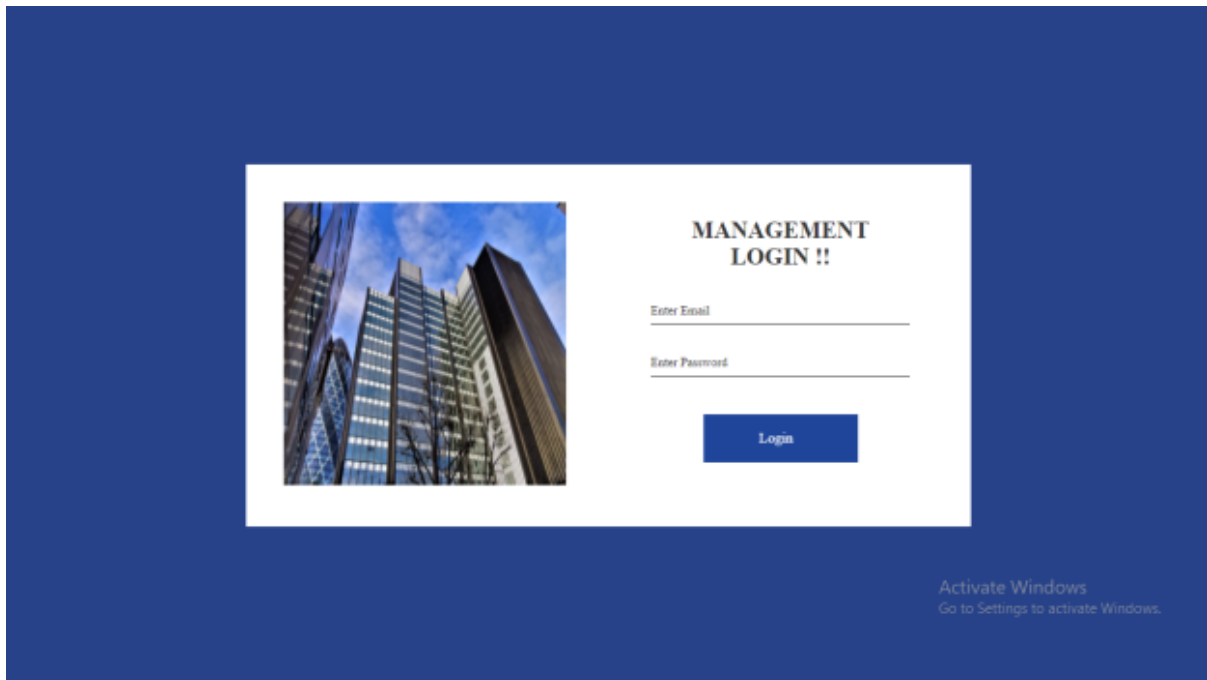
Home page:



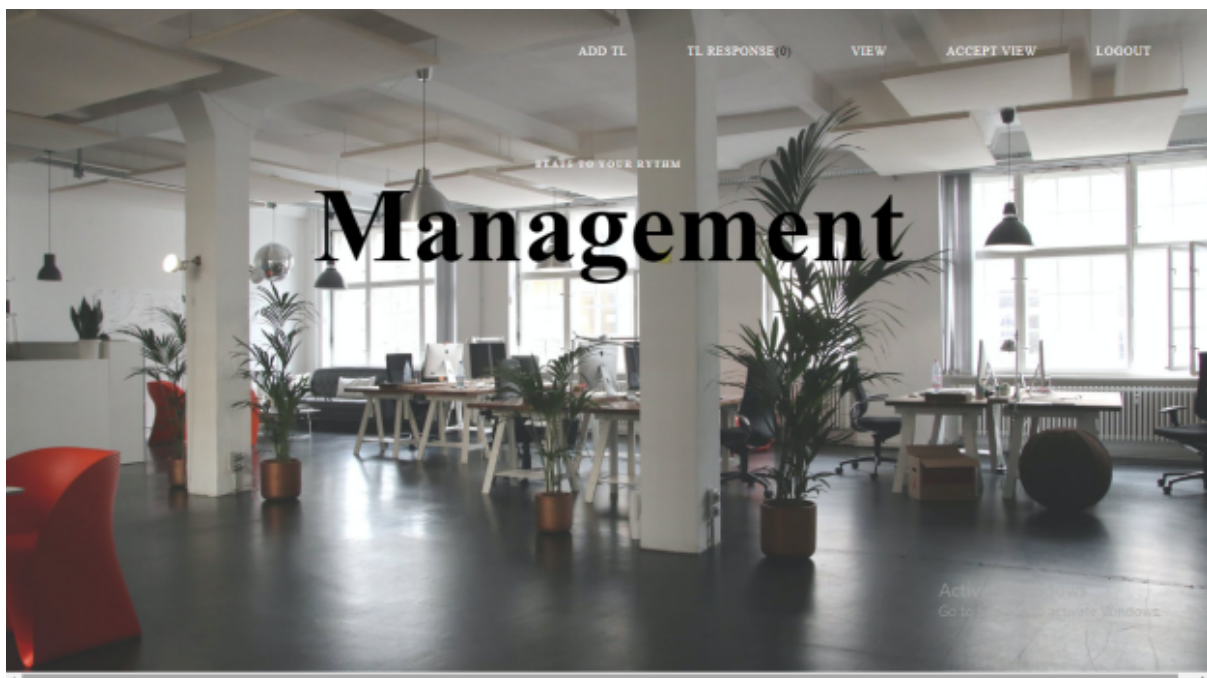
Team Leader Login Page:



Management Login Page:



Management Home Page:



Team Leader Add Page:

New Team Leader...!!! Back

Choose Team:
Team A

Name:
Enter Full Name

Email :
Enter Email

Mobile :
Enter contact No

Password :
Enter Password

Re-Enter password:
Confirm Password

Upload photo :
Choose File No file chosen

Picture Here!!!

Activate Windows
Go to Settings to activate Windows.

Team View Page:

Back

TeamA

Name : aaa
Email :aaa@gmail.com
Mobile :9966558877

Members

TeamB

Name : bbb
Email :bbb@gmail.com
Mobile :99556688

Members

TeamC

Name : ccc
Email :ccc@gmail.com
Mobile :9988556622

Members

TeamD

TeamC

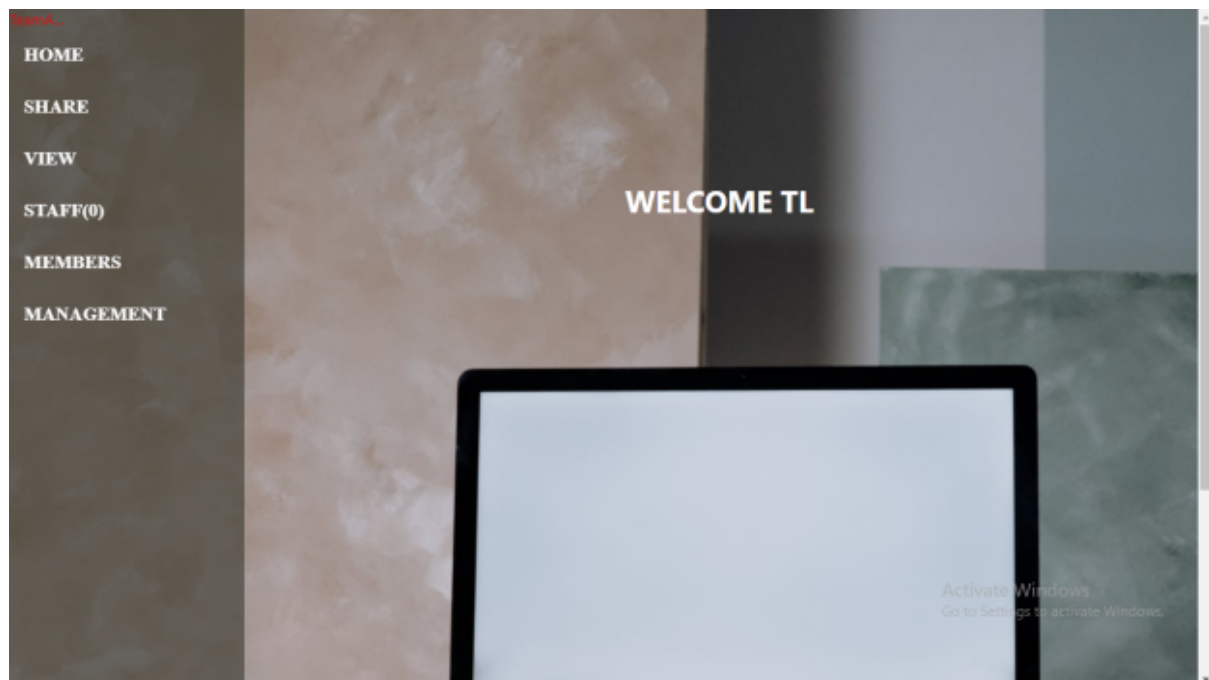
Activate Windows
Go to Settings to activate Windows.

Management Accept View Page:

[Back](#)

Title	Task	Filename	Staff	TL Email
test	testcase	bc024.pdf	mani@gmail.com	venkat@gmail.com
Project	for Java developer	new.pdf	prajith@gmail.com	syed@gmail.com
Review	Complete fromt end.,	daily statement.pdf	prajith@gmail.com	syed@gmail.com
project	Complete fromt end.,	new.pdf	paul@gmail.com	venkat@gmail.com
testing	testcase	bc024.pdf	ramu@gmail.com	dhina@gmail.com
sample project	using all	new.pdf	naveen@gmail.com	praveen@gmail.com
rdfv4qg	Complete fromt end.,	b5.pdf	aaa@gmail.com	aaa@gmail.com
weds	for Java developer	b6.pdf	bbb@gmail.com	bbb@gmail.com
weds	for Java developer	b6.pdf	bbb@gmail.com	bbb@gmail.com
dfcghihk	using all	base 2.pdf	ccc@gmail.com	ccc@gmail.com
qqwrtty	python	base4.pdf	ddd@gmail.com	ddd@gmail.com
cvrg	Complete fromt end.,	base 2.pdf	bbb@gmail.com	bbb@gmail.com
new project	testcase	b5.pdf	mani@gmail.com	saran@gmail.com
java	java work	sample.pdf	aaa@gmail.com	aaa@gmail.com
qqwrtty	python	base4.pdf	vig@gmail.com	ddd@gmail.com

Team Leader Login Page:



Team leader file Share page:

Back


SHARE FROM TEAMMEMBER

TITLE	<input type="text" value="Title"/>
Description	<input type="text" value="Description"/>
Email	<input type="text" value="aaa@gmail.com"/>
Team	<input type="text" value="TeamA"/>
File	<input type="button" value="Choose File"/> No file chosen
<input type="button" value="Submit"/>	


Activate Windows
Go to Settings to activate Windows.

Team members List:


Back




Name : paul
Email :paul@gmail.com
Mobile :9966558877
Picture :123



Name : aaa
Email :aaa@gmail.com
Mobile :9988556622
Picture :123



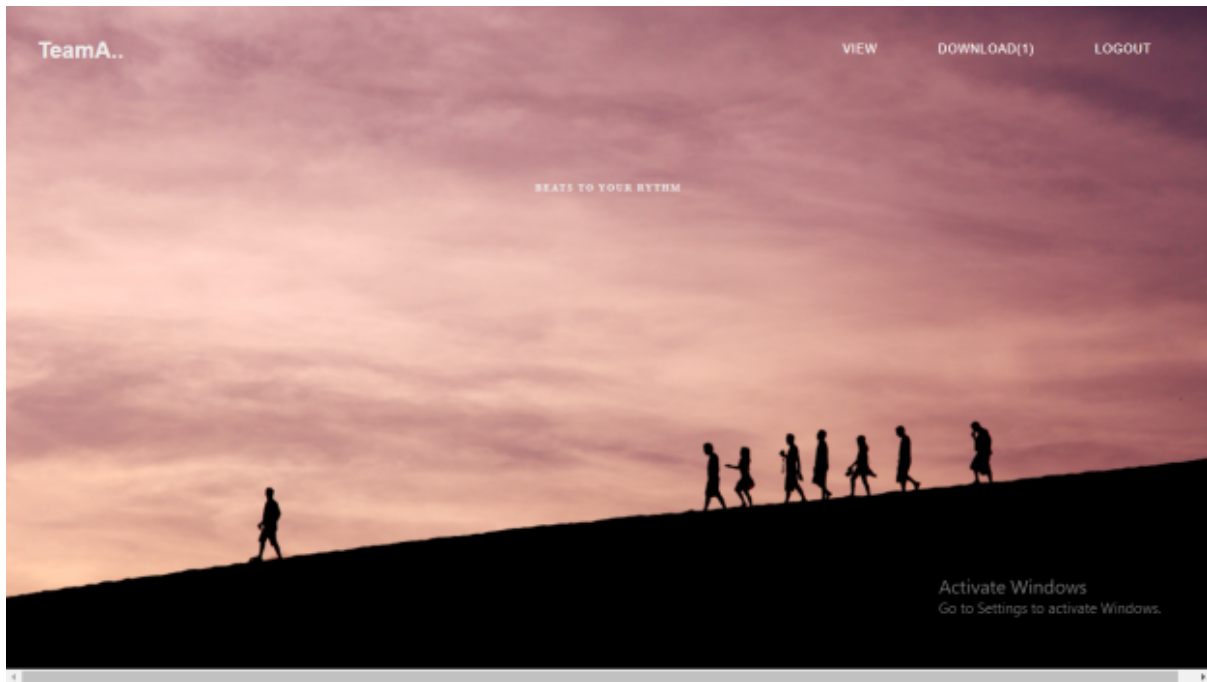
Name : ccc
Email :ccc@gmail.com
Mobile :9988556622
Picture :123



Name : ccc

Activate Windows
Go to Settings to activate Windows.

Staff Main Page:



Staff File View Page:

						Go Back
TITLE	DESCRIPTION	FILENAME	TLMAIL	TEAM	REQUEST	
test	testcase	bc024.pdf	venikat@gmail.com	TeamA	REQUEST	
project	Complete fromt end..	new.pdf	venikat@gmail.com	TeamA	REQUEST	
rdfv4qg	Complete fromt end..	b5.pdf	aaa@gmail.com	TeamA	REQUEST	
java	java work	sample.pdf	aaa@gmail.com	TeamA	REQUEST	

Activate Windows
Go to Settings to activate Windows.

Staff Download page:

Staff Download here!!!

[back](#)

Filename	TL Email	Filekey	QR Generate	Download
new.pdf	venkat@gmail.com	1BHDB	generate	Download

Activate Windows
Go to Settings to activate Windows.

REFERENCES:

1. D. Liu and J. Lee (2020), "CNN based Malicious Website Detection by Invalidating Multiple Web Spams," IEEE Access, vol. 8, no. 1, pp. 97258-97266.
2. D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos and G. Das (2018), "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.
3. N. Kakade and U. Patel, "Secure Secret Sharing Using Homomorphic Encryption (2020)," in Proc.11th International Conference on Computing, Communication and Networking Technologies, pp. 1-7.

4. P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo (2020), "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147-156.
5. S. Jiao, T. Lei, Y. Gao, Z. Xie and X. Yuan (2019), "Known-Plaintext Attack and Ciphertext-Only Attack for Encrypted Single-Pixel Imaging," *IEEE Access*, vol. 7, no.2, pp. 119557-119565.
6. S. Sundari and M. Ananthi (2020), "Secure multi-party computation in differential private data with Data Integrity Protection," in *Proc International Conference on Computing and Communications Technologies*, pp. 180-184.
7. S. Zhang, and J H. Lee (2020). "Mitigations on Sybil-based Double-spend Attacks in Bitcoin," *IEEE Consumer Electronics Magazine*, vol.7, no. 2, pp. 1-1.
8. W. Martin, V. Friedhelm, and K. Axel (2019), "Tracing manufacturing processes using blockchain-based token compositions," *Digital Communications and Networks*, vol. 6, no 2, pp. 167-176.
9. W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li and R. Chen (2019), "NutBaaS: A Blockchain-as-a-Service Platform," *IEEE Access*, vol. 7, pp. 134422-134433.
10. X. Wang, Q. Feng and J. Chai (2018), "The Research of Consortium Blockchain Dynamic Consensus Based on Data Transaction Evaluation," in *Proc.11th International Symposium on Computational Intelligence and Design*, pp. 214-217.

