CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

In the recent decades, cloud-based storage service has attracted considerable attention from both academia and industries. It may be widely used in many Internet-based commercial applications (e.g., Apple iCloud) due to its long-list benefits including access flexibility and free of local data management. Increasing number of individuals and companies nowadays prefer to outsource their data to remote cloud in such a way that they may reduce the cost of upgrading their local data management facilities/devices. However, the worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service. In many practical applications, outsourced data may need to be further shared with others. For example, a Dropbox user Alice may share photos with her friends. Without using data encryption, prior to sharing the photos, Alice needs to generate a sharing link and further share the link with friends. Although guaranteeing some level of access control over unauthorized users (e.g., those are not Alice's friends), the sharing link may be visible within the Dropbox administration level (e.g., administrator could reach the link).

Since the cloud (which is deployed in an open network) is not be fully trusted, it is generally recommended to encrypt the data prior to being uploaded to the cloud to ensure data security and privacy. One of the corresponding solutions is to directly employ an encryption technique (e.g., AES) on the outsourced data before uploading to cloud, so that only specified cloud user (with valid decryption key) can gain access to the data via valid decryption

Moreover, emerging solutions further enhance cloud security by enabling computation on encrypted data without revealing the actual content. As cloud storage technologies continue to evolve, research and innovation in access control frameworks, decentralized trust models, and efficient encryption techniques will play a pivotal role in shaping the future of secure cloud computing.

1.2 PROBLEM STATEMENT

Security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service. In many practical applications, outsourced data may need to be further shared with others. For example, a Dropbox user Alice may share photos with her friends.

Key Security Challenges in Cloud-Based Data Sharing

1. **Unauthorized Access :** Cloud providers manage vast amounts of data, and weak authentication measures can allow hackers or malicious insiders to access personal or corporate information.
2. **Data Interception :** Files shared over insecure channels can be intercepted by attackers using techniques like man-in-the-middle attacks.
3. **Data Integrity Risks :** Without proper verification mechanisms, data stored in the cloud could be altered or corrupted, leading to misinformation and lost trust.
4. **Lack of User Control :** Users might not have granular control over who can access, modify, or forward their shared files, increasing exposure to unintentional data leaks.
5. **Compliance & Legal Issues :** Many industries require strict adherence to regulations such as GDPR or HIPAA, and failure to secure data could result in legal consequences.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

A ROBUST AND SECURE MULTI-AUTHORITY ACCESS CONTROL SYSTEM FOR CLOUD STORAGE :

This research introduces a multi-authority access control system designed to enhance security and robustness in cloud storage environments. By involving multiple authorities in the access control process, the system aims to distribute trust and improve the overall security of data storage and sharing in the cloud.

Distributed Trust Model – Instead of a single authority managing access control, multiple entities oversee authentication and authorization, reducing the risk of a single point of failure.

Fine-Grained Access Control – Users are granted specific privileges based on predefined attributes, ensuring that data access is highly customizable.

Improved Resistance to Attacks – By decentralizing the access control mechanism, the system becomes more resilient against malicious attacks, insider threats, and unauthorized access.

ACCESS CONTROL TECHNOLOGIES FOR BIG DATA MANAGEMENT SYSTEMS :

This literature review examines various access control technologies applicable to big data management systems, including cloud storage. It discusses existing models, challenges, and potential future trends in access control mechanisms, providing a comprehensive overview of the field

Traditional Access Control Models – Overview of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), along with their limitations in large-scale systems.

Privacy-Preserving Access Control – Examination of techniques like differential privacy, homomorphic encryption, and secure multiparty computation to protect user data while enabling controlled access.

Policy-Based and Context-Aware Access Control – The role of dynamic policies and contextual attributes in enhancing security while allowing flexible data access.

Decentralized Access Control Approaches – Discussion of blockchain-based access control mechanisms that enable immutable and transparent permissions for secure data sharing.

COMBINING DATA OF OWNER SIDE AND CLOUD SIDE ACCESS CONTROL FOR ENCRYPTION CLOUD STORAGE :

This paper presents a scheme that combines owner-side and cloud-side access control mechanisms for encrypted cloud storage. By integrating Ciphertext-Policy Attribute-Based Encryption (CP-ABE) with cloud-side verification, the approach enhances security and prevents unauthorized access to sensitive data

Dual Access Control Mechanism – Combines owner-side encryption with cloud-side authorization, ensuring stronger security enforcement.

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) – Enables fine-grained access control by encrypting data based on predefined attributes.

Cloud-Side Verification & Authentication – Adds an additional layer of authentication, ensuring unauthorized users are blocked even if they obtain encrypted data.

Security & Performance Evaluation – The study includes experimental analysis to assess the effectiveness of the proposed mechanism in maintaining security while optimizing efficiency.

Owner-Side Encryption – Data owners encrypt their files before uploading them to the cloud, ensuring that only authorized users with the correct attributes can decrypt the data.

A DUAL SECURITY PROTECTION MECHANISM FOR CLOUD-BASED DATA STORAGE AND SHARING :

This research proposes a dual security protection mechanism combining data encryption and access control to safeguard cloud-based data storage. The mechanism addresses the need for secure data sharing while mitigating potential security threats in cloud environments.

Data Encryption – Sensitive data is encrypted before being uploaded to the cloud, ensuring that only authorized users with decryption keys can access it. Common encryption techniques include Advanced Encryption Standard.

Access Control – The mechanism enforces strict access control policies, preventing unauthorized users from downloading or modifying data. This helps mitigate threats like Economic Denial of Sustainability (EDoS) attacks, which aim to exhaust cloud resources.

Dual Access Control Systems – Two distinct access control models are designed. That are one for managing "data access" and another for regulating "download requests". This ensures both security and efficiency in cloud environments.

Security & Experimental Analysis – The research evaluates the effectiveness of the proposed mechanism through security assessments and experimental analysis.

DUAL ACCESS CONTROL FOR CLOUD-BASED DATA STORAGE AND SHARING USING AES ALGORITHM :

This study explores a dual access control mechanism that integrates data encryption using the AES algorithm with controls over download requests. The approach aims to protect data confidentiality and prevent Economic Denial of Sustainability (EDoS) attacks, ensuring both security and efficiency in cloud storage environment

AES Encryption – Data is encrypted using the AES algorithm before being stored in the cloud, ensuring that only authorized users with decryption keys can access it.

Access Control Mechanism – The system implements two layers of access control one for managing data access and another for regulating download requests to prevent unauthorized retrieval.

Mitigation of EDoS Attacks – By controlling download requests, the mechanism prevents malicious actors from exhausting cloud resources through excessive access attempts.

Security & Performance Evaluation – The study includes experimental analysis to assess the effectiveness of the proposed system in maintaining security without compromising efficiency.

A FRAMEWORK FOR RAPIDLY PROTOTYPING CRYPTOSYSTEMS :

The framework focuses on transitioning cryptographic schemes from research concepts to practical systems efficiently and effectively.

Implementation of Cryptographic Schemes – The framework supports the development of various cryptographic systems, including identity-based encryption, attribute-based encryption, and privacy-preserving schemes like ring signatures and group signatures.

It provides modular building blocks, allowing developers to combine components efficiently to create new cryptographic protocols.

Interactive Protocols – It facilitates the creation of interactive cryptographic protocols, which are essential for secure communication and data exchange in distributed systems.

Benchmarking and Performance – The framework includes tools for benchmarking, enabling developers to compare the performance of their implementations with existing systems.

While there may be a slight performance trade-off compared to traditional C implementations, the reduction in code complexity and size is a significant advantage.

Interoperability – Specialized tools ensure that different cryptosystems can work together seamlessly, making it easier to integrate new schemes into existing systems.

CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION :

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a cryptographic system that integrates access control policies directly into the encryption process.

Ciphertext-Policy Attribute-Based Encryption is a type of Attribute-Based Encryption (ABE) where

- **Encryption** – Defines an access policy over a set of attributes and encrypts the data under this policy. When an encryptor defines an access policy over a set of attributes and encrypts data under this policy, it typically refers to cryptographic methods used in systems like Attribute-Based Encryption (ABE).
- **Access Policy** – It determines who can access the encrypted data. It is a logical statement composed of attributes, such as "department: finance" or "role: manager."
- **Attributes** – These are descriptive values associated with users or entities and attributes act as criteria for decrypting the data.
- **Data Encryption Process** – The encryptor uses the access policy to encode restrictions into the encryption process. When data is encrypted, it incorporates these policy rules.
- **Key Issuance** – Users who want to access the encrypted data must have private keys associated with the relevant attributes. These keys are issued by a trusted authority or key issuer.
- **Decryption** – Only users whose attributes satisfy the defined access policy can decrypt the data. The decryptor possesses a private key associated with a set of attributes and it is possible only if the attributes of the decryptor's key match the access policy in the ciphertext. This includes the encrypted data plus the access policy defined by the

encryptor.A Private Key with Attributes are linked to the user and reflect their identity, role, department, clearance level, etc. and the key is generated using a master secret by the authority.

Core Idea – CP-ABE allows data owners to define access policies that determine who can decrypt the data. These policies are embedded in the ciphertext itself, ensuring that only users with the appropriate attributes can access the information.

Attributes and Policies – Attributes represent user credentials or characteristics (e.g., "Role: Manager" or "Department: HR").

The encryptor specifies a policy, such as "Role: Manager AND Department: HR," which defines the conditions under which the data can be decrypted.

Security – CP-ABE is resistant to collusion attacks, meaning unauthorized users cannot combine their attributes to gain access.

It ensures data confidentiality even if the storage server is compromised, as the access control is enforced cryptographically.

INNOVATIVE TECHNOLOGY FOR CPU BASED ATTESTATION AND SEALING :

In an era of increasing reliance on cloud computing, virtualization, and distributed systems, ensuring data confidentiality and integrity, even from privileged software like the operating system or hypervisor is critical. Traditional security architectures assume the OS is trusted, but this model breaks down when the OS can be compromised by rootkits or attackers with administrative access.

Attestation – The process of proving to an external party that a piece of code is running in a trusted, unmodified environment. Software Guard Extensions(SGX) enables remote attestation through

- Measurement of the code and data loaded into a secure enclave.
- Signing of this measurement using a cryptographic key that is embedded in the hardware and certified by Intel.
- A remote verifier can check this signature to ensure the correct enclave is running and it is running on genuine Intel hardware.

Sealing – Allows an enclave to store data securely so that only the same code (or enclave) can access it again in the future. Software Guard Extensions provides

- Hardware-backed encryption keys that are unique to the CPU or enclave identity.

- These keys are never exposed to software, even the OS, and are used to encrypt ("seal") the data.
- This ensures that even if the OS is compromised, previously sealed data remains confidential and tamper-proof.

Enclave Model and Security Guarantees – Software Guard Extensions introduces the concept of "enclaves", which are

- Protected memory regions created by user applications.
- Isolated from all other software, including the OS and hypervisor.
- Capable of executing code and accessing data while keeping it private and tamper-proof.

A REVOCABLE HYBRID ENCRYPTION SCHEME BASED ON ATTRIBUTE-BASED ENCRYPTION, SYMMETRIC SEARCHABLE ENCRYPTION AND SGX :

It introduces a hybrid encryption framework designed to enhance secure cloud storage by integrating multiple cryptographic techniques.

Attribute-Based Encryption (ABE) – ABE allows data to be encrypted under an access policy defined over user attributes. Only users possessing attributes that satisfy the policy can decrypt the data. This ensures that access control is enforced cryptographically, reducing reliance on external access control mechanisms.

Symmetric Searchable Encryption (SSE) – SSE permits users to perform keyword searches on encrypted data without revealing the plaintext or the search terms to the server. This maintains data confidentiality while providing search functionality, which is essential for usability in cloud storage systems.

Intel SGX for Revocation – User revocation is a critical aspect of access control. Traditional ABE schemes often face challenges in efficiently revoking users without re-encrypting data or updating keys for all users. By leveraging SGX, the scheme offloads revocation management to a secure enclave, enabling Cryptology ePrint Archive

- **Efficient Revocation :** Revocation decisions are made within the SGX enclave, allowing for immediate and secure enforcement without extensive re-encryption.
- **Separation of Concerns :** Revocation is handled independently of the ABE scheme, simplifying the overall system design and improving scalability.

- **Fine-Grained Access Control :** ABE ensures that only users with appropriate attributes can access specific data.
- **Efficient Search :** SSE allows users to search over encrypted data without compromising security.
- **Secure and Efficient Revocation :** SGX provides a secure environment for managing revocation, reducing the overhead associated with traditional revocation methods.
- **Enhanced Security :** By combining these technologies, the scheme offers robust protection against both external and internal threats.

SECURE SCHEMES FOR SECRET SHARING AND KEY DISTRIBUTION:

The work titled *"*Secure Schemes For Secret Sharing And Key Distribution*"* is a foundational contribution to the fields of cryptography, specifically focusing on secret sharing and key distribution mechanisms. It investigates how sensitive information or cryptographic keys can be divided among multiple users in such a way that only certain predefined groups, known as authorized subsets, are capable of reconstructing the original secret. These groups are defined through logical structures called access structures, which allow for a flexible and secure way to control who gets access to specific data. The research explores general access structures beyond simple threshold-based models, enabling more sophisticated forms of secret management in distributed systems.

A key concept introduced is that of monotonic access structures, where if a group of users is permitted to reconstruct a secret, then any larger group containing that subset is also permitted. This idea is important in real-world applications where hierarchical or role-based access is needed.

The work also emphasizes the importance of efficiency, presenting methods to reduce the size of the data shares distributed to users while maintaining strong security guarantees. In addition, it explores secure key distribution protocols that ensure only users in authorized sets can derive shared keys, a principle crucial to many secure multi-user systems.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Although being able to support fine-grained data access, CP-ABE, acting as a single solution, is far from practical and effective to hold against EDoS attack which is the case of DDoS in the cloud setting. Several countermeasures to the attack have been proposed in the literature. But stated that the previous works could not fully defend the EDoS attack in the algorithmic or protocol level, and they further proposed a solution to secure cloud data sharing from the attack.

However, suffers from two disadvantages. First, the data owner is required to generate a set of challenge ciphertexts in order to resist the attack, which enhances its computational burden. Second, a data user is required to decrypt one of the challenges ciphertexts as a test, which costs a plenty of expensive operations (e.g., pairing). Here the computational complexity of both parties is inevitably increased and meanwhile, high network bandwidth is required for the delivery of ciphertexts. The considerable computational power of cloud is not fully considered in will present a new solution that requires less computation and communication cost to stand still in front of the EDoS attack.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

- The system was not implemented Ciphertext-Policy Attribute-based-Encryption Method which leads less security on outsourced data.
- The system is less security due to lack of Authenticated Encryption with Associated Data.

3.2 PROPOSED SYSTEM

- In this project, propose a new mechanism, dubbed dual access control, to tackle the above aforementioned two problems. To secure data in cloud-based storage service, Attribute Based Encryption (ABE) is one of the promising candidates that enables the confidentiality of outsourced data as well as fine-grained control over the outsourced data.
- In particular, Ciphertext-Policy ABE (CP-ABE) provides an effective way of data encryption such that access policies, defining the access privilege of potential data receivers, can be specified over encrypted data. Note that consider the use of CP-ABE in

our mechanism in this paper. Nevertheless, simply employing CP-ABE technique is not sufficient to design an elegant mechanism guaranteeing the control of both data access and download request.

- A strawman solution to the control of download request is to leverage dummy ciphertexts to verify data receiver's decryption rights. It, concretely, requires data owner, say Alice, to upload multiple "testing" ciphertexts along with the "real" encryption of data to cloud, where the "testing" ciphertexts are the encryptions of dummy messages under the same access policy as that of the "real" data. After receiving a download request from a user, say Bob, cloud asks Bob to randomly decrypt one of the "testing" ciphertexts. If a correct result/decryption is returned (i.e. indicating Bob is with valid decryption rights), Bob is authorized by Alice to access the" real" data, so that the cloud allows Bob to download the corresponding ciphertext.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

(1) Confidentiality of outsourced data : In our proposed systems, the outsourced data is encrypted prior to being uploaded to cloud. No one can access them without valid access rights.

(2) Anonymity of data sharing : Given an outsourced data, cloud server cannot identify data owner, so that the anonymity of owner can be guaranteed in data storage and sharing.

(3) Fine-grained access control over outsourced (encrypted) data : Data owner keeps controlling his encrypted data via access policy after uploading the data to cloud. In particular, a data owner can encrypt his outsourced data under a specified access policy such that only a group of authorized data users, matching the access policy, can access the data.

(4) Control over anonymous download request and EDoS attacks resistance : A cloud server is able to control the download request issued by any system user, where the download request can set to be anonymous. With the control over download request, we state that our systems are resistant to EDoS attacks.

(5) High efficiency : Our proposed systems are built on the top of the CP-ABE system. Compared with, they do not incur significant additional computation and communication overhead. This makes the systems feasible for real-world applications.

3.3 PROJECT ALGORITHM

Diffie-Hellman algorithm:

The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

For the sake of simplicity and practical implementation of the algorithm, we will consider only 4 variables, one prime P and G (a primitive root of P) and two private values a and b.

P and G are both publicly available numbers. Users (say Alice and Bob) pick private values a and b and they generate a key and exchange it publicly. The opposite person receives the key and that generates a secret key, after which they have the same secret key to encrypt.

How Diffie-Hellman Works:

Public Parameters Selection:

Two parties (Alice and Bob) agree on a **large prime number** ppp and a **primitive root (generator)** ggg.

These values are publicly known.

Private Key Selection:

Alice chooses a private key aaa and keeps it secret.

Bob chooses a private key bbb and keeps it secret.

Public Key Computation:

Alice computes her public key: A=gamod pA = g^a \mod pA=gamodp and sends it to Bob.

Bob computes his public key: B=gbmod pB = g^b \mod pB=gbmodp and sends it to Alice.

Shared Secret Key Computation:

Alice computes: S=Bamod pS = B^a \mod pS=Bamodp.

Bob computes: S=Abmod pS = A^b \mod pS=Abmodp.

Both result in the same shared secret SSS, which is used for encryption.

Security of Diffie-Hellman:

The security relies on the Discrete Logarithm Problem (DLP)**,** which is computationally difficult to solve.

Even if an attacker intercepts g,p,A,Bg, p, A, Bg,p,A,B, they cannot easily determine the private keys aaa or bbb or compute the shared secret SSS.

Use in Cloud Security:

Diffie-Hellman can be used to establish **secure session keys** for encrypting cloud data storage and sharing.

It enhances **multi-authority access control** by enabling secure communication between cloud servers and authorized users.

3.4 SYSTEM REQUIREMENTS:

3.4.1 HARDWARE REQUIREMENTS

- Processor - Intel (R) Core (TM) i3-4200U
- CPU - 1.6GHz
- RAM - 4 GB
- Hard Disk - 500 GB.

3.4.2 SOFTWARE REQUIREMENTS

- Operating System - Windows 10
- Server - Tomcat
- Database - MYSQL
- Frontend - HTML, CSS, JS
- Backend - JSP

3.5 FEASIBILITY STUDY

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

- **Operational Feasibility**
- **Economic Feasibility**
- **Technical Feasibility**

3.5.1 OPERATIONAL FEASIBILITY

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

It is one of the key components of a feasibility study. It focuses on determining whether the proposed system will function effectively within the existing organizational environment and whether it will be accepted and used by the intended users. This aspect of feasibility deals with the practical implementation of the system and assesses whether it will meet the needs of the users, improve current processes, and integrate smoothly into daily operations.

The goal of operational feasibility is to evaluate the extent to which the proposed system will solve the existing problems and support the business or operational goals of the organization. It involves analysing how the system will operate once implemented and how well it aligns with current workflows, user expectations, organizational culture, and management structures.

In the context of the current project, the system under development is designed to automate the administrative functions involved in tracking project progress. Previously, the administration had to rely on manual processes, which were often time-consuming, error-prone, and inefficient. These manual tasks not only consumed significant amounts of time and energy but also increased the likelihood of miscommunication, data loss, or delayed responses.

The new automated system aims to streamline these tasks by providing real-time tracking, notifications, data entry, and reporting functionalities. With the implementation of this system, the administrator will be able to monitor progress more efficiently, allocate resources more effectively, and ensure that all tasks are completed within the desired timeline. The system is designed with user-friendly interfaces and logical workflows, ensuring that users can adopt the system with minimal training. The operational benefits of the proposed system are significant. It enhances the administrator's control over the project by providing immediate access to project data and status updates. The system also facilitates better decision-making through accurate and timely reporting tools. These features contribute to a more organized and responsive administrative process, reducing the stress and workload on the administrator.

Moreover, operational feasibility is supported by the system's alignment with the organization's existing infrastructure. The system does not require drastic changes to hardware or existing IT resources, making it easy to adopt and integrate. User involvement during the development process ensures that the system is tailored to real-world requirements, further increasing the likelihood of acceptance. After conducting a thorough analysis and testing, it has been concluded that the system is operationally feasible. It effectively meets the operational goals, fits well into the current administrative framework, and significantly improves the overall efficiency of project management tasks. The system's automation features, ease of use, and relevance to the administrator's daily responsibilities demonstrate that it will be a valuable addition to the organization.

In conclusion, operational feasibility confirms that the proposed system is not only technically sound and functionally capable but also practically useful and acceptable to the users. Its successful implementation is expected to bring about improved performance, higher productivity, and a more streamlined workflow.

3.5.2 ECONOMIC FEASIBILITY

Economic Feasibility is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at any time. The Virtual

Private Network is to be developed using the existing resources of the organization. So, the project is economically feasible.

Economic feasibility, also known as cost-benefit analysis, is a critical component of the overall feasibility study. It focuses on evaluating whether the benefits of implementing a proposed system outweigh its associated costs. The objective is to determine whether the system is a sound financial investment and whether the organization has sufficient financial resources to support its development, deployment, and maintenance. A system is considered economically feasible if it delivers clear value while minimizing expenditure and making efficient use of available resources.

In the context of this project, economic feasibility has been carefully examined, and the findings indicate that the system is both cost-effective and resource-efficient. One of the key factors contributing to its economic viability is the existing hardware infrastructure. The organization had already installed the necessary hardware for other operational purposes, which significantly reduces the need for additional investment in new equipment. As a result, the project does not require substantial upfront hardware costs, making it financially sustainable from the start.

Another major economic advantage is the system's network-based architecture. Since the tool is designed to function over a Local Area Network (LAN), it can be accessed by multiple employees within the organization without the need for individual installations or separate systems. This centralized access model minimizes deployment costs and ensures that the organization gets maximum value from a single implementation. Employees connected to the LAN can conveniently access the system at any time, thereby improving operational efficiency without incurring extra usage costs. Moreover, the use of a Virtual Private Network (VPN) enhances both security and remote accessibility. Importantly, the VPN infrastructure will be developed using existing resources, which further reduces overall costs. There is no need for external service subscriptions or additional hardware purchases to enable this secure communication channel. The reuse of current organizational resources demonstrates a strategic approach to minimizing expenses while maximizing functionality.

In addition to reducing upfront costs, the system is expected to generate long-term economic benefits. By automating administrative tasks and project tracking processes, it significantly reduces the time and effort required to manage these functions manually. This increased efficiency leads to reduced labor costs, faster decision-making, and better allocation of organizational

resources. In the long run, these benefits translate into improved productivity and cost savings, making the project a worthwhile financial investment. There are also indirect cost benefits associated with improved accuracy and data handling. Since manual processes are more prone to errors, automation reduces the likelihood of costly mistakes, delays, or miscommunication. These factors can greatly influence the overall financial impact of the system, supporting the conclusion that the economic advantages outweigh the minimal investment involved.

In conclusion, the analysis confirms that the system is economically feasible. It makes use of the organization's existing hardware and network infrastructure, avoids unnecessary expenditures, and promises long-term savings through increased operational efficiency. The careful balance between cost control and benefit realization ensures that the project will deliver financial value, making it a smart and sustainable investment for the organization.

3.5.3 TECHNICAL FEASIBILITY

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform independent environment, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

Technical feasibility is one of the fundamental aspects of a feasibility study. It evaluates the technical resources, infrastructure, and skills available within an organization to determine whether the proposed system can be successfully developed and implemented. According to software engineering expert Roger S. Pressman, technical feasibility is the assessment of whether the organization has the hardware, software, and human expertise required to build and maintain the system. In this case, the organization already possesses the essential technical foundation necessary to support the development of the proposed system. The required hardware includes IBM-compatible machines that are widely used and standardized within the organization. These machines are equipped with modern graphical web browsers and are connected to both the Internet and the organization's internal Intranet. This ensures that the end-users will have seamless access to the system from any compatible terminal within the network. The system is designed to be platform-independent, making it flexible and scalable across various devices and operating

systems. This characteristic not only ensures long-term adaptability but also reduces dependency on a specific platform or environment, which is beneficial for maintenance and future upgrades. The development tools and technologies used include Python, JavaScript, HTML, SQL Server, and WebLogic Server all of which are modern, widely-supported, and well-understood within the IT industry.

The use of these technologies is significant for several reasons. Python offers a powerful and flexible backend with excellent support for automation, data handling, and integration with other systems. JavaScript and HTML form the foundation of the system's user interface, ensuring an interactive, responsive, and browser-friendly experience for users. SQL Server is employed to manage the database layer, ensuring secure, reliable, and efficient storage and retrieval of data. WebLogic Server provides the necessary support for deploying enterprise-level web applications in a stable, scalable, and secure environment. The organization has access to skilled personnel with experience in all of these technologies, which removes the need for external hiring or costly training programs. The IT team is already familiar with the technical infrastructure, enabling faster development and smoother deployment. In addition, because these technologies are mainstream and well-documented, the risk of technical failure or incompatibility is minimal. Moreover, the existing technical setup — including hardware, networking, development tools, and human expertise — aligns well with the requirements of the system. There is no need for significant investment in new equipment or technology, making the system not only technically feasible but also economically sensible. The technical feasibility study confirms that the system can be successfully developed and integrated into the current environment without any major obstacles.

In conclusion, the technical feasibility of the project has been thoroughly assessed and validated. The available infrastructure, compatible hardware, experienced staff, and chosen technology stack all support the development of the proposed system.

CHAPTER-4

SYSTEM DESIGN

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

4.1 ARCHITECTURE

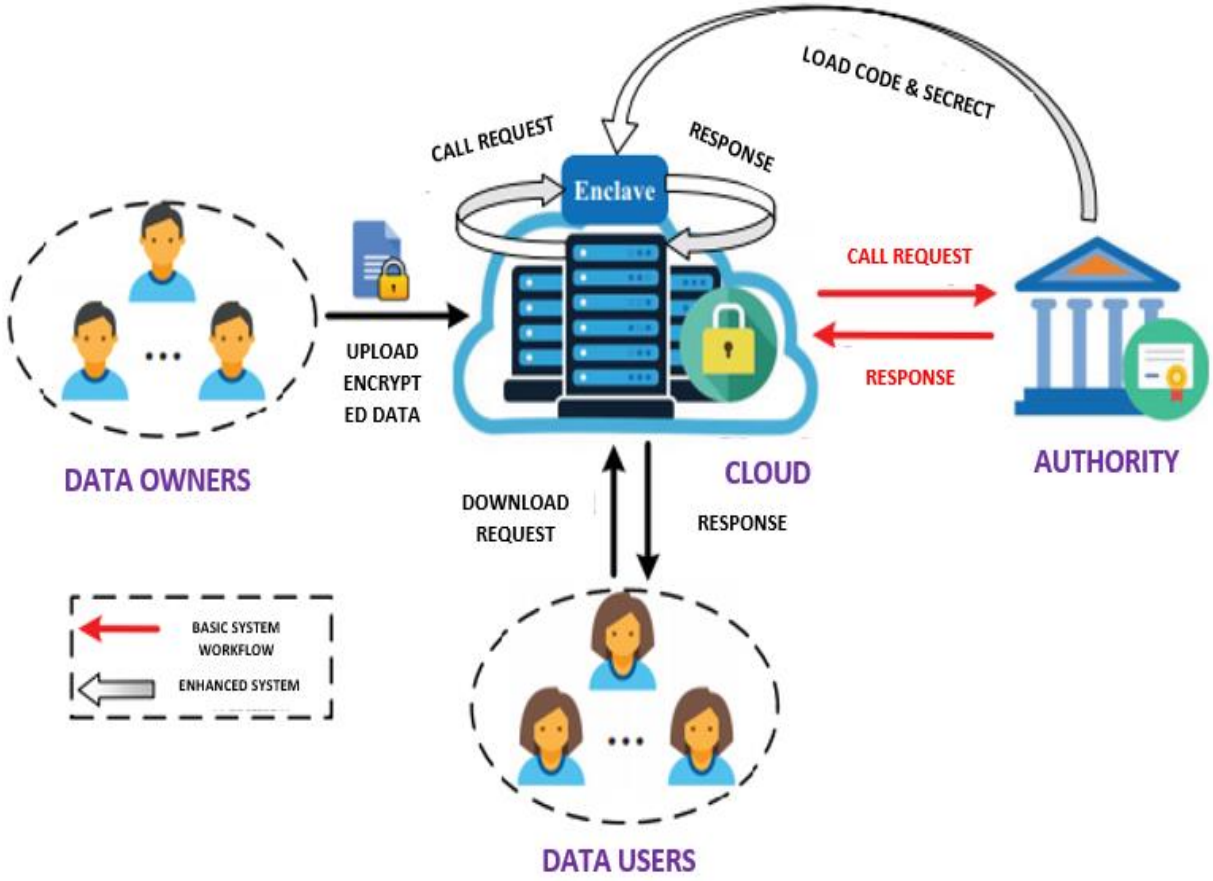


Fig 4.1 System Architecture

The image illustrates a dual access control system for cloud-based data storage and sharing, highlighting the interaction between data owners, cloud infrastructure, an authority, and data users. The system ensures that data remains secure and accessible only to authorized users by implementing encryption and an enclave for secure execution.

In this framework, data owners upload their encrypted data to the cloud, ensuring confidentiality before storage. The cloud serves as the central repository, where data is securely stored and processed. Within the cloud, an enclave is used to execute sensitive operations, providing a trusted environment for handling security mechanisms. The authority plays a crucial role in managing authentication and access control policies, responding to verification requests from the cloud. It helps in ensuring that only legitimate users gain access by providing necessary security credentials.

Data users request access to the stored data, and the cloud verifies their credentials before responding with the requested information. The interaction between the cloud and the authority, represented by red arrows, signifies the basic system workflow, while the black arrows represent an enhanced security mechanism integrating the enclave for improved protection. This system is designed to prevent unauthorized access and data breaches while maintaining efficient and seamless sharing of data in a cloud environment

4.2 MODULES

In this Proposed System Modules are:

Data Owner

In this module, the data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the file and then store in the cloud. The data owner can have capable of updating and deleting of a specific file. And also, he can view the transactions based on the files he uploaded to cloud.

End User

In this module, receiver's login is by using his/her user's name and password. After Login receiver will Search for files and request for secret key of a particular file from Authority, and get the secret key. After getting secret key he is trying to download file by entering file name and secret key from cloud server.

Authority

In this module, the authority helps to check transaction of files and also. If receiver exists and the profile. Authority also views the requests from the receivers and generates the secret key and send to the requested data receivers.

Cloud

In this module the functionalities are: View & authorize Data owners. View& authorize End Users, View Files with master Secret Key, View MSK Req / Res time and View files without master secret key

4.3 UML DIAGRAMS

Introduction To UML

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well asfor business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling oflarge and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostlygraphical 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.

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality andreduce cost and time to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modeling Language (UML) was designed to respond to these needs. Simply, Systems design refers to the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

Object-Oriented Concepts

UML can be described as the successor of object-oriented (OO) analysis and design.

An object contains both data and methods that control the data. The data represents the state of the object. A class describes an object and they also form a hierarchy to model the real-world system. The hierarchy is represented as inheritance and the classes can also be associated in different ways as per the requirement.

Objects are the real-world entities that exist around us and the basic concepts such as abstraction, encapsulation, inheritance, and polymorphism all can be represented using UML.

UML is powerful enough to represent all the concepts that exist in object-oriented analysis and design. UML diagrams are representation of object-oriented concepts only. Thus, before learning UML, it becomes important to understand OO concept in detail.

Following are some fundamental concepts of the object-oriented world –

- **Objects** − Objects represent an entity and the basic building block.
- **Class** − Class is the blue print of an object.
- **Abstraction** − Abstraction represents the behavior of a real-world entity.
- **Encapsulation** − Encapsulation is the mechanism of binding the data together and hiding them from the outside world.
- **Inheritance** − It is the mechanism of making new classes from existing ones.
- **Polymorphism** − It defines the mechanism to exists in different forms.

OO Analysis and Design

OO can be defined as an investigation and to be more specific, it is the investigation of objects. Design means collaboration of identified objects.

Thus, it is important to understand the OO analysis and design concepts. The most important purpose of OO analysis is to identify objects of a system to be designed. This analysis is also done for an existing system. Now an efficient analysis is only possible when we are able to start thinking in a way where objects can be identified. After identifying the objects, their relationships are identified and finally the design is produced.

The purpose of OO analysis and design can describe as

- Identifying the objects of a system.
- Identifying their relationships.
- Making a design, which can be converted to executables using OO languages.

There are three basic steps where the OO concepts are applied and implemented. The steps can be defined as

OO Analysis→ OO Design→ OO implementation using OO languages

The above three points can be described in detail as

- During OO analysis, the most important purpose is to identify objects and describe them in a proper way. If these objects are identified efficiently, then the next job of design is easy. The objects should be identified with responsibilities.. Each and every object has some type of responsibilities to be performed. When these responsibilities are collaborated, the purpose of the system is fulfilled.
- The second phase is OO design. During this phase, emphasis is placed on the requirements and their fulfilment. In this stage, the objects are collaborated according to their intended association. After the association is complete, the design is also complete.
- The third phase is OO implementation. In this phase, the design is implemented using OO languages such as Java, C++, etc.

Role Of UML In OO Design

UML is a modeling language used to model software and non-software systems. Although UML is used for non-software systems, the emphasis is on modeling OO software applications. Most of the UML diagrams discussed so far are used to model different aspects such as static, dynamic, etc. Now whatever be the aspect, the artifacts are nothing but objects.

Hence, the relation between OO design and UML is very important to understand. The OO design is transformed into UML diagrams according to the requirement.

In this project, basic UML diagrams have been explained

1) Use Case Diagram
2) Class Diagram
3) Sequence Diagram
4) Collaboration Diagram
5) Activity Diagram
6) Deployment Diagram

4.3.1 CLASS DIAGRAM

UML class diagrams model static class relationships that represent the fundamental architecture of the system. Note that these diagrams describe the relationships between classes, not those between specific objects instantiated from those classes. Thus, the diagram applies to all the objects in the system.

A class diagram consists of the following features:

- **Classes :** These titled boxes represent the classes in the system and contain information about the name of the class, fields, methods and access specifies. Abstract roles of the Class in the system can also be indicated
- **Interfaces :** These titled boxes represent interfaces in the system and contain information about the name of the interface and its methods. Relationship Linesthat model the relationships between classes and interfaces in the system.

- **Dependency** **:** A dotted line with an open arrowhead that shows one entity depends on the behavior of another entity. Typical usages are to represent that one class instantiates another or that it uses the other as an input parameter

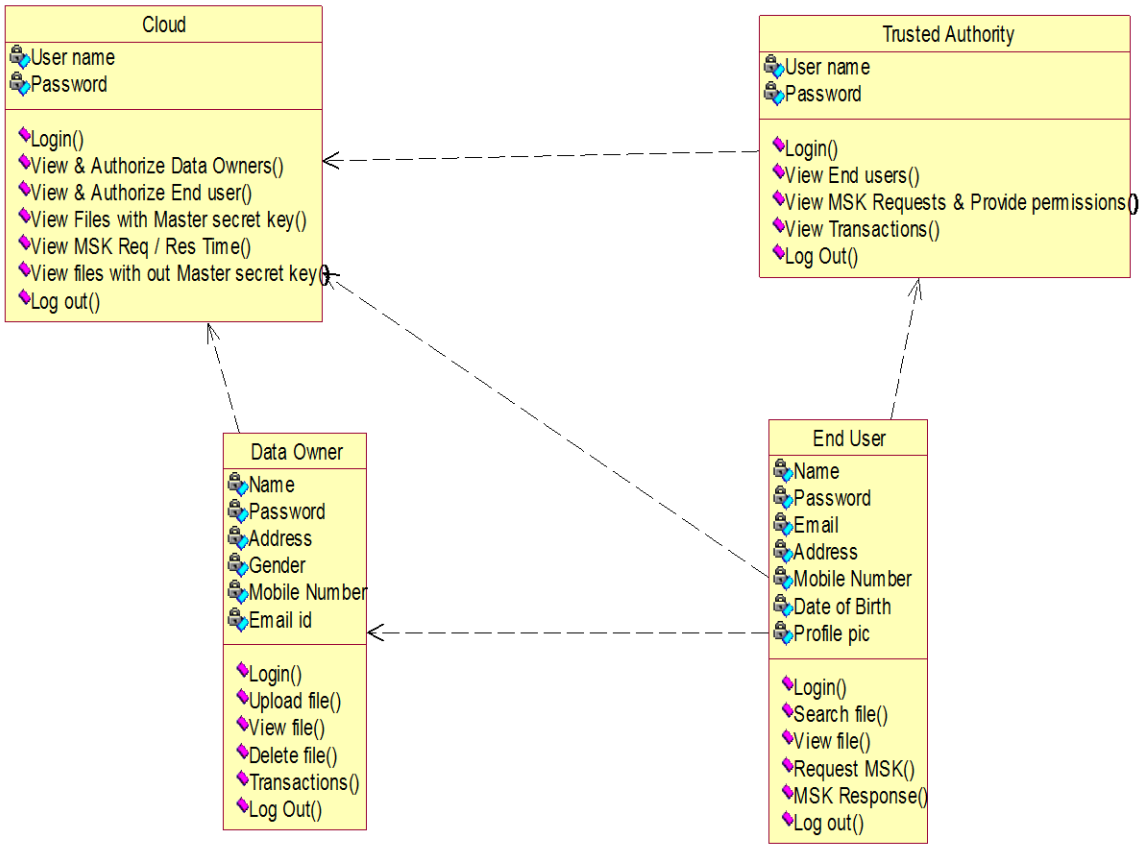


Fig 4.3.1 Class Diagram

4.3.2 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Parts of Use Case Diagram

System boundary boxes

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not

Include

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case".

The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviors from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»". This usage resembles a macro expansion where the included use case behavior is placed inline in the base use case behavior. There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behavior of another, you simply write include followed by the name of use case you want to include, as in the following flow for track order.

Extend

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". The notes or constraints may be associated with this relationship to illustrate the conditions under which this behavior will be executed. Modelers use the «extend» relationship to indicate use cases that are "optional" to the base use case. Depending on the modeler's approach "optional" may mean "potentially not executed with the base use case" or it may mean "not required to achieve the base use case goal".

4.3.2.1 USE CASE DIAGRAM FOR END USER

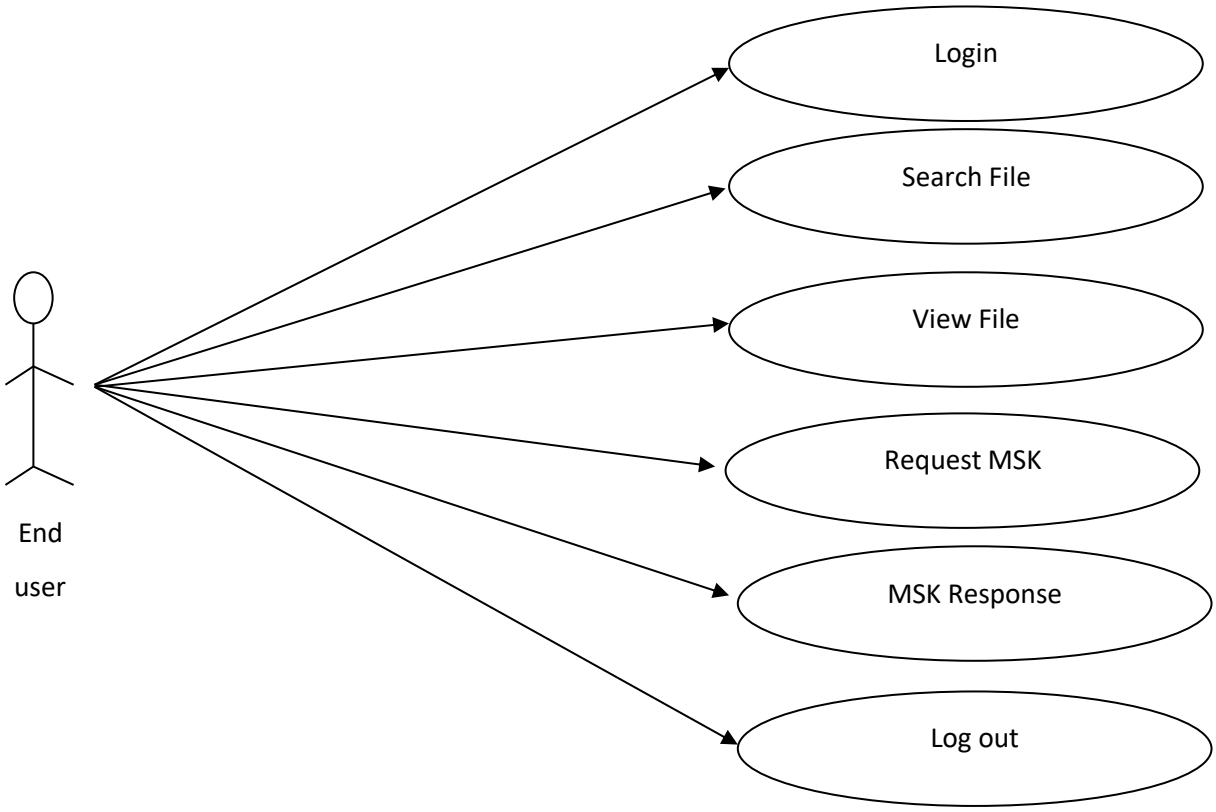


Fig 4.3.2.1 Use Case Diagram for End User

4.3.2.2 USE CASE DIAGRAM FOR DATA OWNER

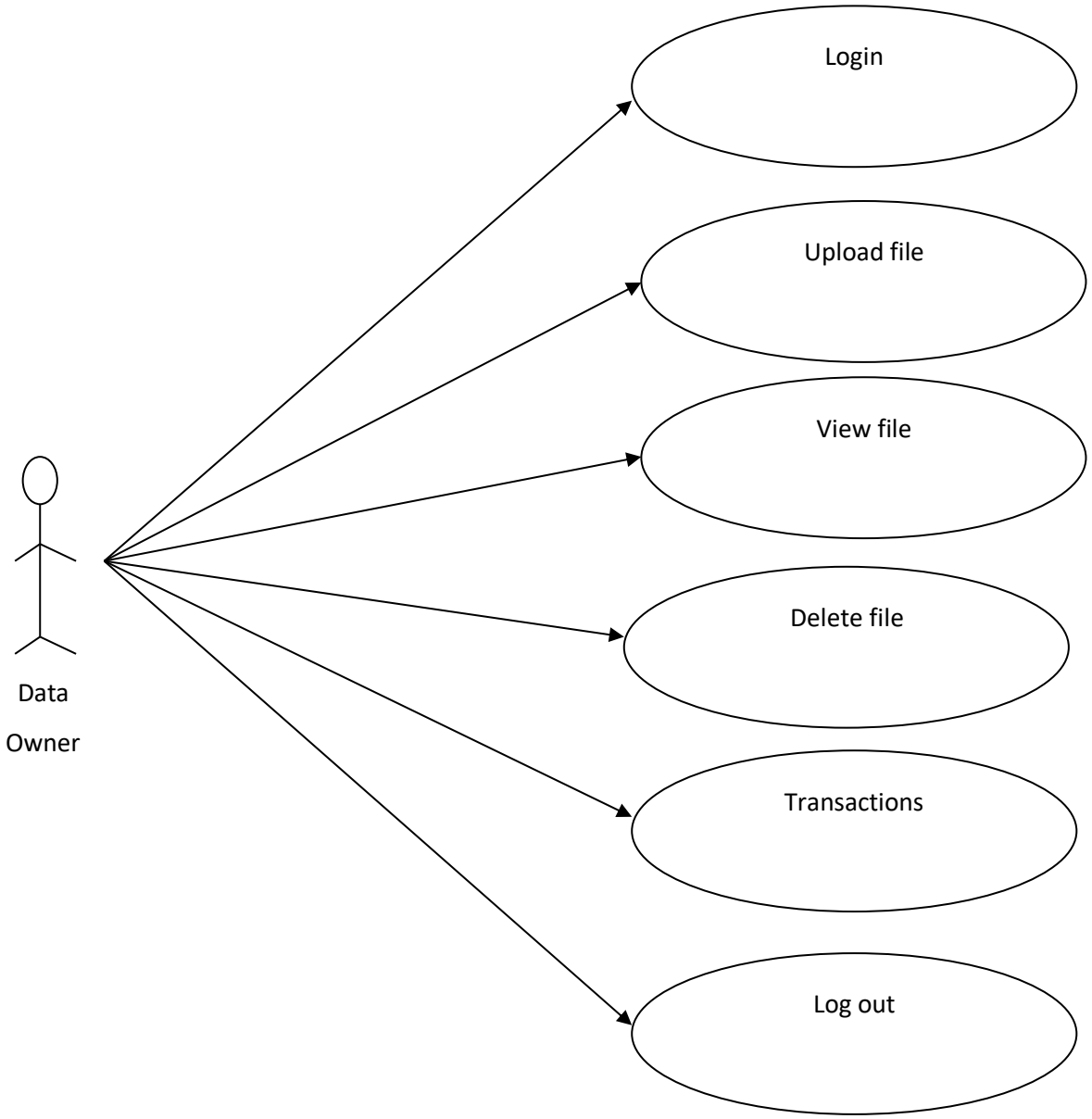


Fig 4.3.2.2 Use case Diagram for data owner

4.3.2.3 USE CASE DIAGRAM FOR TRUSTED AUTHORITY

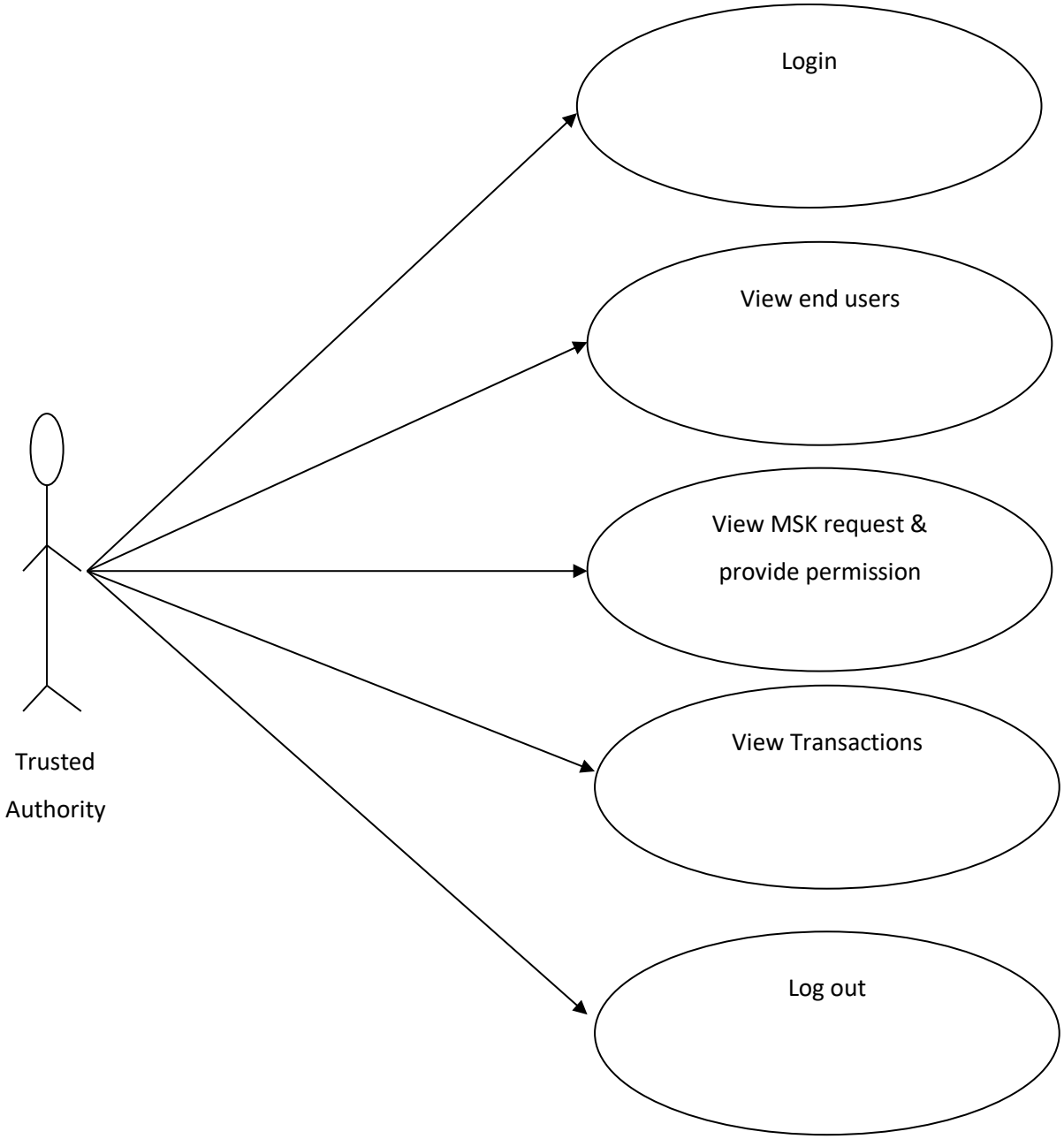


Fig 4.3.2.3 Use Case Diagram for Trusted Authority

4.3.2.4 USE CASE DIAGRAM FOR CLOUD

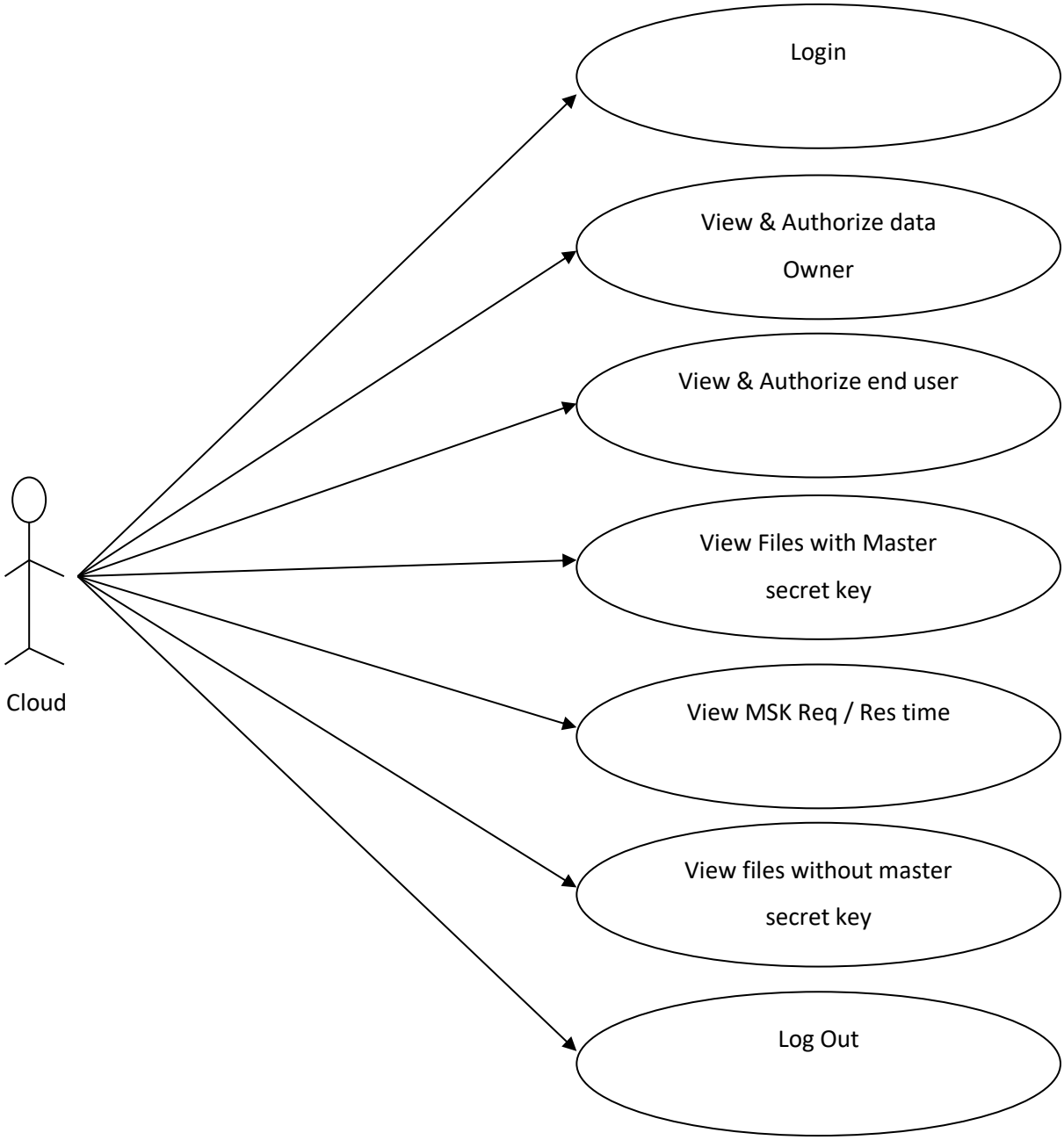


Fig 4.3.2.4 Use Case Diagram for Cloud

4.3.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamicbehavior of a system.

Elements of sequence diagram

The sequence diagram is an element that is used primarily to showcase the interaction that occurs between multiple objects. This interaction will be shown over certain period of time. Because of this, the first symbol that is used is one that symbolizes the object.

Lifeline

A lifeline will generally be generated, and it is a dashed line that sits vertically,and the top will be in the form of a rectangle. This rectangle is used to indicate both theinstance and the class. If the lifeline must be used to denote an object, it will be underlined.

Messages

To showcase an interaction, messages will be used. These messages will come in the form of horizontal arrows, and the messages should be written on top of the arrows. If the arrow has a full head, and it’s solid, it will be called a synchronous call. If the solid arrow has a stick head, it will be an asynchronous call. Stick heads with dash arrows are used to represent return messages.

Objects

Objects will also be given the ability to call methods upon themselves, and theycan add net activation boxes. Because of this, they can communicate with others to show multiple levels of processing. Whenever an object is eradicated or erased from memory, the "X" will be drawn at the lifeline's top, and the dash line will not be drawnbeneath it. This will often occur as a result of a message. If a message is sent from the outside of the diagram, it can be used to define a message that comes from a circle thatis filled in. Within a UML based model, a Super step is a collection of steps which resultfrom outside stimuli.

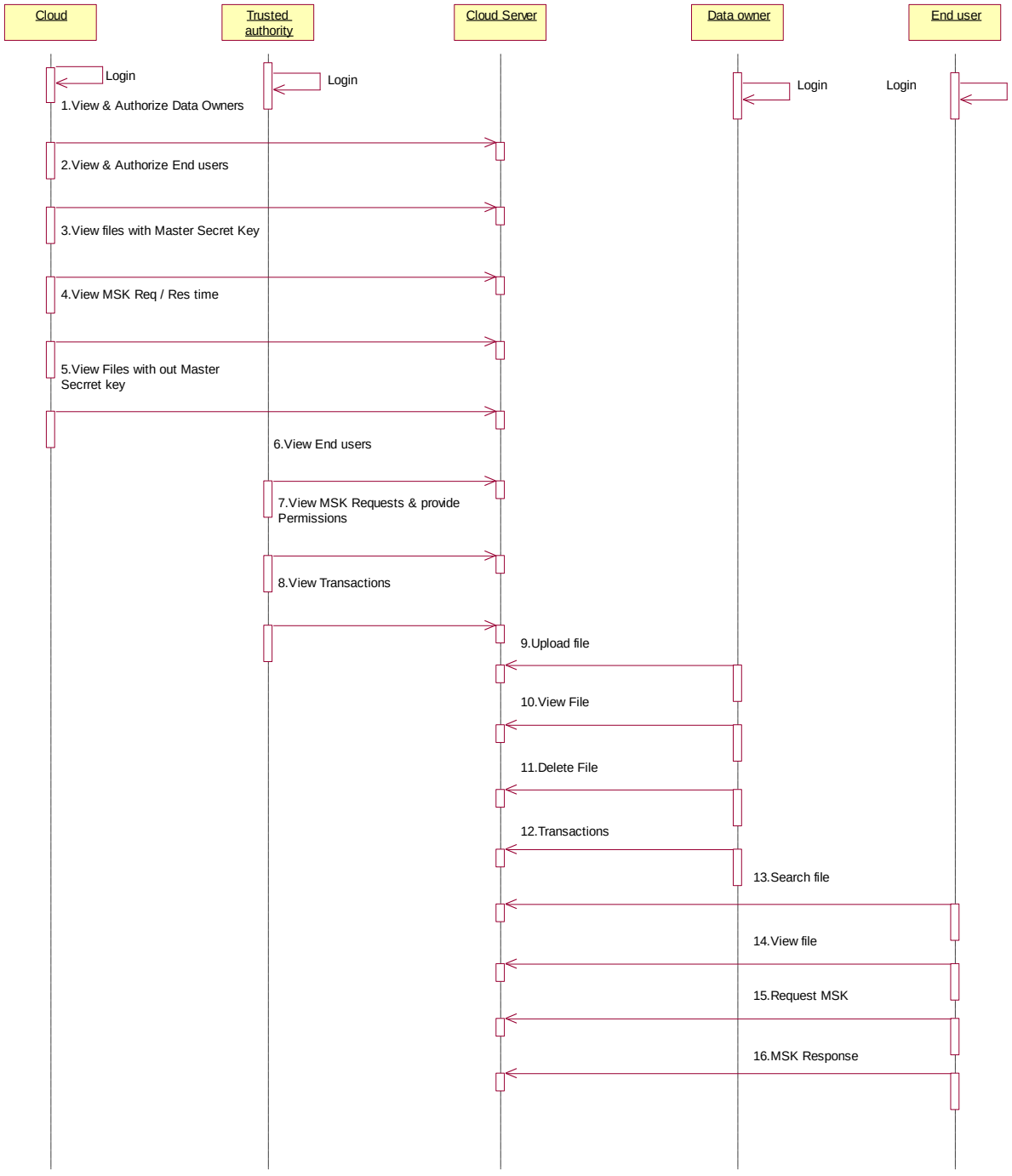


Fig 4.3.3 Sequence Diagram

4.3.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deal with all type of flow control by using different elements like fork, join etc.

How to draw Activity Diagram?

Activity diagrams are mainly used as a flow chart consists of activities performed by the system. But activity diagram is not exactly a flow chart as they havesome additional capabilities. These additional capabilities include branching, parallel flow, swim lane etc. Before drawing an activity diagram, we must have a clearunderstanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions. So before drawing an activity diagram we should identify the following elements.

- Activities
- Association
- Conditions
- Constraints

The following are the basic notational elements that can be used to make up a diagram:

Initial state

An initial state represents a default vertex that is the source for a single transitionto the default state of a composite state. The outgoing transition from the initial vertex may have a behavior, but not a trigger or guard. It is represented by Filled circle, pointing to the initial state.

Final state

A special kind of state signifying that the enclosing region is completed. If the enclosing region is directly contained in a state machine and all other regions in the state machine also are completed, then it means that the entire state machine iscompleted.

Rounded rectangle

It denotes a state. Top of the rectangle contains a name of the state. Can containa horizontal line in the middle, below which the activities that are done in that state areindicated.

Arrow

It denotes transition. The name of the event causing this transition labelsthe arrow body.

4.3.4.1 ACTIVITY DIAGRAM FOR USER

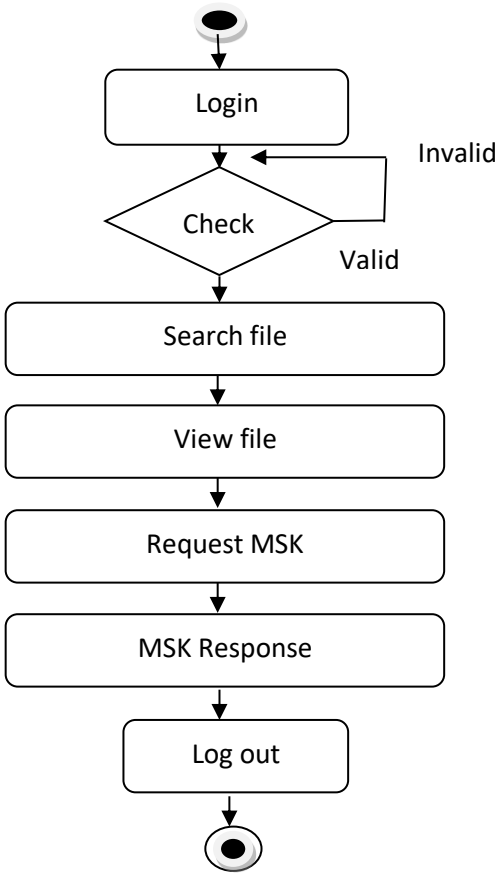


Fig 4.3.4.1 Activity diagram for user

4.3.4.2 ACTIVITY DIAGRAM FOR DATA OWNER

.

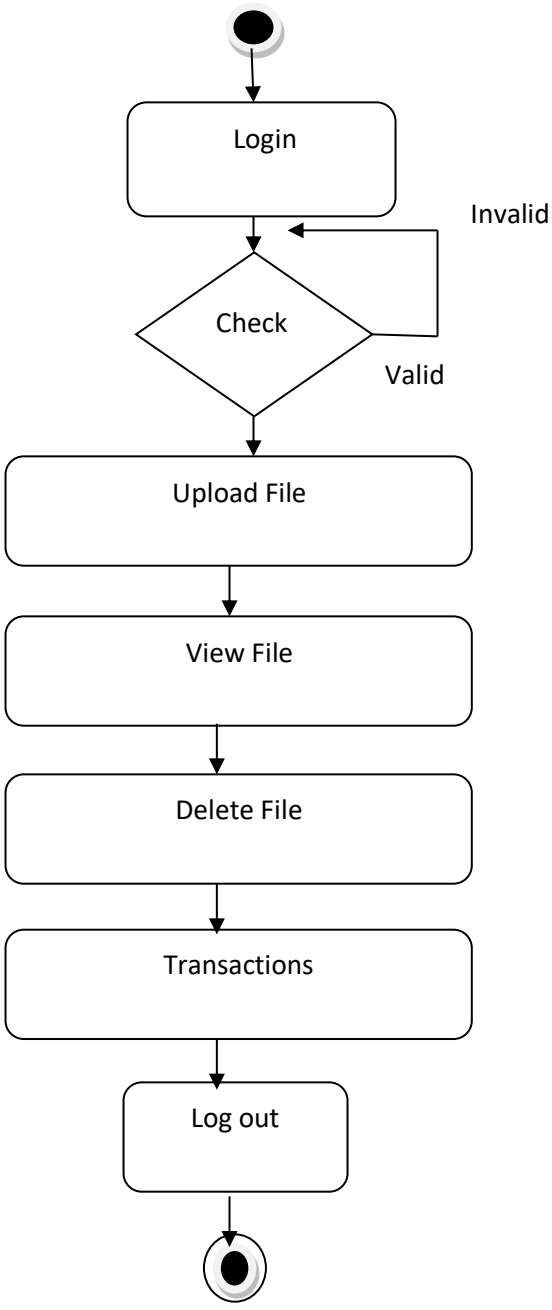


Fig 4.3.4.2 Activity diagram For Data Owner

4.3.4.3 ACTIVITY DIAGRAM FOR CLOUD

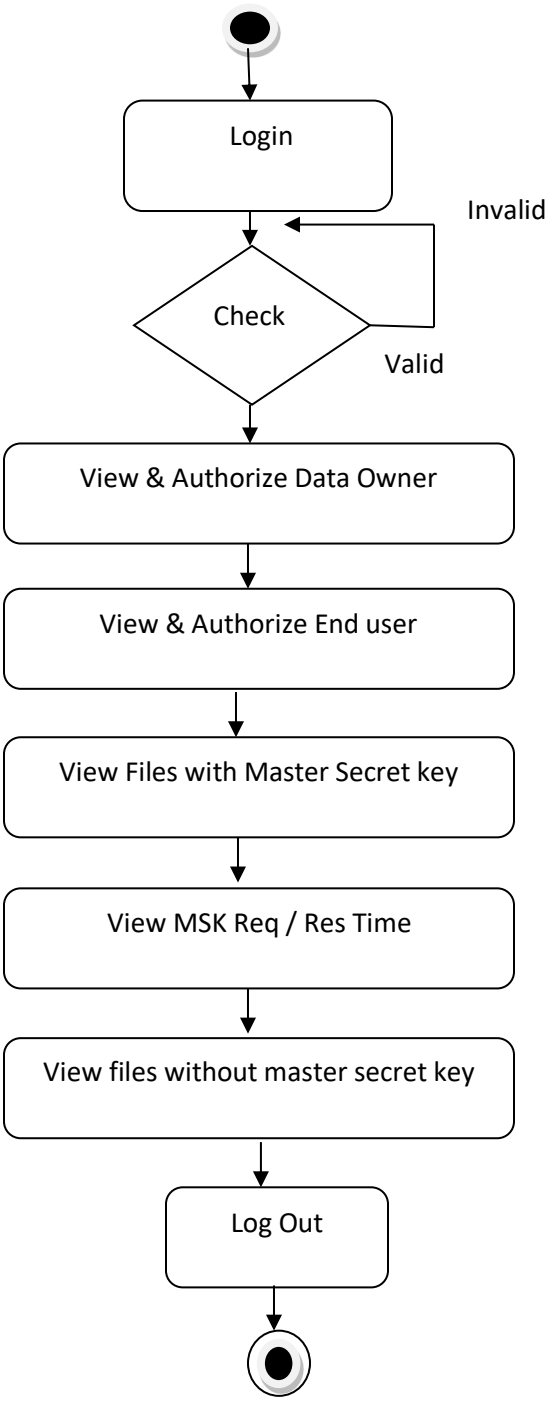


Fig 4.3.4.3 Activity Diagram for Cloud

4.3.4.4 ACTIVITY DIAGRAM FOR TRUSTED AUTHORITY

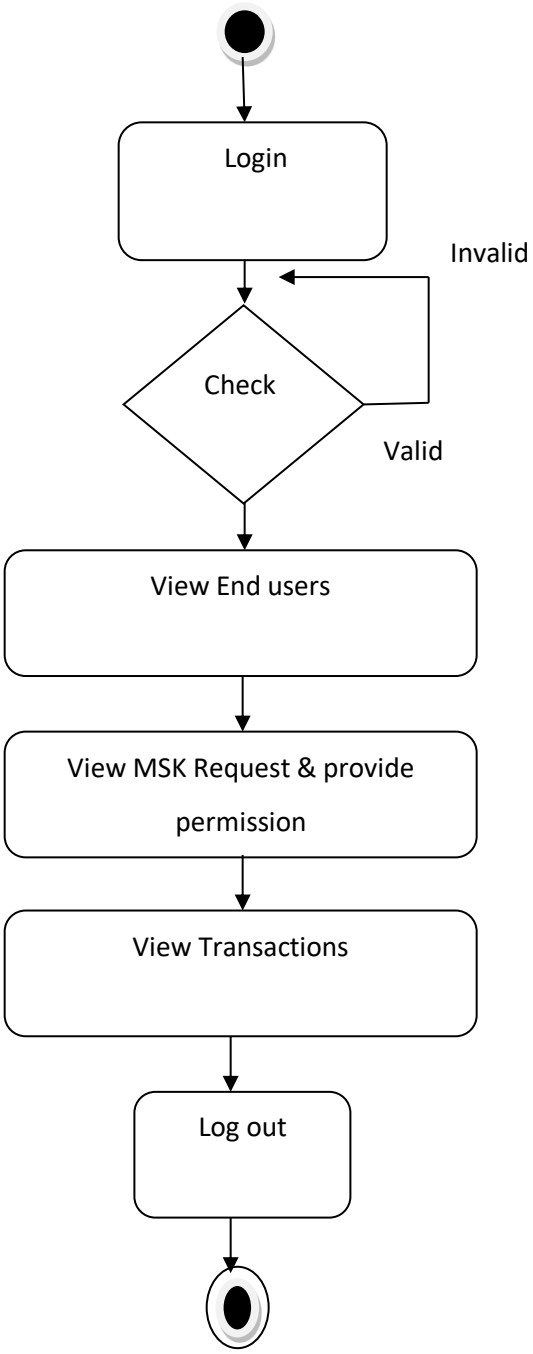


Fig 4.3.4.4 Activity Diagram for Trusted Authority

4.3.5 DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the Component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical Hardware's used to deploy the Applications.

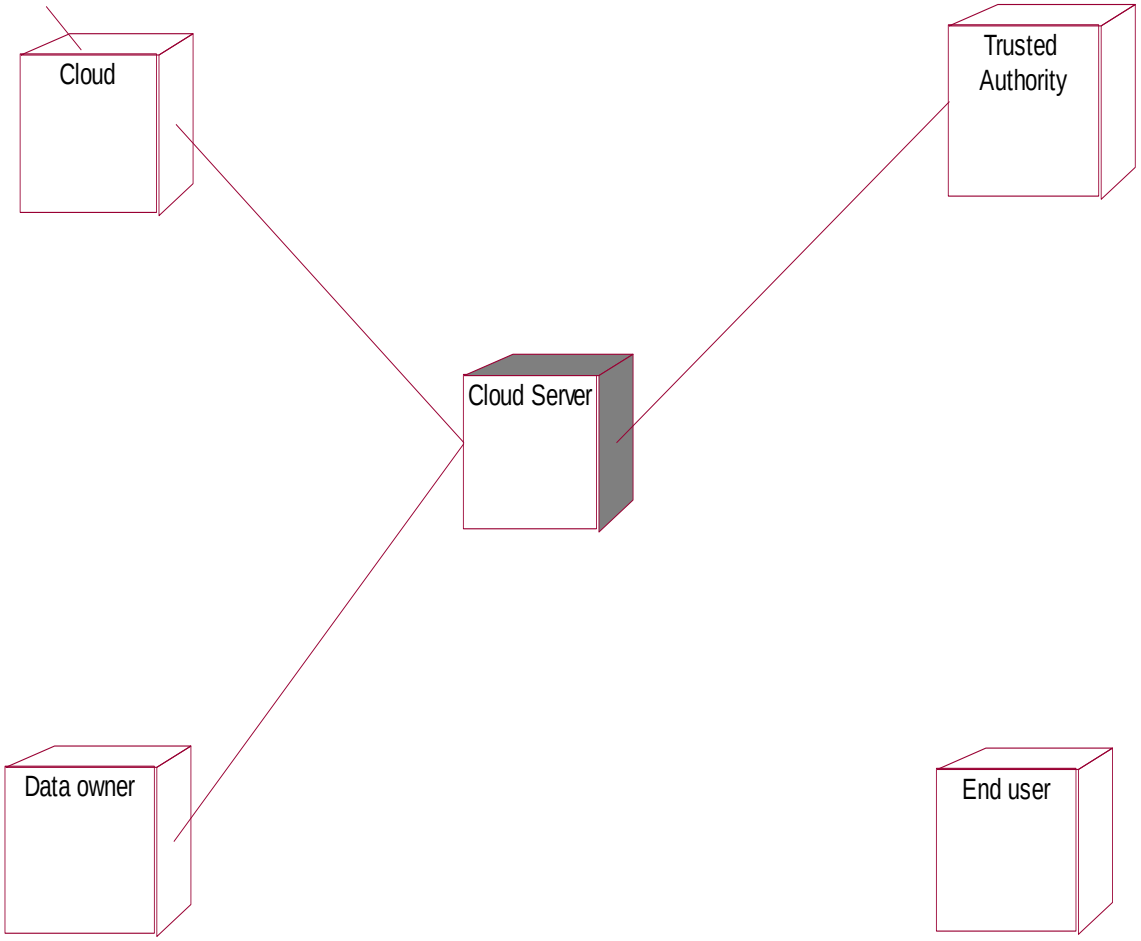


Fig 4.3.5 Deployment Diagram

4.3.6 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

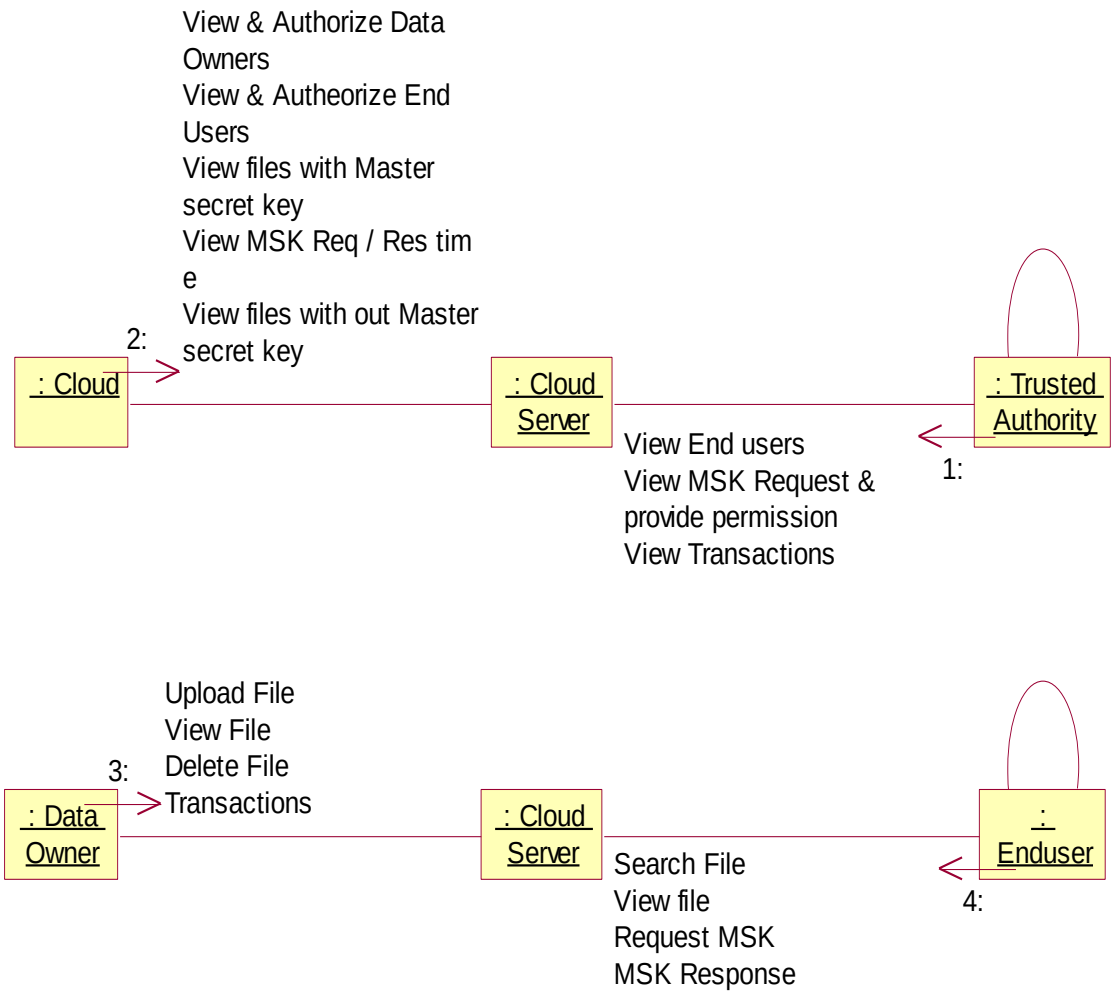


Fig 4.3.6 Collaboration Diagram

CHAPTER-5

SYSTEM IMPLEMENTATION

5.1 SOFTWARE DESCRIPTION

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the Webpages. It is an interpreter, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the Webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB use JavaScript as their scripting and query language.

JavaScript is the full form of **JS.** It is a programming language that is utilized to build an interactive website as it adds functionality to the web pages. It first appeared on **December 4, 1995.** Initially, it was created by **Brendan Eich of Netscape.** Web browsers consist of a separate JavaScript engine which is utilized to execute the JS code. One of the most famous JavaScript runtime environments is **Node.js.** It is a high-level, lightweight, cross-platform, single-threaded and just-in-time compile language that follows **ECMAScript** standard.

JS is the most popular programming in the world right now. It is growing faster than any other language. Big companies like PayPal, Netflix and more build internal applications around JS.

JavaScript Code

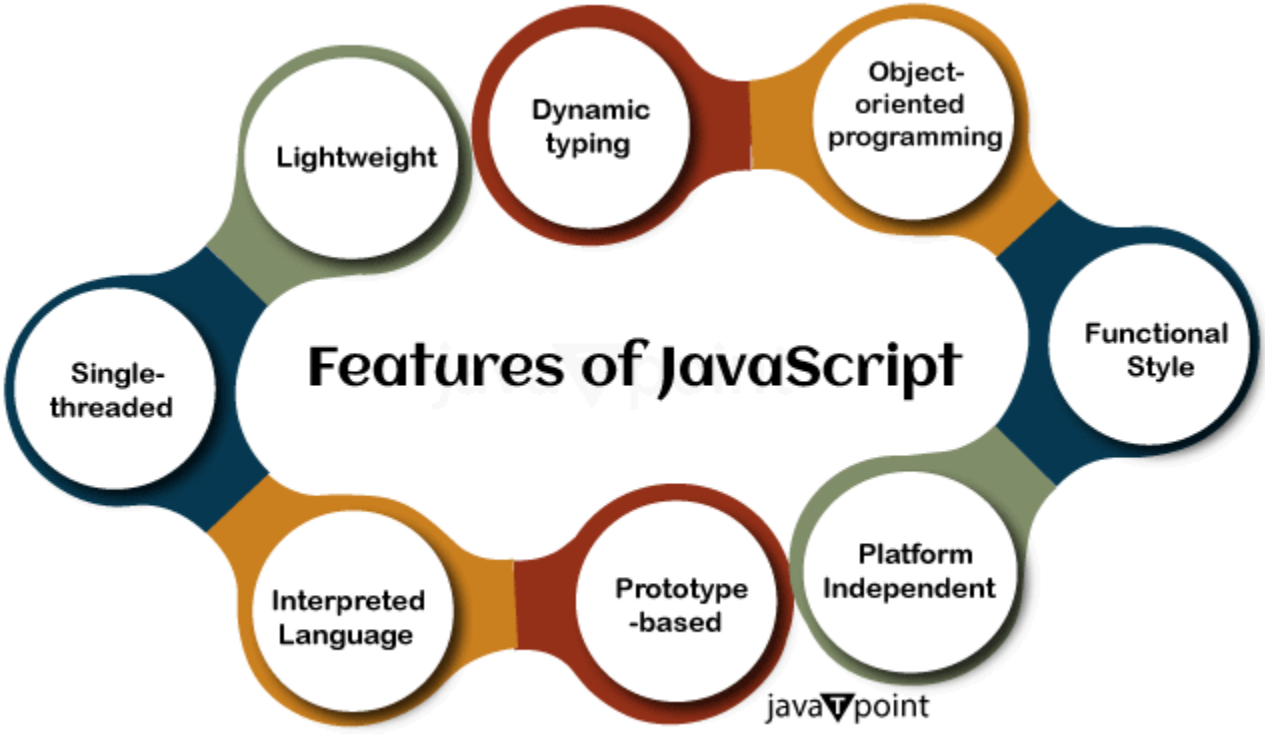
Hey, Execute JavaScript Code!

```
1. console.log ("Hello, World!");
```

Execute JavaScript Code

Features of JavaScript

There are many features of JavaScript which are as follows:



Lightweight

It is a lightweight programming language that is made for handling data on the client side.

Dynamic typing

It is a dynamic typing programming language which means variable types are specified on the basis of the stored value. Some data types are cast implicitly on the basis of the operation utilized.

Object-oriented programming

It supports object-oriented programming. There are two essential principles in JavaScript which are encapsulation and inheritance.

Functional Style

It utilizes a functional style as the objects can be created by utilizing a constructor. In JavaScript, functions are utilized as objects and passed to the other functions also.

Platform Independent

It is a platform independent language which means it is portable and can be run on any operating system. You have to write it once and run it anywhere.

Prototype-based

It is a prototype-based language which means it utilizes prototypes in place of classes. We can specify an object prototype and after that, we can construct more objects with the help of the specified object prototype.

Interpreted Language

It is an interpreted programming language which means it executes the code line by line. A JavaScript interpreter is a built-in feature which is provided by each browser and utilized to interpret the code. At present, an interpreter called just-in-time is utilized by many browsers.

Single-threaded

It supports single threading which means it can only do a single task at a time but JavaScript can implement parallel execution with the help of async processing and web workers.

Async Processing

If there are many functions that need to be executed then utilizing async processing allows us to process functions in parallel that means functions are not executed one by one but can be executed in parallel.

Web Worker

When there is heavy-duty work then web worker is utilized to run tasks in background threads by processing a parallel execution.

Zero-based numbering

It is a zero-index programming language which means the initial element of a sequence is given the index 0 instead of index 1.

Web Development

The main application of JavaScript is web development. JavaScript is utilized to create web pages. It brings web pages to life by creating a dynamic and interactive web pages. Many big companies use this language to build a web page which gives better user experience. There are various websites that are constructed with the help of JavaScript such as Google, Facebook, Yahoo, Twitter, Amazon, Wikipedia, LinkedIn, YouTube, Quora, and more.

Server applications

Server-side applications can be created by utilizing JavaScript. A JS runtime environment called Node.js is utilized to construct server-side applications. Some of the big companies such as PayPal, GoDaddy, etc., are utilizing Node.js for server apps.

Web Applications

Web applications are constructed with the help of JavaScript. The frameworks of JavaScript such as Angular, React, etc., are utilized to build robust web apps.

Game development

JavaScript is a great language for creating games on the web. Both HTML5 and JavaScript are used together to develop games. The library called EaselJS provides rich graphics so it is utilized for graphics. Some of the games created using HTML5 and JS are Rock Paper Scissor, Snake Game, Tic Tac Toe, etc.

Presentations

JavaScript is used to make interactive presentations by utilizing libraries like BespokeJS and RevealJS. The BespokeJS framework is used to add animated bullets, syntax highlighting, and

more. The RevealJS framework is used to create stylish presentations consisting of themes, slide backgrounds, etc.

Mobile App Development

Mobile applications are very popular these days as they are convenient to use. There are various frameworks of JavaScript such as React Native, Apache Cordova, Mobile Angular UI, jQuery Mobile, and more, are used to create mobile applications. Some of the mobile apps created using JS frameworks are Netflix, Candy Crush, Uber, Facebook, etc.

Drawing Applications

HTML in JavaScript provides a feature called the canvas element which is used to make drawings on web pages. This feature has opened a gateway for creating digital art projects. The canvas element allows us to draw freely, provides many colors, and much more.

Smartwatches Applications

Smartwatches are getting popular these days as they act as a substitute for mobiles. JS libraries like Pebble are utilized for creating apps for smartwatches.

Charts and Reports

JavaScript consists of a library called Chart.js which is utilized to create charts and reports. The Chart.js provides various features such as device pixel ratio, data decimation, responsive charts, etc.It supports features like maintaining a high device pixel ratio to ensure sharp graphics, data decimation for handling large datasets efficiently, and responsive charts that adapt seamlessly to different screen sizes.

Frameworks and Libraries of JavaScript

ReactJS

It is a popular JavaScript library which was developed by Meta and community. The original author of ReactJS was Jordan Walke. It was initially released eleven years ago on May 28, 2013. It is a free and open-source front-end JS library that means anybody can utilize this framework without paying a single penny.

It is utilized to create single page or mobile applications. It is utilized by various big companies such as PayPal, Uber, etc.

- **Component-Based Structure**: React promotes reusable components, making development more organized and maintainable.
- **Virtual DOM for Efficiency**: React optimizes updates with a virtual DOM, making rendering fast and smooth.
- **Strong Ecosystem & Community**: Extensive libraries, tools, and support from the community make development easier.
- **Cross-Platform Compatibility**: Works well with React Native if you plan to expand into mobile app development.
- **State Management**: Built-in state handling, and integration with Redux or Zustand helps manage application state efficiently.

jQuery

It is one of the oldest JavaScript frameworks that was developed by the jQuery Team. It was originally authorized by John Resig. It is a free and open-source library which was initially released seventeen years ago on August 26, 2006.

It is utilized to create simple web applications. It helps in simplifying HTML DOM elements, Ajax, CSS animations, and event handling.

Vue.js

It is an open-source JavaScript framework which means anybody can use it for free. It was originally authorized by Evan You and was initially released ten years ago in February 2014.

It is utilized to make user interfaces and single-page applications.

AngularJS

It is a robust, free and open-source JavaScript-based web framework that was developed by Google. It was originally authorized by Miško Hevery and was initially released thirteen years ago on October 20, 2010. It is utilized in simplifying development and testing applications. It is utilized as the front-end of the MEAN stack.

Express

It is a fast and open-source back-end web application framework that was developed by TJ Holowaychuk, StrongLoop and others. It was initially released thirteen years ago on November 16, 2010.

It is utilized to create complex web applications and APIs. It has become the prime choice when using the MEAN stack.

Ember.js

It is an open-source JS web framework that was developed by Ember Core Team. It was originally authorized by Yehuda Katz. It was initially released twelve years ago on December 8, 2011.

It is utilized to construct scalable SPAs. It is used on various websites such as Nordstrom, HashiCrop, Apple Music, Live Nation, Twitch, Intercom, Ghost, Square, etc. This framework has the capability to create desktop and mobile applications.

5.2 HARDWARE REQUIREMENTS

RAM

Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off. When the computer is rebooted, the OS and other files are reloaded into RAM, usually from an HDD or SSD.

Because of its volatility, RAM can't store permanent data. RAM can be compared to a person's short-term memory, and a hard disk drive to a person's long-term memory. Short-term memory is focused on immediate work, but it can only keep a limited number of facts in view at any one time. When a person's short-term memory fills up, it can be refreshed with facts stored in the brain's long-term memory.

A computer also works this way. If RAM fills up, the computer's processor must repeatedly go to the hard disk to overlay the old data in RAM with new data. This process slows the computer's operation.

HARD DISK

Hard disk, also called hard disk drive or hard drive. A computer’s hard drive is a device consisting of several hard disks, read/write heads, a drive motor to spin the disks, and a small amount of circuitry, all sealed in a metal case to protect the disks from dust. In addition to referring to the disks themselves, the term hard disk is also used to refer to the whole of a computer’s internal data storage. Beginning in the early 21st century, some personal computers and laptops were produced that used solid-state drives (SSDs) that relied on flash memory chips instead of hard disks to store information

MOUSE

A computer mouse is a hand-held pointing device that detects two-dimensional motion relative to a surface. This motion is typically translated into the motion of a pointer on a display, which allows a smooth control of the graphical user interface of a computer.

KEYBOARD

A keyboard is for putting information including letters, words and numbers into your computer. You press the individual buttons on the keyboard when you type. The number keys across the top of the keyboard are also found on the right of the keyboard.

5.3 SOURCE CODE

Login,jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<!--
<!-- CONTENT -->
<h3>Cloud Login </h3>
<form action="C_Authentication.jsp" method="post" id="leavereply">
<p>
<input name="imageField" type="submit"  class="LOGIN" id="imageField" value="Login" />
<input type="reset" name="imageField" id="imageField"  class="RESET" />
</p>
<p>  </p>
</form>
<p> </p>
</div>
<h4> </h4>
</div>
</div>
<div class="col-2">
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="R_Login.jsp">End User</a></li>
<li><a href="E_Login.jsp">Data Owner</a></li>
<li><a href="TA_Login.jsp">T-Authority</a></li>
<li><a href="C_Login.jsp">Cloud</a></li>
</ul>
</div>
</div>
</div>
</div>
<script type="text/javascript">Cufon.now(); </script>
<div align=center></div>
</body>
</html>
```

AuthorizeDataOwner.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
<head>
<h2 class="style1">Dual Access Control for Cloud-Based Data Storage and Sharing<br />
</h2>
<ul>
<li class="m5"><a href="C_Main.jsp" class="active">Cloud</a></li>
<li class="m4"><a href="C_Login.jsp">Logout</a></li>
</ul>
<!-- CONTENT -->
<div id="content">
<h3>Authorize Data Owner </h3>
href="C_EncrypterrDetails.jsp?name=<%=s2%>"><%=s2%></a></div></td>
<%
if(s5.equalsIgnoreCase("Waiting"))
{
%>
<td><div>
<div align="center"><a href="C_StatusOwner.jsp?id=<%=i%>"><%=s5%></a></div>
</div></td>
<%
}
connection.close();
}
<ul>
<li><a href="C_Main.jsp">Home</a></li>
<li><a href="C_Attackers.jsp">Attackers</a></li>
<li><a href="C_Transactions.jsp">Transactions</a></li>
<li><a href="C_FileWithMSK.jsp">Files With MSK </a></li>
<li><a href="C_MSKTime.jsp">MSK Req/Res Time</a></li>
<li><a href="C_FileWithoutMSK.jsp">Files Without MSK </a></li>
<li><a href="C_AuthorizeEndUser.jsp">Authorize End User </a></li>
<li><a href="C_ViewResults.jsp">Rank Results</a></li>
<li><a href="C_Login.jsp">Logout</a></li>
</ul>
</body>
</html>
```

AuthorizeEndUser.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<head>
<title>Cloud </title>
<h2 class="style1">Dual Access Control for Cloud-Based <br />
Data Storage and Sharing</h2>
<ul>
<li class="m5"><a href="C_Main.jsp" class="active">Cloud</a></li>
<li class="m4"><a href="C_Login.jsp">Logout</a></li>
</ul>
<!-- CONTENT -->
try
{
String query="select * from receiver ";
Statement st=connection.createStatement();
ResultSetrs=st.executeQuery(query);
while ( rs.next() )
{
i=rs.getInt(1);
s2=rs.getString(2);
s5=rs.getString(12);
%>
</tr>
<tr>
}
connection.close();
}
<ul>
<li><a href="C_Main.jsp">Home</a></li>
<li><a href="C_Attackers.jsp">Attackers</a></li>
<li><a href="C_Transactions.jsp">Transactions</a></li>
<li><a href="C_FileWithMSK.jsp">Files With MSK </a></li>
<li><a href="C_MSKTime.jsp">MSK Req/Res Time</a></li>
<li><a href="C_FileWithoutMSK.jsp">Files Without MSK </a></li>
<li><a href="C_AuthorizeDataOwner.jsp">Authorize Data Owner </a></li>
<li><a href="C_ViewResults.jsp">Rank Results</a></li>
```

```
<li><a href="C_Login.jsp">Logout</a></li>
</ul>
<script type="text/javascript">Cufon.now(); </script>
</body>
</html>
```

Register.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<ul>
<li class="m1"><a href="index.html" >Home</a></li>
<li class="m3"><a href="E_Login.jsp" class="active">Data Owner</a></li>
<li class="m4"><a href="TA_Login.jsp">T-Authority</a></li>
<li class="m5"><a href="C_Login.jsp">Cloud</a></li>
</ul>
<!-- CONTENT -->
<label for="name">Data Owner Name (required)<br />
</label>
<p>
<input id="name" name="userid" class="text" />
</p>
<label for="password">Password (required)<br />
</label>
<p>
<input type="password" id="password" name="pass" class="text" />
</p>
<p>
<label for="email">Email Address (required)<br />
</label>
<input id="email" name="email" class="text" />
</p>
<label for="mobile">Mobile Number (required)<br />
</label>
<p>
<input id="mobile" name="mobile" class="text" />
</p>
<label for="address">Your Address<br />
</label>
```

```
<p>
<textarea name="address" cols="50" id="address"></textarea>
</p>
<label for="dob">Date of Birth (required)<br />
</label>
<p>
<input id="dob" name="dob" class="text" />
</p>
<label for="gender">Select Gender (required)<br />
</label>
<p>
<select id="s1" name="gender" class="text">
<option>-Select-</option>
<option>Male</option>
<option>Female</option>
</select>
</p>
<label for="pincode">Enter Pincode (required)<br />
</label>
<p>
<input id="pincode" name="pincode" class="text" />
</p>
<label for="location">Enter Location (required)<br />
</label>
<p>
<input id="loc" name="location" class="text" />
</p>
<label for="pic">Select Profile Picture (required)<br />
</label>
<p>
<input type="file" id="pic" name="pic" class="text" />
</p>
<p>
<input name="submit" type="submit" value="REGISTER" />
</p>
</form>
<p align="justify"> </p>
</body>
</html>
```

RegisterAuthentication.jsp

```
<title>Registration authen</title>
<%@page
import="com.oreilly.servlet.*,java.sql.*,java.lang.*,java.text.SimpleDateFormat,java.util.*,java.i
o.*,javax.servlet.*,javax.servlet.http.*" %>
while (params.hasMoreElements())
{
paramname = (String) params.nextElement();
if(paramname.equalsIgnoreCase("userid"))
{
uname=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("pass"))
{
pass=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("email"))
{
email=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("mobile"))
{
mno=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("address"))
{
addr=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("dob"))
{
dob=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("gender"))
{
gender=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("pincode"))
{
pincode=multi.getParameter(paramname);
```

```
}
if(paramname.equalsIgnoreCase("location"))
{
location=multi.getParameter(paramname);
}
if(paramname.equalsIgnoreCase("pic"))
{
image=multi.getParameter(paramname);
}
out.println("Data Owner Name Already Exits");
%>
```

Download.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<%@page import ="java.util.*"%>
<%@page import ="java.sql.*"%>
<%@page
import="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>
<%@
import="java.sql.*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileInputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.BufferedInputStream" %>
<%@
import="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>
<%@
import="java.util.*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStream,java.io.PrintStream"%>
<%@ include file="connect.jsp" %>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>End User Main</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script src="js/cufon-yui.js" type="text/javascript"></script>
<script src="js/cufon-replace.js" type="text/javascript"></script>
<script src="js/Myriad_Pro_300.font.js" type="text/javascript"></script>
```

```
<!--[if lt IE 7]>
<script type="text/javascript" src="js/ie_png.js"></script>
<script type="text/javascript">ie_png.fix('.png, #header .row-2 ul li a, #content, .list li');</script>
<![endif]-->
<style type="text/css">
<!--
.style1 {
font-size: 37px;
color: #CCCCCC;
}
.style2 {color: #20b7c9}
.style3 {font-size: 12px}
.style13 {font-size: 14px; color: #3f3f3f; font-weight: bold; }
-->
Data Storage and Sharing</h2>
</div>
<div class="fright"></div>
</div>
<div class="row-2">
<ul>
<li class="m2"><a href="R_Main.jsp"class="active">End User</a></li>
<li class="m3"><a href="R_Login.jsp">Logout</a></li>
</ul>
</div>
<div class="row-3"><imgsrc="images/slogan.gif" alt="" />
<form action="#" method="post" id="search-form">
<fieldset>
<div><span>
<input type="text" value="Enter keyword here" onfocus="if(this.value=='Enter keyword here'){this.value="}" onblur="if(this.value=="){this.value='Enter keyword here'}" />
</span><a href="#"><imgsrc="images/button.gif" alt="" /></a></div>
</fieldset>
</form>
</div>
</div>
<!-- CONTENT -->
<div id="content">
<div class="inner_copy">More <a href="#">Website Templates</a> @ Templates.com!</div>
<div class="tail-right">
<div class="wrapper">
```

```
<div class="col-1">
<div class="indent">
<div class="indent1">
<h3>Download File   </h3>
<p align="justify"> </p>
<form action="R_Download1.jsp" method="post" id="form1">
<p> </p>
<table width="524" border="0" style="border-collapse:collapse" cellpadding="0" cellspacing="0" align="center">
<tr>
<td width="223" height="37"><span class="style13">Enter File Name :</span></td>
<td width="245"><label>
<input required="required" name="t1" type="text" size="40" />
</label></td>
</tr>
<tr>
<td height="38"><span class="style13">Trapdoor :</span></td>
<td><input name="t12" type="text" size="40" /></td>
</tr>
<tr>
<td height="31"><span class="style13">Secret Key :</span></td>
<td><input name="t13" type="text" size="40" /></td>
</tr>
<tr>
<td height="43"><div align="right">
<input type="submit" name="Submit" value="Req Trapdoor" />
</div></td>
<td><input type="submit" name="Submit2" value="Download" /></td>
</tr>
</table>

</form>
<p align="justify"> </p>
</div>
<h4> </h4>
</div>
</div>
<div class="col-2">
<ul>
<li><a href="R_Main.jsp">Home</a></li>
```

```
<li><a href="R_Search.jsp">Search</a></li>
<li><a href="R_Download.jsp">Download</a></li>
<li><a href="R_ViewFiles.jsp">View Files</a></li>
<li><a href="R_ReqMSK.jsp">Request MSK </a></li>
<li><a href="R_MSKRes.jsp">MSK Response </a></li>
<li><a href="R_Login.jsp">Logout</a></li>
</ul>
</div>
</div>
</div>
</div>
<!-- FOOTER -->
<div id="footer">
<div class="indent">
<div class="fleft"></div>
<div class="fright"></div>
</div>
</div>
</div>
</div>
<script type="text/javascript">Cufon.now(); </script>
<div align=center></div>
</body>
</html>
```

CHAPTER-6

SYSTEM TESTING

6.1 SOFTWARE TESTING

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

6.2 TYPES OF TESTS

6.2.1 UNIT TESTING

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. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

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

Functional testing is centered on the following items:

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

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

Functions **:** identified functions must be exercised.

Output **:** identified classes of application outputs must be exercised.

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

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive

Processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2.4 SYSTEM TESTING

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

6.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 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 testing 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.

6.3 UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.3.1 TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

6.3.2 FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.4 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.5 ACCEPTANCE TESTING

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

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.6 TEST CASES

Test Case-1 : Service provider Home Page

Expected Output : Page is successful and Admin main menu is displayed

Actual Output : When Service provider username and Password is entered into login page. On Successful credentials, Service provider home page is displayed and main menu is displayed

Test Case-2 : User Home Page

Expected Output : User Home Page is successful and User main menu is displayed

Actual Output : When User's username and Password is entered into login page. On Successful credentials, home page is displayed and main menu is displayed in user home page

CHAPTER-7

EXPERIMENTAL RESULTS

7.1 EXECUTION PROCEDURE

The Execution procedure is as follows :

1. In this research work with data with attributes are observable and then all of them are floating data. And there's a decision class/class variable. This data was collected from Kaggle machine learning repository.
2. In this research 70% data use for train model and 30% data use for testing purpose.
3. Diffie Hellmann is used as Classifier .
4. In the classification report we were able to find out the desired result
5. In this analysis the result depends on some part of this research. However, which algorithm gives the best true positive, false positive, true negative, and false negative are the best algorithms in this analysis.

7.2 OUTPUT SCREENSHOTS

Fig 7.2.1 Home Page

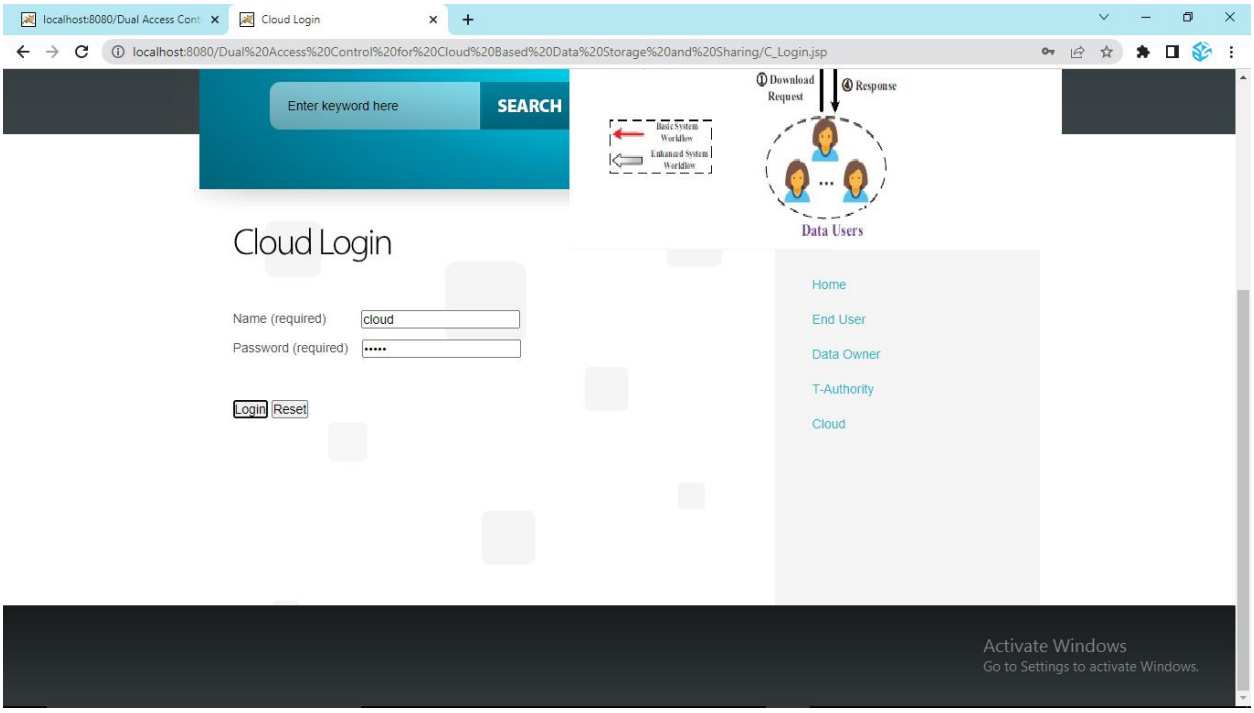


Fig7.2.2 Cloud Login

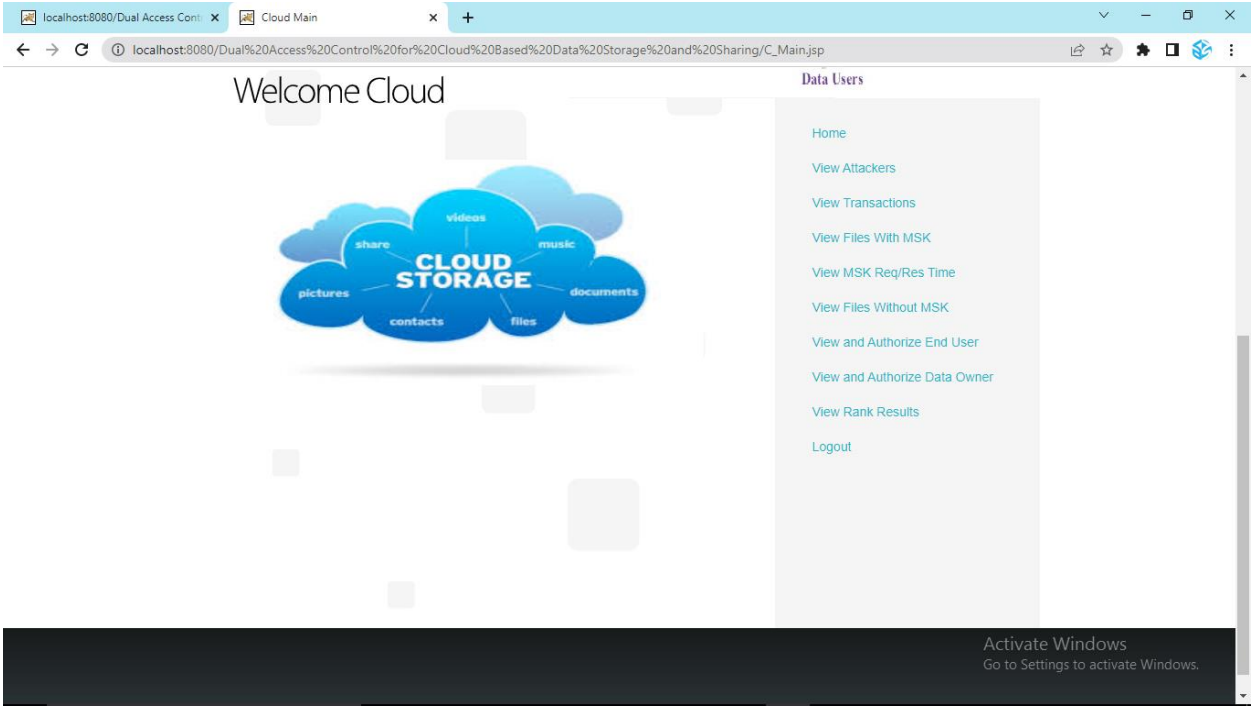


Fig 7.2.3 Cloud Main

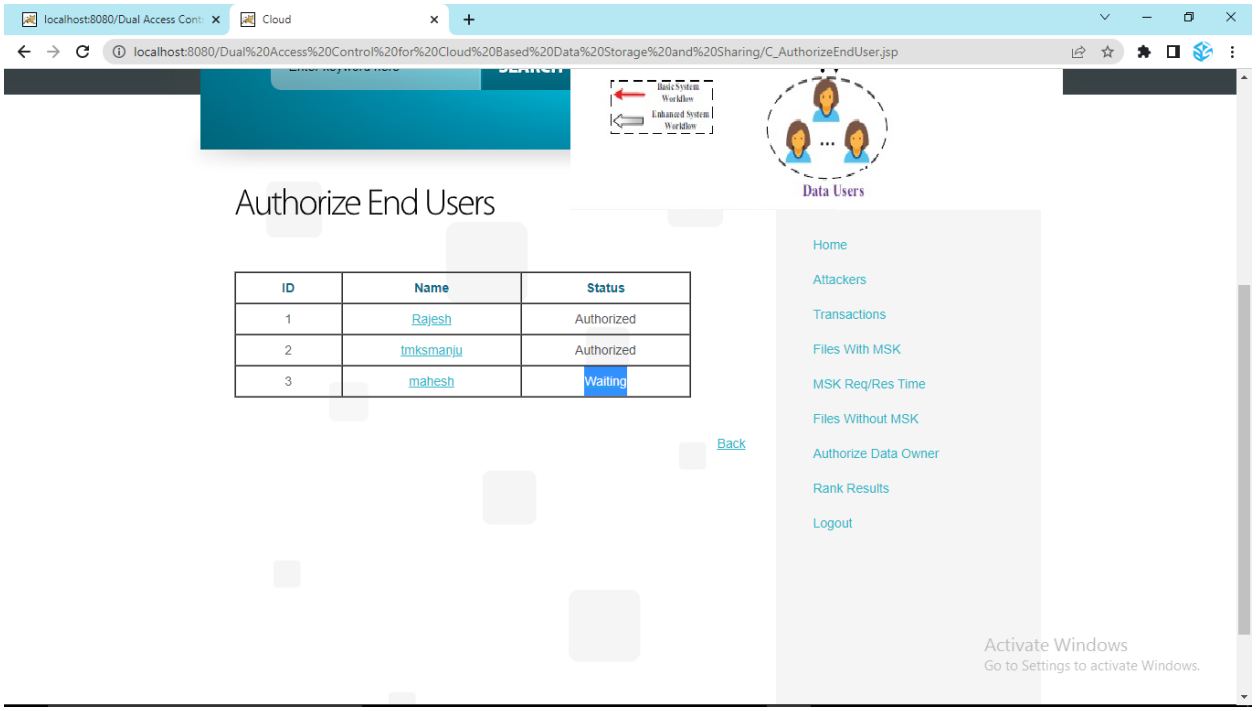


Fig 7.2.4 All Authorize Users

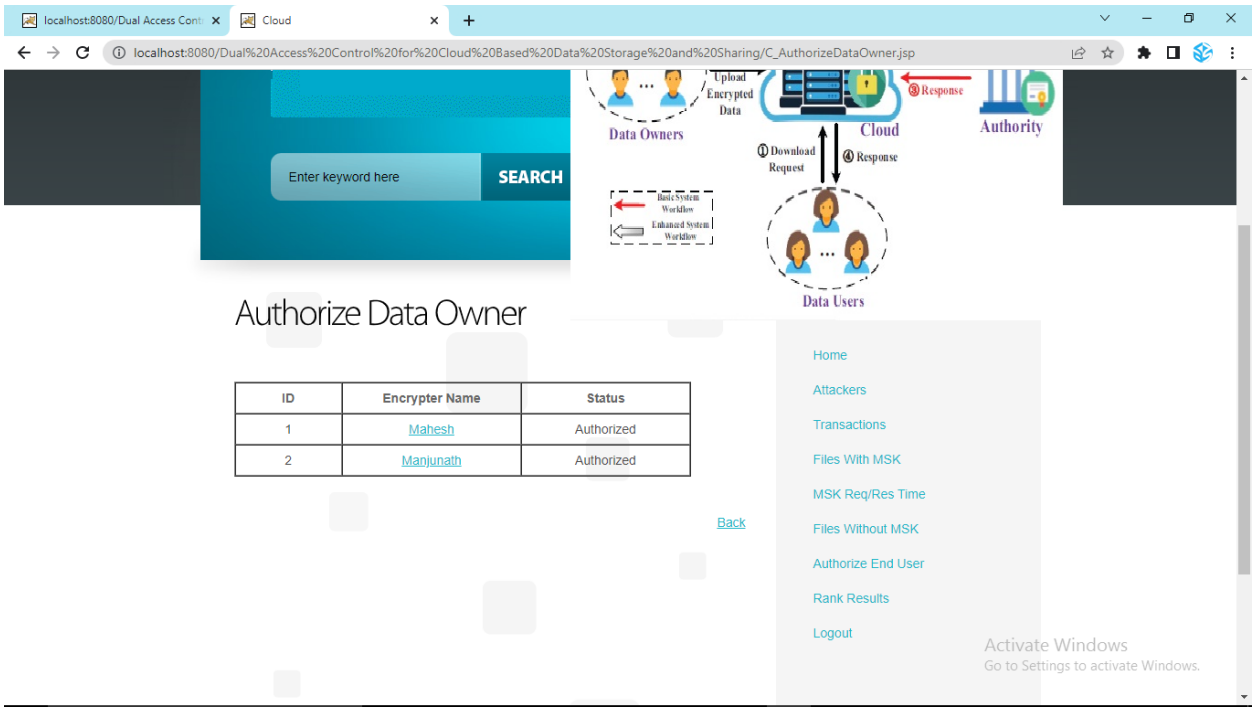


Fig 7.2.5 All Authorize Data Owner

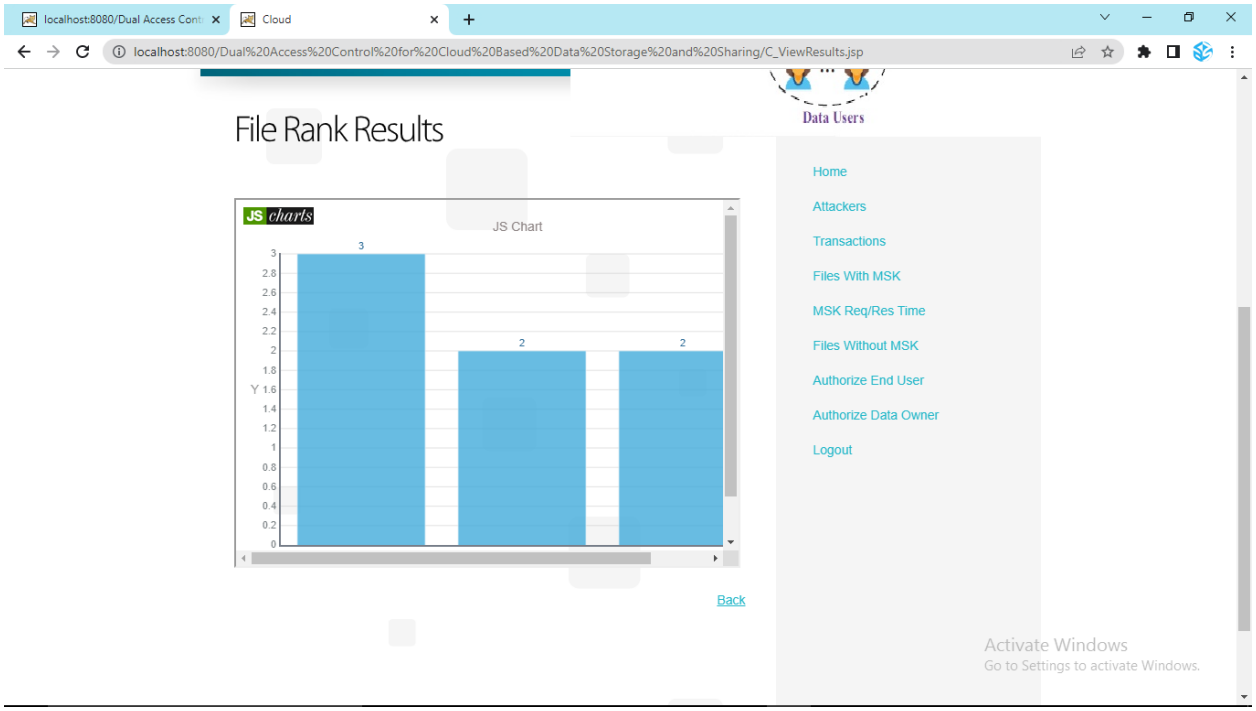


Fig.7.2.6 Rank Result

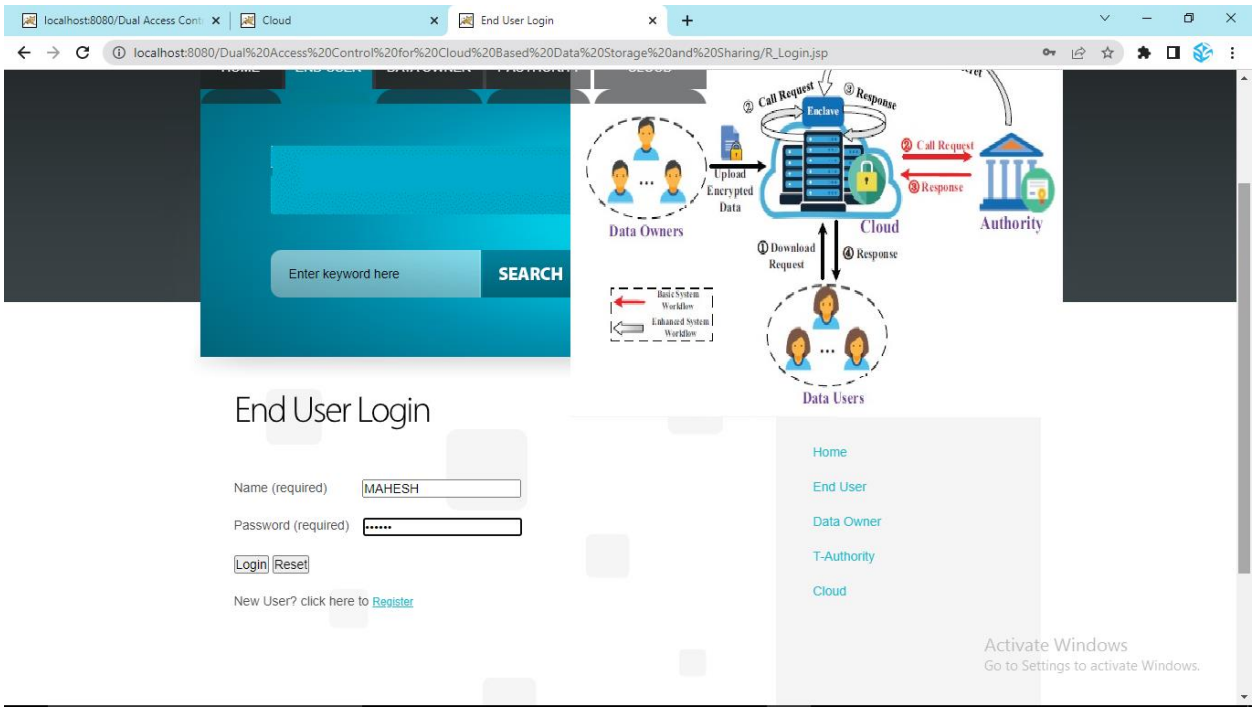


Fig.7.2.7 End User Login

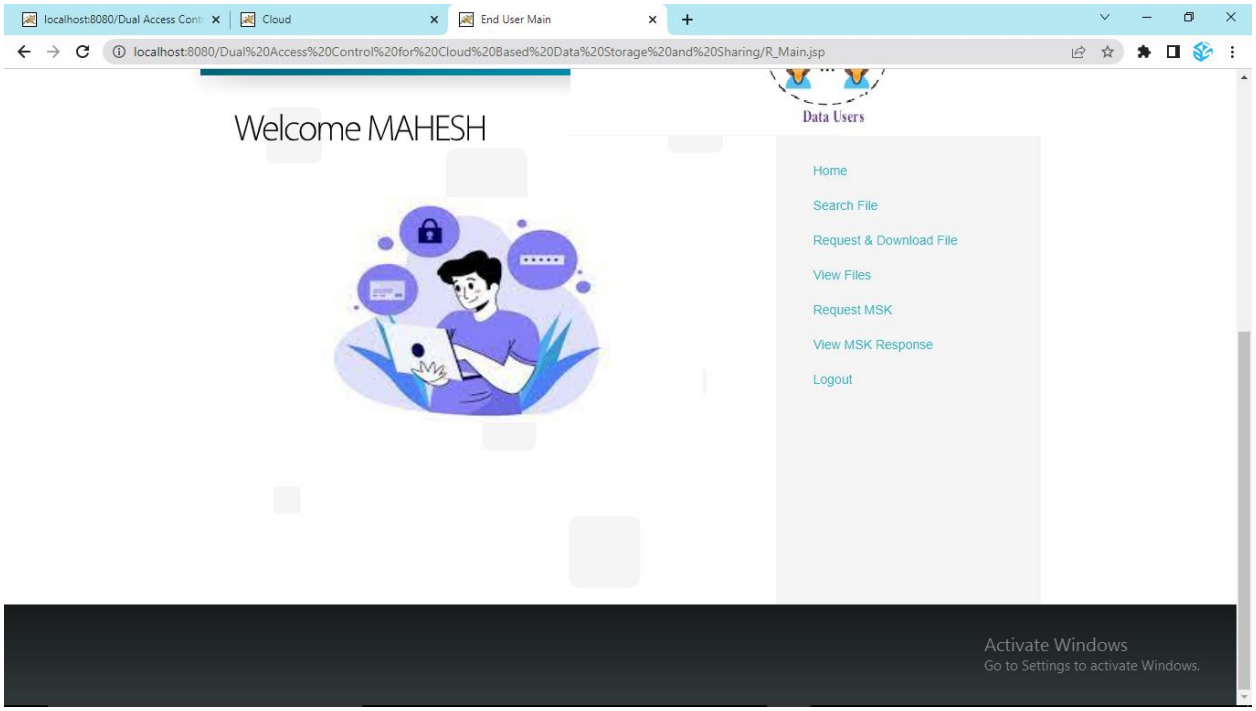


Fig.7.2.8 End Users Home Page

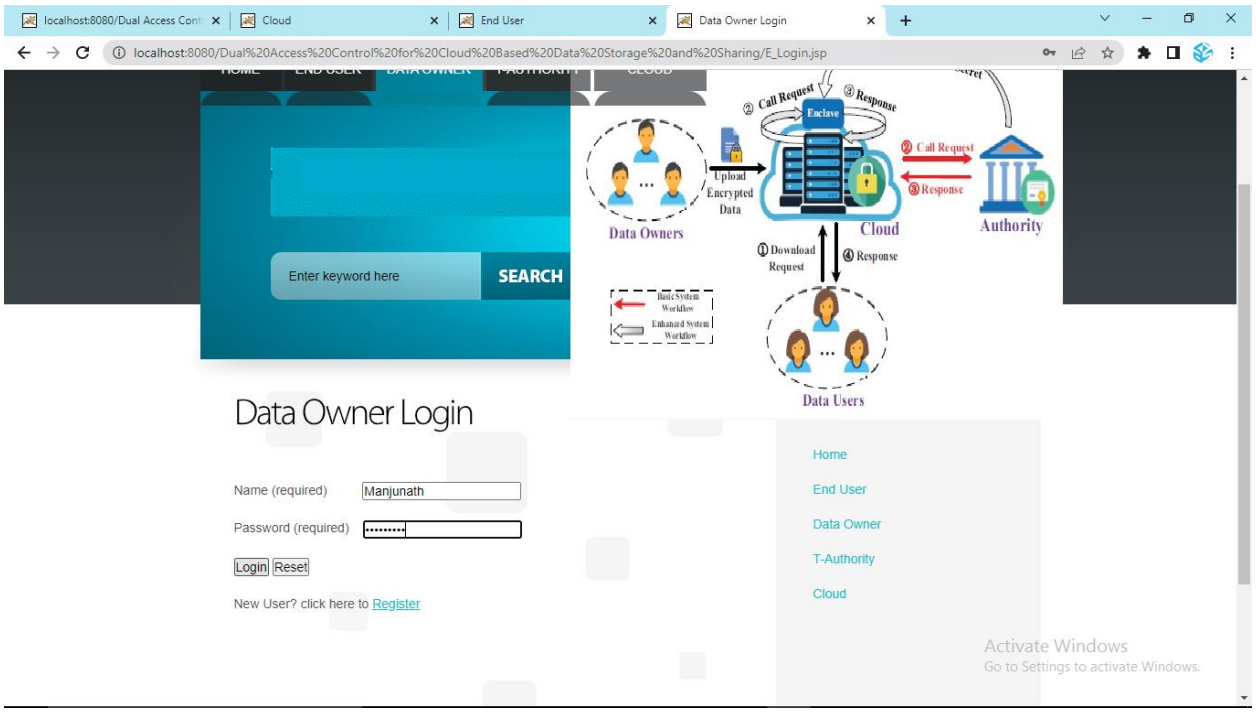


Fig.7.2.9 Data Owner Login

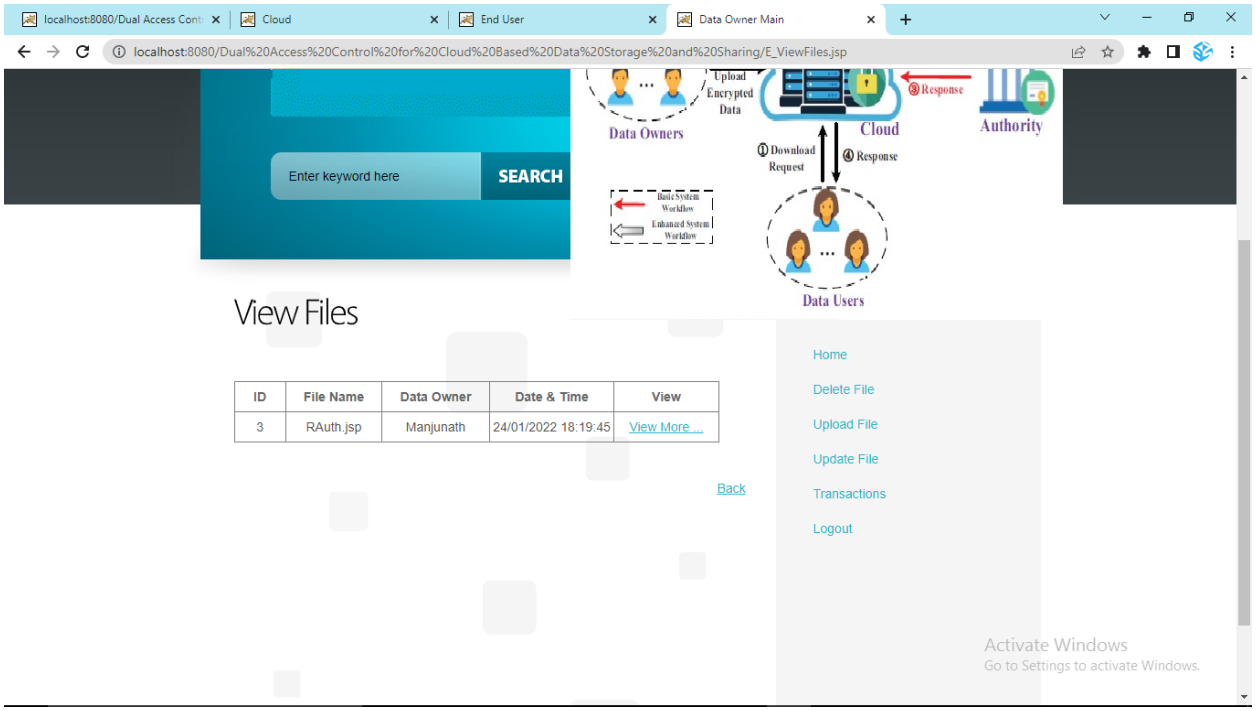


Fig.7.2.10 View All Files

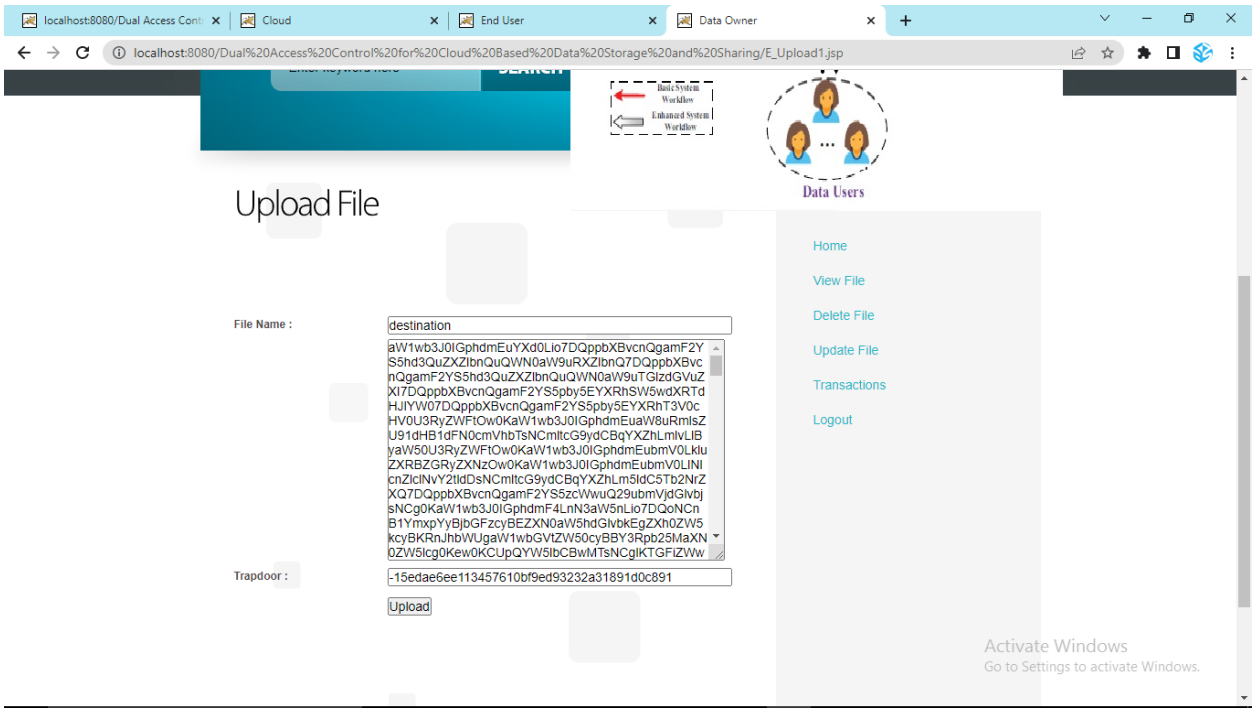


Fig.7.2.11 Upload Files

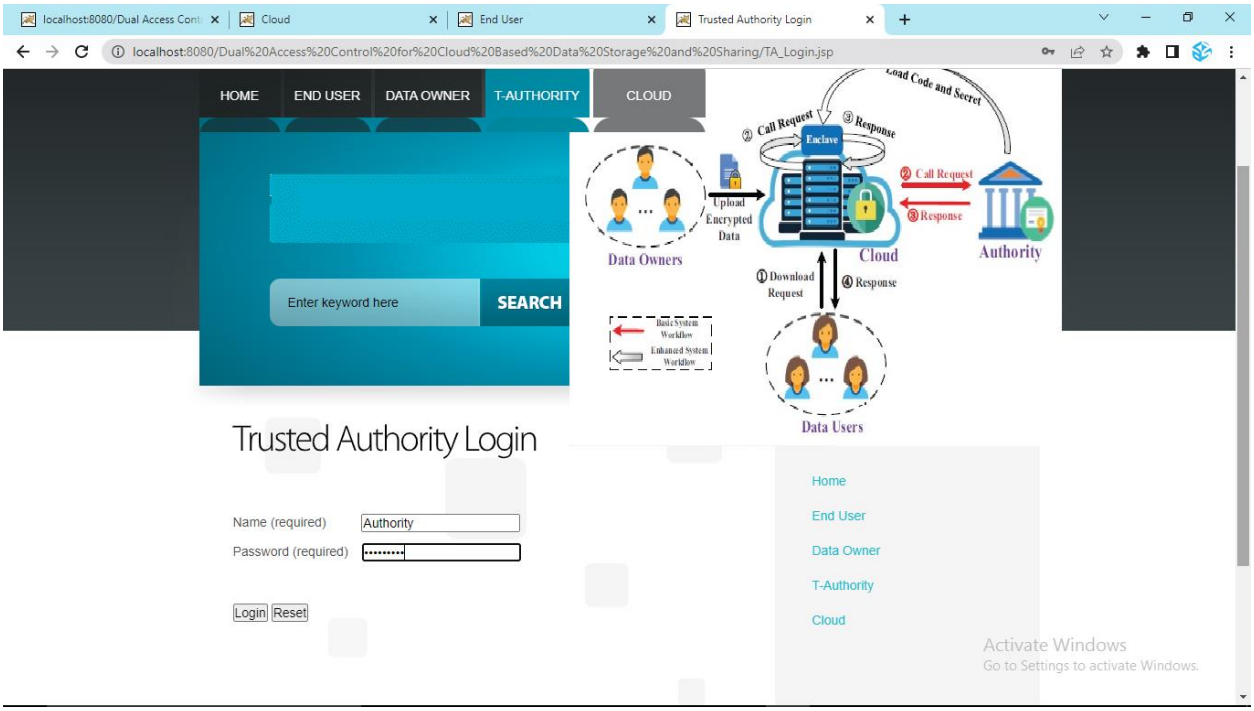


Fig.7.2.12 Trusted Authority Login

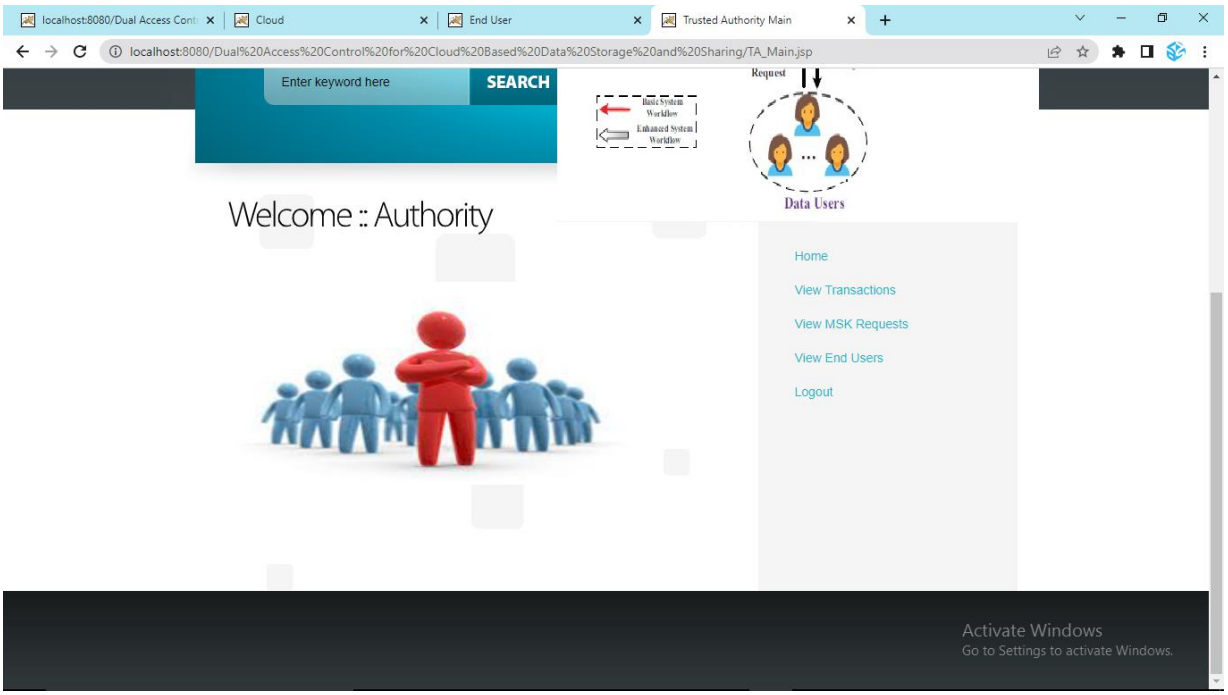


Fig.7.2.13 Trusted Authority Main Page

CHAPTER-8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

We addressed an interesting and long-lasting problem in cloud-based data sharing, and presented two dual access control systems. The proposed systems are resistant to DDoS/EDoS attacks. We state that the technique used to achieve the feature of control on download request is "transplantable" to other CP-ABE constructions. Our experimental results show that the proposed systems do not impose any significant computational and communication overhead (compared to its underlying CP-ABE building block). In our enhanced system, we employ the fact that the secret information loaded into the enclave cannot be extracted. However, recent work shows that enclave may leak some amounts of its secrets to a malicious host through the memory access patterns or other related side-channel attacks. The model of transparent enclave execution is hence introduced in. Constructing a dual access control system for cloud data sharing from transparent enclave is an interesting problem. In our future work, we will consider the corresponding solution to the problem.

8.2 FUTURE SCOPE

To improve performance, the system could adopt **edge computing** and **distributed search indexing**, reducing search latency by processing queries closer to users rather relying solely on centralized cloud servers. Furthermore, optimizing **Secure Multi-Party Computation (SMPC)** could enable collaborative yet privacy-preserving searches across multiple cloud storage providers

REFERENCES

[1] Joseph A Akinyele, Christina Garman, Ian Miers, Matthew WPagano, Michael Rushanan, Matthew Green, and Aviel D Rubin.Charm: A framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3(2):111–128, 2013.

[2] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata: Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.

[3] Alexandros Bakas and Antonis Michalas. Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm2019, pages 472–486, 2019.

[4] Amos Beimel: Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[5] John Bethencourt, Amit Sahai, and BrentWaters: Ciphertext-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE,2007.

[6] Victor Costan and Srinivas Devadas: Intel sgx explained. IACRCryptology ePrint Archive, 2016(086):1–118, 2016.

[7] Ben Fisch, Dhinakaran Vinayagamurthy, Dan Boneh, and SergeyGorbunov. IRON:Functional encryption using intel SGX. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 765–782, 2017.