1.INTRODUCTION

1.1 purpose:

The popularity of cloud computing is obtained from the benefits of rich storage resources and instant access. It aggregates the resources of computing infra- structure, and then provides on-demand services over the Internet. Many famous companies are now providing public cloud services, such as Amazon, Google, Alibaba. These services allow individual users and enterprise us- ers to upload data (e.g. photos, videos and documents) to cloud service provider (CSP), for the purpose of accessing the data at any time anywhere and sharing the data with others. In order to protect the privacy of users, most cloud services achieve access control by maintaining access con- trol list (ACL). In this way, users can choose to either publish their data to anyone or grant access rights merely to their approved people. However, the security risks have raised concerns in people, due to the data is stored in plaintext form by the CSP. Once the data is posted to the CSP, it is out of the data owner's control. Unfortunately, the CSP is usually a semi-trusted server which honestly follows the designated protocol, but might col- lect the users' data and even use them for benefits with- out users' consents. On the other hand, the data has tre- mendous usages by varous data consumers to learn the behavior of users.

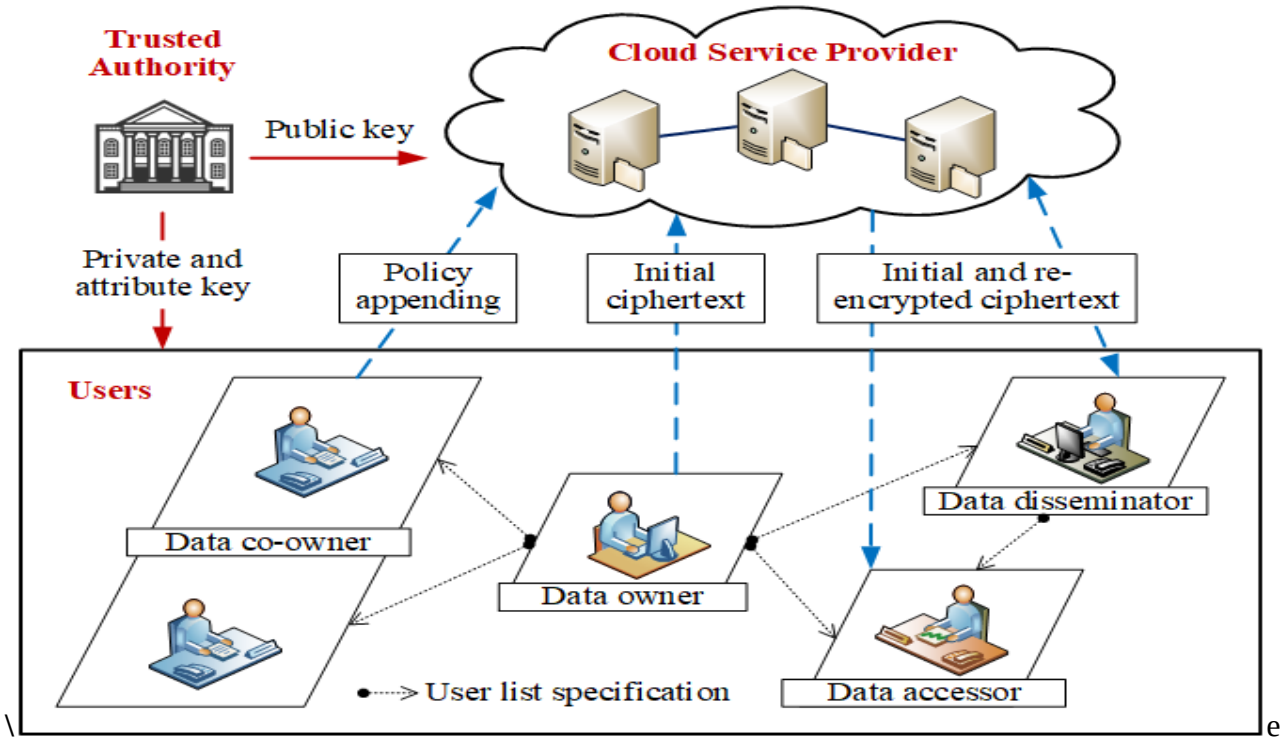
These security issues motivate the effective solutions to protect data confidentiality. It is essential to adopt access control mechanisms to achieve secure data sharing in cloud computing. Currently, cryptographic mechanisms such as attribute-based encryption (ABE), identity-based broadcast encryption (IBBE), and remote attestation have been exploited to settle these security and privacy problems. ABE is one of the new crypto- graphic mechanisms used in cloud computing to reach secure and fine-grained data sharing. It features a mechanism that enables an access control ove r encrypted data using access policies and ascribed attributes among decryption keys and ciphertexts. As long as the attribute set satisfies the access policy that the ciphertext can be decrypted. IBBE is another prevalent technique employed in cloud computing, in which users could share their encrypted data with multiple receivers at one time and the public key of the receiver can be regarded as any valid strings, such as unique identity and email. In fact, IBBE can be seen as a special case of ABE for policies con- sisting of an OR gate. Compared to ABE in which the se- cret key and ciphertext are both correspond to a set of attributes, IBBE incurs low-cost key management and small constant policy sizes, which is more suitable for securely broadcasting data to specific receivers in cloud computing. Hence, by using identities, data owner can share data with a group of users in a secure and efficient manner, which motivates more users to share their private data via cloud.

Actually, these encryption techniques can prevent un- authorized entities (e.g. semi-trusted CSP and malicious users) from accessing the data, but it may not consider data dissemination in cloud computing. In the cloud collaboration scenario such as Box and OneDrive, the data disseminators (e.g. editor and collaborator) may share the documents with new users even those outside the organization. However, once the data is encrypted with the above techniques, data disseminators are not able to modify the ciphertext uploaded by data owners. Proxy re-encryption (PRE) scheme is employed to achieve secure data dissemination in cloud computing by delegating a re-encryption key associated with the new receivers to the CSP. However, the data disseminator can disseminate all of the data owner's data to others with this re-encryption key, A refined concept referred to as conditional PRE (CPRE) could address this issue, in which data owner can enforce re-encryption control over the initial ciphertexts and only the ciphertexts satisfying specific condition can be re-encrypted with corresponding re- encryption key. However, traditional CPRE

schemes only support conditions rather than keywords, at- tribute-based CPRE is proposed, which deploys an access policy in the ciphertext. The re-encryption key is associated with a set of attributes, thus the proxy can re- encrypt the ciphertext only when the re-encryption key matches the access policy. In this way, data owner can customize fine-grained dissemination condition for the shared data. For example, data owner allows project managers in the organization to disseminate the progress report in OneDrive, while only permits executive directors in finance department to disseminate the project budget in OneDrive during a specific time period.

1.2 Scope

We achieve fine-grained conditional dissemination over the ciphertext in cloud computing with attribute-based CPRE. The ciphertext is firstly deployed with an initial access policy customized by data owner. Our pro- posed multiparty access control mechanism allows the data co-owners to append new access policies to the ci- phertext due to their privacy preferences. Hence, the ci- phertext can be re-encrypted by sthe data disseminator only if th



\ e

2.Literature Survey

te-Based Encryption Scheme with Constant Ciphertext Length

An Attribute-Based Encryption (ABE) is an encryption scheme, where users with some attributes can decrypt ciphertexts associated with these attributes. The length of the ciphertext depends on the number of attributes in previous ABE schemes. In this paper, we propose a new Ciphertext-Policy Attribute-Based Encryp- tion (CP-ABE) with constant ciphertext length. In our scheme, the number of pairing computations is also constant. In addition, the number of additional bits required from CPA-secure CP-ABE to CCA-secure CP-ABE is reduced by 90% with respect to that of the previous scheme.

A user identity (such as the name, e-mail address and so on) can be used for accessing control of some resources. For example, in Identity-Based Encryption (IBE) schemes such as [9, 12], an encryptor can restrict a decryptor to in- dicate the identity of the decryptor. An Attribute-Based Encryption (ABE) is an encryption scheme, where users with some attributes can decrypt the ciphertext associ- ated with these attributes. Although IBE schemes have a restriction such that an encryptor only indicates a sin- gle decryptor, in ABE schemes, an encryptor can indicate many decryptors by assigning common attributes of these decryptors such as gender, age, affiliation andso on. There are two kinds of ABE, Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). KP-ABE [18, 27] are schemes such that each private key is associated with an access structure. CP-ABE [5, 15, 17, 25, 32] areschemessuch that each ciphertext is associated with an access structure. An application of KP-ABE is for a biometric system. If an IBE scheme is used to construct the biometric system, then a user's information (such as a finger-print, iris data and so on) is registered as the identity of the user. How- ever, these values are somewhat changed since they depend on a user's condition, on humidity and so on. Therefore, the user is forced to manage secret keys corresponding to all identities. KP-ABE schemes with threshold structures can solve this problem to indicate a threshold value as an error-tolerant value. An application of CP-ABE is for an encrypted storage system. If 1 data is encrypted by using 1 encryption key, then the total number of encryption and decryption keys increases. If plural data are encrypted by using one encryption key, then a fine-grained access con- trol is not achieved. To indicate the set of attributes of a decryptor such as affiliation, the CP-ABE scheme can achieve a fine-grained access controlwithout increasing the number of keys. There are some extended ABE schemes such as ABE schemes with the multi-authority [14, 22], an attribute-based broadcast encryption scheme [23], and a CP-ABE scheme with recipient anonymity [25]. A prob- lem of previous ABE schemes is that the length of the ciphertext depends on the number of attributes. Also, the number of pairing computations depends on the number of attributes. A Predicate Encryption Scheme (PES), where secret keys correspond to predicates, and where cipher- texts are associated with attributes, has been proposed in [11, 21]. It is shown that PES can be regarded as a kind of CP-ABE (see Appendix A and B in [25] for details). Both the [11] and [21] schemes also have the same prob- lems, in that the length of the ciphertext and the number of pairing computations are not constant.

Achieving Scalable Access Control Over Encrypted Data for Edge Computing Networks

The concept of Internet of things (IoT) has raised in the cloud computing paradigm as it adds latency when migrating all pieces of data from the network edge to the data center for them to be approached. Edge computing has been introduced to extend the cloud computing architecture to the edge of the network, which analyses most of

the IoT data near the devices that produce and act on that data. Though edge computing solves the latency problem of data processing, it also brings issues to the data security and privacy preservation. One technique which is potential to provide scalable access control to support data security and privacy in edge computing is attribute-based encryption (ABE). We, in this paper, propose a notion named proxy-aided ciphertext-policy attribute-based encryption (PA-CPABE), which outsources the majority of the decryption computations to edge devices. Compared to the existing ABE with outsourced decryption (ABE-OD) schemes, PA-CPABE has an advantage in that the key distribution does not require any secure channels. We present a generic construction of PA-CPABE, and then foramlly prove its security. In addition, we implement an instantiation of the proposed PA-CPABE framework to evaluate its performance.

The idea of Internet of Things (IoT) has become increas- ingly popular, which enables various objects including physi- cal devices, vehicles, buildings and other items embedded with computing and communication capabilities to exchange data. However, because of limitations in the computation capability, battery, storage and bandwidth, smart devices sometimes may decrease the quality of services and weaken the user expe- rience. Cloud computing supplies resources to end users in terms of software, infrastructure and platform, and delivers services to applications at a comparatively small cost, which has been considered as a promising solution to mitigate the limitation of devices with constrained resources. Unfortunately, cloud computing cannot be an answer to all emerging problems, since some IoT applications need to be instantly responded, some contain sensitive information, and some generate a large amount of data and cause a heavy workload to the network. The demand for distributing the IoT workloads between the local data centre and the cloud has resulted in an architectural model called Edge Computing.

Edge computing extends cloud computing and facilitates cloud computing in significantly reducing the delays incurred by service deployments. End devices, edge and cloud form a three-layerhierarchicalarchitecture(asshowninFig.1)forthe service delivery, which supports a wide range of applications (e.g., the smart city network). Take the autonomous vehicle network as an instance, where the vehicle might produce gigabyte data in one second, and the real-time processing is in necessity as any delay in practice could lead the vehicle to make false resolutions. In such a situation, the responding time could be extremely long if all data items are going to be forwarded to and processed by the cloud, and thus it would be very demanding for the current network to support a large number of vehicles in the same area. Therefore, it is essential for all kinds of data items to be managed at the network edge to reach a more effective management and a shorter response time.

Edge devices reduce communication and computation over- heads by providing computing, networking and storage ser- vices and making decisions at the network edge. Unfortu- nately, edge devices requiring less cost than cloud servers can be easily compromised by adversaries and cannot be trusted, especially in the data sharing (e.g., vehicles may need to share the traffic data when travelling on the same motorway) situation. Therefore, it is indispensable to arm an edge computing network with an access control mechanism to allow the data to be shared among data users possessing certain attributes while preventing other entities (including the cloud server, edge devices and unprivileged data users) from learning the original data.

KEYD: SECURE KEY-DEDUPLICATION WITH IDENTITY-BASED BROADCAST ENCRYPTION

De-duplication is a technique used for removing duplicate copies of data in the cloud in order to reduce the storage space and upload bandwidth. Before outsourced, the data which is about to be upload in the cloud will be encrypted for ensuring data confidentiality. Traditional encryption will produce different ciphertexts which

are produced from the same plain text by different user's secret key, which makes difficult for de-duplication. To overcome this problem, we go for Convergent Encryption which naturally encrypts the same plain texts into same ciphertexts. This project explains the problem of achieving reliable key management in secure de-duplication. since we use the baseline approach for key management for maintaining an enormous number of keys with the increasing number of users where user have to protect their master key from the third party. So we designed a novel client-side de-duplication protocol named KeyD by using Identity-based broadcast Encryption (IBBE) instead of the independent key management server. The user will interact with the cloud service provider(CSP) while uploading files and downloading it. Security Analysis explains that KeyD ensures data confidentiality and convergent key security at the same time it provides ownership privacy. Our scheme makes better tradeoff among storage cost, communication and computation overhead.

The system in which "KeyD secure Deduplication" is a web application which is used for avoiding the deduplication in the cloud where it relatively increases the storage space. By the year 2020, the volume of data will reach up to 40 trillion gigabytes. To make data management scalable, deduplication has been introduced. The process of providing ownership for files is to remove duplicates or replica information which is done automatically while uploading files. In our existing system, the encryption process of a single file for the number of times with different keys so that encrypted files are different in a different manner. In order to avoid multiple data copies with the same content, we go for both file level and block level granularities.

Privacy-preserving Machine Learning in Cloud

Machine learning algorithms based on deep neural networks (NN) have achieved remarkable results and are being extensively used in different domains. On the other hand, with increasing growth of cloud services, several Machine Learning as a Service (MLaaS) are offered where training and deploying machine learning model sare performed on cloud providers' infrastructure. However, machine learning algorithms require access to raw data which is of ten privacy sensitive and can create potential security and privacy risks. To address this issue, we develop new techniques to provide solutions for applying deep neural network algorithms to the encrypted data. In this paper, we show that it is feasible and practical to train neural networks using encrypted data and to make encrypted predictions, and also return the predictions in an encrypted form. We demonstrate applicability of the proposed techniques and evaluate its performance. The empirical results show that it provides accurate privacy-preserving training and classification.

In this paper, we present a solution to apply neural network algorithms to encrypted data and allow the parties to provide/receive the service without thaving to reveal their sensitive data to the other parties. The main components of our proposed approach are homomorphic encryption and neural networks. We need to adopt neural network algorithms with in limitations of homomorphic encryption.

Providing User Security Guarantees in Public Infrastructure Clouds

The infrastructure cloud (IaaS) service model offers improved resource flexibility and availability, where tenants – insulated from the minutiae of hardware maintenance – rent computing resources to deploy and operate complex systems. Large-scale services running on IaaS platforms demonstrate the viability of this model; nevertheless, many organizations operating on sensitive data avoid migrating operations to IaaS platforms due to security concerns. In this paper, we describe a framework for data and operation security in IaaS, consisting of protocols for a trusted launch of virtual machines and domain-based storage protection. We continue with an extensive theoretical analysis with proofs about protocol resistance against attacks in the defined threat model. The protocols allow trust to be established by remotely attesting host platform configuration prior to launching guest virtual machines and a public electronic health record system, showing that the proposed protocols can be integrated into existing cloud environments.

Cloud computing has progressed from a bold vision to massive deployments in various application domains. However, the complexity of technology underlying cloud computing introduces novel security risks and challenges. Threats and mitigation techniques for the IaaS model have been under intensive scrutiny in recent years, while the

industry has invested in such mechanisms. There is a clear need for usable and cost-effective cloud platform security mechanisms suitable for organizations that rely on cloud infrastructure. \

One such mechanism is platform integrity verification for compute hosts that support the virtualized cloud infrastructure. Several large cloud vendors have signaled practical implementations of this mechanism, primarily to protect the cloud infrastructure from insider threats and advanced persistent threats.. To address this, we propose a set of protocols for trusted launch of virtual machines (VM) in IaaS, which provide tenants with a proof that the requested VM instances were launched on a host with an expected software stack.

Secure Data Group Sharing and Dissemination with Attribute and Time Conditions in Public Cloud

Cloud computing has become increasingly popular among users and businesses around the world. Although cryptographic techniques can provide data protection for users in public cloud, several issues also remain problematic, such as secure data group dissemination and fine-grained access control of time-sensitive data. In this paper, we propose an identity- based data group sharing and dissemination scheme in public cloud, in which data owner could broadcast encrypted data to a group of receivers at one time by specifying these receivers' identities in a convenient and secure way., we adopt attribute-based and timed-release conditional proxy re-encryption to guarantee that only data disseminators whose attributes satisfy the access policy of encrypted data can disseminate it to other groups after the releasing time by delegating a re-encryption key to cloud server. The re-encryption conditions are associated with attributes and releasing time, which allows data owner to enforce fine-grained and timed-release access control over disseminated ciphertexts. The theoretical analysis and experimental results show our proposed scheme makes a tradeoff between computational overhead and expressive dissemination conditions.

CLOUD computing is regarded as such a computing paradigm in which resources in the computing infra- structure are provided as services over the Internet. The cloud computing, security problems may somehow im- pede its quick development. Currently, more and more users would outsource their data to cloud service provid- er (CSP) for sharing. However, the CSP which deprives data owners' direct control over their data is assumed to be honest-but-curious, that may prompt security concerns. These security matters existing in public cloud motivate the requirement to appropriately keep data confidential. Several schemes exploiting cryptographic mechanisms to settle the security problems have been proposed. In order to guarantee secure data group sharing, identity-based broadcast encryption (IBBE) scheme is employed in public cloud. The data owners could broadcast their en- crypted data to a group of receivers at one time and the public key of the user can be regarded as email, unique id and username. Hence, by using an identity, data own- er can share data with other group users in a convenient and secure manner. Attribute-based encryption (ABE) is one of new cryptographic mechanisms used in cloud to reach flexible and fine-grained. policies to authorize others to see this photo. In this case, Bob is a disseminator of the photo. The proxy re-encryption (PRE) scheme in authorized users who can view his data to disseminate data or allow the dissemina- tors to disseminate all of his data. For example, Alice au- thorizes Bob and Carol to access her data, but she only allows Bob to disseminate some specific photos or videos to his space. The conditional PRE (CPRE) scheme could address this issue by allowing a user to generate a re-encryption key associated with a condition, and only the encrypted data meeting the condition can be re- encrypted. However, conditions in traditional CPRE which are only keywords may not well match situations in cloud because data owner may have a large number of requirements for different disseminators to disseminate his different data, such as photos taken in home only for families to disseminate and travelling photos allowed to be disseminated by friends. Thus, fine-grained conditions are inevitably needed in data group dissemination situation in public cloud.

3.Fundamental Concepts on (Domain)

Cloud Computing & its Applications

1) What is Cloud Computing?

- Cloud computing is a kind of computing system in which various hardware, software and applications share their facilities over the internet.
- Role of Cloud Computing is very remarkable & it is one of the emerging technology in the world of computers.
- The cloud computing is a better way to run your business instead of having your own resource you can use resources as services. Cloud run on a shared data centers virtually, hence the name Cloud Computing.

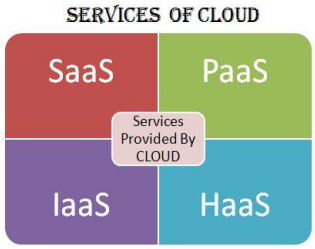
❖ **Types of clouds:-**

❖ There are different types of clouds that you can subscribe to depending on your needs.

- ➢ **Public Cloud**
- ➢ **Private Cloud**
- ➢ **Community Cloud**
- ➢ **Hybrid Cloud**

SERVICES PROVIDED BY CLOUD

- **Software as a Service (SaaS)**
- **Platform as a Service (PaaS)**
- **Infrastructure as a Service (IaaS)**
- **Hardware as Service (HaaS)**



2) Why Cloud Computing?

The cloud computing provides the following benefits to the world wide uers.

Flexibility – Scale up and down to meet your organization's requirements.

- **Security** – The data in the cloud is much more secure than what lives on a tower under your desk or in yourunsecured server room.

- **Capacity** – In the past, you had to spend a lot of your IT budget on human resources to manage your software. With cloud computing, that's no longer an issue. Now, you can focus on how the solution will help you further your mission.
- **Cost** – Using cloud technology reduces your maintenance fees. No more servers, software, and update fees. Many of the hidden costs typically associated with software implementation, customization, hardware, maintenance, and training are rolled into a transparent subscription fee.
- **It's open** – Internet standards and web services allow you to connect services to each other. This means that you can centralize your information and access it from anywhere in the world, on any computer or mobile device, at any time.

Applications of cloud computing

1) **Cloud Computing in Business**

The business delivery model provides a user experience by which hardware, software and network resources are optimally leveraged to provide innovative services over the Web, and servers are provisioned in accordance with the logical needs of the service using advanced, automated tools.

2) Cloud Computing in Education

Today's IT professionals in educational institutions need to respond quickly to increasing demands from students and faculty, while coping with fixed or declining budgets and staff. In this challenging environment, cloud-based computing has become an increasingly attractive option for delivering education services more securely, reliably, and economically It is one of the fastest-growing industries in the world.

3) Online Entertainment

Most people come on the internet for entertainment; Therefore, cloud computing is the perfect place for reaching to a varied consumer base. Cloud-based entertainment can reach any device be it TV, mobile, set top box, or any other form.

4) Telecommunication

Telecommunication companies can use cloud computing to provide both private and public cloud networks to customers and organizations for domestic and commercial purposes.

5) Finance and Banking

As the international market grew so did the need for a more condensed and easier financial reach. Cloud computing eliminates the need for having a separate banking portal and client database for every location. This means faster and better business.

4. System Analysis

4.1Existing System:

However, the data disseminator can disseminate all of the data owner's data to others with this re-encryption key, which may not meet the practical requirement since the data owner may only permit the data disseminator to disseminate a particular document. A refined concept referred to as conditional PRE (CPRE) could address this issue, in which data owner can enforce re-encryption control over the initial ciphertexts and only the ciphertexts satisfying specific condition can be re-encrypted with corresponding reencryption key. However, traditional CPRE schemes only support simple keyword conditions, so they cannot match complex situations in cloud computing well which deploys an access policy in the ciphertext. The re-encryption key is associated with a set of attributes, thus the proxy can reencrypt the ciphertext, data owner allows project managers in the organization to disseminate the progress report in OneDrive, while department to disseminate the project budget in OneDrive during a specific time period.

4.1.1 Disadvantages:

Besides the requirement of conditional data dissemination, multiparty access control problem for data sharing in cloud computing such as cloud collaboration and cloud-based social networks comes along which means the special authorization requirements from multiple associated users can be accommodated together to control the shared data., and Carol. If Alice who is the data owner uploads this co-authoring document or cophoto to the CSP and tags both Bob and Carol as the co-owners. Alice can restrict this data to serious privacy problem if applying the preference of only one party, which may cause

4.3 Proposed System

However, merging privacy preferences of data owner and multiple co-owners is not an easy task, due to privacy conflict is inevitable in multiparty authorization enforcement Privacy conflict happens when the coowners have opposite privacy policies, and it results in data being impossibly accessed with anyone. To deal with this dilemma, multiparty access control mechanisms (e.g. voting scheme) are further provided. To mitigate the problems mentioned above, we introduce a solution to achieve ciphertext group sharing among multiple users, and capture the core feature of multiparty authorization requirements.

4.3.1 Advantages

Multiparty access control mechanism allows the data co-owners to append new access policies to the cipher text due to their privacy preferences. Hence, the cipher text can be re-encrypted by the data

The majority permit strategy, data owner can firstly choose a threshold value for data co-owners, and the cipher text can be disseminated if and only if the sum of the access policies satisfied by data disseminator's attributes is greater than or equal to this fixed threshold.

4.4 Modules Description

1) Trusted authority: The trusted authority is a fully trusted part that initializes the system public key, and generates private keys as well as attribute keys for users. For example, it can be acted by the administrator of the organization or social security administration.

2) CSP: The CSP is a semi-trusted part that provides each user with a virtual space and convenient data storage service with the cloud infrastructure. It also appends access policies to the cipher texts for data co-owners and generates re encrypted cipher texts for users.

3) User: We divide the user role into the following categories: data owner, data co-owner, data access policy to enforce dissemination conditions. Then he encrypts data for a set of receivers, and outsources the cipher text to CSP

for sharing and dissemination. The data co-owners tagged by data owner can append access policies to the encrypted data with CSP and generate the renewed cipher text. to disseminate data owner's data to others if he satisfies enough access policies in the cipher text. The data accessor can decrypt the initial, renewed and re-encrypted ciphertext with her or his private key.

4.5 FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

4.5.1 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

4.5.2 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help.

4.5.3 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Therefore, itare not many and are already available in-house at NIC or are available as free as open source.

5. System Requirements Specification

5.1 Introduction

A **Software Requirements Specification** (**SRS**) – a requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- **Business requirements** describe in business terms *what* must be delivered or accomplished to provide value.
- **Product requirements** describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- **Process requirements** describe activities performed by the developing organization. For instance, process requirements could specify specific methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement); a requirement that the product be maintainable (a Product requirement) often is addressed by imposing requirements to follow particular development styles

5.2 PURPOSE

An systems engineering, a **requirement** can be a description of *what* a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called Non-functional requirements, or 'quality of service requirements.' A collection of requirements define the characteristics or features of the desired system. A 'good' list of requirements as far as possible avoids saying, leaving such decisions to the system designer. Specifying how the system should be implemented is called "implementation bias" or "solution engineering". However, *implementation constraints* on the solution In software engineering, the same meanings of requirements apply, except that the focus of interest is the software itself.

5.3 FUNCTIONAL REQUIREMENTS

Admin

Admin can add movies, view embedding layer, region layer, locally attentive layer and non linear attentive layer Admin recommended movies to users.

User

User view his/her own profile and view recommended movie and high rated movies.

5.4 NON FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability

The system is designed with completely automated process hence there is no or less user intervention.

Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

Performance

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

Implementation

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform.

Interface the user interface is based on Struts provides HTML Tag

5.5 Input & Output Design

Input Design:

Inaccurate input data are the most common causes of errors in data processing. Errors entered by data entry operators can be controlled by the Input design. "Input design is the process of converting user originated inputs to computer based formats". It consists of developing specification and procedure for data preparation.

Objectives of input design:

- The main objectives of input design are:
- 1. Controlling amount of input: Due to so many reasons, design should control the quantity of data for input. Reducing the data requirement can lower cost by reducing labour expenses. By reducing input requirement, the analyst can speed the entire process from data capture to providing results to the users.
- Avoiding delay: A processing delay resulting from data preparation or data entry operator is called bottlenec . Avoiding bottleneck should always be one objective of the analyst while designing output.
- Avoiding errors in data: The rate at which errors occurs depends on the quantity of data, i.e. smaller the amount of data to input the fewer the opportunities for errors.
- Keeping the process simple: Simplicity works and is accepted by the users. Complexity should be avoided when there are simple alternatives.

Output Design:

- The term output necessarily implies to information on printed or displayed by an information system. Following are the activities that are carried out in output design stage.
- Ø Identification of specific output required to meet the information requirements.
- Ø Selecting of methods for processing outputs.
- Ø Designing of reports, formats or other documents that acts as a carrier of information.

Output Design Activities

The output design of an information system must meet the following objectives:

1. The output design should provide information about the past, present or future events. The operational control level outputs provide operations of the past and present events. On the other hand, strategic planning level

2. The output design should indicate the important events, opportunities and problems.

3. The output design should be designed keeping in mind that an action must be triggered in response to some event. A set of rule is pre- designed for such trigger.

4. The output design should produce some action to the transaction. For e.g. when the telephone bill is generated, a receipt is printed.

5.6 Software Requirements:

Language	:	JDK (1.7.0)
Frontend	:	JSP, Servlets
Backend	:	Oracle10g
IDE	:	my eclipse 8.6
Operating System	:	windows XP
Server	:	tomcat

5.7 Hardware Requirements:

Processor	:	Pentium IV
Hard Disk	:	80GB
RAM	:	2GB

6. System Design

6.1Introduction

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design. System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. During, Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module is usually specified in a high-level design description language, which is independent of the target language in which the software,., the interaction between parts, Developers bridge the gap between the requirements specification , produced during requirements elicitation and analysis , and the system that is delivered to the user. Design is the place where the quality is fostered in development . Software design is a process through which requirements are translated into a representation of software.

6.2 System Model

Introduction to UML

The unified Modeling Language (UML) is a standard language for writing software blueprints. The UML may be used to visualize, specify , construct and document the artifacts of software-intensive system.

The goal of UML is to provide a standard notation that can be used by all object - oriented methods and to select and integrate the best elements .UML is itself does not prescribe or advice on how to use that notation in a software development process or as part of an object - design methodology. The UML is more than just bunch of graphical symbols. Rather , behind each symbol in the UML notation is well-defined semantics.

The system development focuses on three different models of the system.

- ➔ Functional model
- ➔ Object model
- ➔ Dynamic model

Functional model in UML is represented with use case diagrams , describing the functionality of the system from user point of view.

Object model in UML is represented with class diagrams , describing the structure of the system in terms of objects , attributes , associations and operations.

Dynamic model in UML is represented with sequence diagrams , start chart diagrams and activity diagrams describing the internal behaviour of the system.

Scenarios

A Use Case is an abstraction that all describes all possible scenarios involving the described functionality . A scenario is an instance of a use case describing a concrete set of actions.

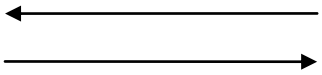
- ➔ The **name** of the scenario enables us to refer it ambiguously. The name of scenario is underlined to indicate it is an instance.
- ➔ The **Participating actor instance** field indicates which actor instance are involved in this scenario. Actor instance also have underlined names.
- ➔ The **Flow of Events** of scenario describe the sequence of events step by step.

6.3 Data Flow Diagrams:

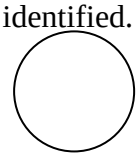
A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

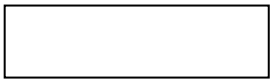
1. Dataflow: Data move in a specific direction from an origin to a destination.



2. Process: People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

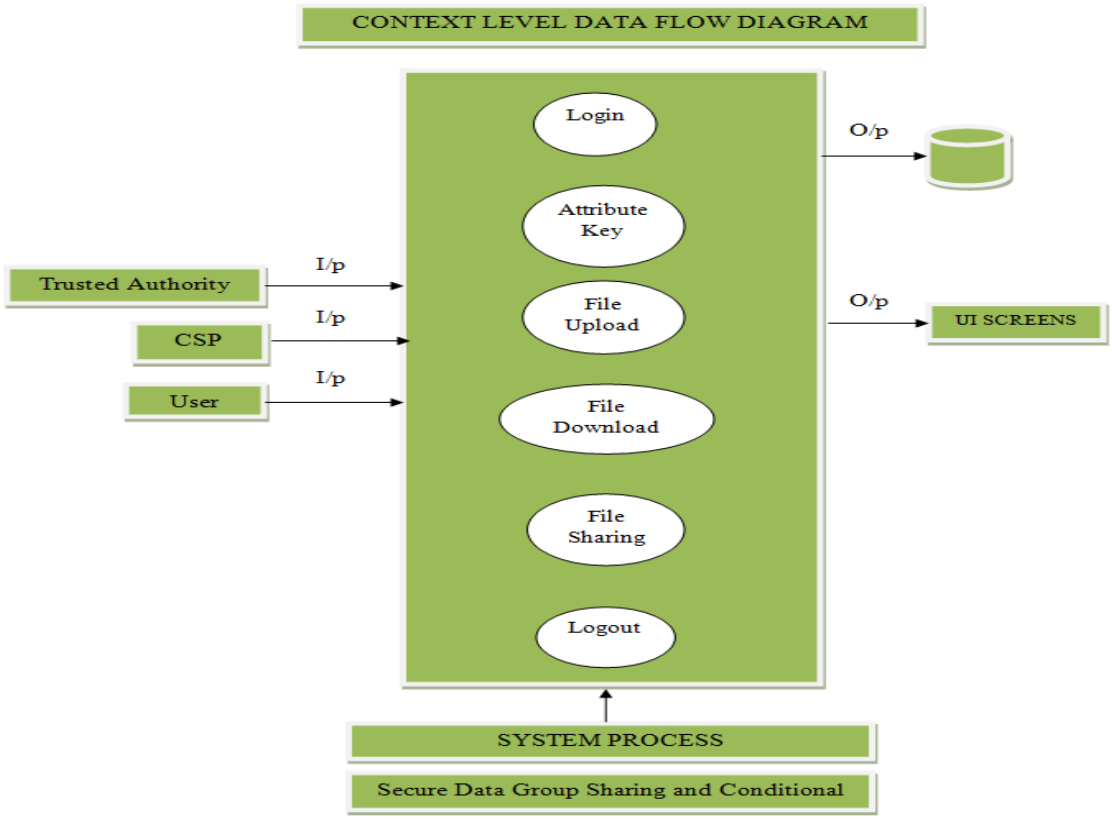


3. Source: External sources or destination of data, which may be People, programs, organizations or other entities.

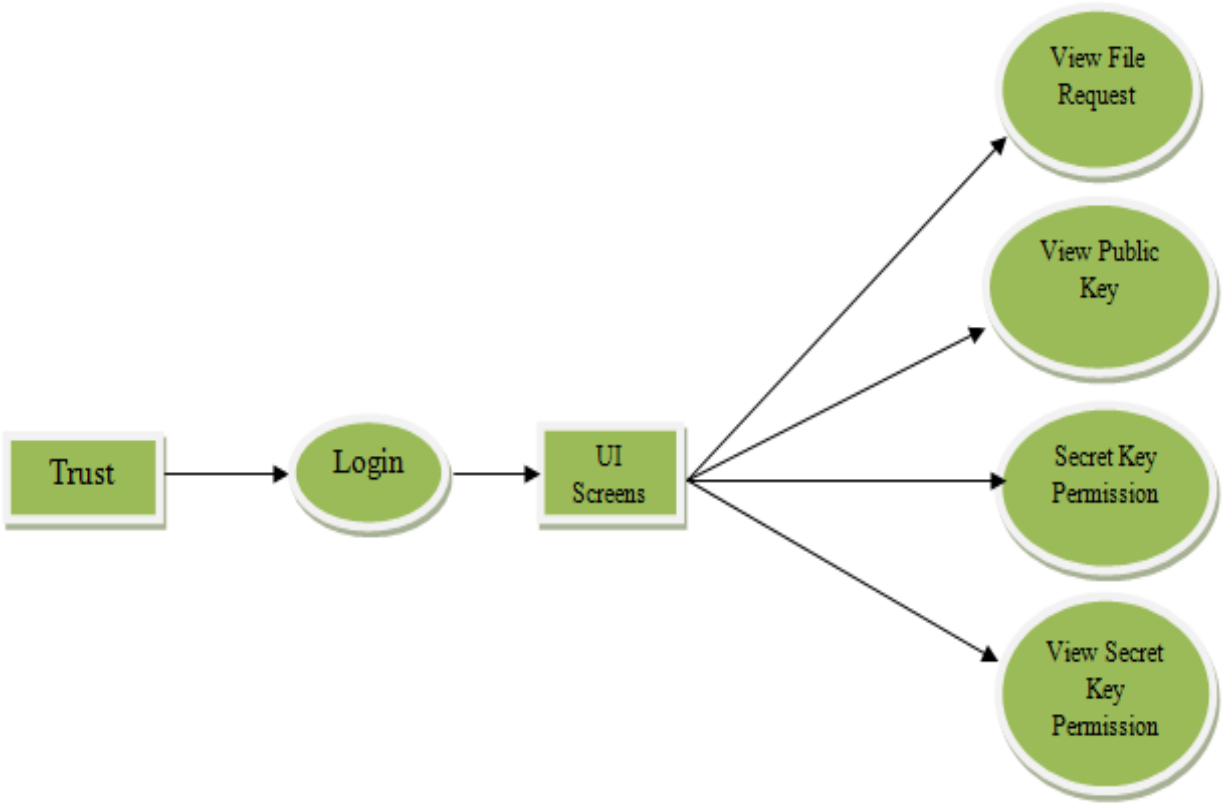


4. Data Store: Here data are stored or referenced by a process in the System.

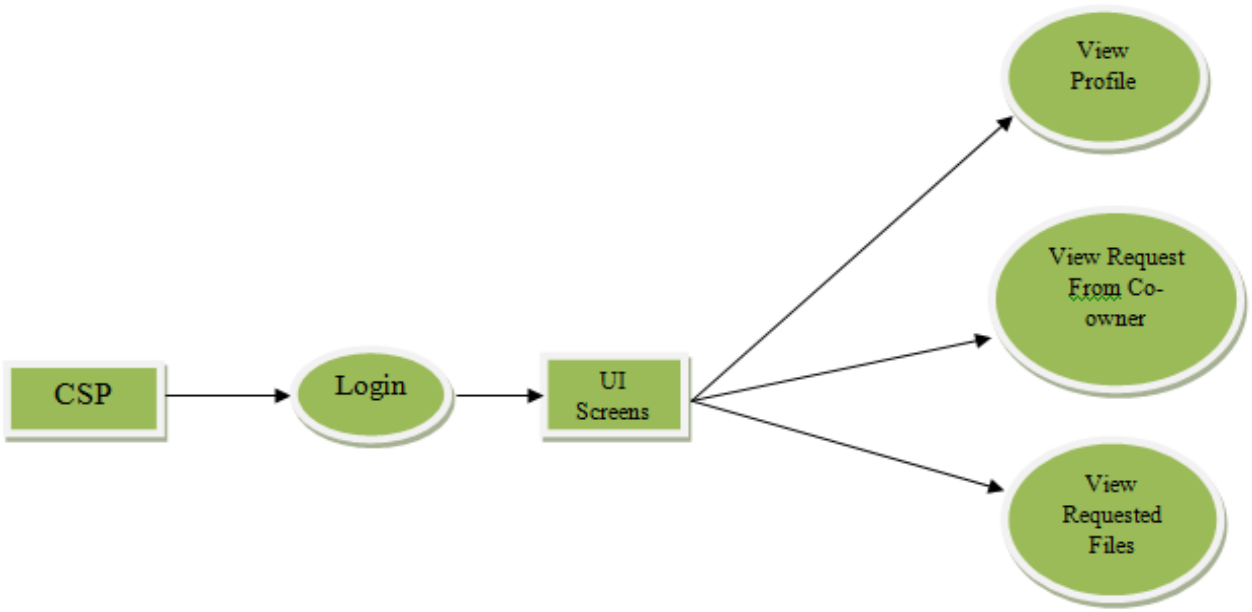
Level 0 Data Flow Diagram



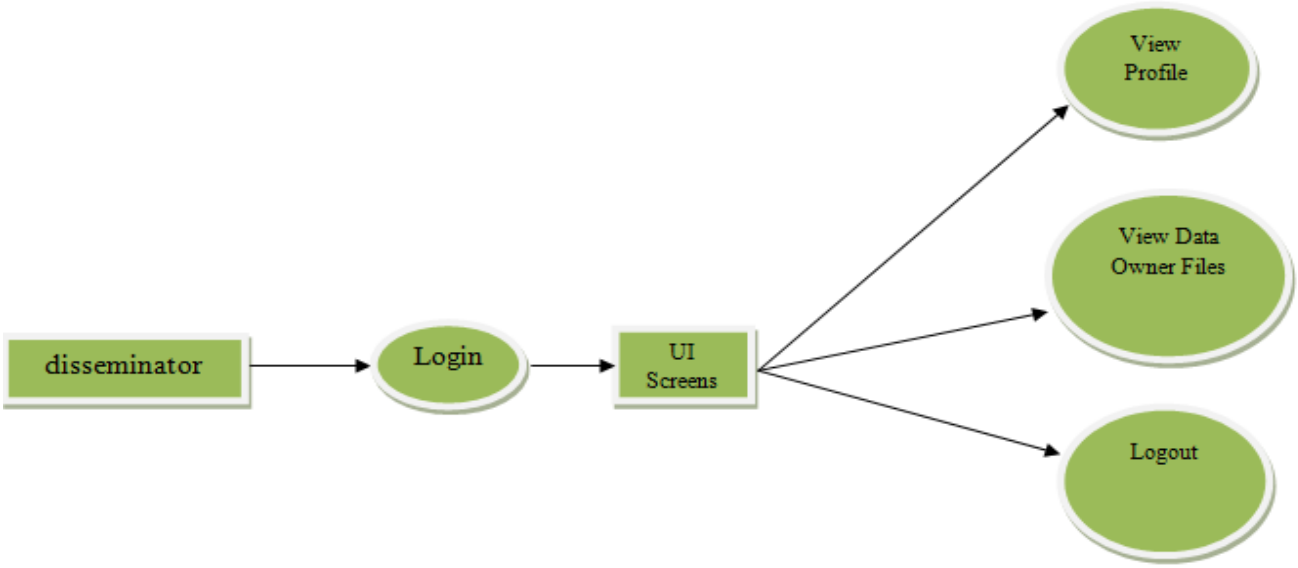
Trust Level 1 Data Flow Diagram



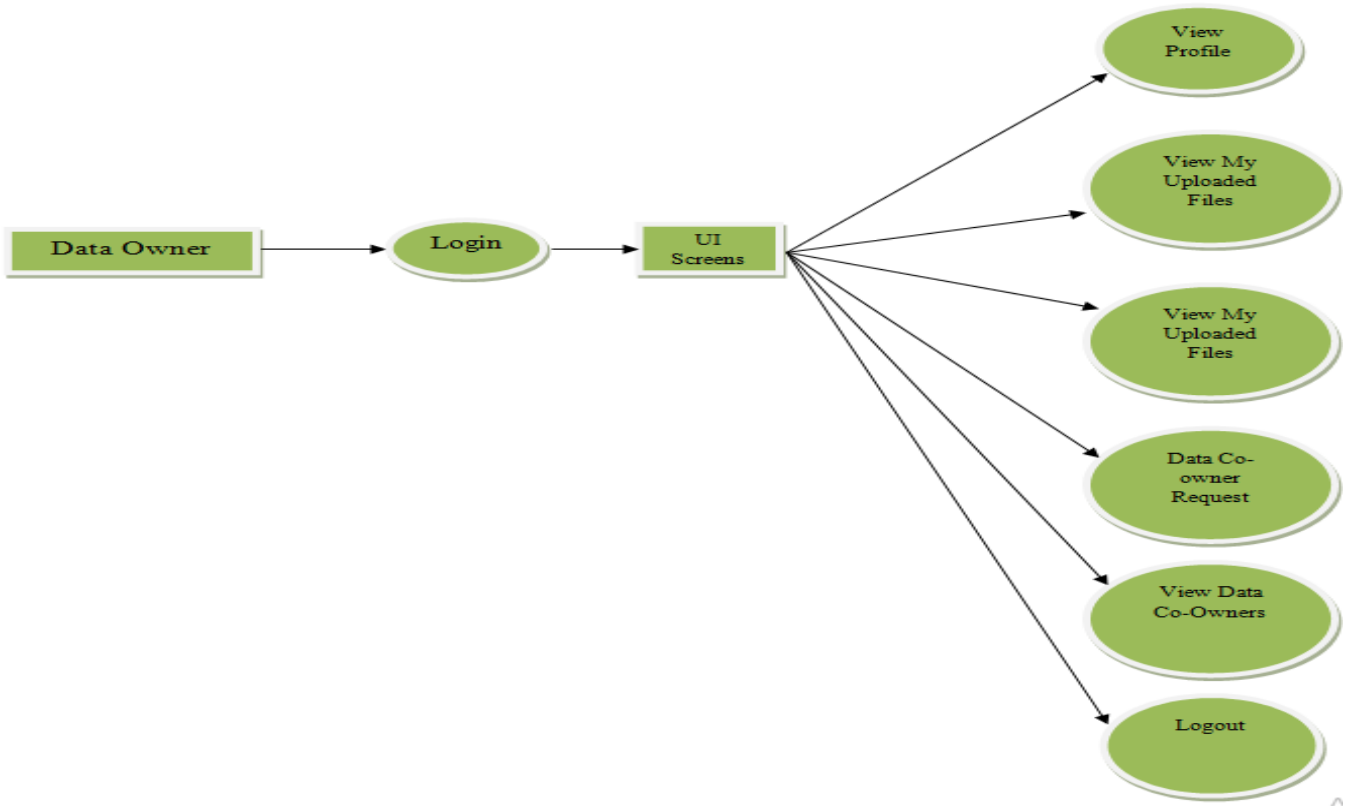
CSP Level1 Data Flow Diagram



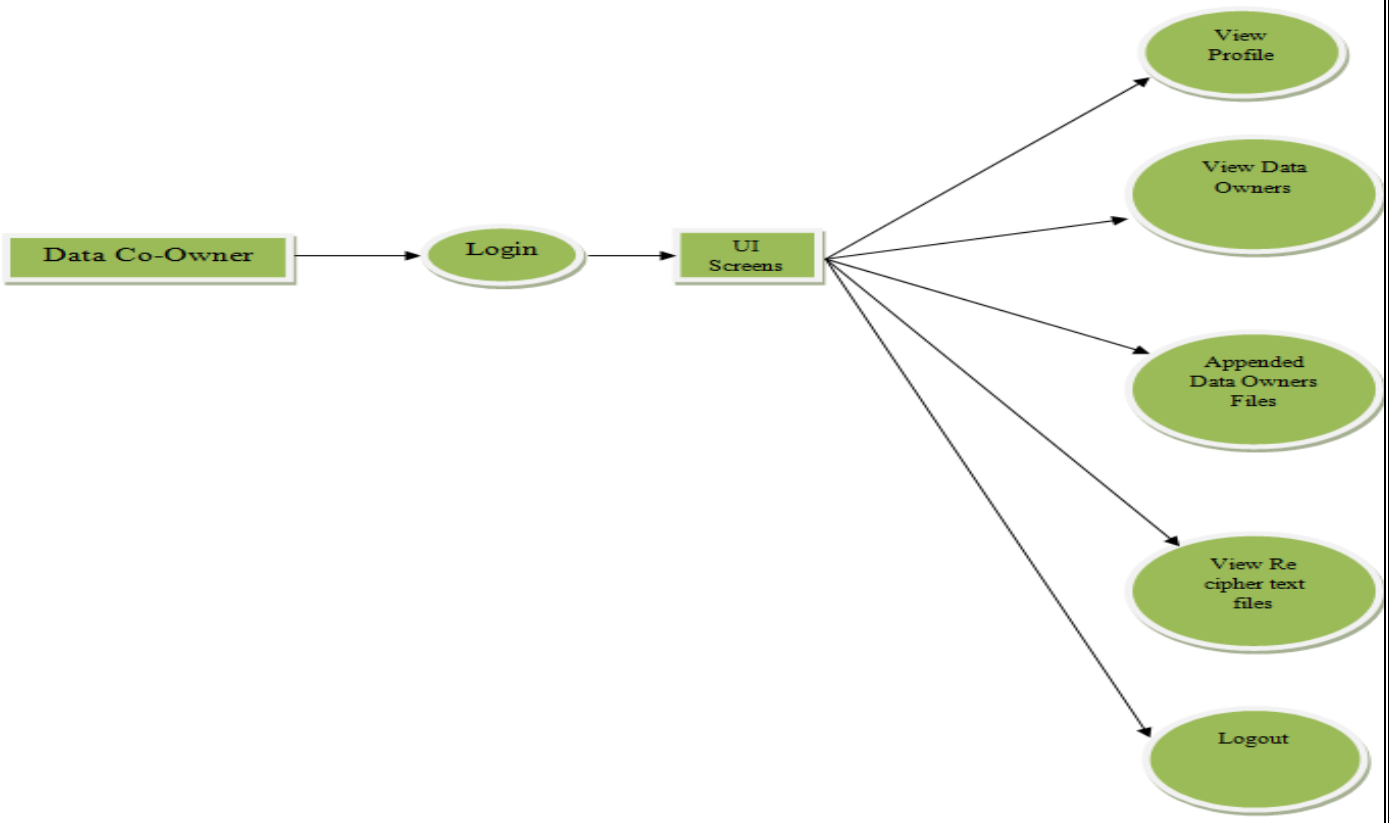
Disseminator Level1 Data Flow Diagram



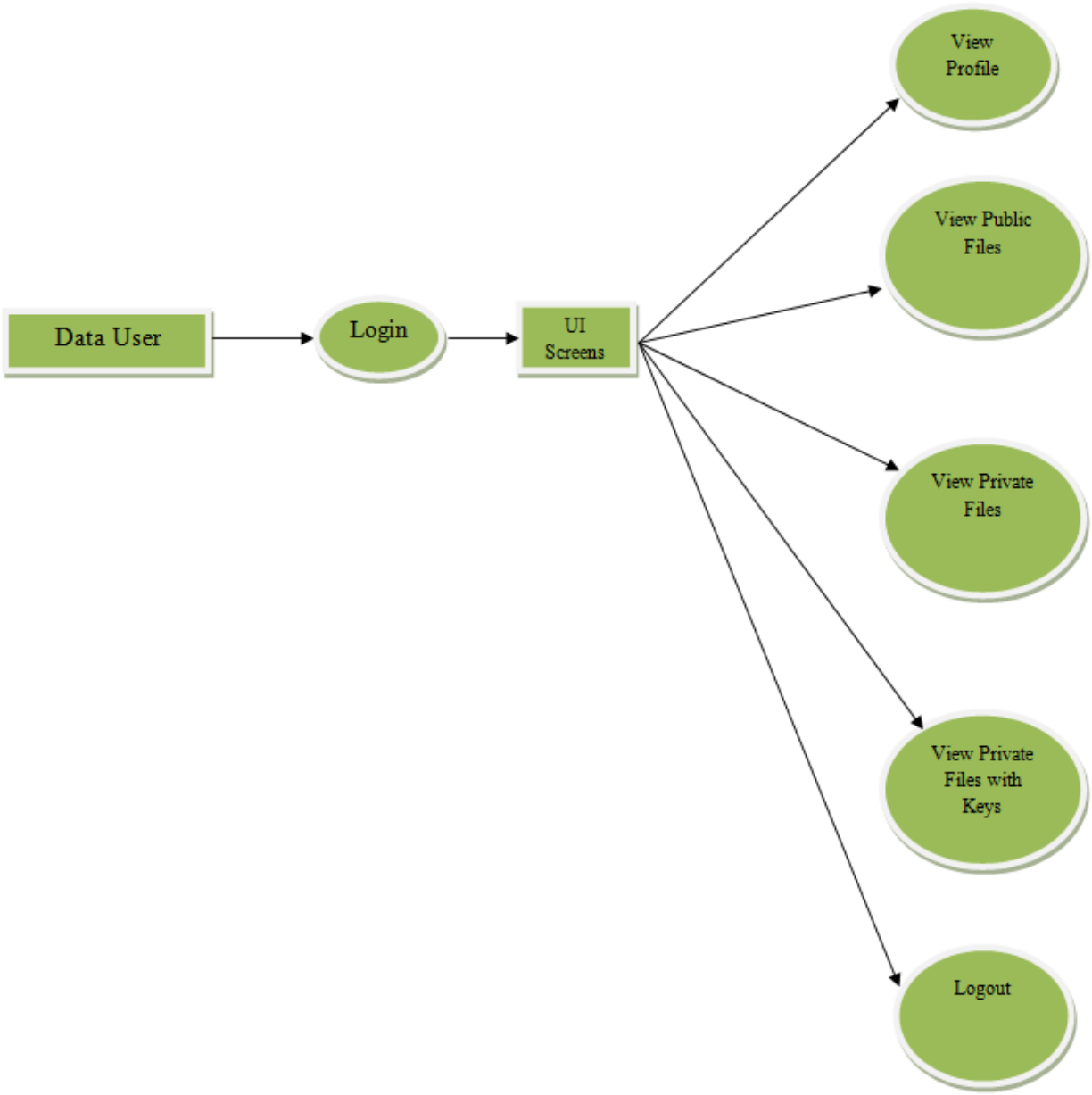
Data Owner Level1 Data Flow Diagram



Data Co-owner Level1 Data Flow Diagrams



Data User Level1 Data Flow Diagrams

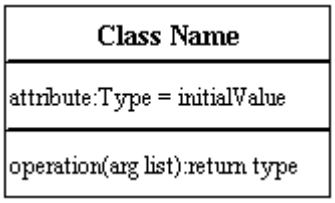


6.4 UML Diagrams

Class diagrams are the backbone of almost every object-oriented method including UML. They describe the static structure of a system.

Basic Class Diagram Symbols and Notations

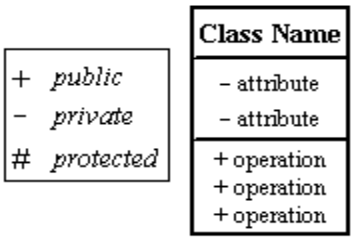
Classes represent an abstraction of entities with common characteristics. Associations represent he relationships between classes.Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition, and write operations into the third.



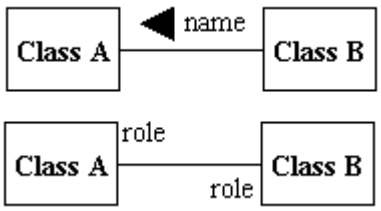
Active Class

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.

Visibility : Use visibility markers to signify who can access the information contained within a class. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access inform ation they inherited from a parent class. .

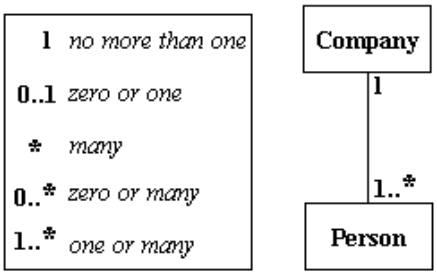


Associations : Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other. ***Note:*** It's uncommon to name both the association and the class roles.



Multiplicity (Cardinality)

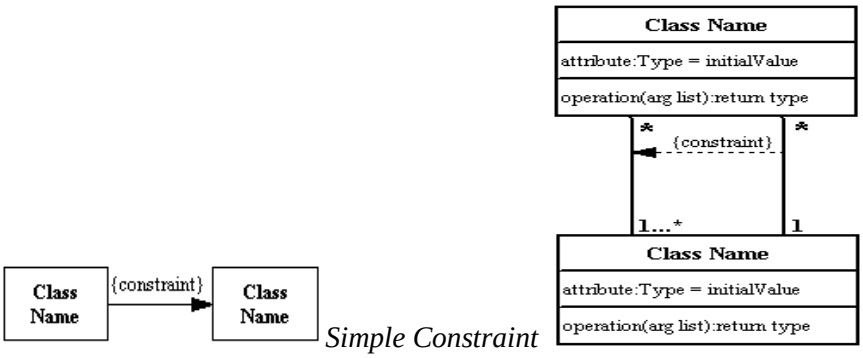
Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but



each employee works for one company only.

Constraint

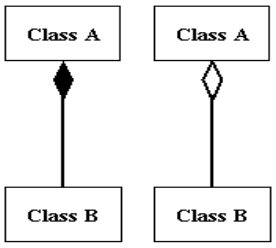
Place constraints inside curly braces {}.



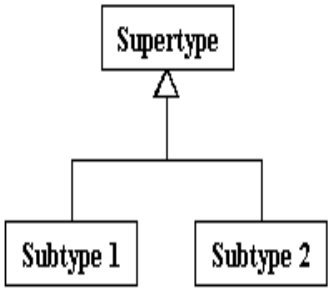
Simple Constraint

Composition and Aggregation

Composition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part. Illustrate **composition** with a filled diamond. Use a hollow diamond to represent a simple **aggregation** relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other. The diamond end in both a composition and aggregation relationship points toward the "whole" class or the aggregate

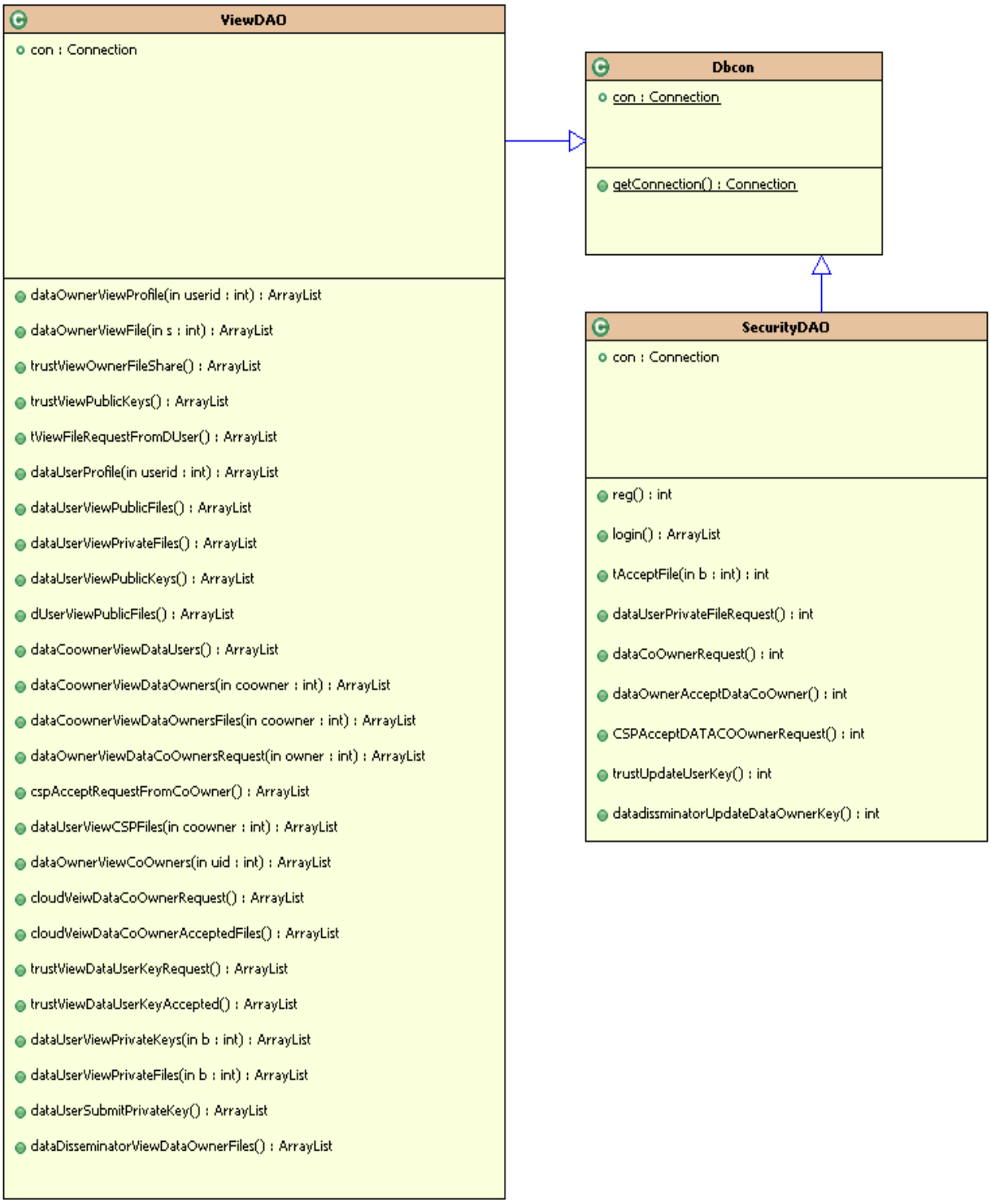


Generalization Generalization is another name for inheritance or an "is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another. For example, Honda is a type of car. So the class Honda would have a generalization relationship with the class car.



In real life coding examples, the difference between inheritance and aggregation can be confusing. If you have an aggregation relationship, the aggregate (the whole) can access only the PUBLIC functions of the part class. On the other hand, inheritance allows the inheriting class to access both the PUBLIC and PROTECTED functions of the super class.

Class Diagram:



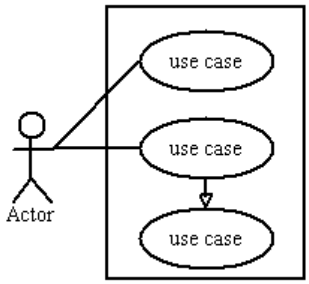
Use Case Diagram

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users.

Basic Use Case Diagram Symbols and Notations

System

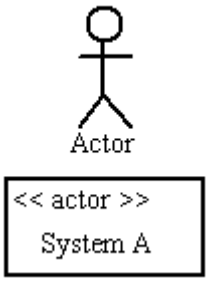
Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



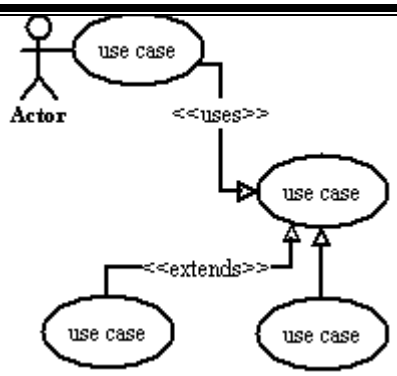
Use Case

Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

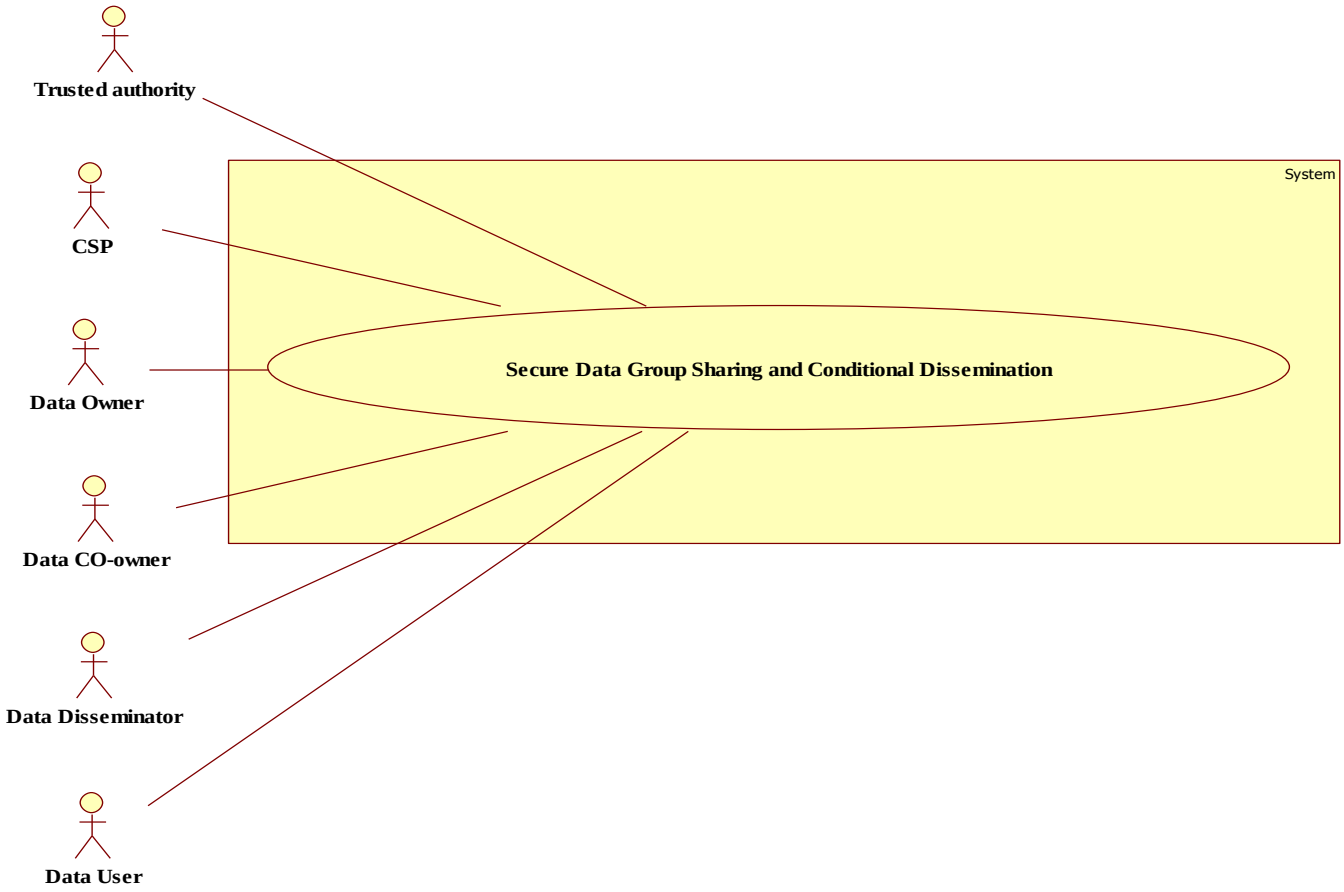
ActorActors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



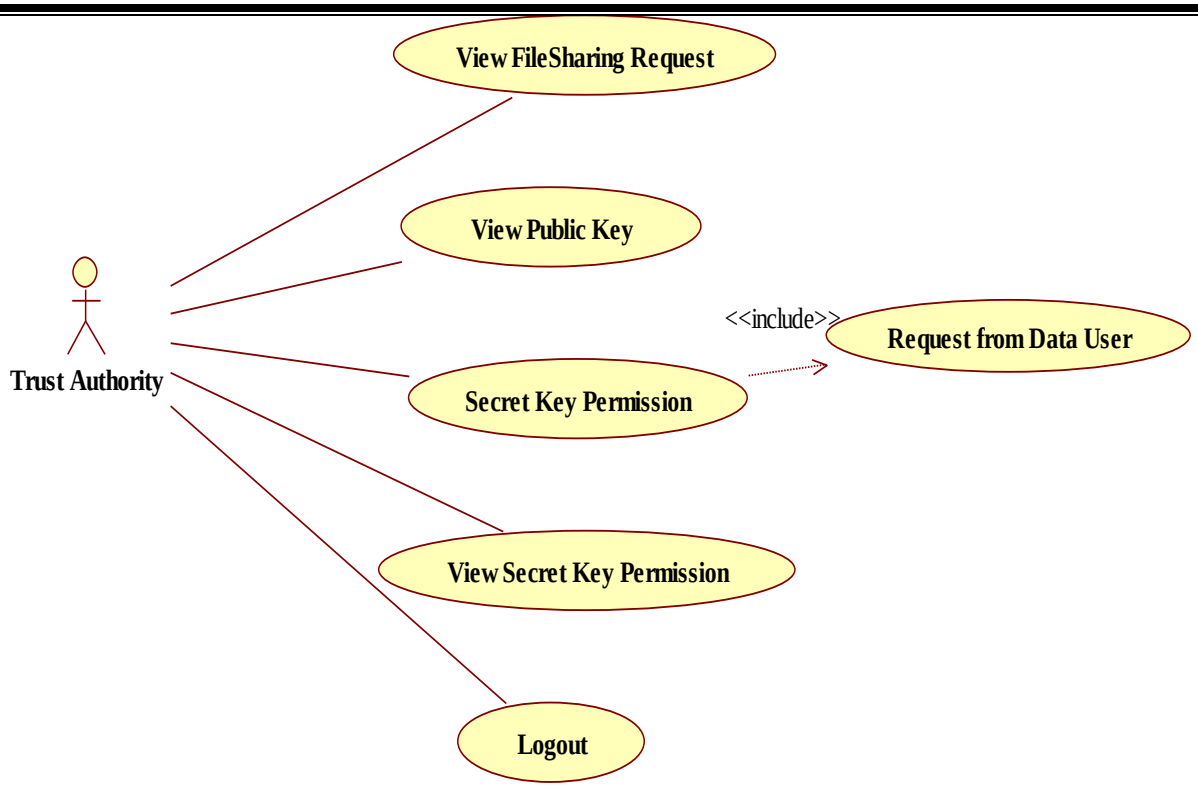
Relationships: llustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



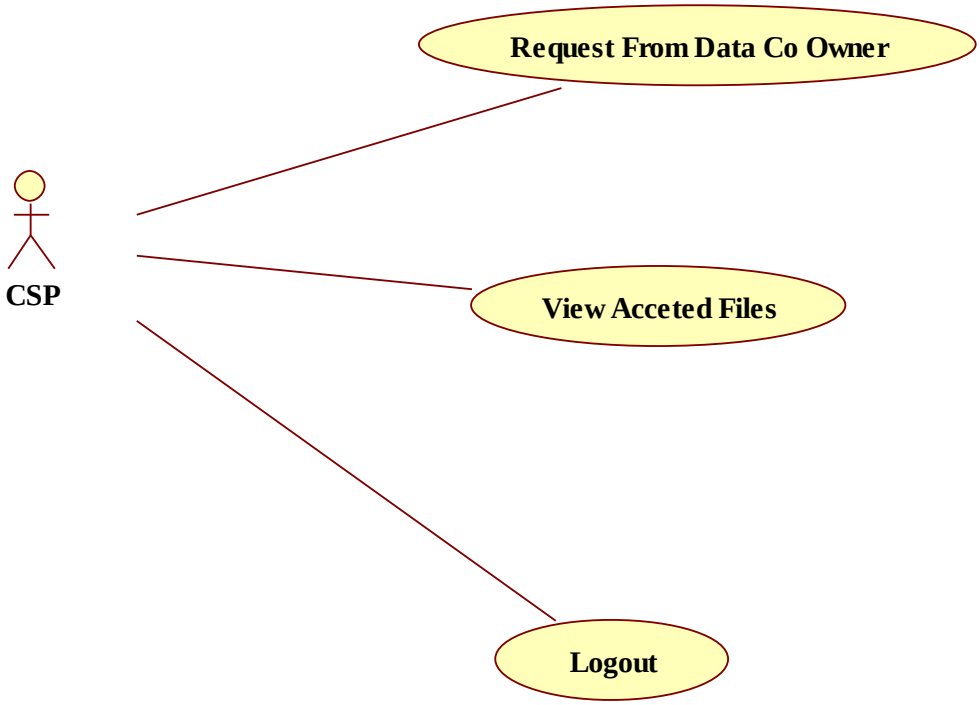
System Use Case Diagram



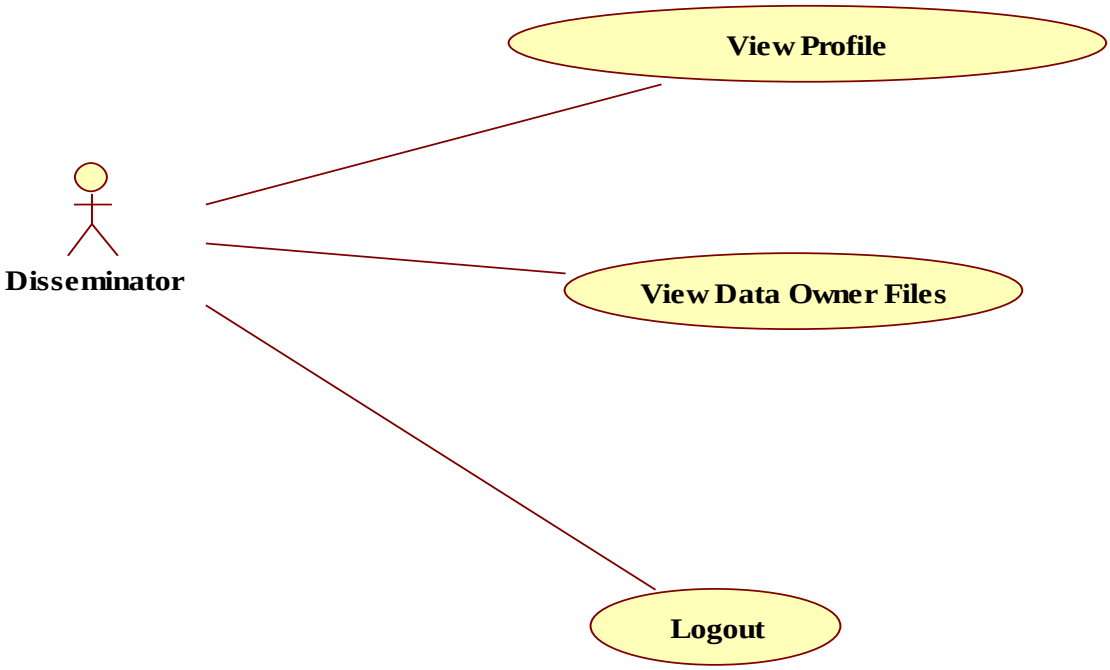
Trust Authority Use Case Diagram



CSP Use Case Diagram



Disseminator Use Case Diagram



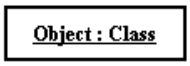
Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

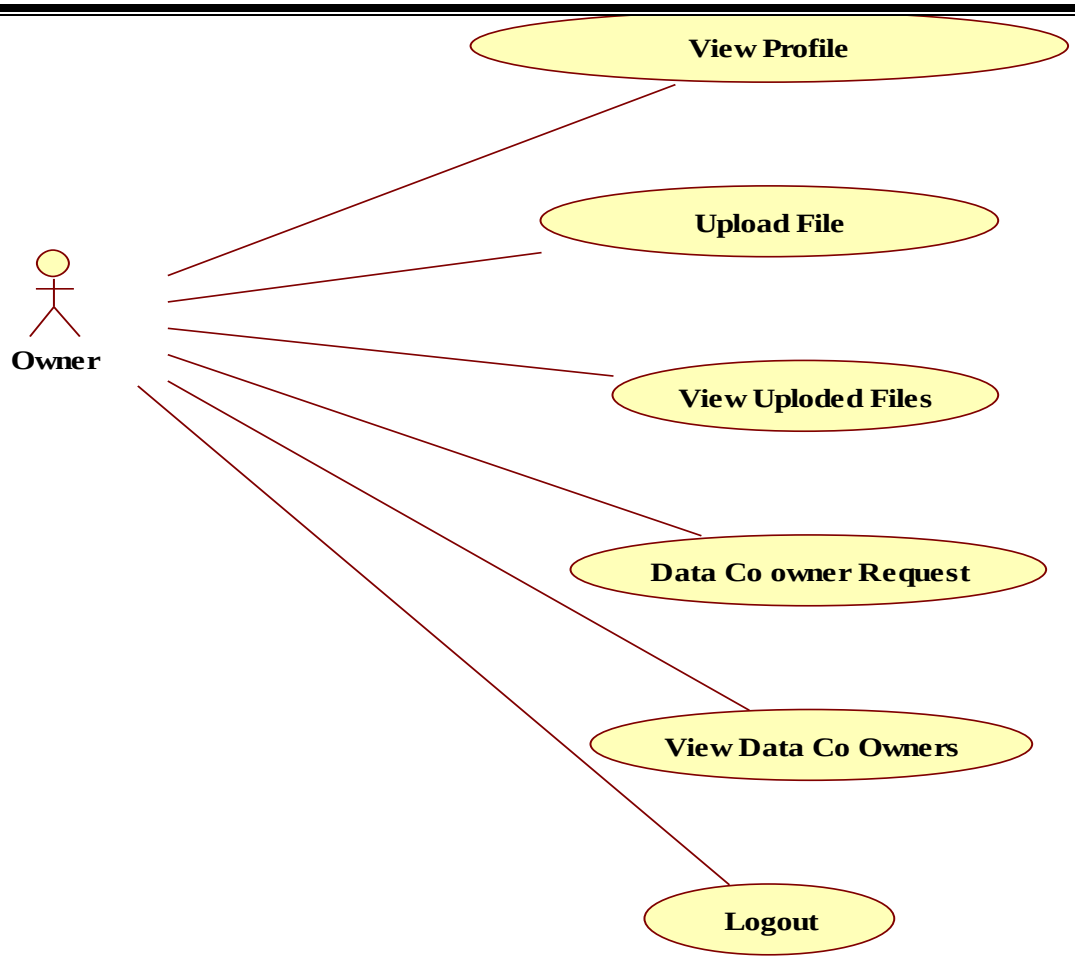
Basic Sequence Diagram Symbols and Notations

Class roles

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

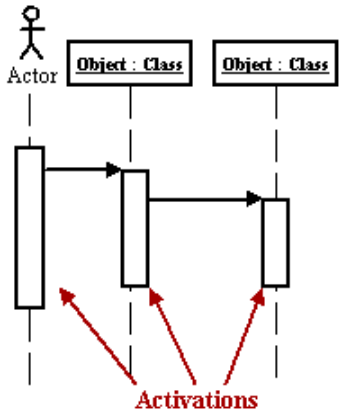


Owner Use Case Diagram

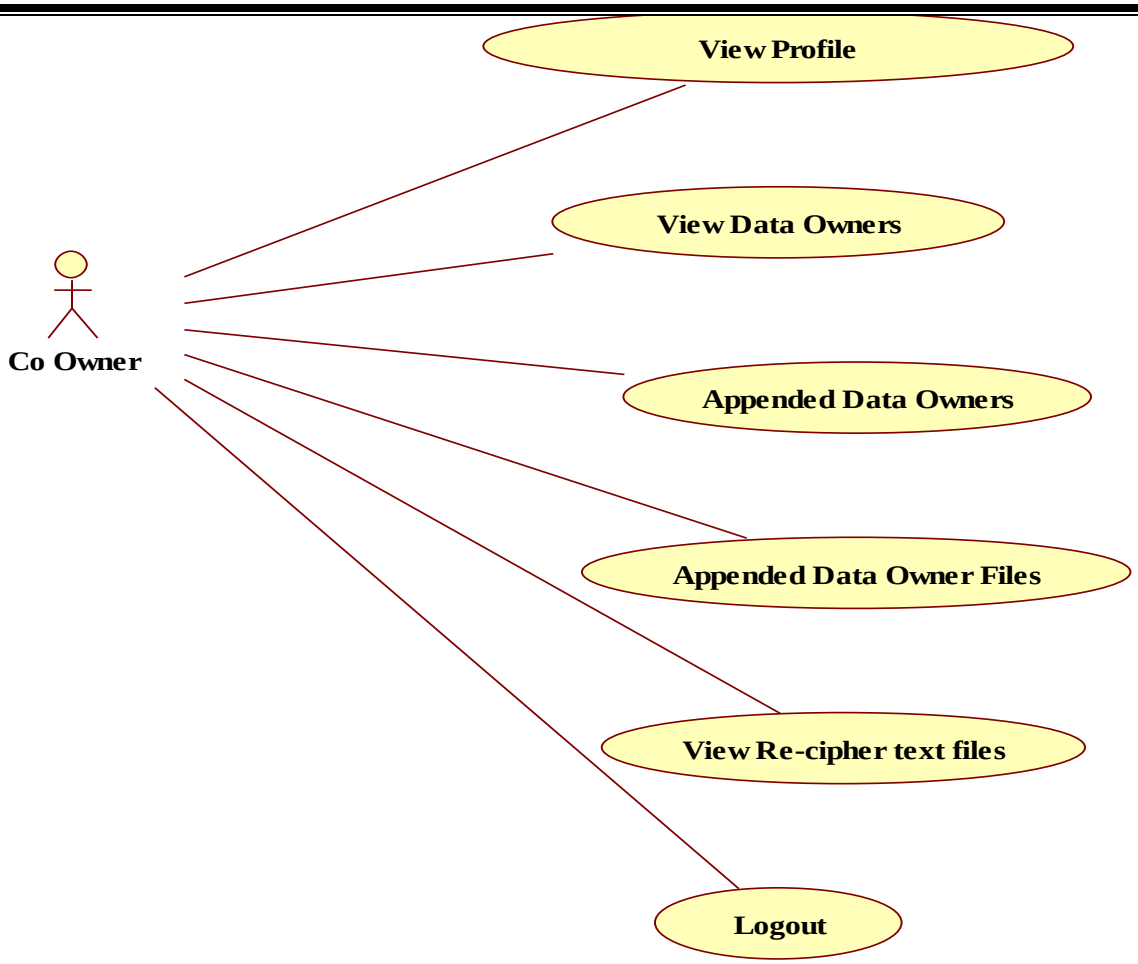


Activation

Activation boxes represent the time an object needs to complete a task.

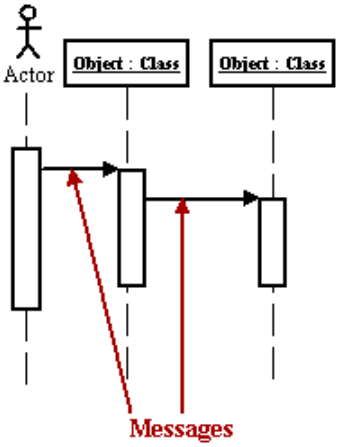


Co Owner Use Case Diagram



Messages

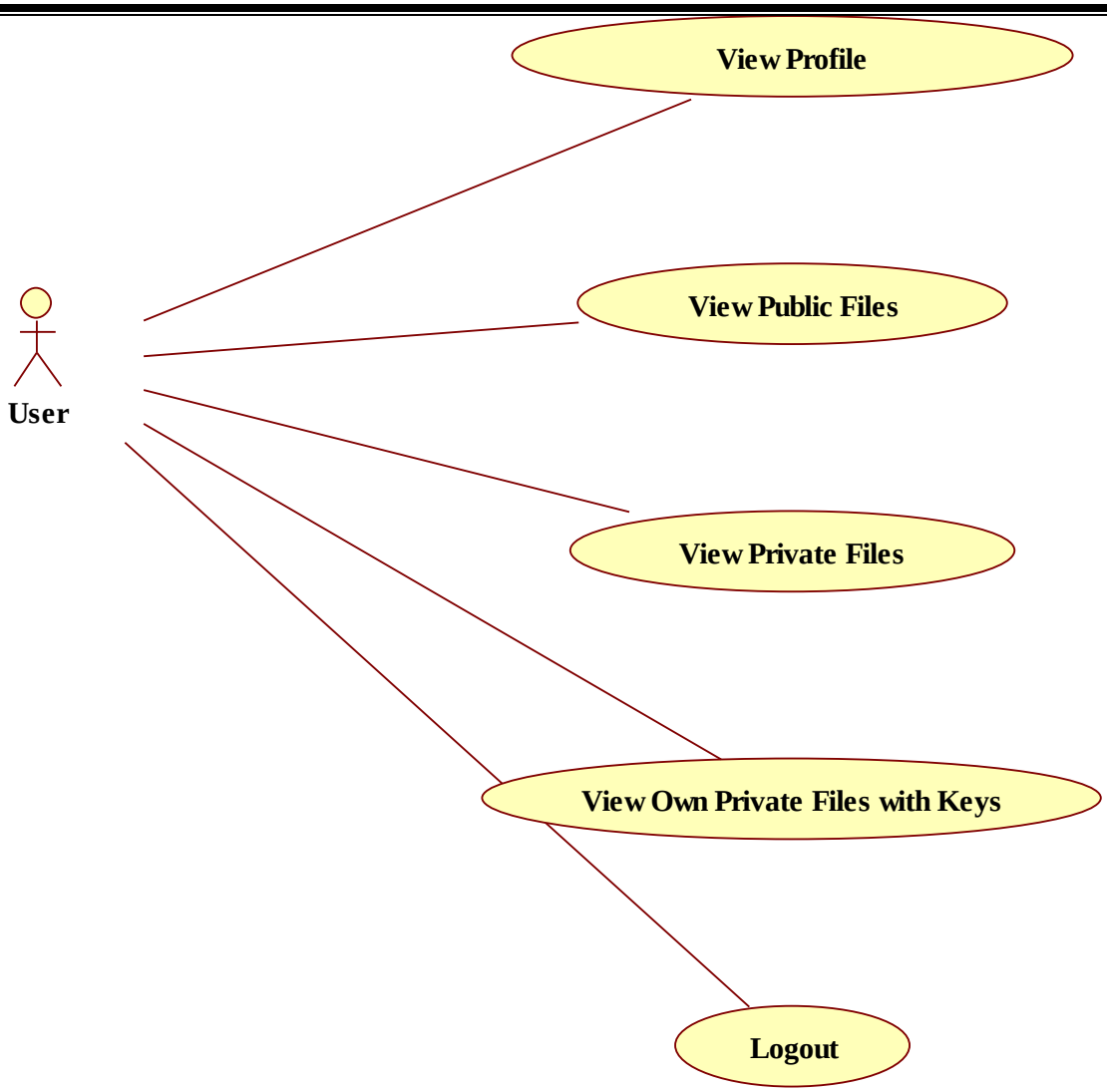
Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.



Arrow	Message type
	Simple
	Synchronous
	Asynchronous
	Balking
	Time out

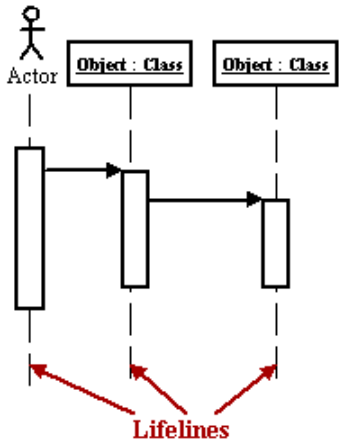
Various message types for Sequence and Collaboration diagrams

Data User Use Case Diagram



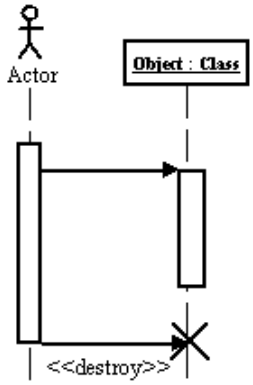
Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.



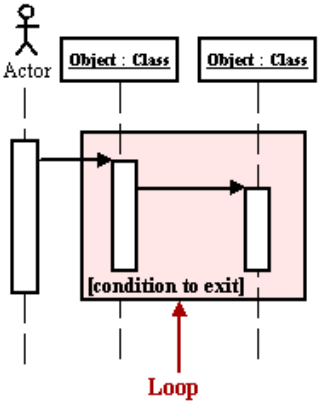
Destroying Object Objects can be terminated early using an arrow labeled "<< destroy >>" that points to

X

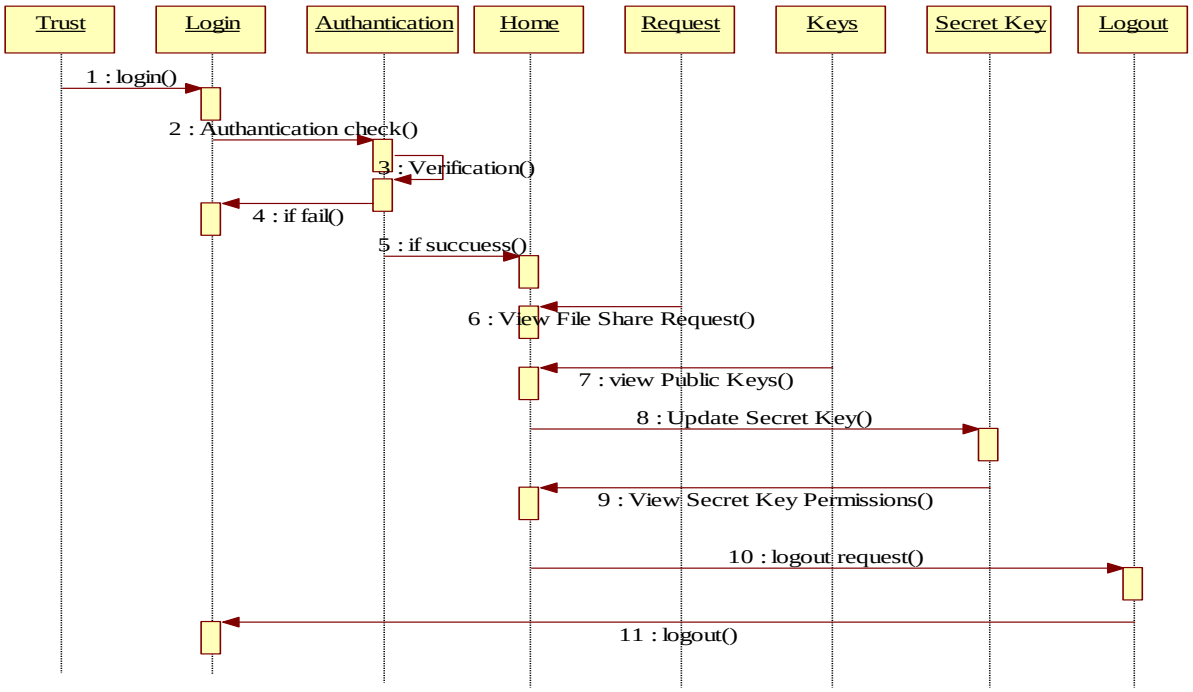


Loops

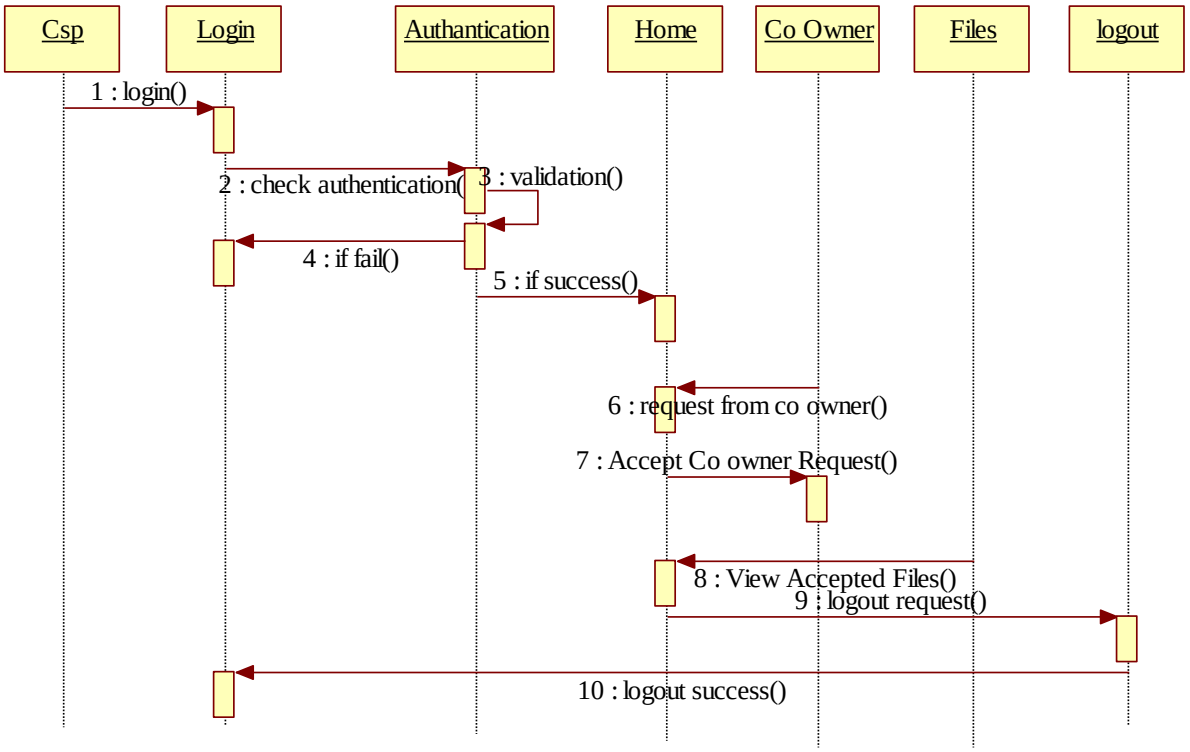
A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].



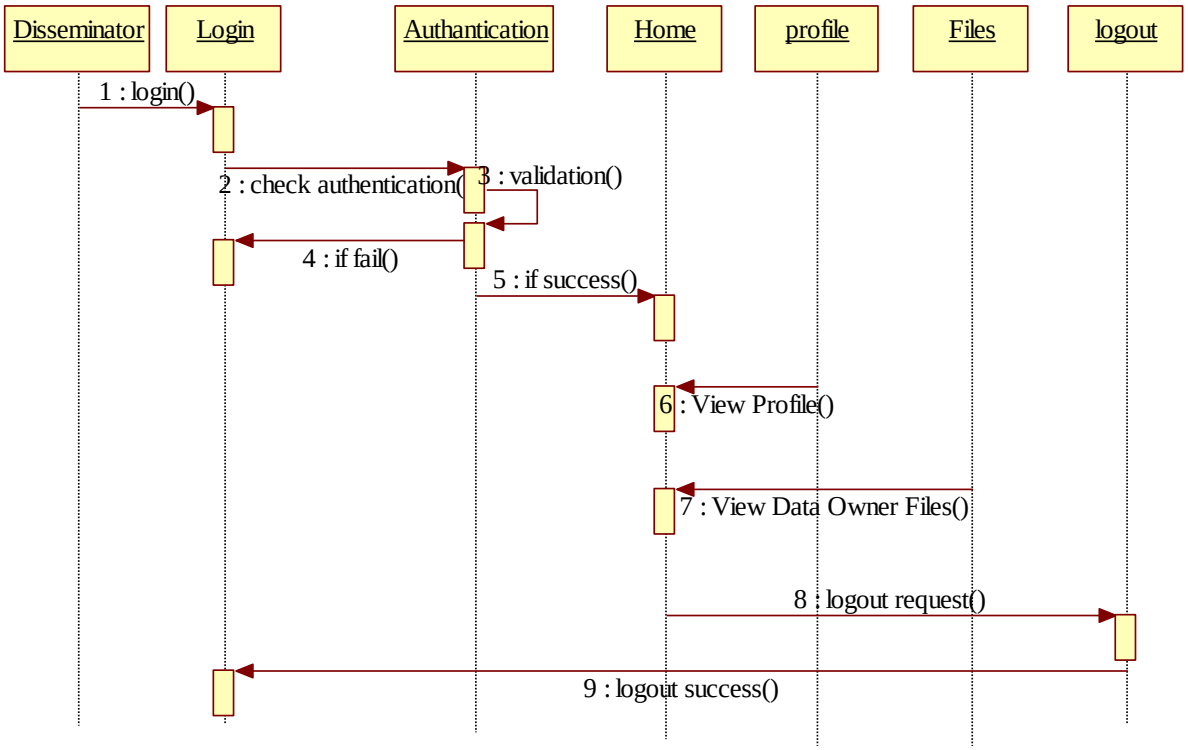
Trust Authority Sequence Diagram



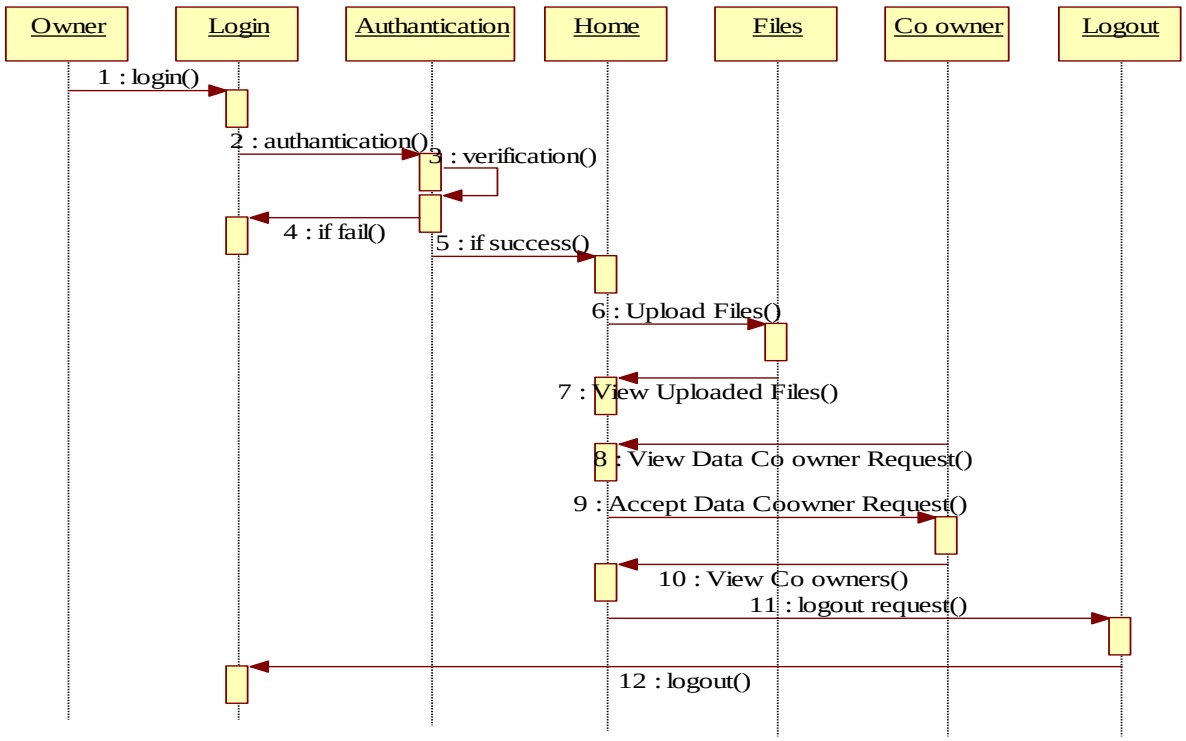
CSP Sequence Diagram



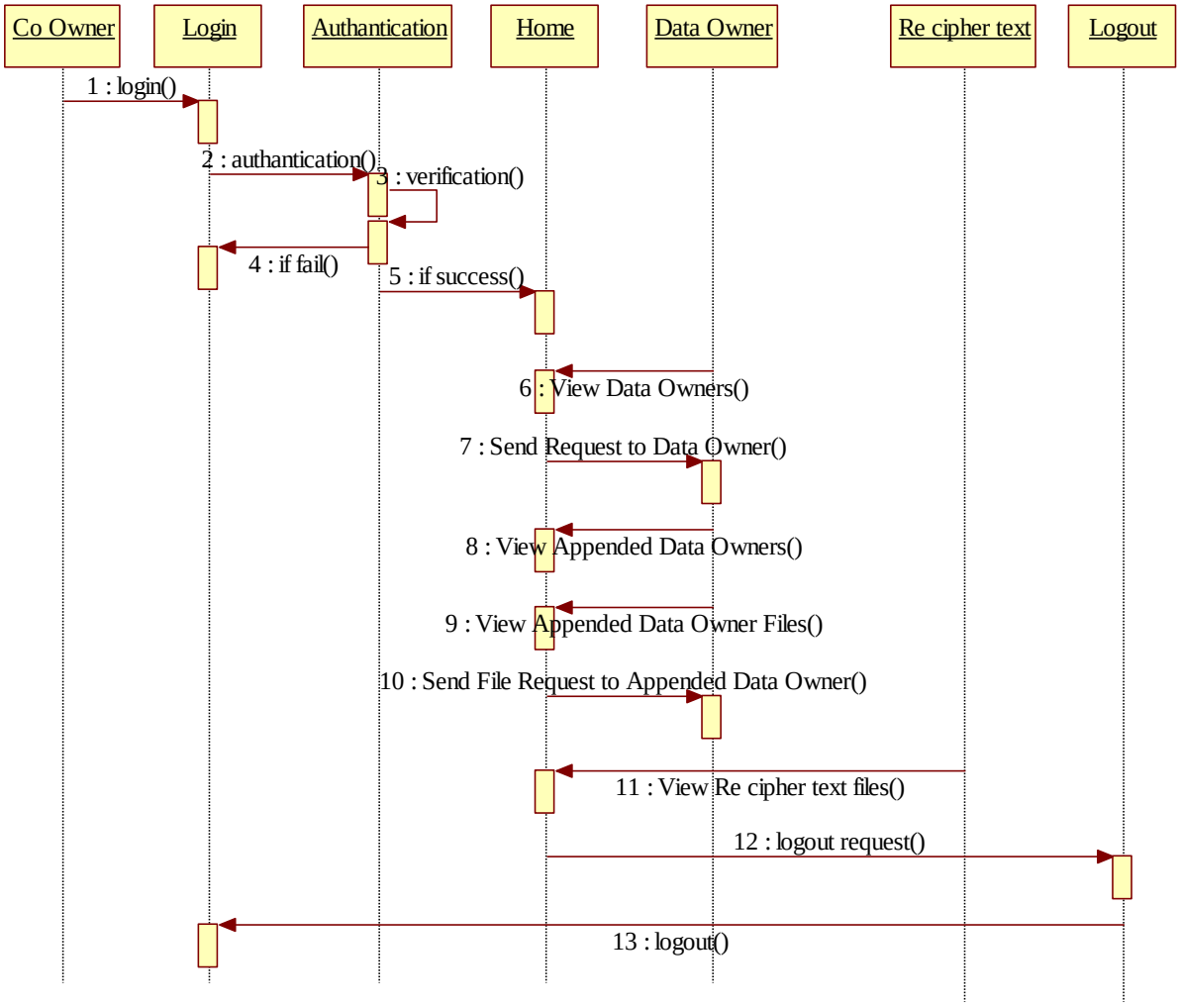
Disseminator Sequence Diagram



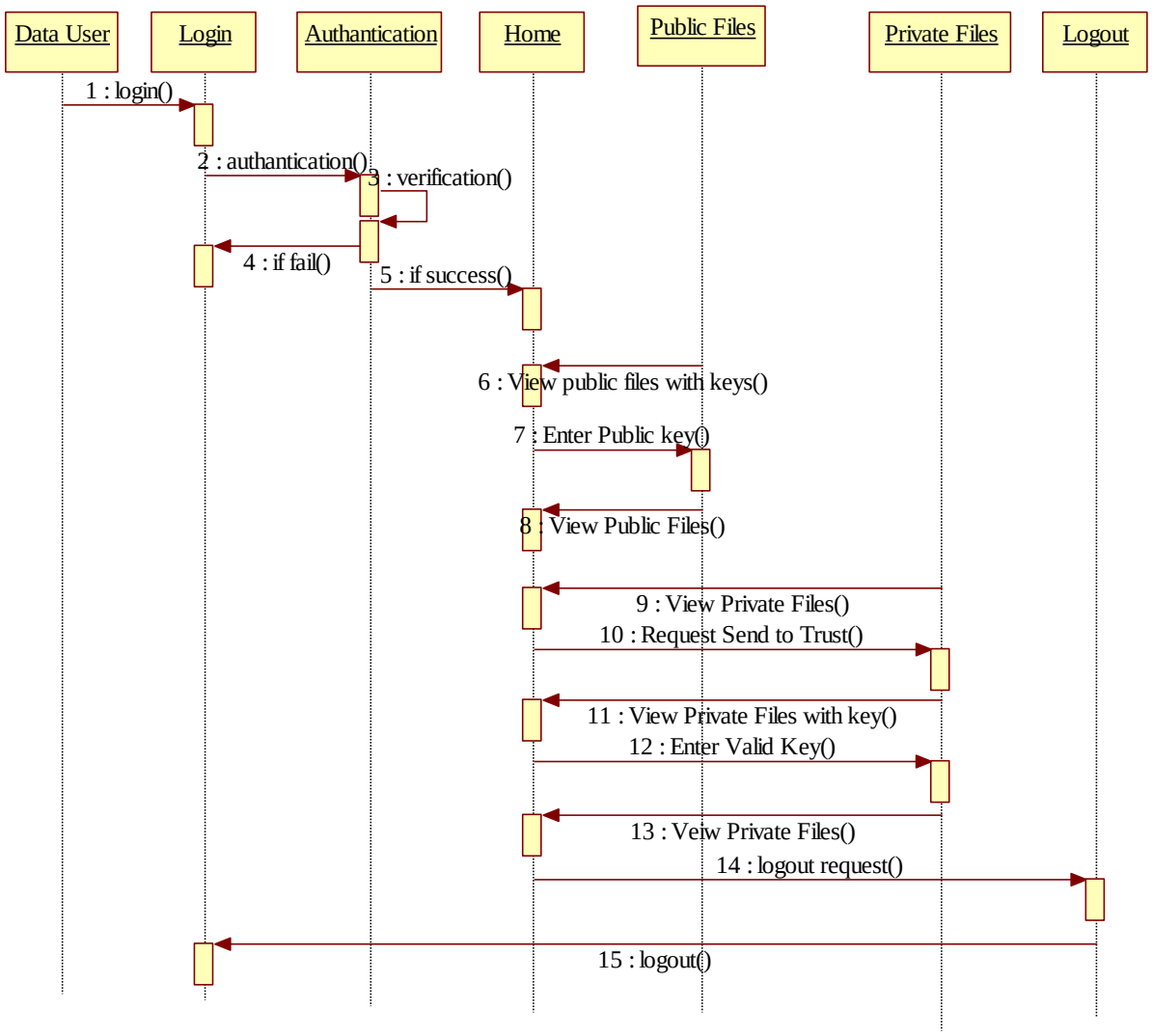
Data Owner Sequence Diagram



Data Co Owner Sequence Diagram



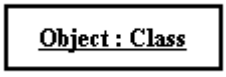
Data User Sequence Diagram



Collaboration Diagram

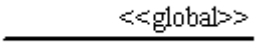
A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.

Basic Collaboration Diagram Symbols and NotationsClass roles Class roles describe how objects behave. Use the UML object symbol to illustrate class roles, but don't list object attributes.



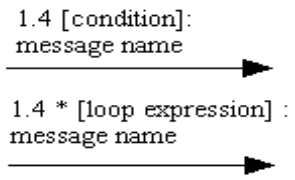
Association roles

Association roles describe how an association will behave given a particular situation. You can draw association roles using simple lines labeled with stereotypes.

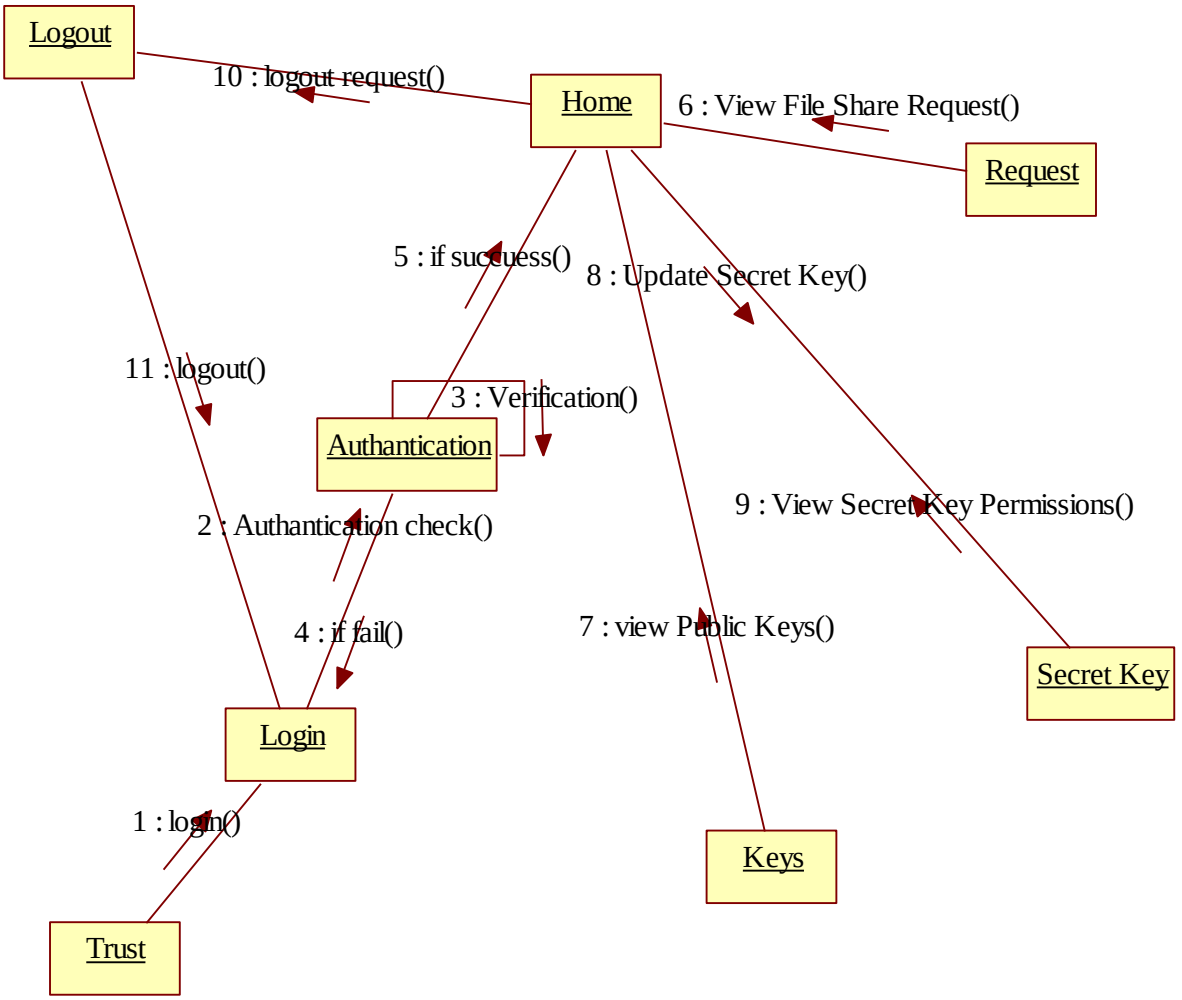


Messages

Unlike sequence diagrams, collaboration diagrams do not have an explicit way to denote time and instead number messages in order of execution. Sequence numbering can become nested using the Dewey decimal system. For example, nested messages under the first message are labeled 1.1, 1.2, 1.3, and so on. The a condition for a message is usually placed in square brackets immediately following the sequence number. Use a * after the sequence number to indicate a loop. Learn how to add arrows to your lines.

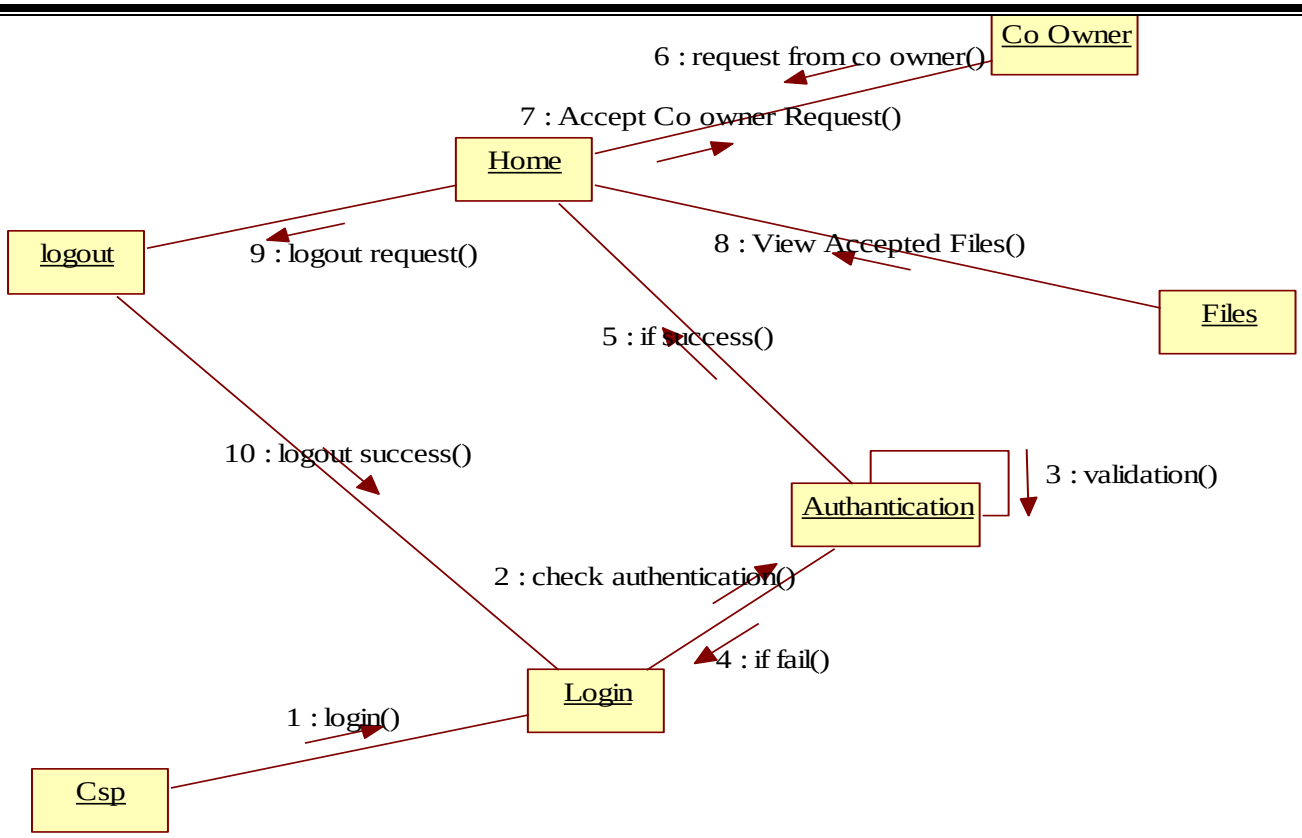


collaboration **Diagram**Trust Collaboration Diagram

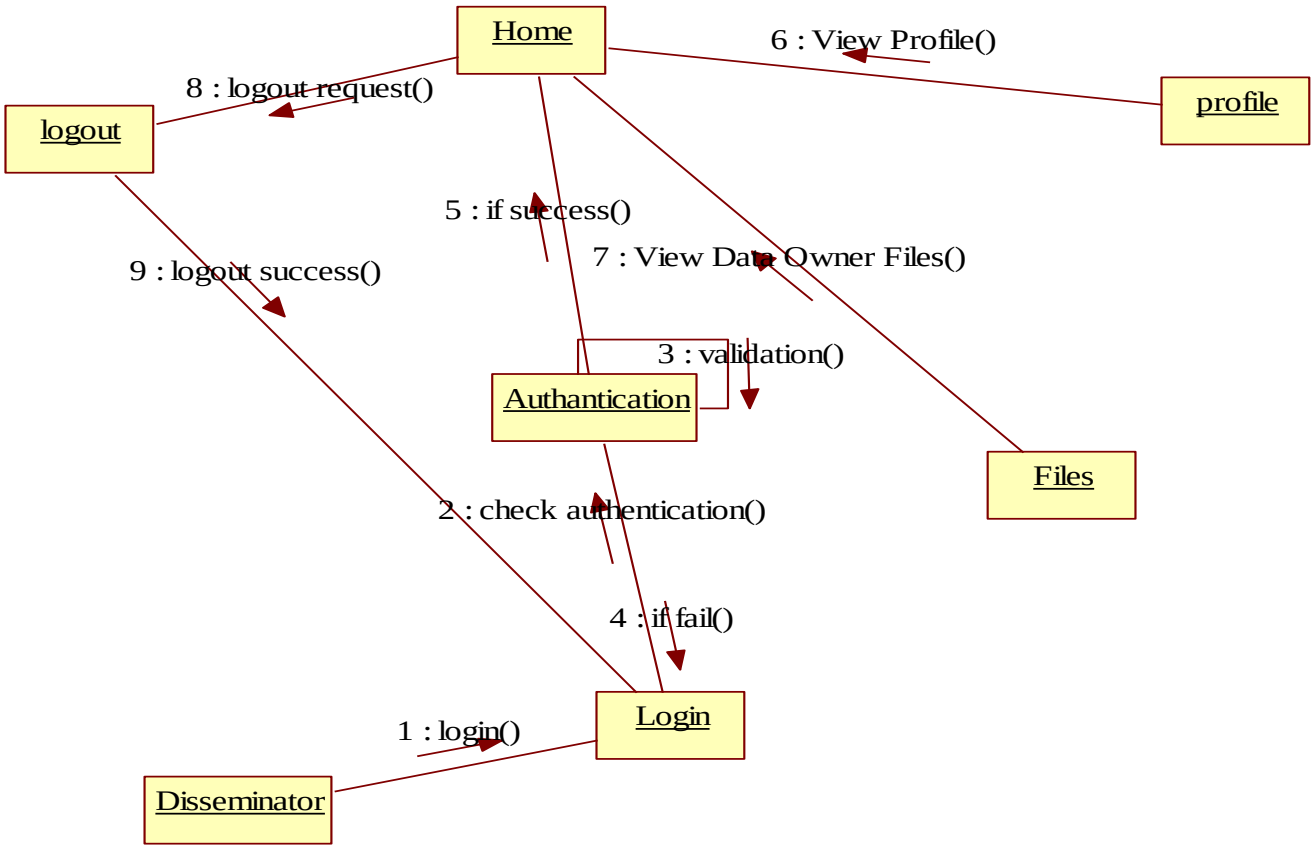


CSP

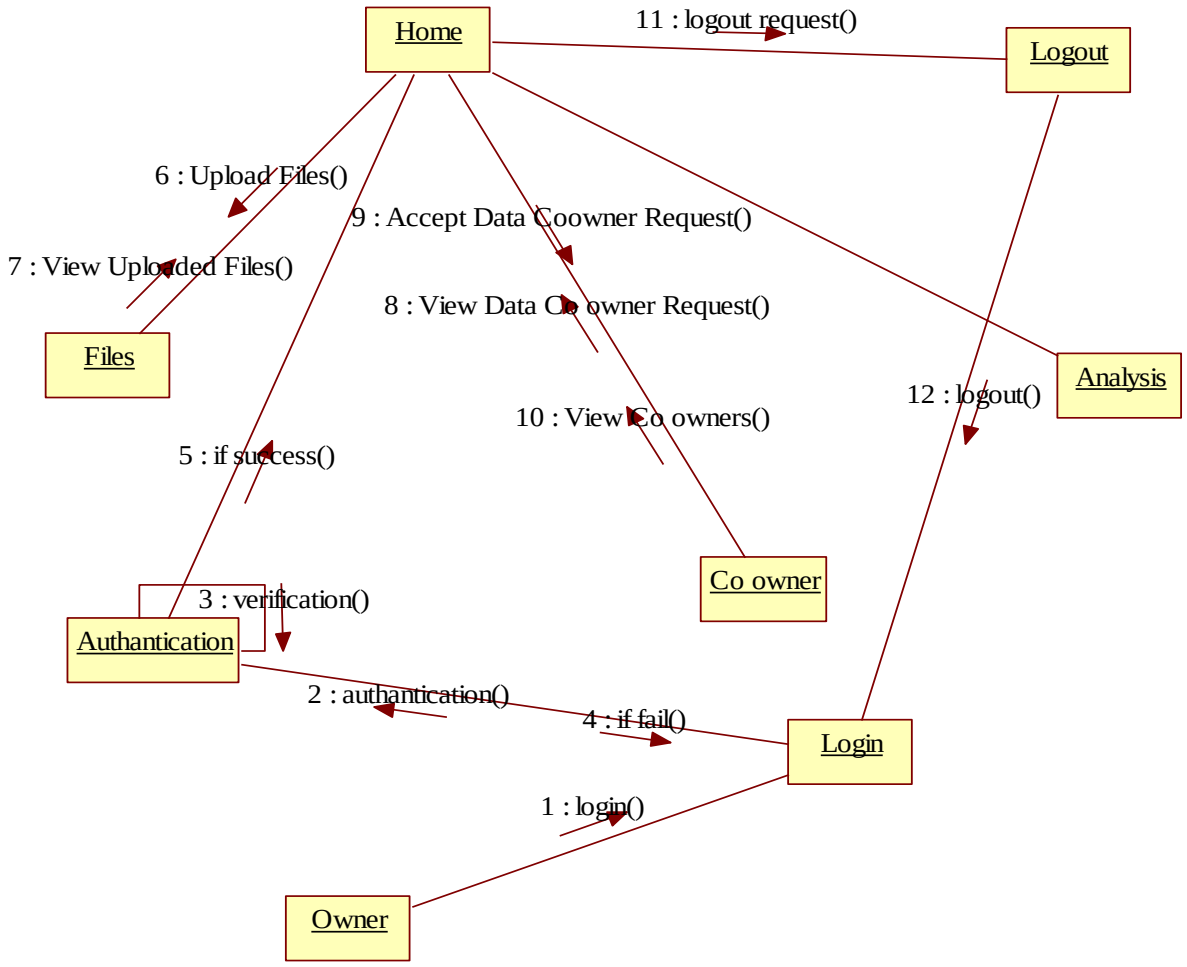
Collaboration Diagram



Data Disseminator Collaboration Diagram



Data Owner Collaboration Diagram



Activity Diagram

An activity diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation. Because an activity diagram is a special kind of state chart diagram, it uses some of the same modeling conventions.

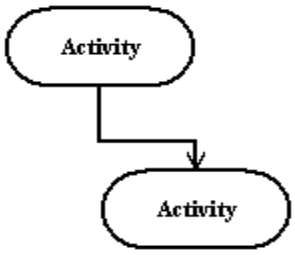
Basic Activity Diagram Symbols and Notations

Action states

Action states represent the non interruptible actions of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.

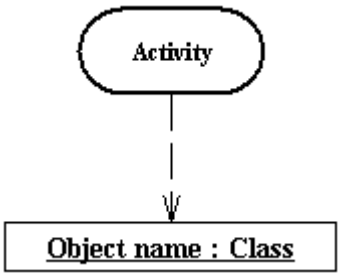
Action Flow

Action flow arrows illustrate the relationships among action states.



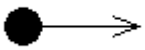
Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



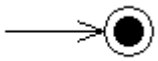
Initial State

A filled circle followed by an arrow represents the initial action state.



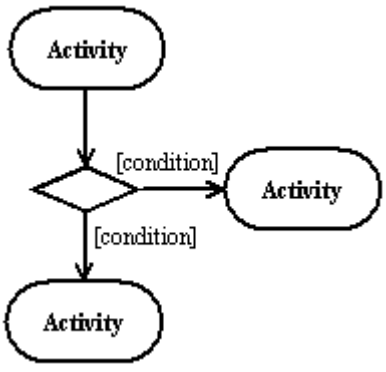
Final State

An arrow pointing to a filled circle nested inside another circle represents the final action state.



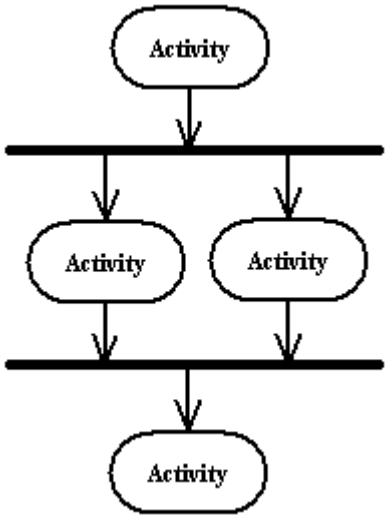
Branching

A diamond represents a decision with alternate paths. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



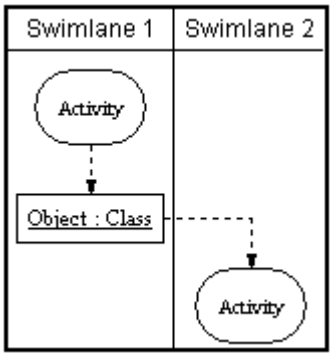
synchronization

A synchronization bar helps illustrate parallel transitions. Synchronization is also called forking and joining.



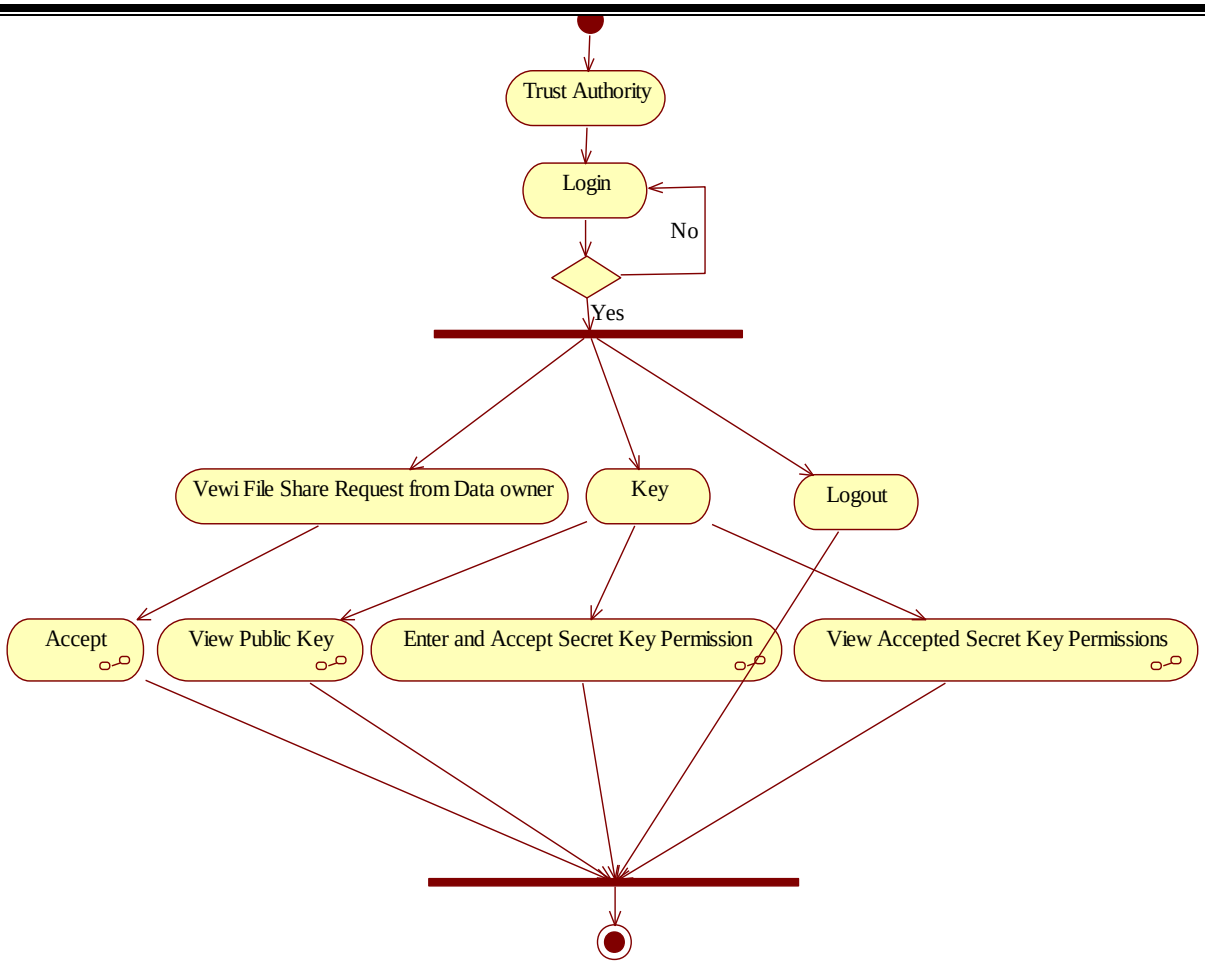
Swimlanes

Swimlanes group related activities into one column.

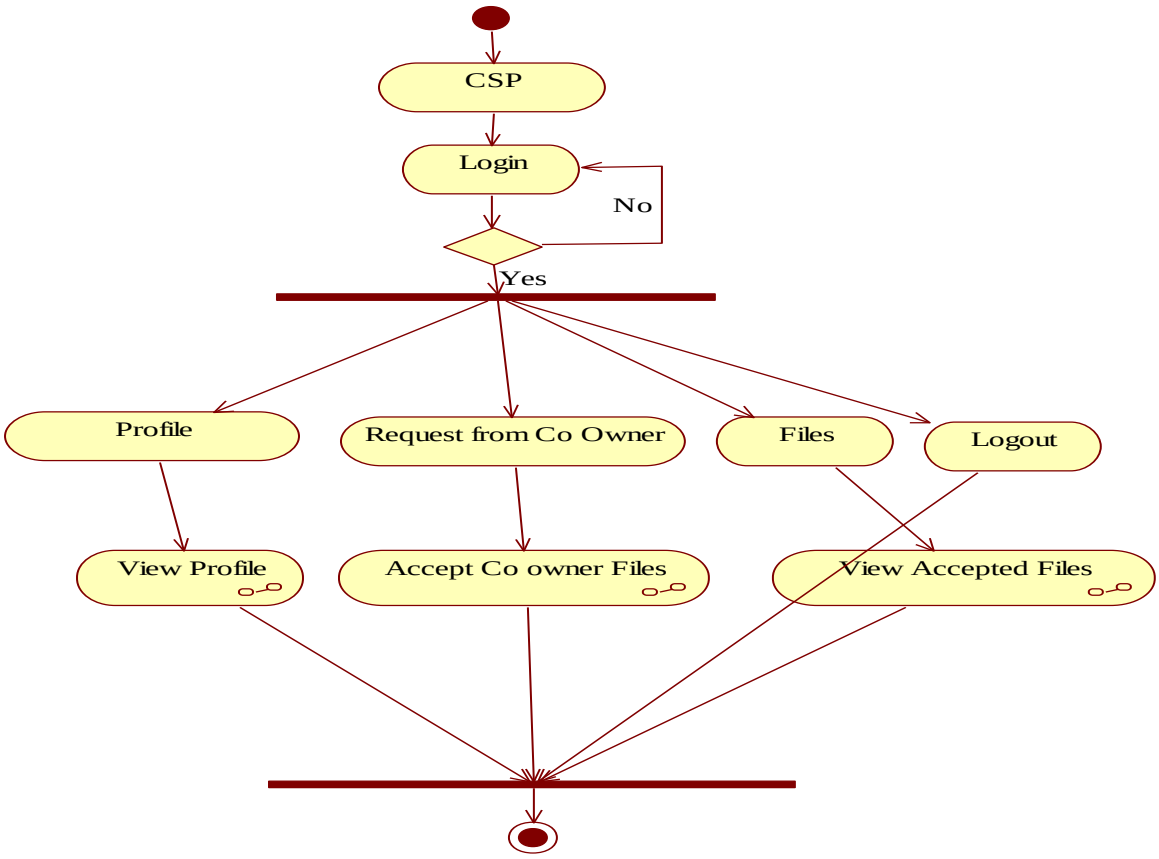


Activity Diagram

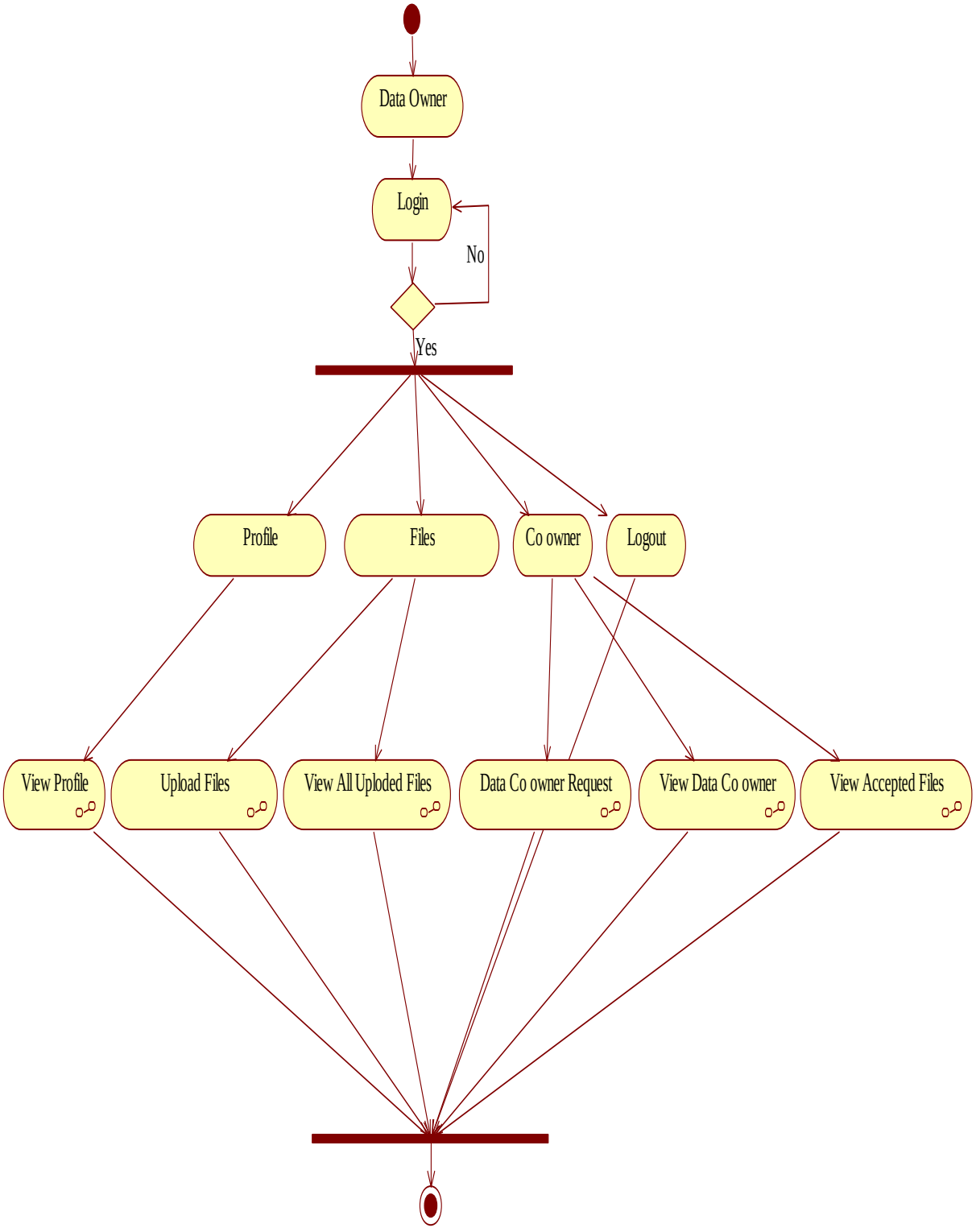
Trusted Authority Activity Diagram



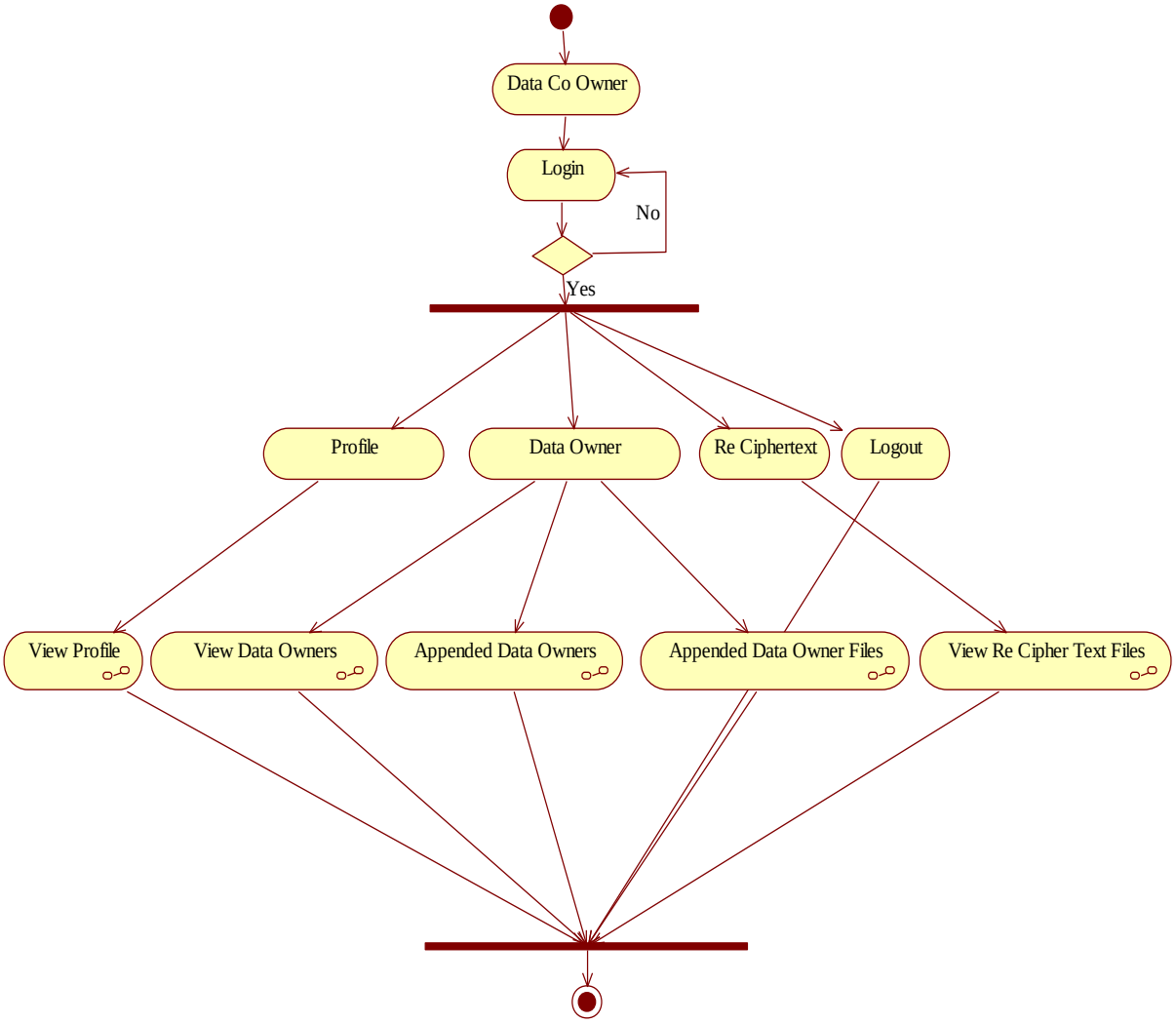
CSP Activity Diagram



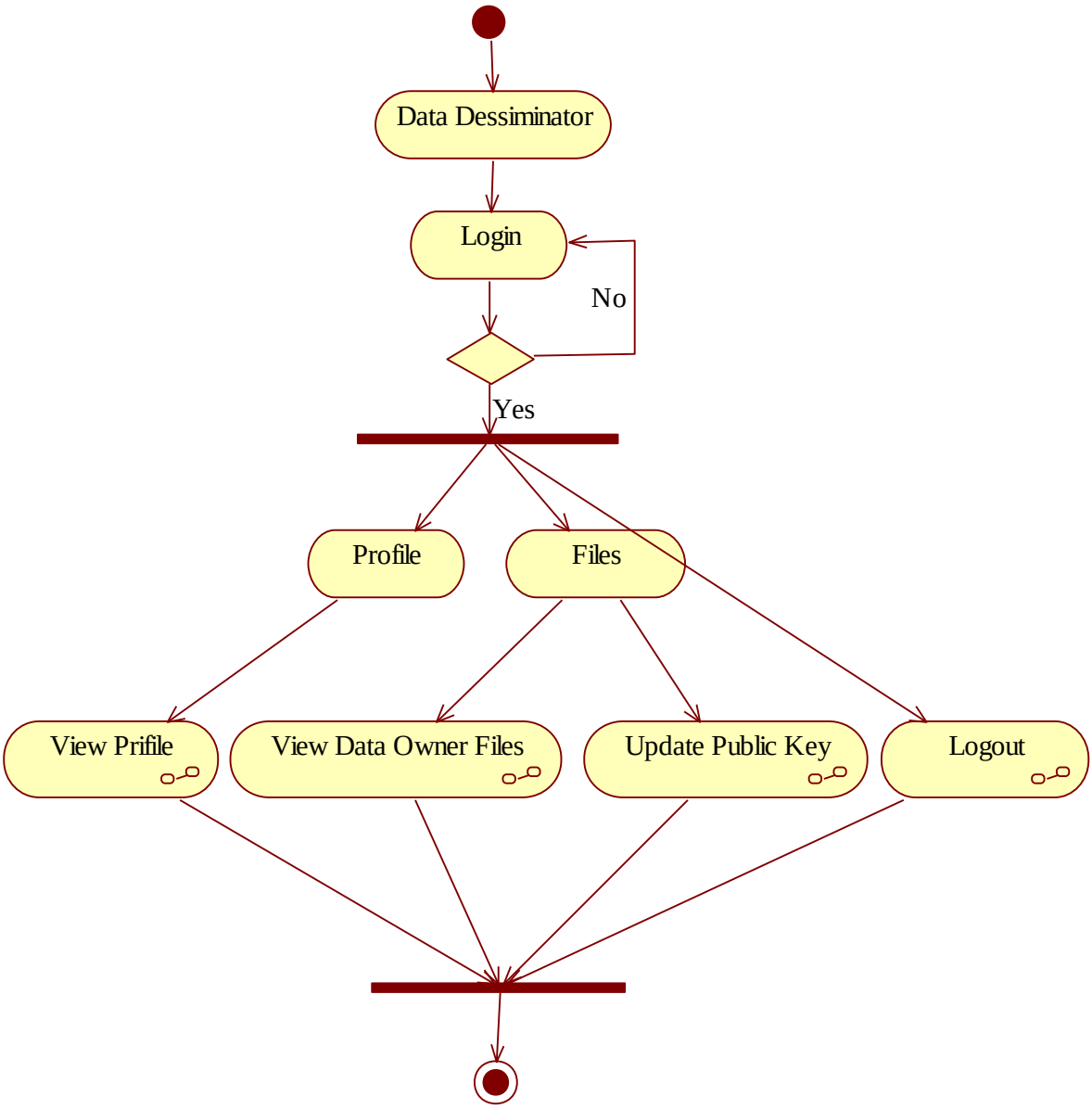
Data Owner Activity Diagram



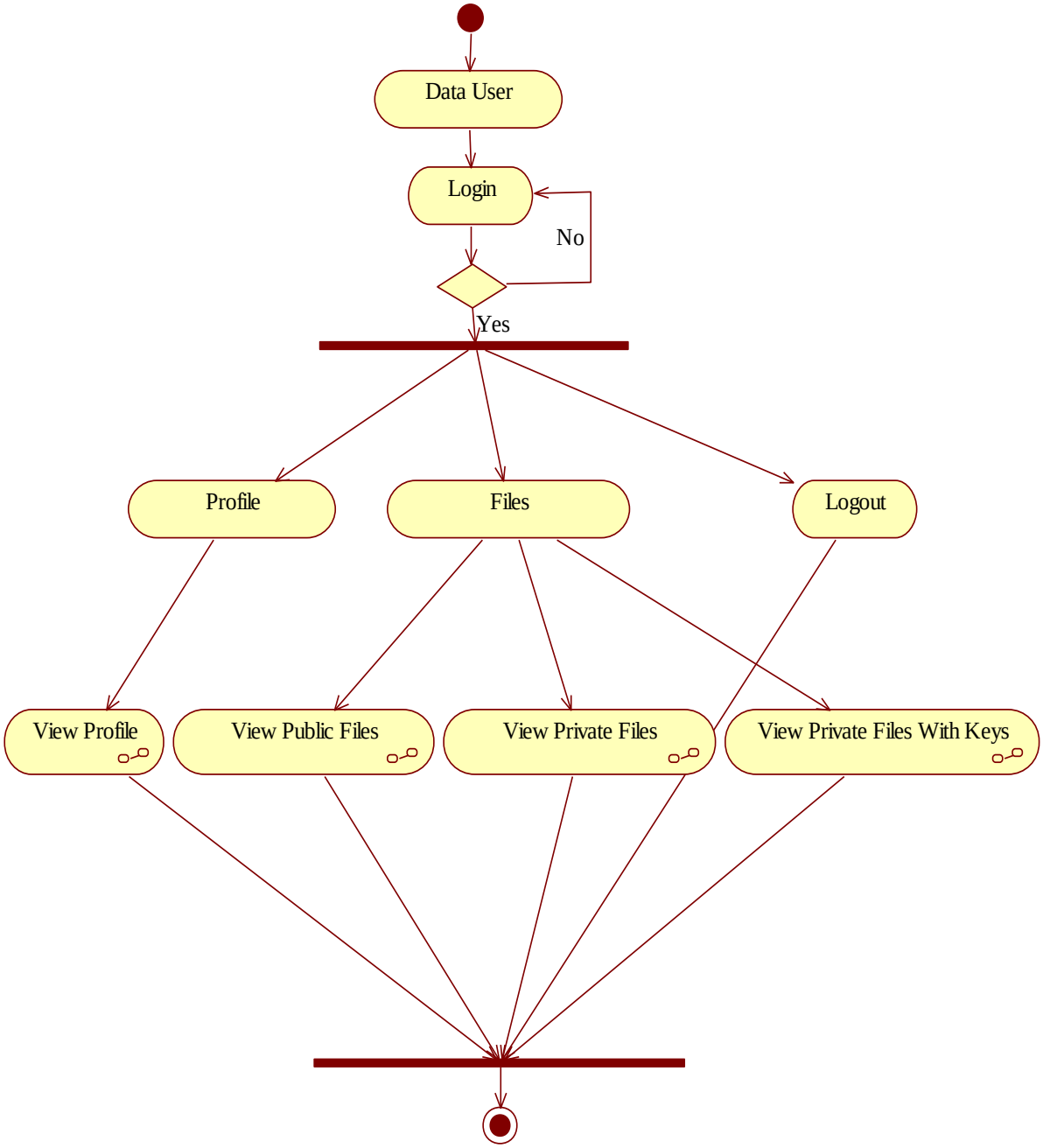
Data Co owner Activity Diagram



Data Disseminator Activity Diagram



Data User Activity Diagram



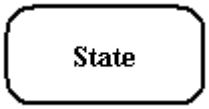
State chart Diagram

A state chart diagram shows the behavior of classes in response to external stimuli. This diagram models the dynamic flow of control from state to state within a system.

Basic State chart Diagram Symbols and Notations

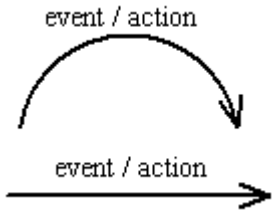
States

States represent situations during the life of an object. You can easily illustrate a state in Smart Draw by using a rectangle with rounded corners.



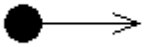
Transition

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it.



Initial State

A filled circle followed by an arrow represents the object's initial state.

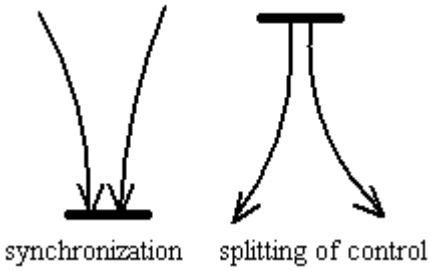


Final State

An arrow pointing to a filled circle nested inside another circle represents the object's final state.

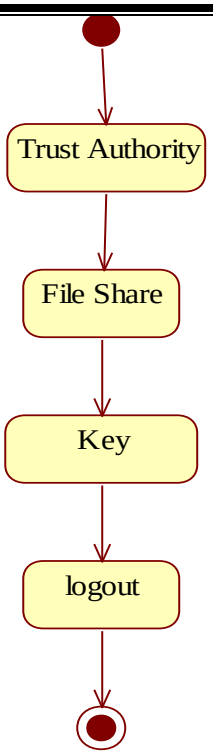
Synchronization and Splitting of Control

A short heavy bar with two transitions entering it represents a synchronization of control. A short heavy bar with two transitions leaving it represents a splitting of control that creates multiple states.

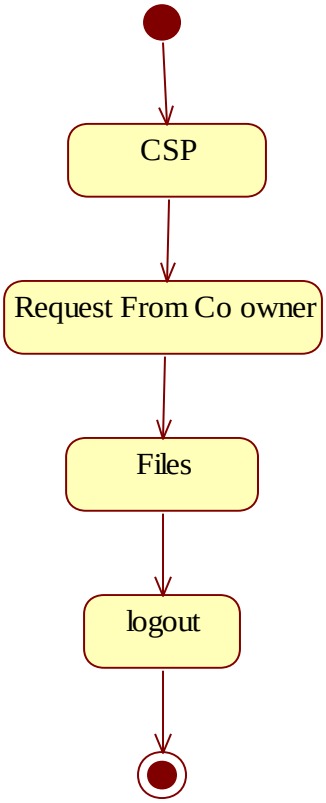


State Chart Diagram

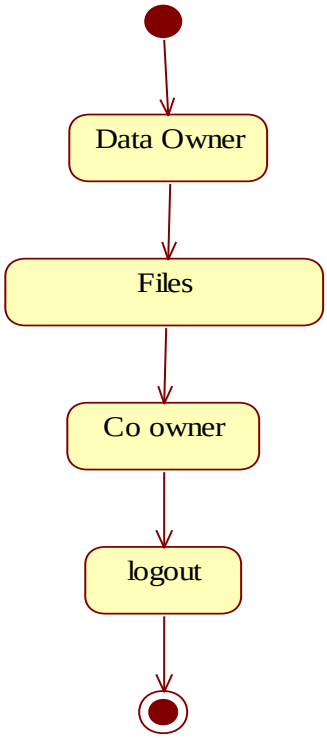
trust Authority State chart Diagram



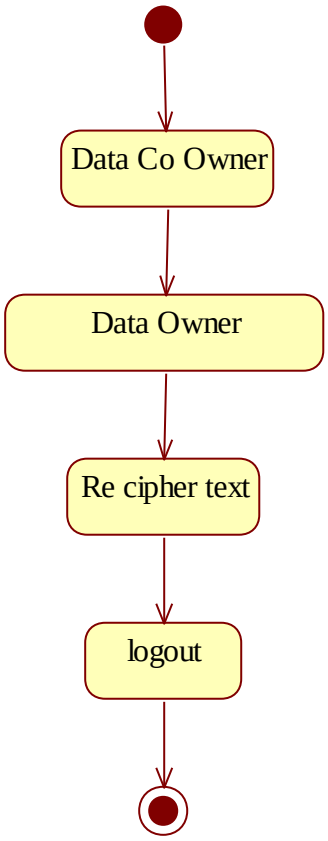
CSP State Chart Diagram



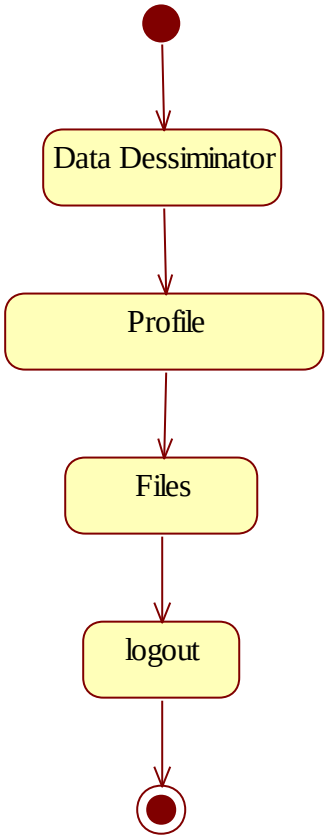
Data Owner State Chart Diagram



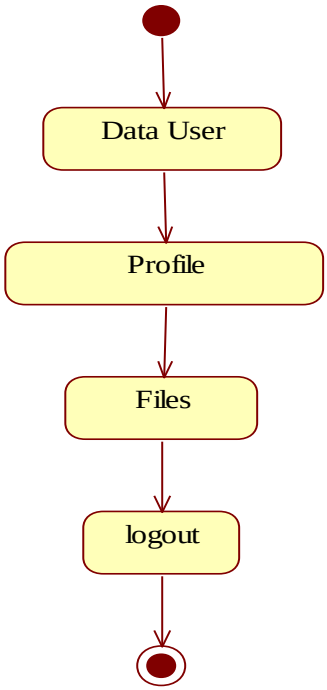
Data Co Owner State Chart Diagram



Data Disseminator State Chart Diagram



Data User State Chart Diagram



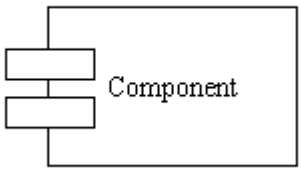
Component Diagram:

A component diagram describes the organization of the physical components in a system.

Basic Component Diagram Symbols and Notations

Component

A component is a physical building block of the system. It is represented as a rectangle with tabs. Learn how to resize grouped objects like components.

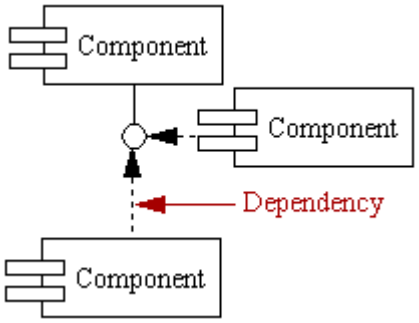


Interface

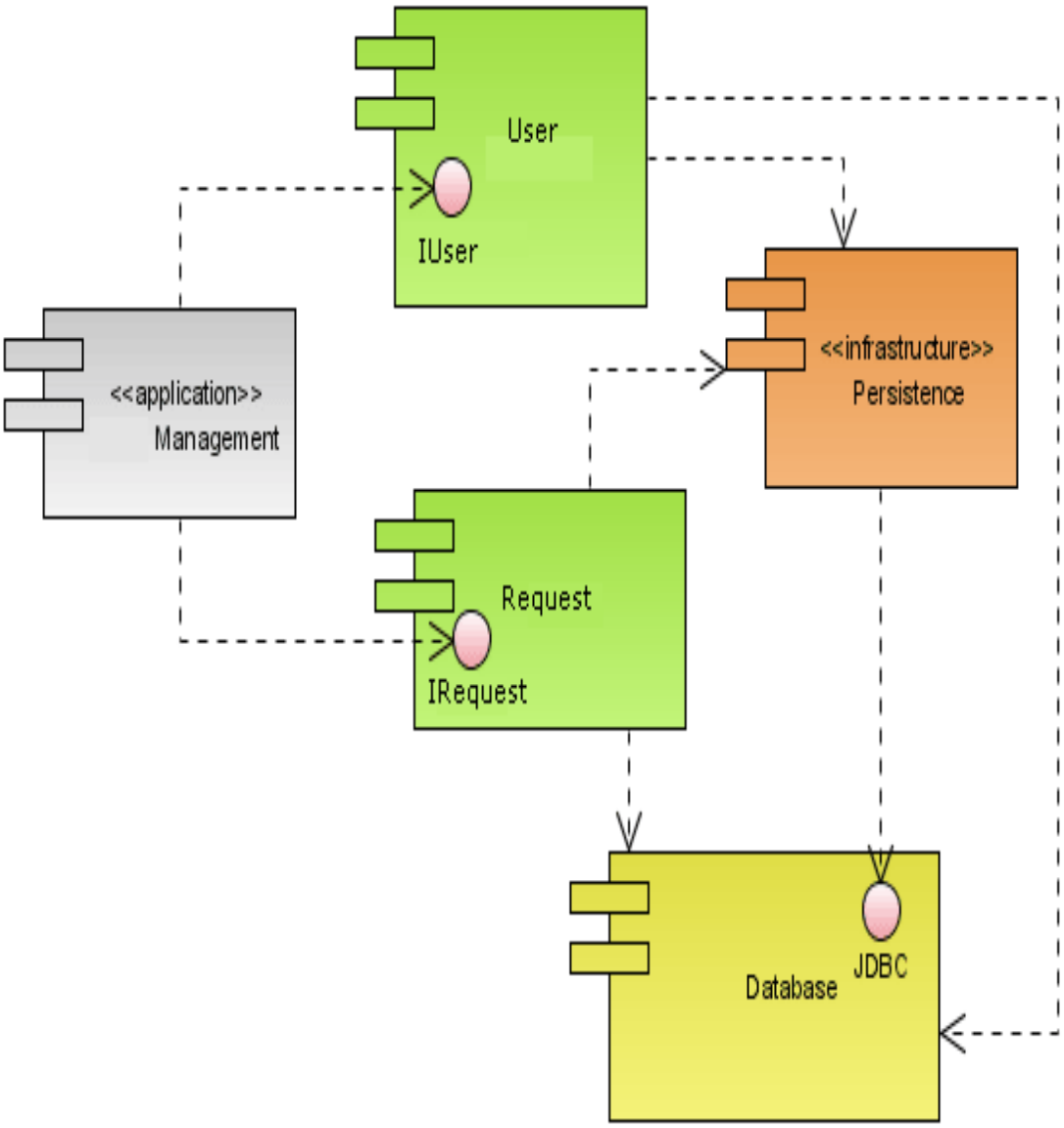
An interface describes a group of operations used or created by components.

Dependencies

Draw dependencies among components using dashed arrows. Learn about line styles in SmartDraw.



COMPONENT DIAGRAM:



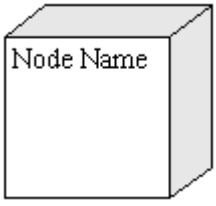
Deployment Diagram:

Deployment diagrams depict the physical resources in a system including nodes, components, and connections.

Basic Deployment Diagram Symbols and Notations

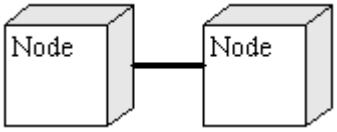
Component

A node is a physical resource that executes code components.
Learn how to resize grouped objects like nodes.



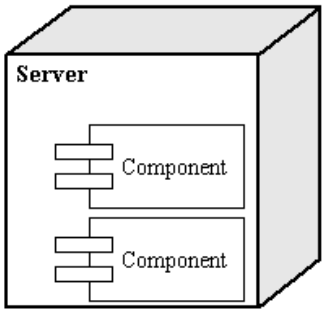
Association

Association refers to a physical connection between nodes, such as Ethernet. Learn how to connect two nodes.

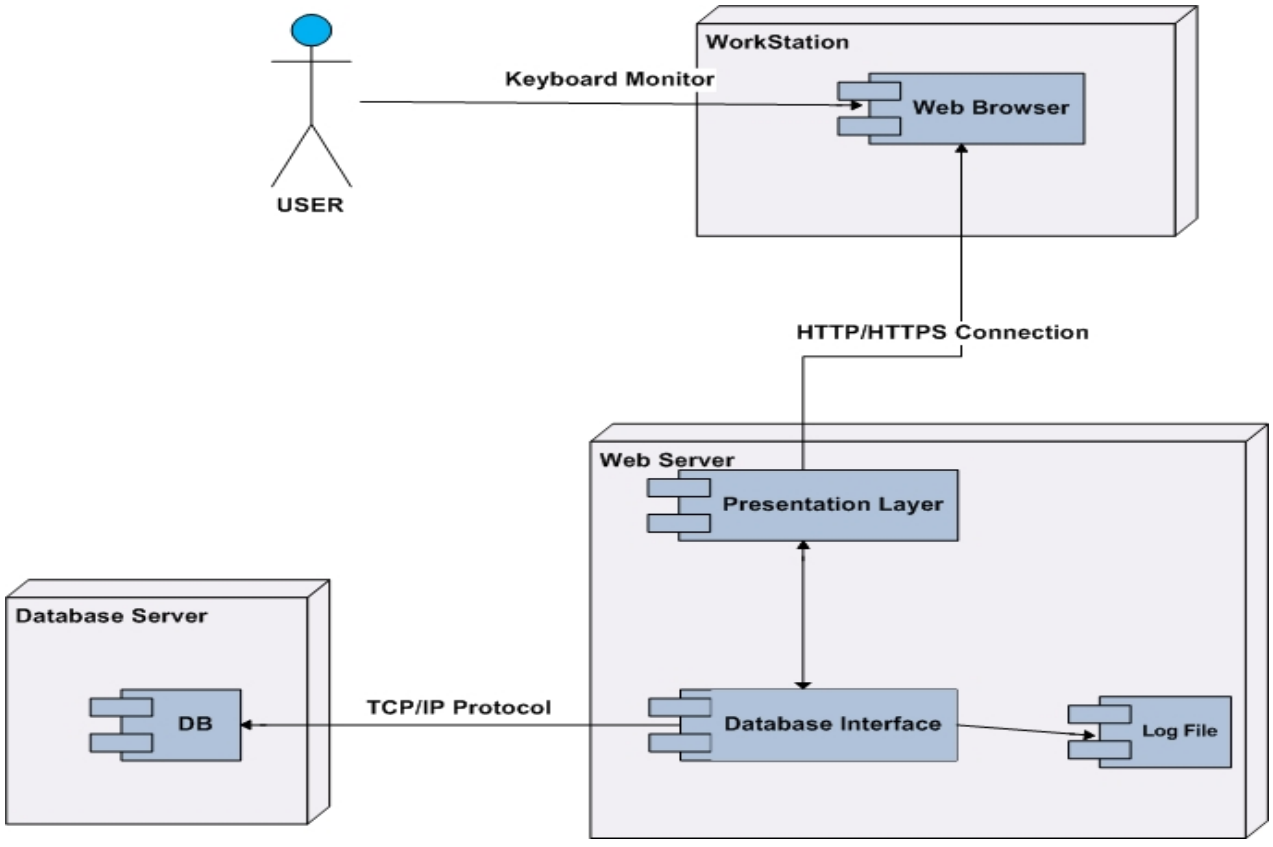


Components and Nodes

Place components inside the node that deploys them.



DEPLOYMENT DIAGRAM:



6.5Data dictionary

User Details

Column Name	Data Type	Nullable	Default	Primary Key
USERID	NUMBER	No	-	1
USERNAME	VARCHAR2(4000)	Yes	-	-
PASSWORD	VARCHAR2(4000)	Yes	-	-
EMAIL	VARCHAR2(4000)	Yes	-	-
MOBILE	NUMBER	Yes	-	-
DOD	DATE	Yes	-	-
UTYPE	VARCHAR2(4000)	Yes	-	-
ADDRESS	VARCHAR2(4000)	Yes	-	-
				1 - 8

UFiles

Column Name	Data Type	Nullable	Default	Primary Key
USERID	NUMBER	Yes	-	-
UPLODEDBY	VARCHAR2(4000)	Yes	-	-
FILENAME	VARCHAR2(4000)	Yes	-	-
UFILE	BLOB	Yes	-	-
CONTENT	VARCHAR2(4000)	Yes	-	-
STATUS	VARCHAR2(4000)	Yes	-	-
FID	NUMBER	Yes	-	-
POLICY	VARCHAR2(4000)	Yes	-	-
PUBLICKEY	NUMBER	Yes	-	-
				1 - 9

File Request

Column Name	Data Type	Nullable	Default	Primary Key
USERID	NUMBER	Yes	-	-
REQUESTEDBY	VARCHAR2(4000)	Yes	-	-
FID	NUMBER	Yes	-	-
FNAME	VARCHAR2(4000)	Yes	-	-
KEY	VARCHAR2(4000)	Yes	-	-
				1 - 5

Ratings

Column Name	Data Type	Nullable	Default	Primary Key
USERID	NUMBER	Yes	-	-
MOVIEID	NUMBER	Yes	-	-
RATING	VARCHAR2(4000)	Yes	-	-
TIMESTAMP	NUMBER	Yes	-	-
				1 - 4

Co Owner Files

Column Name	Data Type	Nullable	Default	Primary Key
DOWNERID	NUMBER	Yes	-	-
DOWNERNAME	VARCHAR2(4000)	Yes	-	-
FID	NUMBER	Yes	-	-
FNAME	VARCHAR2(4000)	Yes	-	-
CONTENT	VARCHAR2(4000)	Yes	-	-
ACCESSPOLICY	VARCHAR2(4000)	Yes	-	-
REQUESTEDBY	VARCHAR2(4000)	Yes	-	-
STATUS	VARCHAR2(4000)	Yes	-	-
REQUESTEDID	VARCHAR2(4000)	Yes	-	-
				1 - 9

Appended Users

Column Name	Data Type	Nullable	Default	Primary Key
OWNERID	NUMBER	Yes	-	-
COOWNERID	NUMBER	Yes	-	-
OWNERNAME	VARCHAR2(4000)	Yes	-	-
COOWNERNAME	VARCHAR2(4000)	Yes	-	-
STATUS	VARCHAR2(4000)	Yes	-	-
				1 - 5

Movies1

Column Name	Data Type	Nullable	Default	Primary Key
MOVIEID	NUMBER	No	-	1
TITLE	VARCHAR2(4000)	Yes	-	-
GENRES	VARCHAR2(4000)	Yes	-	-
MYEAR	VARCHAR2(16)	Yes	-	-
				1 - 4

Links1

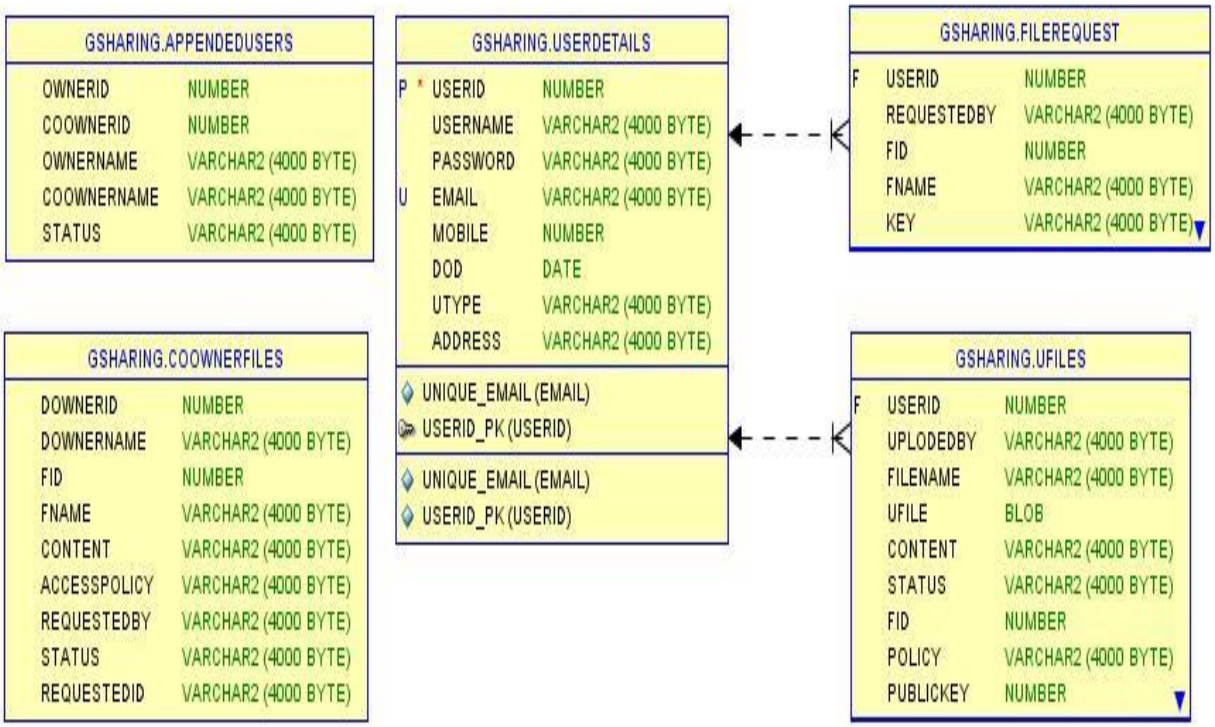
Column Name	Data Type	Nullable	Default	Primary Key
MOVIEID	NUMBER	Yes	-	-
IMDBID	NUMBER	Yes	-	-
TMDBID	NUMBER	Yes	-	-
				1 - 3

Comment_Tab

Column Name	Data Type	Nullable	Default	Primary Key
USER_ID	VARCHAR2(4000)	Yes	-	-
FILMNAME	VARCHAR2(4000)	Yes	-	-
RATINGS	VARCHAR2(4000)	Yes	-	-
COMMENTS	VARCHAR2(4000)	Yes	-	-
MOVIETYPE	VARCHAR2(4000)	Yes	-	-
				1 - 5

ER Diagrams

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.



7. Implementation

7.1 Technology Description

About the Java Technology

The Java platform consists of the Java application programming interfaces (APIs) and the Java virtual machine (JVM).

The following Java technology lets developers, designers, and business partners develop and deliver a consistent user experience, with one environment for applications on mobile and embedded devices. Java meshes the power of a rich stack with the ability to deliver customized experiences across such devices.

Java APIs are libraries of compiled code that you can use in your programs. They let you add ready-made and customizable functionality to save you programming time. Java programs are run (or interpreted) by another program called the Java Virtual Machine. Rather than running directly on the native operating system, the program is interpreted by the Java VM for the native operating system. This means that any computer system with the Java VM installed can run Java programs regardless of the computer system on which the applications were originally developed.

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris TM Operating System (Solaris OS), Linux, or Mac OS.

Java technology is both a programming language and a platform.

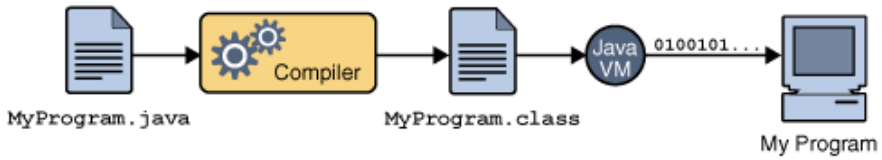
The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Object oriented
- Distributed
- Multithreaded
- Dynamic
- Architecture neutral
- Portable
- High performance
- Robust
- Secure

Each of the preceding buzzwords is explained in *The Java Language Environment* , a white paper written by James Gosling and Henry McGilton.

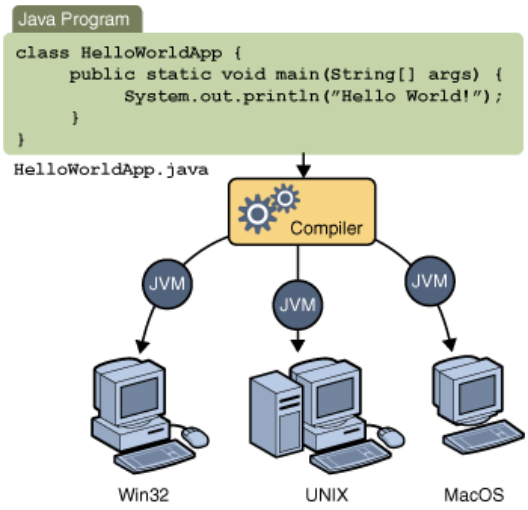
In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains *bytecodes* — the machine language of the Java Virtual Machine[1] (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.



An overview of the software development process.

An overview of the software development process.

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



Through the Java VM, the same application is capable of running on multiple platforms.

Servlet and JSP technology

Servlet and JSP technology has become the technology of choice for developing online stores, interactive

A Servlet's Job

Servlets are Java programs that run on Web or application servers, acting as a middle layer between requests coming from Web browsers or other HTTP clients and databases or applications on the HTTP server. Their job is to perform the following tasks,

as illustrated in Figure 1–1.

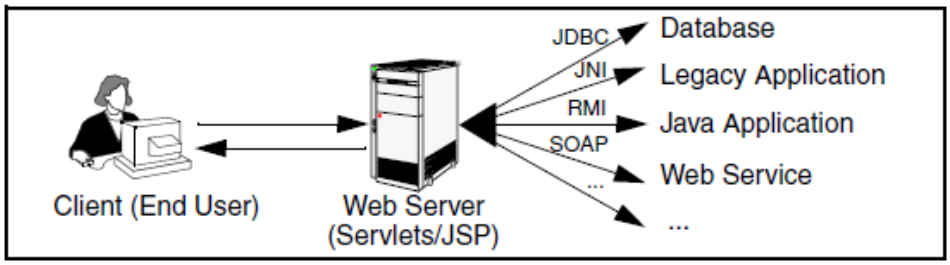


Figure 1–1 The role of Web middleware.

1. **Read the explicit data sent by the client.**

The end user normally enters this data in an HTML form on a Web page. However, the data could also come from an applet or a custom HTTP client program. Chapter 4 discusses how servlets read this data.

2. **Read the implicit HTTP request data sent by the browser.**

Figure 1–1 shows a single arrow going from the client to the Web server (the layer where servlets and JSP execute), but there are really *two* varieties of data: the explicit data that the end user enters in a form and the behind-the-scenes HTTP information.. The HTTP information includes cookies, information about media types and compression schemes the browser understands,

3. **Generate the results.**

This process may require talking to a database, executing an RMI or EJB call, invoking a Web service, or computing the response directly. Your real data may be in a relational database. Fine. But your database probably doesn't speak HTTP or return results in HTML,. You need the Web middle layer to extract the incoming data from the HTTP.

4. **Send the explicit data (i.e., the document) to the client.**

This document can be sent in a variety of formats, including text (HTML or XML), binary or even a compressed format like gzip that is layered on top of some other underlying format. But, HTML is by far the most common format, so an important servlet/JSP task is to wrap the results inside of HTML.

5. **Send the implicit HTTP response data.**

shows a single arrow going from the Web middle layer (the servlet or JSP page) to the client. But, there are really *two* varieties of data sent: the document itself and the behind-the-scenes HTTP information. Again, both varieties are critical to effective development. Sending HTTP response data involves telling the browser or other client what type of document is being returned setting cookies and caching parameters

The Advantages of Servlets Over "Traditional" CGI

Java servlets are more efficient, easier to use, more powerful, more portable, safer, and cheaper than traditional CGI and many alternative CGI-like technologies. With traditional CGI, a new process is started for each HTTP request. If the CGI program itself is relatively short, the execution time. With servlets, not a heavyweight operating system process.. With servlets, however, there would be *N* threads, but only a single copy of the servlet class would be loaded. This approach reduces server, the program terminates. This approach makes it difficult to cache computations,. Servlets remain in memory even after they complete a response, so it is straightforward to store arbitrarily complex data between client requests.

Convenient

Servlets have an extensive infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, tracking sessions, and many other such high-level utilities. In CGI, you have to do much of this yourself. that Java technology makes for more reliable and reusable code than does Visual Basic, VBScript, or C

Powerful

Servlets support several capabilities that are difficult or impossible to accomplish with regular CGI. Servlets can talk directly to the Web server, whereas regular CGI programs cannot, at least not without using a server-specific API. into concrete path names, for instance. Multiple servlets can also share data, Servlets can also maintain information from request to request, simplifying techniques like session tracking and caching of previous computations.

Portable

Servlets are written in the Java programming language and follow a standard API. Servlets are supported directly or by a plug-in on virtually Web server. Consequently, servlets written for, say, Macromedia Run can run virtually unchanged on Apache Tomcat, Microsoft Internet Information Server IBM Web Sphere, planet Enterprise Server, Oracle9i AS, or Star Nine Webster. They are part of the Java 2 Platform, Enterprise Edition (J2EE; see http://java.sun.com/j2ee/), so industry support for servlets is becoming even more pervasive.

Inexpensive

A number of free or very inexpensive Web servers are good for development use or deployment of low- or medium-volume Web sites. Thus, with servlets and JSP you can start with a free or and migrate to more expensive servers with high-performance capabilities or advanced utilities only after your project meets initial success. For example, India was We surmise that the answer is twofold. First, both countries have large pools of well-educated software developers.

Secure

One of the main sources of vulnerabilities in traditional CGI stems from the fact that the programs are often executed by general-purpose operating system shells. So, the CGI programmer must be careful to filter out characters such as, in widely used CGI libraries. CGI programs are processed by languages that do not automatically check array or string bounds. For example, in C and C++ it is perfectly legal to allocate a 100-element array and then write into the 999th "element," which is really some random part of program memory. So, programmers who forget to perform this check open up their system to deliberate or accidental buffer overflow attacks.

Mainstream

There are a lot of good technologies out there. But if vendors don't support them and developers don't know how to use them, what good are they? Servlet and JSP technology is supported by servers from Apache, Oracle, IBM, Sybase, BEA, Macromedia, Caucho, Sun/planet, New Atlanta, ATG, Fujitsu, Ultras, Silver stream, the World Wide Web Consortium (W3C), and many others. Several low-cost plugins add support to Microsoft IIS and Zeus as well. They run on Windows, Unix/Linux, Maces, VMS, and IBM mainframe operating systems. They are the single most popular application of the Java programming language. They are used by the airline
industry (most United Airlines and Delta Airlines Web sites), e- online banking (First USA Bank, Blanco Popular de Puerto Rico), Web search engines/portals), large financial sites (American Century Investments), and hundreds of other sites that you visit every day when you work with server-side Java.

The Role of JSP

A somewhat oversimplified view of servlets is that they are Java programs with HTML embedded inside of them. A somewhat oversimplified view of JSP documents is that they are HTML pages with Java code embedded inside of them. e, behind the scenes they are the same. In fact, a JSP document is just another way of writing a servlet. JSP pages get translated into servlets, the servlets get compiled, and it is the servlets that run at

8. Coding

Index.jsp

```
<!DOCTYPE HTML>

<html>
	<head>
		<title>Menu</title>
		<meta charset="utf-8" />
		<meta name="viewport" content="width=device-width, initial-scale=1" />
		<!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->
		<link rel="stylesheet" href="assets/css/main.css" />
		<!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->
		<!--[if lte IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->
	</head>
<script type="text/javascript">
function valid()
{
	var name = document.forms["signup"]["name"].value;
		if (name == "") {
	alert("UserName must filled");
	return false;
	}

	if(!/^[a-zA-Z]*$/g.test(uname))
	{
		alert("Username Allows only characters");
		return false;
	}
	var email = document.forms["signup"]["email"].value;
	if(email =="")
	{
		alert("Please Enter Email");
		return false;
	}
	if(!/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/.test(email))
	{
```

```
alert("Please Enter Valid Email Address");
                return false;
        }
        var mobile = document.forms["signup"]["mobile"].value;
        if(mobile == "")
        {
                alert("Please Enter Mobile Number");
                return false;
        }
        if(!/^[0-9]{1,10}$/.test(mobile))
        {
                alert("Mobile Number should be in digits");
                return false;
        }
        if(!/^\d{10}$/.test(mobile))
        {
                alert("Mobile Number should be 10 digits");
                return false;
        }
}
</script>
<body class="landing">
<!-- Header -->
                        <header id="header" class="alt">
                                <h1><a href="#">Secure Data Group Sharing</a></h1>
                                <a href="#nav">Menu</a>
                        </header>
                <!-- Nav -->
                <jsp:include page="HomeMenu.jsp"></jsp:include>
                <!-- Banner -->
                        <section id="banner">
                                <i class="icon fa-diamond"></i>
                                <h2>Secure Data Group Sharing and Conditional Dissemination
with Multi-Owner in Cloud Computing</h2>
                        </section>

                <!-- Four -->
```

```
<section id="four" class="wrapper style2 special">
                                        <div class="inner">
                                                <header class="major narrow">
                                                        <h2>Sign UP</h2>
                                                        <p>All Fields are Mandatory</p>
                                                        <%String s = request.getParameter("status");
                                                        if(s!=null)
                                                        {
                                                                out.print(s);
                                                        }
                                                        %>
                                                </header>
                                                <form action="./RegisterSer" method="post"
name="signup" onsubmit="return valid();">
                                                        <div class="container 75%">
                                                                <div class="row uniform 50%">
                                                                        <div class="6u
12u$(xsmall)">
                                                                                <input
name="name" placeholder="Enter Name" type="text" required/>
                                                                        </div>
                                                                        <div class="6u
12u$(xsmall)">
                                                                                <input
name="password" placeholder="Enter Password" type="password" required/>
                                                                        </div>
                                                                        <div class="6u
12u$(xsmall)">
                                                                                <input
name="mobile" placeholder="Enter Mobile" type="text" pattern="[0-9]{10}" maxlength="10" required/>
                                                                        </div>
                                                                        <div class="6u$
12u$(xsmall)">
                                                                                <input
name="email" placeholder="Enter Email" type="email" required/>
                                                                        </div>
```

```
<div class="6u 12u$(xsmall)">
                                                <input name="date"
placeholder="Enter date" type="date" required/>
                                            </div>
                                            <div class="6u
12u$(xsmall)">
                                                <select name="user"
required>
                                                <option value=""
style="color: black;">--Select User Type--</option>
                                                <option
value="dataowner" style="color: black;">Data Owner</option>
                                                <option
value="datauser" style="color: black;">Data User</option>
                                                <option
value="datacoowner" style="color: black;">Data Co-Owner</option>
                                                </select>
                                            </div>
                                            <div class="12u$">
                                                <textarea
name="address" placeholder="Address" rows="4" required></textarea>
                                            </div>
                                        </div>
                                    </div>
                                    <ul class="actions">
                                        <li><input type="submit"
class="special" value="Submit" /></li>
                                        <li><input type="reset" class="alt"
value="Reset" /></li>
                                    </ul>
                                </form>
                            </div>
                        </section>
                    <!-- Scripts -->
                        <script src="assets/js/jquery.min.js"></script>
                        <script src="assets/js/skel.min.js"></script>
                        <script src="assets/js/util.js"></script>
```

```
<!--[if lte IE 8]><script src="assets/js/ie/respond.min.js"></script><![endif]-->
                                <script src="assets/js/main.js"></script>

        </body>
</html>
```

RegisterSer.java

```
package com.secure.data.controller;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.secure.data.bean.Bean;

import com.secure.data.dao.SecurityDAO;

public class RegisterSer extends HttpServlet {

        public void doPost(HttpServletRequest request, HttpServletResponse response)

                        throws ServletException, IOException {

                response.setContentType("text/html");

                PrintWriter out = response.getWriter();

                Bean b = new Bean();

                b.setUname(request.getParameter("name"));

                b.setPassword(request.getParameter("password"));

                b.setMobile(request.getParameter("mobile"));
```

```
                b.setEmail(request.getParameter("email"));
                b.setDob(request.getParameter("date"));
                b.setUtype(request.getParameter("user"));
                b.setAddress(request.getParameter("address"));
                try{
                int i = new SecurityDAO().reg(b);
                if(i!=0)
                {
                        RequestDispatcher rd =
request.getRequestDispatcher("index.jsp?status=Successfully Signed Up");
                        rd.include(request, response);
                }
                else{
                        RequestDispatcher rd = request.getRequestDispatcher("index.jsp?status=Not
Successfull");
                        rd.include(request, response);
                }
                }catch
                (Exception e) {
                        e.printStackTrace();
                        RequestDispatcher rd = request.getRequestDispatcher("index.jsp?status=Some
Internal Error");
                        rd.include(request, response);
                }
        }
 }
```

SecurityDAO.java

```
package com.secure.data.dao;

import java.sql.Connection;

import java.sql.Date;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.text.SimpleDateFormat;

import java.util.ArrayList;


import com.secure.data.bean.Bean;

import com.secure.data.util.Dbcon;

import com.secure.data.util.MsgEncrypt;


public class SecurityDAO extends Dbcon {

	Connection con;

	public int reg(Bean b)throws Exception

	{

		con=getConnection();

		int i=0;

		PreparedStatement ps = con.prepareStatement("insert into
userdetails(userid,username,password,email,mobile,dod,utype,address)values((select
nvl(max(userid),0)+1 from userdetails),?,?,?,?,?,?,?)");

		ps.setString(1, b.getUname());

		ps.setString(2, b.getPassword());

		ps.setString(3, b.getEmail());

		ps.setString(4, b.getMobile());
```

```
            String d = b.getDob();
            SimpleDateFormat sd = new SimpleDateFormat("yyyy-MM-dd");
            Date dd =new Date(sd.parse(d).getTime());
            ps.setDate(5,dd);
            ps.setString(6, b.getUtype());
            ps.setString(7, b.getAddress());
            i=ps.executeUpdate();
            return i;
        }
        public ArrayList<Bean> login(Bean b)throws Exception
        {
            con=getConnection();
            ArrayList<Bean> al = new ArrayList<Bean>();
            PreparedStatement ps = con.prepareStatement("select userid,email,username,utype from
userdetails where email=? and password=?");
            ps.setString(1, b.getEmail());
            ps.setString(2, b.getPassword());
            ResultSet rs = ps.executeQuery();
            while(rs.next()){
            Bean bb = new Bean();
            bb.setUid(rs.getInt(1));
            bb.setEmail(rs.getString(2));
            bb.setUname(rs.getString(3));
            bb.setUtype(rs.getString(4));
            al.add(bb);
```

```
	}

		return al;

	}

	public int tAcceptFile(int b)throws Exception

	{

		con=getConnection();

		PreparedStatement ps = con.prepareStatement("update ufiles set status='Accepted by Trust' where fid=? and STATUS='sent to trust'");

		ps.setInt(1, b);

		int i =  ps.executeUpdate();

		return i;

	}

	public int dataUserPrivateFileRequest(Bean b)throws Exception

	{

		con=getConnection();

		int uid=0;

		int fid=0;

		PreparedStatement ps1 = con.prepareStatement("select USERID,FID from FILEREQUEST where USERID=?");

		ps1.setInt(1, b.getUid());

		ResultSet rs=ps1.executeQuery();

		while(rs.next())

		{

			uid = rs.getInt(1);

			fid = rs.getInt(2);

		}
```

```
	if(uid!=0 && fid!=0)
		{
			return 0;
		}
		int i=0;
		PreparedStatement ps = con.prepareStatement("insert into
FILEREQUEST(userid,REQUESTEDBY,FID,FNAME,KEY)values(?,?,?,?,'Not Generated')");
		ps.setInt(1, b.getUid());
		ps.setString(2, b.getUname());
		ps.setInt(3, b.getFid());
		ps.setString(4, b.getFname());
		i=ps.executeUpdate();
		return i;
	}
	public int dataCoOwnerRequest(Bean b)throws Exception
	{
		con=getConnection();
		int i=0;
		int ownerid =0;
		int coownerid =0;
		PreparedStatement ps = con.prepareStatement("select OWNERID,COOWNERID from
APPENDEDUSERS where OWNERID=? and COOWNERID=?");
		ps.setInt(1, b.getFid());
		ps.setInt(2, b.getUid());
		ResultSet rs = ps.executeQuery();
		System.out.println("ResultSet===>"+rs);
```

```
while(rs.next())
        {
            ownerid =rs.getInt(1);
            coownerid = rs.getInt(2);
        }
        if(ownerid==0 && coownerid==0)
        {
            PreparedStatement ps1 = con.prepareStatement("insert into
APPENDEDUSERS(OWNERID,COOWNERID,OWNERNAME,COOWNERNAME,STATUS)
values(?,?,?,?,'Requested')");
            ps1.setInt(1, b.getFid());
            ps1.setInt(2, b.getUid());
            ps1.setString(3, b.getFname());
            ps1.setString(4, b.getUname());
            i=ps1.executeUpdate();
            System.out.println("Value of i===>"+i);
        }
        else
        {
            return 0;
        }
        return i;
    }
    public int dataOwnerAcceptDataCoOwner(Bean b)throws Exception
    {
        con=getConnection();
```

```
	int  uid = b.getUid();
		System.out.println("Uid===DAO===>"+uid);
		PreparedStatement ps = con.prepareStatement("update APPENDEDUSERS set STATUS='Accept' where OWNERID=? and COOWNERID=?");
		ps.setInt(1, b.getFid());
		ps.setInt(2, uid);
		int i =  ps.executeUpdate();
		return i;
	}
	public int CSPAcceptDATACOOwnerRequest(Bean b)throws Exception
	{
		con=getConnection();
		PreparedStatement ps = con.prepareStatement("update COOWNERFILES set STATUS='Accepted' where FID=? and REQUESTEDID=?");
		ps.setInt(1, b.getFid());
		ps.setInt(2, b.getUid());
		int i =  ps.executeUpdate();
		return i;
	}
	public int trustUpdateUserKey(Bean b)throws Exception
	{
		con=getConnection();
		PreparedStatement ps = con.prepareStatement("update FILEREQUEST set KEY=? where FID=? and userid=?");
		ps.setString(1, b.getPublickey());
		ps.setInt(2, b.getFid());
```

```
        ps.setInt(3, b.getUid());
                int i =  ps.executeUpdate();
                return i;
        }
        public int datadissminatorUpdateDataOwnerKey(Bean b)throws Exception
        {
                con=getConnection();
                PreparedStatement ps = con.prepareStatement("update UFILES set PUBLICKEY=? where
FID=? and userid=?");
                ps.setString(1, b.getPublickey());
                ps.setInt(2, b.getFid());
                ps.setInt(3, b.getUid());
                int i =  ps.executeUpdate();
                return i;
        }
}
```

9. SYSTEM TESTING

9.1TESTING METHODOLOGIES

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1)Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program

module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a
- the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic other tables Incorrect entry always flashes and message.

Numeric Field

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested..

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing., assuming that the live data in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and it can be put to efficient use Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system

development. Depending on the requirements, , it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution,.A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future The future holds a lot to offer to the development and refinement of this project.

9.2TEST CASES

Test cases can be divided in to two types. First one is Positive test cases and second one is negative test cases. In positive test cases are conducted by the developer intention is to get the output. In negative test cases are conducted by the developer intention is to don't get the output.

+VE TEST CASES

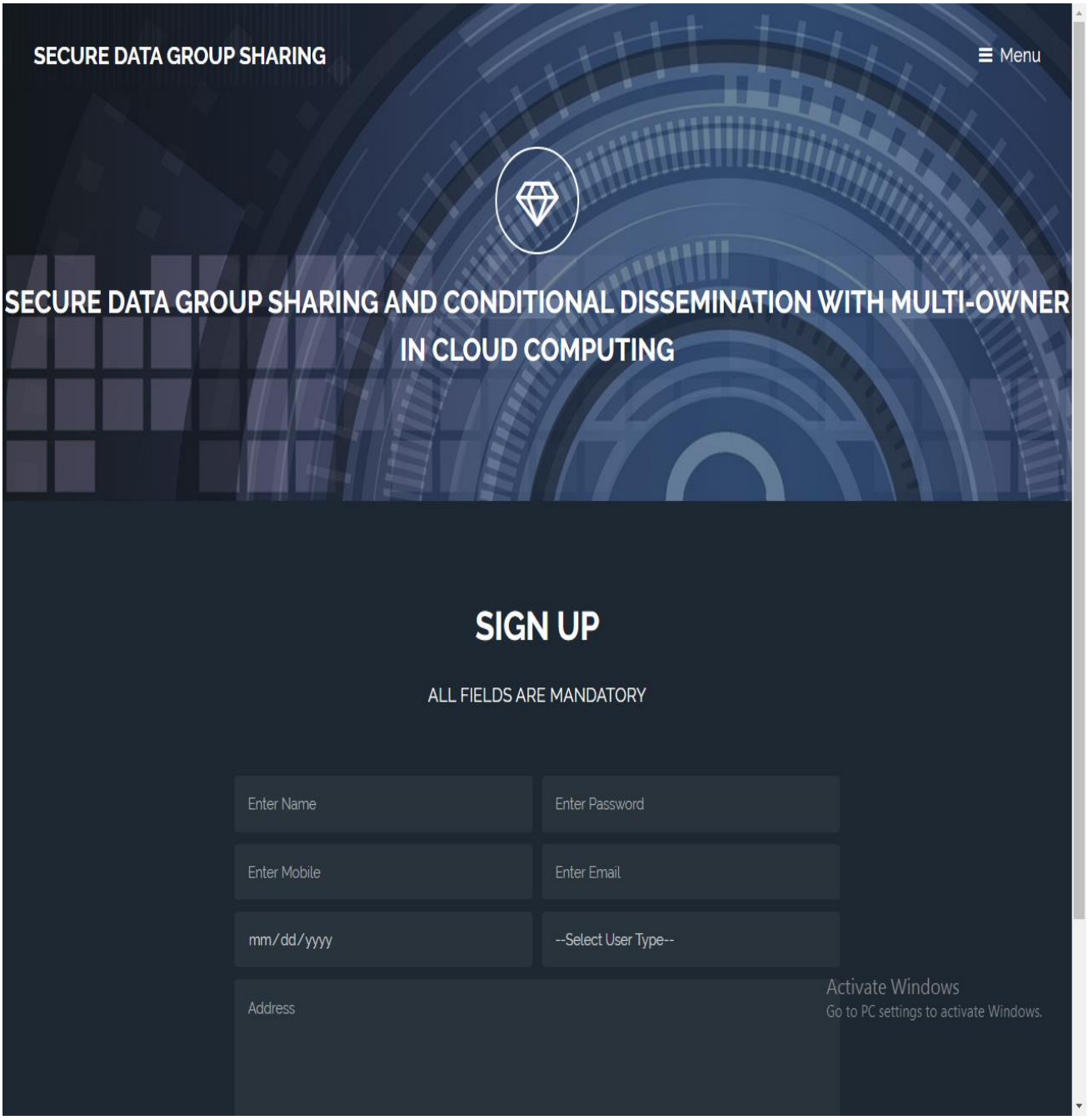
S .No	Test case Description	Actual value	Expected value	Result
1	Create new user registration process	Enter the personal info and address info.	Update personal info and address info in to oracle database successfully	True
2	Enter the username and password	Verification of login details.	Login Successfully	True
3	Upload New File into cloud information	Enter all fields	Web data uploaded successfully	True
4	Data User Download File using key	Enter key	Show data in database	True

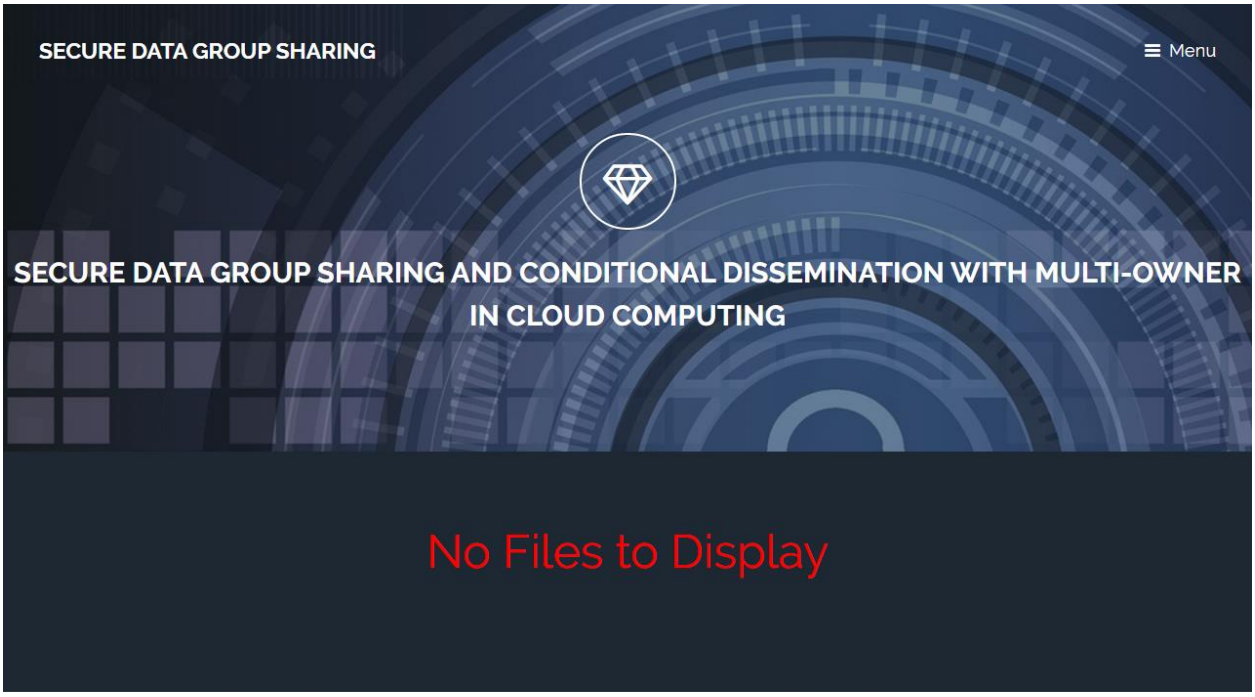
-VE TEST CASES

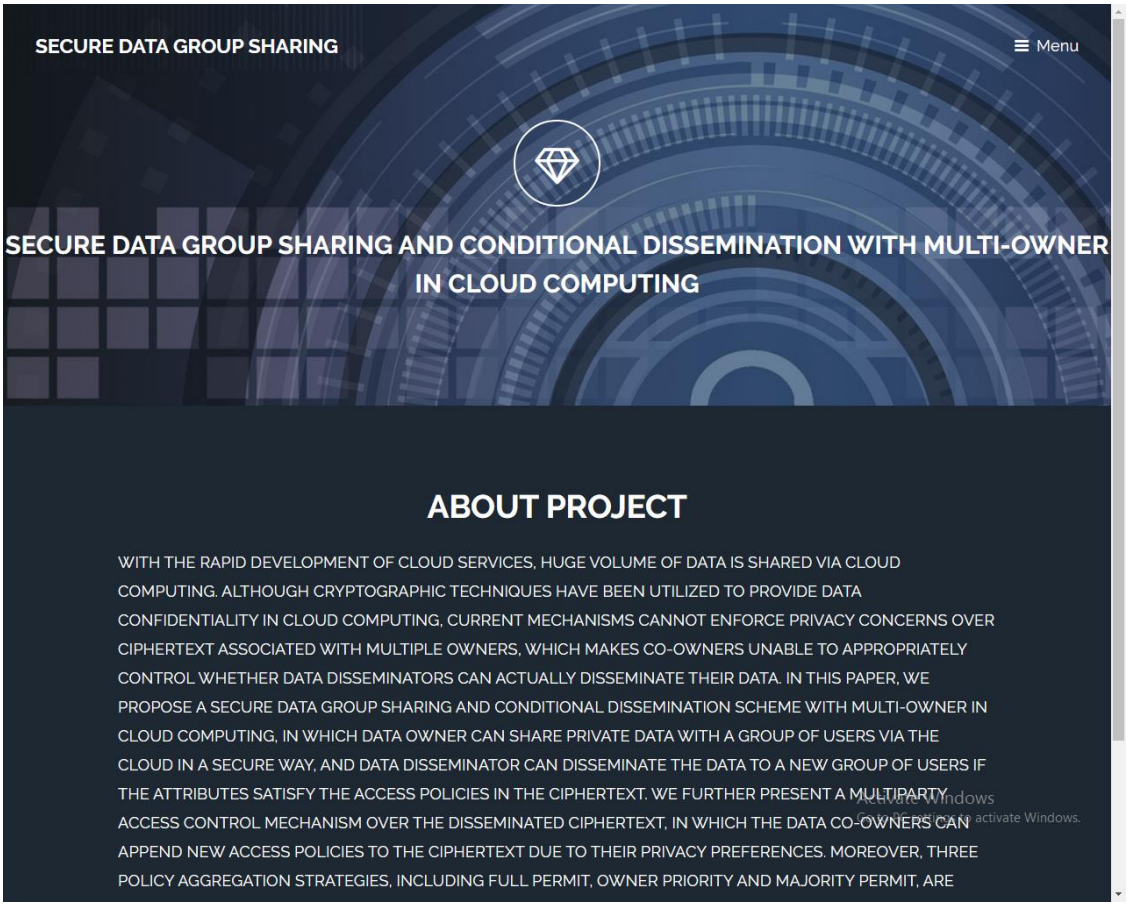
S .No	Test case Description	Actual value	Expected value	Result
1	Create the new user registration process	Enter the personal info and address info.	Personal info and address info its not update into database successfully.	False
2	Enter the username and password	Verification of login details.	Login failed	False
3	Upload data Owner File	Enter all fields	Web data is not create successfully.	False
4	Data User Download File using key	Enter Key	Web data score store in database	False

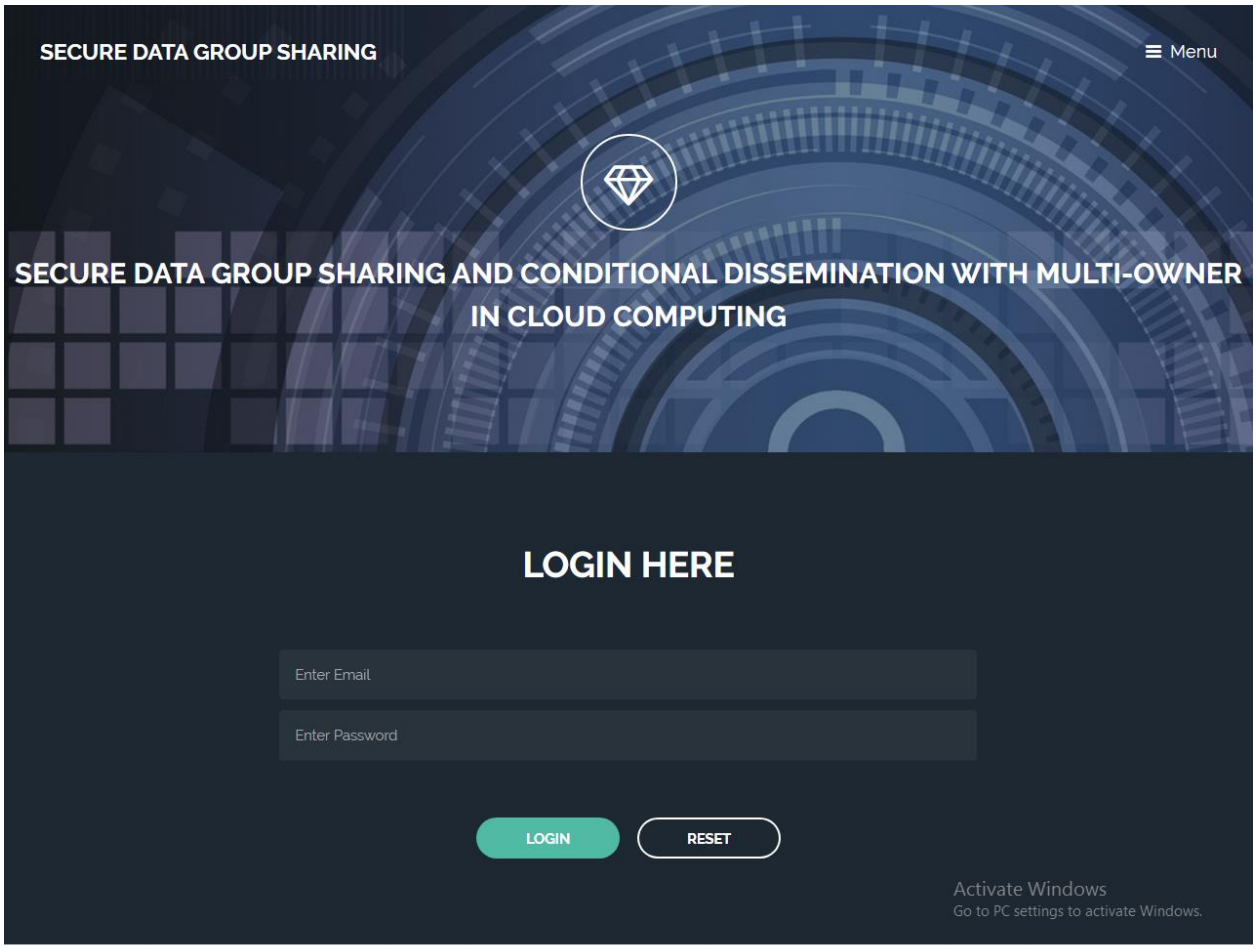
10. Output Screens

home

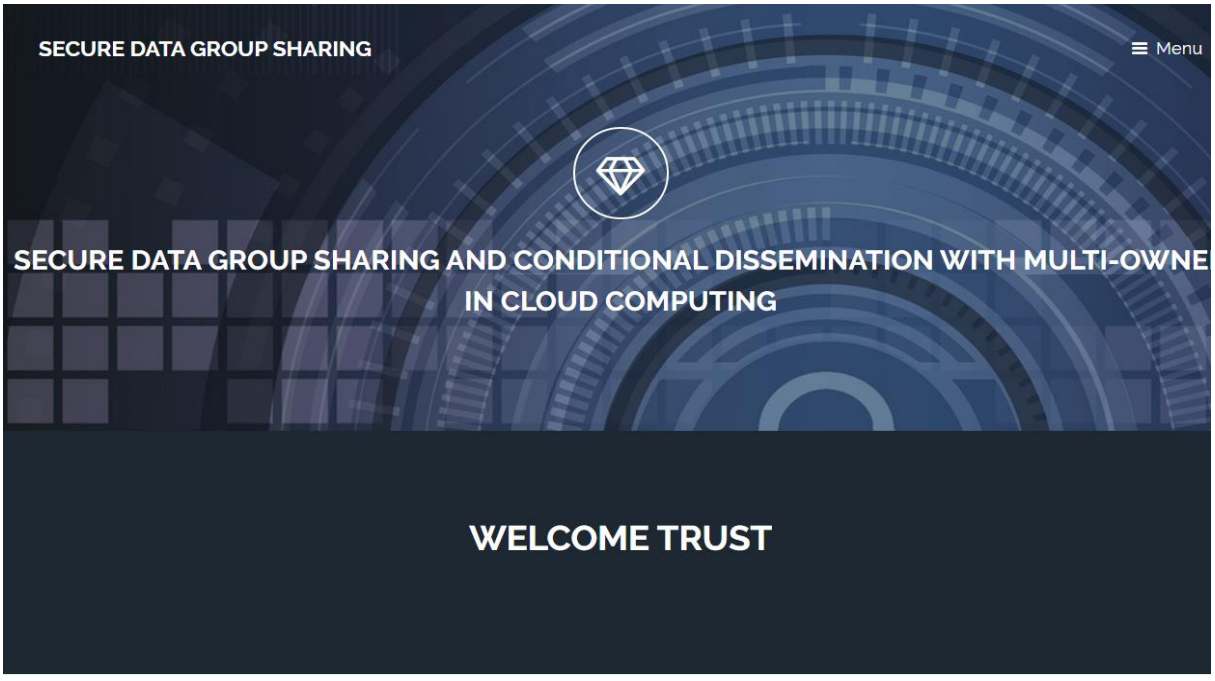


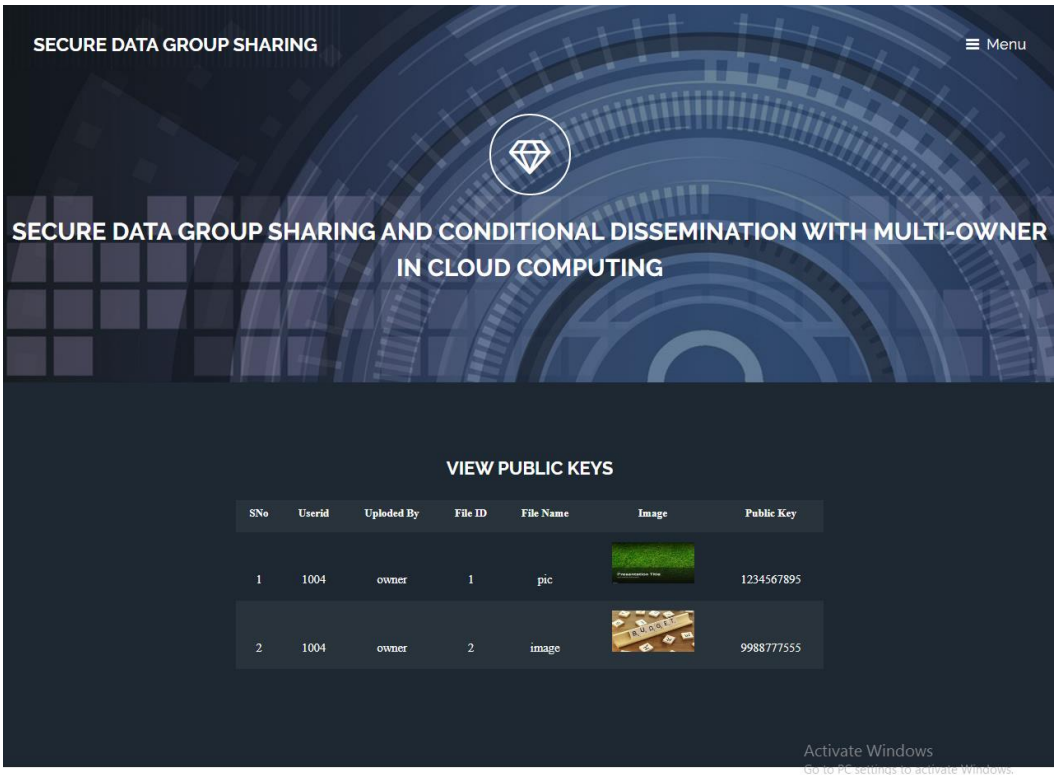

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
No Files to Display

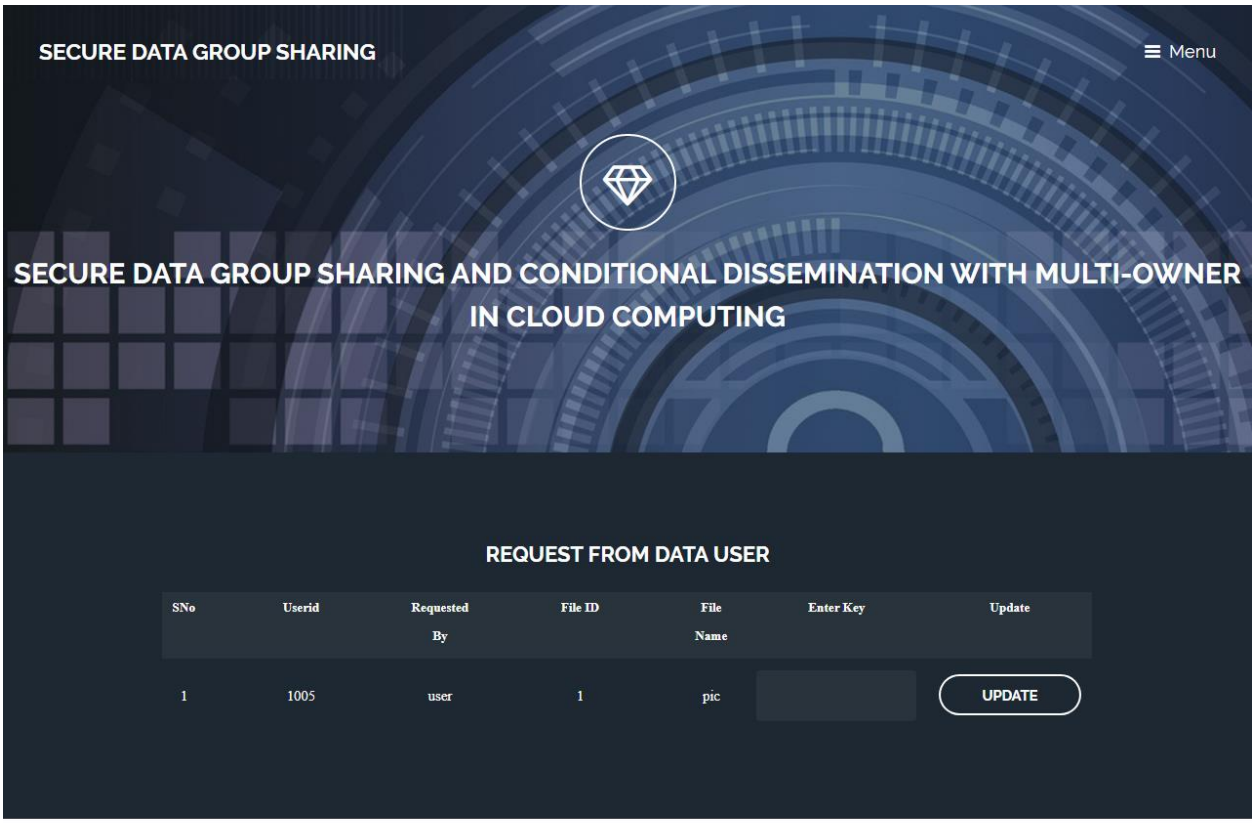

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
ABOUT PROJECT
WITH THE RAPID DEVELOPMENT OF CLOUD SERVICES, HUGE VOLUME OF DATA IS SHARED VIA CLOUD COMPUTING. ALTHOUGH CRYPTOGRAPHIC TECHNIQUES HAVE BEEN UTILIZED TO PROVIDE DATA CONFIDENTIALITY IN CLOUD COMPUTING, CURRENT MECHANISMS CANNOT ENFORCE PRIVACY CONCERNS OVER CIPHERTEXT ASSOCIATED WITH MULTIPLE OWNERS, WHICH MAKES CO-OWNERS UNABLE TO APPROPRIATELY CONTROL WHETHER DATA DISSEMINATORS CAN ACTUALLY DISSEMINATE THEIR DATA. IN THIS PAPER, WE PROPOSE A SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION SCHEME WITH MULTI-OWNER IN CLOUD COMPUTING, IN WHICH DATA OWNER CAN SHARE PRIVATE DATA WITH A GROUP OF USERS VIA THE CLOUD IN A SECURE WAY, AND DATA DISSEMINATOR CAN DISSEMINATE THE DATA TO A NEW GROUP OF USERS IF THE ATTRIBUTES SATISFY THE ACCESS POLICIES IN THE CIPHERTEXT. WE FURTHER PRESENT A MULTIPARTY ACCESS CONTROL MECHANISM OVER THE DISSEMINATED CIPHERTEXT, IN WHICH THE DATA CO-OWNERS CAN APPEND NEW ACCESS POLICIES TO THE CIPHERTEXT DUE TO THEIR PRIVACY PREFERENCES. MOREOVER, THREE POLICY AGGREGATION STRATEGIES, INCLUDING FULL PERMIT, OWNER PRIORITY AND MAJORITY PERMIT, ARE
Activate Windows
Go to PC settings to activate Windows.

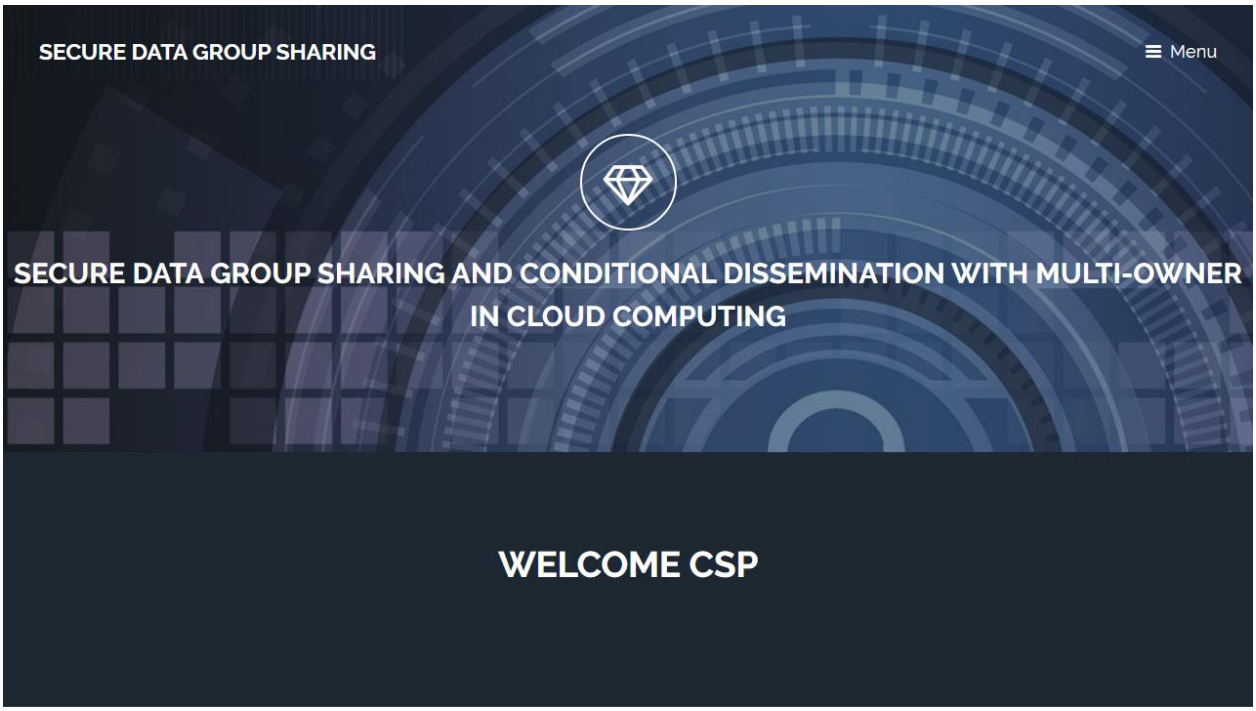

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
LOGIN HERE
Enter Email
Enter Password
LOGIN
RESET
Activate Windows
Go to PC settings to activate Windows.

Trust

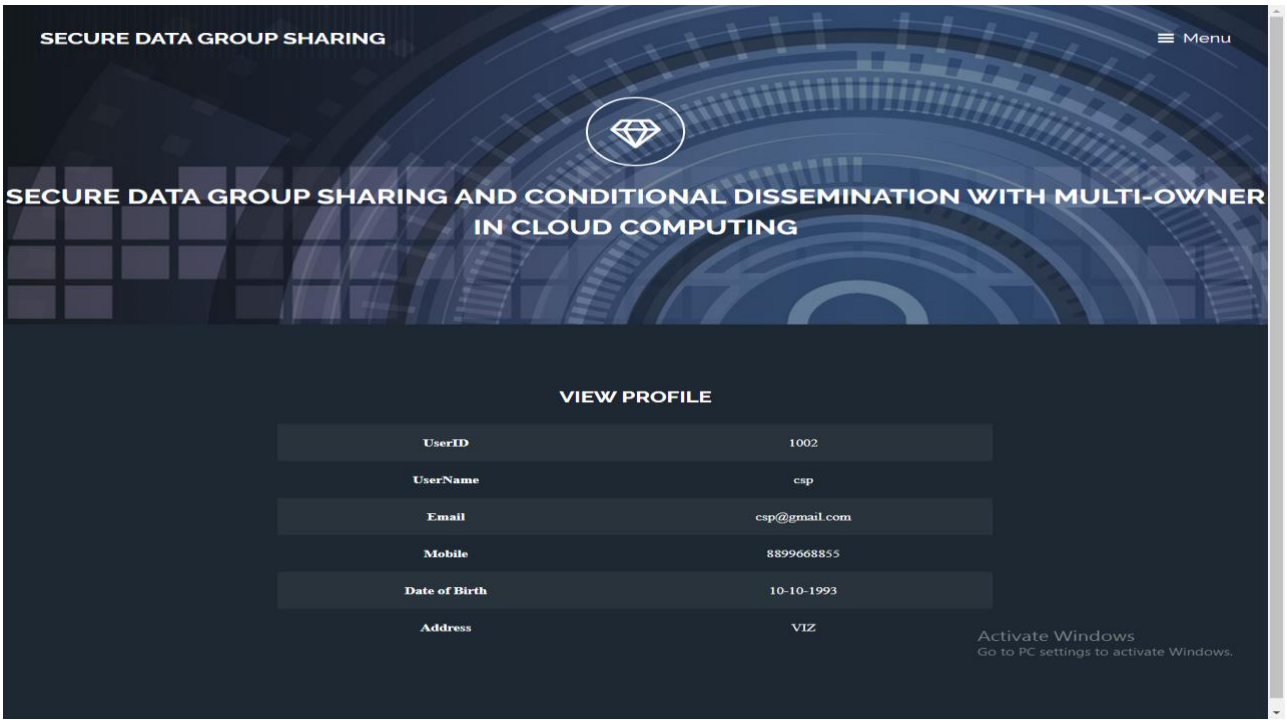

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNE IN CLOUD COMPUTING
WELCOME TRUST

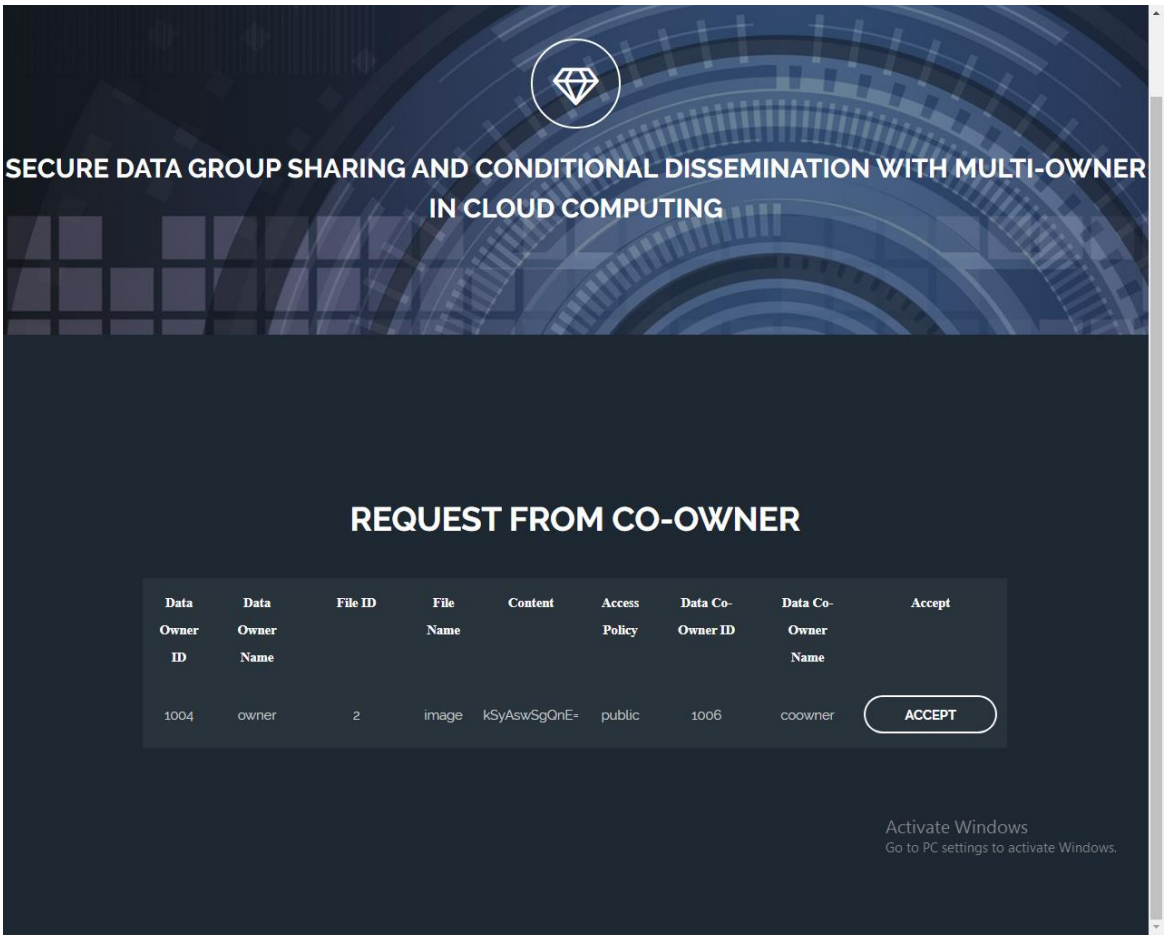

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
VIEW PUBLIC KEYS
SNo Userid Uploded By File ID File Name Image Public Key
1 1004 owner 1 pic 1234567895
2 1004 owner 2 image 9988777555
Activate Windows

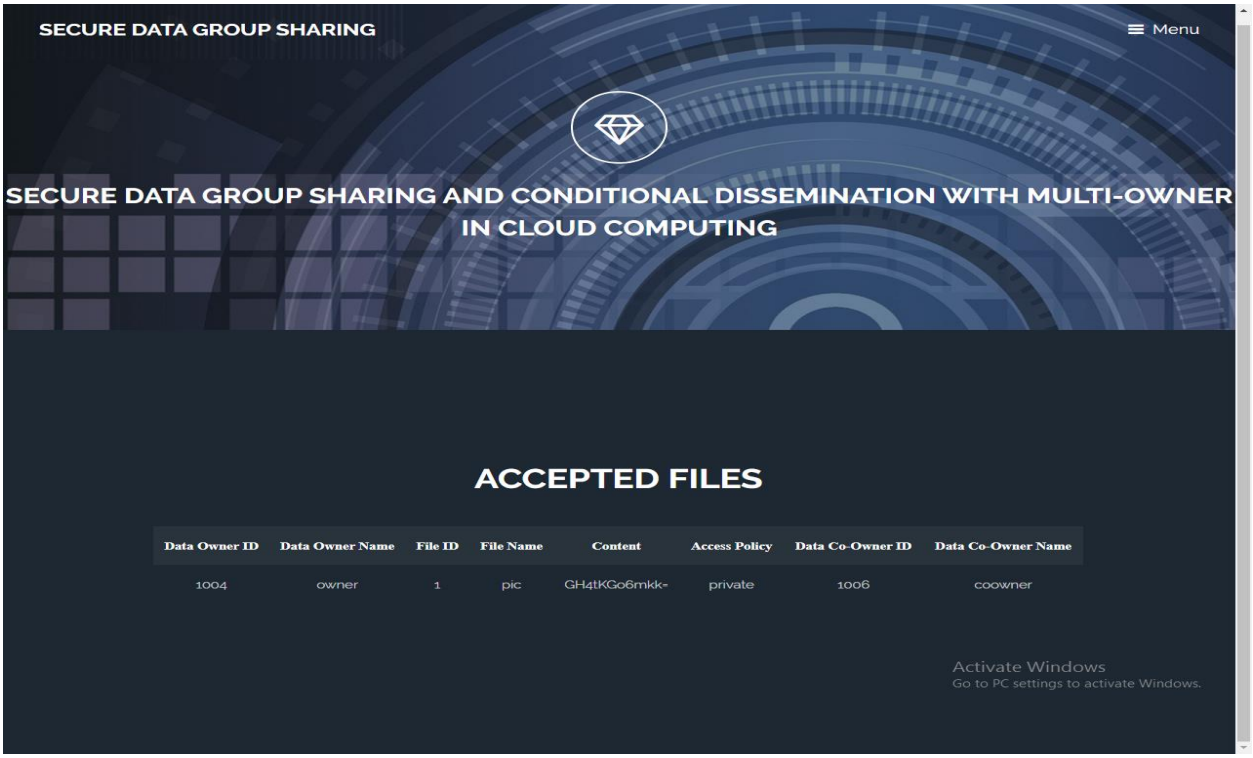

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
REQUEST FROM DATA USER
SNo Userid Requested By File ID File Name Enter Key Update
1 1005 user 1 pic
UPDATE


SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
WELCOME CSP

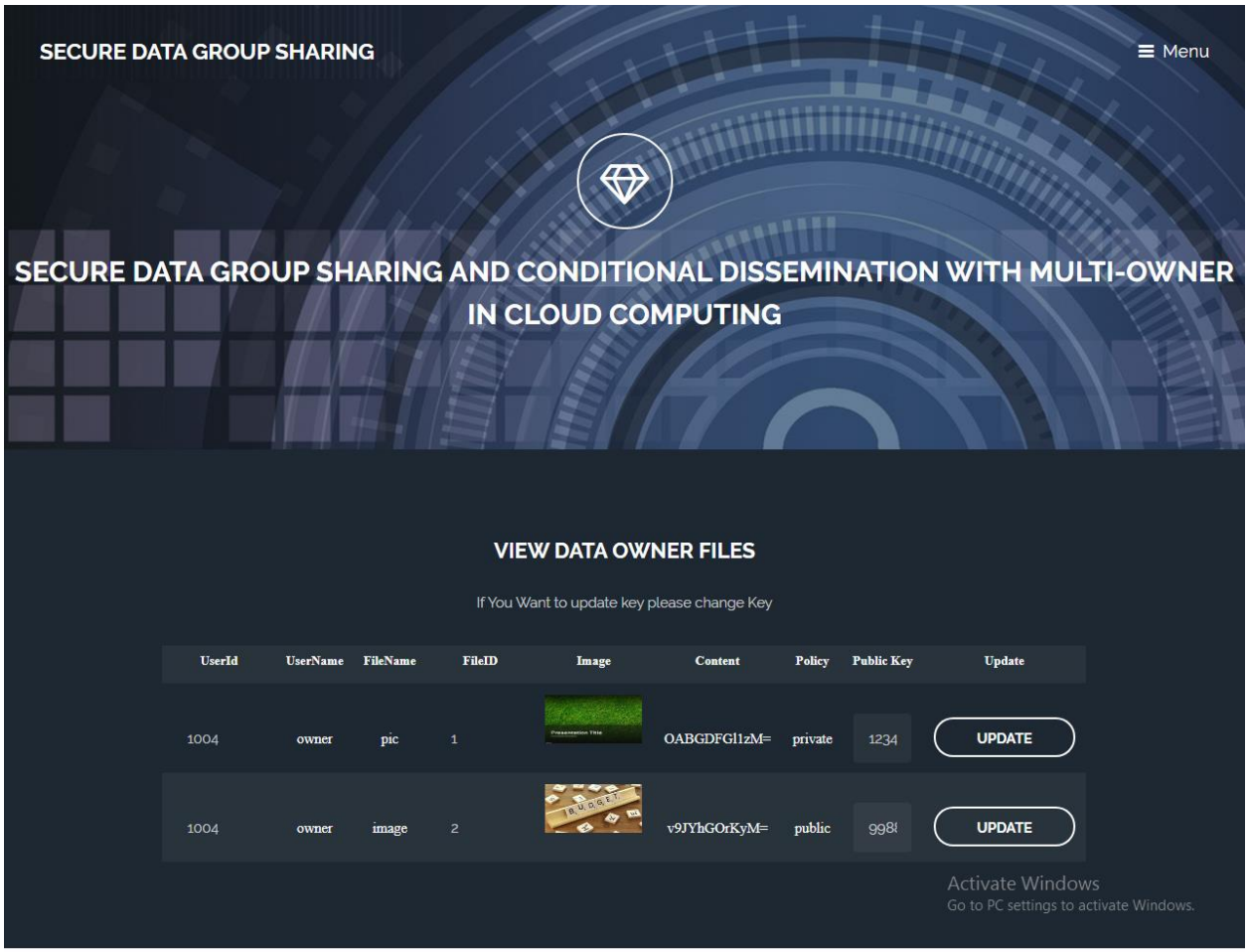
CSP

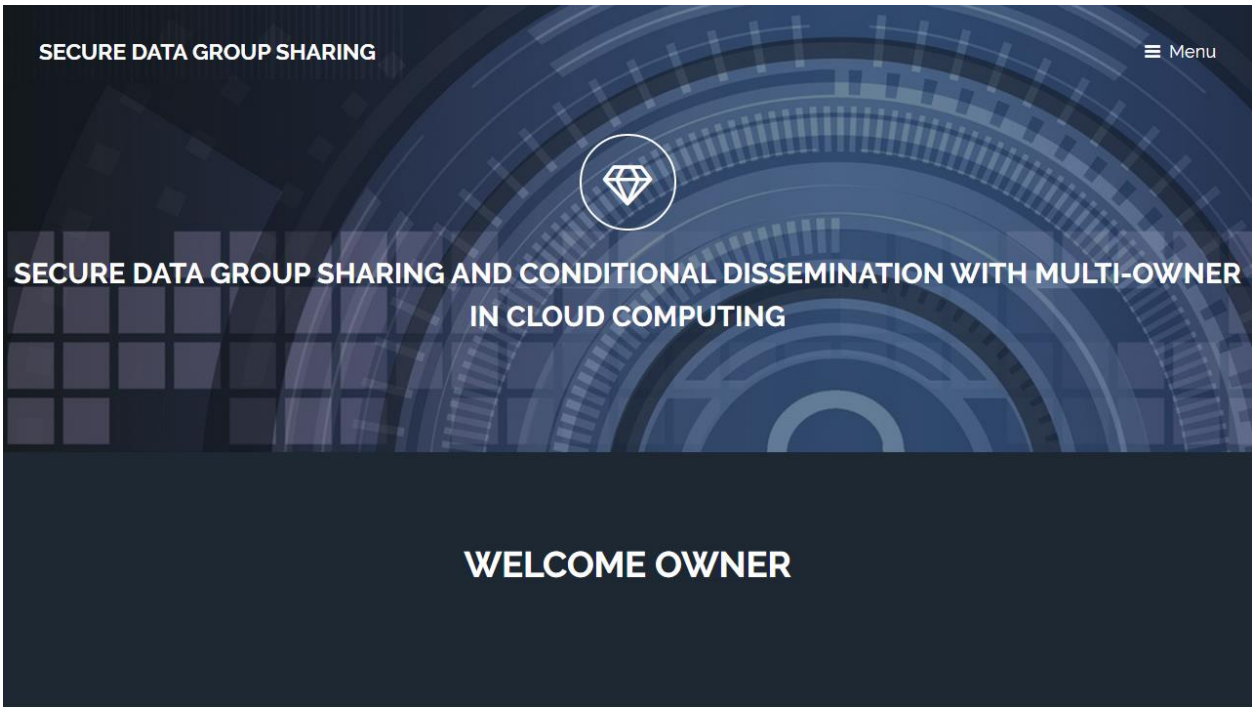

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
VIEW PROFILE
UserID 1002
UserName csp
Email csp@gmail.com
Mobile 8899668855
Date of Birth 10-10-1993
Address VIZ
Activate Windows
Go to PC settings to activate Windows.


SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
REQUEST FROM CO-OWNER
Data Owner ID
Data Owner Name
File ID
File Name
Content
Access Policy
Data Co-Owner ID
Data Co-Owner Name
Accept
1004
owner
2
image
kSyAswSgQnE=
public
1006
coowner
ACCEPT
Activate Windows
Go to PC settings to activate Windows.

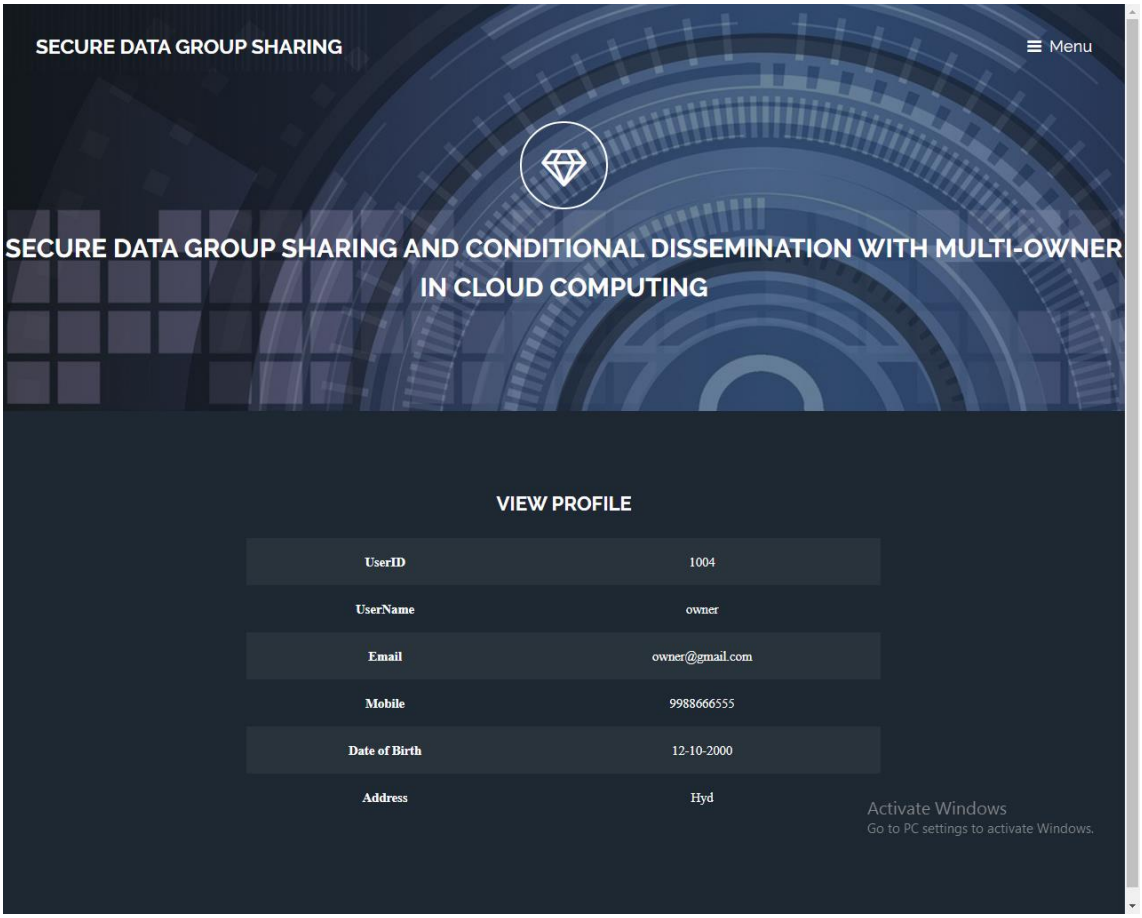

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
ACCEPTED FILES
Data Owner ID
Data Owner Name
File ID
File Name
Content
Access Policy
Data Co-Owner ID
Data Co-Owner Name
1004
owner
1
pic
GH4tKGo6mkk=
private
1006
coowner
Activate Windows
Go to PC settings to activate Windows.

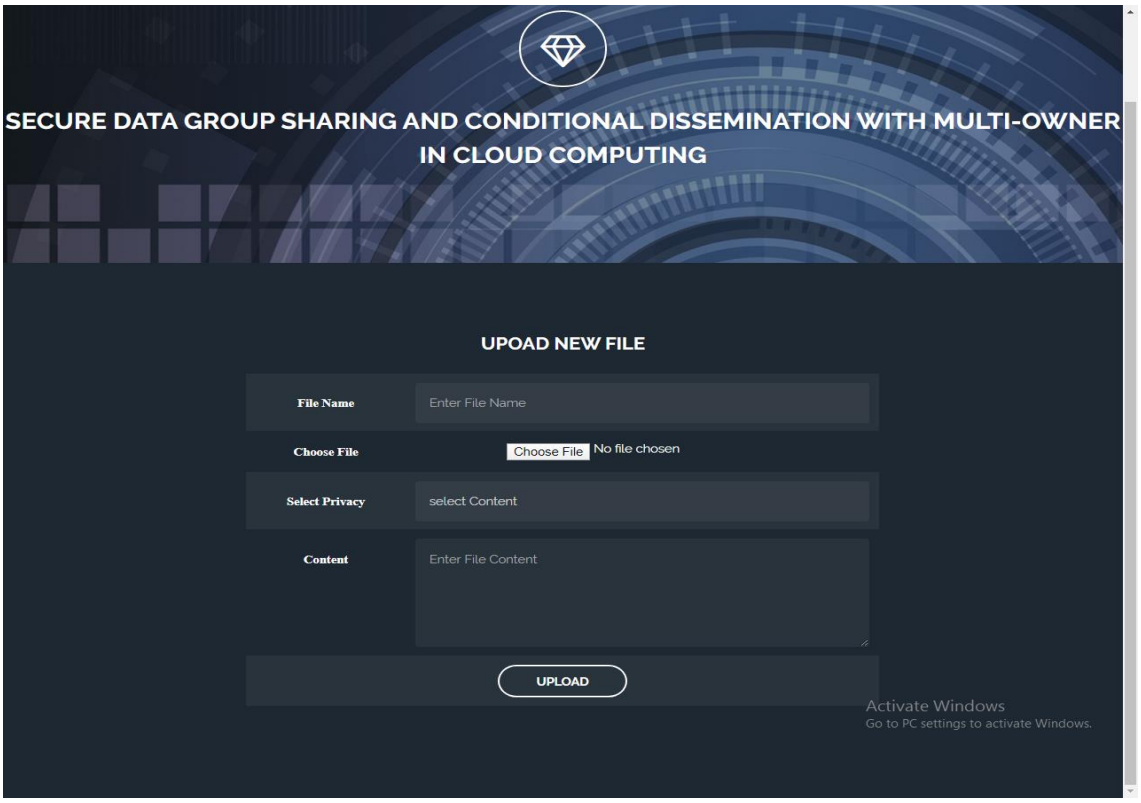
Disseminater

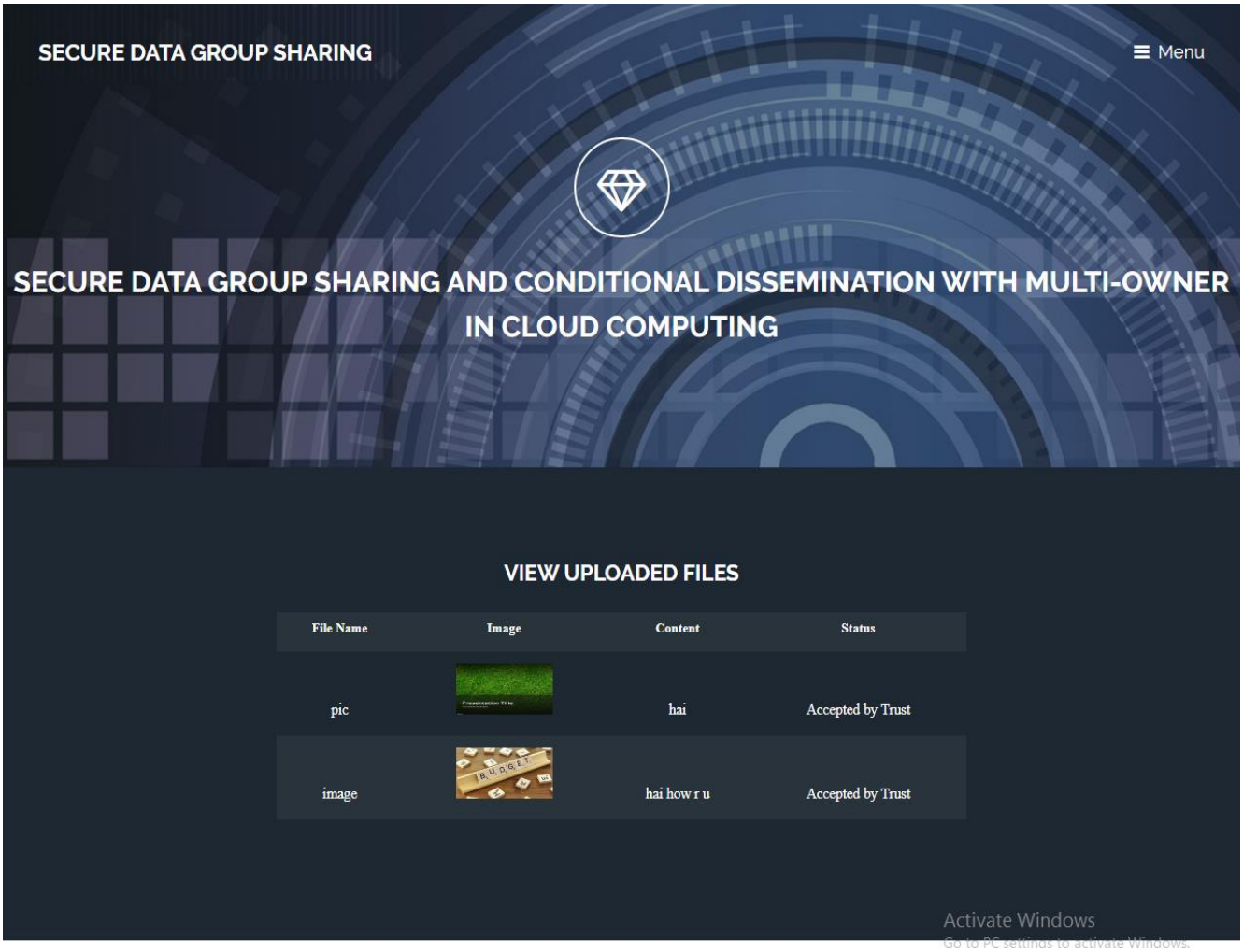


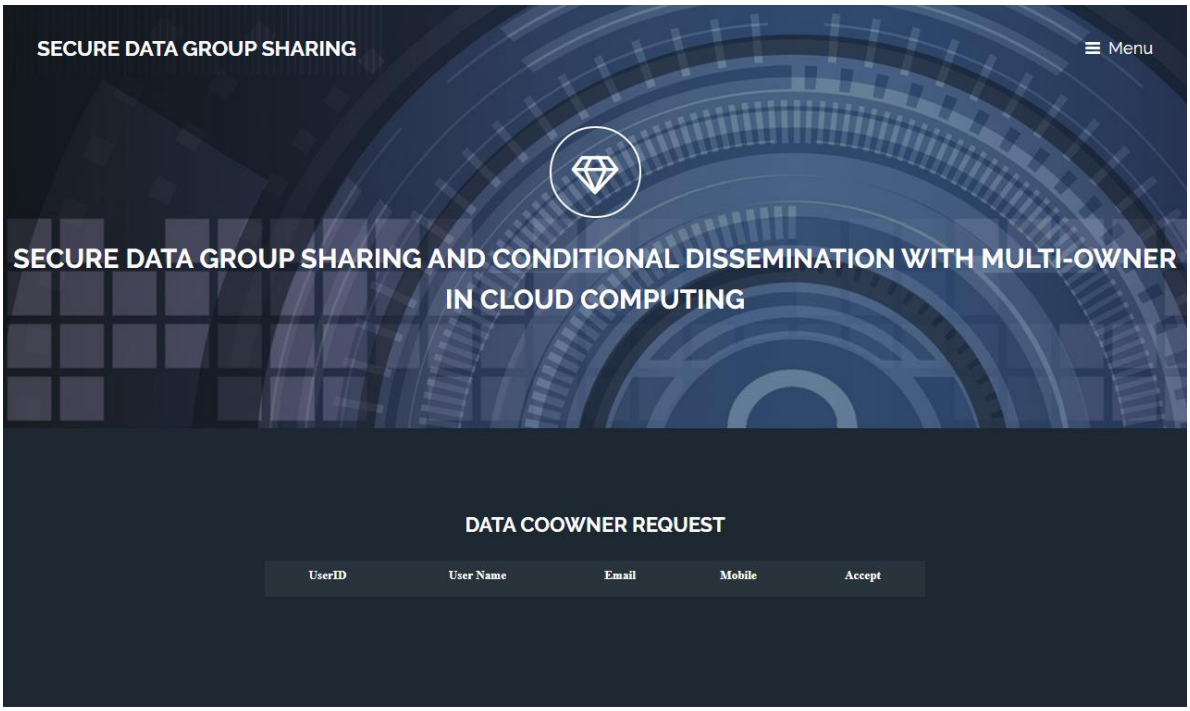

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
WELCOME OWNER

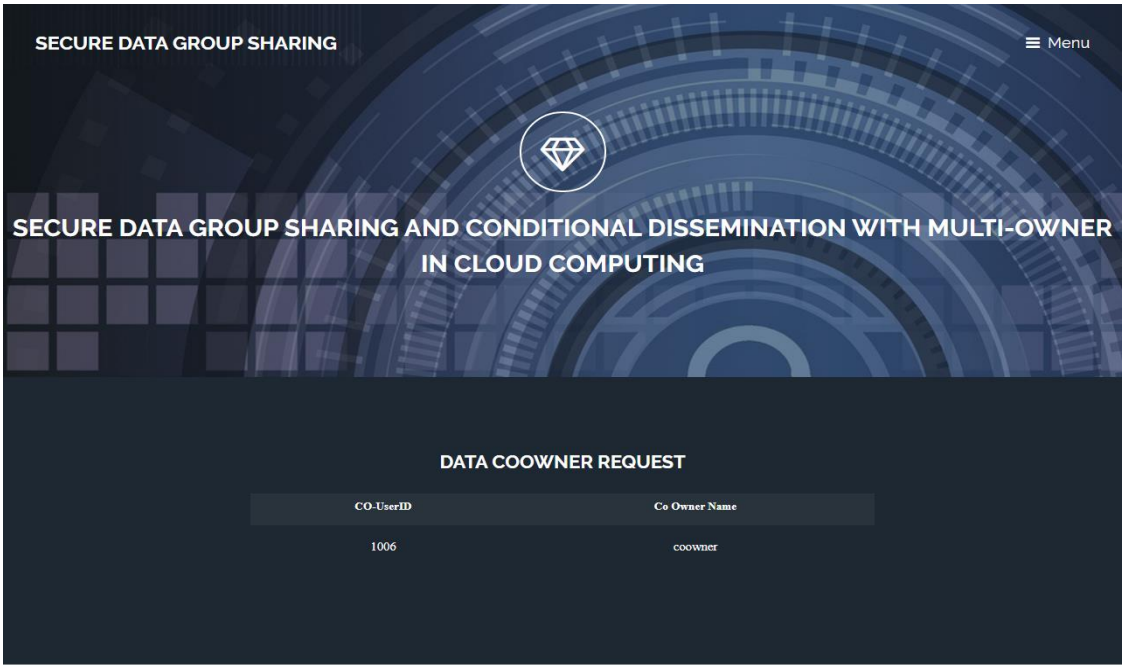
owner


SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
VIEW PROFILE
UserID 1004
UserName owner
Email owner@gmail.com
Mobile 9988666555
Date of Birth 12-10-2000
Address Hyd
Activate Windows
Go to PC settings to activate Windows.

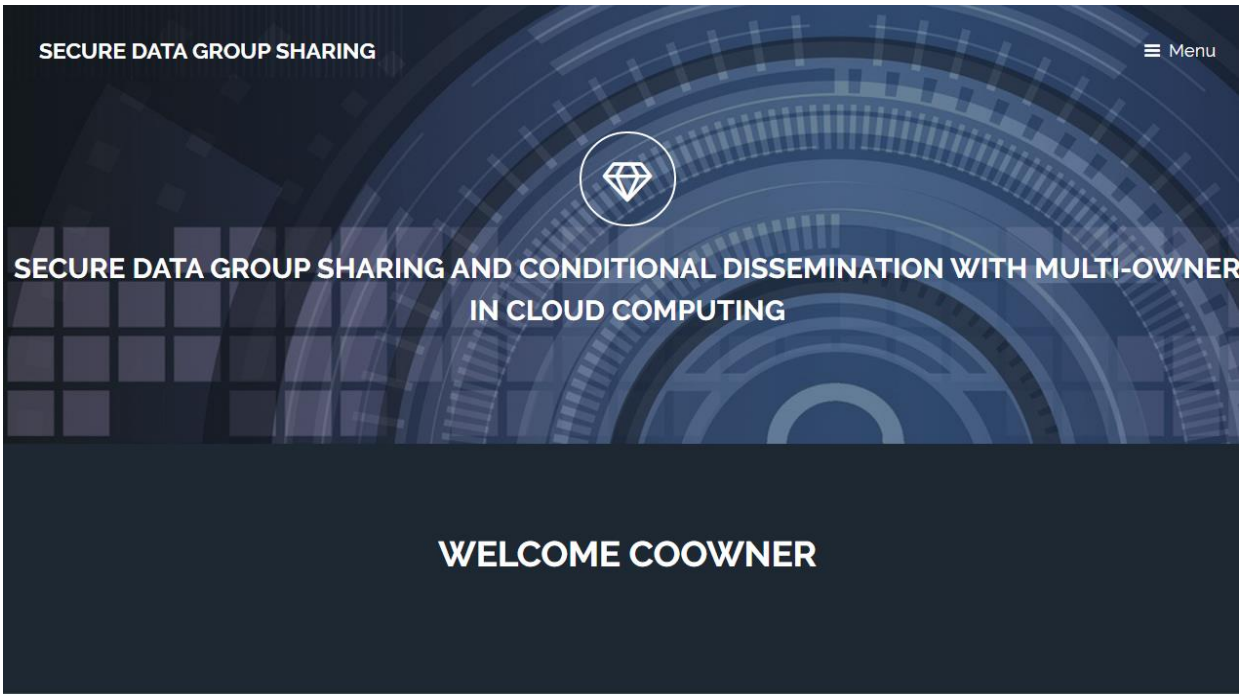

SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
UPOAD NEW FILE
File Name
Enter File Name
Choose File
Choose File
No file chosen
Select Privacy
select Content
Content
Enter File Content
UPLOAD
Activate Windows
Go to PC settings to activate Windows.

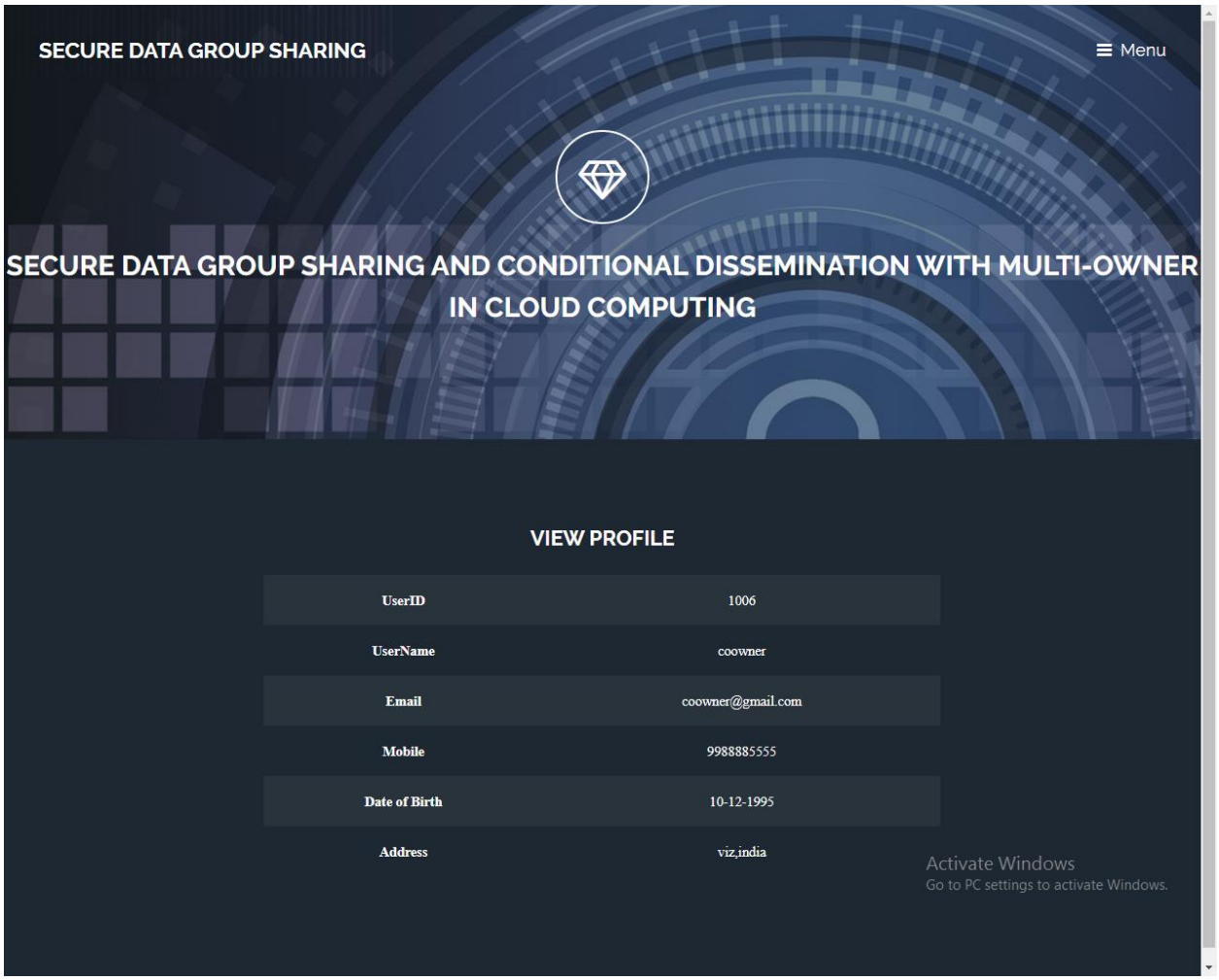

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
VIEW UPLOADED FILES
File Name
Image
Content
Status
pic
hai
Accepted by Trust
image
hai how r u
Accepted by Trust
Activate Windows

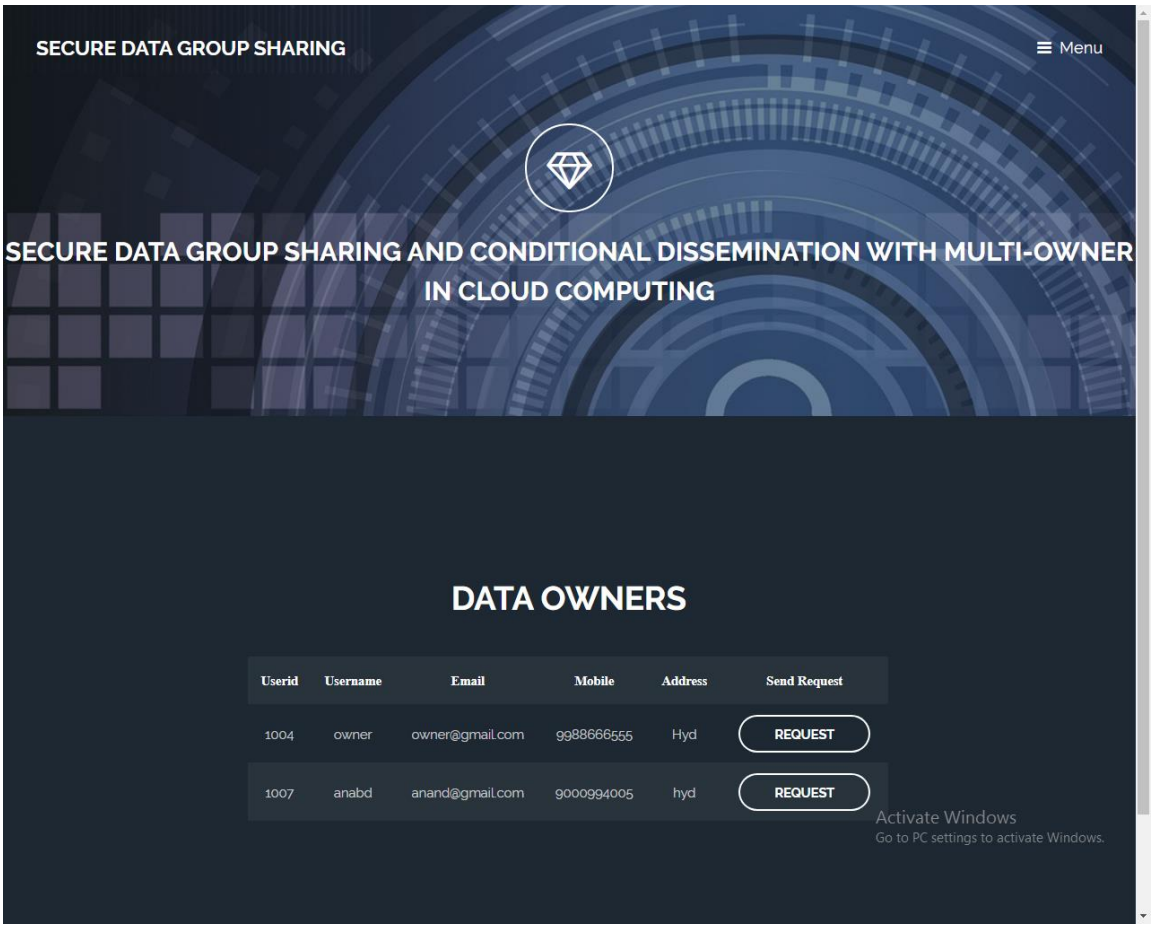

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
DATA COOWNER REQUEST
UserID
User Name
Email
Mobile
Accept

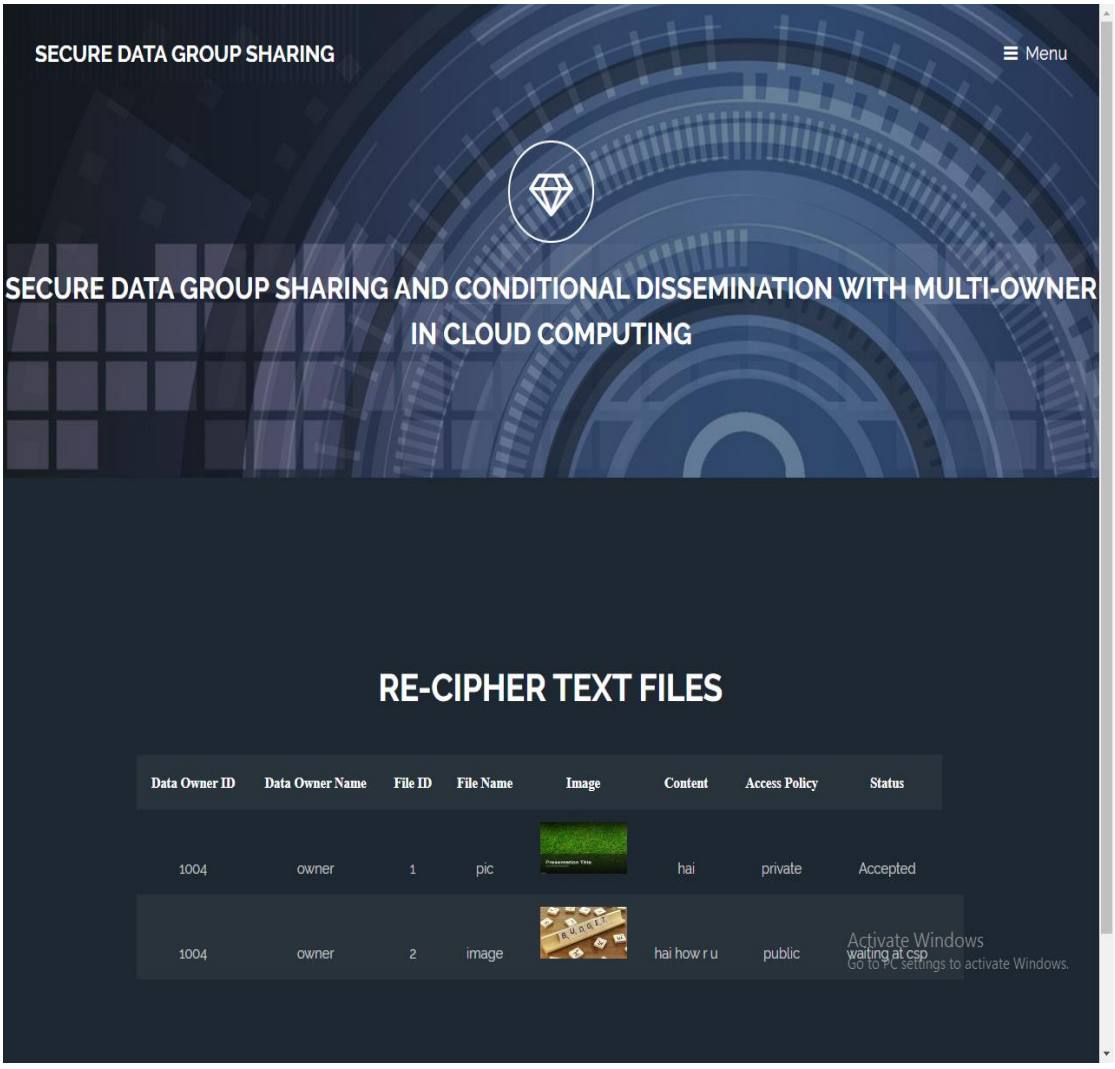

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
DATA COOWNER REQUEST
CO-UserID
Co Owner Name
1006
coowner

Co owner

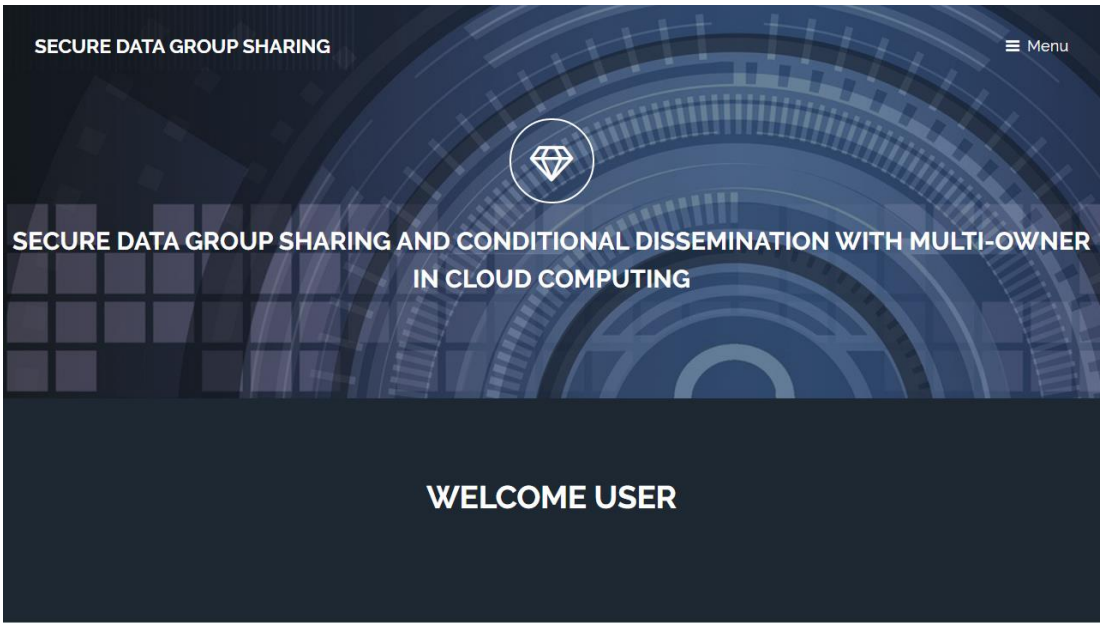

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
WELCOME COOWNER

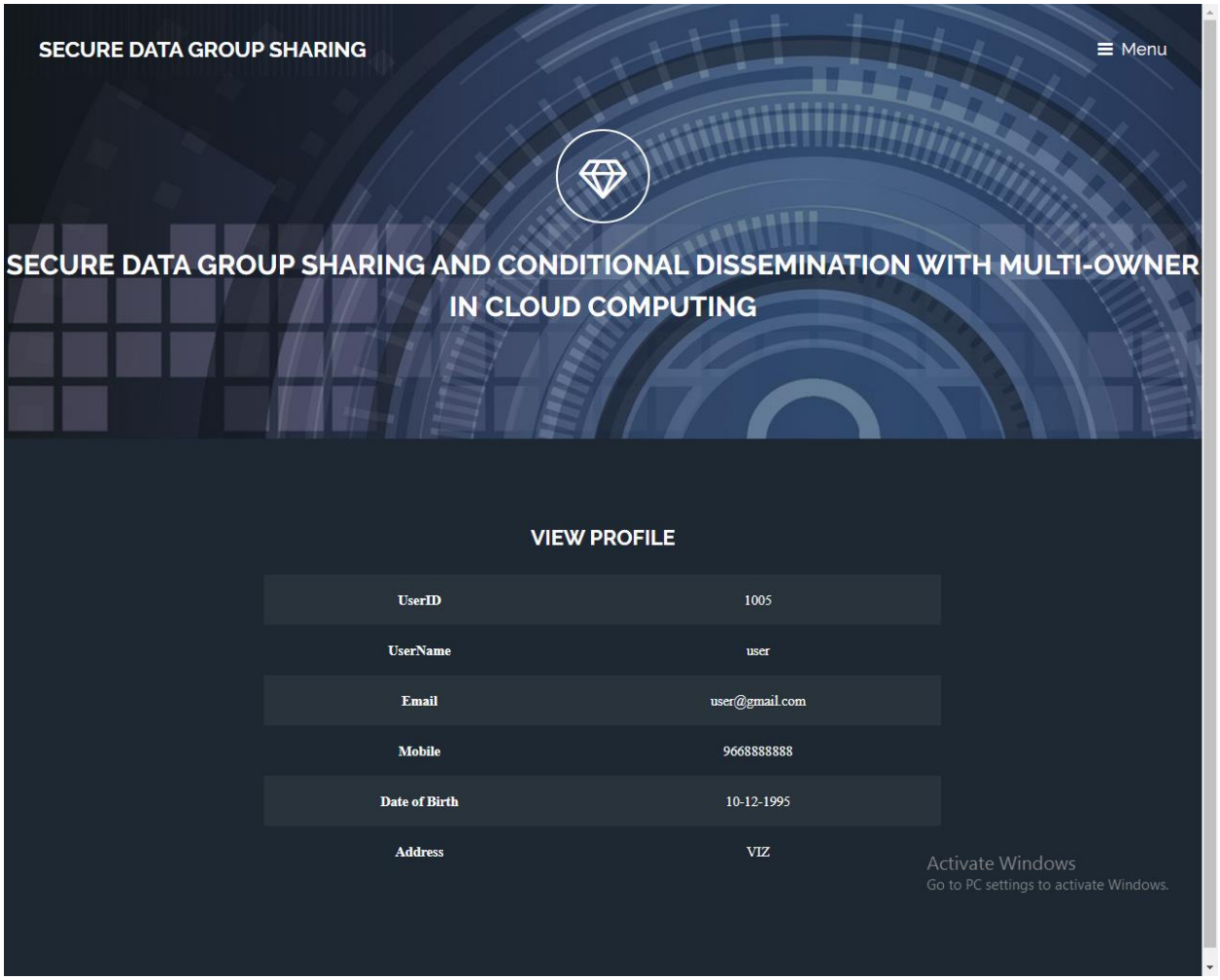

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
VIEW PROFILE
UserID 1006
UserName coowner
Email coowner@gmail.com
Mobile 9988885555
Date of Birth 10-12-1995
Address viz,india
Activate Windows
Go to PC settings to activate Windows.

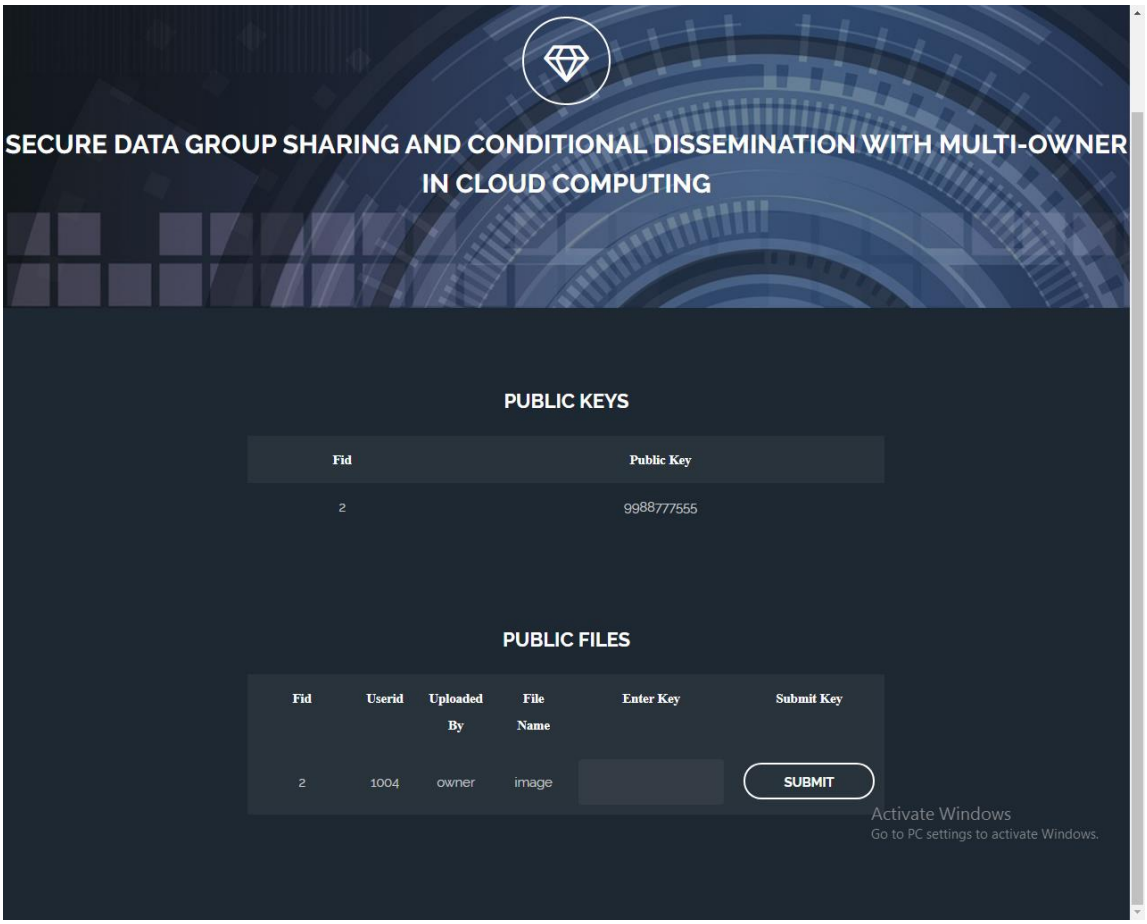

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
DATA OWNERS
Userid Username Email Mobile Address Send Request
1004 owner owner@gmail.com 9988666555 Hyd REQUEST
1007 anabd anand@gmail.com 9000994005 hyd REQUEST
Activate Windows
Go to PC settings to activate Windows.

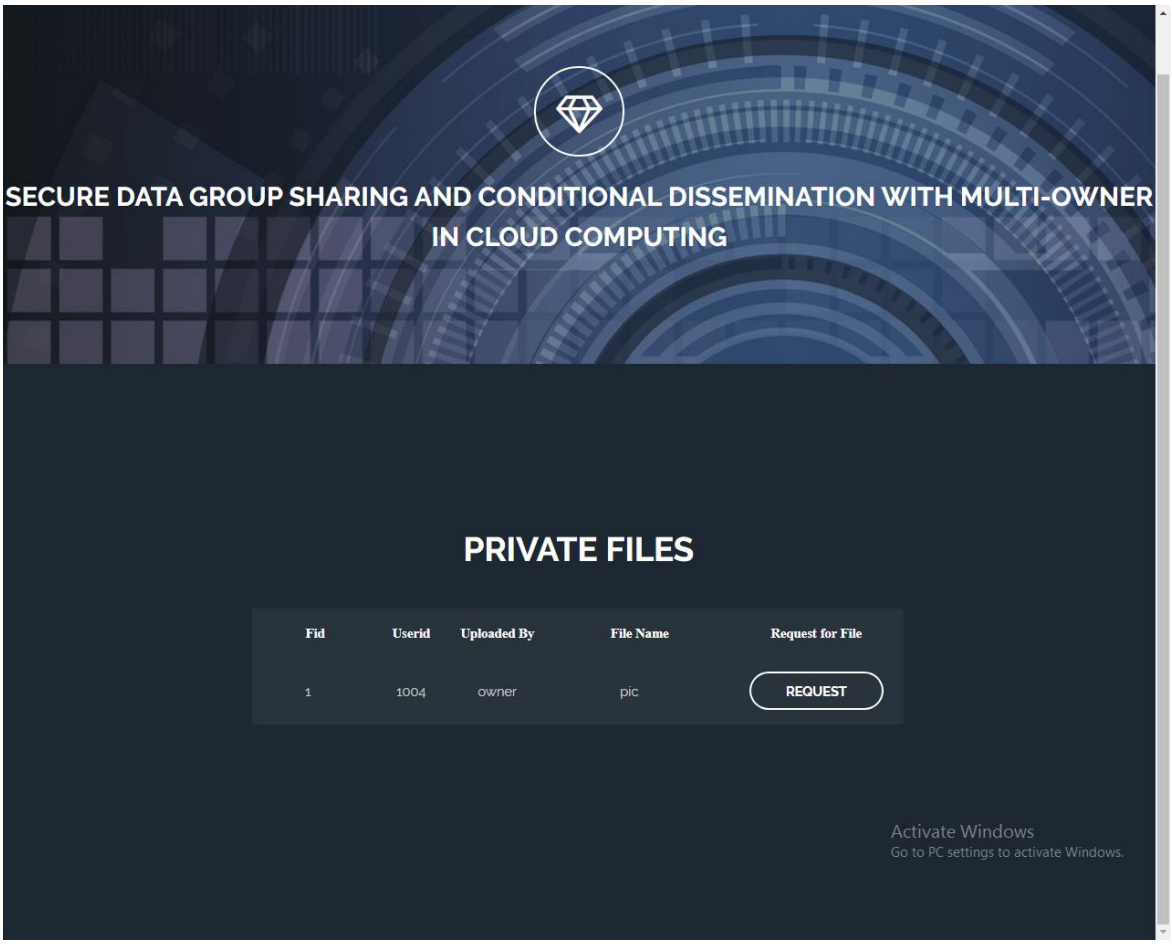

SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
RE-CIPHER TEXT FILES
Data Owner ID Data Owner Name File ID File Name Image Content Access Policy Status
1004 owner 1 pic hai private Accepted
1004 owner 2 image hai how r u public waiting at csp
Activate Windows
Go to PC settings to activate Windows.

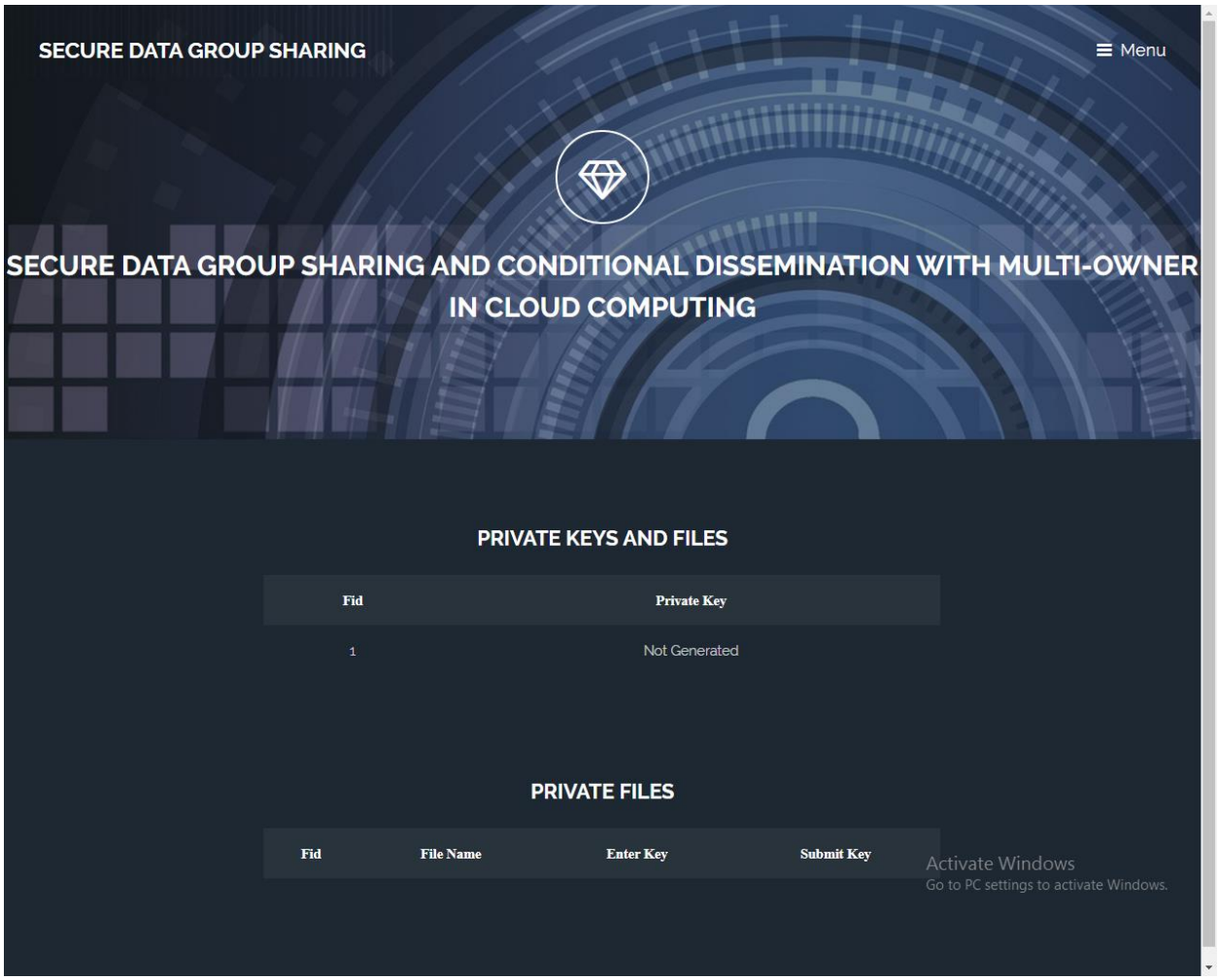
User


SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
WELCOME USER


SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
VIEW PROFILE
UserID 1005
UserName user
Email user@gmail.com
Mobile 9668888888
Date of Birth 10-12-1995
Address VIZ
Activate Windows
Go to PC settings to activate Windows.


SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
PUBLIC KEYS
Fid
Public Key
2
9988777555
PUBLIC FILES
Fid
Userid
Uploaded By
File Name
Enter Key
Submit Key
2
1004
owner
image
SUBMIT
Activate Windows
Go to PC settings to activate Windows.


SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
PRIVATE FILES
Fid
Userid
Uploaded By
File Name
Request for File
1
1004
owner
pic
REQUEST
Activate Windows
Go to PC settings to activate Windows.


SECURE DATA GROUP SHARING
Menu
SECURE DATA GROUP SHARING AND CONDITIONAL DISSEMINATION WITH MULTI-OWNER IN CLOUD COMPUTING
PRIVATE KEYS AND FILES
Fid
Private Key
1
Not Generated
PRIVATE FILES
Fid
File Name
Enter Key
Submit Key
Activate Windows
Go to PC settings to activate Windows.

11. CONCLUSIONS

The data security and privacy is a concern for users in cloud computing. In particular, how to enforce privacy concerns of multiple owners and protect the data confidentiality becomes a challenge. In this, we present a secure data group sharing and conditional dissemination scheme with multi-owner in cloud computing. In our scheme, the data owner could encrypt her or his private data and share it with a group of data accessors at one time in a convenient way based on IBBE technique. Meanwhile, the data owner can specify fine-grained access policy to the cipher text based on attribute-based CPRE, thus the cipher text can only be re-encrypted by Data disseminator whose attributes satisfy the access policy in the cipher text. We further present a multiparty access control mechanism over the cipher text, which allows the data co-owners to append their access policies to the cipher text. Besides, we provide three policy aggregation strategies including full permit, owner priority and majority permit to solve the problem of privacy conflicts.

12. Future Enhancements

In the future, we will enhance our scheme by supporting keyword search over the ciphertext.

13. References

Core Java™ 2 Volume I – Fundamentals 7th Edition - Cay S. Hortsman
Pearson Education – Sun Microsystems Gary Cornell

Core Java™ 2 Volume II – Advanced - Cay S. Hortsman
Pearson Education – Sun Microsystems Gary Cornell

Head First Servlets & JSP - Eric Freeman
O'Reilly – SPD Elisabeth Freeman

The Book of JavaScript 2nd Edition - Thau
SPD

Effective Java – Programming Language Guide - Joshua Bloch
Pearson Education – Sun Microsystems

Java Database Best Practices - George Reese
O'Reilly – SPD

JBoss – A Developers Notebook - Norman Richards
O'Reilly – SPD Sam Griffith

14. Bibliography

(1) Java Complete Reference by Herbert Shield

(2) Database Programming with JDBC and Java by George Reese

(3) Java and XML By Brett McLaughlin

(4) Wikipedia, URL: http://www.wikipedia.org.

(5) Answers.com, Online Dictionary, Encyclopedia and much more, URL: http://www.answers.com

(6) Google, URL: http://www.google.co.in

(7)Project Management URL: http://www.startwright.com/project.htm