PRIVACY-PRESERVING USING HOMOMORPHIC ENCRYPTION IN A MULTIKEY ENVIRONMENT

A PROJECT REPORT

Submitted by

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

In

INFORMATION TECHNOLOGY

C.ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY, MELVISHARAM, VELLORE-632509

ANNA UNIVERSITY:: CHENNAI 600 025

JUNE 2023

ANNA UNIVERSITY:CHENNAI 600 025

BONAFIDE CERTIFICATE

ACKNOWLEDGEMENT

ABSTRACT

Computing power of the cloud has been an important new source of energy because of its excellent capabilities of storing, analyzing, and processing a great deal of data. Customers with limited computing resources may resort to the cloud to perform some association rule mining tasks. Data owners may have a risk of personal sensitive information leakage in this process.

To preserve privacy in outsourced data, data owners may encrypt raw data before uploading. Data analysis of encrypted data is a challenge that has attracted the attention of many researchers in recent years. Homomorphic encryption is a cryptographic tool, which is one of the ways to solve this challenge. It allows data processing of encrypted data without decryption. Researching homomorphic encryption schemes that support privacy-preserving data mining in a multikey environment has become a significant direction. In this project, we propose a novel homomorphic cryptosystem, which supports multiple cloud users to have different public keys.

Besides, we propose a privacy-preserving association rule mining scheme on outsourced data uploaded from multiple parties in twin-cloud architecture.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DB	**:**Data Base
JVM	**:**Java Virtual Machine
JSM	**:**Java Server Page
CB	**:** Collective Behaviour
SB	**:**Social Dimension
JRE	**:**Java Runtime Environment
SSD	**:** Sparse Social Dimension
LGB	**:** Line Graph Partition

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Mining encompasses various algorithms such as clustering, classification, association rule mining and sequence detection. Traditionally, all these algorithms have been developed within a centralized model, with all data being gathered into a central site, and algorithms being run against that data. Privacy concerns can prevent this approach – there may not be a central site with authority to see all the data. We present a privacy preserving algorithm to mine association rules from vertically partitioned data.

There are more realistic examples. In the sub-assembly manufacturing process, different manufacturers provide components of the finished product.Cars incorporate several subcomponents; tires, electrical equipment, etc.; made by independent producers. Again, we have proprietary data collected by several parties, with a single key joining all the data sets, where mining would help detect/predict malfunctions.The recent trouble between Ford Motor and Firestone Tire provide a real-life example. Ford Explorers with Firestone tires from a specific factory had tread separation problems in certain situations, resulting in 800 injuries. Since the tires did not have problems on other vehicles, and other tires on Ford Explorers did not pose a problem, neither side felt responsible.

1.2 NEED FOR THE PROJECT

Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers to develop more effective marketing strategies, increase sales and decrease costs.

1.3 OBJECTIVE OF THE PROJECT

Data mining has opened a world of possibilities for business. This field of computational statistics compares millions of isolated pieces of data and is used by companies to detect and predict consumer behaviour. Its objective is to generate new market opportunities. Data mining converts information into knowledge.

1.4 SCOPE OF THE PROJECT

- The overall code of this project is organised in a very efficient way.
- Identifying patterns and trends from large quantities of data.
- Best possible techniques have been used to enhance the overall performance of the project.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

2.1.1 HISTORICAL BACKGROUND

Survey 1:

TITLE:"Using association rules for product assortment decisions: A case study".

YEAR: 2001

AUTHOR: Tom Brijs, Gilbert Swinnen, Koen Vanhoof

DESCRIPTION:

It has been claimed that the discovery of association rules is well-suited for applications of market basket analysis to reveal regularities in the purchase behaviour of customers. Moreover, recent work indicates that the discovery of interesting rules can in fact only be addressed within a microeconomic framework. This study integrates the discovery of frequent item sets with a (microeconomic) model for product selection (PROFSET). The model enables the integration of both quantitative and qualitative (domain knowledge) criteria. Sales transaction data from a fully automated convenience store is used to demonstrate the effectiveness of the model against a heuristic for product selection based on product-specific profitability. We show that with the use of frequent item sets we are able to identify the cross-sales potential of product items and use this information for better product selection.

Survey 2:

TITLE: Web mining: Pattern discovery from World Wide Web transactions

YEAR:1999

AUTHOR: BamshadMobasher, Jaideep Srivastava

DESCRIPTION:

Web-based organizations often generate and collect large volumes of data in their daily operations. Analyzing such data can help these organizations to determine the life time value of clients, design cross marketing strategies across products and services, evaluate the effectiveness of promotional campaigns, and

find the most effective logical structure for their Web space. This type of analysis involves the discovery of meaningful relationships from a large collection of primarily unstructured data, often stored in Web server access logs. We propose a framework for Web mining, the applications of data mining and knowledge discovery techniques to data collected in World Wide Web transactions. We present data and transaction models for various Web mining tasks such as the discovery of association rules and sequential patterns from the Web data. We also present a Web mining system, WEBMINER, which has been implemented based upon the proposed framework

Survey 3:

TITLE:"On data banks and privacy homomorphisms"

YEAR: 1978

AUTHOR: Ernest F. Brickell, Ernest F. Brickell

DESCRIPTION:

An additive privacy homomorphism is an encryption function in which the decryption of a sum (or possibly some other operation) of ciphers is the sum of the corresponding messages. Rivest, Adleman, and Dertouzos have proposed four different additive privacy homomorphisms. In this paper, we show that two of them are insecure under a ciphertext only attack and the other two can be broken by a known plaintext attack. We also introduce the notion of an R -additive privacy homomorphism, which is essentially an additive privacy homomorphism in which only at most R messages need to be added together. We give an example of an R -additive privacy homomorphism that appears to be secure against a ciphertext only attack.

Survey 4:

TITLE:"Towards semantically secure outsourcing of association rule mining on categorical data".

YEAR: 2014

AUTHOR: Jian Weng, Yingjiu Li

DESCRIPTION:

When outsourcing association rule mining to cloud, it is critical for data owners to protect both sensitive raw data and valuable mining results from being snooped at cloud servers. Previous solutions addressing this concern add random noise to the raw data and/or encrypt the raw data with a substitution mapping. However, these solutions do not provide semantic security; partial information about raw data or mining results can be potentially discovered by an adversary at cloud servers under a reasonable assumption that the adversary knows some plaintext–ciphertext pairs. In this paper, we propose the first semantically secure solution for outsourcing association rule mining with both data privacy and mining privacy. In our solution, we assume that the data is categorical. Additionally, our solution is sound, which enables data owners to verify whether there exists any false data in the mining results returned by a cloud server. Experimental study shows that our solution is feasible and efficient.

Survey 5:

TITLE: "Cryptanalysis of a symmetric fully homomorphic encryption scheme"

YEAR: 2018

AUTHOR: BaocangWang, Yu Zhan

DESCRIPTION:

Fully homomorphic encryption supports meaningful computations on encrypted data, and hence, is widely used in cloud computing and big data

environments. Recently, Li et al. constructed an efficient symmetric fully homomorphic encryption scheme and utilized it to design a privacy-preserving-outsourced association rule mining scheme. Their proposal allows multiple data owners to jointly mine some association rules without sacrificing the data privacy. The security of the homomorphic encryption scheme against the known-plaintext attacks was established by examining the hardness of solving nonlinear systems. However, in this paper, we illustrate that the security of Li et al.'s homomorphic encryption is overvalued. First, we show that we can recover the first part of the secret key from several known plaintext/ciphertext pairs with the continued fraction algorithm. Second, we find that we can retrieve the second part of the secret key through the Euclidean algorithm for the greatest common divisor problem. Experiments on the suggested parameters demonstrate that in case of more than two homomorphic multiplications, all the secret keys of the randomly instantiated Li et al.'s encryption schemes can be very efficiently recovered, and the success probability is at least 98% for one homomorphic multiplication.

2.1.2 DISADVANTAGES

- It has less accuracy
- Low performance on maintain the large amount of data.
- It’s not cost effective and time consuming.
- Less reliable to use.

3. PROPOSED SYSTEM

In this project we propose a novel homomorphic cryptosystem, which supports multiple cloud users to have different public keys. Besides, we propose a privacy-preserving association rule mining scheme on outsourced data uploaded. To preserve privacy in outsourced data, data owners may encrypt raw data before uploading. Researching homomorphic encryption schemes that support privacy-preserving data mining in a multikey environment has become a significant direction. Cloud computing tasks include data classification, cluster analysis, data mining, and so on. Data mining is a technology that searches for information hidden in mass data, and association rule mining is one of the classic algorithms in data mining. Association rule mining can find valuable associations between items from the database.

The problem with encrypting data is that sooner or later, you have to decrypt it. And decrypting data makes it vulnerable to hackers. You can keep your cloud files cryptographically scrambled using a secret key, but as soon as you want to actually do something with those files—anything from editing a word document or querying a database of financial data—you have to unlock the data and leave it vulnerable. Homomorphic encryption, an advancement in the science of cryptography, could change that.

AES ALGORITHM

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

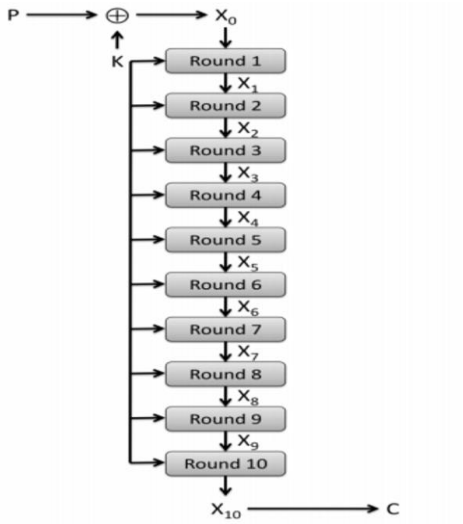
Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

Working of AES Algorithm

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

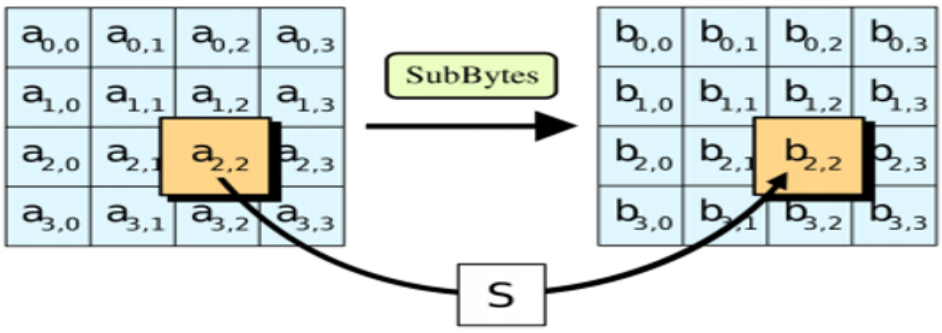


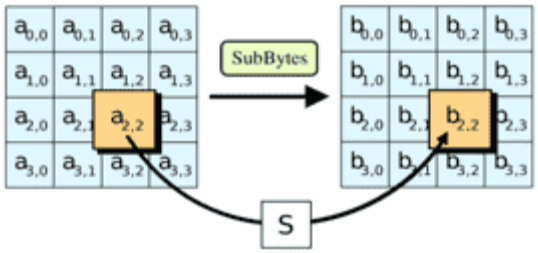
Steps in each round

Each round in the algorithm consists of four steps.

1. Substitution of the bytes

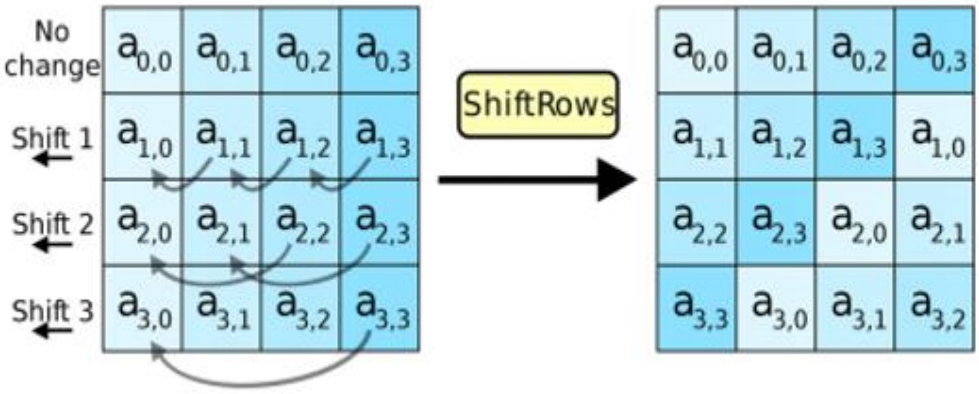
In the first step, the bytes of the block text are substituted based on rules dictated by predefined S-boxes (short for substitution boxes).

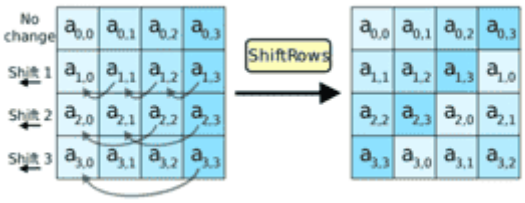




2. Shifting the rows

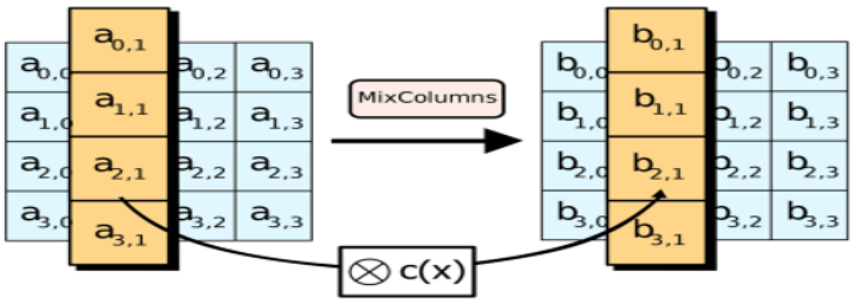
Next comes the permutation step. In this step, all rows except the first are shifted by one, as shown below.

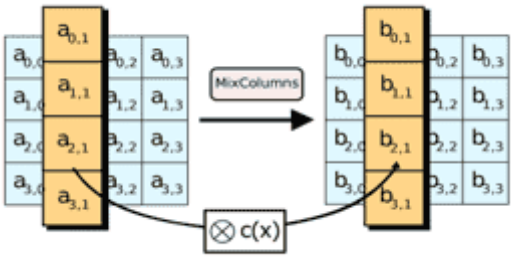




3. Mixing the columns

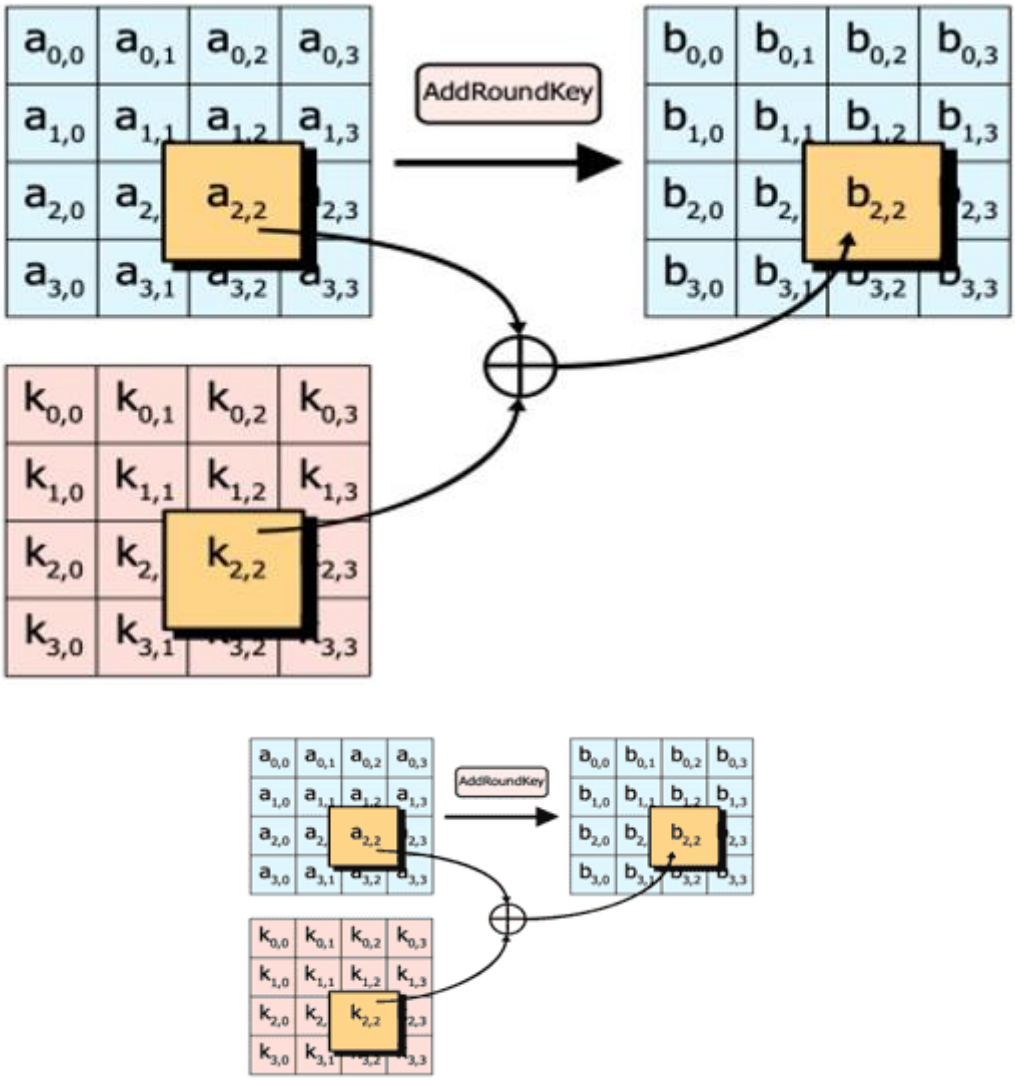
In the third step, the [Hill cipher] is used to jumble up the message more by mixing the block's columns.





4. Adding the round key

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



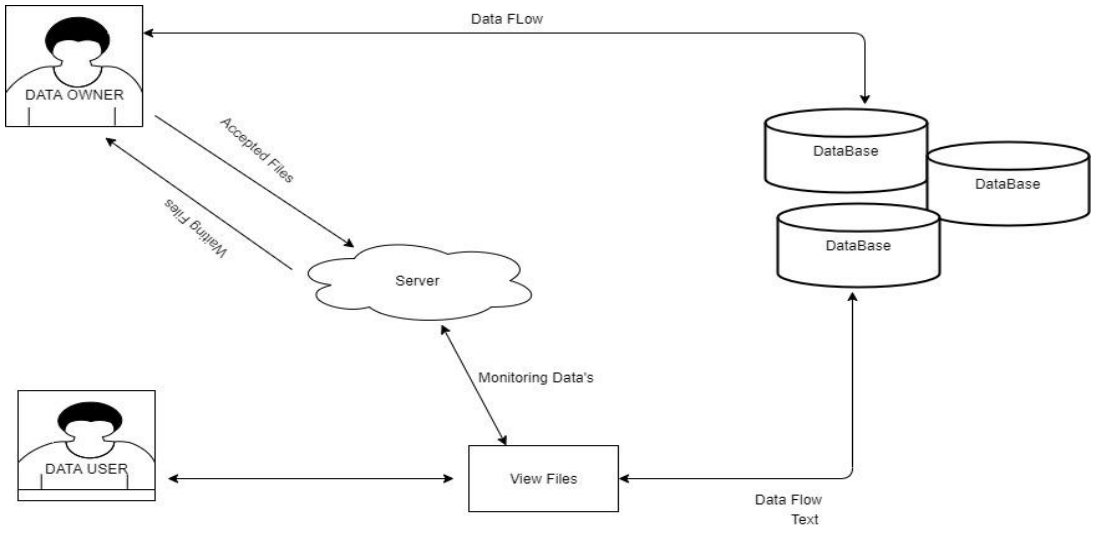
When done repeatedly, these steps ensure that the final ciphertext is secure.

AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.

However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

System Architecture



Association Rule Mining

Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and otherforms of data repositories.

Given a set of transactions, association rule mining aims to find the rules which enable us to predict the occurrence of a specific item based on the occurrences of the other items in the transaction.

Procedure of Association Rule Mining

Frequent itemset lattice, where the color of the box indicates how many transactions contain the combination of items. Note that lower levels of the lattice can contain at most the minimum number of their parents' items; e.g. {ac} can have only at most $min(a, c)$ items. This is called the *downward-closure property*.

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all *frequent itemsets* in a database.
2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

While the second step is straightforward, the first step needs more attention. Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over I and has size $2^n - 1$ (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items n in I, efficient search is possible using the ***downward-closure property*** of support (also called *anti-monotonicity*) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets.

Steps involved in Association Rule Mining

Association Rule Mining can be described as a two-step process.

Step 1: Find all frequent itemsets.

*An **itemset** is a set of items that occurs in a shopping basket.*

A set of items in a shopping basket can be referred to as an itemset. It can consist of any number of products. For example, [bread, butter, eggs] is an itemset from a supermarket database.

A frequent itemset is one that occurs frequently in a database. This begs the question of how frequency is defined. This is where **support count** comes in.

*The **support count** of an item is defined as the frequency of the item in the dataset.*

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Itemset	Support Count
[Beer]	3
[Bread]	4
[Cola]	2
[Diapers]	4
[Milk]	4
[Eggs]	1

Itemsets and their respective support counts

The support count can only speak for the frequency of an itemset. It does not take into account relative frequency i.e., the frequency with respect to the number of transactions. This is called the support of an itemset.

***Support** of an itemset is the frequency of the itemset with respect to the number of transactions.*

$$\text{Support (Itemset)} = \frac{\text{Frequency of Itemset (Support Count)}}{\text{Total Number of Transactions}}$$

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Itemset	Support	%
[Beer]	3/5	0.6
[Bread]	4/5	0.8
[Cola]	2/5	0.4
[Diapers]	4/5	0.8
[Milk]	4/5	0.8
[Eggs]	1/5	0.2

Itemsets and their respective supports

Consider the itemset [Bread] which has 80% support. This means that in every 100 transactions, bread occurs 80 times.

Defining support as percentage helps us set a threshold for frequency called **min_support**. If we set support at 50%, this means that we define a frequent itemset as one that occurs at least 50 times in 100 transactions. For instance, for the above dataset, we set threshold_support at 60%.

60% minimum support ⇨ 60% of (total # of transactions) ⇨ 0.6 x 5 = 3

For an Itemset to be be frequent, it should occur at least 3 times in 5 transactions in the given dataset.

We always **eliminate those items whose support is less than min_support** as is seen from the greyed-out parts of the table above. The generation of frequent itemsets depends on the algorithm used.

Step 2: Generate strong association rules from the frequent itemsets.

Association rules are generated by building associations from frequent itemsets generated in step 1. This uses a measure called confidence to find strong associations. This has been covered in the example section.

The main applications of association rule mining:

- Basket data analysis - is to analyse the association of purchased items in a single basket or single purchase as per the examples given above.
- Cross marketing - is to work with other businesses that complement your own, not competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
- Catalog design - the selection of items in a business' catalog are often designed to complement each other so that buying one item will lead to buying of another. So these items are often complements or very related.

Advantages of Proposed System

- Increase accuracy
- Increase performance

4. SYSTEM ANALYSIS

4.1 REQUIREMENTS SPECIFICATIONS

4.1.1 HARDWARE REQUIREMENTS

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

Processor	:	Pentium Dual Core 2.00GHZ
Hard disk	:	40 GB
Mouse	:	Logitech.
RAM	:	4 GB (minimum)
Keyboard	:	110 keys enhanced.

4.1.2 SOFTWARE REQUIREMENTS

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

Operating system	:	Windows 7(Service Pack 1), 8, 8.1and 10
Front End	:	Html, CSS, Bootstarp, Javascript
Coding Language	:	Java Servlets
Backend	:	MySQL

Software Tool : ECLIPSE or NetBeans

4.2 SOFTWARE DESCRIPTION

4.2.1 FEATURES OF JAVA

4.2.1.1 JAVA TECHNOLOGY

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others.

4.2.1.2 Java Language

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1991 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere."

It's also important to understand that Java is much different from JavaScript. Javascript does not need to be compiled, while Java code does need to be compiled. Also, Javascript only runs on web browsers while Java can be run anywhere.

4.2.1.3 Java Programming Characteristics

- In Java everything is object oriented. Java can be easily extended since it is based on the Object model.
- Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code.

- Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- With the use of Just-In-Time compilers, Java enables high performance.
- Java is designed for the distributed environment of the internet.

OBJECTIVES OF JAVA

To see places of Java in Action in our daily life, explore java.com.

Why Software Developers Choose Java

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

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

Some Ways Software Developers Learn Java

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

Object Oriented

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

1. Inheritance: It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding addition a features as needed.

2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.

3. Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

4. Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

4.2.1.4 Applications of Java Programming

Mobile Applications

Java is considered as the official programming language for mobile app development. It is compatible with software such as Android Studio and Kotlin. Now you must be wondering why only Java? The reason is that it can run on Java Virtual Machine(JVM), whereas Android uses DVK(Dalvik Virtual

Machine) to execute class files. These files are further bundled as Android application Package(APK). With Java and its OOPs principles, it provides better security and ease of simplicity with Android.

Web-based Applications

Java is also used to develop web applications. It provides a vast support for web applications through Servlets, Struts or JSPs. With the help of these technologies, you can develop any kind of web application that you require. The easy coding and high security offered by this programming language allow the development of a large number of applications for health, social security, education, and insurance.

Scientific Applications

Software developers see Java is the weapon of choice when it comes to coding the scientific calculations and mathematical operations. These programs are designed to be highly secure and lighting fast. they support a higher degree of portability and offer low maintenance. Some of the most powerful applications like the MATLAB use Java for interacting user interface as well as part of the core system.

Gaming Applications

Java has the support of the open-source most powerful 3D-Engine, the jMonkeyEngine that has the unparalleled capability when it comes to the designing of 3D games. However, it does cause an occasional latency issue for games as garbage collection cycles can cause noticeable pauses. This issue will be solved in the newer versions of JVMs.

4.2.2 ALGORITHMS

- AES Algorithm

AES ALGORITHM

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit

keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key

4.2.3 DATASET

A dataset in java is mostly used for providing a type of safe view to the data present as part of the SQL queries. It is part of a library called java.util.list which holds for the mostly parameterized types of data. When a select annotation for the method is selected then in that case the query parameter is used for any data class which have other access modifiers like the public to make the accessibility from the queries to the methods present within the class. A dataset in java can behave in either a connected or disconnected way.

4.2.3.1Need of Dataset

- Dataset java is a type of public interface which allows passing parameterized type data at the time of SQL query to the interface using select annotation.
- The query can be performed in either of the modes like it can be in connected mode or in disconnected mode. If any function is getting in connected mode, then in that scenario dataset is like a result set.
- If any function is getting in a disconnected dataset then in that scenario the function will get reflected as CachedRowSet.
- Dataset java nowadays is found in gel with machine learning as well because the entire area of machine learning basically deals with a huge amount of data which makes it required for those data to have a view and manipulations.
- In fact, Dataset Java also helps in building many schemas with a view for dealing with the sql interface that helps in providing many exceptions and connection issues catching prior these issues help in getting the entire implementation and development a safer action from security breaches.

5. SYSTEM ARCHITECTURE & DESIGN

5.1 SYSTEM ARCHITECTURE

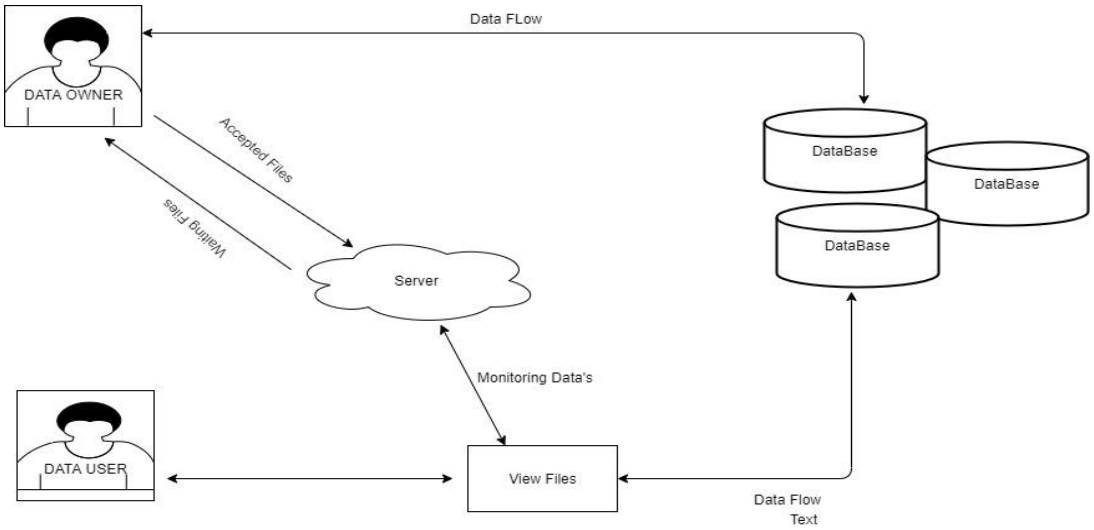


Fig 5.7 System Architecture

5.2USE CASE DIAGRAM

Use case diagrams are a way to capture the system's functionality and requirements in UML diagrams. It captures the dynamic behavior of a live system. A use case diagram consists of a use case and an actor.

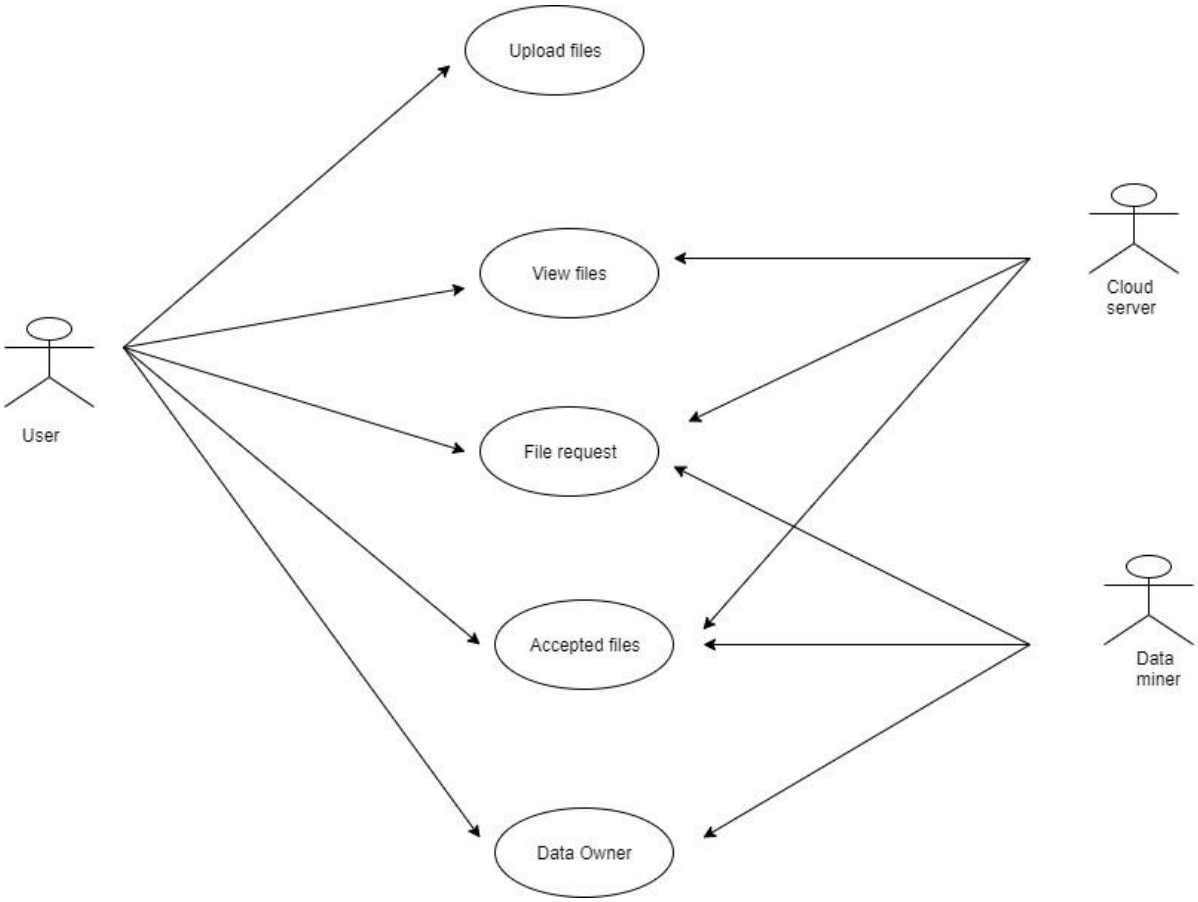


Fig 5.2 Use Case Diagram

5.3 DATA FLOW DIAGRAM

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

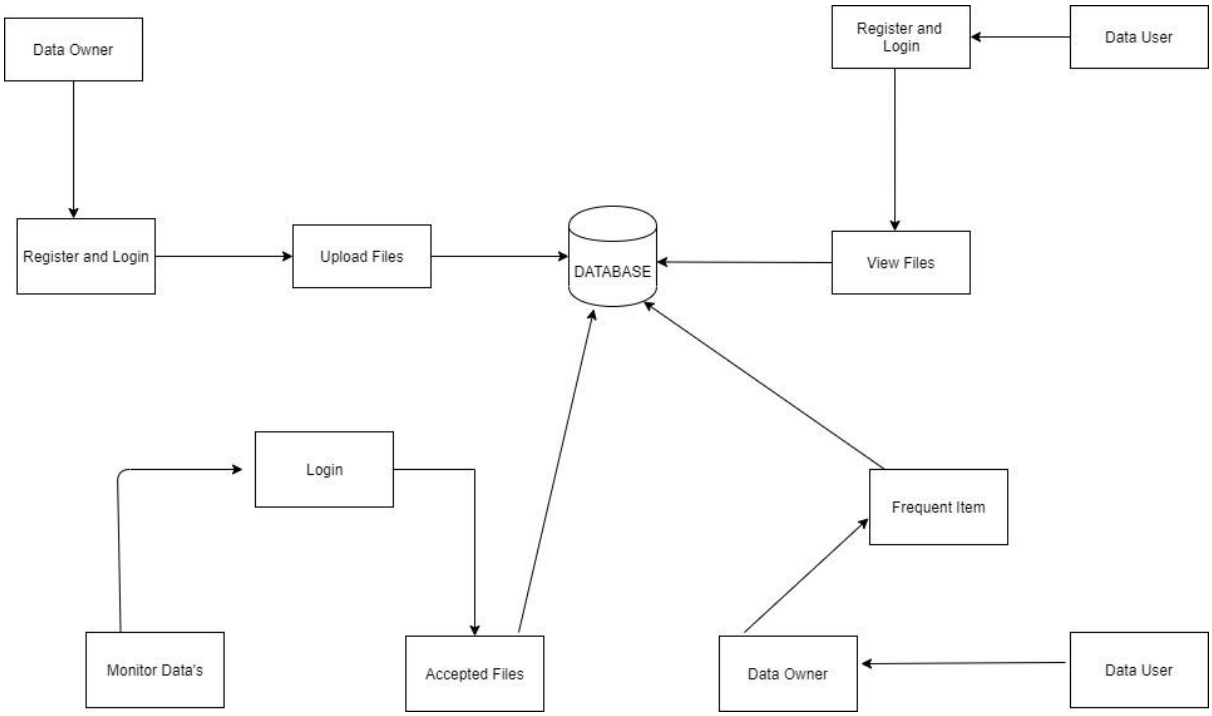


Fig 5.3 Data Flow Diagram

5.4 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

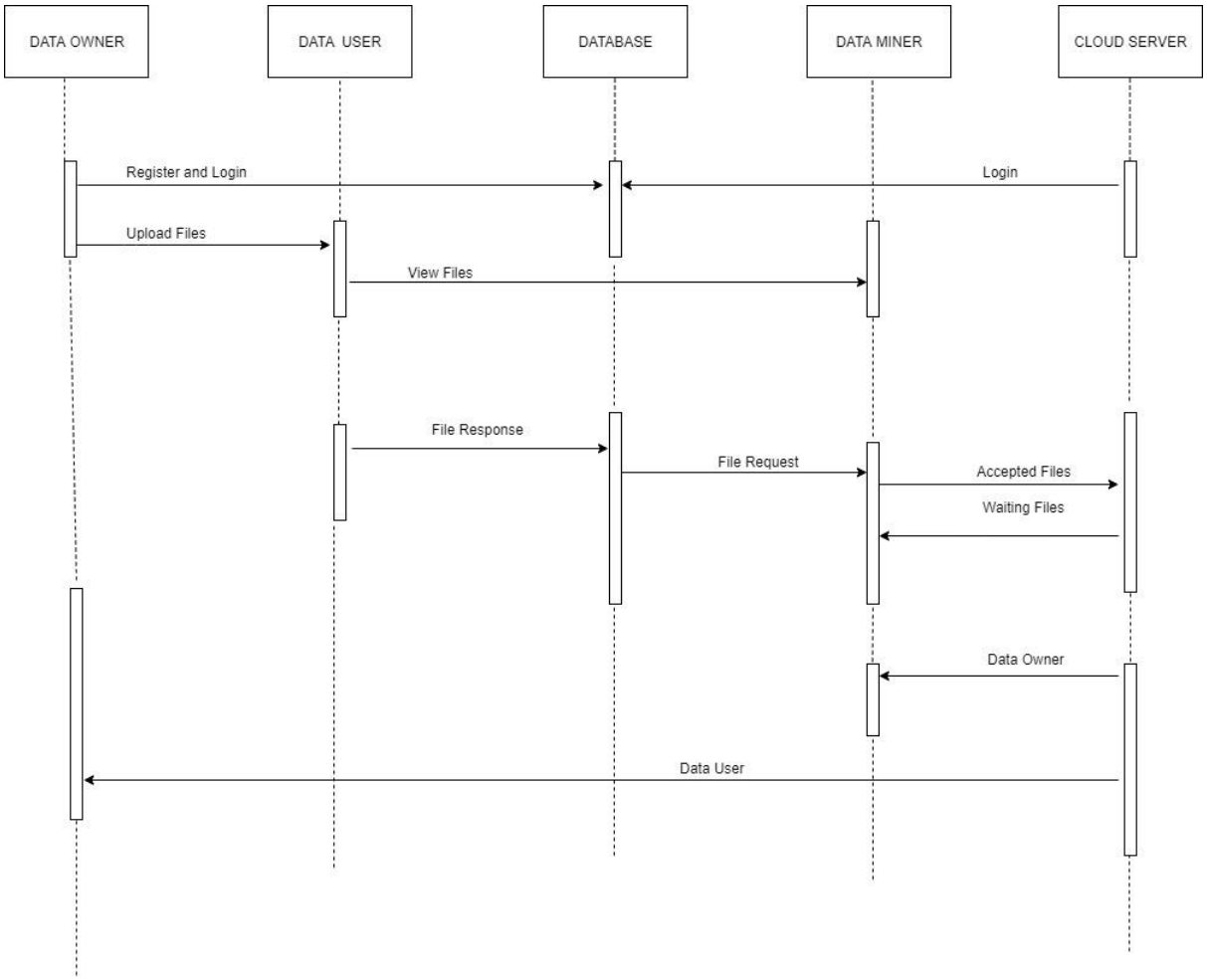


Fig 5.4 Sequence Diagram

5.5 COLLABORATION DIAGRAM

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

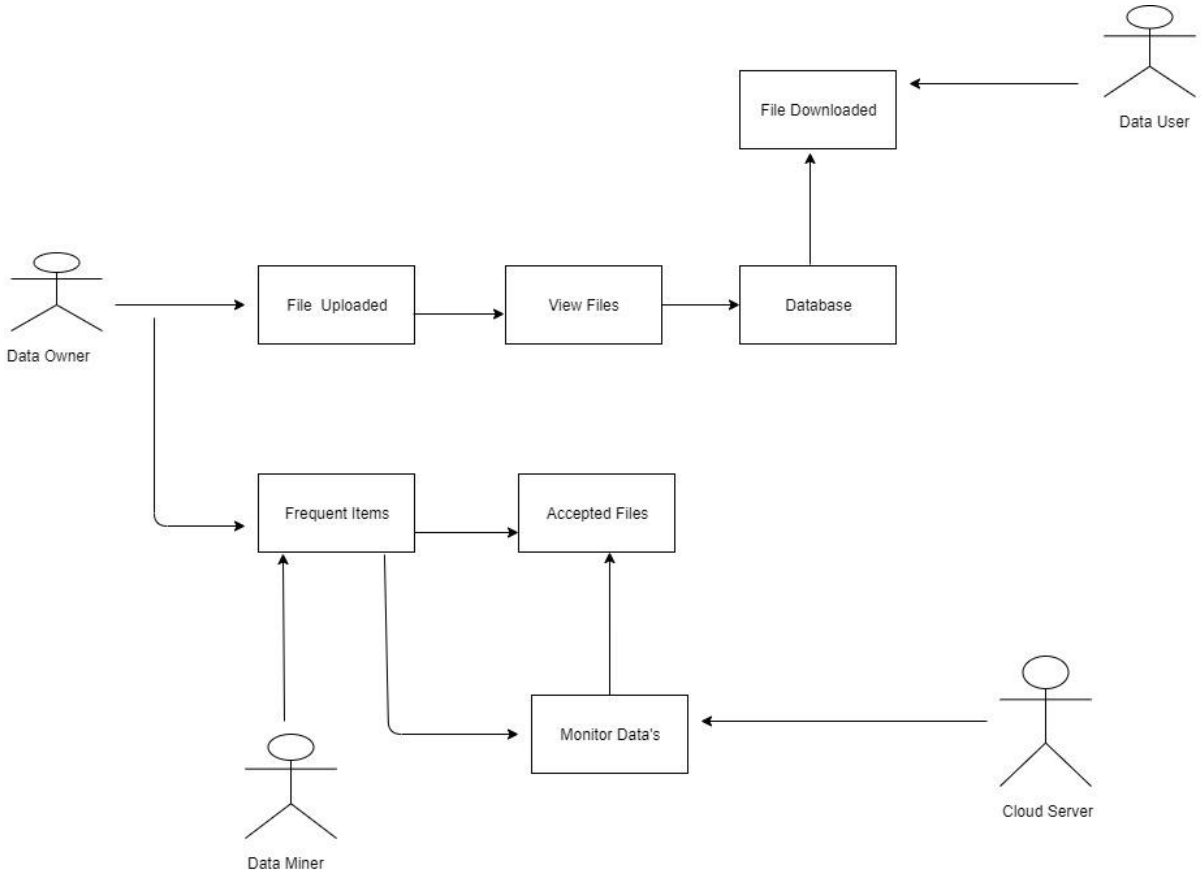


Fig 5.5 Collaboration Diagram

5.6 CLASS DIAGRAM

Class diagrams are the main building block in object-oriented 29odelling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them.

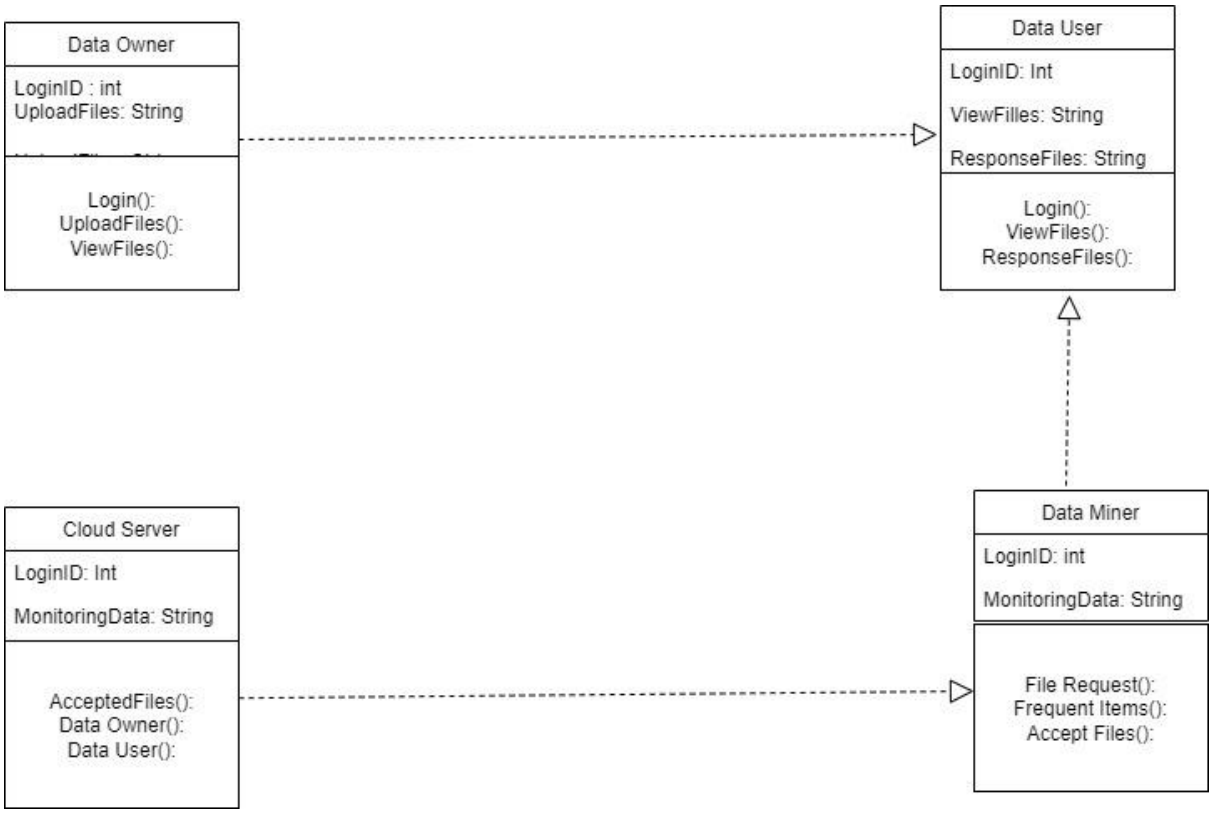


Fig 5.6 Class Diagram

5.7ACTIVITY DIAGRAM

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination**.**

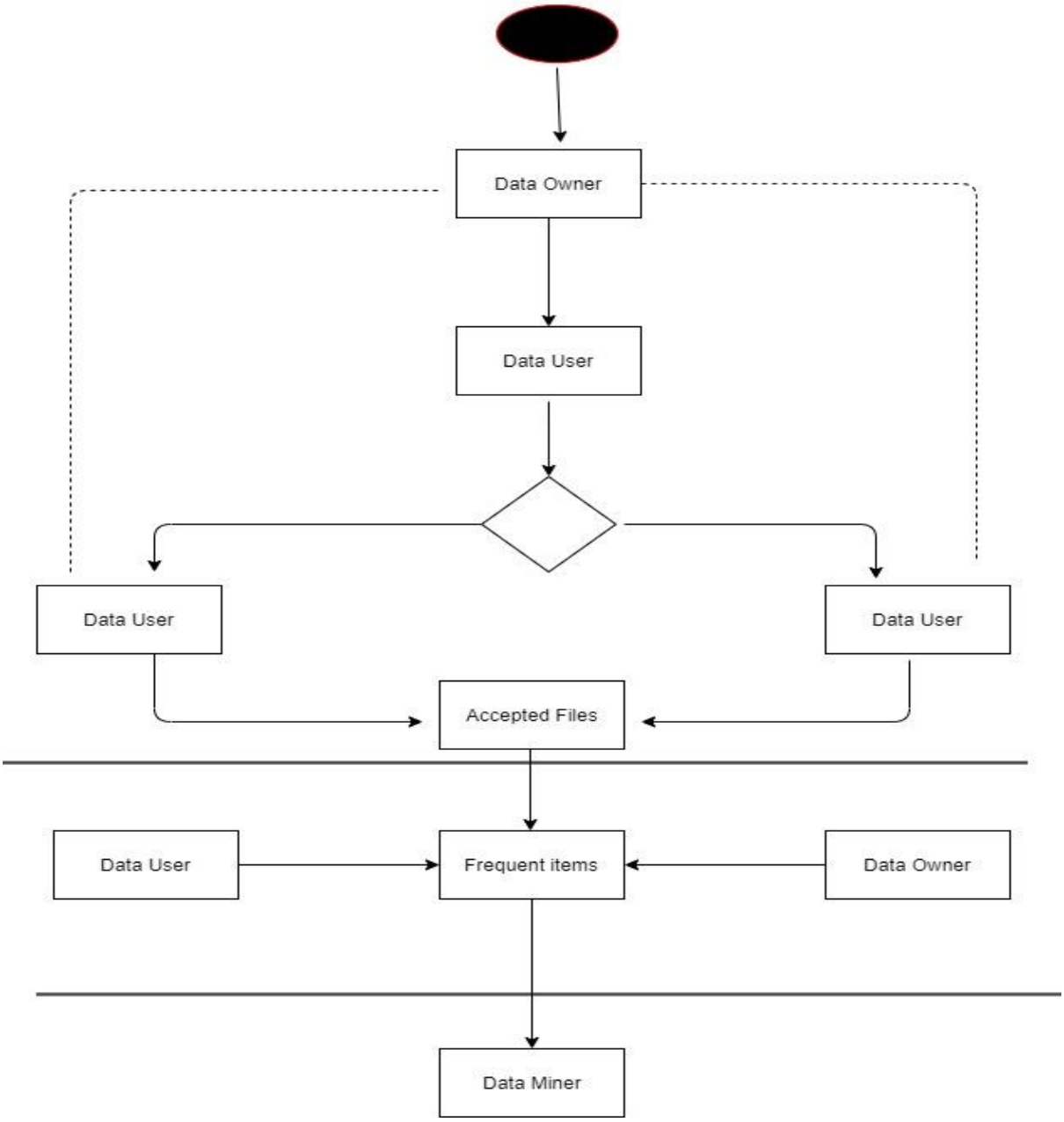


Fig 5.1 Activity Diagram

5.8 OBJECT DIAGRAM

An object diagram shows this relation between the instantiated classes and the defined class, and the relation between these objects in the system. They are be useful to explain smaller portions of your system, when your system class diagram is very complex, and also sometimes modeling recursive relationship in diagram.

The best way to illustrate what an object diagram look like is to show the object diagram derived from the corresponding class diagram.

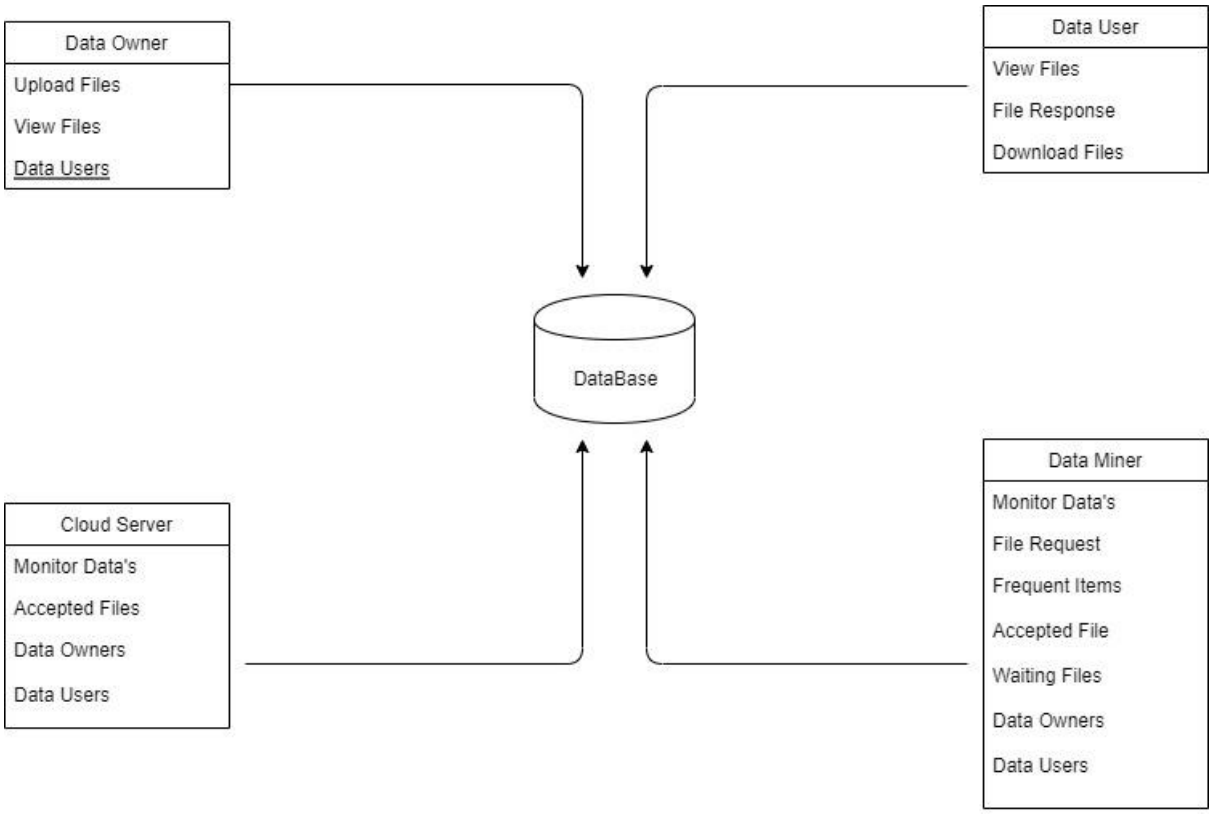


Fig 5.8 Object Diagram

5.9 STATE DIAGRAM

A state diagram, also known as a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language. Then, all of the possible existing states are placed in relation to the beginning and the end.

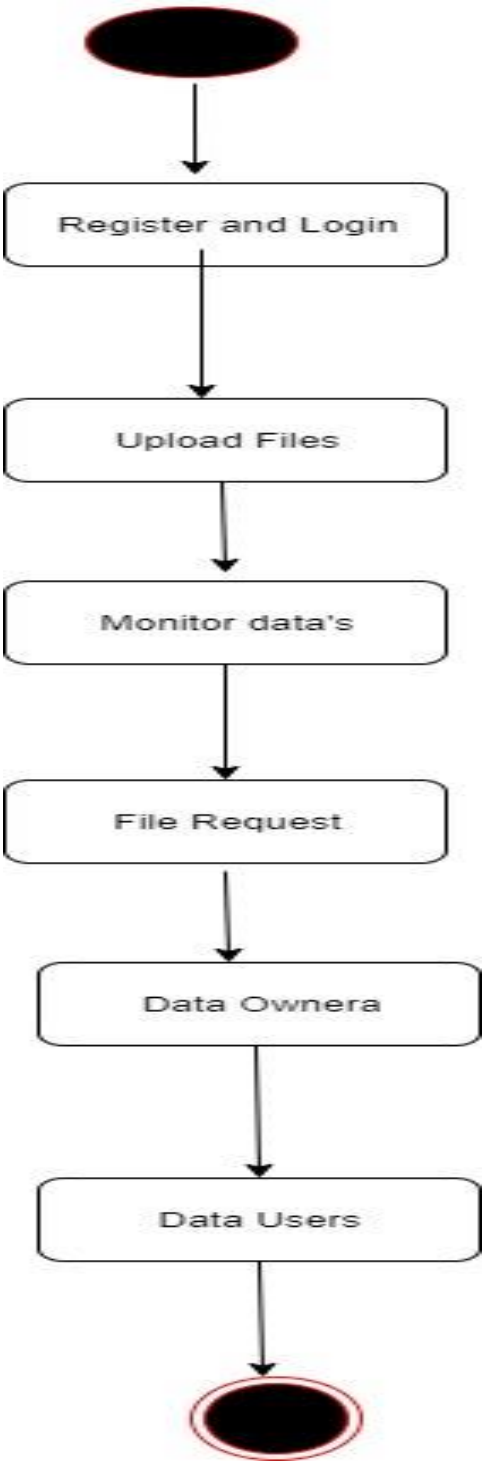


Fig 5.9 State Diagram

5.10 COMPONENT DIAGRAM

The Component diagrams are special type of UML diagrams used for different purposes. These diagrams show the physical components of a system. To clarify it, we can say that component diagrams describe the organization of the components in a system.

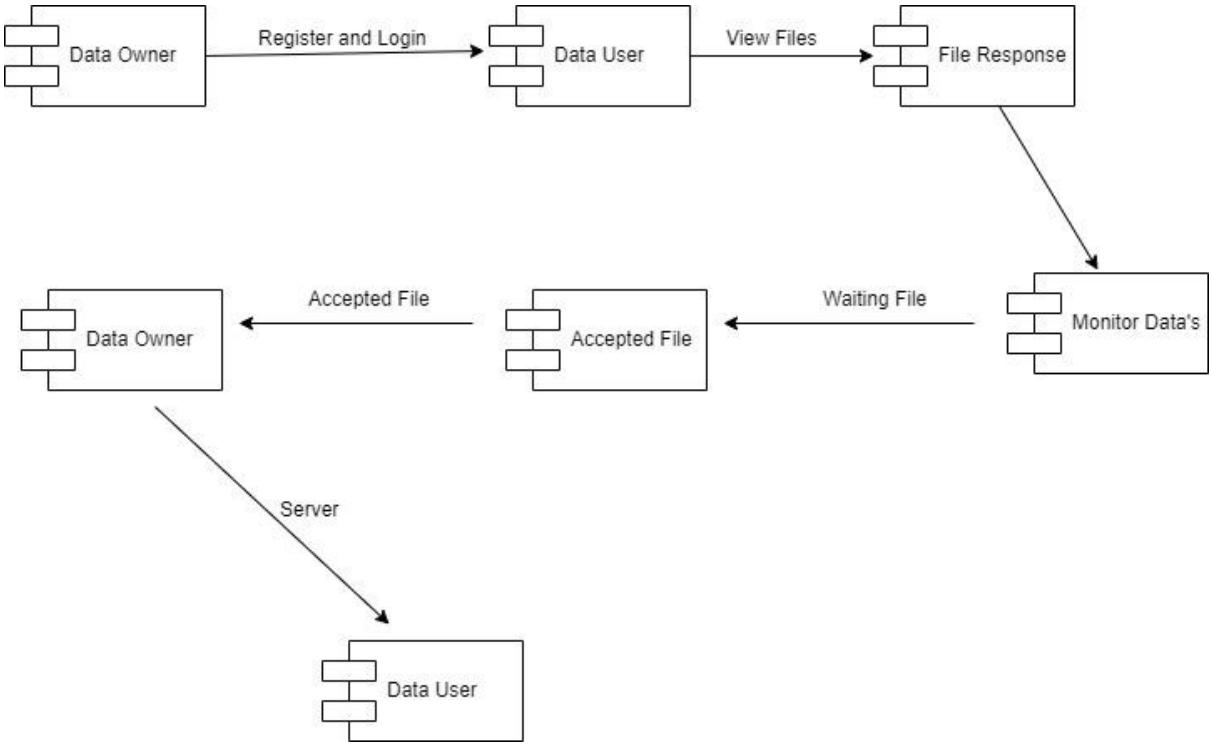


Fig 5.10 Component Diagram

5.11 ER-DIAGRAM

E-R Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

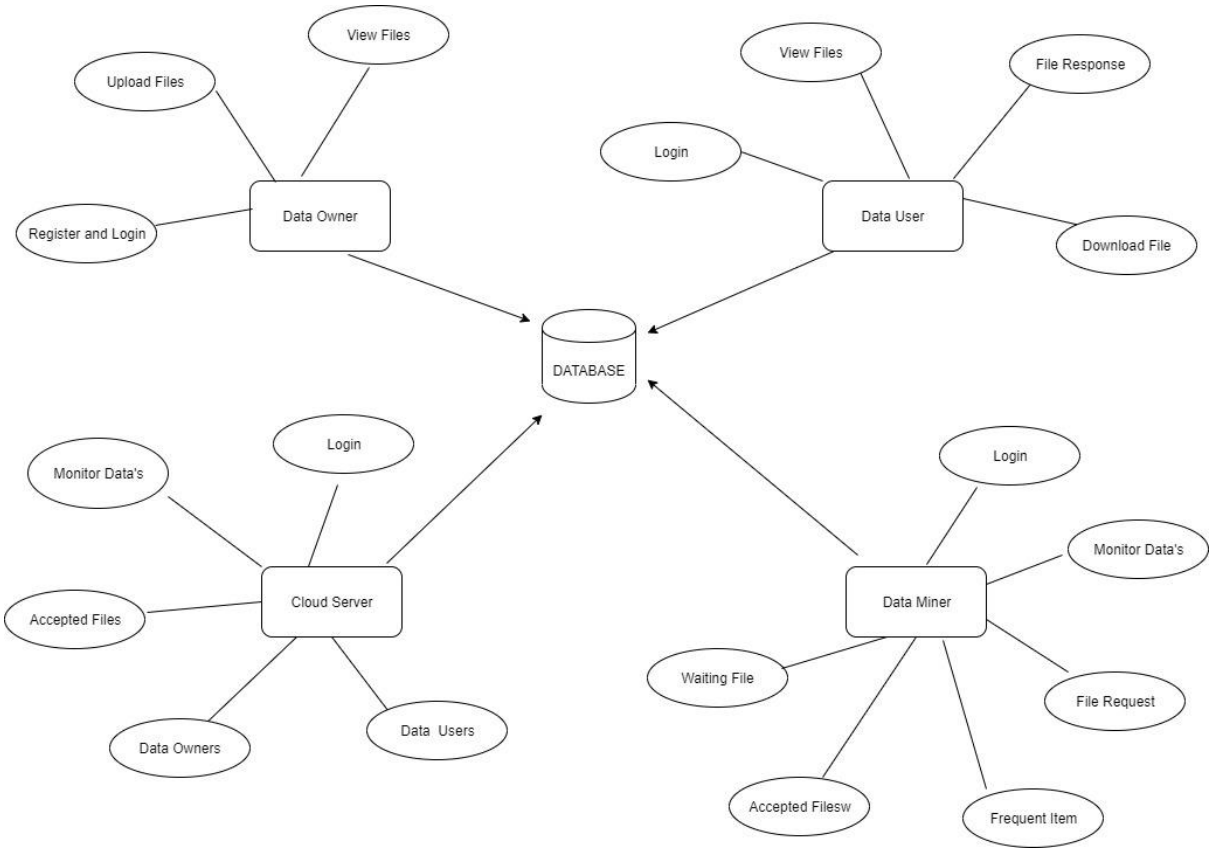


Fig 5.11 ER Diagram

6. MODULES

6.1 MODULE DESCRIPTION

- Data Owner
- Data Miner
- Data User
- Cloud Server

6.1.1DATA OWNER:

- Data owner will uploading file into the cloud using some key for encrypting the particular text document. The data owner can upload and download the file.

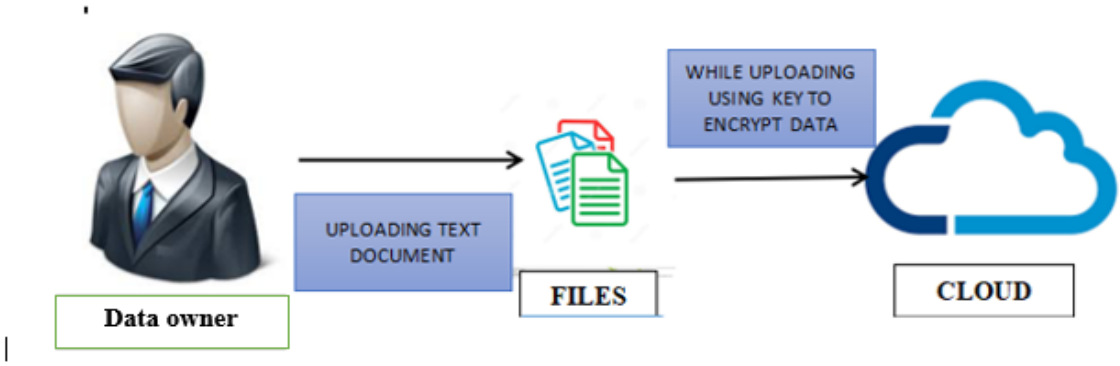


Fig 6.1.1 Data Owner

6.1.2DATA MINER:

- Data miner login then accept the request given by the users and maintain process like accepted files, waiting files then can view the details of data owner and data users details.

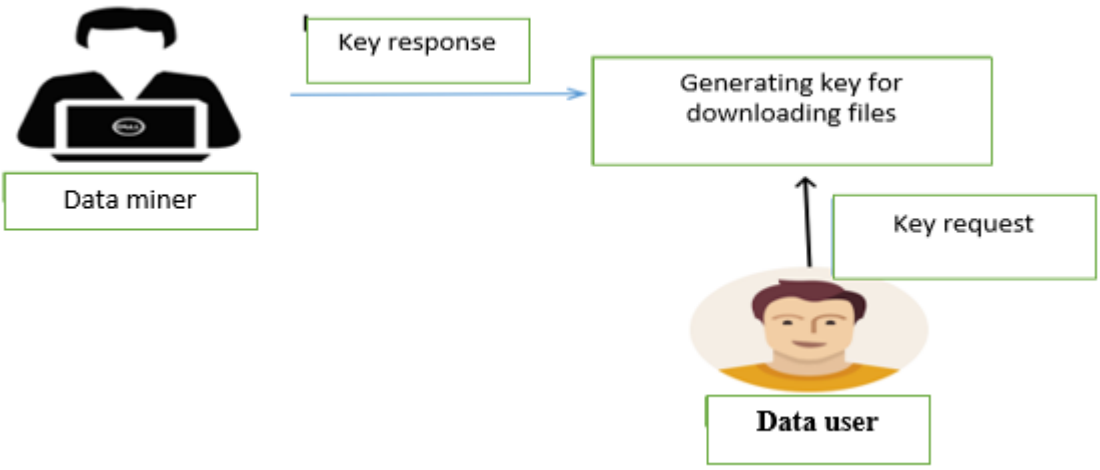


Fig 6.1.2 Data Miner

6.1.3DATA USER DOWNLOADING FILES USING KEY

- The data user downloading files before that they need to request the key from the auditor after that key will be apply in the cloud finally they can download required file.

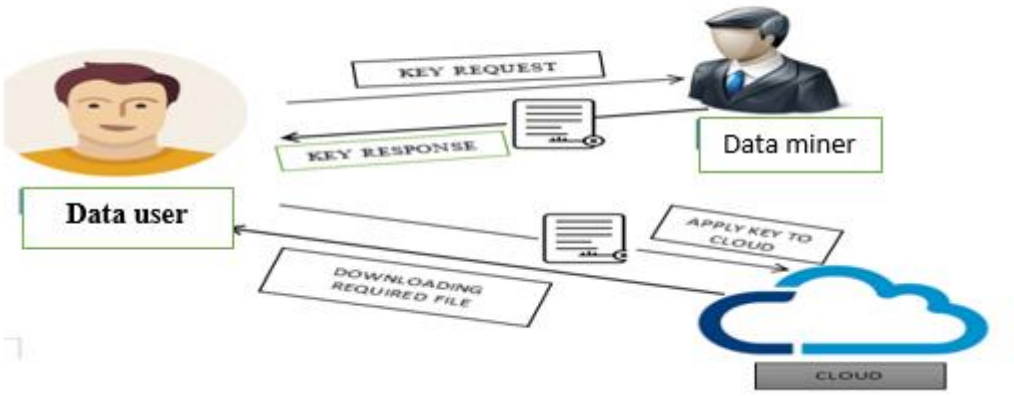


Fig 6.1.3 Data User

6.1.4CLOUD SERVER:

- Cloud server login then maintain the files uploaded by the data owner, and details of data owner and data user, accepted files.
- A cloud server is a pooled, centralized server resource that is hosted and delivered over a network—typically the Internet—and accessed on demand by multiple users.
- Cloud servers can perform all the same functions of a traditional physical server, delivering processing power, storage and applications.

7.IMPLEMENTATION AND TESTING

7.1 IMPLEMENTATION

With the development of big data and cloud computing, data analysis technologies play an important role to produce huge market values. Customers with limited computing resources may resort to the cloud to perform some association rule mining tasks. Data owners may have a risk of personal sensitive information leakage in this process. To preserve privacy in outsourced data, data owners may encrypt raw data before uploading. Data analysis of encrypted data is a challenge that has attracted the attention of many researchers in recent years. Homomorphic encryption is a cryptographic tool, which is one of the ways to solve this challenge. It allows data processing of encrypted data without decryption. Researching homomorphic encryption schemes that support privacy-preserving data mining in a multikey environment has become a significant direction. In this article, we propose a novel homomorphic cryptosystem, which supports multiple cloud users to have different public keys. Besides, we propose a privacy-preserving association rule mining scheme on outsourced data uploaded from multiple parties in a twin-cloud architecture. Our scheme uses a transaction record representation method in databases for large shopping malls based on real-world situations, and our experiments on a real transaction database show that our technology is reasonably feasible.

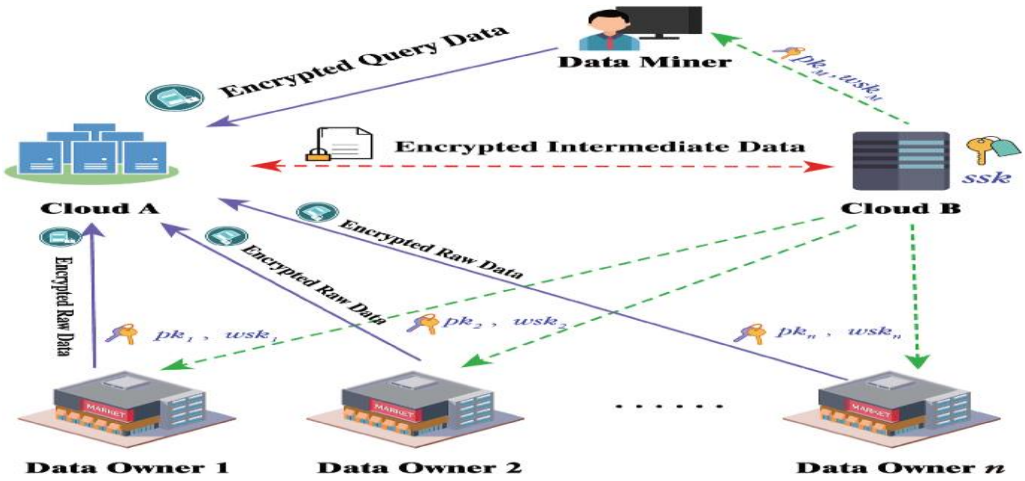


Fig 7.1 Implementation

7.2 TESTING

In this paper, we develop achievability protocols and outer bounds for the secure network coding setting, where the edges are subject to packet erasures, and public feedback of the channel state is available to both Eve and the legitimate network nodes. Secure network coding assumes that the underlying network channels are error-free; thus, if our channels introduce errors, we need to first apply a channel code to correct them, and then build security on top of the resulting error-free network. We show that by leveraging erasures and feedback, we can achieve secrecy rates that are in some cases multiple times higher than the alternative of separate channel-error-correction followed by secure network coding; moreover, we develop outer bounds and prove optimality of our proposed schemes in some special cases.

Types of Testing

7.2.1 Unit Testing

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

7.2.2 Functional Testing

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

Functional testing is cantered on the following items:

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

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

Functions : identified functions must be exercised.

Output :identified classes of application outputs must be exercised.

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

7.2.3 System Testing

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

7.2.4 Performance Testing

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

7.2.5 Integration Testing

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

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

7.2.6 Acceptance Testing

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

Acceptance testing for Data Synchronization:

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

7.3 Build the test plan

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

8. CONCLUSION

In privacy preserving association researching homomorphic encryption schemes that support privacy-preserving data mining in a multikey environment has become a significant direction. In this we propose a novel homomorphic cryptosystem, which supports multiple cloud users to have different public keys.

In this project, we develop achievability protocols and outer bounds for the secure network coding setting, where the edges are subject to packet erasures, and public feedback of the channel state is available to both Eve and the legitimate network nodes. Secure network coding assumes that the underlying network channels are error-free; thus, if our channels introduce errors, we need to first apply a channel code to correct them, and then build security on top of the resulting error-free network. We show that by leveraging erasures and feedback, we can achieve secrecy rates that are in some cases multiple times higher than the alternative of separate channel-error-correction followed by secure network coding

Data owner will uploading file into the cloud using some key for encrypting the particular text document. The data owner can upload and download the file. Data miner login then accept the request given by the users and maintain process like accepted files, waiting files then can view the details of data owner and data users details.

9. FUTURE ENHANCEMENT

The PRIVACY-PRESERVING ASSOCIATION RULE MINING USING HOMOMORPHIC ENCRYPTION IN A MULTIKEY ENVIRONMENT plan to investigate some interesting future work as follows

In the future, we plan to research on applying the principles of privacy preserving homomorphic to implement and enhance future based security.

APPENDICES

APPENDIX 1

SAMPLE CODINGS

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Association_Rule_Mining | Home</title>
        <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">


<link rel="stylesheet"href="https://unpkg.com/swiper/swiper-bundle.min.css"
/>

<!-- font awesome cdn link  -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">

<!-- custom css file link  -->
<link rel="stylesheet"href="css/style.css">



</head>
<body>
<!-- header section starts  -->

<header>
```

```
<a href="#" class="logo"><i class="fas fa-
cloud"></i>Association_Rule_Mining</a>

<div id="menu-bar" class="fas fa-bars"></div>

<nav class="navbar">
<a href="Index.html">Home</a>
<a href="OwnerLogin.jsp">Data Owner</a>
<a href="UserLogin.jsp">Data User</a>
<a href="CloudServerLogin.jsp">Cloud Server</a>
<a href="DataMinerLogin.jsp">Data Miner</a>

</nav>

</header>

<!-- header section ends -->

<!-- home section starts  -->

<section class="home" id="home">

<div class="content">
<h3>Association Rule Mining </h3>
<p>Cloud computing is an evolving technology that provides data storage and
highly fast computing services at a very low cost. All data stored in the cloud is
handled by their cloud service providers or the caretaker of the cloud.</p>
```

 The data owner is concerned about the authenticity and reliability of the data stored in the cloud as the data owners. Data can be mis appropriated or altered by any unauthorized user or person!

<!--get started -->

<!--

<imgsrc="images1/c-1.jpg" alt="">

<imgsrc="images1/c-2.jpg" alt="">

<imgsrc="images1/c-3.jpg" alt="">

<imgsrc="images1/c-4.jpg" alt="">

<imgsrc="images1/c-5.jpg" alt="">

-->

</section>

<!-- home section ends -->

DataOwnerRegisterServlet.java

```
package com.association.Servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.association.Bean.RegisterBean;
import com.association.Implementation.AssociationImplementation;
import com.association.Interface.AssociationInterface;
/**
 * Servlet implementation class DataOwnerRegisterServlet
 */
@WebServlet("/DataOwnerRegisterServlet")
public class DataOwnerRegisterServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public DataOwnerRegisterServlet() {
super();
        // TODO Auto-generated constructor stub
    }
        /**
```

```
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
        String name = request.getParameter("name");
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String cpassword = request.getParameter("cpassword");
        String email = request.getParameter("email");
```

```
String contact = request.getParameter("contact");
String address = request.getParameter("address");

System.out.println(name+" "+username+" "+password+" "+cpassword+" "+email+" "+contact+" "+address);

if(password.equals(cpassword)){
    RegisterBeanrb = new RegisterBean();
    rb.setName(name);
    rb.setUsername(username);
    rb.setPassword(password);
    rb.setEmail(email);
    rb.setContact(contact);
    rb.setAddress(address);

    AssociationInterface ai = new AssociationImplementation();
    int i = ai.dataownerregister(rb);
    System.out.println("The value of i is: "+i);
    if(i == 1){
        response.sendRedirect("OwnerLogin.jsp");
    }else{
```

```
                    response.sendRedirect("Error.jsp");
                }
            }
        }
}
```

DataUserLoginServlet.java

```
package com.association.Servlet;


import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


import com.association.Bean.RegisterBean;
import com.association.Implementation.AssociationImplementation;
import com.association.Interface.AssociationInterface;
```

```
/**
 * Servlet implementation class DataUserLoginServlet
 */
@WebServlet("/DataUserLoginServlet")
public class DataUserLoginServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public DataUserLoginServlet() {
super();
        // TODO Auto-generated constructor stub
    }
	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
```

```
		response.getWriter().append("Served at: ").append(request.getContextPath());
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);

		String username = request.getParameter("username");
		String password = request.getParameter("password");

		System.out.println(username+" "+password);

		AssociationInterface ai = new AssociationImplementation();
		int i = ai.datauserlogin(username, password);
		System.out.println("The value of i is: "+i);
```

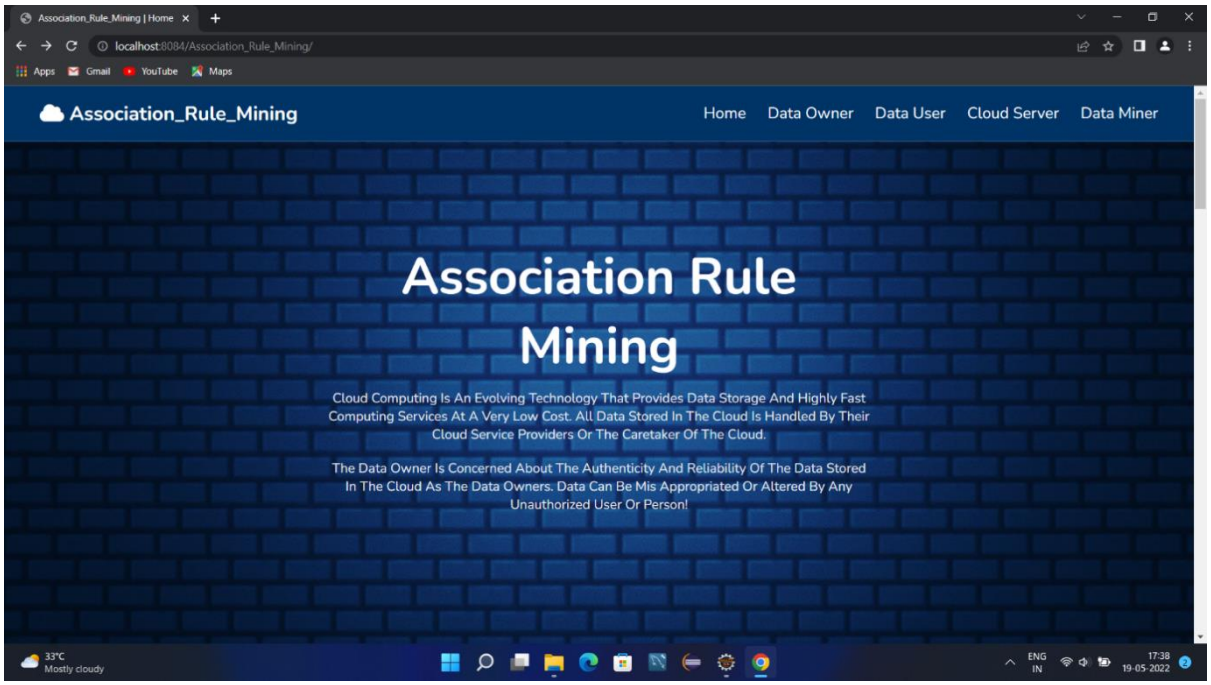
```
            if(i == 1){
                HttpSession session = request.getSession();
                session.setAttribute("duser", username);
                response.sendRedirect("UserHome.jsp");
            }else{
                response.sendRedirect("Error.jsp");
            }
        }
}
```

APPENDIX II

SCREENSHOTS

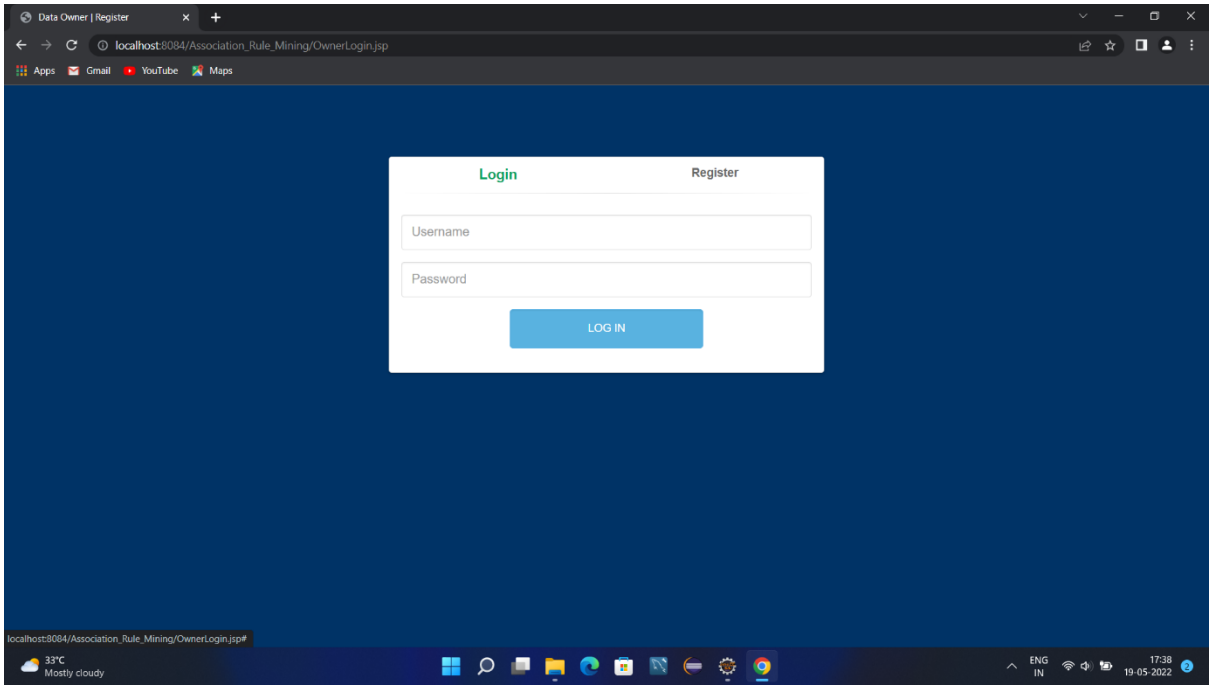
EXECUTION

HOME PAGE



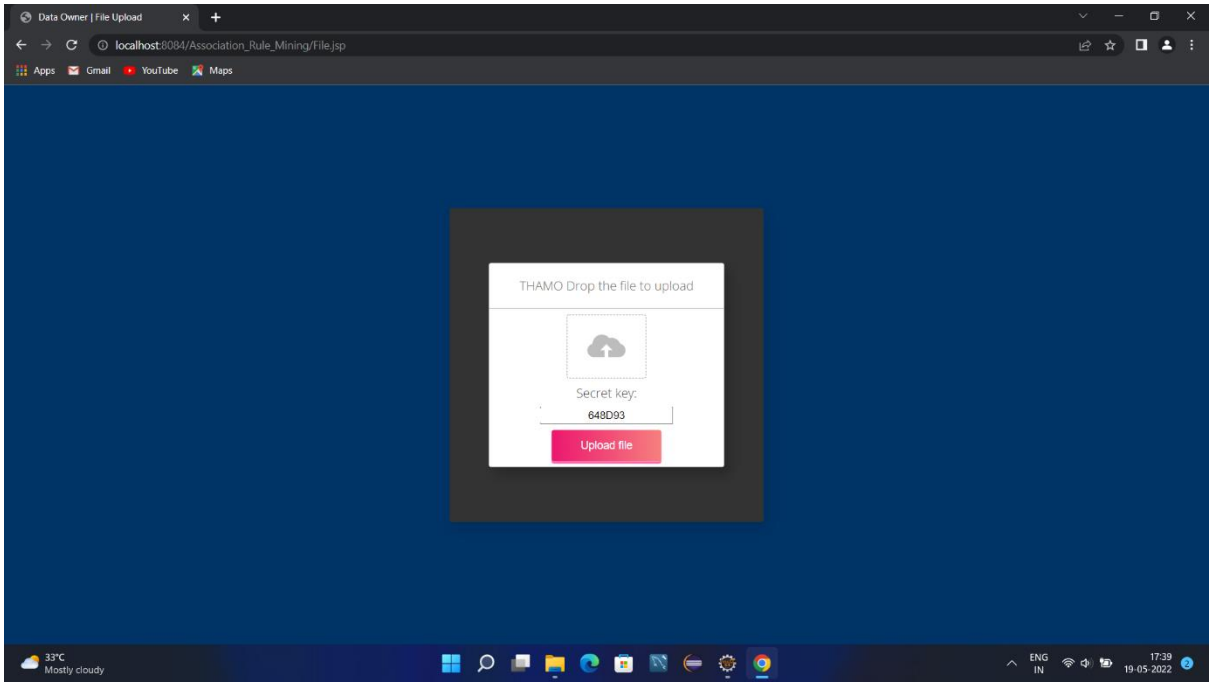
Home Page

USER LOGIN PAGE



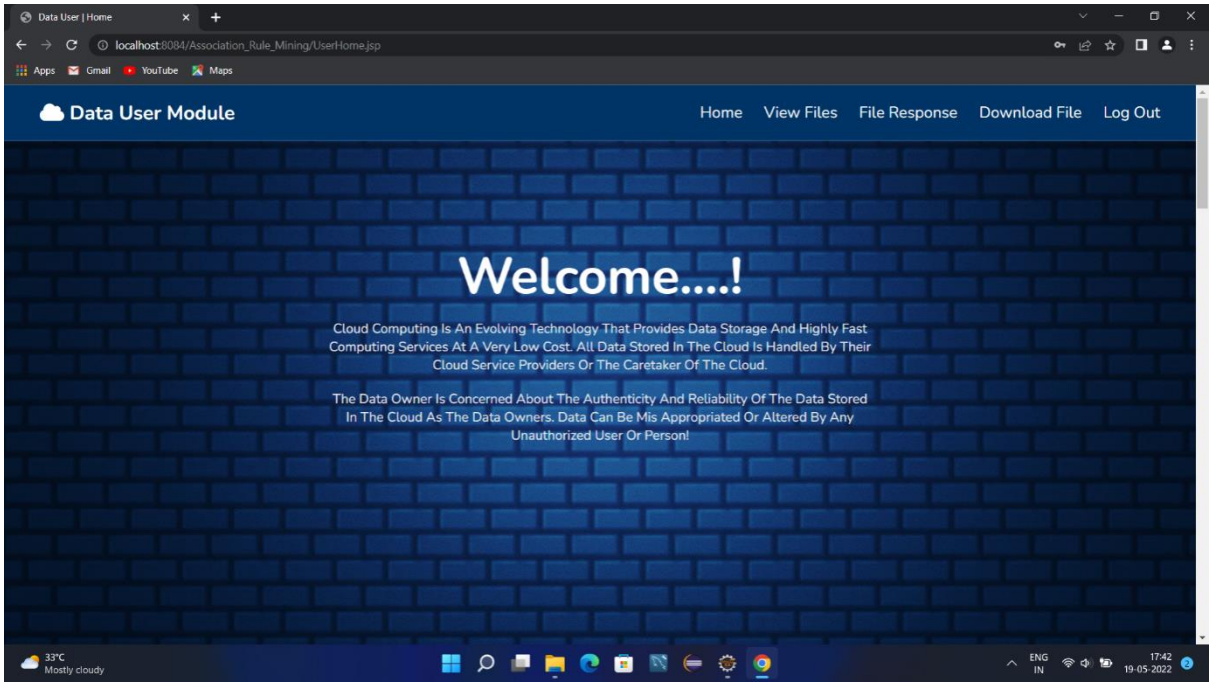
User login page

FILE UPLOADNG



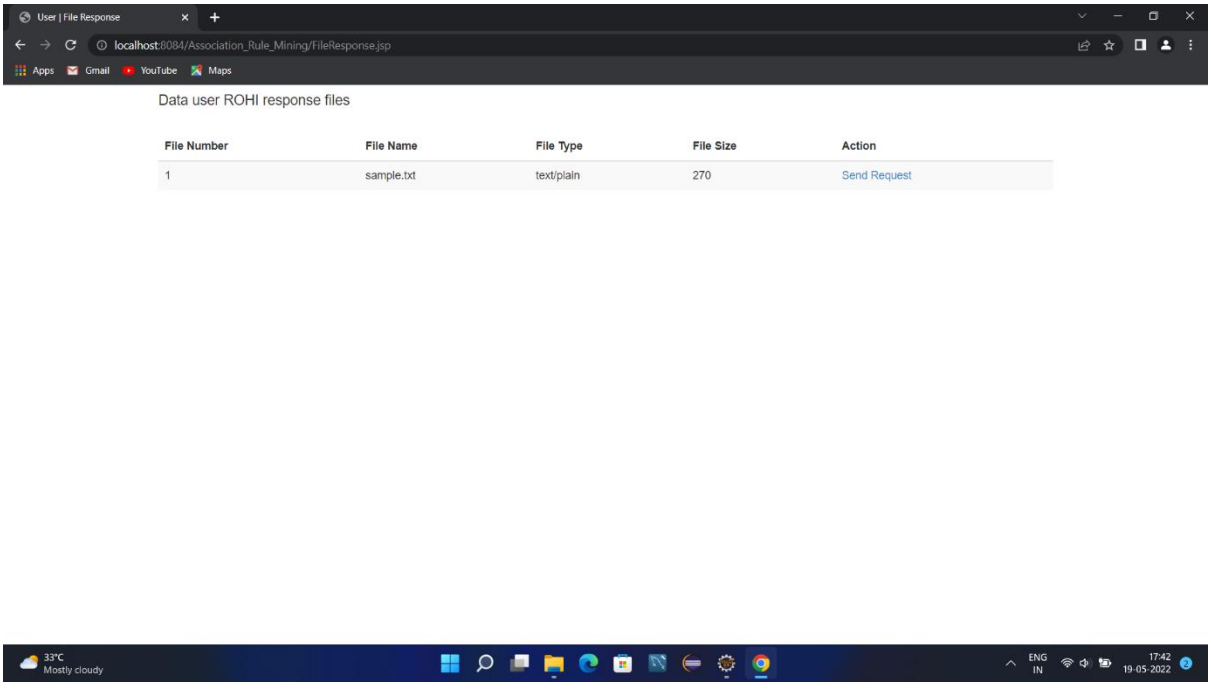
File Uploading by User

DATA USER

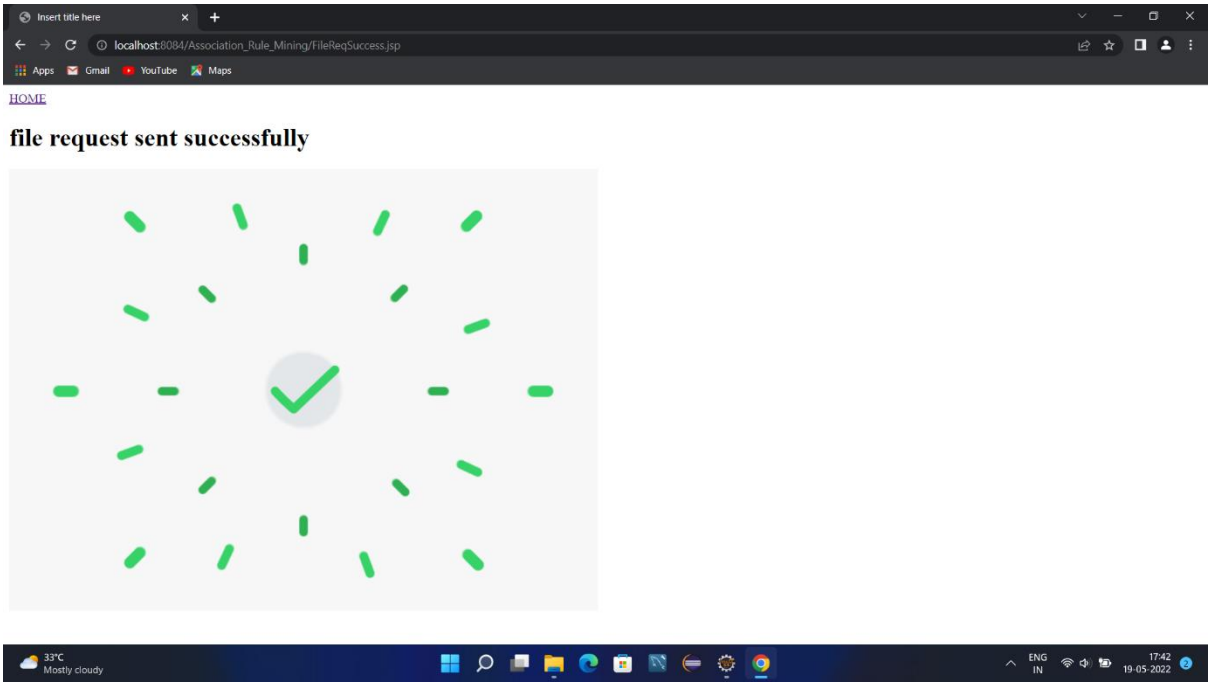


Data user login

REQUESTING FILE

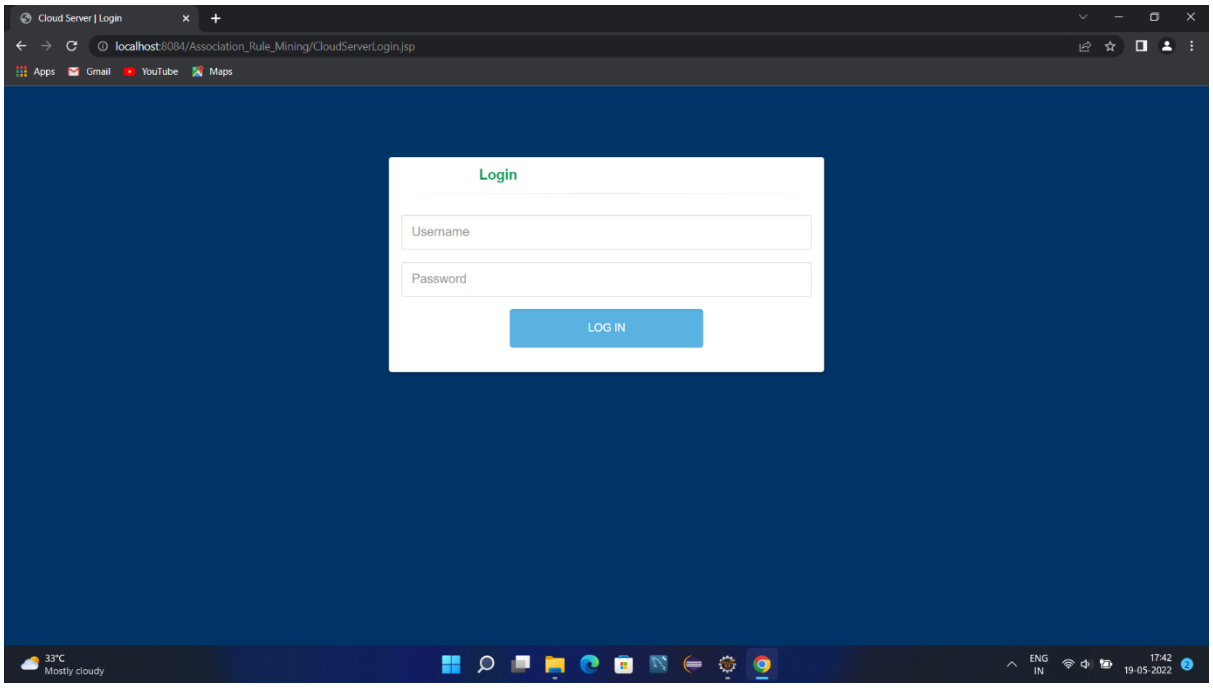


Requesting file from user

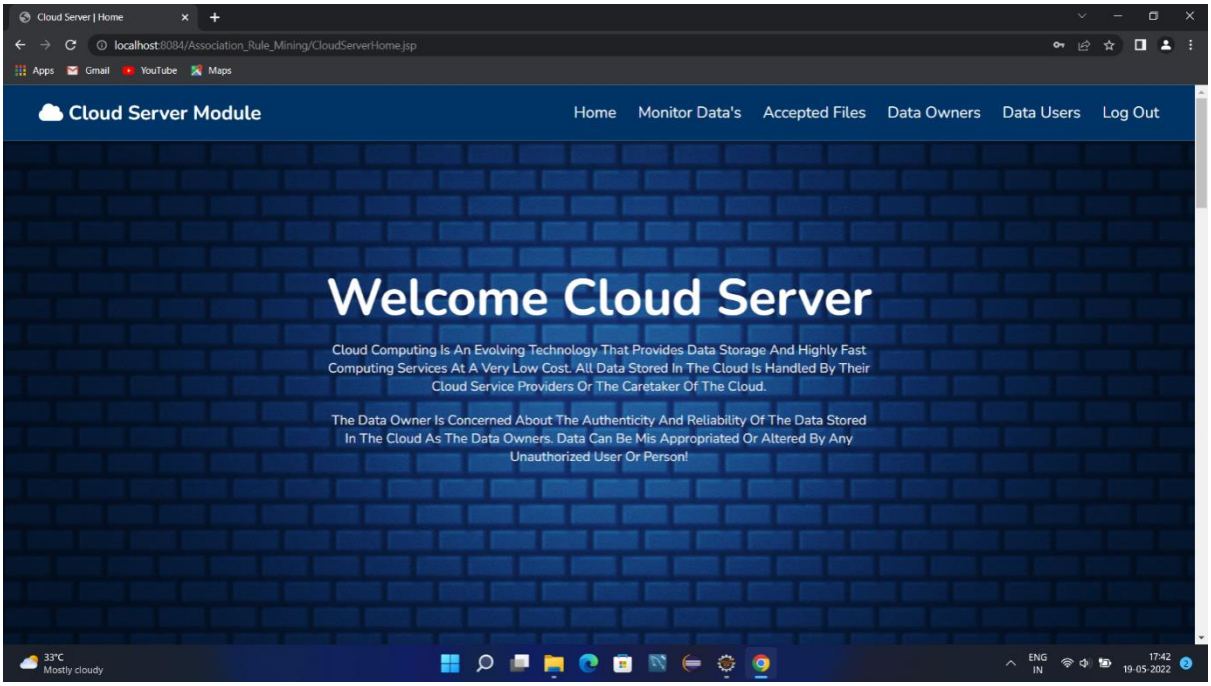


File Request send successfully

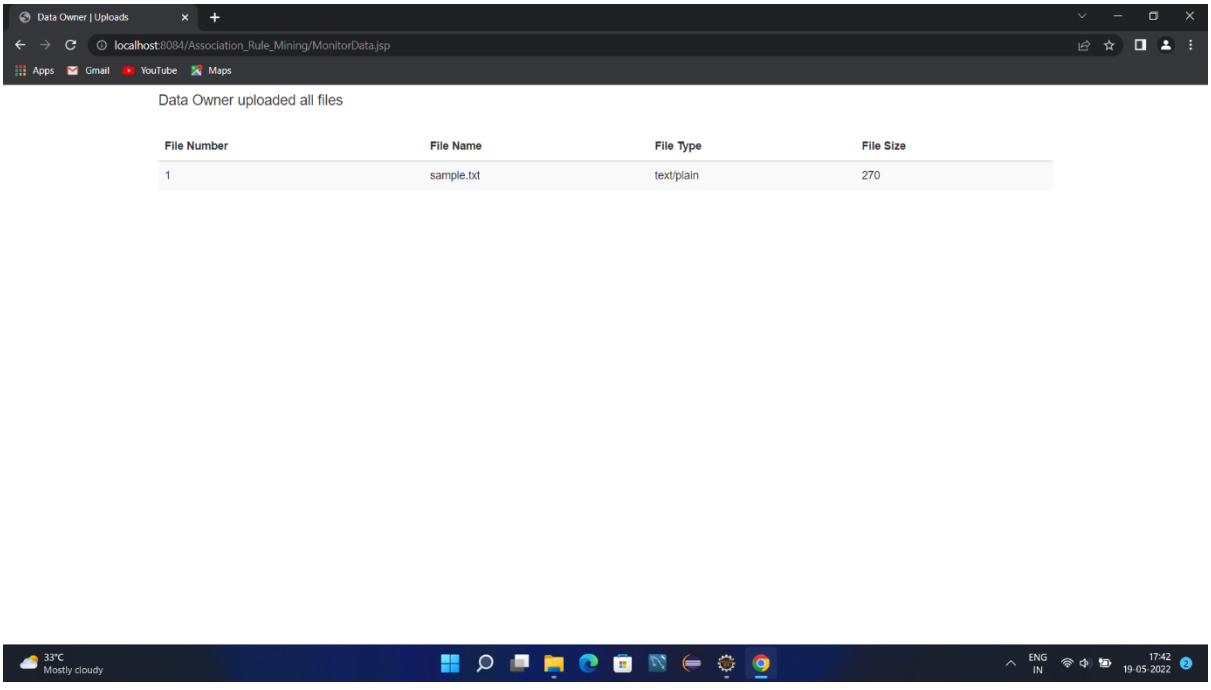
CLOUD SERVER LOGIN



Cloud server login page

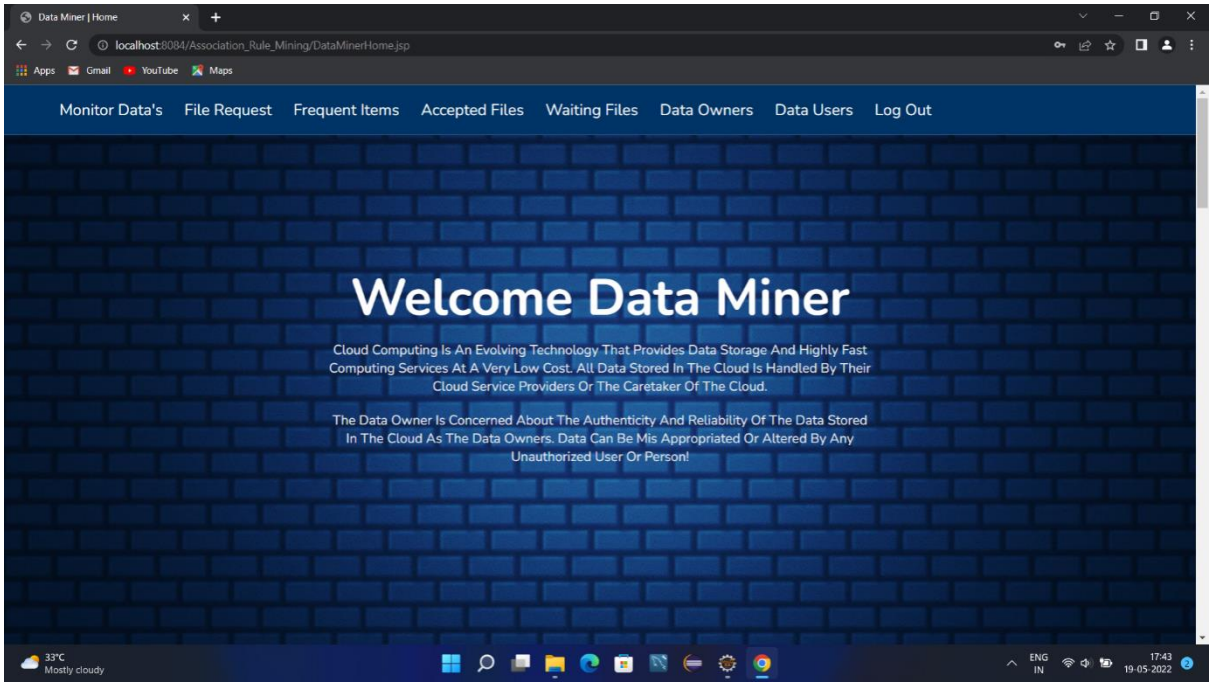


Cloud server home page

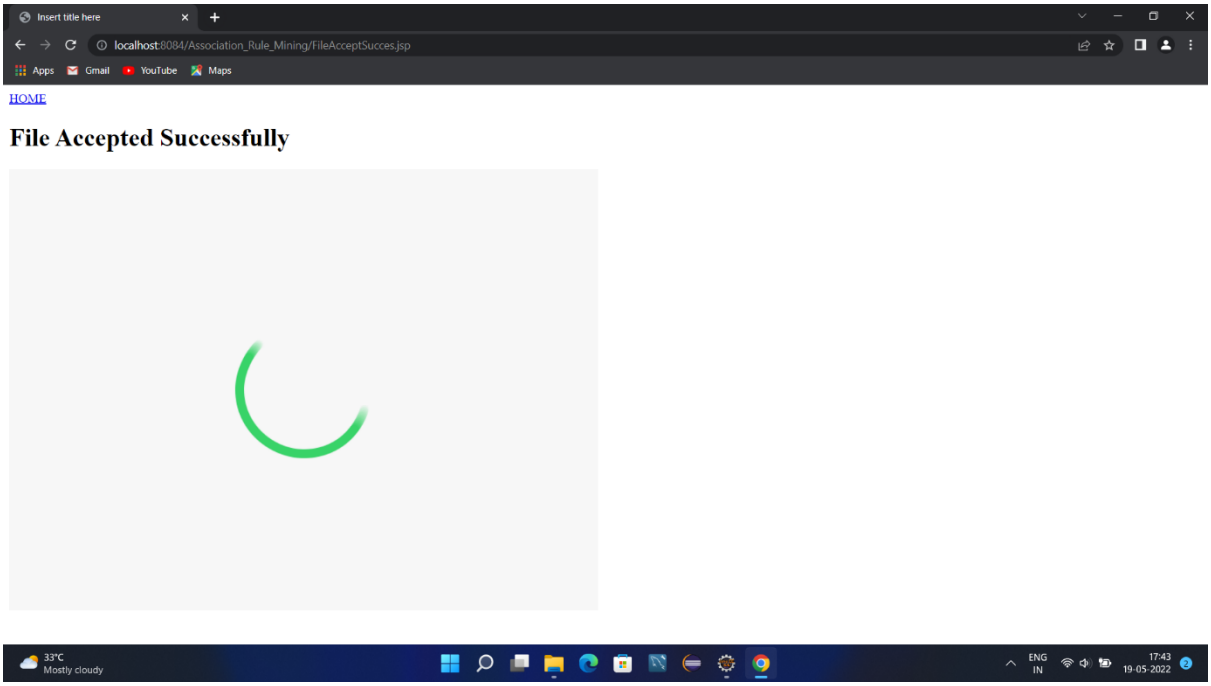


Viewing updated files on cloud

DATA MINER



Data miner home page



File accepting by data miner

REFERENCES

1. Agrawal.R and Srikant.R (2014),” Fast algorithms for mining association rules”, *Proc. 20th Int. Conf. Very Large Data Bases*.
2. Brijis.T, Swinnen.G, Vanhoof.K and Wets.G (2011) “Using association rules for product assortment decisions” A case study, *Proc. 5th ACM SIGKDD Int. Conf. Knowl.*
3. Brossette.S.E et al.(2016),” Association rules and data mining in hospital infection control and public health surveillance”.
4. Bresson.E, Catalano.D and Pointchevel.D (2015),” A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications”.
5. Creighton.C and Hanash.S (2015), “Mining gene expression databases for association rules”.
6. Goldreich.O (2009), Foundations of Cryptography, Cambridge, U.K:Cambridge Univ. Press.
7. Gentry .C, (2018), "Fully homomorphic encryption using ideal lattices"
8. Kim.H-J, Shin.J-H, Song.Y-H and Chang.J-W (2019), “Privacy-preserving association rule mining algorithm for encrypted data in cloud computing”, *Proc. IEEE 12th Int. Conf. Cloud Comput.*, pp. 487-489
9. Lai.J, Li.Y, Deng.R.H, Weng.J, Guan.C and Yan.Q, (2013)” Towards semantically secure outsourcing of association rule mining on categorical data”.
10. Li.L, Lu.R,K-K,Choo.R, Datta.A, and Shao.J(2018),” Privacy-preserving-outsourced association rule mining on vertically partitioned databases”.
11. Mobasher.B ,Jain.N Han E-H and Srivastava.J (1996)“Web mining: Pattern discovery from world wide web transactions”.

12. Peter.ATews.E and Katzenbeisser.S," Efficiently outsourcing multiparty computation under multiple keys", *IEEE Trans. Inf. Forensics Secur.*

13. Rivest.R.L ,Adleman.L and Dertouzos.M.L(2014), "On data banks and privacy homomorphisms", *Found. Secure Comput.*

14. Vaidya.J and Clifton.C(2017) ," Privacy preserving association rule mining in vertically partitioned data", *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*.

15. Wang.B, Zhan.Y and Zhang.Z(2019), "Cryptanalysis of a symmetric fully homomorphic encryption scheme:", *IEEE Trans. Inf. Forensics Secure.*