

DETECTION OF CYBERATTACKS USING ENHANCED SECURE ALGORITHMS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering in Computer Science Engineering

By

Rapole Akhil (37110623)

Pendela Bala Venkata Mahesh Babu (37110560)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600 119

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Rapole Akhil (37110623)** and **Pendela Bala Venkata Mahesh Babu (37110560)** who carried out the project entitled **"DETECTION OF CYBERATTACKS USING ENHANCED SECURE ALGORITHMS IN MACHINE LEARNING"** under my supervision from **August 2020 to March 2021**.

Internal Guide

Mrs. B. Keerthi Samhitha, M.Tech.,(Ph.D).,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Rapole Akhil (37110623)** hereby declare that the Project Report entitled “**DETECTION OF CYBERATTACKS USING ENHANCED SECURE ALGORITHMS IN MACHINE LEARNING**” is done by me under the guidance of **MRS. B. KEERTHI SAMHITHA, M.Tech.,(Ph.D)**., Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **MRS. B. KEERTHI SAMHITHA, M.Tech., (Ph.D.)**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Today information sharing and keeping up its security is significant test. Customer in the data sharing system move their record with the encryption using private key. This property is particularly critical to any huge scope information sharing framework, as any client release the key data then it will get hard for the information proprietor to keep up security of the data. In this paper give a strong and capable dispatch of plan, exhibit its security and give a utilization to show its sound judgment. There are bunches of difficulties for information proprietor to share their information on workers or cloud. There are various answers for take care of these issues. These procedures are a lot of basic to deal with key shared by the information proprietor. This paper will acquaint the confided in power with confirm client the individuals who have the admittance to the information on cloud. SHA calculation is utilized by the believed position to produce the key and that key will get offer to client just as the proprietor. The believed authority module gets encoded record utilizing AES Algorithm from the information proprietor and processes hash esteem utilizing MD-5 calculation. It stores key in its information base which will be utilized during the unique tasks and to decide the tricking party in the framework. Believed authority send document to CSP module to store on cloud. The subsequent key sets are appeared to have various alluring properties that guarantee the privacy of correspondence meetings against intrigue assaults by other organization hubs.

TABLE OF CONTENTS

ABSTRACT	V
LIST OF FIGURES	IX
LIST OF ABBREVIATIONS	XI

CHAPTER NO	TITLE	PAGE NO
1.	INTRODUCTION	1
	1.1 CLOUD COMPUTING	1
	1.2 CONTRIBUTIONS	1
	1.3 AIM AND OBJECTIVE	2
	1.4 CHARACTERISTICS	2
2.	LANGUAGE OVERVIEW	3
	2.1 INTRODUCTION TO JAVA	3
	2.1.1 COLLECTION FRAMEWORK	5
	2.2 MY SQL	6
	2.2.1 SQL SERVER MANAGEMENT STUDIO	7
	2.2.2 CREATE A NEW DATABASE	8
	2.3 NET BEANS	10
	2.4 JAVA NETWORKING	10

3.	LITERATURE SURVEY	11
	3.1 ANALYSIS OF THE LITERATURE	11
	3.2 LITERARY REVIEWS	12
4.	PROBLEM AND REQUIREMENT ANALYSIS	14
	4.1 EXISTING SYSTEM	14
	4.2 PROPOSED SYSTEM	14
	4.3 REQUIREMENT ANALYSIS	15
	4.3.1 FUNCTIONAL REQUIREMENTS	15
	4.3.2 NON-FUNCTIONAL REQUIREMENTS	16
5.	SOFTWARE DEVELOPMENT METHODOLOGY	18
	5.1 ARCHITECTURAL DESIGN	18
	5.2 MODULE DESCRIPTION	19
	5.3 DATA FLOW DIAGRAMS	22
	5.4 UML DIAGRAMS	24
	5.4.1 USE CASE DIAGRAM	24
	5.4.2 CLASS DIAGRAM	25
	5.4.3 SEQUENCE DIAGRAM	26
	5.4.4 ACTIVITY DIAGRAM	27
	5.4.5 COMPONENT DIAGRAM	28
	5.5 REQUIREMENT SPECIFICATION	28
	5.5.1 HARDWARE REQUIREMENTS	28
	5.5.2 SOFTWARE REQUIREMENTS	29
	5.6 PERFORMANCE ANALYSIS	29
	5.7 SYSTEM DESIGN	30
	5.7.1 INPUT DESIGN	30
	5.7.2 OUTPUT DESIGN	31
	5.8 SYSTEM TESTING	32
	5.8.1 TESTING PROCESS	32

	5.8.2 TESTING STRATEGY AND APPROACH	35
6.	RESULTS AND DISCUSSIONS	36
7.	CONCLUSION AND FUTURE WORK	37
	REFERENCES	38
	APPENDIX	
	A. SAMPLE CODE	39
	B. SCREENSHOTS	63
	C. PUBLICATION WITH PLAGIARISM REPORT	69

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
2.1	Java compilation	4
2.2	Collection framework	5
2.3	Working Database Engine	7
2.4	SQL server management studio	8
2.5	Creating a new DB	8
4.1	Proposed system for Cyber-attacks detection using Bayes Classifier	15
5.1	Architecture Diagram	18
5.2	Login module	19
5.3	Registration module	20
5.4	Creating storage and instance	20
5.5	Find collusion module	21
5.6	Find Third-Party module	21
5.7	DFD–Level 0:Data Owner	22
5.8	DFD–Level 1	22
5.9	DFD–Level 2	23
5.10	Use case diagram	24
5.11	Class diagram	25

5.12	Sequence diagram	26
5.13	Activity diagram	27
5.14	Component diagram	28
5.15	Computational outsourcing	28
5.16	Unit testing	33
5.17	White Box testing	34
5.18	Black Box testing	34

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
JRE	Java Runtime Environment
OS	Operating System
J2SE	Java Standard Edition
AES	Advanced Encryption Standard
CSP	Content Security Provider
MD-5	Message Digest Algorithm 5
SHA	Secure Hash Algorithm
E-SC	Electronic Stability Control
SQL	Structured Query Language
IP	Internet Protocol
UML	Unified Modeling Language
OLAP	Online Analytical Processing
TCP	Transfer Control Processing
UDP	User Datagram Protocol
NFR	Non-Functional Requirements

DFD

Data Flow Diagram

IDE

Integrated Development Environment

CHAPTER 1

INTRODUCTION

1.1 CLOUD CLOMPUTING

One possible solution is to migrate character sequences to public cloud computing platforms and to request that Cloud Service Provider process sequence comparisons. At present, primary sequence comparison algorithms are deployed as a universal outsourcing service on public clouds. But at the same time, its security and privacy issues are increasingly emerging. The outsourced data stored as plaintext could easily be exposed to malicious external intruders and internal attackers in the CSP, and the individual private information carried by character sequences (e.g., personal identification, financial transaction records, genetic markers for some diseases, information that is used to identify paternity or maternity, etc.) could more or less be disclosed or abused. Therefore, secure outsourcing is designed to protect the privacy of character sequences, and to ensure that the scheduled computing requests are normally performed on the cloud servers.

1.2 CONTRIBUTIONS

Our scheme is easy in deployment, efficient in processing and controllable in overhead. The contributions of this paper mainly in the following four aspects.

Based on the universal model of a public cloud outsourcing, we propose an overall architecture for E-SC. This architecture is built on the end user and the unmodified CSP. Its overall system model, which has been demonstrated to be secure under the threat model, is user-friendly and implementation-friendly.

A salted hash algorithm is improved to hash the character sequences and the indexes of cost matrices, so as to defend against statistical attacks. An additive order preserving encryption algorithm is designed to encrypt the elements of cost

matrices. Also, this algorithm can achieve an indistinguishability under additive ordered chosen-plaintext attack with linear time complexity.

A single cloud server works for the first time to provide a privacy-preserving computable outsourcing service to effectively resist collusion attacks from the cloud. With per-processing modules of padding, partition and expansion, there is no need to decrypt any outsourced data in the non-interactive sequence comparison stage.

Simulation results show that the overall execution performance of our E-SC is negatively correlated with its security.

1.3 AIM AND OBJECTIVE

The main objective of this system is, SHA algorithm is used by the trusted authority to generate the key and that key will get share to user as well as the owner. The trusted authority module receives encrypted file using AES Algorithm from the data owner and computes hash value using MD-5 algorithm.

1.4 CHARACTERISTICS

The main characteristic of a cloud computing include

- High reliability
- More flexibility
- Low cost
- Provide security
- User comfortable

CHAPTER 2

LANGUAGE OVERVIEW

2.1 INTRODUCTION TO JAVA

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has waned with passage of time, while Java's has grown.

Application of Java:

Java is widely used in every corner of world and of human life. Java is not only used in software but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

- Developing Desktop Applications
- Web Applications like LinkedIn.com, Snapdeal.com etc
- Mobile Operating System like Android
- Embedded Systems
- Robotics and games etc.

Features of Java:

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in molding out the final form of this outstanding language. Those features are:-

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

2) Object Oriented

In java everything is Object which has some data and behavior. Java can be easily extended as it is based on Object Model.

3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language. On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.

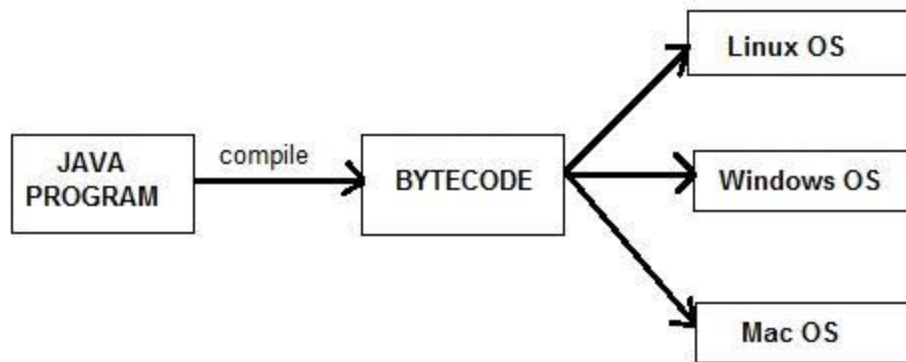


Fig: 2.1 Java Compilation

5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

8) Portable

Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types.

10) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

2.1.1 Collection framework

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.

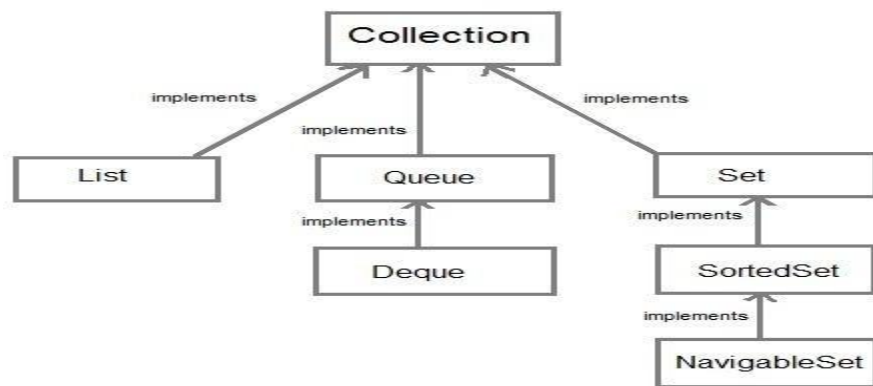


Fig: 2.2 Collection framework

2.2 MY SQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

LAMP is an acronym for "Linux, Apache, MySQL and Perl/PHP/Python." Free software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, Word Press, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr and YouTube.

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, data processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both). SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

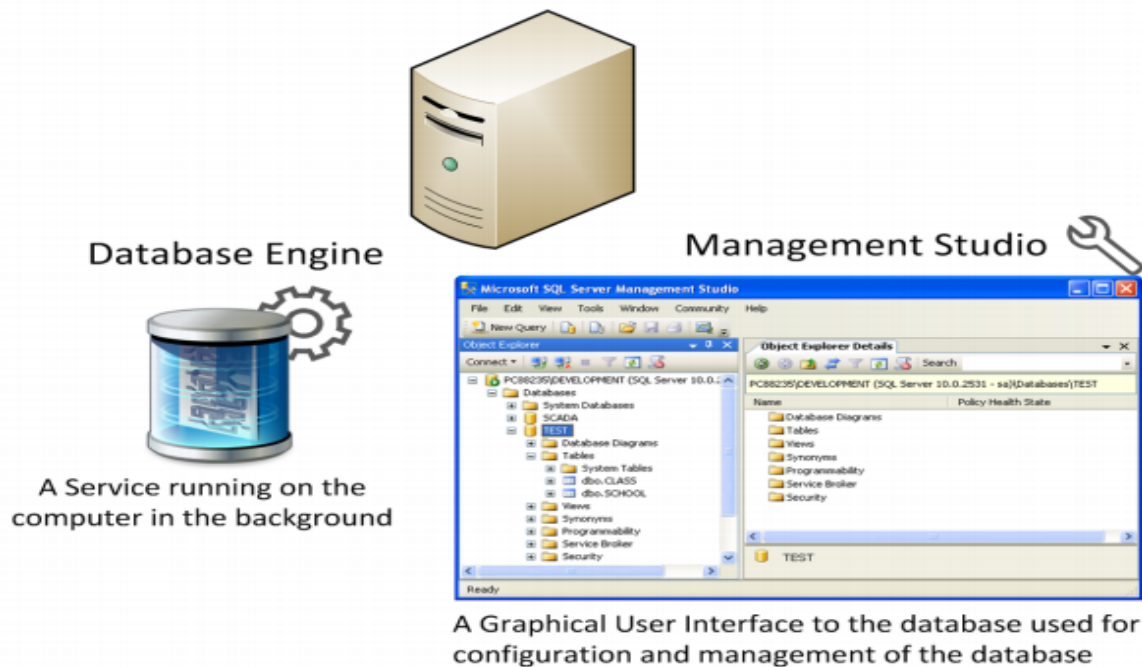


Fig: 2.3 Working Database Engine

2.2.1 SQL Server management studio

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express. A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above). With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

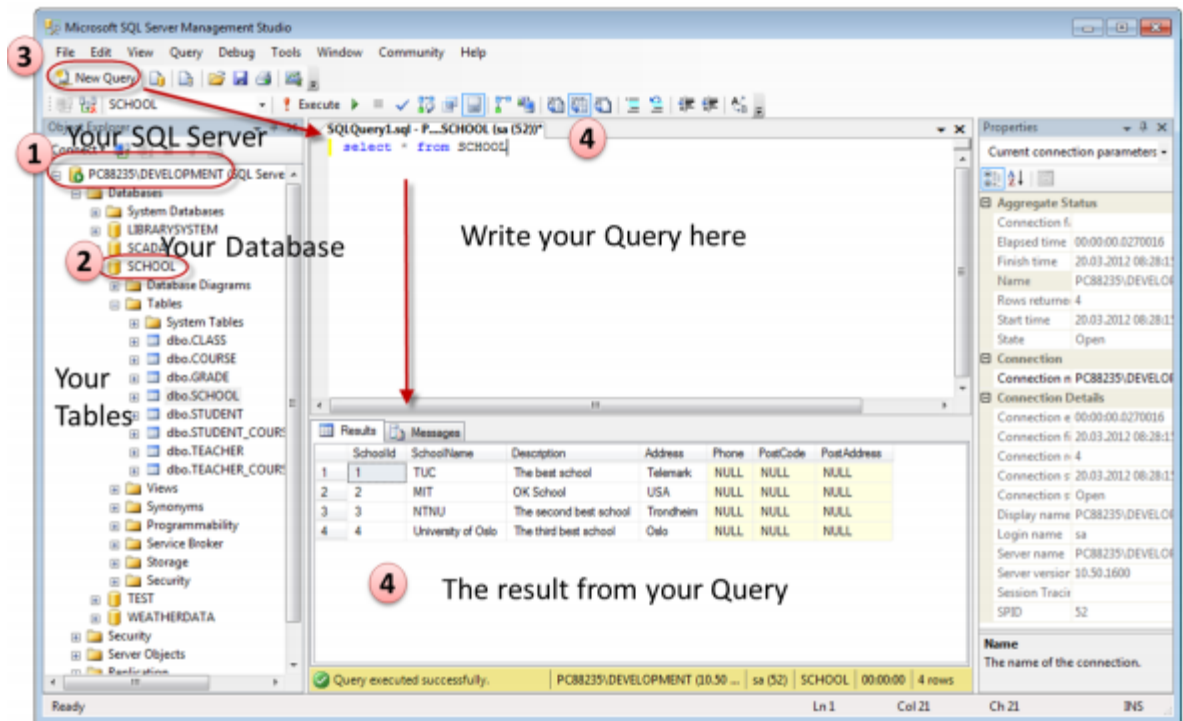


Fig: 2.4 SQL Server management studio

2.2.2 Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the “Databases” node and select “New Database...”

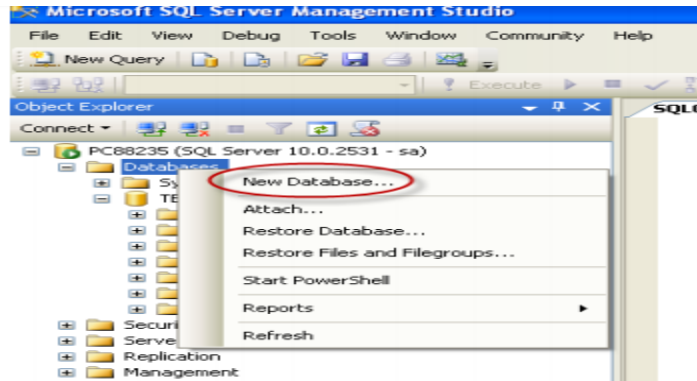


Fig: 2.5 Creating a new DB

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services.

SQL- SERVER database consist of six types of objects:-

They are:

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

Table

A database is a collection of data about a specific topic.

Views of Table

We can work with a table in two types:

1. Design View
2. Datasheet View

Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

Query

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dataset (if you edit it) or a snapshot (it cannot be edited). Each time we run query, we get latest information in the dataset. Access either displays the dataset or snapshot for us to view or perform an action on it, such as deleting or updating.

2.3 NET BEANS

Net Beans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyzers and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, Net Beans IDE sets the standard for developing with cutting edge technologies out of the box.

2.4 JAVA NETWORKING

The term network programming refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network. The `java.net` package provides support for the two common network protocols:-

- **TCP:** TCP stands for Transmission Control Protocol, which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.
- **UDP:** UDP stands for User Datagram Protocol, a connection-less protocol that allows for packets of data to be transmitted between applications.

This tutorial gives good understanding on the following two subjects:

- **Socket Programming:** This is most widely used concept in Networking and it has been explained in very detail.
- **URL Processing:** This would be covered separately. [Click here](#) to learn about URL Processing in Java language.

CHAPTER 3

LITERATURE SURVEY

3.1 ANALYSIS OF THE LITERATURE

Literature survey is the main advance in programming improvement measure. Prior to building up the instrument it is important to decide the time factor, economy and friends strength. When these things are fulfilled, at that point the subsequent stage is to figure out which working framework and language can be utilized for building up the device. When the developers begin assembling the apparatus the software engineers need parcel of outer help. This help can be gotten from senior developers, from book or from sites. The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Prior to building the framework the above

thought are considered for building up the proposed framework. The significant piece of the undertaking advancement area considers and completely survey all the necessary requirements for building up the venture. For each undertaking Literature survey is the main area in programming improvement measure. Prior to building up the instruments and the related planning it is important to decide and survey the time factor, asset prerequisite, labor, economy, and friends strength. When these things are fulfilled and completely surveyed, at that point the following stage is to decide about the product details in the separate framework, for example, what kind of working framework the venture would require and what are largely the important programming are expected to continue with the subsequent stage like building up the apparatuses, and the related activities. Here we have taken the general surveys of different creators and noted down the fundamental central issues with respect to their work. In this venture literature survey assumes a prevailing part in get assets from different areas and all the connected points that are exceptionally valuable under this segment. The most awesome aspect if this is the manner in which things get all together and encourages us to suite our work according to the current information.

3.2 LITERARY REVIEWS

“Secure outsourcing of sequence comparisons”, M.J. Atallah and J.Li (2014)

With the coming of distributed computing, secure rethinking strategies of grouping correlations are getting progressively important, particularly for customers with restricted assets. Quite possibly the most basic functionalities in information reevaluating is obviousness. Specifically, our development re-distorts the circuit just for twisted reactions and subsequently is effective. Plus, we likewise present the conventional examination for our proposed development. Constraint of the procedures in that they don't stretch out to the current circumstance where the strings are of various length additions and erasures are essential for the definition.

“Secure and private sequence comparisons”, M.J Atallah, F.Kerschbaum, and W.Du (2014)

Here we are utilizing Wagner-Fischer procedure. The consequences of the examinations. One approach to dodge this issue is to do the accompanying prior to playing out the previously mentioned credulous least discovering convention. A Limitation of the strategy in that they don't reach out to the current circumstance where the strings are of various length and hence additions and cancellations must be permitted.

“Towards a practical data privacy scheme for a distributed implementation of the Smith-Waterman genome sequence comparison algorithm”, D.Szajda, M.Pohl, J.Owen, and B.Lawson (2016)

Smith-Waterman Sequence Comparison. A careful treatment of arrangement methods would fill a few writings. By and by there is adaptability in these necessities. A few applications may endure a couple of missed significant outcomes are produced. Others may acknowledge some adaptability on the quantity of bogus positives, given that no significant outcomes are missed. This methodology, where correspondence is needed among members and every member creates huge organization traffic, isn't as of now down to earth because of stage limits.

“New algorithms for secure outsourcing of modular exponentiations”, X.Chen, J.Li, J.Ma, Q.Tang and W.Lou (2017)

The arrangement utilized in the camouflage method and in this way permitted spillage of private data. We propose the first rethink secure and effective calculation for synchronous secluded exponentiations. Besides, we demonstrate that both the calculations can accomplish the ideal security thoughts. Very tedious, can't recognize cancellations, movements or duplicate number changes. Versatility requirements limit applications to problem areas.

“Efficient and verifiable outsourcing scheme of sequence comparisons”, Y.Feng, H.Ma, and X.Chen (2019)

With the quick improvement of distributed computing, the strategies for safely re-appropriating restrictively costly calculations are getting broad considerations in mainstream researchers. Here the proposed arrangement empowers customers to effectively identify the misconduct of unscrupulous workers. Moreover, our development re-distorts the circuit just for deformed reactions and in this way is exceptionally productive for genuine applications. Plus, we likewise present the proper examination for our proposed development. Exceptionally dependent on client contribution for right recognizable proof oxPTMs, in any case bogus positives and negative happen.

CHAPTER 4

PROBLEM AND REQUIREMENT ANALYSIS

4.1 EXISTING SYSTEM

Large-scale problems in the physical and life sciences are being revolutionized by Internet computing technologies, like grid computing, that make possible the massive cooperative sharing of computational power, bandwidth, storage, and data. A weak computational device, once connected to such a grid, is no longer limited by its slow

speed, small amounts of local storage, and limited bandwidth: It can avail itself of the abundance of these resources that is available elsewhere on the network. Without revealing to the remote agents whose computational power is being used, either one's data or the outcome of the computation on the data.

Disadvantages

- Secure outsourcing for widely applicable sequence comparison problems.
- Risk of Leak of Secret Information.

4.2 PROPOSED SYSTEM

We propose a secure data sharing scheme, which can achieve secure key distribution and data sharing for dynamic group. We provide a secure way for key distribution without any secure communication channels. The users can securely obtain their private keys from group manager without any Certificate Authorities due to the verification for the public key of the user. Our scheme can achieve fine-grained access control, with the help of the group user list, any user in the group can use the source in the cloud and revoked users cannot access the cloud again after they are revoked. We propose a secure data sharing scheme which can be protected from collusion attack. The revoked users can not be able to get the original data files once they are revoked even if they conspire with the untrusted cloud.

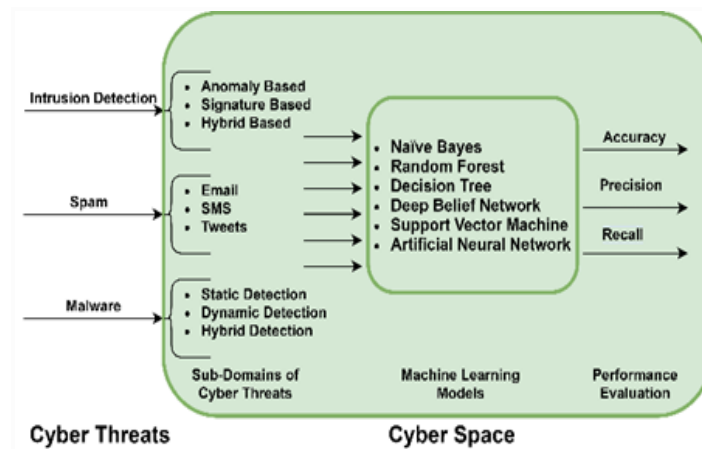


Fig: 4.1 Proposed system for Cyber Attacks detection using Bayes Classifier

Advantages

- Power Means of Persuasion and control
- More Reliable
- It's more secure and efficient.
- Data confidentiality

4.3 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.3.1 Functional Requirements

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

i. Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

ii. Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

iii. Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

iv. Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and even capable of working around any failure.

v. Compatibility

It is supported by version above all web browsers. Using any web servers like local host makes the system real-time experience.

vi. Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

vii. Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

4.3.2 Non-Functional Requirements

Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

i. Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

ii. Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

iii. Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

iv. Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 5

SOFTWARE DEVELOPMENT METHODOLOGY

5.1 ARCHITECTURAL DESIGN

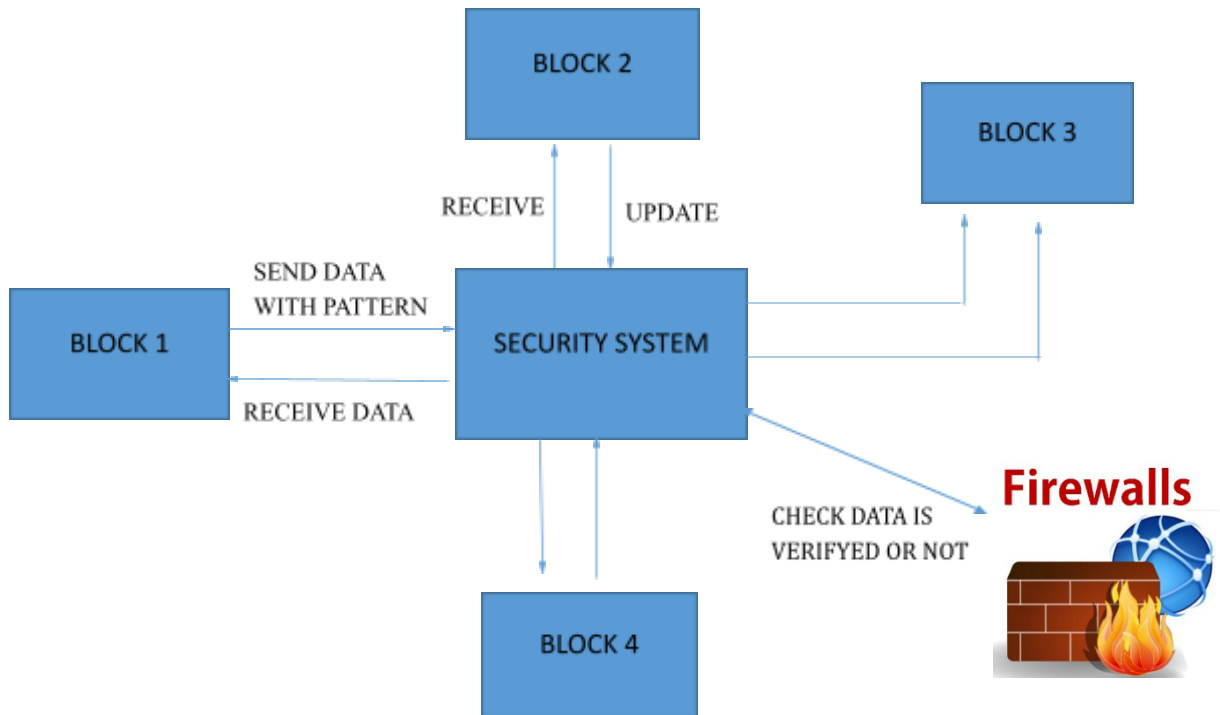


Fig: 5.1 Architecture diagram

The above figure 5.1 shows the overall architecture of our system. We consider a cloud system composed of three major entities that are cloud server, data owner and the multiple users. Block 1 send the data to security system and then received the data. Security system provide security using firewall. Firewall is used to check data is verified or not. The data owner has not control over the data after it is uploaded on cloud. In this module, the original data get encrypted into two different values.

5.2 MODULE DESCRIPTION

There are Used five Different Modules

- Login Module
- Registration Module
- Creation Storage and Instance
- Find collusion Module
- Find Third-Party Module

i. Login Module:

This is the first activity, User needs to provide a correct contact number and a password, which user enters while registering, in order to login into the app. If information provided by the user matches with the data in the database table then user successfully login into the app else message of login failed is displayed and user need to re-enter correct information. A link to the register activity is also provided for registration of new users.



Fig: 5.2 Login Module

INPUT: User Name and Password

OUTPUT: Admin Login

ii. Registration Module:

A new user who wants to access the app needs to register first before login. By clicking on register button in login activity, the register activity gets open. A new user registers by entering full name, password and contact number. A user needs to enter password again in confirm password textbox for confirmation. When user enters the information in all textboxes, on the click of register button, the data is transferred to database and user is directed to login activity again.

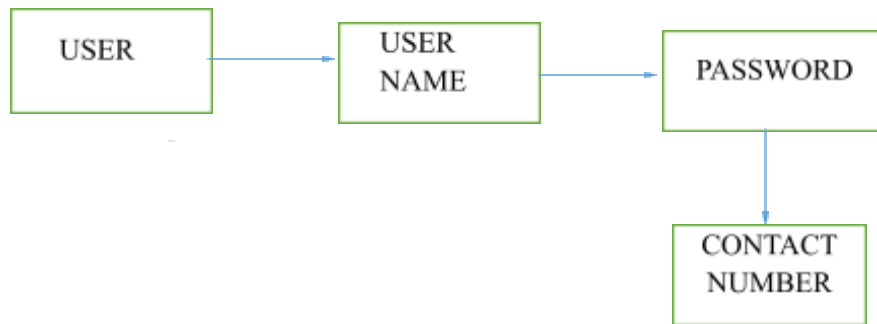


Fig: 5.3 Registration Module

INPUT : User Name and Password

OUTPUT: Database

iii. Creation Storage and Instance:

The data owner has not control over the data after it is uploaded on cloud. In this module, the original data get encrypted into two different values. The data in each slice can be encrypted by using different cryptographic algorithm's and encryption key before storing them in the Cloud.

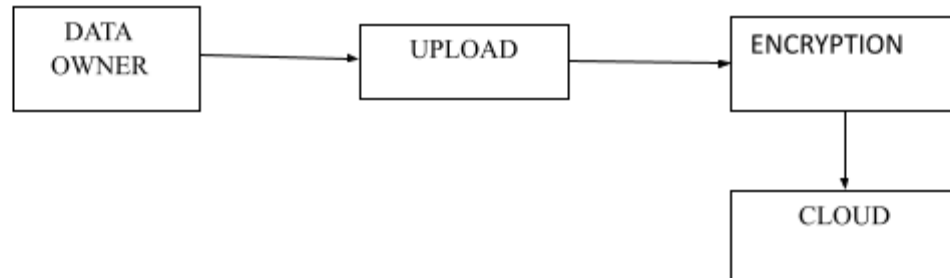


Fig: 5.4 Creation Storage and Instance

INPUT : User Name and Password

OUTPUT: data uploaded

iv. Find collusion Module:

In this Module, Receiver can find collusion occurring or not using calculating a distance.

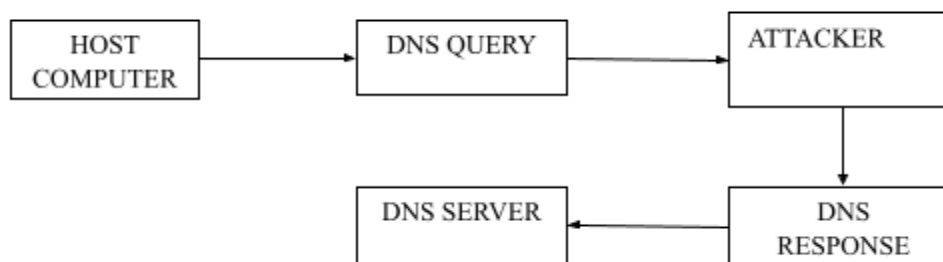


Fig: 5.5 Find Collusion Module

INPUT : User Name and Password

OUTPUT: Database

v. Find Third Party Module:

In this Module, receiver can also find third-parties. Third party refers to another company making software for the original vendor's product.

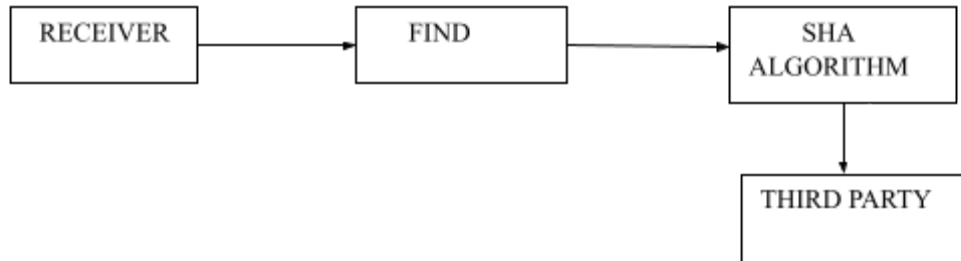


Fig: 5.6 Find Third-Party Module

INPUT : User Name and Password

OUTPUT: find third-parties

5.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFD's can also be used for the visualization of data processing (structured design).

i. DFD - Level 0: Data owner:

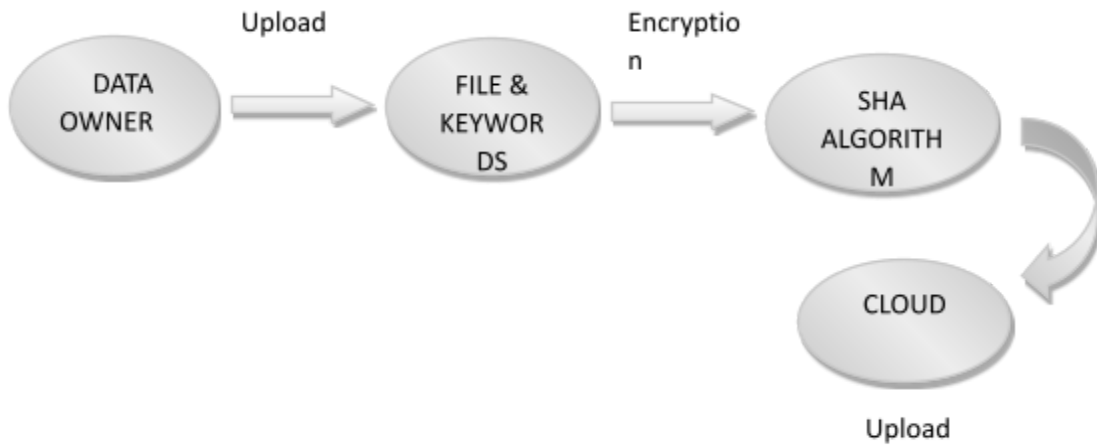


Fig: 5.7 DFD-Level 0: Data Owner

ii. DFD - Level 1:

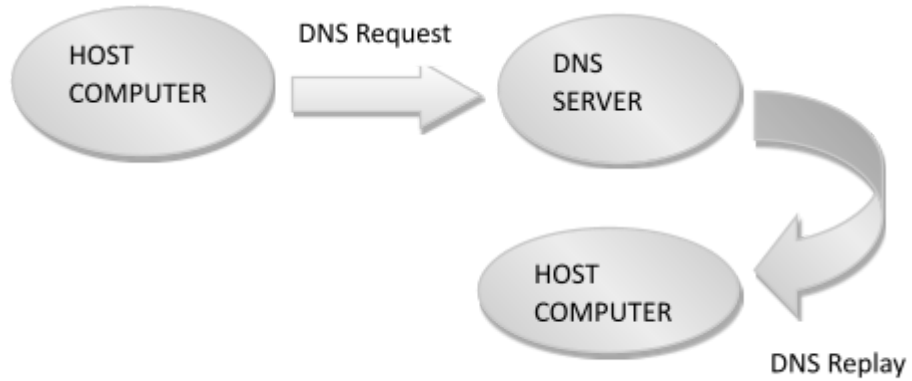
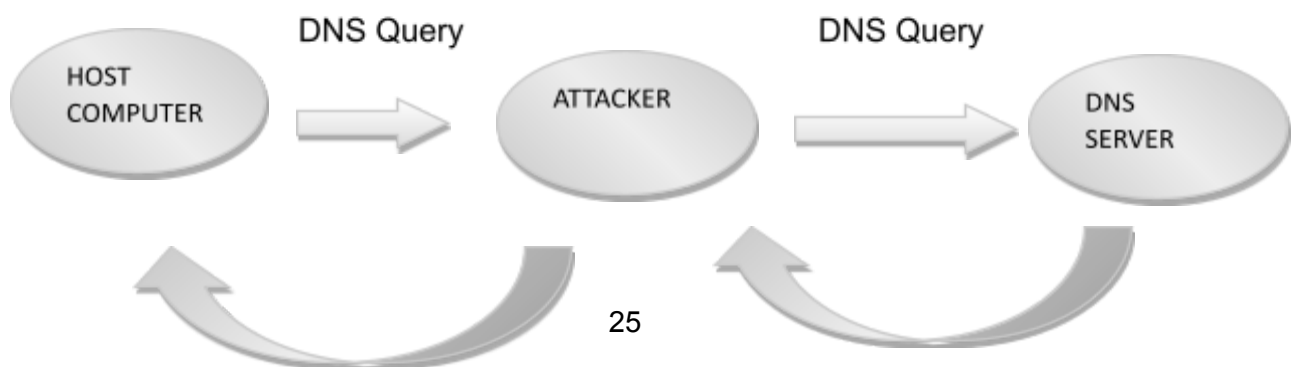


Fig 5.8 DFD-Level 1

iii. DFD-Level 2:



DNS Response

DNS Response

Fig: 5.9 DFD-Level 2

5.4 UML DIAGRAMS

UML is a method for describing the system architecture in detail using the blueprint. UML represents a collection of best engineering practices that have proven successful in the

modeling of large and complex systems. UML is a very important part of developing objects oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects. Using the UML Helps project teams communicate, explore potential designs, and validate the architectural design of the software.

5.4.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

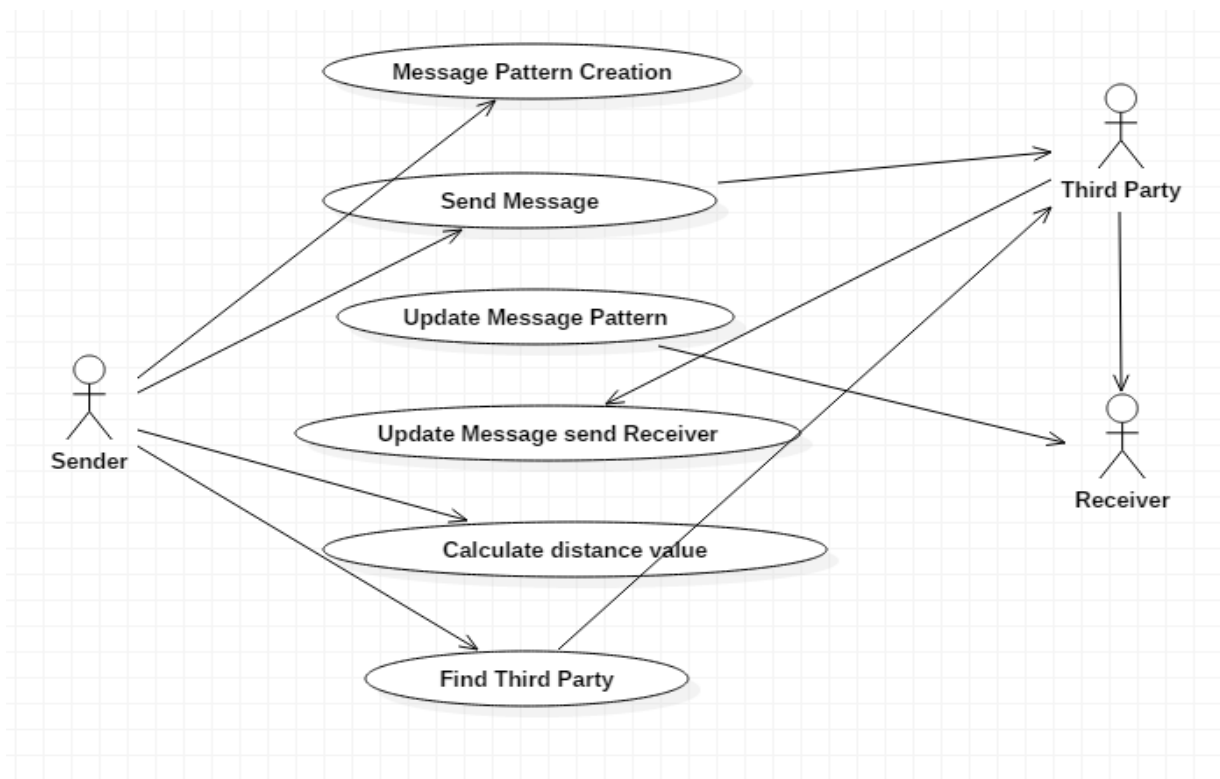


Fig: 5.10 Use Case Diagram

5.4.2 Class Diagram

A class is the main building block of object oriented modeling. It is used for general conceptual modeling of the systematic of the application and for detailed modeling translated the models into the program code.

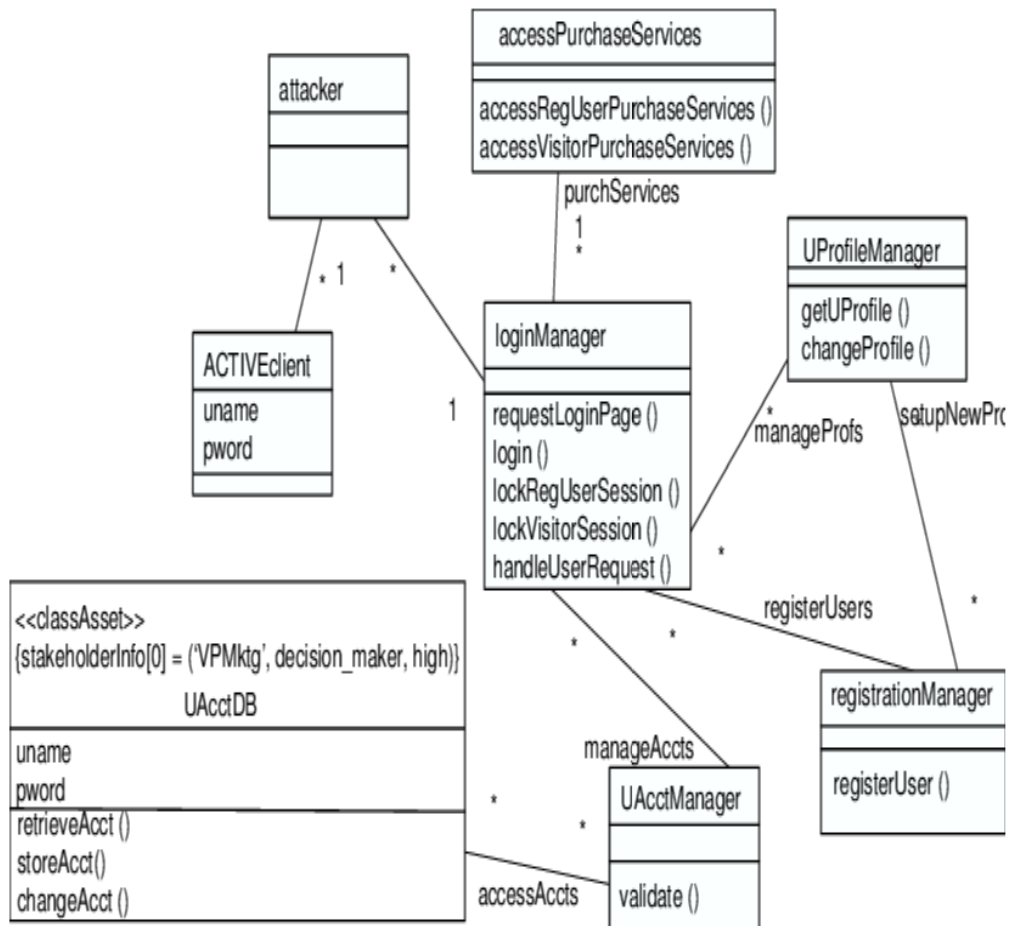


Fig: 5.11 Class Diagram

5.4.3 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message Sequence Chart. Sequence diagram are sometime called event diagrams, and timing diagram.

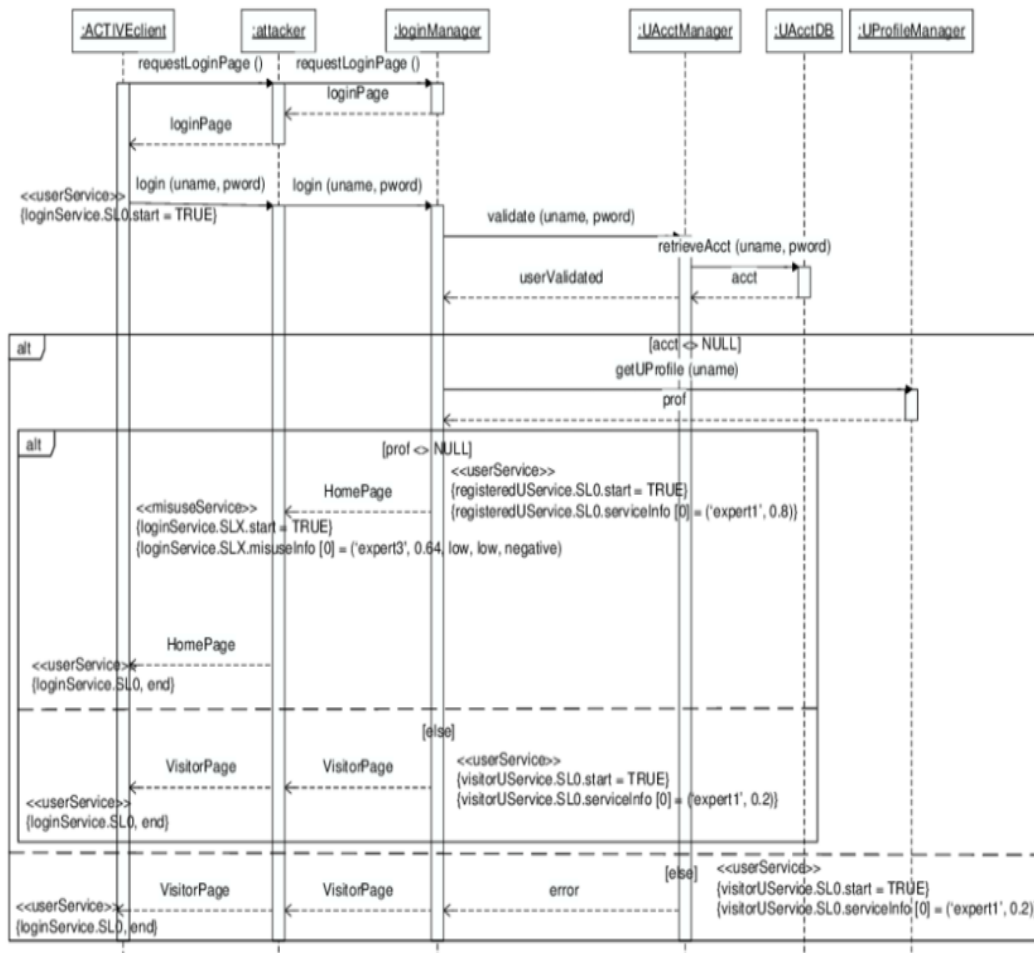


Fig: 5.12 Sequence Diagram

5.4.5 Activity Diagram

Activity diagrams are graphical representations of workflow of stepwise activity and action with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagram can be used to describe the business and operational step by step workflow of components in a system. An activity diagram shows the overflow of control.

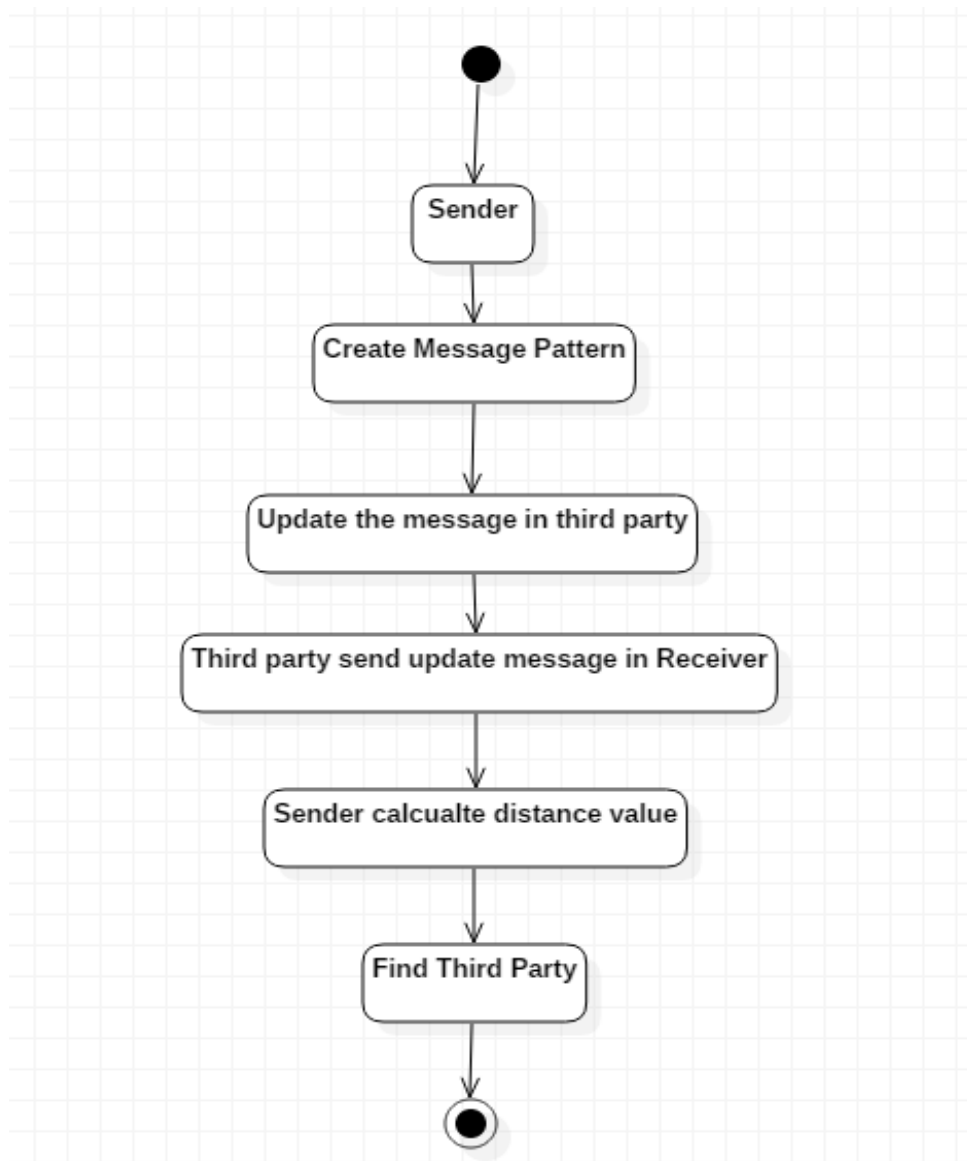


Fig: 5.13 Activity Diagram

5.4.6 Component Diagram

A component diagram is user to visualize organization and relationship among them. The systems are more component useful while making the executable system. The user, master user and Third party auditor are executable parts of the system.

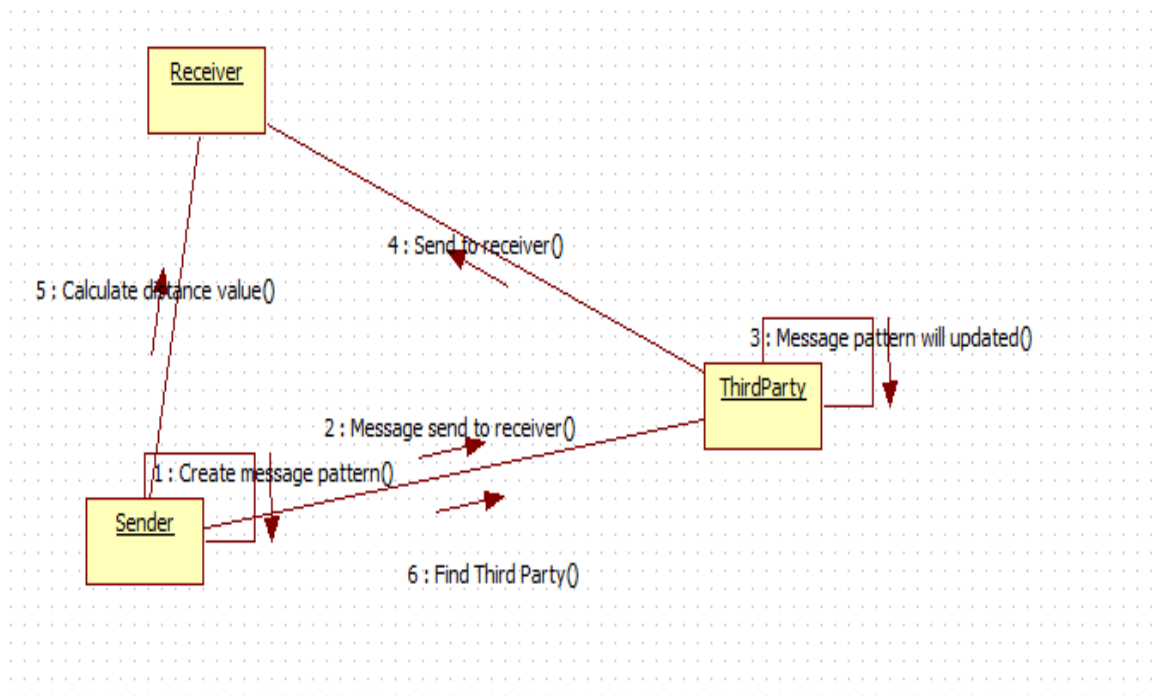


Fig: 5.14 Component Diagram

5.5 REQUIREMENT SPECIFICATION

5.5.1 Hardware Requirements:

- System – Pentium IV
- Hard Disk – 40GB
- Speed – 2.4 GHZ
- Monitor – 15 VGA color
- RAM – 512 GB

5.5.2 Software Requirements:

- Operating system – Windows XP
- Coding language – JAVA
- IDE – Net Beans
- Database – MYSQL

5.6 PERFORMANCE ANALYSIS

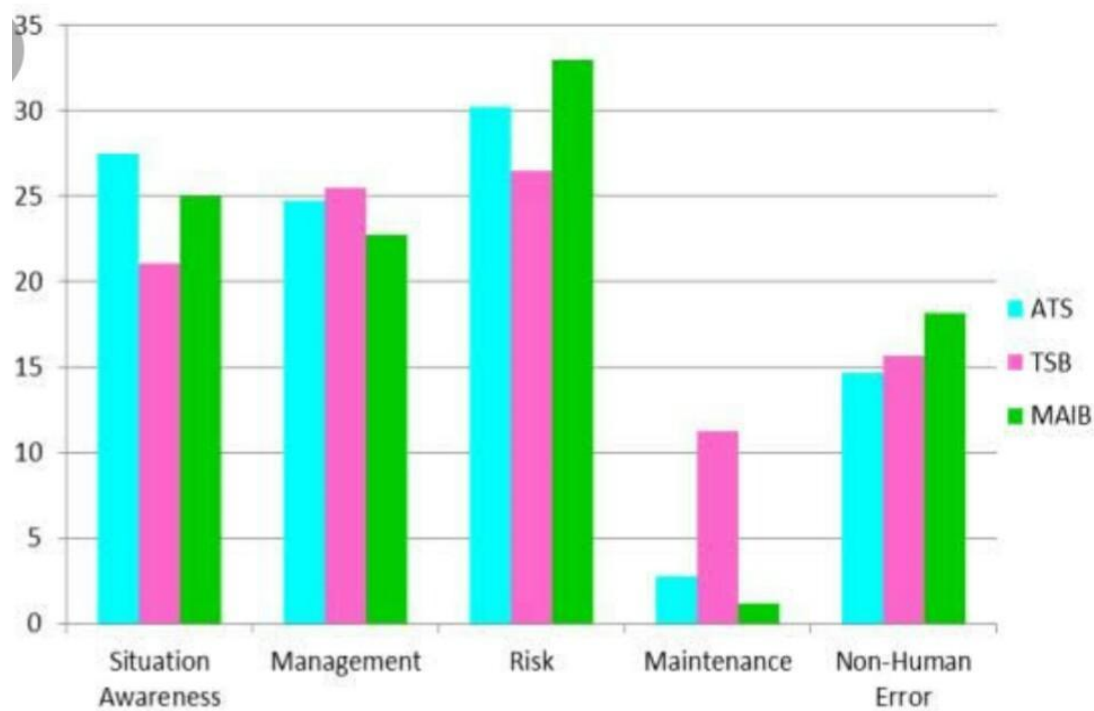


Fig: 5.15 Computational outsourcing

Large-scale problems in the physical and life sciences are being revolutionized by Internet computing technologies, like grid computing, that make possible the massive cooperative sharing of computational power, bandwidth, storage, and data. Our scheme is able to support dynamic groups efficiently, when a new user joins in the group or a user is revoked from the group, the private keys of the other users do not need to be

recomputed and updated. We provide security analysis to prove the security of our scheme.

5.7 SYSTEM DESIGN

System design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization. A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modelling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. The designs can be defined in graphical or textual modelling languages. User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase. Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

5.7.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?

- How the data should be arranged or coded?

- The dialog to guide the operating personnel in providing input.

- Methods for preparing input validations and steps to follow when error occur.

5.7.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.

- Signal important events, opportunities, problems, or warnings.

- Trigger an action.

Confirm an action.

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts. Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

5.8 SYSTEM TESTING

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified. System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together (called assemblages). System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behavior produced or observed when a component or system is tested. System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification, or both. System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification.

5.8.1 Testing Process

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

manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Testing:-

i. Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces 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 system configuration. Unit tests ensure that each unique path of business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

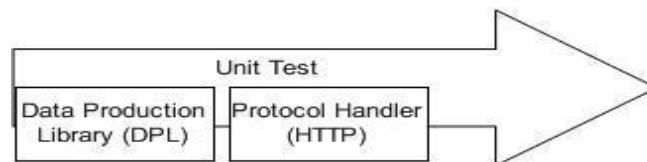


Fig: 5.16 Unit Testing

ii. Integration Testing:

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

iii. Functional Testing:

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

Functional testing is centered on the following items:

- *Valid Input* is used to identified classes of valid input must be accepted.

- *Invalid Input* is used to identified classes of invalid input must be rejected.
- *Functions* are used to identified functions must be exercised.
- *Output* is used to identify classes of application outputs.

Systems/Procedures is used to interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive Processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

iv. *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.

v. *White Box Testing:*

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

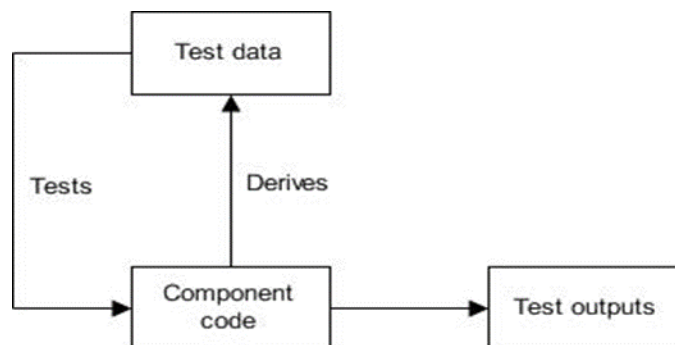


Fig: 5.17 White Box Testing

vi. *Black Box Testing:*

Black Box Testing is testing the software without any knowledge of the inner workings,

structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

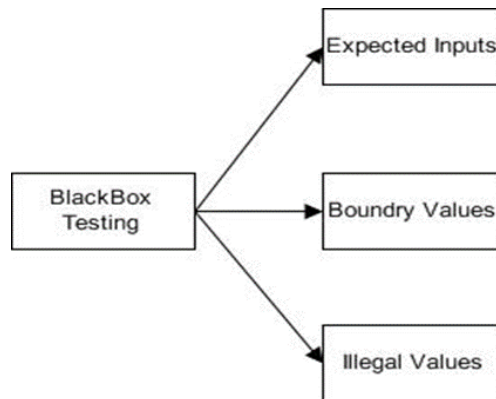


Fig: 5.18 Black Box Testing

5.8.2 Test Strategy and Approach

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Features to be tested.
- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

i. 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.

ii. 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.

iii. Alpha Testing:

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and pre loaders work.

iv. Beta Testing:

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and pre loaders work.

CHAPTER 6

RESULTS AND DISCUSSIONS

In this paper, a safe information sharing plan which can be shielded from agreement assault. The renounced clients cannot have the option to get the first information documents whenever they are denied regardless of whether they contrive with the untrusted cloud. Our plan can accomplish secure client renouncement with the assistance of polynomial capacity. Our plan can uphold dynamic gatherings effectively, when another client participates in the gathering or a client is repudiated from the

gathering, the private keys of different clients don't should be recomputed and refreshed. We give security examination to demonstrate the security of our plan.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Through the above summary, due to the problems about the collusion attacks that are widespread in the secure outsourcing of sequence comparison algorithms, this paper will introduce the trusted authority to authenticate user those who have the access to the data on cloud. SHA algorithm is used by the trusted authority to generate the key and

that key will get share to user as well as the owner. The trusted authority module receives encrypted file using AES Algorithm from the data owner and computes hash value using MD-5 algorithm. It stores key in its database which will be used during the dynamic operations and to determine the cheating party in the system. Trusted authority send file to CSP module to store on cloud. The resulting key sets are shown to have a number of desirable properties that ensure the confidentiality of communication sessions against collusion attacks by other network nodes. It is somewhat hard to extend the work in our paper to certain applications with multi-data source. Firstly, two character sequences from different sources should be encrypted respectively with different keys. Secondly, three cost matrices should be encrypted together after being constructed by the negotiation between both sides. The security target is to complete sequence comparison on a single cloud server in the way of privacy preservation and to ensure that the string typed data of the end user on any side will not be arbitrarily stolen by the other user or the CSP.

REFERENCES

- [1] Y.Feng,H.Ma,andX.Chen,“Efficient and verifiable outsourcing scheme of sequence comparisons,” *Intell. Autom. Soft Comput.*, vol. 21, no. 1, pp. 51–63, Jan. 2015.
- [2] M. J. Atallah and J. Li, “Secure outsourcing of sequence comparisons,” in *Proc. Int.*

- [3] M. J. Atallah, F. Kerschbaum, and W. Du, "Secure and private sequence comparisons," in Proc. ACM Workshop Privacy Electron. Soc. (WPES), Washington, DC, USA, 2003, pp. 39–44.
- [4] D. Szajda, M. Pohl, J. Owen, and B. Lawson, "Toward a practical data privacy scheme for a distributed implementation of the Smith-Waterman genome sequence comparison algorithm," in Proc. Netw. Distrib. Syst. Secur. Symp.
- [5] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 9, pp. 2386–2396, Sep. 2014.
- [6] R. Akimana, O. Markowitch, and Y. Roggeman, "Secure outsourcing of DNA sequences comparisons in a Grid environment," WSEAS Trans. Comput. Res., vol. 2, no. 2, pp.
- [7] M. Blanton, M. J. Atallah, K. B. Frikken, and Q. Malluhi, "Secure and efficient outsourcing of sequence comparisons," in Proc. Eur. Symp. Res. Comput. Secur. (ESORICS), Pisa, Italy, 2012, pp. 505–522.
- [8] Y. Feng, H. Ma, X. Chen, and H. Zhu, "Secure and verifiable outsourcing of sequence comparisons," in Proc. Int. Conf. Inf. Commun. Technol.
- [9] S. Salinas, X. Chen, J. Li, and P. Li, "A tutorial on secure outsourcing of large-scale computations for big data,".
- [10] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," IEEE Trans. Comput., vol. 65, no. 10, pp. 3184–3195, Oct. 2016.

APPENDIX

A. SAMPLE CODE

AES Encrypter

```
import java.io.*;
import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.CipherInputStream;
import javax.crypto.CipherOutputStream;
import javax.crypto.KeyGenerator;
import java.security.spec.AlgorithmParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AESEncrypter
{
    Cipher ecipher;
    Cipher dcipher;

    public AESEncrypter(SecretKey key)
    {
        byte[] iv = new byte[]
        {
            0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d,
            0x0e, 0x0f
        };

        AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);

        try
        {
            ecipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
```

```

dcipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
}
catch (Exception e)
{
e.printStackTrace();
}
}
byte[] buf = new byte[1024];
public void encrypt(InputStream in, OutputStream out)
{
try
{
out = new CipherOutputStream(out, ecipher);
int numRead = 0;
while ((numRead = in.read(buf)) >= 0)
{
out.write(buf, 0, numRead);
}
out.close();
}
catch (java.io.IOException e)
{
}
}
}

```

```

public void decrypt(InputStream in, OutputStream out)
{
    try
    {
        in = new CipherInputStream(in, dcipher);
        int numRead = 0;
        while ((numRead = in.read(buf)) >= 0)
        {
            out.write(buf, 0, numRead);
        }
        out.close();
    }
    catch (java.io.IOException e)
    {
    }
}

private static Key generateKey() {
    String keyValue="TheBestSecretKey";
    Key key = new SecretKeySpec(keyValue.getBytes(), "AES");
    return key;
}

public static void main(String args[])
{
    try
    {
        Key key1 = generateKey();
        /*KeyGenerator      kgen=KeyGenerator.getInstance("AES");

```

```

kgen.init(128);
SecretKey key=kgen.generateKey();
System.out.println();*/
AEEncrypter encrypter = new AEEncrypter((SecretKey)key1);
encrypter.encrypt(new FileInputStream("D:\\Documents and Settings\\JAVA-SAN\\My
Documents\\My Pictures\\2.JPG"),new FileOutputStream("C:/Encrypted.jpg"));
FileOutputStream("C:/Decrypted.jpg"));
}
catch (Exception e)
{
e.printStackTrace();
}}

```

Admin servlet

```

import com.commondb.Common_DB;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
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;

```

```

/**
 *
 * @author sentamilpandi.m
 */
@WebServlet(name = "AdminServlet", urlPatterns = {"/AdminServlet"})
public class AdminServlet extends HttpServlet {
    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.*
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            /*
             * TODO output your page here. You may use following sample code.
             */
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet AdminServlet</title>");
            out.println("</head>");
            out.println("<body>");

```

```

out.println("<h1>Servlet AdminServlet at " + request.getContextPath() + "</h1>");
out.println("</body>");
out.println("</html>");
} finally {
out.close();
}
}
/**
 * Handles the HTTP
 * <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}

/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs

```

```

* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    try {
        String adminuser=request.getParameter("adminuser");
        String adminpass=request.getParameter("adminpass");
        ResultSet      rs=Common_DB.LoginCheck("mona","adminlogin",      "Username",
        "password", adminuser, adminpass);
        if(rs.next()) {
            ArrayList<String> list=new ArrayList<String>();
            HttpSession ses1=request.getSession(true);
            System.out.println("????????");
            ResultSet rr=Common_DB.ViewTable("mona","adminlogin");
            while(rr.next())
            {
                String n=rr.getString(1);
                System.out.println("????????"+n);
                list.add(n);
            }
            ses1.setAttribute("groupname", list);
            response.sendRedirect("AdminLinks.jsp");
        }
        else {
            response.sendRedirect("Error.jsp");
        }
    } catch (Exception ex) {
        Logger.getLogger(AdminServlet.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

    }
}
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Approved Servlet

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.sql.*;

/**
 *
 * @author sentamilpandi.m
 */

```

```

public class ApprovedServlet extends HttpServlet {
    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

    protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            /*
             * TODO output your page here. You may use following sample code.
             */

            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ApprovedServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet ApprovedServlet at " + request.getContextPath() + "</h1>");

```

```

out.println("</body>");
out.println("</html>");
} finally {
out.close();
}
}

/**
 * Handles the HTTP
 * <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}

/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response

```

```

* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    HttpSession session=request.getSession();
    String UserName=session.getAttribute("username").toString();
    String group=session.getAttribute("group").toString();
    String uname = request.getParameter("slist1");
    String filename = request.getParameter("slist3");
    System.out.println("&&& "+filename +"&&&&& "+uname);
    String Approved="Approved";
    int counting=0;
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mona","root","root");
        st=con.createStatement();

        String qry = "select * from othergroup where username='"+uname+"' &&
        groupname='"+group+"' && filename='"+filename+"'";
        rs=st.executeQuery(qry);
    }
}

```

```

rs=st.executeQuery("select usercount from othergroup where groupname='"+group+"'
and username='"+UserName+"' ");
if(rs.next())
{
counting=Integer.parseInt(rs.getString("usercount"));
}
System.out.println("@@@@@"+counting);

intrs1=st.executeUpdate("updateothergroupSET
usercount='"+(counting+1)+"',status='"+Approved+"'whereusername='"+uname+"'&&
filename='"+filename+"'");
if(rs1>0)
{
response.sendRedirect("AppSuccess.jsp");
}
else
{
response.sendRedirect("AppFail.jsp");
}}
catch(Exception ex)
{
ex.printStackTrace();
}}
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */

```

```
@Override  
public String getServletInfo() {  
    return "Short description";  
}  
}
```

Delete Servlet

```
import com.commondb.Common_DB;  
import static com.mona.DownloadServlet.filename;  
import java.io.*;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.security.Key;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.Properties;  
import java.util.Random;  
import java.util.Scanner;  
import javax.crypto.SecretKey;  
import javax.crypto.spec.SecretKeySpec;  
import javax.mail.Message;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Session;  
import javax.mail.Transport;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;
```

```

import javax.servlet.RequestDispatcher;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author sentamilpandi.m
 */
public class DeleteServlet extends HttpServlet {
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;
    RequestDispatcher rd=null;
    static Properties properties=new Properties();
    static
    {
        properties.put("mail.smtp.host", "smtp.gmail.com");
        properties.put("mail.smtp.socketFactory.port", "465");
        properties.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        properties.put("mail.smtp.auth", "true");
        properties.put("mail.smtp.port", "465");
    }
    private static Key generateKey(String keyvalidation) {
        String keyValue=keyvalidation;
        Key key = new SecretKeySpec(keyValue.getBytes(), "AES");
        return key;
    }
}

```

```

}
/**
 * Processes requests for both HTTP
 * <code>GET</code> and
 * <code>POST</code> methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        /*
         * TODO output your page here. You may use following sample code.
         */
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet DeleteServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet DeleteServlet at " + request.getContextPath() + "</h1>");
        out.println("</body>");
        out.println("</html>");
    } finally {

```

```

out.close();
}
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign
on the left to edit the code.">

/**
 * Handles the HTTP
 * <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    try
    {
        HttpSession session1=request.getSession();
        String UserName=session1.getAttribute("username").toString();
        String list=session1.getAttribute("filename").toString();
        String group=session1.getAttribute("group").toString();
        String fname=request.getParameter("filename");
        String TempDownloadDirectory="D:/temp1/";
        String ln="";
        String key = null;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mona","root","password");
            st=con.createStatement();
            rs=st.executeQuery("Select product from groupname where groupname='"+group+"'");
            if(rs.next())
            {
                key=rs.getString(1);
            }
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }
}

```

```

Key key1 = generateKey(key);
"+TempDownloadDirectory+filename+"????? "+group);
AEEncrypter Decrypter = new AEEncrypter((SecretKey)key1);
Decrypter.decrypt(new FileInputStream("D:"+group+"/"+fname),new
FileOutputStream(TempDownloadDirectory+fname));
System.out.println("D:"+group+"/"+fname);
File ff=new File(TempDownloadDirectory+fname);
Scanner sc=new Scanner(ff);
while(sc.hasNextLine())
{
ln=ln+sc.nextLine().toString();
}
session1.setAttribute("data", ln);

final String from="javaredquene@gmail.com";
final String password="mona1234";
final String to="javaredquene@gmail.com";
Session session = Session.getInstance(properties, new javax.mail.Authenticator()
{
protected PasswordAuthentication getPasswordAuthentication() {
return new PasswordAuthentication(from, password);
}});
Random generator = new Random();
int r = generator.nextInt(999999);
String ran12=new Integer(r).toString();
Message message = new MimeMessage(session);
message.setFrom(new InternetAddress(from));
message.setRecipients(Message.RecipientType.TO,

```

```

    InternetAddress.parse(to));
    message.setSubject("OTP FOR YOU");
    message.setText("otp for this session is :"+ran12);
    Transport.send(message);
    System.out.println("Email Sent Successfully");
    response.sendRedirect("Delete.jsp");
}
catch(Exception ex)
{
    ex.printStackTrace();
}
}
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Download Servlet

```

import com.commondb.Common_DB;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.io.*;
import javax.servlet.ServletOutputStream;

```

```

import java.security.Key;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
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 org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;
/**
 *
 * @author sentamilpandi.m
 */
@WebServlet(name = "DownloadServlet", urlPatterns = {"/DownloadServlet"})
public class DownloadServlet extends HttpServlet {
    static String filename;
    private static Key generateKey(String keyvalidation) {
        String keyValue=keyvalidation;
        Key key = new SecretKeySpec(keyValue.getBytes(), "AES");
    }

```

```

return key;
}

private static final String dCase = "abcdefghijklmnopqrstuvwxyz";
private static final String uCase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
private static final String sChar = "!@#$%^&*";
private static final String intChar = "0123456789";

/**
 * Processes requests for both HTTP
 * <code>GET</code> and
 * <code>POST</code> methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
try {
/*
 * TODO output your page here. You may use following sample code.
 */
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet DownloadServlet</title>");

```

```

out.println("</head>");
out.println("<body>");
out.println("<h1>Servlet DownloadServlet at " + request.getContextPath() + "</h1>");
out.println("</body>");
out.println("</html>");
} finally {
out.close();
}
}
/**
 * Handles the HTTP
 * <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}
/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request

```

```

* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    try {
        HttpSession session1=request.getSession(true);
        String keyvalidation=request.getParameter("keyvalidation");
        String group=session1.getAttribute("group").toString();
        String fn = request.getParameter("slist3");
        filename=request.getParameter("filename");
        String TempDownloadDirectory="D:/temp1/";
        File file=new File(TempDownloadDirectory);
        if(!(file.exists())) {
            file.mkdir();
        }
        Key key1 = generateKey(keyvalidation);
        System.out.println(keyvalidation+"#####"+TempDownloadDirectory+filename+"??
        ??? "+group);
        AESEncrypter Decrypter = new AESEncrypter((SecretKey)key1);
        Decrypter.decrypt(new FileInputStream("D:"+group+"/"+filename),new
        FileOutputStream(TempDownloadDirectory+filename));
        if(ServletFileUpload.isMultipartContent(request)){
            try {
                String nn="";
                List<FileItem> multiparts = new ServletFileUpload(
                new DiskFileItemFactory()).parseRequest(request);

```

```

for(FileItem item : multipart){
    if(!item.isFormField()){
        String name = new File(item.getName()).getName();
        item.write( new File(TempDownloadDirectory+name));
        nn=name;
    }
}

request.setAttribute("message", "File download Successfully");
System.out.println("???????");
} catch (Exception ex) {
    request.setAttribute("message", "File download Failed due to " + ex);
}
}

Key key1 = generateKey(keyvalidation);
System.out.println("#####"+TempDownloadDirectory+filename);
AESEncrypter Decrypter = new AESEncrypter((SecretKey)key1);
Decrypter.decrypt(new FileInputStream("D:"+group+"/"+filename),new
FileOutputStream(TempDownloadDirectory+filename));
String filepath1=TempDownloadDirectory+filename;
System.out.println(filepath1+"???????"+filename+"....."+TempDownloadDirectory);
System.out.println("***** "+fn);
FileInputStream filetoDownload=new FileInputStream(filepath1);
ServletOutputStream output=response.getOutputStream();
response.setContentType("images/jpg");
response.addHeader("Content-Disposition","attachment;filename="+filename);
response.setContentLength(filetoDownload.available());
int readBytes=0;
byte[] buffer=new byte[1024];
while(filetoDownload.available()>0)

```

```

{
output.write(filetoDownload.read());
}
session1.setAttribute("keyvalidation1", keyvalidation);
output.close();
filetoDownload.close();
} catch (Exception ex) {
ex.printStackTrace();
}
}
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
return "Short description";
}
}

```

B. SCREENSHOTS

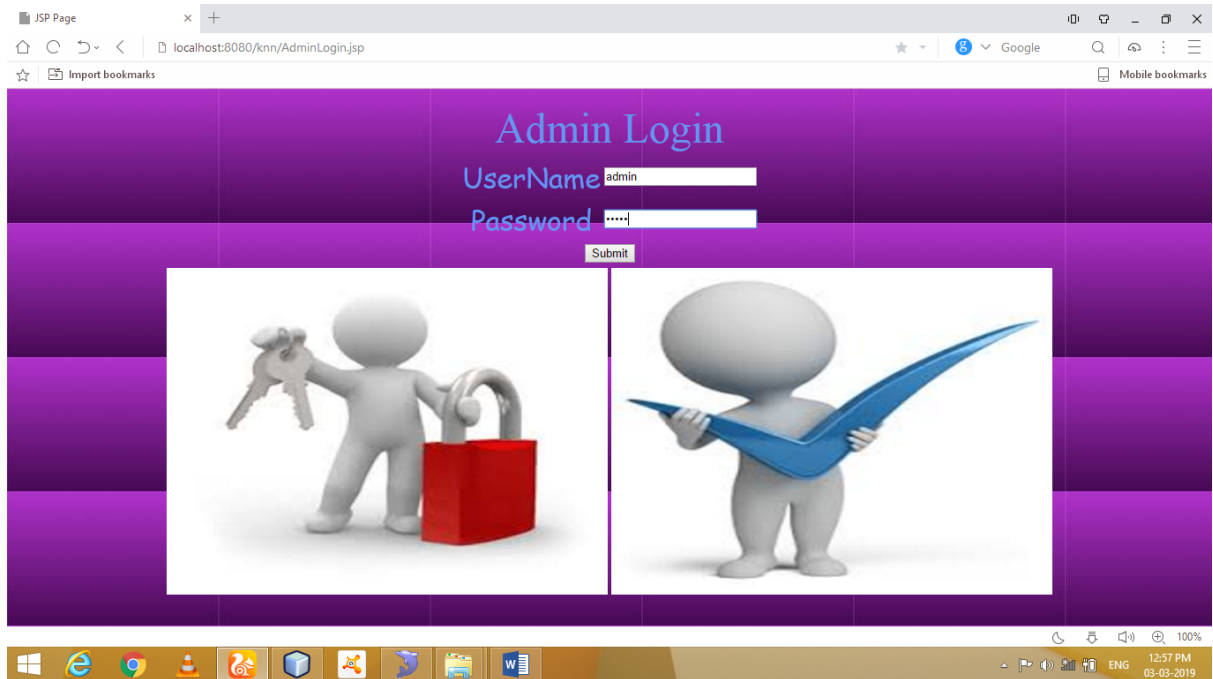


Fig: Admin Login

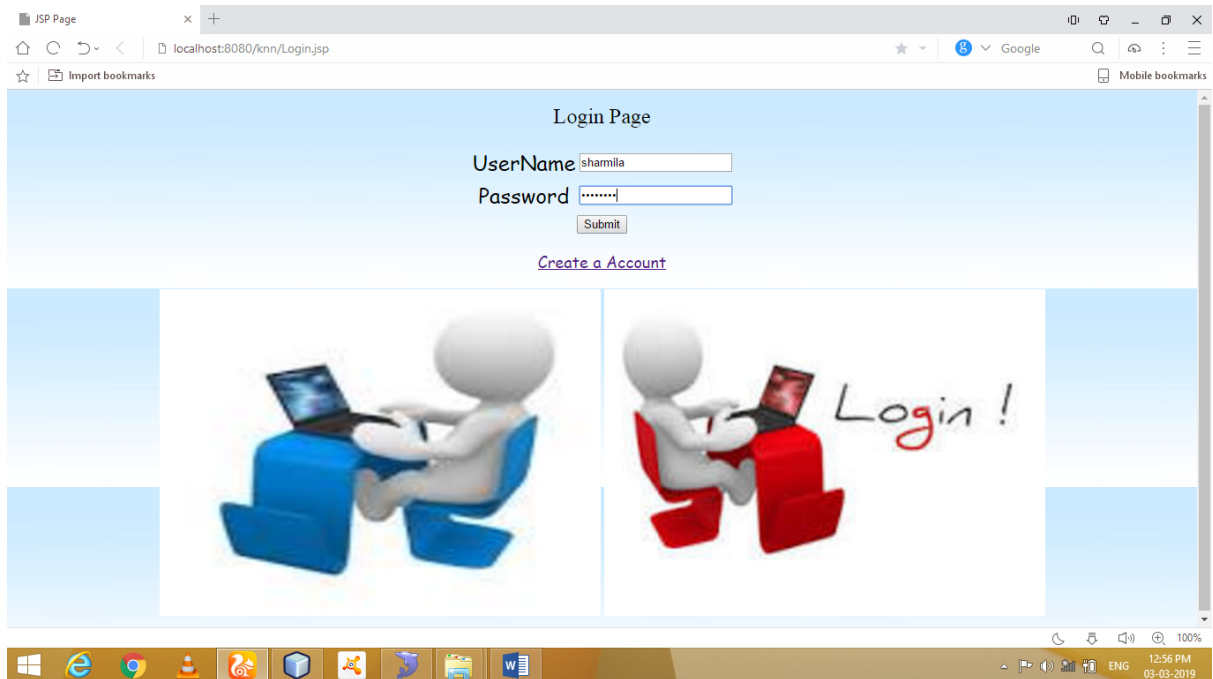


Fig: User Login

User Registration Details

UserName:

Password:

Confirm Password:

Email:

Group Name:

NEW USER REGISTRATION

REGISTER NOW!

Fig: Registration

GroupName Registration

GroupName:

GroupKey:

Online Registration

CLICK HERE

Fig: Create New Group

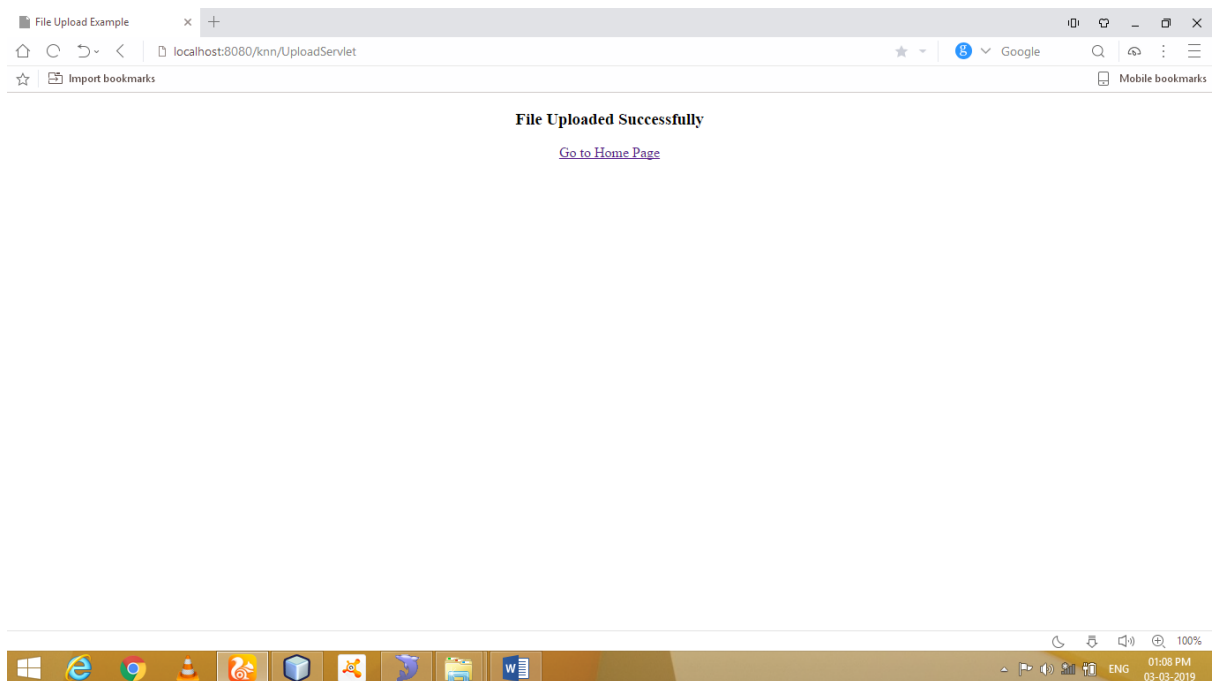
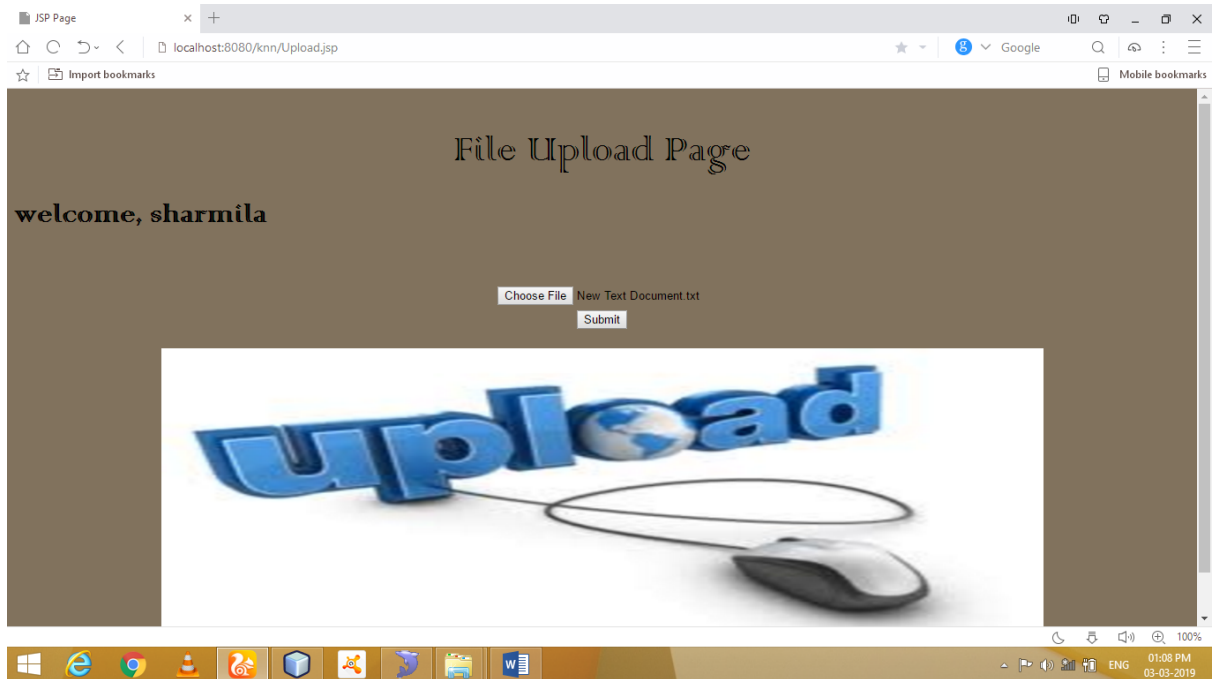
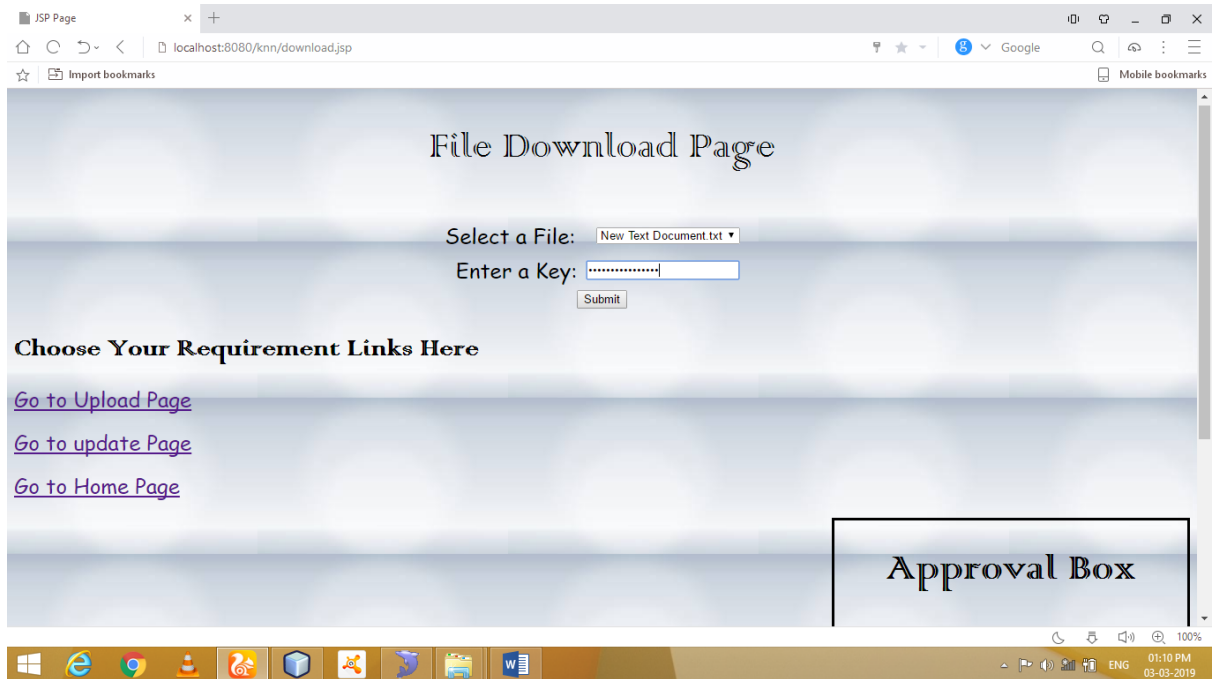


Fig: Data Upload Page



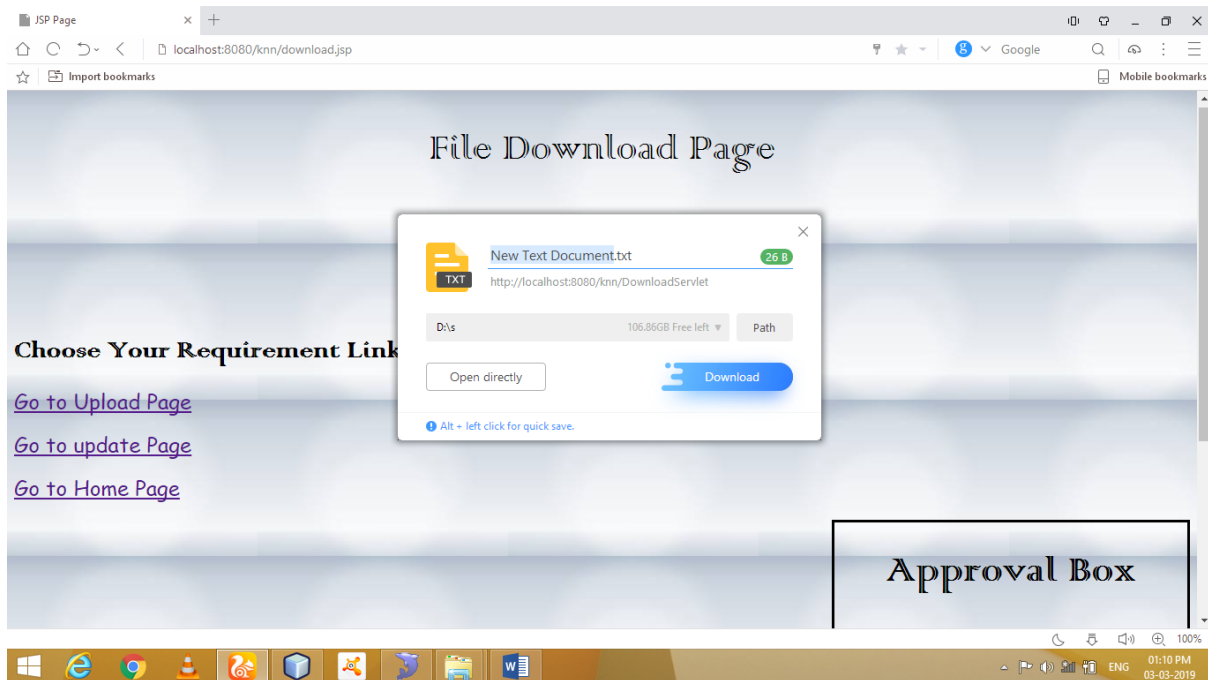


Fig: File Download Page

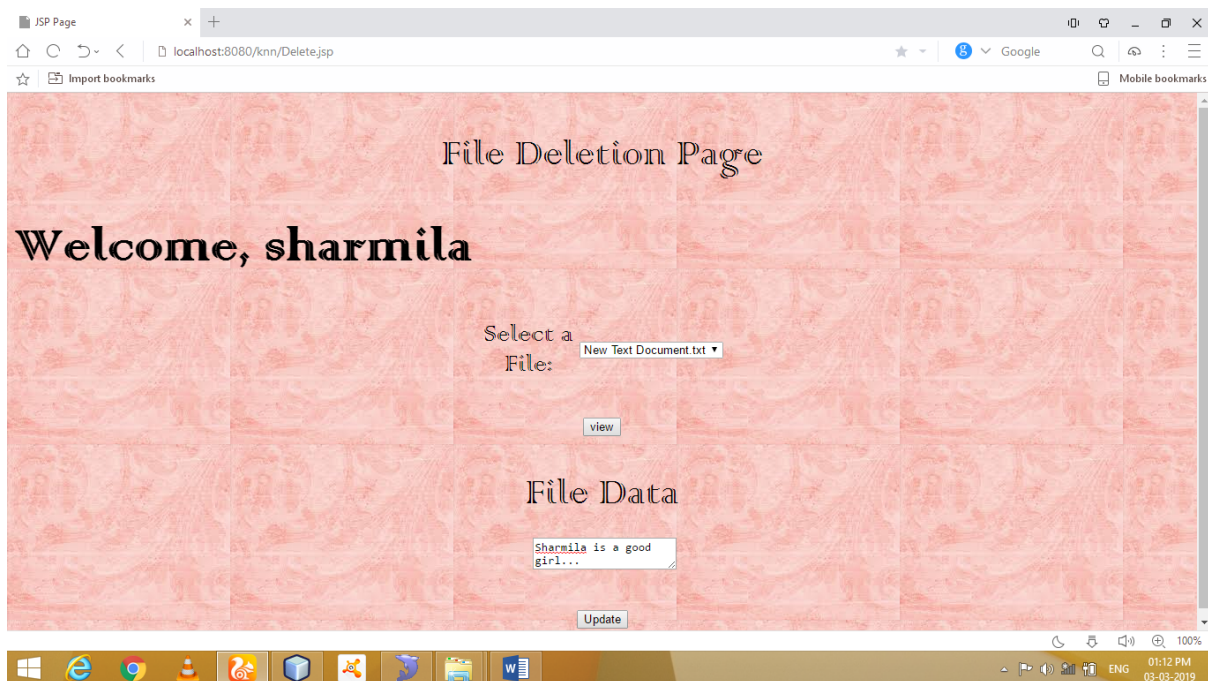


Fig: Update Page

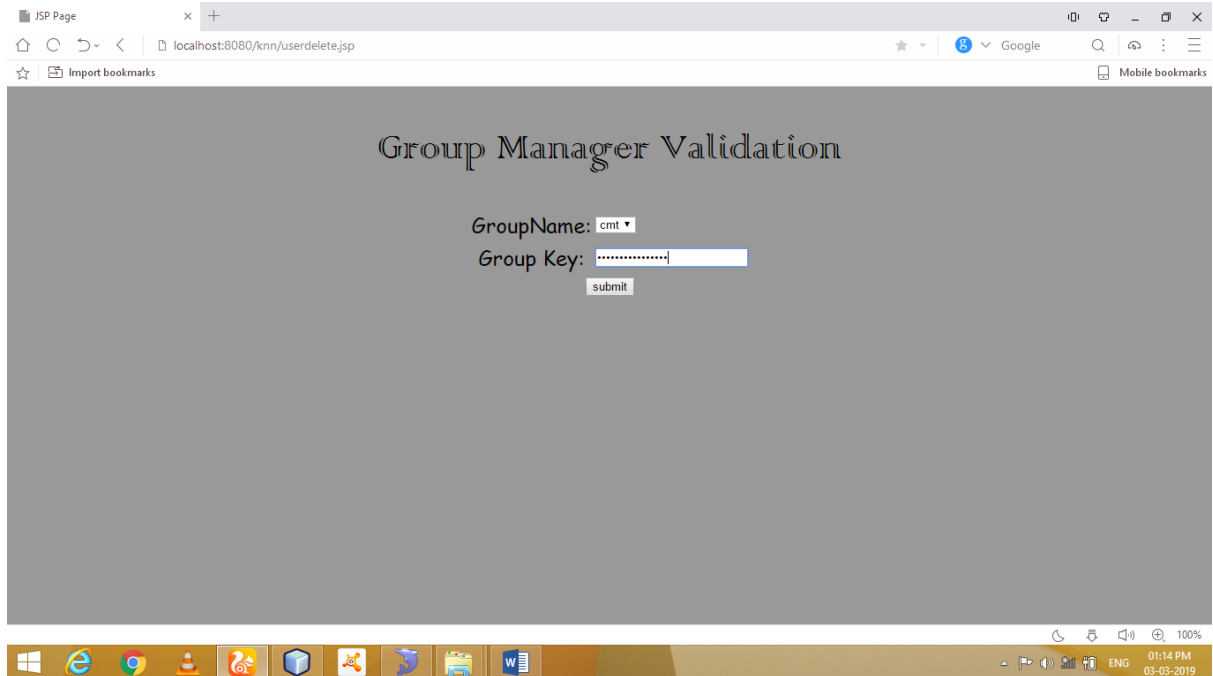


Fig: User Revocation Details

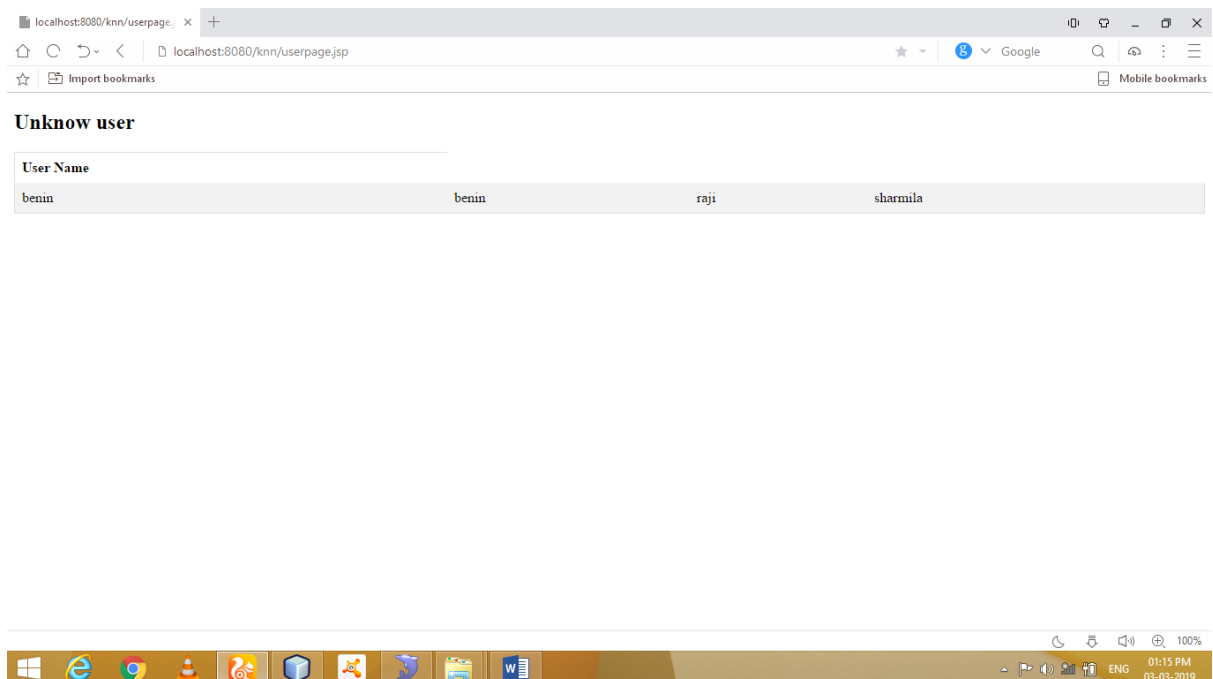


Fig: Find Third Party

C. PUBLICATION WITH PLAGIARISM REPORT