



School of Information Technology and Engineering

Department of Software and Systems Engineering
NOVEMBER 2019

**“AUTHENTICATION BY ENCRYPTED
NEGATIVE PASSWORD”**

AN INDUSTRIAL INTERNSHIP REPORT
Submitted in partial fulfillment for the award of the degree of

M.Tech
in
Software Engineering
by
SHAIK.SOHAIL ANWAR(17MIS0497)



VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering
Department of Software and Systems Engineering

DECLARATION BY THE CANDIDATE

I hereby declare that the Industrial Internship report entitled **“AUTHENTICATION BY ENCRYPTED NEGATIVE PASSWORD”** submitted by me to VIT, Vellore, in partial fulfillment of the requirement for the award of the degree of **MTech (Software Engineering)** is a record of bonafide **Industrial Internship -SWE3099** carried out by me under the guidance of **SAI PRAVEEN, MANAGER**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or university.

Place: Vellore

Date:

Signature of the Candidate



VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**School of Information Technology and Engineering
Department of Software and Systems Engineering**

BONAFIDE CERTIFICATE

This is to certify that the Industrial Internship report entitled **“AUTHENTICATION BY ENCRYPTED NEGATIVE PASSWORD”** by **SHAIK.SOHAIL ANWAR (17MIS0497)** to VIT, Vellore, in partial fulfillment of the requirement for the award of the degree of **M.Tech (Software Engineering)** is a record of bonafide work carried out by him /her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

<Prof. Name of the Internal Guide>
Signature of Internal Guide

Examiner(s) Signature

- 1.**
- 2.**



CERTIFICATE OF COMPLETION

Is here by granted to

S. SOHAIL ANWAR

Regd No: 17MIS0497

This to certify that **Mr. S. SOHAIL ANWAR** of **VELLORE INSTITUTE OF TECHNOLOGY**, has successfully completed his internship at **INFOANALYTIC SYSTEMS PVT LTD**. From **18/04/2020** to **28/05/2020**.

He has worked on the project titled "AUTHENTICATION BY ENCRYPTED NEGATIVE PASSWORD"

We found him extremely inquisitive and hardworking. He was very much interested to learn the functions of new technologies. During the internship he demonstrated good skills with self – motivative attitude towards learning.

His Association with the team was very fruitful. We wish him **All the best in the future Endeavour's**

For Infoanalytic Systems Pvt. Ltd.

Sai Manan



Authorized Manager.

Infoanalytic Systems Address - Level 5, Tech Park One Airport Road, Yerwada, Pune 411006 India
Ph : +91 9703313736 Email : info@infoanalyticsystems.com
www.infoanalyticsystems.com

ACKNOWLEDGEMENT

I wish to express our heartfelt gratitude to **Dr.G.Viswanathan**, Chancellor, VIT, Vellore, for providing facilities for the Industrial Internship. I am highly grateful to our Vice President, **Dr.G. Sekar Viswanathan**, Vice chancellor **Dr. Anand A. Samuel**, and Pro-Vice Chancellor **Dr.S.Narayanan**, for providing the necessary resources.

My sincere gratitude to **Dr. Balakrushna Tripathy**, Dean, School of Information Technology and Engineering, for giving me the opportunity to undertake the project.

I wish to express my sincere gratitude to **Dr. S. Sree Dharinya**, Head of the Department, Software and Systems Engineering, **Prof. S. Kalaivani & Prof.J. Vellingiri**, Industrial Internship Coordinators, M.Tech (Software Engineering), School of Information Technology and Engineering for providing me continuous support to do my project work.

I would like to express my special gratitude and thanks to my external guide **Mr. Sai Praveen**. Manager, INFOANALYTIC SYSTEMS PVT LTD. and internal guide **Prof. Santhosh Kumar.S.V.N** ,Assistant professor, SITE for their esteemed guidance, immense support and encouragement to complete the internship successfully.

I thank the management of VIT, Vellore for permitting me to use the library resources. I also thank all the faculty members of VIT, Vellore for giving me the courage and strength I needed to complete my goals. This acknowledgement would be incomplete without expressing my whole hearted thanks to my family and friends who motivated me during the course of the work.

Place: Vellore

Date:

SHAIK.SOHAIL ANWAR

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
	LIST OF TABLES	7
	LIST OF FIGURES	7
	LIST OF SYMBOLS	7

1.	INTRODUCTION	8
	1.1.Problem Statement	8
	1.2.Motivation	8
	1.3.Objective	8
	1.3.1.Proposed System	9
	1.3.2.Advantages of Proposed system	9
2.	TECHNOLOGIES LEARNT	9
3.	SYSTEM DESIGN	24
	3.1 system architecture	24
	3.2 Module description	24
	3.3 System Specification	25
	3.3.1 Software Requirements	25
	3..3.2 Hardware Requirements	26
	3.3 Detailed Design	26
4.	IMPLEMENTATION	30
	4.1 Implementation details	30
5.	TEST RESULTS	42
6.	RESULTS AND DISCUSSIONS	46
7.	CONCLUSION AND FUTURE WORK	56
	7.1 Conclusion	56
	7.2 Future Work	56

8.	REFERENCES	57
----	------------	----

LIST OF TABLES

Table no	Title	Page No
5	Test Case Table	24

LIST OF FIGURES

Figure no	Title	Page No
3.1	System Architecture	24
3.4.1	Use Case Diagram	27
3.4.2	Sequence diagram	28
3.4.3	Class diagram	29

LIST OF SYMBOLS

Acronym	Expansion
JVM	Java virtual Machine
API	Application programming interface
JDBC	Java Database Connectivity
ODBC	Open Data Base Connectivity
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
JRE	Java Runtime Environment

CHAPTER 1 INTRODUCTION

1.1 Problem statement :

OWING to the development of the Internet, a vast number of online services have emerged, in which password authentication is the most widely used authentication technique, for it is available at a low cost and easy to deploy. Hence, password security always attracts great interest from academia and industry. Despite great research achievements on password security, passwords are still cracked since users' careless behaviours . For instance, many users often

select weak passwords ; they tend to reuse same passwords in different systems ; they usually set their passwords using familiar vocabulary for its convenience to remember. In addition, system problems may cause password compromises. It is very difficult to obtain passwords from high security systems. On the one hand, stealing authentication data tables (containing user names and passwords) in high security systems is difficult. On the other hand, when carrying out an online guessing attack, there is usually a limit to the number of login attempts .

1.2 Motivation:

However, passwords may be leaked from weak systems . Vulnerabilities are constantly being discovered, and not all systems could be timely patched to resist attacks, which gives adversaries an opportunity to illegally access weak systems . In fact, some old systems are more vulnerable due to their lack of maintenance. Finally, since passwords are often reused, adversaries may log into high security systems through cracked passwords from systems of low security.

1.3 Objective:

After obtaining authentication data tables from weak systems, adversaries can carry out offline attacks . Passwords in the authentication data table are usually in the form of hashed passwords . However, because processor resources and storage resources are becoming more and more abundant, hashed passwords cannot resist pre computation attacks, such as rainbow table attack and lookup table attack .

Note that there is a trend of generalization of adversaries, because anyone could obtain access to information on vulnerabilities from vulnerability databases, such as the Open Source Vulnerability .Database (OSVDB), National Vulnerability Database (NVD), and the Common Vulnerabilities and Exposures (CVE) , and then make use of these information to crack systems. Moreover, they could download and use attack tools without the need for very professional security knowledge. Some powerful attack tools, such as hash cat, Rainbow Crack and John the Ripper , provide a variety of functions, such as multiple hash algorithms, multiple attack models, multiple operating systems, and multiple platforms, which raises a higher demand for secure password storage.

1.3.1 Proposed System:

❖ In the proposed system, a password protection scheme called Encrypted Negative Password (abbreviated as ENP) is proposed, which is based on the Negative Database (abbreviated as NDB) , cryptographic hash function and symmetric encryption, and a password authentication framework based on the ENP is presented. The NDB is a new security technique that is inspired by biological immune systems and has a wide range of applications.

❖ Symmetric encryption is usually deemed inappropriate for password protection. Because the secret key is usually shared by all encrypted passwords and stored together with the authentication data table, once the authentication data table is stolen, the shared key may be

stolen at the same time. Thus, these passwords are immediately compromised. However, in the ENP, the secret key is the hash value of the password of each user, so it is almost always different and does not need to be specially generated and stored. Consequently, the ENP enables symmetric encryption to be used for password protection.

1.3.2 Advantages of proposed system :

The system is more effective due to improved dynamic Key-Hashed Message Authentication Code function (abbreviated as d-HMAC) was proposed for password storage.

The system more powerful password scheme by dynamic salt generation and placement are used to improve password security.

CHAPTER 2 TECHNOLOGIES LEARNT

Software Environment

Java Technology

Java technology is both a programming language and a platform.

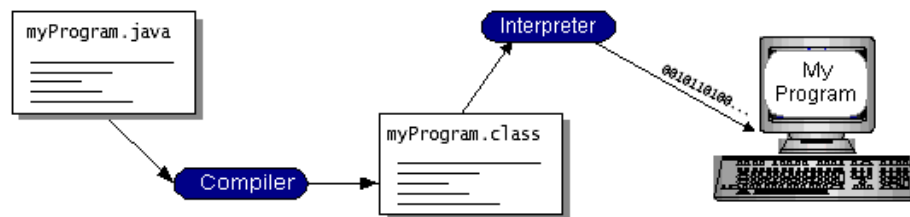
The Java Programming Language

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

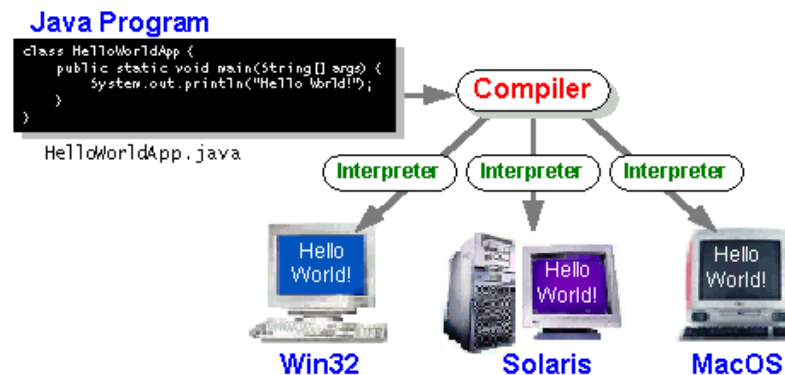
- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter

on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make —write once, run anywhere! possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

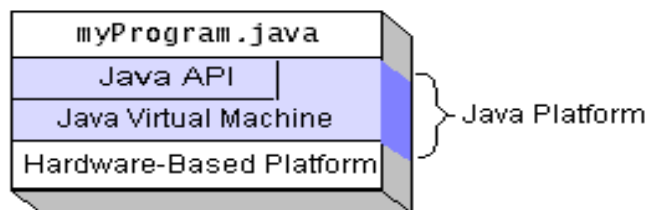
The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, *What Can Java Technology Do?* Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

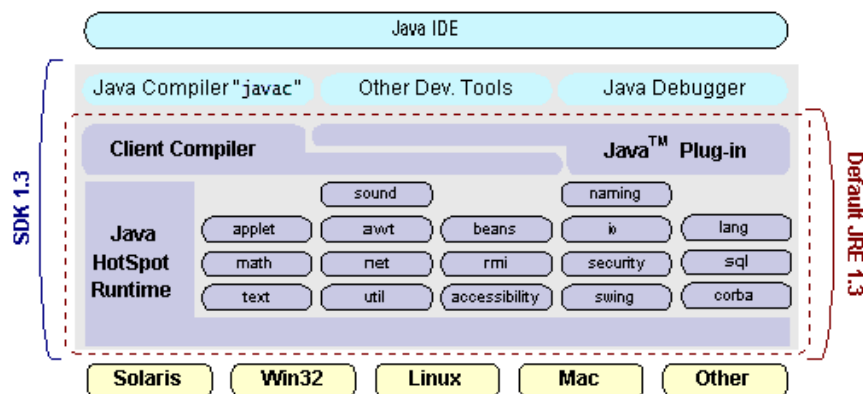
The most common types of programs written in the Java programming language are applets and applications. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs. An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server. How does the API support all these kinds of programs? It does so with packages of

software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets:** The set of conventions used by applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components:** Known as JavaBean™, can plug into existing component architectures.
- **Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™):** Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly:** Although the Java programming language is a powerful objectoriented language, it's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code:** Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Write better code:** The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly:** Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java:** You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
- **Write once, run anywhere:** Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily:** You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded —on the fly, without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should

be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems.

Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed

Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of —plug-in database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after. The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java. The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to —generate JDBC code and to hide many of JDBC's complexities from the end user.

2. SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. JDBC must be implemental on top of common database interfaces

The JDBC SQL API must —sit on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. Provide a Java interface that is consistent with the rest of the Java system

Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6. Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

7. Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

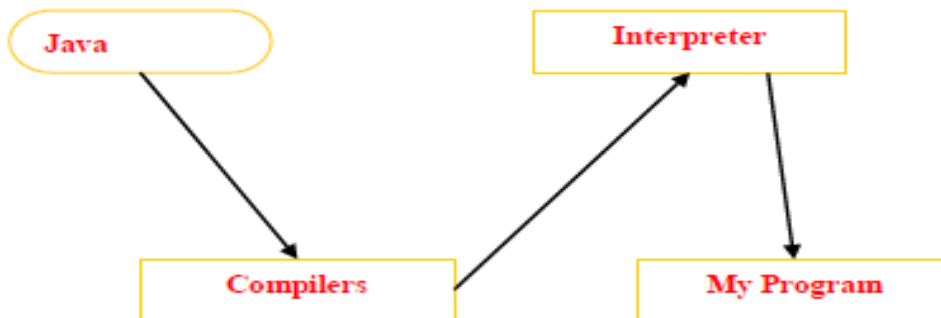
Java has two things: a programming language and a platform.
Java is a high-level programming language that is all of the following

Simple	Architecture-neutral
Object-oriented	Portable
Distributed	High-performance
Interpreted	multithreaded
Robust	Dynami
Secure	

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed.

The figure illustrates how this works.



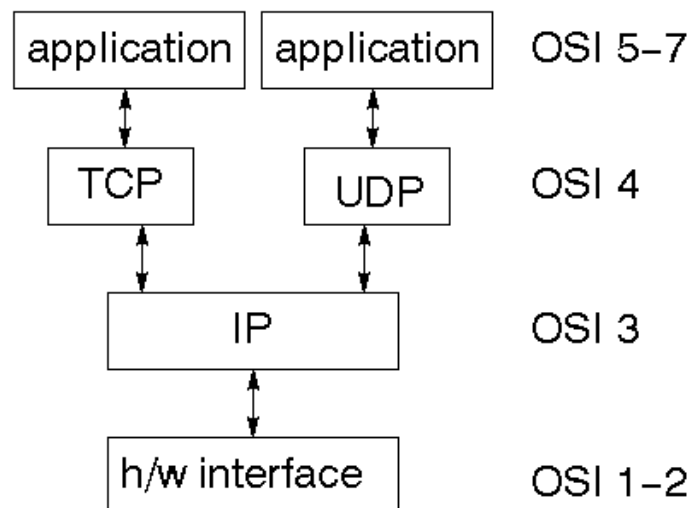
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make —write once, run anywhere possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

Networking

TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

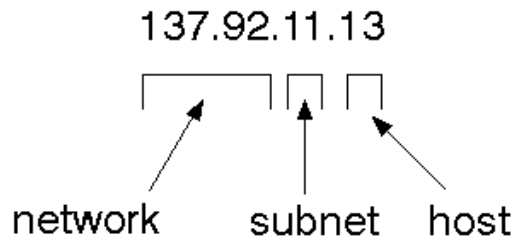
Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Total address



The 32 bit address is usually written as 4 integers separated by dots.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call `socket`. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int family, int type, int protocol);
```

Here "family" will be `AF_INET` for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

- A consistent and well-documented API, supporting a wide range of chart types;

- A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart; Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

4 Property Editors

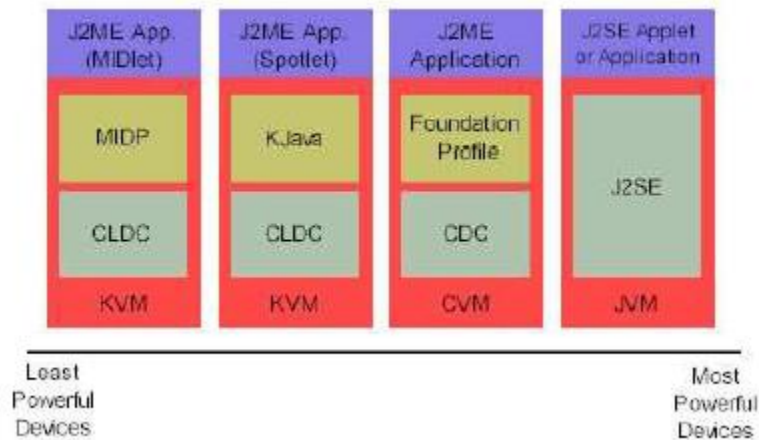
The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the Java One Developer

Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

1. General J2ME architecture



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

2. Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role pre verification plays in this process.

3. Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

- * Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

- * Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

- * Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

4.Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

- * **Connected Limited Device Configuration (CLDC)** is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

- * **Connected Device Configuration (CDC)** is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

5.J2ME profiles

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations. A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: KJava

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is `com.sun.kjava`. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

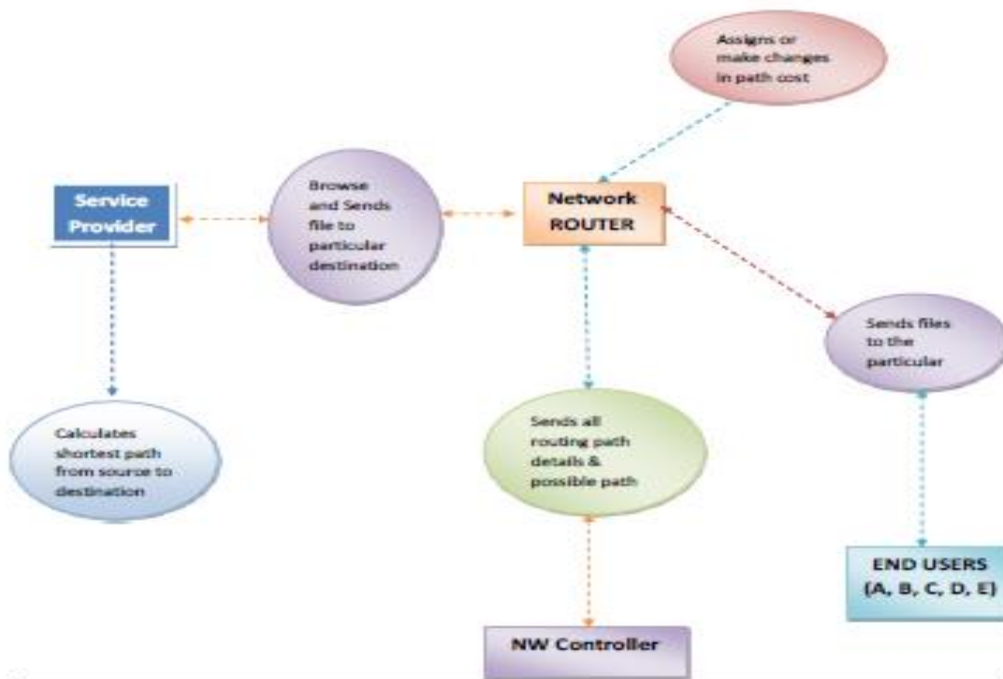
Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile Application development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

- * `java.lang`
- * `java.io`
- * `java.util`
- * `javax.microedition.io`
- * `javax.microedition.lcdui`
- * `javax.microedition.midlet`
- * `javax.microedition.rms`

CHAPTER 3 SYSTEM DESIGN

3.1 System Architecture



3.2 Module description

- **Data Owner**

In this module, the data owner uploads their data in the Web server. For the security purpose the data owner encrypts the data file and then store in the Web. The Data owner can have capable of manipulating the typed data file. The data owner will send Meta data to Audit Web. In audit Web raw or metadata information is available for auditing and data integrity checking purpose. Data owner will create an end user and the data owner can set the access permission (read or write) to user and also Verifies Password.

Data Auditing and Verification

The data owner can also audit the data integrity in the corresponding Web for verifying whether the data is safe or not using digital sign and web URL. If the data is not safe then he will delete the data and re upload the data to the corresponding Web server.

- **Web Server**

The Web server is responsible for data storage and file authorization for an end user. The data file will be stored with their tags such as file name, secret key, digital sign, and owner name. The data file will be sending based on the privileges. If the privilege is correct then the data will be sent to the corresponding user and also will check the file name, end user name and secret key.

If all are true then it will send to the corresponding user or he will be captured as attacker. The Web server can also act as attacker to modify the data which will be auditing by the audit Web and also View All Encrypted Negative Password, View All Attacker, View All Password Attackers.

- **Data Consumer(End User)**

The data consumer is nothing but the end user who will request and gets file contents response from the corresponding Web servers. If the file name and secret key, access permission is correct then the end is getting the file response from the Web or else he will be considered as an attacker and also he will be blocked in corresponding Web. If he wants to access the file after blocking he wants to UN block from the Web and also verifies password.

- **Attacker**

Attacker is one who is integrating the Web file by adding malicious data to the corresponding Web. They may be within a Web or from outside the Web. If attacker is from inside the Web then those attackers are called as internal attackers. If the attacker is from outside the Web then those attackers are called as external attackers.

3.3 System Specification

3.3.1 Software Requirements

Functional requirements for a secure cloud storage service are straightforward:

1. The service should be able to store the user's data;
2. The data should be accessible through any devices connected to the Internet;
3. The service should be capable to synchronize the user's data between multiple devices (notebooks, smart phones, etc.);
4. The service should preserve all historical changes (versioning);
5. Data should be shareable with other users;
6. The service should support SSO; and
7. The service should be interoperable with other cloud storage services, enabling data migration from one CSP to another.

Operating System : Windows XP

Application Server : Tomcat 5.0

Front End : HTML, Java, Jsp

Scripts : JavaScript.

Server side Script : Java Server Pages.

Database : Mysql 5.0

Database Connectivity : JDBC.

3.3.2 Hardware Requirements:

- Processor : Intel Core2 Duo
- Seed : 2.4 GHz
- RAM : 2 GB(minimum)
- Hard Disk : 180 GB

Cloud computing has three fundamental models, these are:

3.4 Detailed Design

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modelling and documenting software. In fact, it's one of the most popular business process modelling techniques. It is based on **diagrammatic representations** of software components. As the old proverb says: picture is worth a thousand words. By using visual representations, we are able to better understand possible flaws or errors in software or business processes. UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

GOALS:

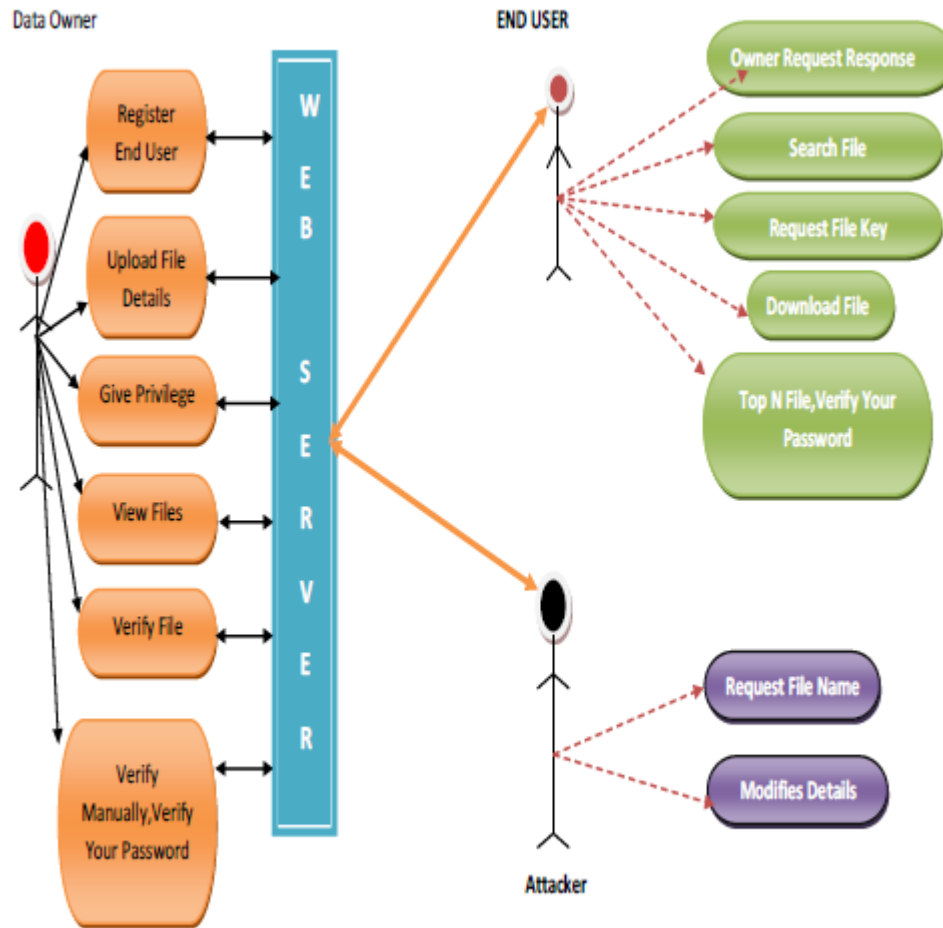
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
- 6 Support higher level development concepts such as collaborations, frameworks, patterns and components.
- 7 Integrate best practices.

i. USE CASE DIAGRAM:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural 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. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

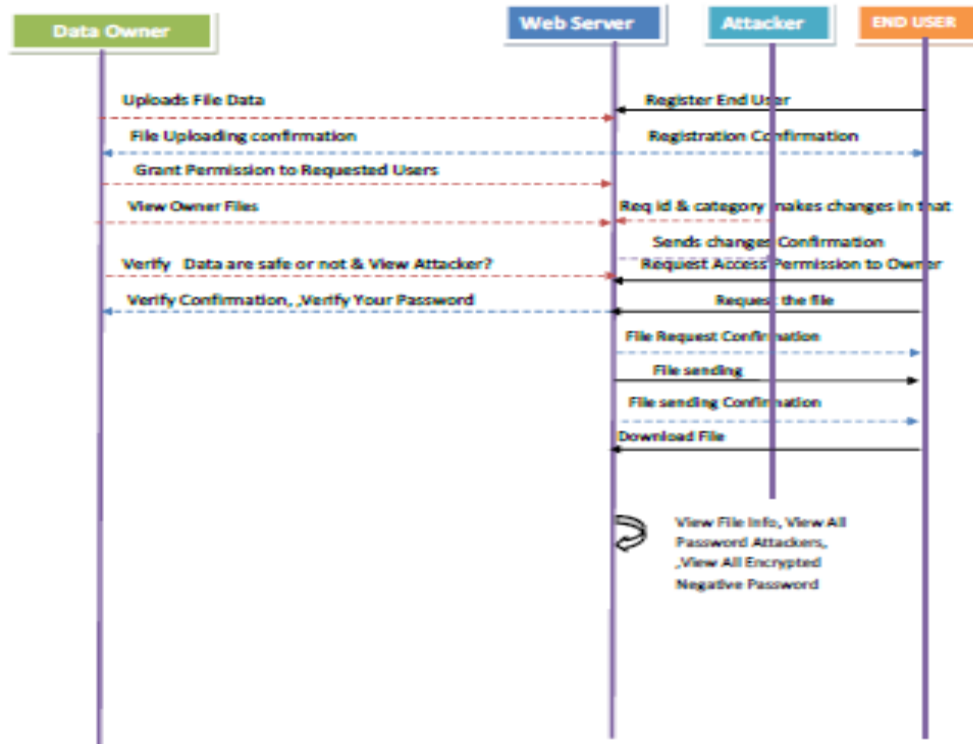
Use Case Diagram



ii. SEQUENCE DIAGRAM:

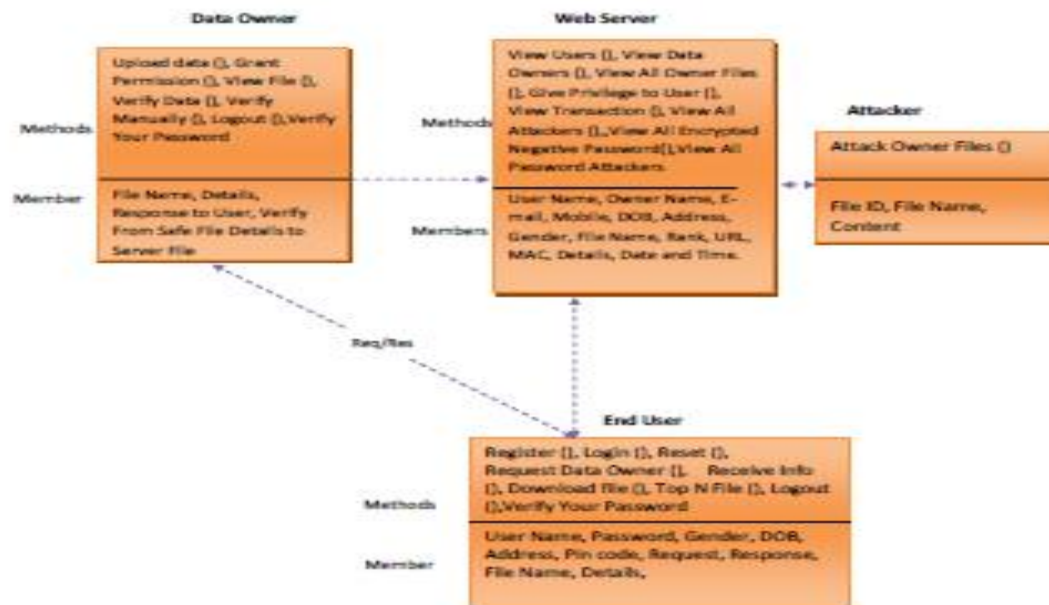
A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

Sequence Diagram



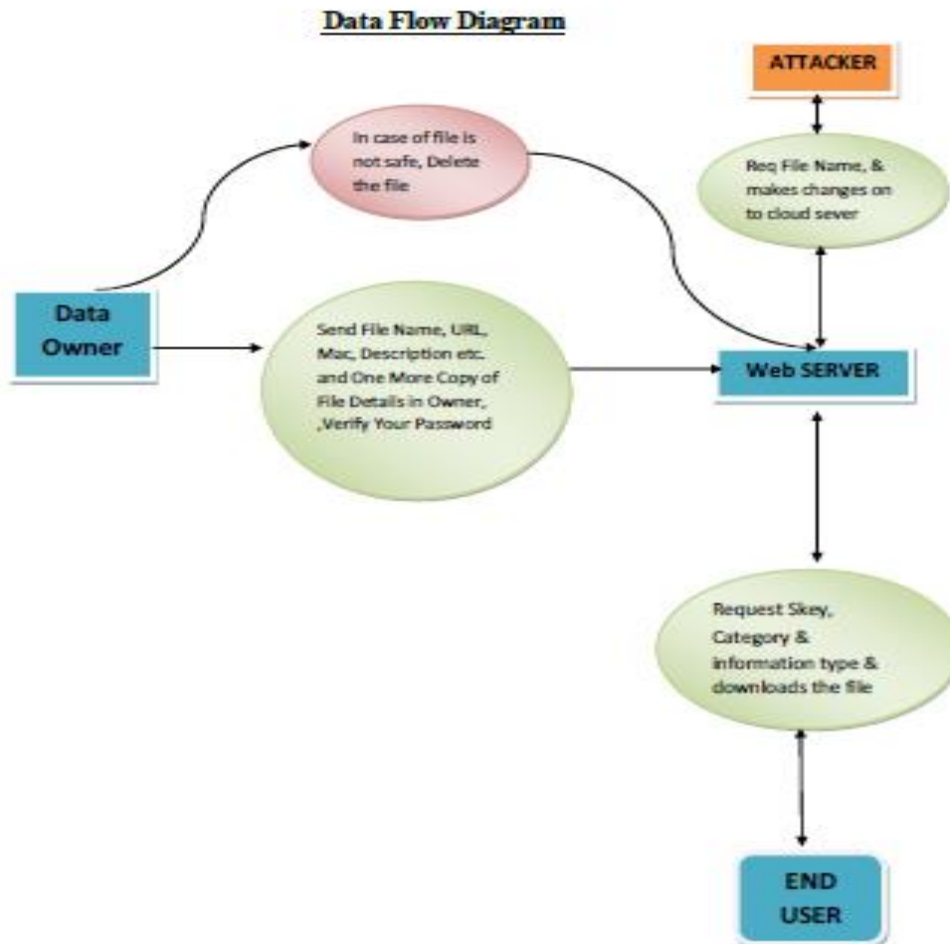
iii. CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



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.. DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:



CHAPTER 4 IMPLEMENTATION

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Main Menu</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>
<script type="text/javascript" src="js/radius.js"></script>

```

```

<style type="text/css">
<!--

.style2 { colour: #1f7fbb}
.style8 {
color: #FFFFFF;
font-weight: bold;
}
-->
</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html">Authentication by ENP</a></h1>
</div>
<div class="clr"></div>
<div class="menu_nav">
<ul>
<li><a href="index.html">Home</a></li>
<li class="active"><a href="endUserLogin.html">Client</a></li>
<li><a href="dataOwnerLogin.html"></a></li>
<li><a href="webServerLogin.html">WebServer</a></li>
<li></li>
<li></li>
</ul>
</div>
<div class="clr"></div>

<div class="clr"></div>
</div>
</div>
<div class="clr"></div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>Search File <span class="style2"></span></span></h2>
<div class="clr"></div>
<div class="clr"></div>
<p> <form action="Downloadadd.jsp" method="post" name="form1" id="form1">
<table width="478" border="0" align="center">
<tr>
<td width="223" bgcolor="#FF0000"><span class="style1 style8">Enter File

```



```

Name</span></td>
<td width="245"><label>
<input required name="t1" type="text" value="" size="40" />
</label></td>

</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td><div align="right"></div></td>
<td><label>
<input type="submit" name="Submit" value="Search" />
</label></td>
</tr>
</table>
</form>
</p>
<p align="right"><a href="endUserMain.jsp">Back</a></p>
</div>
</div>
<div class="sidebar">
<div class="gadget">
<h2>Sidebar Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><a href="endUserMain.jsp">Client Main</a></li>
<li><a href="endUserLogin.html">Log Out</a></li>
</ul>
</div>
</div>
<div class="clr"></div>
<div class="fbg"></div>
<div class="clr"></div>

</div>
</div>
<div class="footer">
<div class="clr"></div>
</div>
</div>
</body>
</html><script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 24px}

```

```
.style2 {color: #0000FF}
.style3 {color: #5E6D4E}
.style4 {font-weight: bold}
-->
</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html" class="style1">Authentication By Encrypted Negative Password<
</a></h1>
</div>
<div class="menu_nav">
<ul>
<li class="active"><a href="index.html"><span>Home</span> Page </a></li>
<li><a href="DataOwnerLogin.html">Data Owner </a></li>
<li><a href="EndUserLogin.html">End User</a></li>
<li><a href="CloudServerLogin.html">Cloud Server</a>

</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"> <a href="#"> </a> <a href="#"> </a> <a
href="#"> </a> </div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>WELCOME TO HOME PAGE </span></h2>
<p class="infopost"><a href="#" class="com"><span>11</span></a></p>
<div class="clr"></div>
<div class="clr"></div>
</div>
<div class="article">
<h2><span></span><form action="Attack1.jsp" method="post" name="form1" id="form1">
<table width="616" border="0" align="left">
```

```

<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td width="333"><span class="style1">Enter File Name :-</span></td>
<td width="273"><label> <input required name="t1"
type="text" size="40" /> </label></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td><span class="style1">Ur Name :-</span></td>
<td><input name="ot" type="text" size="40" /></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>
<div align="right"><input type="submit" name="Submit"
value="Attack" /></div>
</td>
</tr>
</table>
</form></h2>
<p class="infopost">&nbsp;</p>
<div class="clr"></div>
<div class="clr"></div>
</div>
</div>
<div class="sidebar">
<div class="searchform">

<form id="formsearch" name="formsearch" method="post" action="#">
<span>
<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80"
value="Search our ste:" type="text" />
</span>
<input name="button_search" src="images/search.gif" class="button_search" type="image" />

```

```

</form>
</div>
<div class="clr"></div>
<div class="gadget">
<h2 class="star"><span>Home</span> Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><a href="index.html">Home Page </a></li>
<li><a href="DataOwnerLogin.html">Data Owner </a></li>
<li><a href="EndUserLogin.html">End User</a></li>
<li><a href="WebServerLogin.html">Cloud Server</a></li>
</ul>
</div>
<div class="gadget">
<h2 class="star">&nbsp;</h2>
</div>
</div>
<div class="clr"></div>
</div>
<div class="fbg">
<div class="fbg_resize">
<div class="clr"></div>
</div>
</div>
<div class="footer">
<div class="footer_resize">
35
<div style="clear:both;"></div>
</div>
</div>
</div>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<% @page import ="java.util.*"%>
<% @page import ="java.sql.*"%>
<% @page import
="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKe
ySpec,or
g.bouncycastle.util.encoders.Base64"%>
<% @ page
import="java.sql.*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileIn
putStrea
m,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.Buf
feredInp
utStream" %>

```

```

<% @ page import
="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"
%>
<% @page import
="java.util.*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutp
utStream,
java.io.PrintStream"%>
<% @ include file="connect.jsp" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>WEB SERVER </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-titillium-250.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style21 {font-size: 14px}
.style24 {color: #FF0000}
.style28 {font-size: 24px}
.style29 {font-size: 14px; color: #FF0000; font-weight: bold; }
-->
</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html" class="style28">A Distributed Trust Evaluation Protocol<br />
with Privacy Protection for Intercloud</a></h1>
</div>
<div class="menu_nav">
<ul>
<li><a href="C_Main.jsp">WEB SERVER </a></li>
<li><a href="C_Login.jsp"><span>Logout </span></a></li>
<li></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">

```

```

<div id="coin-slider"> <a href="#"></a> <a href="#"></a> <a
href="#"></a> </div>
<div class="clr"></div>
</div>
<div class="clr"></div>

</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>Web Server </span></h2>
<p>&nbsp;</p>
<table width="565" border="2" cellpadding="0" style="border-collapse:collapse"
cellspacing="0"
align="center">
<tr>
<td width="99" height="30" bgcolor="#FFFF00"><div align="center" class="style21
style24"><strong> ID </strong></div></td>
<td width="174" bgcolor="#FFFF00"><div align="center" class="style29">User Name
</div></td>
<td width="252" bgcolor="#FFFF00"><div align="center"
class="style29">Permission</div></td>
<%
String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8,s9="",s10,s11,s12,s13;
int i=0,j=0,k=0;
try
{
String query="select * from request";
Statement st=connection.createStatement();
ResultSet rs=st.executeQuery(query);
while ( rs.next() )
{
i=rs.getInt(1);
s2=rs.getString(2);
s3=rs.getString(3);
%>
</tr>
<tr>
<td height="28"><div align="center" class="style24"><%=i%></div></td>
<td><div align="center" class="style24"><a
href="C_UserSGDetails.jsp?uname=<%=s2%>"><%=s2%></a></div></td>
<%
if(s3.equalsIgnoreCase("Requested"))

```

```

{
%>
<td><div class="style24">
<div align="center"><a href="C_GrantSearch1.jsp?usid=<%=i%>"><%=s3%></a></div>
</div></td>
<%
}else
{
%>
<td width="28"><div class="style24">
<div align="center"><%=s3%></div>
</div></td>
</tr>
<%
}
}
connection.close();
}
catch(Exception e)
{
out.println(e.getMessage());
}
%>

</table>
<p>&nbsp;</p>
<p align="right"><a href="C_Main.jsp">Back</a></p>
</div>
</div>
<div class="sidebar">
<div class="gadget">
<h2 class="star">Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><strong><a href="C_Main.jsp">Home</a></strong></li>
<li><strong><a href="C_Login.jsp">Logout</a></strong></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg"></div>
<div class="footer">
<div class="footer_resize">
<p class="lf">&nbsp;</p>
<p class="rf">&nbsp;</p>

```

```

<div style="clear:both;"></div>
</div>
</div>
</div>
<div align=center></div>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<% @page import ="java.util.*"%>
<% @page import ="java.sql.*"%>
40
<% @page import
="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKey
KeySpec,org
g.bouncycastle.util.encoders.Base64"%>
<% @ page
import="java.sql.*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileIn
putStream,
java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.Buf
feredInp
utStream" %>
<% @ page import
="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"
%>
<% @page import
="java.util.*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutp
utStream,
java.io.PrintStream"%>
<% @ include file="connect.jsp" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>WEB SERVER </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-titillium-250.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style21 {font-size: 14px}
.style24 {color: #FF0000}
.style28 {font-size: 24px}
.style29 {font-size: 14px; color: #FF0000; font-weight: bold; }

```



```

-->
</style>
</head>

<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html" class="style28">A Distributed Trust Evaluation Protocol<br />
with Privacy Protection for Intercloud</a></h1>
</div>
<div class="menu_nav">
<ul>
<li><a href="C_Main.jsp">WEB SERVER</a></li>
<li><a href="C_Login.jsp"><span>Logout </span></a></li>
<li></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"> <a href="#"></a> <a href="#"></a> <a
href="#"></a> </div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>Grant Search Control </span></h2>
<p>&nbsp;</p>
<table width="565" border="2" cellpadding="0" style="border-collapse:collapse"
cellspacing="0"
align="center">
<tr>

<td width="99" height="30" bgcolor="#FFFF00"><div align="center" class="style21
style24"><strong> ID </strong></div></td>
<td width="174" bgcolor="#FFFF00"><div align="center" class="style29">User Name
</div></td>
<td width="252" bgcolor="#FFFF00"><div align="center"
class="style29">Permission</div></td>

```

```

<%
String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8,s9="",s10,s11,s12,s13;
int i=0,j=0,k=0;
try
{
String query="select * from request";
Statement st=connection.createStatement();
ResultSet rs=st.executeQuery(query);
while ( rs.next() )
{
i=rs.getInt(1);
s2=rs.getString(2);
s3=rs.getString(3);
%>
</tr>
<tr>
<td height="28"><div align="center" class="style24"><%=i%></div></td>
<td><div align="center" class="style24"><a
href="C_UserSGDetails.jsp?uname=<%=s2%>"><%=s2%></a></div></td>
<%
if(s3.equalsIgnoreCase("Requested"))
{
%>

<td><div class="style24">
<div align="center"><a href="C_GrantSearch1.jsp?usid=<%=i%>"><%=s3%></a></div>
</div></td>
<%
}else
{
%>
<td width="28"><div class="style24">
<div align="center"><%=s3%></div>
</div></td>
</tr>
<%
}
}
connection.close();
}
catch(Exception e)
{
out.println(e.getMessage());
}
%>
</table>
<p>&nbsp;</p>

```

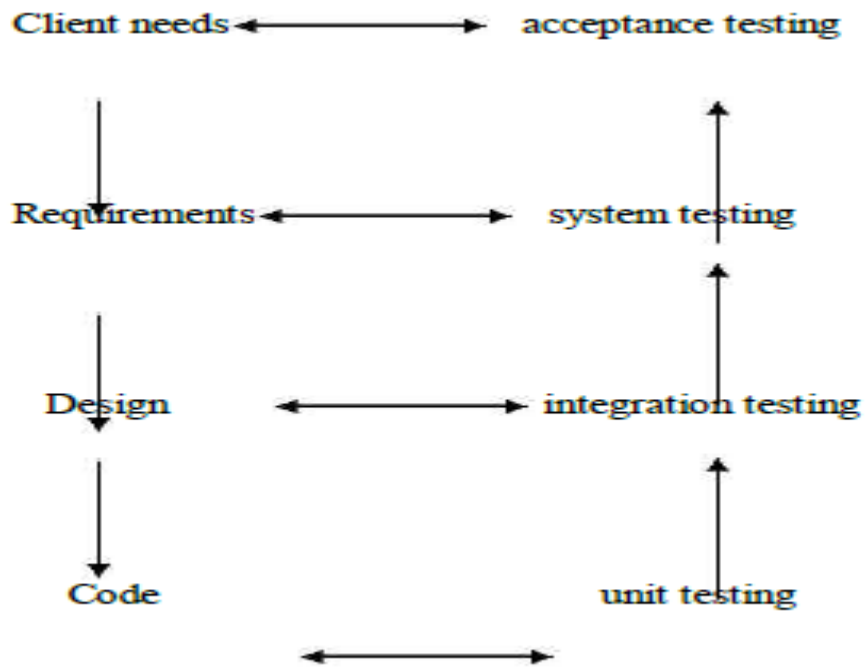
```
<p align="right"><a href="C_Main.jsp">Back</a></p>
</div>
</div>
<div class="sidebar">
<div class="gadget">
<h2 class="star">Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><strong><a href="C_Main.jsp">Home</a></strong></li>
44
<li><strong><a href="C_Login.jsp">Logout</a></strong></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg"></div>
<div class="footer">
<div class="footer_resize">
<p class="lf">&nbsp;</p>
<p class="rf">&nbsp;</p>
<div style="clear:both;"></div>
</div>
</div>
</div>
<div align=center></div>
</body>
</html>
```

CHAPTER – 5

TEST RESULTS

Types of Testing

The basic levels of Testing:



Functional Testing

Real tests give efficient protests that functions tested are attainable as specific by the business and technical requirements, system documentation, and user manuals. known categories of application outputs should be exercised.

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

TYPES OF TESTS

Unit testing:

A unit is the smallest piece of source code that can be tested. It is also known as a module which consists of numerous lines of code that are processed by a single programmer. The key purpose of performing unit testing is to expose that a particular unit doesn't satisfy the specified functional

requirements and also to show that the structural implementation is not like to the projected structure designed.

Integration testing:

Tests are intended to test incorporated programming segments to figure out whether they really keep running as one system. Testing is occasion driven and is more worried with the fundamental result of screens or fields. Reconciliation tests exhibit that in spite of the fact that the parts were separately fulfilment, as appeared by effectively unit testing, the blend of segments is right and comprised.

Integration testing is specifically aimed at revealing the problems that rise from the mixture of components.

Functional test:

Functional tests give efficient challenges that capacities tried are accessible as determined by the business and specialized necessities, framework documentation, and client manuals Functional testing is centered on the following items:

Output : identified classes of application outputs must be exercised.

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

Association and arrangement of practical tests is centered around prerequisites, key capacities, or unique

experiments. Likewise, efficient scope relating to recognize Business procedure streams; information fields, predefined procedures, and progressive procedures must be considered for testing. Before

utilitarian testing is finished, extra tests are distinguished and the powerful estimation of current tests is resolved.

System Test

System testing guarantees that the entire coordinated programming framework meets prerequisites. It tests a design to guarantee known and unsurprising results. A sample of framework testing is the arrangement situated framework combination test. Framework testing depends on procedure portrayals and streams, stressing pre-driven procedure connections and mix focuses.

White Box Testing

It is a testing in which the product analyzer has information of the internal workings, structure and dialect of the product, or if nothing else its motivation. It is reason. It is utilized to test ranges that can't be gotten a handle on from a discovery level.

Black Box Testing

It is the testing the product with no information of within workings, structure or dialect of the part

being tried. Discovery tests, as most different sorts of tests, must be composed from a complete source report, for example, prerequisite or necessities archive, for example, determination or necessities record. It is a trying in which the product under test is dealt with, as a discovery .you can't "see" into it. The test gives inputs and reacts to yields without considering how the product functions.

Unit Testing:

Unit testing is by and large appeared as a major aspect of a joined code and unit test period of the product lifecycle, in spite of the fact that it is not exceptional for coding and unit testing to be directed as two unmistakable stages.

Test strategy and approach

Ground testing will be done physically and functional tests will be inscribed in detail.

Test objectives

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

Features to be tested

- Validate that the accesses are of the correct format
- No duplicate entries should be allowed
- Entire links must gross the user to the accurate page.

Integration Testing

Software integration testing is the incremental combination analysis of two or more joint software components on a single platform to generate failures created by boundary faults.

The task of the integration test is to design those components or s/w applications, e.g. modules in a software system or – one step up – software presentations at the company level – interact without faults.

Test Results: All the test cases stated above passed effectively. No defects met.

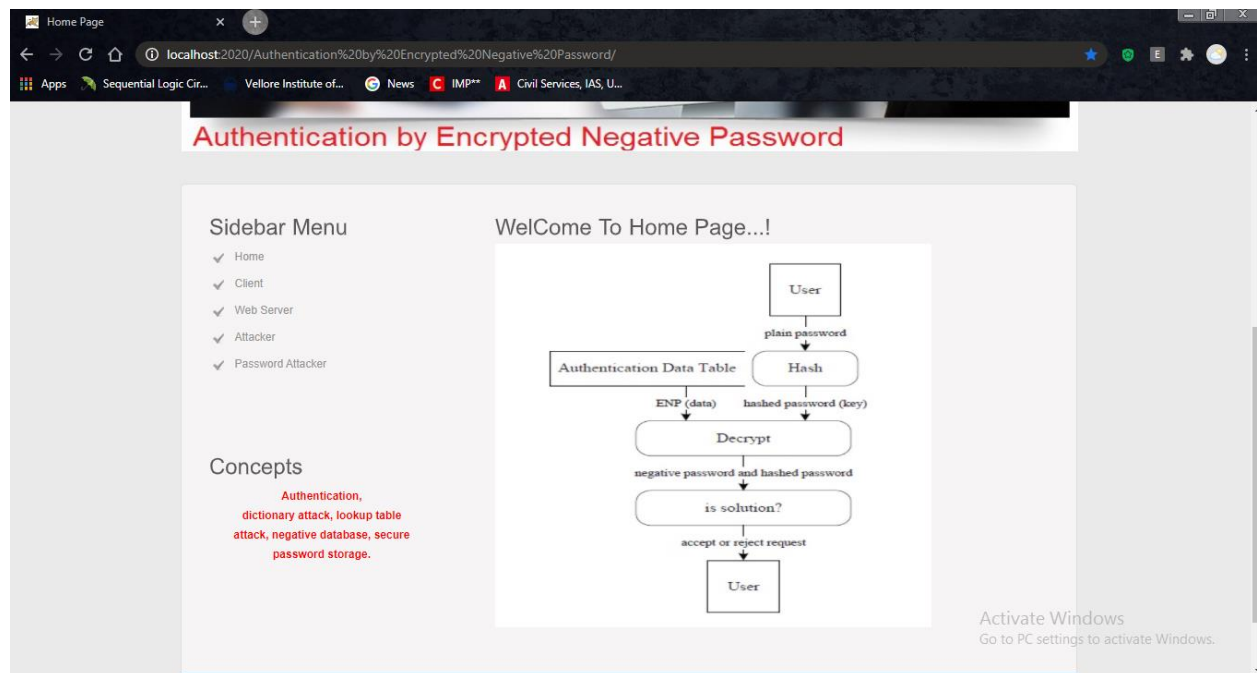
Acceptance Testing

User Acceptance Testing is a serious phase of any project and needs important contribution by the end user. It also guarantees that the system encounters the functional requirements.

Test case id	Test case description	Actual value	Entered value	Status
1	Register user details in registration page	Fill all the fields while registering user	All the fields are filled	Pass
2	Give user name in text box	User name must be given in alphabets	User name given in alphabets and numeric values	Fail
3	Password to be entered in password box	Password must be given correctly	Password is entered wrongly	Fail
4	Phone number must be entered in phone number box during registration	Phone number must be given in 10 digits	Phone number given in 10 digits	Pass
5	Validating the functionality of Browse button	System should select the corresponding file	selected the file what we expected	Pass

Test Results: All the test cases stated above passed effectively. No defects met.

CHAPTER 6 RESULTS



User Register Page

localhost:2020/Authentication%20by%20Encrypted%20Negative%20Password/newUserRegister.jsp

Sequential Logic Cir... Vellore Institute of... News IMP** Civil Services, IAS, U...

Client Registration...!

Select User Type

User Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

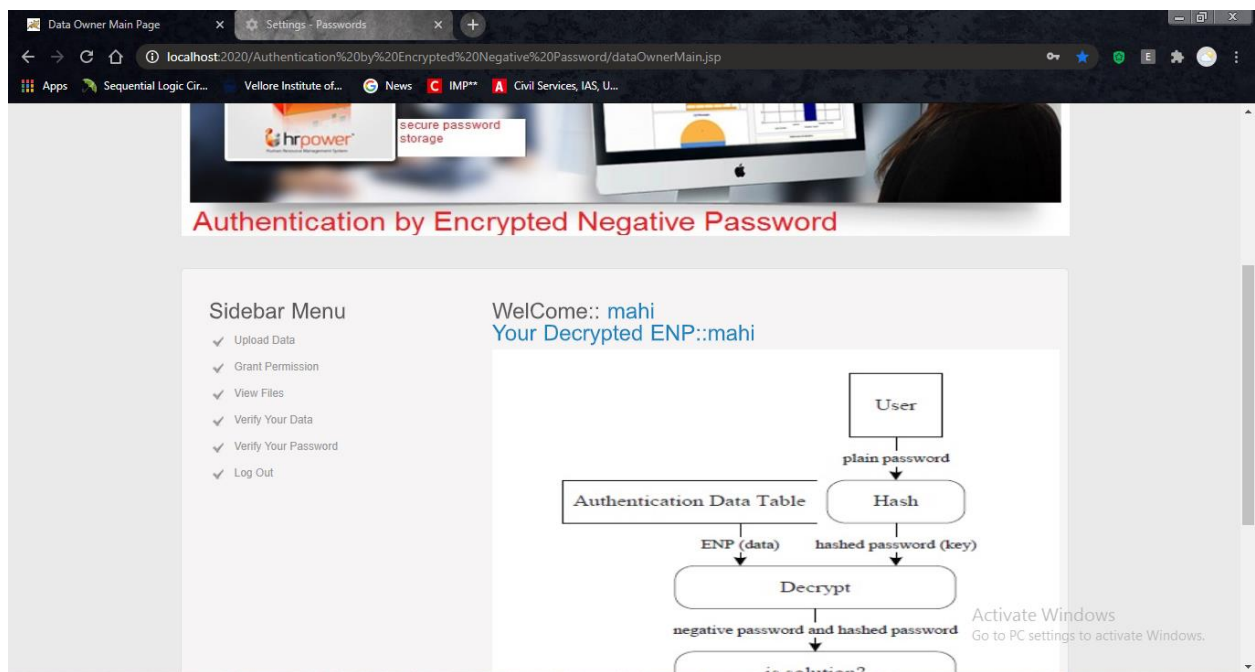
Your Address

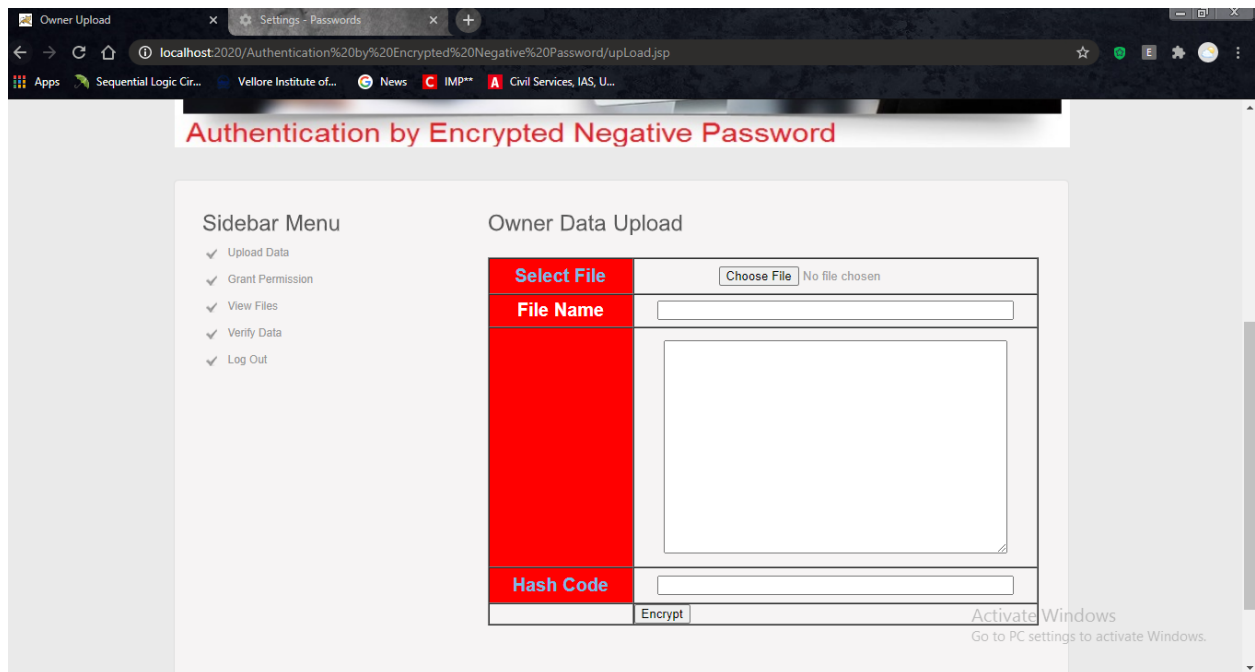
Date of Birth (required)

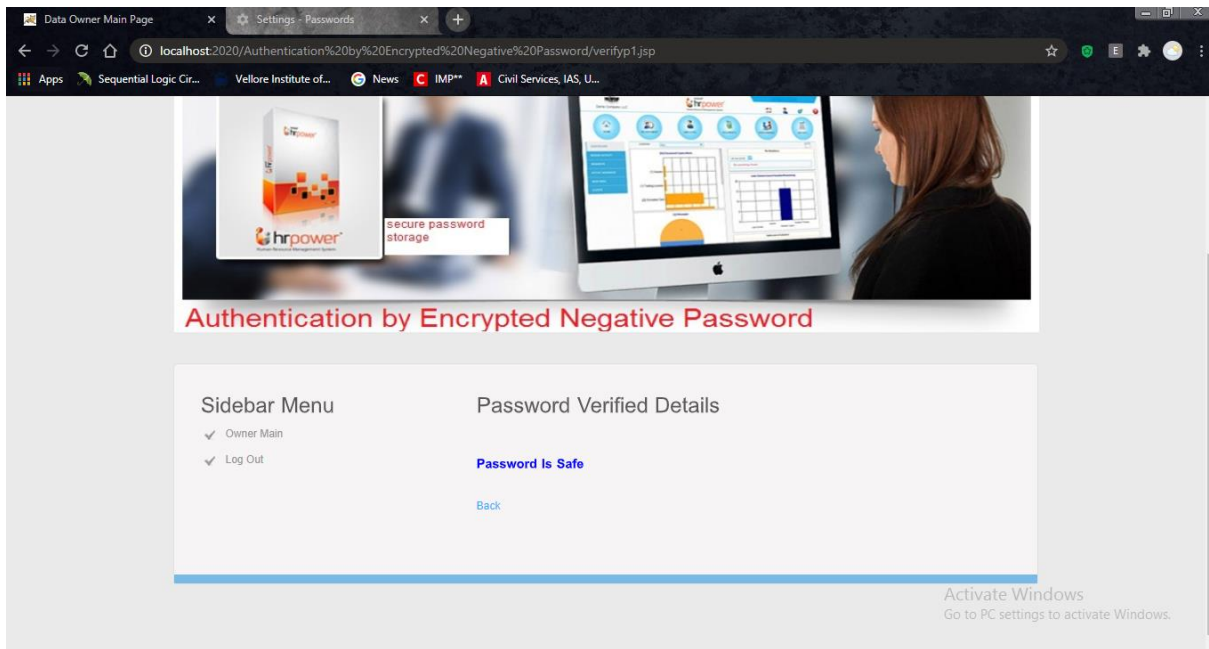
Select Gender (required)

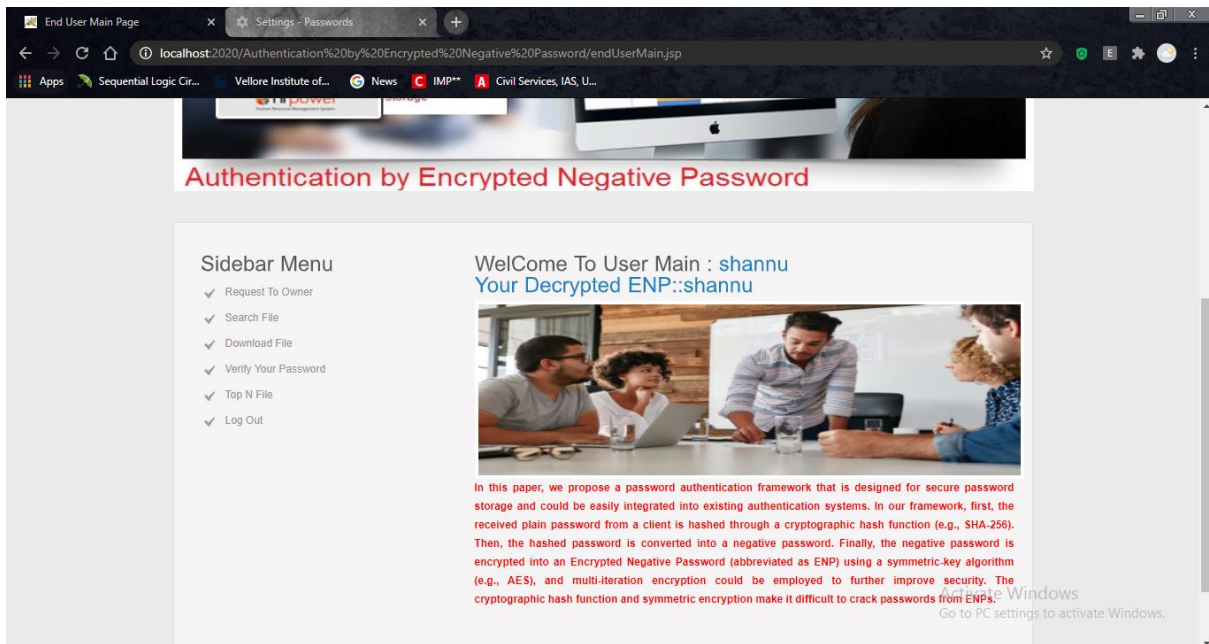
Enter Pincode (required)

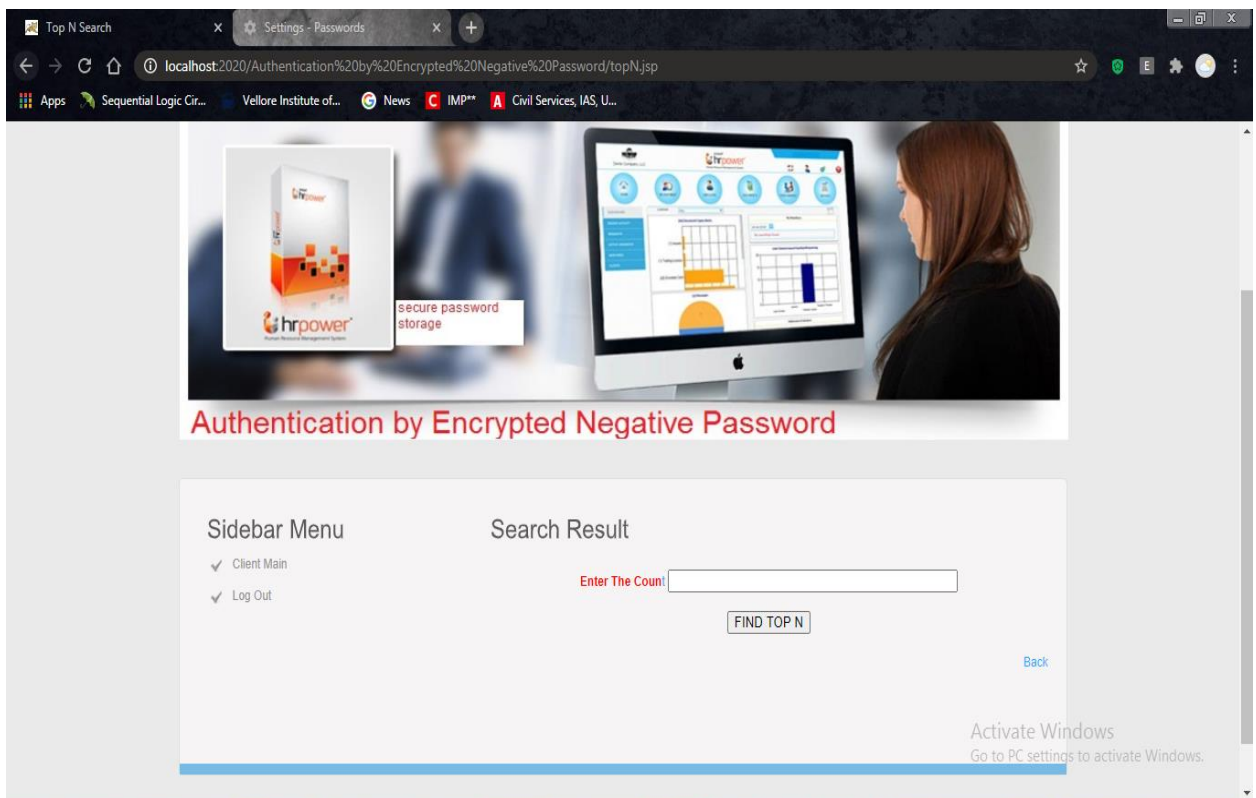
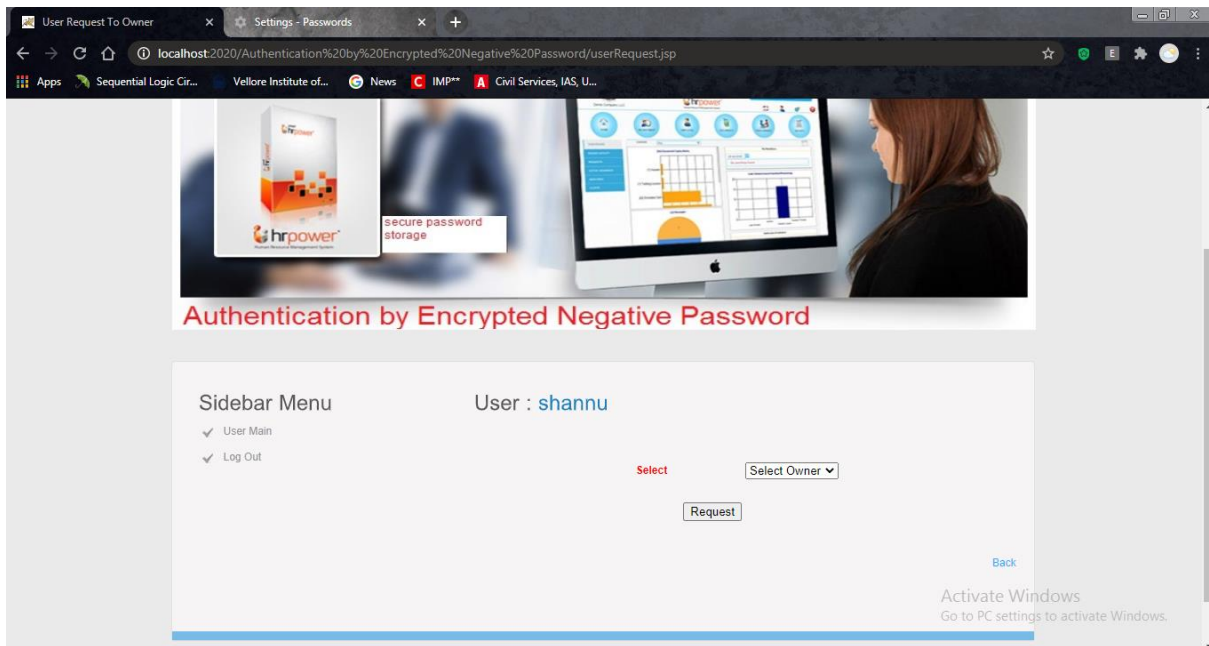
Activate Windows
Go to PC settings to activate Windows.











Web Server Main Page

localhost:2020/Authentication%20by%20Encrypted%20Negative%20Password/webServerMain.jsp

Authentication by Encrypted Negative Password

Sidebar Menu

- ✓ View All Users
- ✓ View All Owner Files
- ✓ Give Privileges To Users
- ✓ View Transactions
- ✓ View All Encrypted Negative Password
- ✓ View All Attacker
- ✓ View All Password Attackers
- ✓ View File Score Results
- ✓ Log Out

WelCome To server Main...!



In this paper, we propose a password authentication framework that is designed for secure password storage and could be easily integrated into existing authentication systems. In our framework, first, the received plain password from a client is hashed through a cryptographic hash function (e.g., SHA-256). Then, the hashed password is converted into a negative password. Finally, the negative password is encrypted into an Encrypted Negative Password (abbreviated as ENP) using a symmetric-key algorithm (e.g., AES), and multi-iteration encryption could be employed to further improve security. The cryptographic hash function and symmetric encryption make it difficult to crack passwords from ENPs.

Activate Windows
Go to PC settings to activate Windows.

View All Users





localhost:2020/Authentication%20by%20Encrypted%20Negative%20Password/viewAllEndUsers.jsp?ab

Authentication by Encrypted Negative Password

Sidebar Menu

- ✓ Server Main
- ✓ Log Out

All Clients...

ID	UserImage	Username	Password	Password's Hash Code	Address	Gender	Status	User Type
6		Kumar	QDY3ODE2OHF3	-21b4494a446726145208 daa6a000e2356cbf92ef	#7827,4th Cross,Rajajinagar	Male	Authorized	Owner
7		Ramesh	UmFIZXNo	-67be3b21e2ecfb18cc3a a76e60bf3986ee76bdf9	#7827,4th Cross,Vijayanagar	Male	Authorized	User
8		Umesh	VW1lc2g=	274d64548be795d51f926 dacbb4e806210f2a10c	#782,4th Cross,Rajajinagar	Male	Authorized	Owner
9		Manjunath	dGVZdA==	-60cf2645a5b13c791222 1533d808b10bedf7a2	#8827,4th Cross,Rajajinagar	Male	Authorized	Owner

Activate Windows
Go to PC settings to activate Windows.

Authentication by Encrypted Negative Password

Sidebar Menu

- ✓ Server Main
- ✓ Log Out

All s...

SN	Owner Name	File Name	Rank	Date & Time	URL and Hash Code
3	Umesh	Connect.jsp	6	05/05/2020 15:21:02	Details
4	Umesh	KeyGen.java	2	05/05/2020 15:26:02	Details
5	Umesh	SQL.txt	0	05/05/2020 15:26:33	Details
6	Manjunath	Android.txt	2	15/05/2020 16:07:27	Details
7	Manjunath	Dotnet.txt	0	15/05/2020 16:07:41	Details
8	Manjunath	Java.txt	3	15/05/2020 16:07:52	Details
9	Manjunath	Silverlight.txt	0	15/05/2020 16:08:06	Details

Activate Windows
Go to PC settings to activate Windows.

User Permission

localhost:2020/Authentication%20by%20Encrypted%20Negative%20Password/viewAllTransaction.jsp

Sidebar Menu

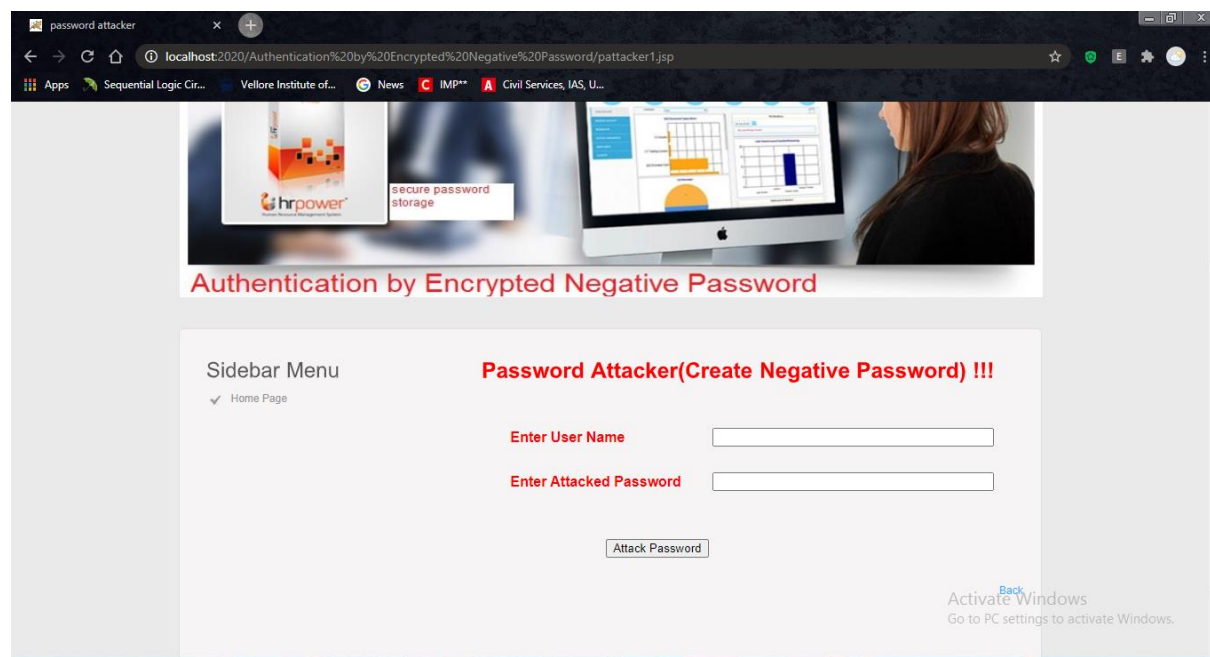
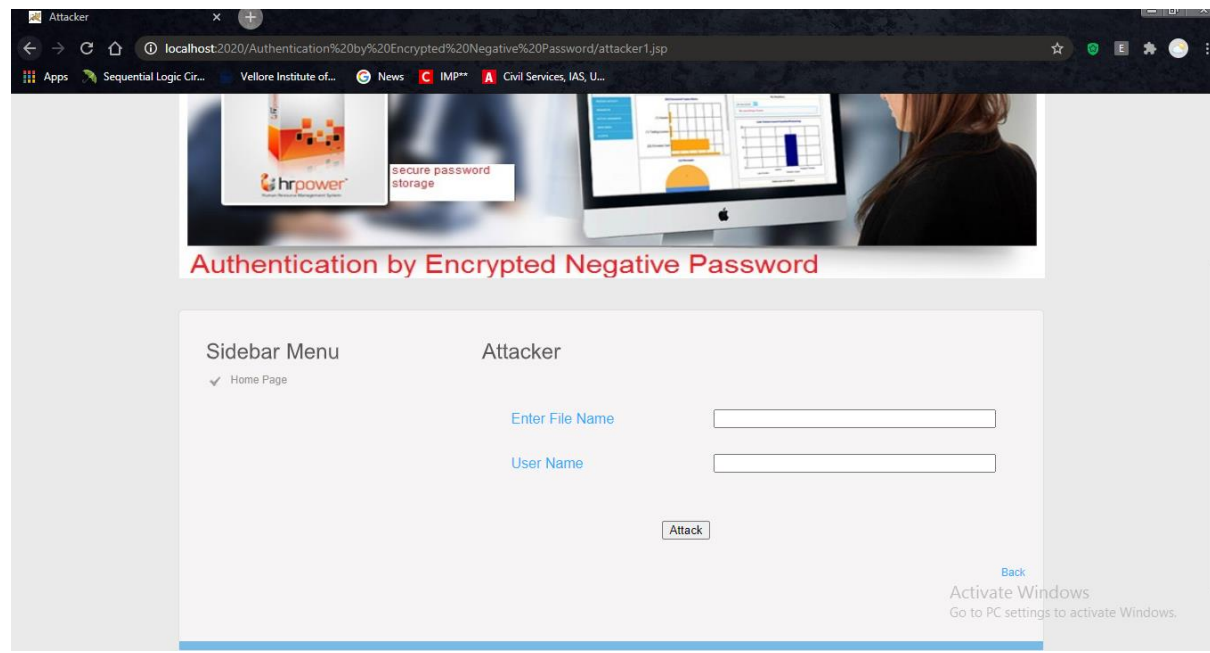
✓ Server Main

✓ Log Out

All Clients And Transactions...

SI NO	User Name	File Name	Secret Key	Operation	Date & Time
4	Umesh	Connect.jsp	[B@1f4fedf	Upload	7/05/2020 15:21:02
5	Ramesh	Connect.jsp	[B@1f4fedf	Download	8/05/2020 15:24:39
6	Umesh	KeyGen.java	[B@27aaf0	Upload	8/05/2020 15:26:02
7	Umesh	SQL.txt	[B@5d9072	Upload	8/05/2020 15:26:33
8	Manjunath	Android.txt	[B@1d0eeefb	Upload	9/05/2020 16:07:27
9	Manjunath	Dotnet.txt	[B@84322	Upload	9/05/2020 16:07:41
10	Manjunath	Java.txt	[B@1b0cc8c	Upload	10/05/2020 16:07:52
11	Manjunath	Silverlight.txt	[B@9955ab	Upload	10/05/2020 16:08:06
12	tmksmanju	Connect.jsp	[B@1f4fedf	Download	11/05/2020 16:10:47
13	tmksmanju	Java.txt	[B@1b0cc8c	Download	11/05/2020 16:12:16
14	tmksmanju	Android.txt	[B@1d0eeefb	Download	12/05/2020 16:12:02

Activate Windows
Go to PC settings to activate Windows.



CHAPTER 7

CONCLUSION & FUTURE WORK

CONCLUSION

In this paper, we proposed a password protection scheme called ENP, and presented a password authentication framework based on the ENP. In our framework, the entries in the authentication data table are ENPs. In the end, we analyzed and compared the attack complexity of hashed password, salted password, key stretching and the ENP. The results show that the ENP could resist lookup table attack and provide stronger password protection under dictionary attack. It is worth mentioning that the ENP does not need extra elements (e.g., salt) while resisting lookup table attack. In the future, other NDB generation algorithms will be studied and introduced to the ENP to further improve password security.

Future Work

Furthermore, other techniques, such as multi-factor authentication and challenge-response authentication, will be introduced into our password authentication framework.

CHAPTER-8

REFERENCES

- [1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, —Passwords and the evolution of imperfect authentication,|| Communications of the ACM, vol. 58, no. 7, pp. 78–87, Jun. 2015.
- [2] M. A. S. Gokhale and V. S. Waghmare, —The shoulder surfing resistant graphical password authentication technique,|| Procedia Computer Science, vol. 79, pp. 490–498, 2016.
- [3] J. Ma, W. Yang, M. Luo, and N. Li, —A study of probabilistic password models,|| in Proceedings of 2014 IEEE Symposium on Security and Privacy, May 2014, pp. 689–704.
- [4] A. Adams and M. A. Sasse, —Users are not the enemy,|| Communications of the ACM, vol. 42, no. 12, pp. 40–46, Dec. 1999.
- [5] E. H. Spafford, —Opus: Preventing weak password choices,|| Computers & Security, vol. 11, no. 3, pp. 273–278, 1992.
- [6] Y. Li, H. Wang, and K. Sun, —Personal information in passwords and its security implications,|| IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2320–2333, Oct. 2017.

- [7] D. Florencio and C. Herley, —A large-scale study of web password habits,‡ in Proceedings of the 16th International Conference on World Wide Web. ACM, 2007, pp. 657–666.
- [8] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, —Designing password policies for strength and usability,‡ ACM Transactions on Information and System Security, vol. 18, no. 4, pp. 13:1–13:34, May 2016.
- [9] D. Wang, D. He, H. Cheng, and P. Wang, —fuzzyPSM: A new password strength meter using fuzzy probabilistic context-free grammars,‡ in Proceedings of 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Jun. 2016, pp. 595–606.
- [10] H. M. Sun, Y. H. Chen, and Y. H. Lin, —oPass: A user authentication protocol resistant to password stealing and password reuse attacks,‡ IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 651–663, Apr. 2012.
- [11] M. Zviran and W. J. Haga, —Password security: An empirical study,‡ Journal of Management Information Systems, vol. 15, no. 4, pp. 161–185, 1999.
- [12] P. Andriotis, T. Tryfonas, and G. Oikonomou, —Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method,‡ in Proceedings of Human Aspects of Information Security, Privacy, and Trust. Springer International Publishing, 2014, pp. 115–126.
- [13] D. P. Jablon, —Strong password-only authenticated key exchange,‡ SIGCOMM Computer Communication Review, vol. 26, no. 5, pp. 5–26, Oct. 1996.
- [14] J. Jose, T. T. Tomy, V. Karunakaran, A. K. V, A. Varkey, and N. C. A., —Securing passwords from dictionary attack with character-tree,‡ in Proceedings of 2016 International Conference on Wireless Communications, Signal Processing and Networking, Mar. 2016, pp. 2301–2307.
- [15] A. Arora, A. Nandkumar, and R. Telang, —Does information security attack frequency increase with vulnerability disclosure? an empirical analysis,‡ Information Systems Frontiers, vol. 8, no. 5, pp. 350–362, Dec. 2006.
- [16] R. Song, —Advanced smart card based password authentication protocol,‡ Computer Standards & Interfaces, vol. 32, no. 5, pp. 321–325, 2010.
- [17] M. C. Ah Kioon, Z. S. Wang, and S. Deb Das, —Security analysis of MD5 algorithm in password storage,‡ in Proceedings of Instruments, Measurement, Electronics and Information Engineering. Trans Tech Publications, Oct. 2013, pp. 2706–2711.
- [18] P. Oechslin, —Making a faster cryptanalytic time-memory trade-off,‡ in Proceedings of Advances in Cryptology - CRYPTO 2003. Springer Berlin Heidelberg, 2003, pp. 617–630.
- [19] S. Noel, M. Elder, S. Jajodia, P. Kalapa, S. O’Hare, and K. Prole, —Advances in topological vulnerability analysis,‡ in Proceedings of 2009 Cybersecurity Applications

Technology Conference for Homeland Security, Mar. 2009, pp. 124–129.

[20] —Hashcat,|| <https://hashcat.net/hashcat/>.

[21] —RainbowCrack,|| <http://project-rainbowcrack.com/>.

[22] —John the Ripper,|| <http://www.openwall.com/john/>.

[23] N. Provos and D. Mazières, —A future-adaptive password scheme,|| in Proceedings of the Annual Conference on USENIX Annual Technical Conference. USENIX Association, 1999, pp. 32–32.

[24] —RFC 7914: The scrypt Password-Based Key Derivation Function,|| <https://tools.ietf.org/html/rfc7914>.