**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT** | **III** |
|  | **LIST OF FIGURES** | **IV** |
|  | **LIST OF SCREENSHOTS** | **V** |
| **1.** | **INTRODUCTION** | **1** |
| **2.** | **LITERATURE SURVEY** | **2-3** |
|  | 2.1. Existing System | 2 |
|  | 2.2. Proposed System | 2 |
|  | 2.3. Advantages of Proposed System | 3 |
|  | 2.4. Disadvantages of Proposed System | 3 |
| **3.** | **ANALYSIS** | **4-8** |
|  | 3.1. Requirement Specification | 4 |
|  | 3.2. Software Requirements | 4 |
|  | 3.3. Hardware Requirements | 4 |
|  | 3.4. Functional Requirements | 5 |
|  | 3.5. Non-Functional Requirements | 5 |
|  | 3.6. Performance Requirements | 6 |
|  | 3.7. Feasibility Study | 6 |
|  | 3.7.1. Technical Feasibility | 7 |
|  | 3.7.2. Operational Feasibility | 7 |
|  | 3.7.3. Economic Feasibility | 8 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **4.** | **DESIGN** | **9-19** |
|  | 4.1. System Architecture | 9 |
|  | 4.2. Data Flow Diagrams | 10 |
|  | 4.3. Uml Diagrams | 11 |
|  | 4.3.1. Use Case Diagram | 13 |
|  | 4.3.2. Class Diagram | 15 |
|  | 4.3.3. Sequence Diagram | 16 |
|  | 4.3.4. Activity Diagram | 17 |
|  | 4.3.5. Component Diagram | 18 |
|  | 4.3.6. Deployment Diagram | 19 |
| **5.** | **IMPLEMENTATION** | **20-33** |
|  | 5.1. Technologies Used | 20 |
|  | 5.2. Sample Code | 22 |
| **6.** | **TESTING AND VALIDATION** | **34-36** |
|  | 6.1. System Testing | 34 |
|  | 6.2. Module Testing | 34 |
|  | 6.3. Integration Testing | 34 |
|  | 6.4. Acceptance Testing | 35 |
|  | 6.5. Test Cases | 35 |
| **7.** | **RESULTS** | **37-43** |
| **8.** | **CONCLUSION AND FUTURE SCOPE** | **44** |
| **9.** | **REFERENCE** | **45-46** |

**ABSTRACT**

The proposed model is liable to meet the required security needs of data center of cloud. AES, DES, and RSA are used for the encryption of file slices, which takes minimum time and has maximum throughput for encryption and decryption from other symmetric algorithms. The idea of splitting and merging adds on to meet the principle of data security. The hybrid approach when deployed in cloud environment makes the remote server more secure and thus, helps the cloud providers to fetch more trust of their users. For data security and privacy protection issues, the fundamental challenge of separation of sensitive data and access control is fulfilled. Cryptography technique translates original data into unreadable form. Cryptography technique is divided into symmetric key cryptography and public key cryptography. This technique uses keys for translating data into unreadable form. So only authorized people can access data from cloud server. Cipher text data is visible for all people.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S. No** | **PARTICULARS** | **PAGE No.** |
|  |  |  |
| 4.1 | Architecture Diagram | 10 |
| 4.2 | Data Flow Diagram | 11 |
| 4.3.1 | Use Case Diagram | 13-14 |
| 4.3.2 | Class Diagram | 15 |
| 4.3.3 | Sequence Diagram | 16 |
| 4.3.4 | Activity Diagram | 17 |
| 4.3.5 | Component Diagram | 18 |
| 4.3.6 | Deployment Diagram | 19 |
| 5.1 | Structure of compilation | 21 |

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **S No.** | **PARTICULAR** | **PAGE NO.** |
|  |  |  |
| 7.1 | Home Page | 37 |
| 7.2 | Owner Login | 37 |
| 7.3 | Owner Registration | 38 |
| 7.4 | Owner Home | 38 |
| 7.5 | File Upload | 39 |
| 7.6 | View File | 39 |
| 7.7 | View Split Data | 40 |
| 7.8 | View Request | 40 |
| 7.9 | User Registration | 41 |
| 7.10 | User Login | 41 |
| 7.11 | User Home | 42 |
| 7.12 | View File | 42 |
| 7.13 | User Verify | 43 |

**1. INTRODUCTION**

Distributed computing has its roots in the early days of extensive-scale distributed computing technology. The National Institute of Standards and Technology (NIST) defines Cloud computing as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." In Cloud computing, both data and software are not entirely confined to the user's local computer. Instead, they reside on the premises of a cloud service provider. This arrangement raises concerns about the security of user data because the user's applications and programs are hosted on the provider's infrastructure.

One way to address these security concerns is through encryption. The cloud provider can implement encryption algorithms to encode the user's files and data, ensuring that even if unauthorized access occurs, the information remains confidential and protected. Encryption plays a pivotal role in securing sensitive data in the cloud, as it transforms information into unreadable gibberish without the proper decryption keys.

**2. LITERATURE SURVEY**

This paper shows a document security model to give a productive answer for the fundamental issue of security in cloud conditions. In this model, half breed encryption is utilized where records are scrambled by blowfish combined with document part and SRNN (modified RSA) is utilized for the secured correspondence amongst clients and the servers. A. Information Security Issues Due to receptiveness and multi-occupant qualities of the cloud, the customary security systems are never again appropriate for applications and information in cloud. A portion of the issues are as following: • Due to dynamic versatility, administration and area straightforwardness highlights of distributed computing model, a wide range of use and information of the cloud stage have no settled framework and security limits. In case of security break, it is hard to segregate a specific asset that has a risk or has been traded off According to benefit conveyance models of Cloud processing, assets and cloud administrations might be possessed by numerous suppliers. As there is an irreconcilable situation, it is hard to convey a brought together safety effort. • Due to the receptiveness of cloud and sharing virtualized assets by multitenant, client information might be gotten to by other unapproved clients.

**2.1 EXISTING SYSTEM:**

* The existing system is a manual process, which means that it is done by humans and not by computers. This makes it a time-consuming process, as humans need to manually enter data and perform tasks.
* The existing system only uses one encryption algorithm. This means that the data is encrypted using the same method every time. This makes it easier for attackers to crack the encryption, as they only need to find one way to break the algorithm.
* The keys that are provided with the existing system are not safe. This means that they are easy to guess or crack. This makes it easy for attackers to gain access to the data, even if it is encrypted.

**2.2 PROPOSED SYSTEM:**

* The stored image file is encrypted using three different algorithms: AES, DES, and RC6.
* This multi-algorithm approach makes it much more difficult for attackers to crack the encryption.
* Even if an attacker were to crack one of the algorithms, they would still need to crack the other two in order to access the file.
* This makes the stored image file much more secure than if it were encrypted using only one algorithm.
* As a result, users can be confident that their image files are safe and secure.

**2.3 ADVANTAGES OF PROPOSED SYSTEM:**

* When this technique is used to encrypt a file, three keys are generated to access the file. All three keys must be entered to access the file, which makes it difficult for hackers to hack the data.
* Data is kept secure on the cloud server, which prevents unauthorized access.
* The system is very secure and robust.

**2.4 DISADVANTAGES OF PROPOSED SYSTEM:**

* Requires an active internet connection to connect with cloud server.

**3.ANALYSIS**

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), is the process of creating or altering systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies).

#### **3.1 Requirements Specification:**

#### Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a server which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operating system. These software and hardware requirements provide compatible support to the operation system in developing an application.

**3.2 Hardware Requirements:**

#### The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

#### System : Pentium IV 2.4 GHz.

#### Hard Disk : 100 GB.

#### RAM : 1 GB.

**3.3 Software Requirements:**

#### The software requirements specify the use of all required software products like data management systems. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

#### Operating system : Windows XP/7/10

* Coding Language : Html, JavaScript, Java/J2EE (Jsp Servlet)
* Development Kit : JDK 1.7
* Database : MySQL
* IDE : Netbeans
* Server : Tomcat 7.0

**3.4. Functional Requirements**

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant “functional requirements” in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and the technology implementation of the business processes.

**3.5. Non-Functional Requirements**

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system can change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features:

* Accessibility
* Availability
* Backup
* Certification
* Compliance
* Configuration Management
* Documentation
* Disaster Recovery
* Efficiency (resource consumption for given load)
* Interoperability

**3.6. Performance Requirements**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements must be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system.
* The system should be accurate.
* The system should be better than the existing system.

The existing system is completely dependent on the user to perform all the duties.

**3.7. Feasibility Study**

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

* Technical Feasibility
* Operation Feasibility
* Economic Feasibility

**3.7.1 Technical Feasibility**

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

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

**3.7.2 Operational Feasibility**

**User-friendly**

Customers will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also, the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

**Reliability**

The package wills pick-up current transactions online. Regarding the old transactions, User will enter them into the system.

**Security**

The web server and database server should be protected from hacking, virus etc

**Portability**

The application will be developed using standard open-source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc. This software will work both on Windows and Linux o/s. Hence portability problems will not arise.

**Availability**

This software will be available always.

**Maintainability**

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end, and the 2nd tier is the database, which uses My-SQL, which is the back end.

The front-end can be run on different systems (clients). The database will be running on the server. Users access these forms by using the user-ids and the passwords.

**3.7.3 Economic Feasibility**

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web-based application with separate web server and database server. This is required as the activities are spread throughout the organization and the customer wants a centralized database. Further some of the linked transactions take place in different locations.

Open-source software like TOMCAT, JAVA, My-SQL, and Linux is used to minimize the cost for the Customer.

**4.DESIGN**

**4.1. System Architecture**

## The purpose of the design phase is to arrange an answer to the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It’s a serious impact on the later part, notably testing and maintenance.

## The output of this part is the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

## System Design, conjointly referred to as top-ranking style, aims to spot the modules that ought to be within the system, the specifications of those modules, and the way they move with one another to supply the specified results.

## At the top of the system style all the main knowledge structures, file formats, output formats, and the major modules within the system and their specifications square measure set. System design is the method or art of processing the design, components, modules, interfaces, and knowledge for a system to satisfy such needs. Users will read it because of the application of systems theory to development.

## Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure is sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

## In system design the main target is on distinguishing the modules, whereas through careful style the main target is on planning the logic for every of the modules.

## 

## Figure 4.1 Architecture Diagram

**4.2. DATA FLOW DIAGRAMS**

## Data Flow Diagram can also be termed as bubble chart. It is a pictorial or graphical form, which can be applied to represent the input data to a system and multiple functions carried out on the data and the generated output by the system.

## A graphical tool accustomed describe and analyze the instant of knowledge through a system manual or automatic together with the method, stores of knowledge, and delays within the system. The transformation of knowledge from input to output, through processes, is also delineate logically and severally of the physical elements related to the system. The DFD is also known as a data flow graph or a bubble chart. The Basic Notation used to create a DFD’s are as follows:

## Dataflow:

## 

## 

## Process:

## 

## Source:

## Rhombus:

## 

## **4.3.UML DIAGRAM:**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspectives. Each view is defined by a set of diagrams, which is as follows.

**User Model View**

This view represents the system from the user’s perspective. The analysis representation describes a usage scenario from the end-user’s perspective.

**Structural Model view**

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

**Behavioral Model View**

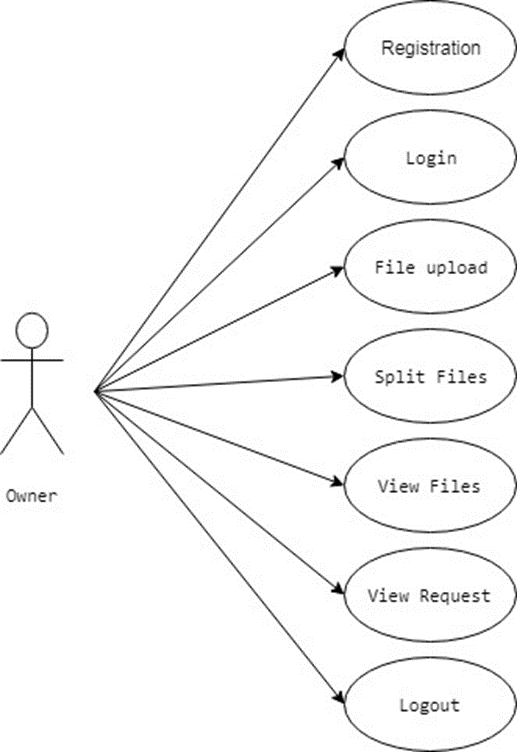
It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

**4.3.1. USE CASE DIAGRAM:**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



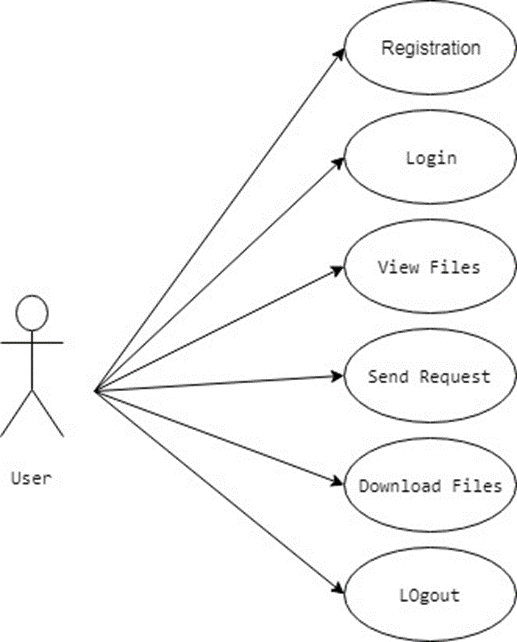


Figure 4.3.1 Use Case Diagram

**4.3.2. CLASS DIAGRAM:**

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. A class with three sections, in the diagram, classes is represented with boxes which contain three parts:

*The upper part holds the name of the class.*

*The middle part contains the attributes of the class.*

The bottom part gives the methods or operations the class can take or undertake.

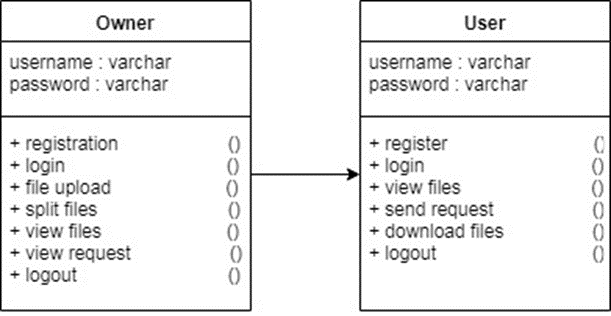
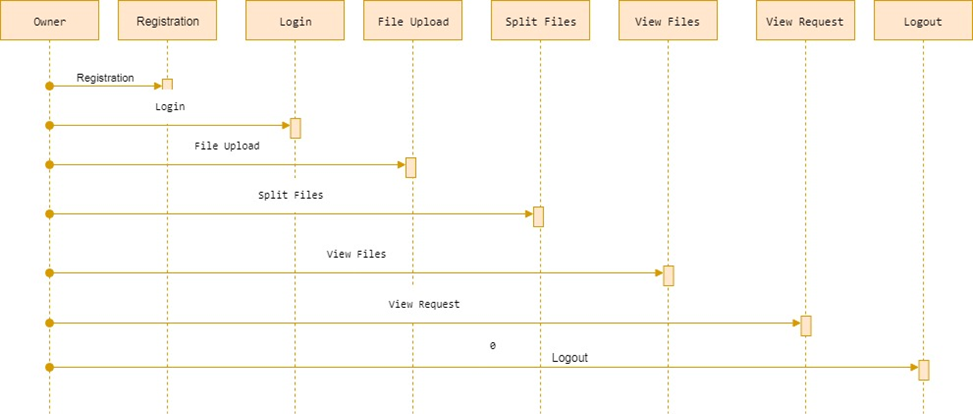


Figure 4.3.2: Class Diagram

**4.3.3. SEQUENCEDIAGRAM:**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



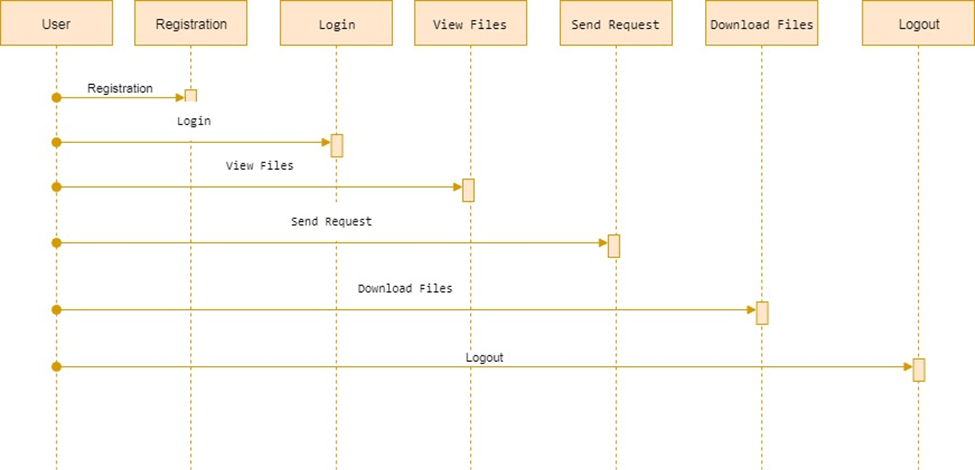


Figure 4.3.3: Sequence diagram

**4.3.4.** ACTIVITY DIAGRAM**:**

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

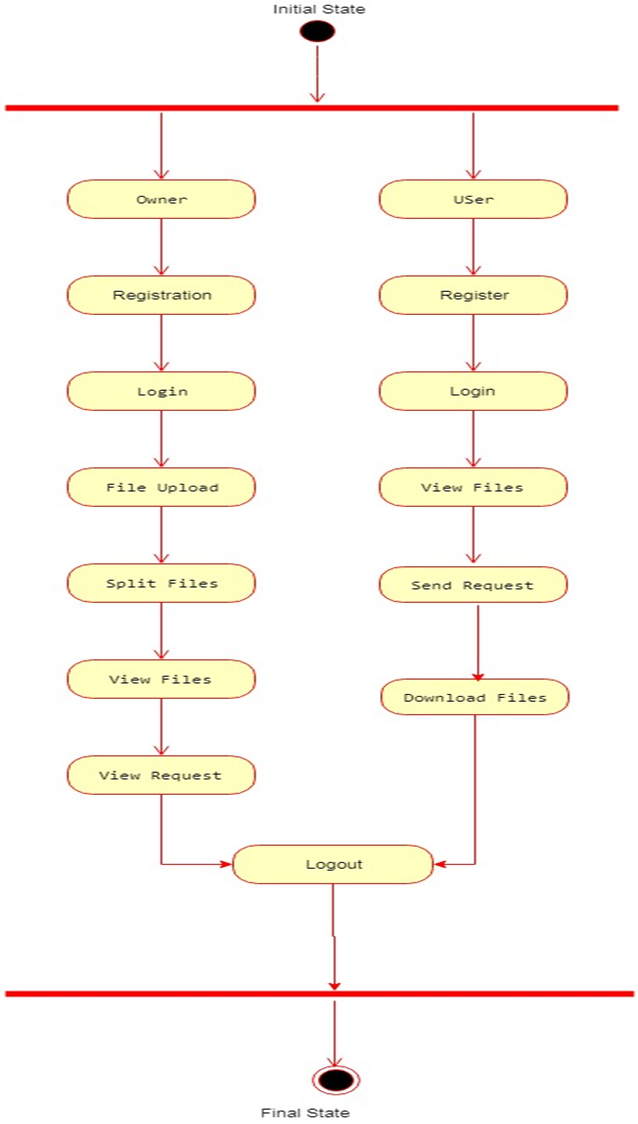
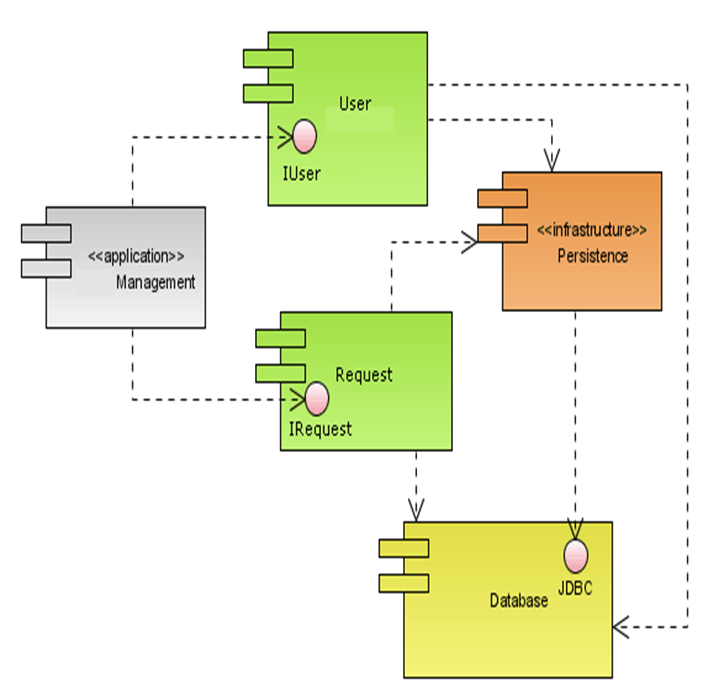
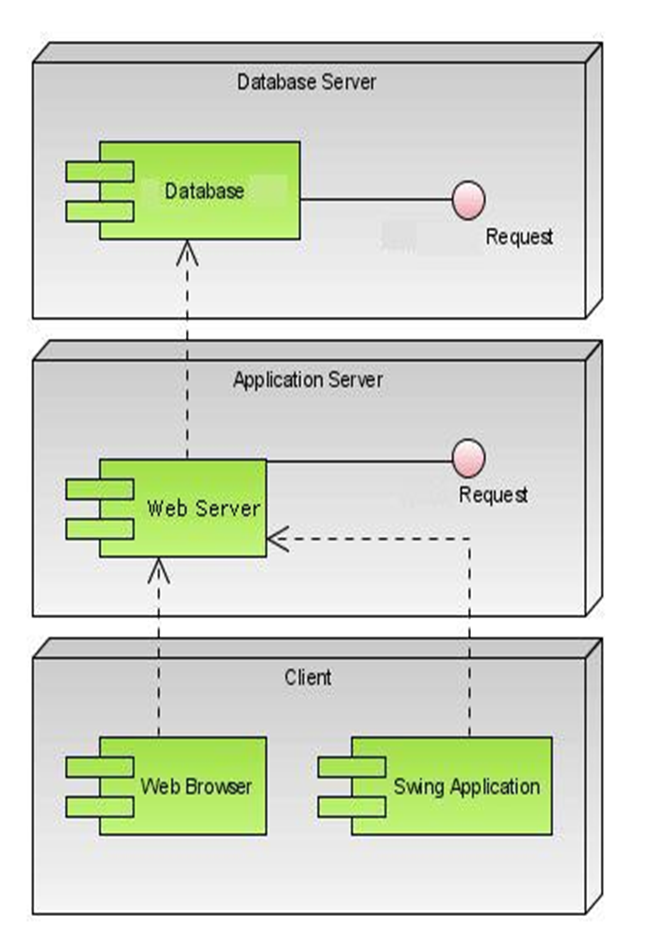


Figure 4.3.4: Activity Diagram

**4.3.5.** COMPONENT DIAGRAM:



**4.3.6. DEPLOYMENT DIAGRAM:**



**5. IMPLEMENTATION:**

**5.1 Technologies Used**

Initially Java language was called as “oak”, but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent i.e., architecture neutral language that could be used to create software to be embedded in various consumer electronic devices.

**Applications and applets**

An application is a program that runs on our computer under the operating system of that computer. It is like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet I San application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I am tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

**Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

**Compilation of code**

When we compile the code, the Java compiler creates machine code called byte code for a hypothetical machine called Java Virtual Machine (JVM). Compiling and interpreting java source code.

****

**Figure 5.1: Structure of compilation**

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. This could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

**Simple**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer, learning Java will orient features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

**Object oriented**

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

**Robust**

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks the code at compile time and runtime.

**5.2. SAMPLE CODE:**

**Database Code:**

/\*

SQL Yog Community Edition- MySQL GUI v7.15

MySQL - 5.5.29: Database - file\_split

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/ /\*!40101 SET NAMES utf8 \*/;

/\*!40101 SET SQL\_MODE=''\*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

CREATE DATABASE /\*!32312 IF NOT EXISTS\*/`file\_split` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

USE `file\_split`;

/\*Table structure for table `cloudadata` \*/

DROP TABLE IF EXISTS `cloudadata`;

CREATE TABLE `cloudadata` (

`filename` varchar (100) DEFAULT NULL,

`owner` varchar (100) DEFAULT NULL,

`f1` text,

`skey` varchar (100) DEFAULT NULL,

`f2` text,

`skey1` varchar (100) DEFAULT NULL,

`f3` varchar (100) DEFAULT NULL,

`skey2` varchar(100) DEFAULT NULL,

`data` text

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Data for the table `cloudadata` \*/

insert into `cloudadata`(`filename`,`owner`,`f1`,`skey`,`f2`,`skey1`,`f3`,`skey2`,`data`) values ('shiva.txt','shiva','llvbOYe158yhYcYNii7d3g==','AU8TXvdcD8QXEfUjS7mTSw==','0hx2yl1M0TI=','ZwSdZONRRTE=','B4+kpXjYIdyRD+d2isWjaQ==','Fs+P1IxMSvT/f0mXgqKRQA==','hi hello how ru ');

/\*Table structure for table `file` \*/

DROP TABLE IF EXISTS `file`;

CREATE TABLE `file` (

`id` int(100) NOT NULL AUTO\_INCREMENT,

`file` text,

`filename` varchar(50) DEFAULT NULL,

`CDate` varchar(20) DEFAULT NULL,

`data` text,

`owner` varchar(40) DEFAULT NULL,

`pk` varchar(40) DEFAULT NULL,

`mk` varchar(40) DEFAULT NULL,

`privatekey` varchar(10) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

/\*Data for the table `file` \*/

insert into `file`(`id`,`file`,`filename`,`CDate`,`data`,`owner`,`pk`,`mk`,`privatekey`) values (3,'hi hello how ru ','shiva.txt','2019-03-27','hi hello how ru \n','shiva','no','no',NULL);

/\*Table structure for table `owner` \*/

DROP TABLE IF EXISTS `owner`;

CREATE TABLE `owner` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(100) NOT NULL,

`password` varchar(1000) DEFAULT NULL,

`dob` date DEFAULT NULL,

`email` varchar(100) DEFAULT NULL,

`city` varchar(100) DEFAULT NULL,

`contact` varchar(100) DEFAULT NULL,

PRIMARY KEY (`id`,`username`),

UNIQUE KEY `id` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1;

/\*Data for the table `owner` \*/

insert into `owner`(`id`,`username`,`password`,`dob`,`email`,`city`,`contact`) values (1,'shiva','shiva','0000-00-00','shiva','sjo','8465'),(2,'aa','a','2019-01-03','shivakrish191@gmail.com','s','7894561200');

/\*Table structure for table `request` \*/

DROP TABLE IF EXISTS `request`;

CREATE TABLE `request` (

`filename` varchar(40) NOT NULL,

`data` text,

`owner` varchar(40) DEFAULT NULL,

`status` varchar(40) DEFAULT NULL,

`email` varchar(40) DEFAULT NULL,

`p1` varchar(40) DEFAULT 'no',

`p2` varchar(40) DEFAULT 'no',

`p3` blob,

`s1` varchar(100) DEFAULT 'no',

`s2` varchar(100) DEFAULT 'no',

`s3` varchar(100) DEFAULT 'no',

PRIMARY KEY (`filename`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Data for the table `request` \*/

insert into `request`(`filename`,`data`,`owner`,`status`,`email`,`p1`,`p2`,`p3`,`s1`,`s2`,`s3`) values ('shiva.txt','hi hello how ru ','shiva','yes','shivakrish191@gmail.com','llvbOYe158yhYcYNii7d3g==','0hx2yl1M0TI=','B4+kpXjYIdyRD+d2isWjaQ==','AU8TXvdcD8QXEfUjS7mTSw==','ZwSdZONRRTE=','Fs+P1IxMSvT/f0mXgqKRQA==');

/\*Table structure for table `user` \*/

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

`id` int (11) NOT NULL AUTO\_INCREMENT,

`username` varchar (100) NOT NULL,

`password` varchar (1000) DEFAULT NULL,

`dob` date DEFAULT NULL,

`email` varchar (100) DEFAULT NULL,

`city` varchar (100) DEFAULT NULL,

`contact` varchar (100) DEFAULT NULL,

PRIMARY KEY (`id`,`username`),

UNIQUE KEY `id` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=latin1;

/\*Data for the table `user` \*/

insert into `user`(`id`,`username`,`password`,`dob`,`email`,`city`,`contact`) values (1,'shiva','shiva','2019-01-10','shivakrish191@gmail.com','n','945449494');

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

**DB Connection Code:**

package databasecon;

import java.sql.Connection;

import java.sql.DriverManager;

public class Dbconnection {

public static Connection getConnection() {

Connection con = null;

try {

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc: mysql://localhost:3306/studentperformance", "root", "root");

} catch (Exception ex) {

ex.printStackTrace();

}

return con;

}

}

**LSB Decode Code:**

package lsb;

import java.awt.image.BufferedImage;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintStream;

import java.io.PrintWriter;

//import java.util.Arrays;

import java.util.Scanner;

import javax.imageio.ImageIO;

public class LSB\_decode {

//static final String STEGIMAGEFILE = "D:\\2.jpg";

//static final String DECODEDMESSAGEFILE = "D:\\message\_dec.txt";

public static String b\_msg="";

public static int len = 0;

public void decode(String STEGIMAGEFILE, String DECODEDMESSAGEFILE) {

try{

BufferedImage yImage=readImageFile(STEGIMAGEFILE);

DecodeTheMessage(yImage);

String msg="";

//System.out.println("len is "+len\*8);

for(int i=0;i<len\*8;i=i+8){

String sub=b\_msg.substring(i,i+8);

int m=Integer.parseInt(sub,2);

char ch=(char) m;

System.out.println("m "+m+" c "+ch);

msg+=ch;

}

PrintWriter out = new PrintWriter(new FileWriter(DECODEDMESSAGEFILE, true), true);

out.write(msg);

out.close();

} catch (Exception e) {

System.out.println(e);

}

}

public static BufferedImage readImageFile(String COVERIMAGEFILE){

BufferedImage theImage = null;

File p = new File (COVERIMAGEFILE);

try{

theImage = ImageIO.read(p);

}catch (IOException e){

e.printStackTrace();

System.exit(1);

}

return theImage;

}

public static void DecodeTheMessage (BufferedImage yImage) throws Exception{

int j=0;

int currentBitEntry=0;

String bx\_msg="";

for (int x = 0; x < yImage.getWidth(); x++){

for ( int y = 0; y < yImage.getHeight(); y++){

if(x==0&&y<8){

//System.out.println("enc "+yImage.getRGB(x, y)+" dec "+yImage.getRGB(x, y)+" "+b\_msg);

int currentPixel = yImage.getRGB(x, y);

int red = currentPixel>>16;

red = red & 255;

int green = currentPixel>>8;

green = green & 255;

int blue = currentPixel;

blue = blue & 255;

String x\_s=Integer.toBinaryString(blue);

bx\_msg+=x\_s.charAt(x\_s.length()-1);

len=Integer.parseInt(bx\_msg,2);

}

else if(currentBitEntry<len\*8){

//System.out.println("enc "+yImage.getRGB(x, y)+" dec "+yImage.getRGB(x, y)+" "+b\_msg);

int currentPixel = yImage.getRGB(x, y);

int red = currentPixel>>16;

red = red & 255;

int green = currentPixel>>8;

green = green & 255;

int blue = currentPixel;

blue = blue & 255;

String x\_s=Integer.toBinaryString(blue);

b\_msg+=x\_s.charAt(x\_s.length()-1);

currentBitEntry++;

//System.out.println("curre "+currentBitEntry);

}

}

}

System.out.println("bin value of msg hided in img is "+b\_msg);

}

}

**Encode Code:**

package lsb;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

//import java.util.Arrays;

import java.util.Scanner;

import javax.imageio.ImageIO;

public class LSB\_encode {

//static final String MESSAGEFILE = "D:\\message.txt";

//static final String COVERIMAGEFILE = "D:\\Desert.jpg";

//static final String STEGIMAGEFILE = "D:\\2.jpg";

public void encode(String MESSAGEFILE, String COVERIMAGEFILE,String STEGIMAGEFILE) {

try{

String contentOfMessageFile = (readMessageFile(MESSAGEFILE));

int[] bits=bit\_Msg(contentOfMessageFile);

System.out.println("msg in file "+contentOfMessageFile);

for(int i=0;i<bits.length;i++)

System.out.print(bits[i]);

System.out.println();

BufferedImage theImage=readImageFile(COVERIMAGEFILE);

hideTheMessage(bits, theImage,STEGIMAGEFILE);

} catch (Exception e) {

System.out.println(e);

}

}

public static String readMessageFile (String MESSAGEFILE) throws FileNotFoundException{

String contentOfMessageFile = "";

File a = new File (MESSAGEFILE);

Scanner scan = new Scanner (a);

while (scan.hasNextLine()){

String next = scan.nextLine();

contentOfMessageFile += next;

if (scan.hasNextLine()){

contentOfMessageFile += "\n";

}

}

scan.close();

return contentOfMessageFile;

}

public static int[] bit\_Msg(String msg){

int j=0;

int[] b\_msg=new int[msg.length()\*8];

for(int i=0;i<msg.length();i++){

int x=msg.charAt(i);

String x\_s=Integer.toBinaryString(x);

while(x\_s.length()!=8){

x\_s='0'+x\_s;

}

System.out.println("dec value for "+x +" is "+x\_s);

for(int i1=0;i1<8;i1++) {

b\_msg[j] = Integer.parseInt(String.valueOf(x\_s.charAt(i1)));

j++;

};

}

return b\_msg;

}

public static BufferedImage readImageFile(String COVERIMAGEFILE){

BufferedImage theImage = null;

File p = new File (COVERIMAGEFILE);

try{

theImage = ImageIO.read(p);

}catch (IOException e){

e.printStackTrace();

System.exit(1);

}

return theImage;

}

public static void hideTheMessage (int[] bits, BufferedImage theImage,String STEGIMAGEFILE) throws Exception{

File f = new File (STEGIMAGEFILE);

BufferedImage sten\_img=null;

int bit\_l=bits.length/8;

int[] bl\_msg=new int[8];

System.out.println("bit lent "+bit\_l);

String bl\_s=Integer.toBinaryString(bit\_l);

while(bl\_s.length()!=8){

bl\_s='0'+bl\_s;

}

for(int i1=0;i1<8;i1++) {

bl\_msg[i1] = Integer.parseInt(String.valueOf(bl\_s.charAt(i1)));

};

int j=0;

int b=0;

int currentBitEntry=8;

for (int x = 0; x < theImage.getWidth(); x++){

for ( int y = 0; y < theImage.getHeight(); y++){

if(x==0&&y<8){

int currentPixel = theImage.getRGB(x, y);

int ori=currentPixel;

int red = currentPixel>>16;

red = red & 255;

int green = currentPixel>>8;

green = green & 255;

int blue = currentPixel;

blue = blue & 255;

String x\_s=Integer.toBinaryString(blue);

String sten\_s=x\_s.substring(0, x\_s.length()-1);

sten\_s=sten\_s+Integer.toString(bl\_msg[b]);

//j++;

int temp=Integer.parseInt(sten\_s,2);

int s\_pixel=Integer.parseInt(sten\_s, 2);

int a=255;

int rgb = (a<<24) | (red<<16) | (green<<8) | s\_pixel;

theImage.setRGB(x, y, rgb);

//System.out.println("original "+ori+" after "+theImage.getRGB(x, y));

ImageIO.write(theImage, "png", f);

b++;

}

else if (currentBitEntry < bits.length+8 ){

int currentPixel = theImage.getRGB(x, y);

int ori=currentPixel;

int red = currentPixel>>16;

red = red & 255;

int green = currentPixel>>8;

green = green & 255;

int blue = currentPixel;

blue = blue & 255;

String x\_s=Integer.toBinaryString(blue);

String sten\_s=x\_s.substring(0, x\_s.length()-1);

sten\_s=sten\_s+Integer.toString(bits[j]);

j++;

int temp=Integer.parseInt(sten\_s,2);

int s\_pixel=Integer.parseInt(sten\_s, 2);

int a=255;

int rgb = (a<<24) | (red<<16) | (green<<8) | s\_pixel;

theImage.setRGB(x, y, rgb);

//System.out.println("original "+ori+" after "+theImage.getRGB(x, y));

ImageIO.write(theImage, "png", f);

currentBitEntry++;

//System.out.println("curre "+currentBitEntry);

}

}

}

}

}

**6.TESTING AND VALIDATION**

## Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property function as a unit. The test data should be chosen such that it passed through all possible conditions. The following is the description of the testing strategies, which were carried out during the testing period.

**6.1 SYSTEM TESTING:**

## Testing has become an integral part of any system or project, especially in the field of information technology. The importance of testing is a method of justifying if one is ready to move further, be it to be check if one is capable to withstand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested to see whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus, the code was exhaustively checked for all possible correct data and the outcomes were also checked.

## **6.2 MODULE TESTING:**

## To locate errors, each module is tested individually. This enables us to detect errors and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus, all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example, the job classification module is tested separately. This module is tested with different jobs and its approximate execution time, and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately, and their corresponding results are obtained which reduces the process waiting time.

## **6.3 INTEGRATION TESTING:**

## After the module testing, the integration testing is applied. When linking the modules there may be a chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus, the mapping of jobs with resources is done correctly by the system.

## **6.4 ACCEPTANCE TESTING**

## When that user finds no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

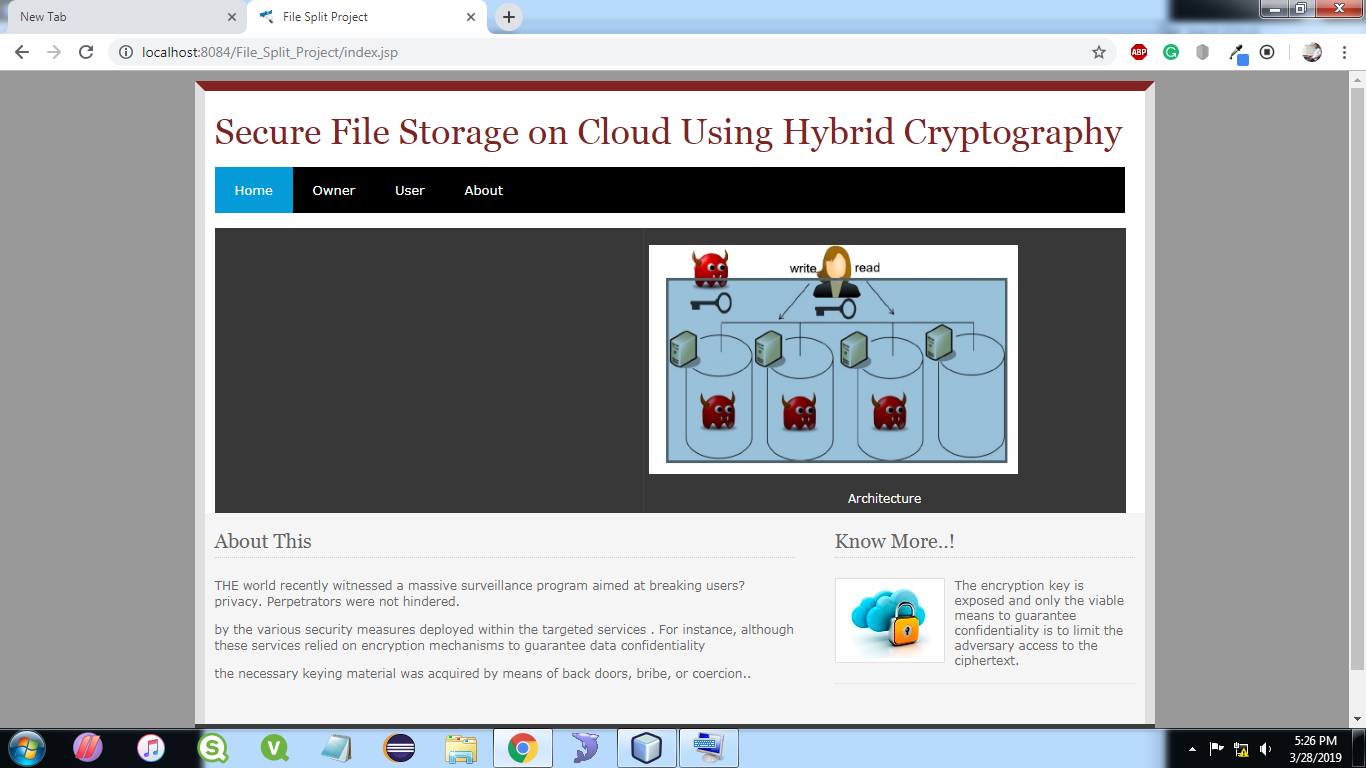
**6.5 TEST CASES:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case Id** | **Test**  **Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload the tasks dataset | Verify either file is loaded or not | If dataset is not uploaded | It cannot display the file loaded message | File is loaded which displays task waiting time | High | High |
| 02 | Upload patient’s dataset | Verify either dataset loaded or not | If dataset is not uploaded | It cannot display dataset reading process completed | It can display dataset reading process completed | low | High |
| 03 | Preprocessing | Whether preprocessing on the dataset applied or not | If not applied | It cannot  display the necessary data for further process | It can display the necessary data for further process | Medium | High |
| 04 | Prediction Random Forest | Whether  Prediction algorithm applied on the data or not | If not applied | Random tree is not generated | Random tree is generated | High | High |
| 05 | Recommendation | Whether predicted data is displayed or not | If not displayed | It cannot view prediction containing patient data | It can view prediction containing patient data | High | High |
| 06 | Noisy Records Chart | Whether the graph is displayed or not | If graph is not displayed | It does not show the variations in between clean and noisy records | It shows the variations in between clean and noisy records | Low | Medium |

TABLE 6.5.1 TESTCASES

**7.RESULTS**

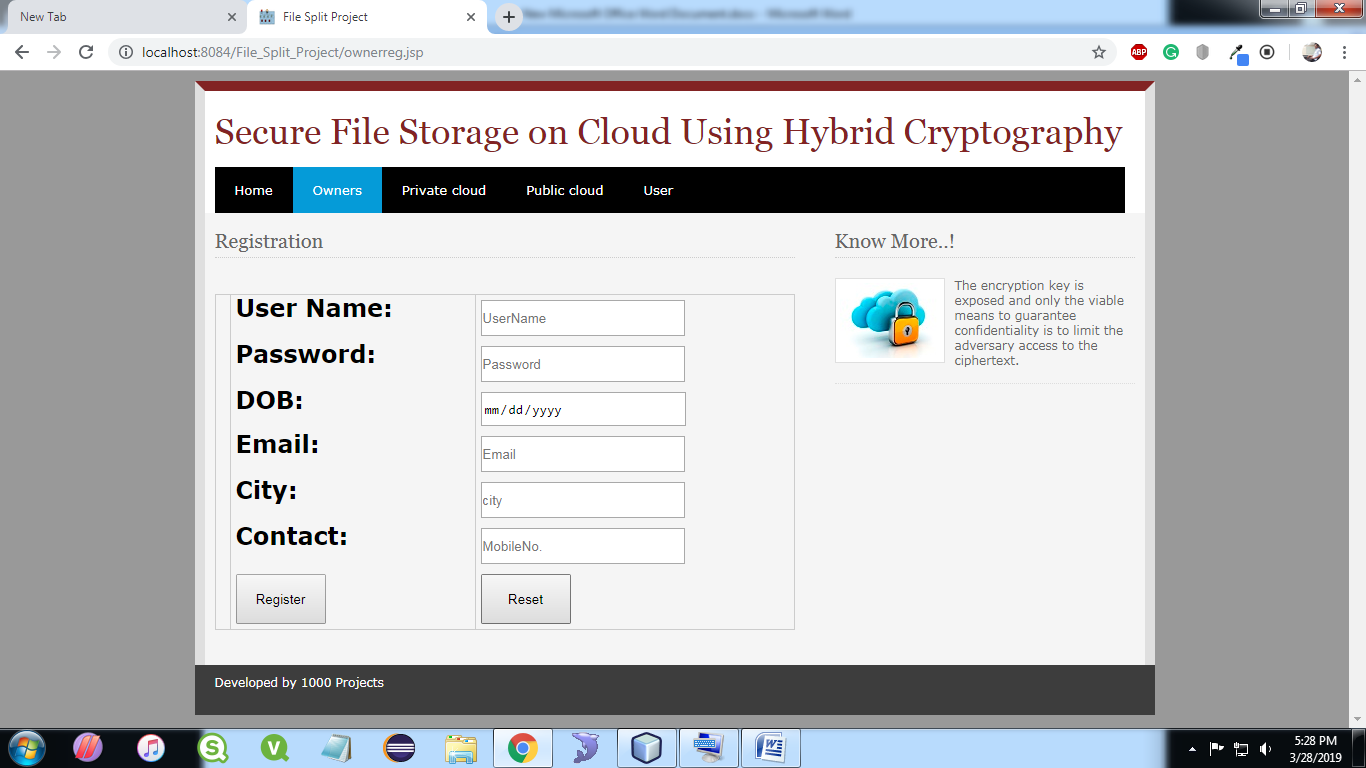
**7.1 HOME**



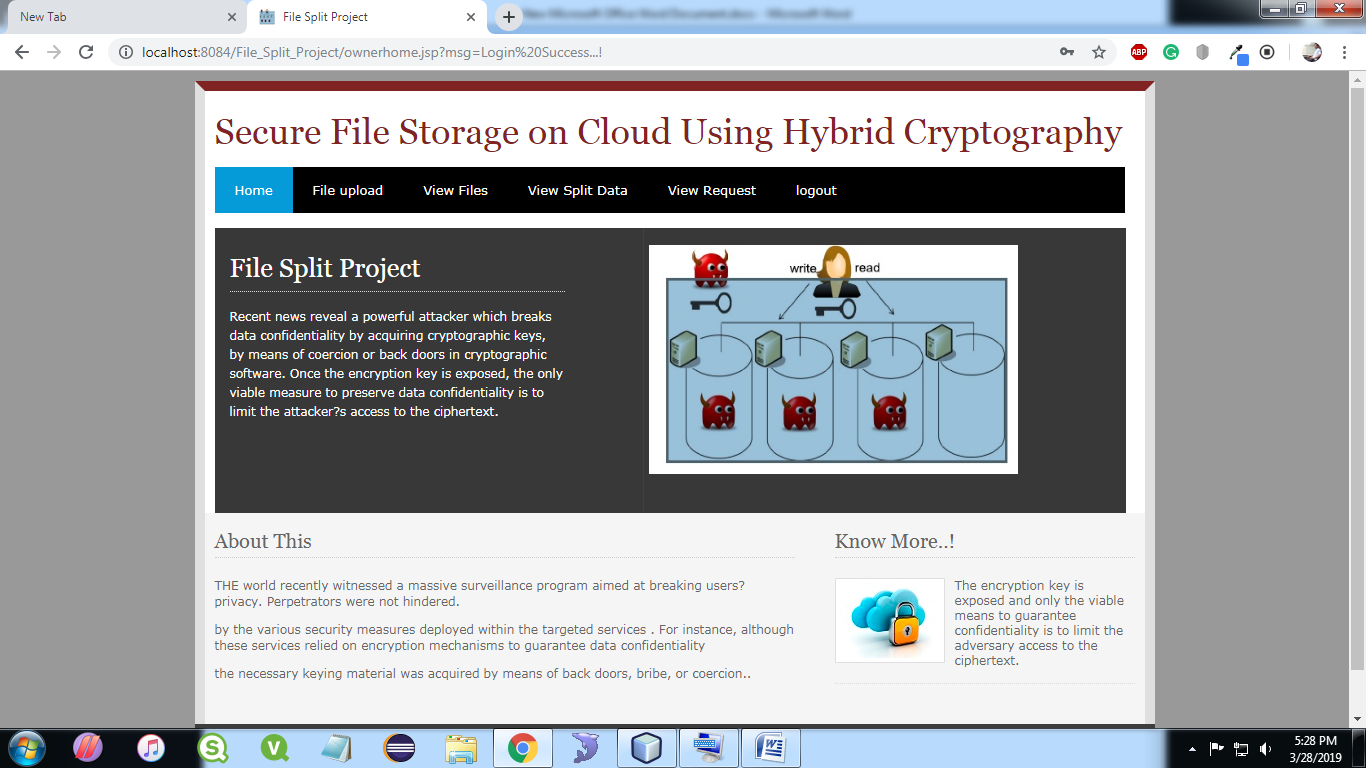
**7.2 OWNER LOGIN**



**7.3 OWNER REGISTRATION**



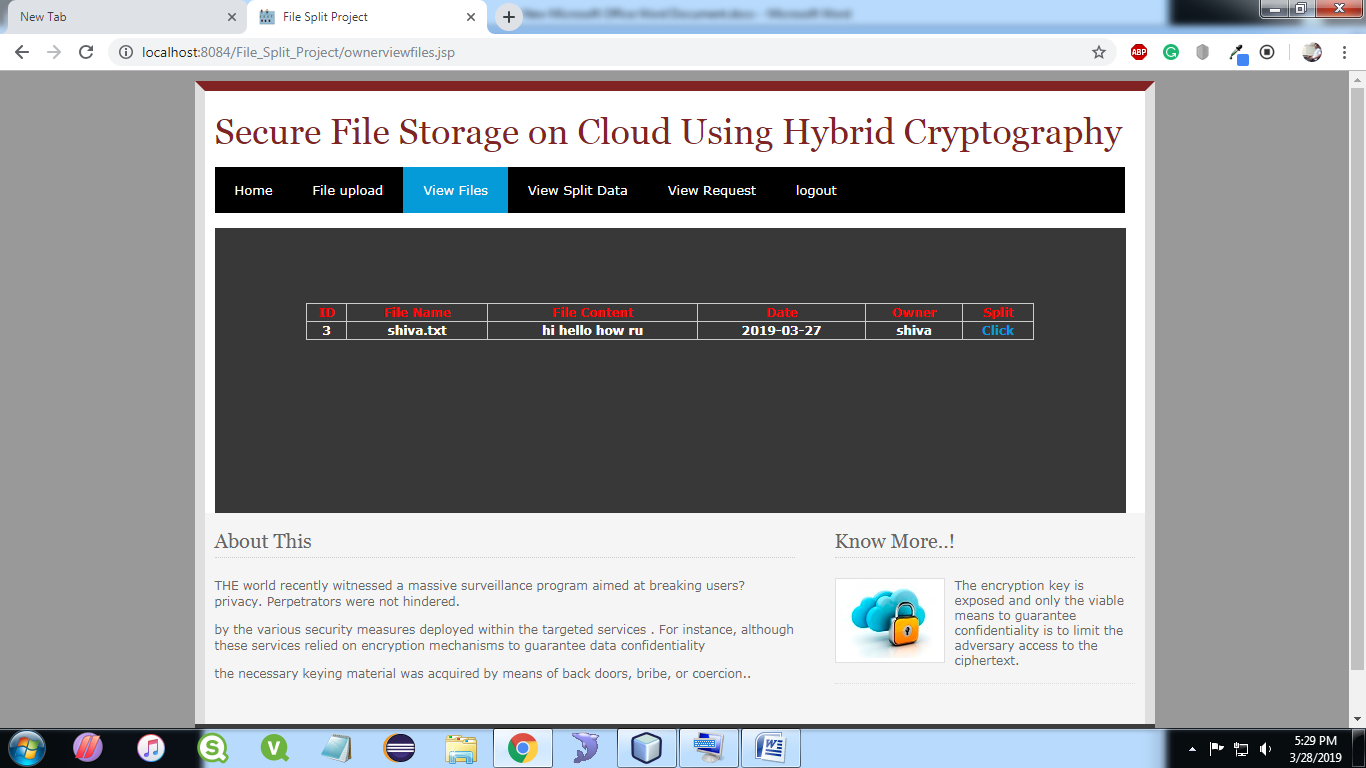
**7.4 OWNER HOME**



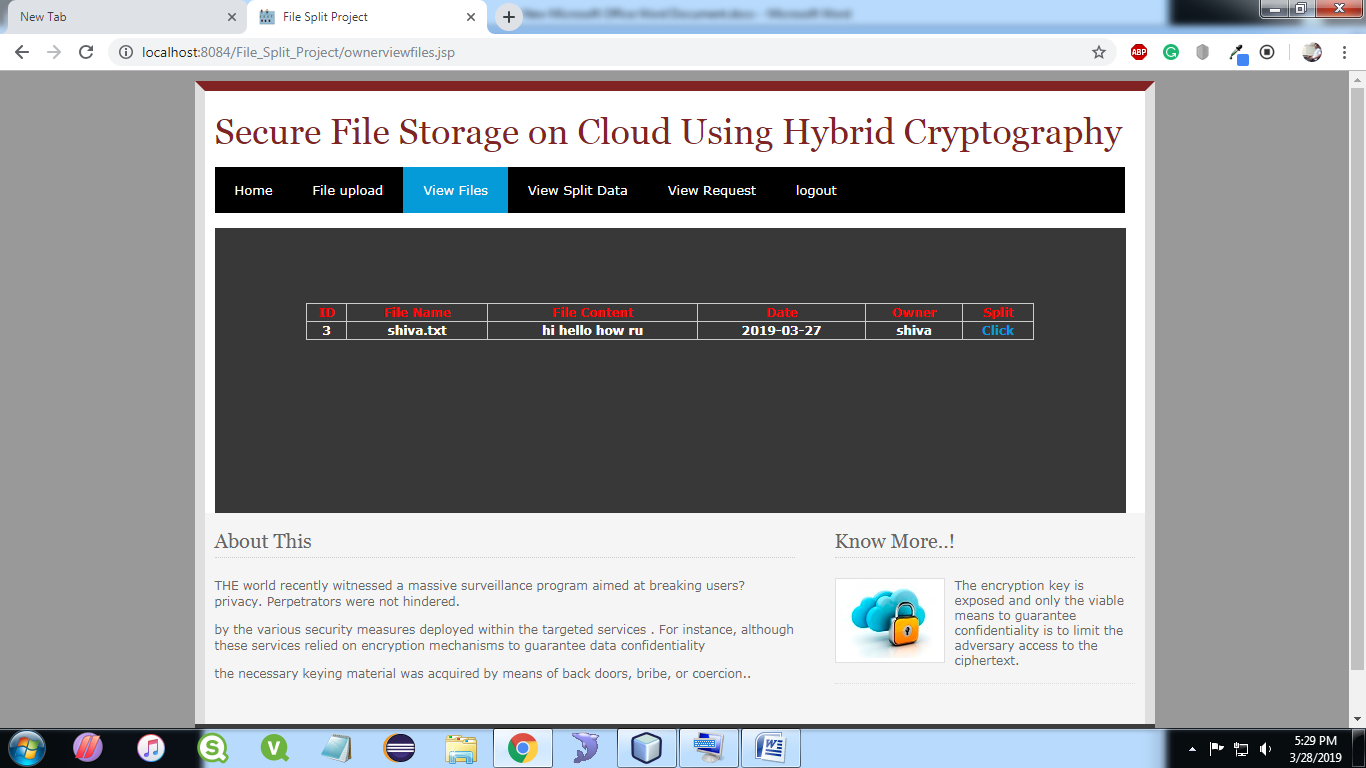
**7.5 FILE UPLOAD**



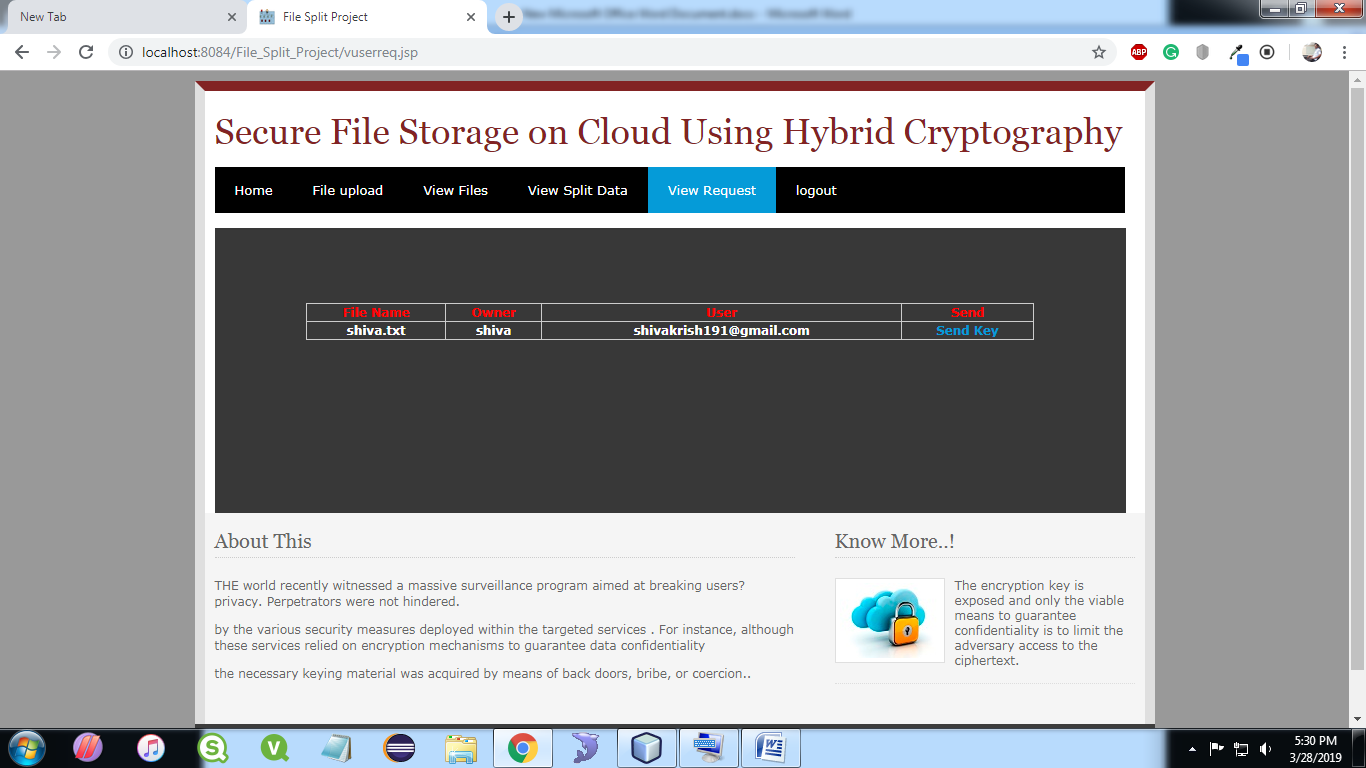
**7.6 VIEW FILE**



**7.7 VIEW SPLIT DATA**



**7.8 VIEW REQUEST**



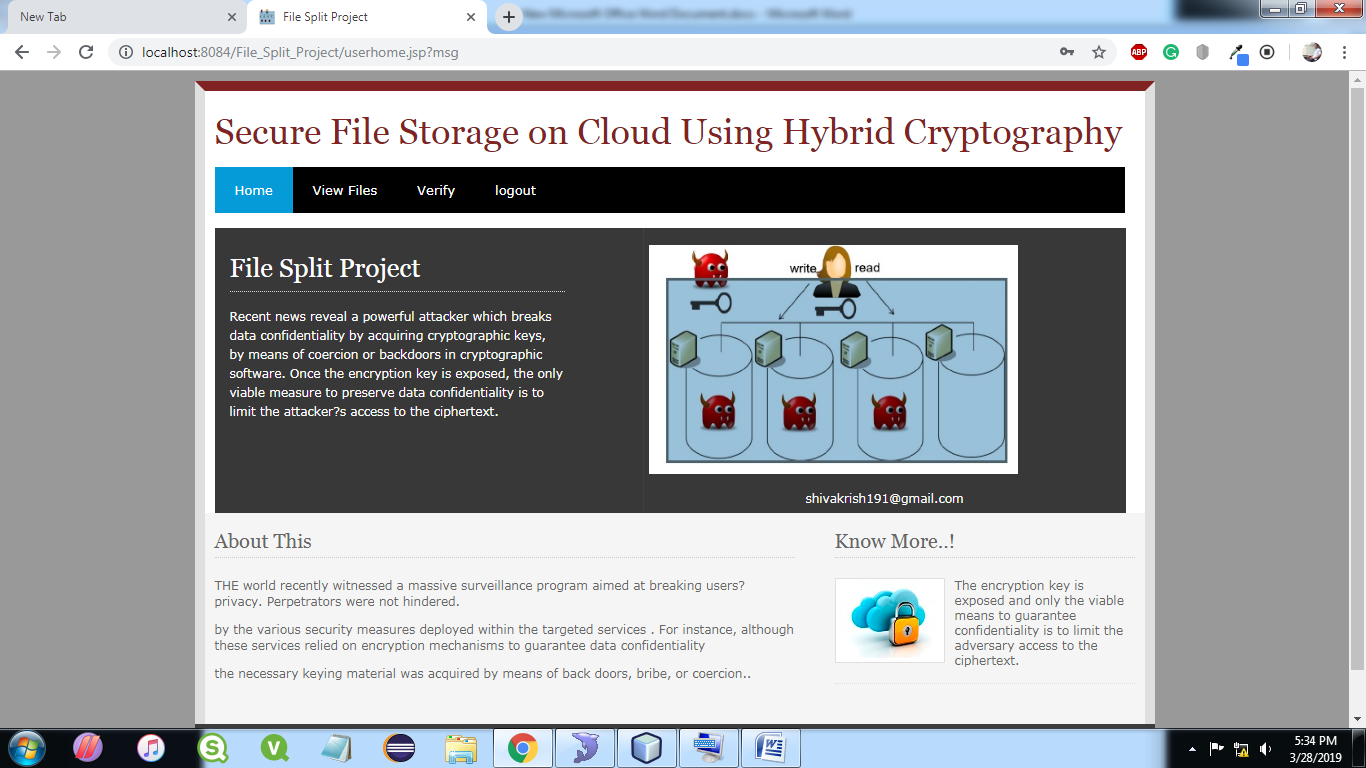
**7.9 USER REGISTRATION**



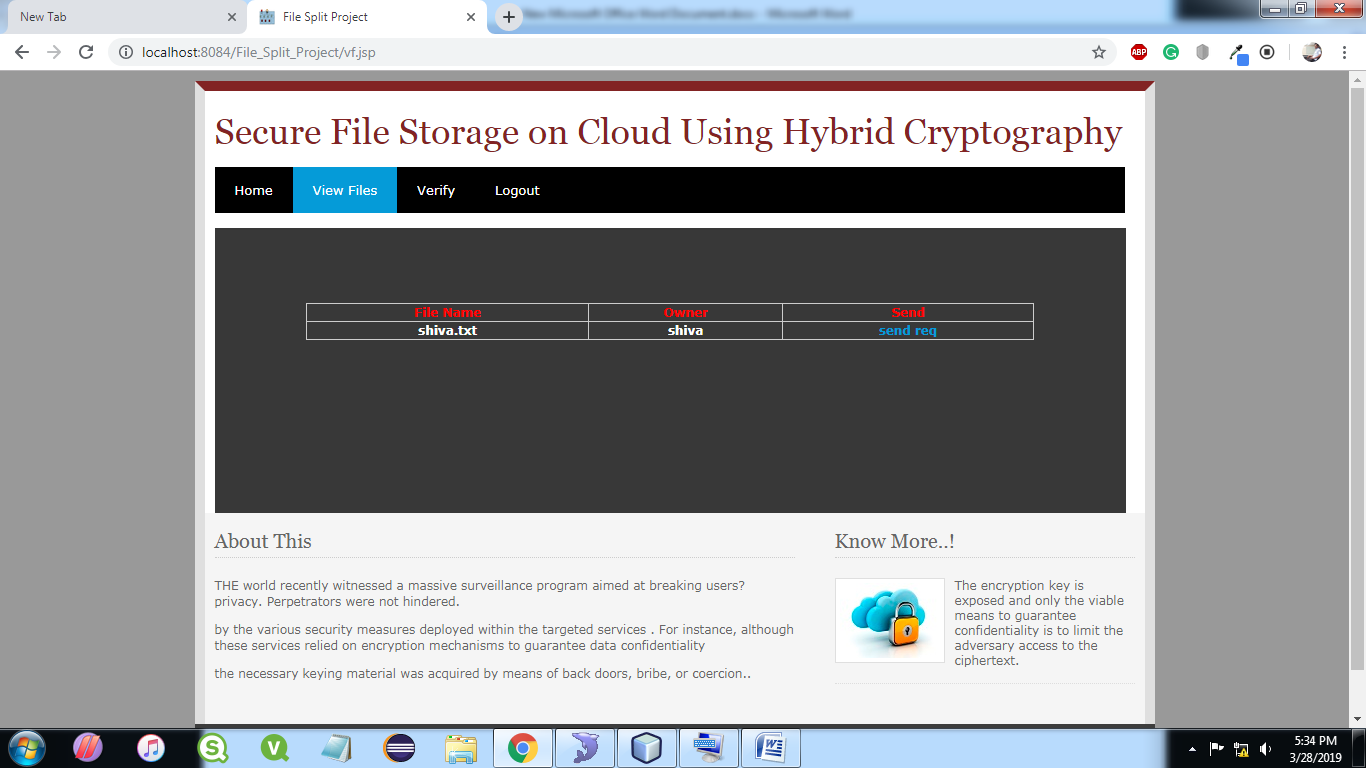
**7.10 USER LOGIN**



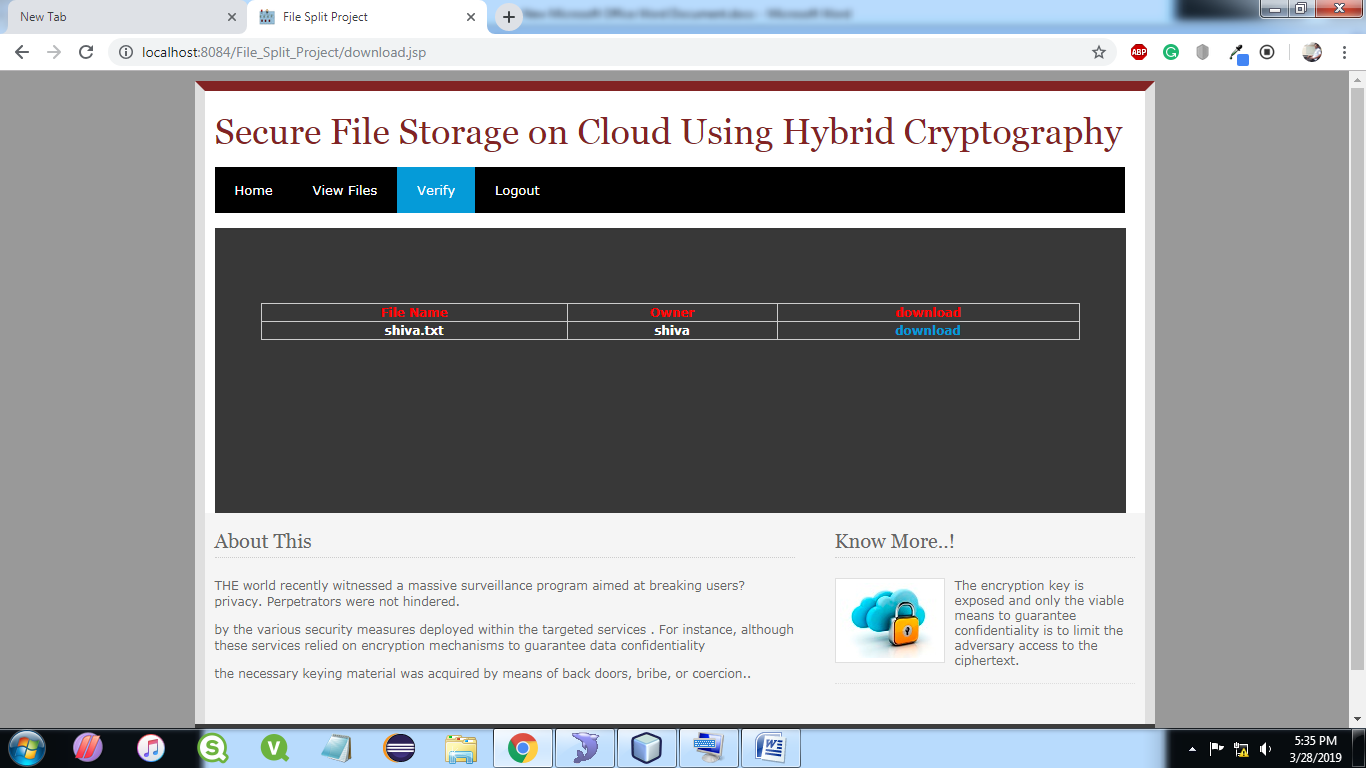
**7.11 USER HOME**



**7.12 VIEW FILE IN USER**



**7.13 USER VREIFY**



**8. CONCLUSION AND FUTURE SCOPE**

The main goal is to securely store and access data in clouds that is not controlled by the owner of the data. We exploit the technique of elliptic curve cryptography encryption to protect data files in the cloud. Two parts of the cloud server improved the performance during storage and accessing of data. The ECC Encryption algorithm used for encryption is another advantage to improve the performance during encryption and decryption process. We assume that this way of storing and accessing data is much more secure and have high performance. Our efforts are going on to solve the problem of group sharing of data in the shared data section as in this scheme only members of group can access the data stored over shared data section. One too many, many to one, many too many communications is not possible.

**Future Enhancements:**

It is not possible to develop a system that meets all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* As technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
* Based on the future security issues, security can be improved using emerging technologies like single sign-on.

**9.REFERENCES**

[1] Peter Mel and Tim Grace, "The NIST Definition of Cloud Computing", NIST, 2010.

[2] Achill Buhl, "Rising Security Challenges in Cloud Computing", in Proc. of World Congress on Information and correspondence Technologies, pp. 217-222, Dec. 2011.

[3] Srinivasarao D et al., "Breaking down the Superlative symmetric Cryptosystem Encryption Algorithm", Journal of Global Research in Computer Science, vol. 7, Jul. 2011

[4] Tingyuan Nye and Tang Zhang "An investigation of DES and Blowfish encryption algorithm‟, in Proc. IEEE Region 10 Conference, pp. 1-4, Jan. 2009.

[5] Jitendra Singh Adam et al.," Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering,vol. 2,Aug. 2012.

[6] Manikandan.G et al., "A changed cryptographic plan improving information", Journal of Theoretical and Applied Information Technology, vol. 35, no.2, Jan. 2012.

[7] Niles Maintain and Subhead Bhingarkar, " The examination and Judgment of Nimbus, Open Nebula and Eucalyptus", International Journal of Computational Biology, vol. 3, issue 1, pp 44-47, 2012.

**BIBLIOGRAPHY**

JAVA Technologies

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

HTML

HTML Black Book by Holzner

JDBC

Java Database Programming with JDBC by Patel moss.

Software Engineering by Roger Pressman