**CHAPTER 1**

**INTRODUCTION**

The necessity for secure communication between entities on the Grid has motivated the development of the Dual Authentication Methodology. DAM provides integrity, protection, confidentiality and authentication for sensitive information. Secure invocation of Grid services brings out the need for a security model that reflects the security components that require be recognizing and defining based on the Grid security requirements. In this project we are going to implement the same dual authentication methodology to provide security and protection for the user from the unauthorized user. This project will be very use full in education sectors and in public libraries, normally in libraries we use only simple authentication methodologies, in this project we use DAM to provide more security and protection.

* 1. **OBJECTIVE**

The objective of this project is to achieve the dual authentication; it necessitates the user to register with the Authentication file. The users have to register with the Authentication file, so that it can hold a part of the interpreted password with itself and another part in the Backend file. Authentication token is retrieved from backend file and is matched with the authentication index, if matched the user is validated

**CHAPTER 2**

**SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

Security is of utmost importance in grid computing applications as grid resources are heterogeneous, dynamic and multi-domain. Authentication remains as the significant security challenge in grid environment. In traditional authentication protocol a single server stores the sensitive user credentials, like username and password.

**Drawbacks**

1. When such a server is compromised, a large number of user passwords, will be exposed
2. Easy to hack the password
3. Doesn’t provide confidentiality for sensitive information.
   1. **PROPOSED SYSTEM**

Our proposed approach uses a dual authentication protocol in order to improve the authentication service in grid environment. The protocol utilizes the fundamental concept of trigon and based on the parameters of the trigon the user authentication will be performed. In the proposed protocol, the password is interpreted and alienated into more than one unit and these units are stored in two different servers, namely Authentication Server and Backend Server. Only when the combined authentication scheme from both the servers authenticates the user, the privilege of accessing the requested resources is obtained by the user.

**Advantages**

1. Dual Authentication Methodology (DAM) providesIntegrity, Protection, Confidentiality and Authentication for sensitive information.
2. Very useful in education sectors and in public libraries, normally in libraries we use only simple authentication methodologies, this project we use DAM to provide more security and protection.

**2.3 FEASIBILITY STUDY**

A Feasibility study is a preliminary study that is undertaken to determine and document a project’s viability. The term feasibility study is used to refer to the resulting document. The results of this study are used to make a decision whether to proceed with the project. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative.

**2.3.1 ECONOMIC FEASIBILITY**

To implement the software, two systems with LAN connection is necessary. So it is economically feasible to develop the project.

**2.3.2 OPERATIONAL FEASIBILITY**

The software developed is easy to operate under Windows application. The end users with basic Computer knowledge can easily work with this software. So it is said to be operationally feasible to develop the project.

**2.3.3 TECHNICAL FEASIBILITY**

The software is developed using Java. It is easy to work, so it is technically feasible for development of the project.

**CHAPTER 3**

**SYSTEM SPECIFICATION**

**3.1 HARDWARE REQUIREMENTS**

PROCESSOR : PENTIUM IV 2.6 GHz

RAM : 512 MB DD RAM

MONITOR : 15” COLOR

HARD DISK : 20 GB

FLOPPY DRIVE : 1.44 MB

CD-DRIVE : LG 52X

KEYBOARD : STANDARD 102 KEYS

MOUSE : 3 BUTTONS

**3.2 SOFTWARE REQUIREMENTS**

FRONT END : JAVA

BACK END : MICROSOFT ACCESS 2007

OPERATING SYSTEM : WINDOWS XP

**CHAPTER 4**

**SOFTWARE DESCRIPTION**

**4.1 FRONT END AND ITS FEATURES**

**4.1.1 JAVA**

Java is one of the most popular languages in the field of software programming. The reason behind it’s widely acceptance in software programming is its large number of important and robust features. Java can be used to develop wide variety of dynamic, fully secure and platform independent applications. Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called Java byte codes--the platform-independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how this works.

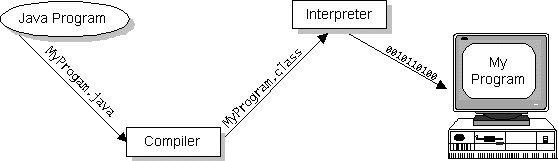


Figure 4.1.1.1 Working of JAVA

**Features**

**Simple**

Java is easy to write and understand. Java programmer doesn’t have to work with pointers and doesn’t need to manage memory explicitly. Java eliminated the use of pointers for security reasons. Its virtual machine is able to handle the memory management and so removes the occupied memory automatically when it is no longer referenced. Programmer can now concentrate on the required application logic rather than wasting time and logic with these managements.

**Java is object – oriented**

There are several languages like C and C++ which are not pure object oriented but Java is fully object oriented language which follows all the rules of OOPs like Inheritance, Encapsulation, Polymorphism etc and everything can be treated in the form of objects. Even primitive data types can also be treated as objects using wrapper classes to make it fully object oriented.

**Java is distributed**

Java supports network programming to communicate with remote objects distributed over the network. Java provides libraries like RMI and CORBA to develop network applications.

**Java is interpreted**

Java is called interpreted language because its byte code, which is generated after compiling the source code, its interpreted by the JVM and converts to machine dependent code which is also, called native code.

**Java is robust and secure**

Java applications are reliable in various ways. It offers compile time checking to detect early the causes of bugs, run time checking, eliminates the use of pointers which can cause memory corruption or unwanted access of memory, garbage collection management to free the unused memories automatically, and exception handling to handle the situation at the time of occurrence of any error and a lot more.

**Java is architecturally neutral**

Java is designed to make the application which is neutral to systems of any architecture with variety of CPUs and Operating systems. For this, java compiler converts the source code to the intermediate platform independent code, called byte code, at the time of compilation which has nothing to do with any system architecture. This byte code makes java an architecture neutral language because this code can be interpreted by any system which has java virtual machine (JVM) installed in it.

**Multithreaded**

In Java, multithreaded applications are easy to develop using synchronization in comparison to other programming languages. Applications performing several tasks at the same time are called multithreaded applications which are useful for providing quick and real time response.

**AWT and Event Handling**

The original design goal of the graphical user interface called as GUI library in Java was to allow the programmer to build a GUI that looks good on all platforms. The Java Abstract Window Toolkit called as AWT produces a GUI that looks equally good on all systems. Thus much of the situation has been improved with the Java Abstract Window Toolkit AWT by using a very good object oriented approach also with the use of Java Beans which is a component programming model that is very useful for the creation of easy and effective visual programming environments. Thus the code attached to these events gets fired when the action or event occurs. This is very important concept supported by Java in an efficient way which takes care of interactive programming methodology thus making it a powerful programming tool.

**Networking**

This is a very important feature of Java and in Java it is possible to achieve networking without much effort because of the powerful features possessed by it. In other words, the concept of networking has become so easy in Java because the underlying details of networking is abstracted and taken away and is taken care of within the JVM and local machine installation of Java.

**Socket**

In Java one creates a socket to make a connection to the other machine and thereby gets an Input Stream and Output Stream from the socket with the appropriate converters namely Reader and Writer which helps to treat the connection as an IO stream object. Generally there are two stream–based socket classes namely a ServerSocket that a server uses to listen for incoming connections and a Socket that a client uses in order to initiate a connection. Once a client makes a socket connection the ServerSocket returns using appropriate method namely accept() method a corresponding server side socket through which direct communications take place. From this point one gets a true Socket to Socket connection and one can treat both ends the same way as they are same.

**Database Connection**

The greatest feature of Java is its ability to build platform-independent client/server database applications. In Java this is possible with Java Database Connectivity also called as JDBC in short, Java Database Connectivity is designed to be platform independent and so one need not worry about the database one use while they are programming.

**4.2 BACK END**

**4.2.1 MS ACCESS 2007**

Access 2007 is an application that allows you to create and manage databases. A database is a place where you actually store in information related to a specific topic. How you intend to use information will determine whether you need an Access database or different program to create and manage your data.

Our project used Ms Access as the back end to store the user name, alpha, product and variance in the tables. We used two data base, one is authentication file and another one is backend file. In backend file we stored values of alpha product and variance. In authentication file we store username and authentication token.

**CHAPTER 5**

**PROJECT DESCRIPTION**

**5.1 PROBLEM DEFINITION**

In traditional authentication protocol a single file stores the sensitive user credentials, like username and password. When such a file is compromised, a large number of user passwords, will be exposed.

**5.2 OVERVIEW OF THE PROJECT**

In this project, the password is interpreted and alienated into more than one unit and these units are stored in two different servers, namely, Authentication Server and Backend Server. Only when the combined authentication scheme from both the servers authenticates the user, the privilege of accessing the requested resources is obtained by the user.

The main advantage of utilizing the dual authentication protocol in grid computing is that an adversary user cannot attain the access privilege by compromising a single consolidated server because of the fact that the split password is stored in different servers.

**5.3 MODULE DESCRIPTION**

**MODULES**

This project contains two modules.

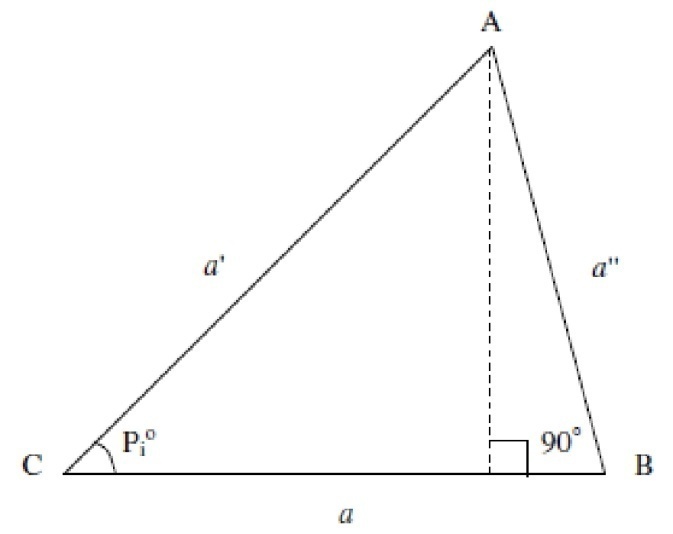
1. New user registration
2. Authentication

**5.3.1 NEW USER REGISTRATION**

To achieve the dual authentication, it necessitates the user to register with the Authentication file. The users have to register with the Authentication file, so that it can hold a part of the interpreted password with itself and another part in the Backend file.

It is used during registration of username and password and it stores the **username, variance and product values in backend file** which is used to find the password.

Designing of backend file for the back end is done in this module. The functionalities of this module are to collect user name and password, calculating product, variance and alpha values from the randomly or automatically generated two prime numbers.



**Fig 5.3.1.1** Sample trigon generated using parameters a, a’, a’’ and Pi

Va a’ = a – a’

Pa a’ = a \* a’

α = 2Pa a’ – a’’2

After the calculation of *α*, *Vaa*' and *Paa*', the **authentication file stores the *α* value and its corresponding username** in a database and forwards *Vaa*' and *Paa*' to the Backend file along with the username. The Backend file maintains the *Vaa*' and *Paa*' for the corresponding username in a database. Hence, the password is interpreted and alienated into two units and stored in two separate files, thereby achieving the concept of dual authentication .The process is repeated for all the users who wish to register in the file, so that both the files can maintain all the users’ account. If the user is an adversary and if it tries to use wrong password or username, the file can validate effectively, asserts the user as invalid and sends a warning to the adversary.

**5.3.2 AUTHENTICATION**

Authentication token is retrieved from backend file and is matched with the authentication index, if matched the user is validated.

In this module, the authentication procedure we have developed is on the basis of the fundamental concepts of a trigon, login to the authentication file using the username and password. Here, *ui* and *pwi* refers to username and password of *ith* user. The authentication file calculates the Password index (Pi) from the password as

Pi = PAi /10n-2 if PAi >= 180

Or

Pi = PAi /10n-3

PAi is the ASCII-interpreted value of the given password pwi, n is the total number of digits in PAi and PAi (j) represents the first j digits of PAi. The PAi can be calculated by the following steps.

* Change the pwi into its corresponding ASCII value.
* Calculate the three-fourth of total digits of the ASCII value modulo 180, which results the first three digits of PAi.
* Append the remaining one-fourth of the ASCII digits to PAi.

Then from Pi, the authentication file determine the authentication index (Ai) for Ui as

AI (i) = Pi/2

Then, the authentication File searches for the username index α i for the corresponding ui which has already been stored in the file database during the process of the registration. Subsequently, αi is sent to the backend file along with ui. When the Backend file receives the index αi and the username from the authentication file, it searches for Vaa’ and Paa’ the variance and the product of the sides a and a' respectively, which have been saved in the backend file database during the process of registration.

**5.4 FLOW CHART EXPLAINING REGISTRATION PROCESS OF USER**

Backend server saves Vaa' and Paa'  for the corresponding username

Saves α and forwards username, Vaa' and Paa' to Backend server

Determines variance and the product of the sides a and a’ and strengthening parameter used as the index to represent user credentials (*α*, *Vaa*' and *Paa*')

Authentication server calculates Pi (the angle between the sides a and a’) and generates the three sides of the trigon a, a’ and a”)

User registers with username and password in Authentication server

**Fig 5.4.1**

**5.5 ACTIVITY DIAGRAM FOR AUTHENTICATION**

to user

user

TVo(if TVo received)

contacts VO using

User Logins

Find PAi

PAi >= 180

Pi = PAi /10n-3

Pi = PAi /10n-2

AI (i) = Pi/2

alpha, Vaa' and

Paa'

Sin (A (i) = (1 - A/2)1/2

ATi = alphai +

V2aa'i/ 2Paa'i

Issues TVo

User

Authentication server

Backend server

ui, pwi

Yes

No

yes

No

Warns the

**Fig 5.5.1**

**CHAPTER 6**

**SYSTEM TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding. The engineer creates a series of test cases that are intended to “demolish” the software that has been built. In fact, testing is the one step in this software engineering process that could be viewed as destructive rather than constructive. The testing process is divided into three major components as follows:

* Unit Testing
* Integration Testing
* Validation Testing

**6.1 UNIT TESTING**

The software is tested using the unit test method. Unit testing focuses verification effort on the smallest unit of the software design module. Using the procedural design description as a guide, important control parts are tested and it covers the errors within the boundary of the module.

**6.2 INTEGRATION TESTING**

This testing is the systematic technique for constructing the program structure by performing the test in each module and later combining the entire individual module to form a very large program.

**6.3 VALIDATION TESTING**

Validation testing is the process of testing the inputs that is been provided by the user. This helps in checking whether the given inputs by the user are valid or invalid.

**Example:**

The error message “invalid user” will be displayed whenever the wrong user id and password entered. Because it accept only authenticated users.

**CHAPTER 7**

**SYSTEM IMPLEMENTATION**

The implementation is the process of converting the design into actual code. The implementation process is done after completing the designing task. The main objective of this system is to achieve the dual authentication where the password is interpreted and alienated into more than one unit and these units are stored in two different servers, namely, Authentication Server and Backend Server. The implementation of any software requires perfect satisfaction of the user. The proposed system takes into account the various difficulties faced by the users in operating system and provides users friendly operations. The system is implemented in the server and checked for its performance and accuracy. Successful implementation of the software is accomplished. This project is to be done using Java as front end and Ms Access as back end, connected through JDBC Connector.

**CHAPTER 8**

**CONCLUSION**

As the servers kept the interpreted and distinct form of user credentials, there is very less chance to reveal the user credentials to the adversary. Moreover, the protocol utilized the fundamental properties of the trigon and the trigon parameters. This simple trigon concept utilization in the authentication protocol introduced a novel and revolutionary idea in the authentication mechanism as well as in grid environment. So in future enhancement we are going to implement in grid environment. This protocol uses two server and trigon parameter, made the grid more secure as the alienated passwords had been derived from these trigon parameters.

The maintenance of the records is made efficient, as all the records are stored in the ACCESS database, through which data can be retrieved easily. The navigation control is provided in all the forms to navigate through the large amount of records. If the numbers of records are very large then user has to just type in the search string and user gets the results immediately.

**APPENDIX 1**

**SAMPLE CODE**

**Client Side**

**MainMenu.java**

import java.io.\*;

import java.sql.\*;

import javax.swing.\*;

import sun.jdbc.odbc.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.lang.\*;

import java.util.Random;

import javax.swing.border.\*;

public class MainMenu extends JPanel implements ActionListener

{

JLabel l1=new JLabel("TRIGON BASED DUAL AUTHENTICATION");

JButton b1=new JButton("NEW USER");

JButton b2=new JButton("EXISTING USER");

JButton b3=new JButton("EXIT");

JButton b4;

ImageIcon im;

JPanel nor, cen, sou, back;

public MainMenu()

{

JFrame fr=new JFrame("Trigon Based Dual Authentication ");

Container c=fr.getContentPane();

back = new JPanel();

nor = new JPanel();

cen = new JPanel();

sou = new JPanel();

im=new ImageIcon("tri.jpg");

b4=new JButton("TRIGON",im);

back.setLayout(new BorderLayout());

nor.setLayout(new FlowLayout());

cen.setLayout(new FlowLayout());

sou.setLayout(new FlowLayout());

b4.setActionCommand("TRIGON");

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

nor.add(l1);

b4.setBounds(50,50,600,400);

c.add(b4);

sou.add(b1);

sou.add(b2);

sou.add(b3);

back.setBorder(new BevelBorder(BevelBorder.LOWERED));

back.add(nor, BorderLayout.NORTH);

back.add(cen, BorderLayout.CENTER);

back.add(sou, BorderLayout.SOUTH);

c.add(back);

setLocation(225, 240);

fr.setSize(500,500);

fr.setVisible(true);

fr.show();

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

if(s.equals("NEW USER"))

{

Sample h=new Sample();

h.show();

}

if(s.equals("EXISTING USER"))

{

NextPage h=new NextPage();

h.show();

}

if(s.equals("EXIT"))

{

System.exit(0);

}

}

public static void main(String args[])

{

MainMenu s=new MainMenu();

s.setSize(500,500);

s.setVisible(true);

}

}

**Sample.java**

import java.io.\*;

import java.sql.\*;

import javax.swing.\*;

import sun.jdbc.odbc.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.lang.\*;

import java.util.Random;

import javax.swing.border.\*;

import java.net.\*;

public class Sample extends JPanel implements ActionListener

{

static final int packetsize=1024;

static BufferedReader stdin=new BufferedReader(new InputStreamReader(System.in));

JLabel l1=new JLabel("USERNAME:");

JLabel l2=new JLabel("PASSWORD:");

JLabel l3=new JLabel("SEX:");

JLabel l4=new JLabel("YOUR SCHOOL NAME:");

JLabel l5=new JLabel("YOUR FAVOURITE MOVIE:");

JLabel l6=new JLabel("YOUR FAVOURITE DISH:");

JTextField t1=new JTextField(10);

JPasswordField t2=new JPasswordField(10);

JTextField t3=new JTextField(10);

JTextField t4=new JTextField(10);

JTextField t5=new JTextField(10);

JTextField t6=new JTextField(10);

JButton b1=new JButton("REGISTER");

JButton b2=new JButton("NEXT");

JButton b3=new JButton("EXIT");

JPanel nor, cen, sou, back;

public Sample()

{

JFrame fr=new JFrame(" User Registration");

Container c=fr.getContentPane();

back = new JPanel();

nor = new JPanel();

cen = new JPanel();

sou = new JPanel();

back.setLayout(new BorderLayout());

nor.setLayout(new FlowLayout());

cen.setLayout(new FlowLayout());

sou.setLayout(new FlowLayout());

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

nor.add(l1);

nor.add(t1);

nor.add(l2);

t2.setEchoChar('\*');

nor.add(t2);

nor.add(l3);

nor.add(t3);

cen.add(l4);

cen.add(t4);

cen.add(l5);

cen.add(t5);

cen.add(l6);

cen.add(t6);

sou.add(b1);

sou.add(b2);

sou.add(b3);

back.setBorder(new BevelBorder(BevelBorder.LOWERED));

back.add(nor, BorderLayout.NORTH);

back.add(cen, BorderLayout.CENTER);

back.add(sou, BorderLayout.SOUTH);

c.add(back);

setLocation(225, 240);

fr.setSize(500,500);

fr.setVisible(true);

fr.show();

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

String s1=t1.getText();

String s2=t2.getText();

String s3=t3.getText();

String s4=t4.getText();

String s5=t5.getText();

String s6=t6.getText();

if(s.equals("REGISTER"))

{

disp(s1,s2,s3,s4,s5,s6);

}

if(s.equals("NEXT"))

{

NextPage h=new NextPage();

h.show();

}

if(s.equals("EXIT"))

{

System.exit(0);

}

}

public void disp(String m1,String m2,String m3,String m4,String m5,String m6)

{

DatagramPacket packet;

byte[] data1;

byte[] data2;

byte[] data3;

byte[] data4;

byte[] data5;

byte[] data6;

byte[] data7;

DatagramSocket socket;

int port=9999;

String s="172.25.1.170";

String messagereturn;

try

{

InetAddress address=InetAddress.getByName(s);

if(m1.length()==0)

{

System.exit(0);

}

socket=new DatagramSocket();

data1=m1.getBytes();

packet=new DatagramPacket(data1,data1.length,address,port);

socket.send(packet);

data2=m2.getBytes();

packet=new DatagramPacket(data2,data2.length,address,port);

socket.send(packet);

data3=m3.getBytes();

packet=new DatagramPacket(data3,data3.length,address,port);

socket.send(packet);

data4=m4.getBytes();

packet=new DatagramPacket(data4,data4.length,address,port);

socket.send(packet);

data5=m5.getBytes();

packet=new DatagramPacket(data5,data5.length,address,port);

socket.send(packet);

data6=m6.getBytes();

packet=new DatagramPacket(data6,data6.length,address,port);

socket.send(packet);

JOptionPane.showMessageDialog(this,"Registering.............");

}

catch(IOException ioe)

{

System.out.println("Could not receive:"+ioe.getMessage());

System.exit(0);

}}

public static void main(String args[])

{

Sample s=new Sample();

s.setSize(500,500);

s.setVisible(true);

}}

**NextPage.java**

import java.io.\*;

import java.sql.\*;

import javax.swing.\*;

import sun.jdbc.odbc.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.lang.\*;

import java.util.Random;

import javax.swing.border.\*;

public class NextPage extends JPanel implements ActionListener

{

JLabel l1=new JLabel("USERNAME:");

JLabel l2=new JLabel("PASSWORD:");

JTextField t1=new JTextField(20);

JPasswordField t2=new JPasswordField(20);

JButton b2=new JButton("LOGIN");

JButton b3=new JButton("EXIT");

JPanel nor, cen, sou, back;

public NextPage()

{

JFrame fr=new JFrame(" User Authentication");

Container c=fr.getContentPane();

back = new JPanel();

nor = new JPanel();

cen = new JPanel();

sou = new JPanel();

back.setLayout(new BorderLayout());

nor.setLayout(new FlowLayout());

cen.setLayout(new FlowLayout());

sou.setLayout(new FlowLayout());

b2.addActionListener(this);

b3.addActionListener(this);

nor.add(l1);

nor.add(t1);

cen.add(l2);

t2.setEchoChar('\*');

cen.add(t2);

sou.add(b2);

sou.add(b3);

back.setBorder(new BevelBorder(BevelBorder.LOWERED));

back.add(nor, BorderLayout.NORTH);

back.add(cen, BorderLayout.CENTER);

back.add(sou, BorderLayout.SOUTH);

c.add(back);

setLocation(225, 240);

fr.setSize(500,500);

fr.setVisible(true);

fr.show();

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

int t=0;

double al=0.0;

String s1=t1.getText();

String s2=t2.getText();

if(s.equals("LOGIN"))

{

try

{

String query3 = "select \* from details";

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("Jdbc:Odbc:pentagon");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(query3);

String sx, sy, sz;

boolean b1 = false;

String q2 = JOptionPane.showInputDialog(this, "Enter your School name");

String q3 = JOptionPane.showInputDialog(this, "Enter your Favourite movie");

String q4 = JOptionPane.showInputDialog(this, "Enter your Favourite dish");

while (rs.next())

{

sx = rs.getString("school");

sy = rs.getString("movie");

sz = rs.getString("dish");

if (q2.equals(sx) && q3.equals(sy) && q4.equals(sz) )

{

JOptionPane.showMessageDialog(this,"Valid verification");

al=auth();

authenticate(s1,s2,al);

b1 = true;

break;

}

}

if (!b1)

{

JOptionPane.showMessageDialog(this,"wrong answer");

System.exit(0);

}

}

catch(Exception e1)

{

System.out.println("Exception in authentication :"+e1);

}}

if(s.equals("EXIT"))

{

System.exit(0);

}}

public double auth()

{

double ap=0.0;

try

{

String nam="";

String query = "select \* from ass";

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("Jdbc:Odbc:pentagon");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(query);

int exist=0,m=0,fag=0;

while (rs.next())

{nam = rs.getString(1);

ap = Double.parseDouble(rs.getString(2));

}

}

catch(Exception e)

{

System.out.println("Exception in authentication :"+e);

}

return ap;

}

public void authenticate(String name, String pword,double alp)

{

try

{

String nam="";double p=0.0,v=0.0,alpha=0.0;

alpha=alp;

String query = "select \* from bs";

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("Jdbc:Odbc:pentagon");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(query);

int exist=0,m=0,fag=0;

while (rs.next())

{nam = rs.getString(1);

if(nam.equals(name))

{System.out.println("user is :"+nam);

m=1;fag=1;

p = Double.parseDouble(rs.getString(2));

v = Double.parseDouble(rs.getString(3));

break;

}

System.out.println(name);

}

if(fag==0)

{

JOptionPane.showMessageDialog(this, "You are not an authenticated user");

}

else

{

System.out.println("p="+p);

System.out.println("v="+v);

System.out.println("alpha="+alpha);

double ati=0.0;

v=Math.pow(v, 2);

ati=(alpha+v);

ati=(ati/(2\*p));

System.out.println("ati="+ati);

String s = pword; pword = new String("");

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

{

pword=pword+(int)s.charAt(i);

}

double pi,ai,x,y;

System.out.println("pword="+pword);

String lastDigit = pword.substring(pword.length()\*3/4,pword.length());

halfDigit = pword.substring(0,pword.length()\*3/4);

double deg = Double.parseDouble(halfDigit);

System.out.println("halfDigit1="+halfDigit);

System.out.println("deg1="+deg);

deg = deg % 180;

System.out.println("deg="+deg);

pword = pword.replace(halfDigit, ""+((int)deg));

System.out.println("halfDigit2="+pword);

halfDigit = pword.substring(0,3);

System.out.println("halfDigit3="+halfDigit);

int pai;

int ld=Integer.parseInt(lastDigit);

halfDigit=halfDigit+ld;

pai=Integer.parseInt(halfDigit);

System.out.println("halfDigit="+pai);

double n=halfDigit.length();

if(pai>=180)

{

pi=pai/(Math.pow(10, n-2));

}

else

{

pi=pai/(Math.pow(10, n-3));

}

ai=pi/2;

System.out.println("ai="+ai);

x=Math.sin(ai)\*Math.sin(ai);

y=((1-(ati))/2);

System.out.println("y1="+y);

y=Math.floor(Math.sqrt(y));y=1;

System.out.println("x="+x+"y="+y);

if(m==y)

{

JOptionPane.showMessageDialog(this,"you are an authenticated user");

}

else

{

JOptionPane.showMessageDialog(this,"you are not an authenticated user");

System.exit(0);

}

}

}

catch(Exception e)

{

System.out.println("Exception in authentication :"+e);

}}

public static void main(String args[])

{

NextPage s=new NextPage();

s.setSize(300,300);

s.setVisible(true);

}

}

**Server Side**

**Sample.java**

import java.net.\*;

import java.io.\*;

import java.io.\*;

import java.sql.\*;

import javax.swing.\*;

import sun.jdbc.odbc.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.lang.\*;

import java.util.Random;

import javax.swing.border.\*;

public class Sample

{

static final int port=9999;

static final int packetsize=1024;

public static void main(String args[])throws IOException

{

DatagramPacket packet1,packet2,packet3,packet4,packet5,packet6,packet7;

DatagramSocket socket;

byte[] data1;

byte[] data2;

byte[] data3;

byte[] data4;

byte[] data5;

byte[] data6;

byte[] data7;

String msg="ok";

String uname,pass,s,sn,fa,di;

int recvsize;

InetAddress address;

int clientport;

socket=new DatagramSocket(port);

data1=new byte[packetsize];

packet1=new DatagramPacket(data1,packetsize);

data2=new byte[packetsize];

packet2=new DatagramPacket(data2,packetsize);

data3=new byte[packetsize];

packet3=new DatagramPacket(data3,packetsize);

data4=new byte[packetsize];

packet4=new DatagramPacket(data4,packetsize);

data5=new byte[packetsize];

packet5=new DatagramPacket(data5,packetsize);

data6=new byte[packetsize];

packet6=new DatagramPacket(data6,packetsize);

System.out.println("To receive packets or port:"+port);

socket.receive(packet1);

socket.receive(packet2);

socket.receive(packet3);

socket.receive(packet4);

socket.receive(packet5);

socket.receive(packet6);

recvsize=packet1.getLength();

uname=new String(data1,0,recvsize);

System.out.println("Username:"+uname.trim());

recvsize=packet2.getLength();

pass=new String(data2,0,recvsize);

recvsize=packet3.getLength();

s=new String(data3,0,recvsize);

recvsize=packet4.getLength();

sn=new String(data4,0,recvsize);

recvsize=packet5.getLength();

fa=new String(data5,0,recvsize);

recvsize=packet6.getLength();

di=new String(data6,0,recvsize);

try

{

System.out.println("you are in ");

String name;

String name1=uname;

System.out.println("username="+name1);

String query = "select \* from ass";

String query2 = "select \* from bs";

String query4 = "select \* from details";

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("Jdbc:Odbc:pentagon");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(query);

int exist=0;

while (rs.next())

{

name = rs.getString(1);

if(name.equals(name1))

{

System.out.println("found");

exist=1;

break;

}

}

if(exist==1)

{

System.out.println("already exist try again ");

System.exit(0);}

else{

double a = prime();

double b = prime();

System.out.println("a = "+a+" b = " + b);

double pro = a \* b;

double vari = a - b;

double degree = getDegree(pass);

if(degree >90D)

{

degree = 180D - degree;

}

double angle = (degree) \* (Math.PI/180D);

double h = b \* Math.sin(angle);

double reminderSide = Math.abs(a - Math.sqrt((Math.pow(b, 2) - Math.pow(h, 2))));

double c = Math.sqrt((Math.pow(h, 2) + Math.pow(reminderSide, 2)));

double cStar = Math.pow(c, 2);

double alpha = (2\*pro) - cStar;

String query1="insert into ass values('"+name1+"','"+alpha+"')";

stmt.executeUpdate(query1);

String query3="insert into bs values('"+name1+"','"+pro+"','"+vari+"')";

stmt.executeUpdate(query3);

String query5="insert into details values('"+name1+"','"+s+"','"+sn+"','"+fa+"','"+di+"')";

stmt.executeUpdate(query5);

System.out.println("Registered Successfully");

}

con.close();}

catch(Exception sql)

{

System.out.println("Sql exception :"+sql);

}

}

public static double getDegree(String pword)

{

double pas = 0.0;

String s = pword; pword = new String("");

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

{

pword=pword+(int)s.charAt(i);

}

String halfDigit = pword.substring(0,pword.length()\*3/4);

double deg = Double.parseDouble(halfDigit);

deg = deg % 180;

pword = pword.replace(halfDigit, ""+((int)deg));

halfDigit = pword.substring(0,3);

deg = Integer.parseInt(halfDigit);

if(deg<180)

{

pword = pword.replace(pword.substring(3), "."+pword.substring(3));

}

else

pword = pword.replace(pword.substring(2), "."+pword.substring(2));

pas = Double.parseDouble(pword);

System.out.println("Pi : "+pas);

return pas;

}

public static double prime()

{

int number=0,nope=0;

Random r = new Random();

number =Math.abs( r.nextInt());

number=number%1000000;

if(number%2 == 0)

number = number+1;

boolean status = true;

while((number -= 2)>0)

{

status = true;

if (number == 1 || number == 2)

return number;

for (int i=2; i<(int)(number/2); i++)

{

if(number%i == 0)

status = false;

if(!status)

i = number;

}

if(status)

break;

}

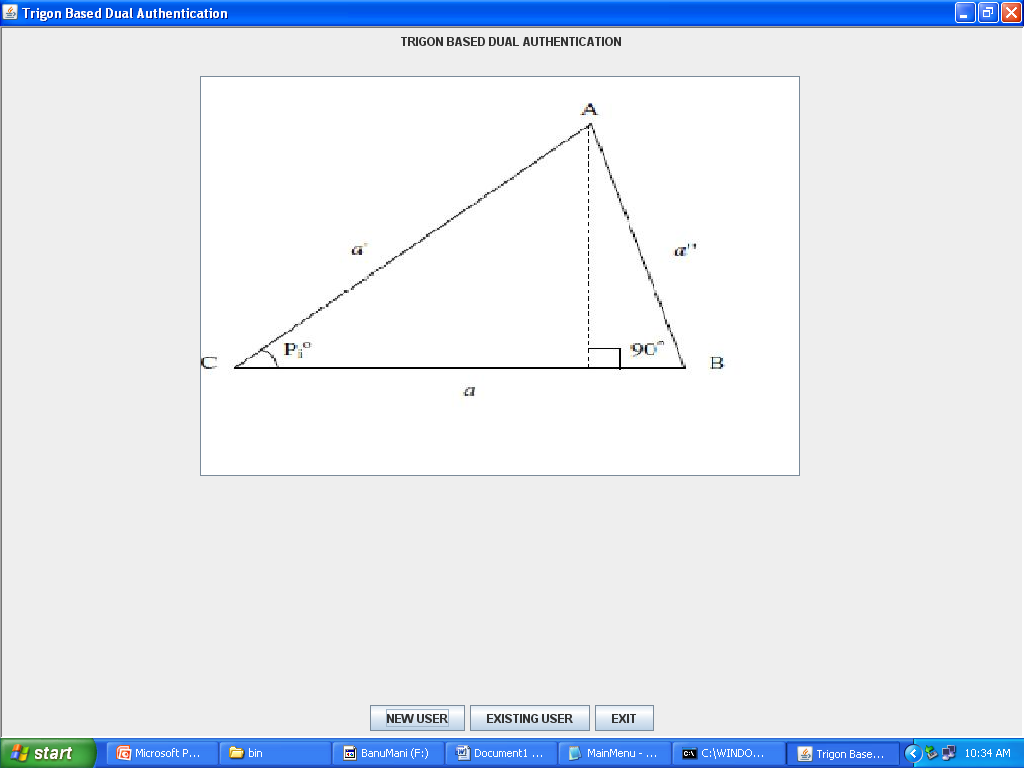
return number;

}

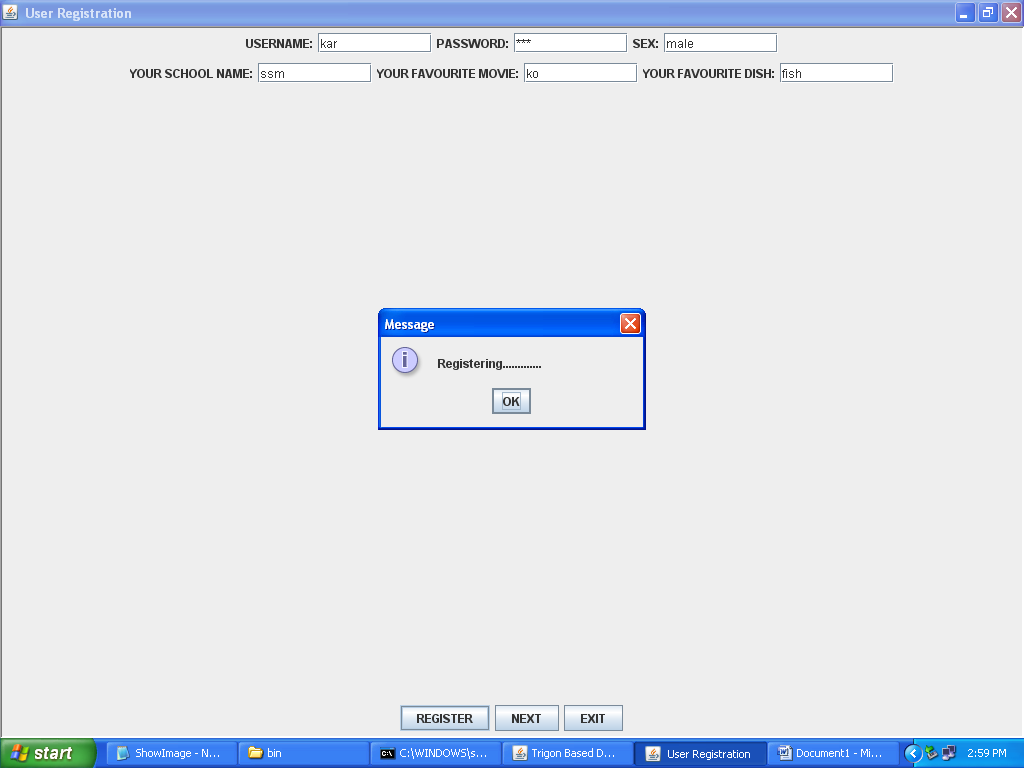
**APPENDIX 2**

**SCREEN SHOTS**

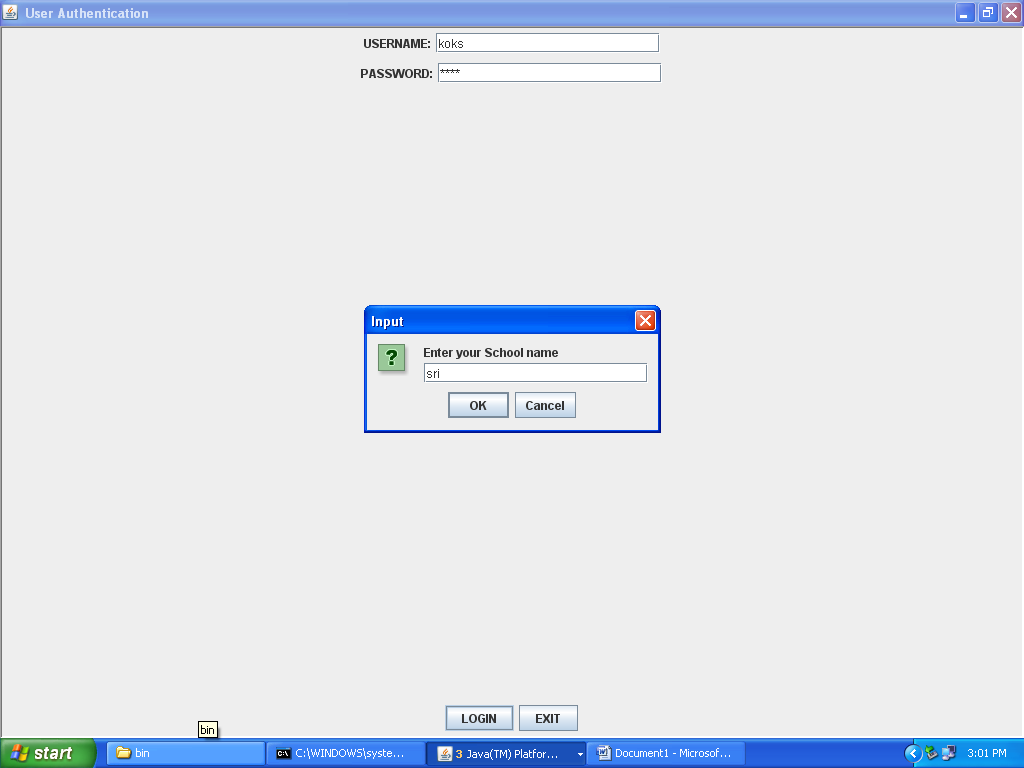
**Main Menu**

****

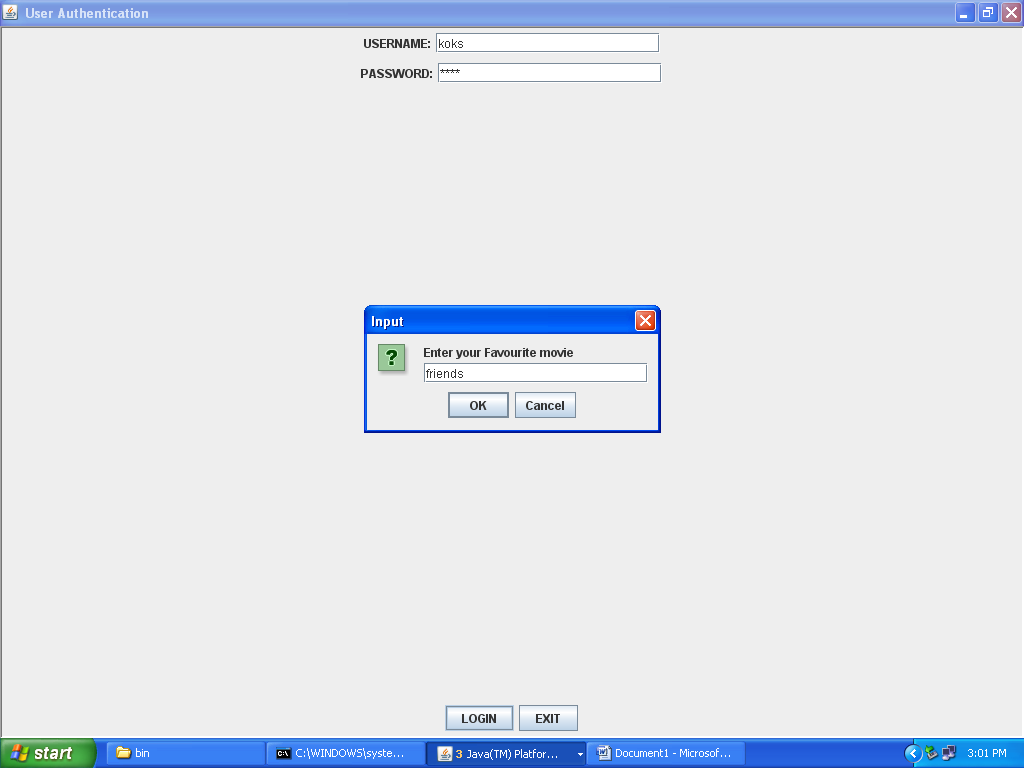
**New User Registration**

****

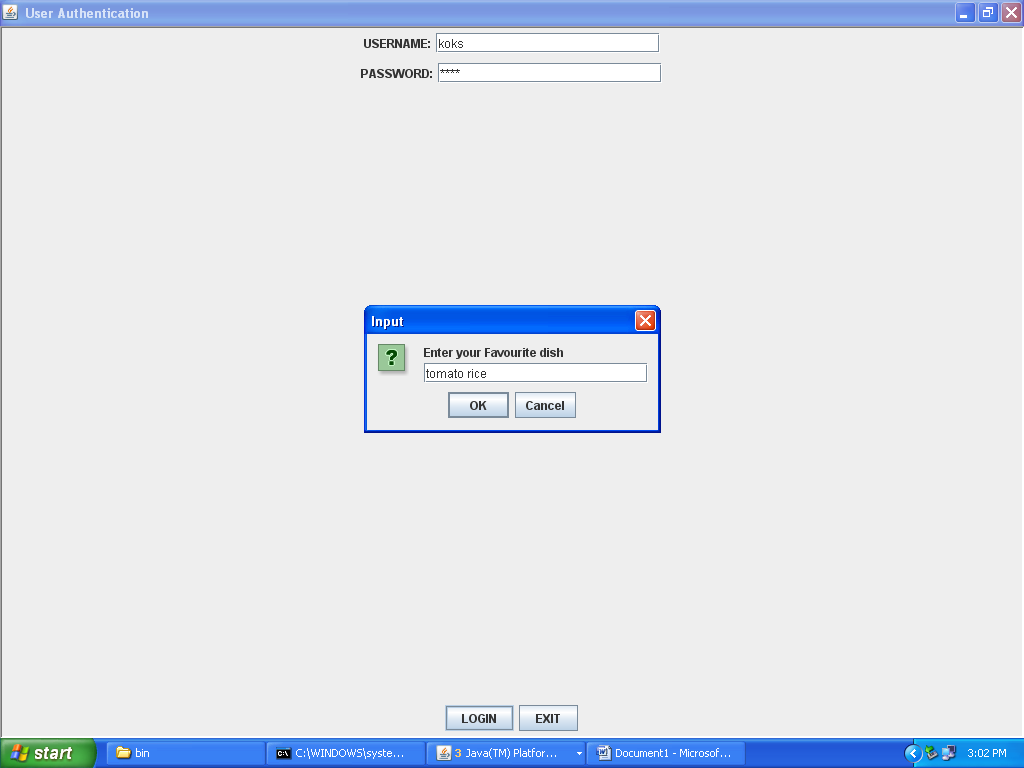
**Authentication**

****

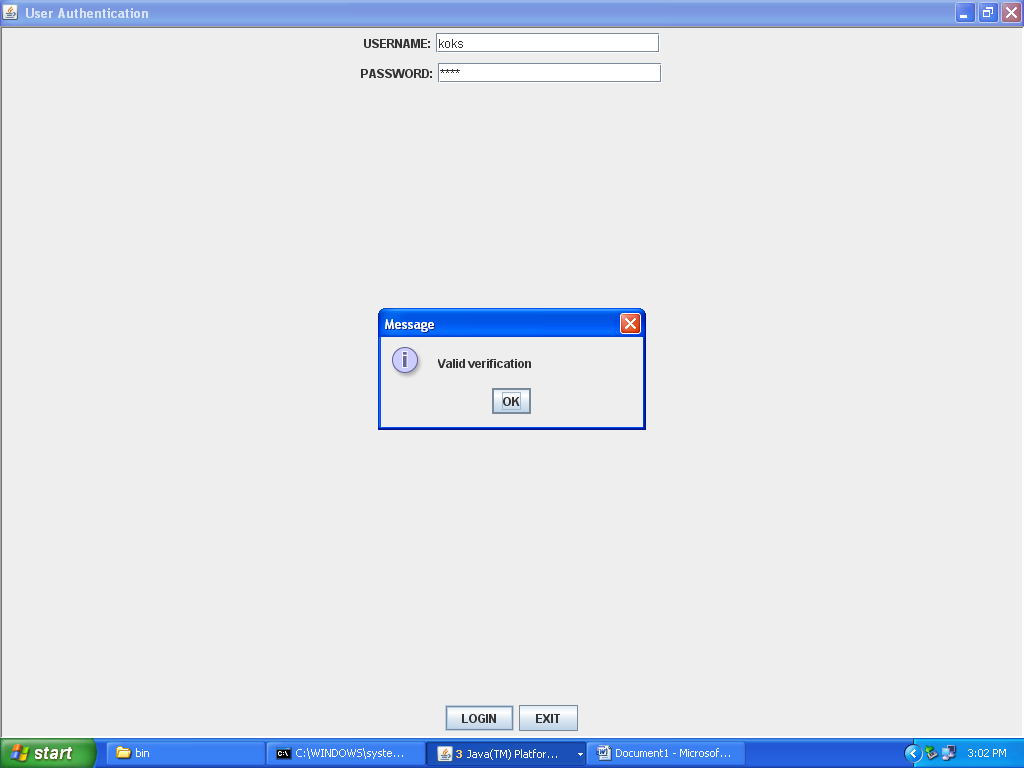
**Authentication**

****

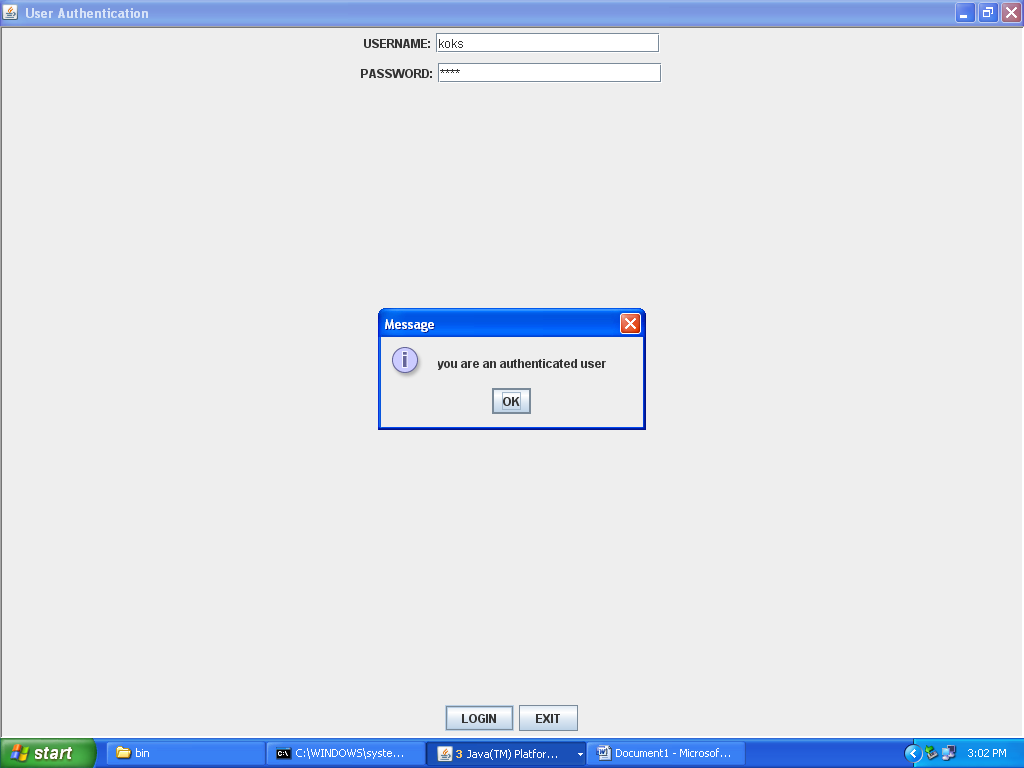
**Authentication**

****

**Authentication**

****

**Authentication**

****

**REFERENCES**

1. V. Ruckmani, Dr. G. Sudha sadasivam,“ A Novel Trigon Based Dual Authentication Protocol for enhancing security in Grid Environment”, Vol. 6, No. 3, 2009
2. Xukai Zou, Yuvan- shun dai, Xiang ran,“ Dual Key Management for secure Grid communication in dynamic and hierarchical groups”, 2006
3. V. Ruckmani, Dr. G. Sudha sadasivam, “Secure group communication in Grid environment”
4. Rajkumar Buyya, Srikumar Venugopal, “A Gentle introduction to Grid Computing and Technologies”
5. www.w3school.com