**DATA SECURITY IN CLOUD**

Project Work  
submitted in partial fulfillment of the   
requirements for the award of degree of

**Bachelor of Technology**

in

**Computer Science & Engineering**

Submitted By

* Tushar Kaushik(1201101253)
* Utkarsh Sharma(1201101309)

Under the super vision of

Mr. Prashant Kumar

Asst. Professor



**School of Computing Science and Engineering**

**Galgotias university**

**Greater noida – 201306**

**may 2015**

Certificate

I hereby certify that the work which is being presented in the project entitled, **“*Software Engineering*”**, in partial fulfillment of the requirements for the award of degree of Bachelor in Technology in Computer Science and Engineering submitted in School of Computing Science and Engineering of Galgotias University, Gr. Noida, is an authentic record of my own work carried out under the supervision of *Name of supervisor* and refers other researcher’s works which are duly listed in the reference section.

The matter presented in this project has not been submitted for the award of any other degree of this or any other university.

(***Name of Candidate)***

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(**Name of Supervisor)**

School of Computing Science and Engineering

Galgotias University

Gr. Noida

## Countersigned by

**(Dr. Parma Nand)**

Professor & Dean

School of Computing Science & Engineering

Galgotias University

Gr. Noida

**ACKNOWLEDGEMENT**

I have taken efforts in this “Data Security in Cloud” project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Mr. Prashant Kumar for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

Sign:

Date:

**ABSTRACT**

. **ABSTRACT**

Cloud computing has been envisioned as the next-generation architecture of IT enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, cloud computing moves the application software and databases to the large data centres, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this article, we focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server (s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

**Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENTS** | **Page No.** |
| 1. | Introduction | 11 |
| 2. | System Overview | 12 |
| 3. | Objective | 12 |
| 3.1 | System objective | 13 |
| 3.2. | Scope | 13 |
| 4. | Feasibility Study | 13 – 16 |
| 5. | Development Environment | 16 – 19 |
| 6. | System Design | 20 – 24 |
| 7. | Implementation | 25 – 28 |
| 8. | Coding | 29 – 51 |
| 9. | Output Screen | 52 – 64 |
| 10. | Methodology | 65-76 |
| 11. | Testing | 77 -82 |
| 11.1 | System Testing | 77 – 78 |
| 11.2 | Performance Testing | 78 - 79 |
| 11.3 | Testing Objective | 81 |
| 11.4 | Test Case | 82 |
| 12. | Conclution | 83 |
| 13. | Bibliography | 84 |

**1. Introduction**

In [computer science](http://en.wikipedia.org/wiki/Computer_science), Secure Transmission refers to the transfer of data such as confidential or proprietary information over a [secure channel](http://en.wikipedia.org/wiki/Secure_channel). Many secure transmission methods require a type of [encryption](http://en.wikipedia.org/wiki/Encryption).

Secure transmissions are put in place to prevent attacks such as [ARP spoofing](http://en.wikipedia.org/wiki/ARP_spoofing) and general [data loss](http://en.wikipedia.org/wiki/Data_loss). Software and hardware implementations which attempt to prevent the unauthorized transmission of information from the computer systems to an organization on the outside.

**Tiny Algorithm**

The Tiny Encryption Algorithm (TEA) is a block cipher encryption algorithm that is very simple to implement, has fast execution time, and takes minimal storage space. The included example is to be compiled and used on a LabVIEW FPGA target.

In [cryptography](http://en.wikipedia.org/wiki/Cryptography), the Tiny Encryption Algorithm (TEA) is a [block cipher](http://en.wikipedia.org/wiki/Block_cipher) notable for its simplicity of description and [implementation](http://en.wikipedia.org/wiki/Implementation), typically a few lines of code. It was designed by [David Wheeler](http://en.wikipedia.org/wiki/David_Wheeler_%28computer_scientist%29) and [Roger Needham](http://en.wikipedia.org/wiki/Roger_Needham) of the [Cambridge Computer Laboratory](http://en.wikipedia.org/wiki/Cambridge_University_Computer_Laboratory); it was first presented at the [Fast Software Encryption](http://en.wikipedia.org/wiki/Fast_Software_Encryption) workshop in [Leuven](http://en.wikipedia.org/wiki/Leuven) in 1994, and first published in the proceedings of that workshop.

The cipher is not subject to any [patents](http://en.wikipedia.org/wiki/Patent).

4. **Feasibility Study**

When complex problem and opportunities are to be defined, it is generally desirable to conduct a preliminary investigation called a feasibility study. A feasibility study is conduct to obtain an overview of the problem and to roughly assess whether feasible solution exists prior to committing substantial resources to a project. During a feasibility study, the system analyst usually works with representatives from the departments(s) expected to benefit from the solution.

Every project is feasible if given unlimited resource and infinite time. Unfortunately, the development of computer based systems is more likely to be plagued by scarcity of resources and difficult delivery of data it is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Precious time and money can be saved and untold professional embarrassment can be averted if an ill conceived system is recognized early in the definition phase. So a detailed study is carried out to check the workability of the system.

Feasibility study is undertaken to evaluate its workability, impact on the organization, ability to meet user needs, and effective se of resources. The main objective of feasibility study is to test the technical, operational and economical feasibility of developing the computer system. Thus, during feasibility analysis for this project the following three primary areas of interest were considered very carefully. The feasibility of a project can be ascertained in terms of technical factors, economic factors, or both. A feasibility study is documented with a report showing all the ramification of the project.

The primary objective of a feasibility study is to assess three types of feasibility.

1. Technical feasibility: can a solution be supported with existing technology?
2. Economical feasibility: is existing technology cost effective?
3. Operational feasibility: will the solution work in the organization if implemented?

**Technical Feasibility**

A systems development project may be regarded as technically feasibility or ‘practical’ if the organization has the necessary expertise and infrastructure to develop, install, operate and maintain the proposed system. Organizations will need to make this assessment based on:

* Knowledge of current and emerging technological solutions.
* Availability of technically qualified staff in house for the duration of the project and subsequent maintenance phase.
* Availability of infrastructure in house to support the development and maintenance of the proposed system.
* Where necessary, the financial and/or technical capacity to procure appropriate infrastructure and expertise from outside.
* Capacity of the proposed system to accommodate increasing levels of use over the medium term and capacity of the proposed system to meet initial performance expectations and accommodate new functionality over the medium term.

The existing computer system has a good hardware configuration and good software facilities in such a way that any alteration can be implemented with slight modifications of the existing process. Hence this project is technically feasible.

**Economic Feasibility**

A systems development project may be regarded as economically feasible or ‘good value’ to the organization if its anticipated benefits outweigh its estimated costs. However, many of the organizational benefits arising from record keeping projects are intangible and may be hard to quantify.In contrasts, many development costs are easier to identify.

These costs may include the time, budget and staff resources invested during the design and implementation phase as well as infrastructure, support, training and maintenance costs incurred after implementation. In these high risk situations it may be appropriate assessments of financial feasibility.

**Operational Feasibility**

A systems development project is likely to be operationally feasible if it meets the ‘needs’ and expectations of the organization. User acceptance is an important determinant of operational feasibility.

**Feasibility study of the proposed system**

The feasibility study of the proposed system has been carried out in all the three areas.

**Technical Feasibility**

The proposed system can be easily developed using resources available in the organization. Hence it is technically feasible.

**Economic feasibility**

The proposed system can be easily developed using the resources available in the organization and they do not invest in procurement of additional hardware or software. The cost of developing the system, including all the phases have been taken into account and it is strict minimum. Hence the system is economically feasible.

**Operational feasibility**

The system has been developed after extensive discussion with the end user and all the operational requirements has been taken into account during the planning and implementation stages. Hence the system is operationally feasible.

5. **Development Environment**

1. **Hardware Configuration**

Processor : Pentium 4 processor

Memory : 1 GB RAM

Display : 14’’ LCD

Hard disk Drive : 80 GB

1. **Software Configuration**

Operating System : Windows XP professional

Environment : Jdk 1.5, Java, Netbeans 6.9

Database : MySql

**4.1.1) NetBeans**

NetBeans is the most comprehensive J2EE IDE() for the open Source netbeans platform.It incorporates most innovative open standard technologies to provide a development environment for J2EE WEB,XML,UML & database & a wide array of application server connectors to streamline development ,deployment, testing & portability.It’s a cross-platform.

**4.1.2) Java:-**

Java is pure object oriented programming language, which has derived C syntax and C++ object oriented programming features. Is compiled and interpreted language and is platform independent and can do graphics, networking, multithreading. It was initially called as OAK. Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. Java can used to create two types of programs: application and applets. Application is a program that runs on your computer, under the operating system of that computer. That is an application created by Java is more or less like one created using C or C++. An applet is an application designed to be transmitted over the Internet and executed by a Java-compatible Web browser. Java provides the Java Virtual Machine (JVM).

Java are following list of buzzwords:-

* Simple
* Secure
* Portable
* Object-oriented
* Robust
* Multithreading
* Architecture-neutral
* Interpreted
* High Performance
* Distributed
* Dynamic

Java supports the different types of editors are EditPlus, Eclips, NetBean, Notepad. EditPlus editor are different types of used in editor such as Text, HTML, PHP, JAVA Script, JAVA, JSP, XML, VBSscript, C#, C/C++, Perl, .NET Config, CSS. Etc. Most of the Eclipse SDK is "pure" JavaTM code and has no direct dependence on the underlying operating system. The chief dependence is therefore on the Java 2 Platform itself. The 3.1 release of the Eclipse Project is written and compiled against version 1.4 of the Java 2 Platform APIs, and targeted to run on version 1.4 of the Java 2 Runtime Environment, Standard Edition

Java technology are performed the different types of version are JDK1.2, JDK1.3, JDK1.4, JDK1.5. etc.

**4.2 Back END:**

**Structure Query Language(SQL)**

A query language for RDBMS based on. Non –procedure approach to retrieve record from RDBMS.

SQL was proposed by IBM and got its standardization by ANSI and adopted by different corporation with bit modification.

SQL can be divided into three categories as given below:

* DML – Data Manipulation Language.
* DCL - Data Control language.
* DDL – Data Definition language
* DML :- Primarily used to retrieve the records from RDBMS
* SELECT [\*|ALL] FROM <TABLE> [WHERE <CONDITION”] <ORDER BY [<FIELD>]
* [HAVING<CONDITION>]
* insert into <table> ( field1, field2, field3 ) values(values1, values2,values3);
* DDL:- Primary used to create tables/indexes etc.
* Create table <table name> (

field name1 type1,

field name2 type2,

field name3 type3

);

* Drop table < table name >;
* DCL:- Primarily used for administrative /option operation like creating if user/assignment of password updation of record/deletion of user/creation of roles/assignment of access right.
* Create user<user name>
* Identified by <password>
* Grant select, insert on EMP to demo;
* Revoke select on EMP from Demo;

In a summarized way it could be concluded that SQL becomes the query engine that resides over the database engine having been designed on the client-server Approach and provided retrieval of data as well as operation on RDBMS. By the Application package and web pages.

**6.System Design**

**Data Flow Diagram**

[**Data flow diagrams**](http://www.edrawsoft.com/Data-Flow-Diagrams.php) illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out to precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

**DataFlow Diagram : (Level-0) :**

**Encription**

**Register/Login**

**Database**

**Server**

**Database**

**Fig 1**. Level 0 DFD

**Level – 1 & 2 DFD :**

Register/Forgot Password

Login

User

Register

Transfer

Server

Encrypt and embded

Dembed and decrypt

**Entity-Relation Ship Diagram**

registration

Reg\_user

User

login

Encrypt/decript/embed/dembed

**7. Implementation**

**Project Description**

The project “Secure Data Transmission” is basically aimed to transfer a file in local area network with security. In this project we use two layer security. i.e cryptography and stenography for secure sensitive data.

**MODULES:**

1. Register
2. Login.
3. Forgot Password
4. Encryption
5. Decryption
6. Embed text file in video fils
7. Dembed text file from video files.
8. Transfer file one system to another system

**Databases Involved**

Databases used in this project are as follows :-

1. Registration
2. Login Details

**8. Sample Coding**

**Database Connection**

package securedata;

import java.sql.PreparedStatement;

import java.sql.Connection;

import java.sql.DriverManager;

public class dbconnection {

static Connection con = null;

PreparedStatement pst = null;

public static Connection makeConnection() {

try {

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

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

} catch (Exception e) {

}

return con;

}

}

**Login**

/\*

\* Login.java

\*

\* Created on Nov 7, 2012, 12:52:40 AM

\*/

package securedata;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

/\*\* Creates new form Login \*/

public Login() {

initComponents();

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

txtuid = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

txtpwd = new javax.swing.JPasswordField();

jLabel4 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Login - Secure Data Transmission");

jPanel1.setBackground(new java.awt.Color(204, 204, 204));

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 30)); // NOI18N

jLabel1.setText("Login");

jPanel2.setBackground(new java.awt.Color(204, 204, 204));

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "Login", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Adobe Caslon Pro", 0, 11))); // NOI18N

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Login");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton3.setText("Register");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel3.setText("Password");

txtuid.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel2.setText("User ID");

txtpwd.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1))

.addComponent(txtuid, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(txtpwd, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 193, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(17, 17, 17)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(txtuid, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2))

.addGap(33, 33, 33)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(3, 3, 3)

.addComponent(txtpwd, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton3))

.addContainerGap())

);

jPanel2Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new java.awt.Component[] {jButton1, jButton3});

jLabel4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/securedata/login.jpg"))); // NOI18N

jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton2.setText("Forgot Password");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jButton2)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 259, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(33, 33, 33)

.addComponent(jLabel1))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(27, 27, 27)

.addComponent(jLabel1)

.addGap(27, 27, 27)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jButton2)

.addContainerGap(38, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Register reg = new Register();

dispose();

reg.show();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String login = null;

String pwd = null;

login = txtuid.getText();

if (login.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Please enter your user id.");

return;

}

pwd = txtpwd.getText();

if (pwd.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Please enter your password.");

return;

}

try {

Connection con = dbconnection.makeConnection();

PreparedStatement ps = con.prepareStatement("select \* from registration where userid=? and password=?");

ps.setString(1, login);

ps.setString(2, pwd);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

//insert into login history

String sql = "insert into login\_history(uname,login\_date,login\_time) values(?,now(),now())";

ps = con.prepareStatement(sql);

ps.setString(1, login);

ps.executeUpdate();

//send profile page

MainForm cpi = new MainForm();

dispose();

cpi.show(true);

} else {

javax.swing.JOptionPane.showMessageDialog(rootPane, "User id ans password are wrong.!Re-try");

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Forgotpassword reg = new Forgotpassword();

dispose();

reg.show();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPasswordField txtpwd;

private javax.swing.JTextField txtuid;

// End of variables declaration

}

}

**User Registration**

/\*

\* Register.java

\*

\* Created on Nov 7, 2012, 12:54:58 AM

\*/

package securedata;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

public class Register extends javax.swing.JFrame {

/\*\*

\* Creates new form Register

\*/

public Register() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

txtname = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

txtage = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

txtaddress = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

txtuid = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

txtskey = new javax.swing.JTextField();

txtpwd = new javax.swing.JPasswordField();

jLabel7 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Register - Secure Data Transmission");

jPanel1.setBackground(new java.awt.Color(204, 204, 204));

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 30)); // NOI18N

jLabel1.setText("Register");

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel2.setText("Name");

txtname.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel3.setText("Age");

txtage.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel4.setText("Address");

txtaddress.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel5.setText("User ID");

txtuid.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel6.setText("Password");

txtskey.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

txtpwd.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel7.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel7.setText("Secret Key");

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Submit");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton2.setText("Go to Login");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(68, 68, 68)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 358, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 366, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel4)

.addComponent(jLabel7))

.addGap(30, 30, 30)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(txtage)

.addComponent(txtname)

.addComponent(txtaddress)

.addComponent(txtuid)

.addComponent(txtpwd)

.addComponent(txtskey)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1)))))))

.addGap(83, 83, 83))

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jButton1, jButton2});

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(36, 36, 36)

.addComponent(jLabel1)

.addGap(34, 34, 34)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(txtname, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel3)

.addComponent(txtage, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel4)

.addComponent(txtaddress, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(36, 36, 36)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(txtuid, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(34, 34, 34)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(txtpwd, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(33, 33, 33)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtskey, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel7))

.addGap(40, 40, 40)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addContainerGap(50, Short.MAX\_VALUE))

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new java.awt.Component[] {jButton1, jButton2});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Login lg = new Login();

dispose();

lg.show();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String name = null;

String address = null;

String age = null;

String login = null;

String pwd = null;

String skey = null;

name = txtname.getText();

if (name.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Please enter your name.");

return;

}

login = txtuid.getText();

if (login.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Please enter your user id.");

return;

}

pwd = txtpwd.getText();

if (pwd.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Please enter your password.");

return;

}

skey = txtskey.getText();

if (skey.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Please enter your seckret key.");

return;

}

//check user availability

try {

Connection con = dbconnection.makeConnection();

PreparedStatement ps = con.prepareStatement("select \* from registration where userid=?");

ps.setString(1, login);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

javax.swing.JOptionPane.showMessageDialog(rootPane, "" + login + " is Available.!Please enter diffrent users.");

txtuid.setText("");

return;

}

} catch (Exception e) {

e.printStackTrace();

}

age = txtage.getText();

address = txtaddress.getText();

try {

Connection con = dbconnection.makeConnection();

String sql = "insert into registration(userid,password,secretkey,name,age,address,reg\_date) values(?,?,?,?,?,?,now())";

PreparedStatement ps = con.prepareStatement(sql);

ps.setString(1, login);

ps.setString(2, pwd);

ps.setString(3, skey);

ps.setString(4, name);

ps.setString(5, age);

ps.setString(6, address);

int i = ps.executeUpdate();

if (i > 0) {

JOptionPane.showMessageDialog(rootPane, "Your information has been successfully submited.");

txtname.setText("");

txtage.setText("");

txtaddress.setText("");

txtuid.setText("");

txtpwd.setText("");

txtskey.setText("");

} else {

JOptionPane.showMessageDialog(rootPane, "Your information has not been submited.! Try-agin");

}

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(rootPane, "Problem in database.");

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Register().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

private javax.swing.JTextField txtaddress;

private javax.swing.JTextField txtage;

private javax.swing.JTextField txtname;

private javax.swing.JPasswordField txtpwd;

private javax.swing.JTextField txtskey;

private javax.swing.JTextField txtuid;

// End of variables declaration

}

**Stenography**

package securedata;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

public class EmbProcess {

String embfilename;

public String emb(String s, String s1) {

try {

File file = new File(s);

File file1 = new File(s1);

FileInputStream fileinputstream = new FileInputStream(s);

FileOutputStream fileoutputstream = new FileOutputStream("temp");

byte abyte0[] = new byte[8];

int i;

int k;

for (k = 0; (i = fileinputstream.read(abyte0, 0, 8)) > 0; k = i) {

fileoutputstream.write(abyte0, 0, i);

}

fileinputstream.close();

for (int l = 1; l <= 8 - k; l++) {

fileoutputstream.write(65);

}

fileoutputstream.write("DATAFILE".getBytes(), 0, 8);

System.out.println("File name===" + file.getName());

StringBuffer stringbuffer = new StringBuffer(file1.getName());

stringbuffer.setLength(50);

fileoutputstream.write(stringbuffer.toString().getBytes(), 0, 50);

fileinputstream = new FileInputStream(s1);

int j;

while ((j = fileinputstream.read(abyte0, 0, 8)) > 0) {

fileoutputstream.write(abyte0, 0, j);

}

fileinputstream.close();

fileoutputstream.close();

file.delete();

File file2 = new File("temp");

file2.renameTo(file);

embfilename = file.getName();

} catch (Exception e) {

e.printStackTrace();

embfilename = "";

}

return embfilename;

}

public String demb(String s) {

boolean flag;

String demfile = "";

try {

File file = new File(s);

String outpath = s.substring(0, s.lastIndexOf("\\") + 1);

FileInputStream fileinputstream = new FileInputStream(s);

char c = '\b';

byte abyte0[] = new byte[c];

String s1 = "";

int i;

while ((i = fileinputstream.read(abyte0, 0, c)) > 0) {

s1 = new String(abyte0);

if (s1.equals("DATAFILE")) {

break;

}

}

if (!s1.equals("DATAFILE")) {

flag = false;

fileinputstream.close();

return demfile;

}

abyte0 = new byte[50];

fileinputstream.read(abyte0, 0, 50);

s1 = new String(abyte0);

String s2 = s1.trim();

String fpath = s2.substring(0, s2.lastIndexOf(".") + 1) + "enc";

System.out.println("fpath------" + fpath);

FileOutputStream fileoutputstream = new FileOutputStream(outpath + fpath);

c = '\u5000';

abyte0 = new byte[c];

while ((i = fileinputstream.read(abyte0, 0, c)) > 0) {

fileoutputstream.write(abyte0, 0, i);

//fileinputstream.delete(abyte0, 0, c);

}

fileinputstream.close();

fileoutputstream.close();

demfile = fpath;

} catch (Exception exception) {

demfile = "";

exception.printStackTrace();

System.out.println(exception);

}

return demfile;

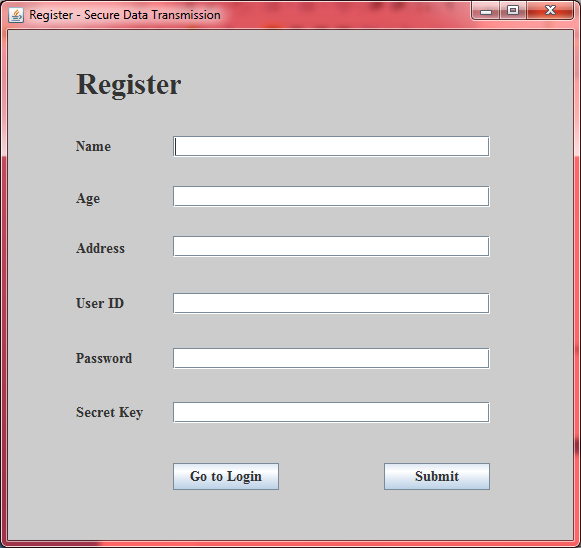
}

}

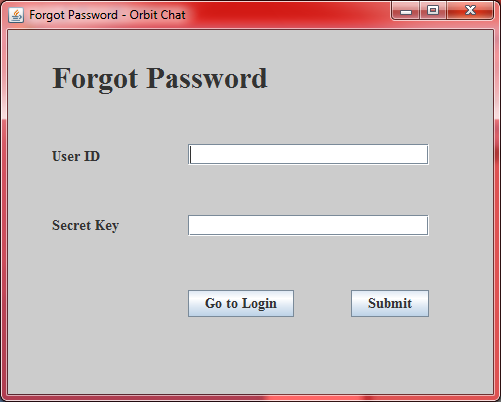
1. **Secure Data Transmission**

****

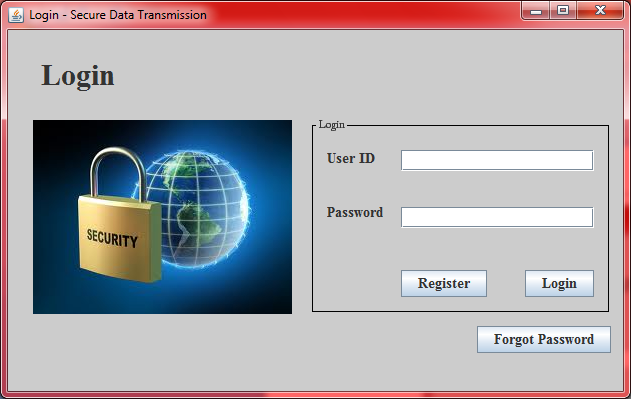
Welcome Page

****

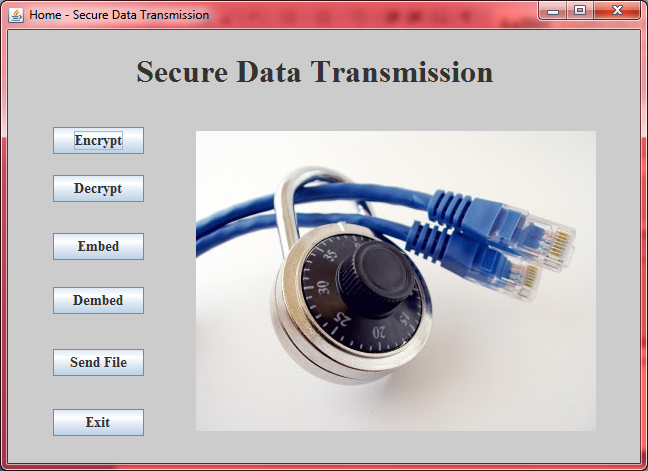
**Registration Form**

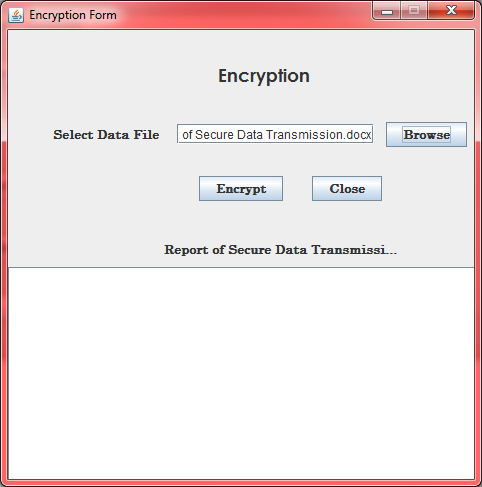
****

**Forgot Password**

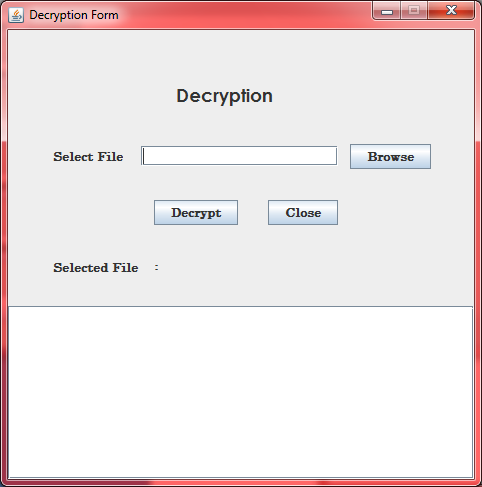
****

**Login**

**Home Page**

****

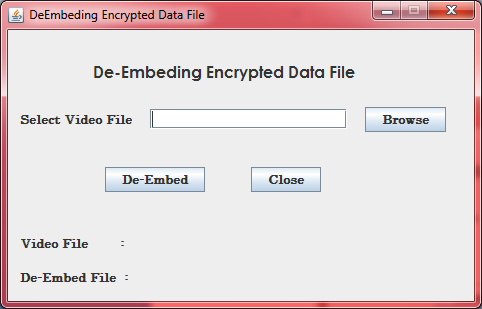
**Encryption**

****

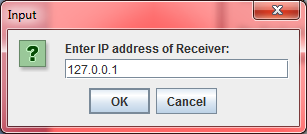
**Decryption**

****

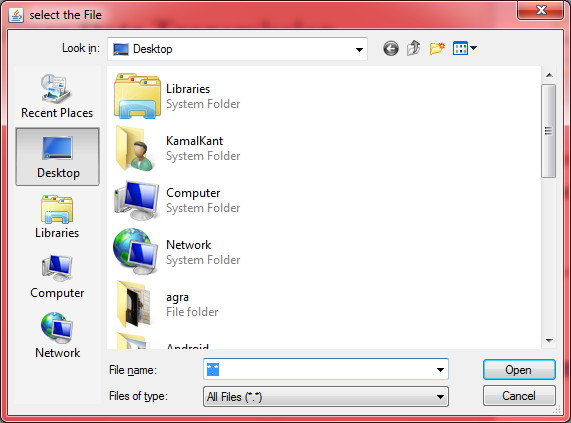
**Embeding**

****

**De-Embeding**

****

**IP Address for sending a file**

****

**Select a file for sending**

**10. Technical Specification**

**What is a Methodology?**

Software engineering is carry out of using preferred procedure techniques to progress the quality of a software development effort. A methodology is defined as a collection of procedures, techniques, tools, and documentation aids which will help developers in their efforts (both product and process related activities) to implement a new system. For successful implementation, a well-organized and systematic approach is crucial. Therefore, several methodologies were developed to encourage the systematic approach to planning, analysis, design, testing and implementation. Methodologies offer various tools and techniques to assist in analysis, design and testing in terms of detailed design of software, data flowcharts and database design.

**Why Methodology?**

1. To complete a project within time and budget with the expected scope and quality we need methodologies which provide for a framework.
2. Most methodologies have a general planning, developing and managing stages in common. They suggest the development team the ways of thinking, learning and arriving at a regular feasible solution.

To select an ideal methodology was based on project requirements and goals.

* Functional Decomposition: The methodology should have stages according to the interrelated activities which can be grouped into different functional areas.
* Requirement Changes: If required, methodology provides scope to change the requirement.
* Manage Risks: Determined the risk is an important activity to develop a project.
* Iterative approach: Iteration allows refinement of requirement as well as design.
* Documentation: Methodology provides support for large documentation.
* Analysis and Design Support: A well defined structure of the methodology helps for analysis and designing to development process..
* Implementation: The system should be implemented as per plan.
* Testing Support: More testing, more reliable the product is.
* Object Oriented Approach: Object oriented concepts will be used in developing the project as it supports component reusability.

**Suitable Methodologies:**

**Waterfall Methodology**: All projects can be managed better when segmented into a hierarchy of chunks such as phases, stages, activities, tasks and steps. It follows a linear structure starting from requirement analysis, through design, implementation and maintenance. Most widely accepted methodology for student projects, this model has been well tried and tested. Each phase of it has sub phases which produce deliverables. Requirements are fixed at initial stages before proceeding with development plans in system development projects; the simplest rendition of this is called the "waterfall" methodology, as shown in the following figure:

Fig 2: waterfall model 

The graphic illustrates a few critical principles of a good methodology:

* Work is done in stages,
* Content reviews are conducted between stages, and
* Reviews represent quality gates and decision points for continuing.

The waterfall provides an orderly sequence of development steps and helps ensure the adequacy of documentation and design reviews to ensure the quality, reliability, and maintainability of the developed software.  While almost everyone these days disparages the "waterfall methodology" as being needlessly slow and cumbersome, it does illustrate.

**Conclusion of Methodology Research**

Each methodology was evaluated against set criteria and performance rating was given. The evaluation proved RUP as a well disciplined industry standard approach that perfectly matches the set criteria. Apart from allowing flexibility in requirement analysis, it also provides strong support for object oriented analysis and design.

**WHY NOT WATERFALL MODEL**

* This model needs all requirements explicitly, but it is frequently not easy for the customer to state all requirements clearly.
* The model tends to consume a lot more time compared to other software advance models though it is able to identify exact starting and ending points for a given project.
* Waterfall model does not support iterative approach.
* In this model developer cannot change the requirement in the middle of the project. If changes are tried to be incorporated it leads to more confusion and further delays.
* It defers testing and integration until the end of development lifecycle resulting in unnecessary risks.

**WHY NOT SPIRAL MODEL**

* Requires considerable expertise in risk evaluation and reduction.
* Complex and relatively difficult to follow strictly.
* It has lack of risk management experience as well as lack of milestones and management is unsure of spiral process.

**11. Testing**

The testing process focuses on the internal logic of the software assuring that all the statements have been tested and also on the functional external by conducting tests to uncover the errors. The amount of testing required related to the size and complexity of the application.

Before testing any system test cases for each screen should be prepared. The test cases are made to check and execute all the functionalities of “Remote Desktop”. The following are the tests conducted for this system:

**System Testing**

Software testing entails running software products under known conditions with defined inputs and documented outcomes that can be compared to their predefined expectations. It is a time consuming, difficult, and imperfect activity. As such, it requires early planning in order to be effective and efficient.

Test plans and test cases should be created as early in the software development process as feasible. They should identify the schedules, environments, resources (personnel, tools, etc.), methodologies, cases (inputs, procedures, outputs, expected results), documentation, and reporting criteria. The magnitude of effort to be applied throughout the testing process can be linked to complexity, critically, reliability, and/or safety issues (e.g., requiring functions or modules that produce critical outcomes to be challenged with intensive testing of their fault tolerance features).

Implementation of software testing and its implications with respect to software quality cannot be over emphasized. Software testing is a critical element of the software quality assurance and represents the ultimate review of specification, design and coding.

A software testing process should be based on principles that foster effective examinations of a software product. Applicable software testing tenets include:

* The expected test outcome is predefined.
* A good test case has a high probability of exposing an error.
* A successful test is one that finds an error.
* There is independence from coding.
* Both application(user) and software (programming) expertise are employed.
* Testers use different tools from coders
* Examining only the usual case is insufficient.

In order to provide a through and rigorous examination of a software product, development testing is typically organized into levels. As an example, a software product’s testing can organized into unit, integration, and system levels of testing.

**Content Testing**

Content testing checks that users can easily understand all items that appear on a interface and, importantly, that they are able to correctly interpret the information they read. Accordingly, it identifies any problem areas that may require simplification or re-writing.

**Interface Testing**

Testing conducted to ensure that the end users enter appropriate data.

**Navigation Testing**

Once the first layers of the interface have been examined, it is time to perform simple tests to probe the quality of the global site navigation. Good sites will provide consistent, well-executed navigation and should provide alternative navigation schemes.

**Security Testing**

Techniques used to confirm the design and/or operational effectiveness of security controls implemented within a system. Examples: Attack and penetration studies to determine whether adequate controls have been implemented to prevent breach of system controls.

**Unit Testing**

Unit (module or component) level testing focuses on the early examination of sub-program functionality and ensures that functionality not visible at the system level is examined by testing. Unit testing ensures that quality software units are furnished for integration into the finished software product.

**Integration Testing**

Integration level testing focuses on the transfer of data and control across a program’s internal and external interfaces. External interfaces are those with other software, system hardware, and the users and can be described as communications links.

**Whitebox Testing**

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Some of the basic white box test cases were cyclomatic complexity and graph matrices.

**Testing and Design**

The development team must plan for testing when they create an application’s design. The team should recognize that the product eventually will be tested for validity and consistency. The verification process uses standard testing procedures (for examples regression testing) that the team can plan for.

The design work generally reveals potential problems such as running out of disk space or a possible failure of the system services. If such problems cannot be “designed out” of the system, having a record of them can be valuable when the team prepares its tests.

The team writes the set of functional validation tests during preliminary design to ensure that the tests measure the functions the software is to perform. If the team writes the tests after the code is written. Their knowledge of the code may affect the way they write the tests. This approach diminishes the objectivity of the test and the value of the results.

Functional tests are usually black box tests and can be written based on the requirements, specifications, and design documents. White box tests can be written at the design phase only if the design documents are quite detailed.

**Performance Testing**

Performance testing helps ensure that a product performs its functions at the required speed. Planning for performance testing starts at the beginning of the project when product goals and requirements are defined. Performance testing is a part of the product’s initial engineering paln.

The development team can approach performance testing is one of the three ways:

* The team can design for performance. Techniques such as modeling and prototyping help to assess the application’s performance. Techniques for validating designs produce an application design that can enhance performance.
* The team can test performance during development. This approach entails testing performance at the unit level. The team writes tests and establishes performance for each unit tested. The drawback to this approach is the significant time and effort required.
* The development team can test the performance of the finished product. This approach also requires tests and benchmarks. However, the team creates the tests and benchmarks only for the full application. For this reasons, it is more practical than testing during development.

Assuming that the team selects the third approach, the developers must first create the tests and benchmarks. This step is difficult because it entails translating information from the requirements and specification stages into tests and benchmarks that are specific to individual products. Without meaningful benchmarks, performance testing serves little purpose.

Depending upon the project and purchaser’s requirements, appropriate testing method is selected. All testing strategies are having advantages as well as disadvantage. Further, there are several tests to be performed before delivering a software system:

* Function Testing
* Performance Testing
* Acceptance Testing
* Installation Testing

The function test checks that the integrated system performs its functional requirements as specified in the requirements. After this, performance test compares the integrated modules with the non-functional requirements. It includes speed, accuracy, security and reliability etc. The acceptance test checks the system characteristics to assure that they are in compliance with the users to exercise the system’s functions at the working environment. The errors occurred during the test are records and rectified it immediately before cleaning for each stage.

**Testing Objective**

The objectives for the testing have to be set keeping in mind the s type of the system. The objectives are:

1. The input values have to be validated for the aeriable type and size specified in the data base.
2. Secured pages should be accessed only trough session.
3. Logical sequence in navigation has to be maintained.

**Test Case**

A test case is as set of conditions or variables under which a tester will determine if a requirement or use case upon an application is partially or fully satisfied. It may take many test cases to determine that a requirement is fully satisfied. In order to fully test that all the requirements of and application met, there must be at least one test case for each requirement unless a requirement has sub requirement. In that situation, each sub requirement must have at least one test case.

The written test case is that there is known input and an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post condition test cases uncover the following categories:

* Erroneous initialization or default values and inconsistent data types
* Incorrect (misspelled or truncated) variable name
* Underflow, overflow and addressing exceptions

**Conclusion**

This Desktop Application provides facility to transfer sensitive file from one user to another user through security system. This project work on client server architecture that means if server is start then server receive client file.

**BIBLOGRAPHY**

[1] Core Java – vol1- Hall & Brown (2nd Edition).

[2] Core Java 2 Volume 1 - Fundamentals Cay S. Horstmann, Gary Cornell

[3] Core Java 2 Volume 2 - Advanced Features Cay S. Horstmann, Gary

Cornell