

**TheOrchid School, Pune**

A Project Report

on

**INSTANT MESSENGER**

For

AISSCE 2019 Examination

[As a part of the Informatics Practices Course (065)]

SUBMITTED BY:

Mittul Daswani

Shubham Dandawate

Anooj Deshpande

Under the Guidance of:

Mrs. Jayashri Samudra

**CERTIFICATE**

This is to certify that the Project entitled **Instant Messenger** is a bonafide work done by Miss Manali Shukla of class XII session 2018-19 in partial fulfillment of CBSE’s Examination 2019 and has been carried out under my direct supervision and guidance.

This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

## ………………………… … …………………………….Signature of Student Signature of Teacher/Guide

**Name: ……………….. Name: Ms. Jayashri Samudra**

*……………………………. ……………………………*

**Signature of Unit Head Signature of Principal**

**Name: Ms. Netre Kulkarni Name: Ms. Namrata Majhail**

**…………………………..**

**Signature of External Examiner**

**Place: The Orchid School, Pune Date:……..………..**

**ACKNOWLEDGEMENT**

I

undertook this Project work, as the part of my XII-Informatics Practices course. I tried to apply my best of knowledge and experience, gained during the study and class work experience. However, developing software system is generally a quite complex and time-consuming process. It requires a systematic study, insight vision and professional approach during the design and development.

Moreover, the developer always feels the need, the help and good wishes of the people near you, who have considerable experience and idea.

I would like to extend my sincere thanks and gratitude to my teachers

Ms. Jayashri Samudra for guiding us throughout the life cycle of Software Development**.**

I would like to take the opportunity to extend my sincere thanks and gratitude to my Unit Head Ms. Netre Kulkarni for giving valuable time and moral support to develop this software.

I am very much thankful to our Principal Ms. Namrata Majhail for being a source of inspiration.

**Class XII**

**C O N T E N T S**

1**. Introduction** 5

**2. Objective & Scope of the Project** 6

**3. Problem Definition & Analysis** 7

**4. System Implementation** 8

4.1 The Hardware used: 8

4.2 The Softwares used: 8

**5. System Design & Development** 9

5.1 Database Design: 9

5.2 Menu Design: 10

5.3 I/O Forms Design & Event Coding: 10

**6. User Manual** 33

6.1 How to install: 33

6.2 Working with Software: 34

**7. References**  37

**1. Introduction**

This software project is developed to automate the functionalities of a Instant Messenger. The purpose of the software project is to develop the Management Information System (MIS) to automate the functioning of a instant messenger capable of sending, receiving messages. Apart from that new user accounts can be added.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users.

During coding and design of the software Project, Java NetBeans IDE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. As a back-end a powerful, open source RDBMS, My SQL is used as per requirement of the CBSE curriculum of Informatics Practices Course.

**2. Objective & Scope of the Project**

T

he objective of the software project is to develop a computerized MIS to automate the functions of a messenger This software project is also aimed to enhance the current record keeping system, which will help users to retrieve the up-to-date information.

The proposed software system is expected to do the following functionality-

* To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for MIS activities.
* The proposed system should maintain all the records , and should generate the required reports and information when required.
* To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
* To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

In its current scope, the software enables user to retrieve and update the information from centralized database designed with MySQL . This software does not require much training time of the users due to limited functionality and simplicity.

During the development of this project, Java NetBeans IDE, a powerful, open source event-driven form-based development environment is used for modular design and future expandability of the system.

Despite of the best effort of the developer, the following limitations and functional boundaries are visible, which limits the scope of this application software.

1.This is just a model of instant messenger it don’t work over internet as an instant messenger but functions over database to save , retrieve messages and user records.

2. This software can only send messages, can receive messages but is not capable of sending objects like images, video etc.

3. Some application area of a messenger like adding friends, checking validity of an email, isn’t a part of this project.

So far as future scope of the project is concerned, firstly it is open to any modular expansion i.e. other modules or functions can be designed and embedded to handle the user need in future. Any part of the software and reports can be modified independently without much effort.

**3. Problem Definition & Analysis**

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the *what* of a system, not *how*. The quality of a software product is only as good as the process that creates it. Problem Definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem Definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

* Identification and Documentation of customer’s or user’s needs.
* Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
* Analysis and Validation of the requirements documents to ensure consistency, completeness, and feasibility
* Evolution of needs.

After the analysis of the functioning of a Restaurant, the proposed System is expected to do the following: -

* To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for a computerized Restaurant.
* The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
* To provide efficient and secured Information storage, flow and retrieval system, ensuring the integrity and validity of records.
* To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
* To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

**4. System Implementation**

## 

## 4.1 The Hardware used:

While developing the system, the hardware used was:

## PC with Intel Core i5 with 4 GB RAM

## 4.2 The Software’s used:

* Microsoft Windows® 10 as Operating System.
* Java NetBeans 8.2 as Front-end Development environment.
* MySQL 5.7 as Back-end Sever with Database for Testing.
* MS-Word for documentation.

**5. System Design & Development**

## 

## 5.1 Database Design:

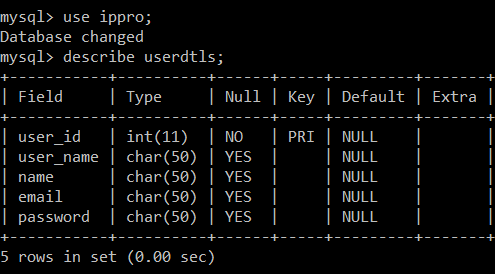
An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or constraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.

This software project maintains a database named **MANALI** which contains the following tables.

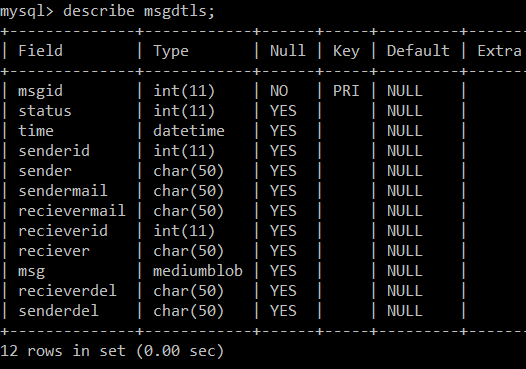
**Table Design:**

**.Table : Userdtls**

**Table Design:**

****

**2.Table : Msgdtls**

****

**6.3 I/O Forms Design & Event Coding:**

The software project for Instant Messenger contains various forms along with programming codes. Forms (JFrames) and their event coding are given below.

**Frame: NewJFrame.java**

import java.sql.\*

import javax.swing.DefaultListModel;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

***public class NewJFrame extends javax.swing.JFrame*** {

Connection con = null;

Statement stmt = null;

ResultSet rs = null;

String query, mailfrom = null , usrnm, pass, mail;

int usrid=0;

***public NewJFrame()*** {

initComponents();

}

***private void startpagebtnActionPerformed(java.awt.event.ActionEvent evt)*** {

login.setVisible(true);

dispose();

}

***Privatevoid create userbtn\_signupActionPerformed(java.awt.event.ActionEvent evt)*** {

usrnm = inputusrname\_signup.getText();

String passwrd = new String(inputpass\_signup.getPassword()),

name = inputnametxt\_signup.getText(),

email = usrnm+"@kvucmail.in",

retypepasswrd = new String(inputretypepass\_signup.getPassword());

if(usrnm!=null && name!=null && passwrd!=null){

int x; //check whether usernm already exists

try {

query = "select count(\*) from userdtls where user\_name='"+usrnm+"'";

rs = sqlexecutequery(query);

rs.next();

x=rs.getInt("count(\*)"); //==> if >0 not valid

if (x==0) {

if(retypepasswrd.equals(passwrd)) {

query = "select max(user\_id)+1 from userdtls";

rs = sqlexecutequery(query);

rs.next();

usrid = rs.getInt("max(user\_id)+1");

usridlbl\_signup.setText("User ID :" + usrid);

query = "insert into userdtls values ("+ usrid + ","+ "'" + usrnm + "'," + "'" +name + "'," +"'" + email +"'," +"'" + passwrd + "'"+ ");";

sqlupdatequery(query);

JOptionPane.showMessageDialog(null, "User Account Created successfully"+'\n'+

" Please note down your user ID: "+ usrid+'\n'+"and your email:" +'\n'+email); inputusermail\_signup.setText(""+email); usridlbl\_signup.setText("Userid"+usrid);

createuserbtn\_signup.setEnabled(false);

}

else{

JOptionPane.showMessageDialog(null, "Your Passwords Don't Match");

}

}

else

{

JOptionPane.showMessageDialog(null, "User name already exists");

}

dispose(); }

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

} }

Photo to be put.

***private void clr\_signupActionPerformed(java.awt.event.ActionEvent evt)*** { createuserbtn\_signup.setEnabled(true);

inputpass\_signup.setText("");

inputusrname\_signup.setText("");

inputnametxt\_signup.setText("");

usridlbl\_signup.setText("User ID");

inputretypepass\_signup.setText("");

inputusermail\_signup.setText("");

}

**Frame: login[JFrame]**

***private void newuserbtn\_loginActionPerformed(java.awt.event.ActionEvent evt)*** {

signup.setVisible(true);

inputusermail\_signup.setText("");

inputpass\_signup.setText("");

inputusrname\_signup.setText("");

inputnametxt\_signup.setText("");

usridlbl\_signup.setText("User ID");

inputretypepass\_signup.setText("");

createuserbtn\_signup.setEnabled(true);

dispose(); }

***private void loginbtn\_loginActionPerformed(java.awt.event.ActionEvent evt)*** {

usrnm = usrnmtxt\_login.getText();

pass = new String(usrpass\_login.getPassword());

if (usrnm.equals("msgdtls") && (pass.equals("kvuc"))) {

msgdtlspg.setVisible(true); //msgpagelogin

}

else { //userpagelogin

try {

query = "select user\_id, password, email, user\_name, name from userdtls where user\_name =" + "'" + usrnm + "'" + ";";

rs = sqlexecutequery(query);

rs.next();

String passwrd = rs.getString("password");

if (pass.equals(passwrd)) { /\*login successfull\*/

usrid=rs.getInt("user\_id");

usrnm=rs.getString("user\_name");

mail=rs.getString("email");

JOptionPane.showMessageDialog(null, "Login Successfull");

useridlabel\_accsttng.setText("" + rs.getInt("user\_id"));

username\_welcome.setText(rs.getString("user\_name"));

txtnml\_accsttng.setText(usrnm);

emaillbll\_accsttng.setText(rs.getString("email"));

nm\_accsttng.setText(rs.getString("name"));

fromlabel\_write.setText(rs.getString("email"));

welcome.setVisible(true);

} }

catch (Exception e) { JOptionPane.showMessageDialog(null,e.getMessage());}

}

login.dispose();

}

**Frame: login[JFrame]**

***private void Logout3\_welcomeActionPerformed(java.awt.event.ActionEvent evt)*** {

logout(); }

***private void accsttngbtn\_welcomeActionPerformed(java.awt.event.ActionEvent evt)*** {

accsttng.setVisible(true); }

***private void writebtn\_welcomeActionPerformed(java.awt.event.ActionEvent evt)*** {

write.setVisible(true);

totxt\_write.setText("");}

***private void inboxbtn\_welcomeActionPerformed(java.awt.event.ActionEvent evt)*** {

inbox.setVisible(true);

msgtextarea\_inbox.removeAll();

int status1 = 0;

DefaultListModel x = ( DefaultListModel) msglist\_inbox.getModel();

DefaultListModel y = (DefaultListModel) msgidlist\_inbox.getModel();

x.removeAllElements();

y.removeAllElements();

try {

query = "Select msgid, senderid , recieverid, time, msg, status , reciever , sender , sendermail "+ "from msgdtls where recieverid= " + usrid +" AND recieverdel = 'no' " +";";

rs = sqlexecutequery(query);

while (rs.next()){

x.add(0,rs.getString("Sender")); y.add(0,rs.getInt("msgid"));

status1= status1 + rs.getInt("status");//check status }

msglist\_inbox.setModel(x); msgidlist\_inbox.setModel(y);

if (status1 != 0) { //msg for new message

JOptionPane.showMessageDialog(null, "you have recieved" + status1 + " new messages");

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());}

}

***private void sentbtn\_welcomeActionPerformed(java.awt.event.ActionEvent evt)*** {

sent.setVisible(true);

try {

DefaultListModel x = ( DefaultListModel) sentmsglist\_sent.getModel();

DefaultListModel y = (DefaultListModel) sentmsgidlist\_sent.getModel();

x.removeAllElements();

y.removeAllElements();

query = "Select msgid, recieverid, time, msg, status , reciever , sender, recievermail, sendermail from msgdtls where sendermail = '"+ mail +"'"+

" and senderdel = 'no' "+";" ;

rs = sqlexecutequery(query);

while (rs.next()){

x.add(0,rs.getString("reciever"));

y.add(0,rs.getInt("msgid"));

}

sentmsgarea\_sent.setText("");

sentmsglist\_sent.setModel(x);

sentmsgidlist\_sent.setModel(y);

sentmsgidlist\_sent.setVisible(false);

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

}

***private void statsbtn\_welcomeActionPerformed(java.awt.event.ActionEvent evt)*** {

msgcounter.setVisible(true);

int avgcharsent ,avgcharrecieved;

try {

query="select count(\*), sum(length(msg)) from msgdtls where senderid = "+ usrid+ ";";

rs= sqlexecutequery( query);rs.next(); //max msg send

msgsent\_msgcounter.setText(""+rs.getInt("count(\*)")); //avg char in msg sent

if(rs.getInt("count(\*)")!=0)

{

avgcharsent= rs.getInt("sum(length(msg))") / rs.getInt("count(\*)");

}

else {

avgcharsent=0;

}

avgsentmsgchar\_msgcounter.setText(""+avgcharsent);

query="select count(\*), sum(length(msg)) from msgdtls where recieverid= "+ usrid+";";

rs= sqlexecutequery( query);rs.next();//max msg recieved

msgrecieved\_msgcounter.setText(""+rs.getInt("count(\*)")); //avg char in recv. msg

if(rs.getInt("count(\*)")!=0) {

avgcharrecieved=rs.getInt("sum(length(msg))") /rs.getInt("count(\*)");

}

else {

avgcharrecieved=0;

}

avgcharmsgrecieved\_msgcounter.setText(""+avgcharrecieved);

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

}

**Frame: accsttng [JFrame]**

***private void Logout7\_accsttngActionPerformed(java.awt.event.ActionEvent evt)*** {

logout(); }

***private void Updatel\_accsttngActionPerformed(java.awt.event.ActionEvent evt)*** {

pass = new String(oldpassl\_accsttng.getPassword());

String npass = new String(newpassl\_accsttng.getPassword()), nm= nm\_accsttng.getText();

try {

query = "select user\_id, password, email from userdtls where user\_name =" +"'" +usrnm + "'"+";";

rs = sqlexecutequery(query); rs.next();

if (pass.equals(rs.getString("password"))) { //check old password

query = "update userdtls set " + "user\_name = '" + usrnm + "'"

+ ",name = '" + nm + "'" + ",password = '" + npass + "'"

+ ", email= '" + emaillbll\_accsttng.getText() + "'"

+ " where user\_id = " + usrid;

sqlupdatequery(query);

JOptionPane.showMessageDialog(null, "Updated Successfully!");

}

else {

JOptionPane.showMessageDialog(null, "Please enter a valid password");

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}}

***private void Logout5\_writeActionPerformed(java.awt.event.ActionEvent evt)*** {

logout();}

***privatevoidClose\_writeActionPerformed(java.awt.event.ActionEvent evt)*** {

write.dispose();

}

***private void sendbtn\_writeActionPerformed(java.awt.event.ActionEvent evt)*** {

String mailto, sendermail, reciever, time, msg, sender, recieveremail;

int msgid, senderid, recieverid, status= 1; /\*yet not read\*/

mailto = totxt\_write.getText();

sendermail = fromlabel\_write.getText();

senderid = usrid;/\*senderid\*/

sender = usrnm;/\*sender name\*/

msg = msgarea\_write.getText();/\*msg\*/

if(msg.equals("") && mailto.equals(""))

{

JOptionPane.showMessageDialog(null, "Can't send a blank mail");}

else{

try {

query = "select max(msgid) from msgdtls";

rs = sqlexecutequery(query);

rs.next();

msgid = rs.getInt("max(msgid)") + 1;/\*msgid\*/

query = "select user\_id, user\_name , email from userdtls where email= '" + mailto + "'";

rs = sqlexecutequery(query);

rs.next();

reciever = rs.getString("user\_name");/\*reciever\*/

recieverid = rs.getInt("user\_id");/\*reciever id\*/

recieveremail = rs.getString("email");/\*reciever email\*/

query = "select now()"; /\*time\*/

rs = sqlexecutequery(query);

rs.next();

time = rs.getString("now()");

query = "insert into msgdtls values ("

+ msgid + "," + status

+ ", '" + time + "'" + "," + senderid + ",'" + sender + "'"

+ ",'" + sendermail + "'"+ "," + recieverid

+ ",'" + reciever + "'"+ ",'" + msg + "'"

+ ",'" + recieveremail + "'," + " 'no', 'no' );";

sqlupdatequery(query);

msgarea\_write.setText("");

totxt\_write.setText("");

JOptionPane.showMessageDialog(null, "Message Sent Successfully");

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

}

}

**Frame: Sent[JFrame]**

***private void Logout6\_sentActionPerformed(java.awt.event.ActionEvent evt)***{

logout();

}

***private void Close\_sentActionPerformed(java.awt.event.ActionEvent evt)***{

sent.dispose();

}

***private void delsentmsg\_sentActionPerformed(java.awt.event.ActionEvent evt)*** {

DefaultListModel dlm1 =(DefaultListModel)sentmsglist\_sent.getModel();

DefaultListModel dlm2 =(DefaultListModel)sentmsgidlist\_sent.getModel();

Object msgid = sentmsgidlist\_sent.getSelectedValue();

if (msgid !=null) {

if (JOptionPane.showConfirmDialog(null, "Are you sure to Delete") ==

JOptionPane.YES\_OPTION) {

dlm1.remove(sentmsglist\_sent.getSelectedIndex());

dlm2.remove(sentmsgidlist\_sent.getSelectedIndex());

sentmsgarea\_sent.setText("");

sentmsglist\_sent.setSelectedIndex(-1);

sentmsgidlist\_sent.setSelectedIndex(-1);

sentmsglist\_sent.setModel(dlm1);

sentmsgidlist\_sent.setModel(dlm2);

query = "update msgdtls set senderdel = 'yes' where msgid=" + msgid;

sqlupdatequery(query);

}

}

else{

JOptionPane.showMessageDialog(null, "Please select a msg to delete");

}

}

***private void sentmsglist\_sentValueChanged(javax.swing.event.ListSelectionEvent evt)*** {

Object msg\_id ;

sentmsgidlist\_sent.setSelectedIndex( sentmsglist\_sent.getSelectedIndex());

msg\_id = sentmsgidlist\_sent.getSelectedValue();

String br = '\t'+"--------------------------------------------------"+'\n'+

'\t'+"--------------------------------------------------"+'\n';

if(msg\_id!=null){

try {

query = "select msgid, status ,reciever, time, sender, sendermail ,msg from msgdtls where msgid = " + msg\_id + ";";

rs = sqlexecutequery(query); rs.next();

sentmsgarea\_sent.setText("");

if (rs.getInt("status") == 0) {

sentmsgarea\_sent.setText(rs.getString("reciever") + " have read the message"+'\n'+"Sent on : " + '\n' + rs.getString("time") + '\n' +br+ "Message:" + '\n' + rs.getString("msg")); }

else {

sentmsgarea\_sent.setText(rs.getString("reciever") + " haven't read the message yet"+'\n'+"Sent on : " + '\n' + rs.getString("time") + '\n'+ br + "Message:" + '\n' + rs.getString("msg"));

}

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "sent msglist"+e.getMessage());

}

}

}

**Frame: Inbox[JFrame]**

***private void Logout8\_inboxActionPerformed(java.awt.event.ActionEvent evt)*** {

logout();

}

***private void replybtn\_inboxActionPerformed(java.awt.event.ActionEvent evt)*** {

if (mailfrom== null) {

JOptionPane.showMessageDialog(null, "Please select a msg to reply back;");

} else {

totxt\_write.setText(mailfrom);

write.setVisible(true);

}

}

***private void Close\_sent1ActionPerformed(java.awt.event.ActionEvent evt)*** {

inbox.dispose();

}

***private void Deletebtn\_inboxActionPerformed(java.awt.event.ActionEvent evt)*** {

DefaultListModel dlm1 = (DefaultListModel) msglist\_inbox.getModel();

DefaultListModel dlm2 = (DefaultListModel) msgidlist\_inbox.getModel();

Object msgid = msgidlist\_inbox.getSelectedValue();

if (msgid != null) {

if (JOptionPane.showConfirmDialog(null, "Are you sure to Delete")==

JOptionPane.YES\_OPTION) {

dlm1.removeElementAt(msglist\_inbox.getSelectedIndex());

dlm2.removeElementAt(msgidlist\_inbox.getSelectedIndex());

msgtextarea\_inbox.setText("");

mailfrom=null;

msglist\_inbox.setSelectedIndex(0);

msgidlist\_inbox.setSelectedIndex(0);

query = "update msgdtls set recieverdel = 'yes' where msgid=" + msgid;

sqlupdatequery(query);

}

}

else

{

JOptionPane.showMessageDialog(null, "Please select a msg to Delete");

}

}

***private void msglist\_inboxValueChanged(javax.swing.event.ListSelectionEvent evt)*** {

Object msg\_id ;

msgidlist\_inbox.setSelectedIndex( msglist\_inbox.getSelectedIndex());

msg\_id = msgidlist\_inbox.getSelectedValue();

msgtextarea\_inbox.setText("");

String br = '\t'+"--------------------------------------------------"+'\n'+

'\t'+"--------------------------------------------------"+'\n';

if( msg\_id!=null){

try {

query = "select msgid, status , time, sender, sendermail ,msg from msgdtls where msgid = " + msg\_id + ";";

rs = sqlexecutequery(query);

rs.next();

msgtextarea\_inbox.setText("Sent by:"+ rs.getString("sender")+'\n'

+"Email:"+rs.getString("sendermail")+'\n'+"Time:"+rs.getString("time")+'\n'+br+

"Message: "+'\n'+ rs.getString("msg") );

mailfrom=rs.getString("sendermail");

query = "update msgdtls set status = 0 where msgid = " + msg\_id + ";"; //status=0==> read

sqlupdatequery(query);

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

}}

**Dialog: msgcounter[JDialog]**

***private void close\_statsActionPerformed(java.awt.event.ActionEvent evt)*** {

msgcounter.dispose();

}

* **METHODS USED IN THE PROGRAM**

**private ResultSet sqlexecutequery(String query)** {**/\*method to execute query\*/**

try {

Class.forName("java.sql.Driver");

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

stmt = con.createStatement();

rs = stmt.executeQuery(query);

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

return rs;

}

**private void sqlupdatequery(String query)** **{/\*method to execute update sql query\*/**

try {

Class.forName("java.sql.Driver");

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

stmt = con.createStatement();

stmt.executeUpdate(query);

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage());

} }

**private void logout()**{ **//to logout**

mailfrom = null ;

usrnm= null;

pass= null;

mail= null;

usrid=0;

//loginframe

usrnmtxt\_login.setText("");

usrpass\_login.setText("");

//welcome frame

username\_welcome.setText("");

//inbox frame

msgtextarea\_inbox.setText("");

msglist\_inbox.removeAll();

msgidlist\_inbox.removeAll();

//writeframe

msgarea\_write.setText("");

totxt\_write.setText("");

//sentframe

sentmsgarea\_sent.setText("");

sentmsglist\_sent.removeAll();

sentmsgidlist\_sent.removeAll();

//accsttng frame

oldpassl\_accsttng.setText("");

newpassl\_accsttng.setText("");

//dispose frames and dialogs

welcome.dispose();

inbox.dispose();

write.dispose();

sent.dispose();

accsttng.dispose();

msgcounter.dispose();

signup.dispose();

msgdtlspg.dispose();

JOptionPane.showMessageDialog(null, "Successfully logged out");

login.setVisible(true);

try{

rs.close();

stmt.close();

con.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(null, e.getMessage());

}

}

**6. User Manual**

## 6.1 How to install Software:

### Hardware Requirement-

### Latest Intel processor based PC at Client/Server end.

### 1 GB RAM and 4GB HDD space (for Database) is desirable.

### Standard I/O devices like Keyboard and Mouse etc.

### Software Requirement-

* Windows 2000/XP OS is desirable.
* NetBeans Ver 5.1 or higher should be installed with JDK and JVM.
* MySQL Ver 6.1 with Library Database must be present at machine.

### 

### Database Installation

The software project is distributed with a backup copy of a Database named **MANALI**with required tables. Some dummy records are present in the tables for testing purposes, which can be deleted before inserting real data. The project is shipped with Lib.sqlfile which installs a database and tables in the computer system.

Note: The PC must have MySQL server with user (***root***) and password (***123456***). If root password is any other password, it can be changed by running MySQL Server Instance Configure Wizard.

Start ⏵Program ⏵ MySQL ⏵MySQL Server ⏵MySQL Server Instance Config Wizard

Provide current password of root and new password as “123456” , this will change the root password.

To install a MySQL database from a dump file ( Lib.sql), simply follow the following steps.

**Step 1:** Copy the Lib.sql file in **C:\Program files\Mysql\MySql server 5.1\Bin** folder.

**Step 2:** Open MySQL and type the following command to create the database named Manali.

mysql> create database **MANALI**;

**Step 3:** Open Command Window (Start ⏵Run ⏵ cmd)

**Step 4:** Go to the following folder using CD command of DOS.

**C:\Program files\Mysql\MySql server 5.1\Bin>**

**Step 5:** type the following command on above prompt -

**C:….\bin>** mysql -u ***root***–p***123456*** Manali< lib.sql

This will create a MANALI database with required tables.

**7. References**

In order to work on this project titled –***INSTANT MESSENGER***,I have referred to the following books and literature during the various phases of development of the project.

(1) Informatics Practices for class XII (NCERT)

(2) Informatics Practices for class XII

-by Sumita Arora

Other than the above-mentioned book, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.