LABMA!L

Informatics Practices Project

By:
Vishakan Subramanian and Joshua Joseph
Class: XII – E3
Roll Nos: 123532 & 123517
Year: 2017-2018

Acknowledgement

Firstly,

We would like to thank our Informatics Practices teacher Mrs.Lakshmi Radhika for her support throughout the making of the project.

Her immense contribution to the project goes without saying and all of this has basically been possible only because of her.

Secondly,

We would like to thank the school and the administration for providing us a chance to do a project like this, and providing us with the materials that we required.

Lastly,

We would like to thank the Almighty for bestowing his blessings upon us.

Contents

Title	Starting Page
❖ Project Description.	1
❖ Project Analysis.	2
❖ MySQL Databases and Tables.	8
❖ NetBeans and Java.	12
❖ Source Code.	17
❖ Outputs & Screenshots.	47
❖ Scope & Limitations.	55
❖ Bibliography.	57

Project Description

This project, named “LabMail” stylized as “LabMa!l.” was an idea conceived by us to quicken communication between users connected through different computers connected to a LAN, which is basically the LAN of an organization working within 1 or 2 buildings, like a school or an office block without the need of an Internet connection.

The application allows users to register themselves as a member of its user community and use the application for sending textual messages to other users, who are also connected to the LAN and are registered in the application’s database.

It is to be used for instant messaging between people without the hassle of walking through the building/using a telephone/using an Internet connection/other communication means.

Since, we had developed the project keeping in mind the school’s lab and the school at large, we decided to name it LabMail.

Project Analysis

Functions Used In MailScreen frame:

1.Loader()

Usage:- To load the jTable1 element with the messages that the user has received in his inbox.

Source Code:-

```
public void Loader() {

      try {
         //To load newly received mail from the database.
         oldk = jTable1.getRowCount();//stores old row count.
         DefaultTableModel tm = (DefaultTableModel) jTable1.getModel();
         tm.setNumRows(0);
         Class.forName("java.sql.DriverManager");
         conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail", "root",
"psbb");
         stmt1 = conn.createStatement();
         String q1 = "Select s_no,r_date,sender,message from " + Login.uid;
         rs1 = stmt1.executeQuery(q1);
         while (rs1.next()) {
            int s_no = rs1.getInt("S_No");
            String r_date = rs1.getString("R_Date");
            String sender = rs1.getString("Sender");
            String message = rs1.getString("Message");
            Object row[] = {s_no, sender, message, r_date};
            tm.addRow(row);
         }
         oldc = jTable1.getRowCount();//stores new row count.
         flag = false;
         //false=inbox.
      } catch (Exception e) {
         JOptionPane.showMessageDialog(this, "An unexpected error has
occured."
                    + "\nWe shall rectify the problem soon."
                    + "\nSorry for the inconvenience.");
      }
  }
```

2.NewMail()

Usage:- To check whether the user has received any new mail, and display an appropriate icon if the above condition is satisfied.

Source Code:-

```
public void NewMail() {
        //To check whether any new message has arrived.
        try {
            newmail.setVisible(true);
            Class.forName("java.sql.DriverManager");
            conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail", "root",
"psbb");
            stmt1 = conn.createStatement();
            String q1 = "Select msgcount from counter where username='" +
Login.uid + "'";
            rs1 = stmt1.executeQuery(q1);
            while (rs1.next()) {
                msgc = rs1.getInt("msgcount");
            }
        } catch (Exception e) {
          JOptionPane.showMessageDialog(this, "An unexpected error has
occured."
                    + "\nWe shall rectify the problem soon."
                    + "\nSorry for the inconvenience.");

        }
        if (oldc == 0) {
            clicker = false;
            newmail.setVisible(false);
        } else if (oldc >= 0) {
            clicker = true;
        }
        if(runner==false){
            if((oldc==msgc)||(oldk==msgc)){
                newmail.setVisible(false);
                runner=true;
            }
        }
        if (oldc == oldk) {
            newmail.setVisible(false);
        }
    }
```

3.Refresher()

Usage:-

To automatically refresh the mail box, so that any newly sent/received messages are synchronized.

Source Code:-

```
public void Refresher() {
        //To auto-refresh and check for newly arrived/newly sent mail, and sync
with the database.
        int delay = 10000; //ms
        ActionListener taskPerformer = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
```

```
            if (flag == false) {
                Loader();
                NewMail();
            }
            if (flag == true) {
                Sent_Mail();
            }
        }
    };
    new javax.swing.Timer(delay, taskPerformer).start();
}
```

4.Sent_Mail()

Usage:- To load the messages sent by the user (or outbox) in the jTable1 element.

Source Code:-

```
public void Sent_Mail() {
        //To load the table with sent messages, and load sent messages
from the database.
        try {
            newmail.setVisible(false);
            DefaultTableModel tm = (DefaultTableModel) jTable1.getModel();
            tm.setNumRows(0);
            Class.forName("java.sql.DriverManager");
            conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail_se
nt", "root", "psbb");
            stmt1 = conn.createStatement();
            String q1 = "Select * from " + Login.uid;
            rs1 = stmt1.executeQuery(q1);
            while (rs1.next()) {
                int s_no = rs1.getInt("S_No");
                String s_date = rs1.getString("S_Date");
            String receiver = rs1.getString("Receiver");
            String message = rs1.getString("Message");
            Object row[] = {s_no, receiver, message, s_date};
            tm.addRow(row);
         }
         flag = true;
         //true=outbox
      } catch (Exception e) {
         JOptionPane.showMessageDialog(this, "An unexpected error has
occured."
                 + "\nWe shall rectify the problem soon."
                 + "\nSorry for the inconvenience.");
      }

   }
```

5.RecUpdater()

Usage:- To update the table after performing deletion of a message from the mail box.

Source Code:-

```
public void RecUpdater(String db) {
      //To update the table and database after performing deletion.
      try {
         Class.forName("java.sql.DriverManager");
         conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/" + db,
"root", "psbb");
         stmt = conn.createStatement();
         stmt.executeUpdate("delete from " + Login.uid + " where S_No="
+ rowc + "");
         stmt.execute("set @r:=0");
         stmt.executeUpdate("update " + Login.uid + " set
s_no=@r:=@r+1");
         newmail.setVisible(false);
```

```
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "An unexpected error has
occured."
                    + "\nWe shall rectify the problem soon."
                    + "\nSorry for the inconvenience.");
        }

    }
```

6.TableUpdater()

Usage:- To change the column names in the jTable1 element appropriately, according to the active content in the jTable1, namely inbox and outbox.

Source Code:-

```
    public void TableUpdater(String s, String d) {
        //To update the table with appropriate column names, while
switching between outbox and inbox.
        JTableHeader th = jTable1.getTableHeader();
        TableColumnModel tcm = th.getColumnModel();
        TableColumn tc1 = tcm.getColumn(1);
        tc1.setHeaderValue(s);
        TableColumn tc2 = tcm.getColumn(3);
        tc2.setHeaderValue(d);
    }
```

MYSQL Databases and Tables

Tables stored under the database *'labmail'*.

show tables;

```
+--------------------+
| Tables_in_labmail  |
+--------------------+
| counter            |
| feedback           |
| jagnm710           |
| joshiee            |
| labadmin           |
| login              |
| sample             |
| shiva123           |
| vinaydevabhaktuni  |
| yugesh             |
+--------------------+
```

Table containing the user credentials.

select * from login;

```
+-------------------+-------------+-----------------------------------+------------+
| Username          | Password    | question                          | Answer     |
+-------------------+-------------+-----------------------------------+------------+
| jagnm710          | jaganm71000 | What is the name of your pet?     | joshiee    |
| joshiee           | idontknow   | What is the name of your pet?     | Woof       |
| labadmin          | labber      | What is the name of your pet?     | Woofy      |
| Sample            | sample      | What is the name of your pet?     | Doggie     |
| shiva123          | shiva123    | Who is your best friend?          | myself     |
| vinaydevabhaktuni | emily2028   | Which is your favorite TV sitcom? | lucifer    |
| yugesh            | 123456      | Who is your favorite athlete?     | usain bolt |
+-------------------+-------------+-----------------------------------+------------+
```

A sample table of an existing user.

select * from labadmin;

```
+------+---------------------+---------+----------------+
| S_No | R_Date              | Sender  | Message        |
+------+---------------------+---------+----------------+
|    1 | 2017/11/28 14:46:02 | sample  | hello          |
|    2 | 2017/11/28 15:11:01 | joshiee | Hello labadmin |
+------+---------------------+---------+----------------+
```

The table holding feedback values.

select * from feedback;

```
+--------+--------+--------------------+
| Name   | Rating | Feedback           |
+--------+--------+--------------------+
| sample |      5 | very user friendly |
+--------+--------+--------------------+
```

The table holding the most recent number of messages.

select * from counter;

```
+-------------------+----------+
| Username          | Msgcount |
+-------------------+----------+
| jagnm710          |        0 |
| joshiee           |        5 |
| labadmin          |        2 |
| Sample            |        0 |
| shiva123          |        0 |
| vinaydevabhaktuni |        0 |
| yugesh            |        1 |
+-------------------+----------+
```

<u>The database *'labmail_sent'* holding the sent mail.</u>

use labmail_sent;

show tables;

```
+------------------------+
| Tables_in_labmail_sent |
+------------------------+
| jagnm710               |
| joshiee                |
| labadmin               |
| sample                 |
| shiva123               |
| vinaydevabhaktuni      |
| yugesh                 |
+------------------------+
```

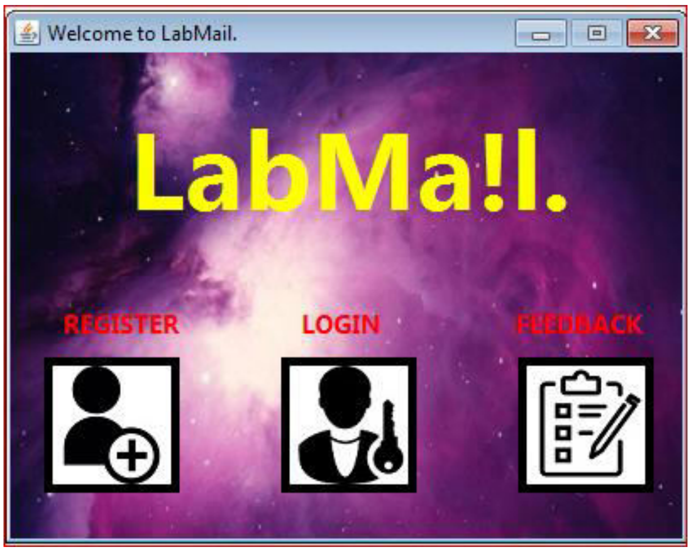
<u>A sample table's records.</u>

select * from labadmin;

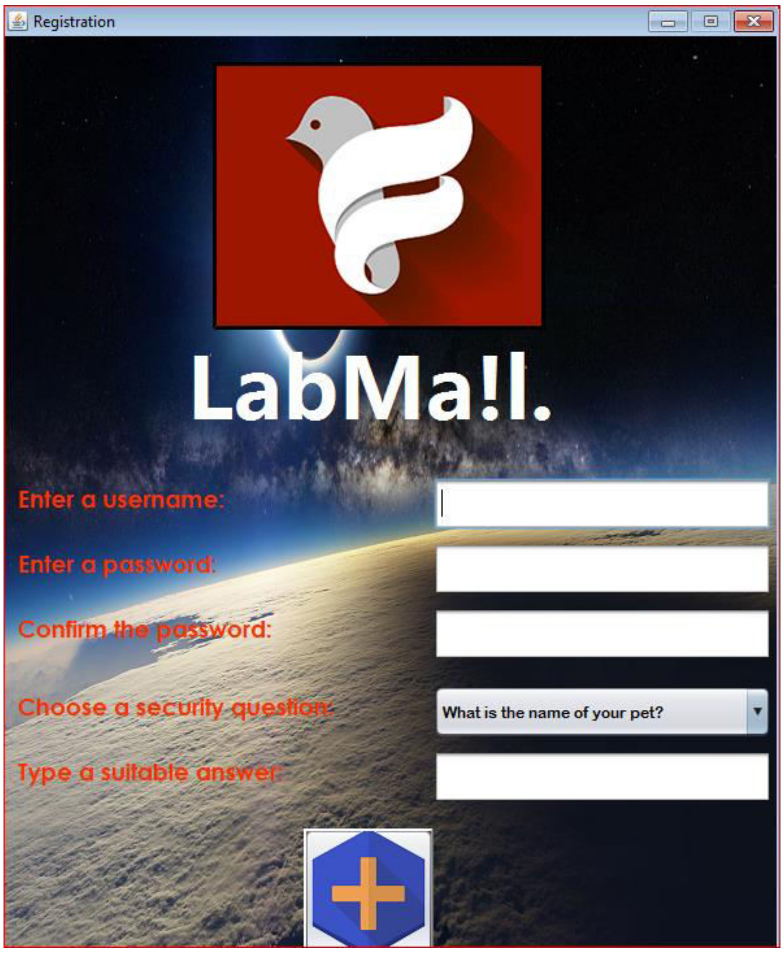
```
+------+----------+---------------------+------------+
| S_No | Receiver | S_Date              | Message    |
+------+----------+---------------------+------------+
|    1 | joshiee  | 2017/11/24 13:54:27 | Hello man  |
+------+----------+---------------------+------------+
```

Frame	Function
IntroFrame	An opening screen/loading screen for the user.
Registration	To register in the application as a new user.
Login	To login as an existing user.
ForgotPass	To change password, if the user has forgotten his/her current password.
MailScreen	To display the mail received and sent by the user.
MsgScr	To view the complete message sent/received by a user.
WriteScreen	To compose a new mail to another user.
FeedbackForm	To send feedback of a user, and store in under the database.

IntroFrame



Registration



Enter a username:

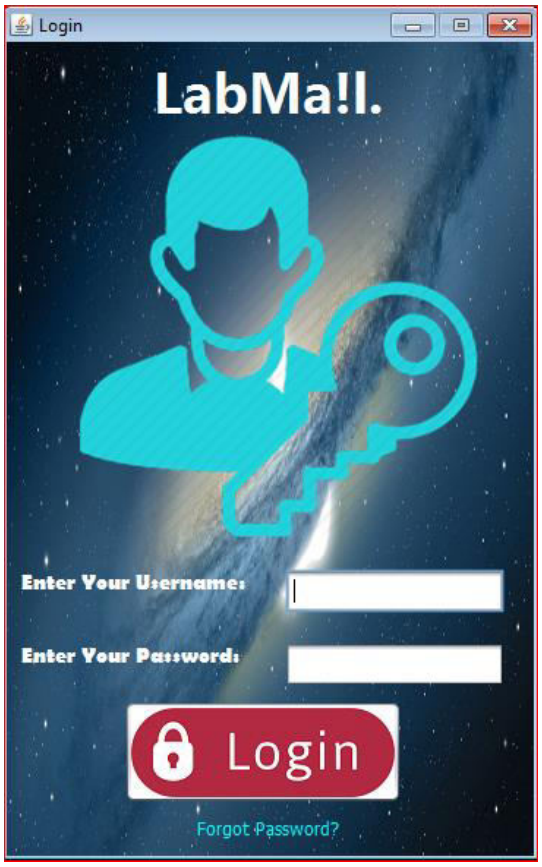
Enter a password:

Confirm the password:

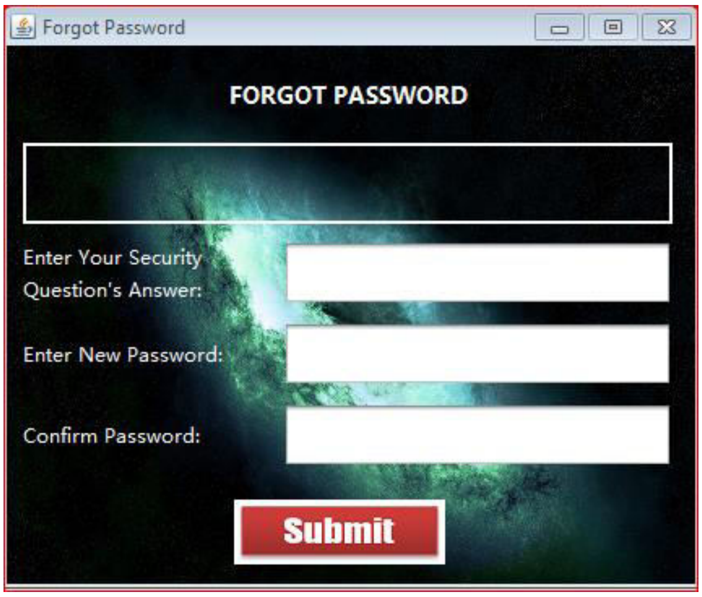
Choose a security question: What is the name of your pet?

Type a suitable answer:

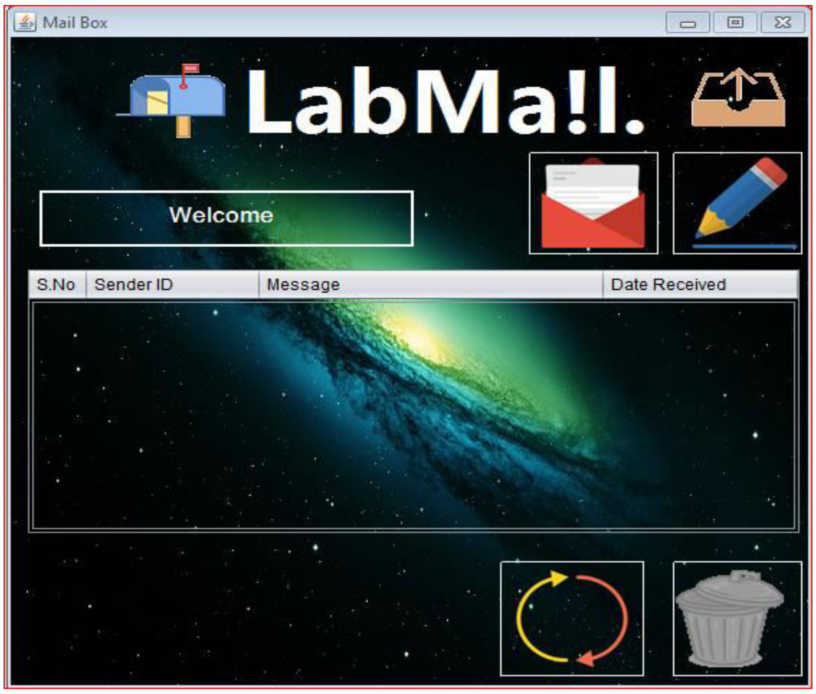
Login



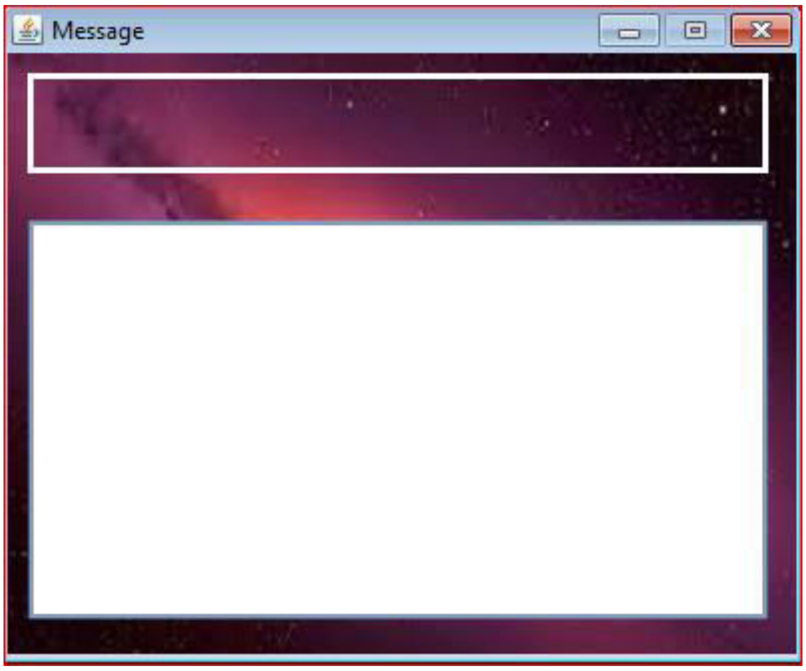
ForgotPass



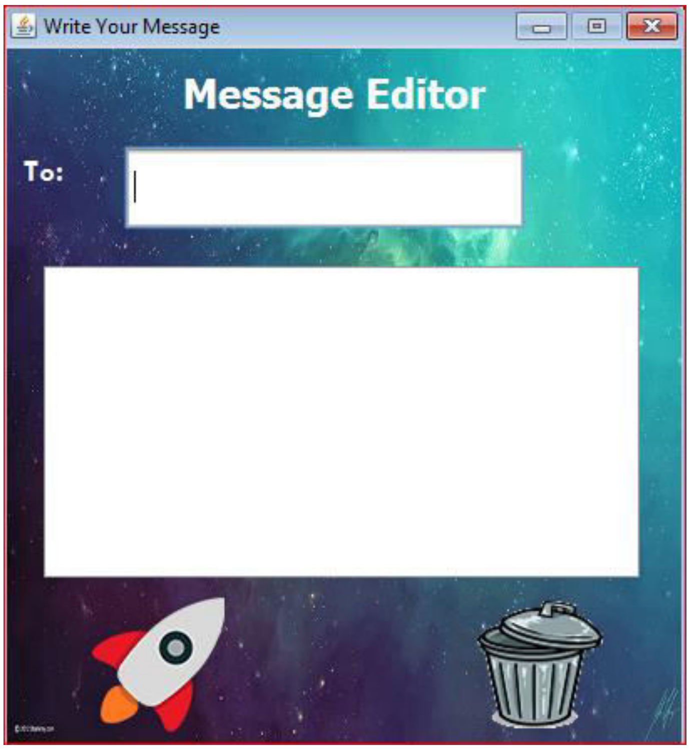
MailScreen



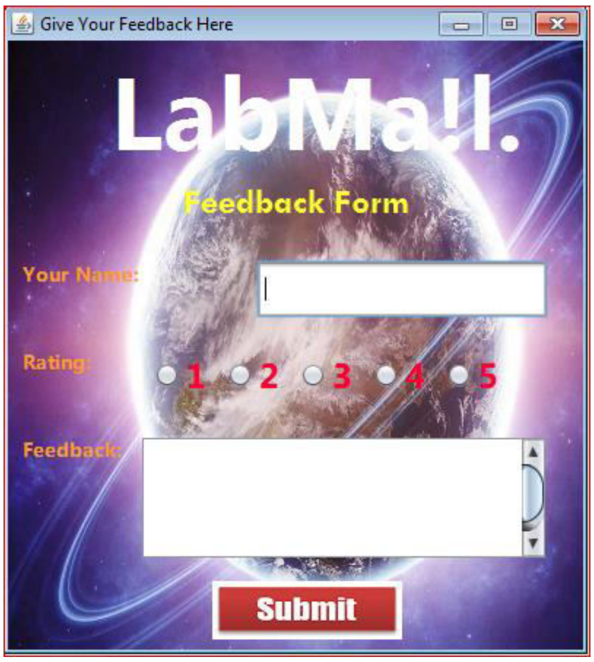
MsgScr



WriteScreen



FeedbackForm



Source Code

IntroFrame

Imports Used:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
```

Under initComponents():

```
        setSize(403, 315);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winCl
osingEvent);
    }
```

Coding:

```
private void feedbackMouseClicked(java.awt.event.MouseEvent evt)
{
        // To go to the feedback form, to give personal user feedback.
        FeedbackForm ff = new FeedbackForm();
        ff.setVisible(true);
        close();

    }

    private void registerMouseClicked(java.awt.event.MouseEvent
evt) {
        // To go to the registration form, to create a new user.
        Registration r = new Registration();
        r.setVisible(true);
        close();
    }

    private void loginMouseClicked(java.awt.event.MouseEvent evt) {
        // To go to the login form, to login as an existing user.
        Login l = new Login();
        l.setVisible(true);
        close(); }
```

Registration

Imports Used:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JOptionPane;
```

Under initComponents():

```
setSize(600, 730);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winCl
osingEvent);
   }
```

Coding:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // To create a new account:
        String ques = (String) question.getSelectedItem();
        String ans = answer.getText().trim();
        String uname = t1.getText().trim();
        String pass = new String(p1.getPassword());
        String passcheck = new String(p2.getPassword());
        int l1 = uname.length();
        int l2 = pass.length();
        int l3 = passcheck.length();
        int l4 = ans.length();
        int k = 0, s = 0;
        if (uname.contains(" ")) {
            JOptionPane.showMessageDialog(this, "Spaces aren't allowed
in username.");
            t1.setText("");
        } else {
            if (l1 == 0 || l2 == 0 || l3 == 0 || l4 == 0) {
                JOptionPane.showMessageDialog(this, "Do not leave any
field empty.");
            }
            if (l1 < 6 || l2 < 6 || l3 < 6) {
                JOptionPane.showMessageDialog(this, "Enter a username
and password with more than 6 characters\nand less than 35
characters");
            }
            if ((l1 <= 35) && (l1 >= 6) && (l2 <= 35) && (l2 >= 6) && (l3 <=
35) && (l3 >= 6)) {
                if (l4 <= 50) {
                    if (pass.equals(passcheck)) {
                        try {
                            Class.forName("java.sql.DriverManager");
                            Connection conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                            String q2 = "select count(*) from login where
username=" + "'" + uname + "'";
                            Statement stmt2 = conn.createStatement();
                            ResultSet count = stmt2.executeQuery(q2);
                    while (count.next()) {
                        k = count.getInt("count(*)");
                    }
                    if (k > 0) {
                        JOptionPane.showMessageDialog(this, "Username
already exists, choose another name.");
                    } else if (k == 0) {
                        try {
                            Class.forName("java.sql.DriverManager");
                            Connection conn1 =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                            String q = "insert into login values('" + uname +
"','" + pass + "','" + ques + "','" + ans + "')";
                            String q1 = "create table " + uname + "(S_No
int(5) primary key,R_Date varchar(19), Sender varchar(35), Message
varchar(65000))";
                            String q3 = "insert into counter values('" + uname
+ "'," + s + ")";
                            Statement stmt = conn1.createStatement();
                            stmt.executeUpdate(q);
                            stmt.executeUpdate(q3);
                            stmt.execute(q1);
                            Class.forName("java.sql.DriverManager");
                            Connection conn2 =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
_sent", "root", "psbb");
                            String p = "create table " + uname + "(S_No int(5)
primary key, Receiver varchar(35), S_Date varchar(19), Message
varchar(65000))";
                            Statement stmte = conn2.createStatement();
                            stmte.execute(p);
                            JOptionPane.showMessageDialog(this,
"Registration successful!");
```

```
                            IntroFrame i = new IntroFrame();
                            i.setVisible(true);
                            close();
                          } catch (Exception ex) {
                            JOptionPane.showMessageDialog(this,
ex.getMessage());
                          }
                        }
                      } catch (Exception ex) {
                        JOptionPane.showMessageDialog(this, "An
unexpected error has occured."
                            + "\nWe shall rectify the problem soon."
                            + "\nSorry for the inconvenience.");
                        close();
                      }
                    } else {
                      JOptionPane.showMessageDialog(this, "Passwords do
not match.");
                    }
                  } else {
                    JOptionPane.showMessageDialog(this, "Enter an answer
less than 50 characters");
                  }
                } else {
                  JOptionPane.showMessageDialog(this, "Do not exceed 35
characters");
                }}}

  private void formWindowClosing(java.awt.event.WindowEvent
evt) {
    // to go back to IntroScreen:
    IntroFrame i = new IntroFrame();
    i.setVisible(true);
  }
```

Login

Imports:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JOptionPane;
```

Under initComponents():

```
jScrollPane1.setOpaque(false);
        jScrollPane1.getViewport().setOpaque(false);
        setSize(367, 584);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winCl
osingEvent);
    }
```

Global Variables:

```
public static String userid = "";
public static String uid = "";
```

Coding:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // To enter user details, for logging in:
        userid = t1.getText();
        String pwdc = "";
        String pwd = new String(p1.getPassword());
        if (userid.equals("") || pwd.equals("")) {
            JOptionPane.showMessageDialog(this, "Enter your username
and password for logging in.");
        } else {
            try {
                Class.forName("java.sql.DriverManager");
                Connection conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                String q = "Select password from login where username='" +
userid + "'";
                Statement stmt = conn.createStatement();
                ResultSet l1 = stmt.executeQuery(q);
                while (l1.next()) {
                    pwdc = l1.getString("password");  }
                if (pwd.equals(pwdc)) {
                    uid = userid;
                    MailScreen m = new MailScreen();
                    m.setVisible(true);
                    this.setVisible(false);
                } else {
                    JOptionPane.showMessageDialog(this, "Wrong
credentials.");  }
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(this, ex.getMessage());
           }}}
private void formWindowClosing(java.awt.event.WindowEvent evt) {
        //To go back to the intro screen.
        IntroFrame i = new IntroFrame();
        i.setVisible(true);
    }

    private void jLabel6MouseClicked(java.awt.event.MouseEvent evt)
{
        // To change password in case password is forgotten.
        userid = t1.getText().trim();
        if (userid.length() != 0) {
            try {
                Class.forName("java.sql.DriverManager");
                Connection conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                String q = "Select password from login where username='" +
userid + "'";
                Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery(q);
                rs.first();
                if (rs.getRow() == 0) {
                    JOptionPane.showMessageDialog(this, "Specified user
does not exist.");
                } else {
                    ForgotPass fp = new ForgotPass();
                    fp.setVisible(true);
                }
            } catch (Exception e) {
            }
        } else {
            JOptionPane.showMessageDialog(this, "Enter your username
to proceed further.");
        }
    }
```

ForgotPass

Imports Used:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.sql.*;
import javax.swing.JOptionPane;
```

Under initComponents():

```
setSize(430, 360);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
        //To load the security question of the specified user.
        try {
            Class.forName("java.sql.DriverManager");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
            Statement stmt1 = conn.createStatement();
            String q1 = "select question from login where username='" +
Login.userid + "'";
            ResultSet rs = stmt1.executeQuery(q1);
            while (rs.next()) {
                ques = rs.getString("question");
            }
            question.setText(ques);
        } catch (Exception e) {
        }
```

Global Variables:

```
  String ques = "";
```

Coding:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // To check the entered answer and update password, if answer
is correct.

        String cans = "";
        String uans = t1.getText();
        String newp = new String(t2.getPassword());
        String newpc = new String(t3.getPassword());
        int l1 = newp.length();
        int l2 = newpc.length();
        int l3 = uans.length();
        if ((l1 > 0) && (l1 <= 35) && (l2 > 0) && (l2 <= 35)) {
            if ((l3 > 0) && (l3 < 50)) {
                try {
                    Class.forName("java.sql.DriverManager");
                    Connection conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                    Statement stmt1 = conn.createStatement();
                    String q1 = "select answer from login where username='"
+ Login.userid + "'";
                    ResultSet rs = stmt1.executeQuery(q1);
                    while (rs.next()) {
                        cans = rs.getString("answer");
                    }
                    if (uans.equalsIgnoreCase(cans)) {
                        if (newp.equals(newpc)) {
```

```
                        Statement stmt2 = conn.createStatement();
                        String q2 = "update login set password='" + newp + "'
where username='" + Login.userid + "'";
                        stmt2.executeUpdate(q2);
                        JOptionPane.showMessageDialog(this, "Password
successfully updated.");
                        this.setVisible(false);
                    } else {
                        JOptionPane.showMessageDialog(this, "Passwords
do not match.");
                    }
                } else {
                    JOptionPane.showMessageDialog(this, "Answer
entered is incorrect.");
                }
            } catch (Exception e) {
            }
        } else {
            JOptionPane.showMessageDialog(this, "Enter an answer
less than 50 characters");
        }
    } else {
        JOptionPane.showMessageDialog(this, "Enter passwords less
than 35 characters");
    }
  }
```

MailScreen

Imports Used:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;
import javax.swing.table.TableColumn;
import javax.swing.table.TableColumnModel;
```

Under initComponents():

```
jScrollPane1.setOpaque(false);
        jScrollPane1.getViewport().setOpaque(false);
        setSize(560, 535);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
        l1.setText("Welcome " + Login.uid);
        runner=false;
        Loader();
        NewMail();
        Refresher();
public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winCl
osingEvent);
    }
```

Global Variables:

```
boolean runner=false;
    public static boolean flag = false, clicker = false;
    public int rowno;
    public String rowc = "";
    public static int recno = -1;
    public static String send_id;
    int count = 0;
    Connection conn;
    Statement stmt2, stmt1, stmt;
    ResultSet rs1, rs2, rs3;
    public int oldc = 0, oldk = 0,msgc=0;
```

Coding:

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        // To select a message for further action.
        rowno = jTable1.getSelectedRow();
        Object o1 = jTable1.getValueAt(rowno, 0);
        rowc = o1.toString();
        recno = Integer.parseInt(rowc);
        if (recno >= 0) {
            delete.setEnabled(true);
        }
    }
    private void deleteMouseClicked(java.awt.event.MouseEvent evt)
{
        // To delete a selected message:
        if (flag == false) {
            RecUpdater("labmail");
            delete.setEnabled(false);
            Loader();
            NewMail();
            newmail.setVisible(false);
        } else if (flag == true) {
            RecUpdater("labmail_sent");
            delete.setEnabled(false);
            Sent_Mail();
            newmail.setVisible(false);
        }
    }
private void syncMouseClicked(java.awt.event.MouseEvent evt) {
        // To check for newly arrived/sent mail manually.
        if (flag == false) {
            Loader();
            NewMail();
        } else if (flag == true) {
            Sent_Mail();
        }
    }

    private void formWindowClosing(java.awt.event.WindowEvent
evt) {
        //To go back to the intro screen.
        IntroFrame i = new IntroFrame();
        i.setVisible(true);
          try {
            Class.forName("java.sql.DriverManager");
            conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
```

```
          stmt1 = conn.createStatement();
          oldc=jTable1.getRowCount();
          String q1 = "Update counter set msgcount="+oldc+" where
username='"+Login.uid+"'";
          stmt1.executeUpdate(q1);
       } catch (Exception e) {
          JOptionPane.showMessageDialog(this, "An unexpected error
has occured."
                + "\nWe shall rectify the problem soon."
                + "\nSorry for the inconvenience.");
       }
    }

    private void viewmailMouseClicked(java.awt.event.MouseEvent
evt) {
       // To open a selected message.
       if (recno >= 0) {
          MsgScr m = new MsgScr();
          m.setVisible(true);
          newmail.setVisible(false);
       } else {
          JOptionPane.showMessageDialog(this, "Select a message to
open");
       }
    }

    private void composeMouseClicked(java.awt.event.MouseEvent
evt) {
       // To compose a new message.
       WriteScreen w = new WriteScreen();
       w.setVisible(true);
       newmail.setVisible(false);
    }
    private void boxMouseClicked(java.awt.event.MouseEvent evt) {
        // To switch between inbox and outbox.
        if (flag == false) {
            TableUpdater("Receiver ID", "Date Sent");
            ImageIcon im = new ImageIcon("src\\outbox.png");
            box.setIcon(im);
            msgc=0;
            oldc=0;
            oldk=0;
            Sent_Mail();
        } else if (flag == true) {
            TableUpdater("Sender ID", "Date Received");
            ImageIcon im = new ImageIcon("src\\inbox.png");
            box.setIcon(im);
            Loader();
            oldk=0;
            oldc=0;
            msgc=0;
            NewMail();

        }

    }
```

<u>Functions Used:</u>

```
public void Loader() {
        try {
            //To load newly received mail from the database.
            oldk = jTable1.getRowCount();//stores old row count.
            DefaultTableModel tm = (DefaultTableModel)
jTable1.getModel();
            tm.setNumRows(0);
            Class.forName("java.sql.DriverManager");
```

```
          conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
          stmt1 = conn.createStatement();
          String q1 = "Select s_no,r_date,sender,message from " +
Login.uid;
          rs1 = stmt1.executeQuery(q1);
          while (rs1.next()) {
             int s_no = rs1.getInt("S_No");
             String r_date = rs1.getString("R_Date");
             String sender = rs1.getString("Sender");
             String message = rs1.getString("Message");
             Object row[] = {s_no, sender, message, r_date};
             tm.addRow(row);
          }
          oldc = jTable1.getRowCount();//stores new row count.
          flag = false;
          //false=inbox.
       } catch (Exception e) {
          JOptionPane.showMessageDialog(this, "An unexpected error
has occured."
                + "\nWe shall rectify the problem soon."
                + "\nSorry for the inconvenience.");
       }}
    public void NewMail() {
       //To check whether any new message has arrived.
       try {
          newmail.setVisible(true);
          Class.forName("java.sql.DriverManager");
          conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
          stmt1 = conn.createStatement();
          String q1 = "Select msgcount from counter where username='"
+ Login.uid + "'";
          rs1 = stmt1.executeQuery(q1);
          while (rs1.next()) {
             msgc = rs1.getInt("msgcount");
          }
       } catch (Exception e) {
          JOptionPane.showMessageDialog(this, "An unexpected error
has occured."
                + "\nWe shall rectify the problem soon."
                + "\nSorry for the inconvenience.");
}
       if (oldc == 0) {
          clicker = false;
          newmail.setVisible(false);
       } else if (oldc >= 0) {
          clicker = true;
       }
       if(runner==false){
          if((oldc==msgc)||(oldk==msgc)){
             newmail.setVisible(false);
             runner=true;
          }
       }
       if (oldc == oldk) {
          newmail.setVisible(false); }}

   public void Refresher() {
       //To auto-refresh and check for newly arrived/newly sent mail,
and sync with the database.
       int delay = 10000; //ms
       ActionListener taskPerformer = new ActionListener() {
          public void actionPerformed(ActionEvent evt) {
             if (flag == false) {
                Loader();
                NewMail();
            }
            if (flag == true) {
                Sent_Mail();
            }}};
        new javax.swing.Timer(delay, taskPerformer).start();  }
    public void RecUpdater(String db) {
        //To update the table and database after performing deletion.
        try {
            Class.forName("java.sql.DriverManager");
            conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/" + db,
"root", "psbb");
            stmt = conn.createStatement();
            stmt.executeUpdate("delete from " + Login.uid + " where
S_No=" + rowc + "");
            stmt.execute("set @r:=0");
            stmt.executeUpdate("update " + Login.uid + " set
s_no=@r:=@r+1");
            newmail.setVisible(false);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "An unexpected error
has occured."
                    + "\nWe shall rectify the problem soon."
                    + "\nSorry for the inconvenience.");
        }}
    public void Sent_Mail() {
        //To load the table with sent messages, and load sent messages
from the database.
        try {
            newmail.setVisible(false);
            DefaultTableModel tm = (DefaultTableModel)
jTable1.getModel();
```

```
            tm.setNumRows(0);
            Class.forName("java.sql.DriverManager");
            conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
_sent", "root", "psbb");
            stmt1 = conn.createStatement();
            String q1 = "Select * from " + Login.uid;
            rs1 = stmt1.executeQuery(q1);
            while (rs1.next()) {
                int s_no = rs1.getInt("S_No");
                String s_date = rs1.getString("S_Date");
                String receiver = rs1.getString("Receiver");
                String message = rs1.getString("Message");
                Object row[] = {s_no, receiver, message, s_date};
                tm.addRow(row);
            }
            flag = true;
            //true=outbox
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "An unexpected error
has occured."
                    + "\nWe shall rectify the problem soon."
                    + "\nSorry for the inconvenience.");
        }}
    public void TableUpdater(String s, String d) {
        //To update the table with appropriate column names, while
switching between outbox and inbox.
        JTableHeader th = jTable1.getTableHeader();
        TableColumnModel tcm = th.getColumnModel();
        TableColumn tc1 = tcm.getColumn(1);
        tc1.setHeaderValue(s);
        TableColumn tc2 = tcm.getColumn(3);
        tc2.setHeaderValue(d);
    }
```

MsgScr

Imports Used:

import java.awt.Dimension;
import java.awt.Toolkit;
import java.sql.;*
import javax.swing.JOptionPane;

Global Variables:

```
Connection conn;
 ResultSet rs;
 Statement stmt;
```

Under initComponents():

```
setSize(400, 330);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
        //To fully load the selected message from the database.
        ta1.setText("");
        int mail_rec = MailScreen.recno;
        if (mail_rec >=0) {
            try {
                String p="";
                Class.forName("java.sql.DriverManager");
                if (MailScreen.flag == false) {
                    conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                    p="sender";
            } else if (MailScreen.flag == true) {
               conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
_sent", "root", "psbb");
               p="receiver";
            }
            String q = "Select message,"+p+" from " + Login.uid + "
where s_no=" + mail_rec;
            stmt = conn.createStatement();
            rs = stmt.executeQuery(q);
            while (rs.next()) {
               String msg = rs.getString("message");
               String sndr = rs.getString(p);
               ta1.append(msg);
               if (MailScreen.flag == false) {
                  l1.setText(sndr + " has sent you the following:");
               } else if (MailScreen.flag == true) {
                  l1.setText("You have sent the following to "+sndr+":");
               }
            }
         } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "An unexpected
error has occured."
                  + "\nWe shall rectify the problem soon."
                  + "\nSorry for the inconvenience.");
         }
      } else {
         this.setVisible(false);
      }
```

WriteScreen

Imports Used:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JOptionPane;
import java.text.*;
import java.util.Date;
```

Under initComponents():

```
setSize(409, 445);
        setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
public void close() {
        WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winCl
osingEvent);
    }
```

Global Variables:

```
public static int k = 0;
```

Coding:

```
private void sendMouseClicked(java.awt.event.MouseEvent evt) {
        // To send the typed message.
        String uname = t1.getText().trim();
        String msg = ta1.getText().trim();
        int msgc = 0, userc = 0, sboxc = 0;
        int l1 = uname.length();
        int l2 = msg.length();
        DateFormat df = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss");
        Date d = new Date();
        String sdate = (df.format(d));
        if (l1 != 0 && l2 != 0) {
            try {
                Class.forName("java.sql.DriverManager");
                Connection c =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                Connection c1 =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
_sent", "root", "psbb");
                String q2 = "select count(*) from " + uname;
                String q3 = "select count(*) from login where username='" +
uname + "'";
                String p2 = "select count(*) from " + Login.uid;
                Statement stmt1 = c.createStatement();
                Statement stmt2 = c1.createStatement();
                ResultSet ucount = stmt1.executeQuery(q3);
                while (ucount.next()) {
                    userc = ucount.getInt("count(*)");
                }
                ResultSet count = stmt1.executeQuery(q2);
                while (count.next()) {
                    msgc = count.getInt("count(*)");
            }
            ResultSet scount = stmt2.executeQuery(p2);
            while (scount.next()) {
               sboxc = scount.getInt("count(*)");
            }
            if (userc == 0) {
               JOptionPane.showMessageDialog(this, "The specified
user does not exist.");
            } else if (msg.equals("")) {
               JOptionPane.showMessageDialog(this, "Enter a message
to send");
            } else {
               uname = t1.getText();
               msgc++;
               sboxc++;
               String q = "insert into " + uname + " values(" + msgc + ",'"
+ sdate + "','" + Login.uid + "','" + msg + "')";
               stmt1.executeUpdate(q);
               String p = "insert into " + Login.uid + " values(" + sboxc +
",'" + uname + "','" + sdate + "','" + msg + "')";
               stmt2.executeUpdate(p);
               JOptionPane.showMessageDialog(this, "Message sent
successfully!");
               this.setVisible(false);

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

         }
      } else if (l1 == 0) {
         JOptionPane.showMessageDialog(this, "Enter a username");
      } else if (l2 == 0) {
         JOptionPane.showMessageDialog(this, "Type a message");
      }
  }

  private void deleteMouseClicked(java.awt.event.MouseEvent evt)
{
      // To clear the typed message.
      ta1.setText("");
  }

  private void ta1KeyPressed(java.awt.event.KeyEvent evt) {
      // To display the number of characters left.
      String temp = ta1.getText();
      int templ = temp.length();
      ta1.setToolTipText("You have " + (65000 - templ) + " characters
left.");
      if (templ == 65000) {
         JOptionPane.showMessageDialog(this, "You've reached the
maximum limit of characters.");
         ta1.setEditable(false);
      }
  }
```

Feedback Form

Imports Used:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JOptionPane;
```

Under initComponents():

```
public class FeedbackForm extends javax.swing.JFrame {

  public FeedbackForm() {
      initComponents();
      setSize(400, 445);
      setResizable(false);
      Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
      this.setLocation(dim.width / 2 - this.getSize().width / 2,
dim.height / 2 - this.getSize().height / 2);
   }

   public void close() {
      WindowEvent winClosingEvent = new WindowEvent(this,
WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winCl
osingEvent);
   }
```

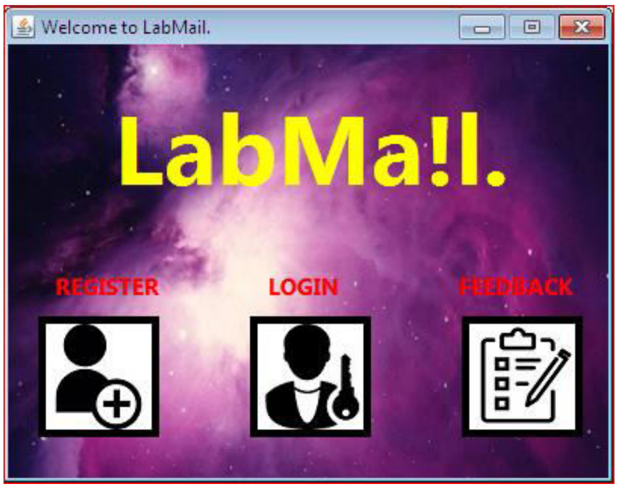
Coding:

```
 private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // To enter the user's feedback into the database.
        String f_name = t1.getText().trim();
        int rating = 0;
        String feedback = ta1.getText().trim();
        if (r1.isSelected()) {
            rating = 1;
        }
        if (r2.isSelected()) {
            rating = 2;
        }
        if (r3.isSelected()) {
            rating = 3;
        }
        if (r4.isSelected()) {
            rating = 4;
        }
        if (r5.isSelected()) {
            rating = 5;
        }
        if (rating == 0) {
            JOptionPane.showMessageDialog(this, "Kindly select a rating
for our application.");
        }
        if ((rating != 0) || (f_name.equals("")) || (feedback.equals(""))) {
            try {
                Class.forName("java.sql.DriverManager");
                Connection conn =
DriverManager.getConnection("jdbc:mysql://192.168.10.184/labmail
", "root", "psbb");
                String q = "insert into feedback values('" + f_name + "'," +
rating + ",'" + feedback + "')";
```

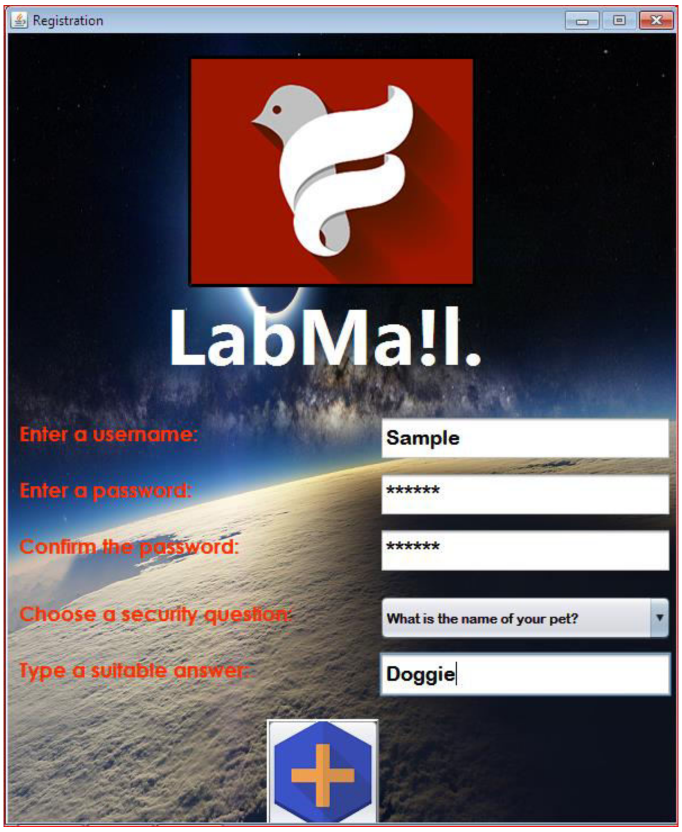
```
            Statement stmt = conn.createStatement();
            stmt.executeUpdate(q);
            JOptionPane.showMessageDialog(this, "Thank you for your
valuable feedback.");
            close();
         } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "An unexpected
error has occured."
                  + "\nWe shall rectify the problem soon."
                  + "\nSorry for the inconvenience.");
            this.setVisible(false);
         }
      }
   }

   private void formWindowClosing(java.awt.event.WindowEvent
evt) {
      // To go back to the intro screen.
      IntroFrame i = new IntroFrame();
      i.setVisible(true);
   }
```

Outputs

Opening Screen during Run-Time:



On Clicking Register Icon,



Enter a username: Sample

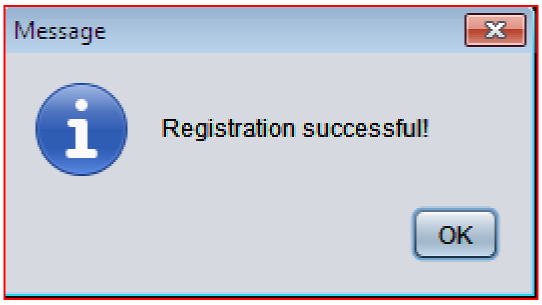
Enter a password: ******

Confirm the password: ******

Choose a security question: What is the name of your pet?

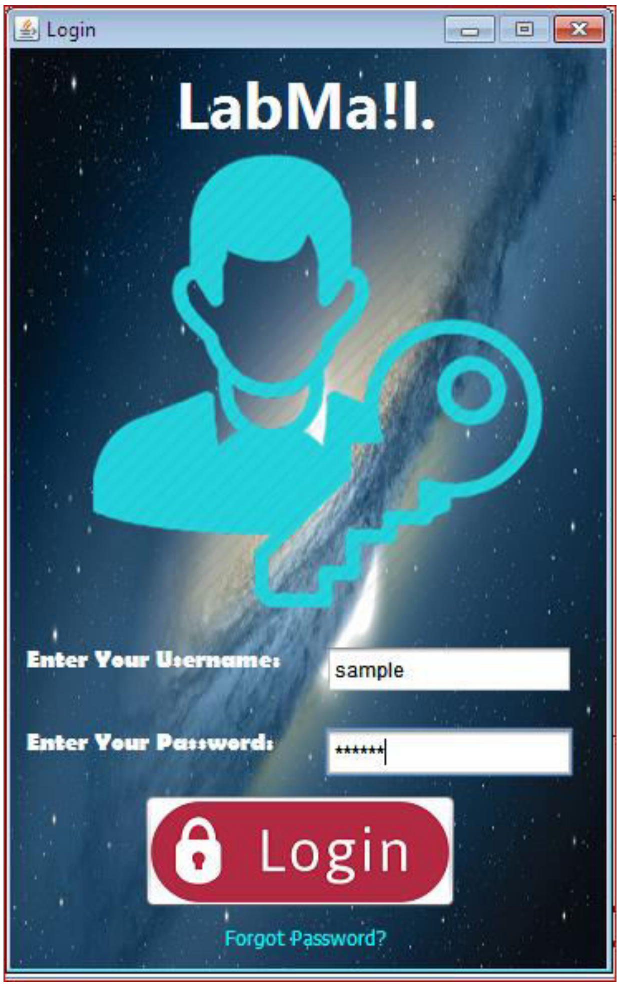
Type a suitable answer: Doggie

On clicking (+) icon,


Message

Registration successful!

On going back to Opening Screen and clicking Login Icon,

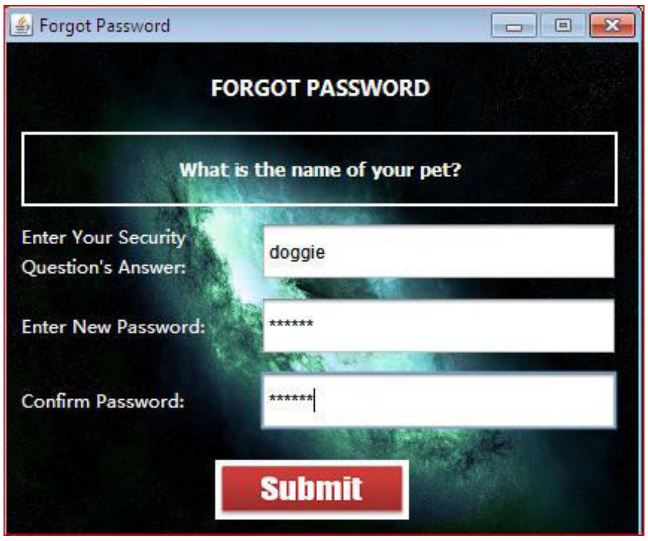

Login

LabMa!l.

Enter Your Username: sample

Enter Your Password: ******

If Password is forgotten by the user *'sample',*



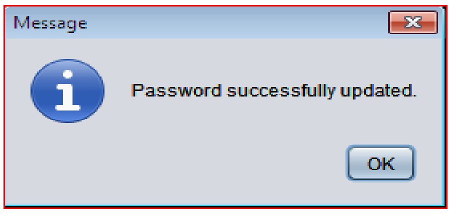
What is the name of your pet?

Enter Your Security Question's Answer: doggie

Enter New Password: ******

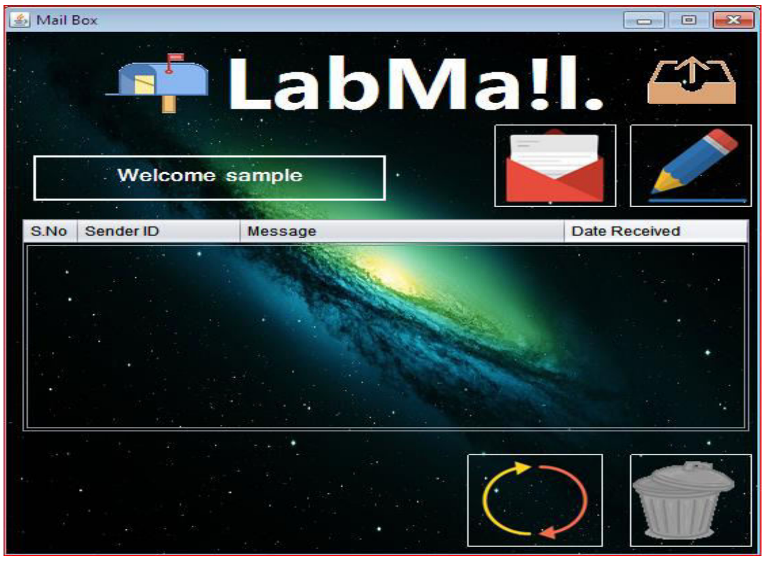
Confirm Password: ******

On submitting the details,



Password successfully updated.

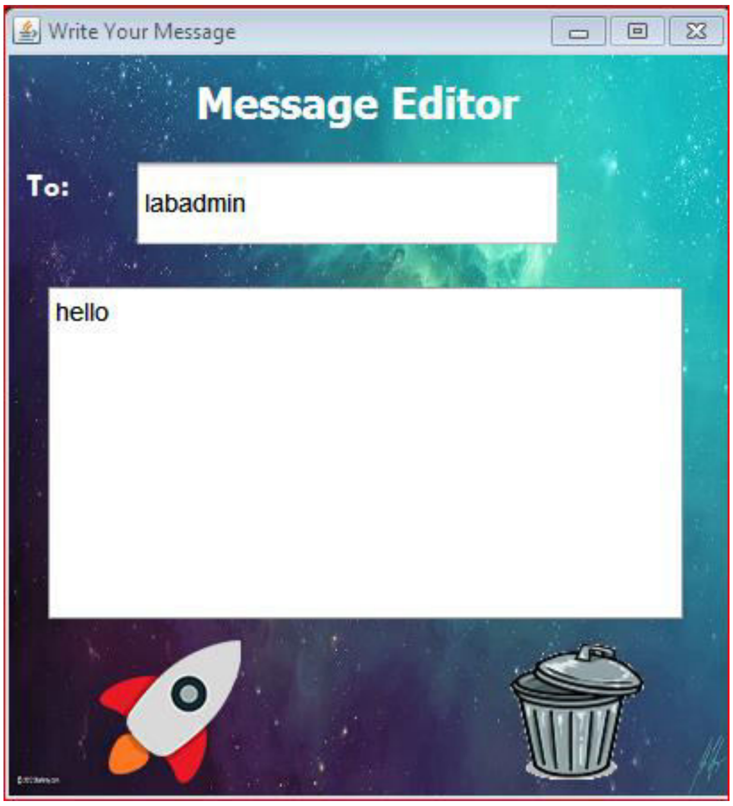
On Logging into user account *'sample',*



Welcome sample

S.No	Sender ID	Message	Date Received

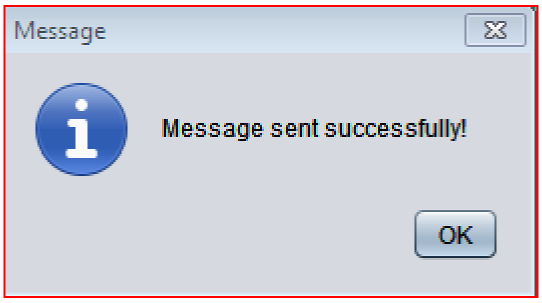
On opening Pencil icon (compose message),



labadmin

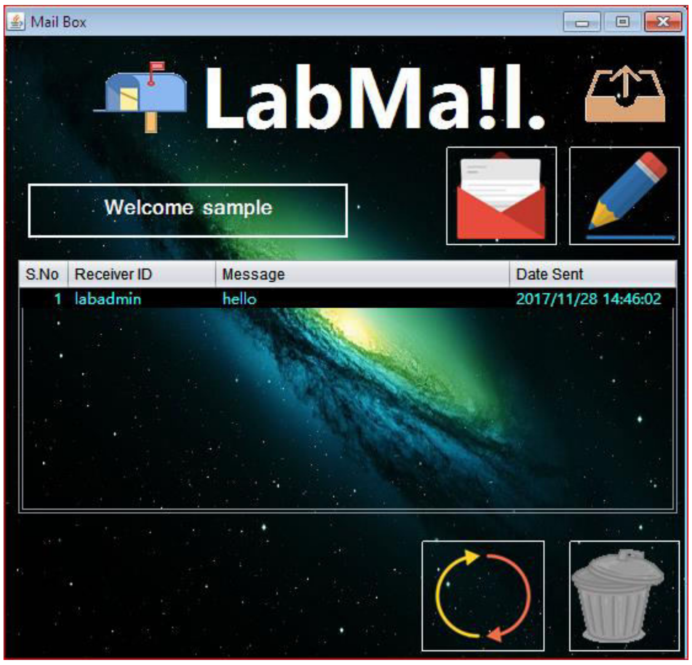
hello

On sending message,



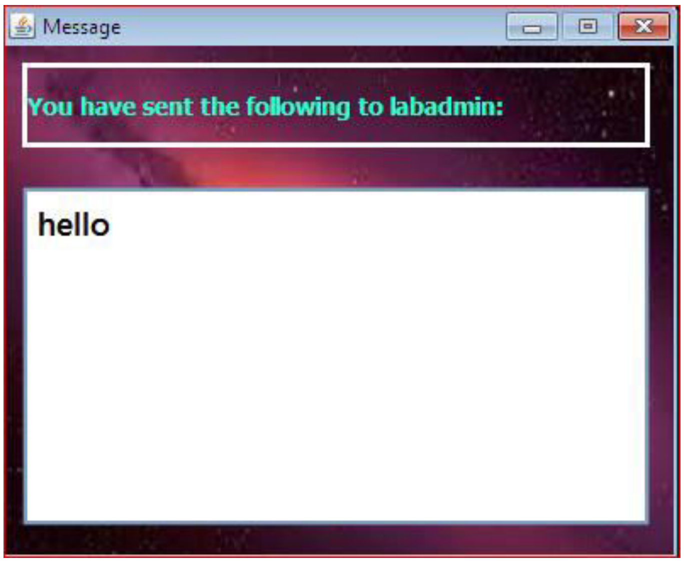
Message sent successfully!

On checking outbox in mail screen,



S.No	Receiver ID	Message	Date Sent
1	labadmin	hello	2017/11/28 14:46:02

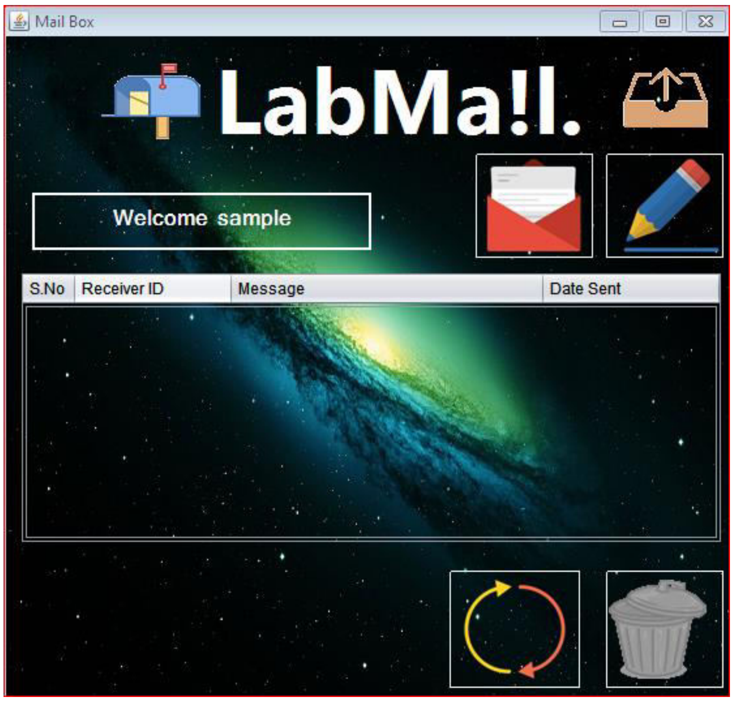
On selecting a particular mail for opening it,



You have sent the following to labadmin:

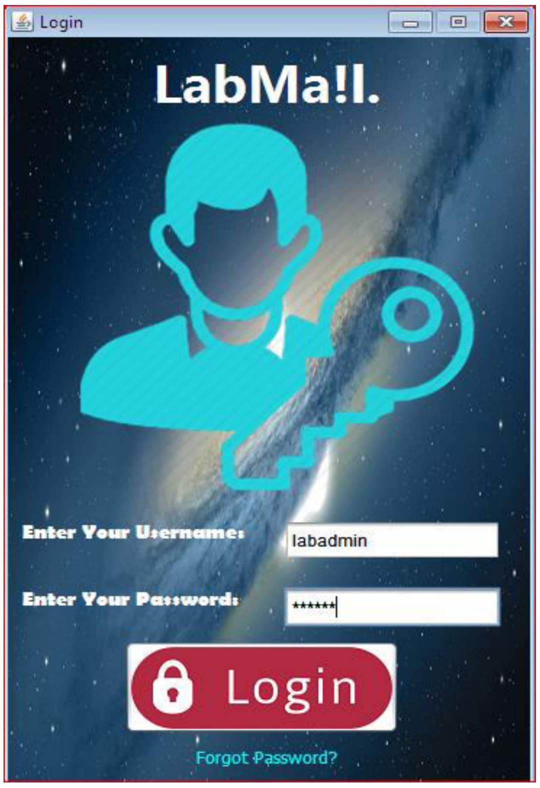
hello

On deleting a particular mail,



S.No	Receiver ID	Message	Date Sent

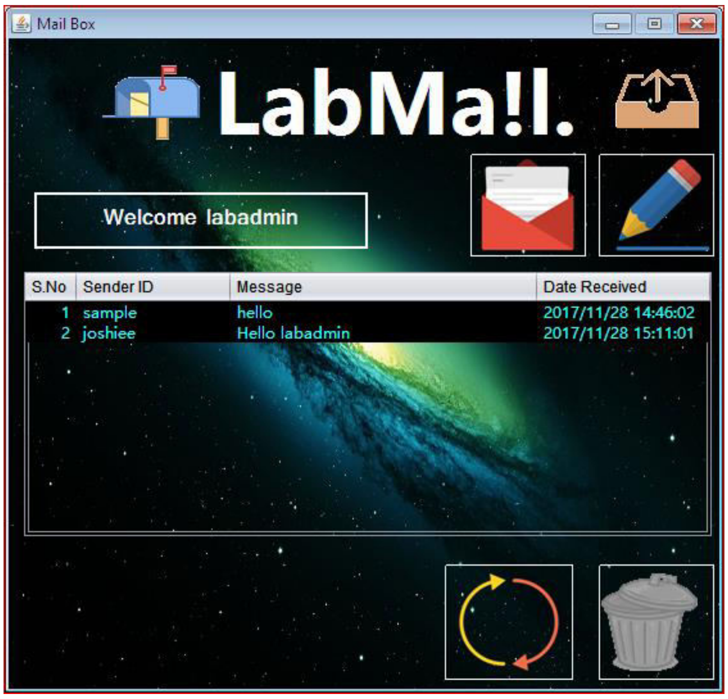
On opening receiver account *'labadmin'*,



Enter Your Username: labadmin

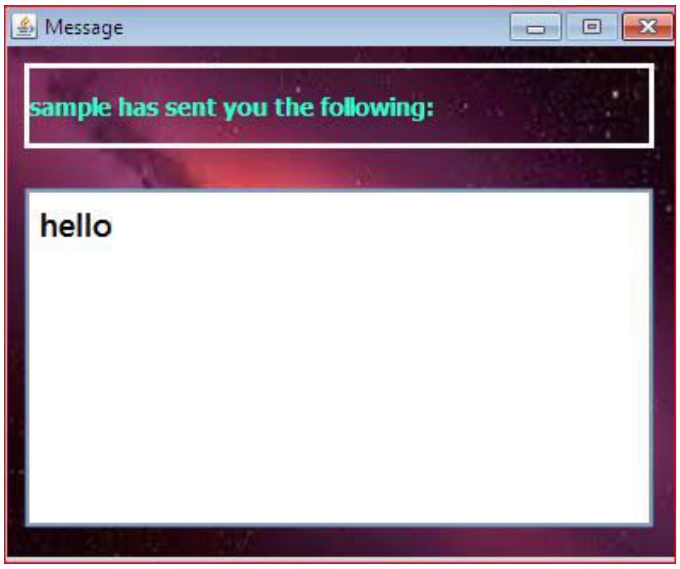
Enter Your Password: ******

On entering receiver account *'labadmin',*



S.No	Sender ID	Message	Date Received
1	sample	hello	2017/11/28 14:46:02
2	joshiee	Hello labadmin	2017/11/28 15:11:01

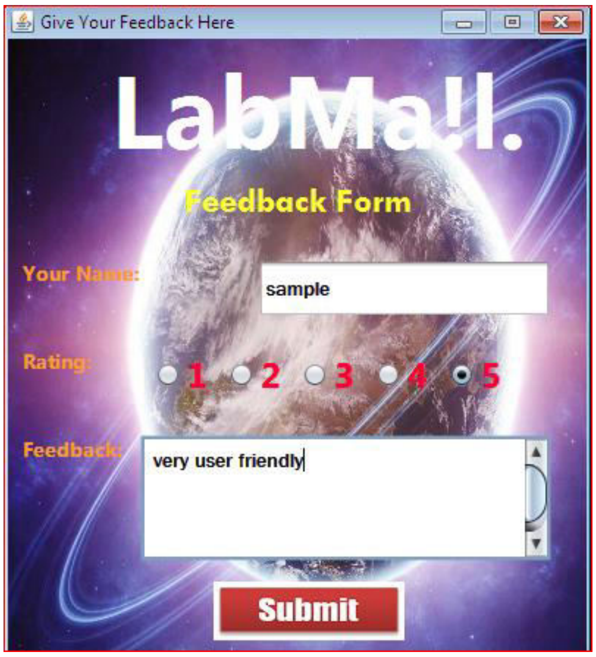
On viewing mail from sender *'sample',*



sample has sent you the following:

hello

On opening Feedback form,

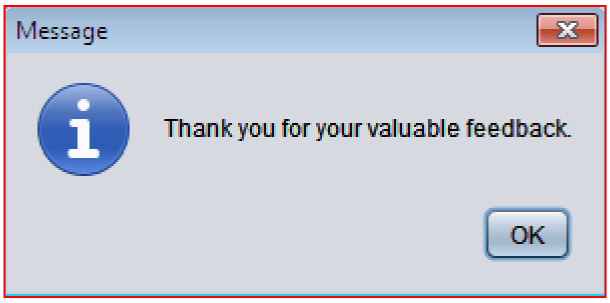


Your Name: sample

Rating: 1 2 3 4 [✓ 5]

Feedback: very user friendly

On submitting Feedback form,



Thank you for your valuable feedback.

Scope and Limitations

Currently, the scope of the project is to send textual messages between computers with the use of the application. It works like an e-mail system which on the larger scale uses the Internet, whereas our application uses the LAN network to connect. Currently, its features are listed as follows.

Features:

- Can send text messages to users.
- Can delete old messages and sent messages.
- The exact date and time of sending/receiving is also shown.
- Users who have forgotten their passwords can recover their accounts by using "Forgot Password" feature.
- The application's mail box automatically refreshes every few seconds to check for newly arrived mail, therefore manual-synchronizing is not required.
- In case there is a newly arrived message, an appropriate image shows up on the application's interface.
- There is almost no time lag in sending & receiving messages.
- The application is free to use, and is user friendly.
- Users can submit their feedback through the application's feedback form.

It is limited by various factors, listed as follows.

Limitations:

- Pictures, Videos and Audio messages cannot be sent.
- It can be used only by computers that are connected to a single LAN network.
- Users who have forgotten their user-ID currently have no means of getting back their account.
- The computer holding the database should be up for communication to happen.
- There is currently a word limit of 65000 characters in a single text message.
- Once messages are deleted, cannot be recovered back.

Bibliography

- Class-12 Informatics Practices, NCERT.
- Class-11 Informatics Practices, NCERT.
- All-In-One Java for Dummies, 4th Edition, by Doug Lowe.
- Pictures used for backgrounds and icons obtained from Google Images.