**PROJECT REPORT**

**ON**

**KIKON CHAT**

**SUBMITTED BY:**

**CHETAN D. PARMAR**

**AGAM D. PAREKH**

**DEPARTMENT OF B.S.C.I.T**

**AT**

**SVKM’S USHA PRAVIN GANDHI COLLEGE OF MANAGEMENT**

**VILE PARLE (W), MUMBAI-400056**

**AFFILIATED TO UNIVERSITY OF MUMBAI**

**2015-2016**

# Acknowledgement

We are pleased to present **"KIKON CHAT”** project and take this opportunity to express our profound gratitude to all those people who helped us in completion of this project.

We thank our college for providing us with excellent facilities that helped us to complete and present this project. We would also like to thank the staff members and lab assistants for permitting us to use computers in the lab as and when required.

We express our deepest gratitude towards our project guide for her valuable and timely advice during the various phases in our project. We would also like to thank her for providing us with all proper facilities and support as the project co-coordinator. We would like to thank her for support, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules.

**Table of Contents**

[Acknowledgement 2](#_Toc446635111)

Production Phase:

[Inception](#_Toc446635112) 6

[The Need](#_Toc446635113) 7

[Why Java](#_Toc446635114) 7

[EXISTING SYSTEM 8](#_Toc446635115)

[PROPOSED SYSTEM 9](#_Toc446635116)

[FEASIBILITY STUDY 10](#_Toc446635117)

[**Technical Feasibility** 10](#_Toc446635118)

[**Economic Feasibility** 10](#_Toc446635119)

[**Legal Feasibility** 10](#_Toc446635120)

[**Operational Analysis** 11](#_Toc446635121)

[**Schedule Feasibility** 11](#_Toc446635122)

[Elaboration 12](#_Toc446635123)

[WBS and Components 13](#_Toc446635124)

[Visual Modelling 14](#_Toc446635125)

[**Sequence Diagram:** 14](#_Toc446635126)

[**Use-Case Diagram:** 19](#_Toc446635127)

[**ACTIVITY DIAGRAM:** 20](#_Toc446635128)

Engineering Phase:

[Construction 21](#_Toc446635129)

[Graphical User Interface 23](#_Toc446635130)

[**Registration Form:-** 23](#_Toc446635131)

[**Login Page:-** 24](#_Toc446635132)

[**Edit Profile Page:-** 24](#_Toc446635133)

[**Profile Page:-** 25](#_Toc446635134)

[**Change Password:-** 26](#_Toc446635135)

[**Friends:-** 26](#_Toc446635136)

[**Personal Chat Window:-** 27](#_Toc446635137)

[**Groups:-** 28](#_Toc446635138)

[**All users List:-** 28](#_Toc446635139)

[**Make Group:-** 29](#_Toc446635140)

[**Group Chat Window:-** 29](#_Toc446635141)

[**Group Information:-** 30](#_Toc446635142)

[**Room Chat:-** 30](#_Toc446635143)

[**Room Chat Conversation Window:-** 31](#_Toc446635144)

[Source code 31](#_Toc446635145)

[**Chatapp.java (Registration)** 31](#_Toc446635146)

[**Backup.java** 42](#_Toc446635147)

[**Login.java** 44](#_Toc446635148)

[**Main.java (Main Window)** 49](#_Toc446635149)

[**Home.java(Profile page)** 58](#_Toc446635150)

[**MyFriends.java** 63](#_Toc446635151)

[**MyGroup.java** 69](#_Toc446635152)

[**FindFriend.java** 73](#_Toc446635153)

[**MakeGroup.java** 79](#_Toc446635154)

[**RoomChat\_StartPage.java** 91](#_Toc446635155)

[**RoomChat.java** 94](#_Toc446635156)

[**GroupChatWindow.java** 101](#_Toc446635157)

[**ChatWindow.java(Personal chat)** 111](#_Toc446635158)

[**DataAccess.java(Data Access Object)** 122](#_Toc446635159)

[**DatabaseConnection.java** 129](#_Toc446635160)

[**PersonalReader.java** 130](#_Toc446635161)

[**PriorityListener.java** 132](#_Toc446635162)

[**PersonalUser.java (server side Thread)** 134](#_Toc446635163)

[Transition](#_Toc446635164) 105

[Deployment Pre-requisites and Summary 142](#_Toc446635165)

[Future enhancements and releases 143](#_Toc446635166)

[BIBLIOGRAPHY 144](#_Toc446635167)

# Inception

Tasks

* Research and analyze basic idea
* Set goals, objectives
* Synthesize an initial demonstration

Objectives

* Common understanding between coordinators and myself on the scope of my project

Deliverables

* Project proposal/synopsis
* Initial demonstration

## The Need

|  |
| --- |
| **Why Java?** |

Java is an excellent language for developing cross platform desktop applications.It's just that nobody knows it. Most of Java's dominance is on the server side coupled with a little bit of love from the J2ME crowd.

Finally, a really great reason to pick Java (as opposed to .NET or Objective-C) is the ecosystem. If you write an application in Java today, you can seamlessly transition to Scala at any time. You also have the option to do scripting in Groovy or JRuby, not to mention taking advantage of the powerful concurrency abstractions in Clojure. Because of its portability and high-level nature, the JVM is increasingly the preferred target for language compilers and interpreters (even more than the CLR). Java is really at the center of that nexus, not really benefiting directly from the vast cloud of languages which swarm around it, but certainly deriving value from all of the interoperability, present and future.

Java VMs (virtual machines) are everywhere! They are in:

* Web browsers
* Cell phones
* PDAs
* Desktop Computers
* Web Servers
* Application Servers

|  |
| --- |
| **EXISTING SYSTEM** |

1. **Annoying ads**

Many applications continuously shows ads which interferes between application and user. The screen is partially covered by the ads and user is not able to enjoy the full screen feature.

2. **Request/Accept Mechanism:**

There is no request sending mechanism, without sending request anybody can send message to opposite person with the help of phone no or email-id and can also view their information available in that app.

3. **Prediction Mechanism**

There is no prediction mechanism in-built the application where it assist in fast reply to the respective sender.

4. **Room Chat**

There is no Room chat feature available in most of application where you can chat, learn, discuss about ongoing topic

|  |
| --- |
| **PROPOSED SYSTEM** |

The proposed system KikOn is a Java chat application with peer-peer chat, group chat, and room chat.

Once downloaded and installed the user needs to register and can enjoy the following features:

1. **Use of Data Access Object Pattern**

Data Access Object (DAO) is an Object that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer**.**

2. **Priority Feature**

There is Priority mechanism, if sender sends an important message but the receiver is offline. Sender can notify him to reply him back urgently using Priority Feature.

3. **Fast Typing Mechanism**

There is fast Typing mechanism in-built the application where it assist in fast reply to the respective sender.

|  |
| --- |
| FEASIBILITY STUDY |

**TELOS** refers to the five areas of feasibility -Technical, Economic, Legal, Operational, and Scheduling. All of these factors were analysed and studied. A highly brief summary of the report and results are given below.

#### **Technical Feasibility**

Technical aspects of the application are taken into account before deciding the development.The technology and the requirements are matched to decide upon many factors.

The application is developed in Net Beans IDE entirely with the support of Java Development Toolkit 8.The application does makes use of Data Structure to store user’s message in the computer and in server.

#### **Economic Feasibility**

Considering that this is an under-graduate project, ROI and profit in terms of money is not expected. However, I must and did take into consideration whether developing the product itself was within budget.

The cost of a suitable test device (or client), and other components was found to be within budget. The project is indeed economically feasible.

#### **Legal Feasibility**

No data is saved by the application without the explicit intention of the user. Therefore, data privacy and other similar legal issues do not apply.The developer, however, has no control of how the user makes use of the application and hence is not responsible for it and is absolved of any liabilities arising therefore. The product is legally feasible.

#### **Operational Analysis**

Operational Feasibility is to check how the software would perform at client side. Since the application is designed to function automatically the user would not have to perform any additional task.

The application is built in such a way that the user doesn’t need to worry about most of the things .The application thus is built user friendly, and works like a charm.

#### **Schedule Feasibility**

We had worked on similar project before, like simple chat using socket connection. Taking past experience as well as expert advice into consideration, the effort and schedule break up showed that the project was likely to be delivered on time.

# Elaboration

Tasks

* Thorough research of concepts and technologies
* Based on research, develop system design
* Identify modules, components
* Develop evaluation criteria (exit criteria)

Objectives

* Detailed design based on solid research

Deliverables

* UML Diagrams demonstrating system design

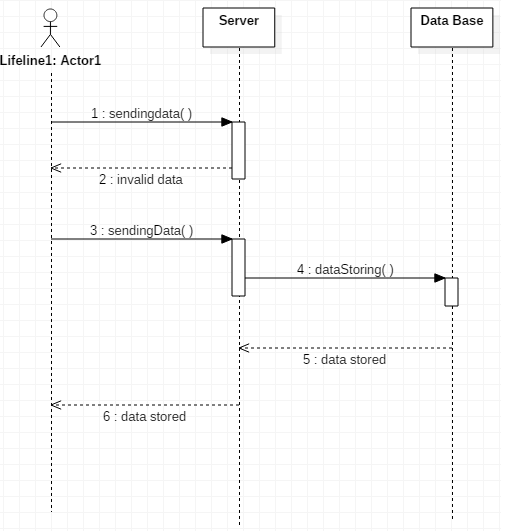
|  |
| --- |
| GANTT CHART |

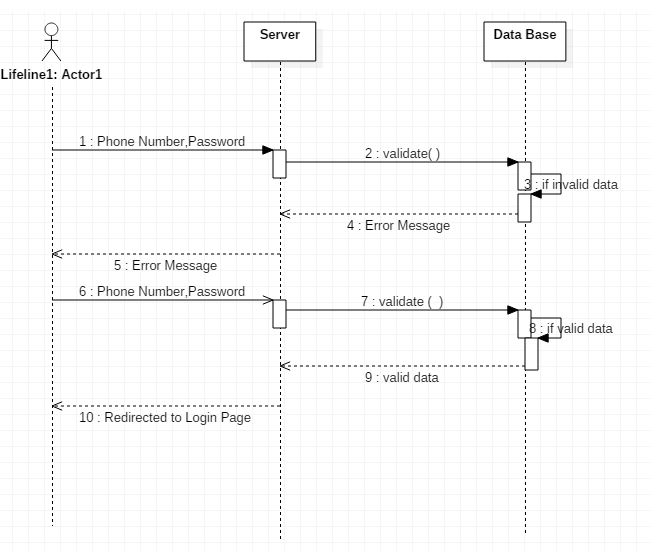
Gantt chart is the Graphical Representation about the progress of the project. Gantt chart shows the details about the how much time taken for any task**.**Gantt chart is a type of bar chart that illustrated the project schedule. It shows the start & finish of the project.Gantt chart shows the time taken by the task for each phase. Gantt chart decides the standard of the system. It is a useful tool for planning & scheduling projects.Gantt chart contains two columns. First is task and other is month (timing). In the task columns there are five options or phase find in the month’s columns shows the progress of the project.

|  |
| --- |
| Visual Modelling |

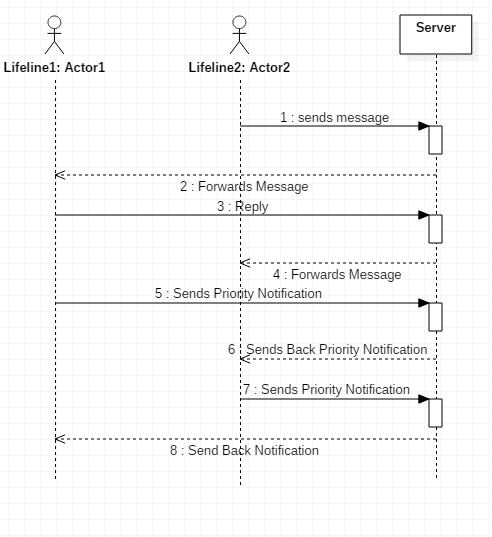
#### **Sequence Diagram:**

**Registration**

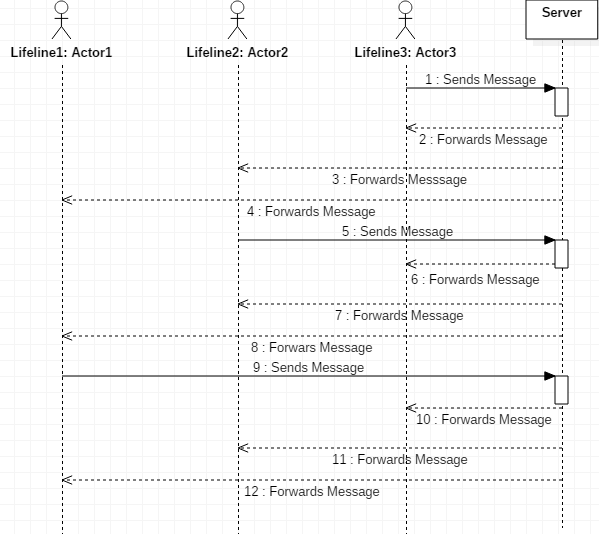
****

**Log-In Scenario**

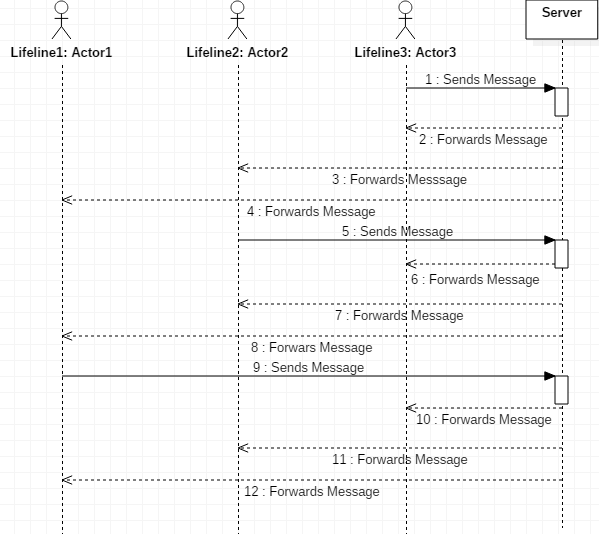
**Peer-Peer Chat**



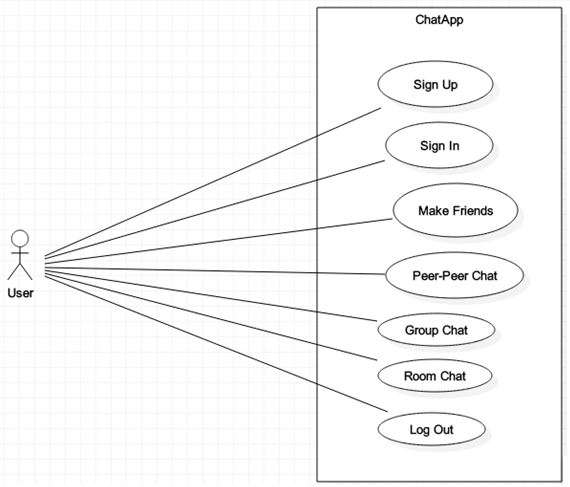
**Group Chat**



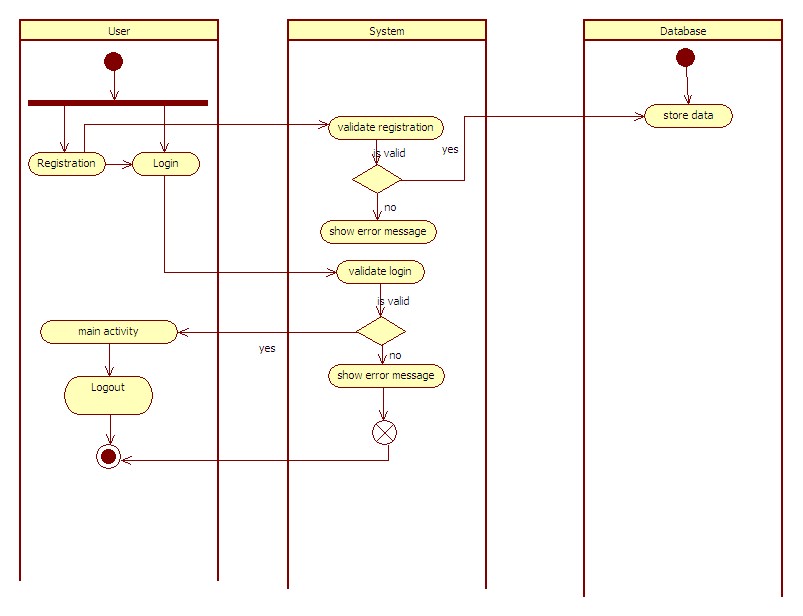
**Room Chat**



#### **Use-Case Diagram:**



#### **ACTIVITY DIAGRAM:**



# Construction

Tasks

* Coding and testing
* Experimentation and analysis

Objectives

* Finish software development

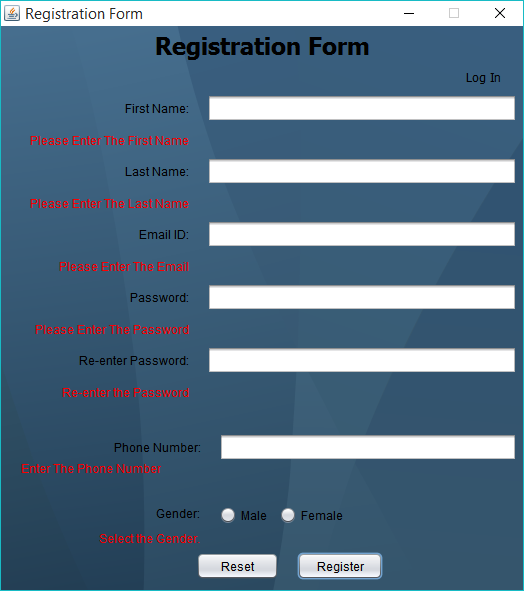
Deliverables

* Source code
* Product demonstration

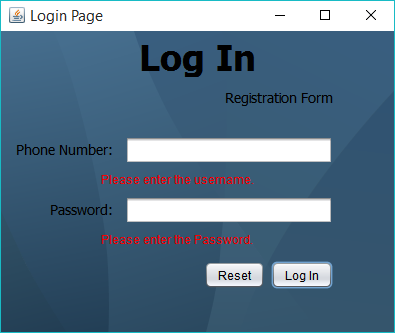
|  |
| --- |
| Graphical User Interface |

The following are a few screenshots of the application at work.

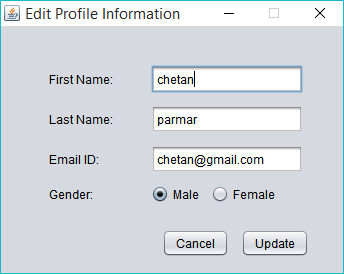
#### **Registration Form:-**



#### **Login Page:-**



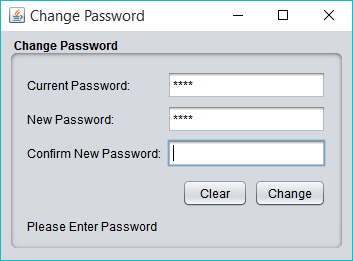
#### **Edit Profile Page:-**



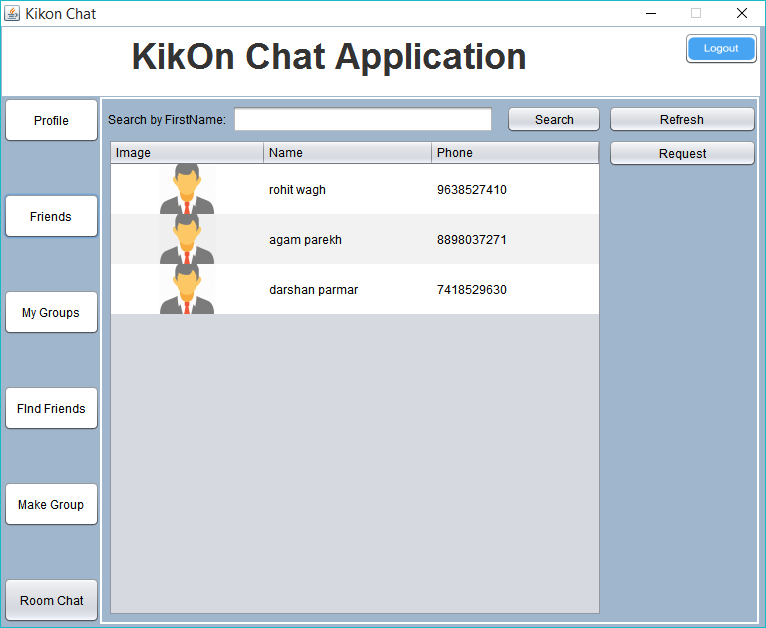
#### **Profile Page:-**



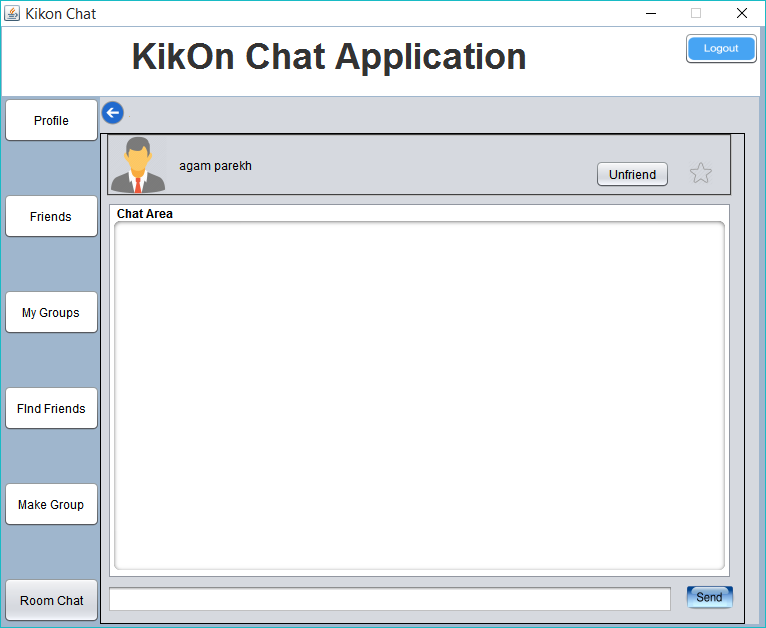
#### **Change Password:-**



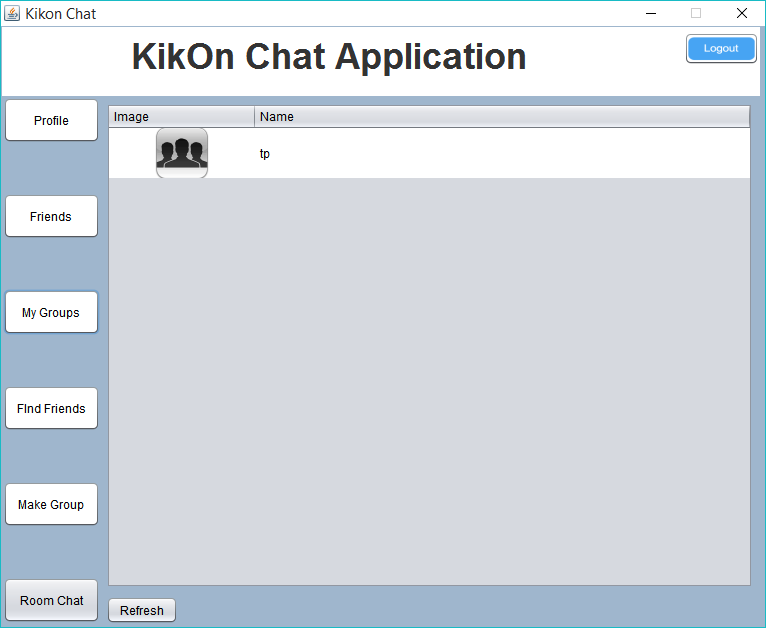
#### **Friends:-**



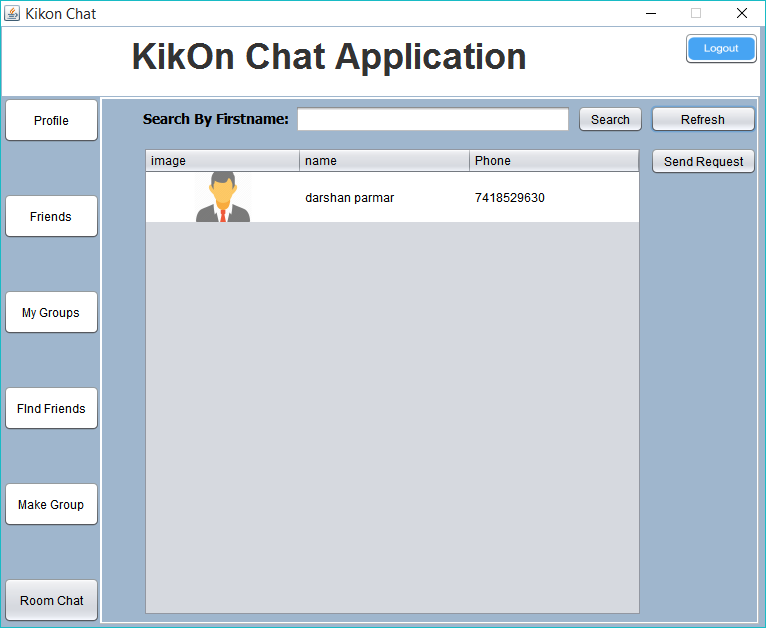
**Personal Chat Window:-**



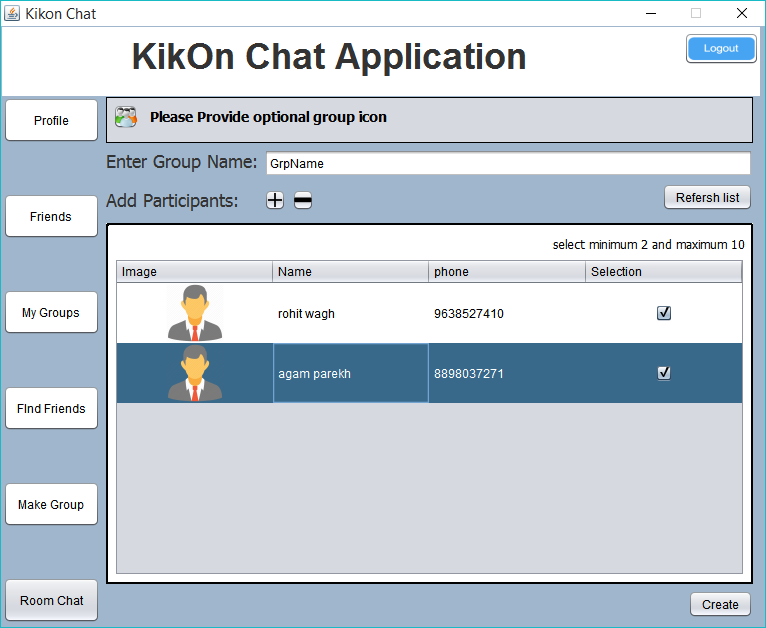
#### **Groups:-**

****

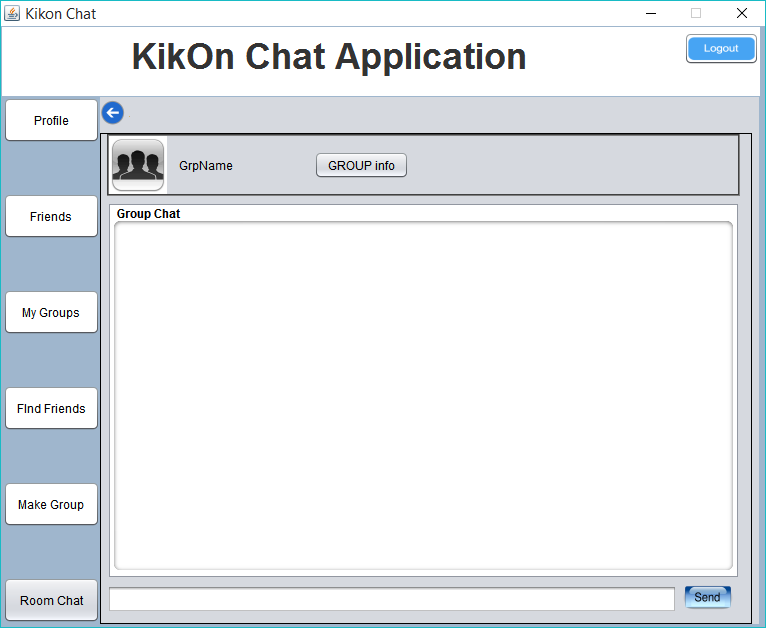
#### **All users List:-**

****

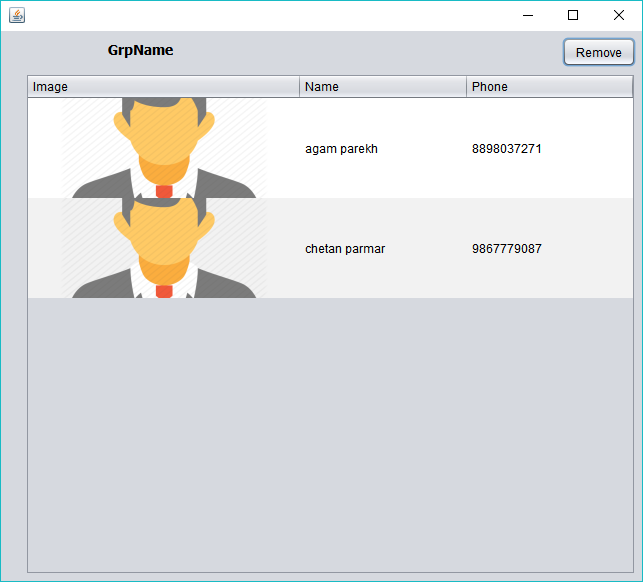
#### **Make Group:-**

****

#### **Group Chat Window:-**

****

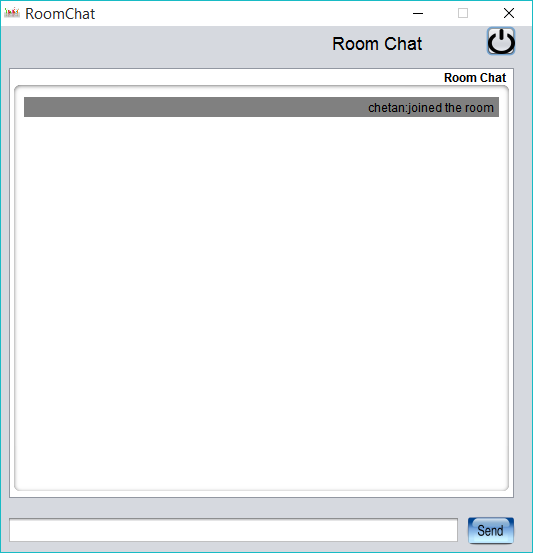
#### **Group Information:-**

****

#### **Room Chat:-**

****

#### **Room Chat Conversation Window:-**

****

|  |
| --- |
| Source code |
|  |

#### **Chatapp.java (Registration)**

package chatapp;

import com.sun.glass.events.KeyEvent;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.net.InetAddress;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class Chatapp extends javax.swing.JFrame {

Connection conn;

String phone = "";// required to create friend request table

String name = "";// required to create friend request table

LoginPage login;

public Chatapp() {

super("Registration Form");

this.setResizable(false);

initComponents();

login = new LoginPage();

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

this.setLocation(dim.width / 2 - this.getSize().width / 2, dim.height / 2 - this.getSize().height / 2);

jLabel8.setForeground(Color.red);

jLabel9.setForeground(Color.red);

jLabel10.setForeground(Color.red);

jLabel11.setForeground(Color.red);

jLabel12.setForeground(Color.red);

jLabel13.setForeground(Color.red);

jLabel16.setForeground(Color.red);

buttonGroup1.add(jRadioButton1);

buttonGroup1.add(jRadioButton2);

}

@SuppressWarnings("unchecked")

private void initComponents() {

buttonGroup1 = new javax.swing.ButtonGroup();

jDesktopPane1 = new javax.swing.JDesktopPane();

jLabel2 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jTextField6 = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jPasswordField1 = new javax.swing.JPasswordField();

jPasswordField2 = new javax.swing.JPasswordField();

jLabel14 = new javax.swing.JLabel();

jLabel15 = new javax.swing.JLabel();

jRadioButton1 = new javax.swing.JRadioButton();

jRadioButton2 = new javax.swing.JRadioButton();

jLabel16 = new javax.swing.JLabel();

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

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel2.setText("Registration Form");

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel1.setText("First Name:");

jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel3.setText("Last Name:");

jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel4.setText("Password:");

jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel5.setText("Re-enter Password:");

jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

jTextField1KeyTyped(evt);

}

});

jTextField3.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

jTextField3KeyTyped(evt);

}

});

jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel6.setText("Email ID:");

jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel7.setText("Phone Number:");

jTextField6.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

jTextField6KeyTyped(evt);

}

});

jLabel13.setToolTipText("");

jButton1.setText("Reset");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Register");

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

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

jButton2ActionPerformed(evt);

}

});

jLabel14.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jLabel14.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel14.setText("Log In");

jLabel14.addMouseListener(new java.awt.event.MouseAdapter() {

public void mousePressed(java.awt.event.MouseEvent evt) {

jLabel14MousePressed(evt);

}

});

jLabel15.setText("Gender:");

jRadioButton1.setText("Male");

jRadioButton2.setText("Female");

}

private void jLabel14MousePressed(java.awt.event.MouseEvent evt) {

this.setVisible(false);

login.setVisible(true);

}

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

int count = validation();

if (count == 7) {

conn = DatabaseConnection.getdatabaseConnection();

try {

if (isNewContact()) {

String gender = (jRadioButton1.isSelected() ? "male" : "female");

phone = jTextField6.getText();

DataAccess db = new DataAccess();

db.register(jTextField1.getText(),

jTextField3.getText(),

jTextField2.getText(),

jPasswordField1.getText(), phone, gender,conn);

createTable();

} else {

JOptionPane.showMessageDialog(null, "User with this Phone Number is already registered\n"

+ "Enter Unregistered Phone Number.");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "There is some problem in"

+ " storing data.\n Please try again." + ex.getMessage());

}

}

}

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

reset();

}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {

char c = evt.getKeyChar();

if (!(Character.isAlphabetic(c) || c == KeyEvent.VK\_BACKSPACE || c == KeyEvent.VK\_DELETE)) {

evt.consume();

this.getToolkit().beep();

}

}

private void jTextField3KeyTyped(java.awt.event.KeyEvent evt) {

char c = evt.getKeyChar();

if (!(Character.isAlphabetic(c) || c == KeyEvent.VK\_BACKSPACE || c == KeyEvent.VK\_DELETE)) {

evt.consume();

this.getToolkit().beep();

}

}

private void jTextField6KeyTyped(java.awt.event.KeyEvent evt) {

char c = evt.getKeyChar();

if (jTextField6.getText().length() > 9) {

evt.consume();

this.getToolkit().beep();

}

if (!(Character.isDigit(c) || c == KeyEvent.VK\_BACKSPACE || c == KeyEvent.VK\_DELETE)) {

evt.consume();

this.getToolkit().beep();

}

}

private int validation() {

int count = 0;

if (jTextField1.getText().equals("")) {

jLabel8.setText("Please Enter The First Name");

} else {

count++;

jLabel8.setText("");

}

if (jTextField3.getText().equals("")) {

jLabel9.setText("Please Enter The Last Name");

} else {

count++;

jLabel9.setText("");

}

if (jTextField2.getText().equals("")) {

jLabel10.setText("Please Enter The Email");

} else {

if (!jTextField2.getText().matches("^[\\w-\_\\.+]\*[\\w-\_\\.]\\@([\\w]+\\.)+[\\w]+[\\w]$")) {

jLabel10.setText("Please Enter Valid Email");

} else {

count++;

jLabel10.setText("");

}

}

if (jPasswordField1.getText().equals("")) {

jLabel11.setText("Please Enter The Password\n");

} else {

count++;

jLabel11.setText("");

}

if (jPasswordField2.getText().equals("")) {

jLabel12.setText("Re-enter the Password\n");

} else {

String pass1 = jPasswordField1.getText();

String pass2 = jPasswordField2.getText();

if (!pass1.equals(pass2)) {

jLabel12.setText("Mis Match Password");

} else {

count++;

jLabel12.setText("");

}

}

if (jTextField6.getText().equals("")) {

jLabel13.setText("Enter The Phone Number");

} else {

if (jTextField6.getText().length() == 10) {

count++;

jLabel13.setText("");

} else {

jLabel13.setText("enter 10 digit mobile number");

}

}

if (jRadioButton1.isSelected() || jRadioButton2.isSelected()) {

count++;

jLabel16.setText("");

} else {

jLabel16.setText("Select the Gender.");

}

return count;

}

private void reset() {

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jPasswordField1.setText("");

jPasswordField2.setText("");

jTextField6.setText("");

buttonGroup1.clearSelection();

}

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch(Exception e){}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new Chatapp().setVisible(true);

}

});

}

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JDesktopPane jDesktopPane1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

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.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JPasswordField jPasswordField2;

private javax.swing.JRadioButton jRadioButton1;

private javax.swing.JRadioButton jRadioButton2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField6;

// End of variables declaration

private void createTable() throws SQLException {

String frdlist = "create table frdlist" + phone + " (phone varchar(10),priority int)";

String grplist = "create table grplist" + phone + " (name varchar(10))";

//the above grplist+ phone is the table of current logged in user

// in the table, the name of all the tables will be stored where this user belong to that grp

Statement st = conn.createStatement();

st.execute(frdlist);

st.execute(grplist);

}

private boolean isNewContact() throws SQLException {

String all = "select phone from register";

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery(all);

String p = jTextField6.getText();

while (rs.next()) {

if (p.equals(rs.getString("phone"))) {

return false;

}

}

return true;

}

}

#### **Backup.java**

package chatapp;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.Serializable;

import java.util.Hashtable;

public class Backup implements Serializable {

Hashtable<String, MessageStack> messagebackup;

private File file;

public Backup() {

messagebackup = new Hashtable<String, MessageStack>();

new MessageStack();

file = new File("C:\\kikon\\" + Main.phone + ".ser");

}

public boolean store() {

try {

FileOutputStream fos = new FileOutputStream(file);

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(messagebackup);

oos.close();

fos.close();

} catch (IOException ex) {

return false;

}

return true;

}

public void load() {

try {

FileInputStream fis = new FileInputStream(file);

ObjectInputStream ois = new ObjectInputStream(fis);

Hashtable<String, MessageStack> messagebackup1 = new Hashtable<String, MessageStack>();

messagebackup1 = (Hashtable<String, MessageStack>) ois.readObject();

ois.close();

fis.close();

messagebackup = messagebackup1;

} catch (IOException e) {

} catch (ClassNotFoundException e) {

}

}

public boolean fileExist() {

return file.exists();

}

}

#### **Login.java**

package chatapp;

import com.sun.glass.events.KeyEvent;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.net.InetAddress;

import java.net.UnknownHostException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class LoginPage extends javax.swing.JFrame {

Connection conn;

public LoginPage() {

initComponents();

this.setTitle("Login Page");

conn=DatabaseConnection.getdatabaseConnection();

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

this.setLocation(dim.width/2-this.getSize().width/2, dim.height/2-this.getSize().height/2);

jLabel4.setForeground(Color.red);

jLabel5.setForeground(Color.red);

}

@SuppressWarnings("unchecked")

private void initComponents() {

jDesktopPane1 = new javax.swing.JDesktopPane();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jPasswordField1 = new javax.swing.JPasswordField();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

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

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("Log In");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel2.setText("Phone Number:");

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel3.setText("Password:");

jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

jTextField1KeyTyped(evt);

}

});

jButton1.setText("Reset");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Log In");

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

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

jButton2ActionPerformed(evt);

}

});

jLabel6.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabel6.setText("Registration Form");

jLabel6.addMouseListener(new java.awt.event.MouseAdapter() {

public void mousePressed(java.awt.event.MouseEvent evt) {

jLabel6MousePressed(evt);

}

});

}

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

reset();

}

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

int count=validateLogin();

if(count==2){

try {

DataAccess db = new DataAccess();

String phone = jTextField1.getText();

String name = db.getName(phone,conn);

if(db.isLoginDataValid(phone, jPasswordField1.getText(),conn)){

this.setVisible(false);

Main main=new Main(conn,phone,name);

}

else

throw new SQLException();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null,"Invalid username "

+ "or password\n");

}

}

}

void reset(){

jTextField1.setText("");

jPasswordField1.setText("");

}

private void jLabel6MousePressed(java.awt.event.MouseEvent evt) {

this.setVisible(false);

Chatapp reg=new Chatapp();

reg.setVisible(true);

}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

char c=evt.getKeyChar();

if(jTextField1.getText().length()>9 ){

evt.consume();

this.getToolkit().beep();

}

if(!(Character.isDigit(c) || c== KeyEvent.VK\_BACKSPACE || c == KeyEvent.VK\_DELETE)){

evt.consume();

this.getToolkit().beep();

}

}

private int validateLogin(){

int count=0;

if(jTextField1.getText().equals("")){

jLabel4.setText("Please enter the username.");

}else{

count++;

jLabel4.setText("");

}

if(jPasswordField1.getText().equals("")){

jLabel5.setText("Please enter the Password.");

}else{

count++;

jLabel5.setText("");

}

return count;

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JDesktopPane jDesktopPane1;

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.JPasswordField jPasswordField1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

#### **Main.java (Main Window)**

package chatapp;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.GridLayout;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.io.Serializable;

import java.net.Socket;

import java.sql.Connection;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Hashtable;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JSplitPane;

import javax.swing.border.Border;

import javax.swing.border.LineBorder;

class MessageStack implements Serializable {

ArrayList<String> message;

public MessageStack() {

message = new ArrayList<String>();

}

}

public class Main extends javax.swing.JFrame {

Statement st;

Connection conn;

static String phone;

JSplitPane splitPanev;

JSplitPane splitPaneh;

Home home;

MyFriends myFriends;

MakeGroup makeGroup;

FindFriend findFriend;

MyGroup myGroup;

Image btn,btn6;

Socket soc;

PrintWriter nos;

BufferedReader nis;

Backup backup;

RoomChat\_StartPage startPage;

String name;

PersonalReader reader;

Socket personalsoc;

Border border;

JPanel left;

Main\_top\_panel top;

static String ip;

PriorityListener prioritylistener;

public Main(Connection conn, String phone, String name) {

this.getContentPane().setBackground(new Color(159, 182, 205));

initComponents();

this.setTitle("Kikon Chat");

this.st = st;

this.conn = conn;

this.phone = phone;

this.name = name;

ip = "localhost";

backup = new Backup();

if (backup.fileExist()) {

backup.load();

}

try {

soc = new Socket(ip, 9000);

} catch (IOException ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

}

reader = new PersonalReader(conn, backup, phone);

reader.start();

System.out.println("reader started");

home = new Home(this, conn, st, phone);

home.setBackground(new Color(159, 182, 205));

myFriends = new MyFriends(this, phone, conn, backup);

myFriends.setBackground(new Color(159, 182, 205));

makeGroup = new MakeGroup(this, phone, soc, this.name);

makeGroup.setBackground(new Color(159, 182, 205));

findFriend = new FindFriend(conn, phone);

findFriend.setBackground(new Color(159, 182, 205));

myGroup = new MyGroup(this, conn, phone, soc, backup);

myGroup.setBackground(new Color(159, 182, 205));

splitPanev = new JSplitPane();

splitPaneh = new JSplitPane();

startPage = new RoomChat\_StartPage(this.name, this);

initImageButton();

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

this.setLocation(dim.width / 2 - this.getSize().width / 2 - 70, dim.height / 2 - this.getSize().height / 2 - 55);

this.setSize(770, 630);

this.setResizable(false);

this.setVisible(true);

top = new Main\_top\_panel();

top.setBackground(new Color(255, 255, 255));

top.setSize(450, 100);

top.setVisible(true);

border = new LineBorder(Color.WHITE, 0);

jButton6.setBorder(border);

JPanel left = new JPanel(new GridLayout(6, 1, 50, 50));

left.setBackground(new Color(159, 182, 205));

left.add(jButton1);

left.add(jButton2);

left.add(jButton3);

left.add(jButton4);

left.add(jButton5);

left.add(jButton7);

splitPanev.setSize(760, 600);

splitPanev.setDividerSize(0);

splitPanev.setDividerLocation(70);

splitPanev.setOrientation(JSplitPane.VERTICAL\_SPLIT);

splitPanev.setTopComponent(top);

splitPaneh = new JSplitPane();

splitPaneh.setSize(760, 600);

splitPaneh.setDividerSize(0);

splitPaneh.setDividerLocation(98);

splitPaneh.setLeftComponent(left);

splitPaneh.setRightComponent(home);

splitPanev.setBottomComponent(splitPaneh);

this.add(splitPanev);

startPriorityListener();

setMouseListener();

}

public void startPriorityListener() {

try {

prioritylistener = new PriorityListener(this.conn, phone);

prioritylistener.start();

} catch (SQLException ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void setMouseListener() {

jButton6.addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent me) {

try {

btn = ImageIO.read(new File("src/logout1.png"));

jButton6.setIcon(new ImageIcon(btn.getScaledInstance(75, 33, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println(ex.toString());

}

}

public void mouseExited(MouseEvent me) {

try {

btn = ImageIO.read(new File("src/logout2.png"));

jButton6.setIcon(new ImageIcon(btn.getScaledInstance(75, 33, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println(ex.toString());

}

}

});

}

@SuppressWarnings("unchecked")

private void initComponents() {

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);

jButton1.setBackground(new java.awt.Color(255, 255, 255));

jButton1.setText("Profile");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(255, 255, 255));

jButton2.setText("Friends");

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

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

jButton2ActionPerformed(evt);

}

});

jButton3.setBackground(new java.awt.Color(255, 255, 255));

jButton3.setText("My Groups");

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

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

jButton3ActionPerformed(evt);

}

});

jButton4.setBackground(new java.awt.Color(255, 255, 255));

jButton4.setText("FInd Friends");

jButton4.addActionListener(new java.awt.event.ActionListener() {

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

jButton4ActionPerformed(evt);

}

});

jButton5.setBackground(new java.awt.Color(255, 255, 255));

jButton5.setText("Make Group");

jButton5.addActionListener(new java.awt.event.ActionListener() {

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

jButton5ActionPerformed(evt);

}

});

jButton6.setBackground(new java.awt.Color(255, 255, 255));

jButton6.addActionListener(new java.awt.event.ActionListener() {

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Room Chat");

jButton7.setToolTipText("");

jButton7.addActionListener(new java.awt.event.ActionListener() {

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

jButton7ActionPerformed(evt);

}

});

}

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

splitPaneh.setRightComponent(home);

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

int isYES = JOptionPane.showConfirmDialog(null, "wanna logout?",

"Choose", JOptionPane.YES\_NO\_OPTION);

if (isYES == JOptionPane.YES\_OPTION) {

try {

DataAccess db = new DataAccess();

db.logout(phone, conn);

backup.store();

this.dispose();

} catch (SQLException ex) {

System.out.println(ex.toString());

}

}

}

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

splitPaneh.setRightComponent(myFriends);

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

splitPaneh.setRightComponent(findFriend);

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

splitPaneh.setRightComponent(makeGroup);

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

splitPaneh.setRightComponent(startPage);

}

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

splitPaneh.setRightComponent(myGroup);

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JButton jButton6;

private javax.swing.JButton jButton7;

// End of variables declaration

private void initImageButton() {

try {

btn6 = ImageIO.read(new File("src/logout2.png"));

jButton6.setIcon(new ImageIcon(btn6.getScaledInstance(75, 33, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println(ex.toString());

}

}

}

#### **Home.java(Profile page)**

package chatapp;

import java.awt.Image;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Vector;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.filechooser.FileNameExtensionFilter;

import javax.swing.table.DefaultTableModel;

public class Home extends javax.swing.JPanel {

Connection conn = null;

Main main = null;

DefaultTableModel dtm;

Image img;

String phone = "";// use to get info for current logged in user

Changepass changepassword;

DataAccess db;

public Home(Main main, Connection conn, Statement st, String phone) {

dtm = new DefaultTableModel() {

@Override

public boolean isCellEditable(int row, int column) {

return false;

}

};

initComponents();

db = new DataAccess();

this.main = main;

this.phone = phone;

this.conn = conn;

jTable1.setTableHeader(null);

jTable1.setFocusable(false);

jTable1.setRowSelectionAllowed(false);

String header[] = new String[]{"image", "name"};

dtm.setColumnIdentifiers(header);

jTable1.setRowHeight(20);

try {

img = ImageIO.read(new File("src/usericon.png"));

jLabel10.setIcon(new ImageIcon(img.getScaledInstance(208, 168, Image.SCALE\_SMOOTH)));

img = ImageIO.read(new File("src/edit.png"));

jButton1.setIcon(new ImageIcon(img.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

jButton2.setIcon(new ImageIcon(img.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

jButton3.setIcon(new ImageIcon(img.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

displayData();

} catch (IOException ex) {

}

}

@SuppressWarnings("unchecked")

private void initComponents() {

jButton1 = new javax.swing.JButton();

jLabel10 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jPanel1 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton3 = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createEtchedBorder(), new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED)), "Profile", javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Tahoma", 1, 18))); // NOI18N

setForeground(new java.awt.Color(255, 255, 255));

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

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

jButton1ActionPerformed(evt);

}

});

jLabel10.setText("jLabel10");

jButton2.setToolTipText("");

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

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

jButton2ActionPerformed(evt);

}

});

jLabel1.setText("Edit Profile Pic");

jLabel2.setText("Chage Password");

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Information", javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.DEFAULT\_POSITION));

jTable1.setModel(dtm);

jScrollPane1.setViewportView(jTable1);

}

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

main.setEnabled(false);

Editprofile edit = new Editprofile(phone, main,conn);

edit.setVisible(true);

edit.setResizable(false);

}

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

JFileChooser file = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter("\*.images", "jpg", "gif");

file.addChoosableFileFilter(filter);

int result = file.showOpenDialog(null);

if (result == JFileChooser.APPROVE\_OPTION) {

try {

File f = file.getSelectedFile();

FileInputStream fis = new FileInputStream(f);

System.out.println(f);

Image bi = ImageIO.read(f);

jLabel10.setText("");

jLabel10.setIcon(new ImageIcon(bi.getScaledInstance(208, 168, Image.SCALE\_SMOOTH)));

int i = db.updateProfilePic(phone, conn, fis);

if (i > 0) {

System.out.println("image inserted into the database..");

} else {

System.out.println("image not inserted");

}

} catch (IOException ex) {

} catch (SQLException ex) {

}

}

}

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

main.setEnabled(false);

this.changepassword = new Changepass(conn,main, phone);

changepassword.setVisible(true);

}

public void displayData(){

db.displayData(dtm, phone, conn, jLabel10);

}

// 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 jLabel10;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration

}

#### **MyFriends.java**

package chatapp;

import java.io.PrintWriter;

import java.net.Socket;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.ListSelectionModel;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

public class MyFriends extends javax.swing.JPanel {

DefaultTableModel dtm;

Main main;

String phone;

Connection conn;

Statement st;

static ImageIcon img;

String username;

Backup backup;

DataAccess db;

String query;

MyFriends myfriends ;

public MyFriends(final Main main,final String phone,final Connection conn,final Backup backup) {

this.main=main;

this.phone=phone;

this.conn=conn;

this.backup=backup;

this.myfriends = this;

db = new DataAccess();

query="Select r.firstname,r.lastname,r.image,r.phone from register r,frdlist"+this.phone+" fl where r.phone=fl.phone";

dtm=new DefaultTableModel(){

@Override

public boolean isCellEditable(int row, int column) {

return false;

}

@Override

// Returning the Class of each column will allow different

// renderers to be used based on Class

public Class getColumnClass(int column)

{

return getValueAt(0, column).getClass();

}

};

initComponents();

String header[] = new String[] { "Image", "Name", "Phone" };

dtm.setColumnIdentifiers(header);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowHeight(50);

jTable1.getColumnModel().getColumn(0).setPreferredWidth(60);

jTable1.setAutoResizeMode(JTable.AUTO\_RESIZE\_LAST\_COLUMN);

try {

displayUsers(null);

} catch (SQLException ex) {

// System.out.println(ex.toString());

}

jTable1.getSelectionModel().addListSelectionListener(new ListSelectionListener(){

@Override

public void valueChanged(ListSelectionEvent e) {

ChatWindow c=new ChatWindow(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString(),conn,main,phone,backup);

myfriends.removeAll();

main.splitPaneh.setRightComponent(c);

}

});

}

private void displayUsers(String name) throws SQLException{

db.displayUser(conn, dtm,name,img, query);

}

@SuppressWarnings("unchecked")

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton5 = new javax.swing.JButton();

setBackground(new java.awt.Color(255, 255, 255));

setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, java.awt.Color.white, java.awt.Color.white, java.awt.Color.white, java.awt.Color.white));

setPreferredSize(new java.awt.Dimension(539, 403));

jLabel1.setText("Search by FirstName:");

jTextField1.addActionListener(new java.awt.event.ActionListener() {

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

jTextField1ActionPerformed(evt);

}

});

jButton1.setText("Search");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Refresh");

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

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

jButton2ActionPerformed(evt);

}

});

jTable1.setModel(dtm);

jScrollPane1.setViewportView(jTable1);

jButton5.setText("Request");

jButton5.addActionListener(new java.awt.event.ActionListener() {

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

jButton5ActionPerformed(evt);

}

});

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

main.setEnabled(false);

Request req=new Request(main,phone,conn);

req.setVisible(true);

}

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

// TODO add your handling code here:

if(jTextField1.getText().equals("")){

JOptionPane.showMessageDialog(null,"Please enter firstname.");

}

else{

try {

String fname=jTextField1.getText();

jTextField1.setText("");

displayUsers(fname);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null,"sorry NO DATA found!");

}

}

}

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

// TODO add your handling code here:

try {

displayUsers(null);

} catch (SQLException ex) {

}

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton5;

private javax.swing.JLabel jLabel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

#### **MyGroup.java**

package chatapp;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.net.Socket;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.ImageIcon;

import javax.swing.JTable;

import javax.swing.ListSelectionModel;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

public class MyGroup extends javax.swing.JPanel {

Main main;

Connection conn;

DefaultTableModel dtm;

Statement st;

String phone;

static ImageIcon img;

Socket soc;

String selectedgrpName;

Backup backup;

public MyGroup(final Main main, final Connection conn, final String phone, final Socket soc, final Backup backup) {

this.main = main;

this.conn = conn;

this.phone = phone;

this.soc = soc;

this.backup = backup;

dtm = new DefaultTableModel() {

@Override

public boolean isCellEditable(int row, int column) {

return false;

}

@Override

// Returning the Class of each column will allow different

// renderers to be used based on Class

public Class getColumnClass(int column) {

return getValueAt(0, column).getClass();

}

};

String header[] = new String[]{"Image", "Name"};

initComponents();

dtm.setColumnIdentifiers(header);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowHeight(50);

jTable1.getColumnModel().getColumn(0).setPreferredWidth(50);

jTable1.getColumnModel().getColumn(1).setPreferredWidth(400);

jTable1.setAutoResizeMode(JTable.AUTO\_RESIZE\_LAST\_COLUMN);

try {

displayGrp();

} catch (SQLException ex) {

System.out.println(ex.toString());

}

jTable1.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

@Override

public void valueChanged(ListSelectionEvent e) {

try {

st = conn.createStatement();

selectedgrpName = jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString();

String port = "select portnumber from groupinfo where name='" + selectedgrpName + "'";

ResultSet rs = st.executeQuery(port);

rs.next();

int getport = rs.getInt("portnumber");

GroupChatWindow c = new GroupChatWindow(getport, conn, main, phone, soc, backup, selectedgrpName);

main.splitPaneh.setRightComponent(c);

} catch (SQLException ex) {

Logger.getLogger(MyGroup.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

jTable1.addMouseListener(new MouseAdapter() {

@Override

public void mouseMoved(MouseEvent e) {

try {

displayGrp();

} catch (SQLException ex) {

Logger.getLogger(MyGroup.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

}

private void initComponents() {

jScrollPane2 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jTable1.setModel(dtm);

jTable1.setAutoResizeMode(javax.swing.JTable.AUTO\_RESIZE\_LAST\_COLUMN);

jScrollPane2.setViewportView(jTable1);

jButton1.setText("Refresh");

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

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

jButton1ActionPerformed(evt);

}

});

}

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

try {

displayGrp();

} catch (SQLException ex) {

Logger.getLogger(MyGroup.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void displayGrp() throws SQLException {

dtm.setRowCount(0);

String query = "Select gl.name,gi.image from grplist" + this.phone + " gl,groupinfo gi where gi.name=gl.name";

st = conn.createStatement();

ResultSet rs = st.executeQuery(query);

while (rs.next()) {

String name = rs.getString("name");

Blob imgData = rs.getBlob("image");

img = FindFriend.getImage(imgData);

dtm.addRow(new Object[]{img, name});

}

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTable jTable1;

// End of variables declaration

}

#### **FindFriend.java**

package chatapp;

import java.awt.Image;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import javax.swing.ListSelectionModel;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

public class FindFriend extends javax.swing.JPanel {

DefaultTableModel dtm;

Connection conn;

Statement st;

ResultSet rs;

String phone = "";

static ImageIcon img;

String query;

DataAccess db;

public FindFriend(final Connection conn, final String phone) {

this.phone = phone;

db = new DataAccess();

//query to display all the record i.e all the registerd users

query = "Select \* from register where not phone='" + this.phone + "' and phone not in (select phone from frdlist" + this.phone + ")";

dtm = new DefaultTableModel() {

@Override

public boolean isCellEditable(int row, int column) {

return false;

}

@Override

// Returning the Class of each column will allow different

// renderers to be used based on Class

public Class getColumnClass(int column) {

return getValueAt(0, column).getClass();

}

};

initComponents();

this.conn = conn;

String header[] = new String[]{"image", "name", "Phone"};

dtm.setColumnIdentifiers(header);

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.setRowHeight(50);

jTable1.getColumnModel().getColumn(0).setPreferredWidth(60);

try {

displayUsers(null);

} catch (SQLException ex) {

System.out.println(ex.toString());

}

}

private void displayUsers(String name) throws SQLException {

db.displayUser(conn, dtm, name, img, query);

}

private void initComponents() {

jLayeredPane1 = new javax.swing.JLayeredPane();

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton3 = new javax.swing.JButton();

setBackground(new java.awt.Color(255, 255, 255));

setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, java.awt.Color.white, java.awt.Color.white, java.awt.Color.lightGray, java.awt.Color.white));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jLabel1.setText("Search By Firstname:");

jButton1.setText("Search");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Refresh");

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

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

jButton2ActionPerformed(evt);

}

});

jTable1.setModel(dtm);

jScrollPane1.setViewportView(jTable1);

jButton3.setText("Send Request");

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

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

jButton3ActionPerformed(evt);

}

});

}

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

if (jTextField1.getText().equals("")) {

JOptionPane.showMessageDialog(null, "Please enter firstname.");

} else {

try {

String fname = jTextField1.getText();

jTextField1.setText("");

displayUsers(fname);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "sorry NO DATA found!");

}

}

}

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

try {

displayUsers(null);

} catch (SQLException ex) {

}

}

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

//sending friend request

Statement st = null;

ResultSet rs;

String str ="",phone2="";

try{

st=conn.createStatement();

phone2=jTable1.getValueAt(jTable1.getSelectedRow(),2).toString();

str="select sender from request where receiver='"+phone2+"' and sender='"+phone+"'";

rs=st.executeQuery(str);

rs.next();

str=rs.getString("sender");

}catch(SQLException s){System.out.println("ERROR"+s.toString());}

catch(ArrayIndexOutOfBoundsException s){System.out.println("ERROR"+s.toString());}

if(jTable1.getSelectedRow()>=0){

if(str.equals(phone)){

int isYES=JOptionPane.showConfirmDialog(null,"Request Already Send , wannna Cancel?","Choose",JOptionPane.YES\_NO\_OPTION);

if(isYES==JOptionPane.YES\_OPTION){

try{

str="delete from request where sender='"+phone2+"' and receiver="+phone;

st.execute(str);

JOptionPane.showMessageDialog(null,"Request has been cancel");

}

catch(SQLException s1){System.out.println(s1.toString());}

}

}

else{

str="insert into request values('"+phone+"','"+phone2+"')";

try {

st.execute(str);

} catch (SQLException ex) { }

JOptionPane.showMessageDialog(null,"request has sent...");

}

}

else

JOptionPane.showMessageDialog(null, "Please Select the person you want to send request to!");

}

public static ImageIcon getImage(Blob imgData) {

if (imgData != null) {

try {

File tmpFile = new File("tmpImage");

FileOutputStream fos = new FileOutputStream(tmpFile);

fos.write(imgData.getBytes(1L, (int) imgData.length()));

fos.close();

Image image = ImageIO.read(tmpFile);

img = new ImageIcon(image.getScaledInstance(58, 58, Image.SCALE\_SMOOTH));

} catch (FileNotFoundException ex) {

JOptionPane.showMessageDialog(null, "Image not found..", "Load Error", JOptionPane.ERROR\_MESSAGE);

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, "Failed To Load Image Data", "Load Error", JOptionPane.ERROR\_MESSAGE);

} catch (SQLException ex) {

}

}

return img;

}

// 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.JLayeredPane jLayeredPane1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

#### **MakeGroup.java**

package chatapp;

import java.awt.Color;

import java.awt.Image;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.net.Socket;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTable;

import javax.swing.ListSelectionModel;

import javax.swing.filechooser.FileNameExtensionFilter;

import javax.swing.table.DefaultTableModel;

public class MakeGroup extends javax.swing.JPanel {

DefaultTableModel dtm;

Image btn;

Connection conn;

Statement st;

String phone;

ImageIcon img;

File f;

FileInputStream fis;

Socket soc;

Main main;

String name;

String query=null;

public MakeGroup(Main main, String phone, Socket soc, String name) {

this.main = main;

this.phone = phone;

this.soc = soc;

this.name = name;

this.query ="Select r.firstname,r.lastname,r.image,r.phone from frdlist" + this.phone + " f,register r where f.phone=r.phone";

conn = DatabaseConnection.getdatabaseConnection();

dtm = new DefaultTableModel() {

@Override

public boolean isCellEditable(int row, int column) {

switch (column) {

case 0:

return false;

case 1:

return false;

case 2:

return false;

case 3:

return true;

default:

return false;

}

}

@Override

public Class getColumnClass(int column) {

return getValueAt(0, column).getClass();

}

};

initComponents();

jTable1.setRowHeight(60);

try {

btn = ImageIO.read(new File("src/plus.png"));

jButton1.setIcon(new ImageIcon(btn.getScaledInstance(22, 22, Image.SCALE\_SMOOTH)));

btn = ImageIO.read(new File("src/default-group-icon.png"));

jButton3.setIcon(new ImageIcon(btn.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

btn = ImageIO.read(new File("src/minus.png"));

jButton4.setIcon(new ImageIcon(btn.getScaledInstance(22, 22, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println(ex.toString());

}

String header[] = new String[]{"Image", "Name", "phone", "Selection"};

dtm.setColumnIdentifiers(header);

jTable1 = new JTable(dtm) {

@Override

public Class getColumnClass(int column) {

switch (column) {

case 0:

return getValueAt(0, column).getClass();

case 1:

return String.class;

case 2:

return String.class;

case 3:

return Boolean.class;

default:

return null;

}

}

};

jTable1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jTable1.getColumnModel().getColumn(0).setPreferredWidth(180);

try {

displayUsers(query);

} catch (SQLException ex) {

System.out.println(ex.toString());

}

JPanel top = new JPanel();

top.setBackground(new Color(159, 182, 205));

top.add(jLabel1);

top.add(jTextField1);

top.add(jLabel2);

top.add(jButton1);

JPanel bottom = new JPanel();

bottom.add(jPanel1);

jPanel1.add(jLabel4);

jPanel1.add(jTable1);

jPanel1.setVisible(false);

}

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jLabel4 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

jLabel5 = new javax.swing.JLabel();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

setPreferredSize(new java.awt.Dimension(539, 490));

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel1.setForeground(new java.awt.Color(51, 51, 51));

jLabel1.setText("Enter Group Name:");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel2.setForeground(new java.awt.Color(51, 51, 51));

jLabel2.setText("Add Participants:");

jButton1.setToolTipText("");

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

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

jButton1ActionPerformed(evt);

}

});

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0), 2, true));

jTable1.setModel(dtm);

jTable1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jTable1.setRowHeight(150);

jTable1.setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_SELECTION);

jTable1.getTableHeader().setResizingAllowed(false);

jTable1.getTableHeader().setReorderingAllowed(false);

jScrollPane1.setViewportView(jTable1);

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jLabel4.setText("select minimum 2 and maximum 10");

jButton2.setText("Create");

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

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

jButton2ActionPerformed(evt);

}

});

jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

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

jLabel5.setText("Please Provide optional group icon");

jButton3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/default-group-icon.png"))); // NOI18N

jButton3.setPreferredSize(new java.awt.Dimension(22, 22));

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

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

jButton3ActionPerformed(evt);

}

});

jButton4.addActionListener(new java.awt.event.ActionListener() {

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Refersh list");

jButton5.addActionListener(new java.awt.event.ActionListener() {

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

jButton5ActionPerformed(evt);

}

});

}

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

jPanel1.setVisible(true);

}

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

PrintWriter nos = null;

int rows = jTable1.getRowCount();

int selected = 0;

boolean tableExist = true;

for (int i = 0; i < rows; i++) {

if (jTable1.getValueAt(i, 3).equals(true)) {

selected++;

}

}

try {

//printwriter object creation to send data to server

nos = new PrintWriter(

new BufferedWriter(

new OutputStreamWriter(

soc.getOutputStream()

)

), true

);

} catch (IOException ex) {

Logger.getLogger(MakeGroup.class.getName()).log(Level.SEVERE, null, ex);

}

try {

String query = "select \* from " + jTextField1.getText();

st = conn.createStatement();

tableExist = st.execute(query);

System.out.println("Table Exist:" + tableExist);

} catch (SQLException ex) {

tableExist = false;

}

if (selected < 2) {

JOptionPane.showMessageDialog(null, "Minimum 2 memeber are required to add in group");

return;

} else if (selected > 11) {

JOptionPane.showMessageDialog(null, "Maximum 10 memeber can be added in group");

return;

} else if (jTextField1.getText().equals("")) {

JOptionPane.showMessageDialog(null, "Enter Group Name");

return;

} else if (jTextField1.getText().contains(" ")) {

JOptionPane.showMessageDialog(null, "Enter Group name without space");

return;

} else if (tableExist) {

JOptionPane.showMessageDialog(null, "Enter Different Group Name as same Group Name Exist ");

} else {

try {

st = conn.createStatement();

String GrpName = "create table " + jTextField1.getText() + " (phone varchar(10))";

st.execute(GrpName);

String InsertQuery;

System.out.println("Count:" + selected);

for (int i = 0; i < jTable1.getRowCount(); i++) {

if (jTable1.getValueAt(i, 3).equals(true)) {

InsertQuery = "insert into " + jTextField1.getText() + " values('" + jTable1.getValueAt(i, 2).toString() + "')";

st.executeUpdate(InsertQuery);

InsertQuery = "insert into grplist" + jTable1.getValueAt(i, 2).toString() + " values('" + jTextField1.getText() + "')";

st.executeUpdate(InsertQuery);

}

}

InsertQuery = "insert into " + jTextField1.getText() + " values('" + phone + "')";

st.executeUpdate(InsertQuery);

InsertQuery = "insert into grplist" + phone + " values('" + jTextField1.getText() + "')";

st.executeUpdate(InsertQuery);

InsertQuery = "insert into groupinfo (name) values('" + jTextField1.getText() + "')";

st.executeUpdate(InsertQuery);

fis = new FileInputStream(new File("src/grpicon.png"));

PreparedStatement st = conn.prepareStatement("Update groupinfo set image=? where name='" + jTextField1.getText() + "'");

st.setBinaryStream(1, (InputStream) fis, fis.available());// possition,stream bits,length

if (st.executeUpdate() > 0) {

System.out.println("image inserted into the database..");

} else {

System.out.println("image not inserted");

}

} catch (SQLException ex) {

System.out.println("Make group sql exception" + ex.toString());

} catch (FileNotFoundException ex) {

Logger.getLogger(MakeGroup.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(MakeGroup.class.getName()).log(Level.SEVERE, null, ex);

}

JOptionPane.showMessageDialog(null, "Group Created");

}

try {

String getPort = "select portnumber from groupinfo where name='" + jTextField1.getText() + "'";

ResultSet rs = st.executeQuery(getPort);

rs.next();

int port = rs.getInt("portnumber");

nos.println(this.name + ":group:" + port);

} catch (SQLException ex) {

Logger.getLogger(MakeGroup.class.getName()).log(Level.SEVERE, null, ex);

}

this.setEnabled(false);

main.setEnabled(true);

}

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

JFileChooser file = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter("\*.images", "jpg", "gif");

file.addChoosableFileFilter(filter);

int result = file.showOpenDialog(null);

if (result == JFileChooser.APPROVE\_OPTION) {

try {

f = file.getSelectedFile();

fis = new FileInputStream(f);

System.out.println(f);

Image bi = ImageIO.read(f);

jButton3.setIcon(new ImageIcon(bi.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

}

}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

jPanel1.setVisible(false);

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

try {

displayUsers(query);

} catch (SQLException ex) {

Logger.getLogger(MakeGroup.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void displayUsers(String query) throws SQLException {

dtm.setRowCount(0);

st = conn.createStatement();

ResultSet rs = st.executeQuery(query);

while (rs.next()) {

String username = rs.getString("firstname") + " " + rs.getString("lastname");

System.out.println(username);

Blob imgData = rs.getBlob("image");

img = FindFriend.getImage(imgData);

String phone = rs.getString("phone");

dtm.addRow(new Object[]{img, username, phone, false});

}

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

#### **RoomChat\_StartPage.java**

package chatapp;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.io.File;

import java.io.IOException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

public class RoomChat\_StartPage extends javax.swing.JPanel {

RoomChat roomChat;

JButton btn;

Image img;

public RoomChat\_StartPage(final String name, final Main main) {

initComponents();

this.btn=jButton1;

try {

img = ImageIO.read(new File("src/room-chat-frame-icon.jpeg"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(550,375,Image.SCALE\_SMOOTH)));

img = ImageIO.read(new File("src/enter-room-chat-button.png"));

jButton1.setIcon(new ImageIcon(img.getScaledInstance(55,25,Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println("roomchat\_startPage"+ex.getMessage());

}

jButton1.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

try {

roomChat = new RoomChat(name,main);

roomChat.setVisible(true);

roomChat.showJoiningMsg(name+":joined the room");

btn.setEnabled(false);

} catch (IOException ex) {

System.out.println("IO exception"+ex.getMessage());

}

}

});

}

private void initComponents() {

jButton1 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Room Chat", javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.DEFAULT\_POSITION));

jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/enter-room-chat-button.png"))); // NOI18N

jButton1.setText("Enter ");

jButton1.setActionCommand("Enter ");

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/room-chat-frame-icon.jpeg"))); // NOI18N

jLabel2.setText("into Room Chat");

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

// End of variables declaration

}

#### **RoomChat.java**

package chatapp;

import java.awt.Color;

import java.awt.Component;

import java.awt.ComponentOrientation;

import java.awt.Dimension;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.net.Socket;

import javax.imageio.ImageIO;

import javax.swing.DefaultListCellRenderer;

import javax.swing.DefaultListModel;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JList;

import javax.swing.JOptionPane;

import javax.swing.JScrollPane;

public class RoomChat extends javax.swing.JFrame {

DefaultListModel dlm;

Image img;

String name;

Socket soc;

Main main;

//reader thread

Reader reader;

JFrame f;//this is for giving reference of this frame to this object so that we can dispose

//this frame when user clicks on logout

public RoomChat(final String name, final Main main) throws IOException {

this.name = name;

this.f = this;

this.main = main;

dlm = new DefaultListModel();

initComponents();

//following code is to display the frame in the center of the screen

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

this.setLocation(dim.width / 2 - this.getSize().width / 2, dim.height / 2 - this.getSize().height / 2);

//setting the frame nonresizable

this.setResizable(false);

//setting the frame icon

img = ImageIO.read(new File("src/room-chat-frame-icon.jpeg"));

this.setIconImage(img);

jList1.setCellRenderer(new ListCellRendererDemo(this.name));

try {

img = ImageIO.read(new File("src/send.png"));

jButton1.setIcon(new ImageIcon(img.getScaledInstance(50, 30, Image.SCALE\_SMOOTH)));

img = ImageIO.read(new File("src/logout-room-chat.png"));

jButton2.setIcon(new ImageIcon(img.getScaledInstance(27, 25, Image.SCALE\_SMOOTH)));

soc = new Socket(Main.ip, 8096);

if (!soc.isConnected()) {

JOptionPane.showMessageDialog(null, "NOT ABLE TO CONNECT..");

} else {

reader = new Reader(soc, dlm, jList1, jScrollPane1);

reader.start();

System.out.println("reader started");

}

System.out.println("welcome " + name + " to chat room");

} catch (IOException ex) {

//if server is not connected, display this message to user.

JOptionPane.showMessageDialog(null, "Server is not connected");

}

sendMessageEvents();

}

public void sendMessageEvents() throws IOException {

final PrintWriter nos = new PrintWriter(

new BufferedWriter(

new OutputStreamWriter(

soc.getOutputStream()

)

), true

);

jButton1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

sendMessage(nos);

}

});

jTextField1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

sendMessage(nos);

}

});

//below code is to send logout request string to server

// so that server can disconnect the connection

jButton2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int isYES = JOptionPane.showConfirmDialog(null, "wanna sure exit from chat room?",

"Choose", JOptionPane.YES\_NO\_OPTION);

if (isYES == JOptionPane.YES\_OPTION) {

nos.println(name + ":logout");

nos.flush();

f.dispose();

main.startPage.btn.setEnabled(true);

}

}

});

}

public void sendMessage(PrintWriter nos) {

String msg = jTextField1.getText();

jTextField1.setText("");

if (!msg.equals("")) {

if (!msg.contains(":")) {

nos.println(name + ":" + msg);

nos.flush();

} else {

JOptionPane.showMessageDialog(null, "can not use \":\" in message");

}

}

}

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jList1 = new javax.swing.JList();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);

setTitle("RoomChat");

jLabel1.setFont(new java.awt.Font("Tekton Pro", 0, 18)); // NOI18N

jLabel1.setText(" Room Chat");

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JList jList1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

void showJoiningMsg(String msg) {

try {

final PrintWriter nos = new PrintWriter(

new BufferedWriter(

new OutputStreamWriter(

soc.getOutputStream()

)

), true

);

nos.println(msg);

} catch (IOException ex) {

}

}

}

class Reader extends Thread {

BufferedReader nis;

Socket soc;

DefaultListModel listModel;

JList list;

JScrollPane jsp;

Reader(Socket soc, DefaultListModel listModel, JList list, JScrollPane jsp) {

this.soc = soc;

this.listModel = listModel;

this.list = list;

this.jsp = jsp;

}

public void run() {

try {

nis = new BufferedReader(

new InputStreamReader(

soc.getInputStream()

)

);

String str = nis.readLine();

while (!str.equals("End")) {

System.out.println("server send back = " + str);

listModel.addElement(str);

int lastIndex = list.getModel().getSize() - 1;

if (lastIndex >= 0) {

list.ensureIndexIsVisible(lastIndex);

}

str = nis.readLine();

}

} catch (Exception e) {

}

}

}

class ListCellRendererDemo extends DefaultListCellRenderer {

String username;

public ListCellRendererDemo(String name) {

this.username = name;

}

@Override

public Component getListCellRendererComponent(JList list, Object value, int index,

boolean isSelected, boolean cellHasFocus) {

Component c = super.getListCellRendererComponent(list, value, index, isSelected, cellHasFocus);

if (value instanceof String) {

String msg = (String) value;

if (msg.contains("left this room") || msg.contains("joined the room")) {

setBackground(Color.GRAY);

} else {

setBackground(Color.WHITE);

}

if (msg.contains(username)) {

list.setComponentOrientation(ComponentOrientation.RIGHT\_TO\_LEFT);

} else {

list.setComponentOrientation(ComponentOrientation.LEFT\_TO\_RIGHT);

}

if (isSelected) {

setBackground(getBackground().darker());

}

}

return c;

}

}

#### **GroupChatWindow.java**

package chatapp;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.net.Socket;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.DefaultListModel;

import javax.swing.ImageIcon;

import javax.swing.JList;

import javax.swing.JOptionPane;

import javax.swing.JScrollPane;

public class GroupChatWindow extends javax.swing.JPanel {

Image img; // this image button is used for displaying image with different scaled instance

DefaultListModel dlm;

int port;

Connection conn;

Statement st = null;

ResultSet rs = null;

Main main;

String currentUserPhone;

Socket soc;

GrpReader reader;

static Backup backup;

static String grpName;

GroupMember grpMember;

public GroupChatWindow(int port, Connection conn, final Main main, String currentUserPhone, Socket soc, Backup backup, String grpName) {

this.port = port;

this.conn = conn;

this.main = main;

this.soc = soc;

this.currentUserPhone = currentUserPhone;

this.backup = backup;

this.grpName = grpName;

try {

dlm = new DefaultListModel();

final String currentUserName = getUserName(this.currentUserPhone);

initComponents();

Socket soc1 = new Socket(Main.ip, this.port);

if (!soc.isConnected()) {

JOptionPane.showMessageDialog(null, "NOT ABLE TO CONNECT..");

} else {

reader = new GrpReader(soc1, dlm, jList1, jScrollPane1);

reader.start();

System.out.println("reader started");

}

jList1.setCellRenderer(new ListCellRendererDemo(currentUserName));

//displaying image in send button

img = ImageIO.read(new File("src/back-button1.png"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

//displaying image in back label

img = ImageIO.read(new File("src/send.png"));

jButton1.setIcon(new ImageIcon(img.getScaledInstance(50, 25, Image.SCALE\_SMOOTH)));

displayUserData();

grpMember = new GroupMember(conn, grpName, main);

final PrintWriter nos = new PrintWriter(

new BufferedWriter(

new OutputStreamWriter(

soc1.getOutputStream()

)

), true

);

jButton1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String msg = jTextField1.getText();

jTextField1.setText("");

if (!msg.equals("")) {

if (!msg.contains(":")) {

nos.println(currentUserName + ":" + msg);

nos.flush();

}

else

JOptionPane.showMessageDialog(null,"can not use \":\" in message.");

}

}

});

jTextField1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String msg = jTextField1.getText();

jTextField1.setText("");

if (!msg.equals("")) {

if (!msg.contains(":")) {

System.out.println("inside if");

nos.println(currentUserName + ":" + msg);

nos.flush();

}

else

JOptionPane.showMessageDialog(null,"can not use \":\" in message.");

}

}

});

jLabel1.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

super.mouseClicked(e); //To change body of generated methods, choose Tools | Templates.

nos.println(currentUserName + ":$back$");

nos.flush();

main.splitPaneh.setRightComponent(main.myGroup);

}

});

} catch (IOException ex) {

} catch (SQLException ex) {

} catch (NullPointerException e) {

//backup.messagebackup.put(this.grpName,new MessageStack());

}

}

private void initComponents() {

jScrollPane2 = new javax.swing.JScrollPane();

jList2 = new javax.swing.JList();

jLabel1 = new javax.swing.JLabel();

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jList1 = new javax.swing.JList();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jList2.setModel(new javax.swing.AbstractListModel() {

String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };

public int getSize() { return strings.length; }

public Object getElementAt(int i) { return strings[i]; }

});

jScrollPane2.setViewportView(jList2);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/back-button1.png"))); // NOI18N

jLabel1.setText("back image");

jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel1MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

jLabel1MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

jLabel1MouseExited(evt);

}

});

jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.gray, java.awt.Color.darkGray));

jLabel2.setText("user pic");

jLabel3.setText("username");

jButton2.setText("GROUP info");

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

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

jButton2ActionPerformed(evt);

}

});

jList1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Group Chat", javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.DEFAULT\_POSITION));

jList1.setModel(dlm);

jList1.setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_SELECTION);

jList1.setToolTipText("Chat Messages");

jScrollPane1.setViewportView(jList1);

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

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

jButton1ActionPerformed(evt);

}

});

}

private void jLabel1MouseExited(java.awt.event.MouseEvent evt) {

try {

//this is to change the back label to different image

img = ImageIO.read(new File("src/back-button1.png"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println(ex.getMessage() + " inside jlabel1 mouse event chatwindow");

}

}

private void jLabel1MouseEntered(java.awt.event.MouseEvent evt) {

try {

//this is to change the back label to different image

img = ImageIO.read(new File("src/back-button2.png"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(25, 25, Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println(ex.getMessage() + " inside jlabel1 mouse event chatwindow");

}

}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

//main.splitPaneh.setRightComponent(new MyGroup(main,conn,currentUserPhone));

main.splitPaneh.setRightComponent(main.myGroup);

}

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

grpMember.setVisible(true);

main.setEnabled(false);

}

private void displayUserData() throws SQLException {

String q = "select name,image from groupinfo where portnumber=" + this.port;

st = conn.createStatement();

ResultSet rs = st.executeQuery(q);

while (rs.next()) {

String name = rs.getString("name");

Blob imgData = rs.getBlob("image");

ImageIcon imgicon = FindFriend.getImage(imgData);

jLabel2.setIcon(imgicon);

jLabel3.setText(name);

}

}

private String getUserName(String currentUserPhone) {

String name = null;

try {

System.out.println(currentUserPhone);

String query = "select firstname from register where phone='" + currentUserPhone + "'";

ResultSet rs = null;

st = conn.createStatement();

rs = st.executeQuery(query);

rs.next();

name = rs.getString("firstname");

} catch (SQLException ex) {

Logger.getLogger(GroupChatWindow.class.getName()).log(Level.SEVERE, null, ex);

System.out.println(ex.getMessage());

} catch (NullPointerException ex) {

Logger.getLogger(GroupChatWindow.class.getName()).log(Level.SEVERE, null, ex);

System.out.println(ex.getMessage());

}

return name;

}

private void displayMessage() {

for (int i = 0; i < backup.messagebackup.get(this.grpName).message.size(); i++) {

dlm.addElement(backup.messagebackup.get(this.grpName).message.get(i));

}

}

// 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.JList jList1;

private javax.swing.JList jList2;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

class GrpReader extends Thread {

BufferedReader nis;

Socket soc;

DefaultListModel listModel;

JList list;

JScrollPane jsp;

GrpReader(Socket soc, DefaultListModel listModel, JList list, JScrollPane jsp) {

this.soc = soc;

this.listModel = listModel;

this.list = list;

this.jsp = jsp;

}

public void run() {

try {

nis = new BufferedReader(

new InputStreamReader(

soc.getInputStream()

)

);

String str = nis.readLine();

while (!str.equals("End")) {

System.out.println("server send back = " + str);

listModel.addElement(str);

//GroupChatWindow.backup.messagebackup.get(GroupChatWindow.grpName).message.add(str);

int lastIndex = list.getModel().getSize() - 1;

if (lastIndex >= 0) {

list.ensureIndexIsVisible(lastIndex);

}

//jsp.revalidate();

//jsp.repaint();

str = nis.readLine();

}

} catch (Exception e) {

}

}

}

#### **ChatWindow.java(Personal chat)**

package chatapp;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.IOException;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.net.Socket;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.DefaultListModel;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class ChatWindow extends javax.swing.JPanel {

Image img; // this image button is used for displaying image with different scaled instance

DefaultListModel dlm;

String phone="";

Connection conn;

Statement st=null;

ResultSet rs=null;

Main main;

String currentUserPhone;

static Backup backup;

PersonalReader reader;

PrintWriter nos;

int priority=0;

public ChatWindow(String phone, Connection conn,Main main,String currentUserPhone,Backup backup) {

this.phone=phone;

this.conn=conn;

this.main=main;

this.currentUserPhone=currentUserPhone;

this.backup=backup;

try {

st=conn.createStatement();

dlm=new DefaultListModel();

initComponents();

//jList1.setCellRenderer(new ListCellRendererDemo("You:"));

//displaying image in send button

img = ImageIO.read(new File("src/back-button1.png"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

//displaying image in back label

img = ImageIO.read(new File("src/send.png"));

jButton1.setIcon(new ImageIcon(img.getScaledInstance(50,25,Image.SCALE\_SMOOTH)));

jLabel4.setText("");

img = ImageIO.read(new File("src/star2.png"));

jLabel4.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

displayUserData();

try{

this.backup.messagebackup.get(this.phone).message.isEmpty();

displayMessage();

}catch(NullPointerException ex){

this.backup.messagebackup.put(phone,new MessageStack());

}

RefreshMessage refreshMessage = new RefreshMessage();

refreshMessage.start();

} catch (IOException ex) {

} catch (SQLException ex) {

}

}

private void initComponents() {

jScrollPane2 = new javax.swing.JScrollPane();

jList2 = new javax.swing.JList();

jLabel1 = new javax.swing.JLabel();

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jList1 = new javax.swing.JList();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jList2.setModel(new javax.swing.AbstractListModel() {

String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };

public int getSize() { return strings.length; }

public Object getElementAt(int i) { return strings[i]; }

});

jScrollPane2.setViewportView(jList2);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/back-button1.png"))); // NOI18N

jLabel1.setText("back image");

jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel1MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

jLabel1MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

jLabel1MouseExited(evt);

}

});

jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGray, java.awt.Color.darkGray));

jLabel2.setText("user pic");

jLabel3.setText("username");

jLabel4.setText("jLabel4");

jLabel4.setToolTipText("Priority ");

jLabel4.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel4MouseClicked(evt);

}

});

jLabel4.getAccessibleContext().setAccessibleName("Priority");

jList1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Chat Area", javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.DEFAULT\_POSITION));

jList1.setModel(dlm);

jList1.setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_SELECTION);

jList1.setToolTipText("Chat Messages");

jScrollPane1.setViewportView(jList1);

jTextField1.addActionListener(new java.awt.event.ActionListener() {

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

jTextField1ActionPerformed(evt);

}

});

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

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

jButton1ActionPerformed(evt);

}

});

}

private void jLabel1MouseExited(java.awt.event.MouseEvent evt) {

try {

//this is to change the back label to different image

img = ImageIO.read(new File("src/back-button1.png"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

//this.setCursor(Cursor.getDefaultCursor());

} catch (IOException ex) {

System.out.println(ex.getMessage() + " inside jlabel1 mouse event chatwindow");

}

}

private void jLabel1MouseEntered(java.awt.event.MouseEvent evt) {

try {

//this is to change the back label to different image

img = ImageIO.read(new File("src/back-button2.png"));

jLabel1.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

//img = ImageIO.read(new File("src/hand-pointer.png"));

//this.setCursor(Toolkit.getDefaultToolkit().createCustomCursor(img,

// new Point(0,3), "cursor"));

} catch (IOException ex) {

System.out.println(ex.getMessage() + " inside jlabel1 mouse event chatwindow");

}

}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {

//going back to friend list window

this.removeAll();

main.splitPaneh.setRightComponent(new MyFriends(main,currentUserPhone,conn,backup));

}

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

displayData();

}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

displayData();

}

private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {

String str;

str="select priority from frdlist"+this.phone+" where phone="+this.currentUserPhone;

try {

ResultSet rs=st.executeQuery(str);

while(rs.next())

this.priority=rs.getInt("priority");

} catch (SQLException ex) {

Logger.getLogger(ChatWindow.class.getName()).log(Level.SEVERE, null, ex);

}

if(this.priority==0){

str="update frdlist"+this.phone+" set priority=1 where phone="+this.currentUserPhone;

try{

st=conn.createStatement();

st.execute(str);

this.priority=1;

img = ImageIO.read(new File("src/star2.png"));

jLabel4.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

} catch (SQLException ex) {System.out.println("Priority problem"+ex.toString());

} catch (IOException ex) {

System.out.println(ex.toString());

}

JOptionPane.showMessageDialog(null,"Priority Message has been send");

}

else{

int isYES=JOptionPane.showConfirmDialog(null,"wanna Cancel priority?",

"Choose",JOptionPane.YES\_NO\_OPTION);

if(isYES==JOptionPane.YES\_OPTION){

try {

str="update frdlist"+this.phone+" set priority=0 where phone="+this.currentUserPhone;

st=conn.createStatement();

st.execute(str);

this.priority=0;

img = ImageIO.read(new File("src/star1.png"));

jLabel4.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

} catch (SQLException ex) {

System.out.println(ex.toString());

} catch (IOException ex) {

System.out.println(ex.toString());

}

}

}

}

private void displayData(){

String msg=jTextField1.getText();

jTextField1.setText("");

if(!msg.equals("")){

System.out.println("Msg snd frm client:"+phone+currentUserPhone+msg);

String str="insert into chat(sender,receiver,message) values('"+currentUserPhone+"','"+phone+"','"+msg+"')";

try {

st.executeUpdate(str);

} catch (SQLException ex) {

System.out.println(ex.toString());

}

dlm.addElement(msg);

backup.messagebackup.get(phone).message.add(msg);

int lastIndex = jList1.getModel().getSize() - 1;

if (lastIndex >= 0) {

jList1.ensureIndexIsVisible(lastIndex);

}

jScrollPane1.repaint();

}

}

private void displayUserData() throws SQLException{

//dtm.setRowCount(0);

//String query="Select r.firstname,r.lastname,r.image,r.phone from register r,frdlist"+this.phone+" fl where r.phone=fl.phone";

String q="select firstname,lastname,image from register where phone='"+this.phone+"'";

st=conn.createStatement();

ResultSet rs=st.executeQuery(q);

while(rs.next()){

String name=rs.getString("firstname")+" "+rs.getString("lastname");

Blob imgData = rs.getBlob("image");

ImageIcon imgicon=FindFriend.getImage(imgData);

jLabel2.setIcon(imgicon);

jLabel3.setText(name);

// dtm.addRow(new Object[] { img,name,phone });

}

}

private void displayMessage(){

for(int i=0;i<backup.messagebackup.get(phone).message.size();i++){

dlm.addElement(backup.messagebackup.get(phone).message.get(i));

}

}

private void refresh(){

int sizeDlm=dlm.getSize();

int sizeBackup=backup.messagebackup.get(phone).message.size();

//System.out.println("Size of backup:"+sizeBackup+" sizeDlm:"+sizeDlm);

if(sizeBackup>sizeDlm){

for(int i=sizeDlm;i<sizeBackup;i++){

System.out.println("phone:"+backup.messagebackup.get(phone).message.get(i));

dlm.addElement(backup.messagebackup.get(phone).message.get(i));

int lastIndex = jList1.getModel().getSize() - 1;

if (lastIndex >= 0) {

jList1.ensureIndexIsVisible(lastIndex);

}

jScrollPane1.repaint();

}

}

}

public void refreshPriority(){

try {

String q="select priority from frdlist"+this.phone+" where phone="+this.currentUserPhone;

try{

ResultSet r=st.executeQuery(q);

r.next();

this.priority=r.getInt("priority");

}catch(NullPointerException e){

System.out.println(e.toString());

}

if(this.priority==1){

try {

//set priority image

img = ImageIO.read(new File("src/star2.png"));

jLabel4.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println("priority image "+ex.toString());

}

}

else{

try {

//set not priority image

img = ImageIO.read(new File("src/star1.png"));

jLabel4.setIcon(new ImageIcon(img.getScaledInstance(25,25,Image.SCALE\_SMOOTH)));

} catch (IOException ex) {

System.out.println("priority image "+ex.toString());

}

}

} catch (SQLException ex) {

Logger.getLogger(ChatWindow.class.getName()).log(Level.SEVERE, null, ex);

}

}

class RefreshMessage extends Thread{

@Override

public void run() {

super.run(); //To change body of generated methods, choose Tools | Templates.

while(true){

refresh();

refreshPriority();

}

}

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JList jList1;

private javax.swing.JList jList2;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

#### **DataAccess.java(Data Access Object)**

package chatapp;

import java.awt.Image;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.net.InetAddress;

import java.sql.Blob;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JRadioButton;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class DataAccess {

Connection conn;

Statement st;

ResultSet rs;

PreparedStatement pst;

Backup backup;

public DataAccess() {

backup = new Backup();

}

public void register(String firstname, String lastname, String email, String password, String phone,

String gender, Connection conn) {

this.conn = conn;

FileInputStream fis;

File f = new File("src/usericon.png");

try {

fis = new FileInputStream(f);

String str = "INSERT INTO register(firstname,lastname,email,password,phone,online,gender,image) values(?,?,?,?,?,?,?,?)";

pst = conn.prepareStatement(str);

ResultSet rs = pst.getResultSet();

pst.setString(1, firstname);

pst.setString(2, lastname);

pst.setString(3, email);

pst.setString(4, password);

pst.setString(5, phone);

pst.setInt(6, 0);

pst.setString(7, gender);

pst.setBinaryStream(8, (InputStream) fis, fis.available());// possition,stream bits,length

pst.execute();

JOptionPane.showMessageDialog(null, "Records inserted successfully..");

} catch (FileNotFoundException ex) {

System.out.println(ex.toString());

} catch (SQLException ex) {

System.out.println(ex.toString());

} catch (IOException ex) {

System.out.println(ex.toString());

}

}

public boolean isLoginDataValid(String phone, String password, Connection conn) throws SQLException {

String str = "select password from register where phone='" + phone + "'";

st = conn.createStatement();

rs = st.executeQuery(str);

rs.next();

String passwd2 = rs.getString("password");

String update = "UPDATE register SET online=1 where phone='" + phone + "'";

rs.close();

if (password.equals(passwd2)) {

st.executeUpdate(update);

return true;

} else {

return false;

}

}

public String getName(String phone, Connection conn) throws SQLException {

String str = "select firstname from register where phone='" + phone + "'";

st = conn.createStatement();

rs = st.executeQuery(str);

rs.next();

String name = rs.getString("firstname");

rs.close();

return name;

}

public void logout(String phone, Connection conn) throws SQLException {

String update = "UPDATE register SET online=0 where phone='" + phone + "'";

st = conn.createStatement();

st.executeUpdate(update);

backup.store();

LoginPage login = new LoginPage();

login.setVisible(true);

}

public void displayData(DefaultTableModel dtm, String phone, Connection conn, JLabel imgLabel) {

dtm.setRowCount(0);

String str = "select \* from register where phone='" + phone + "'";

try {

st = conn.createStatement();

rs = st.executeQuery(str);

rs.next();

String firstName = rs.getString("firstname");

String lastName = rs.getString("lastname");

String email = rs.getString("email");

String gender = rs.getString("gender");

String phone2 = rs.getString("phone");

Blob imgData = rs.getBlob("image");

if (imgData != null) {

try {

File tmpFile = new File("tmpImage");

FileOutputStream fos = new FileOutputStream(tmpFile);

fos.write(imgData.getBytes(1L, (int) imgData.length()));

fos.close();

Image image = ImageIO.read(tmpFile);

imgLabel.setIcon(new ImageIcon(image.getScaledInstance(208, 168, Image.SCALE\_SMOOTH)));

} catch (FileNotFoundException ex) {

JOptionPane.showMessageDialog(null, "Image not found..", "Load Error", JOptionPane.ERROR\_MESSAGE);

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, "Failed To Load Image Data", "Load Error", JOptionPane.ERROR\_MESSAGE);

}

}

String[] title = new String[]{"UserName:", "Email ID:", "Gender:", "Phone Number:"};

String[] data = new String[]{firstName + " " + lastName, email, gender, phone};

for (int i = 0; i < 4; i++) {

dtm.addRow(new Object[]{title[i], data[i]});

}

} catch (SQLException ex) {

}

}

public int updateProfilePic(String phone, Connection conn, FileInputStream fis) throws SQLException, IOException {

PreparedStatement st = conn.prepareStatement("Update register set image=? where phone='" + phone + "'");

st.setBinaryStream(1, (InputStream) fis, fis.available());// possition,stream bits,length

int i = st.executeUpdate();

return i;

}

public void updateRecord(Connection conn, String phone,

JTextField t1, JTextField t2,

JTextField t3, JRadioButton b1, JRadioButton b2) throws SQLException {

String record = "select \* from register where phone='" + phone + "'";

st = conn.createStatement();

rs = st.executeQuery(record);

rs.next();

t1.setText(rs.getString("firstname"));

t2.setText(rs.getString("lastname"));

t3.setText(rs.getString("email"));

String gender = rs.getString("gender");

if (gender.equals("male")) {

b1.setSelected(true);

}

if (gender.equals("female")) {

b2.setSelected(true);

}

}

public void displayUser(Connection conn, DefaultTableModel dtm, String name, ImageIcon img, String query) throws SQLException {

//this method is used to display friend list/grplist/all user list

dtm.setRowCount(0);

st = conn.createStatement();

if (name != null) {

query = "Select \* from register where firstname='" + name + "'";

}

rs = st.executeQuery(query);

while (rs.next()) {

String name2 = rs.getString("firstname") + " " + rs.getString("lastname");

Blob imgData = rs.getBlob("image");

img = FindFriend.getImage(imgData);

String phone2 = rs.getString("phone");

dtm.addRow(new Object[]{img, name2, phone2});

}

}

public void requestSend(String query1, String query2, String query3, JTable table, String phone, Connection conn) {

String str = null;

try {

st = conn.createStatement();

rs = st.executeQuery(query1);

rs.next();

str = rs.getString("sender");

} catch (SQLException s) {

System.out.println("ERROR" + s.toString());

} catch (ArrayIndexOutOfBoundsException s) {

System.out.println("ERROR" + s.toString());

}

if (table.getSelectedRow() >= 0) {

if (str.equals(phone)) {

int isYES = JOptionPane.showConfirmDialog(null, "Request Already Send , wannna Cancel?", "Choose", JOptionPane.YES\_NO\_OPTION);

if (isYES == JOptionPane.YES\_OPTION) {

try {

st.execute(query2);

JOptionPane.showMessageDialog(null, "Request has been cancel");

} catch (SQLException s1) {

System.out.println(s1.toString());

}

}

} else {

try {

st.execute(query3);

} catch (SQLException ex) {

}

JOptionPane.showMessageDialog(null, "request has sent...");

}

} else {

JOptionPane.showMessageDialog(null, "Please Select the person you want to send request to!");

}

}

public int getGrpPort(Connection conn,String query) throws SQLException {

st = conn.createStatement();

rs = st.executeQuery(query);

rs.next();

return rs.getInt("portnumber");

}

}

#### **DatabaseConnection.java**

package chatapp;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

static Connection conn;

public static Connection getdatabaseConnection(){

try {

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

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

System.out.println("connection established to 192.168.108.108");

} catch (ClassNotFoundException ex) {

ex.printStackTrace();

System.out.println("database connection error");

} catch (SQLException ex) {

System.out.println("database connection error");

}

return conn;

}

}

#### **PersonalReader.java**

package chatapp;

import java.net.Socket;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

public class PersonalReader extends Thread {

Socket soc;

String senderPhone;

Backup backup;

Connection conn;

Statement st;

String phone;

PersonalReader(Connection conn, Backup backup, String phone) {

try {

this.conn = conn;

st = conn.createStatement();

this.soc = soc;

this.backup = backup;

this.phone = phone;

} catch (SQLException ex) {

Logger.getLogger(PersonalReader.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void run() {

String str = "select \* from chat";

String strdelete = "";

while (true) {

try {

ResultSet rs = st.executeQuery(str);

while (rs.next()) {

if (rs.getString("receiver").equals(phone)) {

senderPhone = rs.getString("sender");

String message = rs.getString("message");

strdelete = "delete from chat where id=" + rs.getInt("id") ;

st.executeUpdate(strdelete);

try {

this.backup.messagebackup.get(senderPhone).message.add(message);

System.out.println("Personal Reader::senderPhone:" + senderPhone + " msg:" + message);

} catch (NullPointerException ex) {

this.backup.messagebackup.put(senderPhone, new MessageStack());

this.backup.messagebackup.get(senderPhone).message.add(message);

}

}

}

} catch (SQLException ex) {

System.out.println(ex.toString());

}

}

}

}

#### **PriorityListener.java**

package chatapp;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class PriorityListener extends Thread {

Connection conn;

Statement st;

String phone;

ArrayList<String> name;

public PriorityListener(Connection conn,String phone) throws SQLException{

this.conn=conn;

st=conn.createStatement();

this.phone=phone;

name=new ArrayList<String>();

}

@Override

public void run() {

while(true){

try {

String str="select f.phone,f.priority,r.firstname from frdlist"+this.phone+" f,register r where f.phone=r.phone";

ResultSet rs=st.executeQuery(str);

while(rs.next()){

if(rs.getInt("priority")==1){

name.add(rs.getString("firstname"));

}

}

} catch (SQLException ex) {

System.out.println("PriorityListener: "+ex.toString());

}

String names="";

if(!name.isEmpty()){

for(String s:name){

names+=s+" ";

}

JOptionPane.showMessageDialog(null,"Priority Message has been send by "+names);

String str="update frdlist"+this.phone+" set priority=0";

try {

st.execute(str);

} catch (SQLException ex) {

System.out.println(ex.toString());

}

name.clear();

}

}

}

}

#### **PersonalUser.java (server side Thread)**

package chatserver;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.ArrayList;

import java.util.Hashtable;

import java.util.logging.Level;

import java.util.logging.Logger;

public class PersonalUser {

private final static int PORT=9000;// server will listen to this port number for each user

public static void main(String[] args) {

try {

ServerSocket ss=new ServerSocket(PORT);

while(true){

Socket soc=ss.accept();

BufferedReader nis = new BufferedReader(

new InputStreamReader(

soc.getInputStream()

)

);

PrintWriter nos = new PrintWriter(

new BufferedWriter(

new OutputStreamWriter(

soc.getOutputStream()

)

), true

);

UserConversation c=new UserConversation(soc,nos,nis);

c.start();

}

} catch (IOException ex) {

}

}

}

class UserConversation extends Thread{

Socket soc;

PrintWriter nos;

BufferedReader nis;

public UserConversation(Socket soc,PrintWriter nos,BufferedReader nis){

this.soc=soc;

this.nos=nos;

this.nis=nis;

}

@Override

public void run() {

try {

super.run(); //To change body of generated methods, choose Tools | Templates.

String str = nis.readLine();

String[] retval=str.split(":");

//user will send information in 3 parts

//1.name 2.message 3.port no for grp

// massage can either be actual message or "logout" or "group"

System.out.println(retval[0]);

System.out.println(retval[1]);

System.out.println(retval[2]);

while (!retval[1].equalsIgnoreCase("logout")) {

if(retval[1].equalsIgnoreCase("group")){

//if user send msg as "group" that means user is creating grp

//we will create group object that will handle the grp msg

createGroupOperation(Integer.parseInt(retval[2]));

}else{

//operation to send msg to intended user

}

str = nis.readLine();

retval=str.split(":");

}

soc.close();

System.out.println("socket closed | user logged out");

} catch (IOException ex) {

Logger.getLogger(UserConversation.class.getName()).log(Level.SEVERE, null, ex);

}}

private void createGroupOperation(int port) {

Group group=new Group(port);

group.createConnection();

}

}

class Group{

int port;

public static Message<String> q;

public static Message<String> allMessage;

public static ArrayList<PrintWriter> noslist;

public Group(int port){

this.port=port;

this.q=new Message<>();

allMessage = new Message<>();

noslist = new ArrayList<>();

}

public void createConnection(){

try {

ServerSocket ss= new ServerSocket(port);

MessageDispacthergrp md = new MessageDispacthergrp();

md.setDaemon(true);

md.start();

while(true){

Socket soc = ss.accept();

System.out.println("Connection established");

BufferedReader nis = new BufferedReader(

new InputStreamReader(

soc.getInputStream()

)

);

PrintWriter nos = new PrintWriter(

new BufferedWriter(

new OutputStreamWriter(

soc.getOutputStream()

)

), true

);

Conversationgrp c = new Conversationgrp(soc,nis,nos);

c.start();

}

} catch (IOException ex) {

Logger.getLogger(Group.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

class Conversationgrp extends Thread {

BufferedReader nis;

PrintWriter nos;

Socket soc;

Conversationgrp(Socket soc,BufferedReader nis,PrintWriter nos) {

this.soc = soc;

this.nos=nos;

this.nis=nis;

}

@Override

public void run() {

try {

Group.noslist.add(nos);

//if user enter into group chat following is the code

// to send 500 previous messages sent by others

//that messages are stored in message generic class

if(Group.allMessage.getSize() > 0){

ArrayList<String> al=Group.allMessage.getAll();

for(String msg : al){

nos.println(msg);

}

}

String str = nis.readLine();

String[] retval=str.split(":");

while (!retval[1].equalsIgnoreCase("exit-group")) {

if(retval[1].equalsIgnoreCase("$back$")){

soc.close();

Group.noslist.remove(nos);

this.destroy();

}

else{

Group.q.enqueue(str);

Group.allMessage.insert(str);

str = nis.readLine();

retval=str.split(":");

}

}

Group.q.enqueue(retval[0] + " left this group");

Group.allMessage.insert(retval[0] + " left this group");

Group.noslist.remove(nos);

soc.close();

System.out.println("socket closed");

} catch (Exception e) {

}

}

}

class MessageDispacthergrp extends Thread {

@Override

public void run() {

while (true) {

try {

String str = Group.q.dequeue();

for (PrintWriter o : Group.noslist) {

o.println(str);

}

} catch (Exception e) {

}

}

}

}

class Message<T>{

ArrayList<T> al=new ArrayList<>();

ArrayList<T> messageBuffere = new ArrayList<>();

synchronized public void enqueue(T i){

al.add(i);

notify();

}

synchronized public T dequeue(){

if(al.isEmpty()){

try{

wait();

}catch(Exception e){}

}

return al.remove(0);

}

synchronized public void insert(T i){

if(messageBuffere.size() == 500){

messageBuffere.remove(0);

}

messageBuffere.add(i);

}

synchronized public int getSize(){

return messageBuffere.size();

}

synchronized public ArrayList<T> getAll(){

if(!messageBuffere.isEmpty())

return messageBuffere;

else

return null;

}

}

|  |
| --- |
| **Iterations/Spirals** |

This part consists of the increments produced over the 4 iterations to develop the product. 4 iterations were required as **new requirements and changes** were required to be added, and also because the product did not quite pass the exit criteria in the first 3 iterations.

At each stage, **at least 3 types of tests were performed:**

1. **Unit tests** of the newly incorporated changes to confirm that they work as intended.All the components/module were tested to check for errors and defect.
2. **Integration Tests** to make sure that they are compatible with the rest of the system and no unintended side effects are produced.All the components were integerated into Main.java and tested for errors and defects.
3. **System test** defined in the Elaboration Phase to check if the product is ready.

# Transition

Tasks

* Final documentation
* Prepare for presentation

Objectives

* Wrap up and present project

Deliverables

* Black book
* Final product

|  |
| --- |
| Future enhancements and releases |

There is always scope for improvement in accuracy and performance :

* To Develop Android Version of KIKON CHAT.
* To Implent Cryptography for beter security.
* To Implement HIDE AND SEEK Feature in Group Chat.
* To Implement Games for User to play.

|  |
| --- |
| **Project Experience And Conclusion** |

# Learned how to manage time and work as a team

# The most important part was that I learned new Concept such as Socket Connection,Threading.This project turned out to be the first actual experience in IT industry.

# Finally got a chance to deal with documentation and making an actual project which I accomplished successfully.

# Learned to deal with Application building.

# The journey was not all that smooth with many many glitches faced while developing this project .

# 

# BIBLIOGRAPHY

# [www.stackoverflow.com](http://www.stackoverflow.com)

# [www.coderanch.com](http://www.coderanch.com)

# [www.tutorialrepublic.com](http://www.tutorialrepublic.com)

# [www.w3schools.com](http://www.w3schools.com)