```
// ********************** Blood Bank Management System ************************

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package blood_bank_management;

import java.awt.Color;
import java.awt.Container;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

/**
 *
 * @author 2019-1-60-256
 */

interface BloodBankInterface
{
    void search_blood(String c_blood_group);

    void search_blood(String c_address, String c_blood_group);

    void search_vlounteer_by_location(String c_address);
}

class userInterface implements BloodBankInterface
{
    private String name;
    private String email;
    private String phone;
    private String address;
```

```
public void set_value_user(String name, String email, String phone, String address)
{
    this.name = name;
    this.email = email;
    this.phone = phone;
    this.address = address;
}

public String get_name()
{
    return name;
}

public String get_email()
{
    return email;
}

public String get_phone()
{
    return phone;
}

public String get_address()
{
    return address;
}
```

```
@Override
public void search_blood(String c_blood_group)
{
    //Code for JTextArea for show only
    JFrame frame = new JFrame();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    Font font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

    JPanel panel = new JPanel();
    panel.setLayout(null);
    panel.setBackground(Color.red);

    JTextArea jta = new JTextArea();
    jta.setForeground(Color.white);
    jta.setBackground(Color.gray);
    jta.setFont(font1);
    jta.setLineWrap(true);
    jta.setWrapStyleWord(true);
    jta.setEditable(false);

    JScrollPane jsp = new JScrollPane(jta);
    jsp.setBounds(15, 20, 560, 380);

    panel.add(jsp);
    frame.add(panel);
    frame.setVisible(true);
    frame.setResizable(false);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Main code of search
    String filelist[] = new String[1000];
    String adminlist2[] = new String[1000];

    fileHandle ob = new fileHandle();
    try {
       ob.readObjectFromFile("AllFileLocation", filelist);
    } catch (IOException ex) {
       Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
    }

    int i,sum = 0;
    String str = "Yes";
    for( i=0; i<demo.n; i++ )
    {
       try {
          ob.readObjectFromFile(filelist[i], adminlist2);
       } catch (IOException ex) {
          Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
       }

       if ( str.equals(adminlist2[7]) )
       {
          String o_blood_group = adminlist2[6];
```

```
    if( c_blood_group.equals(o_blood_group) )
    {
      sum++;

      jta.append("Name : "+adminlist2[0]);

      jta.append("\nEmail : "+adminlist2[1]);

      jta.append("\nPhone Number : "+adminlist2[2]);

      jta.append("\nAddress : "+adminlist2[3]);

      jta.append("\nBlood Group : "+adminlist2[6]);

      jta.append("\nDonation Status : "+adminlist2[7]);

      jta.append("\n\n......................................\n\n");

    }
  }
}

if ( sum <= 0)
{
  jta.append("Sorry, No donor available at this moment. Check again after sometime.....");
}
else
```

```
    {
        jta.append("Total donors of this blood group : "+sum);
    }

}

@Override
public void search_blood(String c_address, String c_blood_group)
{
    //Code for JTextArea for show only
    JFrame frame = new JFrame();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    Font font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

    JPanel panel = new JPanel();
    panel.setLayout(null);
    panel.setBackground(Color.red);

    JTextArea jta = new JTextArea();
    jta.setForeground(Color.white);
    jta.setBackground(Color.gray);
    jta.setFont(font1);
    jta.setLineWrap(true);
    jta.setWrapStyleWord(true);
    jta.setEditable(false);
```

```
JScrollPane jsp = new JScrollPane(jta);
jsp.setBounds(15, 20, 560, 380);

panel.add(jsp);
frame.add(panel);

frame.setVisible(true);
frame.setResizable(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Main code of search
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
try {
   ob.readObjectFromFile("AllFileLocation", filelist);
} catch (IOException ex) {
   Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

int i,sum = 0;
String str = "Yes";
for( i=0; i<demo.n; i++ )
{
   try {
      ob.readObjectFromFile(filelist[i], adminlist2);
   } catch (IOException ex) {
    Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

if ( str.equals(adminlist2[7]) )
{
    String o_address = adminlist2[3];
    String o_blood_group = adminlist2[6];

    if( c_address.equals(o_address) && c_blood_group.equals(o_blood_group))
    {
        sum++;

        jta.append("Name : "+adminlist2[0]);

        jta.append("\nEmail : "+adminlist2[1]);

        jta.append("\nPhone Number : "+adminlist2[2]);

        jta.append("\nAddress : "+adminlist2[3]);

        jta.append("\nBlood Group : "+adminlist2[6]);

        jta.append("\nDonation Status : "+adminlist2[7]);

        jta.append("\n\n..................................\n\n");

    }
```

```
        }
    }

    if ( sum <= 0)
    {
        jta.append("Sorry, No donor available at this moment. Check again after sometime.....");
    }
    else
    {
        jta.append("Total donors of this blood group near the location : "+sum);
    }
}

@Override
public void search_vlounteer_by_location(String c_address)
{
    //Code for JTextArea for show only
    JFrame frame = new JFrame();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    Font font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

    JPanel panel = new JPanel();
    panel.setLayout(null);
    panel.setBackground(Color.red);

    JTextArea jta = new JTextArea();
```

```
jta.setForeground(Color.white);
jta.setBackground(Color.gray);
jta.setFont(font1);
jta.setLineWrap(true);
jta.setWrapStyleWord(true);
jta.setEditable(false);

JScrollPane jsp = new JScrollPane(jta);
jsp.setBounds(15, 20, 560, 380);

panel.add(jsp);
frame.add(panel);

frame.setVisible(true);
frame.setResizable(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Main code of search
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
try {
   ob.readObjectFromFile("AllFileLocation", filelist);
} catch (IOException ex) {
   Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}
```

```
int i,sum = 0;
String str = "No";
for ( i=0; i<demo.n; i++ )
{
   try {
      ob.readObjectFromFile(filelist[i], adminlist2);
   } catch (IOException ex) {
      Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
   }

   if ( str.equals(adminlist2[8]))
   {
      String o_address = adminlist2[3];

      if ( c_address.equals(o_address) )
      {
         sum++;

         jta.append("Name : "+adminlist2[0]);

         jta.append("\nEmail : "+adminlist2[1]);

         jta.append("\nPhone Number : "+adminlist2[2]);

         jta.append("\nAddress : "+adminlist2[3]);

         jta.append("\nBlood Group : "+adminlist2[6]);
```

```
                    jta.append("\nDonation Status : "+adminlist2[7]);


                    jta.append("\n\n.................................\n\n");
                }
            }
        }
        if ( sum <= 0)
        {
            jta.append("Sorry, No volunteer available at this moment. Check again after sometime.....");
        }
        else
        {
            jta.append("Total volunteers around this location in the list : "+sum);
        }
    }


}


class adminInterface extends userInterface
{
    generateRandom id_generate = new generateRandom();


    private String id;
    private String password;
    private String blood_group;
    private String donation_status;
    private String donor_visibility;
```

```
public void set_value_admin(String password, String blood_group, String donation_status)
{
    id = id_generate.generateID();
    this.password = password;
    this.blood_group = blood_group;
    this.donation_status = donation_status;
    donor_visibility = "";
}

public void set_value_admin_id(String id)
{
    this.id = id;
}

public void set_value_admin_password(String password)
{
    this.password = password;
}

public void set_value_admin_donation_status(String donation_status)
{
    this.donation_status = donation_status;
}

public void set_value_donor_visibility(String donor_visibility)
{
    this.donor_visibility = donor_visibility;
}
```

```
public String get_id()
{
   return id;
}

public String get_password()
{
   return password;
}

public String get_blood_group()
{
   return blood_group;
}

public String get_donation_status()
{
   return donation_status;
}

public String get_donor_visility()
{
   return donor_visibility;
}

public int login( String id, String pass) throws IOException
{
```

```
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
try {
    ob.readObjectFromFile("AllFileLocation", filelist);
} catch (IOException ex) {
    Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

int i, flag = 0;
for ( i=0; i<demo.n; i++ )
{
    ob.readObjectFromFile(filelist[i], adminlist2);

    String o_id = adminlist2[4];
    String o_pass = adminlist2[5];
    String donor_visibility = adminlist2[8];

    if( id.equals(o_id) && pass.equals(o_pass) )
    {
        if ( donor_visibility.equals("No") )
        {
            //Volunteer
            flag = 1;
        }
        else if ( donor_visibility.equals("Yes") )
        {
                //Donor
                flag = 2;
            }
        }
        else
        {
            //User id and password doesn't matched.

        }
    }
    return flag;
}

public void logout()
{
    System.exit(0);
}

public int reset_password(String id, String pass, String n_pass) throws IOException
{
    String filelist[] = new String[1000];
    String adminlist2[] = new String[1000];

    fileHandle ob = new fileHandle();
    try {
        ob.readObjectFromFile("AllFileLocation", filelist);
    } catch (IOException ex) {
        Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
    }


    int i, flag = 0;
    for ( i=0; i<demo.n; i++ )
    {
      ob.readObjectFromFile(filelist[i], adminlist2);


      String o_id = adminlist2[4];
      String o_pass = adminlist2[5];


      if( id.equals(o_id) && pass.equals(o_pass) )
      {
        //System.out.println("User id & password matched. You Can Change your
password......");
        flag = 1;


        adminlist2[5] = n_pass;


        ob.writeInCleanFile(filelist[i], adminlist2[0]);


        ob.writeInFile(filelist[i], adminlist2[1]);


        ob.writeInFile(filelist[i], adminlist2[2]);


        ob.writeInFile(filelist[i], adminlist2[3]);


        ob.writeInFile(filelist[i], adminlist2[4]);


        ob.writeInFile(filelist[i], adminlist2[5]);
```

```
                ob.writeInFile(filelist[i], adminlist2[6]);


                ob.writeInFile(filelist[i], adminlist2[7]);


                ob.writeInFile(filelist[i], adminlist2[8]);


            }
            else
            {
                //System.out.println("User id & password not matched.....");
            }
        }
      return flag;
    }

    public int donation_status(String id, String pass, String c_donation_status) throws
IOException
    {
        String filelist[] = new String[1000];
        String adminlist2[] = new String[1000];


        fileHandle ob = new fileHandle();
        try {
            ob.readObjectFromFile("AllFileLocation", filelist);
        } catch (IOException ex) {
            Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
        }
```

```
int i, flag = 0;
for( i=0; i<demo.n; i++ )
{
    ob.readObjectFromFile(filelist[i], adminlist2);

    String o_id = adminlist2[4];
    String o_pass = adminlist2[5];

    String o_donation_status = adminlist2[7];

    if( id.equals(o_id) && pass.equals(o_pass) )
    {
        //System.out.println("User id & password matched......");

        adminlist2[7] = c_donation_status;

        ob.writeInCleanFile(filelist[i], adminlist2[0]);

        ob.writeInFile(filelist[i], adminlist2[1]);

        ob.writeInFile(filelist[i], adminlist2[2]);

        ob.writeInFile(filelist[i], adminlist2[3]);

        ob.writeInFile(filelist[i], adminlist2[4]);

        ob.writeInFile(filelist[i], adminlist2[5]);
```

```
                ob.writeInFile(filelist[i], adminlist2[6]);

                ob.writeInFile(filelist[i], adminlist2[7]);

                ob.writeInFile(filelist[i], adminlist2[8]);

                flag = 1;
            }
            else
            {
                //System.out.println("User id & password not matched.....");
            }
        }
      return flag;
    }

}


class volunteerInterface extends adminInterface
{

    public void add_volunteer(adminInterface ob) throws IOException
    {
        generateRandom obid = new generateRandom();
        fileHandle obfile = new fileHandle();

        ob.set_value_admin_id(obid.generateID());
        ob.set_value_donor_visibility("No");
        obfile.writeObjectInFile(ob.get_id(), ob);
        obfile.writeInFile("AllFileLocation", ob.get_id());
        demo.n++;
        obfile.writeIntegerIncleanFile("Value_of_n", demo.n);

        //System.out.println("New volunteer succesfully added.....\n"+"New volunteer's id is : 
"+ob.get_id());

    }

    public void add_donor(adminInterface ob) throws IOException
    {
        generateRandom obid = new generateRandom();
        fileHandle obfile = new fileHandle();

        ob.set_value_admin_id(obid.generateID());

        ob.set_value_donor_visibility("Yes");
        obfile.writeObjectInFile(ob.get_id(), ob);
        obfile.writeInFile("AllFileLocation", ob.get_id());
        demo.n++;
        obfile.writeIntegerIncleanFile("Value_of_n", demo.n);

    }

    public int delete_donor(String id) throws IOException
    {
```

```
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
try {
    ob.readObjectFromFile("AllFileLocation", filelist);
} catch (IOException ex) {
    Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

int i, r=0;
int flag = 0;
for( i=0; i<demo.n; i++)
{
    ob.readObjectFromFile(filelist[i], adminlist2);

    String o_id = adminlist2[4];

    if( o_id.equals(id))
    {
        flag = 0;
        r = i;
    }
    else
    {
        flag = 1;
    }
}
```

```
    if( flag == 1 )
    {
        return 2;
        //System.out.println("Invalid id. User not found.....");
    }
    else
    {
        ob.writeBlankInCleanFile("AllFileLocation", "");
        for( i=0; i<r; i++ )
        {
            ob.writeInFile("AllFileLocation", filelist[i]);
        }
        for ( i=r; i<demo.n-1; i++)
        {
            filelist[i] = filelist[i+1];
            ob.writeInFile("AllFileLocation", filelist[i]);
        }
        demo.n--;
        ob.writeIntegerIncleanFile("Value_of_n", demo.n);

        return 1;
        //System.out.println("User succesfully deleted.....");
    }
}

void show_donarlist() throws IOException
{
    //Code for JTextArea for show only
```

```
JFrame frame = new JFrame();
frame.setBounds(350, 150, 600, 450);
frame.setTitle("Blood Bank Management System");

Font font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

JPanel panel = new JPanel();
panel.setLayout(null);
panel.setBackground(Color.red);

JTextArea jta = new JTextArea();
jta.setForeground(Color.white);
jta.setBackground(Color.gray);
jta.setFont(font1);
jta.setLineWrap(true);
jta.setWrapStyleWord(true);
jta.setEditable(false);

JScrollPane jsp = new JScrollPane(jta);
jsp.setBounds(15, 20, 560, 380);

panel.add(jsp);
frame.add(panel);

frame.setVisible(true);
frame.setResizable(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
// Main code of search
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
try {
    ob.readObjectFromFile("AllFileLocation", filelist);
} catch (IOException ex) {
    Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

int i,sum = 0;
for ( i=0; i<demo.n; i++ )
{
    ob.readObjectFromFile(filelist[i], adminlist2);

    String str = adminlist2[7];

    if ( str.equals("Yes"))
    {
        sum++;

        jta.append("Name : "+adminlist2[0]);

        jta.append("\nEmail : "+adminlist2[1]);

        jta.append("\nPhone Number : "+adminlist2[2]);
```

```
            jta.append("\nAddress : "+adminlist2[3]);


            jta.append("\nBlood Group : "+adminlist2[6]);


            jta.append("\nDonation Status : "+adminlist2[7]);


            jta.append("\n\n..................................\n\n");
        }
    }
     if ( sum <= 0)
    {
        jta.append("Sorry, No donor available at this moment. Check again after sometime.....");
    }
    else
    {
        jta.append("Total donors in the list : "+sum);
    }

}

void show_donorlist(String c_blood_group) throws IOException
{
    //Code for JTextArea for show only
    JFrame frame = new JFrame();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    Font font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);
```

```
JPanel panel = new JPanel();
panel.setLayout(null);
panel.setBackground(Color.red);

JTextArea jta = new JTextArea();
jta.setForeground(Color.white);
jta.setBackground(Color.gray);
jta.setFont(font1);
jta.setLineWrap(true);
jta.setWrapStyleWord(true);
jta.setEditable(false);

JScrollPane jsp = new JScrollPane(jta);
jsp.setBounds(15, 20, 560, 380);

panel.add(jsp);
frame.add(panel);

frame.setVisible(true);
frame.setResizable(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Main code of search
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
```

```
try {
   ob.readObjectFromFile("AllFileLocation", filelist);
} catch (IOException ex) {
   Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

int i,sum = 0;
for ( i=0; i<demo.n; i++ )
{
   ob.readObjectFromFile(filelist[i], adminlist2);
   String str = adminlist2[7];

   if ( str.equals("Yes"))
   {
      String o_blood_group = adminlist2[6];

      if ( c_blood_group.equals(o_blood_group) )
      {
         sum++;

         jta.append("Name : "+adminlist2[0]);

         jta.append("\nEmail : "+adminlist2[1]);

         jta.append("\nPhone Number : "+adminlist2[2]);

         jta.append("\nAddress : "+adminlist2[3]);
```

```
                jta.append("\nBlood Group : "+adminlist2[6]);


                jta.append("\nDonation Status : "+adminlist2[7]);


                jta.append("\n\n..................................\n\n");
            }
        }
    }
     if ( sum <= 0)
    {
        jta.append("Sorry, No donor available at this moment. Check again after sometime.....");
    }
    else
    {
        jta.append("Total donors of this blood in the list : "+sum);
    }
}

void show_donorlist_location(String c_address) throws IOException
{
    //Code for JTextArea for show only
    JFrame frame = new JFrame();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    Font font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

    JPanel panel = new JPanel();
```

```
panel.setLayout(null);
panel.setBackground(Color.red);

JTextArea jta = new JTextArea();
jta.setForeground(Color.white);
jta.setBackground(Color.gray);
jta.setFont(font1);
jta.setLineWrap(true);
jta.setWrapStyleWord(true);
jta.setEditable(false);

JScrollPane jsp = new JScrollPane(jta);
jsp.setBounds(15, 20, 560, 380);

panel.add(jsp);
frame.add(panel);

frame.setVisible(true);
frame.setResizable(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Main code of search
String filelist[] = new String[1000];
String adminlist2[] = new String[1000];

fileHandle ob = new fileHandle();
try {
   ob.readObjectFromFile("AllFileLocation", filelist);
```

```
} catch (IOException ex) {
   Logger.getLogger(userInterface.class.getName()).log(Level.SEVERE, null, ex);
}

int i,sum = 0;
for ( i=0; i<demo.n; i++ )
{
   ob.readObjectFromFile(filelist[i], adminlist2);
   String str = adminlist2[7];

   if ( str.equals("Yes"))
   {
      ob.readObjectFromFile(filelist[i], adminlist2);
      String o_address = adminlist2[3];

      if ( c_address.equals(o_address) )
      {
         sum++;

         jta.append("Name : "+adminlist2[0]);

         jta.append("\nEmail : "+adminlist2[1]);

         jta.append("\nPhone Number : "+adminlist2[2]);

         jta.append("\nAddress : "+adminlist2[3]);

         jta.append("\nBlood Group : "+adminlist2[6]);
```

```
                    jta.append("\nDonation Status : "+adminlist2[7]);


                    jta.append("\n\n.....................................\n\n");
                }
            }
        }
         if ( sum <= 0)
        {
            jta.append("Sorry, No donor available at this moment. Check again after sometime.....");
        }
        else
        {
            jta.append("Total donors of this blood in the list : "+sum);
        }
    }

}

class donorInterface extends adminInterface
{

}



class patientInterface extends userInterface
{
}

class generateRandom
{
   public String generateID()
   {
      // create instance of Random class
         Random rand = new Random();



         // Generate random integers in range 0 to 999
      int Rand_ID = rand.nextInt(1000000);

      String rand_id = "" + Rand_ID;

      return rand_id;
   }
}

//************************************************** Control Part (File Handle)
************************************************

class file
{
   private String filename;

   public void setFileName(String filename)
   {
      this.filename = filename;
```

```
    }

    public String get_filename()
    {
        return filename;
    }
}


class fileHandle
{

    public void writeInFile(String filename, String contents) throws IOException
    {
        try (FileWriter fw = new FileWriter(filename, true)) {
            fw.write(contents);
            fw.write("\n");
        }
    }

    public void writeInCleanFile(String filename, String contents) throws IOException
    {
        try (FileWriter fw = new FileWriter(filename)) {
            fw.write(contents);
            fw.write("\n");
        }
    }
```

```
public void writeBlankInCleanFile(String filename, String contents) throws IOException
{
   try (FileWriter fw = new FileWriter(filename)) {
      fw.write(contents);
   }
}

public String readFromFile(String filename) throws FileNotFoundException, IOException
{
   String con = "";
   try (FileReader fr = new FileReader(filename)) {
      BufferedReader br = new BufferedReader(fr);

      String line;
      while((line = br.readLine()) != null)
      {
         con =  con + line;
      }
   }

   return con;
}

public String searchFileFromFolder(file filelist[], int n, String id)
{
   int i,flag = 0;
   String fileLocation = "";
   for ( i=0; i<n; i++ )
```

```
    {
        if ( id.equals(filelist[i].get_filename()))
        {
            fileLocation = fileLocation + id;
            return fileLocation;
        }
        else
        {
            flag = 1;
        }
    }
    if ( flag == 1 )
    {
        String str = "Invaliad User";
        return str;
    }
    return null;
}

public void writeObjectInFile (String filename, adminInterface ob) throws IOException
{
    writeInFile(filename, ob.get_name());

    writeInFile(filename, ob.get_email());

    writeInFile(filename, ob.get_phone());

    writeInFile(filename, ob.get_address());
    writeInFile(filename, ob.get_id());

    writeInFile(filename, ob.get_password());

    writeInFile(filename, ob.get_blood_group());

    writeInFile(filename, ob.get_donation_status());

    writeInFile(filename, ob.get_donor_visility());

}

public void writeObjectInCleanFile (String filename, adminInterface ob) throws IOException
{
    writeInCleanFile(filename, ob.get_name());

    writeInFile(filename, ob.get_email());

    writeInFile(filename, ob.get_phone());

    writeInFile(filename, ob.get_address());

    writeInFile(filename, ob.get_id());

    writeInFile(filename, ob.get_password());

    writeInFile(filename, ob.get_blood_group());
```

```
        writeInFile(filename, ob.get_donation_status());

        writeInFile(filename, ob.get_donor_visility());

    }

    public String[] readObjectFromFile(String filename, String adminlist2[]) throws
FileNotFoundException, IOException
    {
        try (FileReader fr = new FileReader(filename)) {
            BufferedReader br = new BufferedReader(fr);

            int i=0;
            String line;
            while((line = br.readLine()) != null)
            {
                adminlist2[i] = line;
                i++;
            }
        }

        return adminlist2;
    }

    public void writeIntegerIncleanFile(String filename, int n) throws IOException
    {
        String str = Integer.toString(n);
        writeInCleanFile(filename, str);
    }

    public int readIntegerFromFile(String filename) throws IOException
    {
        String myString = readFromFile(filename);

        int foo = Integer.parseInt(myString);

        return foo;

    }
}

//*********************************************** View Part (GUI Create)
**********************************************

class GUI1 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label;
    private Font font1, font2;
    private JButton button1, button2;

    GUI1()
    {
        initComponents();
    }
```

```
public void initComponents()
{
   icon = new ImageIcon(getClass().getResource("icon.png"));
   this.setIconImage(icon.getImage());

   con = this.getContentPane();
   con.setLayout(null);
   con.setBackground(Color.red);

   font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
   font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);

   label = new JLabel("Blood Bank Management System");
   label.setBounds(110, 90, 400, 50);
   label.setFont(font1);
   con.add(label);

   button1 = new JButton("Admin");
   button1.setFont(font2);
   button1.setBounds(110, 220, 150, 50);
   con.add(button1);

   button1.addActionListener((ActionListener) new ActionListener(){

      @Override
      public void actionPerformed(ActionEvent e)
      {
           GUI6 gui6 = new GUI6();
           gui6.AdminGui();
      }
   });

   button2 = new JButton("Patient");
   button2.setFont(font2);
   button2.setBounds(340, 220, 150, 50);
   con.add(button2);

   button2.addActionListener((ActionListener) new ActionListener(){

      @Override
      public void actionPerformed(ActionEvent e)
      {
           GUI2 gui2 = new GUI2();
           gui2.PatientGUI();
      }
   });
}

public void HomepageGUI()
{
   GUI1 frame = new GUI1();
   frame.setBounds(350, 150, 600, 450);
   frame.setTitle("Blood Bank Management System");

   frame.setVisible(true);
```

```
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}

class GUI2 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label;
    private Font font1, font2;
    private JButton button1, button2, button3, button4;

    GUI2()
    {
        initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());

        con = this.getContentPane();
        con.setLayout(null);
        con.setBackground(Color.red);
```

```
font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 30);
font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);

label = new JLabel("WELCOME");
label.setBounds(220, 50, 200, 50);
label.setFont(font1);
con.add(label);

button1 = new JButton("Search by Blood Group");
button1.setFont(font2);
button1.setBounds(40, 120, 330, 50);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

  @Override
  public void actionPerformed(ActionEvent e)
  {
    dispose();

    GUI3 gui3 = new GUI3();
    gui3.SearchByBloodGUI();
  }
});

button2 = new JButton("Search by Blood Group & Location");
button2.setFont(font2);
button2.setBounds(40, 200, 330, 50);
```

```
con.add(button2);

button2.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      dispose();

      GUI4 gui4 = new GUI4();
      gui4.SearchByBlooadLocationGUI();
   }
});

button3 = new JButton("Search Volunteers by Location");
button3.setFont(font2);
button3.setBounds(40, 280, 330, 50);
con.add(button3);

button3.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      dispose();

      GUI5 gui5 = new GUI5();
      gui5.SearchVolunteerByLocationGUI();
```

```
        }
    });

    button4 = new JButton("Exit");
    button4.setFont(font2);
    button4.setBounds(450, 200, 100, 40);
    con.add(button4);

    button4.addActionListener((ActionListener) new ActionListener(){

        @Override
        public void actionPerformed(ActionEvent e)
        {
            dispose();

            //System.exit(0);
        }
    });

}

public void PatientGUI()
{
    GUI2 frame = new GUI2();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    frame.setVisible(true);
```

```
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}

//*********Patient Part

class GUI3 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label1, label2;
    private Font font1, font2;
    private JComboBox comb;
    private JButton button1, button2;

    private final String[] BloodGroup = {"A+", "A-", "B+", "B-", "AB+", "AB-", "O+"};

    GUI3()
    {
        initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());
```

```
con = this.getContentPane();
con.setLayout(null);
con.setBackground(Color.red);

font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);

label1 = new JLabel("Thanks for Using Our Blood Bank");
label1.setBounds(110, 90, 400, 50);
label1.setFont(font1);
con.add(label1);

label2 = new JLabel("Blood Group : ");
label2.setBounds(150, 180, 170, 40);
label2.setFont(font2);
con.add(label2);

comb = new JComboBox(BloodGroup);
comb.setBounds(320, 180, 100, 40);
comb.setFont(font2);
con.add(comb);

button1 = new JButton("Submit");
button1.setBounds(150, 270, 100, 40);
button1.setFont(font2);
con.add(button1);
```

```
button1.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        dispose();

        if ( comb.getSelectedItem().toString().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Please enter all field.");
        }
        else
        {
            patientInterface ob = new patientInterface();
            ob.search_blood(comb.getSelectedItem().toString());
        }
    }
});

button2 = new JButton("Exit");
button2.setBounds(320, 270, 100, 40);
button2.setFont(font2);
con.add(button2);

button2.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
```

```
            {
                dispose();

                //System.exit(0);
            }
        });

    }

    public void SearchByBloodGUI()
    {
        GUI3 frame = new GUI3();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}

class GUI4 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private Font font1, font2;
    private JLabel label1, label2, label3;
private JComboBox comb;
private JTextField tf;
private JButton button1, button2;

private final String[] BloodGroup = {"A+", "A-", "B+", "B-", "AB+", "AB-", "O+"};

GUI4()
{
   initComponents();
}

public void initComponents()
{
   icon = new ImageIcon(getClass().getResource("icon.png"));
   this.setIconImage(icon.getImage());

   con = this.getContentPane();
   con.setLayout(null);
   con.setBackground(Color.red);

   font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
   font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);

   label1 = new JLabel("Thanks for Using Our Blood Bank");
   label1.setBounds(110, 90, 400, 50);
   label1.setFont(font1);
   con.add(label1);
```

```
label2 = new JLabel("Blood Group : ");
label2.setBounds(110, 180, 170, 40);
label2.setFont(font2);
con.add(label2);

comb = new JComboBox(BloodGroup);
comb.setBounds(290, 180, 220, 40);
comb.setFont(font2);
con.add(comb);

label3 = new JLabel("Location : ");
label3.setBounds(110, 230, 170, 40);
label3.setFont(font2);
con.add(label3);

tf = new JTextField();
tf.setBounds(290, 230, 220, 40);
tf.setFont(font2);
con.add(tf);

button1 = new JButton("Submit");
button1.setBounds(150, 300, 100, 40);
button1.setFont(font2);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

   @Override
```

```
    public void actionPerformed(ActionEvent e)
    {
        dispose();

        if (tf.getText().isEmpty() || comb.getSelectedItem().toString().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Please enter all field.");
        }
        else
        {
            patientInterface ob = new patientInterface();
            ob.search_blood(tf.getText(), comb.getSelectedItem().toString());
        }
    }
});

button2 = new JButton("Exit");
button2.setBounds(320, 300, 100, 40);
button2.setFont(font2);
con.add(button2);

button2.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        dispose();
```

```
            //System.exit(0);
        }
    });

  }

  public void SearchByBlooadLocationGUI()
  {
    GUI4 frame = new GUI4();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");


    frame.setVisible(true);
    frame.setResizable(false);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}

class GUI5 extends JFrame
{
  private ImageIcon icon;
  private Container con;
  private Font font1, font2;
  private JLabel label1, label2;
  private JTextField tf;
  private JButton button1, button2;

  GUI5()
```

```
{
    initComponents();
}

public void initComponents()
{
    icon = new ImageIcon(getClass().getResource("icon.png"));
    this.setIconImage(icon.getImage());

    con = this.getContentPane();
    con.setLayout(null);
    con.setBackground(Color.red);

    font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
    font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);

    label1 = new JLabel("Thanks for Using Our Blood Bank");
    label1.setBounds(110, 90, 400, 50);
    label1.setFont(font1);
    con.add(label1);

    label2 = new JLabel("Location : ");
    label2.setBounds(110, 190, 170, 40);
    label2.setFont(font2);
    con.add(label2);

    tf = new JTextField();
    tf.setBounds(290, 190, 220, 40);
tf.setFont(font2);
con.add(tf);

button1 = new JButton("Submit");
button1.setBounds(130, 270, 100, 40);
button1.setFont(font2);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      dispose();

      if ( tf.getText().isEmpty())
      {
         JOptionPane.showMessageDialog(null, "Please enter all field.");
      }
      else
      {
         patientInterface ob = new patientInterface();
         ob.search_vlounteer_by_location(tf.getText());
      }
   }
});

button2 = new JButton("Exit");
```

```
        button2.setBounds(390, 270, 100, 40);
        button2.setFont(font2);
        con.add(button2);

        button2.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                dispose();

                //System.exit(0);
            }
        });

    }

    public void SearchVolunteerByLocationGUI()
    {
        GUI5 frame = new GUI5();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```
//*********Admin Part

class GUI6 extends JFrame
{
   private ImageIcon icon;
   private Container con;
   private Font font1, font2;
   private JLabel label1, label2, label3;
   JTextField tf;
   JPasswordField pf;
   private JButton button1, button2;

   GUI6()
   {
      initComponents();
   }

   public void initComponents()
   {
      icon = new ImageIcon(getClass().getResource("icon.png"));
      this.setIconImage(icon.getImage());

      con = this.getContentPane();
      con.setLayout(null);
      con.setBackground(Color.red);

      font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 30);
```

```
font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);

label1 = new JLabel("Donate Blood, Save Life");
label1.setBounds(150, 80, 400, 50);
label1.setFont(font1);
con.add(label1);

label2 = new JLabel("User ID :");
label2.setBounds(130, 170, 100, 40);
label2.setFont(font2);
con.add(label2);

tf = new JTextField();
tf.setBounds(240, 170, 230, 40);
tf.setFont(font2);
con.add(tf);

label3 = new JLabel("Password :");
label3.setBounds(130, 230, 100, 40);
label3.setFont(font2);
con.add(label3);

pf = new JPasswordField();
pf.setBounds(240, 230, 230, 40);
pf.setFont(font2);
pf.setEchoChar('*');
con.add(pf);
```

```
button1 = new JButton("Exit");
button1.setBounds(160, 290, 100, 40);
button1.setFont(font2);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      dispose();

      GUI1 gui1 = new GUI1();
      gui1.HomepageGUI();
   }
});

button2 = new JButton("Log in");
button2.setBounds(330, 290, 100, 40);
button2.setFont(font2);
con.add(button2);

button2.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      dispose();
```

```
if ( tf.getText().isEmpty() || pf.getText().isEmpty() )
{
    JOptionPane.showMessageDialog(null, "Please enter all field.");
}
else
{
    adminInterface ob = new adminInterface();

    int x = 0;
    try {
        x = ob.login(tf.getText(), pf.getText());
    } catch (IOException ex) {
        Logger.getLogger(GUI6.class.getName()).log(Level.SEVERE, null, ex);
    }

    if ( x == 1 )
    {
        //Volunteer
        dispose();
        GUI8 gui8 = new GUI8();
        gui8.VolunteerGUI();
    }

    else if ( x == 2 )
    {
        //Donor
        dispose();
```

```
                        GUI7 gui7 = new GUI7();
                        gui7.DonorGUI();
                    }

                    else //if ( x == 3 )
                    {
                        JOptionPane.showMessageDialog(null, "Sorry!\nUser id and password doesn't matched.\nPlease try again.");
                        //System.exit(0);
                    }
                }

            }
        });

    }

    public void AdminGui()
    {
        GUI6 frame = new GUI6();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
```

```
//*********Donor Part

class GUI7 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label;
    private Font font1, font2;
    private JButton button1, button2, button3, button4, button5, button6, button7;

    GUI7()
    {
        initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());

        con = this.getContentPane();
        con.setLayout(null);
        con.setBackground(Color.red);

        font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 30);
        font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 20);
```

```
label = new JLabel("Welcome Dear Donor");
label.setBounds(170, 30, 300, 50);
label.setFont(font1);
con.add(label);

button1 = new JButton("Search by Blood Group");
button1.setFont(font2);
button1.setBounds(20, 90, 330, 40);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

  @Override
  public void actionPerformed(ActionEvent e)
  {
    //dispose();

    GUI3 gui3 = new GUI3();
    gui3.SearchByBloodGUI();
  }
});

button2 = new JButton("Search by Blood Group & Location");
button2.setFont(font2);
button2.setBounds(20, 170, 330, 40);
con.add(button2);

button2.addActionListener((ActionListener) new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();

        GUI4 gui4 = new GUI4();
        gui4.SearchByBlooadLocationGUI();
    }
});

button3 = new JButton("Search Volunteers by Location");
button3.setFont(font2);
button3.setBounds(20, 250, 330, 40);
con.add(button3);

button3.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();

        GUI5 gui5 = new GUI5();
        gui5.SearchVolunteerByLocationGUI();
    }
});
```

```
button4 = new JButton("Update Donation Status");
button4.setFont(font2);
button4.setBounds(20, 330, 330, 40);
con.add(button4);

button4.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      //dispose();

      GUI11 gui11 = new GUI11();
      gui11.UpdateDonationStatusGUI();
   }
});

button5 = new JButton("Log Out");
button5.setFont(font2);
button5.setBounds(370, 130, 200, 40);
con.add(button5);

button5.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      //dispose();
```

```
        adminInterface ob = new adminInterface();
        JOptionPane.showMessageDialog(null, "Thanks for using our system.");
        ob.logout();
    }
});

button6 = new JButton("Reset Password");
button6.setFont(font2);
button6.setBounds(370, 210, 200, 40);
con.add(button6);

button6.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();

        GUI12 gui12 = new GUI12();
        gui12.ResetPasswordGUI();
    }
});

button7 = new JButton("Exit");
button7.setFont(font2);
button7.setBounds(370, 290, 200, 40);
con.add(button7);
```

```
        button7.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                dispose();

                //System.exit(0);
            }
        });

    }

    public void DonorGUI()
    {
        GUI7 frame = new GUI7();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}

//*********Volunteer Part
```

```
class GUI8 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label;
    private Font font1, font2;
    private JButton button1, button2, button3, button4, button5, button6, button7, button8,
button9, button10, button11, button12, button13;

    GUI8()
    {
        initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());

        con = this.getContentPane();
        con.setLayout(null);
        con.setBackground(Color.red);

        font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
        font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 14);

        label = new JLabel("Welcome Dear Volunteer");
        label.setBounds(170, 30, 300, 50);
        label.setFont(font1);
        con.add(label);

        button1 = new JButton("Search by Blood Group");
        button1.setFont(font2);
        button1.setBounds(20, 90, 250, 30);
        con.add(button1);

        button1.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                //dispose();

                GUI3 gui3 = new GUI3();
                gui3.SearchByBloodGUI();
            }
        });

        button2 = new JButton("Search by Blood Group & Location");
        button2.setFont(font2);
        button2.setBounds(20, 140, 250, 30);
        con.add(button2);

        button2.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                //dispose();

                GUI4 gui4 = new GUI4();
                gui4.SearchByBlooadLocationGUI();
            }
        });

        button3 = new JButton("Search Volunteers by Location");
        button3.setFont(font2);
        button3.setBounds(20, 190, 250, 30);
        con.add(button3);

        button3.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                //dispose();

                GUI5 gui5 = new GUI5();
                gui5.SearchVolunteerByLocationGUI();
            }
        });

        button4 = new JButton("Update Donation Status");
        button4.setFont(font2);
```

```
button4.setBounds(20, 240, 250, 30);
con.add(button4);

button4.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      //dispose();

      GUI11 gui11 = new GUI11();
      gui11.UpdateDonationStatusGUI();
   }
});

button5 = new JButton("Add Volunteer");
button5.setFont(font2);
button5.setBounds(20, 290, 250, 30);
con.add(button5);

button5.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      //dispose();

      GUI9 gui9 = new GUI9();
```

```
        gui9.AddVolunteerGUI();
    }
});

button6 = new JButton("Add Donor");
button6.setFont(font2);
button6.setBounds(20, 340, 250, 30);
con.add(button6);

button6.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();


        GUI10 gui10= new GUI10();
        gui10.AddDonorGUI();
    }
});

button7 = new JButton("Delete Donor");
button7.setFont(font2);
button7.setBounds(320, 90, 250, 30);
con.add(button7);

button7.addActionListener((ActionListener) new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();


        GUI13 gui13 = new GUI13();
        gui13.DeleteDonorGUI();
    }
});

button8 = new JButton("Show All Donors");
button8.setFont(font2);
button8.setBounds(320, 140, 250, 30);
con.add(button8);

button8.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();
        volunteerInterface ob = new volunteerInterface();
        try {
            ob.show_donarlist();
        } catch (IOException ex) {
            Logger.getLogger(GUI8.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});

button9 = new JButton("Show All donors by Blood Group");
button9.setFont(font2);
button9.setBounds(320, 190, 250, 30);
con.add(button9);

button9.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();

        GUI14 gui14 = new GUI14();
        gui14.ShowDonorlistBloodGroupGUI();
    }
});

button10 = new JButton("Show All donors by Location");
button10.setFont(font2);
button10.setBounds(320, 240, 250, 30);
con.add(button10);

button10.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();

        GUI15 gui15 = new GUI15();
        gui15.ShowDonorlistLocationGUI();
    }
});

button11 = new JButton("Reset Password");
button11.setFont(font2);
button11.setBounds(320, 290, 250, 30);
con.add(button11);

button11.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        //dispose();

        GUI12 gui12 = new GUI12();
        gui12.ResetPasswordGUI();
    }
});

button12 = new JButton("Log Out");
button12.setFont(font2);
```

```
button12.setBounds(320, 340, 250, 30);
con.add(button12);

button12.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      //dispose();

      volunteerInterface ob = new volunteerInterface();
      JOptionPane.showMessageDialog(null, "Thanks for using our system.");
      ob.logout();
   }
});

button13 = new JButton("Exit");
button13.setFont(font2);
button13.setBounds(170, 380, 250, 30);
con.add(button13);

button13.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
      dispose();
```

```
                //System.exit(0);
            }
        });

    }

    public void VolunteerGUI()
    {
        GUI8 frame = new GUI8();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}

class GUI9 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label1, label2, label3, label4, label5, label6, label7, label8;
    private JTextField tf1, tf2, tf3, tf4;
    private JPasswordField pf;
    private Font font1, font2;
    private JComboBox comb1, comb2;
private final String[] str1 = {"A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-"};
private final String[] str2 = {"Yes", "No"};
private JButton button1, button2;

GUI9()
{
  initComponents();
}

public void initComponents()
{
    icon = new ImageIcon(getClass().getResource("icon.png"));
    this.setIconImage(icon.getImage());

    con = this.getContentPane();
    con.setLayout(null);
    con.setBackground(Color.red);

    font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 24);
    font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 16);

    label1 = new JLabel("Be a blood donor, be a real hero");
    label1.setBounds(140, 20, 400, 40);
    label1.setFont(font1);
    con.add(label1);

    label2 = new JLabel("Name :");
    label2.setBounds(50, 80, 100, 30);
```

```
label2.setFont(font2);
con.add(label2);

tf1 = new JTextField();
tf1.setBounds(160, 80, 350, 30);
tf1.setFont(font2);
con.add(tf1);

label3 = new JLabel("Email :");
label3.setBounds(50, 120, 100, 30);
label3.setFont(font2);
con.add(label3);

tf2 = new JTextField();
tf2.setBounds(160, 120, 350, 30);
tf2.setFont(font2);
con.add(tf2);

label4 = new JLabel("Phone :");
label4.setBounds(50, 160, 100, 30);
label4.setFont(font2);
con.add(label4);

tf3 = new JTextField();
tf3.setBounds(160, 160, 350, 30);
tf3.setFont(font2);
con.add(tf3);
```

```
label5 = new JLabel("Address :");
label5.setBounds(50, 200, 100, 30);
label5.setFont(font2);
con.add(label5);

tf4 = new JTextField();
tf4.setBounds(160, 200, 350, 30);
tf4.setFont(font2);
con.add(tf4);

label6 = new JLabel("Password :");
label6.setBounds(50, 240, 100, 30);
label6.setFont(font2);
con.add(label6);

pf = new JPasswordField();
pf.setBounds(160, 240, 350, 30);
pf.setFont(font2);
con.add(pf);

label7 = new JLabel("Blood Group :");
label7.setBounds(50, 280, 100, 30);
label7.setFont(font2);
con.add(label7);

comb1 = new JComboBox(str1);
comb1.setBounds(160, 280, 100, 30);
comb1.setFont(font2);
```

```
        con.add(comb1);


        label8 = new JLabel("Donation Status :");
        label8.setBounds(280, 280, 120, 30);
        label8.setFont(font2);
        con.add(label8);
        comb2 = new JComboBox(str2);
        comb2.setBounds(410, 280, 100, 30);
        comb2.setFont(font2);
        con.add(comb2);


        button1 = new JButton("Submit");
        button1.setFont(font2);
        button1.setBounds(360, 340, 100, 30);
        con.add(button1);


        button1.addActionListener((ActionListener) new ActionListener(){


            @Override
            public void actionPerformed(ActionEvent e)
            {
                dispose();


                if ( tf1.getText().isEmpty() || tf2.getText().isEmpty() || tf3.getText().isEmpty() ||
    tf4.getText().isEmpty() || pf.getText().isEmpty() || comb1.getSelectedItem().toString().isEmpty()
    || comb2.getSelectedItem().toString().isEmpty() )
                {
                    JOptionPane.showMessageDialog(null, "Please enter all field");
                }
                else
                {
                    volunteerInterface ob = new volunteerInterface();
                    ob.set_value_user(tf1.getText(), tf2.getText(), tf3.getText(), tf4.getText());
                    ob.set_value_admin(pf.getText(), comb1.getSelectedItem().toString(),
    comb2.getSelectedItem().toString());

                    try {
                      ob.add_volunteer(ob);
                    } catch (IOException ex) {
                      Logger.getLogger(GUI9.class.getName()).log(Level.SEVERE, null, ex);
                    }

                    JOptionPane.showMessageDialog(null, "New volunteer succesfully
    added.\n"+"New volunteer's id is : "+ob.get_id());
                    //System.exit(0);
                }
            }
        });

        button2 = new JButton("Back");
        button2.setFont(font2);
        button2.setBounds(110, 340, 100, 30);
        con.add(button2);

        button2.addActionListener((ActionListener) new ActionListener(){

            @Override
```

```
            public void actionPerformed(ActionEvent e)
            {
                dispose();

                GUI8 gui8 = new GUI8();
                gui8.VolunteerGUI();
            }
        });

    }

    public void AddVolunteerGUI()
    {
        GUI9 frame = new GUI9();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class GUI10 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label1, label2, label3, label4, label5, label6, label7, label8;
    private JTextField tf1, tf2, tf3, tf4;
    private JPasswordField pf;
    private Font font1, font2;
    private JComboBox comb1, comb2;
    private final String[] str1 = {"A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-"};
    private final String[] str2 = {"Yes", "No"};
    private JButton button1, button2;

    GUI10()
    {
      initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());

        con = this.getContentPane();
        con.setLayout(null);
        con.setBackground(Color.red);

        font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 24);
        font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 16);

        label1 = new JLabel("Be a blood donor, be a real hero");
        label1.setBounds(140, 20, 400, 40);
        label1.setFont(font1);
```

```
con.add(label1);

label2 = new JLabel("Name :");
label2.setBounds(50, 80, 100, 30);
label2.setFont(font2);
con.add(label2);

tf1 = new JTextField();
tf1.setBounds(160, 80, 350, 30);
tf1.setFont(font2);
con.add(tf1);

label3 = new JLabel("Email :");
label3.setBounds(50, 120, 100, 30);
label3.setFont(font2);
con.add(label3);

tf2 = new JTextField();
tf2.setBounds(160, 120, 350, 30);
tf2.setFont(font2);
con.add(tf2);

label4 = new JLabel("Phone :");
label4.setBounds(50, 160, 100, 30);
label4.setFont(font2);
con.add(label4);

tf3 = new JTextField();
```

```
tf3.setBounds(160, 160, 350, 30);
tf3.setFont(font2);
con.add(tf3);

label5 = new JLabel("Address :");
label5.setBounds(50, 200, 100, 30);
label5.setFont(font2);
con.add(label5);

tf4 = new JTextField();
tf4.setBounds(160, 200, 350, 30);
tf4.setFont(font2);
con.add(tf4);

label6 = new JLabel("Password :");
label6.setBounds(50, 240, 100, 30);
label6.setFont(font2);
con.add(label6);

pf = new JPasswordField();
pf.setBounds(160, 240, 350, 30);
pf.setFont(font2);
con.add(pf);

label7 = new JLabel("Blood Group :");
label7.setBounds(50, 280, 100, 30);
label7.setFont(font2);
con.add(label7);
```

```
comb1 = new JComboBox(str1);
comb1.setBounds(160, 280, 100, 30);
comb1.setFont(font2);
con.add(comb1);

label8 = new JLabel("Donation Status :");
label8.setBounds(280, 280, 120, 30);
label8.setFont(font2);
con.add(label8);
comb2 = new JComboBox(str2);
comb2.setBounds(410, 280, 100, 30);
comb2.setFont(font2);
con.add(comb2);

button1 = new JButton("Submit");
button1.setFont(font2);
button1.setBounds(360, 340, 100, 30);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
       dispose();
```

```
                if ( tf1.getText().isEmpty() || tf2.getText().isEmpty() || tf3.getText().isEmpty() ||
tf4.getText().isEmpty() || pf.getText().isEmpty() || comb1.getSelectedItem().toString().isEmpty()
|| comb2.getSelectedItem().toString().isEmpty() )

                {

                    JOptionPane.showMessageDialog(null, "Please enter all field");

                }


                else

                {

                    donorInterface ob = new donorInterface();

                    ob.set_value_user(tf1.getText(), tf2.getText(), tf3.getText(), tf4.getText());

                    ob.set_value_admin(pf.getText(), comb1.getSelectedItem().toString(),
comb2.getSelectedItem().toString());


                    volunteerInterface ob2 = new volunteerInterface();

                    try {

                        ob2.add_donor(ob);

                    } catch (IOException ex) {

                        Logger.getLogger(GUI10.class.getName()).log(Level.SEVERE, null, ex);

                    }


                    JOptionPane.showMessageDialog(null, "New donor succesfully added.\n"+"New
donor's id is : "+ob.get_id());

                    //System.exit(0);

                }

            }

        });


        button2 = new JButton("Back");

        button2.setFont(font2);
```

```
}

    button2.setBounds(110, 340, 100, 30);
    con.add(button2);

    button2.addActionListener((ActionListener) new ActionListener(){

      @Override
      public void actionPerformed(ActionEvent e)
      {
        dispose();

        GUI8 gui8 = new GUI8();
        gui8.VolunteerGUI();
      }
     });

  }

  public void AddDonorGUI()
  {
    GUI10 frame = new GUI10();
    frame.setBounds(350, 150, 600, 450);
    frame.setTitle("Blood Bank Management System");

    frame.setVisible(true);
    frame.setResizable(false);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

```
class GUI11 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private Font font1, font2;
    private JTextField tf;
    private JPasswordField pf;
    private JLabel label1, label2, label3, label4;
    private JComboBox comb;
    private JButton button1, button2;
    private final String[] str = {"Yes", "No"};

    GUI11()
    {
        initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());

        con = this.getContentPane();
        con.setLayout(null);
        con.setBackground(Color.red);

        font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
```

```
font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

label1 = new JLabel("Update Donation Status");
label1.setBounds(170, 50, 300, 40);
label1.setFont(font1);
con.add(label1);

label2 = new JLabel("Enter your id : ");
label2.setBounds(70, 120, 170, 40);
label2.setFont(font2);
con.add(label2);

tf = new JTextField();
tf.setBounds(250, 120, 300, 40);
tf.setFont(font2);
con.add(tf);

label3 = new JLabel("Enter your password :");
label3.setBounds(70, 180, 170, 40);
label3.setFont(font2);
con.add(label3);

pf = new JPasswordField();
pf.setBounds(250, 180, 300, 40);
pf.setFont(font2);
con.add(pf);

label4 = new JLabel("Donation Status :");
```

```
        label4.setBounds(70, 240, 170, 40);
        label4.setFont(font2);
        con.add(label4);



        comb = new JComboBox(str);
        comb.setBounds(340, 240, 100, 40);
        comb.setFont(font2);
        con.add(comb);

        button1 = new JButton("Update Status");
        button1.setFont(font2);
        button1.setBounds(100, 300, 160, 40);
        con.add(button1);

        button1.addActionListener((ActionListener) new ActionListener(){

          @Override
          public void actionPerformed(ActionEvent e)
          {
            dispose();

            if ( tf.getText().isEmpty() || pf.getText().isEmpty() ||
    comb.getSelectedItem().toString().isEmpty() )
            {
              JOptionPane.showMessageDialog(null, "Please enter all field");
            }
            else
            {
```

```
          int x = 0;
          donorInterface ob = new donorInterface();
          try {
            x = ob.donation_status(tf.getText(), pf.getText(),
comb.getSelectedItem().toString());
          } catch (IOException ex) {
            Logger.getLogger(GUI10.class.getName()).log(Level.SEVERE, null, ex);
          }

          if ( x == 1 )
          {
            JOptionPane.showMessageDialog(null, "Donation Status Updated...");
            //System.exit(0);
          }

          else //if ( x == 3 )
          {
            JOptionPane.showMessageDialog(null, "Sorry!\nUser id and password doesn't
matched.\nPlease try again.");
            //System.exit(0);
          }
        }
      }
    });

    button2 = new JButton("Exit");
    button2.setFont(font2);
    button2.setBounds(360, 300, 160, 40);
    con.add(button2);
```

```
        button2.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                dispose();
                //System.exit(0);
            }
         });

    }

    public void UpdateDonationStatusGUI()
    {
        GUI11 frame = new GUI11();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class GUI12 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label1, label2, label3, label4;
    private JTextField tf;
    private JPasswordField pf1, pf2;
    private Font font1, font2;
    private JButton button1, button2;

    GUI12()
    {
        initComponents();
    }

    public void initComponents()
    {
        icon = new ImageIcon(getClass().getResource("icon.png"));
        this.setIconImage(icon.getImage());

        con = this.getContentPane();
        con.setLayout(null);
        con.setBackground(Color.red);

        font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
        font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

        label1 = new JLabel("Change Your Password");
        label1.setBounds(160, 50, 300, 40);
        label1.setFont(font1);
        con.add(label1);
```

```
label2 = new JLabel("Enter your id : ");
label2.setBounds(70, 110, 170, 40);
label2.setFont(font2);
con.add(label2);

tf = new JTextField();
tf.setBounds(250, 110, 300, 40);
tf.setFont(font2);
con.add(tf);

label3 = new JLabel("Enter old password :");
label3.setBounds(70, 160, 170, 40);
label3.setFont(font2);
con.add(label3);

pf1 = new JPasswordField();
pf1.setBounds(250, 160, 300, 40);
pf1.setFont(font2);
con.add(pf1);

label4 = new JLabel("Enter new password :");
label4.setBounds(70, 210, 170, 40);
label4.setFont(font2);
con.add(label4);



pf2 = new JPasswordField();
```

```
pf2.setBounds(250, 210, 300, 40);
pf2.setFont(font2);
con.add(pf2);

button1 = new JButton("Reset Password");
button1.setFont(font2);
button1.setBounds(100, 280, 160, 40);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        dispose();

        if ( tf.getText().isEmpty() || pf1.getText().isEmpty() || pf2.getText().isEmpty() )
        {
            JOptionPane.showMessageDialog(null, "Please enter all field");
        }
        else
        {
            int x = 0;
            adminInterface ob = new adminInterface();
            try {
                x = ob.reset_password(tf.getText(),pf1.getText(), pf2.getText());
            } catch (IOException ex) {
                Logger.getLogger(GUI12.class.getName()).log(Level.SEVERE, null, ex);
```

```
                    }

                    if ( x == 1 )
                    {
                        JOptionPane.showMessageDialog(null, "Password Updated.");
                        //System.exit(0);
                    }
                    else //if ( x == 2)
                    {
                        JOptionPane.showMessageDialog(null, "Sorry!\nUser id and password doesn't
matched.\nPlease try again.");
                    }
                }
            }
        });

        button2 = new JButton("Exit");
        button2.setFont(font2);
        button2.setBounds(360, 280, 160, 40);
        con.add(button2);

        button2.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                dispose();
                //System.exit(0);
            }
```

```
         });
    }

    public void ResetPasswordGUI()
    {
        GUI12 frame = new GUI12();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class GUI13 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label1, label2, label3;
    private JTextField tf;
    private Font font1, font2;
    private JButton button1, button2;

    GUI13()
    {
        initComponents();
    }
```

```
public void initComponents()
{
    icon = new ImageIcon(getClass().getResource("icon.png"));
    this.setIconImage(icon.getImage());

    con = this.getContentPane();
    con.setLayout(null);
    con.setBackground(Color.red);

    font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
    font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

    label1 = new JLabel("Delete a Donor");
    label1.setBounds(200, 70, 300, 40);
    label1.setFont(font1);
    con.add(label1);

    label2 = new JLabel("Enter id :");
    label2.setBounds(80, 180, 100, 40);
    label2.setFont(font2);
    con.add(label2);

    tf = new JTextField();
    tf.setBounds(195, 180, 300, 40);
    tf.setFont(font2);
    con.add(tf);
```

```
button1 = new JButton("Delete Donor");
button1.setFont(font2);
button1.setBounds(340, 280, 160, 40);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        dispose();

        if ( tf.getText().isEmpty() )
        {
            JOptionPane.showMessageDialog(null, "Please enter all field");
        }
        else
        {
            int x = 0;
            volunteerInterface ob = new volunteerInterface();
            try {
                x = ob.delete_donor(tf.getText());
             } catch (IOException ex) {
                Logger.getLogger(GUI13.class.getName()).log(Level.SEVERE, null, ex);
            }
            if ( x == 1 )
            {
                JOptionPane.showMessageDialog(null, "Donor deleted successfully!");
```

```
                    //System.exit(0);
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Sorry!\nUser not found.");
                }
            }
        }
    });

    button2 = new JButton("Back");
    button2.setFont(font2);
    button2.setBounds(80, 280, 160, 40);
    con.add(button2);

    button2.addActionListener((ActionListener) new ActionListener(){

        @Override
        public void actionPerformed(ActionEvent e)
        {
            dispose();

            GUI8 gui8 = new GUI8();
            gui8.VolunteerGUI();
        }
     });
}
```

```
    public void DeleteDonorGUI()
    {
        GUI13 frame = new GUI13();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class GUI14 extends JFrame
{
    private ImageIcon icon;
    private Container con;
    private JLabel label1, label2;
    private JComboBox comb;
    private Font font1, font2;
    private JButton button1, button2;
    private final String[] str1 = {"A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-"};

    GUI14()
    {
        initComponents();
    }

    public void initComponents()
```

```
{
 icon = new ImageIcon(getClass().getResource("icon.png"));
 this.setIconImage(icon.getImage());

 con = this.getContentPane();
 con.setLayout(null);
 con.setBackground(Color.red);

 font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
 font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

 label1 = new JLabel("Show Donor List by Blood Group");
 label1.setBounds(100, 70, 400, 40);
 label1.setFont(font1);
 con.add(label1);

 label2 = new JLabel("Blood Group :");
 label2.setBounds(140, 160, 150, 40);
 label2.setFont(font2);
 con.add(label2);

 comb = new JComboBox(str1);
 comb.setBounds(320, 160, 120, 40);
 comb.setFont(font2);
 con.add(comb);

 button1 = new JButton("Search");
 button1.setFont(font2);
```

```
button1.setBounds(340, 240, 100, 40);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

   @Override
   public void actionPerformed(ActionEvent e)
   {
     dispose();

     if ( comb.getSelectedItem().toString().isEmpty() )
     {
        JOptionPane.showMessageDialog(null, "Please enter all field");
     }
     else
     {
        volunteerInterface ob = new volunteerInterface();
        try {
           ob.show_donorlist(comb.getSelectedItem().toString());
        } catch (IOException ex) {
           Logger.getLogger(GUI14.class.getName()).log(Level.SEVERE, null, ex);
        }
     }
   }
});

button2 = new JButton("Back");
button2.setFont(font2);
```

```
        button2.setBounds(140, 240, 100, 40);
        con.add(button2);

        button2.addActionListener((ActionListener) new ActionListener(){

            @Override
            public void actionPerformed(ActionEvent e)
            {
                dispose();

                GUI8 gui8 = new GUI8();
                gui8.VolunteerGUI();
            }
         });
    }

    public void ShowDonorlistBloodGroupGUI()
    {
        GUI14 frame = new GUI14();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}
```

```
class GUI15 extends JFrame
{
   private ImageIcon icon;
   private Container con;
   private JLabel label1, label2, label3;
   private JTextField tf;
   private Font font1, font2;
   private JButton button1, button2;

   GUI15()
   {
      initComponents();
   }

   public void initComponents()
   {
      icon = new ImageIcon(getClass().getResource("icon.png"));
      this.setIconImage(icon.getImage());

      con = this.getContentPane();
      con.setLayout(null);
      con.setBackground(Color.red);

      font1 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 28);
      font2 = new Font("Times New Roman", Font.ITALIC+Font.BOLD, 18);

      label1 = new JLabel("Show Donor List by Location");
```

```
label1.setBounds(120, 70, 400, 40);
label1.setFont(font1);
con.add(label1);

label2 = new JLabel("Location :");
label2.setBounds(120, 160, 100, 40);
label2.setFont(font2);
con.add(label2);

tf = new JTextField();
tf.setBounds(260, 160, 200, 40);
tf.setFont(font2);
con.add(tf);

button1 = new JButton("Search");
button1.setFont(font2);
button1.setBounds(340, 240, 100, 40);
con.add(button1);

button1.addActionListener((ActionListener) new ActionListener(){

  @Override
  public void actionPerformed(ActionEvent e)
  {
    dispose();

    if ( tf.getText().isEmpty() )
    {
            JOptionPane.showMessageDialog(null, "Please enter all field");
        }
        else
        {
            volunteerInterface ob = new volunteerInterface();
            try {
                ob.show_donorlist_location(tf.getText());
            } catch (IOException ex) {
                Logger.getLogger(GUI14.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
});

button2 = new JButton("Back");
button2.setFont(font2);
button2.setBounds(140, 240, 100, 40);
con.add(button2);

button2.addActionListener((ActionListener) new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent e)
    {
        dispose();

        GUI8 gui8 = new GUI8();
        gui8.VolunteerGUI();
```

```
            }
        });
    }

    public void ShowDonorlistLocationGUI()
    {
        GUI15 frame = new GUI15();
        frame.setBounds(350, 150, 600, 450);
        frame.setTitle("Blood Bank Management System");

        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}

class demo
{
    static int n ;
    static adminInterface adminlist2[] = new adminInterface[1000];
}



public class Blood_Bank_Management {

    /**
```

```
     * @param args the command line arguments
     * @throws java.io.IOException
     */
    public static void main(String[] args) throws IOException {
        // TODO code application logic here
        try {
            fileHandle ob = new fileHandle();
            demo.n = ob.readIntegerFromFile("Value_of_n");
            /*
             demo.adminlist2[demo.n] = new adminInterface();
             demo.adminlist2[demo.n].set_value_user("Sabik Rahat",
"sabikrahat72428@gmail.com", "01647629698", "Merul Badda");
             demo.adminlist2[demo.n].set_value_admin("72428", "O+", "Yes");
             demo.adminlist2[demo.n].set_value_admin_id("2019-1-60-256");
             demo.adminlist2[demo.n].set_value_donor_visibility("No");

             String FileName = demo.adminlist2[demo.n].get_id();
             String FileLocation = demo.MainPath + FileName;
             ob.writeInFile(demo.MainPath + "AllFileLocation", FileLocation);
             ob.writeObjectInCleanFile(FileLocation, demo.adminlist2[demo.n]);

             demo.n++;
             ob.writeIntegerIncleanFile(demo.MainPath + "Value_of_n", demo.n);
             System.out.println(demo.n);
                    */
        //Owner Information finished......
        }
        catch (FileNotFoundException e)
        {
            System.out.println("Exception : "+e);
            System.out.println("Set your file path");

        }

        GUI1 gui1 = new GUI1();
        gui1.HomepageGUI();

    }

}
```