**JSPM’s**

**Jayawantrao Sawant College of Engineering**

Department of Information Technology

Academic Year: 2019-2020



Project Report on

**BLOOD BANK MANAGEMENT SYSTEM**

* **: GROUP MEMBERS: -**

|  |  |  |
| --- | --- | --- |
| **Sr. no** | **Name** | **Seat no** |
| 1. | Omkar Somnath Tikar | 40 |
| 2. | Saiteja Ramesh Murari | 38 |
| 3. | Sumedh Sonba Sable | 50 |
| 4. | Kalpesh Subash Patil | 42 |

**INDEX**

| **Chapter No.** | **Name of Topic** | **Page no.** |
| --- | --- | --- |
| 1. | Abstract | 3 |
| 2. | Introduction | 4 |
| 3. | Project Specification | 5 |
|  | 3.1 Problem statement | 6 |
|  | 3.2 Scope | 6 |
|  | 3.3 Requirement | 6 |
|  | 3.3.1 Hardware Requirement | 6 |
|  | 3.3.2 Software Requirement | 6 |
|  | 3.4 Assumptions | 6 |
| 4. | E-R model | 7 |
| 5. | Relational Model with Normalization | 8 |
| 6. | Data dictionary | 9 |
| 7. | Graphical user interface (GUI) | 11 |
| 8. | Source code | 16 |
| 9. | Conclusion | 33 |

1. **ABSTRACT**

Blood Bank Management System service focuses on the Donor details and the and record keeping of blood as well as bills. This application includes different modules such as, donor and recipient details, billing, record keeping blood, blood type and their transactions, purchase orders. This application will work on any windows-based operating system higher than Windows XP.

This application will be a Java based application developed using NetBeans IDE 8.0.2 on Windows 7. This application requires minimum hardware configuration and hence can be easily implemented on Dual core processor with minimum 1 GB memory. In case of details of donor and recipient and collection blood, the backend of this application will be MySQL. This database will contain 4 main tables and 1 extra tables for collection blood.

**2. INTRODUCTION**

The main aim of this project Blood Bank Management Systems to maintaining all the information pertaining to blood donors. different blood groups available in each blood bank and help them manage in a best way. This System provides Transparency in this field, make the process of obtaining blood from a blood bank corruption free and make the system of blood bank management collective.

Blood Bank Management System can help to collect blood from many donors from various sources and distribute that blood to needy people who require blood This project can manage List of Donors who eligible for donation on a particular date with contact Number. This system is blood bank management system that helps in managing various blood bank operations effectively. The project consists of a central repository containing various blood groups deposit available along with related details These details include blood type, storage area and date of storage These details help in maintaining the blood deposits.

The project is an online system that allows to check weather required blood deposits of a particular group are available in the blood bank This system is developed on JAVA platform and supported by MySQL database to store blood and user specific details. Blood Bank Management System provides the Donor information system. Acceptor information system. Done information system deals with the organization of the voluntary data.

**Modules Description:**

* **User Management: -**In this module owner can login for insert data of donor or recipient details.
* **Payment: -**Customer can pay their price through cash.
* **Feedback: -**Customer can give feedback.
* **Report Generate: -**Generate different types of reports.

**3. PROJECT SPECIFICATION**

**3.1 PROBLEM STATEMENT**

To develop a Blood Bank Management System is to maintaining

all the information pertaining to blood donors and recipient.

**3.2 SCOPE**

The main use of this project is as follows:

1) Add and maintain records of donor and recipient.

2) Add and maintain customer details.

3) Add and maintain details of new Donor and Recipient.

4) Add and maintain the enters of Blood.

5) Provides financial and economic report of the origination.

6) Make an easy to use environment for users and customers.

**3.3 REQUIREMENTS:**

**Software Requirements (Minimum):**

* + - **Operating System: -** NetBeans IDE,Windows 7
    - **Back-end Tool: -** MySQL
    - **Front End**: - Java

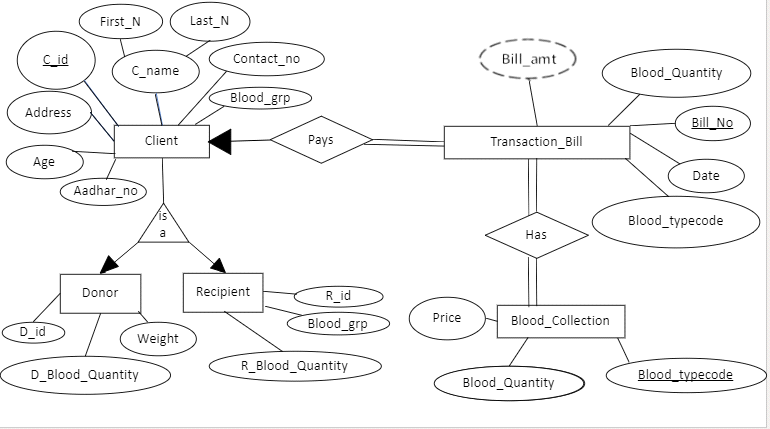
**Hardware Requirement (Minimum):**

* + - **Processor: -** 2.0 GHZ
    - **RAM: -** 1 GB
    - **Hard disk: -** 20 GB free space

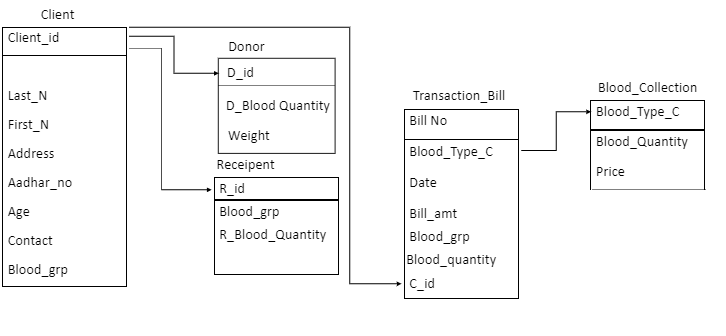
**3.4 ASSUMPTIONS:**

* Contact no is considered as a single value.
* While billing a client can choose the blood type for blood transfusion.
* On performing delete operation the primary key value must not be present as foreign key value in associated table.

**4. E-R MODEL**



**5. RELATIONAL MODEL WITH NORMALIZATION**

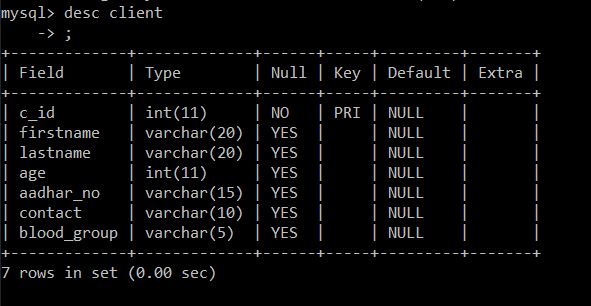
**The Normal forms are:**  
  
>1 NF (First normal form):  
Only one value is associated with each attribute i.e. The value in  
the domain of each attribute of the relation are atomic.

>2NF (Second normal form):  
A table is said to be in 2NF if both the following conditions hold:  
Table is in 1NF  
No non-prime attribute is dependent on the proper subset  
of any candidate key of table i.e. partial dependency  
should not be present.

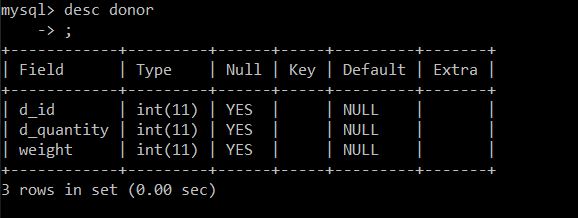
>3NF (Third normal form):  
A table design is said to be in 3NF if both the following conditions  
hold:  
Table must be in 2NF  
Transitive functional dependency of non-prime  
attribute on any super key should be removed.  
The tables package, channel and bill are in 1NF, 2NF, 3NF  
as they are atomic, contain no partial or transitive dependency.

**6. DATA DICTIONARY**

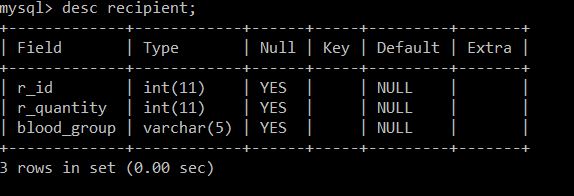
* **Client:**



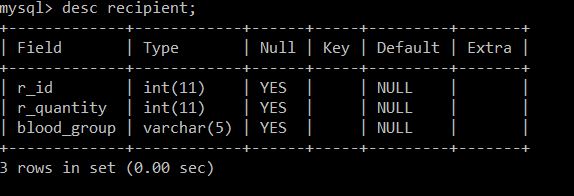
* **Donor:**



* **Recipient:**

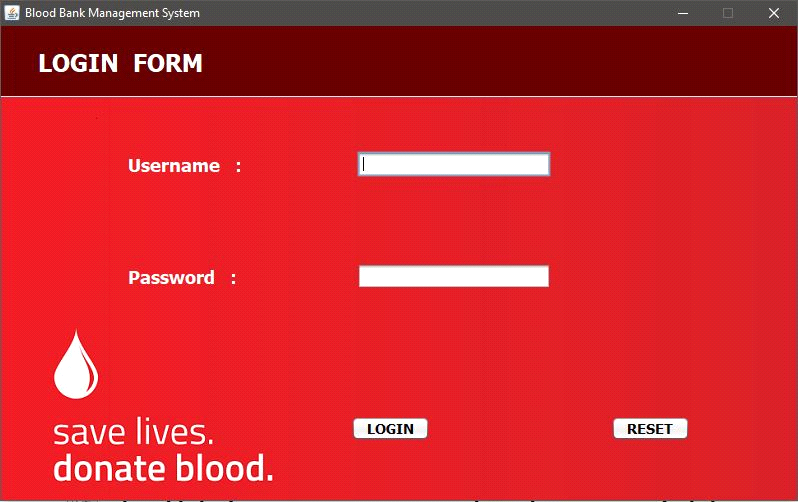


* **Transaction Bill:**

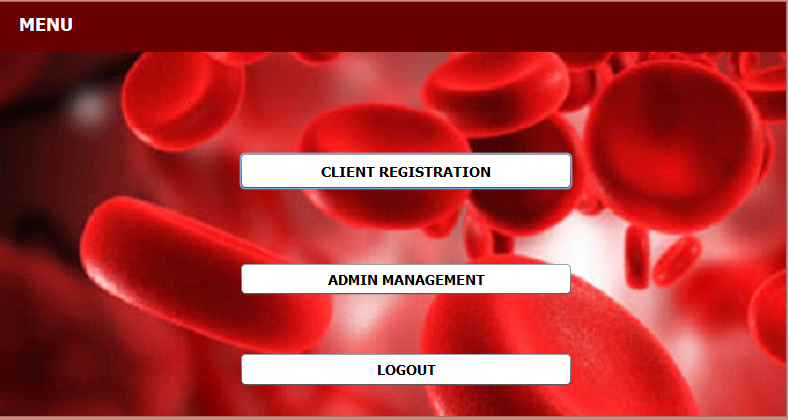


**7. GRAPHIC USER INTERFACE**

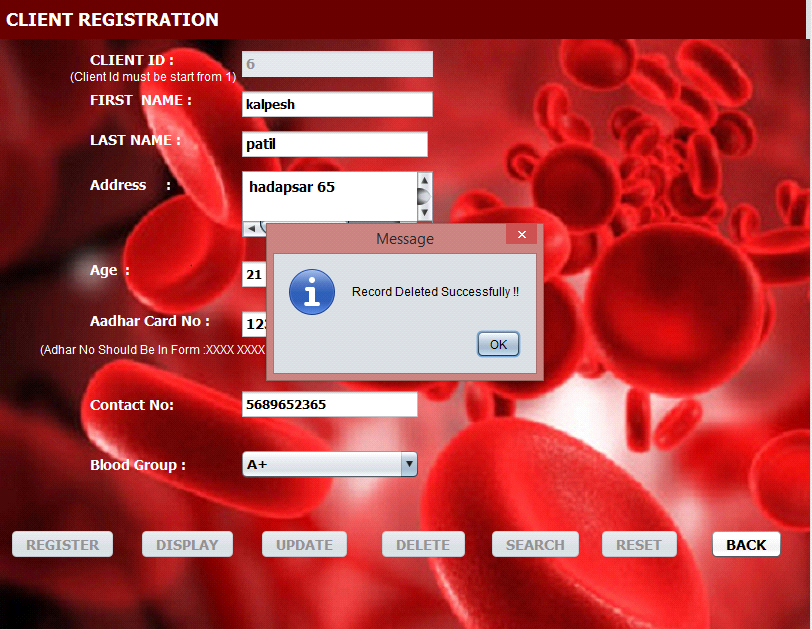
1. **Login form:**



1. **Menu form:**

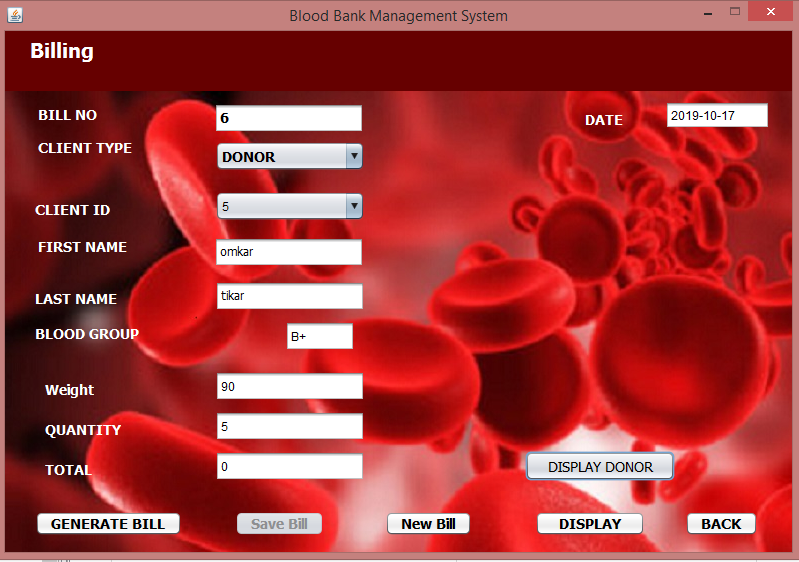


1. **Customer form:**

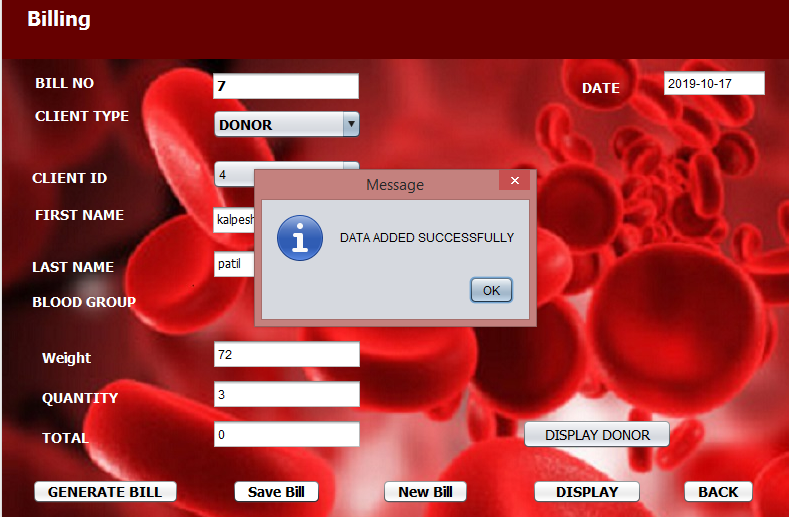


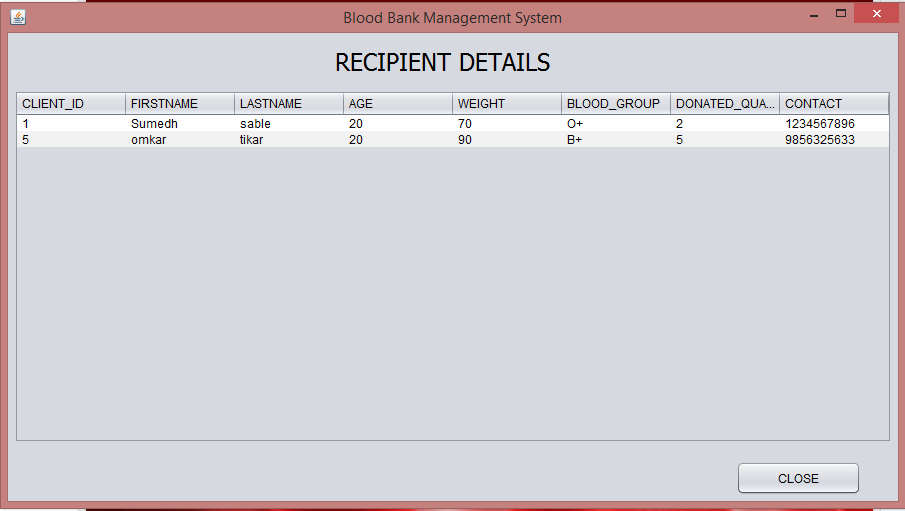
1. **Bill form:**

* **Receiver:**



* **Donor:**



* **Display Donor**

**8. SOURCE CODE**

**package loginandregister;**

**import java.awt.Dimension;**

**import java.sql.\*;**

**import java.awt.Toolkit;**

**import java.awt.event.WindowEvent;**

**import javax.swing.JFrame;**

**import java.util.regex.\*;**

**import javax.swing.JOptionPane;**

**/\***

**\* 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.**

**\*/**

**/\*\***

**\***

**\* @author Rucha**

**\*/**

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

**PreparedStatement addChPS, fetChPS, searchChPS, modifyChPS, deleteChPS,searchChPS1;**

**Connection con;**

**ResultSet fetChRes, searchChRes,searchChRes1;**

**public bldstorage() {**

**/\* try {**

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

**con = DriverManager.getConnection("jdbc:derby://localhost:3306/blood\_bank", "omkar", "12345");**

**} catch (Exception ex)**

**{**

**JOptionPane.showMessageDialog(null, ex);**

**}\*/**

**initComponents();**

**//this.setAlwaysOnTop(false);**

**this.setSize(500,450);**

**setExtendedState(JFrame.MAXIMIZED\_HORIZ);**

**setVisible(true);**

**setTitle("Blood Bank Management System");**

**Toolkit toolkit=getToolkit();**

**Dimension size=toolkit.getScreenSize();**

**setLocation(size.width/2 - getWidth()/2,size.height/2 - getHeight()/2);**

**}**

**public void close()**

**{**

**WindowEvent winClosingEvent=new WindowEvent(this,WindowEvent.WINDOW\_CLOSING);**

**Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);**

**}**

**/\*\***

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

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

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

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

**private void initComponents() {**

**jPanel1 = new javax.swing.JPanel();**

**jLabel5 = new javax.swing.JLabel();**

**jPanel2 = new javax.swing.JPanel();**

**jLabel2 = new javax.swing.JLabel();**

**jLabel3 = new javax.swing.JLabel();**

**jLabel4 = new javax.swing.JLabel();**

**quantity = new javax.swing.JTextField();**

**charges = new javax.swing.JTextField();**

**bldtypcode = new javax.swing.JComboBox<>();**

**back = new javax.swing.JButton();**

**reset = new javax.swing.JButton();**

**jLabel1 = new javax.swing.JLabel();**

**jLabel6 = new javax.swing.JLabel();**

**setDefaultCloseOperation(0);**

**setAlwaysOnTop(true);**

**setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));**

**setMinimumSize(new java.awt.Dimension(793, 443));**

**setSize(new java.awt.Dimension(793, 443));**

**getContentPane().setLayout(null);**

**jPanel1.setBackground(new java.awt.Color(102, 0, 0));**

**jLabel5.setBackground(new java.awt.Color(51, 51, 0));**

**jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N**

**jLabel5.setForeground(new java.awt.Color(255, 255, 255));**

**jLabel5.setText(" BLOOD STORAGE DETAILS");**

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

**jPanel1.setLayout(jPanel1Layout);**

**jPanel1Layout.setHorizontalGroup(**

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

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

**.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 384, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addContainerGap(416, Short.MAX\_VALUE))**

**);**

**jPanel1Layout.setVerticalGroup(**

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

**.addGroup(jPanel1Layout.createSequentialGroup()**

**.addContainerGap()**

**.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addContainerGap(19, Short.MAX\_VALUE))**

**);**

**getContentPane().add(jPanel1);**

**jPanel1.setBounds(0, 0, 800, 67);**

**jPanel2.setBackground(new java.awt.Color(202, 242, 255));**

**jPanel2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N**

**jPanel2.setLayout(null);**

**jLabel2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N**

**jLabel2.setForeground(new java.awt.Color(255, 255, 255));**

**jLabel2.setText("Blood Type Code");**

**jPanel2.add(jLabel2);**

**jLabel2.setBounds(79, 42, 127, 24);**

**jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N**

**jLabel3.setForeground(new java.awt.Color(255, 255, 255));**

**jLabel3.setText("Quantity");**

**jPanel2.add(jLabel3);**

**jLabel3.setBounds(79, 128, 127, 23);**

**jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N**

**jLabel4.setForeground(new java.awt.Color(255, 255, 255));**

**jLabel4.setText("Charges");**

**jPanel2.add(jLabel4);**

**jLabel4.setBounds(79, 209, 127, 22);**

**quantity.setEditable(false);**

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

**quantity.addActionListener(new java.awt.event.ActionListener() {**

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

**quantityActionPerformed(evt);**

**}**

**});**

**jPanel2.add(quantity);**

**quantity.setBounds(316, 128, 128, 23);**

**charges.setEditable(false);**

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

**jPanel2.add(charges);**

**charges.setBounds(316, 209, 128, 23);**

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

**bldtypcode.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "---- SELECT---", "A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-" }));**

**bldtypcode.addFocusListener(new java.awt.event.FocusAdapter() {**

**public void focusLost(java.awt.event.FocusEvent evt) {**

**bldtypcodeFocusLost(evt);**

**}**

**});**

**bldtypcode.addActionListener(new java.awt.event.ActionListener() {**

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

**bldtypcodeActionPerformed(evt);**

**}**

**});**

**jPanel2.add(bldtypcode);**

**bldtypcode.setBounds(316, 42, 128, 24);**

**back.setBackground(new java.awt.Color(204, 204, 204));**

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

**back.setText("Back");**

**back.addActionListener(new java.awt.event.ActionListener() {**

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

**backActionPerformed(evt);**

**}**

**});**

**jPanel2.add(back);**

**back.setBounds(457, 305, 95, 25);**

**reset.setBackground(new java.awt.Color(204, 204, 204));**

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

**reset.setText("RESET");**

**reset.addActionListener(new java.awt.event.ActionListener() {**

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

**resetActionPerformed(evt);**

**}**

**});**

**jPanel2.add(reset);**

**reset.setBounds(168, 305, 100, 25);**

**jLabel1.setForeground(new java.awt.Color(255, 255, 255));**

**jLabel1.setText("(per bag)");**

**jPanel2.add(jLabel1);**

**jLabel1.setBounds(454, 215, 55, 14);**

**jLabel6.setIcon(new javax.swing.ImageIcon(getClass().getResource("/loginandregister/140508100350\_1\_540x360.jpg"))); // NOI18N**

**jLabel6.setText("jLabel6");**

**jPanel2.add(jLabel6);**

**jLabel6.setBounds(0, -10, 800, 440);**

**getContentPane().add(jPanel2);**

**jPanel2.setBounds(0, 60, 800, 430);**

**pack();**

**setLocationRelativeTo(null);**

**}// </editor-fold>**

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

**// TODO add your handling code here:**

**quantity.setText("");**

**charges.setText("");**

**bldtypcode.setSelectedIndex(0);**

**}**

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

**// TODO add your handling code here:**

**close();**

**this.dispose();**

**adminmanagement adm =new adminmanagement();**

**adm.setVisible(true);**

**}**

**private void bldtypcodeFocusLost(java.awt.event.FocusEvent evt) {**

**// TODO add your handling code here:**

**}**

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

**// TODO add your handling code here:**

**}**

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

**// TODO add your handling code here:**

**try {**

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

**con = DriverManager.getConnection("jdbc:derby://localhost:1527/blood\_bank", "omkar", "12345");**

**} catch (Exception ex)**

**{**

**JOptionPane.showMessageDialog(null, ex);**

**}**

**if(bldtypcode.getSelectedItem()=="A+")**

**{**

**//charges.setText("100");**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='A+'";**

**String query1="select charges from blood\_collection where blood\_typecode='A+'";**

**searchChPS=con.prepareStatement(query);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}**

**else if(bldtypcode.getSelectedItem()=="A-")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='A-'";**

**String query1="select charges from blood\_collection where blood\_typecode='A-'";**

**searchChPS=con.prepareStatement(query);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}**

**else if(bldtypcode.getSelectedItem()=="B+")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='B+'";**

**String query1="select charges from blood\_collection where blood\_typecode='B+'";**

**searchChPS=con.prepareStatement(query);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}**

**else if(bldtypcode.getSelectedItem()=="B-")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='B-'";**

**String query1="select charges from blood\_collection where blood\_typecode='B-'";**

**searchChPS=con.prepareStatement(query);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}else if(bldtypcode.getSelectedItem()=="AB+")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='AB+'";**

**String query1="select charges from blood\_collection where blood\_typecode='AB+'";**

**searchChPS=con.prepareStatement(query);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}else if(bldtypcode.getSelectedItem()=="AB-")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='AB-'";**

**String query1="select charges from blood\_collection where blood\_typecode='AB-'";**

**searchChPS=con.prepareStatement(query);**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}**

**else if(bldtypcode.getSelectedItem()=="O+")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='O+'";**

**String query1="select charges from blood\_collection where blood\_typecode='O+'";**

**searchChPS=con.prepareStatement(query);**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}else if(bldtypcode.getSelectedItem()=="O-")**

**{**

**try{**

**Statement sta=con.createStatement();**

**String query="select quantity from blood\_collection where blood\_typecode='O-'";**

**String query1="select charges from blood\_collection where blood\_typecode='O-'";**

**searchChPS=con.prepareStatement(query);**

**searchChRes=searchChPS.executeQuery();**

**if(searchChRes.next())**

**{**

**quantity.setText(searchChRes.getString("quantity"));**

**}**

**searchChPS1=con.prepareStatement(query1);**

**//searchChPS.setInt(1,Integer.parseInt();**

**searchChRes1=searchChPS1.executeQuery();**

**if(searchChRes1.next())**

**{**

**charges.setText(searchChRes1.getString("charges"));**

**}**

**}catch(Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex);**

**}**

**}**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**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 (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(bldstorage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(bldstorage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(bldstorage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(bldstorage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

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

**public void run() {**

**new bldstorage().setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JButton back;**

**private javax.swing.JComboBox<String> bldtypcode;**

**private javax.swing.JTextField charges;**

**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.JPanel jPanel1;**

**private javax.swing.JPanel jPanel2;**

**private javax.swing.JTextField quantity;**

**private javax.swing.JButton reset;**

**// End of variables declaration**

**9. CONCLUSION**

This was an effort to develop a blood bank database management system that can be used to effectively get client details and information about available blood groups. As the number of people who are in need of blood increasing in large number day by day this system aims to make blood bank database management much simpler and more efficient.