

BLOOD BANKING SYSTEM



An

Object-Oriented Programming through Java Course Project Report

in partial fulfilment of the degree

Bachelor of Technology
in
Computer Science & Engineering

By

Shiva Krishna Beeraboina

2103A51390

Bhuvan Sai Rebba

2103A51288

Submitted to

Mr. N. Mahender

Assistant Professor





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **Object-Oriented Programming through Java - Course Project** Report entitled “**BLOOD BANKING SYSTEM (Reforming Blood Donation and Emergency Blood Requests)**” is a record of bonafide work carried out by the student(s) **Shiva Krishna Beeraboina, Bhuvan Sai Rebba** bearing Roll No(s) **2103A51390, 2103A51288** during the academic year 2023-24 in partial fulfillment of the award of the degree of *Bachelor of Technology* in **Computer Science & Engineering** by the SR University, Ananthasagar, Warangal.

Lab In-charge

Head of the Department

TABLE OF CONTENTS

S. No	Title	Page No
1.	Acknowledgement	4
2.	Abstract	5
3.	Objectives	6
4.	Key Definitions	7 – 9
5.	Design	10 – 12
6.	Implementation	13 - 39
7.	Result Screens	40 - 45
8.	Conclusion	46

1. ACKNOWLEDGEMENT

We express our thanks to our lecturer **Mr. N. Mahender, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M. Sheshikala**, mam for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, sir for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

2. ABSTRACT

Blood, the life-sustaining fluid that passes through our veins, plays an irreplaceable role in maintaining human health and liveliness. Every day, many individuals face medical emergencies, surgeries, accidents and chronic illnesses that needs blood transfusions. In these critical situations, the availability of safe and compatible blood matters life and death. The urgent need for blood donation, a selfless act directly impacts the well-being of both donors and recipients.

Despite of its importance the availability of safe and sufficient blood remains as a challenge in many parts of world due to misconceptions, fears, and reachability. The **“Blood Banking System – Reforming Blood Donation and Emergency Blood Requests”** presents a comprehensive digital solution designed to reshape the blood donation and emergency blood requests. Integrated with advance technology with healthcare needs, this system offers a user-centric functionalities. With this user-friendly system, people wanting to donate blood can easily sign up, provide important health information, and verify their blood test results with integrated security to maintain user data privacy. An additional reward system encourages regular donors to make positive impact in the community that can increase social awareness.

When an emergency strikes, this system can help. People who need blood can quickly find potential donors with matching blood types in app. It also has integrated real time location tracking of blood banks and users ensuring that help is always within reach. This streamlined approach ensures timely access to life-saving resources during emergencies.

This systematic approach to blood banking not only revolutionizes donation but also encourages the culture of taking and giving back by bridging the gap between donors and recipients. By providing an efficient and user-friendly interface, this project builds the way for a future where the act of donating blood and saving lives is made more accessible, engaging and impactful.

3. OBJECTIVES

- We firmly believe that our mission to enhance blood donation practices can thrive with unwavering passion and persistent commitment. Our goal is to provide a free and accessible platform where blood donors and recipients can connect seamlessly, fostering a community dedicated to saving lives.
- We aspire to offer a 24/7 service, tirelessly working to enhance the Blood Banking System by adding new functionalities and features. Our aim is to continually improve user experience by incorporating the latest technologies and best practices in blood donation and emergency response.
- Key Objectives of our project includes:
 - i. Developing a User-Friendly Interface**
 - Creating a responsive and user-friendly interface for donors and recipients for easy access of functionalities and navigation.
 - ii. Implementing User Registration and Authentication System**
 - Implementing secure user registration and authentication via email and mobile number permitting users to use the application.
 - iii. Collecting and Storing Donors and Blood Banks Information**
 - Designing and implementing a database schema for storing donors, recipients and blood banks information and performing constant updates on the data.
 - Collecting personal and health related fields from users and storing it in database that can be used in critical situations.
 - iv. Developing a Search Functionality and Integrating with Location Tracking**
 - Creating a search feature for recipients to find potential donors based on blood group and location.
 - Integrating with mapping services to find real-time locations of nearby blood bank centers and donors.

4. KEY DEFINITIONS

a. Programming Languages

- a) **Java** is a versatile, object-oriented, high-level programming language that is widely used for building software applications. It is known for its platform independence, security features, and extensive libraries, making it an excellent choice for developing a variety of applications, including web and desktop applications.



According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are:

- a. Desktop Applications such as acrobat reader, media player, antivirus, etc.
 - b. Web Applications such as irctc.co.in, javatpoint.com, etc.
 - c. Enterprise Applications such as banking applications.
 - d. Mobile
 - e. Embedded System
 - f. Smart Card
 - g. Robotics
 - h. Games, etc.
- b) **MYSQL** is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database.



b. Frameworks and Libraries

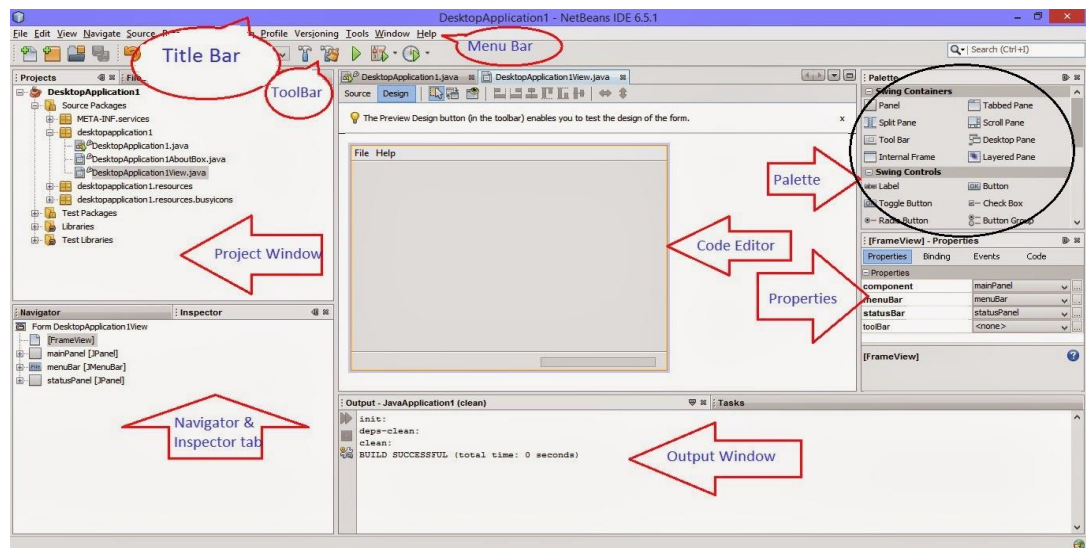
A. Java Swings: Java Swing serves as a versatile toolkit for constructing graphical user interfaces (GUIs) in Java, enabling the development of platform-independent desktop applications. Popular for its high-level nature, robustness, and commitment to object-oriented principles, Swing has become a go-to choice for Java developers. Unlike its predecessor, AWT, Swing components are lightweight, ensuring consistent behavior across various operating systems. The framework adheres to the Model-View-Controller (MVC) architecture, fostering modular and maintainable code by separating application logic, user interface, and user input.

B. Swing AWT Components: Abstract Window Toolkit (AWT) is a foundational component of Java for building graphical user interfaces (GUIs) in desktop applications. It provides a set of heavyweight components that rely on the underlying operating system's native GUI elements, allowing for a seamless integration with the platform's look and feel. AWT includes essential UI components such as buttons, text fields, and windows, enabling developers to create basic GUIs for their Java applications. Swing, AWT components are considered heavyweight, potentially leading to variations in appearance across different operating systems.

C. Java SQL(JDBC): Java SQL, part of the Java Standard Edition (Java SE), provides a comprehensive set of APIs for database connectivity and interaction. It enables Java applications to seamlessly connect to relational databases, execute SQL queries, and manage database transactions. Key components of Java SQL include the JDBC (Java Database Connectivity) API, which serves as a standardized interface for database access, allowing developers to interact with various databases using a uniform set of methods. JDBC supports both simple and complex database operations, including querying, updating, and transaction management.

c. Development Environment

A. IDE:



Apache NetBeans, an open-source integrated development environment (IDE), is a powerful tool for Java application development. Offering a rich set of features and tools, NetBeans simplifies the development process, allowing programmers to create, debug, and deploy Java applications with efficiency. The IDE supports various Java technologies, including JavaFX and Swing, and provides a user-friendly interface for designing graphical user interfaces (GUIs). NetBeans boasts seamless integration with Apache Maven for project management and dependency handling. Its built-in editor, code analyzers, and version control system support enhance productivity, while features like Smart Javadoc and instant rename aid in code maintenance. With support for multiple languages and platforms, Apache NetBeans is a versatile and extensible IDE, fostering a conducive environment for developers to build robust and scalable Java applications.

B. SYSTEM SOFTWARE SPECIFICATIONS

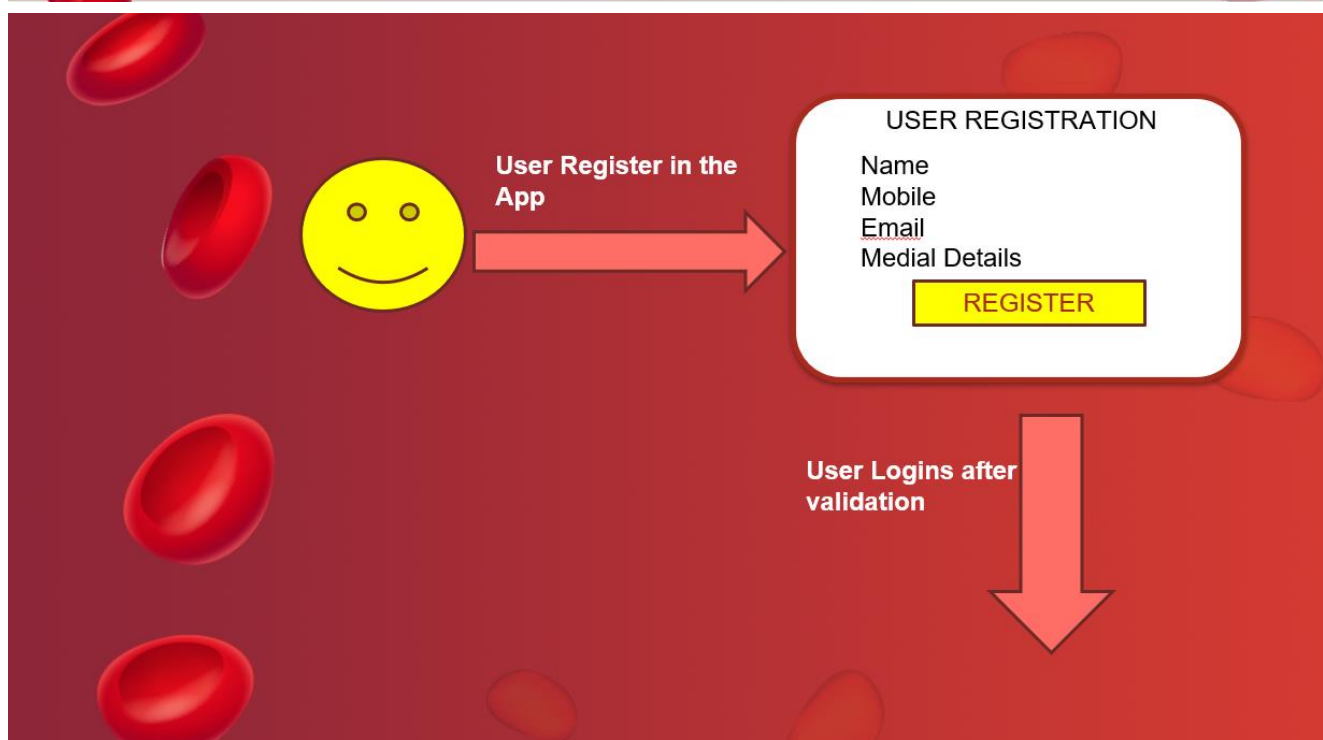
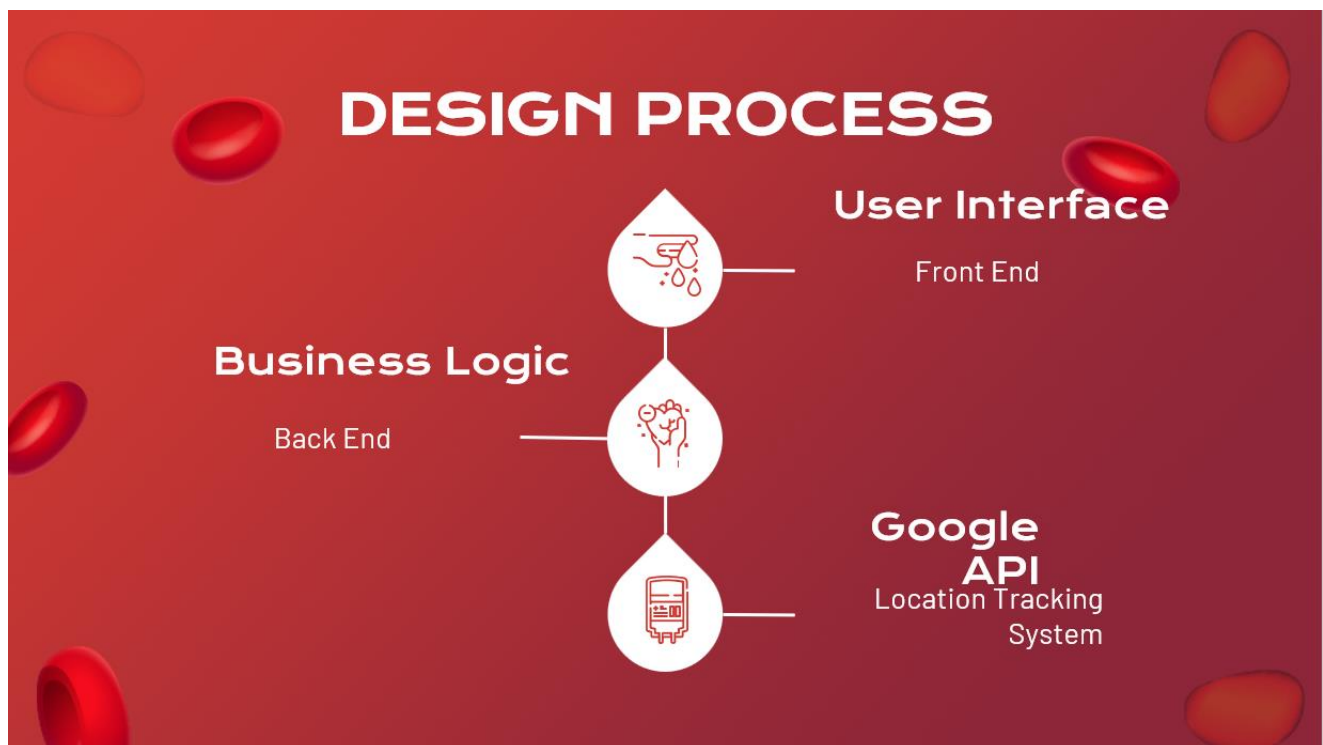
Laptop model: Dell Inspiron 15 3000

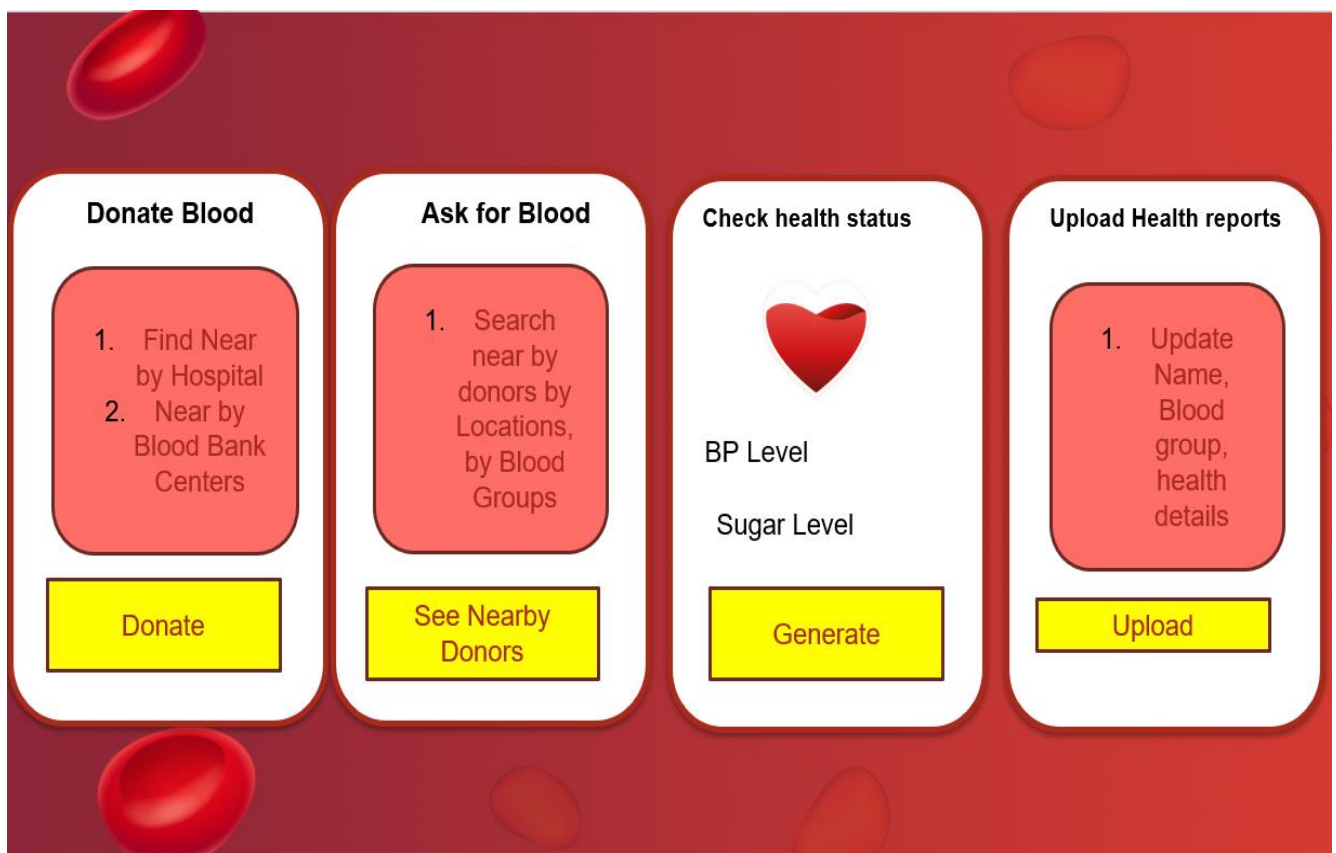
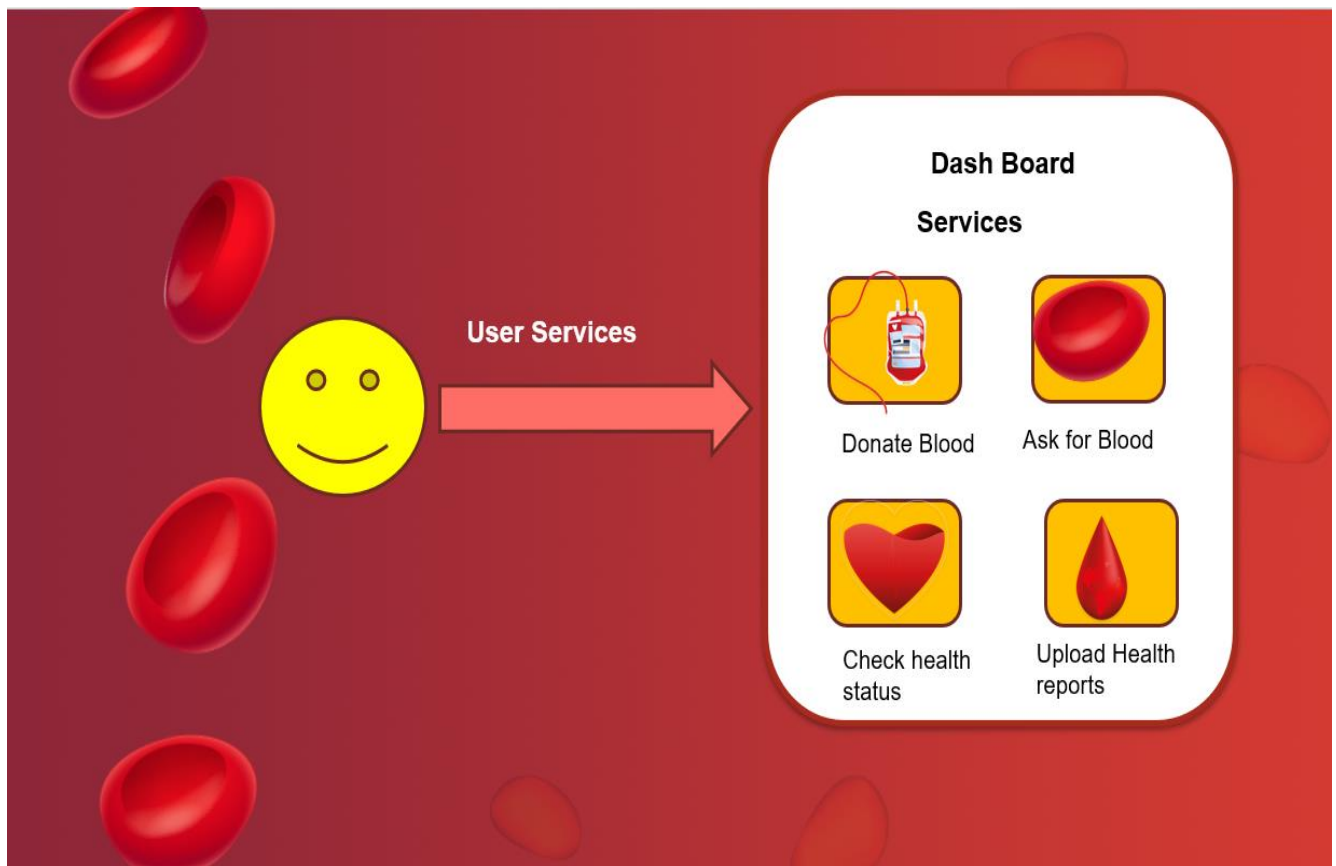
Operating System: Windows 11

Processor: intel i5

5. DESIGN

Architectural Design





DATABASE SCHEMAS

```
mysql> use bloodbanksystem
Database changed
mysql> desc register
-> ;
```

Field	Type	Null	Key	Default	Extra
name	varchar(100)	NO		NULL	
email	varchar(200)	NO	PRI	NULL	
mobile	varchar(12)	NO	PRI	NULL	
password	varchar(100)	NO	PRI	NULL	

4 rows in set (0.05 sec)

```
mysql> desc donors;
```

Field	Type	Null	Key	Default	Extra
donorId	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
age	int	YES		NULL	
fatherName	varchar(100)	YES		NULL	
motherName	varchar(100)	YES		NULL	
streetAddress	varchar(200)	YES		NULL	
city	varchar(100)	YES		NULL	
district	varchar(100)	YES		NULL	
pinCode	varchar(10)	YES		NULL	
bloodGroup	varchar(6)	YES		NULL	
mobile	varchar(12)	YES	UNI	NULL	
gender	varchar(10)	YES		NULL	

12 rows in set (0.01 sec)

```
mysql> desc bloodbanks;
```

Field	Type	Null	Key	Default	Extra
bankId	int	NO	PRI	NULL	
bankName	varchar(100)	YES		NULL	
bankAddress	varchar(200)	YES		NULL	
city	varchar(100)	YES		NULL	
district	varchar(100)	YES		NULL	
state	varchar(100)	YES		NULL	
email	varchar(200)	NO	PRI	NULL	
mobile	varchar(12)	YES	UNI	NULL	
license	varchar(100)	NO	PRI	NULL	
validity	varchar(100)	YES		NULL	
testFacilities	varchar(200)	YES		NULL	
storageUnit	varchar(200)	YES		NULL	

12 rows in set (0.00 sec)

6. IMPLEMENTATION

4.1. Code

HOME PAGE

```
package bloodbank;

import javax.swing.JOptionPane;

/**
 * @author SHIVA KRISHNA
 */

public class Home extends javax.swing.JFrame {

    public Home() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        MainList = new javax.swing.JMenuBar();
        newDonor = new javax.swing.JMenu();
        addDonor = new javax.swing.JMenuItem();
        jSeparator1 = new javax.swing.JPopupMenu.Separator();
        updateDonor = new javax.swing.JMenuItem();
        Search = new javax.swing.JMenu();
        locationFind = new javax.swing.JMenuItem();
        jSeparator2 = new javax.swing.JPopupMenu.Separator();
        bloodGroupFind = new javax.swing.JMenuItem();
        jSeparator3 = new javax.swing.JPopupMenu.Separator();
        listDonors = new javax.swing.JMenuItem();
        Bank = new javax.swing.JMenu();
        addBank = new javax.swing.JMenuItem();
        jSeparator4 = new javax.swing.JPopupMenu.Separator();
        deleteBank = new javax.swing.JMenuItem();
        jSeparator5 = new javax.swing.JPopupMenu.Separator();
        listBanks = new javax.swing.JMenuItem();
        Delete = new javax.swing.JMenu();
        deleteInner = new javax.swing.JMenuItem();
        Exit = new javax.swing.JMenu();
        logout = new javax.swing.JMenuItem();
        jSeparator6 = new javax.swing.JPopupMenu.Separator();
    }
}
```

```

exitInner = new javax.swing.JMenuItem();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(1600, 900));
setUndecorated(true);
getContentPane().setLayout(null);
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/back11.png")));
getContentPane().add(jLabel1);
jLabel1.setBounds(0, 0, 1600, 840);
MainList.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
MainList.setMargin(new java.awt.Insets(0, 0, 10, 0));
MainList.setMaximumSize(new java.awt.Dimension(1920, 60));
MainList.setMinimumSize(new java.awt.Dimension(1920, 60));
MainList.setPreferredSize(new java.awt.Dimension(1920, 60));
newDonor.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Donor.png")));
// NOI18N
newDonor.setText("Donor");
addDonor.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_DOWN_MASK));
addDonor.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Add
new.png"))); // NOI18N
addDonor.setText("Add Me");
addDonor.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addDonorActionPerformed(evt);
    }
});
newDonor.add(addDonor);
newDonor.add(jSeparator1);
updateDonor.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_U,
java.awt.event.InputEvent.CTRL_DOWN_MASK));
updateDonor.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Update
details.png"))); // NOI18N
updateDonor.setText("Update Details");
updateDonor.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateDonorActionPerformed(evt);
    }
});

```

```

    }
});
newDonor.add(updateDonor);
MainList.add(newDonor);
Search.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/search user.png")));
// NOI18N
Search.setText("Search Donor");
locationFind.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.ALT_DOWN_MASK));
locationFind.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/bloodbank/Location.png"))); // NOI18N
locationFind.setText("Search by Location");
locationFind.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        locationFindActionPerformed(evt);
    }
});
Search.add(locationFind);
Search.add(jSeparator2);
bloodGroupFind.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_B,
java.awt.event.InputEvent.ALT_DOWN_MASK));
bloodGroupFind.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Blood
group.png"))); // NOI18N
bloodGroupFind.setText("Search by Blood Group");
bloodGroupFind.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bloodGroupFindActionPerformed(evt);
    }
});
Search.add(bloodGroupFind);
Search.add(jSeparator3);
listDonors.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.ALT_DOWN_MASK));
listDonors.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Details.png")));
// NOI18N
listDonors.setText("List all Blood Donors");
listDonors.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            listDonorsActionPerformed(evt);
        }
    });
    Search.add(listDonors);
    MainList.add(Search);
    Bank.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/stock.png")));
    Bank.setText("Blood Bank");
    addBank.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_N,
java.awt.event.InputEvent.SHIFT_DOWN_MASK));
    addBank.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Inc.png"))); //
NOI18N
    addBank.setText("Add Blood Bank");
    addBank.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            addBankActionPerformed(evt);
        }
    });
    Bank.add(addBank);
    Bank.add(jSeparator4);
    deleteBank.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.SHIFT_DOWN_MASK));
    deleteBank.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Dec.png")));
    deleteBank.setText("Delete Blood Bank");
    deleteBank.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            deleteBankActionPerformed(evt);
        }
    });
    Bank.add(deleteBank);
    Bank.add(jSeparator5);
    listBanks.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_L,
java.awt.event.InputEvent.SHIFT_DOWN_MASK));
    listBanks.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Details.png"))); //
NOI18N
    listBanks.setText("ListAll");
    listBanks.addActionListener(new java.awt.event.ActionListener() {

```



```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            listBanksActionPerformed(evt);
        }
    });
    Bank.add(listBanks);
    MainList.add(Bank);
    Delete.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/delete
donor.png"))); // NOI18N
    Delete.setText("Delete My Info");
    deleteInner.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_DOWN_MASK));
    deleteInner.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/delete.png")));
// NOI18N
    deleteInner.setText("Delete");
    deleteInner.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            deleteInnerActionPerformed(evt);
        }
    });
    Delete.add(deleteInner);
    MainList.add(Delete);

    Exit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/exit.png")));
    Exit.setText("Exit");
    logout.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_E,
java.awt.event.InputEvent.CTRL_DOWN_MASK));
    logout.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Logout.png")));
    logout.setText("LogOut");
    logout.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            logoutActionPerformed(evt);
        }
    });
    Exit.add(logout);
    Exit.add(jSeparator6);
    exitInner.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_W,
java.awt.event.InputEvent.CTRL_DOWN_MASK));

```

```

        exitInner.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Exit
application.png"))); // NOI18N
        exitInner.setText("Exit");
        exitInner.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                exitInnerActionPerformed(evt);
            }
        });
        Exit.add(exitInner);
        MainList.add(Exit);
        setJMenuBar(MainList);
        pack();
    }
    private void addDonorActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new addNewDonor().setVisible(true);
    }
    private void listDonorsActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new AllDonorDetails().setVisible(true);
    }
    private void deleteBankActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new DeleteBloodBank().setVisible(true);
    }
    private void listBanksActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new ListAllBanks().setVisible(true);
    }
    private void deleteInnerActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new DeleteDonor().setVisible(true);
    }
    private void logoutActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

```

```

        int a = JOptionPane.showConfirmDialog(null,"Select Yes by Confirming","Do you Want to
Logout",JOptionPane.YES_NO_OPTION);
        if(a==0)
        {
            new LoginPage().setVisible(true);
        }
    }

    private void exitInnerActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int a = JOptionPane.showConfirmDialog(null,"Select Yes by Confirming","Do you Want to
Exit",JOptionPane.YES_NO_OPTION);
        if(a==0)
        {
            JOptionPane.showMessageDialog(null, "Thanks for Using our App");
            System.exit(0);
        } }

    private void updateDonorActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new UpdateDonor().setVisible(true);
    }

    private void locationFindActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new SeachByLocation().setVisible(true);
    }

    private void bloodGroupFindActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        new SearchByBloodGroup().setVisible(true);
    }

    private void addBankActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new AddBloodBank().setVisible(true);
    }

    public static void main(String args[]) {
        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(Home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Home().setVisible(true);
        }
    });
}

private javax.swing.JMenu Bank;
private javax.swing.JMenu Delete;
private javax.swing.JMenu Exit;
private javax.swing.JMenuBar MainList;
private javax.swing.JMenu Search;
private javax.swing.JMenuItem addBank;
private javax.swing.JMenuItem addDonor;
private javax.swing.JMenuItem bloodGroupFind;
private javax.swing.JMenuItem deleteBank;
private javax.swing.JMenuItem deleteInner;
private javax.swing.JMenuItem exitInner;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JPopupMenu.Separator jSeparator1;
private javax.swing.JPopupMenu.Separator jSeparator2;
private javax.swing.JPopupMenu.Separator jSeparator3;
private javax.swing.JPopupMenu.Separator jSeparator4;
private javax.swing.JPopupMenu.Separator jSeparator5;
private javax.swing.JPopupMenu.Separator jSeparator6;
private javax.swing.JMenuItem listBanks;
private javax.swing.JMenuItem listDonors;
private javax.swing.JMenuItem locationFind;
private javax.swing.JMenuItem logout;
private javax.swing.JMenu newDonor;
private javax.swing.JMenuItem updateDonor;
}

```

ADD DONOR PAGE

```

package bloodbank;
import DataBaseConnections.ConnectionProvider;
import java.sql.*;
import javax.swing.JOptionPane;
/**
 *
 * @author SHIVA KRISHNA
 */
public class addNewDonor extends javax.swing.JFrame {
    public addNewDonor() {
        initComponents();
    }
    private void initComponents() {
        Title = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        donorID = new javax.swing.JLabel();
        idField = new javax.swing.JLabel();
        name = new javax.swing.JLabel();
        fatherName = new javax.swing.JLabel();
        motherName = new javax.swing.JLabel();
        dob = new javax.swing.JLabel();
        hno = new javax.swing.JLabel();
    }
}

```

```

city = new javax.swing.JLabel();
district = new javax.swing.JLabel();
pinCode = new javax.swing.JLabel();
bloodGroup = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jTextField3 = new javax.swing.JTextField();
jTextField4 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();
jTextField6 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jTextField2 = new javax.swing.JTextField();
Mobile = new javax.swing.JLabel();
gender = new javax.swing.JLabel();
jTextField10 = new javax.swing.JTextField();
jComboBox1 = new javax.swing.JComboBox<>();
jComboBox2 = new javax.swing.JComboBox<>();
jSeparator2 = new javax.swing.JSeparator();
save = new javax.swing.JButton();
Exit = new javax.swing.JButton();
Clear = new javax.swing.JButton();
image = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(1600, 900));
setUndecorated(true);
addComponentListener(new java.awt.event.ComponentAdapter() {
    public void componentShown(java.awt.event.ComponentEvent evt) {
        formComponentShown(evt);
    }
});
getContentPane().setLayout(null);
Title.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
Title.setText("Add NEW Donor");
getContentPane().add(Title);
Title.setBounds(497, 42, 298, 48);
jSeparator1.setPreferredSize(new java.awt.Dimension(800, 10));
getContentPane().add(jSeparator1);

```

```

jSeparator1.setBounds(0, 108, 1373, 10);
donorID.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
donorID.setText("New Donor Id");
getContentPane().add(donorID);
donorID.setBounds(180, 150, 128, 25);
idField.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
idField.setForeground(new java.awt.Color(255, 51, 0));
getContentPane().add(idField);
idField.setBounds(371, 148, 200, 30);
name.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
name.setText("Full Name");
getContentPane().add(name);
name.setBounds(180, 210, 128, 25);

fatherName.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
fatherName.setText("Father Name");
getContentPane().add(fatherName);
fatherName.setBounds(170, 330, 128, 25);
motherName.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
motherName.setText("Mother Name");
getContentPane().add(motherName);
motherName.setBounds(170, 390, 128, 25);
dob.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
dob.setText("Age");
getContentPane().add(dob);
dob.setBounds(180, 270, 128, 25);
hno.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
hno.setText("Street Address");
getContentPane().add(hno);
hno.setBounds(170, 460, 128, 25);
city.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
city.setText("City");
getContentPane().add(city);
city.setBounds(160, 520, 128, 25);
district.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
district.setText("District");
getContentPane().add(district);

```

```

district.setBounds(160, 570, 128, 25);
pinCode.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
pinCode.setText("pinCode");
getContentPane().add(pinCode);
pinCode.setBounds(707, 267, 128, 25);
bloodGroup.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
bloodGroup.setText("BloodGroup");
getContentPane().add(bloodGroup);
bloodGroup.setBounds(707, 328, 128, 25);
jTextField1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField1);
jTextField1.setBounds(371, 203, 200, 31);
jTextField3.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jTextField3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField3ActionPerformed(evt);
    }
});
getContentPane().add(jTextField3);
jTextField3.setBounds(371, 325, 200, 31);
jTextField4.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField4);
jTextField4.setBounds(371, 386, 200, 31);
jTextField5.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField5);
jTextField5.setBounds(371, 447, 200, 31);
jTextField6.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField6);
jTextField6.setBounds(371, 508, 200, 31);
jTextField7.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField7);
jTextField7.setBounds(371, 569, 200, 31);
jTextField8.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jTextField8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField8ActionPerformed(evt);
    }
}

```



```

});
getContentPane().add(jTextField8);
jTextField8.setBounds(1089, 264, 200, 31);
jTextField2.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField2);
jTextField2.setBounds(371, 264, 200, 31);
Mobile.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
Mobile.setText("Mobile Number");
getContentPane().add(Mobile);
Mobile.setBounds(707, 389, 144, 25);
gender.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
gender.setText("Gender");
getContentPane().add(gender);
gender.setBounds(707, 450, 144, 25);
jTextField10.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
getContentPane().add(jTextField10);
jTextField10.setBounds(1089, 386, 200, 31);
jComboBox1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Male",
"Female", "Other" }));
jComboBox1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox1ActionPerformed(evt);
    }
});
getContentPane().add(jComboBox1);
jComboBox1.setBounds(1089, 447, 200, 31);
jComboBox2.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "A+", "A-",
"B+", "B-", "AB+", "AB-", "O+", "O-" }));
getContentPane().add(jComboBox2);
jComboBox2.setBounds(1089, 325, 200, 31);
jSeparator2.setPreferredSize(new java.awt.Dimension(800, 10));
getContentPane().add(jSeparator2);
jSeparator2.setBounds(0, 734, 1373, 10);
save.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
save.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/save.png"))); //

```

NOI18N

```
save.setText("Save");
save.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        saveActionPerformed(evt);
    }
});
getContentPane().add(save);
save.setBounds(229, 661, 130, 36);
Exit.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
Exit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Exit
application.png"))); // NOI18N
Exit.setText("Exit");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});
getContentPane().add(Exit);
Exit.setBounds(866, 661, 130, 36);
Clear.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
Clear.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Dec.png"))); //
```

NOI18N

```
Clear.setText("Clear");
Clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClearActionPerformed(evt);
    }
});
getContentPane().add(Clear);
Clear.setBounds(562, 661, 140, 36);
image.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/back62.jpg"))); //
```

NOI18N

```
image.setMaximumSize(new java.awt.Dimension(1600, 900));
image.setMinimumSize(new java.awt.Dimension(1600, 900));
image.setPreferredSize(new java.awt.Dimension(1600, 900));
getContentPane().add(image);
```

```

        image.setBounds(0, 0, 1600, 900);

        pack();
    }

    private void formComponentShown(java.awt.event.ComponentEvent evt) {
        // TODO add your handling code here:
        try{
            ConnectionProvider connectionProvider = new ConnectionProvider();
            Connection con = connectionProvider.getCon();
            Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            ResultSet re = st.executeQuery("select max(donorId) from donors");
            if(re.first())
            {
                int id = re.getInt(1);
                id+=1;
                String str = String.valueOf(id);
                idField.setText(str);
            }
            else{
                idField.setText("1");
            }
            re.close();
            st.close();
            con.close();
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
    }

    private void saveActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String donorId = idField.getText();
        String Name = jTextField1.getText();
        String age = jTextField2.getText();
        String Father = jTextField3.getText();
        String Mother = jTextField4.getText();
    }

```

```

String streetAddress = jTextField5.getText();
String City = jTextField6.getText();
String District = jTextField7.getText();
String PinCode = jTextField8.getText();
String mobile = jTextField10.getText();
String BloodGroup = jComboBox2.getSelectedItem().toString();
String Gender = jComboBox1.getSelectedItem().toString();
try{
    ConnectionProvider connectionProvider = new ConnectionProvider();
    Connection con = connectionProvider.getCon();
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
    st.executeUpdate("insert into donors values('" + donorId + "','" + Name + "','" + age + "','" + Father
+ "','" + Mother + "','" + streetAddress + "','" + City + "','" + District + "','" + PinCode + "','" + BloodGroup
+ "','" + mobile + "','" + Gender + "')");
    JOptionPane.showMessageDialog(null,"Data Successfully inserted");
    setVisible(false);
    new addNewDonor().setVisible(true);
}
catch(Exception e){
    JOptionPane.showMessageDialog(null,"Invalid Data");
    setVisible(false);
    new addNewDonor().setVisible(true);
}
}
private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    new Home().setVisible(true);
}
private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    new addNewDonor().setVisible(true);
}
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(addNewDonor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(addNewDonor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(addNewDonor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(addNewDonor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new addNewDonor().setVisible(true);
        }
    });
}

private javax.swing.JButton Clear;
private javax.swing.JButton Exit;
private javax.swing.JLabel Mobile;
private javax.swing.JLabel Title;
private javax.swing.JLabel bloodGroup;
private javax.swing.JLabel city;
private javax.swing.JLabel district;
private javax.swing.JLabel dob;
private javax.swing.JLabel donorID;
private javax.swing.JLabel fatherName;
private javax.swing.JLabel gender;
private javax.swing.JLabel hno;

```

```

private javax.swing.JLabel idField;
private javax.swing.JLabel image;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JComboBox<String> jComboBox2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JLabel motherName;
private javax.swing.JLabel name;
private javax.swing.JLabel pinCode;
private javax.swing.JButton save;
}

```

SEARCH NEAR BY LOCATION

```

/**
 *
 * @author SHIVA KRISHNA
 */
public class SeachByLocation extends javax.swing.JFrame {
    public SeachByLocation() {
        initComponents();
    }
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jLabel2 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jSeparator2 = new javax.swing.JSeparator();
        jScrollPane1 = new javax.swing.JScrollPane();

```

```

jTable1 = new javax.swing.JTable();
jSeparator3 = new javax.swing.JSeparator();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMaximumSize(new java.awt.Dimension(1600, 900));
setMinimumSize(new java.awt.Dimension(1600, 900));
setUndecorated(true);
setPreferredSize(new java.awt.Dimension(1600, 900));
getContentPane().setLayout(null);
jLabel1.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
jLabel1.setText("Search Near by Donors");
getContentPane().add(jLabel1);
jLabel1.setBounds(592, 21, 480, 51);
getContentPane().add(jSeparator1);
jSeparator1.setBounds(0, 78, 1594, 10);
jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Location.png"))); //
NOI18N
jLabel2.setText("City");
getContentPane().add(jLabel2);
jLabel2.setBounds(470, 100, 92, 32);
jTextField1.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});
jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jTextField1KeyReleased(evt);
    }
});
getContentPane().add(jTextField1);
jTextField1.setBounds(590, 100, 292, 32);
getContentPane().add(jSeparator2);

```

```

jSeparator2.setBounds(6, 151, 1588, 10);
jTable1.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportViewView(jTable1);
getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(6, 189, 1588, 447);
getContentPane().add(jSeparator3);
jSeparator3.setBounds(0, 654, 1594, 10);
jButton1.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/print.png"))); //
NOI18N
jButton1.setText("Print");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
getContentPane().add(jButton1);
jButton1.setBounds(688, 682, 150, 46);
jButton2.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Exit
application.png"))); // NOI18N
jButton2.setText("Exit");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
}

```



```

});
getContentPane().add(jButton2);
jButton2.setBounds(1144, 682, 170, 46);
jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/back8.jpg"))); //
NOI18N
getContentPane().add(jLabel3);
jLabel3.setBounds(0, 0, 1600, 900);
pack();
} // </editor-fold>
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        jTable1.print(JTable.PrintMode.NORMAL);
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    new Home().setVisible(true);
}
private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String city = jTextField1.getText();
    try{
        ConnectionProvider connectionProvider = new ConnectionProvider();
        Connection con = connectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select * from donors where city like '%" + city + "%' or district like
 '%" + city + "%'");
        if(rs.next())
        {
            rs.beforeFirst();
            jTable1.setModel(DbUtils.resultSetToTableModel(rs));

```

```

    }
    else{
        JOptionPane.showMessageDialog(null,"Donors List is Empty");
        setVisible(false);
        new SeachByLocation().setVisible(true);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
}

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(SeachByLocation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(SeachByLocation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(SeachByLocation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(SeachByLocation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new SeachByLocation().setVisible(true);
    }
});

```

```

    }
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JTable jTable1;
    private javax.swing.JTextField jTextField1;
}

```

LIST ALL BLOOD BANKS

```

package bloodbank;
import DataBaseConnections.ConnectionProvider;
import javax.swing.JTable;
import java.sql.*;
import net.proteanit.sql.DbUtils;
import javax.swing.JOptionPane;
/**
 *
 * @author SHIVA KRISHNA
 */
public class ListAllBanks extends javax.swing.JFrame {
    public ListAllBanks() {
        initComponents();
    }
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jSeparator2 = new javax.swing.JSeparator();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

```

```

jLabel2 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("BloodBankDetails");
setMaximumSize(new java.awt.Dimension(1600, 900));
setMinimumSize(new java.awt.Dimension(1600, 900));
setUndecorated(true);
setPreferredSize(new java.awt.Dimension(1600, 900));
addComponentListener(new java.awt.event.ComponentAdapter() {
    public void componentShown(java.awt.event.ComponentEvent evt) {
        formComponentShown(evt);
    }
});
getContentPane().setLayout(null);
jLabel1.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
jLabel1.setText("ALL BLOOD BANK DETAILS");
getContentPane().add(jLabel1);
jLabel1.setBounds(617, 41, 496, 48);
getContentPane().add(jSeparator1);
jSeparator1.setBounds(6, 95, 1588, 9);
jTable1.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportView(jTable1);
getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(0, 116, 1600, 492);
getContentPane().add(jSeparator2);
jSeparator2.setBounds(0, 647, 1600, 13);
jButton1.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/print.png"))); //
NOI18N
jButton1.setText("Print");

```

```

jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
getContentPane().add(jButton1);
jButton1.setBounds(687, 699, 177, 44);
jButton2.setFont(new java.awt.Font("Segoe UI", 3, 18)); // NOI18N
jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/Exit
application.png"))); // NOI18N
jButton2.setText("Exit");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
getContentPane().add(jButton2);
jButton2.setBounds(968, 699, 155, 44);
jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/bloodbank/back9.jpg"))); //
NOI18N
getContentPane().add(jLabel2);
jLabel2.setBounds(0, 0, 1600, 900);
pack();
} // </editor-fold>
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    new Home().setVisible(true);
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        jTable1.print(JTable.PrintMode.NORMAL);
    }
    catch (Exception e){
        JOptionPane.showMessageDialog(null,e);
    }
}
private void formComponentShown(java.awt.event.ComponentEvent evt) {

```

```

// TODO add your handling code here:
try{
    ConnectionProvider connectionProvider = new ConnectionProvider();
    Connection con = connectionProvider.getCon();
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
    ResultSet rs = st.executeQuery("select * from bloodbanks");
    if(rs.next())
    {
        rs.beforeFirst();
        jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    }
    else{
        JOptionPane.showMessageDialog(null,"Blood Banks List is Empty");
        setVisible(false);
        new Home().setVisible(true);
    }
} catch (Exception e){
    JOptionPane.showMessageDialog(null,e); } }

public static void main(String args[]) {
    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(ListAllBanks.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(ListAllBanks.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(ListAllBanks.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(ListAllBanks.class.getName()).log(java.util.logging.Level.SEVERE,

```

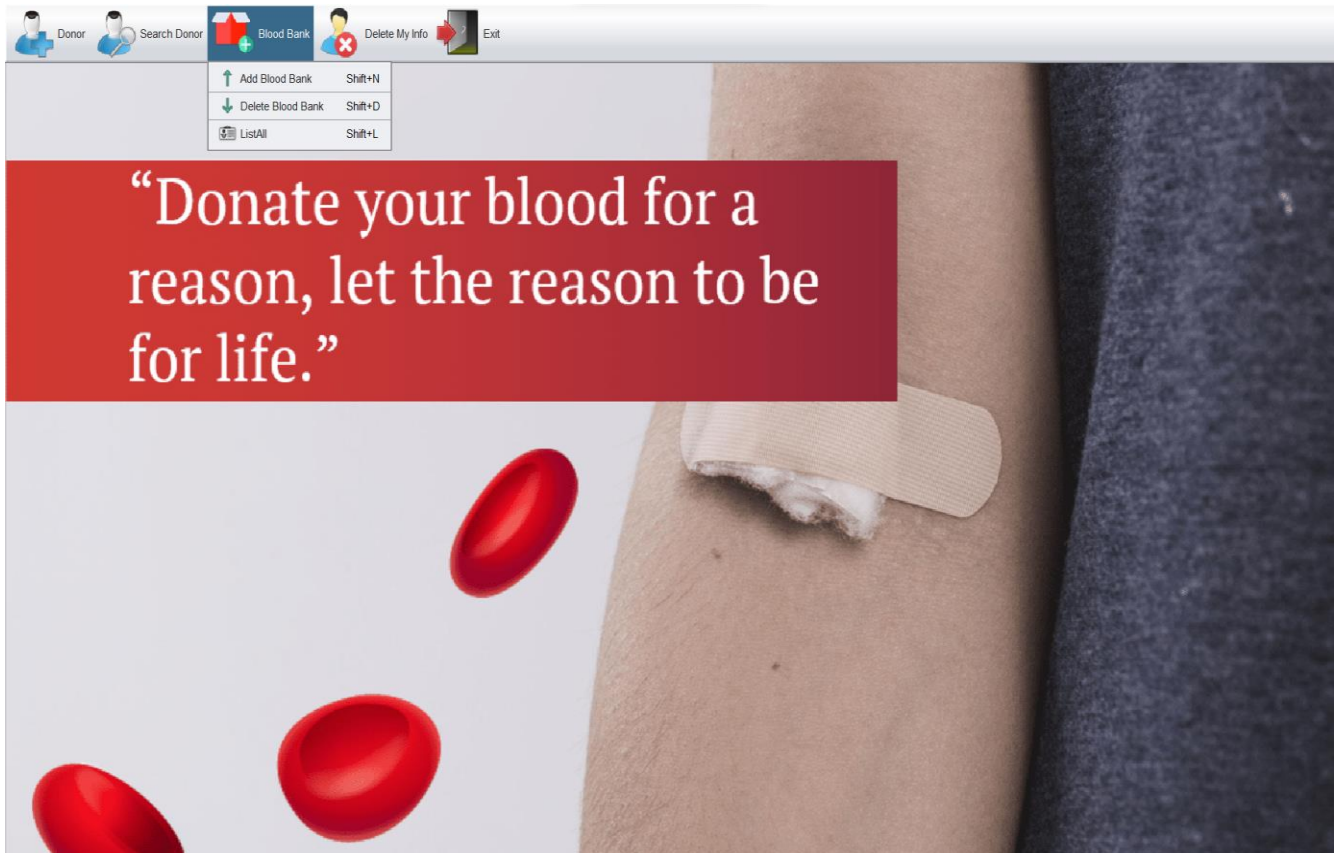
```
    null, ex);}

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ListAllBanks().setVisible(true);
        }
    });

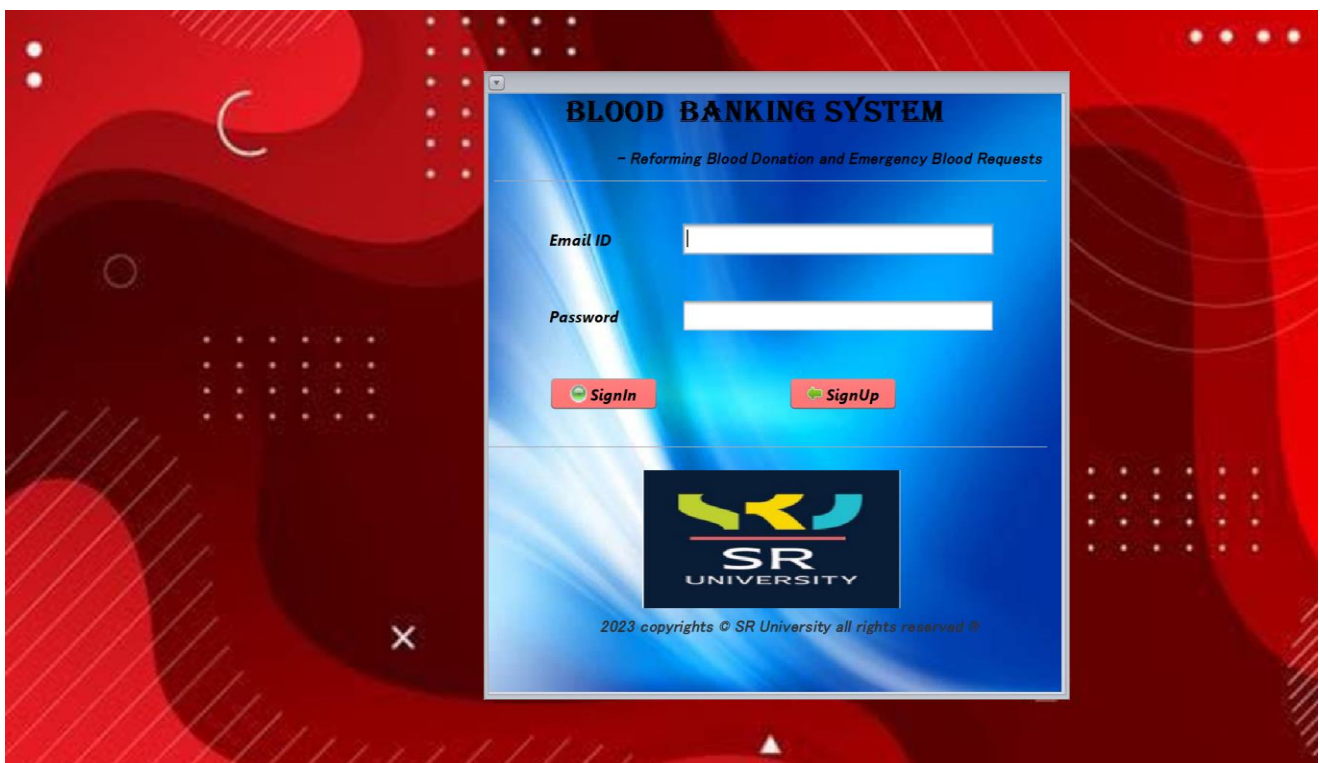
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JTable jTable1;
}
```

7. RESULT SCREENS

1. Home Page



2. Login Page



3. Register Page



The screenshot shows a web browser window titled "Register Form" for the "BLOOD BANKING SYSTEM". The system's tagline is "Reforming Blood Donation and Emergency Blood Requests". The form is set against a red background with a white ECG line and a globe with white crosses. The form fields are: Full Name, Email Id, Mobile, and Password. Below the fields are "Signup" and "Login" buttons. The SR University logo is centered at the bottom, with a copyright notice: "2023 copyrights © SR University all rights reserved ®".

BLOOD BANKING SYSTEM
- Reforming Blood Donation and Emergency Blood Requests

Full Name

Email Id

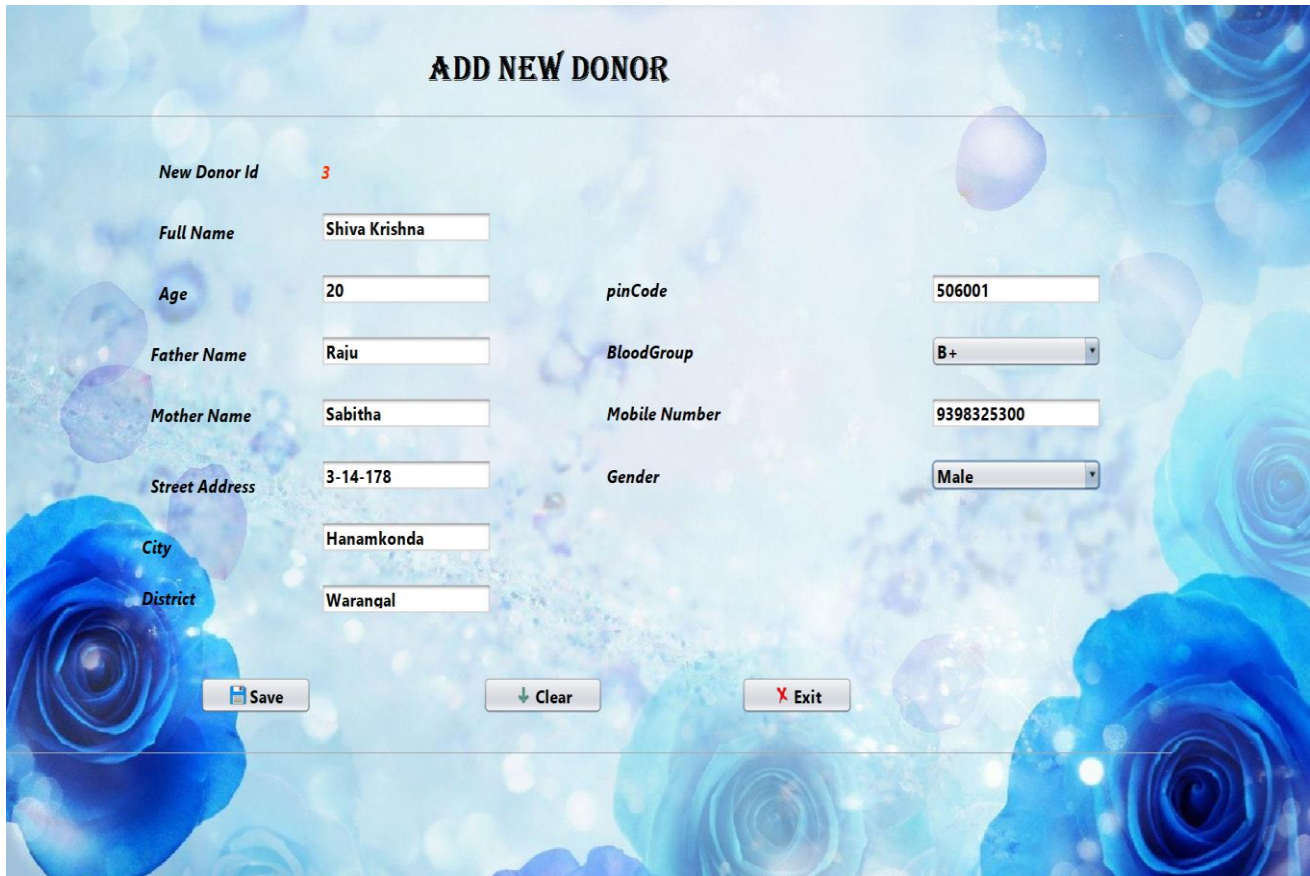
Mobile

Password



2023 copyrights © SR University all rights reserved ®

4. Add Donor Page



The screenshot shows the "ADD NEW DONOR" form on a light blue background with a floral pattern. The form contains the following fields: New Donor Id (3), Full Name (Shiva Krishna), Age (20), pinCode (506001), Father Name (Raju), BloodGroup (B+), Mother Name (Sabitha), Mobile Number (9398325300), Street Address (3-14-178), Gender (Male), City (Hanamkonda), and District (Waranqal). At the bottom are "Save", "Clear", and "Exit" buttons.

ADD NEW DONOR

New Donor Id **3**

Full Name

Age pinCode

Father Name BloodGroup

Mother Name Mobile Number

Street Address Gender

City

District

5. Update Donor Page

Wallpapershd

UPDATE DONOR DETAILS

Mobile Number 9398325300 Search

Full Name	<input type="text" value="Shiva Krishna Beeraboina"/>	District	<input type="text" value="Warangal"/>
Age	<input type="text" value="20"/>	Pin Code	<input type="text" value="506002"/>
Father Name	<input type="text" value="Raju"/>	Mobile	<input type="text" value="9398325300"/>
Mother Name	<input type="text" value="Sabitha"/>	Blood Group	<input type="text" value="B+"/>
Street Address	<input type="text" value="3-14-178"/>	Gender	<input type="text" value="Male"/>
City	<input type="text" value="Hanamkonda"/>		

Update Reset Exit

6. Delete Donor Screen

DELETE DONOR DETAILS


Mobile Number 9989095747 Search

Name	<input type="text" value="Shirisha Beeraboina"/>	PinCode	<input type="text" value="506002"/>
Age	<input type="text" value="22"/>	BloodGroup	<input type="text" value="AB+"/>
Father Name	<input type="text" value="Raju"/>	Gender	<input type="text" value="Female"/>
Mother Name	<input type="text" value="Sabitha"/>	Mobile	<input type="text" value="9989095747"/>
Street Address	<input type="text" value="4-177/A"/>	District	<input type="text" value="Warangal"/>
City	<input type="text" value="Nagaram"/>		



Delete Reset Exit

7. Search by Location Screen

SEARCH NEAR BY DONORS


 **City**

donorId	name	age	fatherName	motherName	streetAddress	city	district	pinCode	bloodGroup	mobile	gender
2	Shirisha Beerab...	22	Raju	Sabitha	4-177/A	Naqaram	Warangal	506002	AB+	9989095747	Female



 **Print** **Exit**

8. Search by Blood Group Page

SEARCH DONOR BY BLOOD GROUP

BloodGroup  **Search**

donorId	name	age	fatherName	motherName	streetAddress	city	district	pinCode	bloodGroup	mobile	gender
2	Shirisha Beerab...	22	Raju	Sabitha	4-177/A	Naqaram	Warangal	506002	AB+	9989095747	Female

 **Print** **Exit**

9. List All Donors Screen

ALL DONOR DETAILS											
donorId	name	age	fatherName	motherName	streetAddress	city	district	pinCode	bloodGroup	mobile	gender
1	Shiva Krishna B...	20	Raju	Sabitha	3-14-178	Hanamkonda	Waranqal	506002	B+	9398325300	Male
2	Shirisha Beerab...	22	Raju	Sabitha	4-177/A	Naqaram	Waranqal	506002	AB+	9989095747	Female

Print

Close

10. Add Blood Bank Page

ADD BLOOD BANK CENTRE			
Blood Bank Id	3		
Bank Name	Redcross	License Number	AZBYCX321
Bank Address	4-167, Near GopalPur	Validity	26-Nov-2025
City	Hanamkonda	Testing Facilities	Platelet Incubators
District	Waranaqal	Storage Unit	Nucleic Acid Testing
State	Telangana		
Email	redcross@blood.in		
Mobile Number	8499010457		
<div><div>Save</div><div>Reset</div><div>Exit</div></div>			

11. Delete Blood Bank Screen

DELETE BLOOD BANK

License Number

AZBYCX321

Search

Bank Name

RedCross

Validity

2028-11-30

Bank Address

4-11/B

Test Facilities

Nucleic Acid Testing

City

Hanamkonda

Storage Unit

Plasma Freezer

District

Warangal

State

Telangana

Email

redcross.blood.in

Mobile

9177410742

Delete

Reset

Exit

12. List All Blood Banks Page

ALL BLOOD BANK DETAILS

bankId	bankName	bankAddress	city	district	state	email	mobile	license	validity	testFacilities	storageUnit
1	RedCross	4-11/B	Hanamkonda	Warangal	Telangana	redcross.blood.in	9177410742	AZBYCX321	2028-11-30	Nucleic Acid Tes...	Plasma Fr
2	Vidya Blood fo...	Near Gopalpur ...	Hanamkonda	Warangal	Telangana	Vidyablood@pv...	8499010457	QAZWSX890	2029-12-19	Transmissible DI...	Aqitators

Print

Exit

8. CONCLUSION

The "**Blood Banking System**" project is not merely a technological achievement but a profound commitment to addressing a critical need in our society. By seamlessly blending innovative features, user-friendly interfaces, the system emerges as a hope for efficient blood donation and emergency response. The unique rewards system adds a layer of inspiration, turning blood donation into a recurring act of kindness and community engagement.

Our project symbolizes more than just successful code implementation—it represents a vision realized, where technology becomes a bridge between compassion and impact. The "**Blood Banking System (reforming blood donation and emergency blood requests)**" stands as a testament to what can be achieved when passion, purpose, and technology converge. May its success inspire future endeavors that harness the power of innovation to create positive, transformative change for the well-being of society.