BLOOD BANK MANAGEMENT SYSTEM

An

Object-Oriented Programming through Java Course Project Report

in partial fulfilment of the degree

Bachelor of Technology
in
Computer Science & Engineering

By

D. Sreeteja	2103A52078
V. Rishika	2103A52116
S. Tejasree	2103A52068
P. Mahanth	2103A52190

Submitted to

P. Sravan

(Assistant professor dept of CS&AI)

SR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **Object Oriented Programming through Java - Course Project** Report entitled **“Blood Bank Management System”** is a record of bonafide work carried out by **D. Sreeteja, V. Rishika, S. Tejasree, P. Mahanth** bearing Roll No(s) **2103A52078, 2103A52116, 2103A52068, 2103A52190** during the academic year 2023-2024 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the SR University.

Lab In-charge

Mr. P. Sravan
Asst. Professor,
S R University,
Ananthasagar, Warangal.

Head of the Department

Dr. M.Sheshikala
Assoc. Prof. & HOD (CSE),
S R University,
Ananthasagar, Warangal.

TABLE OF CONTENTS

1. ABSTRACT

This project is aimed to developing an online Blood Donation Information System. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The website is designed to handle the daily transactions of the blood bank and search the details when required.

There are number of scenarios where urgent need of blood comes. At these critical time, Online Blood Bank project aims at maintaining all the information related to blood donors. The Blood Donation Agent is to create an e-Information about the donor and organization that are related to donating the blood. Through this application any person who is interested in donating the blood can register himself. In the same way if any organization wants to register itself with this site that can also register.

The number of persons who are in need of blood are increasing in large number day by day. In order to help people who are in need of blood, my Online Blood Bank can be used effectively for getting the details of blood donors having the same blood group and within the same city. With the help of my Online Blood Bank people who are having the thought of donating blood gets registered in my Online Blood Bank giving his total details.

Online Blood Bank website is available to everyone easily. A person who likes to donate blood gives his entire details i.e., fill in the registration form and can create a username with a password by which he can modify his details if at all there are any changes in his information given before.

My site also helps people who are in need of blood by giving the details of the donors by searching, if at all there are no donors having the same group and within their own city they will be given the addresses with phone numbers of some contact persons in major cities who represent a club or an organization with free of cost. If at all the people find any difficulty in getting blood from the contact persons we will give them a another contact information i.e.,
India's Largest Paging Service number through which they can give the message on every ones pagers with the blood group and city they are living in, such that the donors who view the messages in their pagers having the same blood group and the in the same city, he contacts the person on phone who are in need of a blood. Such that the person gets help from us which
saves his life. The present project elucidates the following features:-

- Registering the Donors
- Searching a Donor

2. INTRODUCTION

The BLOOD BANK MANAGEMENT SYSTEM is great project. this project is designed for successful completion of project on blood bank management system. the basic building aim is to provide blood donation service to the city recently. Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

The *Blood bank system project report* contain information

- Name of Donor
- Blood type
- Contact Information
- Available Blood group

(i) Overall Description Objective

The main objective of this application is to automate the complete operations of the blood bank. They need maintain hundreds of thousands of records. Also searching should be very faster so they can find required details instantly.

To develop a web-based portal to facilitate the co-ordination between supply and demand of blood . This system makes conveniently available good quality, safe blood and other blood components, which can be provided in a sound, ethical and acceptable manner, consistent with the long-term well being of the community. It actively encourage voluntary blood donation, motivate and maintain a well-indexed record of blood donors and educate the community on the benefits of blood donation. This will also serve as the site for interaction of best practices in reducing unnecessary utilization of blood and help the state work more efficiently towards self sufficiency in blood.

The system will provide the user the option to look at the details of the existing Donor List, Blood Group and to add a new Donor.

(ii) Purpose need of Blood Bank Management System

Bank blood donation system in java is planned to collect blood from many donators in short from various sources and distribute that blood to needy people who require blood. To do all this we require high quality software to manage those jobs. The government spending lot of money to develop high quality “Blood Bank management system project”.

Existing system of blood bank is very much impeded and disrupt. There is a constant need to replenish stocks in blood bank. Blood bank faces lack of information storage. Hence customers have to go around in search of blood unit. For do all those kinds of need blood bank management system project in java contain modules which are include the detail of following areas:

- Blood Donor
- Blood collection
- Stock details
- blood bank system project Reports

3. PROBLEM STATEMENT

Despite advances in technology, nowadays, most blood bank systems are running in manual system. As such, there is a prevalent problem in the availability of needed blood types. For instance, when a person needs a certain type of blood and this type is not available in the hospital, family members send messages through social media to those who can donate to them and this process takes longer than the life of the patient to the most dangerous. In addition, it seems that there is la ck of proper documentation about blood donors and its medical history. This may lead to blood bag contamination and may affect the blood transfusion safety. Generally, this study aims to determine how the use of online bank management system enhance blood transfusion safety**.**

4. EXISTING SYSTEM

In the existing system there is no use of Web Services and Remoting every thing is done manually in the process of records in that existing process the risk of mismanagement and of data is very high. There is very less Security for the data which is stored manually.

5. PROPOSED SYSTEM

The proposed Blood Bank Management System (BBMS) is a comprehensive software solution designed to streamline the management of blood donations, inventory, and blood requests for healthcare institutions and blood banks. This system is intended to be user-friendly, efficient, and secure, serving as a valuable tool for blood resource management.

Advantages:

- User friendliness I provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- It provides high level of security with different level of authentication.

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

6. IMPLEMENTATION

Bank blood donation system in java is planned to collect blood from many donators in short from various sources and distribute that blood to needy people who require blood. To do all this we require high quality software to manage those jobs. The government spending lot of money to develop high quality “Blood Bank management system project”.

6.1 INTRODUCTION TO JAVA

Java is a widely-used, object-oriented, high-level programming language developed by James Gosling and his team at Sun Microsystems. Java is known for its portability, security features, and ease of use, making it suitable for various applications, including web development, mobile apps etc.

JDK stands for Java Development Kit. It is a software development kit used to develop Java applications and applets. The JDK includes a variety of tools and libraries that are essential for developing and running Java programs. Here are the main components of the JDK:

1. Java Compiler (javac):

The Java compiler is used to compile Java source code (.java files) into bytecode (.class files), which can be executed by the Java Virtual Machine (JVM).

2. Java Virtual Machine (JVM):

The JVM is an integral part of the JDK. It interprets compiled Java bytecode and executes the Java program. JVMs are available for various platforms, allowing Java programs to run on different operating systems without modification.

3. Java Runtime Environment (JRE):

The JRE is an environment in which Java programs run. It includes the JVM and other libraries and components required for running Java applications but does not include development tools like compilers and debuggers.

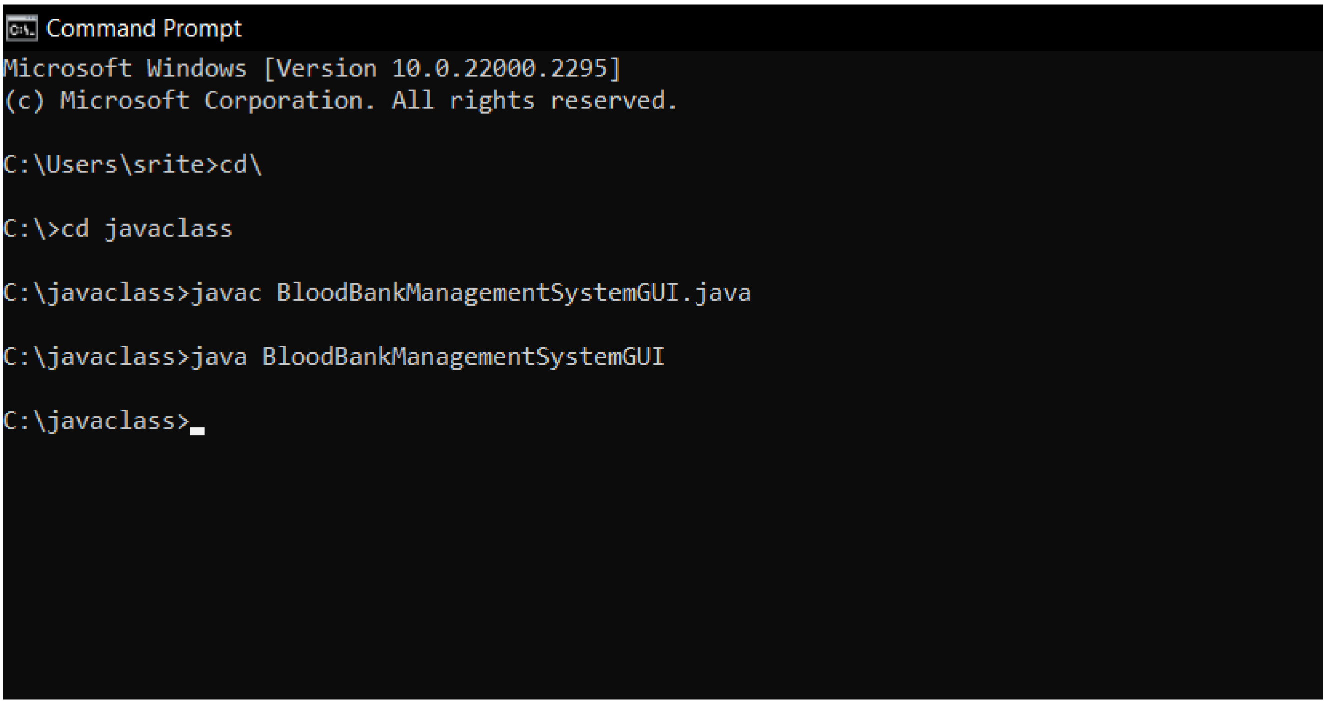
4. Command Prompt:

Using the Command Prompt or Command Line Interface (CLI) within a Java program can be achieved by executing shell commands.

Handling Output: The output of the Command Prompt command is read line by line and printed in the Java console. You can modify the code to process this output in any way you want.

Error Handling: The code includes basic error handling. If the command fails to execute, an IOException will be thrown. If the command does not complete successfully, you can check the exit code using process.exitValue().

Security Considerations : Careful when executing commands obtained from user input. Sanitize and validate any input used to construct the command to prevent security vulnerabilities like command injection.



```
Microsoft Windows [Version 10.0.22000.2295]
(c) Microsoft Corporation. All rights reserved.

C:\Users\srite>cd\

C:\>cd javaclass

C:\javaclass>javac BloodBankManagementSystemGUI.java

C:\javaclass>java BloodBankManagementSystemGUI

C:\javaclass>_
```

Fig 1 : Command prompt for the code

6.2 ELEMENTS USED TO CREATE

1. Swings

Swing is a part of the Java Foundation Classes (JFC) that provides a set of GUI components for building desktop applications in Java. Swing components are lightweight, platform-independent, and customizable, making them ideal for creating graphical user interfaces.

i. Platform Independence:

Swing components are rendered using Java code instead of relying on the native platform's components. This means Swing applications have a consistent look and feel across different operating systems.

ii. **Lightweight Components:**

Swing components are lightweight because they don't depend on the native platform's GUI components. They are implemented entirely in Java and do not rely on native peers.

iii. GUI Components:

Swing provides a wide range of GUI components, including buttons, labels, text fields, text areas, check boxes, radio buttons, combo boxes, lists, tables, trees, and more. These components can be customized and combined to create complex user interfaces.

iv. Layout Managers:

Swing provides layout managers such as BorderLayout, FlowLayout, GridLayout, and BoxLayout, which help in arranging components within containers. Layout managers automatically handle the sizing and positioning of components, ensuring a consistent layout across different screen sizes.

2. GUI [Graphical User Interface]:

Graphical User Interface (GUI) is a user interface that allows users to interact with electronic devices through graphical elements such as icons, buttons, and windows. It provides a more intuitive and user-friendly way to interact with software applications.

GUI components are placed and organized within a JFrame and JPanel, providing a basic but functional interface for adding donors and searching for donors based on their blood group. Users interact with the system by entering information into the text fields and clicking the corresponding buttons. The results are displayed in the text area for user feedback.

JFrame : JFrame is a top-level container that represents the main window of a Java Swing application. It provides the basic functionalities of a window, such as minimizing, maximizing, and closing the application. In the project, the JFrame represents the main window of the Blood Bank Management System application.

JPanel : JPanel is a container that can hold other GUI components like buttons, labels, and text fields. It is used to organize and group components within a window.The JPanel holds all the other GUI components, arranging them in a specific layout.

JButton : JButton is a button component that can trigger actions when clicked by the user. It is often used to initiate some action, like submitting a form or executing a search. In the code, JButtons are used for actions like adding a donor and searching for donors based on the blood group.

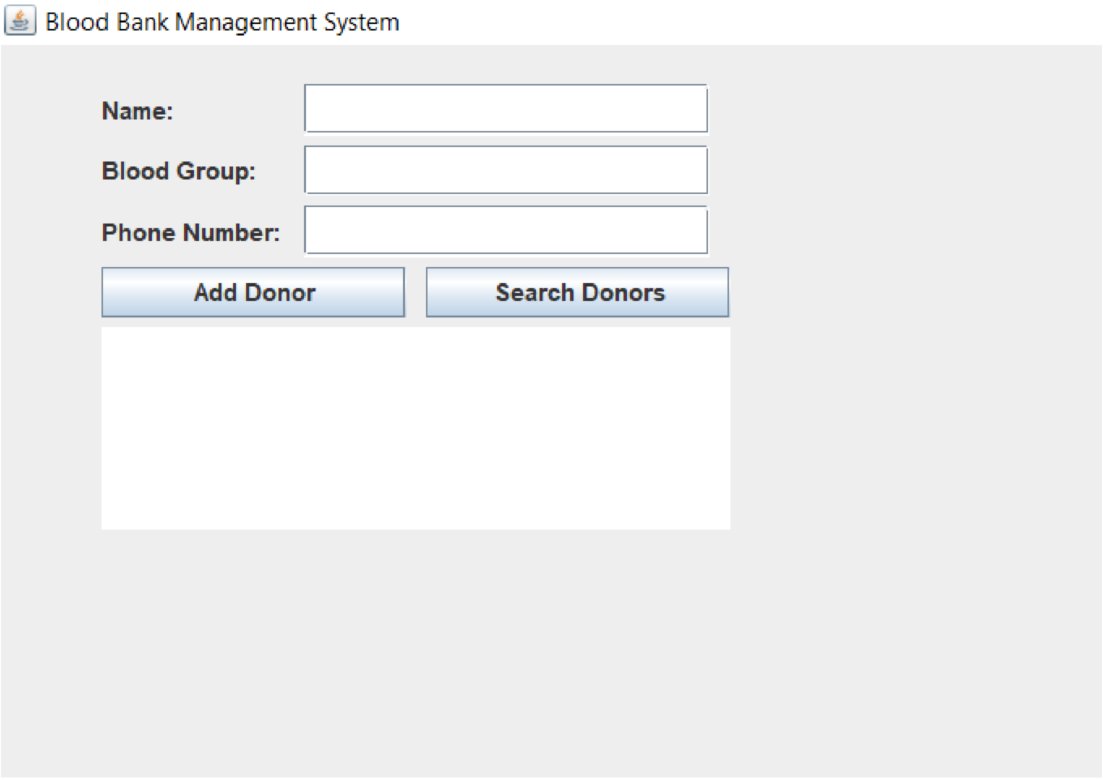


Fig 2 : frame for the registration

6.3 CODE

```
import javax.swing.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.stream.Collectors;


class Donor {

    private String name;

    private String bloodGroup;

    private String phoneNumber;


    public Donor(String name, String bloodGroup, String phoneNumber) {

        this.name = name;

        this.bloodGroup = bloodGroup;

        this.phoneNumber = phoneNumber;

    }


    public String getName() {

        return name;

    }


    public String getBloodGroup() {

        return bloodGroup;

    }


    public String getPhoneNumber() {
        return phoneNumber;
    }
}

class BloodBank {
    private ArrayList<Donor> donors;

    public BloodBank() {
        donors = new ArrayList<>();
    }

    public void addDonor(String name, String bloodGroup, String phoneNumber) {
        Donor donor = new Donor(name, bloodGroup, phoneNumber);
        donors.add(donor);
    }

    public ArrayList<Donor> searchDonors(String bloodGroup) {
        return donors.stream()
                .filter(donor -> donor.getBloodGroup().equalsIgnoreCase(bloodGroup))
                .collect(Collectors.toCollection(ArrayList::new));
    }
}

public class BloodBankManagementSystemGUI {
    private BloodBank bloodBank;

    public BloodBankManagementSystemGUI() {
        bloodBank = new BloodBank();
    }

    public void run() {
        JFrame frame = new JFrame("Blood Bank Management System");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        JPanel panel = new JPanel();
        frame.add(panel);
        placeComponents(panel);

        frame.setVisible(true);
    }

    private void placeComponents(JPanel panel) {
        panel.setLayout(null);

        JLabel nameLabel = new JLabel("Name:");
        nameLabel.setBounds(50, 20, 80, 25);
        panel.add(nameLabel);

        JTextField nameText = new JTextField(20);
        nameText.setBounds(150, 20, 200, 25);
        panel.add(nameText);

        JLabel bloodGroupLabel = new JLabel("Blood Group:");
```

```
bloodGroupLabel.setBounds(50, 50, 80, 25);
panel.add(bloodGroupLabel);

JTextField bloodGroupText = new JTextField(20);
bloodGroupText.setBounds(150, 50, 200, 25);
panel.add(bloodGroupText);

JLabel phoneLabel = new JLabel("Phone Number:");
phoneLabel.setBounds(50, 80, 100, 25);
panel.add(phoneLabel);

JTextField phoneText = new JTextField(20);
phoneText.setBounds(150, 80, 200, 25);
panel.add(phoneText);

JButton addButton = new JButton("Add Donor");
addButton.setBounds(50, 110, 150, 25);
panel.add(addButton);

JButton searchButton = new JButton("Search Donors");
searchButton.setBounds(210, 110, 150, 25);
panel.add(searchButton);

JTextArea resultArea = new JTextArea();
resultArea.setBounds(50, 140, 310, 100);
panel.add(resultArea);
```

```
    addButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String name = nameText.getText();
            String bloodGroup = bloodGroupText.getText();
            String phoneNumber = phoneText.getText();

            bloodBank.addDonor(name, bloodGroup, phoneNumber);
            resultArea.setText("Donor added successfully!");
        }
    });

    searchButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String bloodGroup = bloodGroupText.getText();
            ArrayList<Donor> searchResult = bloodBank.searchDonors(bloodGroup);
            updateResultArea(resultArea, searchResult);
        }
    });
}
private void updateResultArea(JTextArea resultArea, ArrayList<Donor> donors) {
    if (donors.isEmpty()) {
        resultArea.setText("No donors found for the given blood group.");
    } else {
        StringBuilder result = new StringBuilder();
        result.append("Donors found:\n");
        for (Donor donor : donors) {
                result.append("Name: ").append(donor.getName()).append(", Blood Group: ").append(donor.getBloodGroup())
                        .append(", Phone Number: ").append(donor.getPhoneNumber()).append("\n");
            }
            resultArea.setText(result.toString());
        }
    }
    public static void main(String[] args) {
        BloodBankManagementSystemGUI system = new BloodBankManagementSystemGUI();
        system.run();
    }
}
```

7. OUTPUT

1) Home page

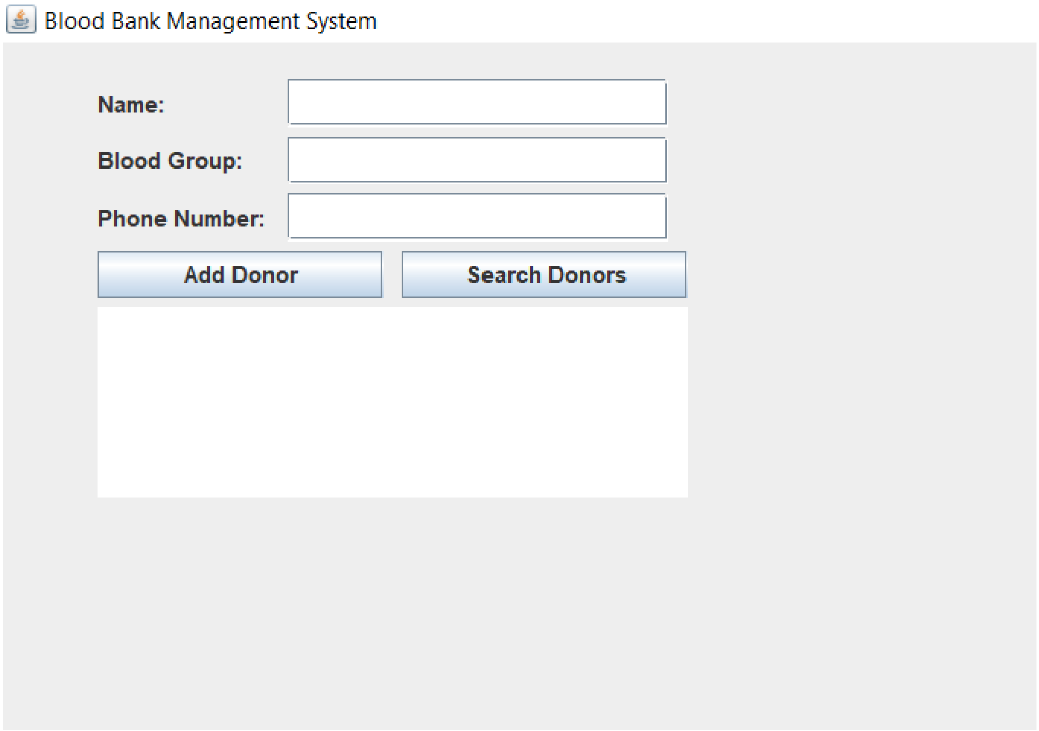
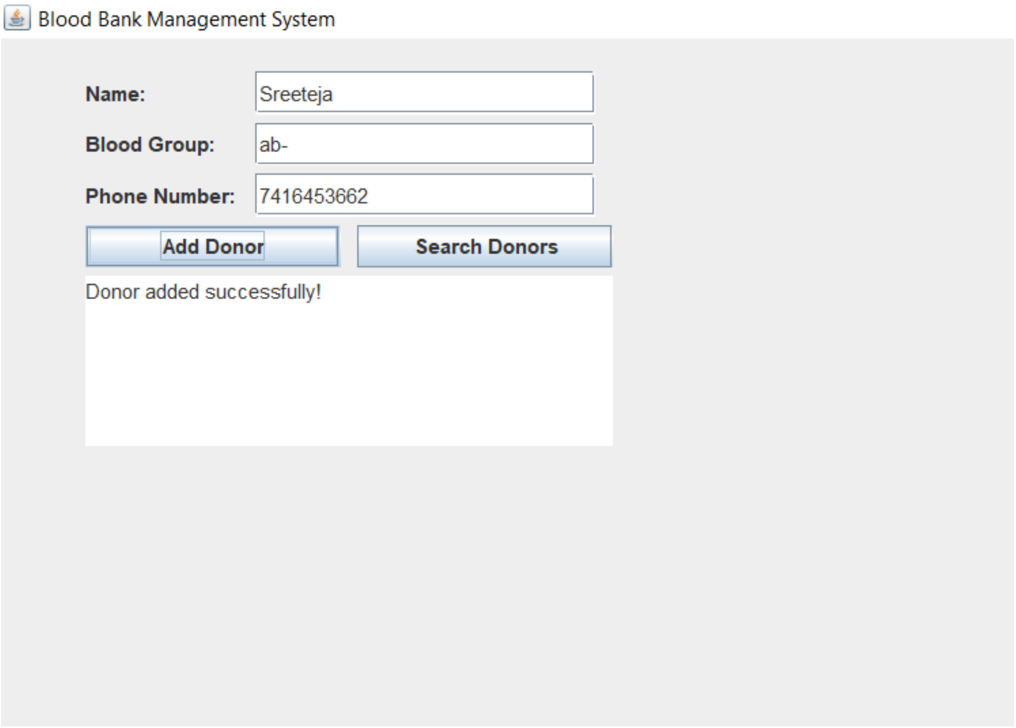


Fig 3 : Homepage for the registration

2) Adding Blood group



Name: Sreeteja

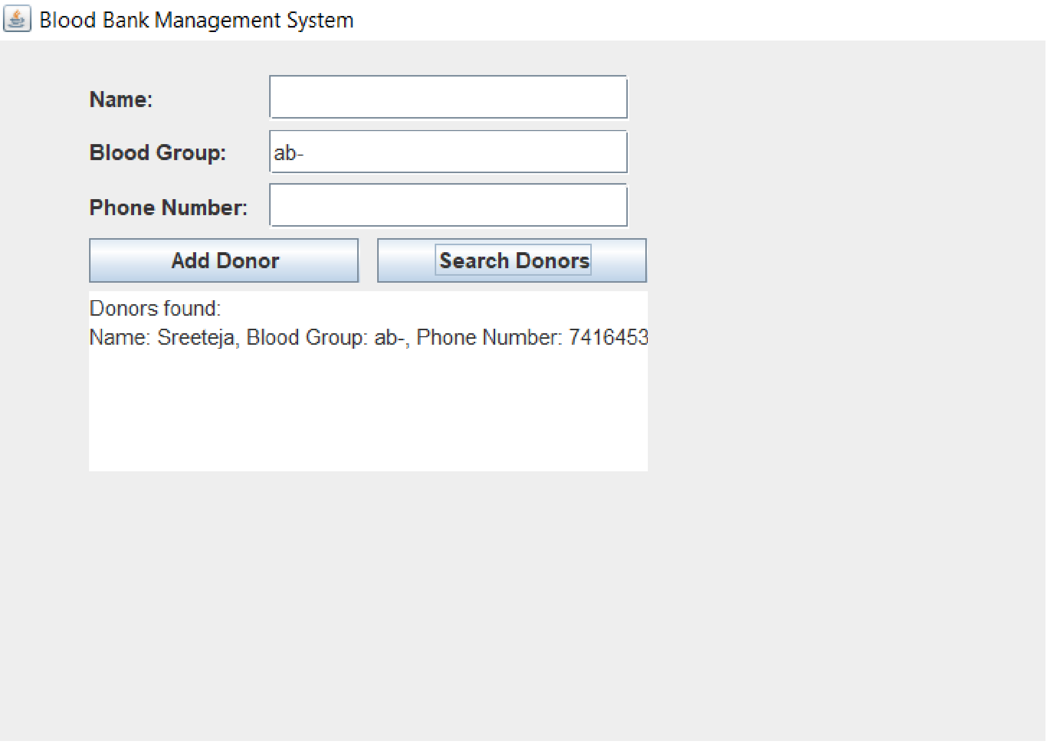
Blood Group: ab-

Phone Number: 7416453662

Donor added successfully!

Fig 4 : Adding page for the Bloodbank

3) Searching blood group



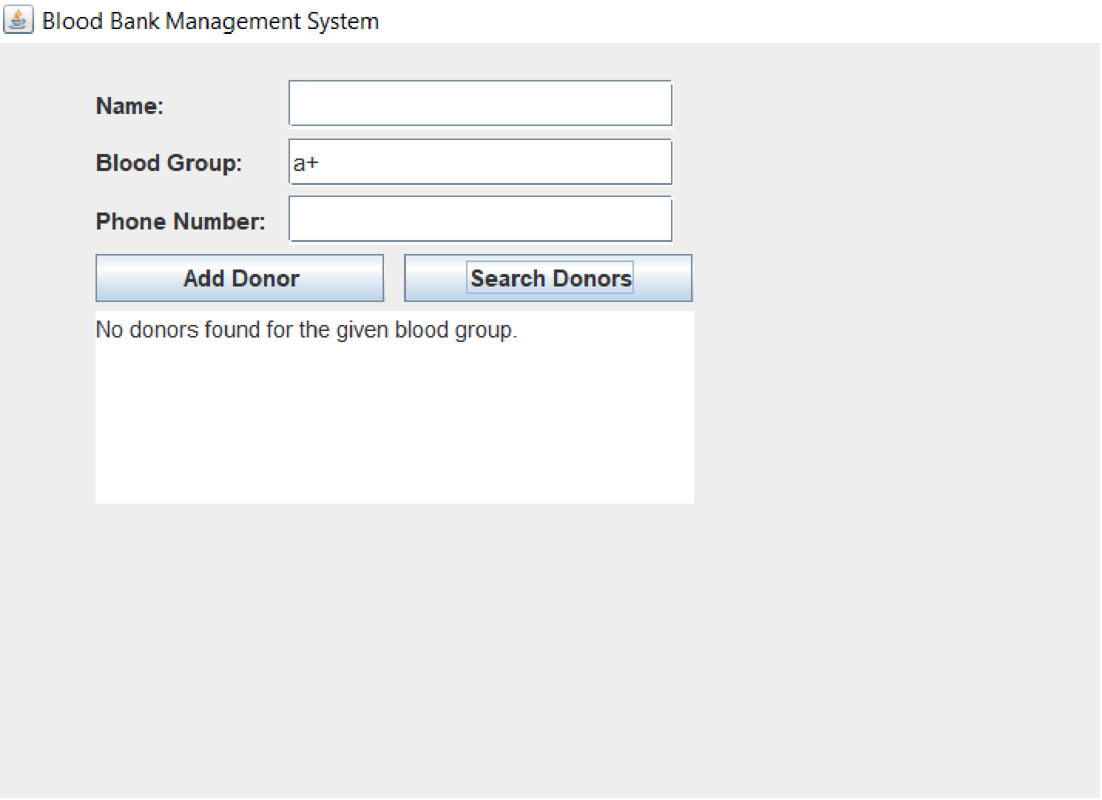
Name:

Blood Group: ab-

Phone Number:

Donors found:
Name: Sreeteja, Blood Group: ab-, Phone Number: 7416453

Fig 5 : Search for blood group



Name:

Blood Group: a+

Phone Number:

No donors found for the given blood group.

Fig 6: No search results founds

8. TEST CASES

Home page with minimize and maximize	Pass
Adding donars	pass
Searching donars	Pass
Displaying donar details	Pass
Availability	pass

9. CONCLUSION

Based on results, this study concluded that online blood bank management system is much better than the manual system. The findings showed that respondents prefer to use online blood bank management system rather than the manual system because it offers many advantages and benefits that lead to its effectiveness, and efficiency. Because of the increased confidence on the users on the system, it can be concluded that the online blood bank management system enhances blood transfusion safety because it provides better ways of handling the various processes in blood bank.

10. FUTURE SCOPE

The future scope for a Blood Bank Management System (BBMS) can encompass several advancements and improvements to enhance its efficiency, usability, and impact. Here are some potential areas of future development for a Blood Bank Management System:

1. **Integration with IoT Devices** : Implement IoT devices to monitor real-time blood storage conditions such as temperature and humidity. Alerts can be sent to the management system in case of any discrepancies.
2. **Mobile Application Development** : Develop mobile applications (iOS and Android) for donors, recipients, and blood bank staff. Mobile apps can enable users to schedule appointments, receive notifications, and track their donation history.
3. **Machine Learning and Predictive Analytics :** Utilize machine learning algorithms to predict blood demand based on historical data, events, and seasonal trends.
4. **Biometric Authentication :** Implement biometric authentication methods, such as fingerprint or facial recognition, to enhance the security of donor information and ensure accurate identification.
5. **AI-Powered Chatbots :** Implement AI-powered chatbots on the website and mobile apps to provide instant responses to common queries, appointment scheduling, and donation eligibility checks.