BLOOD BANK MANAGEMENT SYSTEM

A Project Report submitted to
Bishop Heber College (Autonomous), Tiruchirappalli
affiliated to Bharathidasan University, Tiruchirappalli – 620024
in partial fulfillment of the requirements for the award of the degree of

Bachelor of Vocation in Information Technology

By

S. MOHANRAJ

(REGISTER NO: 225915135)

Under the guidance of
Mrs.P.USHA,MCA .,M.Phil., SET., NET..

Department of Information Technology

Bishop Heber College(Autonomous)

(Nationally Re-accredited with 'A' Grade by NAAC with a CGPA of 3.58 out of 4)

(Recognized by UGC as "College of Excellence")

Tiruchirappalli 620 017

APRIL 2023

Department of Information Technology
Bishop Heber College(Autonomous)
Tiruchirappalli-620 017,Tamilnadu,India
Phone No:0431-277 0136

CERTIFICATE

This Viva-Voce examination for the candidate **S.MOHANRAJ** **(Reg No:225915135)** was held on__________________________

Signature of HOD

Examiner

1.

2.

Mrs. P.USHA, MCA .,M. Phil.,SET.,NET..

Assistant Professor,
Department of Information Technology,
Bishop Heber College (Autonomous),
Tiruchirapalli-620 017

Date:

CERTIFICATE

This is to certify that the project work entitled **" BLOOD BANK MANAGEMENT SYSTEM"** is a bonafide work done under my supervision by **S.MOHANRAJ** and submitted to Bishop Heber College (Autonomous), Tiruchirappalli – 620 017 in partial fulfillment of the requirements for the award of the degree of Bachelor of Vocation in Information Technology during the even semester of the academic year (2022-2023).

Signature of the Guide

DECLARATION

I hereby declare that the work presented in this project work report is the original work done by me under the guidance of **Mrs. P.USHA, MCA .,M. Phil.,SET.,NET.,** Assistant Professor of Information Technology, Bishop Heber College (Autonomous),Tiruchirapalli-620 017and has not been included in any other project work submitted for any other degree.

Name of the Candidate	:S.MOHANRAJ
Register Number	:225915135
Semester	:SECOND
Academic Year	:2022 – 2023
Course Code	:U21ITPJ2

Signature of the Candidate

ACKNOWLEDGEMENTS

Words are beyond our expression as I remain with grateful heart at the thershold of presenting our consolidated project report joyfully acknowledging the divine providence that I experience throughout my project.

First of foremost, I thank **THE ALMIGHTY GOD** for granting abundant grace. Good health and knowledge to do this Project.

I express my sincere gratitude to Dr**.D.PAUL DHAYABARAN, M.Sc., M.Phil., PGDCA., Ph.D.,** Principal., Bishop Heber College (Autonomous), Tiruchirapalli for providing the opportunity to pursue this programme.

I am highly indebted to thanks **Dr. J. JOHN RAYBIN JOSE, M.sc.,MCA, M.Phill, PGDCA ,Ph.D, SET.** Associate professor and Head , Department of Information Dechnology , Bishop Heber College (Autonomous), Trichy for providing all the support and facilities to do this project work.

I wish to place on record my gratitude to **Mrs. P.USHA, MCA.,M.Phil.,SET.,NET.,** Assistant Professor in Department of Information Technology, Bishop Heber College (Autonomous), Tiruchirapalli for grant me permission to pursue carry on with this project and guiding me to complete the project successfully.

I thank all the Staff members of the Department of Information Technology for their contribution to complete this project work successfully.

I record my deep sense of gratitude to my beloved parents and my friends for their encouragement and moral support extended during the period of my project.

S.MOHANRAJ

ABSTRACT

This project entitled as”BLOOD BANK MANAGEMENT SYSTEM".In the existing system donor details are stored in the form of records.In this project ,Donor details are computerized.There are two types of user in this project .Admin can register donor details such as donor Id,name,stock in numbers,donor location and donor blood type can register in our blood bank . The person who need blood can request to admin. This system very useful to customers who want blood and developed in Java and MySql as backend .This project is simple of user friendly.

CONTENTS

S. NO.	TITLE	PAGE NO.
1	**INTRODUCTION**	1
2	**SYSTEM STUDY**	2
	2.1. Project Description	2
	2.1.1. Existing System	2
	2.1.2. Proposed System	2
	2.2 Requirement Analysis	5
	2.2.1 Hardware Requirements	5
	2.2.2 Software Requirements	5
3.	**SYSTEM DESIGN**	9
	3.1. Logical Design	9
	3.2 Database Design	10
4.	**SYSTEM DEVELOPMENT**	11
	4.1. Program Code	11
5.	**SYSTEM TESTING**	23
	5.1. Unit testing	23
	5.2.Integration testing	24
	5.3 Validation Testing	25
6.	**SOFTWARE DEMONSTRATION**	26
7.	**CONCLUSION**	33
	BIBLIOGRAPHY	34

1. INTRODUCTION

A blood bank is a center where blood gathered as a result of blood donation is stored and preserved for later use in blood transfusion. The term "blood bank" typically refers to a department of a hospital usually within a Clinical Pathology laboratory where the storage of blood product occurs and where pre-transfusion and Blood compatibility testing is performed. However, it sometimes refers to a collection center, and some hospitals also perform collection. Blood banking includes tasks related to blood collection, processing, testing, separation, and storage.

The blood bank plays an important role in patient care. Transfusion-related errors have serious consequence for patients, including death. Proper identification of the patient and blood products is critical in avoiding such transfusion-related reactions.

Routine blood storage is 42 days or 6 weeks for stored packed red blood cells (also called "StRBC" or "pRBC"), by far the most commonly transfused blood product, and involves refrigeration but usually not freezing. There has been increasing controversy about whether a given product unit's age is a factor in transfusion efficacy, specifically on whether "older" blood directly or indirectly increases risks of complications. Studies have not been consistent on answering this question, with some showing that older blood is indeed less effective but with others showing no such difference; nevertheless, as storage time remains the only available way to estimate quality status or loss, a first-in-first-out inventory management approach is standard presently. It is also important to consider that there is large variability in storage results for different donors, which combined with limited available quality testing, poses challenges to clinicians and regulators seeking reliable indicators of quality for blood products and storage systems.

2. SYSTEM STUDY

System analysis is a process of gathering the facts concerning the system breaking them into elements and relationship between elements; it provides a framework for visualizing the organizational and environmental factors that operate on a system. The quality of work performed by a machine is usually uniform, neat and more reliable when compared to doing the same operations manually.

2.1.Project Description

This project Blood Bank Management System includes **registration of donors, storing their details into the database**. Our software has the facility to give a unique id for every patient and stores the details of every donor and the user automatically. This software application helps administrator to update data in step by step process. Admin will login using for authentication purpose.

2.1.1. Existing System

The existing system is handled manually. The system has a formatted blood bank for details in paper work. The indent is prepared when stock are to be view and copy is generated for stock details . The system follows large number of paper work for maintaining donor details and user can be difficult to search the blood in database.

- Time Consuming
- Less Efficient

2.1.2.Proposed System.

Using a manual system makes it challenging to accomplish this goal since the data is dispersed, often duplicated, and gathering pertinent data might take a lot of time. These issues are all resolved. simple banking management system This method prioritizes the easy and clear maintenance of information.Blood Bank Managemnt System is a computerized one donor details are in this project donor details are simple form customer can register and enter our project.They can search donor details using donor Id.If they are needed they can communicate easily.

Advantages:

- The system automates the manual procedure of managing blood bank activities.
- It even generates an instant result.
- The system is convenient and flexible to be used.
- It saves their time, efforts, money and resources.

Disadvantages:

- Requires large database.
- The admin has to manually keep updating the information by entering the details in the system.
- Need Internet connection.

Module Description

Blood Bank Management is the add record and delete record to all details.

Modules:

Implementation is the stage of the project when the theoretical design is turned out into a working System. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

- **Donor**

  In this module ,admin login to the system to add donor like donor id,donor name , donor bloodgroup,address etc..,and edit the donor.

  User login to the system to view the donor details.

- **Search Donor**

  This module ,user login to the system to search the donor using the location or blood type.

- **Stock**

  In this module ,admin login to the system to update the stock like bloodgroupand Units.

  User login to the system to view the stock details.

- **Delete Donor**

  To delete the donors using the donor id.

- **Exit**

  To exit the Application.

2.2.Requirment Analysis

Recruitments analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in recruitments as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process final from start to finish. Energy should be directed towards ensuring that the system or product conforms to client needs rather than attempting to user expectations to fit the recruitments. Recruitments analysis is a team effort that demands a combination of hardware, software and human factors engineering as well as skills in dealing with people.

2.2.1.Hardware Requirement

The hardware specification of the Laptop computer system available for developing the project is given below.

Processor	:Intel Core I3
Hard disk	:500 GB
RAM	:4 GB
Keyboard	:Standard Keyboard.
Mouse	:Standard Mousepad

2.2.2.Software Requirement

A Software requirement specification (SRS) is a complete description of the behavior or the system to be developed. It includes a set of use cases that describes all the interaction the users will have with the software.

Operating System	**:**	**Windows 11**
Software application	**:**	**NetBeans IDE 8.1**
Front End	**:**	**Java**
Back End	**:**	**Mysql**
Server	**:**	**Xamp**

Operating system

An operating system is software that communicates with the hardware and allows other to run. 1t comprises the system software, and other utilities. The Operating system used as platform to develop this project in windows 10 Ultimate and later. It is a multi-user operating system.

Windows 10 is a major release of Microsoft's Windows NT operating system. It is the direct successor to Windows 8.1, which was released nearly two years earlier. It was released to manufacturing on July 15, 2015, and later to retail on July 29, 2015. Windows 10 was made available for download via MSDN and TechNet, as a free upgrade for retail copies of Windows 8 and Windows 8.1 users via the Windows Store, and to Windows 7 users via Windows Update. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10, which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support. In June 2021, Microsoft announced that support for Windows 10 editions which are not in the Long-Term Servicing Channel (LTSC) will end on October 14, 2025.

NetBeans:

NetBeans IDE is a free and open-source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems. The IDE simplifies the development of web, enterprise, desktop, and mobile applications that use the Java and HTML5 platforms. NetBeans IDE lets you quickly and easily develop Java desktop, mobile, and web applications, as well as HTML5 applications with HTML, JavaScript, and CSS. The IDE also provides a great set of tools for PHP and C/C developers.

Programming language

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

As of September 2022, Java 19 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.

Mysql

A database management system is essential for helping users to manage and develop databases. The most popular one is **MySQL**. It lets developers build multiple databases that can be accessed by multiple users at the same time. MySQL is perfect for managing relational databases, which typically handle very large amounts of data for all sorts of purposes. These usually come in the form of tables with columns and rows. Indexes are often used to make access quicker. Lots of large familiar companies like Facebook and Flickr depend on MySQL for their services. There’s a free version of MySQL and you can also pay for extra features that expand the possibilities.MySQL is part of the LAMP collection of software which is often used with web servers. Programs like WordPress, phpBB, TYPO3 and Drupal use MySQL. MySQL is popular and widely used because versions are available for around 30 operating systems, including Linux, Microsoft Windows, Mac OS X, AIX, Solaris, and Symbian.For the management of MySQL there are various graphical applications that can be used for a simple user-friendly creation and manipulation of databases, most popular of which is phpMyAdmin—also free web-based application offered by most hosts worldwide. Databases are an essential back-end feature of most software applications. They are where all data is collected, stored and organized. Whenever you conduct a web search, log in to an account or complete an online transaction, a database system is silently recording the information in a structured manner. This makes data access simple and reliable in the future. Given how much data we now generate, it’s easy to see why robust database structures are so important in web development today. Organizations are collecting vast amounts of quantitative and qualitative data, but need reliable data models and database software to turn this into a competitive advantage.In reality, any application with even a moderate level of complexity will likely require the use of Database Management Systems (DBMS) that can handle all the database files. A DBMS offers a systematic solution for the collection of data - a way to create, maintain and control access to database files. The database design optimizes the way in which stored data can be structured,manipulated and retrieved in a secure manner.

3.SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of system theory to product development.

3.1. Logical Design

Logical design is an abstract concept in computer programming by which programmers arrange data in a series of logical relationships known as attributes or entities. An entity refers to a chunk of information, whereas an attribute defines the unique properties of an entity.

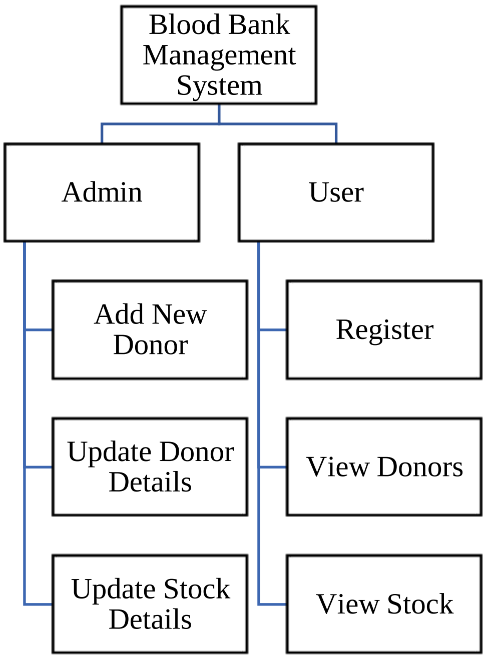


Fig.3.1 BloodBankManageMent System

3.2Database Design

Database design is the process of producing a detailed data model of the database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. A good database design is important in ensuring consistent data, elimination of data redundancy, efficient execution of queries and high-performance application.

Login Table

Field	Type	Null
User	varchar(50)	No
DOB	varchar(50)	No
Fname	varchar(50)	No
Gender	varchar(50)	No
BG	varchar(50)	No
Pno	varchar(50)	No
Email	varchar(50)	No
City	varchar(50)	No
Username	varchar(50)	No
Password	varchar(50)	No

Donor Table

Field	Type	Null
Donorid	Int	No
Name	varchar(50)	No
Fathername	varchar(50)	No
Mothername	varchar(50)	No
DOB	varchar(50)	No
MobileNo	varchar(50)	No
Gender	varchar(50)	No
Email	varchar(50)	No
Bloodgroup	varchar(50)	No
City	varchar(50)	No
Address	varchar(50)	No

Stock Table

Field	Type	Null
Bloodgroup	Varchar(50)	No
Uits	Int	No

4.SYSTEM DEVELOPMENT

The Software Development Life Cycle(SDLC), also referred to as the application development life-cycle, is a term used in system engineering, information system and software engineering to describe a process for planing, creating testing and deploying an information system. The system developments life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

4.1 Program Code

Here I displayed the coding of my program:

Login:

```
private void loginActionPerformed(java.awt.event.ActionEvent evt) {
      String uname1= uname.getText();
      String pass1= pass.getText();
      if(uname1.equals("admin")&&pass1.equals("admin")){
this.dispose();
          home n=new home();
          n.setVisible(true); }
      else{
          JOptionPane.showMessageDialog(null,"Username Password does not exits");
       }String passwordcheck="";
       try{
              Class.forName("com.mysql.jdbc.Driver");
 Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3307/bbms","root","");
                 Statement stmt=(Statement) con.createStatement();
                 ResultSet rs;
```

```
                rs = stmt.executeQuery("SELECT * FROM `login` WHERE
`username`='"+uname1+"'");
                while(rs.next())
                passwordcheck=rs.getString("password");
                if(passwordcheck.equals(pass1)) {
                       this.dispose();
                       home n=new home();
                       n.setVisible(true);
                    }
                    else{
                            JOptionPane.showMessageDialog(null,"Username Password
does not exits");
                         }  con.close();
                          }catch(Exception e){ System.out.println(e);}
   }
```

Signup:

```
private void signupActionPerformed(java.awt.event.ActionEvent evt) {
       String user=name.getText();
       SimpleDateFormat dFormat=new SimpleDateFormat("dd-MM-yyyy");
       String DOB=dFormat.format(dob.getDate());
       String fname=pname.getText();
       String Gender=(String)gender.getSelectedItem();
       String BG=(String)bgrop.getSelectedItem();
       String Pno=pno.getText();
       String EMAIL=email.getText();
       String City=city.getText();
       String username=uname.getText();
       String password=pass.getText();
        try{
          Connection con=ConnectionProvider.getCon();
          Statement st=con.createStatement();
```

```
            st.executeUpdate("insert into login values('"+user+"','"+DOB+"','"+fname+"','"+Gender+"','"+BG+"','"+Pno+"','"+EMAIL+"','"+City+"','"+username+"','"+password+"')");
            JOptionPane.showMessageDialog(null,"SUCCESSFULLY UPDATED");
            setVisible(false);
            new signup().setVisible(true);
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
```

Home:

```
private void addnewActionPerformed(java.awt.event.ActionEvent evt) {
        new home().setVisible(false);
        new addNewDonors().setVisible(true);
    }

    private void searchbloodgroupActionPerformed(java.awt.event.ActionEvent evt) {
        new home().setVisible(false);
        new searchDonorBloodGroup().setVisible(true);
    }

    private void stockincreaseActionPerformed(java.awt.event.ActionEvent evt) {
        new home().setVisible(false);
        new stockIncrease().setVisible(true);
    }

    private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
        new home().setVisible(false);
        new deleteDonor().setVisible(true);
    }
```

```
        private  void  updatedetailActionPerformed(java.awt.event.ActionEvent  evt)  {
        new home().setVisible(false);
        new updateDetailsDonor().setVisible(true);
    }

    private void alldonordetailActionPerformed(java.awt.event.ActionEvent evt) {
      new home().setVisible(false);
      new allDonorDetails().setVisible(true);
    }

    private void searchlocationActionPerformed(java.awt.event.ActionEvent evt) {
      new home().setVisible(false);
      new searchDonorLocation().setVisible(true);
    }

    private void stockdecreaseActionPerformed(java.awt.event.ActionEvent evt) {
      new home().setVisible(false);
      new stockDecrease().setVisible(true);
    }

    private void stockdetailActionPerformed(java.awt.event.ActionEvent evt) {
      new home().setVisible(false);
      new stockDetails().setVisible(true);
    }
```

AddNewDonor:

```
private void saveActionPerformed(java.awt.event.ActionEvent evt) {
      String donorId=newid.getText();
      String name=fullname.getText();
      String fatherName=fname.getText();
      String motherName=mname.getText();
      SimpleDateFormat dFormat=new SimpleDateFormat("dd-MM-yyyy");
      String DOB=dFormat.format(dob.getDate());
```

```
        String MobileNo=mobileno.getText();

        String gender=(String)Gender.getSelectedItem();

        String email=Email.getText();

        String bloodgroup=(String)BG.getSelectedItem();

        String city=City.getText();

        String address=Address.getText();

try{

            Connection con=ConnectionProvider.getCon();

            Statement st=con.createStatement();

            st.executeUpdate("insert into donor
values('"+donorId+"','"+name+"','"+fatherName+"','"+motherName+"','"+DOB+"','"+
MobileNo+"','"+gender+"','"+email+"','"+bloodgroup+"','"+city+"','"+address+"')");

            JOptionPane.showMessageDialog(null,"SUCCESSFULLY ADDED");

            setVisible(false);

            new addNewDonors().setVisible(true);

        }catch(Exception e){

            JOptionPane.showMessageDialog(null, e);

        }

    }

 private void formComponentShown(java.awt.event.ComponentEvent evt) {

        try {

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

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3307/bbms","root","");

            Statement s = con.createStatement();

            ResultSet rs = s.executeQuery("SELECT MAX(donorId) FROM donor");

            rs.next();

            rs.getString("MAX(donorId)");

            if (rs.getString("MAX(donorId)") == null) {

                newid.setText("1");

            } else {
```

```
        int id=rs.getInt("MAX(donorId)");
        id=id+1;
        String str=String.valueOf(id);
        newid.setText(String.format(str));
      }
    } catch (Exception ex) {
      ex.printStackTrace();
    }
  }
```

Update Donor:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String donorId=jTextField1.getText();
    String name=jTextField2.getText();
    String fatherName=jTextField3.getText();
    String motherName=jTextField4.getText();
    String DOB=jTextField5.getText();
    String MobileNo=jTextField6.getText();
    String gender=jTextField7.getText();
    String email=jTextField8.getText();
    String bloodGroup=jTextField9.getText();
    String city=jTextField10.getText();
    String address=jTextArea1.getText();
     try
    {
      Connection con=ConnectionProvider.getCon();
      Statement st=con.createStatement();
      st.executeUpdate("update donor set
name='"+name+"',fatherName='"+fatherName+"',motherName='"+motherName+"',D
OB='"+DOB+"',MobileNo='"+MobileNo+"',gender='"+gender+"',email='"+email+"',b
loodGroup='"+bloodGroup+"',city='"+city+"',address='"+address+"' where
donorId='"+donorId+"'");
            JOptionPane.showMessageDialog(null,"SUCCESSFULLY UPDATED");
            setVisible(false);
            new updateDetailsDonor().setVisible(true);
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,e);
        }
    }
```

All Donor Details:

```
private void formComponentShown(java.awt.event.ComponentEvent evt) {
      try
        {
           Connection con=ConnectionProvider.getCon();
           Statement st=con.createStatement();
           ResultSet rs=st.executeQuery("select *from donor");
           jTable1.setAutoResizeMode(jTable1.AUTO_RESIZE_OFF);
           jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,e);
        }
    }
```

Search Donor:

```
private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {
      String location=jTextField1.getText();
       try
       {
           Connection con=ConnectionProvider.getCon();
           Statement st=con.createStatement();
           ResultSet rs=st.executeQuery("select *from donor where city like
'%"+location+"%' or address like '%"+location+"%'");
           jTable1.setAutoResizeMode(jTable1.AUTO_RESIZE_OFF);
           jTable1.setModel(DbUtils.resultSetToTableModel(rs));
       }
       catch(Exception e)
       {
           JOptionPane.showMessageDialog(null,e);
       }
    }
```

Stock:

```
private void formComponentShown(java.awt.event.ComponentEvent evt) {
        try
       {
           Connection con=ConnectionProvider.getCon();
           Statement st=con.createStatement();
           ResultSet rs=st.executeQuery("select * from stock");
           jTable1.setModel(DbUtils.resultSetToTableModel(rs));
       }
       catch(Exception e)
       {
           JOptionPane.showMessageDialog(null,e);
       }
    }
```

```
  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
           String bloodGroup=(String) jComboBox1.getSelectedItem();
           String units=jTextField1.getText();
           int unit1=Integer.parseInt(units);
    try
    {
      Connection con=ConnectionProvider.getCon();
      Statement st=con.createStatement();
      st.executeUpdate("update stock set units=units-'"+unit1+"'where bloodGroup='"+bloodGroup+"'");
      JOptionPane.showMessageDialog(null,"Successfully updated");
      setVisible(false);
      new stockDecrease().setVisible(true);
    }
    catch(Exception e)
    {
      JOptionPane.showMessageDialog(null,e);
    }
  }
```

Stock Details:

```
private void formComponentShown(java.awt.event.ComponentEvent evt) {
    try
    {
      Connection con=ConnectionProvider.getCon();
      Statement st=con.createStatement();
      ResultSet rs=st.executeQuery("select * from stock");
      jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    }
```

```
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,e);
        }
    }
```

Delete Donor:

```
private void searchActionPerformed(java.awt.event.ActionEvent evt) {
        String donorId=jTextField1.getText();
        try {
                Connection con=ConnectionProvider.getCon();
                Statement st=con.createStatement();
                ResultSet rs=st.executeQuery("select *from donor where donorId='"+donorId+"'");
                if (rs.next())
                {
                    jTextField2.setText(rs.getString(2));
                    jTextField3.setText(rs.getString(3));
                    jTextField4.setText(rs.getString(4));
                    jTextField5.setText(rs.getString(5));
                    jTextField6.setText(rs.getString(6));
                    jTextField7.setText(rs.getString(7));
                    jTextField8.setText(rs.getString(8));
                    jTextField9.setText(rs.getString(9));
                    jTextField10.setText(rs.getString(10));
                    jTextArea1.setText(rs.getString(11));
                    jTextField1.setEditable(false);
                }
                else
                {
```

```
              JOptionPane.showMessageDialog(null,"DonorId does not Exist");
            }
        }
        catch (Exception ex) {
        ex.printStackTrace();
    }


}


private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
    String donorId=jTextField1.getText();
     try
    {
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        st.executeUpdate("delete from donor where donorId='"+donorId+"'");
        JOptionPane.showMessageDialog(null," Deleted SUCCESSFULLY");
        setVisible(false);
        new deleteDonor().setVisible(true);
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}
```

Exit Application:

```
private void exitapplicationActionPerformed(java.awt.event.ActionEvent evt) {
         int  a=JOptionPane.showConfirmDialog(null,"DO  YOU  REALLY  WANT  TO
CLOSE APPLICATION","SELECT",JOptionPane.YES_NO_OPTION);
        if(a==0)
```

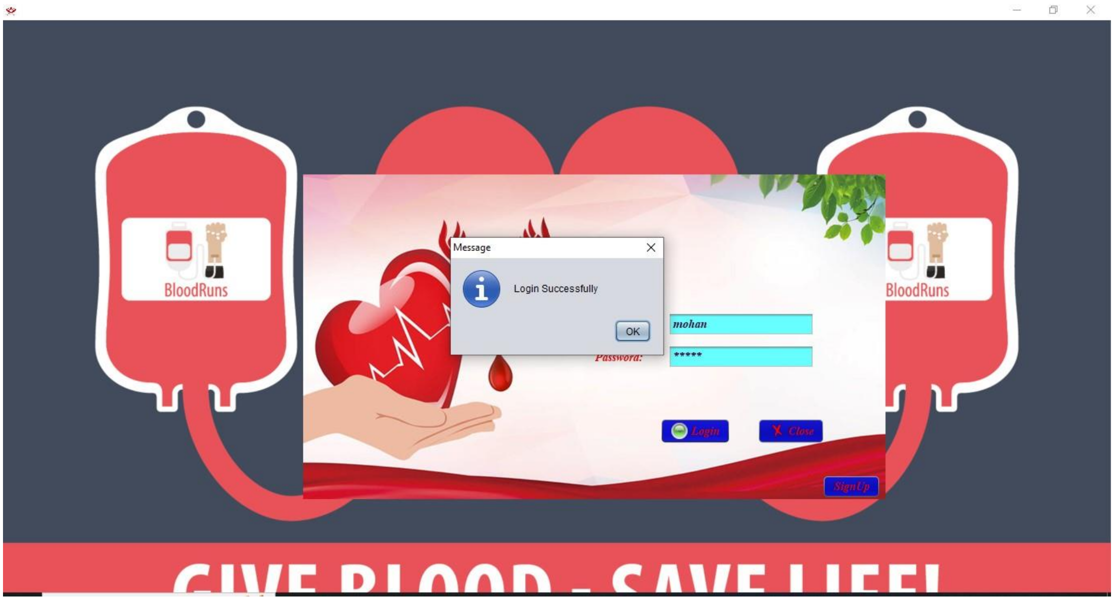
```
    { private void exitapplicationActionPerformed(java.awt.event.ActionEvent evt) {
   int a=JOptionPane.showConfirmDialog(null,"DO YOU REALLY WANT TO
CLOSE APPLICATION","SELECT",JOptionPane.YES_NO_OPTION);
    if(a==0)
    {
      System.exit(0);
    }


  }


  private void logoutActionPerformed(java.awt.event.ActionEvent evt) {         int
a=JOptionPane.showConfirmDialog(null,"DO YOU REALLY WANT TO
LOGOUT","SELECT",JOptionPane.YES_NO_OPTION)
 if(a==0)
    {
       setVisible(false);
       new login().setVisible(true);
    }
  }
```

5.SYSTEM TESTING

System testing is the process of evaluation and software item to detect differences between given input and expected output.Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

5.1. Unit Testing:

Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.



Message

Login Successfully

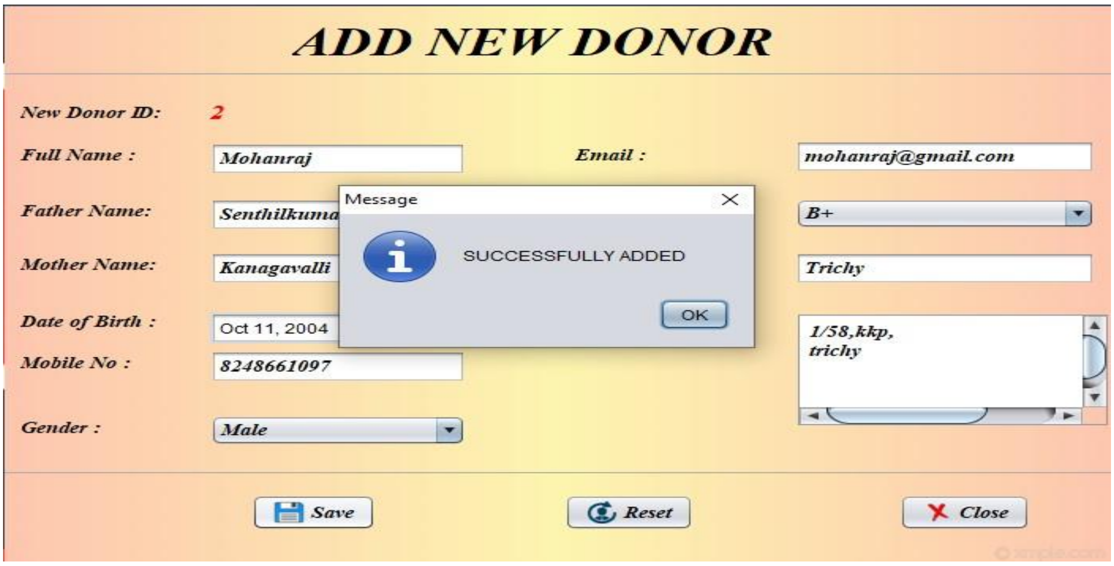
mohan

Password: *****

Fig:5.1 Login Page

5.2. Integration Testing:

Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.



ADD NEW DONOR

New Donor ID: 2

Full Name : Mohanraj

Email : mohanraj@gmail.com

Father Name: Senthilkuma

Mother Name: Kanagavalli

Date of Birth : Oct 11, 2004

Mobile No : 8248661097

Gender : Male

B+

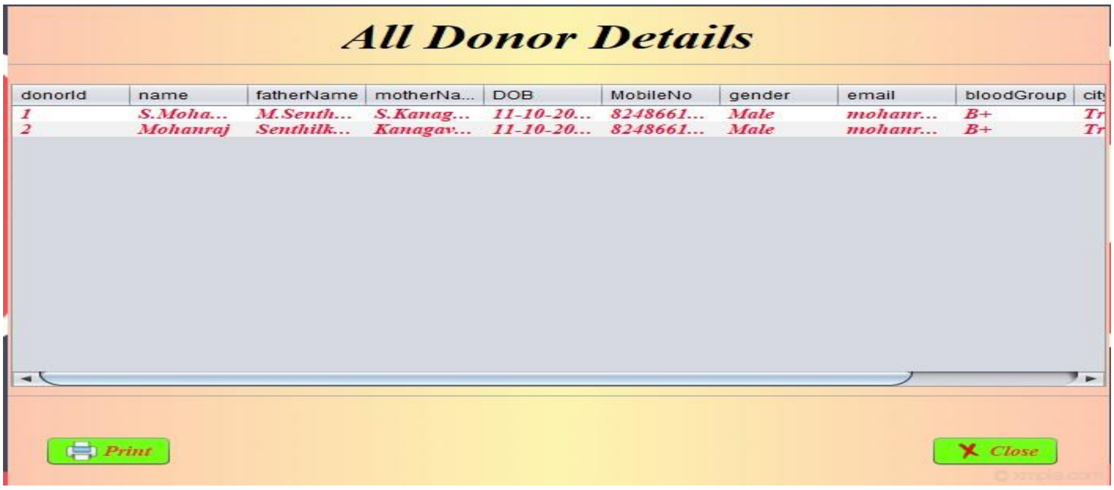
Trichy

1/58,kkp,
trichy

Message

SUCCESSFULLY ADDED

Fig:5.2 Add Donor Output Page



All Donor Details

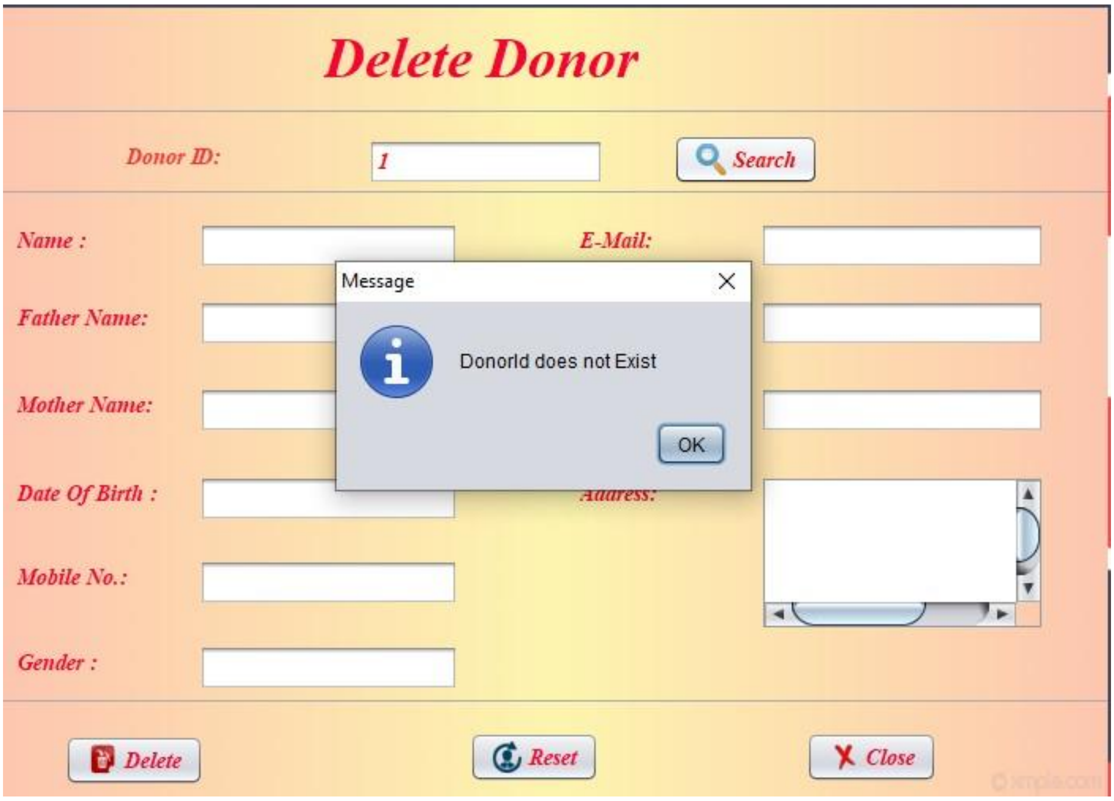
donorId	name	fatherName	motherNa...	DOB	MobileNo	gender	email	bloodGroup	city
1	S.Moha...	M.Senth...	S.Kanag...	11-10-20...	8248661...	Male	mohanr...	B+	Tr
2	Mohanraj	Senthilk...	Kanagav...	11-10-20...	8248661...	Male	mohanr...	B+	Tr

Fig:5.3 Donor Details Output Page

5.3. Validation Testing:

The process of validating a software product, or determining if it meets high level criteria, involves determining whether it is up to par. It is the procedure used to verify that the product we are producing is the proper one. The actual and anticipated product are being validated.

Testing in motion is validation.



Delete Donor

Donor ID: 1

Message

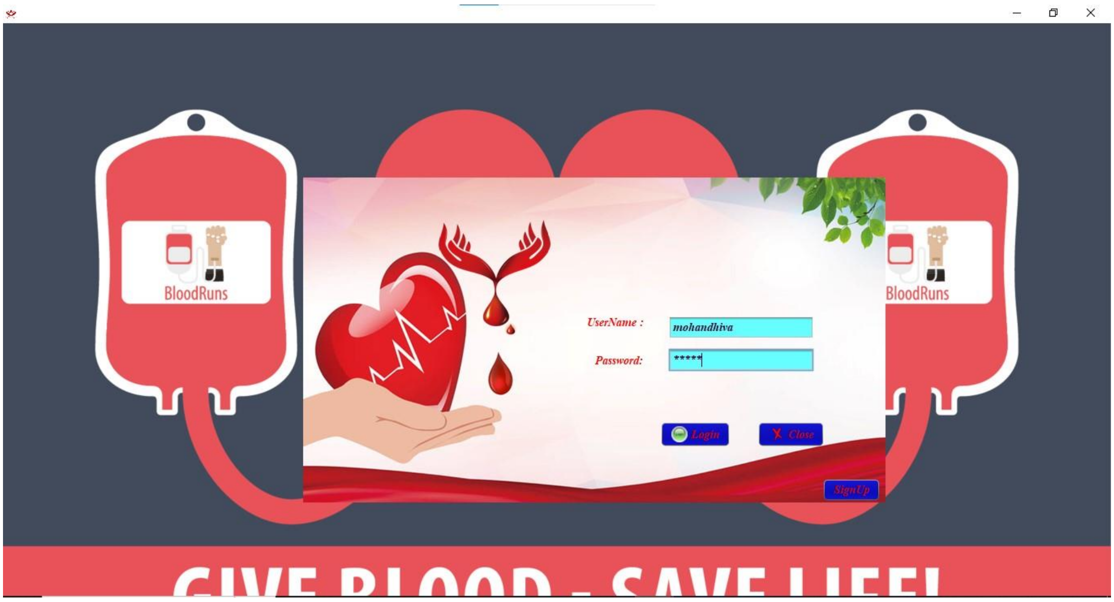
Donorld does not Exist

Fig:5.4 Delete donor Validation

6. SOFTWARE DEMONSTRATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considering the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

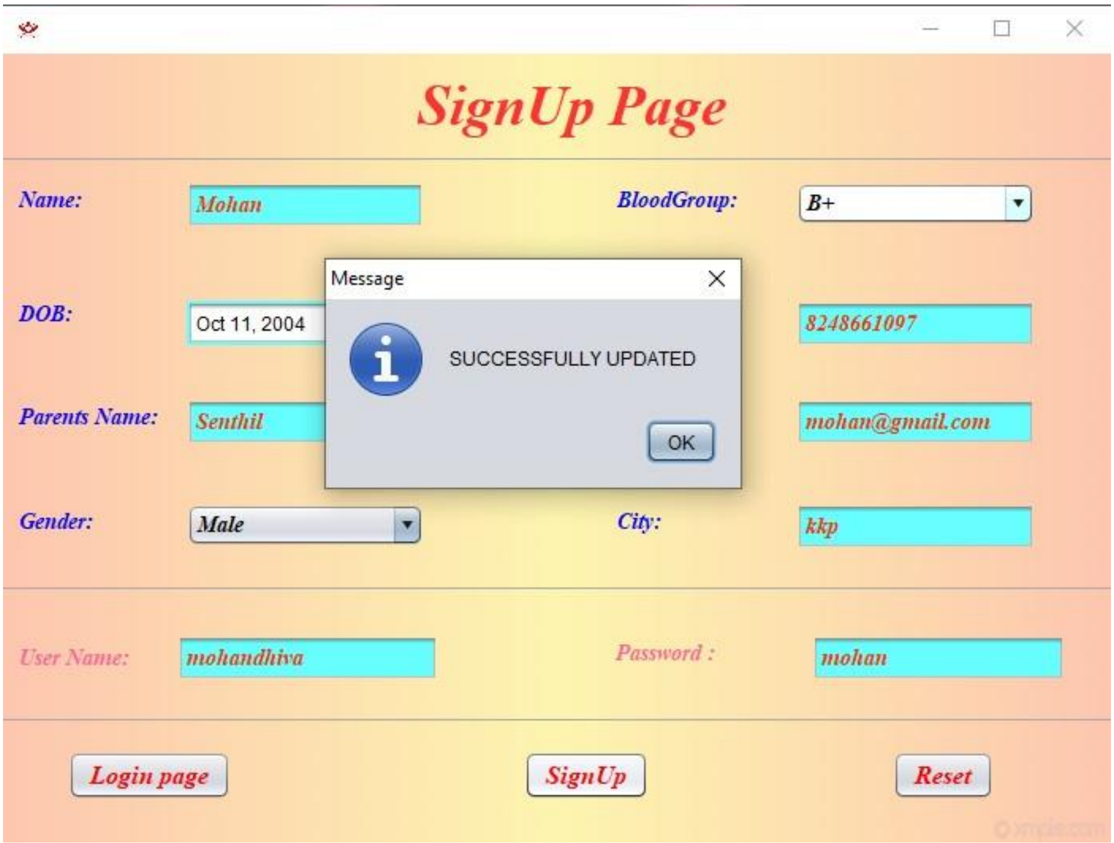
6.1 Demonstration



UserName : mohandhiva

Password: *****

Fig:6.1 Login Page



SignUp Page

Name: Mohan

BloodGroup: B+

DOB: Oct 11, 2004

8248661097

Parents Name: Senthil

mohan@gmail.com

Gender: Male

City: kkp

User Name: mohandhiva

Password : mohan

Message

SUCCESSFULLY UPDATED

Fig:6.2 SignUp Page

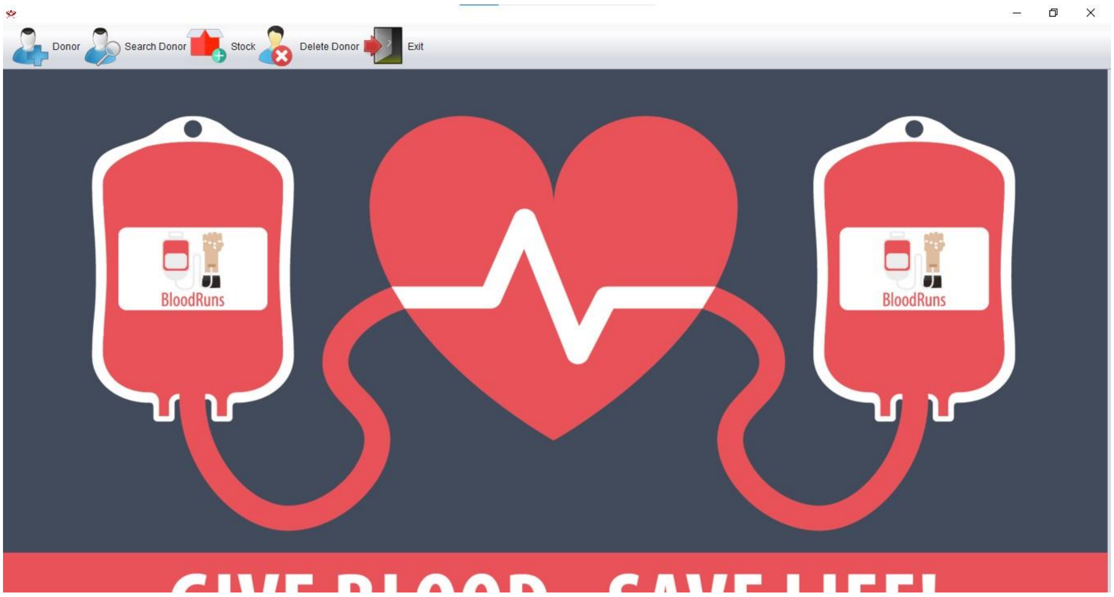
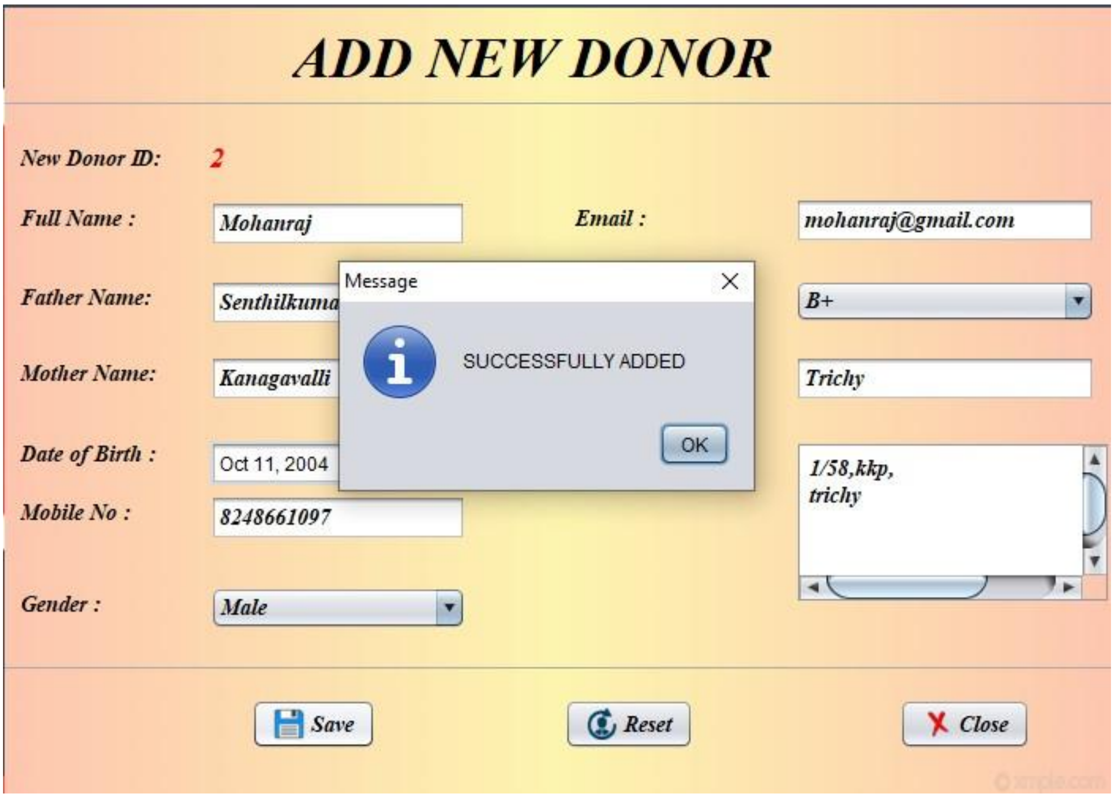


Fig:6.3 Home Page



ADD NEW DONOR

New Donor ID: 2

Full Name : Mohanraj

Email : mohanraj@gmail.com

Father Name: Senthilkuma

B+

Mother Name: Kanagavalli

Trichy

Date of Birth : Oct 11, 2004

1/58,kkp,
trichy

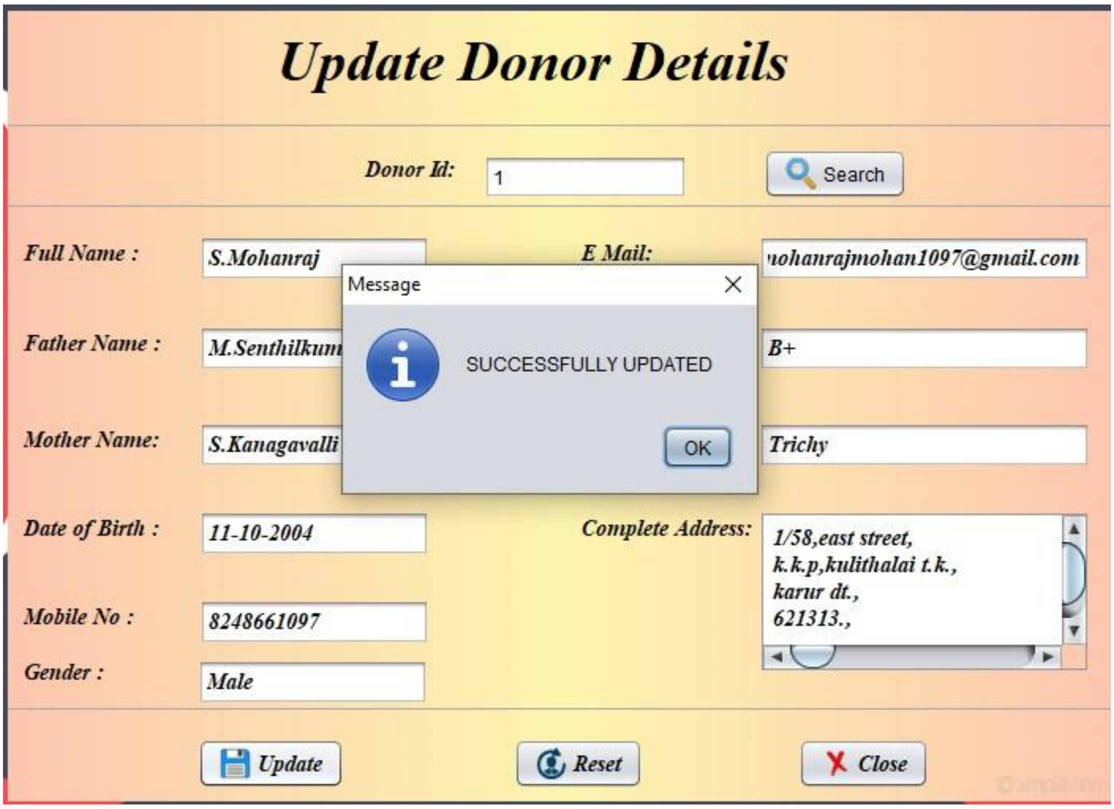
Mobile No : 8248661097

Gender : Male

Message

SUCCESSFULLY ADDED

Fig:6.4 Add New Donor



Update Donor Details

Donor Id: 1

Full Name : S.Mohanraj

E Mail: ıohanrajmohan1097@gmail.com

Father Name : M.Senthilkum

B+

Mother Name: S.Kanagavalli

Trichy

Date of Birth : 11-10-2004

Complete Address: 1/58,east street,
k.k.p,kulithalai t.k.,
karur dt.,
621313.,

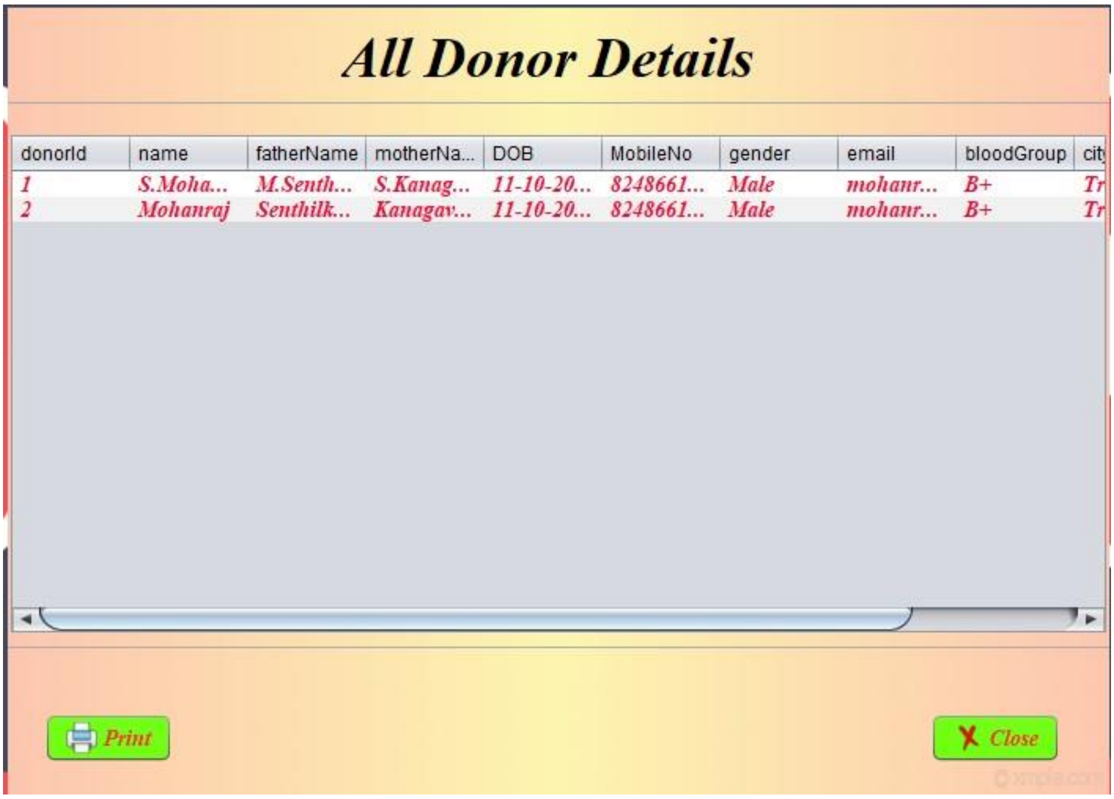
Mobile No : 8248661097

Gender : Male

Message

SUCCESSFULLY UPDATED

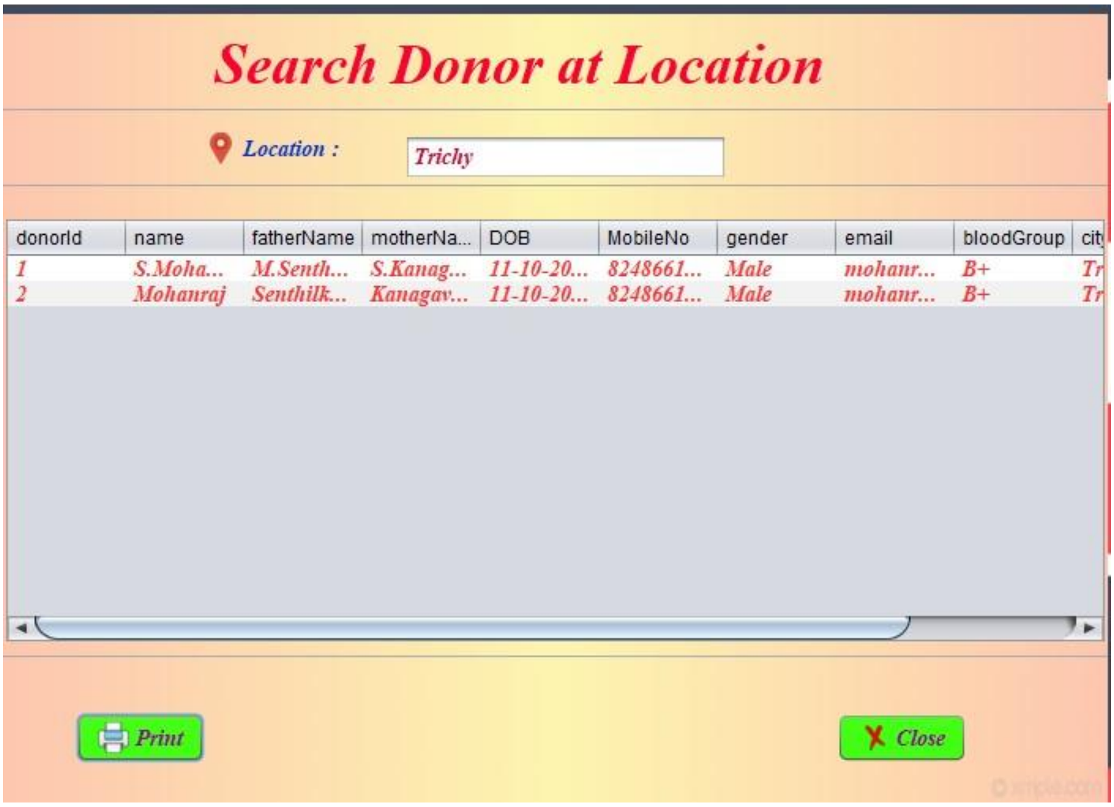
Fig:6.5 Update Donor



All Donor Details

donorId	name	fatherName	motherNa...	DOB	MobileNo	gender	email	bloodGroup	cit
1	S.Moha...	M.Senth...	S.Kanag...	11-10-20...	8248661...	Male	mohanr...	B+	Tr
2	Mohanraj	Senthilk...	Kanagav...	11-10-20...	8248661...	Male	mohanr...	B+	Tr

Fig:6.6 Donor Details

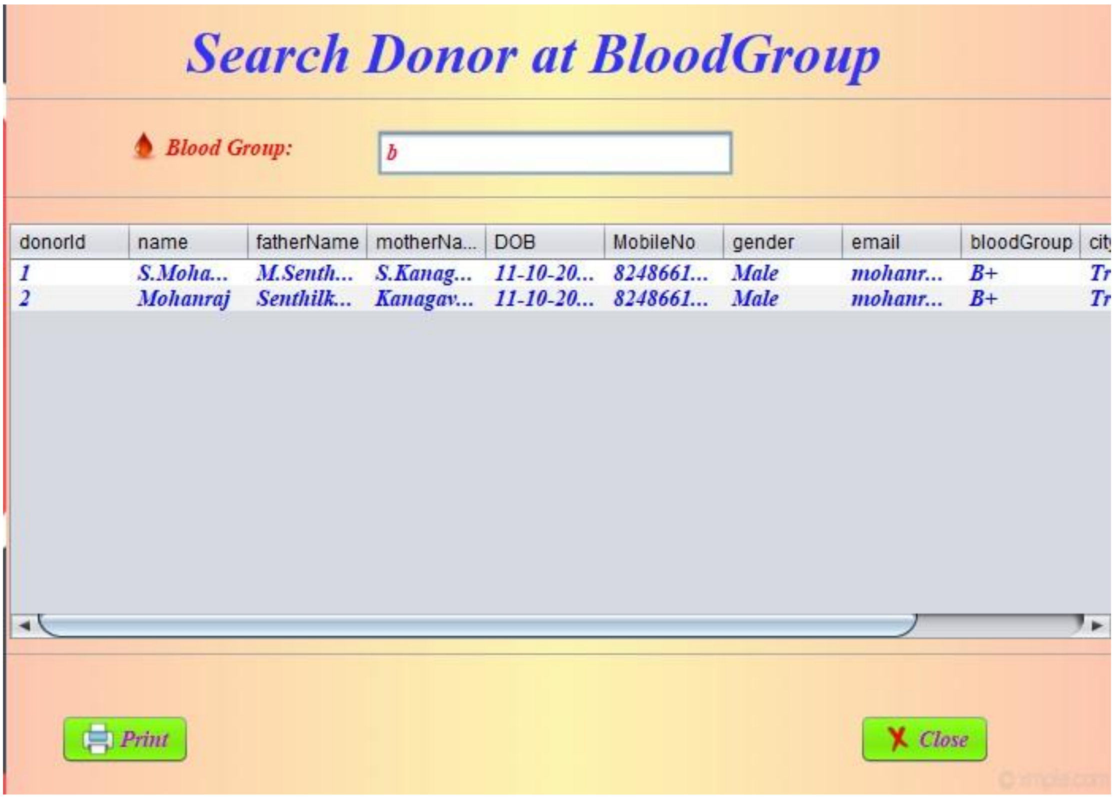


Search Donor at Location

Location : Trichy

donorId	name	fatherName	motherNa...	DOB	MobileNo	gender	email	bloodGroup	cit
1	S.Moha...	M.Senth...	S.Kanag...	11-10-20...	8248661...	Male	mohanr...	B+	Tr
2	Mohanraj	Senthilk...	Kanagav...	11-10-20...	8248661...	Male	mohanr...	B+	Tr

Fig:6.6 Search Donor By Location

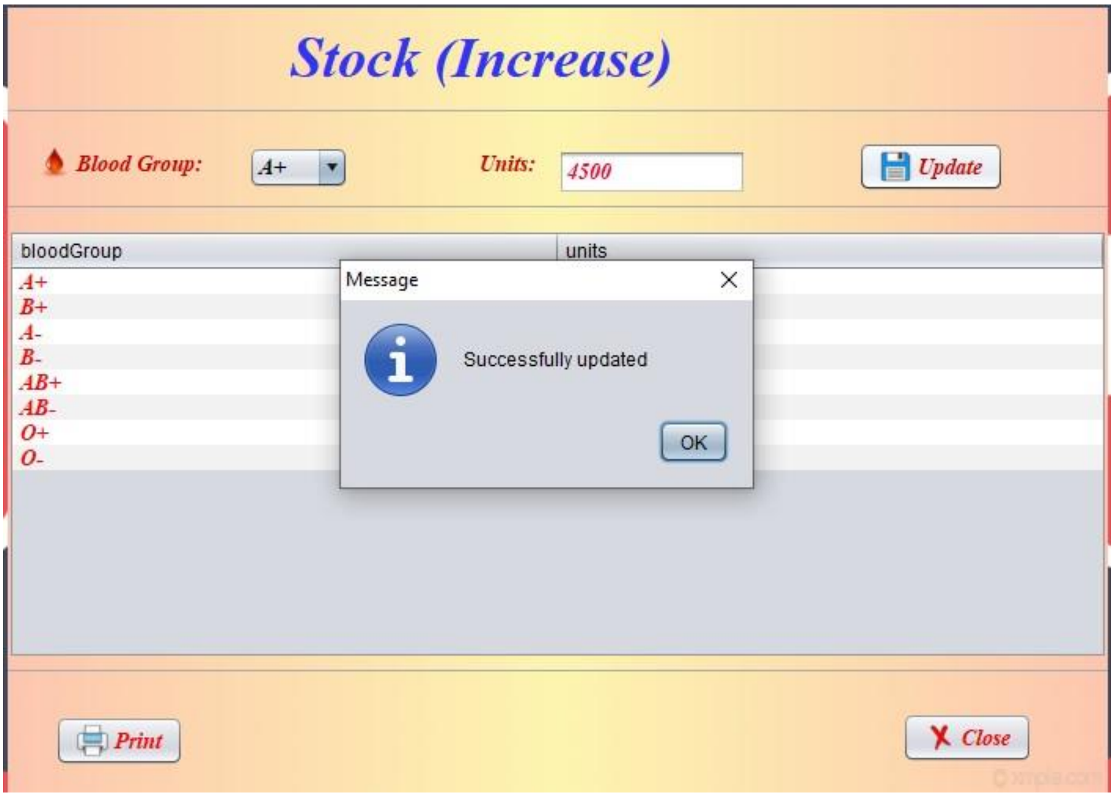


Search Donor at BloodGroup

Blood Group: b

donorId	name	fatherName	motherNa...	DOB	MobileNo	gender	email	bloodGroup	city
1	S.Moha...	M.Senth...	S.Kanag...	11-10-20...	8248661...	Male	mohanr...	B+	Tr
2	Mohanraj	Senthilk...	Kanagav...	11-10-20...	8248661...	Male	mohanr...	B+	Tr

Fig:6.7 Search Donor By Bloodgroup



Stock (Increase)

Blood Group: A+

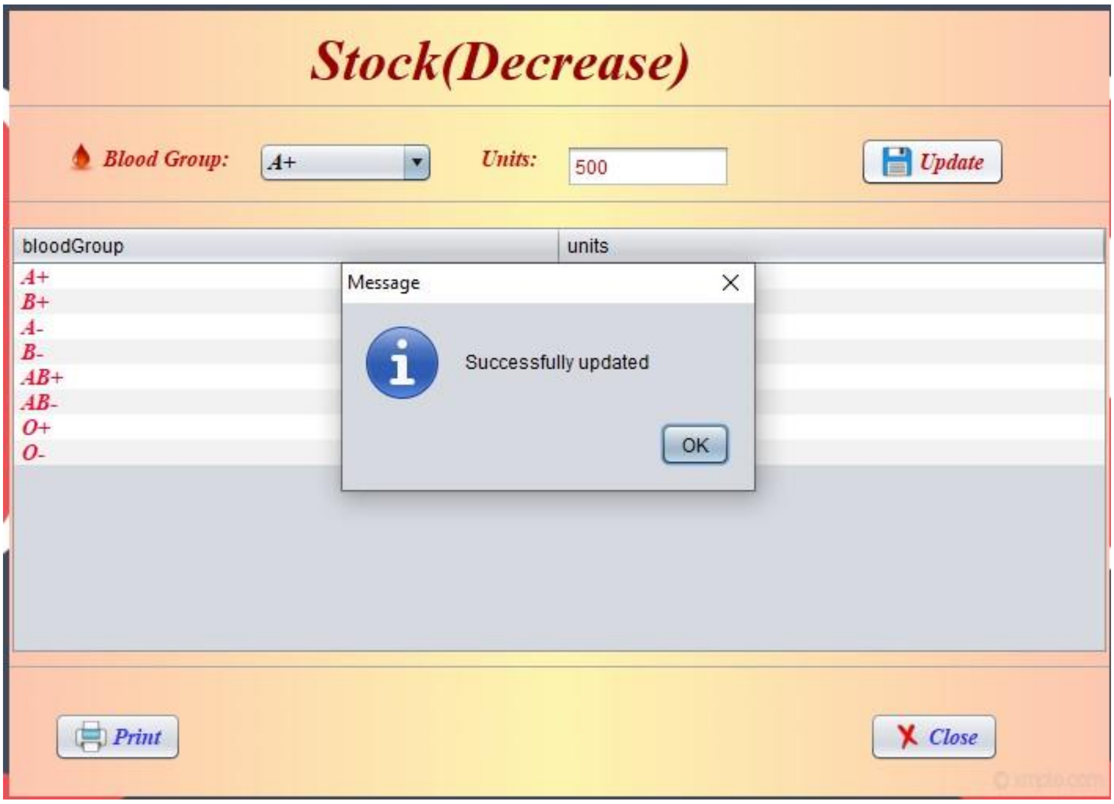
Units: 4500

bloodGroup	units
A+	
B+	
A-	
B-	
AB+	
AB-	
O+	
O-	

Message

Successfully updated

Fig:6.8 Stock Increase



Stock(Decrease)

Blood Group: A+

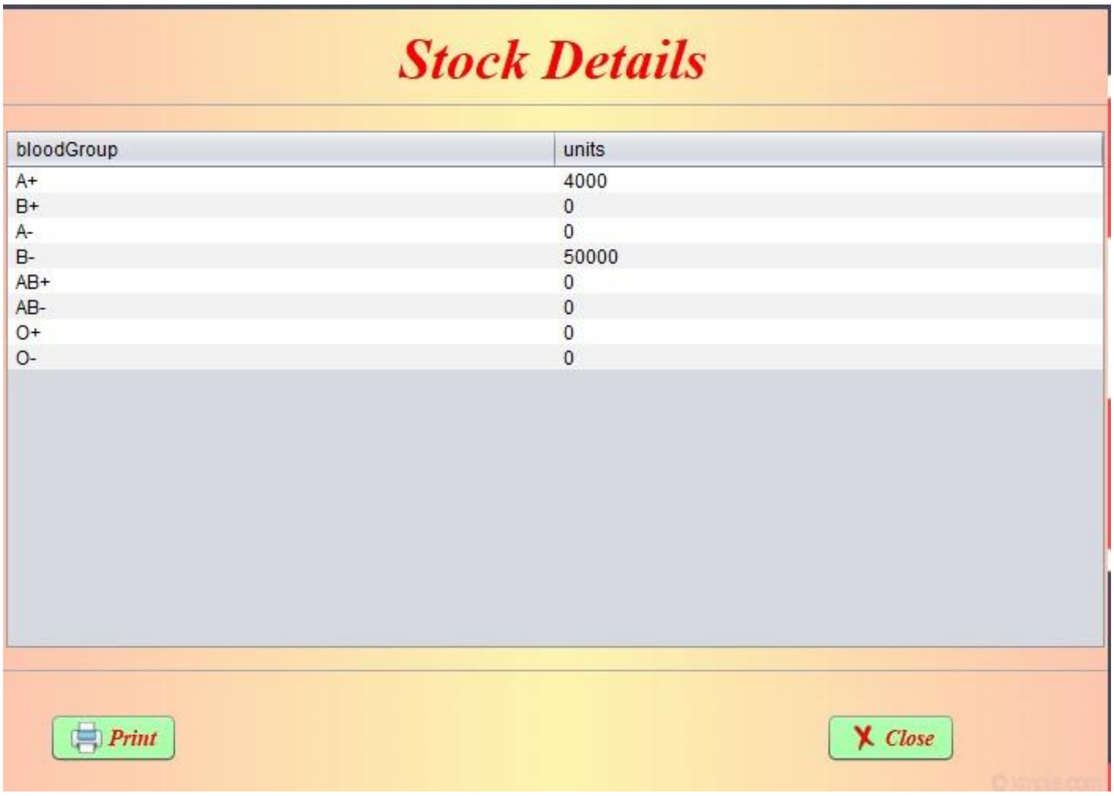
Units: 500

bloodGroup	units
A+	
B+	
A-	
B-	
AB+	
AB-	
O+	
O-	

Message

Successfully updated

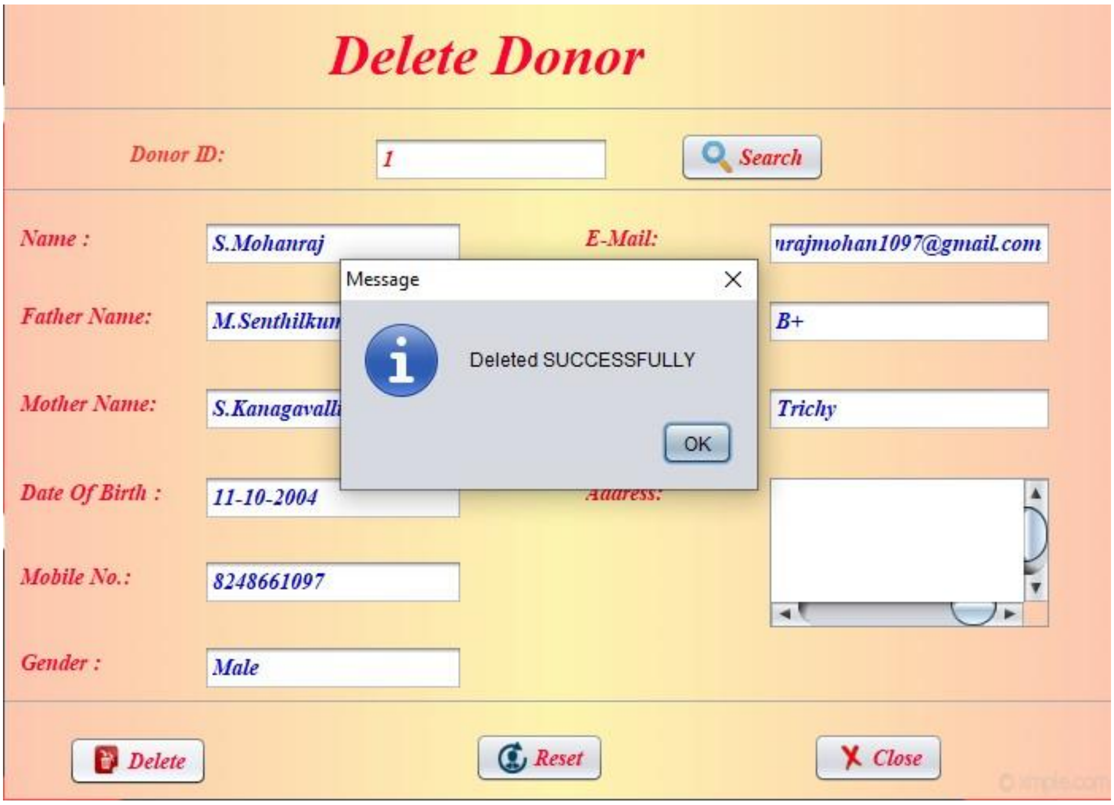
Fig:6.9 Stock Decrease



Stock Details

bloodGroup	units
A+	4000
B+	0
A-	0
B-	50000
AB+	0
AB-	0
O+	0
O-	0

Fig:6.10 Stock Details



Delete Donor

Donor ID: 1

Name : S.Mohanraj

E-Mail: urajmohan1097@gmail.com

Father Name: M.Senthilkum

B+

Mother Name: S.Kanagavalli

Trichy

Date Of Birth : 11-10-2004

Address:

Mobile No.: 8248661097

Gender : Male

Message

Deleted SUCCESSFULLY

Fig:6.11 Delete Donor

7.CONCLUSION

In "Blood Bank Management System" admin can maintain donor details very easily. The Blood bank management system is a very helpful for patients who need blood and admin who work in the blood bank. This project is platform independent so it can be used in any operating system. Also, we wish to concentrate on the future to use web applications to make it a better and more effective product. Future plans will include new features to improve user productivity and ease of use.

BIBLIOGRAPHY

BOOK REFERENCE:

1. Muthu C, ”Programming in Java”.
2. Atul Kahate”Introduction to DBMS”.
3. Abraham Silbeuhanth,Henry korth &S.Sudhanson”DBMS Concepts”.

WEBSITE REFRENCES:

1. https://www.programiz.com/Java-programming
2. https://www.freecodecamp.org/news/what-is-the-Java-programminglanguage-beginner-tutorial/
3. https://www.\geeksforgeeks.org/Java-language-set-1-introduction/