

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



A mini project report on

## *Blood Bank Management System*

*Submitted in partial fulfillment of the requirements as a part of the DBMS Lab for the award of degree of*

## **Bachelor of Engineering in Information Science and Engineering**

by

**P.Yuktha  
1RN16IS061**

**Faculty Incharge**

**Mrs. Hema N**

Assistant professor  
Dept. of ISE, RNSIT

**Lab Incharge**

**Mr. R Rajkumar**

Assistant professor  
Dept. of ISE, RNSIT



**Department of Information Science and Engineering  
RNS Institute of Technology**

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,  
Bengaluru – 560 098

2018 – 2019

# **RNS Institute of Technology**

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,  
Bengaluru – 560 098

## **DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**



### **CERTIFICATE**

This is to certify that the Mini project report entitled **BLOOD BANK MANAGEMENT SYSTEM** has been successfully completed by **P.YUKTHA** bearing USN **1RN16IS061**, presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree **Bachelor of Engineering in Information Science and Engineering** under **Visvesvaraya Technological University, Belagavi** during academic year 2018 – 2019. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

---

**Mrs. Hema N**  
Faculty Incharge

---

**Mr.R Rajkumar**  
Lab Incharge

---

**Dr. M V Sudhamani**  
Professor and HoD

#### **External Viva**

**Name of the Examiners**

1. \_\_\_\_\_

2. \_\_\_\_\_

**Signature with date**

\_\_\_\_\_

\_\_\_\_\_

# **ABSTRACT**

The purpose of the Blood Bank Management System is to automate the existing system by the help of full-fledged computer software fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy manipulation of the same. The required software and hardware are easily available to work with. The main objective of the project on Blood bank management system is to manage the details of Blood bank, blood group, donor, blood stock, blood requests and transactions. 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. Moreover if any general consumer wants to make request blood online he can also take the help of this site.

Blood bank management system, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on other activities other than record keeping. Thus it helps organisations in better utilization of resources. The organization can maintain computerized records without redundant entries. The purpose of this project is to reduce the manual work for managing the blood bank, blood group, donor, blood stock, blood requests and transactions. Basically the project describes how to manage for good performance and better services for the clients.

## ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

I would like to profoundly thank **Management of RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

I would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

I would like to express my thanks to our Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

I wish to place on record my words of gratitude to **Dr. M V Sudhamani**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my Project work.

I would like to express my profound and cordial gratitude to my Faculty incharge **Mr. R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance, constructive comments and continuous encouragement throughout the Project work.

I would like to express my profound and cordial gratitude to my guide **Mrs. Hema N**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance in preparing Project report.

I would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

And lastly, I would hereby acknowledge and thank my parents who have been a source of inspiration and also instrumental in carrying out this Project work.

**P.YUKTHA**

**USN: 1RN16IS061**

# TABLE OF CONTENTS

<b>CERTIFICATE</b>	
<b>ABSTRACT</b>	<b>iii</b>
<b>ACKNOWLEDGMENT</b>	<b>iv</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>vi</b>
<b>ABBREVIATIONS</b>	<b>vii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Background	1
1.2 Introduction about the project	2
<b>2. E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM</b>	<b>3</b>
<b>3. SYSTEM DESIGN</b>	<b>5</b>
3.1 Tables description	5
<b>4. IMPLEMENTATION</b>	<b>8</b>
4.1 Requirements	8
4.2 Discussion of code segments	14
4.3 Applications of project work	19
4.4 Discussion of the results	20
<b>5. CONCLUSION AND FUTURE ENHANCEMENTS</b>	<b>31</b>
<b>REFERENCES</b>	<b>32</b>

# LIST OF FIGURES

Fig. No.	Description	Page
Fig. 2.1	E R Diagram	03
Fig. 2.2	Relational schema diagram	04
Fig. 3.1	Blood table	05
Fig. 3.2	Branch table	05
Fig. 3.3	Donor table	06
Fig. 3.4	Requests table	06
Fig. 3.5	Stock table	06
Fig. 3.6	Transaction table	07
Fig. 3.7	Rating table	07
Fig 4.1	Start page of the application	20
Fig 4.2	Home page	21
Fig 4.3	Login page	21
Fig 4.4	Admin login	22
Fig 4.5	Admin home page	22
Fig 4.6	Admin blood bank portal-add blood bank	23
Fig 4.7	Admin blood bank portal-delete blood bank	23
Fig 4.8	Admin blood bank portal-update blood banks	24
Fig 4.9	Admin blood bank portal-view blood banks	24
Fig 4.10	Admin donor portal	25
Fig 4.11	Admin requests portal	25
Fig 4.12	Branch manager portal-home page	26
Fig 4.13	Branch manager portal-transactions page	26
Fig 4.14	Branch manager portal-stock information	27
Fig 4.15	Donor portal-home page	27
Fig 4.16	Donor portal-donor transactions	28
Fig 4.17	Requester portal	28
Fig 4.18	View donor information	29
Fig 4.19	View requests information	29
Fig 4.20	View available blood banks	30

## **ABBREVIATIONS**

ERD	-	Entity Relationship Diagram
SQL	-	Structured Query Language
IIS	-	Internet Information Services
Xampp	-	Cross platform(X), Apache(A), MySQL(M), PHP(P) and Perl(P)
RAM	-	Random Access Memory
HDD	-	Hard Disk Drive
JVM	-	Java Virtual Machine
OOP	-	Object Oriented Programming
ANSI	-	American National Standard Institute
JDBC	-	Java Data Base Connection
API	-	Application Program Interface
JSP	-	Java Servlet Package
EJB	-	Enterprise Java Bean
ODBC	-	Open Data Base Connectivity
HTTP	-	Hyper Text Transfer Protocol
PHP	-	Personal Home Page
URL	-	Uniform Resource Locator
GUI	-	Graphical User Interface
CGI	-	Common Gateway Interface
PL/SQL	-	Procedural Language extension to structured query language

# Chapter 1

## INTRODUCTION

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received have to be managed thoroughly so that there will be no negative effect to the blood receiver once they received blood.

### 1.1 Background

At present, the public can only know about the blood donation events through conventional media means such as radio, news paper or television advertisements. There is no information regarding the blood donation programs available on any of the portal.

The current system that is using by the blood bank is manual system. With the manual system, there are problems in managing the donors' records. The records of the donor might not be kept safely and there might be missing of donor's records due to human error or disasters.

Besides that, errors might occur when the staff keeps more than one record for the same donor. There is no centralized database of volunteer donors. So, it becomes really tedious for a person to search blood in case of emergency. The only option is to manually search and match donors and then make phone calls to every donor. There is also no centralized database used to keep the donors' records. Each bank is having their own records of donors. If a donor makes donation in different hospital, no previous records can be traced except if the donor brings along the donation certificate. Hence, the donor is considered to be a first-timer if they make blood donation in a new place. Without an automated management system, there are also problems in keeping track of the actual amount of each and every blood type in the blood bank.

A blood bank is a cache or bank of blood or blood components, gathered as a result of blood donation or collection, stored and preserved for later use in blood transfusion. The term "blood bank" typically refers to a division of a hospital where the storage of blood product occurs and where proper testing is performed (to reduce the risk of transfusion related adverse events). However, it sometimes refers to a collection center, and indeed some hospitals also perform collection.



## 1.2 Introduction about the project

The project Blood Bank Management System is known to be a pilot project that is designed for various blood banks to gather blood from various sources and distribute it to the needy people who have high requirements for it. The software application is designed to handle the daily transactions of the blood bank and search the details when required.

This blood bank management system can be used for getting the details of blood donors effectively. This project is available to everyone easily. A person who likes to donate blood gives 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 any changes in his information given before.

The goal of this system is to ensure the safe and efficient use of the many resources involves in the complex process of blood donation. It also helps the people who are in the need of the blood. They can register their username and password similar to as that of donor. They can also see the donors available and their blood group and details so that they can contact the donors by their phone numbers and other information provided. Such that person gets help from us which saves his life.

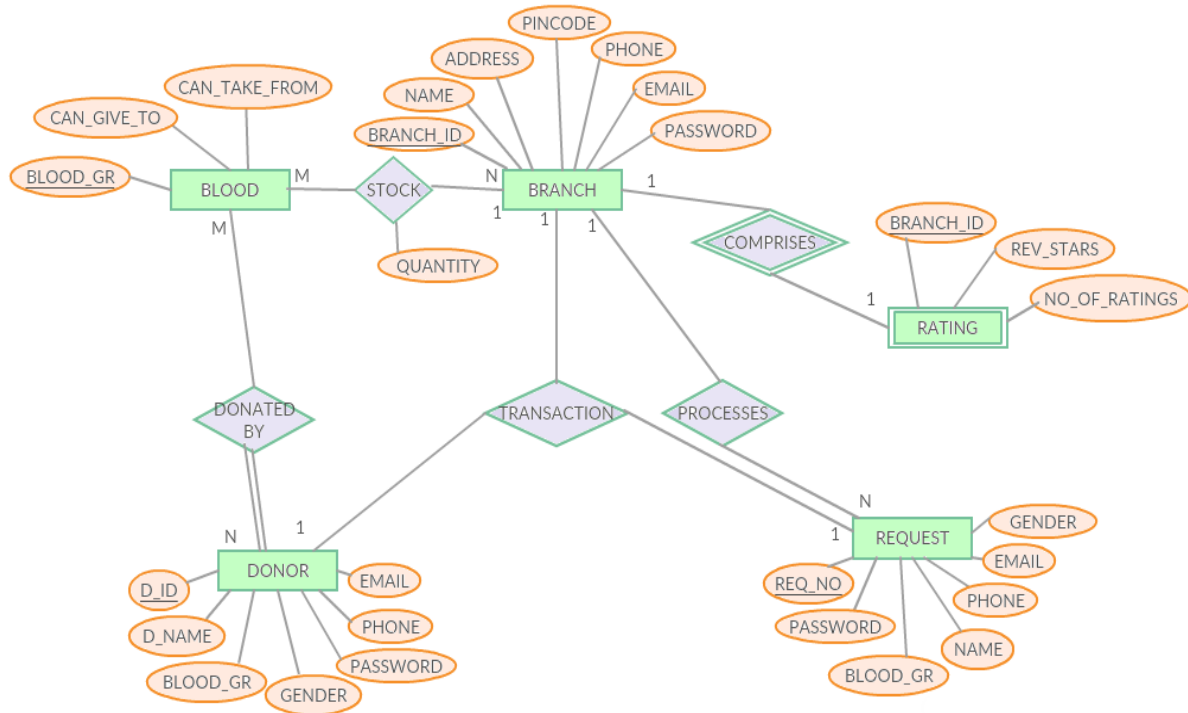
This system also has the ability to keep track of the donor's donation records and the blood stock in the blood bank. This project intends to computerize the blood and donor management system in a blood bank in order to improve the record management efficiency due to the grown size of records of data.

## Chapter 2

# E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM

### E R Diagram:

An entity relationship model, also called an entity-relationship (**ER**) **diagram**, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within **databases** or information systems. An entity is a piece of data-an object or concept about which data is stored.



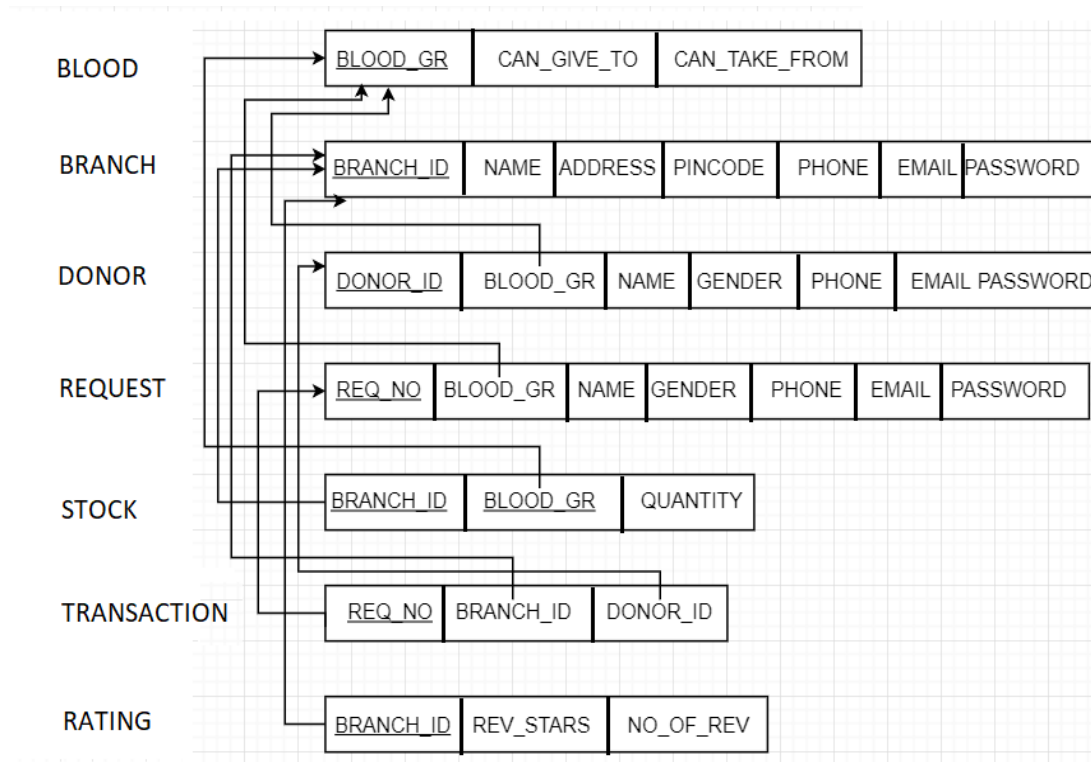
**Fig 2.1: E R Diagram**

**Entities:** Blood, Branch, Rating, Donor, Request

**Attributes:** Blood\_gr, can\_give\_to, can\_take\_from, branch\_id, Name, Address, Pincode, Phone, Email, Password, D\_id, Gender, Quantity, Rev\_stars, No\_of\_ratings

## Relational Schema Diagram:

A database schema represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of formulas known as integrity constraints that govern a database. These formulas are expressed in a data definition language, such as SQL.



**Fig 2.2: Relational schema diagram**

## Chapter 3

# SYSTEM DESIGN

Here we discuss about various tables and their properties used in the project.

### 3.1 Tables Description

#### Blood table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
BLOOD	BLOOD_GR	VARCHAR2	3
BLOOD	CANGIVETO	VARCHAR2	25
BLOOD	CANTAKEFROM	VARCHAR2	25

**Fig 3.1: Blood table**

This table stores different blood groups namely A+, A-, B+, B-, AB+, AB-, O+, O- and their respective compatible blood groups for donating and receiving the blood respectively. Here blood group is the primary key.

#### Branch table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
BRANCH	B_ID	NUMBER	22
BRANCH	B_NAME	VARCHAR2	50
BRANCH	B_ADDRESS	VARCHAR2	100
BRANCH	PINCODE	NUMBER	6
BRANCH	B_PHONE	VARCHAR2	13
BRANCH	B_EMAIL	VARCHAR2	50
BRANCH	B_PASSWORD	VARCHAR2	6

**Fig 3.2: Branch table**

This table stores different blood banks that process the blood transaction. From this table, all details except the password is visible to all the people who use this application. A branch can be added to the table only by the admin. But branch managers can manage the respective blood bank by entering their respective passwords. Here branch Id represented as b\_id is the primary key.

## Donor table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
DONOR	D_PASSWORD	VARCHAR2	6
DONOR	D_ID	NUMBER	22
DONOR	BLOOD_GR	VARCHAR2	3
DONOR	D_NAME	VARCHAR2	25
DONOR	D_GENDER	VARCHAR2	6
DONOR	D_PHONE	VARCHAR2	14
DONOR	D_EMAIL	VARCHAR2	50

**Fig 3.3 Donor table**

This table contains the donor information. Any person who wants to register as donor should fill all the values and set a password. Here donor Id represented as d\_id is the primary key.

## Requests table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
REQUEST	REQ_NO	VARCHAR2	5
REQUEST	BLOOD_GR	VARCHAR2	3
REQUEST	R_NAME	VARCHAR2	25
REQUEST	R_GENDER	VARCHAR2	6
REQUEST	R_PHONE	VARCHAR2	12
REQUEST	R_EMAIL	VARCHAR2	50
REQUEST	R_PASSWORD	VARCHAR2	6

**Fig 3.4: Requests table**

This table contains the details about the people who had requested for blood. Here request number represented by req\_no is the primary key.

## Stock table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
STOCK	B_ID	NUMBER	22
STOCK	BLOOD_GR	VARCHAR2	3
STOCK	QUANTITY	NUMBER	3

**Fig 3.5: Stock table**

This table contains the stock information i.e., the quantity of particular blood group present in particular branch. Only blood bank manager has privileges to modify the contents of this table.

## Transaction table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
TRANSACTION	REQ_NO	VARCHAR2	5
TRANSACTION	B_ID	NUMBER	22
TRANSACTION	D_ID	NUMBER	22

**Fig 3.6: Transaction table**

This table contains the blood transaction information. Only blood bank manager has privileges to modify the contents of this table if and only if it is a valid transaction.

## Rating table

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
RATING	B_ID	NUMBER	22
RATING	REV_STARS	NUMBER	2,1
RATING	NO_OF_REV	NUMBER	3

**Fig 3.7: Rating table**

This table contains the average rating of all blood bank branches available here, It is viewed to all the users who use this application.

## Chapter 4

# IMPLEMENTATION

In this chapter , all the requirements and associated information are discussed in brief.

### 4.1 Requirements

#### 4.1.1 Resource requirement

##### Software requirements

Software used in the project are as follows

- Language used: JAVA(J2EE)
- Database: Oracle
- User interface design: Swings
- IDE: Netbeans

##### Hardware requirements

Hardware used in the project are as follows

- Processor: Intel i3
- Processor speed: 500MHz or above
- RAM: 1GB or more
- Storage space: Approx. 1GB

#### 4.1.2 JAVA

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This tutorial gives a complete understanding of Java. This reference will take you through simple and practical approaches while learning Java Programming language.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be **Write Once, Run Anywhere**.

Java is –

- **Object Oriented** – In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** – Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** – Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** – With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** – Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** – Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust** – Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded** – With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- **Interpreted** – Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
- **High Performance** – With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed** – Java is designed for the distributed environment of the internet.



- **Dynamic** – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

#### 4.1.3 JAVA DATABASE CONNECTION

JDBC stands for **J**ava **D**atabase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

```
import java.sql.*;
```

```
public class orconnect
```

```
{
```

```
    public static Connection connectdb()
```

```
    {
```

```
        Connection conn=null;
```

```
        try
```

```
        {
```

```
            String driverName="oracle.jdbc.driver.OracleDriver";
```

```
            Class.forName(driverName);
```

```
            conn = DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
```

```
                IUPKBLF:1521:xe","seenu","123");
```

```
            System.out.println("Successfully connected to the database");
```

```
        } catch (ClassNotFoundException e) {
```

```
            System.out.println("Could not find the database driver" + e.getMessage());
```

```
        } catch (SQLException e) {
```

```
            System.out.println("Could not connect to the database" + e.getMessage());
```

```
    }  
  
    return conn;  
  
}  
  
}
```

#### 4.1.4 JDBC

JDBC stands for **J**ava **D**atabase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

Fundamentally, JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as –

- Java Applications
- Java Applets
- Java Servlets
- Java ServerPages (JSPs)
- Enterprise JavaBeans (EJBs).

All of these different executables are able to use a JDBC driver to access a database, and take advantage of the stored data. JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

#### 4.1.5 Oracle database

Oracle database (Oracle DB) is a relational database management system (RDBMS) from the Oracle Corporation. Originally developed in 1977 by Lawrence Ellison and other developers, Oracle DB is one of the most trusted and widely-used relational database engines.

The system is built around a relational database framework in which data objects may be directly accessed by users (or an application front end) through structured query language (SQL). Oracle is a fully scalable relational database architecture and is often used by global enterprises, which manage and process data across wide and local area networks. The Oracle database has its own network component to allow communications across networks.

Oracle DB is also known as Oracle RDBMS and, sometimes, just Oracle.

Oracle DB rivals Microsoft's SQL Server in the enterprise database market. There are other database offerings, but most of these command a tiny market share compared to Oracle DB and SQL Server. Fortunately, the structures of Oracle DB and SQL Server are quite similar, which is a benefit when learning database administration.

Oracle DB runs on most major platforms, including Windows, UNIX, Linux and Mac OS. Different software versions are available, based on requirements and budget. Oracle DB editions are hierarchically broken down as follows:

- Enterprise Edition: Offers all features, including superior performance and security, and is the most robust
- Standard Edition: Contains base functionality for users that do not require Enterprise Edition's robust package
- Express Edition (XE): The lightweight, free and limited Windows and Linux edition
- Oracle Lite: For mobile devices

A key feature of Oracle is that its architecture is split between the logical and the physical. This structure means that for large-scale distributed computing, also known as grid computing, the data location is irrelevant and transparent to the user, allowing for a more modular physical structure that can be added to and altered without affecting the activity of the database, its data or users. The sharing of resources in this way allows for very flexible data networks whose capacity can be adjusted up or down to suit demand, without degradation of service. It also allows for a robust system to be devised as there is no single point at

which a failure can bring down the database, as the networked schema of the storage resources means that any failure would be local only.

#### 4.1.6 What is Oracle SQL

**Oracle SQL Developer** is an Integrated development environment (IDE) for working with SQL in Oracle databases. Oracle Corporation provides this product free, it uses the Java Development Kit. Structured Query Language (SQL) is the set of statements with which all programs and users access data in an Oracle database. Application programs and Oracle tools often allow users access to the database without using SQL directly, but these applications in turn must use SQL when executing the user's request. This chapter provides background information on SQL as used by most database systems.

The strengths of SQL provide benefits for all types of users, including application programmers, database administrators, managers, and end users. Technically speaking, SQL is a data sublanguage. The purpose of SQL is to provide an interface to a relational database such as Oracle Database, and all SQL statements are instructions to the database. In this SQL differs from general-purpose programming languages like C and BASIC. Among the features of SQL are the following:

- It processes sets of data as groups rather than as individual units.
- It provides automatic navigation to the data.
- It uses statements that are complex and powerful individually, and that therefore stand alone. Flow-control statements were not part of SQL originally, but they are found in the recently accepted optional part of SQL, ISO/IEC 9075-5: 1996. Flow-control statements are commonly known as "persistent stored modules" (PSM), and the PL/SQL extension to Oracle SQL is similar to PSM.
- SQL lets you work with data at the logical level. You need not be concerned with the implementation details only when you want to manipulate the data. For example, to retrieve a set of rows from a table, you define a condition used to filter the rows. All rows satisfying the condition are retrieved in a single step and can be passed as a unit to the user, to another SQL statement, or to an application. You need not deal with the rows one by one, nor do you have to worry about how they are physically stored or retrieved. All SQL statements use the **optimizer**, a part of Oracle Database that determines the most efficient means of

accessing the specified data. Oracle also provides techniques that you can use to make the optimizer perform its job better.

SQL provides statements for a variety of tasks, including:

- Querying data
- Inserting, updating, and deleting rows in a table
- Creating, replacing, altering, and dropping objects
- Controlling access to the database and its objects
- Guaranteeing database consistency and integrity

SQL unifies all of the preceding tasks in one consistent language.

All major relational database management systems support SQL, so you can transfer all skills you have gained with SQL from one database to another. In addition, all programs written in SQL are portable. They can often be moved from one database to another with very little modification.

## **4.2 Discussion of code segments**

In the following segment, we are going to discuss about the code segment.

### **4.2.1 Creating Database Connection**

Creating connection between front end and database.

```
import java.sql.*;

public class orconnect
{
    public static Connection connectdb()
    {
        //Intializing the connection variable to null

        Connection conn=null;
```

```
try

{

    String driverName="oracle.jdbc.driver.OracleDriver";

    Class.forName(driverName);

        conn = DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
            IUPKBLF:1521:xe","seenu","123");

        System.out.println("Successfully connected to the database");

    } catch (ClassNotFoundException e) {

        System.out.println("Could not find the database driver" + e.getMessage());

    } catch (SQLException e) {

        System.out.println("Could not connect to the database" + e.getMessage());

    }

    return conn;

}

}
```

#### **4.2.2 Code for displaying Database on JTable :**

```
public void showtabledata()

{

    //declaring JDBC objects

    Connection conn;

    ResultSet rs;

    PreparedStatement pst;
```

```
try

{

    String sql="select * from Branch";

    conn=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
        IUPKBLF:1521:xe","seenu","123");

    pst=conn.prepareStatement(sql);

    rs=pst.executeQuery();

    jTable1.setRowHeight(40);

    jTable1.setModel(DbUtils.resultSetToTableModel(rs));

} catch(Exception) {

    JOptionPane.showMessageDialog(null,e);

}

}
```

#### 4.2.3 Code for Using Updaate Statement:

```
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

    try

    {

        String sql="update branc hset b_name=? ,b_address=? ,pincode=? ,b_phone=?
            ,b_email=? ,b_password=? where b_id=?";

        pst=conn.prepareStatement(sql);

        pst.setString(1,jTextField15.getText());

        pst.setString(2,jTextField16.getText());

    }

}
```

```
pst.setString(3,jTextField17.getText());

pst.setString(4,jTextField18.getText());

pst.setString(5,jTextField19.getText());

pst.setString(6,jTextField20.getText());

pst.setString(7,jTextField14.getText());

pst.execute();

JOptionPane.showMessageDialog(null,"Updated Successfully");

showtabledata();

} catch(Exception e) {

    JOptionPane.showMessageDialog(null,e);

}

}
```

#### 4.2.4 Code for Using Insert Statement:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    try

    {

        String sql="insert into branch (b_id, b_name, b_address, pincode, b_phone, b_email,
            b_password) values (?,?,?,?,?,?,?) ";

        pst=conn.prepareStatement(sql);

        pst.setString(1,jTextField1.getText());

        pst.setString(2,jTextField2.getText());
```



```
pst.setString(3,jTextField3.getText());

pst.setString(4,jTextField4.getText());

pst.setString(5,jTextField5.getText());

pst.setString(6,jTextField6.getText());

pst.setString(7,jTextField7.getText());

pst.execute();

JOptionPane.showMessageDialog(null,"Inserted Successfully");

showtabledata();

}catch(Exception e)

    JOptionPane.showMessageDialog(null,e);

}

}
```

#### 4.2.5 Code for Using Delete Statement:

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    String sql="delete from Branch where b_id=?";

    try

    {

        pst=conn.prepareStatement(sql);

        pst.setString(1,jTextField8.getText());

        pst.execute();

        JOptionPane.showMessageDialog(null,"deleted Successfully");

        clearfields();

    }

}
```

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

### 4.3 Applications of project work

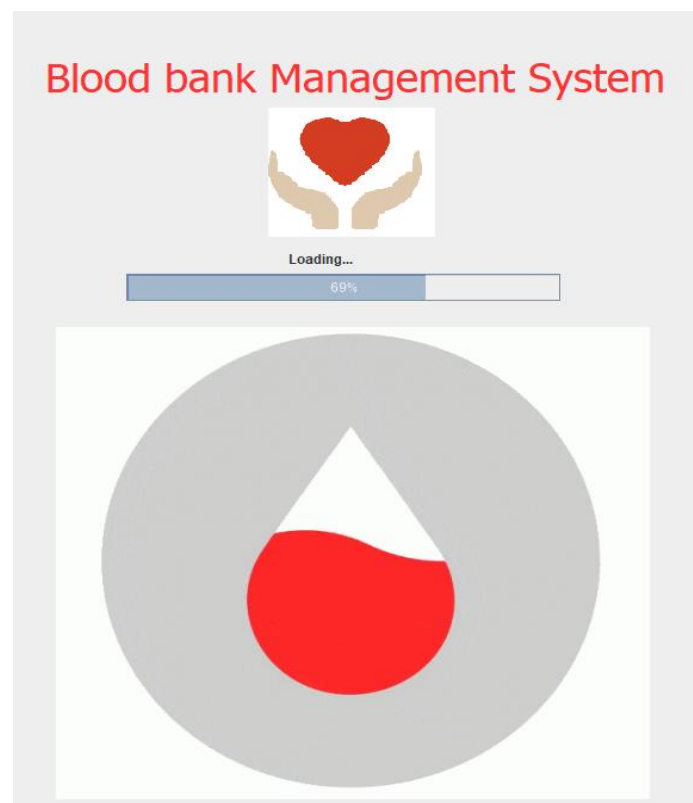
The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- It's a web-enabled project.
- It is helpful in reducing human efforts. Most of all are interested in computer rather than manual registers and notebooks, for all of them this program has been designed so that they can enjoy creating timetable without any effort. This program doesn't cover all the requirements but it covers major. In future this program will be able to develop some more facilities so that it covers everything.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for him that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.

- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.
- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency,

## 4.4 Discussion of the results

### 1. Start page



**Fig 4.1: Start page of the application**

Whenever we run the project, this is the page that is seen. If the progress bar is 100%, then we will be directed to next page.

## 2. Home page



Fig 4.2: Home page

Here, there are many options as shown above. Blood banks, donors, requests can be viewed.

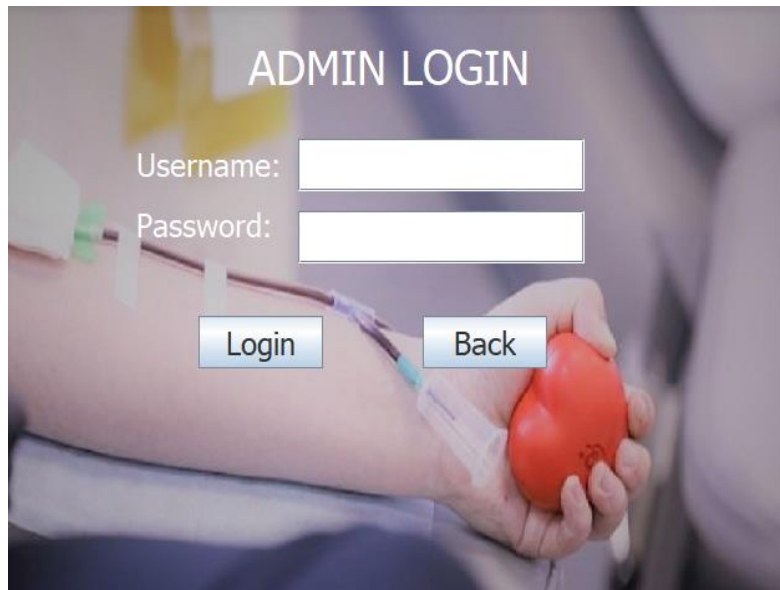
## 3. Login page



Fig 4.3: Login page

Here the user who has account as admin/request/donor/manager can login accordingly.

#### 4. Admin login



**Fig 4.4: Admin login**

This is the login page for admin. On correct username and password, one can login through admin account.

#### 5. Admin home page

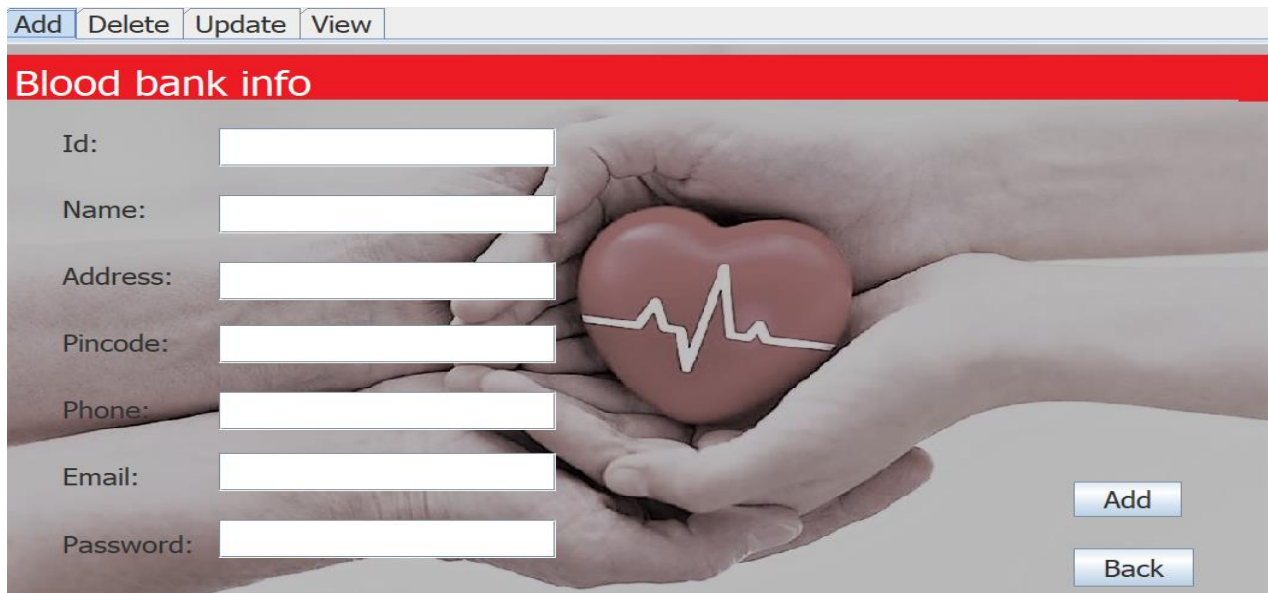


**Fig 4.5: Admin home page**

This is admin home portal where admin can add/delete/modify blood bank/donors/requests.

## 6. Admin Blood bank portal

### 6.1.Add Blood bank

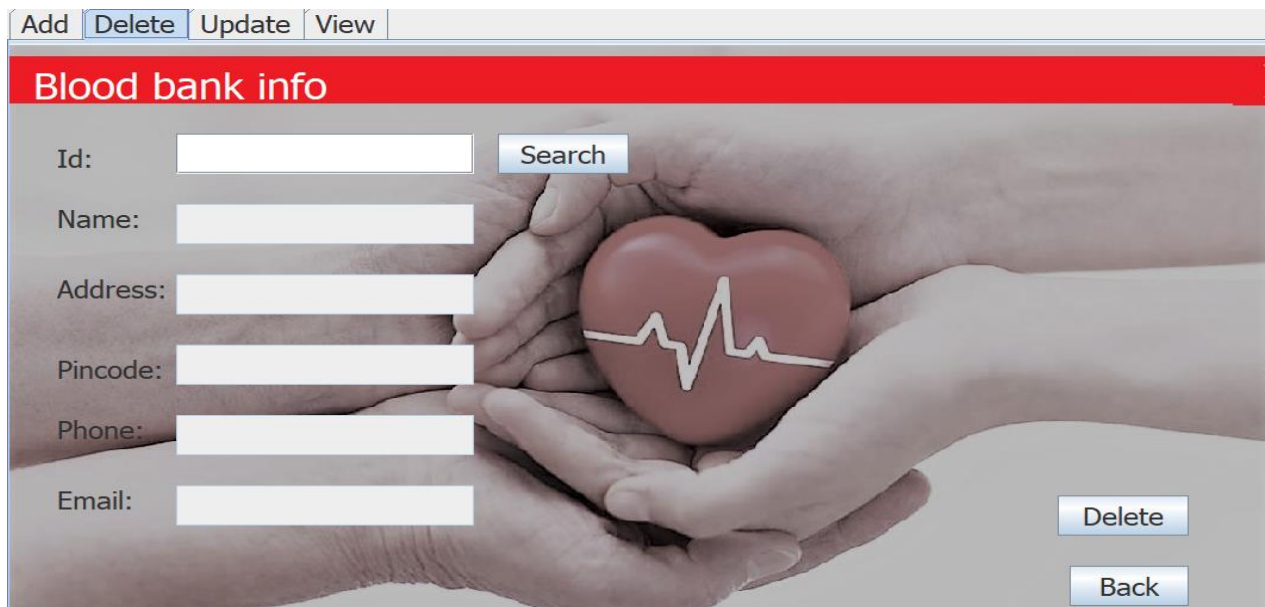


Admin Blood bank portal - Add Blood bank form. The form includes fields for Id, Name, Address, Pincode, Phone, Email, and Password. The 'Add' button is highlighted.

**Fig 4.6: Admin blood bank portal-Add blood bank**

Here, admin can add the blood bank by filling the above details and clicking add button.

### 6.2. Delete blood bank



Admin Blood bank portal - Delete Blood bank form. The form includes fields for Id, Name, Address, Pincode, Phone, and Email. A 'Search' button is next to the Id field. The 'Delete' button is highlighted.

**Fig 4.7: Admin blood bank portal-Delete blood bank**

Here, admin can delete the blood bank by entering blood bank Id.



### 6.3 Update blood bank

Add Delete **Update** View

**Blood bank info**

Id:

Name:

Address:

Pincode:

Phone:

Email:

Password:

**Fig 4.8: Admin blood bank portal-Update blood bank**

Here, admin can update the blood bank by entering blood bank Id and modifying the details.

### 6.4 View blood bank

Add Delete Update **View**

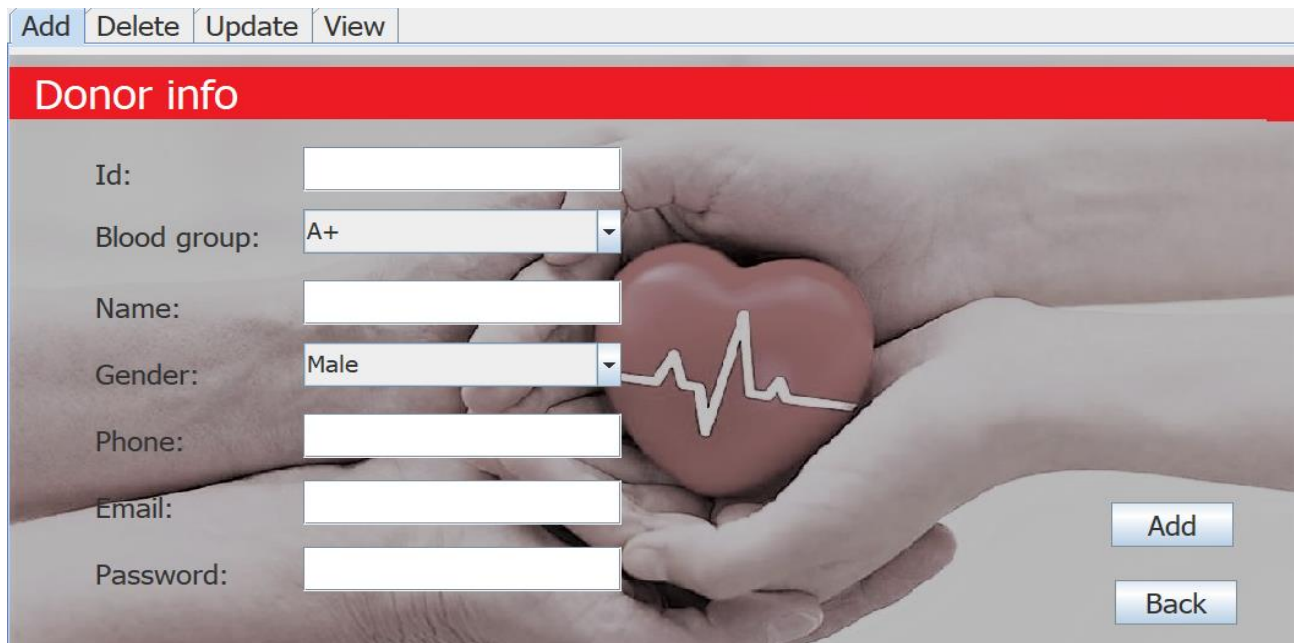
**Blood bank info**

B_ID	B_NAME	B_ADDRESS	PINCODE	B_PHONE	B_EMAIL	B_PASSWORD
3	Life care volu...	Jayanagar west	560011	8123533773	lifecare@gma...	life
4	Jeeva voluntary	New Tharagu...	560002	9845801539	jvbb02@gmai...	jeeva
1	Lions blood b...	Ganganagar	560050	9740100485	lbb@gmail.com	lbb
2	BMS blood ba...	Channasandra	560098	9177764452	bmsbb@outlo...	bmsbb
5	Swamy vivek...	Sampangiram...	560027	9886763931	swamyviveka...	swamy
6	Rashthrothana	Kempegowda...	560019	8026612730	rbloodbank@...	rashttr
7	M S Ramaiah ...	New Bel road	560054	9008300559	contact@msr...	msram
8	Grace blood b...	Ganganagar	560032	9740330720	krishnakamal...	grace

**Fig 4.9: Admin blood bank portal-View blood banks**

Here, admin can view all the available blood banks and their details.

## 7. Admin Donors page

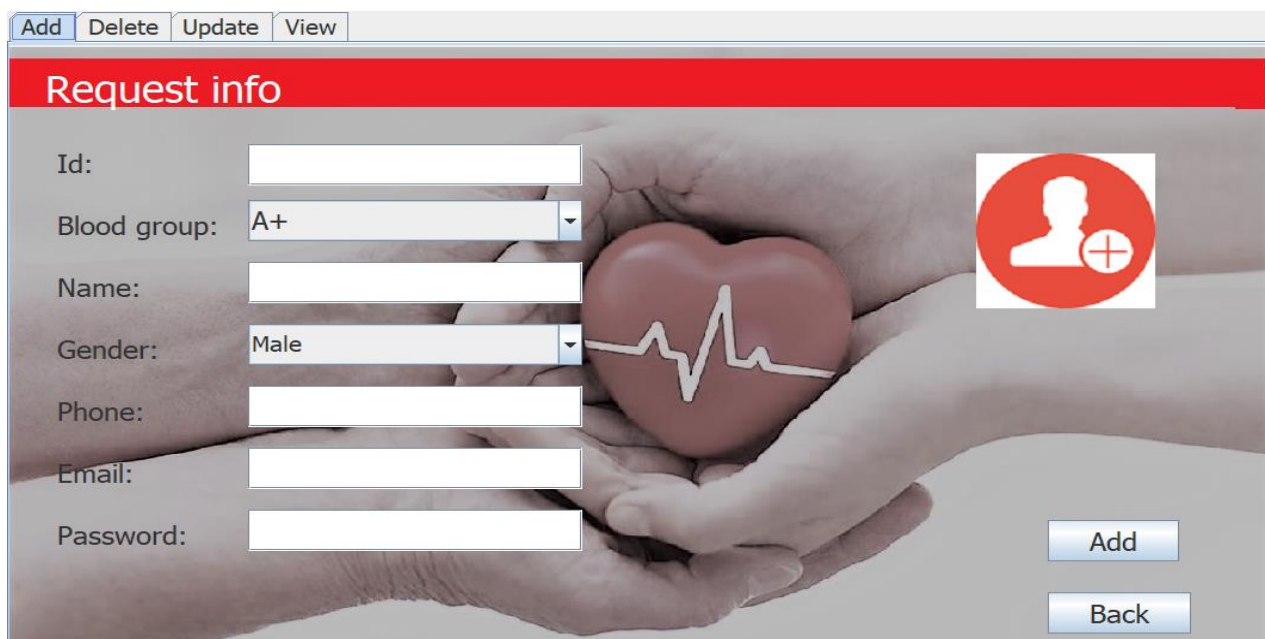


The screenshot shows the 'Admin Donors page' of a Blood Bank Management System. At the top, there is a navigation bar with buttons for 'Add', 'Delete', 'Update', and 'View'. Below this is a red header bar with the text 'Donor info'. The main content area features a form with the following fields: 'Id:' (text input), 'Blood group:' (dropdown menu showing 'A+'), 'Name:' (text input), 'Gender:' (dropdown menu showing 'Male'), 'Phone:' (text input), 'Email:' (text input), and 'Password:' (text input). To the right of the form, there is a large image of hands holding a heart with a heartbeat line. At the bottom right, there are two buttons: 'Add' and 'Back'.

**Fig 4.10: Admin Donor portal**

Here, admin can add/delete/update/view the donor information alike blood bank information.

## 8. Admin Requests page



The screenshot shows the 'Admin Requests page' of a Blood Bank Management System. At the top, there is a navigation bar with buttons for 'Add', 'Delete', 'Update', and 'View'. Below this is a red header bar with the text 'Request info'. The main content area features a form with the following fields: 'Id:' (text input), 'Blood group:' (dropdown menu showing 'A+'), 'Name:' (text input), 'Gender:' (dropdown menu showing 'Male'), 'Phone:' (text input), 'Email:' (text input), and 'Password:' (text input). To the right of the form, there is a large image of hands holding a heart with a heartbeat line. In the top right corner, there is a red circular icon with a white silhouette of a person and a plus sign. At the bottom right, there are two buttons: 'Add' and 'Back'.

**Fig 4.11: Admin requests portal**

Here, admin can add/delete/update/view the requests information alike blood bank information.



## 9. Branch Manager Login

### 9.1 Branch Manager Home page

Home Transactions Stock information

**Blood bank Info:**

Enter Id:

Enter password:

Name:

Address:

Pincode:

Phone:

Email:

**Fig 4.12: Branch manager portal-home page**

Here, branch manager can login into the account and can update branch details.

### 9.2 Transactions Page

Home Transactions Stock information

Donor Id:

Request No:

Title 1	Title 2	Title 3	Title 4

**Fig 4.13: Branch manager portal-Transactions page**

Here, branch manager can add/delete valid transactions by entering donor Id and request number.

### 9.3 Stock Information page

Home Transactions **Stock information**

Blood group: A+

Quantity:

Title 1	Title 2	Title 3	Title 4

**Fig 4.14: Branch manager portal-Stock information**

Here, branch manager can add/update the quantity of selected blood group.

## 10. Donor login page

### 10.1 Donor home page

Donor information transactions

Enter Id: 4

Enter password: .....

Blood group: B+

Name: Kavya M

Gender: Female

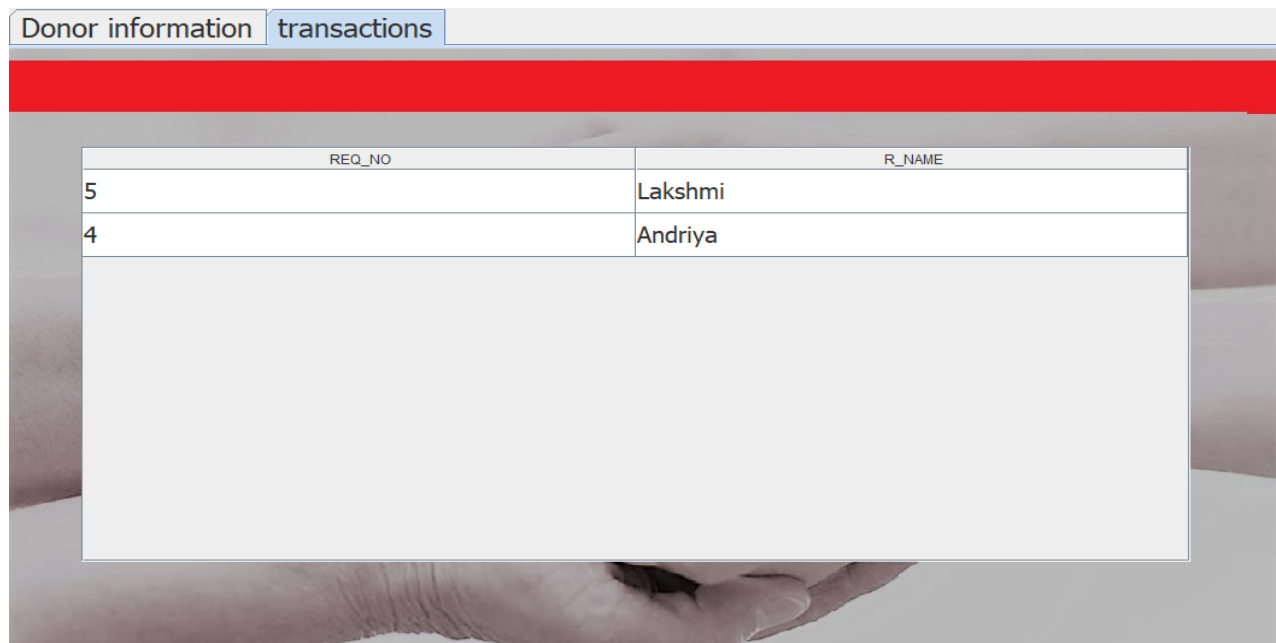
Phone: 8700675434

Email: kavyam@newmail.com

**Fig 4.15: Donor portal-home page**

Here, donor can login to his/her account and update/delete the account.

## 10.2 Donor transactions page

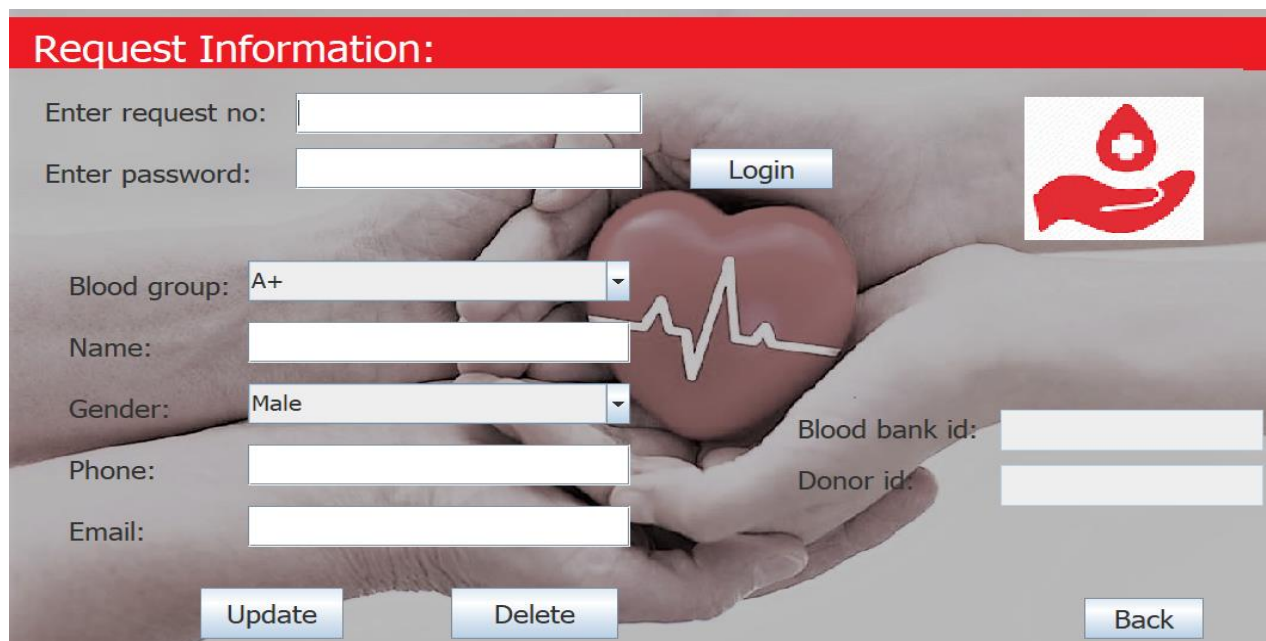


REQ_NO	R_NAME
5	Lakshmi
4	Andriya

**Fig 4.16: Donor portal-donor transaction**

Here, the donor who had login to their account can view his/her transactions.

## 11. Requester Login page



**Fig 4.17: Requester portal**

Here, the people who had requested for blood can login and modify/delete their details.

## 12. View donors page

Donors:					
D_ID	BLOOD_GR	D_NAME	D_GENDER	D_PHONE	D_EMAIL
1	B+	Amulya	Female	9703074871	amulya1@yaho...
2	O+	Ranjith	Male	7019191899	rangith@outloo...
3	A+	Nagaraj	Male	7890567856	nagaraj@yahoo...
4	B+	Kavya M	Female	8700675434	kavyam@new...
5	O+	asha rani	Female	9703074871	asha.aswarth@...
6	AB+	Rakshith	Male	8976567876	rakshith@outlo...
7	AB-	Rahul	Male	9008789090	rahul@yahoo.c...
8	A+	Bharath	Male	8999787890	bharath@gmail...
9	B-	Sireesha	Female	9008777889	sireesha@outlo...

Back

**Fig 4.18: View donor information**

Here, the details of all registered donors can be seen and is useful during emergency conditions.

## 13. View requests page

Requests:					
REQ_NO	BLOOD_GR	R_NAME	R_GENDER	R_PHONE	R_EMAIL
2	B+	Lasya	Female	7668665644	lasya@yahoo.c...
1	AB+	Pradeep	Male	8990805678	pradeep@gmail...
3	A+	Tharun	Male	7889990890	tharun@gmail.c...
4	B-	Andriya	Female	9880789678	andriya@hotm...
5	AB+	Lakshmi	Female	9887766556	lakshmi@yahoo...
6	AB+	Vishal	Male	9787856009	vishal@newmai...

Back

**Fig 4.19: View requests information**

Here, the details of all the requests can be seen. This page is visible to everyone.

**14. View blood banks**

Available blood banks						
B_ID	B_NAME	B_ADDRESS	PINCODE	B_PHONE	B_EMAIL	REV_STARS
1	Lions blood ...	Ganganagar	560050	9740100485	lbb@gmail.c...	4
2	BMS blood b...	Channasandra	560098	9177764452	bmsbb@outl...	4.3
3	Life care vol...	Jayanagar w...	560011	8123533773	lifecare@gm...	4.2
4	Jeeva volunt...	New Tharag...	560002	9845801539	jvbb02@gm...	4.3
5	Swamy vivek...	Sampangira...	560027	9886763931	swamyvivek...	4
6	Rashthrohana	Kempegowd...	560019	8026612730	rbloodbank...	4
7	M S Ramaiah...	New Bel road	560054	9008300559	contact@msr...	3.9
8	Grace blood ...	Ganganagar	560032	9740330720	krishnakama...	3.9

**Fig 4.20: View available blood banks**

Here, all the available blood banks and corresponding details are seen. This page can be seen by all the users.

## Chapter 5

# CONCLUSION AND FUTURE ENHANCEMENTS

Our project is only a humble venture to satisfy the needs in a blood bank to manage their stock, inventory. Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database. The objective of this project is to provide a framework that enables people to be easily in the part of the blood donation process. Easier and faster data transfer through latest technology associated with the computer and communication.

This project can be extended so that it maintains the details of several blood camps. From this any user can register their slot to the blood camps. It can also be made to store the data about different types of donation, height weight requirements of donation, different components of blood stored, records about the blood expired etc. It can be made more secured by introducing some options and emergency options can also be used. It can be connected to the server so that it can be accessible for everyone. Compatibility testing details can be implemented. Before a recipient receives a transfusion, compatibility testing between donor and recipient blood must be done. The first step before a transfusion is given is to type and screen the recipient's blood.

## REFERENCES

- [1] <https://netbeans.org/kb/73/java/project-setup.html>
- [2] <https://netbeans.org/projects>
- [3] [www.stackoverflow.com](http://www.stackoverflow.com)
- [4] [www.oracle.org/netbeans\\_ide](http://www.oracle.org/netbeans_ide)
- [5] [www.javatpoint.org](http://www.javatpoint.org)
- [6] [https://docs.oracle.com/netbeans/nb81/netbeans/develop/create\\_japps.html](https://docs.oracle.com/netbeans/nb81/netbeans/develop/create_japps.html)
- [7] <https://openjdk.java.net/groups/nb-projects/>