



IBM CAREER EDUCATION

MAIN PROJECT

DOMAIN NAME: JAVA

Blood Bank Management System

Submitted By,

- Tanvi Trivedi (18162171031)
- Meshva Gupta (181621710)
- Jhanvi Zala(18162171035)

II Year – (CS) ‘B’ Section

Ganpat University, Ahmedabad.

Submitted To,

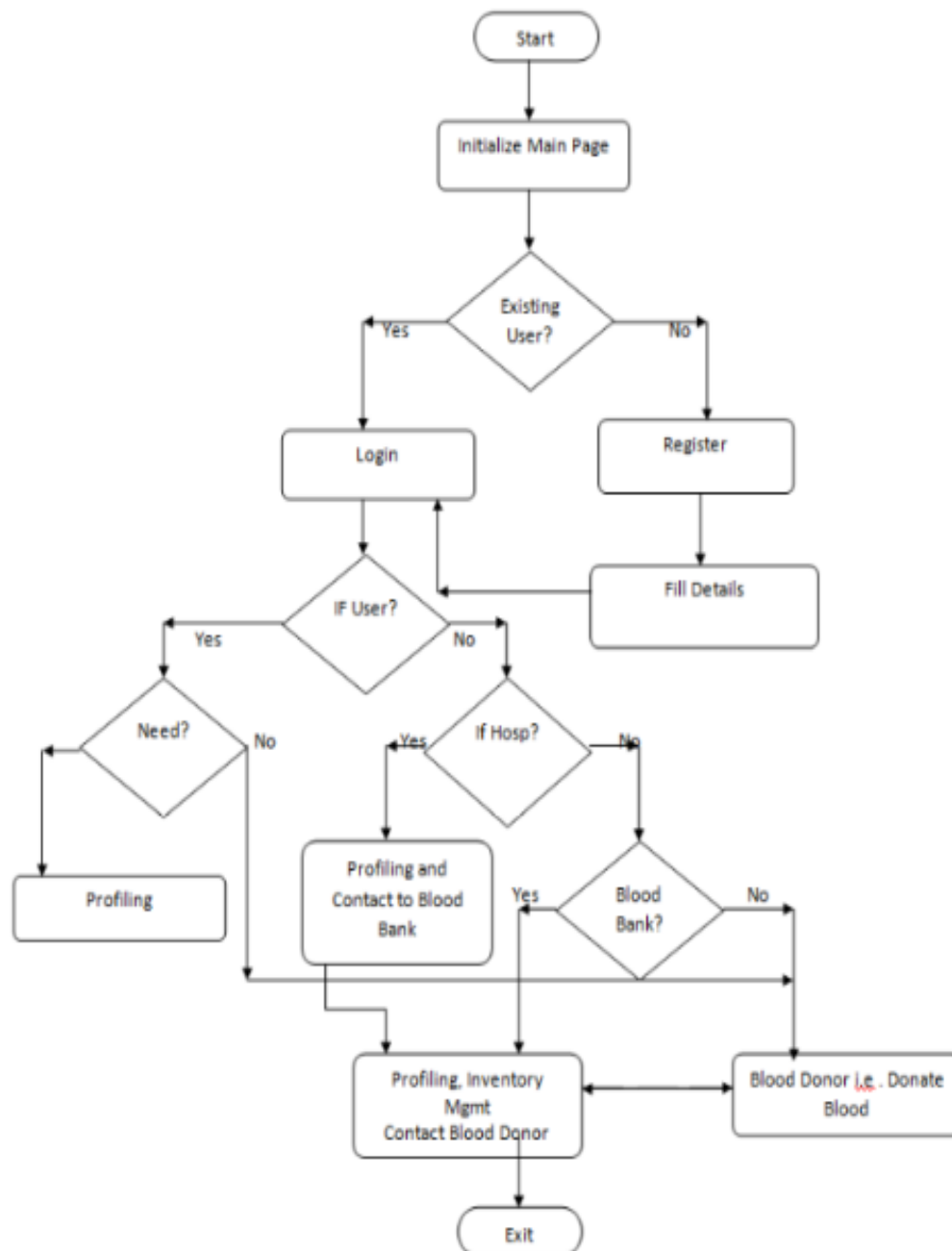
A. Saai Sanjeev Achaarya

IBM Software Technical Trainer

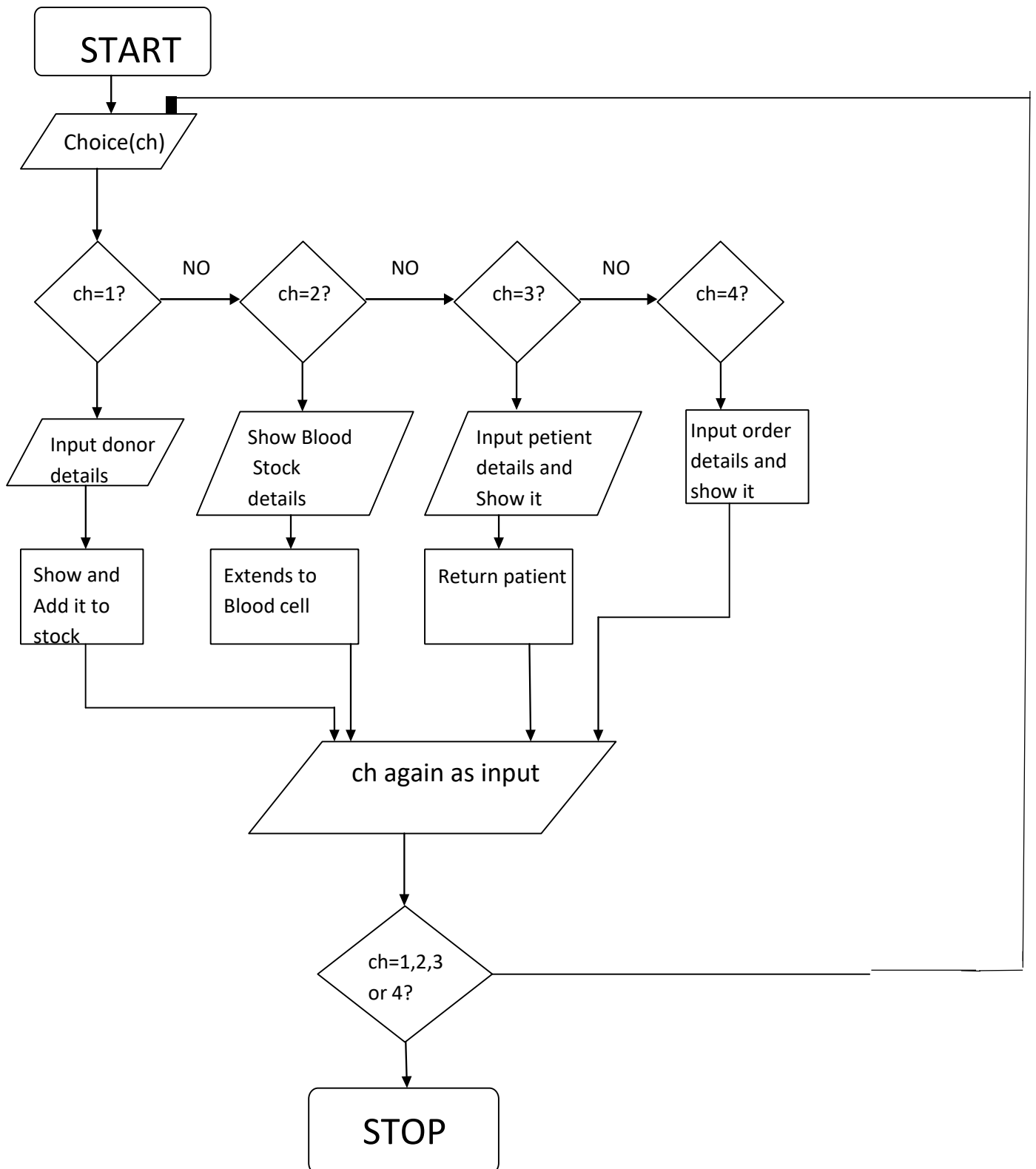


FLOWCHART

- System Architecture



- Flowchart of our program



SOFTWARE SPECIFICATIONS

- OPERATING SYSTEM : Linux / Windows / IOS
- ENVIRONMENT : IBM RAD Software

HARDWARE SPECIFICATIONS

- ☐ PROCESSOR : PENTIUM IV 2.8MHz
- ☐ RAM : 256 MB SD RAM
- ☐ MONITOR : 15" COLOR
- ☐ HARD DISK : 40 GB
- ☐ FLOPPY DRIVE : 1.44 MB

LANGUAGES USED:

- ☐ JAVA : For main coding
- ☐ HTML : For Webpage
- ☐ CSS : For Webpage
- ☐ JAVASCRIPT : For Webpage
- ☐ JQUERY : For Webpage

DESCRIPTION

INTRODUCTION:

The BLOOD BANK MANAGEMENT SYSTEM is great project. This project is designed for successful completion of project on blood bank management system. Blood Bank Management System (BBMS) is a Java Project 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. Project 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.

DESCRIPTION:

Purpose:

- The proposed system (Blood Bank Management System) is designed to help the Blood Bank administrator to meet the demand of Blood by sending and/or serving the request for Blood as and when required. The proposed system gives the procedural approach of how to bridge the gap between Recipient, Donor, and Blood Banks. This Application will provide

a common ground for all the three parties (i.e. Recipient, Donor, and Blood Banks) and will ensure the fulfillment of demand for Blood requested by Recipient and/or Blood Bank

Goals:

- To ease the process of blood donation and reception.
- To improve the existing system.
- To develop a scalable system.
- To be highly available.

Scope:

- Ensure that all the functionalities of a manual blood bank are covered.
- To include all the blood banks at least within a city.
- Make sure the program is simple and easy to use.

PROS:

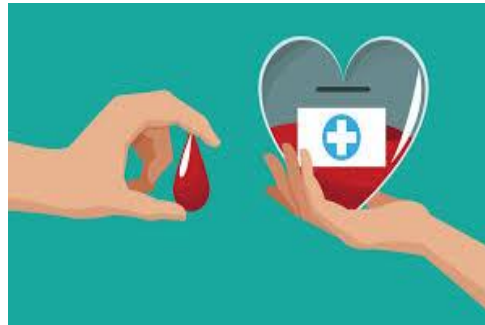
- **Donors:** person who wants to donate the blood voluntarily at the blood donation camp.
- **Seekers:** person who wants the blood from the blood bank due to various reasons like accidents, surgeries, delivery and many more.
- **Blood bank:** staff people which are working in the blood bank which includes staff member, operator, blood bank in charge, head of pathological department.

CONS:

- Wastage of blood due to expiry.

BLOOD BANK MANAGEMENT SYSTEM

PICTURES



AIM:

- The **aim** of this socially relevant project is to manage the donation and storing in blood bank so people can contact easily.

PROGRAM:

- **Java code**

```
//Importing all the relevant packages and their classes
import java.io.*;
import java.util.*;

class blood_stock
{
//Method to show blood stock

static int
qua1=0,qua2=0,qua3=0,qua4=0,qua5=0,qua6=0,qua7=0,qua8=0
;
void blood_stockk()
{
System.out.println("\nBlood Stocks");
System.out.println("Blood Group : A+ \nStock :" + qua1 + "
ml");
System.out.println("Blood Group : A- \nStock :" + qua2 + " ml");
System.out.println("Blood Group : B+ \nStock :" + qua3 + " ml");
System.out.println("Blood Group : B- \nStock :" + qua4 + " ml");
System.out.println("Blood Group : O+ \nStock :" + qua5 + "
ml");
System.out.println("Blood Group : O- \nStock :" + qua6 + " ml");
System.out.println("Blood Group : AB+ \nStock :" + qua7 + "
ml");
```

```

System.out.println("Blood Group : AB- \nStock :" + qua8 + "
ml");
}
}
class blood_cell extends blood_stock
{
void blood_stockk()
{
System.out.println("\nBlood Stocks");
System.out.println("Blood Group : A+ \nStock :" + (qua1*5000)
+ " units");
System.out.println("Blood Group : A- \nStock :" + (qua2*5000)
+ " units");
System.out.println("Blood Group : B+ \nStock :" + (qua3*5000)
+ " units");
System.out.println("Blood Group : B- \nStock :" + (qua4*5000)
+ " units");
System.out.println("Blood Group : O+ \nStock :" + (qua5*5000)
+ " units");
System.out.println("Blood Group : O- \nStock :" + (qua6*5000)
+ " units");
System.out.println("Blood Group : AB+ \nStock :" +
(qua7*5000) + " units");
System.out.println("Blood Group : AB- \nStock :" +
(qua8*5000) + " units");
}
}
class patient
{
//Declaring variables.

String p_name,pblg,padd,rbg4,psex;
int quan,i,pid;

//Method to get patient details

void new_patient()
{
Scanner input = new Scanner(System.in);

```

```
System.out.print("Patient id : ");  
pid = input.nextInt();
```

```
System.out.print("Patient name : ");  
p_name = input.next();
```

```
System.out.print("Address : ");  
padd = input.next();
```

```
do{  
    System.out.print("Gender : ");  
    psex = input.next();  
    if(!(psex.equals("F")  
||psex.equals("M")||psex.equals("f")||psex.equals("m")))  
    {  
        System.out.println("Wrong input please try again");  
    }  
    else  
    {  
        break;  
    }  
}while(true);
```

```
do{  
    System.out.print("Blood Group of patient : ");  
    pblg = input.next();
```

```
    if(!(pblg.equals("A+")||pblg.equals("B+")||pblg.equals("AB+")||  
pblg.equals("O+")||pblg.equals("A-")||pblg.equals("B-")  
||pblg.equals("AB-")||pblg.equals("O-")  
||pblg.equals("a+")||pblg.equals("b+")||pblg.equals("ab+")||pblg.  
equals("o+")||pblg.equals("a-")||pblg.equals("b-")  
||pblg.equals("ab-")||pblg.equals("o-")))  
    {
```

```
        System.out.println("Input you entered is not a blood group.  
Please try again");
```

```

    }
    else
    {
        break;
    }
}while(true);}

```

//Method to show patient's details

```

void patient_info()
{
    System.out.println("ID" + "\t" + "Name" + "\t" + "Gender" + "\t"
+ "Address" + "\t" + "Blg_grp");
    System.out.println(pid + "\t" + p_name + "\t" + psex + "\t" + padd
+ "\t" + pblg);
}
}

```

class donor extends blood_stock

```

{
//Declaring variables

```

```

String don_name,en_date,bg5,hc_name,dsex;
int id_don,qua;

```

//Method to get donor's details

```

void new_donor()
{
    Scanner input = new Scanner(System.in);

```

```

    System.out.print("Donor id : ");
    id_don = input.nextInt();

```

```

    System.out.print("Donor name : ");
    don_name = input.next();

```

```

    System.out.print("Date of Donation : ");
    en_date = input.next();

```

```

do{

```

```

System.out.print("Gender : ");
dsex = input.next();
if(!(dsex.equals("F")
||dsex.equals("M")||dsex.equals("f")||dsex.equals("m")))
{
    System.out.println("Wrong input please try again");
}
else
{
    break;
}
}while(true);

do{
    System.out.print("Blood Group of donor : ");
    bg5 = input.next();

    if(!(bg5.equals("A+")||bg5.equals("B+")||bg5.equals("AB+")||bg
5.equals("O+")||bg5.equals("A-")||bg5.equals("B-
")||bg5.equals("AB-")||bg5.equals("O-
")||bg5.equals("a+")||bg5.equals("b+")||bg5.equals("ab+")||bg5.e
quals("o+")||bg5.equals("a-")||bg5.equals("b-")||bg5.equals("ab-
")||bg5.equals("o-")))
    {
        System.out.println("Input you entered is not a blood
group. Please try again");
    }
    else
    {
        break;
    }
}while(true);

System.out.print("Quantity of Blood donated : ");

```

```

qua = input.nextInt();

if(bg5.equals("A+") || bg5.equals("a+"))
    qua1 += qua;
else if(bg5.equals("A-") || bg5.equals("a-"))
    qua2 += qua;
else if(bg5.equals("B+") || bg5.equals("b+"))
    qua3 += qua;
else if(bg5.equals("B-") || bg5.equals("b-"))
    qua4 += qua;
else if(bg5.equals("O+") || bg5.equals("o+"))
    qua5 += qua;
else if(bg5.equals("O-") || bg5.equals("o-"))
    qua6 += qua;
else if(bg5.equals("AB+") || bg5.equals("ab+"))
    qua7 += qua;
else if(bg5.equals("AB-") || bg5.equals("ab-"))
    qua8 += qua;
System.out.print("Hospital or camp name : ");
hc_name = input.next();
}

```

//Method to show donor's details

```

void donor_info()
{
System.out.println("ID" + "\t" + "Name" + "\t" + "Gender" + "\t"
+ "Entry_date" + "\t" + "Bld_grp" + "\t" + "Quantity" + "\t" +
"Hospital_name");
System.out.println(id_don + "\t" + don_name + "\t" + dsex + "\t"
+ en_date + "\t\t" + bg5 + "\t\t" + qua5 + "\t" + hc_name);
}
}
class order extends blood_stock
{
//Method to show order details

void orderr()
{
String p_name,pblg,date3,avl;

```

```

int quan,eml3;
Scanner input = new Scanner(System.in);
System.out.print("Patient name : ");
p_name = input.next();

do{
System.out.print("Ordered blood group : ");
pblg = input.next();

if(!(pblg.equals("A+")||pblg.equals("B+")||pblg.equals("AB+")||
pblg.equals("O+")||pblg.equals("A-")||pblg.equals("B-")||pblg.equals("AB-")||pblg.equals("O-")||pblg.equals("a+")||pblg.equals("b+")||pblg.equals("ab+")||pblg.equals("o+")||pblg.equals("a-")||pblg.equals("b-")||pblg.equals("ab-")||pblg.equals("o-")))
{
    System.out.println("Input you entered is not a blood group.
Please try again");
}
else
{
    System.out.print("Quantity of Blood Group : ");
    quan = input.nextInt();
    if(pblg.equals("A+") || pblg.equals("a+"))
    {
        if(quan>qual)
        {
            System.out.println("Sorry the quantity
exceeds the stock.");
        }
        else
        {
            qual -= quan;
            break;
        }
    }
}
}

```

```
else if(pblg.equals("A-") || pblg.equals("a-"))
{
    if(quan>qua2)
    {
        System.out.println("Sorry the quantity
exceeds the stock.");
    }
    else
    {
        qua2 -= quan;
        break;
    }
}
else if(pblg.equals("B+") || pblg.equals("b+"))
{
    if(quan>qua3)
    {
        System.out.println("Sorry the quantity
exceeds the stock.");
    }
    else
    {
        qua3 -= quan;
        break;
    }
}
else if(pblg.equals("B-") || pblg.equals("b-"))
{
    if(quan>qua4)
    {
        System.out.println("Sorry the quantity
exceeds the stock.");
    }
    else
    {
        qua4 -= quan;
        break;
    }
}
else if(pblg.equals("O+") || pblg.equals("o+"))
```



```
{
    if(quan>qua5)
    {
        System.out.println("Sorry the quantity
exceeds the stock.");
    }
    else
    {
        qua5 -= quan;
        break;
    }
}
else if(pblg.equals("O-") || pblg.equals("o-"))
{
    if(quan>qua6)
    {
        System.out.println("Sorry the quantity
exceeds the stock.");
    }
    else
    {
        qua6 -= quan;
        break;
    }
}
else if(pblg.equals("AB+") || pblg.equals("ab+"))
{
    if(quan>qua7)
    {
        System.out.println("Sorry the quantity
exceeds the stock.");
    }
    else
    {
        qua7 -= quan;
        break;
    }
}
```

```

        }
    }
    else if(pblg.equals("AB-") || pblg.equals("ab-"))
    {
        if(quan>qua8)
        {
            System.out.println("Sorry the quantity
exceeds the stock.");
        }
        else
        {
            qua8 -= quan;
            break;
        }
    }
    // break;
}
}while(true);

```

```

System.out.print("Emergency level(1 to 5) : ");
eml3 = input.nextInt();

```

```

}
}

```

```

class total
{
    public static void main(String[] args)
    {

```

```

        Scanner input=new Scanner(System.in);
        patient[] p=new patient[25];
        donor d=new donor();
        blood_stock bd=new blood_stock();
        blood_cell bc=new blood_cell();
        order o=new order();
        int choice,c1,s1,count1=0,j=0,status=1;

```

```

s:
while(status==1)
{
    System.out.println("\n                                MAIN
MENU");
    System.out.println("-----
-----");
    System.out.println("1.DONOR    2. BLOOD STOCK
3.PATIENT 4.ORDER 5.BLOOD CELL 6.EXIT");
    System.out.println("-----
-----");
    System.out.print("                                Enter choice : ");
    choice = input.nextInt();
    switch (choice)
    {
        case 3:
        {
            System.out.println("-----
-----");
            System.out.println("                                **PATIENT
SECTION**");
            System.out.println("-----
-----");
            s1=1;
            while(s1==1)
            {
                System.out.println("\n
MAIN MENU");
                System.out.println("-----
-----");
                System.out.println("1.Add a new data
2.Display 3.EXIT");
                System.out.println("-----
-----");

                c1=input.nextInt();

```

```

//switch(c1)
if(c1==1)
{
    //case 1:
    p[count1] = new patient();

    p[count1].new_patient();
    count1++;
    //break;
}
//case 2:
else if(c1==2)
{
    for(j=0;j<count1;j++)
    {
        System.out.println("The
patient's details : ");

        System.out.println();
        p[j].patient_info();
    }
    //break;
}
else if(c1==3)
{
    System.out.println("Return back
press (1/0) for more");
    s1=input.nextInt();
}
else
{
    System.out.print("default");
}
}
break;
}
case 1:
    System.out.println("-----
-----");
    System.out.println("                **DONOR
SECTION**");

```

```

        System.out.println("-----
-----");
        d.new_donor();
        System.out.println("The donor's details : ");
        d.donor_info();
        break;
    case 2:
        System.out.println("-----
-----");
        System.out.println("
**BLOOD_STOCK SECTION**");
        System.out.println("-----
-----");
        bd.blood_stockk();
        break;
    case 4:
        System.out.println("-----
-----");
        System.out.println("                **ORDER
SECTION**");
        System.out.println("-----
-----");
        o.orderr();
        break;
    case 5:
        System.out.println("-----
-----");
        System.out.println("
**BLOOD_CELL SECTION**");
        System.out.println("-----
-----");
        bc.blood_stockk();
        break;
    case 6: break s;
}
}

```

```
}
}
```

- **Html code (including .html, .css, .js files)**

1. Home page



Home Page.rar

2. Sign_up Page



Sign_up Page.rar

3. Admin Page



Admin page.rar

EXPLANATION ABOUT PROJECT:

- The project of bloodbank management system is based on JAVA domain here, first of all, We declare main header Files and then we declare classes like blood_stock, blood_cells , patient, donor, order,total where the main method is in class total.
- In first class blood_Stock returns the stock based on the donor's donation and patient's order for that to be done, here we declare variables and extends it to the blood_cell to return the units of blood (i.e., 1ml = 5000units of blood cell).

- Then in patient class the variables are declared taken input for new entry of patient. There is a *validation that in patient gender, you can only add F or M else it will return “Wrong input please try again”* and then break. Again using do while loop, validation is for only 8 blood group you can enter. *Random entry for blood group is not possible otherwise it will return “Input you entered is not a blood group. Please try again” (i.e., validation for bloodgroup is used).*
- Class donor, which also extends blood_stock, so if we are entering in donor, then the which blood group he/she donates, will directly goes to blood_Stock in to that blood_group and also goes to blood_cell and multiply by 5000 units(for returning the unit of blood cells). *There is also validation in gender and blood group.*
- Class order extends blood_stock , *if ordered blood group quantity is less than the blood_stock then it will execute the other conditions otherwise if the quantity is greater than the required blood group it will print “Sorry the quantity exceeds the stock”*. Then it will ask you emergency level of order out of 5.
- Main method is declared in class total using while loop and in that we have used switch case for main menu and for other sub menus.

- There is a option of exit and it ask you that you want exit only this or directly go to main menu.
- This was the briefing of our java code.

OUTPUT SCREENSHOTS:

- Java code output

```

                                MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
Enter choice : 1
-----
                                **DONOR SECTION**
-----
Donor id : 101
Donor name : Tanvi
Date of Donation : 12/12/2018
Gender : F
Blood Group of donor : AB-
Quantity of Blood donated : 200
Hospital or camp name : zydus
The donor's details :
ID      Name    Gender  Entry_date    Bld_grp    Quantity    Hospital_name
101     Tanvi    F       12/12/2018    AB-        200         0          zydus

                                MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
Enter choice : 1
-----
                                **DONOR SECTION**
-----
Donor id : 102
Donor name : Jhanvi
Date of Donation : 16/4/2019
Gender : F
Blood Group of donor : B+
Quantity of Blood donated : 150
Hospital or camp name : ganpat
The donor's details :
ID      Name    Gender  Entry_date    Bld_grp    Quantity    Hospital_name
102     Jhanvi    F       16/4/2019    B+         150         0          ganpat

```

```

                                MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
                        Enter choice : 2
-----
                        **BLOOD_STOCK SECTION**
-----

Blood Stocks
Blood Group : A+
Stock :0 ml
Blood Group : A-
Stock :0 ml
Blood Group : B+
Stock :150 ml
Blood Group : B-
Stock :0 ml
Blood Group : O+
Stock :0 ml
Blood Group : O-
Stock :0 ml
Blood Group : AB+
Stock :0 ml
Blood Group : AB-
Stock :200 ml

                                MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
                        Enter choice : 3
-----
                        **PATIENT SECTION**
-----

                                MAIN MENU
-----
1.Add a new data 2.Display 3.EXIT
-----
1
Patient id : 1
Patient name : Meshva
Address : Ahmedabad
Gender : F
Blood Group of patient : B+

                                MAIN MENU
-----
1.Add a new data 2.Display 3.EXIT
-----
1
Patient id : 2
Patient name : Shreya
Address : Prahaladnagar
Gender : A
Wrong input please try again
Gender : F
Blood Group of patient : Q+
Input you entered is not a blood group. Please try again
Blood Group of patient : O-

                                MAIN MENU
-----
1.Add a new data 2.Display 3.EXIT
-----
2
The patient's details :

ID      Name      Gender  Address  Blg_grp
1      Meshva    F      Ahmedabad  B+
The patient's details :

ID      Name      Gender  Address  Blg_grp
2      Shreya    F      Prahaladnagar  O-

```

```

MAIN MENU
-----
1.Add a new data 2.Display 3.EXIT
-----
3
Return back press (1/0) for more
0

MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
Enter choice : 4
-----
**ORDER SECTION**
-----
Patient name : Shreya
Ordered blood group : O-
Quantity of Blood Group : 100
Sorry the quantity exceeds the stock.
Ordered blood group : AB-
Quantity of Blood Group : 100
Emergency level(1 to 5) : 3

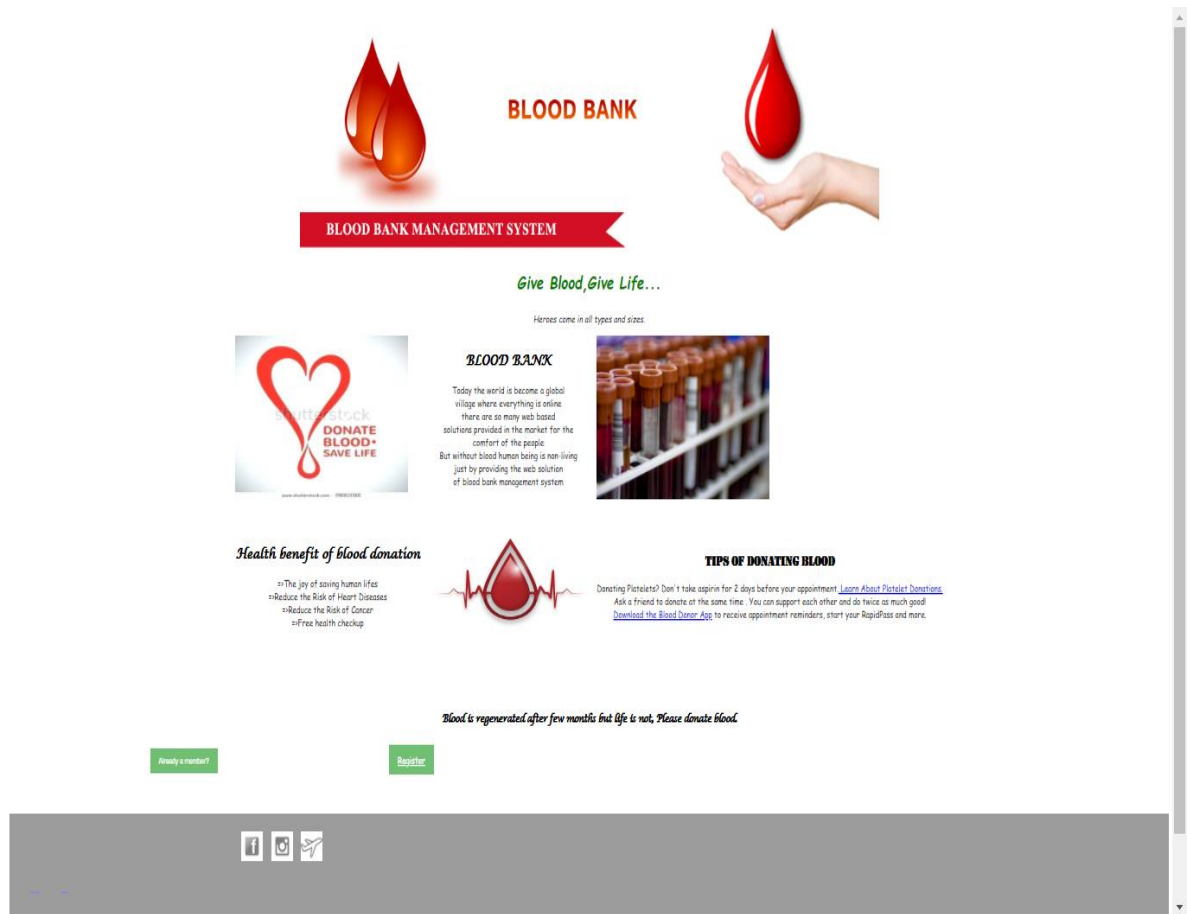
MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
Enter choice : 5
-----
**BLOOD_CELL SECTION**
-----
Blood Stocks
Blood Group : A+
Stock :0 units
Blood Group : A-
Stock :0 units
Blood Group : B+
Stock :750000 units
Blood Group : B-
Stock :0 units
Blood Group : O+
Stock :0 units
Blood Group : O-
Stock :0 units
Blood Group : AB+
Stock :0 units
Blood Group : AB-
Stock :500000 units

MAIN MENU
-----
1.DONOR  2. BLOOD STOCK  3.PATIENT  4.ORDER  5.BLOOD CELL  6.EXIT
-----
Enter choice : 6

```

Here it exits the execution and returns back to the terminal.

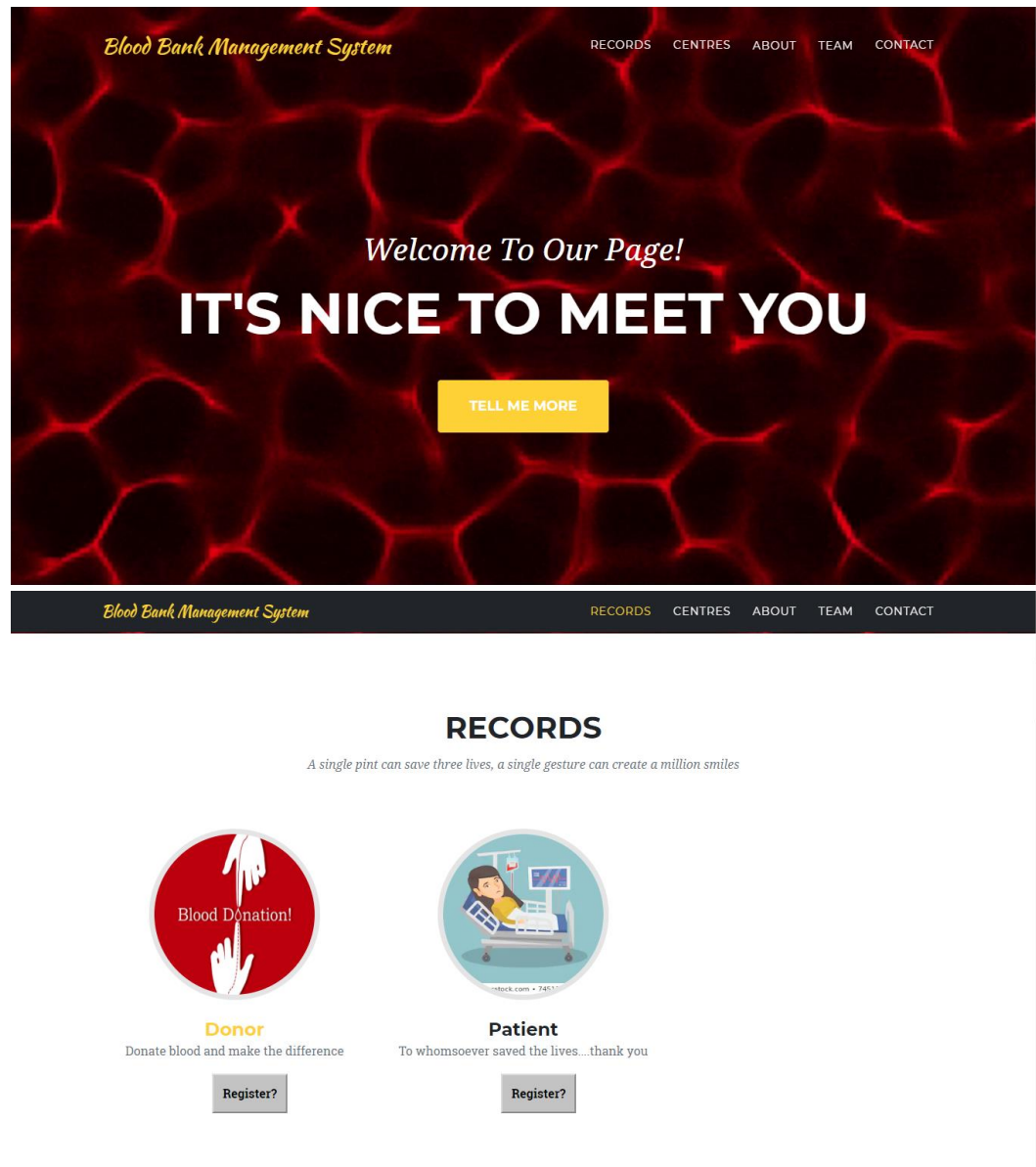
• Home interface



• Sign_up interface

The screenshot displays the sign-up interface. It features a central form titled "CREATE ACCOUNT" with the following fields: "Your Name", "Your Email", "Password" (with an eye icon for visibility), and "Repeat your password". Below these fields is a checkbox labeled "I agree all statements in [Terms of service](#)". A large blue button labeled "SIGN UP" is positioned below the checkbox. At the bottom of the form, there's a link "Have already an account ?" and a link "Login Here". A blue link "Go to home page" is located at the very bottom. The background of the interface is a dark red color with a pattern of red blood cells.

- **Admin interface**





ABOUT

There is no great joy than saving a soul.

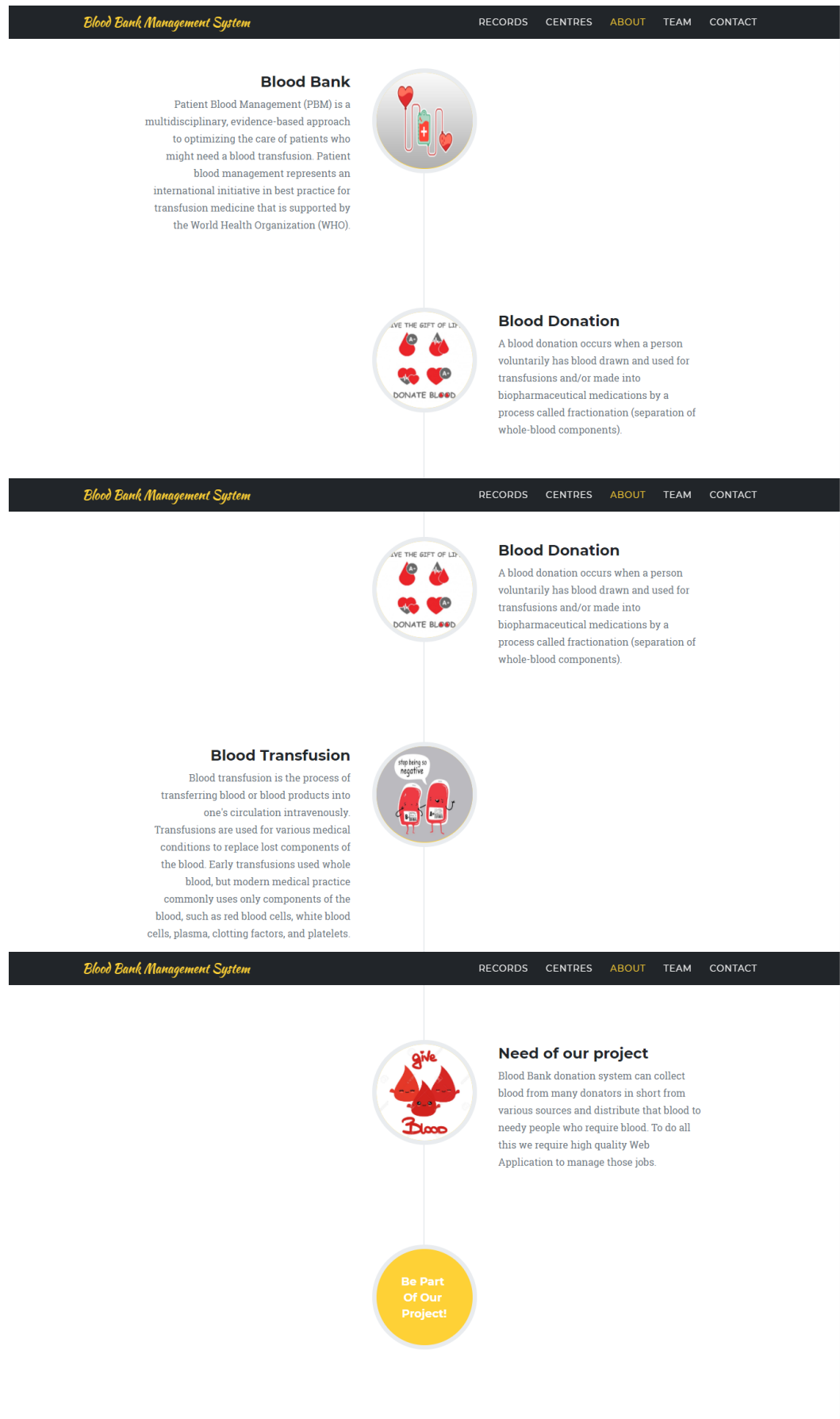
Blood Bank

Patient Blood Management (PBM) is a multidisciplinary, evidence-based approach to optimizing the care of patients who might need a blood transfusion. Patient blood management represents an international initiative in best practice for transfusion medicine that is supported by the World Health Organization (WHO).



Blood Donation

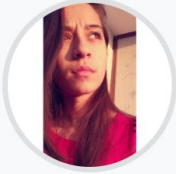
A blood donation occurs when a person



Blood Bank Management System
RECORDS CENTRES ABOUT TEAM CONTACT


OUR AMAZING TEAM

..Teamwork Makes the Dream Work..




Tanvi Trivedi
Programmer

[Twitter](#) [LinkedIn](#)



Jhanvi Zala
Programmer

[Twitter](#) [LinkedIn](#)



Meshva Gupta
Website Developer




[Twitter](#) [LinkedIn](#)

Don't let fools or mosquitoes suck your blood, put it to good use

-Donate blood....Save a life



Blood Bank Management System
RECORDS CENTRES ABOUT TEAM CONTACT

With help of....

WIKIPEDIA
The Free Encyclopedia

Under Guidance of....

Blood Bank Management System
RECORDS CENTRES ABOUT TEAM CONTACT

CONTACT US

Your Name *
Your Email *
Your Phone *

Your Message *

[SEND MESSAGE](#)

Copyright © Your Website 2019

[Twitter](#) [Facebook](#) [LinkedIn](#)

[Privacy Policy](#) [Terms of Use](#)

REFERENCES: (3 References Should Be There)

- <https://www.academia.edu › Blood Bank Management System A Project>
- <https://www.freeprojectz.com › project-report>

CONCLUSION:

- Hereby, we conclude that In the world of information technology where whole world becomes global village, where end user can get the information just sitting at home on one click, infact government has taken a step in order to transform the system. Blood Bank Management information system helps to make the system paperless. It is small contribution of the researcher in order to serve the mankind. This is to make sure that the management of the blood stock became effective, systematic and meeting user requirements.