FINAL PROJECT SUBMISSION

Blood Bank Management System

Course Instructor: Dr.Ashish Phophalia

&

Dr.Novarun Deb

Course Code: CS261

Group members:-

CHINIGEPALLI CHAKRADHAR SRINIVAS -201951048

ABHIYANK RAJ TIWARI -201951011

GANUGACHINTALA RAJA SEKHAR -201951062

SYED SHAHID NAZEER -201951159

Group No:-38

CONTEXT:-

- What is Blood Bank Management System?
- Existing System
- Problem Statement
- Proposed Solution
- Solution In Detail
- Function Specification
- Details for various Users
- Primary Code
- Code for virtual representation
- Snapshots of output
- UML Diagrams

What is a Blood bank management system?

- The project blood bank management system is known to be a pilot project that is designed for the blood bank to gather blood from various sources and distribute it to the needy people who have high requirements for it
- The system is designed to handle the daily transactions of the blood bank and search the details when required.
- It also helps to register the details of donors, blood collection details.
- The system is designed in such a manner that it can suit the needs of the blood bank requirements in the course of the future.

Existing System:-

- The operation of the blood bank still now is maintained in the manual system.
- The operation is tedious, time-consuming, and space-consuming.
- It creates room for errors as the data is entered manually by the persons
- It includes the risk of the documents being lost over years and maintenance of the records is difficult.
- The data recorded during testing or while acquiring the details of different aspects of blood bank management system is not so accurate and precise.
- Maintaining the stock of blood and daily transactions without computerization also poses a challenge.

Problem statement:-

Despite advances in technology, The current blood bank system is still running in a manual system. As such, there is a prevalent problem in the availability of needed blood types, and the problems penetrated in the current system are discussed below

- Scarcity of rare blood group. Unavailability of blood during an emergency.
- Deaths due to lack of blood during operations.
- The Blood Bank Management System project aims to make all the procedures automated and therefore with a computer system it can be more fast and accurate.
- This project concentrate and is set to manage all these cumbersome jobs.

Proposed Solution:-

In this project, the following solution is given

- We are providing a better platform for the users to view the nearest blood donors, blood banks anywhere anytime in the world.
- It keeps track of previous donor data which saves lives in absence of a rare blood group.
- It acts as a connecting bridge between patient and donor and blood bank. It also displays the data required in all possible ways.
- This program has different classes for different users and it also has different operations that they can perform.
- It stores the data using function handling and also saves human effort and time.

Solution In detail:-

The users on the login page:-

1. *Administrator*
2. *Blood bank employee*
3. *Blood Donor*
4. *Patient*

Whenever you choose the option the corresponding class is called and its menu will be popped up as shown below
Admins have a menu

1. *Administrator data*
2. *Blood bank employee data*
3. *Feedback*
4. *Monthly Blood bank count*

The administrator data option will display all the details of admin data with its corresponding options like Editing data.
The Bloodbank employee data option will display all the details of Bloodbank employee data with its corresponding options like Editing data.
The feedback option enables one to read the feedback and suggestions given by Donors and patients.
The monthly Bloodbak count display a linear chart for no of blood banks in a month

A blood bank employee will have a menu like

1. *Blood blank employee*
2. *Blood and Donor data*
3. *Monthly Donation Count*
4. *Linear chart*

The Bloodbank employee data option will display all the details of Bloodbank employee data.
The Blood and Donor data option enables one to view their respective data and their corresponding options like adding, removing, and updating blood and Donor details.
The update list option in Blooddata removes the spoiled blood that is as per Biological norms if blood is unused for more than 42 days it gets spoiled so by choosing this option the unused blood is removed by storing its adopted date in a date array. This will also not accept the donation from a donor if the duration from his last donation is less than 56 days as it is not good for his health which can be controlled again using the date array which stores the last donated date of the donor
.
The monthly Donation count option enables the employee to add no donation on each day and stores for a month.

The linear chart option displays a linear chart of monthly donations in a year.
The Blood donor Discussed here is different from the donor details in the Blood bank employee class as The donor in the blood bank employee is the one which donated their blood and stored in the blood bank, and the donor details here are the one who is ready to donate their blood in emergency purposes and we store their each and every particular.

A Blood Donor will have a menu like

1. *Update and Display details*
2. *Make a Donation*
3. *Update my last donated date*

The Update and Display details usually provide users to view and edit their personal profile.
The make a Donation option enables the user to change their status that they are willing to donate blood.
The last 10 donated dates of donors are stored in a 2D date array in order to avoid multiple unwanted blood donations. And by choosing this he can edit his last donated date.
A patient will have a menu like

1. *Update and display details*
2. *Make a request for blood*

The Update and Display details usually provide users to view and edit their personal profile.
The request option enables to request blood, the user has an option to request even for him or for his friend or relatives or to anyone. The Donors and Blood details will be displayed in the nearer first this will be achieved by Srno od blood bank and Closest class which displays the nearer first.

Some more important points:-

- The feedback and suggestions are taken from the patient and blood bank every 30 days and store in an array. And we also maintain the feedback anonymously.
- We have used another package to plot the data that is to visualize the data and to represent it in form of linear plots.
- We have used a class named closest to display the nearer one first to the patient.

FUNCTION SPECIFICATION

---

1. *Admin*
2. *Blood bank Employee*
3. *Donor*
4. *Patient*

ADMIN:-

- Manage Registration for Admin and Blood bank employee
- Manage Blood bank and Admin information like (update, delete)
- Manage feedback given by Donor and Patient

BLOOD BANK EMPLOYEE:-

- Blood bank information view/update
- View Donor information

DONOR:-

- Manage Donor Information(profile)
- Add new Donation for Blood
- Give feedback

PATIENT:-

- Manage patient information(profile)
- Give the Request for the patient for blood

Details for various Users

Administrator Data

- *Name*
- *Position*
- *Mobile No*
- *Email ID*

Blood bank employee

- *ID of bank*
- *Name*
- *City*
- *Mobile no*
- *Email id*

Donor* :-

- *ID of the Bank*
- *Name*
- *AGE*
- *Address*
- *mobile no*
- *Last donated date*

Blood Bank Donor :-

- *ID of the Bank*
- *Name*
- *AGE*
- *Address*
- *mobile no*
- *Last donated date*
- *Status*

Patient:-

- *ID of the Bank*
- *Name*
- *AGE*
- *Address*
- *mobile no*

There are many other Data types used in this project for many other features which can be read from the comments.

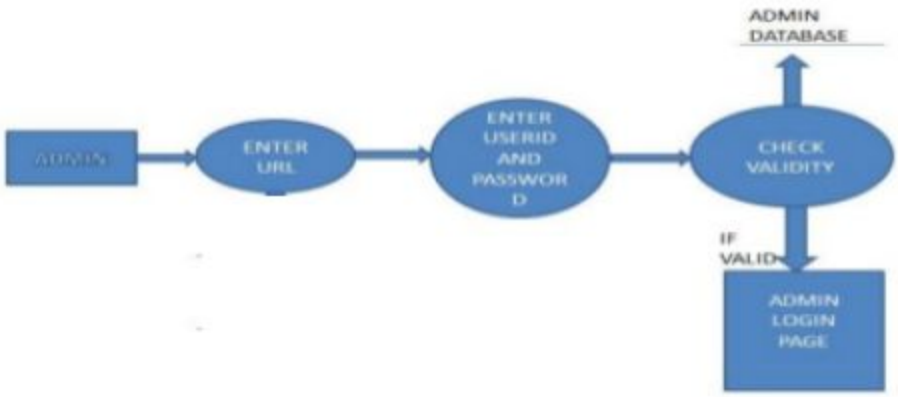
---

*As said above the Donor class is different from the Blood Donor class

DataBases requied:-

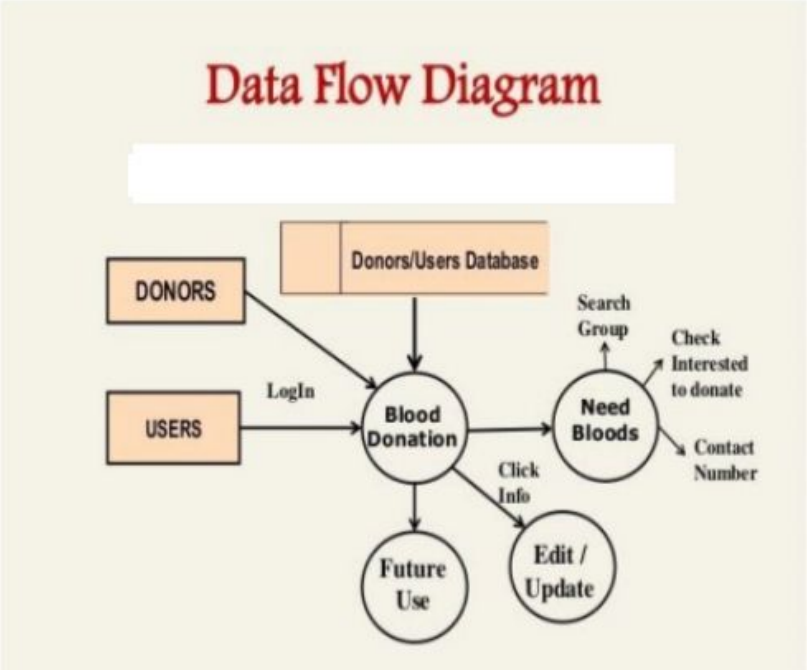
Here we maintain total 10 Data bases which has data for various classes and we are importing these data using filehandling class which has various functions to read and write data.

Diagramatic representation of login class-



Similarly every class login works in the same pattern

Diagramatic representation of Data Flow :-



PRIMARY CODE:-

```
package cs;
import java.util.*;
import java.io.*;
import java.text.*;//for the use of date class
import java.time.LocalDate;
import java.time.ZoneId;//for the use of time class
import sergio.*;//importing required classes from another package
import cu.*;//importing to draw linear chart
class administrator{
	public static String [][] admin = new String[10][2];//admin login details
	public static String [][] bloodbank = new String[10][2];//blood bank employee login details
	public static String [][] admindata = new String[10][4];//admin data
	public static String [][] bloodbankdata = new String[10][5];//blood bank employee data
	public static String [][] feedback = new String[10][8];//Feedback data
	public static int noofbloodbanks[][]=new int[2][12];//no of bloodbanks in a year monthwise
	administrator() //admin class constructor
		{
			filehandling obj = new filehandling();
			sergio object = new sergio();
			cu object1=new cu();
			admin=obj.readdata(0);//calling function in filehandling class to import
data from txt files
			bloodbank=obj.readdata(2);
			admindata=obj.readdata(1);
		bloodbankdata=obj.readdata(3);
			Scanner scan = new Scanner(System.in);
	login obj1 = new login(admin,admindata,0);//calling login class use of 0 is to avoid new
registration without authorisation
	while(true) {
	System.out.println("Choose the data to be verified:-");
	System.out.println("1)Administrator data");
	System.out.println("2)Blood bank employee data");
```

```
        System.out.println("3)Feedback");
        System.out.println("4)Monthly blood bank count");
        System.out.println("5)logout");
        int a= scan.nextInt();
switch(a){
case 1:
        System.out.println("Name   position  mobile no  mail id");
        cs.object1.Display(admindata);//calling display class in sergio package to display
        int b=0;
        while(b!=3){
        System.out.println("Choose the operation to be performed on Admin data:-");
        System.out.println("1)Edit the Data");
        System.out.println("2)Display the Data");
        System.out.println("3)Go back to previous menu");
        b=scan.nextInt();
        switch(b){
        case 1 :
        Editadmindata obj11 = new Editadmindata(admindata,admin,0);//calling editadmindata
class to edit data
        admindata=obj11.getarray();//calling function to get a return value frm a constructor
        admin=obj11.getarray(1);//calling function to get a return value frm a constructor
        continue;
        case 2:System.out.println("Name   position  mobile no  mail id");
                cs.object1.Display(admindata);
        continue;
        case 3 : continue;
        default :
                System.out.println("You have entered wrong key!!!");}}
        continue;
case 2:System.out.println("Srno    Name    city   mobile no  mail id");//calling nested
editblooddata class to edit data
        cs.object1.Display(bloodbankdata);
int b1=0;
while(b1!=3){
System.out.println("Choose the operation to be performed on Blood bank employee data:-");
System.out.println("1)Edit the Data");
System.out.println("2)Display the Data");
System.out.println("3)Go back to previous menu");
b1= scan.nextInt();
switch(b1){
case 1:
Editadmindata obj101=new Editadmindata(bloodbankdata,bloodbank,1);//use of 1 is to enable
some special functions
bloodbankdata=obj101.getarray();
```

```
bloodbank=obj101.getarray(1);
continue;
case 2:System.out.println("Srno    Name    city   mobile no  mail id");
            cs.object1.Display(bloodbankdata);
continue;
case 3:continue;
default :
        System.out.println("You have entered wrong key!!!");}}
continue;
case 3:
        System.out.println("Donors feedback");
   feedback=obj.readdata(4);//importing feedback data
   System.out.println("Q1 Q2 Q3 Q4 Q5 Q6 Q7 Suggestions");
   cs.object1.Display(feedback);
        obj.adddata(feedback, 4);
        feedback=obj.readdata(5);
System.out.println("Patient's feedback");
System.out.println("Q1 Q2 Q3 Q4 Q5 Q6 Q7 Suggestions");
cs.object1.Display(feedback);
continue;
case 4:object1.main(null);//using cu class to draw linearchart
continue;
case 5:obj.adddata(feedback, 5);//writing back data into file using filehandling classes
obj.adddata(admin, 0);
obj.adddata(admindata, 1);
obj.adddata(bloodbank, 2);
obj.adddata(bloodbankdata,3);
cs.main(null);
default :
        System.out.println("You have entered wrong key!!!");}}}}
class bloodbankemployee{
        //intilizing neccesary variables
        public static String bloodgroups[] = {"Bank
id","AB+ve","AB-ve","A+ve","A-ve","B-ve","B+ve","O-ve","O+ve"};
        public static String Donor[][] = new String[10][7];
        public static String Donor1[][] = new String[10][7];
        public static String bloodbank[][]=new String[10][2];
        public static String bloodbankdata[][]=new String[10][5];
        public static int Blooddata[][]=new int[10][9];
        public static int Blooddata1[][]=new int[1][9];
        public static int noofblooddonations[][]=new int[10][13];
        bloodbankemployee(){
                filehandling obj = new filehandling();
                sergio object = new sergio();
```

```
        cu object1=new cu();
                    bloodbank=obj.readdata(2);
                    bloodbankdata=obj.readdata(3);
            Donor=obj.readdata(6);
               Blooddata=obj.read(0);
            Scanner sc = new Scanner(System.in);
            login obj1 = new login(bloodbank,bloodbankdata,0);
            while(true) {
            System.out.println("Choose the data to be verified:-");
      System.out.println("1)Blood bank employee data");
            System.out.println("2)Donor and blood data");
            System.out.println("3)Monthly Donations");
            System.out.println("4)Linear Chart");
            System.out.println("5)logout");
            int a= sc.nextInt();
switch(a) {
case 1:System.out.println("Srno    Name    city   mobile no  mail id");
            cs.object1.Display(bloodbankdata);
continue;
case 2:System.out.println("Enter your Blood bank serial no:-");
String str=sc.next();
            int i,j=0,n=0;
            //getting length of array excluding null data in it to avoid error in compareTo function
for( i=0;i<Donor.length;i++){
            if(Donor[i][0]!=null)
                    n++;}
            for(i=0;i<n;i++)//getting particular donar data from a set of donar data
      {
            a=0;
            int m=str.compareTo(Donor[i][0]);
            if(a==m){
            Donor1[j]=Donor[i];
             j++;     }}
            for(i=0;i<Blooddata.length;i++)//getting particular blood data from a set of blood data
{
            int m=Integer.parseInt(str);//parse function to change string to int
            if(Blooddata[i][0]==m){
            Blooddata1[i]=Blooddata[i];}}
            System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last donated date");
            cs.object1.Display(Donor1);
int b1=0;
while(b1!=2){
System.out.println("Choose the operation to be performed on Donor data:-");
System.out.println("1)Edit the Data");
```

```
System.out.println("2)Blood data");
b1= sc.nextInt();
switch(b1){
case 1:EditDonordata obj101 = new EditDonordata(Donor1);
Donor1=obj101.getarray();
continue;
case 2: break;
default:System.out.println("You have entered wrong key!!!");}}
b1=0;
while(b1!=3){
System.out.println("Choose the operation to be performed on Blood data:-");
System.out.println("1)Edit the Data");
System.out.println("2)Display the Data");
System.out.println("3)Go back to primary menu");
b1= sc.nextInt();
switch(b1){
case 1:cs.object1.Display(bloodgroups);
cs.object1.Display(Blooddata1);
Blooddata1=EditDonordata.Editbloodata(Blooddata1,Donor1);//accessing function in a different
class
for(i=0;i<Donor1.length;i++) {
if(Donor1[i][0]==null);
break;}
for(j=0;j<Donor.length;j++) {
if(Donor[j][0]==null);
break;}
Donor[j]=Donor1[i-1];
continue;
case 2:cs.object1.Display(bloodgroups);
 cs.object1.Display(Blooddata1);
continue;
case 3: continue;
default:System.out.println("You have entered wrong key!!!");}}
continue;
case 3:noofblooddonations=obj.read(1);
int p;
Date date= new Date();//lines to store no of donations in a month for a year
LocalDate localDate = date.toInstant().atZone(ZoneId.systemDefault()).toLocalDate();
int month = localDate.getMonthValue();
for(p=0;p<noofblooddonations.length;p++) {
        int u=Integer.parseInt(Donor1[0][0]);
        if(u==noofblooddonations[p][0])
                        break;}
System.out.println("Enter your day donations count");
```

```
noofblooddonations[p][month]=noofblooddonations[p][month]+sc.nextInt();
obj.add(noofblooddonations, 1);
continue;
case 4:System.out.println("Linear chart representing no of donations month wise");
object1.main(null);
continue;
case 5:
obj.adddata(Donor, 6);
obj.add(Blooddata,0);
cs.main(null);
default :System.out.println("You have entered wrong key!!!");}}}}
class blooddonor{
        Date date11[][] = new Date[10][10];//2D Date array storing dates of donations of donors
        public static String blooddonor[][]=new String[10][2];
        public static String blooddonordata[][]=new String[10][8];
        public static String blooddonordata1[][] = new String[10][8];
        blooddonor(){
        sergio object = new sergio();
        filehandling obj = new filehandling();
        blooddonor=obj.readdata(7);
        blooddonordata=obj.readdata(8);
        Scanner sc = new Scanner(System.in);
        login objx = new login(blooddonor,blooddonordata,7);//calling login class use of 7 is to
write in file
        System.out.println("Please enter your credentials for two step verification");
        System.out.println("Enter the your login id:-");
        String str1=sc.next();
        System.out.println("Enter the your password:-");
        String str2=sc.next();
int n=0,j=0,i;
        for( i=0;i<blooddonordata.length;i++){
                if(blooddonordata[i][0]!=null)  {
        n++;}}
                for( i=0;i<n;i++)//getting particular donar data from a set of donar data
           {
                int a=0;
                int m=str1.compareTo(blooddonordata[i][1]);
                if(a==m){
                blooddonordata1[j]=blooddonordata[i];
                 break; }}
                //getting feedback from user for every 30 days
                        Date date111= new Date();
                        long duration1  = date111.getTime() - cs.joined[i].getTime();//joined Date
array consists of joining dates of Donors
```

```
                    int numofDays1 = (int) (duration1 / (1000 * 60 * 60 * 24));
                    int a1=30;
                    if(numofDays1>a1){
                    System.out.println("Please give you feedback(yes/no)");
                if("yes".equals(sc.next())) {
                String feed[][]=new String[10][8];
                feed=obj.readdata(6);
        feedback objf = new feedback(feed);
        obj.adddata(feed, 6);
                a1=a1+30;
                }
                else {
                        System.out.println("Dont forget to give the feedback");
        }}
        while(true) {
                object.Display(blooddonordata1);
        System.out.println("Choose the operation to be performed :-");
        System.out.println("1)Update my details");
        System.out.println("2)Display my details");
        System.out.println("3)Make a donation");
        System.out.println("4)Update my last donated date");
        System.out.println("5)logout");
        int j1 = 0;
        int a=sc.nextInt();
        switch(a){
        case 1:
        Editmydata o = new Editmydata(blooddonor,blooddonordata,0);
        blooddonordata=o.getarray();//calling function to get a return value frm a constructor
        blooddonor=o.getarray(1);//calling function to get a return value frm a constructor
        continue;
        case 2:System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last donated
date Status");
        cs.object1.Display(blooddonordata1);
        continue;
        case 3:System.out.println("Are u sure u want to make a donation ");//lines to change the
status of donor
String str=sc.next();
Date date = new Date();
if(str.equals("yes")) {
        for(j1=0;j1<date11[0].length;j1++)
if(date11[i][j1]==null)
        break;
        long duration  = date.getTime() - date11[i][j1-1].getTime();
        int numofDays = (int) (duration / (1000 * 60 * 60 * 24));
```

```
        if(numofDays<56)//as per biological norms a person cannot donate blood before 56 days from his last donation
{
                System.out.println("You are not applicable from donation as he donated blood "+numofDays+" ago");
}
        else {
blooddonordata[i][7]="yes";
        }}
continue;
        case 4:Date donationdate = null;//updating the last donated date
                System.out.println("Enter your last donation date");
        blooddonordata[i][6]=sc.next();
        SimpleDateFormat formatter1=new SimpleDateFormat("dd/MM/yyyy");
                try {
                        donationdate=formatter1.parse(blooddonordata[i][6]);
                } catch (ParseException e) {
                        e.printStackTrace();
                }
                date11[i][j1]=donationdate;
                continue;
        case 5:obj.adddata(blooddonor, 7);
             obj.adddata(blooddonordata, 8);
                cs.main(null);
        }}}}
class patient{
        String [][] patient=new String[10][2];
        String [][] patientdata=new String[10][6];
        String [][] patientdata1=new String[10][6];
        public static String blooddonordata[][]=new String[10][8];
        public static int blooddata[][]=new int[10][7];
  patient(){
                sergio object = new sergio();
                filehandling obj = new filehandling();
                patient=obj.readdata(9);
                patientdata=obj.readdata(10);
                blooddonordata=obj.readdata(8);
                blooddata=obj.read(7);
                Scanner sc = new Scanner(System.in);
                login objx = new login(patient,patientdata,9);//calling login class use of 9 is to write in respective file
                System.out.println("Please enter your credentials for two step verification");
                System.out.println("Enter the your login id:-");
                String str1=sc.next();
```

```
                System.out.println("Enter the your password:-");
                String str2=sc.next();
        int n=0,j=0,i;
                for( i=0;i<patientdata.length;i++){
                        if(patientdata[i][0]!=null)         {
                n++;}}
                        for( i=0;i<n;i++)//getting particular patient data from a set of patient data
                   {
                        int a=0;
                        int m=str1.compareTo(patientdata[i][0]);
                        if(a==m){
                        patientdata1[j]=patientdata[i];
                         break; }}
//getting feedback from user for every 30 days
        Date date= new Date();
        long duration  = date.getTime() - cs.joined1[i].getTime();
        int numofDays = (int) (duration / (1000 * 60 * 60 * 24));
        int a=30;
        if(numofDays>a)
        {
        System.out.println("Please give you feedback(yes/no)");
if("yes".equals(sc.next())) {
        String feed[][]=new String[10][8];
        feed=obj.readdata(5);
feedback objf = new feedback(feed);
obj.adddata(feed,5);
a=a+30;
}
else {
        System.out.println("Dont forget to give the feedback");
}}
while(true) {
System.out.println("Srno    Name    Blood grp  Age  City  Mobileno");
object.Display(patientdata1);
                System.out.println("Choose the operation to be performed :-");
                System.out.println("1)Update my details");
                System.out.println("2)Check my details");
                System.out.println("3)Make a request for blood");
                System.out.println("4)logout");
                int j1 = 0;
                int a1=sc.nextInt();
                switch(a1){
                case 1:
                Editmydata o = new Editmydata(patient,patientdata,1);
```

```
                patientdata=o.getarray();//calling function to get a return value frm a constructor
                patient=o.getarray(1);//calling function to get a return value frm a constructor
                continue;
                case 2:System.out.println("Srno    Name    Blood grp  Age  City  Mobileno");
                        cs.object1.Display(patientdata1);
                continue;
                case 3:System.out.println("Do you want to search in your location or other
location?(yes/no)");
                String str="yes";//Displaying near Donors and near bloodbanks
                if(str.equals(sc.next())){
                   System.out.println("Choose the bloodgroup");

System.out.print("1)AB+ve\n"+"2)AB-ve\n"+"3)A+ve\n"+"4)A-ve\n"+"3)B-ve\n"+"4)B+ve\n"+"4)O-
ve\n"+"6)O+ve\n");
                int b=sc.nextInt();String h[]=
{"AB+ve","AB-ve","A+ve","A-ve","B-ve","B+ve","O-ve","O+ve"};
                System.out.println("Donors list:-");
                closest ob=new closest(blooddonordata,patientdata1[0][0],h[b+1]);//calling closest
class which displays the nearer bloodbank and Donor first
                System.out.println("Bloodbank list:-");
                closest ob1=new closest(blooddata,patientdata1[0][0],h[b+1]);}
                   else{System.out.println("Enter your nearer bloodbank serial no");
             String str11=sc.next();
                        System.out.println("Choose the bloodgroup");

System.out.print("1)AB+ve\n"+"2)AB-ve\n"+"3)A+ve\n"+"4)A-ve\n"+"3)B-ve\n"+"4)B+ve\n"+"4)O-
ve\n"+"6)O+ve\n");
                int b=sc.nextInt();String h[]=
{"AB+ve","AB-ve","A+ve","A-ve","B-ve","B+ve","O-ve","O+ve"};
System.out.println("Donors list:-");
                closest ob=new closest(blooddonordata,str11,h[b+1]);
                System.out.println("Bloodbank list:-");
                closest ob1=new closest(blooddata,str11,h[b+1]);}
                   continue;
                   case 4:obj.adddata(patient,9);
                   obj.adddata(patientdata,10);
                   obj.add(blooddata, 0);
                   obj.adddata(blooddonordata, 8);
                           cs.main(null);}}}}
class feedback{//feedback class to collect and store feedback from donor and patient
feedback(String feed[][]) {
        int i;
        for(i=0;i<feed.length;i++)
        if(feed[i][0]==null)
```

```
                break;
        Scanner sc = new Scanner(System.in);//several feedback questions
        System.out.println("Your feedback will be anonymous");      //we matain feedback to be
anonymous
System.out.println("Direction: Using the Likert scale given below, please tick or check the
corresponding box of each statement given.\r\n" +
                "          5- Strongly Agree\n" +
                "          4- Agree\n" +
                "   3- Neutral\n" +
                "   2- Disagree\n" +
                "          1- Strongly Disagree\n" +
                "");
System.out.println("BBMS provides a clear documentation about the blood donor and its blood
donation activities");
feed[i][0]=sc.next();
System.out.println("BBMS can search fast the list of possible blood donors through its donors'
files.");
feed[i][1]=sc.next();
System.out.println("BBMS can clearly monitor the availability of blood bags or products of all
blood types");
feed[i][2]=sc.next();
System.out.println("BBMS allows user to know easily the period of expiration of blood
bags/products.\r\n");
feed[i][3]=sc.next();
System.out.println("BBMS offers an organized and systematized filing or record system.");
feed[i][4]=sc.next();
System.out.println("BBMS provides easy to use, efficient, effective system to the users.");
feed[i][5]=sc.next();
System.out.println("BBMS acts as bridge between patient and donor");
feed[i][6]=sc.next();
System.out.println("Any suggestion to improve this system??");
feed[i][7]=sc.nextLine();
}}
class closest{//class to display the nearer bloodbank and donor first
        String [][] string2;
        int [][] string22;
        closest(String[][]string1,String str,String s){//constructor to display nearer Donor first
                int i,l=0;
                String str1="yes";
                for( i=0;i<string1.length;i++) {
if(str.equals(string1[i][0])) {
        if(s.equals(string1[i][2]))
        if(str1.equals(string1[i][7]))
        {    string2[l]=string1[i];
```

```
            l++;
            break;}
            else {
                    break;              }}}
int j=i;
++i;--j;  for(i=i;i<string1.length;i++) {
                  if(str1.equals(string1[i][7]))
                     if(s.equals(string1[i][2])){
                  string2[l]=string1[i];
                  ++l;}
                 first :for(j=j;j>=0;j--) {
                           if(str1.equals(string1[j][7]))
                           if(s.equals(string1[j][2])){
                           string2[l]=string1[j];
                          ++l;
                          --j;
                           break first;}}}
                     cs.object1.Display(string2);}
            closest(int[][]string1,String str,String s){//constructor to display nearer blood bank first
        int b;String h[]= {" ","AB+ve","AB-ve","A+ve","A-ve","B-ve","B+ve","O-ve","O+ve"};
                     for( b=0;b<9;b++) {
if(s.equals(h[b]))
{           break;}}
            int i,l=0;
                     int str1=Integer.parseInt(str);
                     for( i=0;i<string1.length;i++) {
        if(str1==string1[i][0]) {
            if(string1[i][b]>0) {
                  string22[l]=string1[i];
            l++;
            break;}
            else {
                     break;}}}
int j=i;
++i;
            --j;       for(i=i;i<string1.length;i++) {
                       if(string1[i][b]>0){
                  string22[l]=string1[i];
                  ++l;}
                 first :for(j=j;j>=0;j--) {
                           if(string1[j][b]>0){
                           string22[l]=string1[j];
                          ++l;
                          --j;
```

```
                    break first;}}}
                cs.object1.Display(string22);}}
class login//login class to verify login credentials
{
        login(String[][] check,String details[][],int k){
                String str1 = new String();
                String str2 = new String();
          Scanner scan = new Scanner(System.in);
                System.out.println("Enter the your login id:-");
                str1=scan.next();
                System.out.println("Enter your password:-");
                str2=scan.next();
                int i=0,n=1,m=1;
                try {
                for(i=0;i<check.length;i++){
                m=check[i][0].compareTo(str1);
                if(m==0)
                break;}
                }catch(NullPointerException e)
                {                       System.out.println("You have entered wrong mail id");
                }
                n=check[i][1].compareTo(str2);
                if(n==0)
                System.out.println("Welcome "+str1);
                else{
                        System.out.println("You have entered wrong password !!!"+str1);
                        if(k!=0) {
                        System.out.println("If you have not registered then click 1");
                        int a= scan.nextInt();
                        if(a==1){
                        newregistration(check,details,k);
                        cs.main(null);}}}}
    void newregistration(String[][] check,String details[][],int k)//registration for new user
    { int i;
         Scanner scan = new Scanner(System.in);
         for(i=0;i<check.length;i++) {
                if(check[i][0]==null)
                        break;}
        System.out.println("Enter your name");
        details[i][0]=scan.next();
        System.out.println("Enter your nearer blood bank id");
        details[i][1]=scan.next();
        System.out.println("Enter the mobile no of the person");
        details[i][2]=scan.next();
```

```
        System.out.println("Enter the email of the person");
        details[i][3]=scan.next();
        System.out.println("Enter your new login ID");
        check[i][0]=scan.next();
        System.out.print(
              "1. A password must have at least eight characters.\n" +
              "2. A password consists of only letters,special characters and digits.\n" +
              "3. A password must contain at least one upper case,lower case,special
character,digit \n" );
        System.out.println("Enter your password");
        String str1=scan.next();
        sergio object1 = new sergio();
        while(!object1.validpassword(str1))//calling password strength checker in sergio package
        {
                System.out.println("your password strength is not strong");
                System.out.println("Enter your password");
                str1=scan.next();}
        System.out.println("Confirm your password");
        String str2=scan.next();
        int m=str2.compareTo(str1);
        if(m==0){
        check[i][1]=str1;
        filehandling obj=new filehandling();
        obj.adddata(check,k);
        System.out.println("Details added successfully");
           obj.adddata(details,k+1);
        }
        else{
                System.out.println("The passwords you entered doesnt match!!!");}
        cs.main(null);}
    }
class filehandling{//consists of various functions to read and write data of various types
void adddata(String[][] board,int k)//function to add data into respective file
        {               String path="C:\\Users\\Chakradhar\\Desktop";
                if(k==0) {
                        path=path+"\\A.txt";
                }else if(k==1) {
                        path=path+"\\B.txt";
                        }else if(k==2) {
                                path=path+"\\C.txt";
                        }else if(k==3) {
                                path=path+"\\D.txt";
                        }else if(k==4) {
                                path=path+"\\E.txt";
```

```
                }else if(k==5) {
                        path=path+"\\F.txt";
                }else if(k==6) {
                        path=path+"\\G.txt";
                }else if(k==7) {
                        path=path+"\\H.txt";
                }else if(k==8) {
                        path=path+"\\I.txt";
                }else if(k==9) {
                        path=path+"\\J.txt";
                }else if(k==10) {
                        path=path+"\\K.txt";            }
    PrintWriter writer1;
        try {
                writer1 = new PrintWriter(path);
                writer1.print("");
                writer1.close();
        } catch (FileNotFoundException e) {
                e.printStackTrace();}
        StringBuilder builder = new StringBuilder();
        for(int i = 0; i < board.length; i++)           {
          for(int j = 0; j < board[0].length; j++)        {
            builder.append(board[i][j]+"");
            if(j < board.length )
              builder.append(" ");        }
          builder.append("\n");}
        BufferedWriter writer;
        try {
                writer = new BufferedWriter(new FileWriter(path));
                try {
                        writer.write(builder.toString());
                } catch (IOException e) {
                                                        e.printStackTrace();}
                try {
                        writer.close();
                } catch (IOException e) {
                        e.printStackTrace();}
        } catch (IOException e1) {
                System.out.println("");}}
void add(int [][] matrix,int k) {//function to add data into respective file
        String path="C:\\Users\\Chakradhar\\Desktop";
        if(k==0) {
                path=path+"\\AA.txt";
        }else if(k==1) {
```

```
                path=path+"\\AB.txt";
                }else if(k==2) {
                        path=path+"\\AC.txt";}
        PrintWriter writer1;
            try {
                writer1 = new PrintWriter(path);
                writer1.print("");
                writer1.close();
            } catch (FileNotFoundException e) {
                e.printStackTrace();            }
            try {
           BufferedWriter bw = new BufferedWriter(new FileWriter(path));
    for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                bw.write(matrix[i][j] + " ");}
            bw.newLine();       }
           bw.flush();
        } catch (IOException e) {System.out.print(""+e);}}
String[][] readdata(int k) {//function to read data from file
        String path="C:\\Users\\Chakradhar\\Desktop";
        if(k==0) {
            path=path+"\\A.txt";
        }else if(k==1) {
            path=path+"\\B.txt";
            }else if(k==2) {
                path=path+"\\C.txt";
            }else if(k==3) {
                path=path+"\\D.txt";
            }else if(k==4) {
                path=path+"\\E.txt";
            }else if(k==5) {
                path=path+"\\F.txt";
            }else if(k==6) {
                path=path+"\\G.txt";
            }else if(k==7) {
                path=path+"\\H.txt";
            }else if(k==8) {
                path=path+"\\I.txt";
            }else if(k==9) {
                path=path+"\\J.txt";
            }else if(k==10) {
                path=path+"\\K.txt";
            }       BufferedReader reader ;
        String board1[][] = new String[10][10] ;
```

```
 try {
reader = new BufferedReader(new FileReader(path));
                String line = "";
                int row = 0;
try {
                        while((line = reader.readLine()) != null){
                          String[] cols = line.split(" ");
                          int col = 0;
                          for(String  c : cols){
                            board1[row][col] = c;
                             col++;}
                          row++;}
                } catch (IOException e) {
                        e.printStackTrace();}
                try {
                        reader.close();
                } catch (IOException e) {
                        e.printStackTrace();}
        } catch (FileNotFoundException e) {
                e.printStackTrace();}
        return board1; }
int[][] read(int k) {//function to read data from file
        String path="C:\\Users\\Chakradhar\\Desktop";
        if(k==0) {
                path=path+"\\AA.txt";
        }else if(k==1) {
                path=path+"\\AB.txt";
                }else if(k==2) {
                        path=path+"\\AC.txt";}
Scanner sc;
int [][] myArray = new int[10][10];
try {
                int i=0,j=0;
                File f=new File(path);
                if(f.createNewFile()) {
                        System.out.println(path);}
                else {}
                if(f.length()==0) {        }
                else {
                        BufferedReader br = null;
                         try{
                        br = new BufferedReader(new FileReader(path));
                                                String contentLine = br.readLine();
                          while (contentLine != null) {
```

```
                                        String p=contentLine.trim();
                                        int d1=Integer.parseInt(p);
                                        if(j==10) {
                                                j=j-10;
                                                i++;
                                                myArray[i][j]=d1;
                                        }else {
                                                myArray[i][j]=d1;}
                                        j++;
                                contentLine = br.readLine();}
                        }catch(IOException e) {
                                System.out.println(e);          }}
                }catch(IOException e) {
                        System.out.println(e); }
return myArray;}}
class Editmydata{//class to edit donor and patient data with help of some constraints
        private static String Editdonordata[][];
        private static String Editdonor[][];
        Editmydata(String [][]Editdonordata,String[][]Editdonor,int h){
                sergio object1 = new sergio();
                Scanner sc = new Scanner(System.in);
                  this.Editdonordata=Editdonordata;
                  this.Editdonor=Editdonor;
                                        while(true){
                                        Scanner scan = new Scanner(System.in);
                                        System.out.println("Enter the operation to be performed on
data:-");
                                        System.out.println("1)Delete my acount");
                                        System.out.println("2)Update the details");
                                        System.out.println("3)Previous menu");
                                        System.out.println("4)logout");
                int a=scan.nextInt();
                                System.out.println("Please enter your credentials for two step
verification");
                                System.out.println("Enter the your login id:-");
                                String str1=sc.next();
                                System.out.println("Enter the your password:-");
                                String str2=sc.next();
        switch(a){
                                case 1:System.out.println("Are you sure you want delete your
account(yes/no)");
                                String str=sc.next();
                                if(str.equals("yes"))
                                 {
```

```
                        int b;
                for(int i=0;i<Editdonordata.length;i++) {
                        if(str1.equals(Editdonordata[i][0])) {
                                b=i;
                                for(i=i;i<Editdonordata.length;i++)
                                        Editdonordata[i-1]=Editdonordata[i];
                for(i=b;i<Editdonor.length;i++)
                        Editdonor[i-1]=Editdonor[i];}}
                        break;}
                else
                object1.Display(Editdonordata);
                continue;
                case 2:
                        if(h==0)
                        System.out.println("Srno    Name    Blood grp  Age  City  Mobileno
Last donated date status");
                        else
                                System.out.println("Srno    Name    Blood grp  Age  City
Mobileno");
                        int i;
                        for(i=0;i<Editdonordata.length;i++) {
                                if(str1.equals(Editdonordata[i][0])) {
            object1.Display(Editdonordata[i]);
            break;}}
                System.out.println("Choose the data to be updated");
                System.out.println("1.Age");
                System.out.println("2.Address");
                System.out.println("3.mobile number");
                int f=scan.nextInt();
                System.out.println("Enter the detail of the person");
                Editdonordata[i][f+2]=scan.next();
                System.out.println("Enter 1 to display the data");
                f =scan.nextInt();
                if(f==1)
                {       if(h==0)
                        System.out.println("Srno    Name    Blood grp  Age  City  Mobileno
Last donated date Status");
                        else
                                System.out.println("Srno    Name    Blood grp  Age  City
Mobileno ");
        object1.Display(Editdonordata);}
                        continue;
                        case 3:break;
                        case 4:filehandling obj=new filehandling();
```

```
                obj.adddata(Editdonordata,8);
                obj.adddata(Editdonor,7);
                                cs.main(null);
                        default :
                                System.out.println("You have entered wrong key!!!");}}}
        public String[][] getarray()      {
                return Editdonordata;}
        public String[][] getarray(int k){
                return Editdonor;}}
class Editadmindata{//class to edit admindata
private static String Editadmindata[][];
private static String adminlogin[][];
 Editadmindata(String Editadmindata[][],String adminlogin[][],int k){
          sergio object1 = new sergio();
          this.adminlogin=adminlogin;
                 this.Editadmindata=Editadmindata;
                String str1,str2;
                while(true){
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter the operation to be performed on data:-");
                System.out.println("1)Enter the details of new person");
                System.out.println("2)Delete the details of existing person");
                System.out.println("3)Update the details");
                System.out.println("4)logout");
                int a=scan.nextInt();
switch(a){
case 1:
int i,b;
if(k==0){
for(i=0;i<Editadmindata.length;i++) {
        if(Editadmindata[i][0]==null)
                break;}
System.out.println("Enter the name of the person");
Editadmindata[i][0]=scan.next();
System.out.println("Enter the position of the person");
Editadmindata[i][1]=scan.next();
System.out.println("Enter the mobile no of the person");
Editadmindata[i][2]=scan.next();
System.out.println("Enter the email of the person");
Editadmindata[i][3]=scan.next();}
else{//accessing nested clasess
Editbloodbankdata obj = new Editbloodbankdata(Editadmindata);
Editadmindata=obj.getarray();}
for(i=0;i<adminlogin.length;i++) {
```

```
        if(adminlogin[i][0]==null)
                break;}
System.out.println("Enter your new login ID");
adminlogin[i][0]=scan.next();
System.out.print(
      "1. A password must have at least eight characters.\n" +
      "2. A password consists of only letters,special characters and digits.\n" +
      "3. A password must contain at least one upper case,lower case,special character,digit \n"
);
System.out.println("Enter your password");
str1=scan.next();
while(!object1.validpassword(str1))//calling password strength checker in sergio package
{
        System.out.println("your password strength is not strong");
        System.out.println("Enter your password");
        str1=scan.next();}
System.out.println("Confirm your password");
str2=scan.next();
int m=str2.compareTo(str1);
if(m==0){
System.out.println("Details added successfully");
adminlogin[i][1]=str1;}
else{
        System.out.println("The passwords you entered doesnt match!!!");}
continue;
case 2:
        object1.Display(Editadmindata);
System.out.println("Enter the serial no of the person");
b=scan.nextInt();
if(b<3){
        System.out.println("You cannot delete primary person");
        break;}
int a1=b;
for(b=b;b<Editadmindata.length;b++){
for(i=0;i<Editadmindata[0].length;i++){
        Editadmindata[b-1][i]=Editadmindata[b][i];}}
for(a1=a1;a1<adminlogin.length;a1++){
for(i=0;i<adminlogin[0].length;i++){
        adminlogin[a1-1][i]=adminlogin[a1][i];}}
System.out.println("Enter 1 to display the data ");
int f =scan.nextInt();
if(f==1) {
        object1.Display(Editadmindata);}
continue;
```

```
case 3:object1.Display(Editadmindata);
System.out.println("Enter the serial no of the person");
b=scan.nextInt();
if(k==0){
System.out.println("Choose the data to be changed");
System.out.println("1.Position");
System.out.println("2.mobile number");
System.out.println("3.Email");}
else{
        System.out.println("Choose the data to be updated");
        System.out.println("1.Name");
        System.out.println("2.Address");
        System.out.println("3.mobile number");
        System.out.println("4.Email");}
        f=scan.nextInt();
System.out.println("Enter the detail of the person");
Editadmindata[b-1][f]=scan.next();
System.out.println("Enter 1 to display the data");
f =scan.nextInt();
if(f==1) {
        object1.Display(Editadmindata);}
continue;
case 4:filehandling  obj = new filehandling();
if(k==0)
        {obj.adddata(adminlogin, 0);
obj.adddata(Editadmindata, 1);}
if(k!=0)
{obj.adddata(adminlogin, 2);
obj.adddata(Editadmindata, 3);}
cs.main(null);
default :
        System.out.println("You have entered wrong key!!!");}}
}//functions to return the edited arrays in above constructor
        public String[][] getarray()      {
                return Editadmindata;}
        public String[][] getarray(int k){
                return adminlogin;}}
class Editbloodbankdata{//function to edit bloodbank data
         static String bloodbankdata[][];
         public static int noofbloodbanks[][]=new int[2][12];
         Editbloodbankdata(String bloodbankdata[][]){
                this.bloodbankdata=bloodbankdata;
int i;
Scanner scan = new Scanner(System.in);
```

```
for(i=0;i<bloodbankdata.length;i++) {
        if(bloodbankdata[i][0]==null)
                break;}
System.out.println("Enter the Sr.no of the blood bank");
bloodbankdata[i][0]=scan.next();
System.out.println("Enter the name of the blood bank");
bloodbankdata[i][1]=scan.next();
System.out.println("Enter the address of the blood bank");
bloodbankdata[i][2]=scan.next();
System.out.println("Enter the mobile no of the person");
bloodbankdata[i][3]=scan.next();
System.out.println("Enter the email of the person");
bloodbankdata[i][4]=scan.next();
filehandling obj1 = new filehandling();
noofbloodbanks=obj1.read(1);
Date date= new Date();//lines to increase count of no of bloodbanks in respective month
SimpleDateFormat formatter = new SimpleDateFormat("MM/dd/yyyy");
LocalDate localDate = date.toInstant().atZone(ZoneId.systemDefault()).toLocalDate();
int month = localDate.getMonthValue();
int y=noofbloodbanks[1][month];
y++;
noofbloodbanks[1][month] = y;
obj1.add(noofbloodbanks, 2);}
public String[][] getarray(){
        return Editbloodbankdata.bloodbankdata;}}
class EditDonordata{
        static String EditDonordata[][];
  EditDonordata(String EditDonordata[][])
        { sergio object1 = new sergio();
          this.EditDonordata=EditDonordata;
                                while(true){
                                Scanner scan = new Scanner(System.in);
                                System.out.println("Enter the operation to be performed on Donor
data:-");
                                System.out.println("1)Delete the details of existing person");
                                System.out.println("2)Update the details");
                                System.out.println("3)previous menu");
                                int a=scan.nextInt();
                switch(a){
                case 1:System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last
donated date");
                        object1.Display(EditDonordata);
                System.out.println("Enter the serial no of the person");
                int b=scan.nextInt();
```

```
                for(b=b;b<EditDonordata.length;b++){
                for(int i=0;i<EditDonordata[0].length;i++){
                        EditDonordata[b-1][i]=EditDonordata[b][i];}}
                System.out.println("Enter 1 to display the data ");
                int f =scan.nextInt();
                if(f==1)
                {System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last donated
date");
                        object1.Display(EditDonordata);}
                continue;
                case 2:System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last
donated date");
                        object1.Display(EditDonordata);
                System.out.println("Enter the serial no of the person");
                b=scan.nextInt();
                System.out.println("Choose the data to be updated");
                System.out.println("1.Age");
                System.out.println("2.Address");
                System.out.println("3.mobile number");
                f=scan.nextInt();
                System.out.println("Enter the detail of the person");
                EditDonordata[b-1][f+2]=scan.next();
                System.out.println("Enter 1 to display the data");
                f =scan.nextInt();
                if(f==1)
                {System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last donated
date");
                        object1.Display(EditDonordata);}
                continue;
                case 3: break;
                default :
                        System.out.println("You have entered wrong key!!!");}}}
        static String Editblooddata[][];
        static int Editblood1[][];
        static int[][] Editbloodata(int Editblood1[][],String Editblooddata[][])
        { //function to edit blood data of respective blood bank
                Scanner scan = new Scanner(System.in);
                EditDonordata=Editblooddata;
           Editblood1=Editblood1;
                filehandling obj1=new filehandling();
           String[] Bloodgropus = {"Bank id","AB+ve", "AB-ve","A+ve", "A-ve", "B+ve",
"B-ve","O+ve","O-ve"};
                Date date1[] = new Date[10];//date array to store date of recent donation
                Date date2[] = new Date[10];
```

```
                Date someday = new Date(System.currentTimeMillis() - 44*(1000L * 60L * 60L *
24L));
                date1[0]=someday;//storing some random date for existing users
                Date someday1 = new Date(System.currentTimeMillis() - 8*(1000L * 60L * 60L *
24L));
                date1[1]=someday1;
                date2=date1;
                while(true){
                        System.out.println("Choose the operation");
                        System.out.println("1)Add Blood");
                        System.out.println("2)Remove Blood");
                        System.out.println("3)Display");
                        System.out.println("4)Update the list");
                        System.out.println("5)logout");
                        int a=scan.nextInt();
  System.out.println("Choose the Blood group to be updated");
                System.out.println("1)AB+ve");
                System.out.println("2)AB-ve");
                System.out.println("3)A+ve");
                System.out.println("4)A-ve");
                System.out.println("5)B+ve");
                System.out.println("6)B-ve");
                System.out.println("7)O+ve");
                System.out.println("8)O-ve");
                int b=scan.nextInt();
      switch(a){
case 1:
int i;
System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last donated date");
cs.object1.Display(EditDonordata);//accessing object in main class
System.out.println("Choose the Donor from the data if new donor press 0");
a=scan.nextInt();
if(a==0){
        for(i=0;i<EditDonordata.length;i++) {
                if(EditDonordata[i][0]==null)
                        break;}
        System.out.println("Enter the serial number of your bank");
        EditDonordata[i][0]=scan.next();
   System.out.println("Enter the name of the person");
        EditDonordata[i][1]=scan.next();
        System.out.println("Enter the Blood group of the person");
        EditDonordata[i][2]=scan.next();
        System.out.println("Enter the mobile no of the person");
        EditDonordata[i][3]=scan.next();
```

```
            System.out.println("Enter the age of the person");
            EditDonordata[i][4]=scan.next();
            System.out.println("Enter the address of the person");
            EditDonordata[i][5]=scan.next();
            Date date = new Date();
            date1[i]=date;
            //storing date by changing its datatype to string in donor array
       SimpleDateFormat formatter = new SimpleDateFormat("MM/dd/yyyy");
       EditDonordata[i][6] = formatter.format(date);
            System.out.println("Details added successfully");
            System.out.println("Enter 1)display the data");
            int f =scan.nextInt();
            if(f==1)
            {System.out.println("Srno    Name    Blood grp  Age  City  Mobileno Last donated date");
                    cs.object1.Display(EditDonordata);}}
else{
            //checking no of days between his last day to today
            Date date = new Date();
            long duration  = date.getTime() - date1[a-1].getTime();
            int numofDays = (int) (duration / (1000 * 60 * 60 * 24));
            if(numofDays<56)//as per biological norms a person cannot donate blood before 56 days
from his last donation
{
                    System.out.println("He is not applicable from donation as he donated blood
"+numofDays+" ago");
          continue;                    }}
System.out.println("Enter the quantity of blood");
int ml=scan.nextInt();
Editblood1[0][b]=Editblood1[0][b]+ml;
System.out.println("Details added successfully");
System.out.println("Enter 1)display the data");
int f =scan.nextInt();
if(f==1) {
            cs.object1.Display(Bloodgropus);
            cs.object1.Display(Editblood1);}
continue;
case 2:
            cs.object1.Display(Bloodgropus);
            cs.object1.Display(Editblood1);
System.out.println("Enter the bloodgrp serial no");
b=scan.nextInt();
Editblood1[0][b-1]=Editblood1[0][b-1]-1;
System.out.println("Enter 1 to display the data ");
f =scan.nextInt();
```

```
if(f==1) {
        cs.object1.Display(Bloodgropus);
        cs.object1.Display(Editblood1);}
continue;
case 3:
        cs.object1.Display(Bloodgropus);
        cs.object1.Display(Editblood1);
continue;
case 4://updates the list
        int n=0;
        for(i=0;i<EditDonordata.length;i++){
        if(EditDonordata[i][0]!=null){
                n++;}}
   Date date = new Date();
   for(i=0;i<n;i++){
        int j,m=1;
        //getting the duration from his last donation date
        long duration  = date.getTime() - date2[i].getTime();
        int numofDays = (int) (duration / (1000 * 60 * 60 * 24));
        //if blood is not utilized for more than 42 days it gets spoiled
        if(numofDays>42){
                for(j=1;j<7;j++){
                        date2[i]=null;//removes spoiled blood
         m=Bloodgropus[j].compareTo(EditDonordata[i][2]);
        if(m==0)
        break;}
                if(m==0)
                Editblood1[0][j]=Editblood1[0][j]-1;}}
        System.out.println("The list was updated successfully");
continue;
case 5:
        cs.main(null);
default :
        System.out.println("You have entered wrong key!!!");
return Editblood1;}}}
                        public String[][] getarray(){
                                return EditDonordata;}}
public class cs {//Assume the date array has some date intially to avoid null pointer exception
static  Date joined1[] = new Date[20];//array to store joining date of patients
static sergio object1 =new sergio(); //object used in every class
static Date joined[]=new Date[20]; //array to store joining date of donors
public static void main(String[] args){
Scanner scan = new Scanner(System.in);
 while(true)
```

```
{
   System.out.println("              Welcome to ABC Blood bank          ");
          System.out.println("             -------------------------          ");
          System.out.println("You want to login as");
          System.out.println("1)Administrator");
          System.out.println("2)Blood bank employee");
          System.out.println("3)Blood bank donor");
          System.out.println("4)Patient");
          int a=scan.nextInt();
          switch(a) {
          case 1:
                    administrator obj1 = new administrator();
               break;
          case 2:
                    bloodbankemployee obj2 = new bloodbankemployee();
          break;
          case 3:
                    blooddonor obj3=new blooddonor();
                    break;
          case 4:patient obj4=new patient();
              break;
          default :
                    System.out.println("You have entered wrong key!!!");}}}}
```

Another package which has most used methods:-

```
package sergio;
import java.util.*;
public class sergio{       //password checker to check the strength
  public static boolean validpassword(String password) {
       if (password.length() < 8)           {
          return false; }
       if (password.contains(" ")) {
          return false; }
       if (true) {
          int count = 0;
          for (int i = 0; i <= 9; i++) {
             String str1 = Integer.toString(i);
             if (password.contains(str1)) {
```

```
                count = 1; } }
            if (count == 0) {
                return false; }}
        if (!(password.contains("@") || password.contains("#")
            || password.contains("!") || password.contains("~")
            || password.contains("$") || password.contains("%")
            || password.contains("^") || password.contains("&")
            || password.contains("*") || password.contains("(")
            || password.contains(")") || password.contains("-")
            || password.contains("+") || password.contains("/")
            || password.contains(":") || password.contains(".")
            || password.contains(", ") || password.contains("<")
            || password.contains(">") || password.contains("?")
            || password.contains("|"))) {
            return false; }
        if (true) {
            int count = 0;
            for (int i = 65; i <= 90; i++) {
                char c = (char)i;
                String str1 = Character.toString(c);
                if (password.contains(str1)) {
                    count = 1; } }
            if (count == 0) {
                return false; }}
        if (true) {
            int count = 0;
            for (int i = 90; i <= 122; i++) {
                char c = (char)i;
                String str1 = Character.toString(c);
                if (password.contains(str1)) {
                    count = 1; } }
            if (count == 0) {
                return false; } }
  return true;     }
            int i,j,n=0,l=0;
            //Display the data
public void       Display(String Disp[][]){
            for(i=0;i<Disp[0].length;i++)
                if(Disp[0][i]!=null)
                        n++;
            for(i=0;i<Disp.length;i++)
                if(Disp[i][0]!=null)
                        l++;
            for(i=0;i<l;i++){
```

```
                for(j=0;j<n;j++){
                System.out.print(""+Disp[i][j]+"  ");}
                System.out.println("");}}
public void     Display(int Disp[][]){
        for(i=0;i<Disp[0].length;i++)
            if(Disp[0][i]!=0)
                n++;
        for(i=0;i<Disp.length;i++)
            if(Disp[i][0]!=0)
                l++;
        for(i=0;i<Disp[0].length;i++) {
            if(Disp[0][i]==0)
                break;              else
                        n++;}
        for(i=0;i<Disp.length;i++){
        if(Disp[i][0]==0)
        break;
        for(j=0;j<Disp[0].length;j++){
        System.out.print(""+Disp[i][j]+"  ");}
        System.out.println("");}}
public void     Display(String Disp[]){
                for(j=0;j<Disp.length;j++){
                System.out.print(""+Disp[j]+"    ");}
                System.out.println("");}
public void Display(int Disp[]){
                for(j=0;j<Disp.length;j++){
                System.out.print(""+Disp[j]+"   \t");}
                System.out.println("");}
public void     Display(String Disp[][][][]){}}
```

CODE FOR VIRTUAL REPRESENTATION:-

```
package cu;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
```

```
import java.util.Scanner;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.chart.LineChart;
import javafx.scene.chart.NumberAxis;
import javafx.scene.chart.XYChart;
import javafx.stage.Stage;
//Using interface to work on data
interface myinterface{
void start1();
}//class to draw linear chart
public class cu extends Application{
        Scanner sc = new Scanner(System.in);
        static int  k;
        public void start(Stage primaryStage) throws Exception{
        myinterface m = new myinterface() {//interface to implement linearchart
      public void start1( )
      {
       filehandling obj = new filehandling();
       System.out.println("Print any key for linear chart");
        k=sc.nextInt();
       if(k==0) {
       System.out.println("This is for no of blood banks");
       plot=obj.read(1);}
       else {
          System.out.println("This is for monthly Donations");
          plot=obj.read(2);}
      }
   };
   m.start1();
NumberAxis xAxis = new NumberAxis(0,12,1);
xAxis.setLabel("Months");
if(k==0)
NumberAxis yAxis = new NumberAxis(0,100,5);
```

```
yAxis.setLabel("No of Bloodbanks");
else{
NumberAxis yAxis = new NumberAxis(0,100,5);
yAxis.setLabel("No of Donations");}
LineChart lineChart = new LineChart(xAxis,yAxis);
XYChart.Series series1 = new XYChart.Series();
series1.setName("Linear Chart for no of Donations");
series1.getData().add(new XYChart.Data(1,plot[0][0]));
series1.getData().add(new XYChart.Data(2,plot[0][1]));
series1.getData().add(new XYChart.Data(3,plot[0][2]));
series1.getData().add(new XYChart.Data(4,plot[0][3]));
series1.getData().add(new XYChart.Data(5,plot[0][4]));
series1.getData().add(new XYChart.Data(6,plot[0][5]));
series1.getData().add(new XYChart.Data(7,plot[0][6]));
series1.getData().add(new XYChart.Data(8,plot[0][7]));
series1.getData().add(new XYChart.Data(9,plot[0][8]));
series1.getData().add(new XYChart.Data(10,plot[0][9]));
series1.getData().add(new XYChart.Data(11,plot[0][10]));
series1.getData().add(new XYChart.Data(12,plot[0][11]));
lineChart.getData().addAll(series1);
Group root = new Group(lineChart);
Scene scene = new Scene(root,550,450);
if(k!=0)
primaryStage.setTitle("LINEAR CHART FOR NO OF DONATIONS");
if(k==0)
primaryStage.setTitle("LINEAR CHART FOR NO OF BLOODBANKS");
primaryStage.setScene(scene);
primaryStage.show();
}
        static public int[][] plot;
public static void main(String []args) {
        launch(args);
}
}
//filehandling class to import various data from various txt files
```

```
class filehandling{
	boolean adddata(String[][] board,int k){
		String path="C:\\Users\\Chakradhar\\Desktop";
		if(k==0) {
			path=path+"\\A.txt";
		}else if(k==1) {
			path=path+"\\B.txt";
			}else if(k==2) {
				path=path+"\\C.txt";		}
		StringBuilder builder = new StringBuilder();
		for(int i = 0; i < board.length; i++){
		  for(int j = 0; j < board.length; j++){
		    builder.append(board[i][j]+"");
		    if(j < board.length )
		      builder.append(" ");}
		  builder.append("\n");}
		BufferedWriter writer;
		try {
			writer = new BufferedWriter(new FileWriter(path));
			try {
				writer.write(builder.toString());
			} catch (IOException e) {
				e.printStackTrace();
				return false;}
			try {
				writer.close();
			} catch (IOException e) {
				e.printStackTrace();
				return false;}
		} catch (IOException e1) {
			System.out.println("");
			return false;}
		return true;   }
	boolean add(int [][] matrix,int k) {
		String path="C:\\Users\\Chakradhar\\Desktop";
```

```
                if(k==0) {
                        path=path+"\\AA.txt";
                }else if(k==1) {
                        path=path+"\\AB.txt";
                        }else if(k==2) {
                                path=path+"\\AC.txt";}
  try {
            BufferedWriter bw = new BufferedWriter(new FileWriter(path));
            for (int i = 0; i < matrix.length; i++) {
              for (int j = 0; j < matrix[i].length; j++) {
                   bw.write(matrix[i][j] + ",");}
              bw.newLine();}
            bw.flush();
          } catch (IOException e) {
          return false;}
        return true;   }
String[][] readdata(int k) {
        String path="C:\\Users\\Chakradhar\\Desktop";
        if(k==0) {
                path=path+"\\AA.txt";
        }else if(k==1) {
                path=path+"\\AB.txt";
                }else if(k==2) {
                        path=path+"\\AC.txt";}
        BufferedReader reader ;
        String board1[][] = new String[10][10] ;
   try {  reader = new BufferedReader(new FileReader(path));
                String line = "";
                int row = 0;
                try {while((line = reader.readLine()) != null){
                        String[] cols = line.split(" ");
                        int col = 0;
                        for(String  c : cols) {
                          board1[row][col] = c;
                           col++;  }
```

```
                    row++;
                }                } catch (IOException e) {
                e.printStackTrace();}
            try {            reader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }        } catch (FileNotFoundException e) {
            e.printStackTrace();}
  PrintWriter writer1;
        try {writer1 = new PrintWriter(path);
            writer1.print("");
            writer1.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }        return board1;        }
int[][] read(int k) {
        String path="C:\\Users\\Chakradhar\\Desktop";
        if(k==0) {
            path=path+"\\AA.txt";
        }else if(k==1) {
            path=path+"\\AB.txt";
            }else if(k==2) {
                path=path+"\\AC.txt";}
 Scanner sc;
 int [][] myArray = new int[10][20];
try {
        sc = new Scanner(new BufferedReader(new FileReader(path)));
  while(sc.hasNextLine()) {
    for (int i=0; i<myArray.length; i++) {
      String[] line = sc.nextLine().trim().split(" ");
      for (int j=0; j<line.length; j++) {
        myArray[i][j] = Integer.parseInt(line[j]);
      }}}}catch (FileNotFoundException e) {
  }return myArray;}}
```

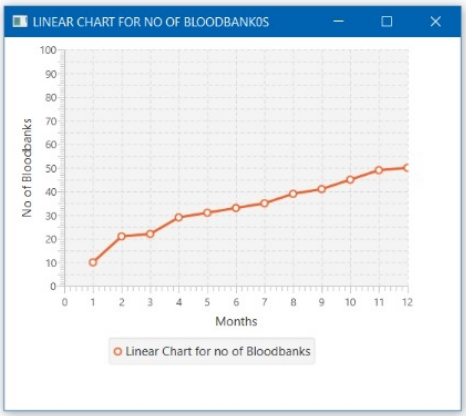
Snap shots of output

Administrator menu and functionality

```
cs [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe  (2 Dec, 2020 4:20:24 PM)
             Welcome to ABC Blood bank
            --------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
1
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Choose the data to be verified:-
1)Administrator data
2)Blood bank employee data
3)Feedback
4)Monthly blood bank count
5)logout
1
Name   position  mobile no  mail id
nazeeer Manager 8688953 manager@gmail.com
 vishal Director 5615615 Director@gmail.com
Choose the operation to be performed on Admin data:-
1)Edit the Data
2)Display the Data
3)Go back to previous menu
1
Enter the operation to be performed on data:-
1)Enter the details of new person
2)Delete the details of existing person
3)Update the details
4)logout
```

```
             Welcome to ABC Blood bank
            --------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
1
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Choose the data to be verified:-
1)Administrator data
2)Blood bank employee data
3)Feedback
4)Monthly blood bank count
5)logout
2
Srno    Name    city   mobile no  mail id
156156 ABBank Ongole 654848 AB@gmail.com
415645 CDBank chiral 565515 CD@gmail.com
Choose the operation to be performed on Blood bank employee data:-
1)Edit the Data
2)Display the Data
3)Go back to previous menu
1
Enter the operation to be performed on data:-
1)Enter the details of new person
2)Delete the details of existing person
3)Update the details
4)logout
```

```
            Welcome to ABC Blood bank
           ---------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
1
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Choose the data to be verified:-
1)Administrator data
2)Blood bank employee data
3)Feedback
4)Monthly blood bank count
5)logout
3
Donors feedback
Q1 Q2 Q3 Q4 Q5 Q6 Q7 Suggestions
4  5  3  4  5  4  3  NOSuggestions
3  5  2  4  4  1  3  NOSuggestions
Patient's feedback
Q1 Q2 Q3 Q4 Q5 Q6 Q7 Suggestions
4  5  3  4  5  4  3  NOSuggestions
3  5  2  4  4  1  3  NOSuggestions
Choose the data to be verified:-
1)Administrator data
2)Blood bank employee data
3)Feedback
4)Monthly blood bank count
5)logout
4
```



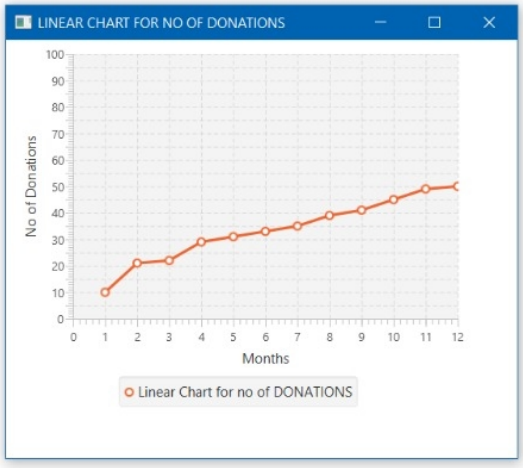
BLOOD BANK EMPLOYEE MENU AND FUNCTIONALITY

```
cs [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe  (2 Dec, 2020 4:50:23 PM)
            Welcome to ABC Blood bank
           ---------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
2
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Choose the data to be verified:-
1)Blood bank employee data
2)Donor and blood data
3)Monthly Donations
4)Linear Chart
5)logout
1
Srno    Name    city   mobile no  mail id
156156 ABBank Ongole 654848 AB@gmail.com
415645 CDBank chiral 565515 CD@gmail.com
Choose the data to be verified:-
1)Blood bank employee data
2)Donor and blood data
3)Monthly Donations
4)Linear Chart
5)logout
2
Enter your Blood bank serial no:-
143143
Srno    Name    Blood grp  Age  City  Mobileno Last donated date
143143 vinaay AB+ve  25 ongole 456656 16/09/2020
Choose the operation to be performed on Donor data:-
```

```
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Choose the data to be verified:-
1)Blood bank employee data
2)Donor and blood data
3)Monthly Donations
4)Linear Chart
5)logout
2
Enter your Blood bank serial no:-
143143
Srno    Name    Blood grp  Age  City  Mobileno Last donated date
143143 vinaay AB+ve  25 ongole 456656 16/09/2020
Choose the operation to be performed on Donor data:-
1)Edit the Data
2)Blood data
2
Choose the operation to be performed on Blood data:-
1)Edit the Data
2)Display the Data
3)Go back to primary menu
1
Bank id AB+ve AB-ve A+ve A-ve B-ve B+ve O-ve O+ve
143143 5 4 11 0 4 3 4 3
Choose the operation
1)Add Blood
2)Remove Blood
3)Display
4)Update the list
5)logout
```

cs [Java Application]

```
             Welcome to ABC Blood bank
            ---------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
2
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Choose the data to be verified:-
1)Blood bank employee data
2)Donor and blood data
3)Monthly Donations
4)Linear Chart
5)logout
4
Linear chart representing no of donations month wise
```



DONOR MENU AND FUNCTIONALITY

```
cs [Java Application]
             Welcome to ABC Blood bank
            ---------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
3
Enter the your login id:-
nikhil
Enter your password:-
nikhil
Welcome nikhil
Please enter your credentials for two step verification
Enter the your login id:-
nikhil
Enter the your password:-
nikhil
Srno    Name    Blood grp  Age  City  Mobileno Last donated date Status
143143 nikhil AB+ve  25 ongole 456656 16/09/2020 yes
Choose the operation to be performed :-
1)Update my details
2)Display my details
3)Make a donation
4)Update my last donated date
5)logout
```

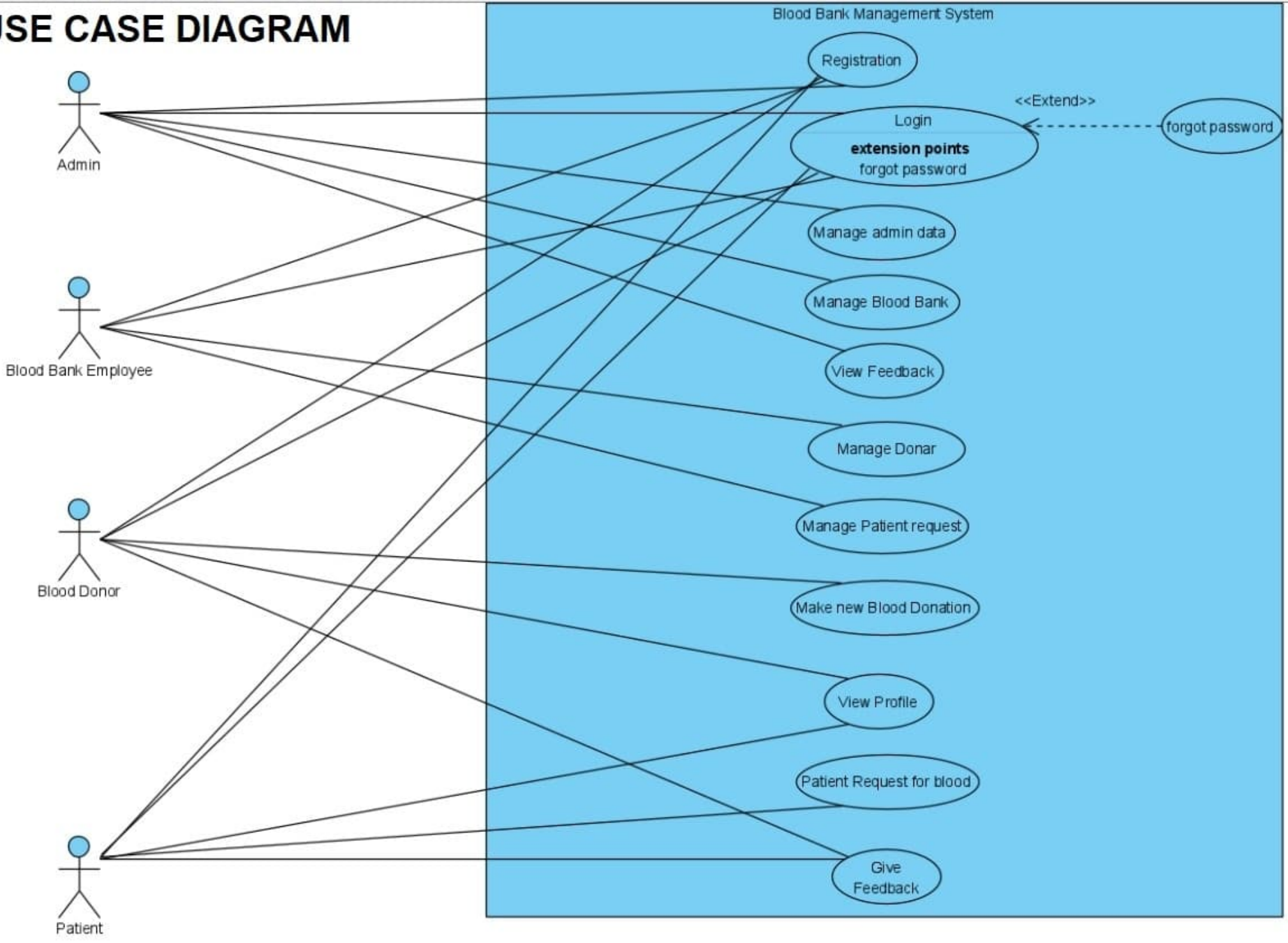
PATIENT MENU AND FUNCTIONALITY

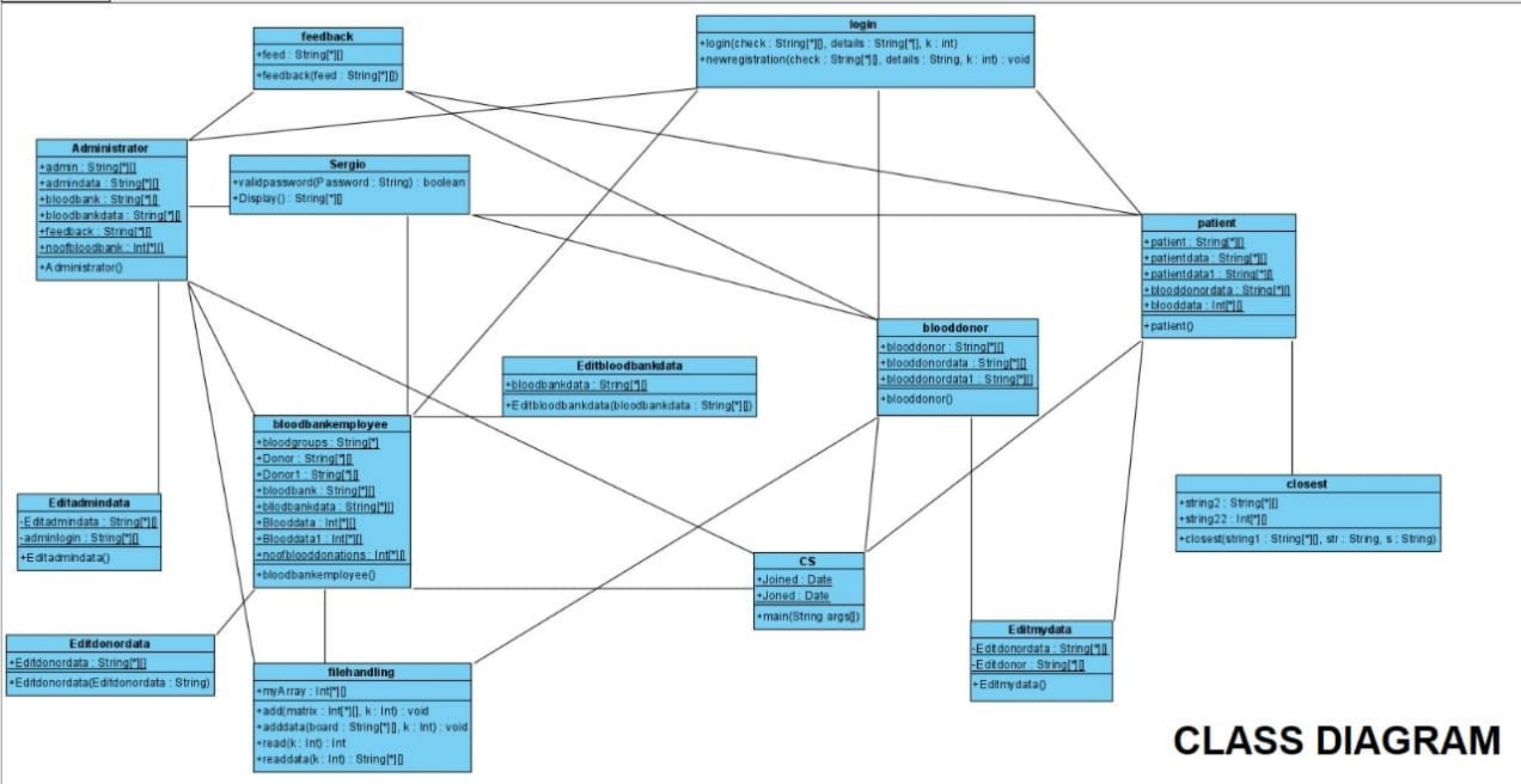
```
             Welcome to ABC Blood bank
            ---------------------------
You want to login as
1)Administrator
2)Blood bank employee
3)Blood bank donor
4)Patient
4
Enter the your login id:-
nikhil
Enter the your password:-
nikhil
Srno    Name    Blood grp  Age  City  Mobileno
143143  lucky    AB+ve     22  ongole  1651651
Choose the operation to be performed :-
1)Update my details
2)Check my details
3)Make a request for blood
4)logout
3
Do you want to search in your location or other location?(yes/no)
yes
Choose the bloodgroup
1)AB+ve/n2)AB-ve/n3)A+ve/n4)A-ve/n3)B-ve/n4)B+ve/n4)O-ve/n6)O+ve/n1
Donors list:-
143143 kiku AB+ve 42 mirz 11/06/2002 yes
143153 kiku AB+ve 42 mirz 11/06/2002 yes
143123 kiku AB+ve 42 mirz 11/06/2002 yes
143163 kiku AB+ve 42 mirz 11/06/2002 yes
Bloodbank list:-
143143 kiku mirz 116515 ch@
143153 kiku mirz 116515 ch@
143123 kiku mirz 116515 ch@
143163 kiku mirz 116515 ch@
```

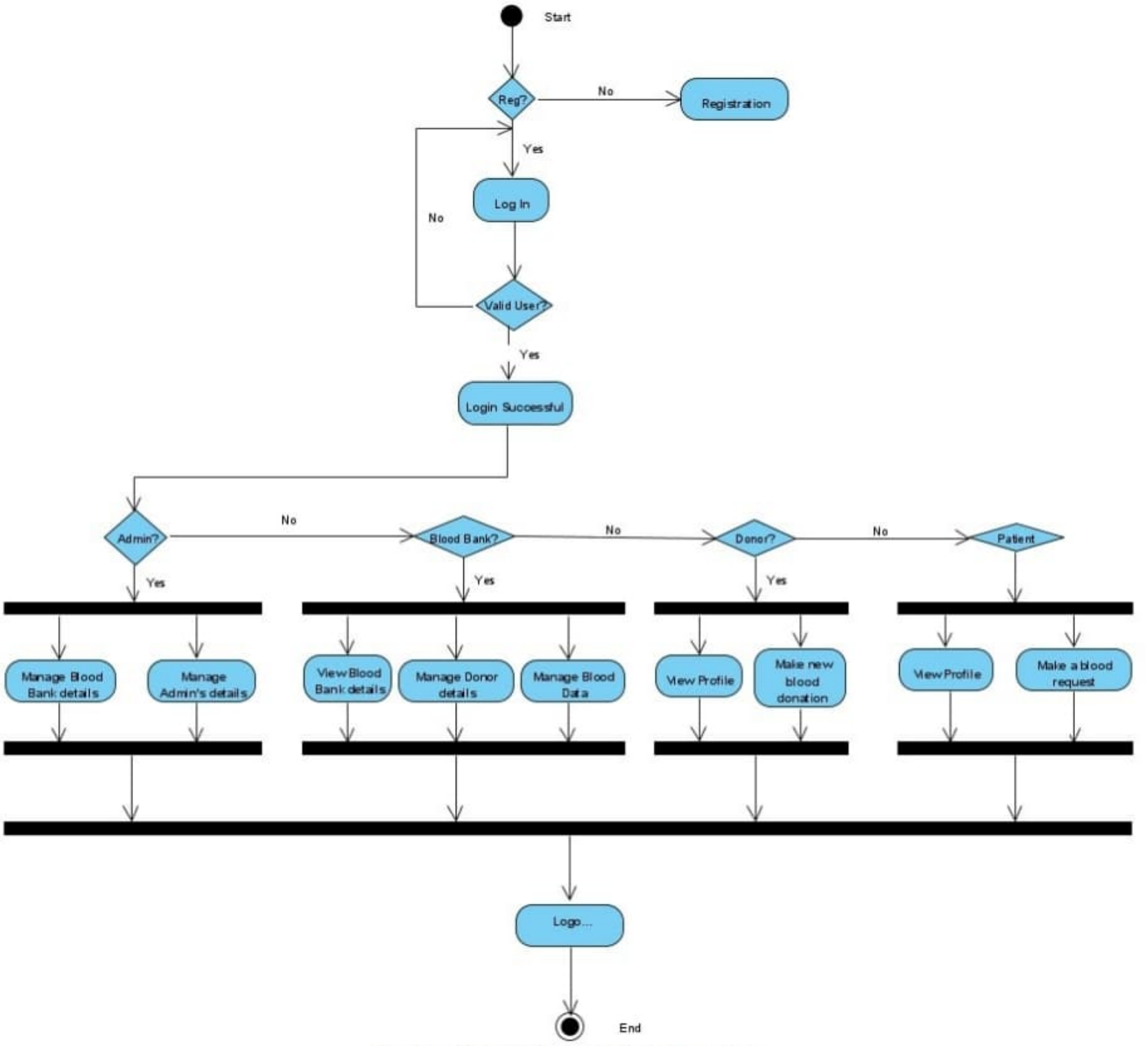
UML

DIAGRAMS

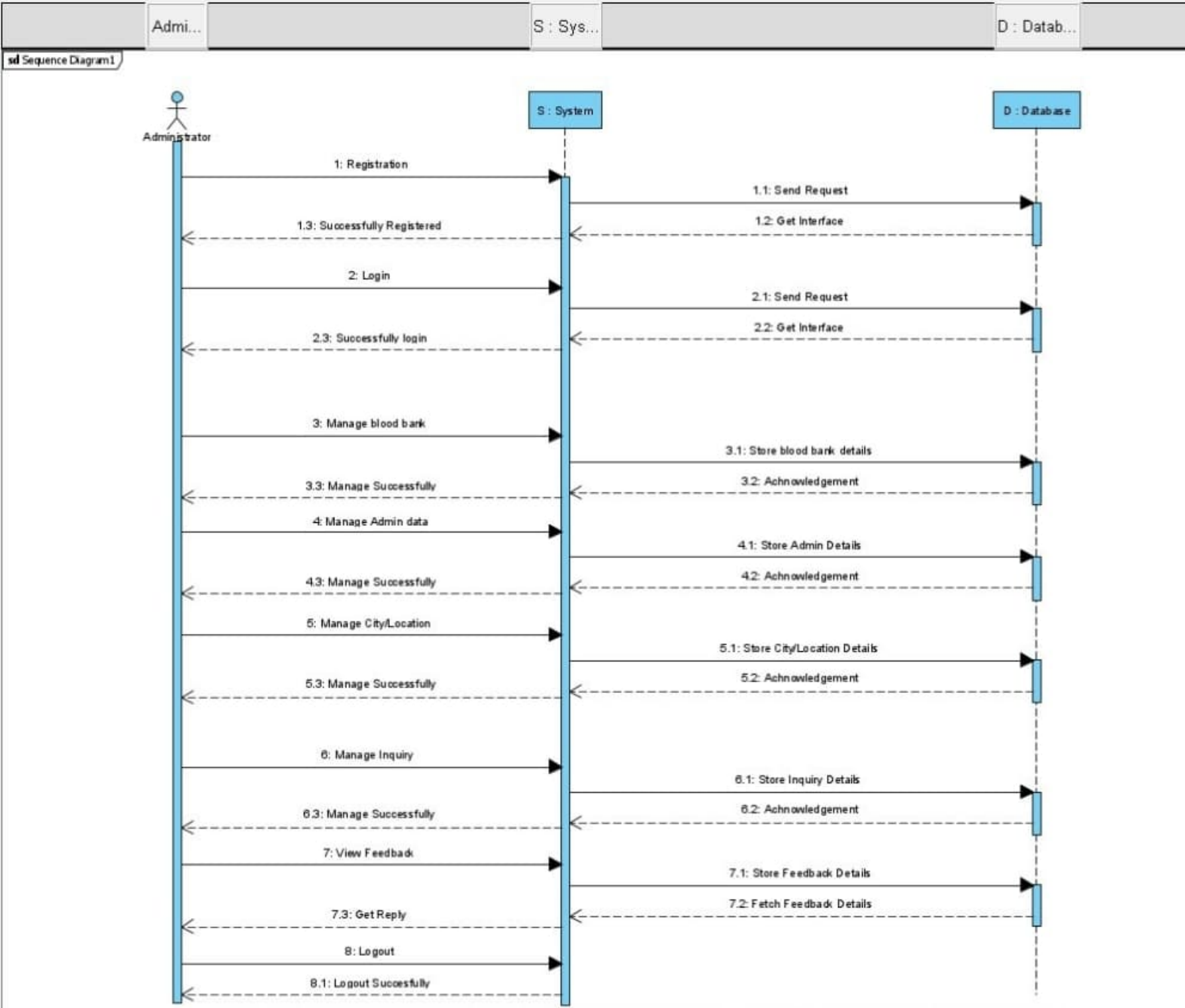
USE CASE DIAGRAM



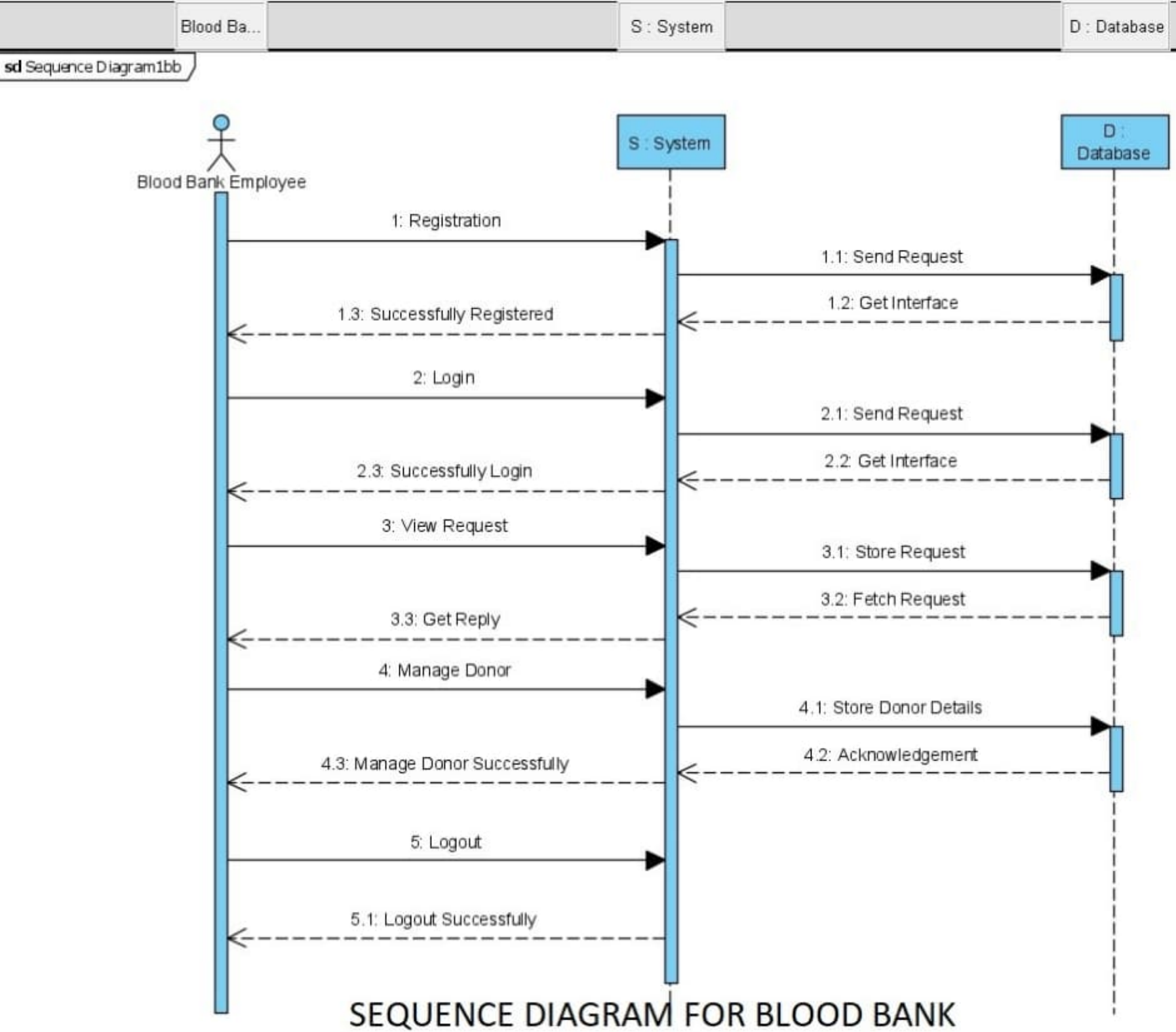

feedback
+feed : String[*][]
+feedback(feed : String[*][])
login
+login(check : String[*][], details : String[*], k : int)
+newregistration(check : String[*][], details : String, k : int) : void
Administrator
+admin : String[*][]
+admindata : String[*][]
+bloodbank : String[*][]
+bloodbankdata : String[*][]
+feedback : String[*][]
+noofbloodbank : Int[*][]
+Administrator()
Sergio
+validpassword(Password : String) : boolean
+Display() : String[*][]
patient
+patient : String[*][]
+patientdata : String[*][]
+patientdata1 : String[*][]
+blooddonordata : String[*][]
+blooddata : Int[*][]
+patient()
blooddonor
+blooddonor : String[*][]
+blooddonordata : String[*][]
+blooddonordata1 : String[*][]
+blooddonor()
Editbloodbankdata
+bloodbankdata : String[*][]
+Editbloodbankdata(bloodbankdata : String[*][])
bloodbankemployee
+bloodgroups : String[*]
+Donor : String[*][]
+Donor1 : String[*][]
+bloodbank : String[*][]
+bloodbankdata : String[*][]
+Blooddata : Int[*][]
+Blooddata1 : Int[*][]
+noofblooddonations : Int[*][]
+bloodbankemployee()
closest
+string2 : String[*][]
+string22 : Int[*][]
+closest(string1 : String[*][], str : String, s : String)
Editadmindata
-Editadmindata : String[*][]
-adminlogin : String[*][]
+Editadmindata()
CS
+Joined : Date
+Joned : Date
+main(String args[])
Editdonordata
+Editdonordata : String[*][]
+Editdonordata(Editdonordata : String)
filehandling
+myArray : Int[*][]
+add(matrix : Int[*][], k : Int) : void
+adddata(board : String[*][], k : Int) : void
+read(k : Int) : Int
+readdata(k : Int) : String[*][]
Editmydata
-Editdonordata : String[*][]
-Editdonor : String[*][]
+Editmydata()
CLASS DIAGRAM



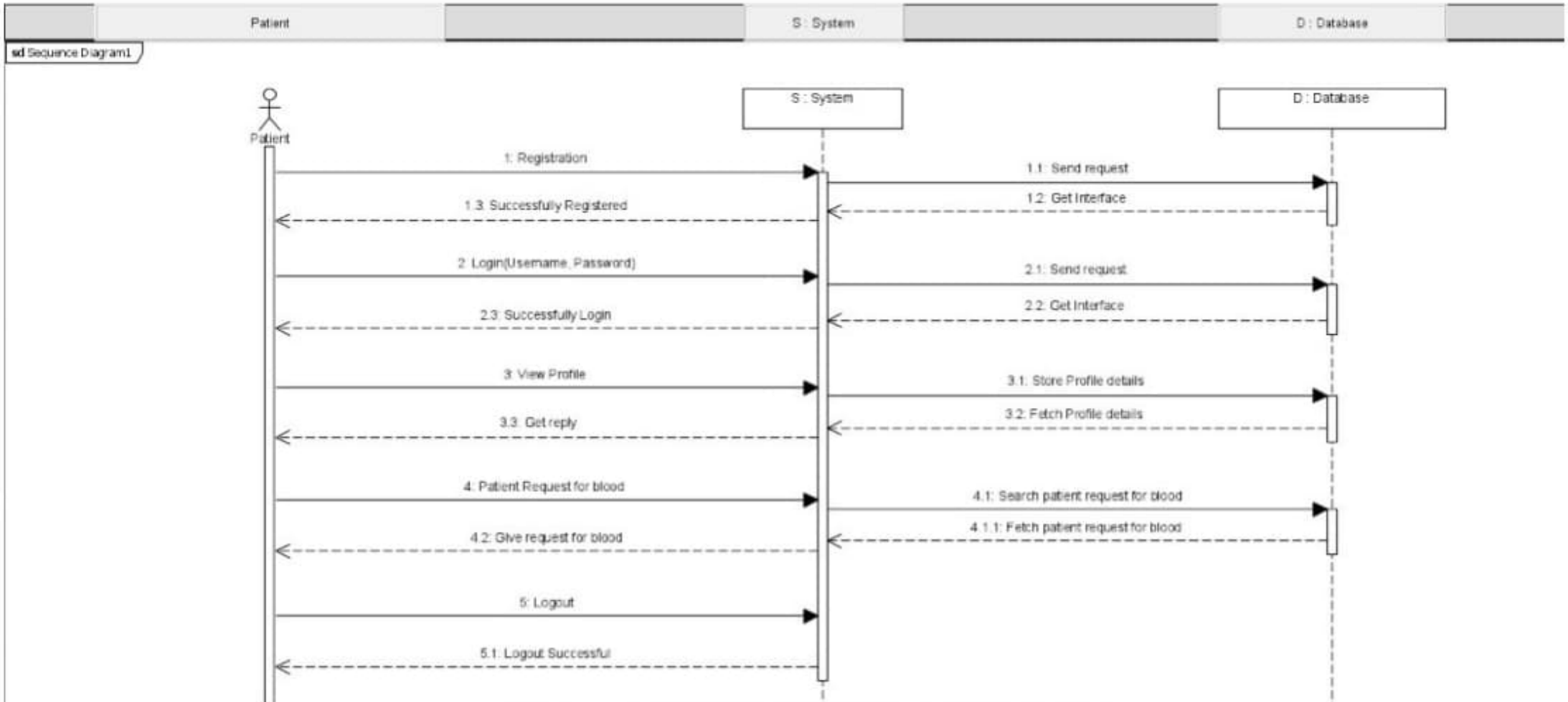
ACTIVITY DIAGRAM



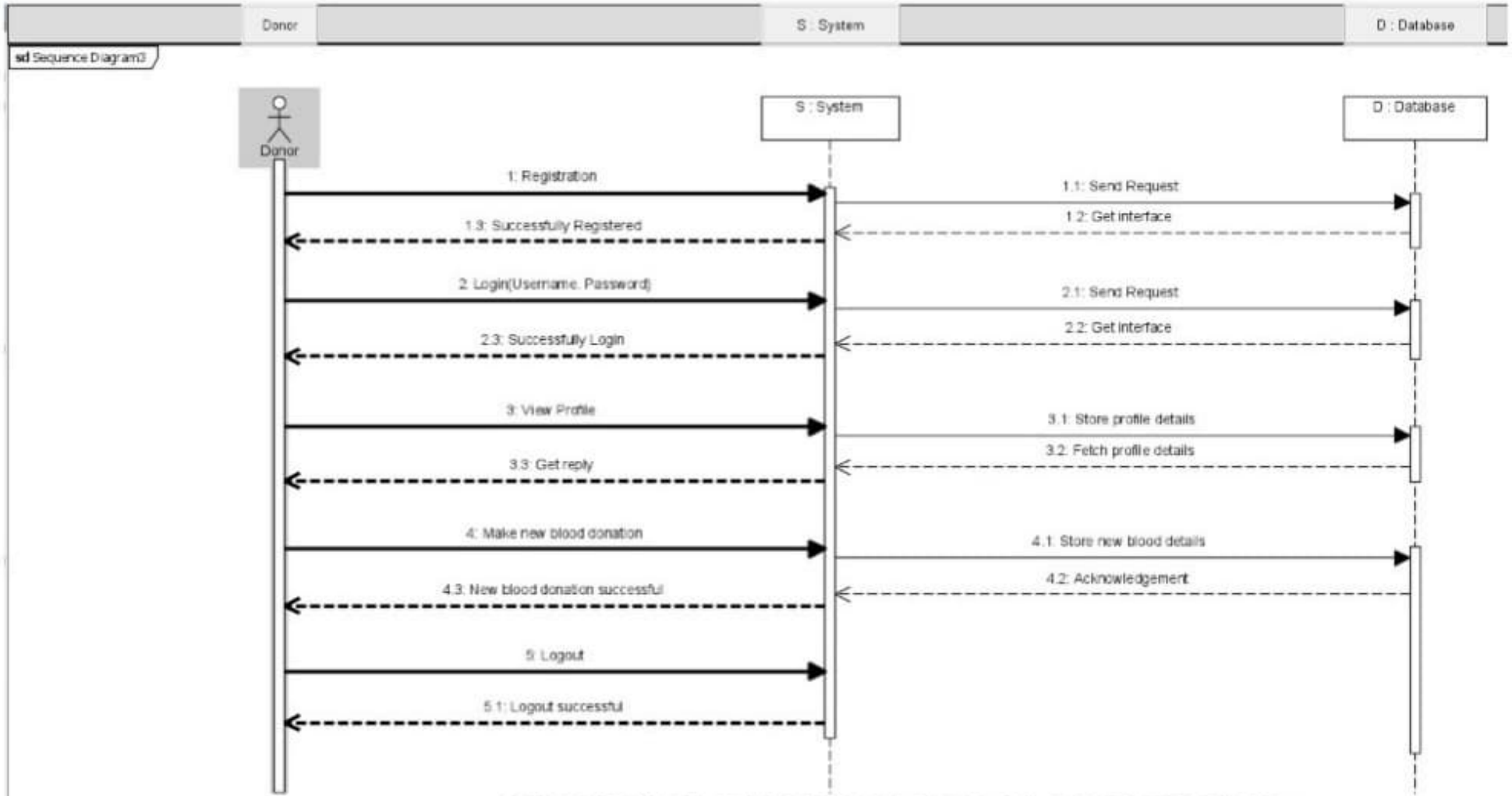
SEQUENCE DIAGRAM FOR ADMINISTRATOR



SEQUENCE DIAGRAM FOR BLOOD BANK



SEQUENCE DIAGRAM FOR PATIENT



SEQUENCE DIAGRAM FOR BLOOD DONOR

THANK

YOU