BLOOD BANK MANAGEMENT

A PROJECT REPORT

Submitted by

ASTHA JAISWAL [Reg No: RA2111042010025]

DHAWAL SAHU [Reg No: RA2111042010015]

SOURABH SHARMA [Reg No: RA2111042010007]

Under the Guidance of

Dr. PAUL T SHEEBA

(Assistant Professor, Department of Data Science and Business Systems)

In partial fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY COMPUTER SCIENCE AND BUSINESS SYSTEMS

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
NOVEMBER 2022

COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(under Section 3 of UGC Act,1956)
SRM NAGAR, KATTANKULATHUR -603 203
CHENGALPATTU DISTRICT

BONAFIDE CERTIFICATE

Register No:RA2111042010007

Certified to be the bonafide record of work done by SOURABH SHARMA of III Semester B.Tech **COMPUTER SCIENCE AND BUSINESS SYSTEMS** Degree course in the Practical **18CSC263J-OBJECT ORIENTED PROGRAMMING** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, Kattankulathur** during the academic year **2022-2023.**

Lab In-charge

Date: **Head of Department**

Submitted for University Examination held in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, Kattankulathur.**

Date: Examiner-1 Examiner-2

ABSTRACT

Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank.

This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way.

Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective. Our client is not interested in blood stocking instead we are stocking blood donors information. The donors who are interested in donating blood has to register in the database. There is no storage of blood so no complications in the project. The software is fully integrated with CRM (customer relationship management) as well as CMS (content management system) solution.

It is developed in a manner that is easily manageable, time saving and relieving one from manual works. The requirement of the blood has to be requested and we supply the information of the donor. The donors can update their status whether they are available or not.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop. •Our project aims to improvise the department of blood management system and its potential accessibility to the public. •Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free. •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.

THE HISTORY

In bullet points, describe the current situation. •Blood Bank Management systems maintain donor details, Blood component stock details, Screening details, component preparation details, Blood request details, Compatibility details, Blood issue details, Monthly statistics. •The current systems store information in an organized manner but are not that user friendly, • No proper coordination between different Applications and Users, •Cannot upload and download the latest updates art right time.

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc. •Manual documentation and data entry. •Only web-based system is available no app-based system available. •Not everybody will be able to use the system especially the computer illiterate people. •New donors will have to go to a hospital to examine their blood types.

APPROACH/ METHODOLOGY

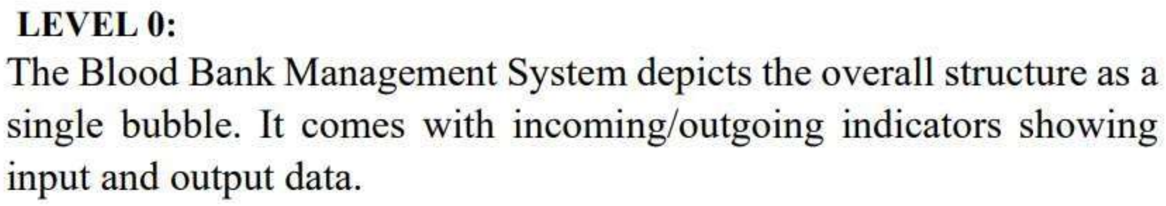
List what is needed to complete the project. •Collaborations with public and private sector hospitals and independent blood banks. •Legal Documentation. •Database system.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization. •The main advantage of a blood bank management system is easy and effective information retrieval. •The system will be very effective in times of emergencies as it saves time and efforts for searching blood donors. •Blood bank in charge can view the list of discarded blood units, they can also view the reason for which the blood units are discarded

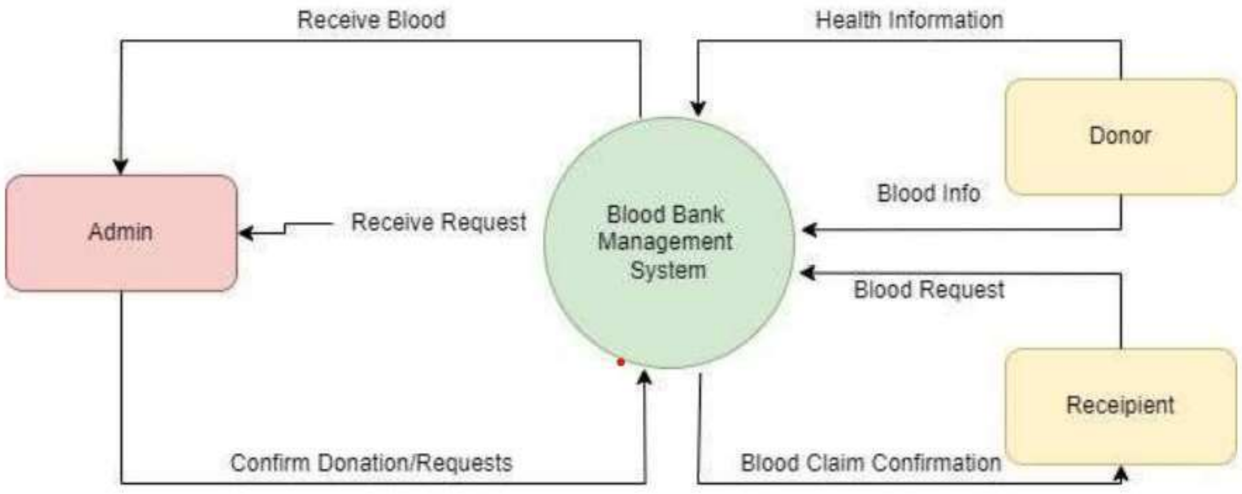
CHAPTER 2
System Design and implementation

2.1 ZERO LEVEL DATA FLOW DIAGRAM OF BLOOD BANK MANAGEMENT SYSTEM(0 LEVEL DFD):



LEVEL 0:

The Blood Bank Management System depicts the overall structure as a single bubble. It comes with incoming/outgoing indicators showing input and output data.

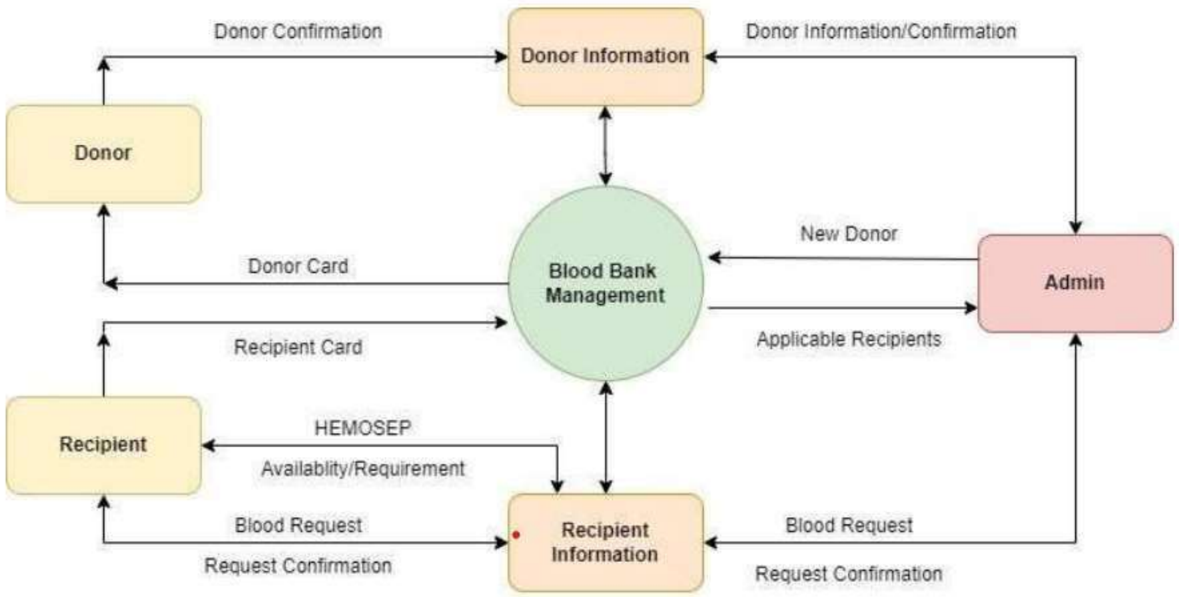


FIRST LEVEL DATA FLOW DIAGRAM OF BLOOD BANK MANAGEMENT

2.2 (1st LEVEL DFD):

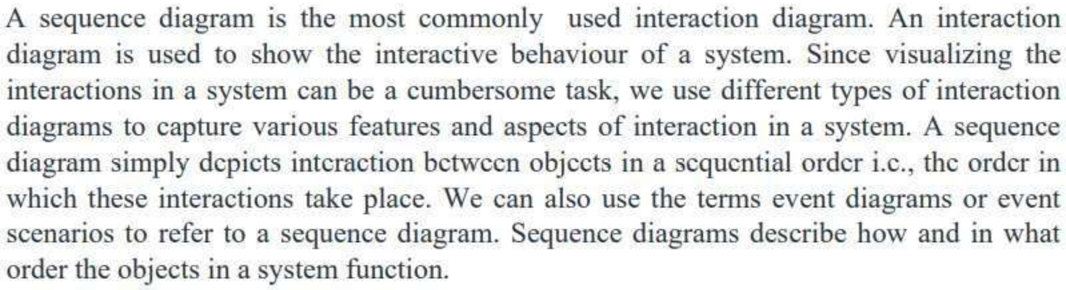
LEVEL 1:

The contents of Blood Bank Management System DFD level 1 must be single process node from the context diagram and is broken down into sub processes.

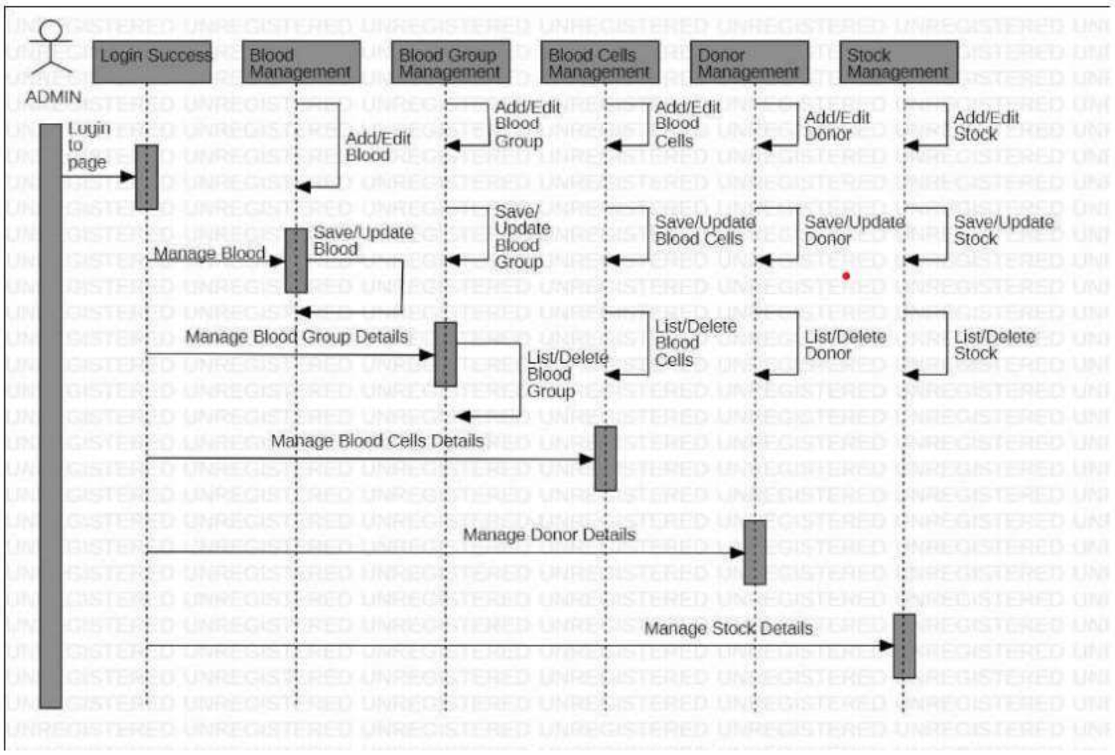


LEVEL 1 DFD

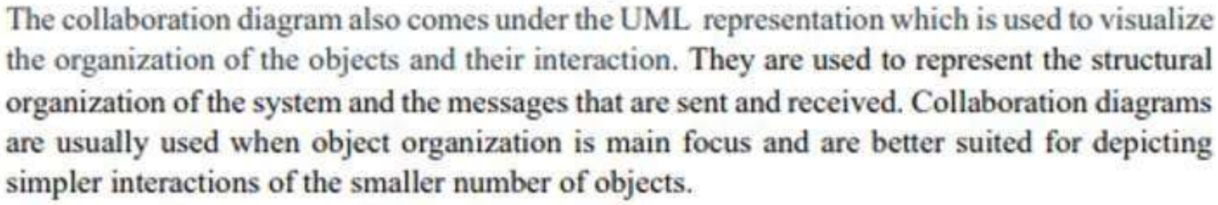
2.3 SEQUENTIAL DIAGRAM FOR BLOOD BANK MANAGEMENT SYSTEM



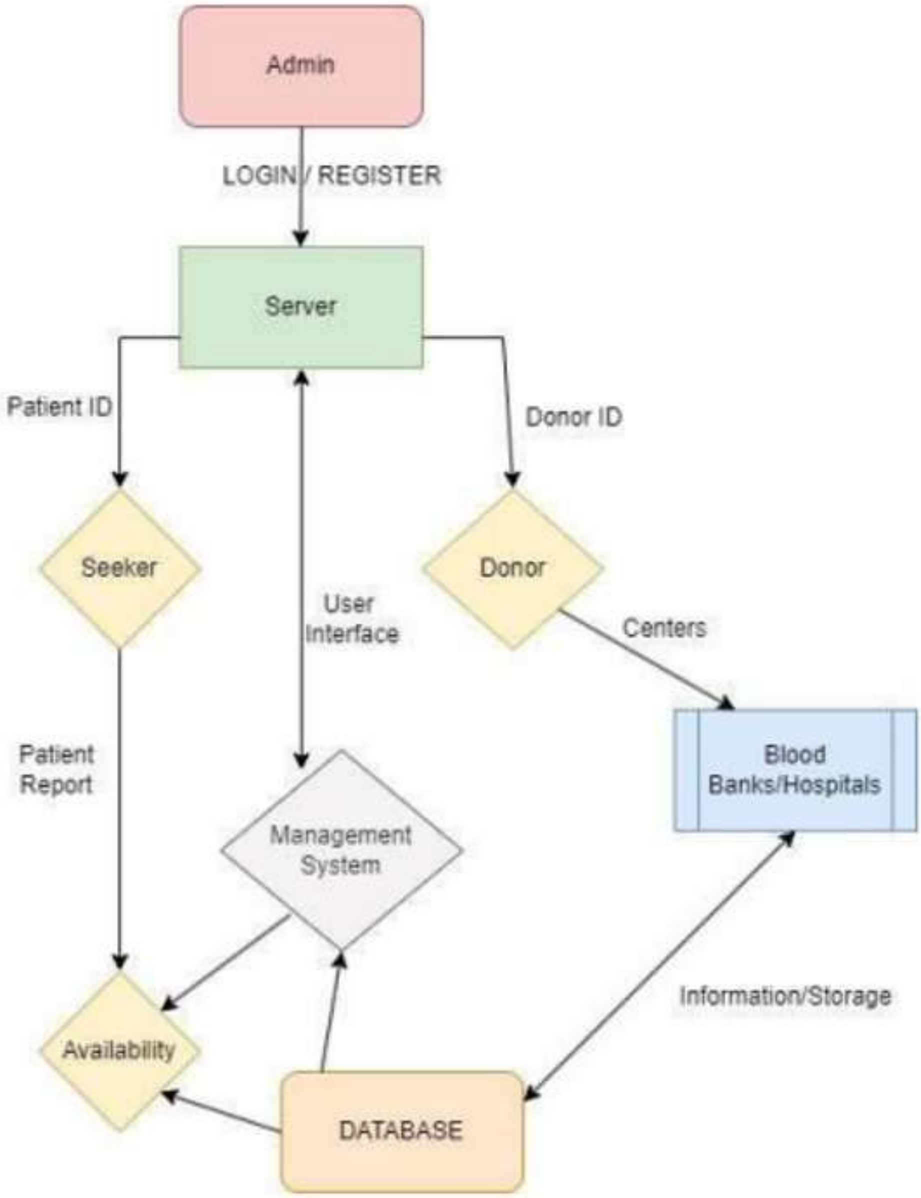
A sequence diagram is the most commonly used interaction diagram. An interaction diagram is used to show the interactive behaviour of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system. A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.



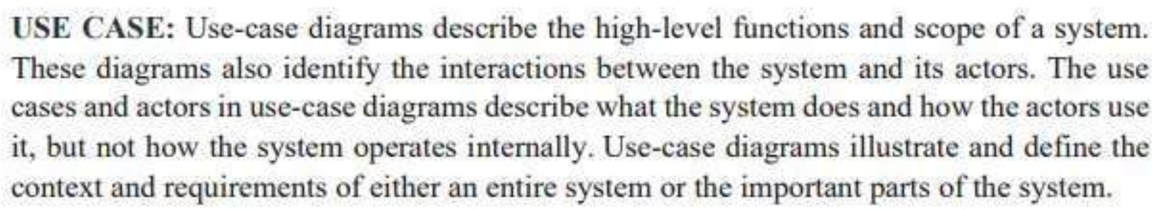
2.4 COLLABRATION DIAGRAM FOR BLOOD BLANK MANAGEMENT SYSTEM:



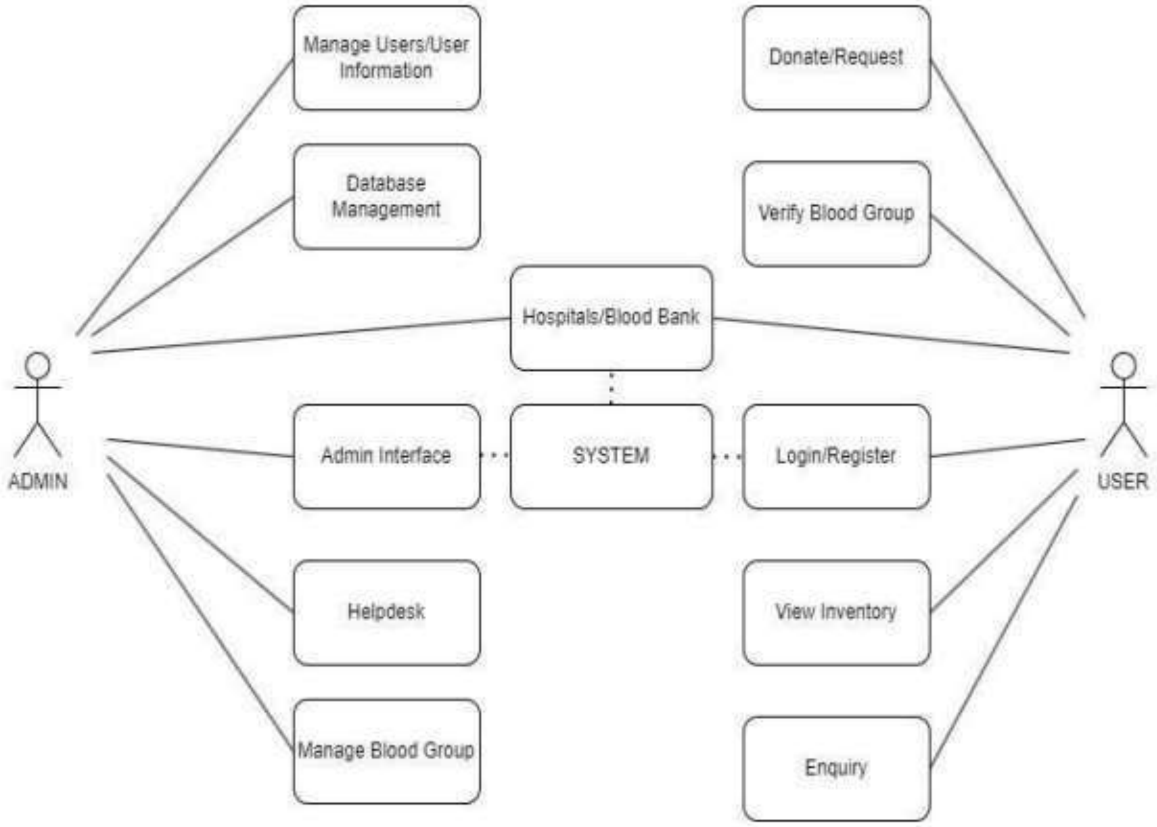
The collaboration diagram also comes under the UML representation which is used to visualize the organization of the objects and their interaction. They are used to represent the structural organization of the system and the messages that are sent and received. Collaboration diagrams are usually used when object organization is main focus and are better suited for depicting simpler interactions of the smaller number of objects.



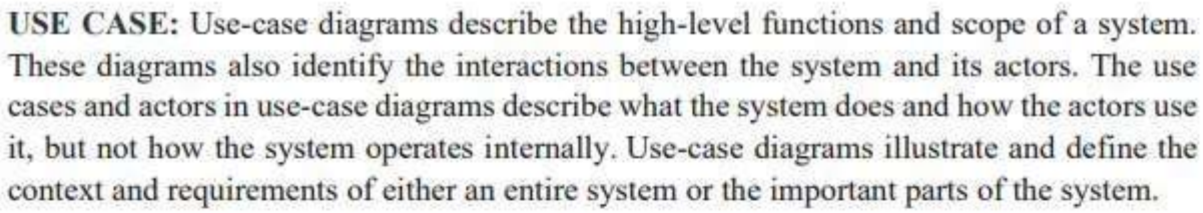
USE CASE DIAGRAM FOR BLOOD BANK MANAGEMENT



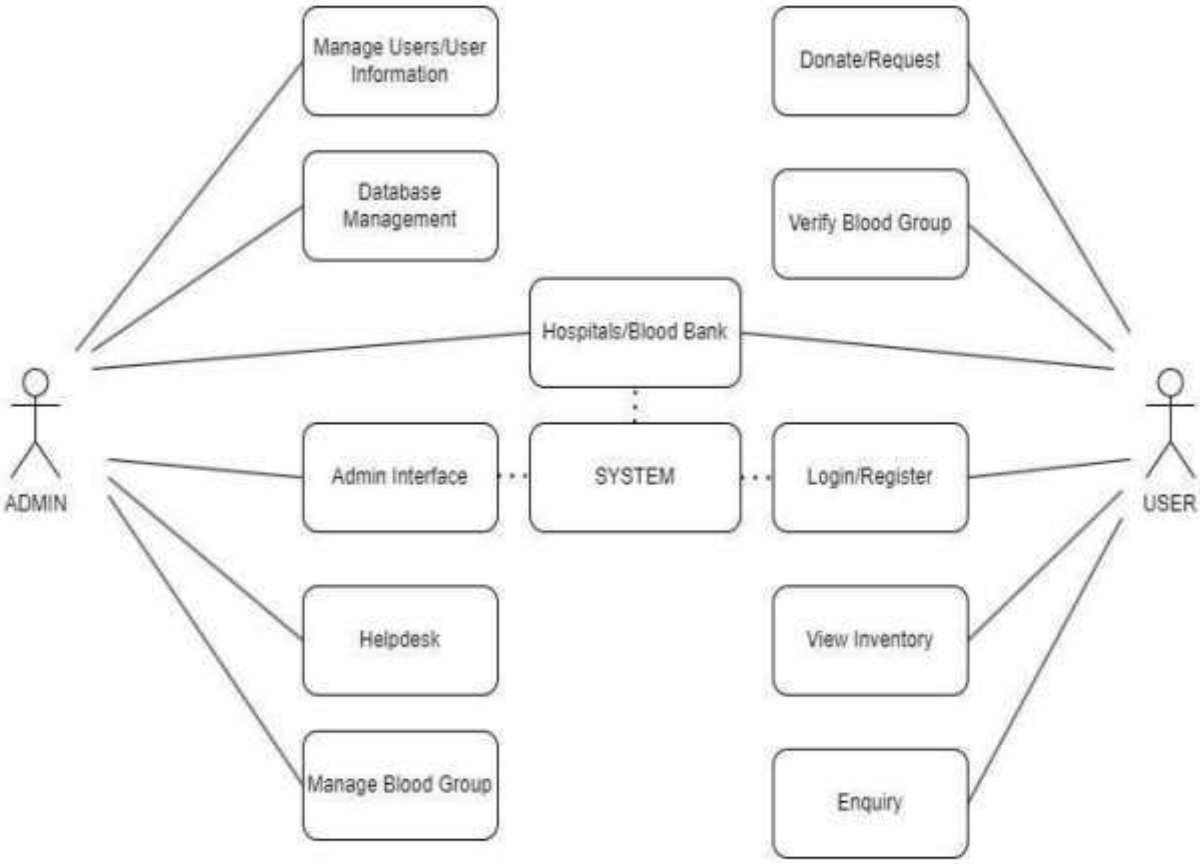
USE CASE: Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system.



STATE CHART DIAGRAM FOR BLOOD BANK MANAGEMENT



USE CASE: Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system.



2.5 SOURCE CODE:

```
#include<bits/stdc++.h> using
namespace std;
class Blood_group
{
protected:
   string blood_group; public:
   Blood_group(string n)
   {
      blood_group=n;
   }
          void set_Blood_group(string n)
          {
             blood_group=n;
}
          string get_Blood_group()
          {
             return blood_group;
}
 };
class User
{
protected:
          string phone_no;
string email;    string
aadhar_no; public:
   User(string a,string b,string c)
   {
      phone_no=a;
email=b;
aadhar_no=c;
   }
   void set_phone_no(string p)
          {
                    phone_no=p;
}
   void set_email(string e){
email=e;          }
   void set_aadhar(string h){ aadhar_no=h;
          }
   string get_phone_no(){
return phone_no;
          }
   string get_email(){
return email;   }
   string get_aadhar(){
            return aadhar_no;

}  };
class Department
{
protected:
   string department; public:
   Department(string d)
   {
      department=d;
   }
   string get_department(string d)
{
                        return department;
            }
   void set_department(string d){
     department=d;       }
};
class Name
{
protected:
string name;
public:
   Name(string n)
   {
      name=n;
   }
            void set_name(string n)
            {
               name=n;
}
            string get_name()
            {
               return name;
            }
};
class Roll
{
protected:
   long int roll; public:
   Roll(long int n)
   {
      roll=n;
   }
          void set_roll(long int  r)
          {
             roll=r;
}
          long int get_roll()
          {
             return roll;
          }

};

class Donor:public Roll,public Name,public Department,public Blood_group,public User
{
protected:
   int no_of_blood_donate;     string
last_date_of_blood_donation; public:
          Donor(string name,long int rollnum,string phone_no,string email,string aadhar_no,string
blood_group,string dept,int num_of_blood,string
lst_dt_of_bd):Roll(rollnum),User(phone_no,email,aadhar_no),Name(name),Department(dept),Blood_grou
p(blood_group)
   {
      no_of_blood_donate=num_of_blood;
      last_date_of_blood_donation=lst_dt_of_bd;
          }
      void set_no_of_blood_donate(int n)
          {
             no_of_blood_donate=n;
          }
          void set_last_date_of_blood_donation(string d)
          {
             last_date_of_blood_donation=d;
          }
          int get_no_of_blood_donate()
          {
             return no_of_blood_donate;
          }
          string get_last_date_of_blood_donation()
          {
             return last_date_of_blood_donation;
          }
          void print1()
          {
                   cout<<endl;
           cout<<"Name: "<<name<<endl;  cout << "Phone
          no.:" << phone_no <<endl;
```

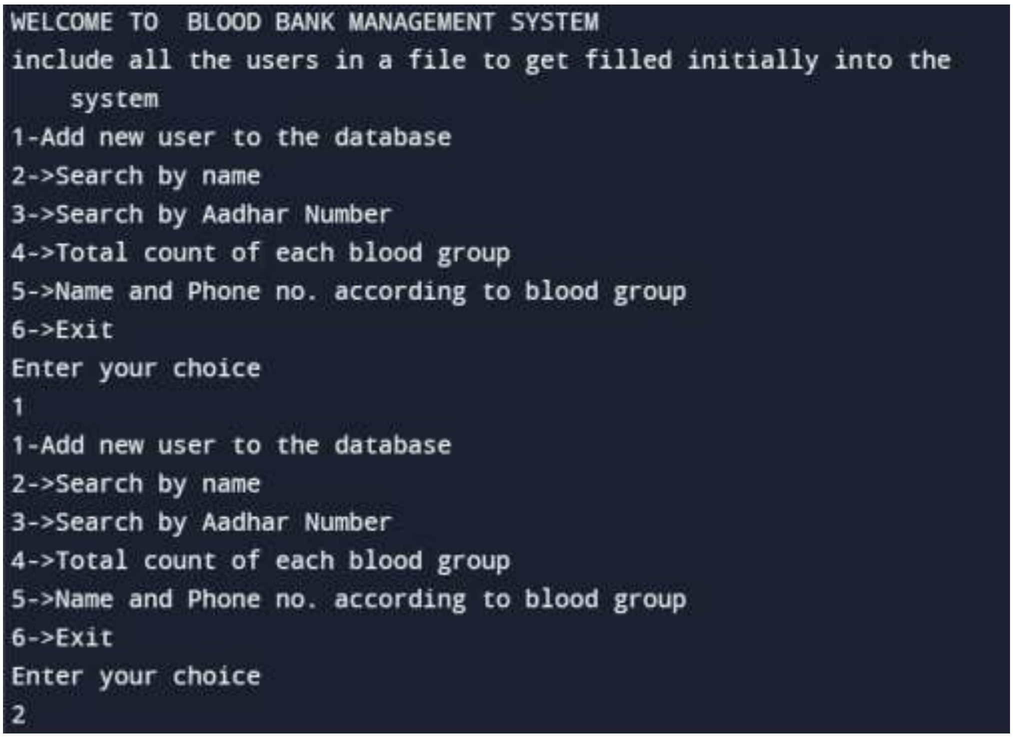
```
            }
            void print()
            {
                        cout<<endl;
                        cout<<"Name: "<<name<<endl;
cout<<"Email: "<<email<<endl;                     cout<<"Phone
No.: "<<phone_no<<endl;                 cout<<"Aadhar No.:
"<<aadhar_no<<endl;                      cout<<"Roll No.:
"<<roll<<endl;                        cout<<"Blood Group:
"<<blood_group<<endl;                   cout<<"Department:
"<<department<<endl;
                        cout<<"Amount of blood donated: "<<no_of_blood_donate<<endl;
cout<<"Last Date of Blood Donation: "<<last_date_of_blood_donation<<endl;
cout<<endl;
            }
 };
int main(){
            int data;
            cout<<"           WELCOME TO  BLOOD BANK MANAGEMENT SYSTEM    " <<endl;
cout<<"include all the users in a file to get filled initially into the system"<<endl;
ifstream infile;             infile.open("input.txt");
            int num;
            //cout<<"Enter number of donors:"<<endl;
            infile>>num;
//cin>>num;
 unordered_map<string,Donor> name_donors;  unordered_map<string,Donor>
aadhar_donors;
            unordered_map<string,vector<Donor>> blood_list;
            while(num--){
  string name,phonenum,email,addhar_no,blood_group,dept,lst_dt_of_bd;   long int rollnum;
                        int num_of_blood;
//cout<<"enter name=";                    infile>>name;
                        //cout<<"enter phonenum:";
      infile>>phonenum;
                        //cout<<"enter email:";
                        infile>>email; //cout<<"enter
            addhar_no:"; infile>>addhar_no;
                        //cout<<"enter blood group:";
                        infile>>blood_group;
                        //cout<<"Enter dept:";
                        infile>>dept;
                        //cout<<"enter rollnum:";
                        infile>>rollnum;
                        //cout<<"Enter blood in ml:";
infile>>num_of_blood;
                        //cout<<"Enter the last donation:";
```

```
                    infile>>lst_dt_of_bd;

          name_donors.insert({name,Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,n
um_of_blood,lst_dt_of_bd)});

          aadhar_donors.insert({addhar_no,Donor(name,rollnum,phonenum,email,addhar_no,blood_group,
dept,num_of_blood,lst_dt_of_bd)});
 auto itr=blood_list.find(blood_group);   if(itr ==
blood_list.end()){
                              vector<Donor> donor_dt;

            donor_dt.push_back(Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,num_of
_blood,lst_dt_of_bd));
                              blood_list.insert({blood_group,donor_dt});
                    }
                    else{
                              itr-
>second.push_back(Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,num_of_blood,ls
t_dt_of_bd));             }
          }
          infile.close();
          ofstream fout;
          fout.open("input.txt", ios::app);
unordered_map<string,Donor>::iterator p;
unordered_map<string,Donor>::iterator t;
          for(p=name_donors.begin();p!=name_donors.end();p++){              p-
>second.print();
          }
 infile.open("input.txt");  string
temp;  int ch;
          while(1)
          {
                    cout << "1-Add new user to the database\n2->Search by name \n3->Search by Aadhar
Number\n4->Total count of each blood group\n5->Name and Phone no. according to blood
group\n6>Exit\nEnter your choice\n";
                    cin >> ch;
                    unordered_map<string,vector<Donor>>::iterator itr2=blood_list.begin();
                    unordered_map<string,vector<Donor>>::iterator itr3=blood_list.end();
                    string name,phonenum,email,addhar_no,blood_group,dept,lst_dt_of_bd;
                                        long int rollnum;
                                        int num_of_blood;
                    switch(ch)
          {
          case 1:
                                        if(fout.is_open()){
                                        cout<<"enter name=";
```

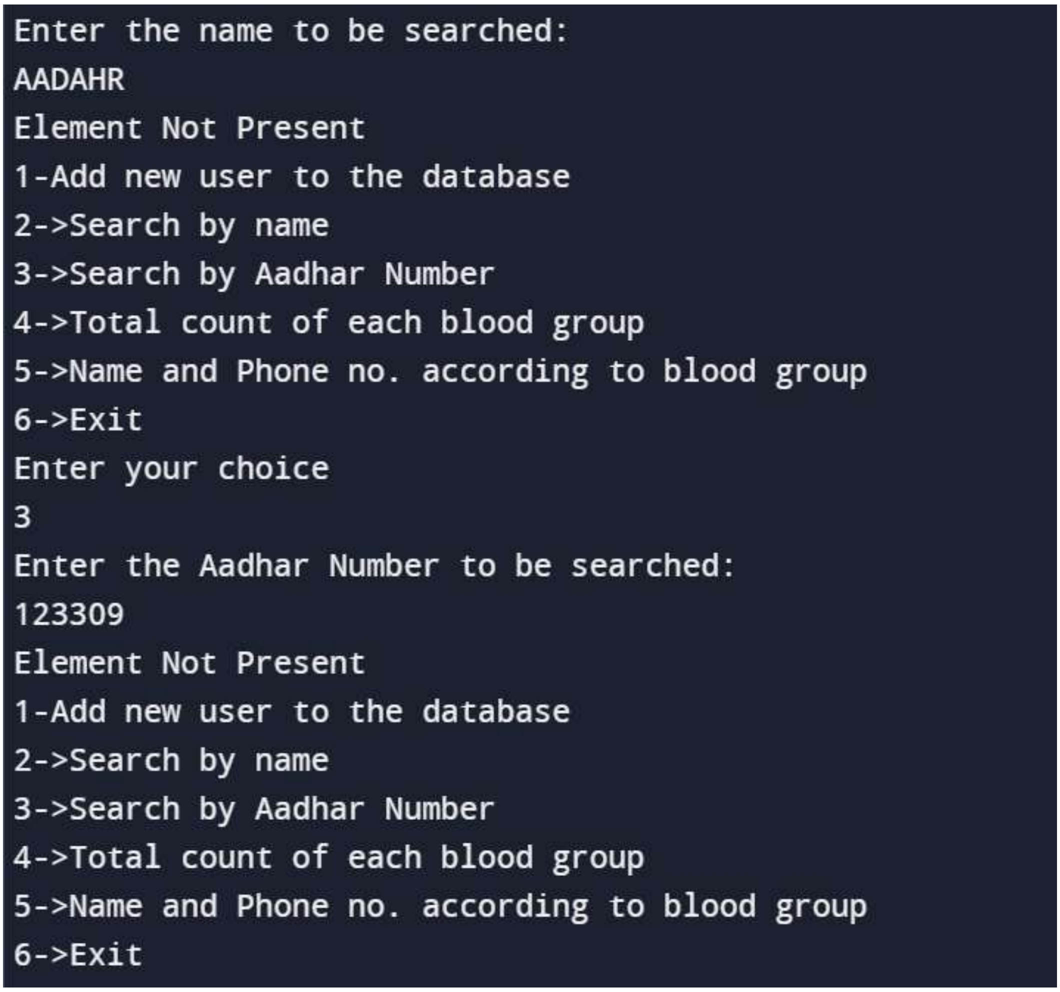
```
                        cin>>name;
                        fout<<name<<endl;
                        cout<<"enter phonenum:";
                cin>>phonenum;
                        fout<<phonenum<<endl;
                        cout<<"enter email:";
                cin>>email;
                        fout<<email<<endl;
            cout<<"enter addhar_no:";
    cin>>addhar_no;
fout<<addhar_no<<endl;
cout<<"enter blood group:";
cin>>blood_group;
fout<<blood_group<<endl;
                        cout<<"Enter dept:";
            cin>>dept;
                        fout<<dept<<endl;
                        cout<<"enter rollnum:";
                cin>>rollnum;
    fout<<rollnum<<endl;
                        cout<<"Enter blood in ml:";
                cin>>num_of_blood;
    fout<<num_of_blood<<endl;
cout<<"Enter the last donation:";
cin>>lst_dt_of_bd;
fout<<lst_dt_of_bd<<endl;
fout.close();

name_donors.insert({name,Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,num_of_blood,lst_dt_of_bd)});
    aadhar_donors.insert({addhar_no,Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,num_of_blood,lst_dt_of_bd)});
                        if(1){
                        auto itr=blood_list.find(blood_group);

                        if(itr == blood_list.end()){
                            vector<Donor> donor_dt;

    donor_dt.push_back(Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,num_of_blood,lst_dt_of_bd));
                            blood_list.insert({blood_group,donor_dt});
                        }
                    else{
                        itr->second.push_back(Donor(name,rollnum,phonenum,email,addhar_no,blood_group,dept,num_of_blood,lst_dt_of_bd));
                        }
                        }
                        }
                    break;
        case 2:
                cout<<"Enter the name to be searched:"<<endl;
                cin>>temp;

                if (name_donors.find(temp) == name_donors.end())
                cout << "Element Not Present\n";
                else
                {
                    p=name_donors.find(temp);
                    p->second.print();
            }
                break;
        case 3:
                cout<<"Enter the Aadhar Number to be searched:"<<endl;
                cin>>temp;

                if (aadhar_donors.find(temp) == aadhar_donors.end())
                cout << "Element Not Present\n";
                else
            {
                    t=aadhar_donors.find(temp);
                    t->second.print();
                }
                break;
        case 4:
                if(1)
                {
                    int sum=0;
                    while(itr2 != itr3)
                {
                        sum=0;
                        cout<<itr2->first;
                        auto ptr=itr2->second.begin();
                    auto ptr1=itr2->second.end();
                        while(ptr != ptr1)
                        {
                            sum=sum+ptr->get_no_of_blood_donate();
                            ptr++;
                        }
                        cout << "\t"<<sum ;
                        itr2++;
                        cout<<endl;
                    }
```

```
                }
                break;
            case 5:

                if(1)

                {
                    while(itr2 != itr3)
                    {

                        cout<<itr2->first;
                        auto ptr=itr2->second.begin();
                        auto ptr1=itr2->second.end();
                        while(ptr != ptr1)
                        {
                            ptr->print1();
                            ptr++;
                        }
                        cout <<"\t\t\t";
                        cout << "There are "<< itr2->second.size() << "people with " <<
itr2->first;

                        itr2++;
                        cout<<endl;
                    }

                }

              break;

            case 6:

                exit(1);

            default:

                cout<<"The option choosen is invalid.\n";
        }

   }
}
```

CHAPTER 3
OUTPUT



```
WELCOME TO  BLOOD BANK MANAGEMENT SYSTEM
include all the users in a file to get filled initially into the
    system
1-Add new user to the database
2->Search by name
3->Search by Aadhar Number
4->Total count of each blood group
5->Name and Phone no. according to blood group
6->Exit
Enter your choice
1
1-Add new user to the database
2->Search by name
3->Search by Aadhar Number
4->Total count of each blood group
5->Name and Phone no. according to blood group
6->Exit
Enter your choice
2
```



```
Enter the name to be searched:
AADAHR
Element Not Present
1-Add new user to the database
2->Search by name
3->Search by Aadhar Number
4->Total count of each blood group
5->Name and Phone no. according to blood group
6->Exit
Enter your choice
3
Enter the Aadhar Number to be searched:
123309
Element Not Present
1-Add new user to the database
2->Search by name
3->Search by Aadhar Number
4->Total count of each blood group
5->Name and Phone no. according to blood group
6->Exit
```

Chapter 4
Conclusion and future enhancement

The proposed system presents a mobile application and web application for collecting and distributing blood bags between blood banks and hospitals in order to enhance the healthcare sector taking benefits of new technologies such as the Internet of things (IoT) and machine learning. IoT affects our daily life, where it has been integrated in daily life objects and things to inform about their states and about their surrounding environment. Taking benefits of this technology in our proposed system could enhance and easy services presented to donators to interact with the system and participate as permanent donators. Using the developed system, family members can help themselves when a critical issue occurs for any member. Once a critical notification has been delivered by family members (i.e., urgent notification concerns a family member), they automatically share their blood points and discover all nearby campaigns to donate, where campaigns cars host GPS devices and update their location in the real-time, where patient was in a hospital and in need for blood bag, the point of life system can help him by informing his friends and family member, so that blood points were added to patient's account.

Taking benefits of machine learning implementations also enhance the smartness of our system, where the proposed system analyzes and classifies data (results from previous campaigns) in order to predict the most relevant and valuable places for starting new campaigns, i.e., scheduling new campaigns automatically considering current needs and shortage in blood types and quantities.

To sum up, outcome could be categorized based on main stakeholders such as follows:The Hospital subsystem:

1. Gaining the required blood bags in an organized manner by registering requests with the required blood bags for every patient assigning his criticality degree.
2. Real-time monitoring for the availability of blood bags for certain request (i.e., patient).
3. Receive sufficiency-schedule for patients' requests to help hospital administrations in organizing surgeries and operations.

The Blood Bank and campaign sub-systems:

1. Monitor the campaigns and the availability of blood types and quantities in real-time.
2. Organize distribution process of the blood bags