DHARMSINH DESAI UNIVERSITY, NADIAD

FACULTY OF TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

B. TECH. CE SEMESTER – IV

SUBJECT: SOFTWARE PROJECT

PROJECT TITLE : **BLOOD BANK MANAGEMENT SYSTEM**

BY:

1) ANIKET DOMADIYA, ROLL NO: CE028

2) SANKET GAJERA, ROLL NO: CE034

GUIDED BY: PROF. PINKAL CHAUHAN
PROF. BRIJESH BHATT
PROF. JIGAR PANDYA

CONTENTS :

1. ABSTRACT

Blood transfusion safety is a relevant and significant public health issue. Since most blood banks are still in paper-based system, various disadvantages are experienced by various stakeholders, which endanger the lives of patients and deter the healthcare system. As such, we aimed to design, develop, and implement a blood bank management system (BBMS).

Blood Bank Management System (BBMS) system project aims at providing transparency in this field, make the process of obtaining blood 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. 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.

2. INTRODUCTION

2.1 Brief Introduction

The basic building aim of this system is to provide blood donation service. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in blood bank and help them manage in a better and easy way as much as possible. Blood donors can register themselves with details of their name, blood, age etc. As well as blood needy people can retrieve details of the nearest blood bank in their area as per their blood needs and for that they have to upload verification details like medical description, reports etc. This system also provides some motivational videos about donating the blood and also the details of places and date of blood donation camps.

2.2 Tools/Technologies Used

Technologies:

- Django
- Python
- MySQL
- JavaScript
- CSS
- HTML

Tools:

- Git
- Visual Studio Code

Platform:

- Local development server

3. SOFTWARE REQUIREMENT SPECIFICATIONS

1. Manage blood request:

R.1.1 : request for donate blood

Input: choose a option ‘Be A Donor’(User selection)

Output: Blood bank tests of donor’s blood.

R.1.2 : confirm request for blood donate

Description: If the blood bank confirms, then the blood details should be stored in

database with relevant donor.

Input: Blood bank confirmation for blood donate

Output: confirmation of blood donate

Next function: Register blood

R.1.3 : Request for blood

Description: User select a option for a request for issue a blood.

Input: user selection and enter blood’s details (blood group)

Output: request has sent

R.1.4 : confirm request for issue blood

Description: If blood bank confirms based on medical reports and availability of blood,

then the blood should be issued to patient and details of blood removed from the

database.

Input: Blood bank confirmation for blood issue.

Output: Confirmation of blood issue.

Next function: Delete blood record

2. Manage blood details:

R.2.1 : Register Blood

Description: To register a blood in Blood bank, the details of blood such as blood group,

donor name, donor address, expired date and price are entered. And this is stored in

database and a unique bottle number is generated.

Input: Blood details

Output: Unique bottle number

R.2.2 : Delete Blood record

Description: Delete a record of issued blood.

Input: Unique bottle number of blood

Output: Confirmation message

R.2.3 : Update Blood record

Description: Update a record of existing blood if updation is needed.

Input: Unique bottle number of blood

Output: Confirmation message

R.2.4 : Check availability of Blood

Description: It checks for availability of blood of particular blood group.

Input: Blood group

Output: display the list of existing blood

3. Manage Donor details:

R.3.1 : Register donor

Description: A donor must be registered before he can donate a blood. After registration,

data of donor should be stored and confirmation message should be displayed.

Input: Donor details

Output: Confirmation message

R.3.1 : Update donor details

Input: Changes needed.

Output: Updated details with Confirmation message.

R.3.1 : Display donor details

Input: Donor’s name

Output: Currently stored details

4. Manage camp details:

R.4.1 : Add upcoming camp’s details

Description: Add all details about upcoming camp included venue and time.

Input: enter camp details option.

Output: details of upcoming camp.

R.4.2 : Remove camp’s details

Description: Remove all details about camp which is completed.

Input: select camp details to be deleted

Output: camp’s detail has been deleted

R.4.3 : Add photos of camp

Description : Add pictures of camp

Input : select gallery option to enter photos.

Output : pictures of blood camp and motivational video.

5. Manage statistics:

R.5.1 : Display blood bottle count

Description: The total number of blood bottles in blood bank should be displayed.

Input: User selection.

Output: Count of blood bottle

R.5.2 : Display amount invested

Description: The total amount invested in blood bank for storing blood and others is

displayed.

Input: User selection

Output: Total amount invest

R.5.3 : Generate bill

Description : The bill is generated for issued blood.

Input : Issued blood details

Output : Bill is generated.

R.5.4 : Display transaction

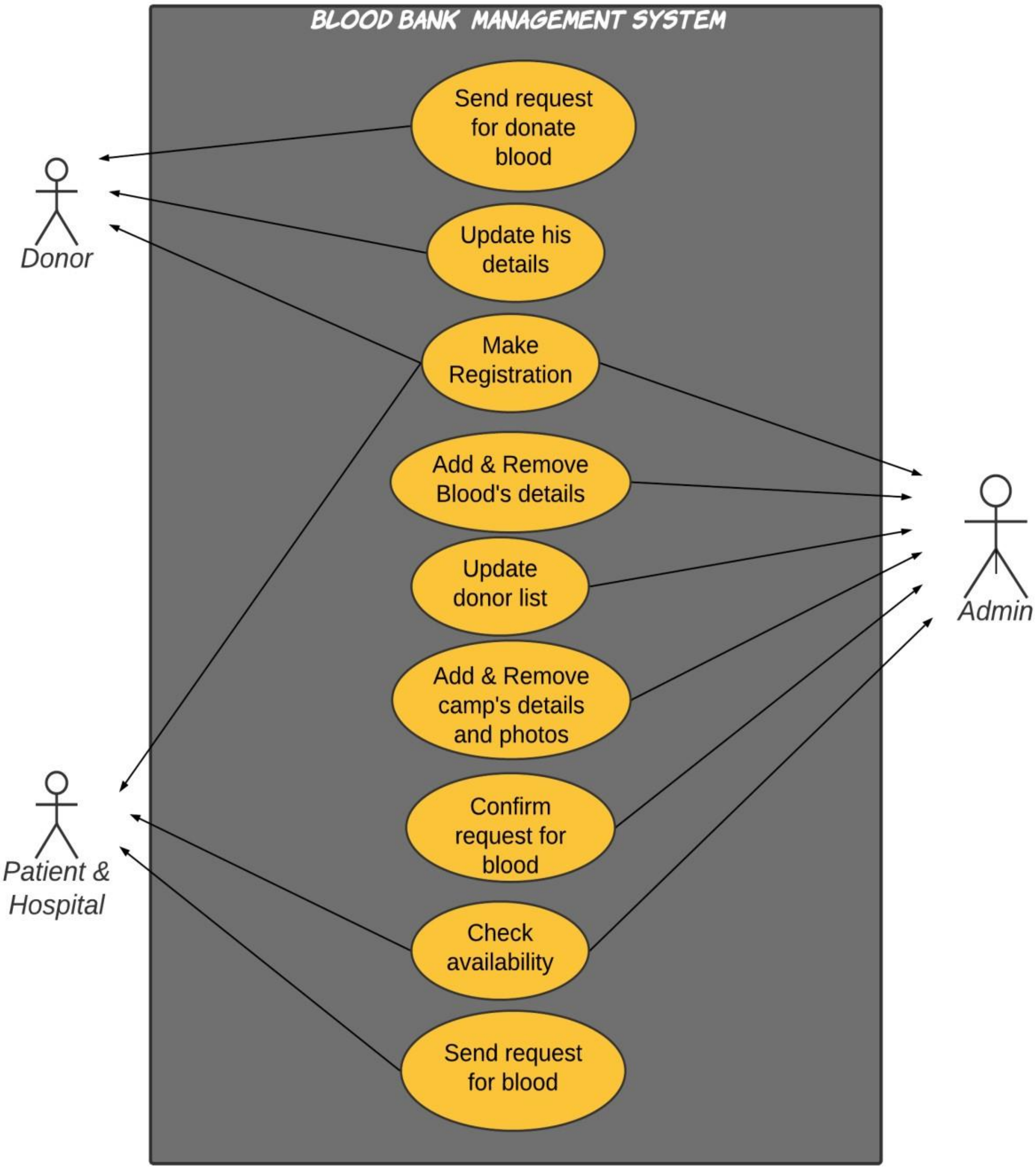
Description : Display transaction include expanses and payment done by patient.

Input : User selection

Output : Transaction list.

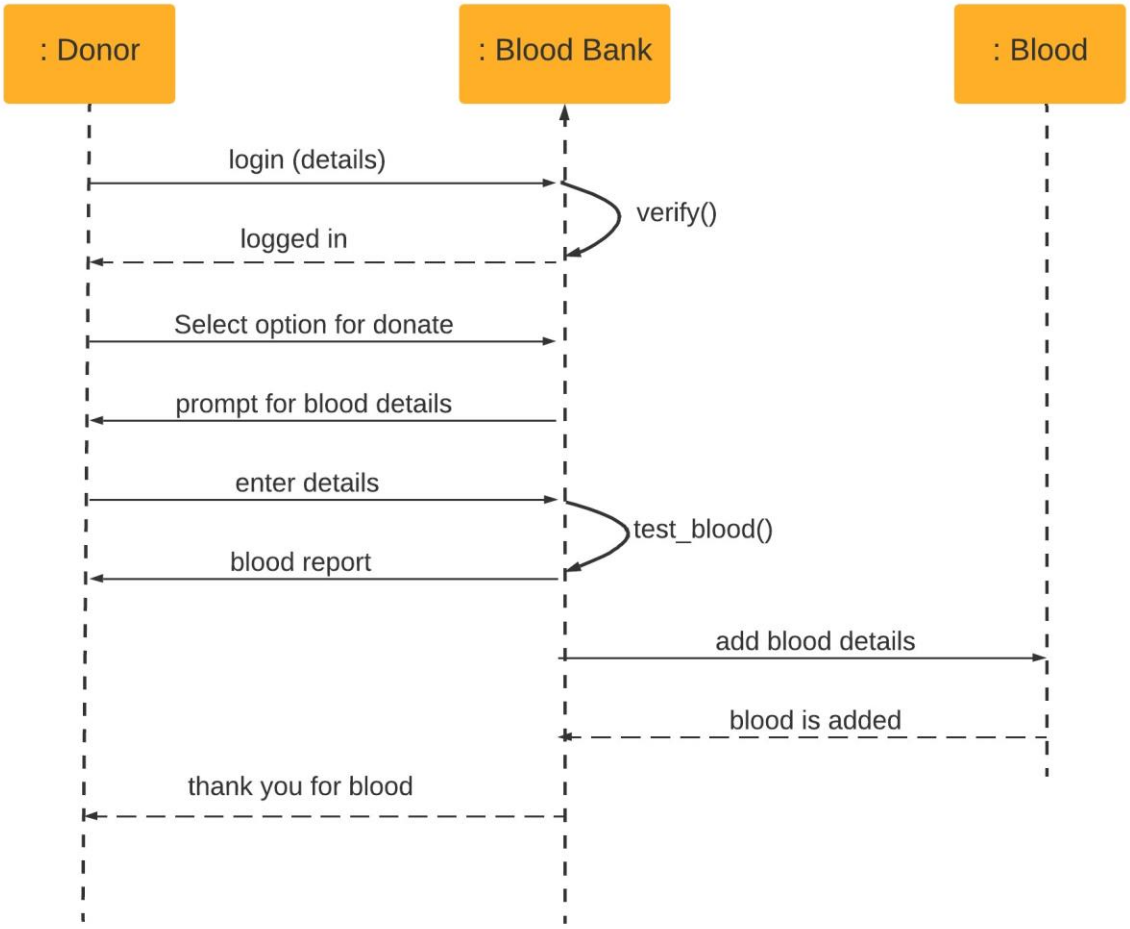
4. DESIGN

4.1 Use Case Diagram

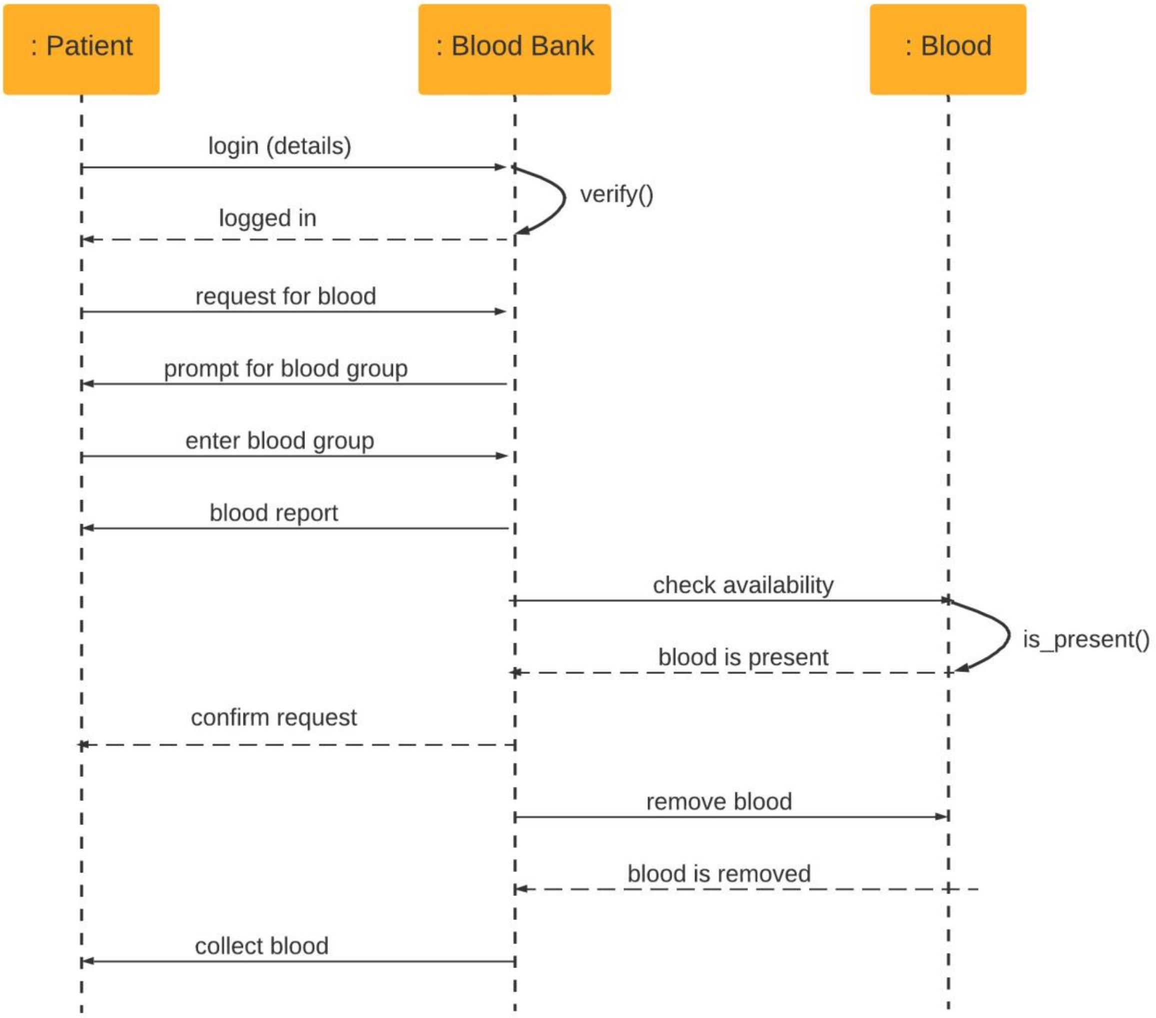

BLOOD BANK MANAGEMENT SYSTEM
Donor
Patient & Hospital
Admin
Send request for donate blood
Update his details
Make Registration
Add & Remove Blood's details
Update donor list
Add & Remove camp's details and photos
Confirm request for blood
Check availability
Send request for blood

4.2 Sequence diagram

- Use case: Make donation

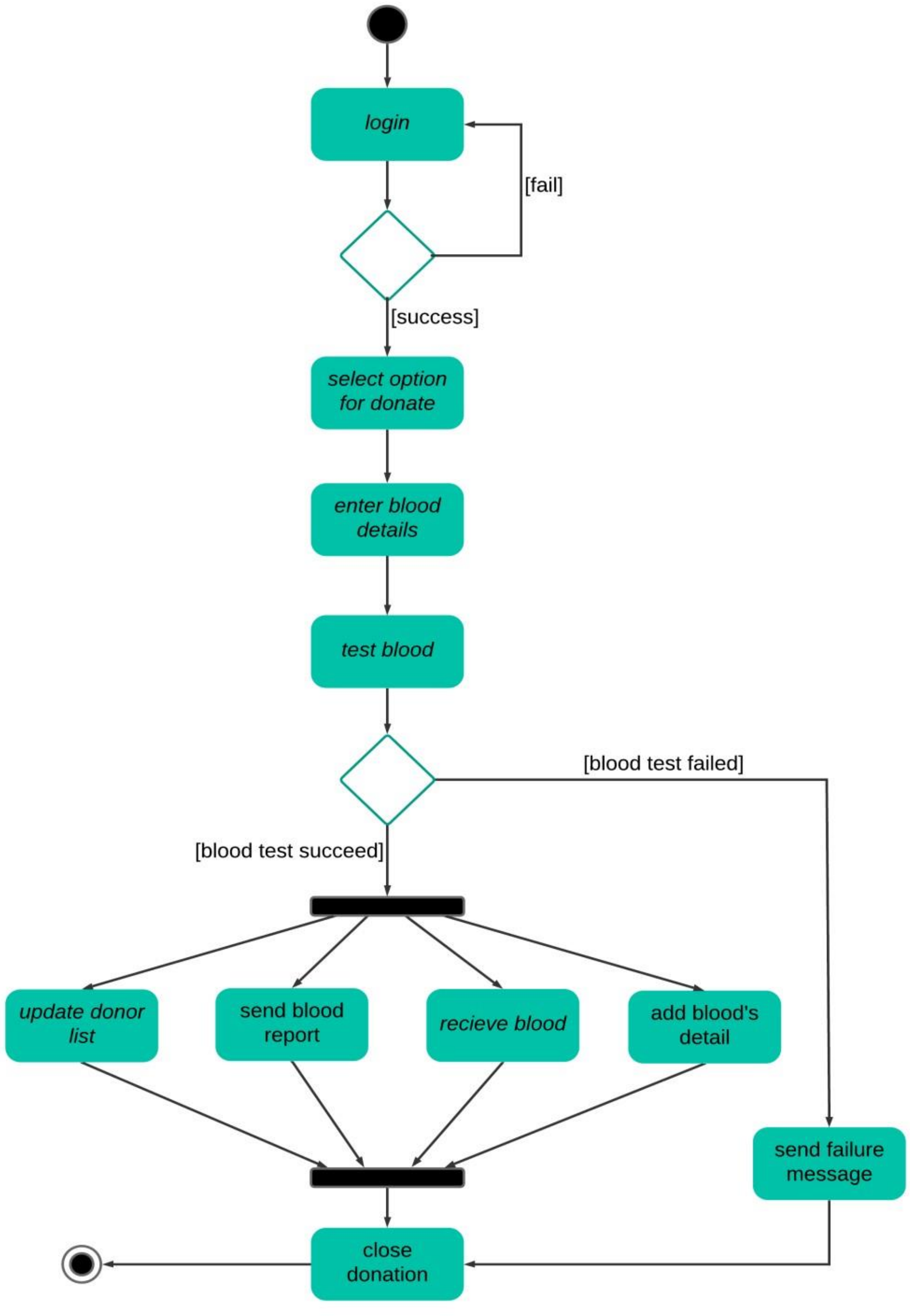


- Use case: request for blood

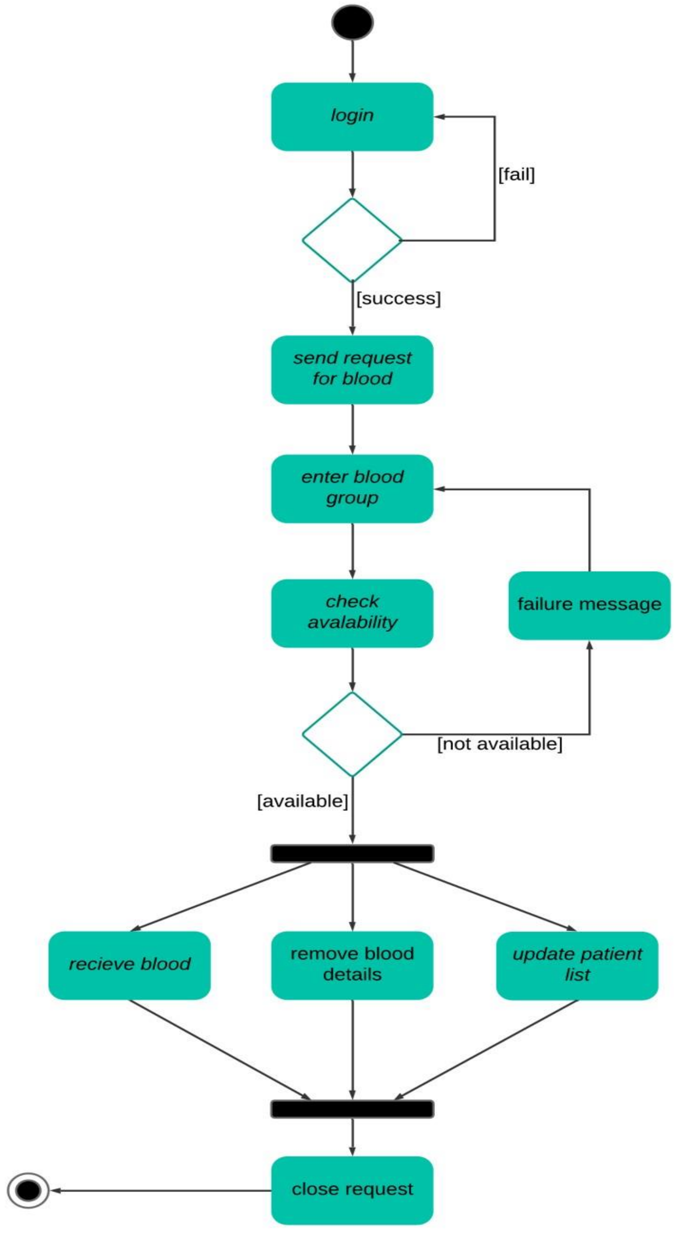


4.3 Activity diagram

- Use case : Make donation

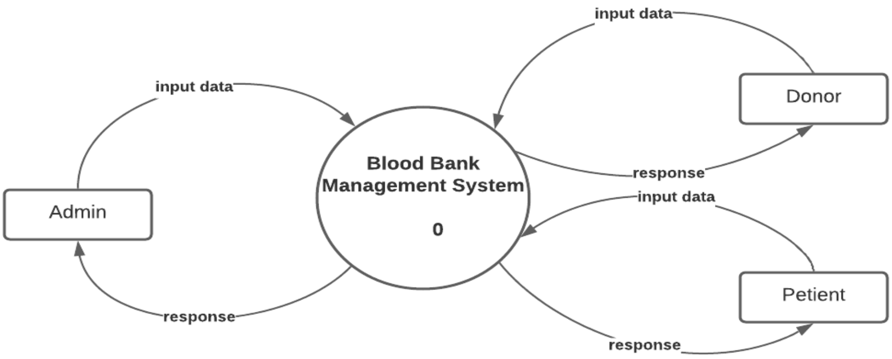


- Use case : request for blood

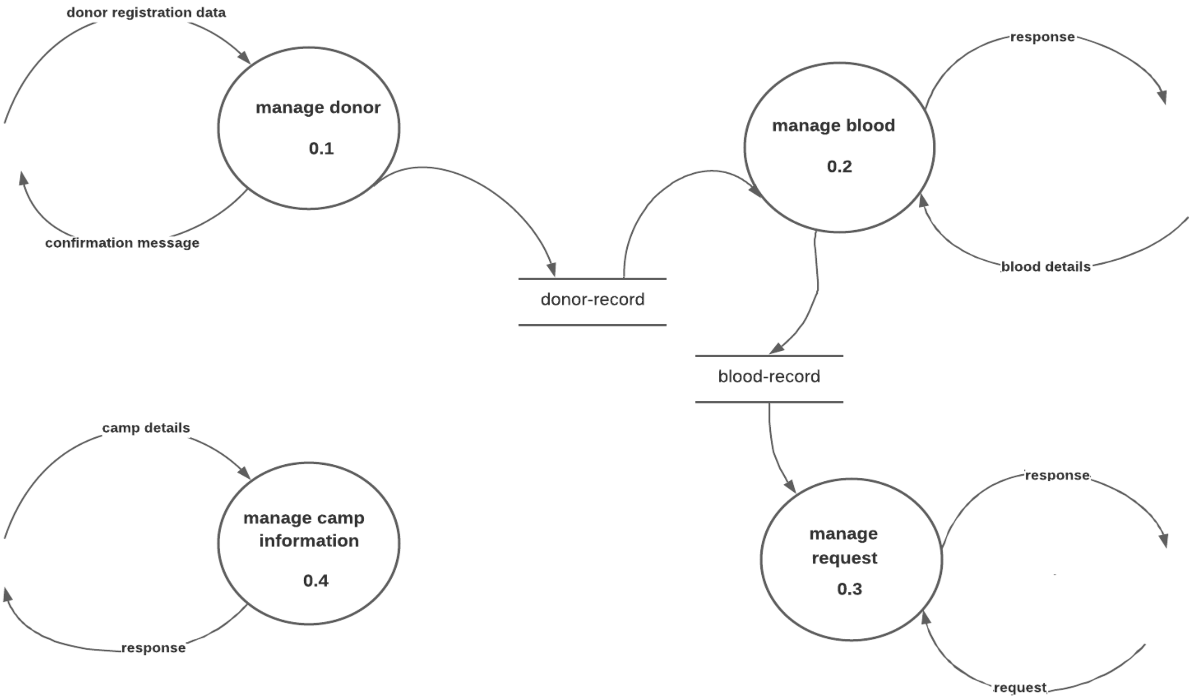


4.4 Data Flow diagram

- Context diagram:

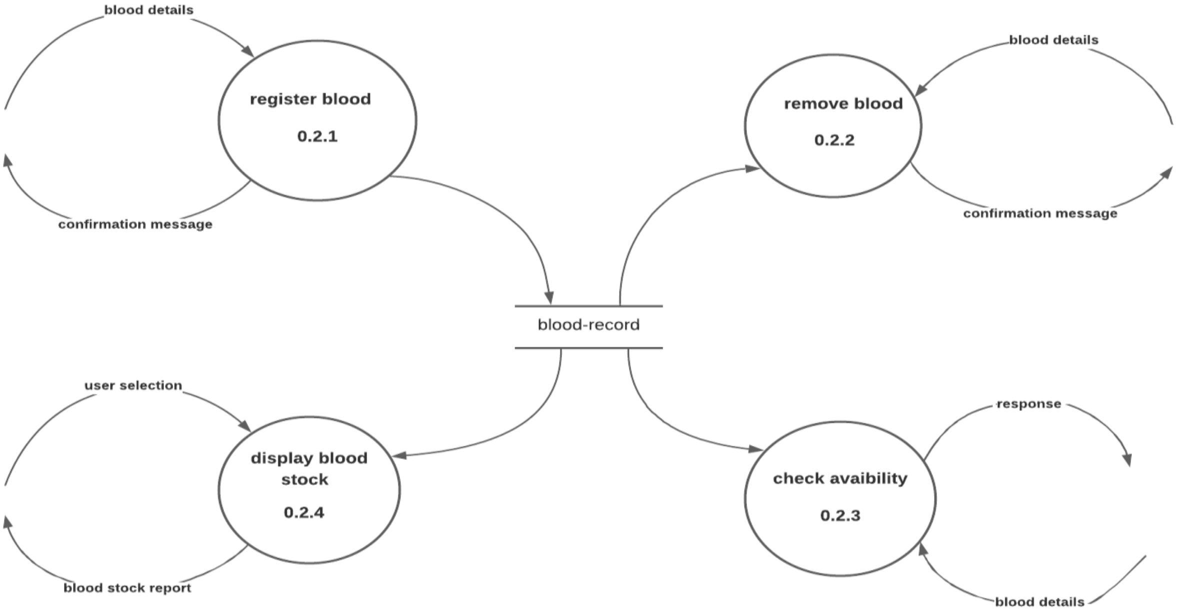


- Level 1 diagram:

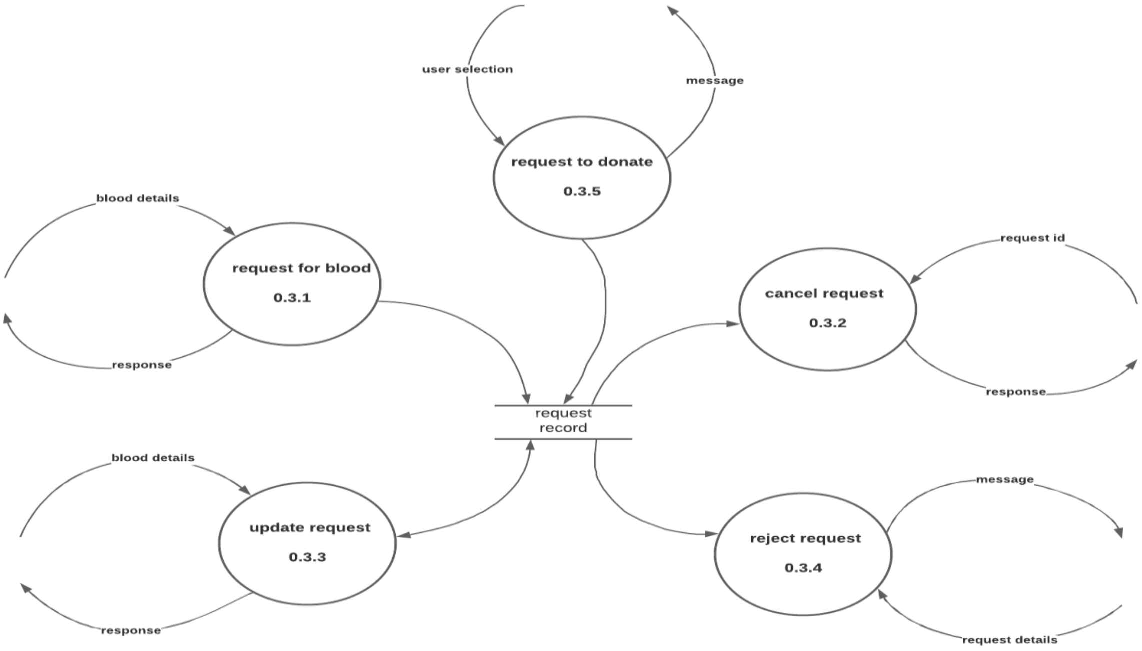


- Level 2 diagrams:

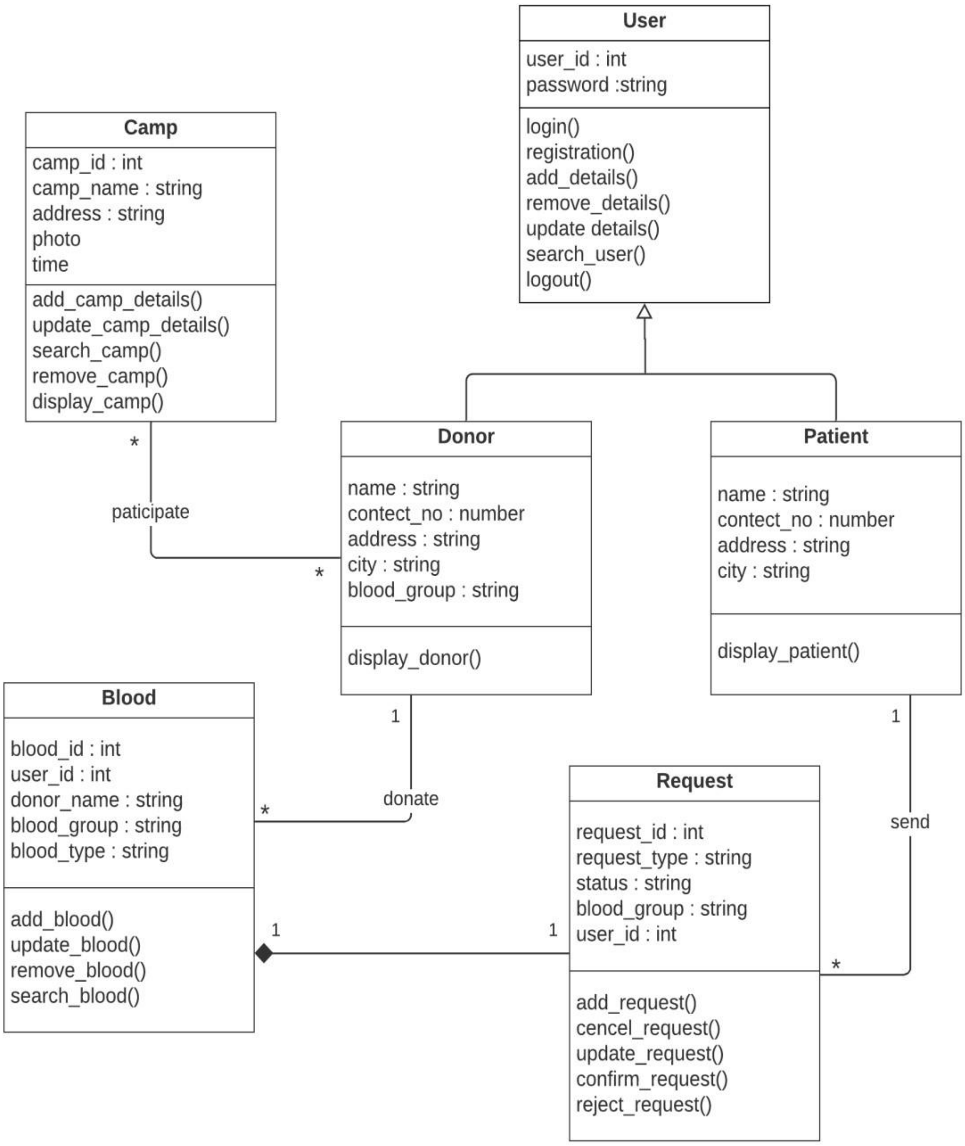
  - ➢ Manage blood:



  - ➢ Manage request:



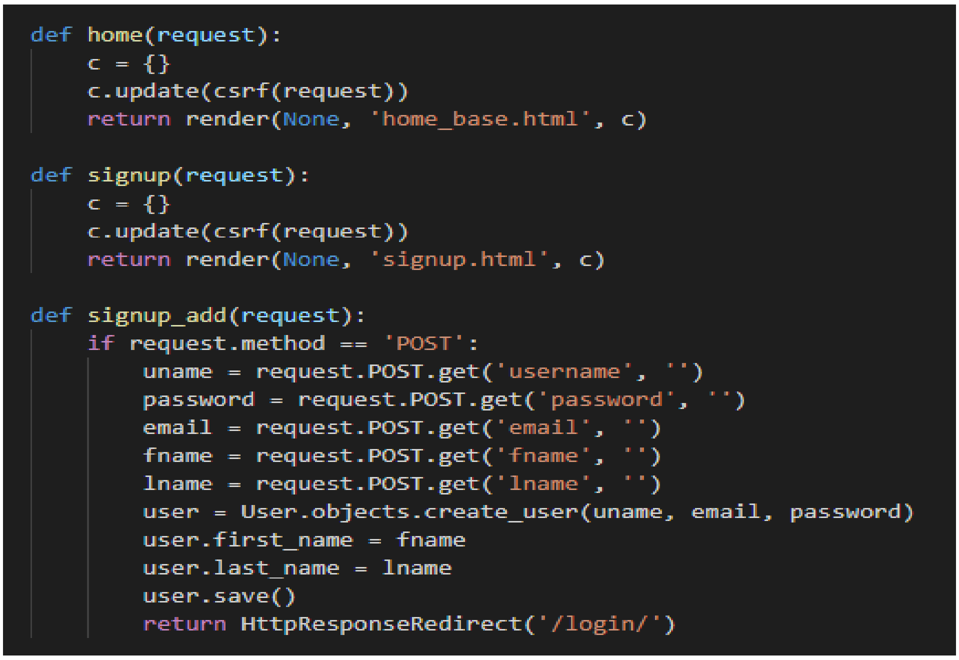
4.5 Class diagram



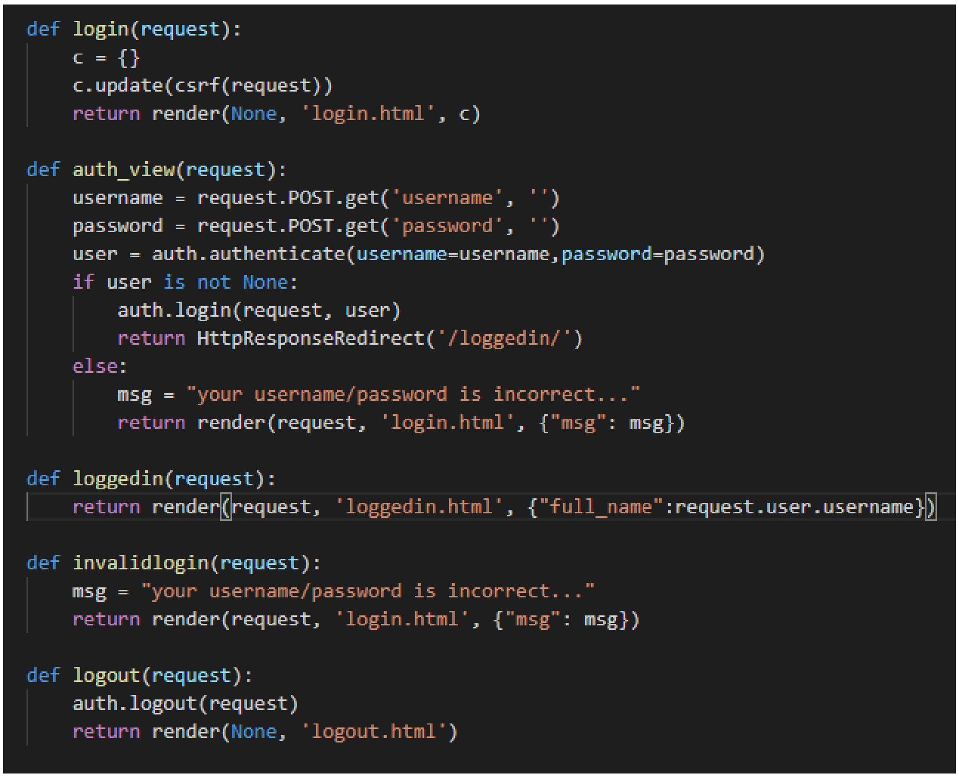
5. IMPLEMENTATION DETAILS

- **Sign up:**
  Guest users can register themselves to use this app by this function.

- **Login:**
  Users or Admin will login into their account by Login function.

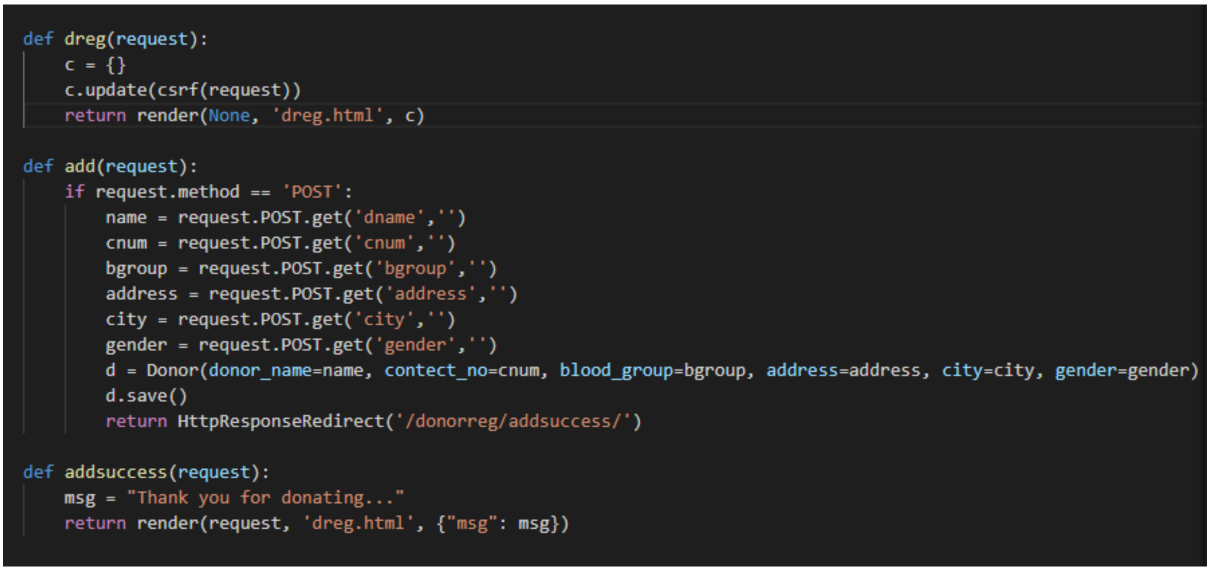


```
def home(request):
    c = {}
    c.update(csrf(request))
    return render(None, 'home_base.html', c)

def signup(request):
    c = {}
    c.update(csrf(request))
    return render(None, 'signup.html', c)

def signup_add(request):
    if request.method == 'POST':
        uname = request.POST.get('username', '')
        password = request.POST.get('password', '')
        email = request.POST.get('email', '')
        fname = request.POST.get('fname', '')
        lname = request.POST.get('lname', '')
        user = User.objects.create_user(uname, email, password)
        user.first_name = fname
        user.last_name = lname
        user.save()
        return HttpResponseRedirect('/login/')
```



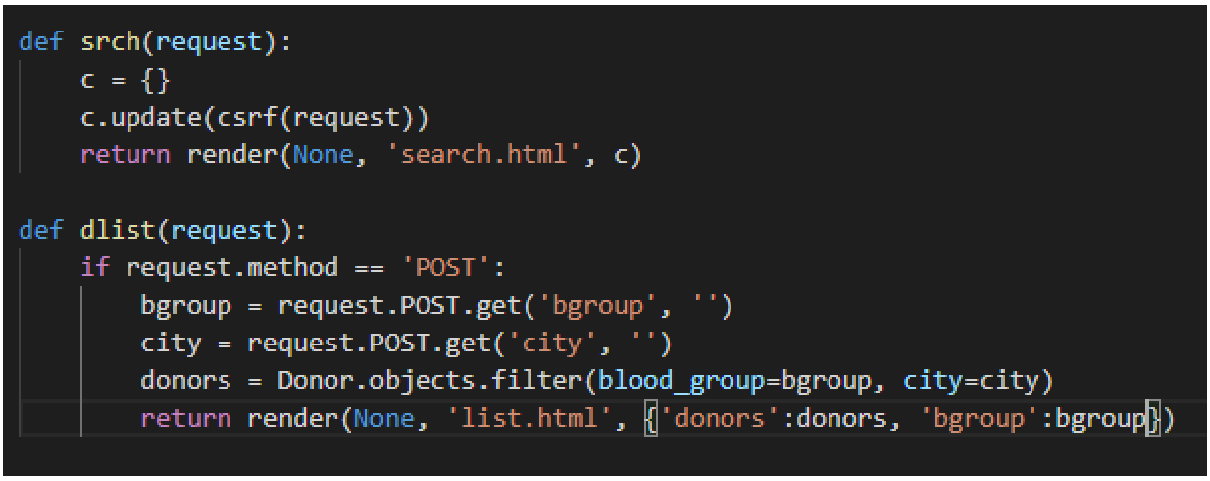
```python
def login(request):
    c = {}
    c.update(csrf(request))
    return render(None, 'login.html', c)

def auth_view(request):
    username = request.POST.get('username', '')
    password = request.POST.get('password', '')
    user = auth.authenticate(username=username,password=password)
    if user is not None:
        auth.login(request, user)
        return HttpResponseRedirect('/loggedin/')
    else:
        msg = "your username/password is incorrect..."
        return render(request, 'login.html', {"msg": msg})

def loggedin(request):
    return render(request, 'loggedin.html', {"full_name":request.user.username})

def invalidlogin(request):
    msg = "your username/password is incorrect..."
    return render(request, 'login.html', {"msg": msg})

def logout(request):
    auth.logout(request)
    return render(None, 'logout.html')
```

- **Register Donor:**
  This Function lets user(Donor) to register himself as a donor.



```python
def dreg(request):
    c = {}
    c.update(csrf(request))
    return render(None, 'dreg.html', c)

def add(request):
    if request.method == 'POST':
        name = request.POST.get('dname','')
        cnum = request.POST.get('cnum','')
        bgroup = request.POST.get('bgroup','')
        address = request.POST.get('address','')
        city = request.POST.get('city','')
        gender = request.POST.get('gender','')
        d = Donor(donor_name=name, contect_no=cnum, blood_group=bgroup, address=address, city=city, gender=gender)
        d.save()
        return HttpResponseRedirect('/donorreg/addsuccess/')

def addsuccess(request):
    msg = "Thank you for donating..."
    return render(request, 'dreg.html', {"msg": msg})
```

- **Search about donor:**
  This Function lets user to find a donors based on city.



```
def srch(request):
    c = {}
    c.update(csrf(request))
    return render(None, 'search.html', c)

def dlist(request):
    if request.method == 'POST':
        bgroup = request.POST.get('bgroup', '')
        city = request.POST.get('city', '')
        donors = Donor.objects.filter(blood_group=bgroup, city=city)
        return render(None, 'list.html', {'donors':donors, 'bgroup':bgroup})
```

6. TESTING

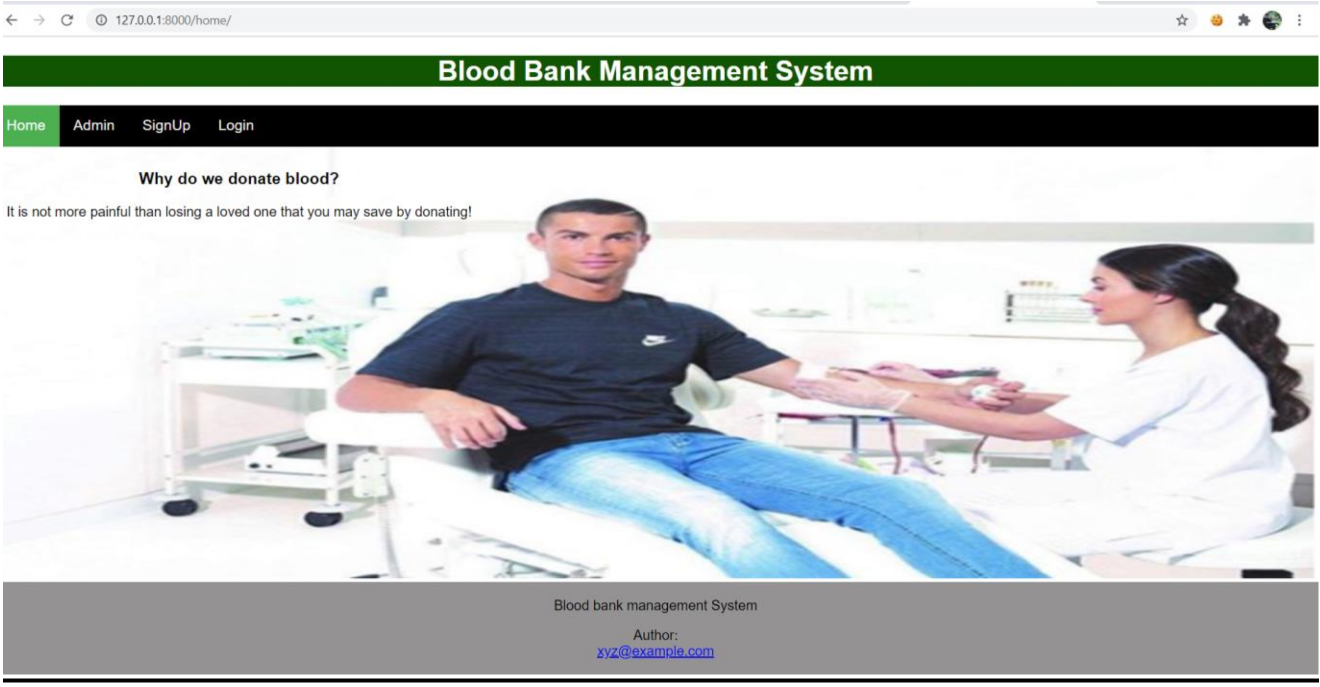
Manual testing was performed in order to find and fix the bugs in development process.

Testing Method: Manual Testing

Sr No.	Test Scenario	Expected Result	Actual Result	Status
1.	Login with incorrect credentials	User should not able to log in.	User is given a message. And redirected to login page.	Success
2.	Login with correct credentials	User should be able to log in.	User is logged in and shown the dashboard.	Success
3.	Validations on registration	User should not be allowed to enter incorrect details	User is shown a message for any incorrect detail	Success
4.	Search Donors	Patient is able to search registered donor.	When given a search query, matching donors are shown.	Success
5.	Log Out	User should be logged out and restricted from the system until next login.	User is successfully logged out and not able to access the system without signing again.	Success
6.	Add Donor	Donor details should be added in database	Donor is added and shown successful message	Success

7. WORK FLOW / LAYOUTS

Home page:



Blood Bank Management System

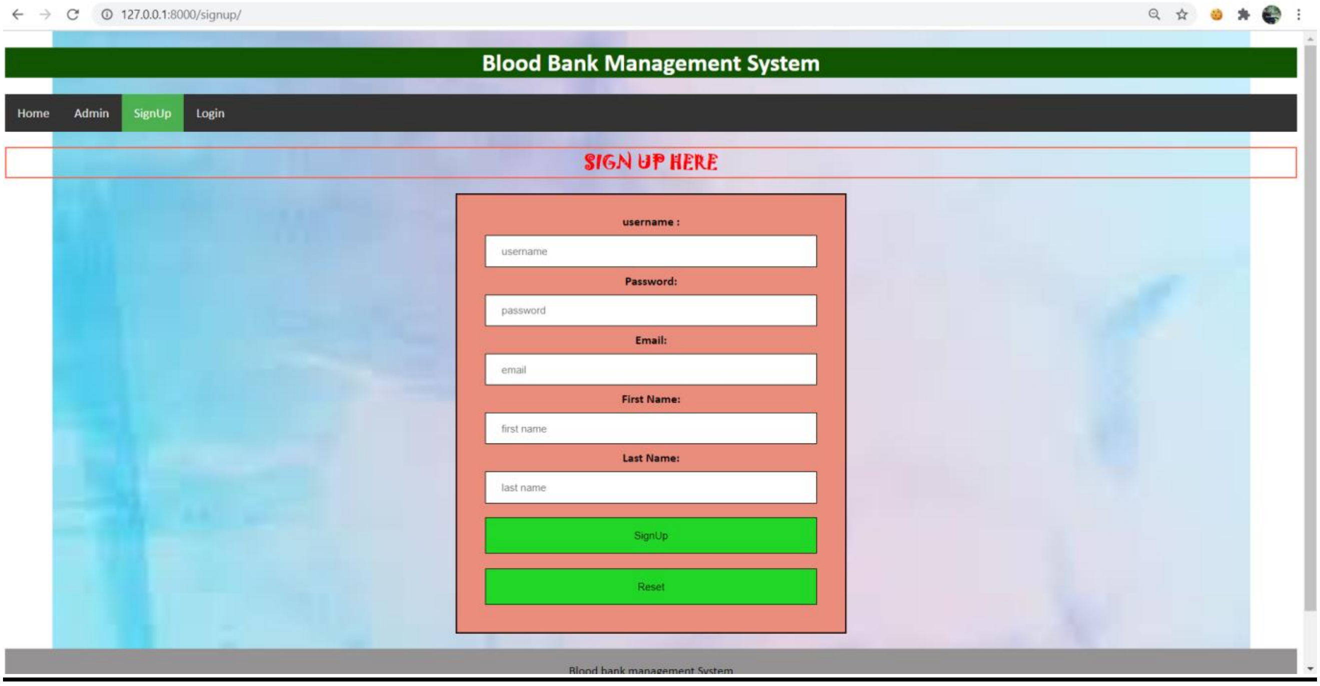
Why do we donate blood?

It is not more painful than losing a loved one that you may save by donating!

Blood bank management System

Author:
xyz@example.com

Sign Up:



Blood Bank Management System

SIGN UP HERE

username : username

Password: password

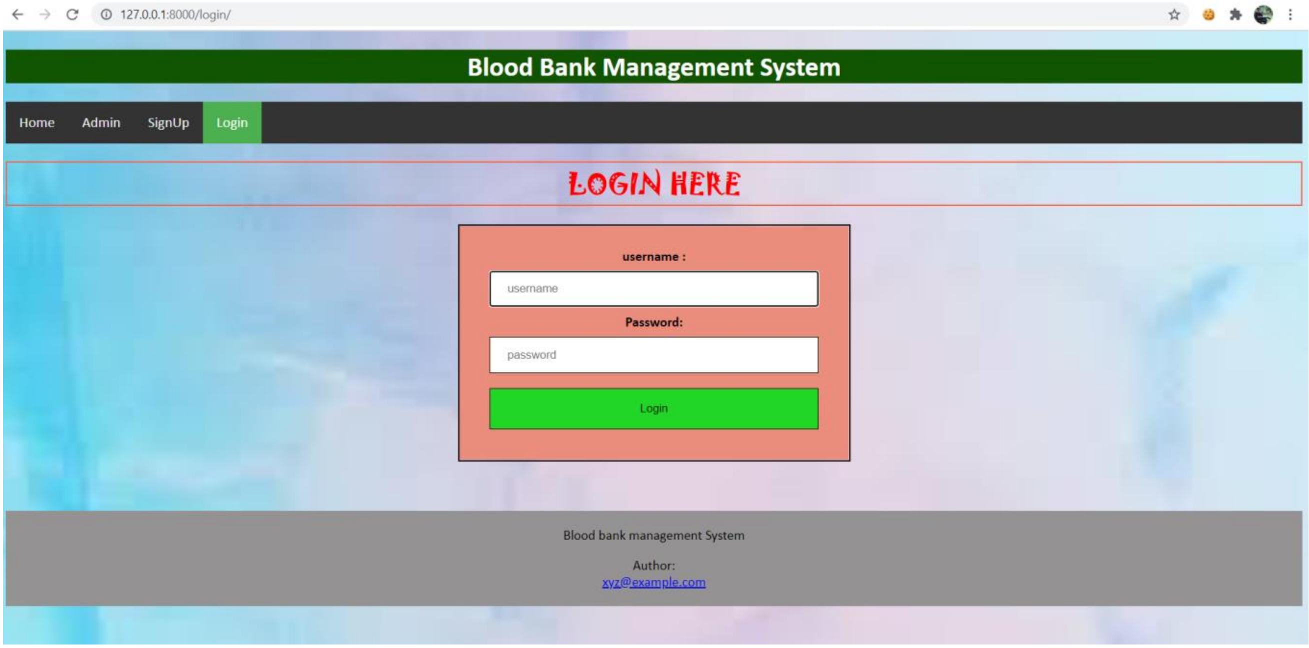
Email: email

First Name: first name

Last Name: last name

Blood bank management System

User Login:



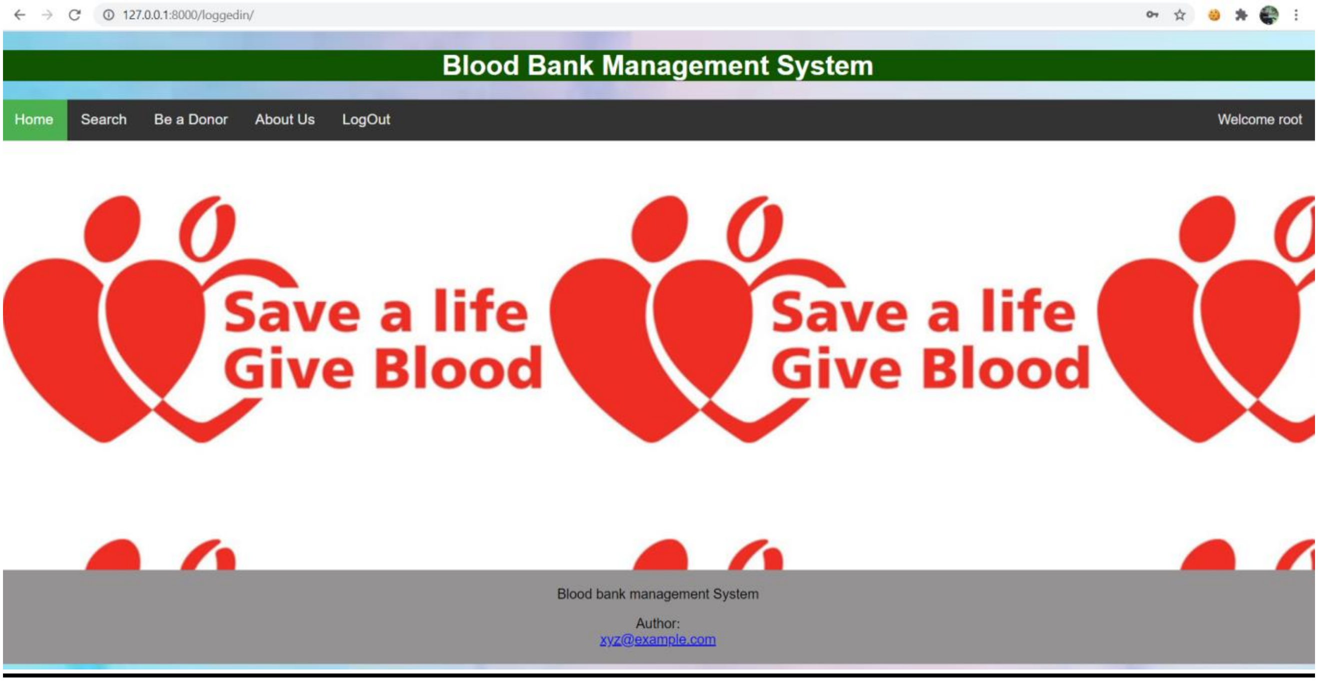
Blood Bank Management System

LOGIN HERE

Blood bank management System

Author:
xyz@example.com

Dash board page:

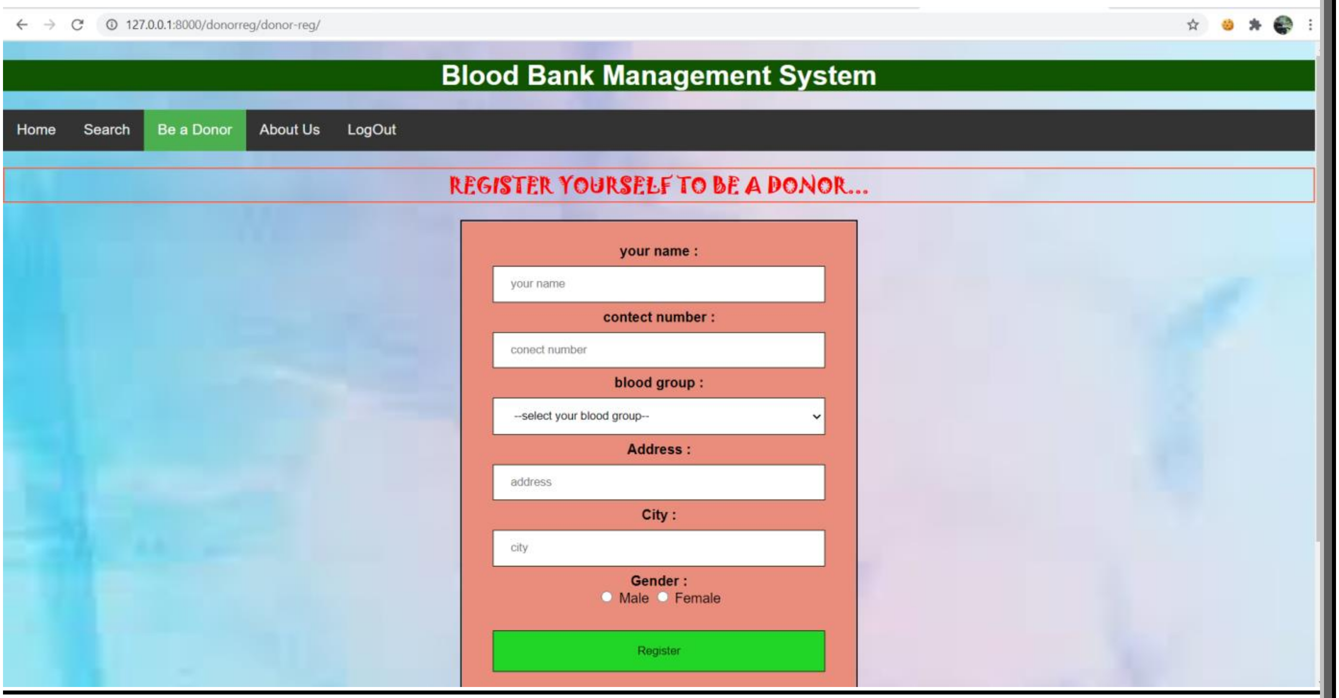


Blood Bank Management System

Blood bank management System

Author:
xyz@example.com

Donor Registration:



127.0.0.1:8000/donorreg/donor-reg/

Blood Bank Management System

REGISTER YOURSELF TO BE A DONOR...

your name :

your name

contect number :

conect number

blood group :

--select your blood group--

Address :

address

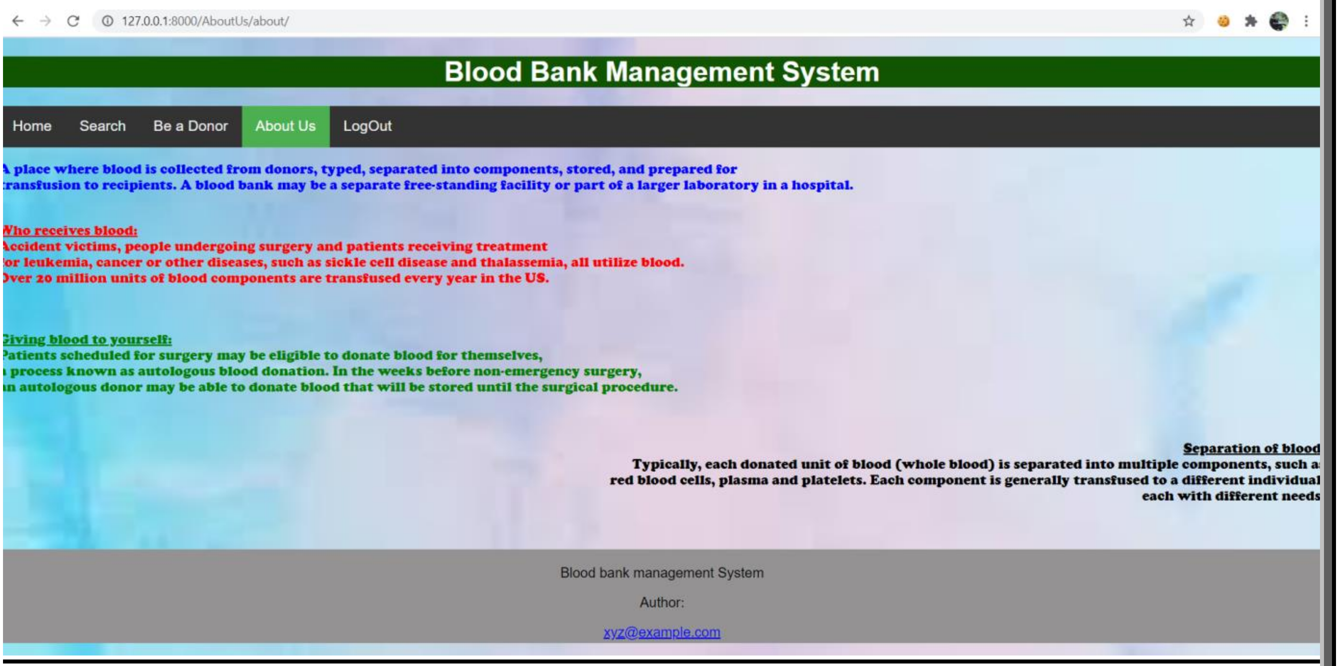
City :

city

Gender :

○ Male ○ Female

About Us:



127.0.0.1:8000/AboutUs/about/

Blood Bank Management System

A place where blood is collected from donors, typed, separated into components, stored, and prepared for
transfusion to recipients. A blood bank may be a separate free-standing facility or part of a larger laboratory in a hospital.

Who receives blood:
Accident victims, people undergoing surgery and patients receiving treatment
for leukemia, cancer or other diseases, such as sickle cell disease and thalassemia, all utilize blood.
Over 20 million units of blood components are transfused every year in the US.

Giving blood to yourself:
Patients scheduled for surgery may be eligible to donate blood for themselves,
a process known as autologous blood donation. In the weeks before non-emergency surgery,
an autologous donor may be able to donate blood that will be stored until the surgical procedure.

Separation of blood
Typically, each donated unit of blood (whole blood) is separated into multiple components, such a
red blood cells, plasma and platelets. Each component is generally transfused to a different individual
each with different needs

Blood bank management System

Author:

xyz@example.com

8. CONCLUSION

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

- User registration
- Login
- Admin module
- Donor registration containing all necessary information
- Search donor details
- About us page
- Logout

9. LIMITATIONS AND FUTURE ENHANCEMENTS

- We have implemented Donor related modules but patient registration module can be further implemented.

- Module which allows to make request for blood which is related to patient can be further implemented.

- Gallery can be added which includes Camp details related module can be implemented which describes arranged camp details in future and motivational video clips for donate the blood.

10. REFERENCE / BIBLIOGRAPHY

Following links and websites were referred during the development of this project:

- Django official documentation
- stackoverflow.com
- github.com