

## Table of Contents

CHAPTER: 1 INTRODUCTION .....	5
1.1 Introduction .....	5
1.2 Problem Statement .....	5
1.3 Objectives.....	5
1.4 Scope and Limitation .....	6
1.4.1 Scope .....	6
1.4.2 Limitation .....	6
1.5 Development Methodology .....	7
1.6 Report Organization .....	8
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW .....	9
2.1 Background study .....	9
2.2 Literature Review.....	9
CHAPTER 3: SYSTEM ANALYSIS .....	12
3.1 System Analysis .....	12
3.1.1 Requirement Analysis .....	12
3.1.2 Feasibility Analysis .....	13
3.1.3 Analysis .....	16
CHAPTER 4: SYSTEM DESIGN.....	21
4.1 Design .....	21
4.1.1 Refinement of object, class and sequence diagram .....	21
4.1.2 Component Diagram .....	23
4.1.3 Deployment Diagram .....	24
4.2 Algorithm Details .....	25
CHAPTER 5: IMPLEMENTATION AND TESTING.....	27
5.1 Implementation.....	27
5.1.1 Tools Used.....	27
5.1.1.1 Programming Language Tools.....	27
5.1.1.2 Database Tools.....	27
5.1.1.3 Drawing Tools .....	27
5.1.2 Implementation Details of Modules .....	27
5.2 Testing.....	34
5.2.1 Test Cases for Unit Testing .....	34
5.2.2 Test Cases for System Testing .....	36
5.3 Result Analysis .....	37
CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION.....	38
6.1 Conclusion.....	38
6.2 Future Recommendation .....	38

REFERENCES .....	39
APPENDIX .....	40

## LIST OF FIGURES

Figure 1.5.1 Waterfall Model.....	7
Figure 3.1.1.1 Use-case Diagram .....	12
Figure 3.1.2.1 Gantt Chart .....	16
Figure 3.1.3.1 Object Diagram.....	16
Figure 3.1.3.2 Class Diagram.....	18
Figure 3.1.3.3 Sequence Diagram .....	20
Figure 3.1.3.3.1 Activity Diagram .....	19
Figure 4.1.1.1 Refinement of Object Diagram.....	21
Figure 4.1.1.2 Refinement of Class Diagram.....	22
Figure 4.1.1.3 Refinement of sequence Diagram.....	23
Figure 4.1.2.1 Component Diagram .....	24
Figure 4.1.2.1 Deployment Diagram.....	24

## **LIST OF TABLES**

Table 3.1.2.1 Hardware Requirement.....	13
Table 3.1.2.2 Software Requirement.....	14
Table 3.1.2.3 Gantt chart Table.....	14
Table 5.2.1.1 Test Cases for Unit Testing .....	35-36
Table 5.2.2.1 Test Cases for system Testing.....	36

# **CHAPTER: 1 INTRODUCTION**

## **1.1 Introduction**

Blood donation is a sensitive process and, therefore, it requires careful management and control. To achieve this, proper planning is essential. A proposed solution to improve the current system is to create a web-based system for blood banks. This system would enable individuals in need of blood to search for available donors and make requests for blood as required. This would help to streamline the process, making it more efficient and effective in meeting the needs of those requiring blood transfusions.

“Rakt Seva” is a blood bank to gather, manage blood from various sources such as blood donation and distribute it to needy people who have high requirement for it.

It is a web-based platform that helps blood donors to locate nearest blood bank to them. The system is designed to notify registered when there is shortage of blood in the blood bank. A SMS will be sent to donors who are eligible for donating.

## **1.2 Problem Statement**

The current system for blood donation and collection is encountering a challenge of both excess and shortage of blood. This is due to the absence of a dedicated location and schedule for individuals who want to donate blood. Moreover, once blood is collected, it has a shelf life of only 42 days. If a considerable amount of blood is collected and not used within this time frame, it results in wastage.

To address this problem, the developed system provides a solution by notifying registered donors through SMS if there is a limited blood supply in the blood banks. This allows interested donors to donate blood based on their proximity to the blood bank in emergency cases. As a result, this system helps to minimize the wastage of blood and ensure that it is available when needed.

## **1.3 Objectives**

The objective of Rakt Seva is to facilitate blood donation by providing a user-friendly web-based platform that connects interested blood donors with nearby blood banks, with the ultimate goal of improving public health outcomes through a consistent supply of high-quality blood products.

## **1.4 Scope and Limitation**

### **1.4.1 Scope**

The scope of the Rakt Seva web-based platform is to provide a convenient and efficient way for blood donors to locate nearby blood banks and to receive alerts when there is a shortage of blood in the blood bank. The platform can help increase awareness about the importance of blood donation and encourage more people to donate blood.

The system can also help streamline the process of blood donation by providing a centralized platform for blood banks to communicate with donors and coordinate blood donations. This can help ensure that blood is available when needed and reduce the risk of shortages or waste.

Additionally, the SMS alerts can help address the issue of blood shortages, which is a common problem. By notifying eligible donors about the need for blood, the system can help increase the number of donors who come forward to donate.

Overall, the scope of the Rakt Seva system is to improve access to blood donations, increase awareness about the importance of blood donation, and help ensure that blood is available when needed. The system has the potential to make a significant impact on public health and improve the quality of life for those who require blood products.

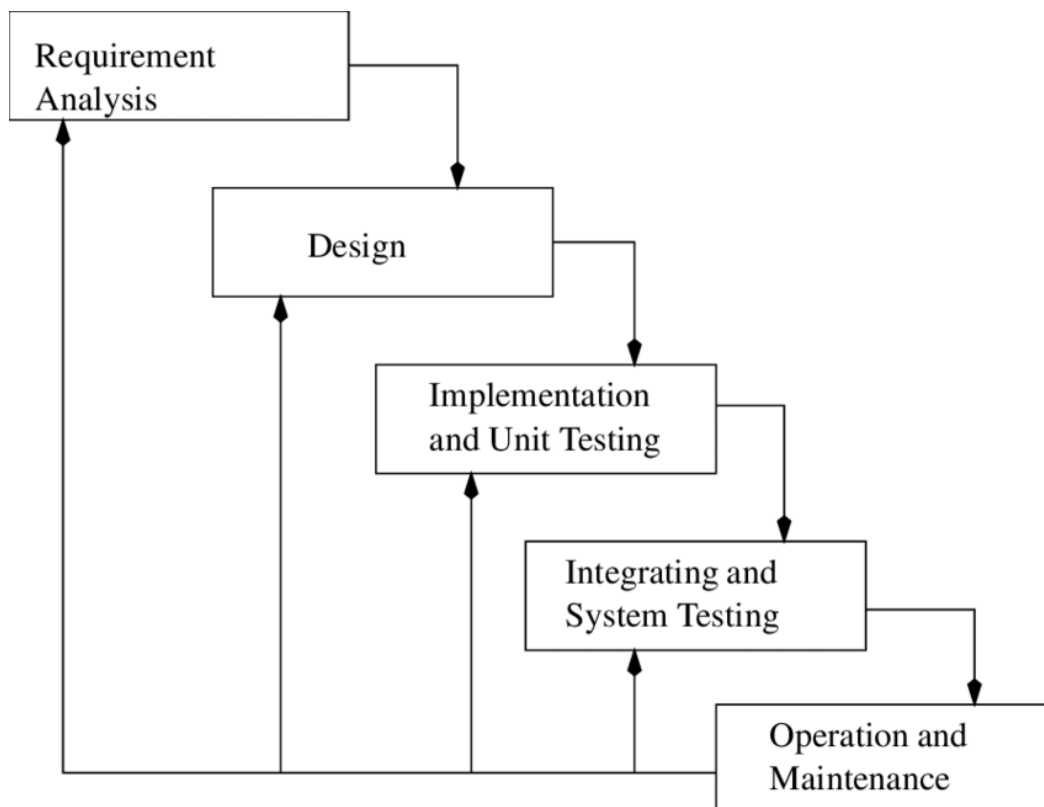
### **1.4.2 Limitation**

There are some criteria that may not be fulfilled by our application implemented. Some of such limitations of our project are mentioned below:

- Incomplete registration of blood donors: The effectiveness of the system depends on the number of registered blood donors, which may not be comprehensive due to incomplete registration.
- Dependence on donor participation: The system relies on donor participation, which may not always be consistent and reliable, leading to potential shortages in blood supply.
- Cannot add donation events or happenings: The system is not capable enough to add any random happenings of blood donation events.

## 1.5 Development Methodology

This project is based on the waterfall model. For this system is designed with the series of processes starting with requirement analysis, design, implementation, testing and maintenance. During requirement analysis, the entire functional and nonfunctional requirement is analyzed and system is developed according to the requirement then designing of the system is carried out. After the design process, coding and development part is started then after integrating the system there is testing of the system. If the testing is positive then system is implemented otherwise some maintenance is done and system come in operation.



**Fig 1.5.1: Waterfall Model**

## **1.6 Report Organization**

This report on “Rakt Seva” follows an object based approach that highlights the key aspects of the project. The report is divided into six chapters that provide a comprehensive overview of the project's design, development, and testing phases.

Chapter 1 serves as an introduction to the project and provides a broad overview of the project's main points. This chapter sets the stage for the subsequent chapters and provides a context for the project's development.

Chapter 2 contains the theoretical literature review, which provides an in-depth analysis of the existing research and identifies potential hypotheses, strategies, and shortfalls. This chapter helps to inform the design and development of the project.

Chapter 3 contains the details of the project's design and development stages, as well as the basic process model. This chapter explains the methodology used in the development of the project, and it provides an overview of the project's architecture.

Chapter 4 contains high-order UML diagrams that give insight into the system design and algorithms used in the development of the project. This chapter helps to visualize the project's architecture and provides an understanding of how the various components of the system work together.

Chapter 5 contains the system testing, which includes the execution of the program or system with the intent of finding errors. This chapter explains the testing methodology used in the project and provides an analysis of the results.

Chapter 6 wraps up the project by providing recommendations for future initiatives. This chapter discusses the strengths and weaknesses of the project and identifies areas for improvement. It also provides suggestions for future research and development.

Overall, this report on "Rakt Seva" follows a systematic approach that provides a comprehensive understanding of the project's design, development, and testing phases. It provides valuable insights into the methodology used in the development of the project and identifies potential areas for improvement in the futur



## **CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW**

### **2.1 Background study**

The concept of a blood bank system has evolved over time with the advancement of technology and the growing need for efficient and effective blood donation systems.

The first successful blood transfusion was performed in 1665, but it was not until the early 20th century that blood typing was discovered, making it possible to match blood types for transfusions. The first blood bank was established in 1937, and the practice of blood donation became more organized and regulated.

With the advent of computers and the internet, blood donation management systems were developed to improve the efficiency of blood donation processes. The first web-based blood donation system was launched in India in 2001, called the Indian Blood Donors Database. This system aimed to connect blood donors with patients in need of blood transfusions.

Since then, many countries have developed their own blood bank systems, with varying degrees of success. Today, technology continues to play a critical role in managing blood donation processes, and many blood banks and organizations use web-based donor locator systems to help ensure a steady supply of blood for those in need.

### **2.2 Literature Review**

Blood bank in Nepal- BPAZES” has the facility of showing different blood banks, its location and contact number which are inside and outside Kathmandu valley, but the database is not updated timely. In this system the location and contact number is shown manually. In our system, the donors can search the blood bank near to them. Thus, the current system is not effective and doesn't give operative output for the blood donation. There are many other organizations or similar systems that try to provide blood bank facilities in need but fail due to various factors such as lack of awareness, false rumor, and outdated system inappropriate blood management systems etc.[1]

Hamro Life bank is a blood bank management system website of Nepal. It is a social initiative for a smart, transparent, and holistic blood management service from collection to supply. When it comes to blood right information at right time can be the answer to the life and death situation. [2]. This website has the feature of providing a list of blood banks and their location and contact information which are inside and outside the Kathmandu valley. It has a systematic blood application but doesn't suggest the nearest blood bank in the time of emergency. Providing the list of donors to the blood bank when blood requested is insufficient is also lacking behind.

It doesn't contact the donors directly from time to time. It has a feature of contacting hotline numbers but not contacting directly to the donor when there is shortage of blood in the blood bank. Contacting the list of donors if requested blood is insufficient, is the important feature lacking behind in this site.

“Nepal Red Cross Society- *NCRS*” is an organization which organize blood donation events. This application has the facility of displaying the location, date, and time of blood donation events in different blood banks or outside the blood banks.[3] The current system is not effective because it keeps the records of donors manually. It would be difficult to manage and store and manage the records. The records of the users might not be kept safely and there might be missing donor records due to human error or disasters. Here only the donation events details is published but it does not contact the donors individually.

Modern medical care, including surgery and medical treatment for many diseases, is not possible without the use of blood products (SPENCER S. ECCLES HEALTH SCIENCE LIBRARY, 2015). According to the American Journal of Epidemiology,” it was found that blood donors are 88 percent less likely to suffer a heart attack than those who do not donate. Blood bank services are mostly offline, and hospital based. In hospital-based blood bank services, each hospital has its own database. Though this database is useful for management of the data they can only be accessed by the hospital staff and administration. The user cannot know the status of blood without reaching the blood banks. And the interested donors also don't know about the events for blood donation programs happening in different areas. People interested in giving blood can also get to use this platform for their convenience.[4]

Resolution WHA63.12 of the World Health Assembly urges all Member States to develop national blood systems based on voluntary unpaid donation and to work towards the goal of self-reliance. An increase of 7.8 million blood donations from voluntary unpaid donors has been reported from 2013 to 2018. [5] In total, 79 countries collect over 90percent of their blood supply from voluntary unpaid blood donors; however, 56 countries collect more than 50 percent of their blood supply from family/replacement or paid donors.

A total of 110 countries reported data on the number of voluntary unpaid whole blood donations given by first-time donors and repeat donors. Overall, the percentage of whole blood donations given by repeat voluntary unpaid blood donors in countries ranged widely from less than 0.1% to 100% (median 45%).[6]

According to WHO, only the donor of age between 18 – 65 are eligible to donate blood. The donors should be in healthy in healthy state and should not have consume any drugs.[7]

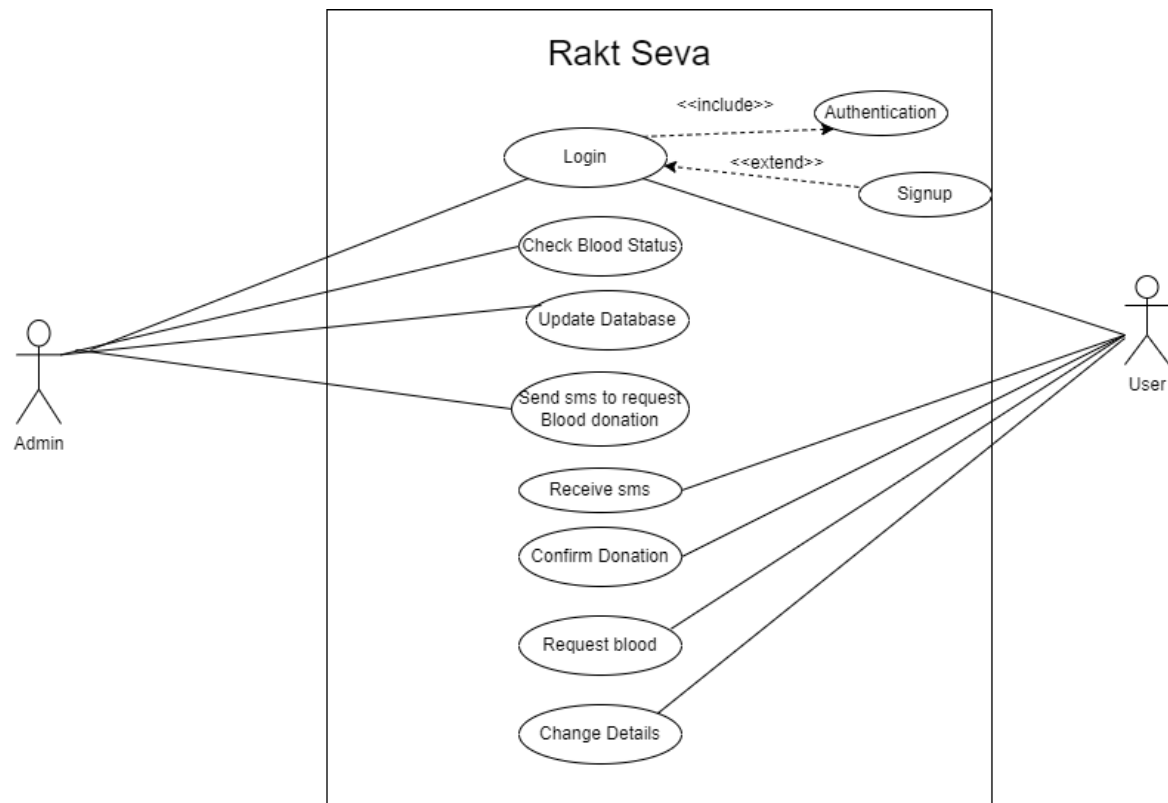
## CHAPTER 3: SYSTEM ANALYSIS

### 3.1 System Analysis

#### 3.1.1 Requirement Analysis

##### i. Functional Requirement

The system should help users to find the nearest blood bank. This system requires every user to register if they want to use any services. They can view the system without registration and without logging in. Secure registration and profile management facilities for different users. The system should provide a platform for donors to donate blood to blood bank locating nearest to them and request blood at any emergency need through this system.



**Fig 3.1.1.1: Use-case Diagram**

## **ii. Non-Functional Requirement**

- Safety and Security Requirements

This system ensures the security of users' information and prevents misuse.

- Scalability

The system is expected to be scalable so that it will be suitable for a very large data list.

- Efficiency

The performance of the system should be efficient.

- Accuracy

The system should be able to provide accurate results.

- Reliability

The system should be reliable with the results produced. To be reliable all the bugs and errors that may appear while running the system must be fixed.

### **3.1.2 Feasibility Analysis**

#### **i. Technical feasibility**

The system is technically feasible as the requirement for the development of the system is easily accessible. The necessary hardware and software required for the development and implementation of the system is available. The basic programming language which is suitable for project is available. For the development and maintenance of the system all the existing resources can be used.

#### **Hardware requirements**

The hardware requirements of this project are:

**Table 3.1.2.1: Hardware Requirements**

Hardware Requirements	
System Architecture	Any standard x86 and x64 bit computer

Memory	4 GB RAM or less
Storage and Type	Minimum 1 GB free HDD/SSD space.

### Software requirements

The software requirements of this project are:

**Table 3.1.2.2: Software Requirements**

Software Requirements	
Web Browsers	Any Modern Browsers: <ol style="list-style-type: none"> <li>1. Chrome - Latest stable release</li> <li>2. Safari - Latest stable release</li> <li>3. Firefox - Latest stable release</li> <li>4. Edge - Latest stable release</li> <li>5. Opera- Latest stable release</li> </ol>
Bandwidth Speed Requirements	bps (Internet Speed Requirement)

### ii. Economic feasibility

All the tools required for the development are freeware tools made with the utilization of easily available hardware and software that we already have access to. So, there is no such investment needed. The development costs, annual operating costs, annual benefits, and intangible costs and benefits make the system affordable and economically feasible. The manpower needed for the databases and resources are quite low which helps in the

minimization of production cost. On behalf of cost-benefit analysis, the developed system is feasible and is economical regarding its pre-assumed cost for making a system.

### **iii. Operational feasibility**

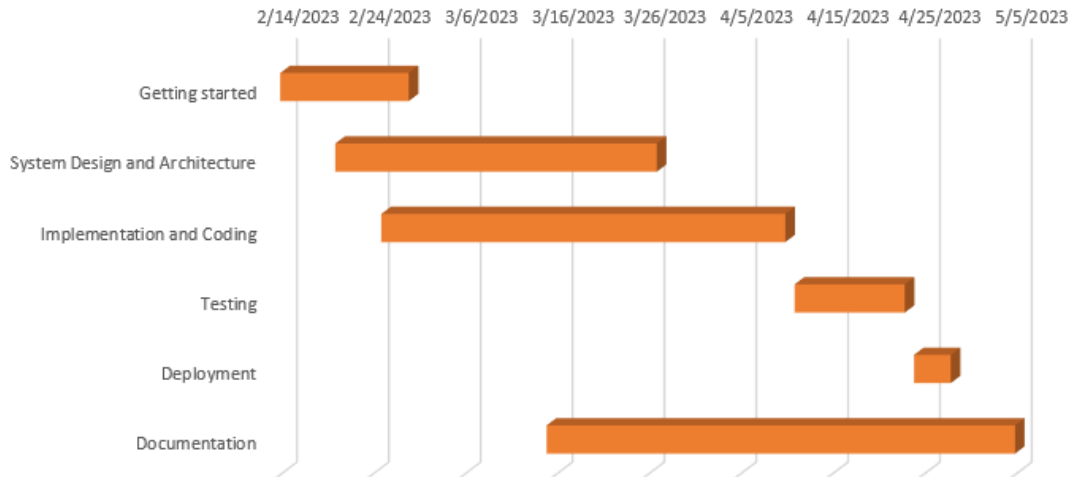
“Rakt Seva” is a good platform to for donors who are interested in donating blood as it provides convenient location for them to donate blood. It is expected to be operationally feasible to provide cost-effective information services to the users. It is designed to make maximum use of available resources, including blood, people, time, and the flow of related collected information. It is expected to solve blood-related issues in time, thus we are certain the web application will be of great use.

### **iv. Schedule Feasibility**

Schedule feasibility is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

**Table 3.1.2.3: Gantt chart Table**

Task Name	Time Duration
Getting Started	14
System Design And Architecture	35
Implementation and coding	44
Testing	12
Deployment	4
Documentation	51



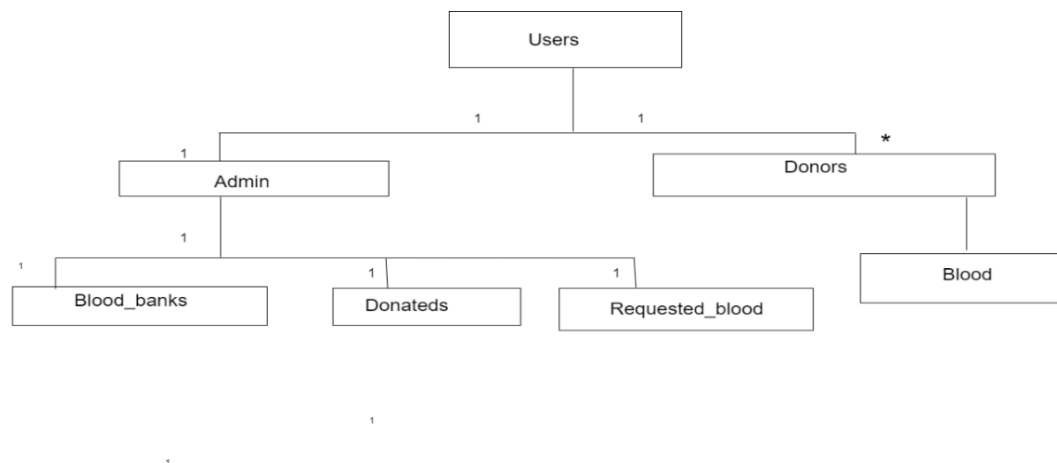
**Fig 3.1.2.1: Gantt Chart**

### 3.1.3 Analysis

#### 3.1.3.1 Object Modeling

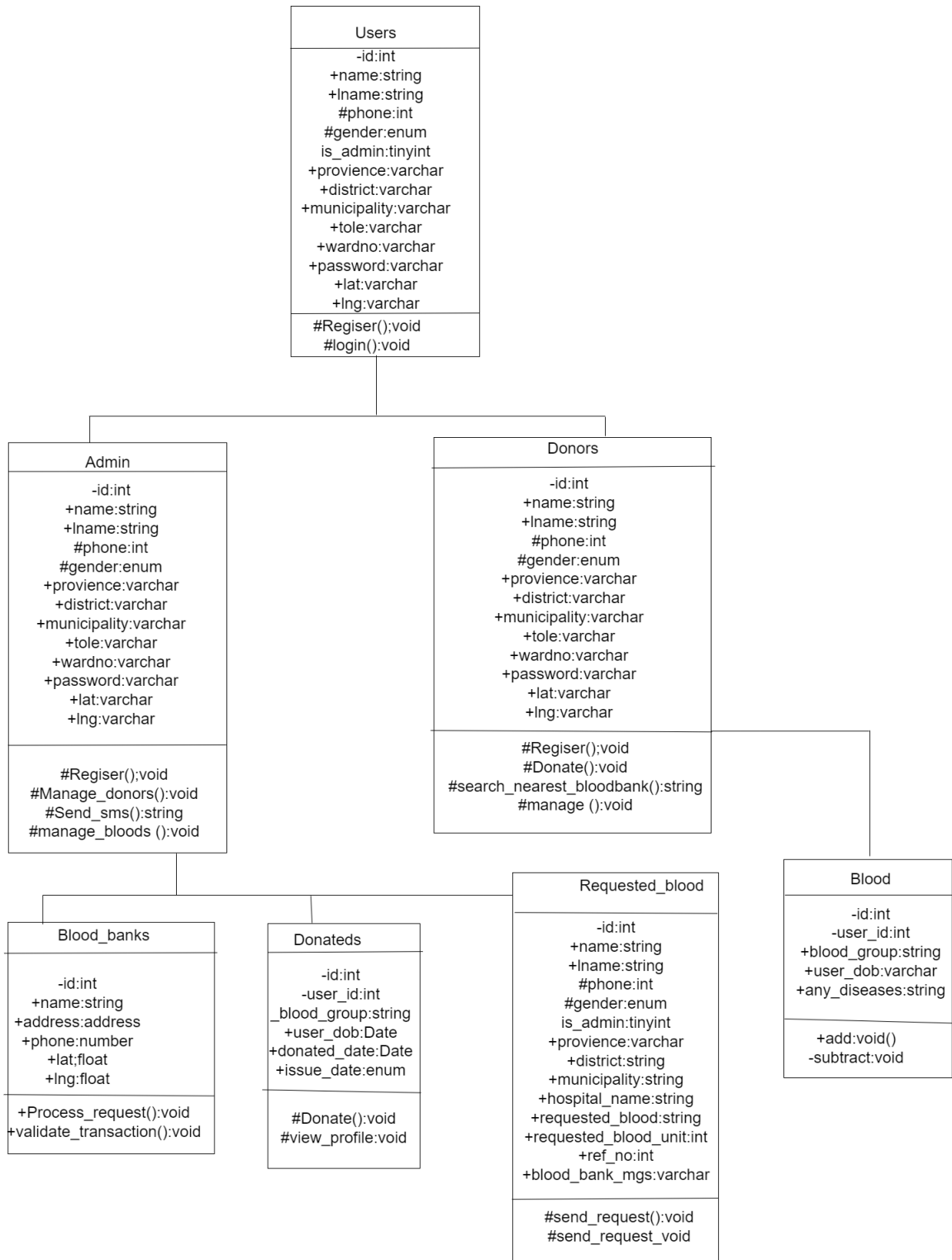
The object model visualizes the elements in a software application in terms of objects and classes. It defines the interfaces or interactions between different models, inheritance, encapsulation and other object-oriented interfaces and features.

The class and object diagram visualizes the particular working functionality of the system. The UML diagram of Rakt Seva only depicts the abstract view of the diagram where the object represents the static view whereas class represents the behavior of those objects.



**Fig 3.1.3.1 Object Diagram**



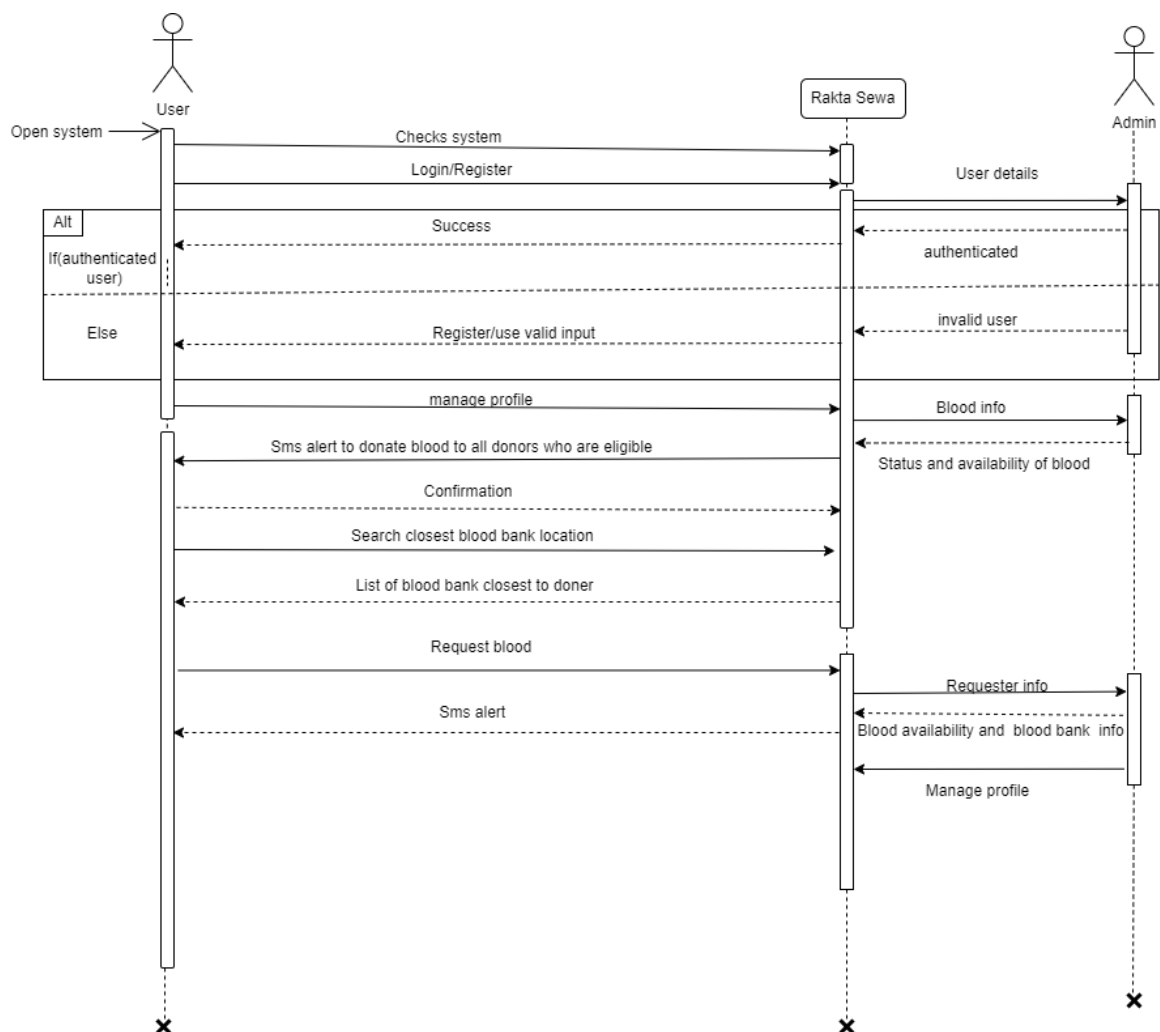


**Fig 3.1.3.2 Class Diagram**

### 3.1.3.2 Dynamic Modeling

Dynamic model describes the internal behavior of a system. Dynamic Modeling represents the temporal aspects of a system, capturing the control elements through which the behavior of objects can be understood over time. The Dynamic Model describes those aspects of a system concerned with time and the sequencing of operations - events that mark changes, sequences of events, and the organizing of events and states. The Dynamic Model does not consider what the operations do, what they operate on, or how they are implemented.

The Sequence diagram presented below is the dynamic model of our system. It describes the internal behavior of our system. It shows how a user will encounter and interact with our system. It also shows how the user query is processed internally.

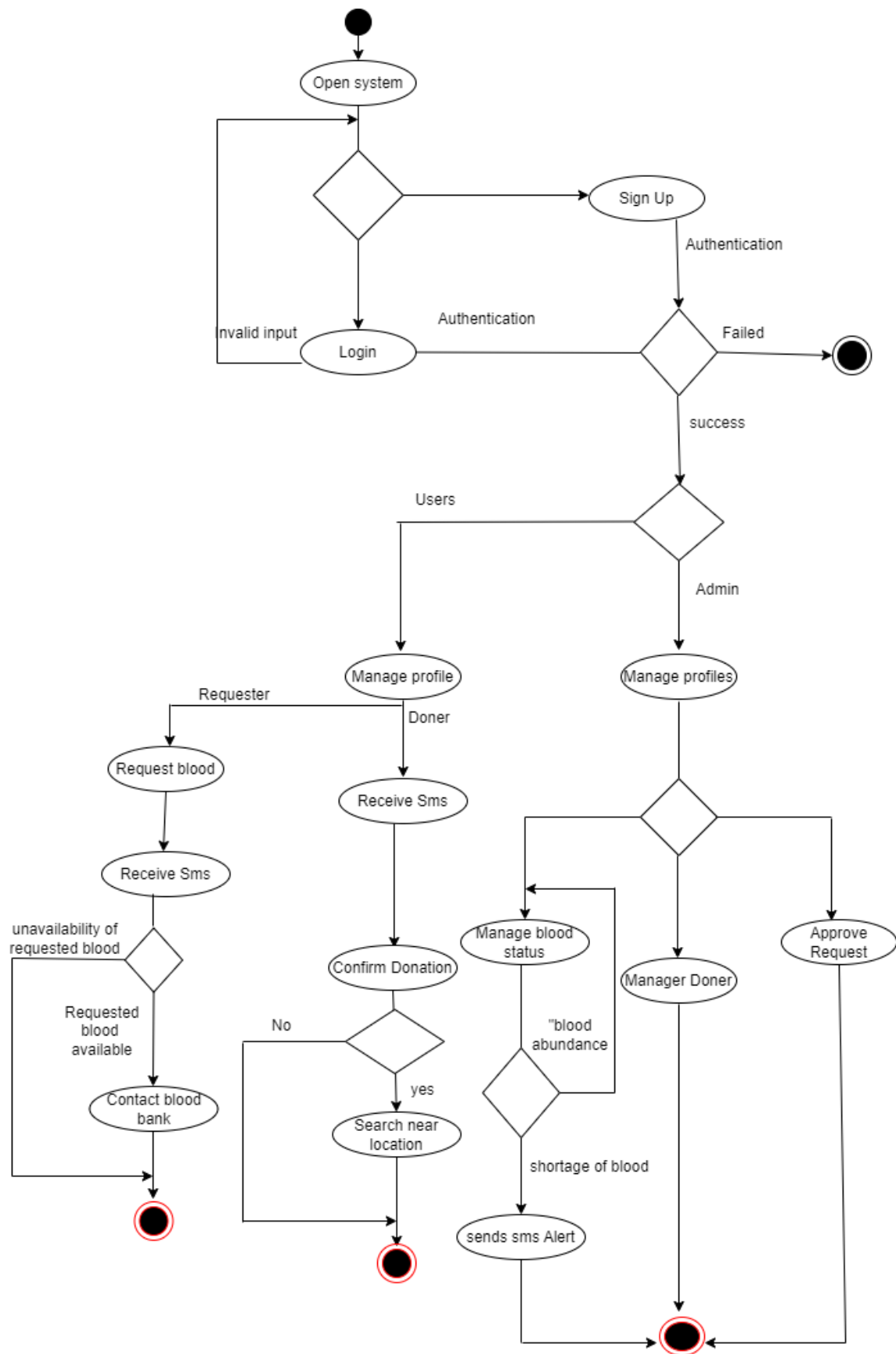


**Fig 3.1.3.3 Sequence Diagram**

### **3.1.3.3 Process Modeling**

The act of process modeling provides a visualization of business processes, which allows them to be inspected more easily, so users can understand how the processes work in their current state and how they can be improved.

The activity diagram presented below shows the process modeling of this system. It shows every step from signup/login that the user follows when he/she arrives in this system.



**Fig 3.1.3.3.1 Activity Diagram**

## CHAPTER 4: SYSTEM DESIGN

### 4.1 Design

In addition to that Component and Deployment diagrams are drawn. In our case we don't need to refine the initially presented Class, Object, Sequence and Activity diagrams.

#### 4.1.1 Refinement of object, class and sequence diagram

The following diagram is the refinement diagram of object diagram which helps to understand the structure and organization of objects of a system.

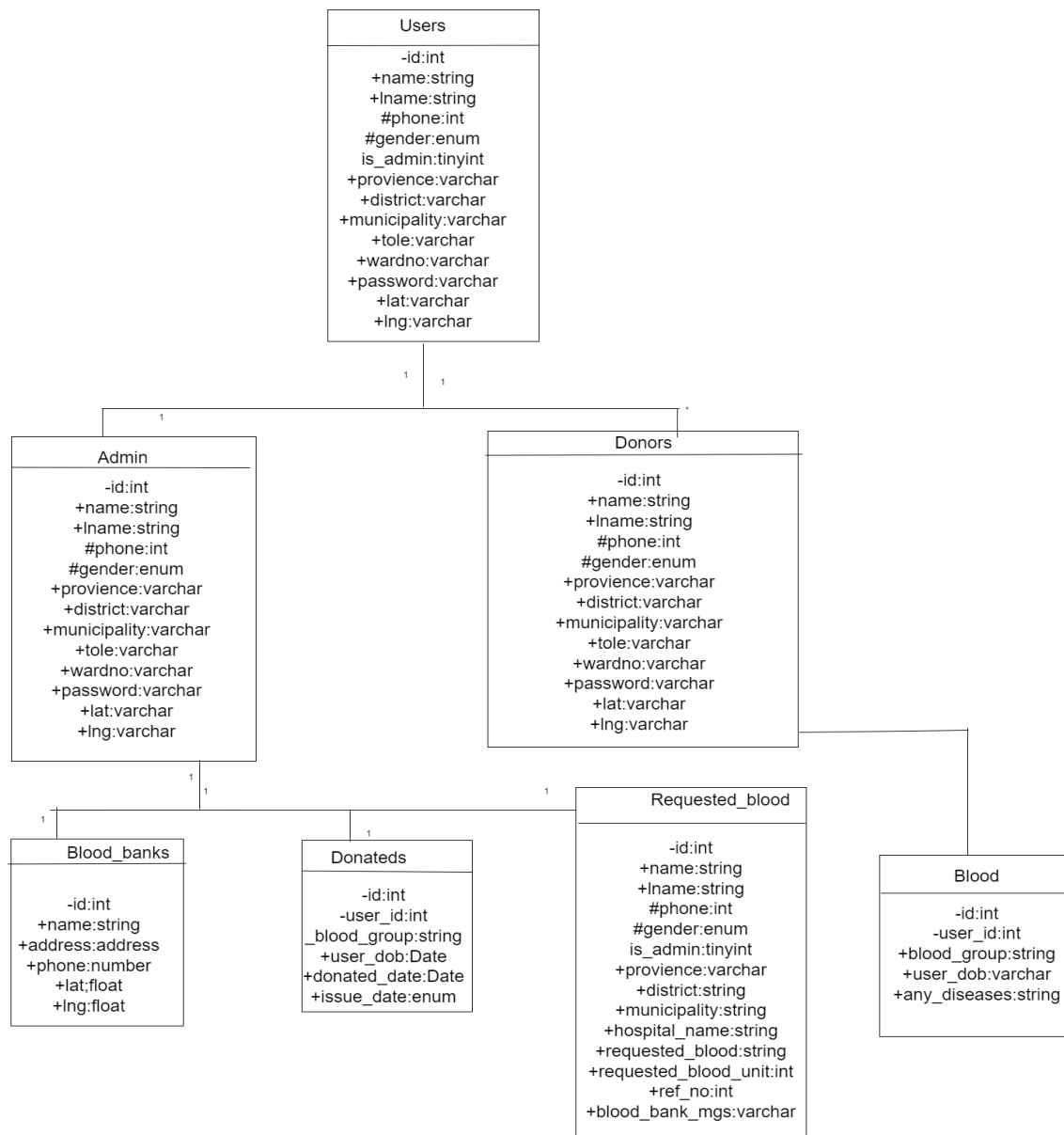
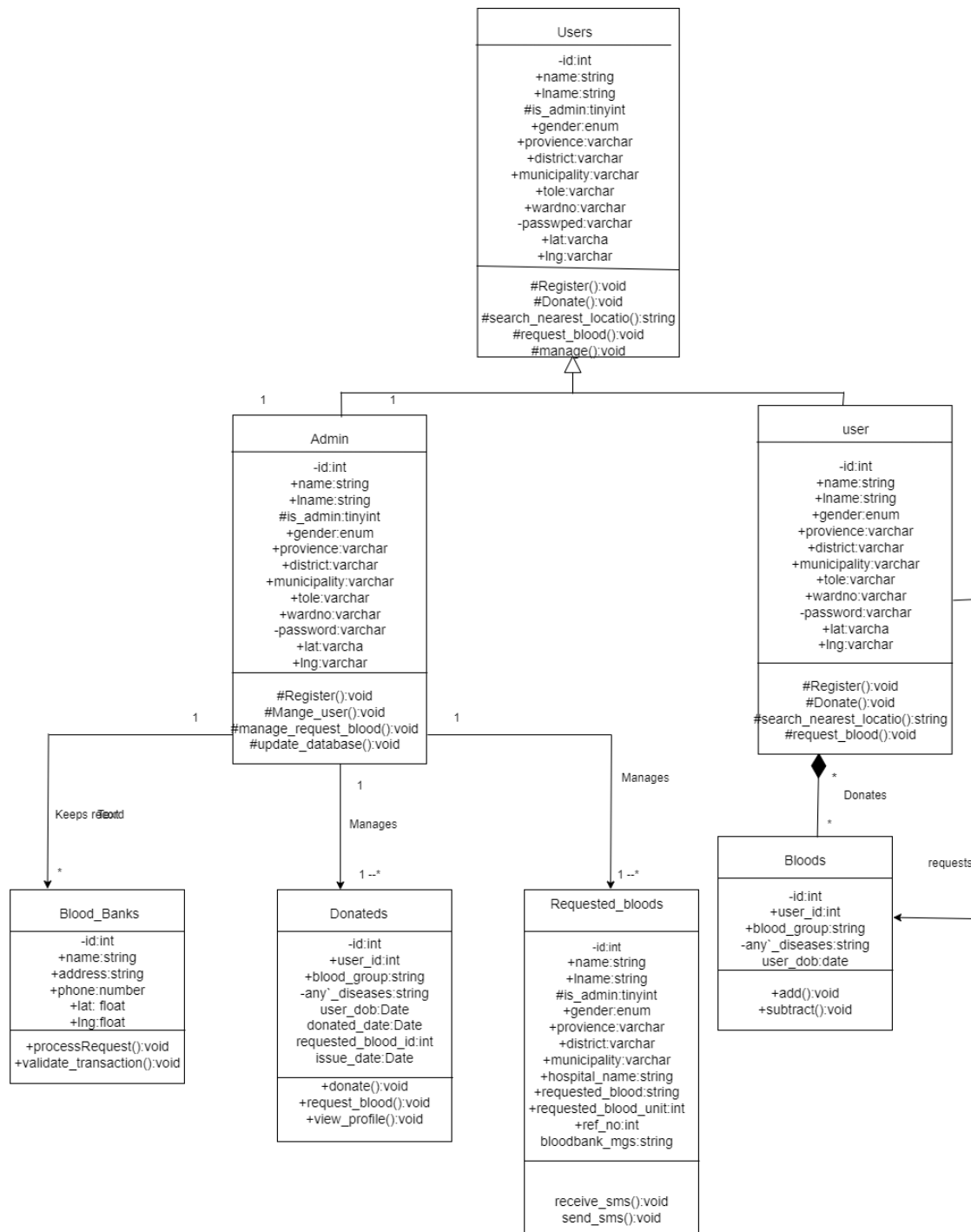


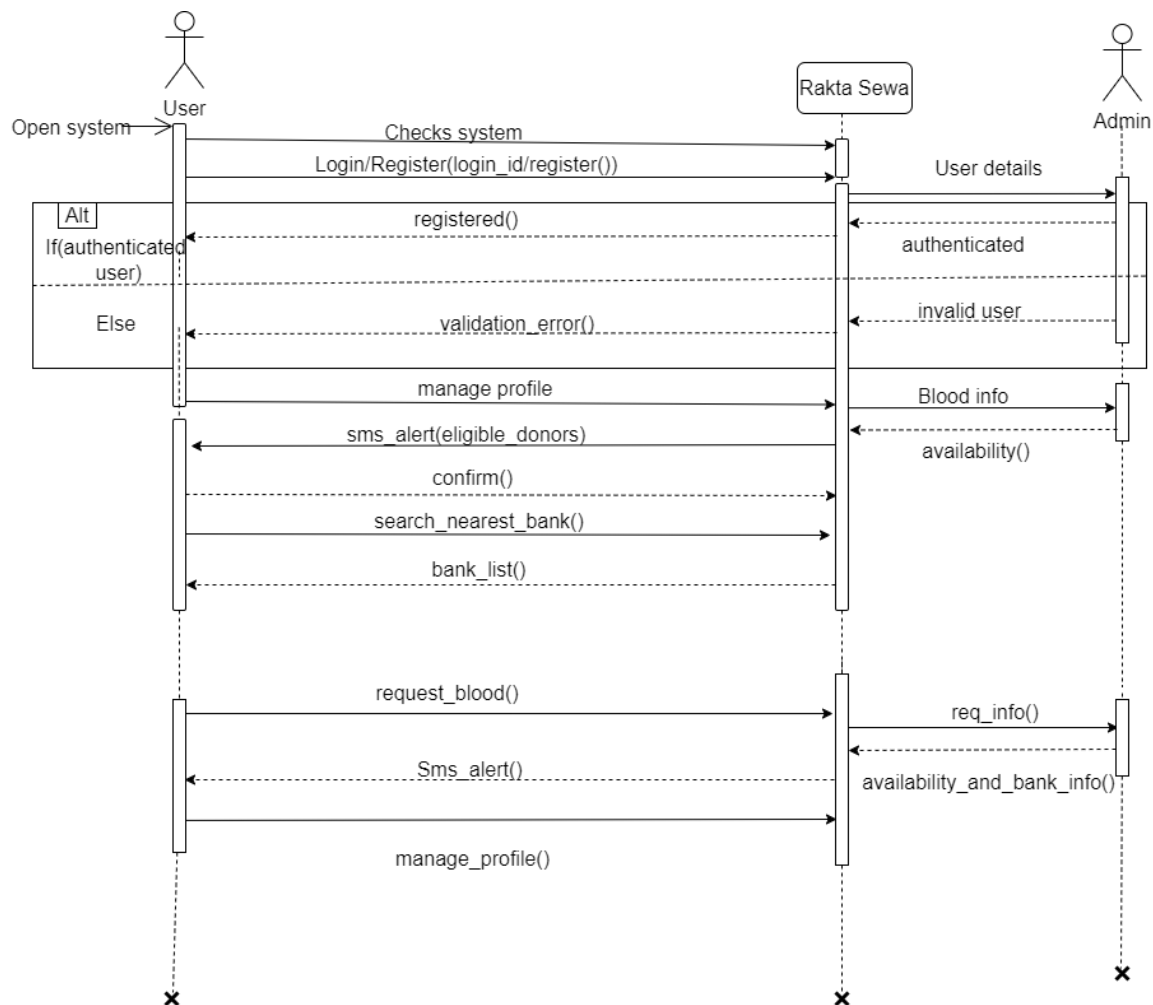
Fig 4.1.1.1 Refinement Object Diagram

The following diagram is the refinement diagram of class diagram which helps in understanding the relationships between classes, identifying the responsibilities and collaborations among classes, and providing a foundation for generating code.



**Fig 4.1.1.2 Refinement of Class Diagram**

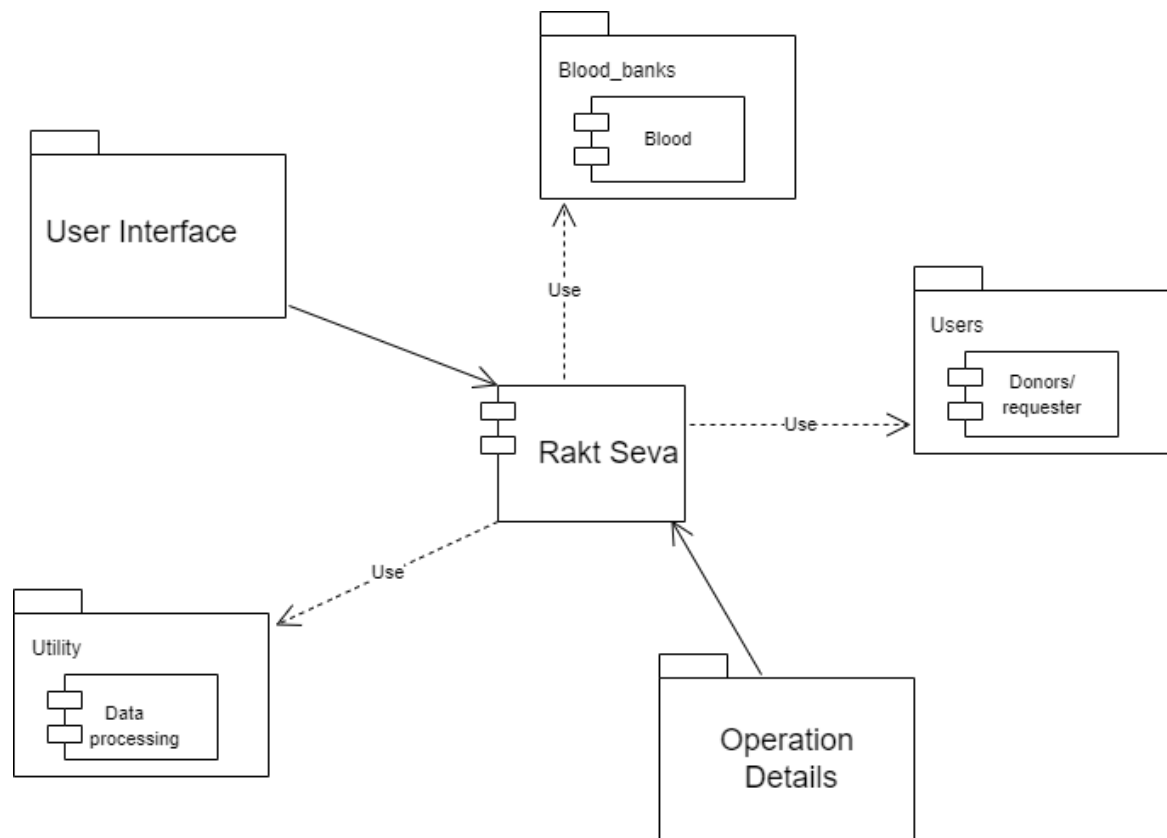
The following diagram is the refinement diagram of sequence diagram which helps in understanding, designing, and documenting the dynamic behavior of a system. They help to visualize the flow of messages and interactions between objects, facilitating requirement analysis, system design, and communication among team members.



**Fig 4.1.1.3 Refinement of Sequence Diagram**

#### 4.1.2 Component Diagram

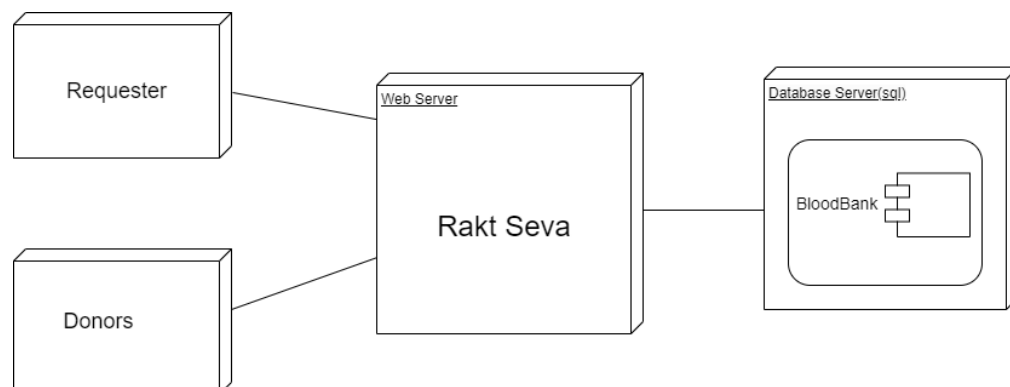
Component diagram helps to understand the structure, relationships, and behavior of this system. This aid in system design, analysis, communication, and facilitate modularity and reusability in this system.



**Fig 4.1.2.1 Component Diagram**

### 4.1.3 Deployment Diagram

Deployment diagram illustrates the physical deployment configuration of this system and aid in understanding the distribution and interaction of software components across hardware nodes. It helps system architects, developers, and system administrators to visualize the hardware infrastructure and software component distribution.



**Fig 4.1.3.1 Deployment Diagram**



## 4.2 Algorithm Details

In this system, Haversian algorithm is used to find the distance between blood-bank and user and provide them the distance between them in km. as well as this system provide the users the nearest blood-bank from their current location. For this following steps have been used:

- Find the longitude and latitude value of their location by marking on graph during registration as well as by entering the name of the current location when required.
- Calculate the difference in latitude and longitude between the two points i.e. user location and blood bank.

Haversian formula has been used to calculate the distance between the two points. The Haversian formula is as follows:

```
$earthRadius = 6371; // radius of the earth in kilometers
$latDiff = deg2rad($lat2 - $lat1);
$lonDiff = deg2rad($lon2 - $lon1);
$a = sin($latDiff / 2) * sin($latDiff / 2) + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) *
sin($lonDiff / 2) * sin($lonDiff / 2);
$c = 2 * atan2(sqrt($a), sqrt(1 - $a));
$distance = $earthRadius * $c; // distance in kilometers
return $distance;
```

Here, Latitude and longitude of any location is calculated using Google OpenSource API i.e Nominatim.

**Code Snippet is shown below:**

```
$UserCoord = array($userlat, $userlng); //stores user coord;
$location = $location;
$bloodcoord = BloodBank::select('lat', 'lng', 'name')->get();
//find user coord
$url = "http://nominatim.openstreetmap.org/";
$nominatim = new Nominatim($url);

$search = $nominatim->newSearch()
->country('Nepal')
->city($location)
```

```
->polygon('geojson') //or 'kml', 'svg' and 'text'
->addressDetails();

$result = $nominatim->find($search);

if($result){

    array_push($UserCoord, $result[0]['lat'], $result[0]['lon']); //gather user coord
}
else{
    dd('Address not found');
}
```

## CHAPTER 5: IMPLEMENTATION AND TESTING

### 5.1 Implementation

#### 5.1.1 Tools Used

CASE as in Computer-Aided Software Engineering is a tool that helps with the analysis and design of software systems. It is a piece of software that helps in the development and maintenance of other software products. Code generators, Unified Modeling Language (UML) tools, refactoring tools, and model transformation tools are examples of such tools.

##### 5.1.1.1 Programming Language Tools

The Rakt Seva uses the .Laravel framework as the primary language to build the backend of the application. For the front-end task we are using HTML, CSS for the structure and styling of the web application, bootstrap for responsiveness, and JS for the front-end for the web interface. The Laravel application architecture follows MVC (Model, View, Controller) to serve the application on the website.

##### 5.1.1.2 Database Tools

MY SQL database has been used in the development phase of our application. It is an open-source object-relational database and is flexible with our Laravel integration

##### 5.1.1.3 Drawing Tools

For the drawing of our diagrams we are using Online Diagram Software (Draw.io), an online development tool for making Activity Diagram, Class Diagram, and UML.

#### 5.1.2 Implementation Details of Modules

##### 5.1.2.1 User and Admin login

```
public function login(Request $request){  
    $input = $request->all();  
    $this->validate($request,[  
        'mobile' => 'required',  
        'password' => 'required',  
    ]);  
    if(auth()->attempt(array('mobile' => $input['mobile'], 'password' =>
```

```

$input['password']))))
{
    if(auth()->user()->is_admin == 1){
        return redirect()->route('admin.home');
    }
    else{
        return redirect()->route('home');
    }
}
}
else{
    return redirect()->route('login')->with('error', 'Mobile number or password wrong
!!!');
}
}
}

```

#### 5.1.2.2 Adding New Donors

```

public function postSearchDonner(Request $request){
    $province = $request->get('province');
    $district = $request->get('district');
    $minicipality = $request->get('minicipality');
    $wordno = $request->get('wordno');
    $data=[
        'results' => Blood::where('blood_group', $request->get('bloodgroup'))->get(),
        'request' => $request
    ];
    return view('user.searchresult', $data);
}

public function getContribution(){
    $data =[
        'donates' => Donated::where('user_id', Auth()->user()->id)->get()
    ];
}

public function AddNewDonor(Request $request){
    $fname = $request->input('name');
    $lname = $request->input('lname');
}

```

```

$gender = $request->input('gender');
$email = $request->input('email');
$bloodgroup = $request->input('bloodgroup');
$dob = $request->input('dob');
$makeaccount = $request->input('makeaccount');
$check = User::where('mobile', $email)->count();
if($check == 0){
    $password = Str::random(4);
    $user = New User;
    $user->name = $fname;
    $user->lname = $lname;
    $user->is_admin = '0';
    $user->mobile = $email;
    $user->gender = $gender;
    $user->password = Hash::make($password);
    $user->save();
}
if($makeaccount == 'on'){
    $client = new Client();
    $text_message = 'Dear Donor, Your New login username: '.$email. ' password: ' .
$password.' For login visit http://localhost:8000/login';
    $res = $client->request('POST', 'https://sms.aakashsms.com/sms/v3/send', [
        'form_params' => [
            'auth_token' =>
'e17b9097e6ec4450ed488ee536924d2b41e4ec8a6ffdac7cff5e2aed0cf4a3c7',
            'from' => '31001',
            'to' => $user->mobile,
            'text' => $text_message,
        ]
    ]);
}
$blood = New Blood;
$blood->blood_group = $bloodgroup;
$blood->user_id = $user->id;
$blood->dob = $dob;

```

```

        $blood->save();
        return redirect()->back()->with('message', 'Donor added successfully');
    }
    else{
        return redirect()->back()->with('message', 'Unable to add, due to mobile number
duplicate entry');
    }}

```

The above code is used for adding new donors into the database.

#### 5.1.2.2 Adding New Blood Banks

```

    public function create()
    {
        return view('admin.bloodbank.create');
    }

    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'name' => 'required',
            'address' => 'required',
            'phone' => 'required',

        ]);

        $bloodBank = new BloodBank();
        $bloodBank->name = $validatedData['name'];
        $bloodBank->address = $validatedData['address'];
        $bloodBank->phone = $validatedData['phone'];

        $bloodBank->save();

        return redirect('/admin/bloodbank')->with('success', 'Blood bank created
successfully!');
    }

```

### 5.1.2.3 Sending SMS to users

```
public function postResponse(Request $request, Requestedblood $bloodrequest){
    $user = User::find($bloodrequest->user_id);
    $bloodrequest->accepted = $request->input('response11');
    $bloodrequest->bloodbankmessage = $request->input('message1');
    $bloodrequest->save();
    if($request->input('sendsms') == 'on'){
        $client = new Client();
        if($request->input('response11') == 'Yes'){
            $text_message = 'Dear ' . $user->name. ', Please visit blood bank to collect your
requested blood.';
        }
        else{
            $text_message = 'Dear ' . $user->name. ', Sorry we do not have requested blood group
yet.';
        }
        $res = $client->request('POST', 'https://sms.aakashsms.com/sms/v3/send', [
            'form_params' => [
                'auth_token' =>
'e17b9097e6ec4450ed488ee536924d2b41e4ec8a6ffdac7cff5e2aed0cf4a3c7',
                'from' => '31001',
                'to' => $user->mobile,
                'text' => $text_message,
            ]
        ]);
    }
    return redirect()->back()->with('message', 'Response send successfully');
}
```

### 5.1.2.3 Comparing the Location of Donors

```
class Distance extends Model
{
    use HasFactory;

    public static function getDistanceBetweenPointsNew($latitude1, $longitude1, $latitude2,
$longitude2, $unit='Km') {
        $theta = $longitude1 - $longitude2;

        $distance = sin(deg2rad($latitude1)) * sin(deg2rad($latitude2)) +
cos(deg2rad($latitude1)) * cos(deg2rad($latitude2)) * cos(deg2rad($theta));

        $distance = acos($distance);
        $distance = rad2deg($distance);
        $distance = $distance * 60 * 1.1515;
        switch($unit)
        {
            case 'Mi': break;
            case 'Km': $distance = $distance * 1.609344;
        }
        return (round($distance,2));
    }

    public static function getUserNerestDistance($userlat, $userlng, $location){
        $UserCoord = array($userlat, $userlng); //stores user coord;
        $location = $location;
        $bloodcoord = BloodBank::select('lat', 'lng', 'name')->get();
        //find user coord
        $url = "http://nominatim.openstreetmap.org/";
        $nominatim = new Nominatim($url);

        $search = $nominatim->newSearch()
            ->country('Nepal')
            ->city($location)
            ->polygon('geojson') //or 'kml', 'svg' and 'text'
            ->addressDetails();

        $result = $nominatim->find($search);
    }
}
```



```

if($result){

array_push($UserCoord, $result[0]['lat'], $result[0]['lon']); //gather user coord
}
else{
    dd('Address not found');
}

//haversine
function haversineDistance($lat1, $lon1, $lat2, $lon2)

{

    $earthRadius = 6371; // radius of the earth in kilometers

    $latDiff = deg2rad($lat2 - $lat1);
    $lonDiff = deg2rad($lon2 - $lon1);

    $a = sin($latDiff / 2) * sin($latDiff / 2) + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) *
sin($lonDiff / 2) * sin($lonDiff / 2);
    $c = 2 * atan2(sqrt($a), sqrt(1 - $a));

    $distance = $earthRadius * $c; // distance in kilometers

    return $distance;

}

$bestDistance = INF;
$closestLocationName = "";

foreach ($bloodcoord as $blood) {
    $distance = haversineDistance($result[0]['lat'], $result[0]['lon'], $blood['lat'],
    $blood['lng']);

```

```

        if ($distance < $bestDistance) {
            $bestDistance = $distance;
            $closestLocationName = $blood['name'];
        }
    }

    return "Our location " . $location . ', Nearest blood bank ' . $closestLocationName;
}
}

```

The above code helps us to search the donors nearest to the location.

## 5.2 Testing

Software testing is a process of evaluating a software product to check whether it meets the specified requirements and works as expected. The testing process involves the use of manual or automated tools to execute the software components and to verify its behavior, performance, functionality, and other properties of interest. The goal of software testing is to identify defects, errors, or missing requirements and to ensure that the software product is of high quality and free of defects. Testing can be done at different stages of the software development life cycle, including unit testing, integration testing, system testing, and acceptance testing.

### 5.2.1 Test Cases for Unit Testing

The unit testing basic approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not. In unit testing, various modules have been tested individually. This has been done manually to test if the expected result is actually seen on the screen. Various unit test cases are shown below:

**Table 5.2.1.1: Test Cases for Unit Testing**

S.N.	Test Case Description	Test Data	Expected Result	Output Result
<b>Donor Registration and Login</b>				
1.	The user registers with a unique email and username.	Email: <a href="mailto:donortest@gmail.com">donortest@gmail.com</a> Password: Donor@123 Role: Donor	Hello donortest@gmail.com (Successful Registration.)	The user successfully registered.
2.	The user registers an invalid email and username.	Email: <a href="mailto:use@ab.com">use@ab.com</a> Password: abc	Error message.	Validation error.
3.	The log in with registered user parameters	Email: donortest@gmail.com Password: Donor@123	Hello donortest@gmail.com (Login successful.)	Login is successful.
4.	The user logs in with incorrect parameters	Email: use@ab.com Password: abc	Login error.	Login error.
<b>Add Donor</b>				
5.	Users log in and provide information.	Email: donortest@gmail.com Password: Donor@123	The admin gets the notification and adds them	The user was added successfully.

<b>Blood Requester</b>				
6.	The user requests for blood by clicking the request button.	Click the “Request” button	After clicking user is taken to request form.	Request form displayed successfully.
<b>Donor Search Location</b>				
7.	Donor types the address of feasible donation location.	Test Data Address name: Satdobato	After clicking search button the list of blood donation location of that address is shown.	Nearest location of blood bank is shown.

The screenshots to demonstrate unit testing is attached in the appendix section.

### 5.2.2 Test Cases for System Testing

System testing is done after the success of unit testing. Once all the individual units are created and tested, we combine all those modules and perform system testing. System testing ensures a complete and fully integrated software product.

**Table 5.2.2.1 : Test Cases for System Testing**

<b>S.N.</b>	<b>Test Case</b>	<b>Expected Outcome</b>	<b>Result</b>
1.	Searching Nearest Location	Entered address along with other nearest blood banks list is shown.	SUCCESS
2.	Responsiveness	The system has some degree of responsiveness but is not intended to be responsive across all devices.	SUCCESS

3.	Login and Registration	The login and registration system works fine.	SUCCESS
----	---------------------------	--	---------

The screenshots to demonstrate system testing is attached in the appendix section.

### 5.3 Result Analysis

More than five different unit test cases under five major features have been tested and the application has worked according to the requirement. This indicates a positive result as the success of all these functions combined ensures the objective of the project. Several screenshots of the successful results have been attached which can provide the details of the result of the test cases. In conclusion, the result analysis of these test cases shows that the system built has been a success.

## **CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION**

### **6.1 Conclusion**

The idea of implementing this system is to provide platform which helps blood donors locate nearby blood banks and also keeping them informed about blood shortages. This platform can send SMS notifications to eligible donors, ensuring that blood supplies are maintained and patients receive timely transfusions. This will help the blood bank to contact the donors when there is scarcity of blood in the blood bank. As well as it also maintain the optimum amount of blood storage to prevent damage of blood.

### **6.2 Future Recommendation**

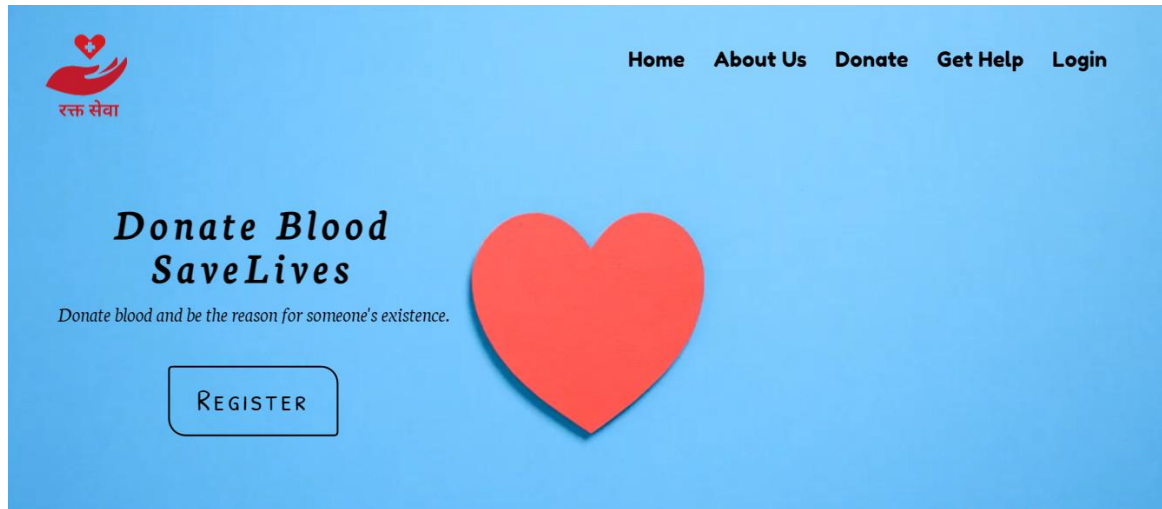
There is always room for improvement, and so does our application. Numbers of functions and features can be added, and bring improvement to the existing ones. Frequent donor list for improving the chances of getting blood. Adding events on system and Pop up notifications of events happening so that the user gets notified and can take part. Hiding the donor from the donor list who has donated the blood for one month so that the seeker won't request for the same person within that time.

## REFERENCES

- [1] “<https://bpazes.com/p/blood-banks>.”
- [2] “hamro lifebank.” [Online]. Available: <https://hamrolifebank.com/>
- [3] “Nepal Red Cross Society.” [Online]. Available: <https://nrccs.org/donate-blood/>
- [4] W. H. Dzik et al., “Patient safety and blood transfusion: new solutions,” *Transfus. Med. Rev.*, vol. 17, no. 3, pp. 169–180, Jul. 2003, doi: 10.1016/s0887-7963(03)00017-8.
- [5] “<https://www.who.int/news-room/fact-sheets/detail/blood-safety-and-availability>.”
- [6] World Health Organization, *The 2016 global status report on blood safety and availability*. Geneva: World Health Organization, 2017. Accessed: Jan. 03, 2023. [Online]. Available: <https://apps.who.int/iris/handle/10665/254987>
- [7] World Health Organization, *Blood donor selection: guidelines on assessing donor suitability for blood donation*. Geneva: World Health Organization, 2012. Accessed: Jan. 04, 2023. [Online]. Available: <https://apps.who.int/iris/handle/10665/76724>

## APPENDIX

### Home page:



### User side

#### User Registration:

The screenshot shows the user registration page of the Rakt Seva website. It has a dark red header with the text 'Rakt Seva' on the left and 'Application' and 'Login' on the right. The main content area is white. On the left side, there is a graphic of a red heart with a pulse line, a red blood drop, and the text 'Save Life'. On the right side, there is a 'SIGN UP' form. The form has a red header with the text 'SIGN UP'. The form fields are: 'First name' and 'Last name' (text boxes), 'Gender \*' (radio buttons for Male, Female, Other), 'Province \*' (dropdown menu), 'District \*' (dropdown menu), 'Municipality \*' (dropdown menu), 'Ward No \*' (dropdown menu), 'City \*' (text box), 'Tole \*' (text box), 'Mobile Number \*' (text box with the value '9863165835'), and 'Password \*' (text box with dots). Below the 'Mobile Number \*' field, there is a small text: 'Mobile number will be your username'. At the bottom of the form is a blue button with the text 'Sign Up'. In the bottom right corner, there is a watermark: 'Activate Windows Go to Settings to activate Windows.'



## User Input:

Application

Login

First name

Last name

Chase Wallace

Camille Ortiz

Gender \*

☒ Male
 ☐ Female
 ☐ Other

Province \*

District \*

Municipality \*

Ward No \*

City \*

Tole \*

Mobile Number \*

Password \*

9849924949

\*\*\*\*\*

Mobile number will be your username

Sign Up

127.0.0.1:8000/home

Rakt Seva

Dashboard

Your Contribution

Search Blood Donor

Requested Blood

New Blood Group Request

List of Blood banks

Naomi Scott (User)

Your Blood Information

Blood Group \*

A+

Date of Birth \*

mm/dd/yyyy

If any diseases or Regular Medicine used, Please define (Optional)

Update

Your Google Location Co-ordinate

lat

lng

Google

This page can't load Google Maps correctly.

Add your Blood Information

Your Blood Profile

Blood Name : O+

Any Diseases or Regular Medicine Used : no

Latest Blood Donate Date : N/A

Your Personal Information

Full Name : Naomi Scott Shellie Bass

Gender : Female Date of Birth: 1999-03-09

Contact Number : 9814829615

Address : Quo in quaerat magni, Ut officia in maxime-12, Tilathi Koiladi Rural Municipality, Saptari, Madesh Province

Your Google Location Co-ordinate

lat

lng

Google

This page can't load Google Maps correctly.

Do you own this website?

OK

Your Google Location Co-ordinate

Submit

## User Login:

Rakt Seva

Login Register

Login

Warning Alerts

Mobile number or password wrong !!!

Mobile Number

9863165835

Password

\*\*\*\*\*

☐ Remember Me

Login

[Forgot Your Password?](#)

Rakt Seva
Dashboard
Your Contribution
Search Blood Donor
Requested Blood
New Blood Group Request
List of Blood banks
Naomi Scott (User)

Add your Blood Information

Your Blood Profile

Your Personal Information

Blood Name : O+
Any Diseases or Regular Medicine Used : no
Latest Blood Donate Date : N/A

Full Name : Naomi Scott Shellie Bass
Gender : Female Date of Birth: 1999-03-09
Contact Number : 9814829615
Address : Quo in quaerat magni, Ut officia in maxime-12, Tilathi Koiladi Rural Municipality, Saptari, Madesh Province

Your Google Location Co-ordinate

lat
lng

Rakt Seva
Dashboard
Your Contribution
Search Blood Donor
Requested Blood
New Blood Group Request
List of Blood banks
Naomi Scott (User)

List of Blood Bank

Blood Bank Name	Address	Contact Number	Distance
Tatiana Dillon	+1 (342) 993-8526	1234567890	7.99 KM Far from You

Search Nearest Blood Bank from anywhere

Location:
Find Shortest Distance

## Request Blood:

Rakt Seva
Dashboard
Your Contribution
Search Blood Donor
Requested Blood
New Blood Group Request
List of Blood banks
Naomi Scott (User)

Blood Request

Patient Information

Full name

Gender \*
Age

☐ Male
☐ Female
☐ Other

Province \*
District \*
Municipality \*

Choose Province
Choose District
Choose Municipality

Word No \*
City \*
Tole \*

Choose word

Admit Hospital Information

Hospital Name \*
Hospital Address \*

Activate Window

Rakt Seva   Dashboard   Your Contribution   Search Blood Donor   Requested Blood   New Blood Group Request   List of Blood banks   Naomi Scott (User) ▾

Requested Blood

Patient Name	Requested Blood	Request Date & Time	Status
No any request yet			

Rakt Seva   Dashboard   Your Contribution   Search Blood Donor   Requested Blood   New Blood Group Request   List of Blood banks   Naomi Scott (User) ▾

Your Contribution

Donate Date	Donate Blood	Donate at
No any contribution yet		

## Search Donors:

Rakt Seva   Dashboard   Your Contribution   Search Blood Donor   Requested Blood   New Blood Group Request   List of Blood banks   Naomi Scott (User) ▾

Search Blood

Province \*

Province 1

Request Blood Group \*

A+

Search

Donor Name	Address	Blood Group	Pervious Donate Date	Contact Number
Audrey Mercer	Incididunt exercitat, Et sunt quidem quia-10 Choose Municipality, Choose District, Choose Province	A+	N/A	9863165835
Vera England	Est ex corporis accu, A sed ad facilis pro-23 Hima, Jumla District, कर्णाली प्रदेश	A+	N/A	1234567890

## Admin Side

Admin Login:

Rakt Seva

LoginRegister

Login

Mobile Number

9863165835

Password

\*\*\*\*\*

☐ Remember Me

Login

[Forgot Your Password?](#)

Rakt Seva

DashboardManage BloodManage DonorBlood RequestBlood BanksAudrey Mercer (Administrator) ▾

Welcome to admin Dashboard

Your Blood Profile

Add /Edit and Update blood bank:

Rakt Seva

DashboardManage BloodManage DonorBlood RequestBlood BanksAudrey Mercer (Administrator) ▾

Blood Banks

Name	Address	Phone	Action
Tatiana Dillon	+1 (342) 993-8526	1234567890	<div>Define GeolocationEditDelete</div>

Add Blood Bank

Rakt Seva
Dashboard
Manage Blood
Manage Donor
Blood Request
Blood Banks
Audrey Mercer (Administrator)

## Blood Banks

Name	Address	Phone	Action
Tatiana Dillon	+1 (342) 993-8526	1234567890	Define Geolocation Edit Delete
Igor Boyer	kadaghari	9876543211	Define Geolocation Edit Delete

Add Blood Bank

## Manage donors:

List of Donor
Add New Donor

Show 10 entries
Search:

ID	FULL NAME	PROVINCE	DISTRICT	MUNICIPALITY	CITY	TOLE	BLOOD GROUP	
4	Naomi Scott Shellie Bass 9814829615	Madesh Province	Saptari	Tilathi Kolladi Rural Municipality	Ut officia in maxime-12	Quo in quaerat magni	O+ Latest Blood donate Date : N/A	Edit Delete
3	Vera England Boris Watson 1234567890	कर्णाली प्रदेश	Jumla District	Hima	A sed ad facilis pro-23	Est ex corporis accu	A+ Latest Blood donate Date : N/A	Edit Delete
2	Walker Curry Whilemina Rodgers 9867061292	कर्णाली प्रदेश	Choose District	Choose Municipality	Laboriosam obcaecat-30	Reiciendis lorem tem	AB+ Latest Blood donate Date : N/A	Edit Delete

Showing 1 to 3 of 3 entries
Previous
1
Next

## View and verify blood request:

Rakt Seva
Dashboard
Manage Blood
Manage Donor
Blood Request
Blood Banks
Audrey Mercer (Administrator)

Requested Blood

Patient Name	Requested Blood	Request Date & Time	Hospital Name
No any request yet			

## Manage blood unit:

Rakt Seva

DashboardManage BloodManage DonorBlood RequestBlood BanksAudrey Mercer (Administrator)

List of Donor

Add Blood Unit

Show 10 entries

Search:

BLOOD GROUP	TOTAL STOCK BLOOD		
A+	0 Unit	Issue Blood	Request Blood from Donor (2)
A-	0 Unit	Issue Blood	Request Blood from Donor (0)
B-	0 Unit	Issue Blood	Request Blood from Donor (0)
AB+	0 Unit	Issue Blood	Request Blood from Donor (1)
B+	0 Unit	Issue Blood	Request Blood from Donor (0)
O+	0 Unit	Issue Blood	Request Blood from Donor (1)

Activate Windows