Project Title

Blood Bank

Submitted By

Md. Rejoan Billah
152-35-1136

Supervised By

Nazia Nishat
Senior Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

A project submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Spring– 2019

DECLARATION

I hereby declare that I have taken this project under the supervision of **Nazia Nishat, Senior Lecturer, Department of Software Engineering, Daffodil International University.** I also declare that neither this project nor any part of this has been submitted elsewhere for award of any degree.

Md. Rejoan Billah
ID: 152-35-1136
Batch: 17th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:

Nazia Nishat

Senior Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Nazia Nishat mam, for providing me an opportunity to do the project work in Php and giving me all support and guidance, which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance. I owe my deep gratitude to my project guide Nazia Nishat mam, who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Software Engineering which helped me in successfully completing our project work.

Lastly, I am pleased to my lovely father and mother who encourage me to achieve this goal.

Dedication

I would like to dedicate this project to my respectable Father and Mother, my instructor, my Honorable teachers who are always dear and near to me and without whose patience, care, understanding, unsparing support, affection and most of all deepest love it would not have been possible to come up to this position.

Abstract

This project is aimed to developing an online Blood Donation Information. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The project entitled “Blood Bank” is a web application. It is developed using PHP and MySQL. This project has tried to incorporate all the advanced features of PHP such as to fulfill the aim of the project. The application has been designed such that it encompasses the managerial as well as public functionalities. Among the managerial functionalities are the task performed by the various offices. The public features are exercised by the users and outsiders according to the permissions level availed to them.

Contents

List of Figures

Chapter 1

Introduction

1.1. Project Overview

About one million babies are born with thalassemia every year in the world. The patient is given blood after a certain period of time, but the Guardian used to make arrangements. So, we want to build a digital blood bank through which the blood donation is very easy to give or to keep a list on the database. Those who give blood and those who need blood, their trouble will be reduced considerably. The BLOOD BANK is great project. The basic building aim is to provide blood donation service to the city recently. Blood Bank is a browser-based system that is designed to store, process, retrieve. This project aims at maintaining all the information pertaining to blood donors, and help them manage in a better way. My 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. In my project, anyone can search for blood. Search results will contain donor's availability status that if he/she is available or not. Anyone can send request for blood. When a user sends a request then all registered users will receive an email containing all information provided in send request form. Anyone can view requests for blood. Registered user can update their blood donated info (where they donated blood, how many bags and donated date). Registered User can edit their donation info if they make any mistake in donated info. Registered user can add other registered user so they can find them easily and know their status.

1.2. Project Purpose

1.2.1. Background

- Lack of rare blood group.
- Unavailability of blood during emergency.
- Less awareness among people about blood donation and blood transfusion.
- Deaths due to lack of blood during operations.

1.2.2. Benefits & Beneficiaries

This project will beneficial for

- The search functionality allows for blood group and location-based search. It means a user can search for any data available in his/her area.
- Users who are not interested to register he/she also can request for blood in Send Request menu.
- Also, users who are not interested to register he/she can view request for blood in View Request menu so that he/she can response for donating blood.
- When a registered user enters his donated info then the system will auto generate that user's status that if he/she is available or not.
- There will be an admin who can update or delete any user data. It is necessary because those users who are not much caring will be removed from the system.
- Users can add other donors to their Favorites list for further need. So that he/she will find their previous donators and their status that they are available or not.
- Users will fill up the donation details so that they can monitor their donating info.
- Users can view or update their donation details form Donation List menu.

1.2.3. Goals

Our main goal is to make this system available all over the country, because most of the peoples are using internet now so that everyone can use it and get benefit from this system. When people search from a blood bank for blood he/she face a lot of problems that he found too many matching contacts but nobody is available right now, our goal is to reduce this hassle so in our system when a user search for blood system will show the donors status that if he/she available or not right now.

1.3. Stakeholders

There are two types of stakeholders in our "Blood Bank". Such as:

- Users
- Admin

Now, I will write a brief description about stakeholders

Users: All donors also are users of the system. Because if anyone don't want to register, he/she can search or send request for blood also view requests from view request section. When a user will register to system then he/she will be a donor. Then he/she can update his blood donating info and also can update their profile info.

Admin: Admin can update donors profile info and also their donating info.

1.4. Proposed System Model (block diagram)

I prepared a system model. This model will clarify my proposed system in brief.

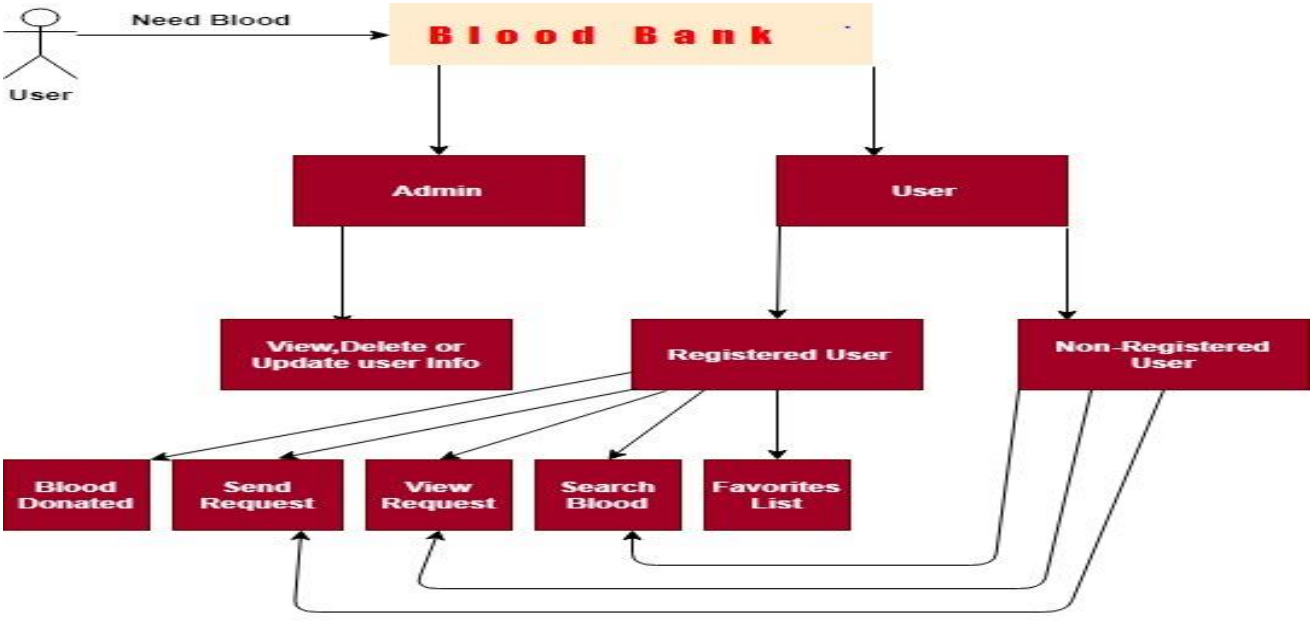

Figure 1.4: Proposed System Model

1.5. Project Schedule

I must prepare schedule for my project to complete it on time.

1.5.1. Gantt chart

A Gantt chart is a visual view of tasks scheduled over time. Gantt charts are used for planning projects of all sizes and they are a useful way of showing what work is scheduled to be done on a specific day. They also help me view the start and end dates of a project in one simple view.

Activities		W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15	W 16
Planning	Ideas																
	Problem definition																
	Proposal planning																
Requirements	Requirement specification																
	Requirement analysis																
QA - 1	Quality assurance																
System design	Design Specification																
	Interface design																
	Database design																
Development	Development System modules																
	Integrate System Modules																
QA - 2	Test cases																
Testing	Unit testing																
	Blackbox testing																
Resolve Issues	Resolve issues found																
Release	Software release																

Figure 1.5.1: Gantt Chart

1.5.2. Release Plan/Milestone

The release plan or milestones are given below:

Activities	Duration in week	Total week
Brainstorming	Week 1	1
Problem identification	Week 1, Week 2	2
Requirement specification	Week 2	1
Requirement analysis	Week 2	1
Design specification	Week 4	1
Interface design	Week 4	1
Database design	Week 5	1
Development System modules	Week 5, Week 6, Week 7, Week 8	4
Integrate System Modules	Week 6, Week 7, Week 8	3
Test case	Week 2, Week 7, Week 8, Week 9, Week 10	5
Unit testing	Week 11, Week 12	2
Black-box testing	Week 13, Week 14, Week 15	3
Resolve Issues	Week 15	1
Software release	Week 16	1

Chapter 2

Software Requirement Specification

2.1. Functional Requirements

A functional requirement is that it essentially specifies something the system should do. Typically, functional requirements will specify a behavior or function.

The Functional Requirements Specification describes what the system must do; how the system does it is described in the Design Specification. If a User Requirement Specification was written, all requirements outlined in the User Requirement Specification should be addressed in the Functional Requirements Specification.

2.1.1. Registration of Donor

Requirements 1	**Registration of Donor**
Description	Requires an interface for the registration the details of the donor. Attributes - Donor ID, Name, Age, Address, Contact, Email, Password, Blood Group, Gender, Date
Stakeholders	**User**

2.1.2. Log in

Requirements 2	**Only registers user can Log In**
Description	Requires registered user's email and password
Stakeholders	**User**

2.1.3. Send Request

Requirements 3	**All users can Send Request**
Description	Registered and Non registered User can send request for blood and all registered donors will receive a notification about it.
Stakeholders	**User**

2.1.4. View Request

Requirements 4	**All users can View Request**
Description	Registered and Non registered User can check their view request section for requested blood
Stakeholders	**User**

2.1.5. Search For blood

Requirements 5	**All users can Search For blood**
Description	Anybody Can use the search section
Stakeholders	**User**

2.1.6. Blood Donated

Requirements 6	**Blood Donated**
Description	Registered Users need to update their blood donated information in blood donated section also admin can update this section
Stakeholders	**User**

2.1.7. View Donations

Requirements 7	**View Donations**
Description	Registered Users can check their donation history
Stakeholders	**User**

2.1.8. Favorites List

Requirements 8	**Favorites List**
Description	Registered Users can add registered donor in the list for further need
Stakeholders	**User**

2.1.9. Update Profile

Requirements 9	**Update Profile**
Description	Registered Users can update their profile
Stakeholders	**User**

2.1.10. Change Password

Requirements 10	**Change Password**
Description	Registered Users can change their password
Stakeholders	**User**

2.1.11. Contact with Admin

Requirements 11	**Contact with Admin**
Description	Anyone can send message to admin
Stakeholders	**User**

2.1.12. Log Out

Requirements 12	**Log Out**
Description	Registered User can log out anytime they want.
Stakeholders	**User**

2.2. Data Requirements

Data requirements establishes the process used to identify, prioritize, precisely formulate, and validate the data needed to achieve objectives. For my project all data will be provided from users so we need to focus on some points. For example:

- Data source sequence
- Types of entity of the system
- Availability of data

2.3. Performance Requirements

Every system needs performance requirements. Because performance requirements prescribe how well the system performs certain functions under specific conditions. Now, I will clarify some viewpoint by which we are going to improve the execution of our system

2.3.1. Speed and Latency Requirements

Speed and latency requirements must be free from doubts while retrieving data from the server.

SLR-1	Search result must be faster
Description	When a user search for blood, then the search result must show within seconds.
Stakeholders	Users

2.3.2. Precision or Accuracy Requirements

Search Results that is to be shown to the user is need to be accurate. Because, wrong information might be cause trouble for the people who need blood.

PAR-1	Search result must be accurate
Description	When a user search for blood, then the search result must be accurate to the search value.
Stakeholders	Users

2.3.3. Capacity Requirements

The developed system must be capable to handle all user data.

CR-1	The system will handle lots of data.
Description	The system needs to handle uploaded data.
Stakeholders	Users

2.4. Dependability Requirements

The term dependability is measured based on four volume. Such as:

- Availability
- Reliability
- Safety
- Security

If we want to say that our system is dependable then it must fulfill the four volumes.

2.4.1. Reliability Requirements

Reliability Requirements is usually defined as the probability that a product will operate without failure.

RAR-1	The system must be working with all required functions
Description	• Our system must be working all the time • The system must be updated regularly
Stakeholders	Users

2.4.2. Availability Requirements

Availability Requirements is defined the availability of the system.

RAR-1	The system must be available on 24 * 7
Description	• Our system must be available 24*7
Stakeholders	Users

2.4.3. Robustness or Fault-Tolerance Requirements

Robustness or Fault-Tolerance Requirements defines that there will be no crush in a system. Now I will show my system's Robustness or Fault-Tolerance Requirements.

RFT-1	The system handles all user access without system errors
Description	Huge amounts of users might hit my system at a time. All their requests must be handled without any fault.
Stakeholders	N/A

2.4.4. Safety-Critical Requirements

There are no safety-critical requirements in my system.

2.5. Maintainability and Supportability Requirements

It is very important to maintain and give 24/7 support to a system after release.

2.5.1. Maintenance Requirements

MR-1	**System helps to update user profile also admins will be there for any solution.**
Description	It is very important to update user profile.
Stakeholders	Users

2.5.2. Supportability Requirements

Supportability essentials probably related to some expands. Such as,

- Testability
- Extensibility
- Adaptability
- Maintainability

I think my system meets all of the above requirements related to supportability.

2.5.3. Adaptability Requirements

This system has the adaptability to use in any environments.

2.5.4. Scalability or Extensibility Requirements

Ability of a computer system to accommodate additions to its capacity or capabilities. From a software point of view, it may include ability to support more network users, greater number of 'hits' from website visitors.

This system has extensibility requirements, such as: Server extend

2.6. Security Requirements

Software security requirements should be its functional requirement. Functionality related to software security can either be directly tested or observed. Some security related requirements are given below:

- Signing in to system
- Get access according to logged in user
- Signing out

While accessing to the system, each and every module must provide authentication mechanism.

2.6.1. Integrity Requirements

There are three Key Requirements to Achieve Data Integrity, Such as:

- **Completeness:** A data record, such as all user data will be recorded.
- **Accuracy:** Wrong or misleading data helps no one. The cause of inaccuracy can be due to manual input errors.
- **Consistency:** This is one of the harder data integrity issues to resolve. If you only have a single source of data, then it is likely to be consistent (although potentially consistently wrong). To truly verify the data, it must be validated against multiple sources. But in my project, I have only single source of data.

So, my project meets the maximum requirements from all above the requirements.

1.1.1. Style Requirements

It is necessary to use stylesheet to the system. In this system I used some bootstrap design.

SR-1	The appearance must be controllable using CSS file
Description	For php system stylesheet file are CSS.
Stakeholders	Software developer.

1.1.2. Privacy Requirements

In every system there should be exist privacy requirements. To ensure privacy of user's data, users are allowed to gain admittance to that information which are being related by them which can be guaranteed by the users sign in to system.

1.2. Usability and Human-Interaction Requirements

My main vision is to make my system user friendly to user and easier to use.

1.2.1. Ease of Use Requirements

EUR-1	**System must be understandable for the users**
Description	This system is easy to use and anyone will understand by which they can operate the system.
Stakeholders	Users

1.2.2. Personalization and Internationalization Requirements

This system is only for use between Dhaka city. The main goal of personalization is to deliver content and functionality that matches specific user needs or interests. I set up the system to identify users and deliver to them the content, experience, or functionality that matches their role.

1.2.3. Understandability and Politeness Requirements

I already declare that this system is easy to use and anyone will understand who can internet browsing.

1.2.4. Accessibility Requirements

This system has some accessibility requirements, such as: non registered user will not get access to the functions which functions are for registered users. Also registered will not get access to admin section.

1.2.5. User Documentation Requirements

User Documentation is prepared to make development life cycle easier for the system engineers or system analysts.

UDR-1	**The system engineer documentation.**
Description	To develop this system first of all I need to make a clear user documentation.
Stakeholders	Software developers.

1.2.6. Training Requirements

It is very necessary to properly train up end users to the system so that they would be capable to operate easily. After release the system we need to appoint some volunteers, they will be trained up the end users to use the system easily.

1.3. Look and Feel Requirements

Look and Feel Requirements mainly refer how the system will look like, how the system's interface will be for the end users.

1.3.1. Appearance Requirements

Registered and non-registered users must know which input fields are required and which are not. For that reason, I will use required labels for all input fields.

AR-1	There should be must require labels
Description	The mandatory field's label must be required labels sign
Stakeholders	Users

1.4. Operational and Environmental Requirements

Operational and environmental requirement refers to the capabilities, performance measurements, process, measurements of effectiveness, measurements of performance, measures of sustainability, measurements of technical performances etc.

1.4.1. Requirements for Interfacing with Adjacent Systems

This system can be interfacing with another adjacent system.

1.5. Legal Requirements

It refers to the terms and conditions or privacy policy of any organizations. But this system is not under any organization so right now there will be no legal requirements.

Chapter 3

System Analysis

3.1. Use Case Diagram

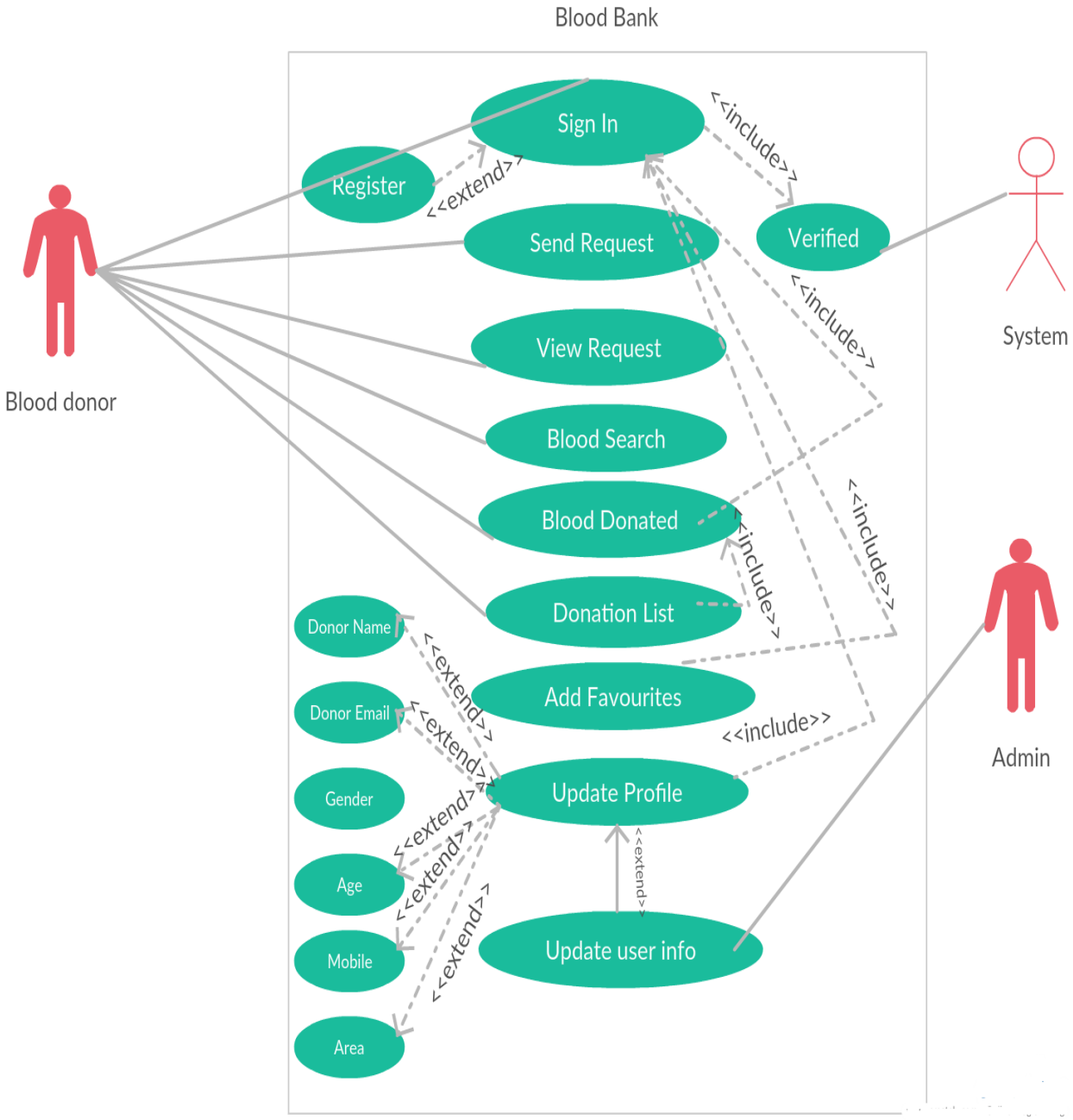

Figure 3.1: Use Case diagram

3.2. Use Case Description (for each use case)

3.2.1. Sign In

Use Case Title	**Sign in**
Goal	Register user can log in to system
Preconditions	User must be authenticated.
Success End Condition	Users can see their facilities
Failure End Condition	Users can't delete any data
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	User's data will be visible to users after providing related information properly
Alternative Flows	N/A
Quality Requirements	N/A

3.2.2. Send Request

Use Case Title	**Send Request**
Goal	Register and non-register users can send request for blood
Preconditions	Authentication isn't necessary
Success End Condition	Users can send request
Failure End Condition	Users can't delete any data
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	User's will be available to see requests
Alternative Flows	N/A
Quality Requirements	N/A

3.2.3. View Request

Use Case Title	**View Request**
Goal	Register and non-register users can view request for blood
Preconditions	Authentication isn't necessary
Success End Condition	Users can view request
Failure End Condition	Users can't delete any data
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	User's will be available to see requests
Alternative Flows	N/A
Quality Requirements	N/A

3.2.4. Blood Search

Use Case Title	**Blood Search**
Goal	Register and non-register users can view search for blood
Preconditions	Authentication isn't necessary
Success End Condition	Users can search their desired blood group
Failure End Condition	Users can't delete or update donor's data
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	Users can operate search operation
Alternative Flows	N/A
Quality Requirements	N/A

3.2.5. Blood Donated

Use Case Title	**Blood Donated**
Goal	Only Register users can update their donating info
Preconditions	User must be authenticated.
Success End Condition	Users can update their blood donated information
Failure End Condition	Users can't delete their donated info
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	User's will be available update blood donated info
Alternative Flows	N/A
Quality Requirements	N/A

3.2.6. Donation list

Use Case Title	**Donation List**
Goal	Only registered users can see donation list
Preconditions	User must be authenticated.
Success End Condition	Users can see donation list
Failure End Condition	Users can't delete donation data
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	User's will be available to see donation list and update list
Alternative Flows	N/A
Quality Requirements	N/A

3.2.7. Add Favorites

Use Case Title	**Add Favorites**
Goal	Only registered users can add others donor to their favorites list
Preconditions	User must be authenticated.
Success End Condition	Users can add donors
Failure End Condition	Users can't delete any donors from favorites list
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	Each user can be available to see their favorites list
Alternative Flows	N/A
Quality Requirements	N/A

3.2.8. Update Profile

Use Case Title	**Update Profile**
Goal	Only registered users can update their profile
Preconditions	User must be authenticated.
Success End Condition	Users can view and update their profile
Failure End Condition	Users can't delete any data from his profile
Primary Actors: **Secondary Actors:**	Blood Donor
Trigger	Data come to the user
Description / Main Success Scenario	User's will be available to see their profile and update data
Alternative Flows	N/A
Quality Requirements	N/A

3.2.9. Update User Info

Use Case Title	**Update User Info**
Goal	Only admins can update all user info
Preconditions	Admin must be authenticated.
Success End Condition	Admin can update any data on any user's profile
Failure End Condition	Admin can't see or change any data from their own profile
Primary Actors: **Secondary Actors:**	Admin
Trigger	Data come to the admin
Description / Main Success Scenario	Admin will be available to see users all data and can update anything
Alternative Flows	N/A
Quality Requirements	N/A

3.3. Activity Diagram (for each use case)

3.3.1. Sign In

Users provided must be match with database to sign in.

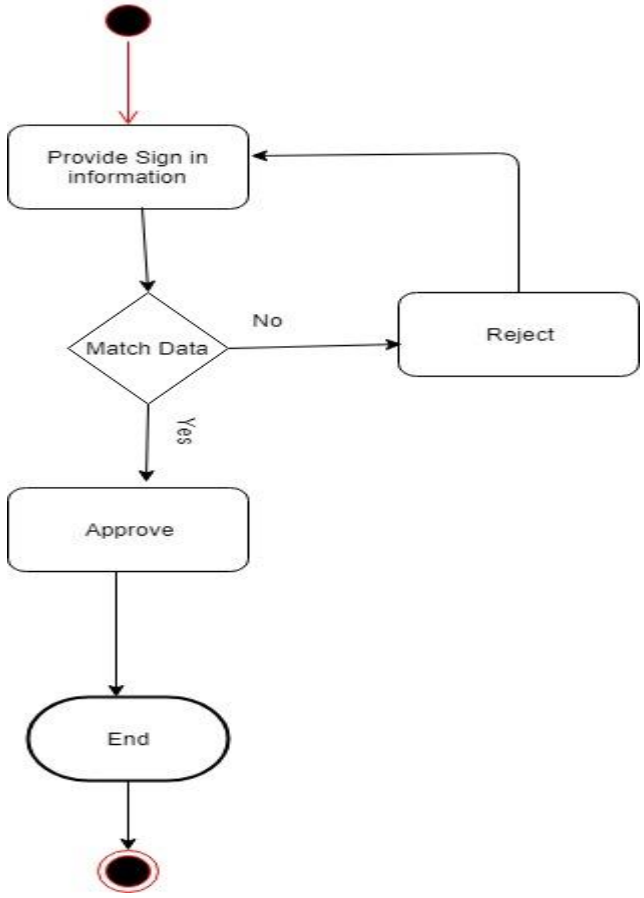

Figure 3.3.1: Activity Diagram for Sign in

3.3.2. Send Request

Users need to provide all information to request for blood.

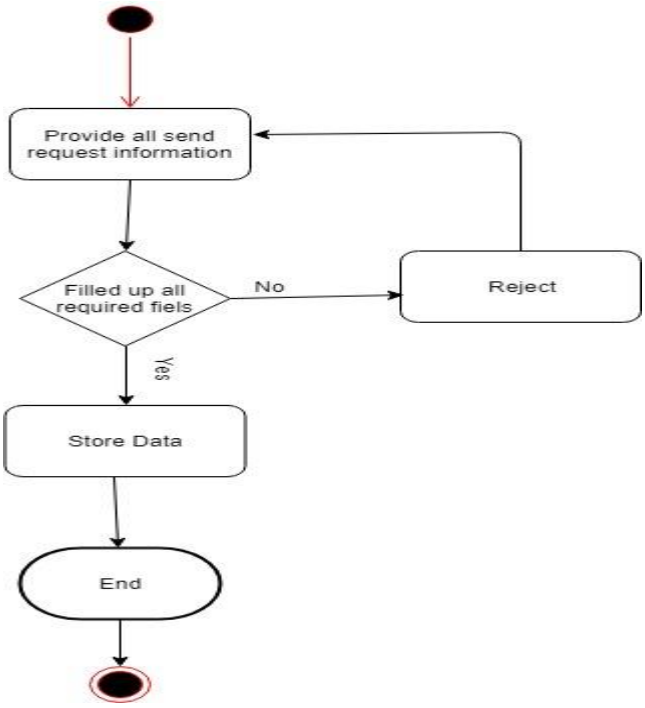

Figure 3.3.2: Activity Diagram for Send request

3.3.3. View Request

All users can view requests from view request section.

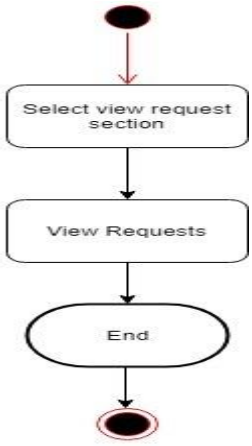

Figure 3.3.3: Activity Diagram for view request

3.3.4. Blood Search

All users can search if the search by providing correct search value

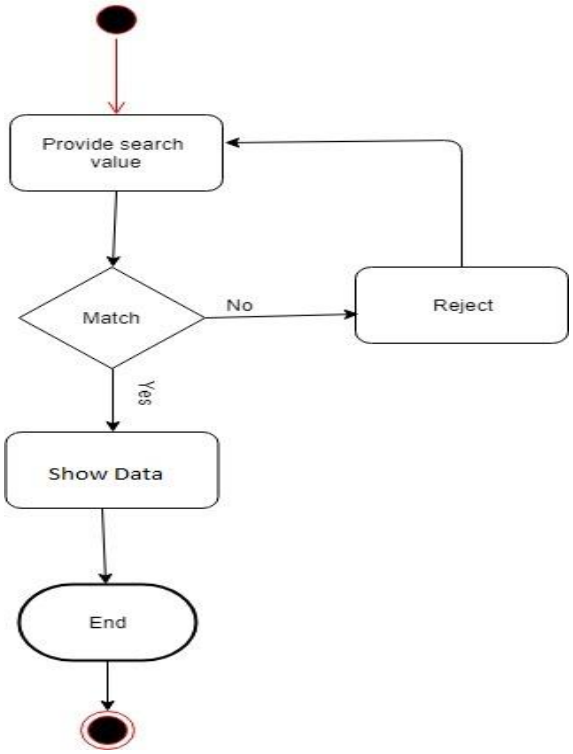

Figure 3.3.4: Activity Diagram for blood search

3.3.5. Blood Donated

Registered users can store blood donated information.

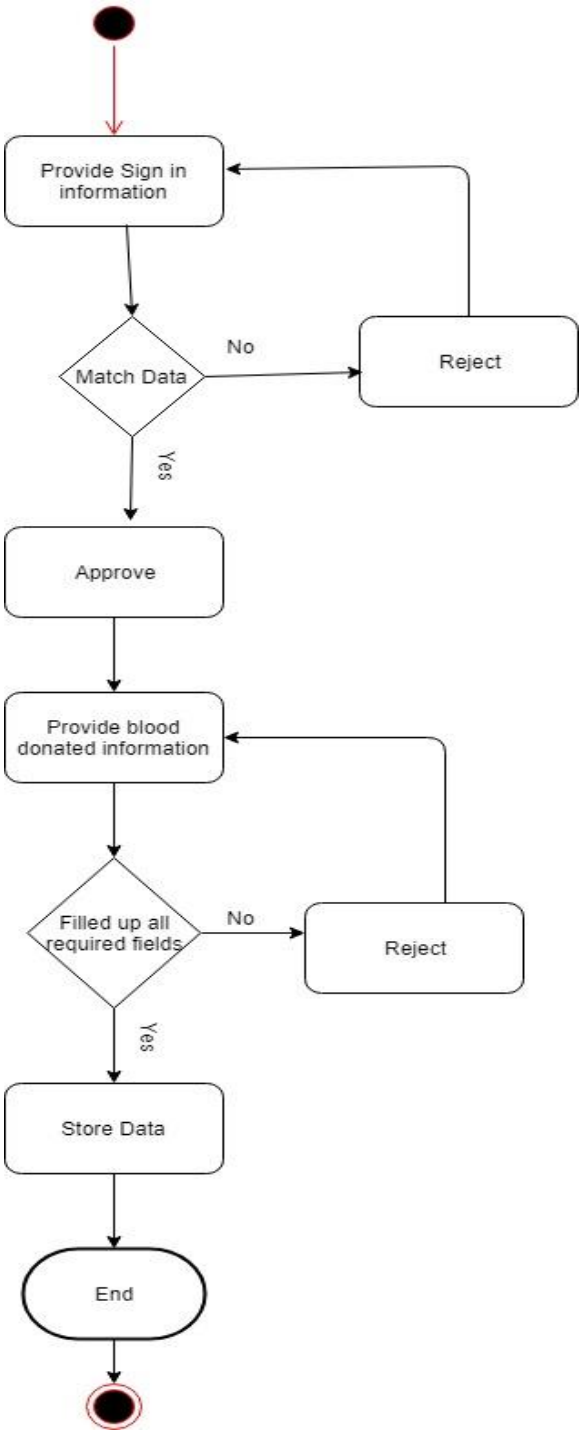

Figure 3.3.5: Activity diagram for blood donated

3.3.6. View Donations

Registered user can view their donation history

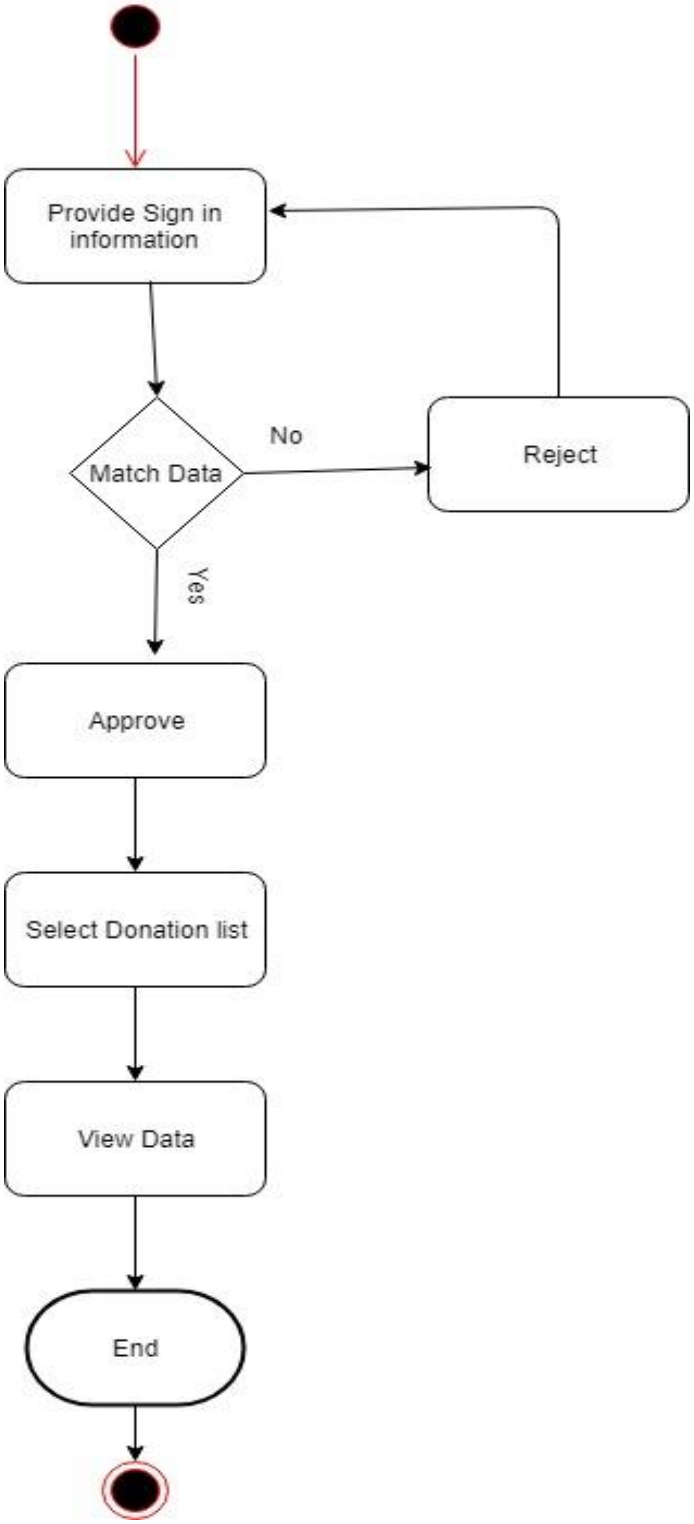

Figure 3.3.6: Activity diagram for view donations

3.3.7. Favorites list

Registered users can add other donors to favorites list.

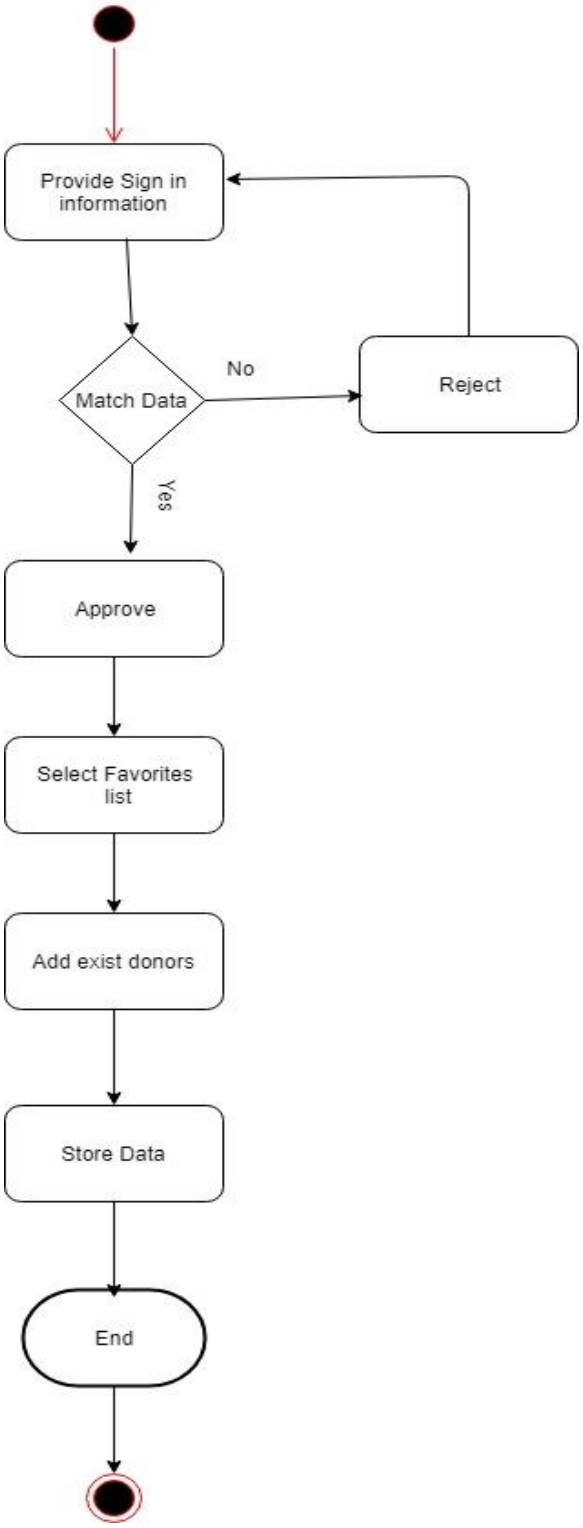

Figure 3.3.7: Activity diagram for favorites list

3.3.8. Update Profile

Registered users can update their profile

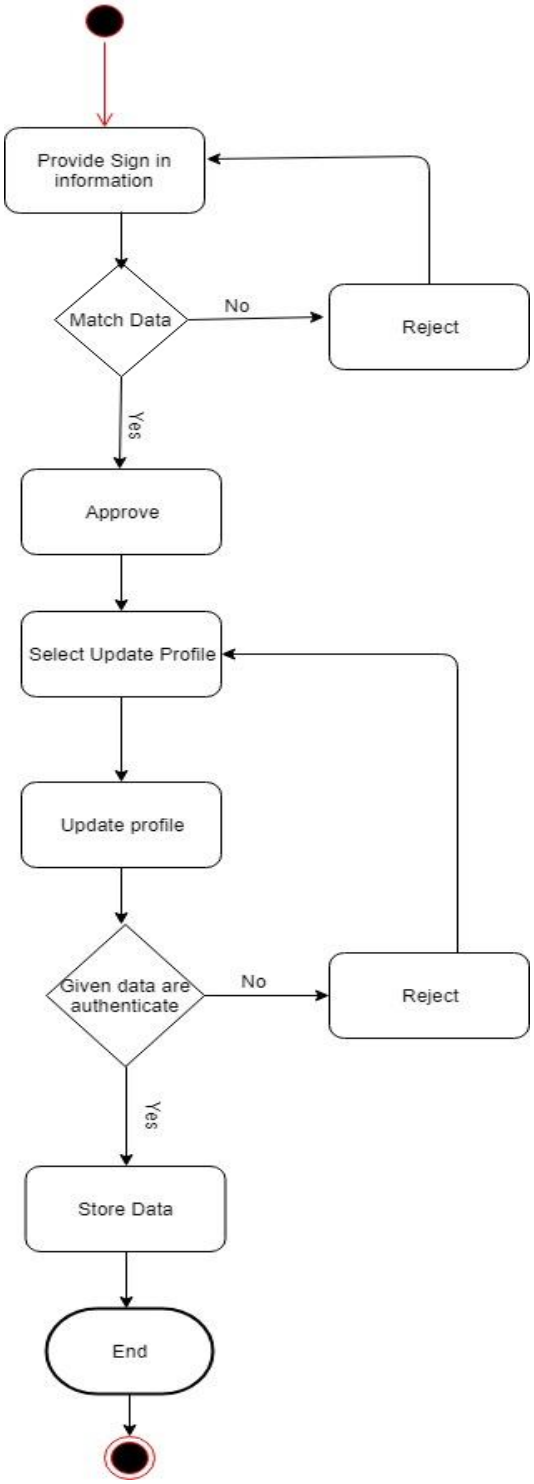

Figure 3.3.8: Activity diagram for update profile

3.3.9. User Update

Admins can update all user data.

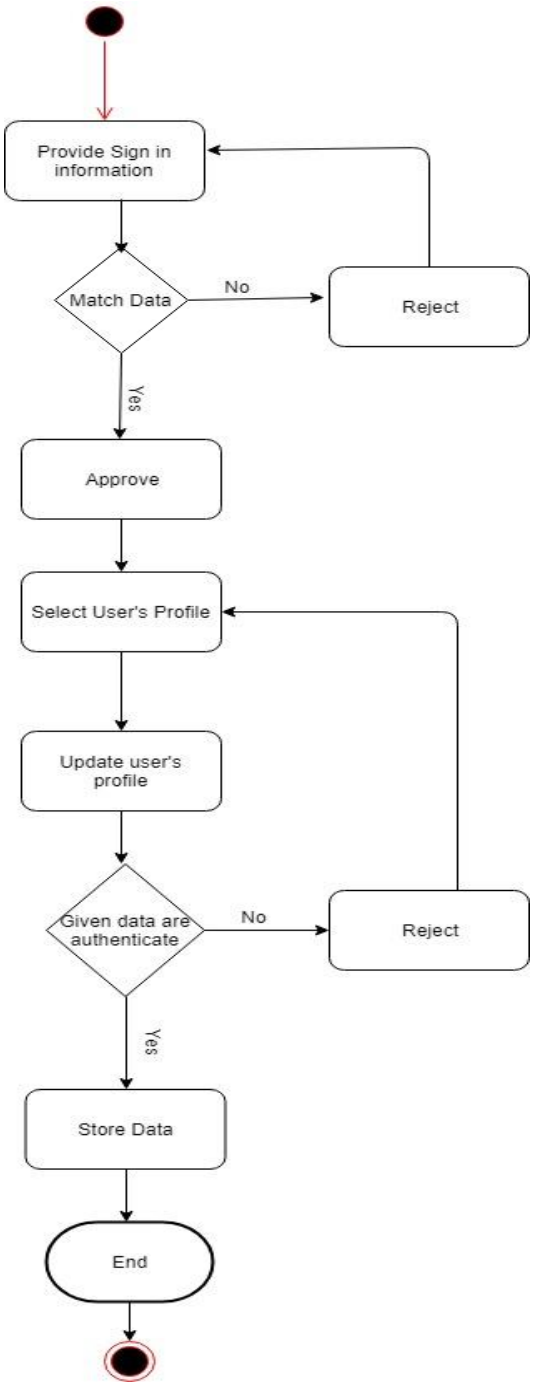

Figure 12: Activity diagram for user update

3.4. System Sequence Diagram (for each use case)

3.4.1. Sign in

Only authenticate users can sign in to system

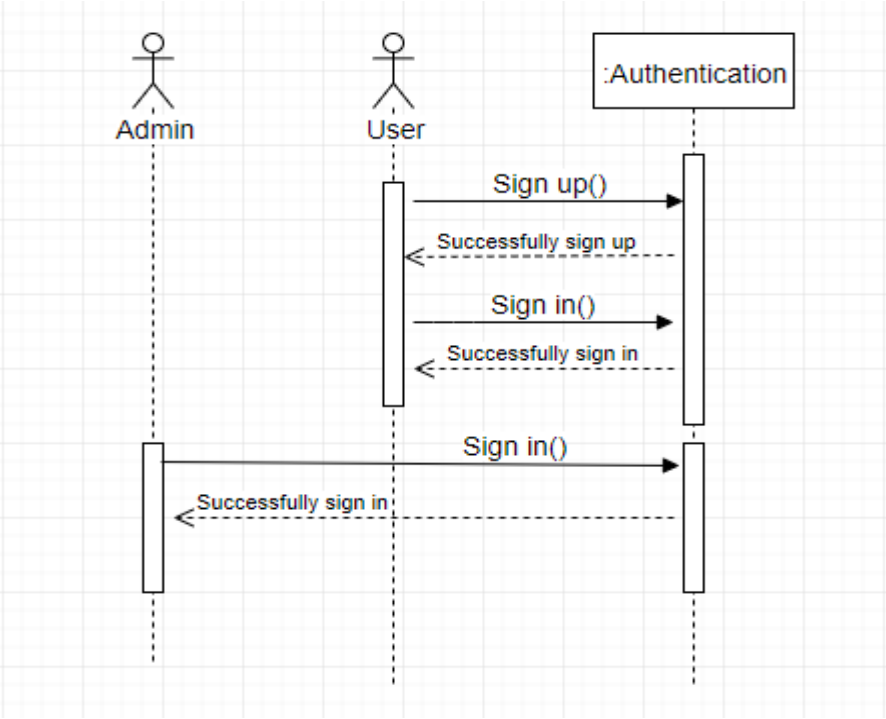

Figure 3.4.1: Sequence Diagram for sign in

3.4.2. Send Request

All users can send request for blood

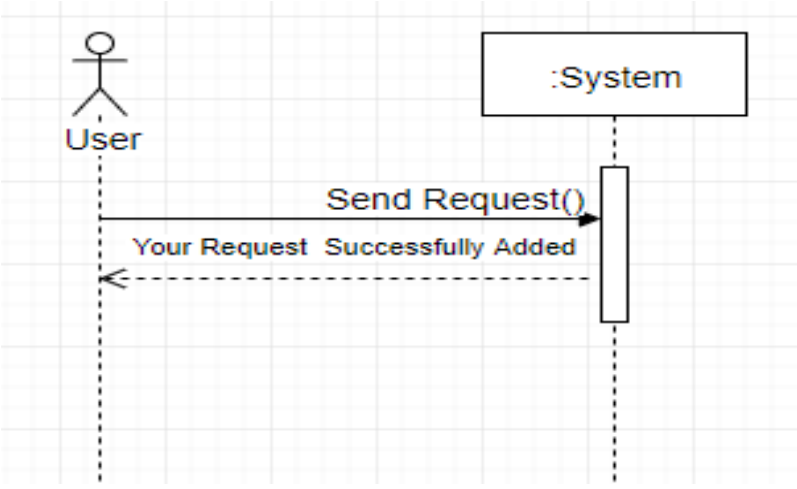

Figure 3.4.2: Sequence Diagram for send request

3.4.3. View Request

All users from system can view request

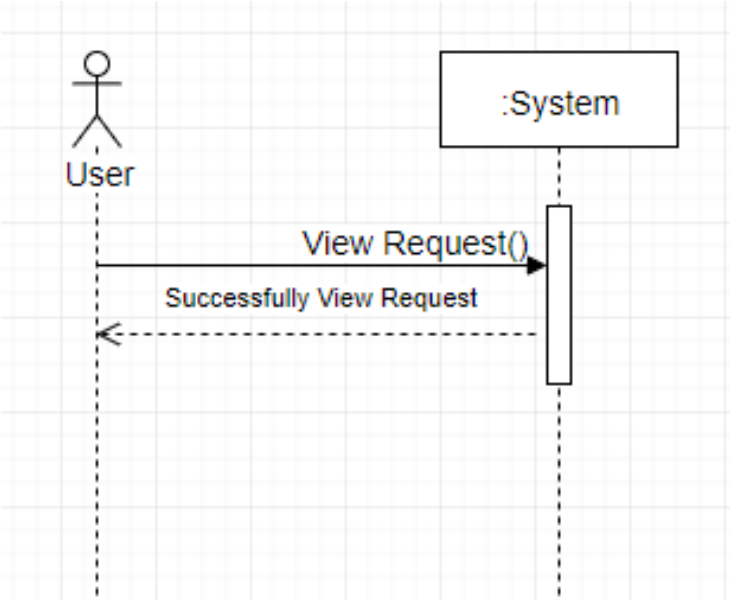

Figure 3.4.3: Sequence Diagram for view request

3.4.4. Blood Donated

Only registered users can update blood donated information

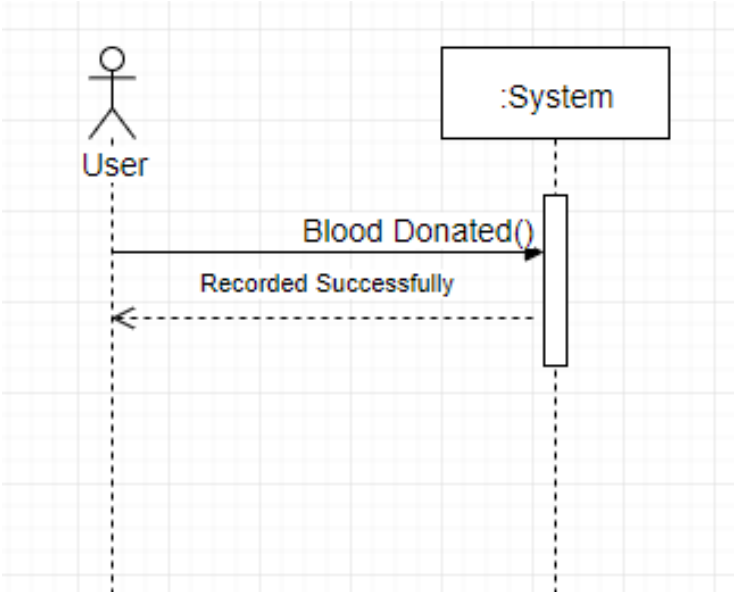

Figure 3.4.4: Sequence Diagram for blood donated

3.4.5. Donation list

Only registered users and admins from system can see donation list

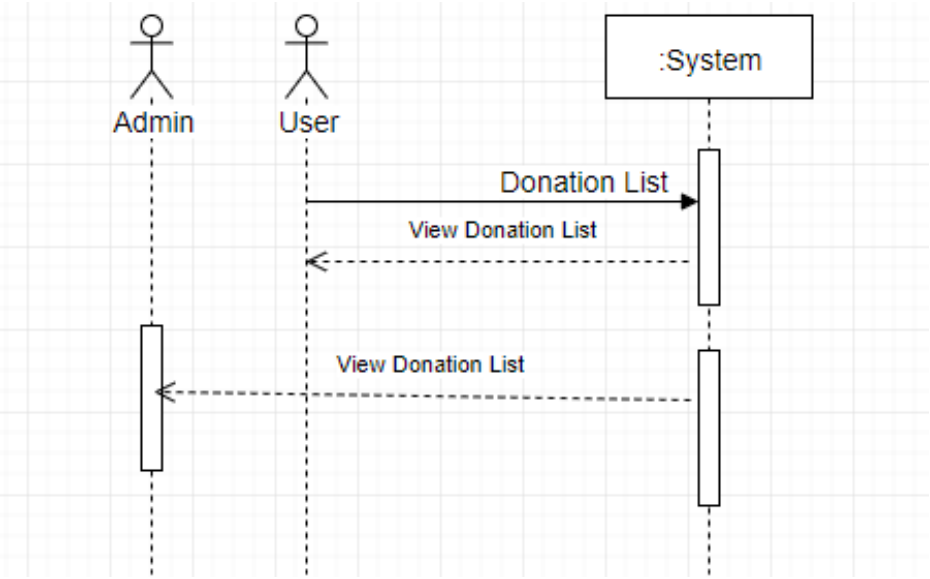

Figure 3.4.5: Sequence Diagram for donation list

3.4.6. Favorites list

Only registered users and admins from system can see favorites list.

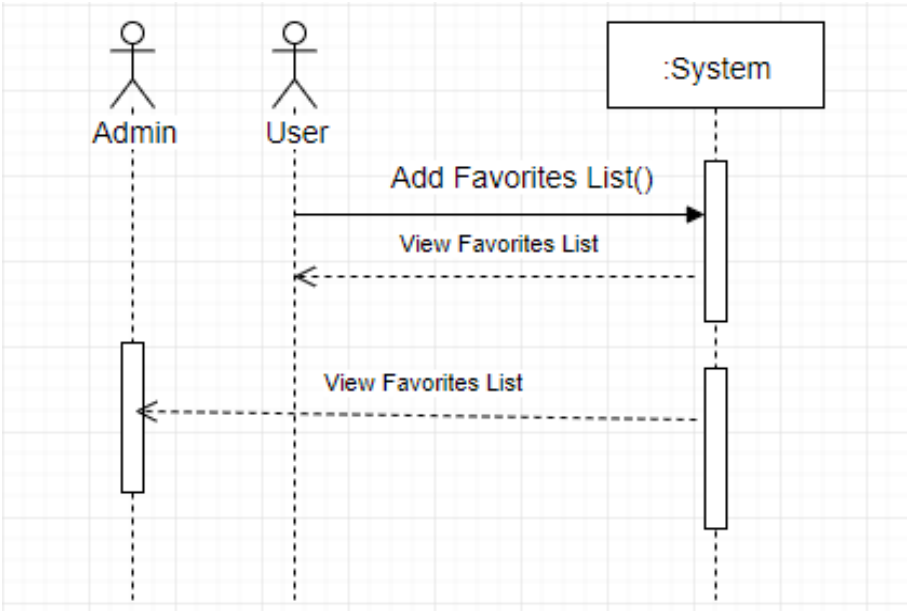

Figure 3.4.6: Sequence Diagram for favorites list

3.4.7. Update Profile

Only registered users can update their profile. And also, admins from system can update any user profile anytime.

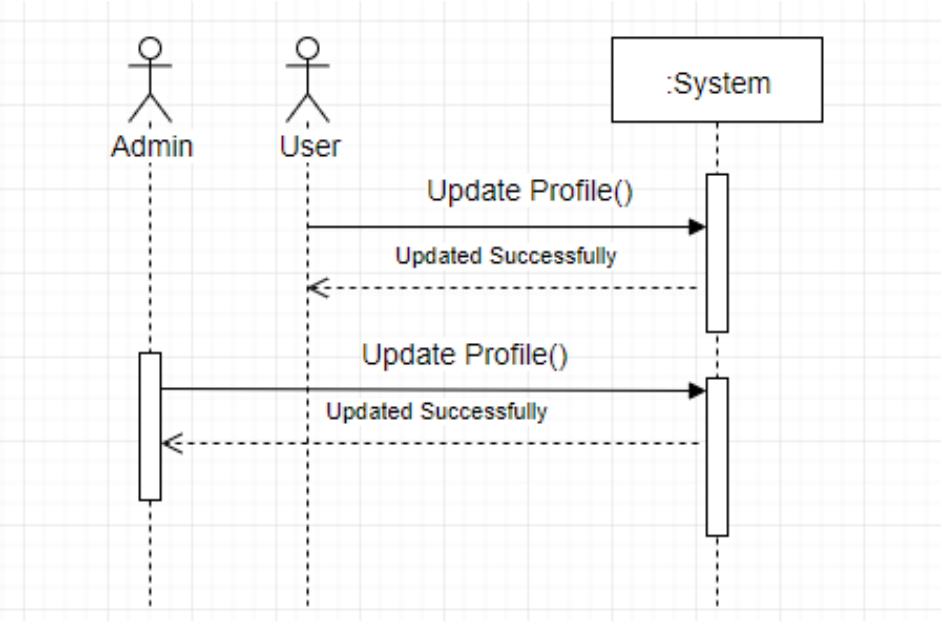

Figure 3.4.7: Sequence Diagram for update profile

3.4.8. Sign Out

Any signed in user can sign out anytime.

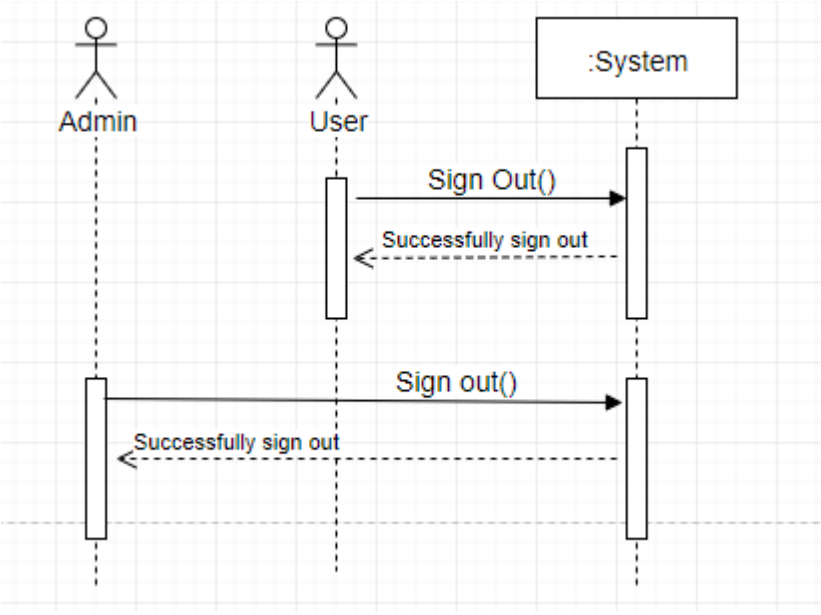

Figure 3.4.8: Sequence Diagram for sign out

Chapter 4

System Design Specification

4.1. Class Responsibilities Collaboration (CRC) Cards

4.1.1. User card

Users	
• Users can Register • Users can sign in • Users can search blood • Users can send request • Users can view request • Users can add favorites • Create Users profile	• favorites • donation Info • area • bloodgroup • blood request

4.1.2. Bloodgroup card

bloodgroup	
• blood group name	• users

4.1.3. Blood request card

blood request	
• send request • view request	• users

4.1.4. Area card

area	
• area name	• users

4.1.5. Donation info card

donation info	
• view donation info	• users

4.1.6. Favorites card

favorites	
• add users to favorites • view donation list	• users

4.2. Class Diagram

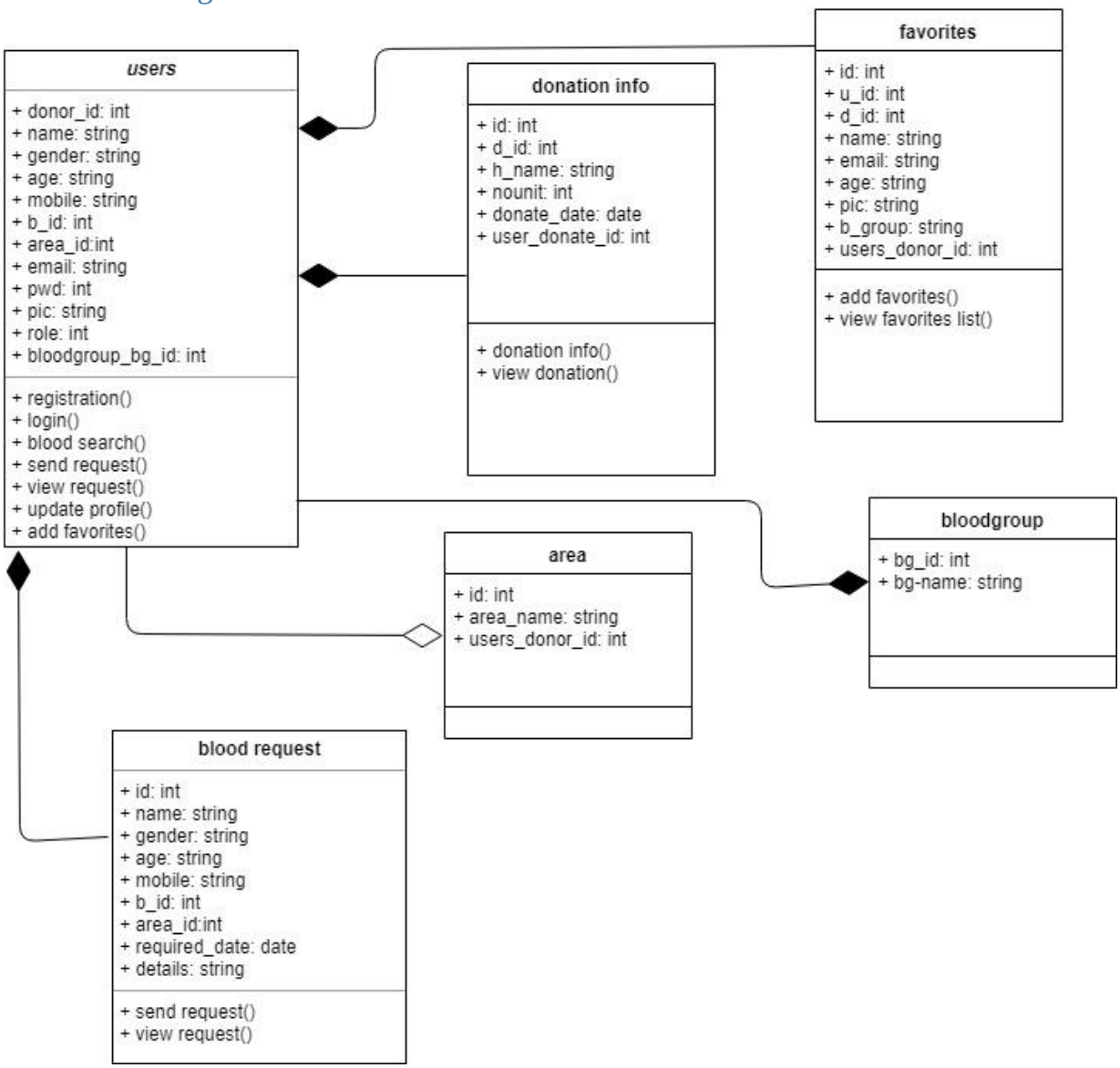

Figure 4.2: Class Diagram

4.3. Database Design Diagram

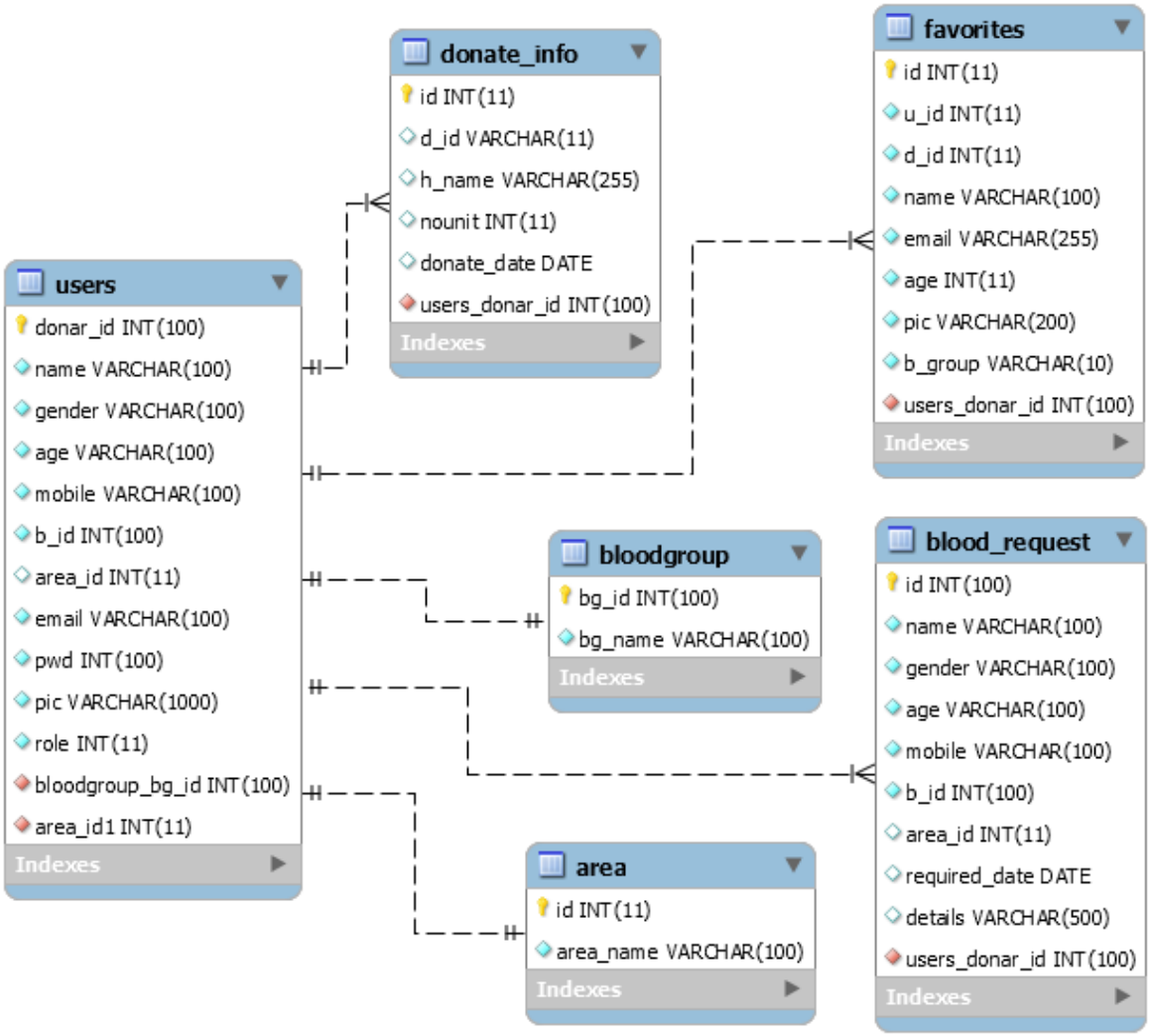

Figure 4.3: Database Design Diagram

4.4. Development Tools & Technology

4.4.1. User Interface Technology

4.4.1.1. Framework

In my project I didn't use any framework. In my project I have used server-side programming language named PHP (Personal Home Page). This is a scripting language.

4.4.1.2. jQuery UI

jQuery UI is a curated set of UI cooperation, effects, gadgets, and themes based over the jQuery JavaScript Library. I used jQuery in my project because jQuery makes easier to understand and work with. It also provides AJAX functionality which I used for search function.

4.4.1.3. CSS Framework

Bootstrap is an open source toolkit for developing with HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JS (Java Script). Quickly prototype my ideas. Bootstrap provides a set of some files which contains stylesheets which gives basic definition. Bootstrap also provides some JavaScript components also. There are some built in components like jQuery UI. By using Bootstrap framework, I get both CSS and JavaScript facilities with a single platform.

4.4.1.4. User Interface Technology

The significance of User interface is high. For accomplishment of any project, a gorgeous UI assumes an imperative job. User interface includes using good image, graphics, stylesheets, scripting etc. I used bootstrap templates for my project.

4.4.2. Implementation Tools & Platforms

There are some tools and technologies that need to be used for developing software. It is very important to determine which tools and platforms are the best match of my requirements. After making a proper decision, I am using php that is a server-side scripting language

4.4.2.1. Integrated Development Environment

There is a lot of PHP editors available that are maintained on Windows, Linux, and Mac and are obtainable for free download. For my project I used Subline text editor. It is an accepted and a competing tool for the PHP editors. It is lightweight with required feature and is supported on OSX, Windows as well as Linux. The Sublime text editor is put up to gain its powers through different plugins and packages.

4.4.2.2. Database Server

I used MySQL server. It is a free to use. It can be used on cross platform. It supports a wide range of features and I have worked with MySQL server before. It is also very easy to use. So, working with this server will be easier to me.

Chapter 5

System Testing

5.1. Testing Features

Highlight testing can be considered as making change to add or modify the new functionality to the existing project. To test the highlights and usefulness, another test set is to be composed for testing reason. Pretty much every component and usefulness have diverse attributes. Those are designed to make the application more useful, intuitive, reliable, secured, effective and efficient.

5.1.1. Features to be tested

Features	**Priority**	**Description**
Sign In	3	Users must be authenticated by system
Logout	1	Session must be destroyed after logout
Send Request	2	All required filled must be inserted properly
View request	2	No sign in required for view request
Blood donated	3	Users must be authenticated by system for store blood donated data.
Donation list	2	User must be authenticated by system for view their donation list
Favorites list	2	User must be authenticated by system for add others users to the favorites list.
Technological Features		
Database	1	Database will be used at almost every operation. So, there will be reliable connection for access.

Here, 1 = Low Priority, 2 = Medium Priority, 3 = High Priority

5.1.2. Features not to be tested

Here are some features that will not be tested in my project.

- Contact with admin: In my project there will be a section for contact with admin. I didn't test this feature yet.
- Data record time: I didn't test this feature yet.

5.2. Testing Strategies

5.2.1. Test Approach

There are mainly two test approach.

- **Automation testing:** Automation testing is a name of testing technique by which test engineers prepare some scripts according to test plan and after that they use suitable tools to perform testing of the software. Nowadays, almost every software company follow the approach of automation testing.
- **Manual testing:** Manual testing is also a name of technique of testing by searching out the bugs or vulnerability in an application. In this process, test engineers manually test and execute the test cases without having any automation tools.

5.2.1.1. Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle. This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

Now I will discuss 2 of the techniques involved in Black Box testing.

- **Equivalence Partitioning Testing:** Equivalence Partitioning also called as equivalence class partitioning. It is abbreviated as ECP. It is a software testing technique that divides the input test data of the application under test into each partition at least once of equivalent data from which test cases can be derived. An advantage of this approach is it reduces the time required for performing testing of a software due to a smaller number of test cases.
- **Boundary Value Analysis:** Boundary value analysis is a type of black box or specification-based testing technique in which tests are performed using the boundary values. Boundary values are validated against both the valid boundaries and invalid boundaries.

5.2.1.2. White Box Testing

White box testing is a testing technique, that examines the program structure and derives test data from the program logic. Here are some white box testing techniques:

- **Statement Coverage:** This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage:** This technique is running a series of tests to ensure that all branches are tested at least once.
- **Path Coverage:** This technique corresponds to testing all possible paths which means that each statement and branch is covered.

5.2.2. Pass/Fail Criteria

Pass/Fail criteria is process to check whether a software item has passed or failed its test.

Now I will give the Pass/Fail criteria below.

- System crash will not be considered as pass case.
- If any criteria pass 100% times, then it will be considered as pass criteria only.
- If data can't be displayed to the users properly, then it is also to be considered as fail criteria.

5.2.3. Suspension and Resumption

Suspension criteria specify the criteria to be used to suspend all or a portion of the testing activities while resumption criteria specify when testing can resume after it has been suspended.

- Unavailability of server during execution.

System Integration Testing in the Integration environment may be resumed under the following circumstances:

- When the server become available again.

5.2.4. Testing Schedule

Test Phase	**Time**
Testing plan create	1 week
Test specification	2 weeks
Unit testing	During development time
Component test	1 week
Test Phase	**Time**
Integration testing	1 week
Validating use cases	1 week
Testing user interfaces	1 week
Load testing	1 week
Performance testing	1 week
Release	1 week

5.2.5. Traceability Matrix

Project Manager			**Business Analyst Lead**		
QA Lead			**Target Implementation Date**		
BR#	**Category / Functionality / Activity**	**Requirement Description**	**Use Case Reference**	**Test Case Reference**	**Comments**
BR-1	Functional	Sign in	Use case 3.2.1	Test case 5.4.1	
BR-4	Functional	Blood Search	Use case 3.2.4	Test case 5.4.2	
BR-11	Non-functional	Reliability	N/A	N/A	

5.3. Testing Environment (hardware/software requirements)

For making the environment for testing, we need:

- Test data
- Database server
- Operating system
- Browser
- System and application
- Network
- User Manuals

5.4. Test Cases

A Test Case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

5.4.1. Sign In

Test case #1			**Test case name:** Sign in		
System: Blood BankBD			**Subsystem: N/A**		
Designed by: Md. Rejoan Billah			**Designed date:** 05-February-2019		
Executed by: Raj			**Executed date:** 05-April-2019		
Short description: The registered users need to login to the system. And before that system will check the authentication and authorization.					
Pre-conditions: • Users will be always redirected to the login page when they input wring information. • Assume that, the email is 'raj@gmail.com' and password is '123'					
Step	**Email**	**Password**	**Expected result**	**Pass/Fail**	**Comment**
1	raj@gmail.com	111	The Password is incorrect	Pass	
2	raj@gmail.com		The Password is incorrect	Pass	
3	raj@gmail.com	123	Successfully login to the application and redirected to the dashboard	Pass	
Post-conditions: Users and admin will successfully login to the application.					

5.4.2. Blood Search

Test case #2			**Test case name:** Blood Search		
System: Blood BankBD			**Subsystem: N/A**		
Designed by: Md. Rejoan Billah			**Designed date:** 05-February-2019		
Executed by: Raj			**Executed date:** 05-April-2019		
Short description: Registered and non-registered users can search blood					
Pre-conditions: • Users will be always in the remain page when they don't input all information. • Assume that, the bloodgroup 'A+', location 'dhanmondi'					
Step	**bloodgroup**	**location**	**Expected result**	**Pass/Fail**	**Comment**
1	A+		No data shown	Pass	
2		dhanmondi	No data shown	Pass	
3	A+	dhanmondi	Matched Data will show	Pass	
Post-conditions: Registered and non-registered users search data will be shown					

Chapter 6

User Manual

6.1. User Manual

6.1.1. Send Request

Registered and Non registered User can send request for blood.

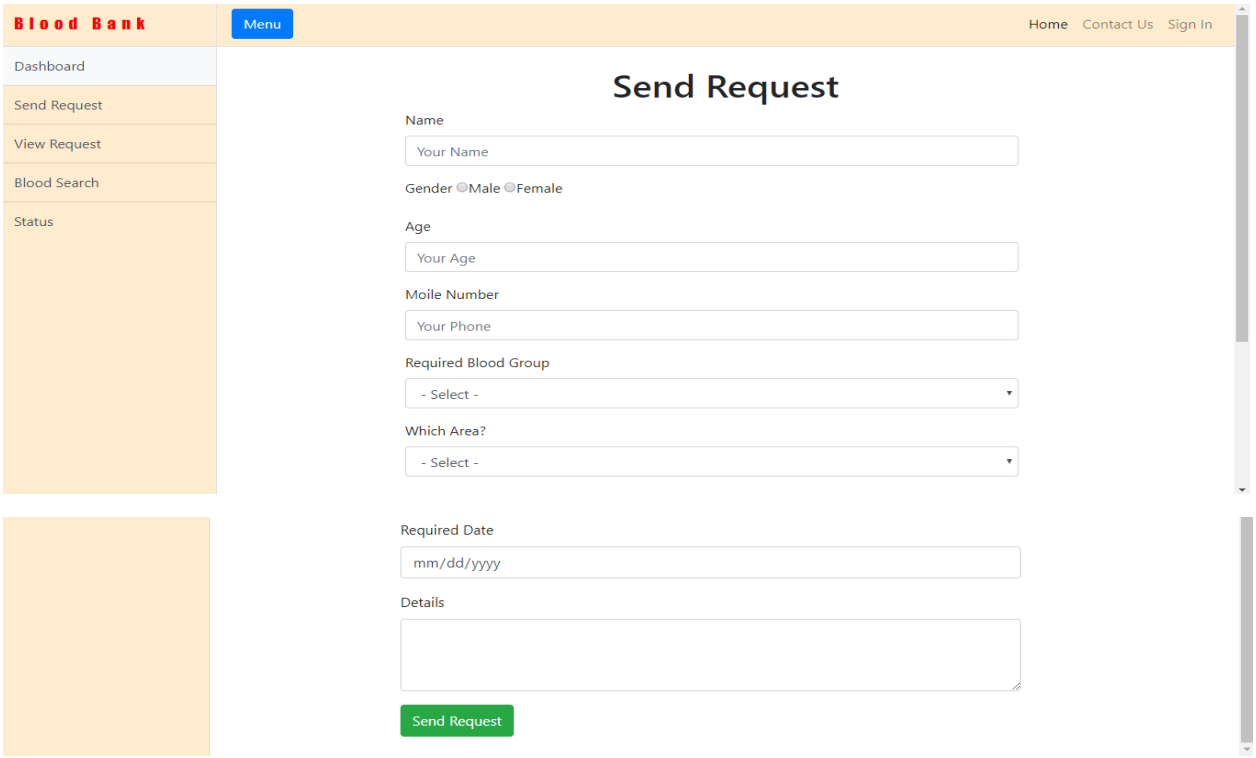

Figure 6.1.1: User Manual of send request

6.1.2. View Request

Registered and Non registered User can check their view request section for requested blood.

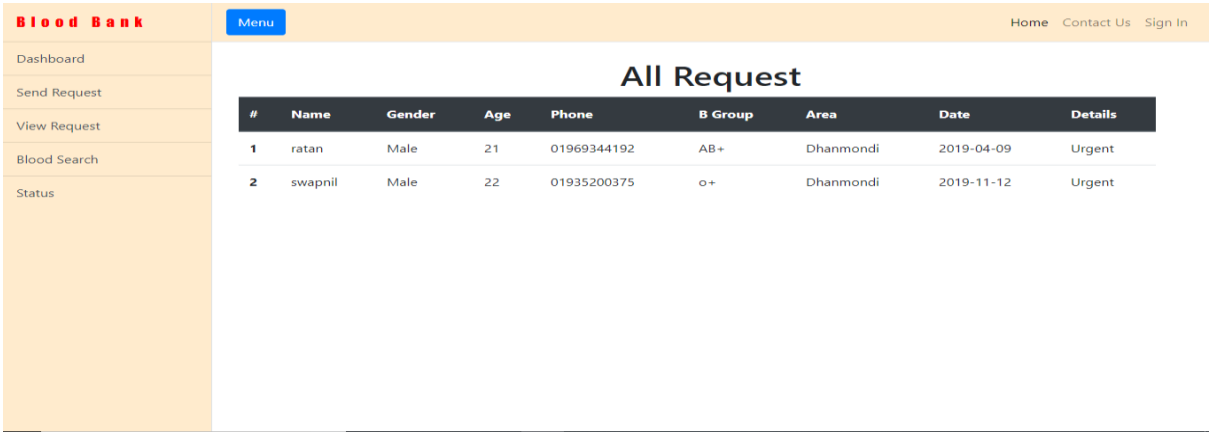

Figure 6.1.2: User Manual of view request

6.1.3. Registration

Anyone who interested to donate blood can register by providing all required information.

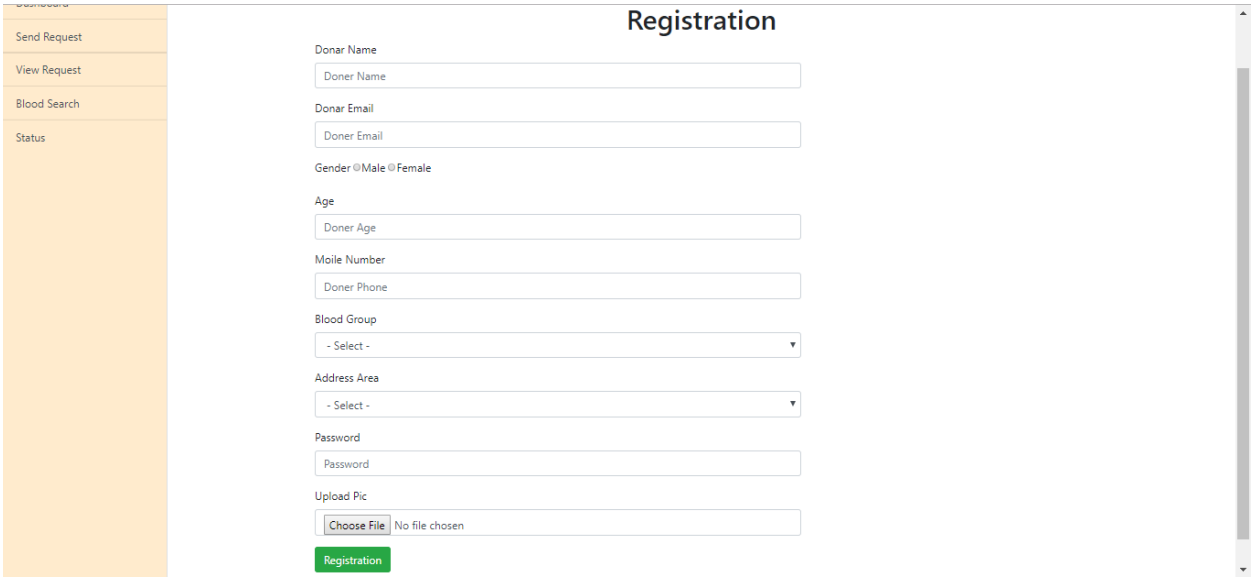

Figure 6.1.3: User Manual of registration

6.1.4. Sign in

Registered users can sign in by providing their registration info.

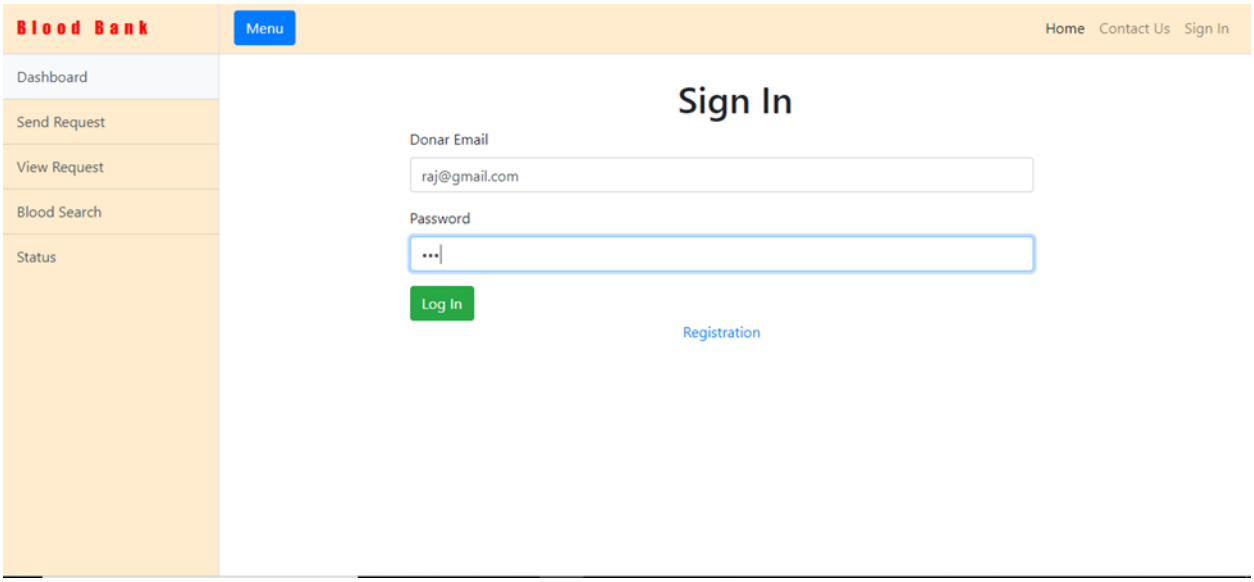

Figure 6.1.4: User Manual of sign in

6.1.5. Blood Donation

Only signed in users can fill up blood donation form.

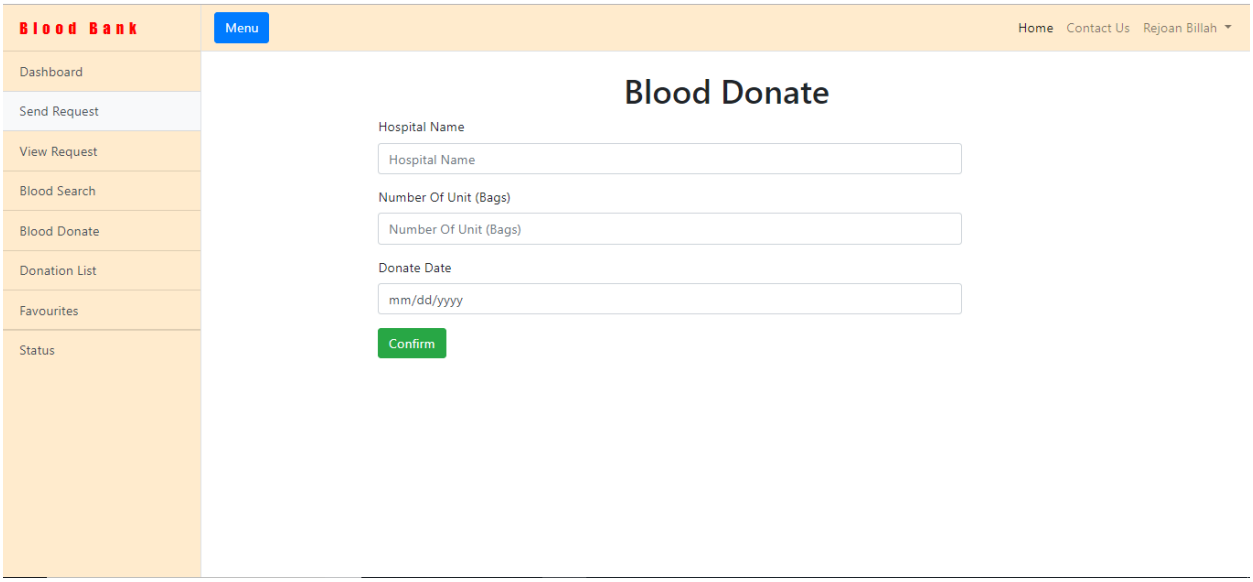

Figure 6.1.5: User Manual of blood donation

6.1.6. Donation List

Only signed in users can view their donation history from donation list. Here users can also update their donation info by clicking update.

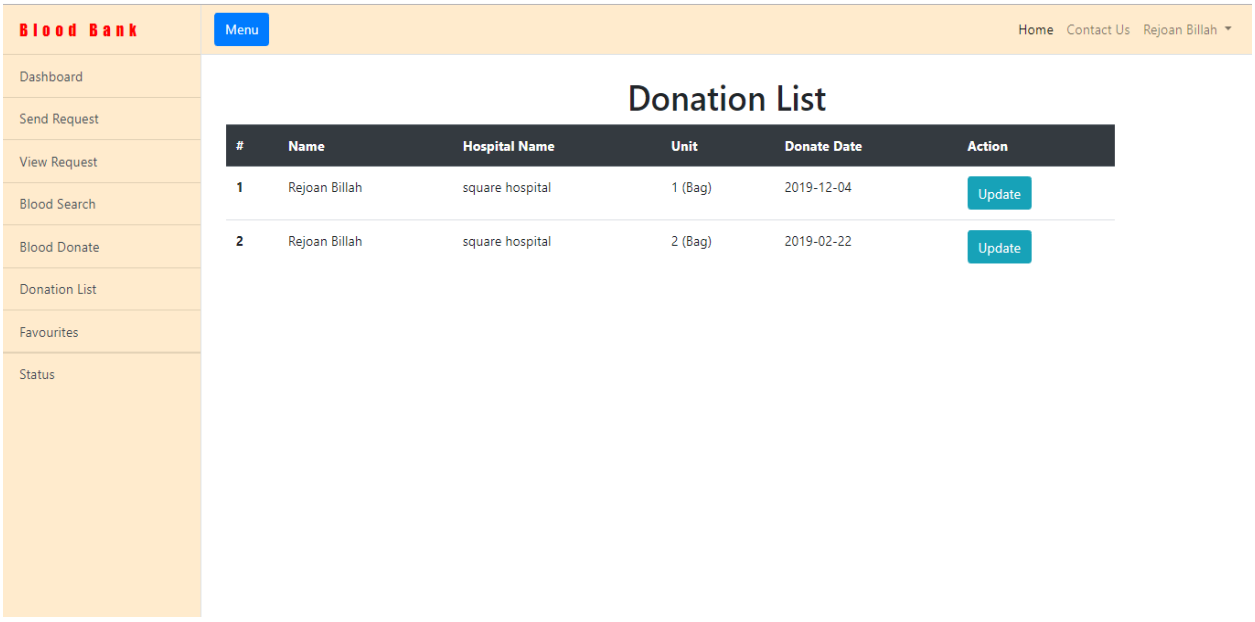

Figure 6.1.6: User Manual of donation list

6.1.7. Favorites

Signed in users can add other donor to their favorites list and also can view from same page.

Figure 6.1.7: User Manual of favorites

6.1.8. Update Profile

Signed in Users can update their profile by selecting update profile from menu bar.

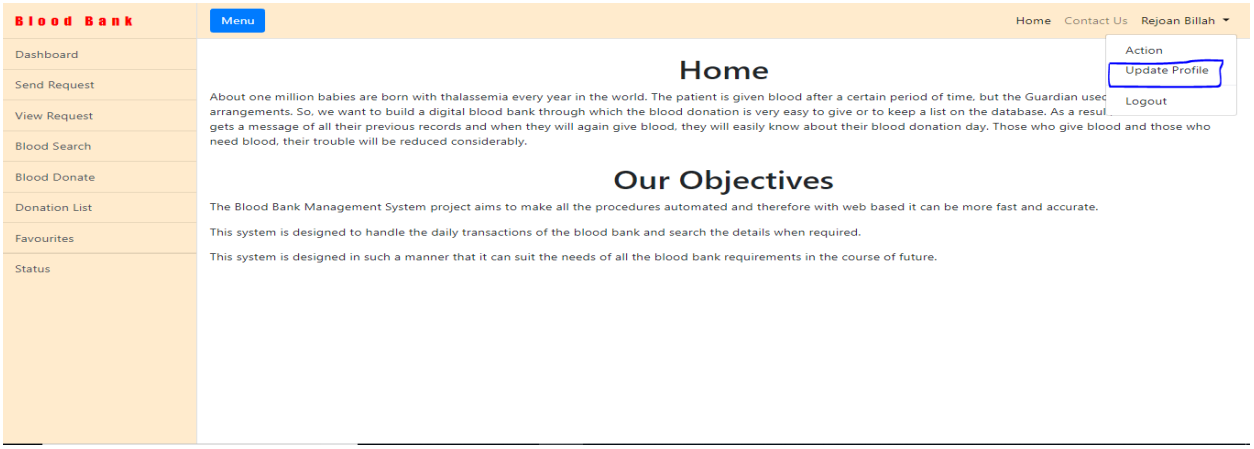

Figure 6.1.8: User Manual of update profile

6.1.9. Sign Out

Signed in users can sign out any time they want.

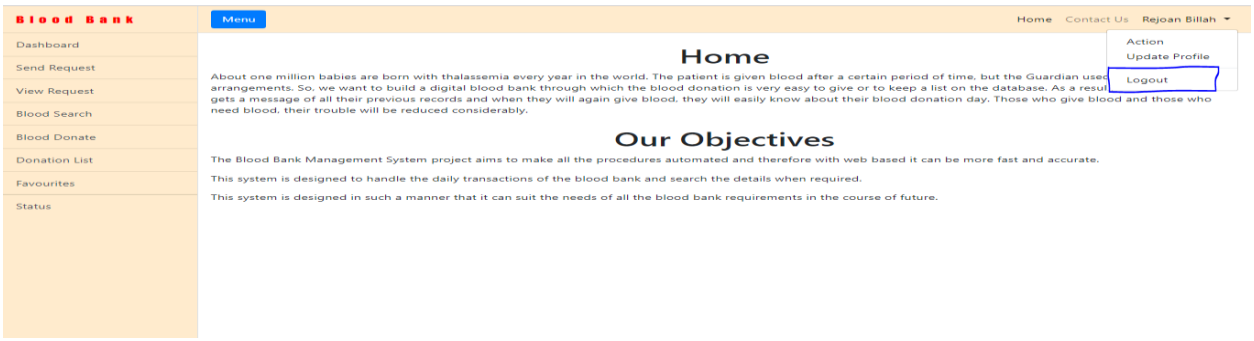

Figure 6.1.9: User Manual of sign out

Chapter 7

Project Summary & Conclusion

7.1. Summary & Conclusion

Thanks to almighty ALLAH WHO blessed us to understand and carry out this work within my expected time-frame. By this project I am trying to help our country. All phases of this project are important because any kind of mistake can damage the whole system. And this documentation is the note of whole process of this Blood BankBD system.

The project from a personal point of view helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

The project also provided us the opportunity of interacting with our teachers and to gain from their best experience

We think that all kind of people of our country will get a great benefit from this project.

7.2. Github Link

https://github.com/rajbx/Blood-BankBD

7.3. Limitations

During developing this project, I have faced some limitations. Now I will describe those limitations in brief.

- **Alert Message to Donors:** In this system I wanted to add an alert message for donors about informing their availability to donate blood. But I couldn't make this function for the donors.
- **Only Web Version:** This system is available only in web version. But we all know now peoples wants an application for any kind of internet works. I didn't make any android oi ISO version of this project. So mobile users need to go their browser to view this system by entering bloodbankbd.ml

7.4. Obstacles & Achievements

7.4.1. Obstacles:

In developing a project, obstacles mean challenges. Obviously, I had several obstacles during development of this project. After some development I was confused that if I am going on right way or my track is wrong. Also, I have one thing in my head is, is this project going to help peoples over country or not. Also, how much benefit they will get from my system. Here are some other obstacles,

- Donors status: I need to show which donor is available and which donors are unavailable right now. This function was a challenge for me. After researching from google I tried too many times for this function and finally I did it.
- Email request to registered donors: When a user send request for blood then this request info will forward to all registered donors' email and they will be informed about the request. That's why I need to search and search all over the google for the solution. Then I bought a domain for my project and after that I did this email function.

7.4.2. Achievements

After completing this project my big achievements are the knowledge about PHP language and also now, I know how software development life cycle works. Also, I have a project to show my experience in job sector.

7.5. Future Scope

I will do this project for Android OS and for IOS. So that this project can reach more people. Also, I will add SMS verification system for registration and alert message to donors.

7.6. References

Here I will add some references from where I got lot of helps:

- Refsnes data. (1998). W3School. Retrieved 05-05-2019, from https://www.w3schools.com/
- John resig. (2006). Jquery. Retrieved 05-05-2019, from http://jquery.com/
- Chad hurley, steve chen, jawed karim. (2005). YouTube. Retrieved 05-05-2019, from https://www.youtube.com/
- Mark otto, jacob thornton. (2011). Bootstrap. Retrieved 05-05-2019, from https://getbootstrap.com/
- Jeff Atwood, Joel Spolsky. (2008). Stackoverflow. Retrieved 05-05-2019, from https://stackoverflow.com/