



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

(AIUB)

Department of Computer Science and Engineering

Project Topic: A Blood Donor Website.

Course Title: Software Engineering

Section: [K]

Class ID: 01495

Submitted To:

Dr. Md. Taimur Ahad

Assistant Professor, Computer Science

Submitted By:

Group Member's	ID
Rakin Sad Aftab	20-41991-1
Md Rabiul Islam Rasel	20-42067-1
Mir Maruf Ahmed	20-42082-1

Blood Donor Website

Problem definition:

People of the required blood group are often not available in an emergency. Moreover, after it becomes difficult to collect the blood of a healthy person, keeping these issues in mind, our primary goal is to create a self-contained blood donor list and attach all their Information.

Our primary goal is to create a self-contained blood donor list and attach their Information for this issue. the Blood Donner Website will have the following features:

1. Donor and Blood Receiver register on the site.
2. For Registration:
 - Provide important Information (Name, Age, Gender, Address, Blood Group, NID or Birth Certificate, Medical Condition Certificate)
 - Take UserID and Password
3. Login Phase:
 - Select Donor Mode or Receiver Mode
4. Donner Mode:
 - Set recent blood donation time
 - Update his Status active/away
 - Show latest blood request post as like user blood group and address
 - Check date-time and confirm his donation.
5. Blood Receiver Mode:
 - Upload a post with date-time, address, and blood group
 - Set date-time and blood group for confirmed his position.
 - Manually search available active donor who matches his required blood group
 - Directly send a request for blood matching active donner
 - If donner documented, then both share their communication details
6. Administration:

It is creating a ranking by listing those donating the most blood and rewarding them socially accordingly. It should be solved if there have any problems.

Mind Map

For

Blood Donor Website



Software Requirements Specification SRS

Version 1.0
<<Annotated Version>>

September 9, 2021

A Blood Donor Website

Submitted in partial fulfillment
Of the requirements of Software Engineering

1.0 Introduction

1.1 Purpose

The purpose of this document is to gift a close description of a blood donation website. It'll make a case for the aim and options of the website, the System's interfaces, what the System can do, the constraints underneath that it should operate, and the way the System can react to external stimuli. This document is meant for all stakeholders, and the System's developers can be projected for its approval to the Regional Historical Society.

1.2 Scope of Project

This website will be an online emergency blood donor and receiver for a local emergency of regional historical society. This website will be designed to maximize the user's facility by providing tools to assist in automating the procedure to get the instant blood donor and receiver information, which would otherwise have to be performed manually. The System will meet the user's needs, easy to understand and use.

More specifically, this website is designed to allow a user to manage and communicate with a group of admins. The online site will facilitate communication between donors, receivers, and admins via phone or text. Preformatted replies to forms will use in every stage of the donation. The receiver progresses through the System to provide a specific process; the form location is configurable with the options of application maintenance. Receivers, donors, and admins are in the list of relational databases, and the system also contains them.

1.3 Glossary

Term	Definition
Active Information	The System tracks the document. It's a narrative that's planned to be announced to the public website.
User	A person is submitting Information for reviewing. In the case of multiple users, this term refers to the principal user, and all communication is established.
Database	This System monitors all collection of the Information.
Admin	A person who receives the Information sends Information for review and makes final judgments for requirements.
Field	A cell within a form.
Historical Society Database	The existing membership database and the HS database.
Software Requirements Specification	A document that thoroughly describes all the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with interest in the project which is not a developer.
User	Blood Donor, Receiver, Admin.

1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5 Document overview

The next chapter of This document gives an overview of the product's functionality in the Overall Description section. Then, it describes the informal requirements and establishes a context for the technical requirements specification in the next chapter.

The third chapter describes the details of the product's functionality, and the Requirements Specification section for the developers is written primarily.

The document describes the same software product in its entirety but is intended for different audiences and thus used in both sections of other languages.

2.0 Overall Description

2.1 System Environment

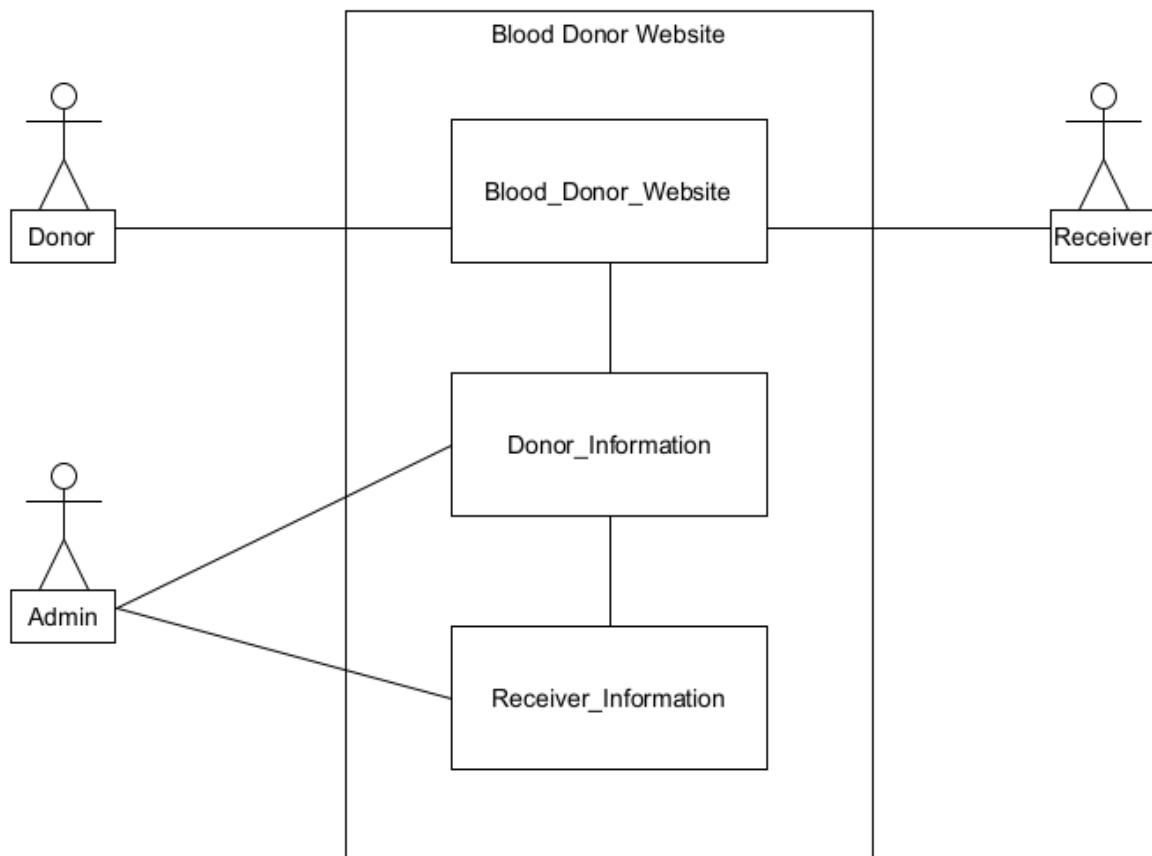


Figure – 1: System Environment

A blood donor website has three active actors and one cooperating System. Actors can access the online System through the Internet. Donors and receivers can communicate in the System via phone call or text message. Finally, the administrator can access the entire System directly. In addition, there is a link to the Historical Society.

<< The division of the Blood donor website into two parts, the donors can see the list of required blood groups, and if any of them is matches his blood group, they can text or call the receiver. On the other hand, the blood receivers can search for the available donor and contact him via phone call or text message.>>

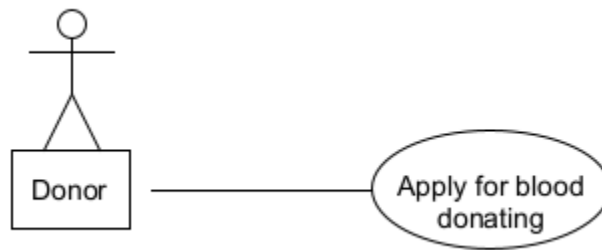
2.2 Functional Requirements Specification

This section indicates the use cases for each of the active readers separately. The users and admins have only one - use case apiece, while the admin is the leading actor in this System.

2.2.1 User Use Case

Use case: Apply for blood donating and receiving.

Diagram:



Brief Description

Donors can access the website after registration and see the list of required blood groups, and if any of them is matches his blood group, they can text or call the receiver.

Initial Step-By-Step Description

Before this use-case can be initiated, and the user has already accessed the Blood donor website.

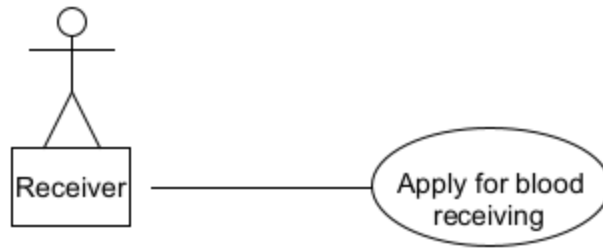
1. The donor can search by request for blood donation.
2. The System displays the searched item to the reader.
3. The donor can view the request if the blood group is mixed with his blood group
4. The System presents the blood receiver information.
5. The user can save the Information for further inquiry.
6. The System provides the receiver location.

2.2.2 Author Use Case

For the same blood group of multiple receivers, the admin will handle this.

Use case: Submit Article

Diagram:



Brief Description

Blood receivers can search for the available donor and contact him via phone call or text message.

Initial Step-By-Step Description: This use case may be initiated before the receiver connects with the donor.

The receiver chooses the *phone call* option.

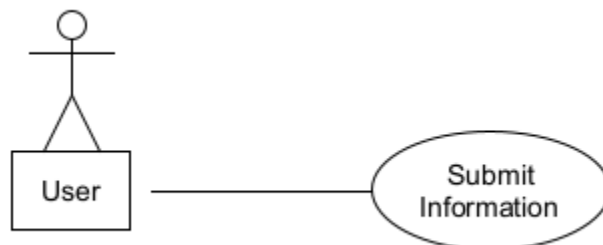
1. The System uses the *callto* HTML tag to bring up the user's call system.
2. The phone number of the donor is automatically popped up on the call section.
3. The receiver calls the donor.

2.2.3 User Use case

For multiple users, it refers to the principal user with whom all communication is made.

Use case: *Submit Article*

Diagram:



Brief Description

The user either submits Information as a donor or a receiver.

Initial Step-By-Step Description

This use case can be initiated before the user has already logged in to the Blood Donor Website.

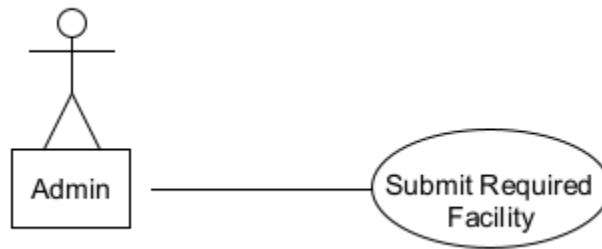
As per submitted Information

1. The user can search by request for blood donation or receiving.
2. The System displays the searched item to the reader.
3. The users can view each other's Information.
4. The System will show them their location.
5. The System uses the *callto* HTML tag to bring up the user's *call* system.
6. The phone number of the users is automatically popped up on the call section.
7. The users call each other.

2.2.4 Admin Use Case

Use case: *Submit Required Facility*

Diagram:



Brief Description

The admin submits a review of the Information.

Initial Step-By-Step Description

This use case can be initiated before the admin has already logged in to the Blood donor website.

1. The admin will receive an email with each request.
2. The admin chooses the *Email button*.
3. The System uses to *mailto* HTML tag to bring up the user's email system.
4. The admin fills in the Subject line and attaches the file as directed, and emails it.
5. The System generates and sends an email acknowledgment.

2.2.5 Editor Use Case

The Editor has followed this set of use cases:

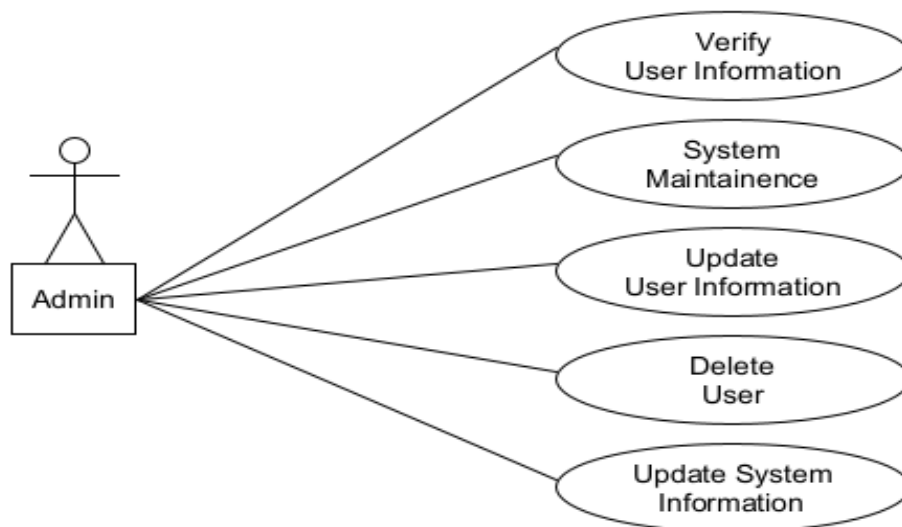
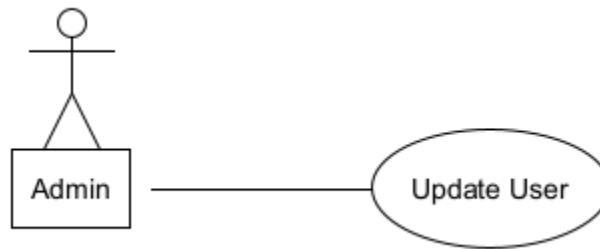


Figure – 2: Editor Use Case

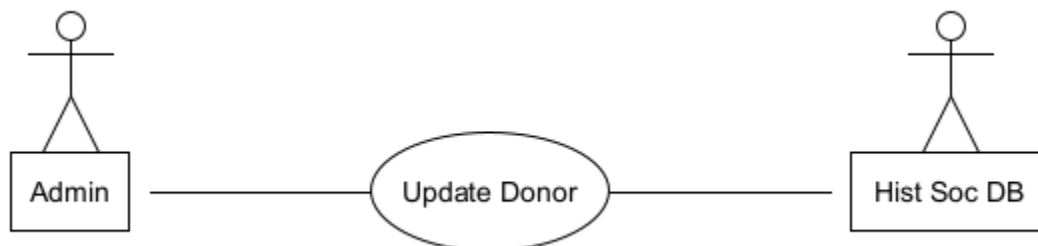
Update information use cases**2.2.6 Use Case: *Update Author*****Diagram:****Brief Description**

The admin enters a new user or updates Information about a current user.

Initial Step-By-Step Description

This use case can be initiated before the admin has already logged in to the main page of the Article Manager.

1. The admin selects to *Add/Update Author*.
2. The System comes up with a choice of updating and adding.
3. The admin chooses to add or update.
4. If the admin is updating a user, the system presents a list of users to choose from and comes up with a grid filling in with the Information; else, the System presents a blank grid.
5. The admin fills in the pieces of information and submits the form.
6. The System verifies the Information and returns it to the admin on the main page.

2.2.7 Use Case: *Update donor***Diagram:****Brief Description**

The admin enters a new donor or updates Information about a current donor.

Initial Step-By-Step Description

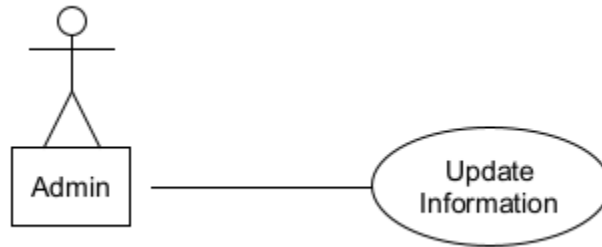
This use case can be initiated before the Editor has already accessed the main page.

1. The admin selects to *Add/Update user*.
2. The System comes up with a choice of adding or updating.
3. The donor chooses to add or to update.
4. The System connects to the Historical Society Database.

5. If the donor is updating a donor, the System presents a grid with the Information about the donor; else, the System presents a list of members for the Editor to select a doctor and shows a grid for the person chosen.
6. The admin fills in the pieces of information and submits the form.
7. The system verifies the information and returns it to the admin on the main page.

2.2.8 Use Case: *Update Information*

Diagram:



Brief Description

The admin enters Information about an existing user.

Initial Step-By-Step Description

This use case can be initiated before the admin has already accessed the main page

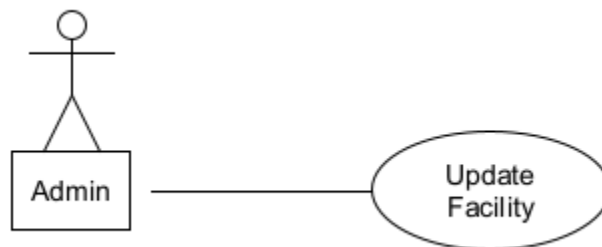
The Editor selects to *Update Article*.

1. The System presents s list of active users.
2. The System comes up with Information about the chosen users.
3. The admin updates and submit the form.
4. The System verifies the Information and returns it to the admin on the main page.

2.2.9 Use Case: *Update Facility*

This use case extends the *update facility* use case.

Diagram:



Brief Description

The admin enters a facility in the system.

Initial Step-By-Step Description

This use case can be initiated before the admin has already logged in to the facility page using the Update facility use case

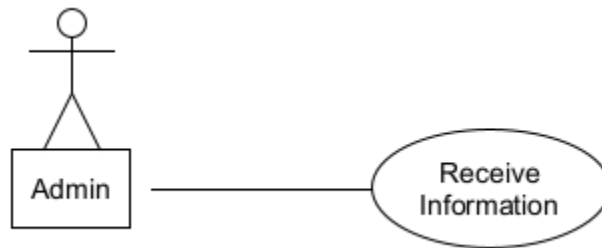
1. The admin selects to *Receive facility*.
2. The System gives a grid for filling with the Information's.

3. The admin fills in the pieces of information and submits the form.
4. The System verifies the Information and returns it to the admin on the main page.

Handle Article use cases

2.2.10 Use case: *Receive Information*

Diagram:



Brief Description

The admin enters new or revised information in the system.

Initial Step-By-Step Description

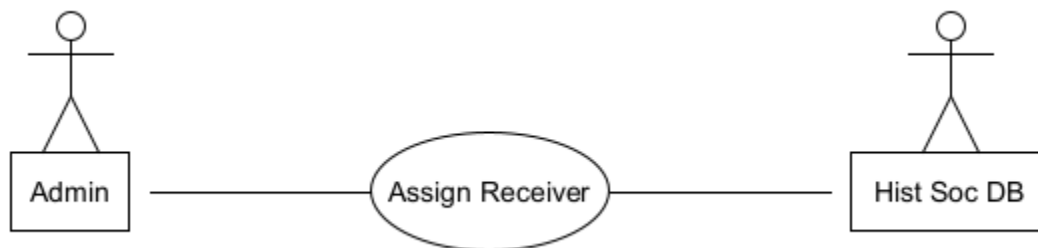
This use case can be initiated before the admin has already logged in to the main page.

1. The admin selects to *Receive Information*.
2. The System presents a choice of entering new Information or updating the existing Information.
3. The admin chooses to update or add.
4. If the admin is updating the Information, the System presents a list of Information to choose from and comes up with a grid for filling with the Information; else, the System presents a blank grid.
5. The admin fills in the pieces of information and submits the form.
6. The System verifies the pieces of information and returns the admin to the main page.

2.2.11 Use Case: *Assign Receiver*

This use case extends the *Update information* use case.

Diagram:



Brief Description

The admin assigns one or more doctors to a user.

Initial Step-By-Step Description

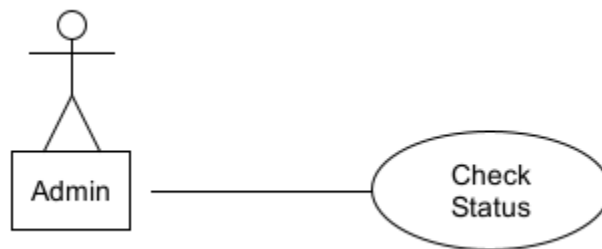
This use case may be initiated before the admin has accessed the user using the *Update information* use case.

1. The admin selects to *Assign receiver*.
2. The System presents a list of receivers with their Status.
3. The admin selects a receiver.
4. The system verifies whether the member is still an active member or not using the Historical Society Database.
5. The admin repeats steps 3 and 4 until good receivers are assigned.
6. The system emails the receiver, attaching the article and requesting that they do the review.
7. The System returns the admin to the *Update information* use case.

Check Status use case:

2.2.12 Use case: *Check Status*

Diagram:



Brief Description, The admin checks the Status of all active users.

Initial Step-By-Step Description

This use case may have been initiated before, and the admin has already accessed the main page.

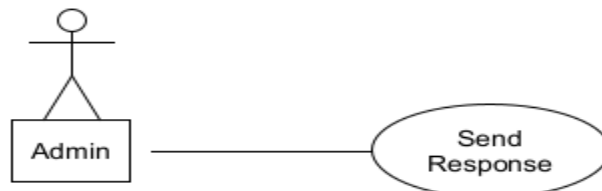
1. The admin selects to *Check Status*.
2. The System returns a list of scrollable all active users with their Status.
3. The System returns the admin to the main page.

Send Recommendation use cases:

2.2.13 Use case: *Send Response*

This use case extends the *updated user's* use case.

Diagram:



Brief Description, The admin sends a response to a user.

Initial Step-By-Step Description

This use case may be initiated before, and the admin has already accessed the user using the *Update user* use case.

1. The admin starts to *Send Responses*.
2. The System links the email system and puts the user's email address in the Recipient line and the article's name on the subject line.
3. The admin fills out the email text and sends the message.
4. The System returns to the main page.

<< Since three actors have only one use case, the diagram summary can only involve the Editor. Adapt the rules of needs for the document rather than adapt the paper to fit the rules.>>

2.3 User Characteristics

The Donors are expected to be Internet literate and be able to use a search engine. The main screen of the blood donor website will have the screen function and a link to "Blood Receiver Information."

The Receivers are expected to be Internet literate and be able to use a search engine. In addition, the main screen of the blood donor website will have the screen function and a link to "Blood Donor Information."

The admin is expecting to be Windows literate and to be able to use the button, pull-down menus, and similar tools.

2.4 Non-Functional Requirements

The blood donor website will be on a server with high-speed Internet capability. The Historical Society will determine the physical machine to be used. The software developed here assumes a tool such as Tomcat to connect the Web pages and the database. Therefore, the speed of the user's connection will depend on the hardware used rather than the characteristics of this System.

3.0 Requirements Specification

3.1 External Interface Requirements

The only connection of an external system is the link to the Historical Society (HS) Database to verify a user's membership. The admin believes that a society member is much more likely to be an influential reviewer and has imposed a membership requirement for a donor. Therefore, the HS Database has fields of heed to the Web Publishing Systems: membership id number, members' name, and email address (an optional field for the HS Database).

The Assigned receiver use case sends the receiver ID to the HS Database, and a Boolean is returned denoting membership status. The Update receiver use case requests a list of member names, membership numbers, and (optional) email addresses when adding a new receiver. It returns a Boolean for membership status while updating a receiver.

1. Study the Software Engineering process model to appropriate process model to develop your proposed system solution.

The project we have undertaken is a website of blood donors. The project aims to develop a website. After carefully reviewing the existing SDLC, we have decided to utilize the Agile model as SDLC in our project.

2. Present your arguments based on your analysis about why your selected method(s) is the best choice among all other methods to develop your proposed software.

The reason for selecting Agile as SDLC are given below:

- a. The benefit of agile methodology is the ability to find problems and create solutions quickly and efficiently.
- b. The flexibility of the agile method allows project teams to respond to customer reactions and constantly improve the product.
- c. People and interactions are emphasized rather than processes and tools.
- d. Continuous attention to technical excellence and sound design.
- e. Working software is delivered frequently (weeks rather than months).

3. Identify all the roles in the project management activities in software development. Describes the responsibilities of the role in the software development.


The roles and responsibilities are distributed in the project given below:

Name	Role	Responsibilities
Mir Maruf Ahmed	Project manager	Manage everything about the project.
Rakin Sad Aftab	Programmer	Programs all the essential functions.
Md Rabiul Islam Rasel	Software quality testing engineer	Test the software quality and report to the programmer.

4. Your Design Specification must address the evaluation rubrics mentioned in the software engineering course outline.

According to the course rubrics, applying the design specifications:

Design Specification	Course Rubrics
Applied a mode: The Agile model will be used in the project decided by the software development team.	We are applying the software development methodology based on the suggestions of our course outline.
Applied UML: The team developed use case diagrams.	We used the use case diagram because of the course outline suggestion.



The image shows a web browser window with an orange background. In the top-left corner is a logo for 'DONATION BLOOD' featuring a stylized black hand holding a red blood drop. The title 'Blood Donor Web' is centered at the top. Below the title are two input fields: 'Username' and 'Password'. Under the 'Password' field are two buttons: 'Log In' and 'Sign Up'. At the bottom center is a blue button labeled 'Know About Us'. The browser window has standard minimize, maximize, and close buttons in the top-right corner.

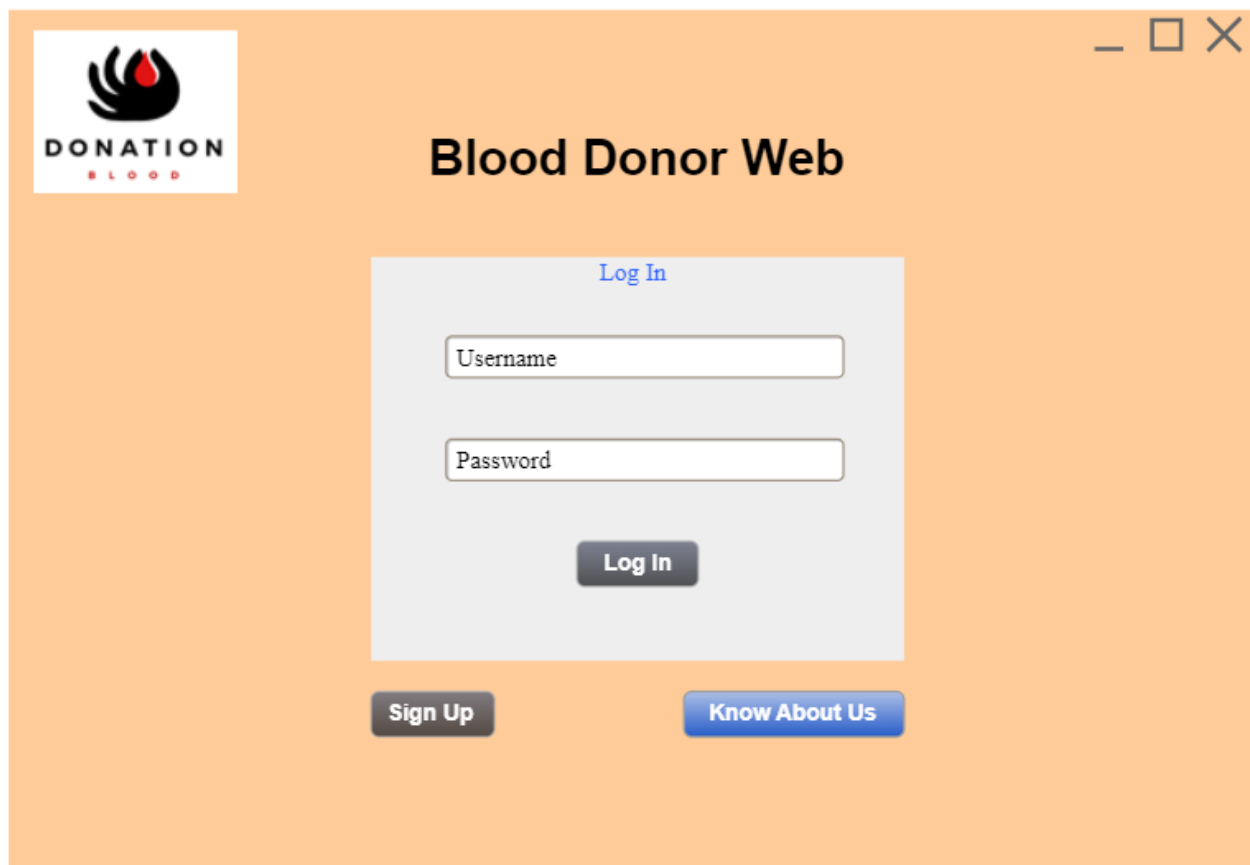
Blood Donor Web

Username

Password

Log In Sign Up

Know About Us



The image shows the same web browser window as above, but with a light gray modal box centered on the screen. The modal contains a 'Log In' link at the top, followed by 'Username' and 'Password' input fields, and a 'Log In' button. Below the modal, the 'Sign Up' and 'Know About Us' buttons are visible. The browser window's title bar and controls remain the same.

Blood Donor Web

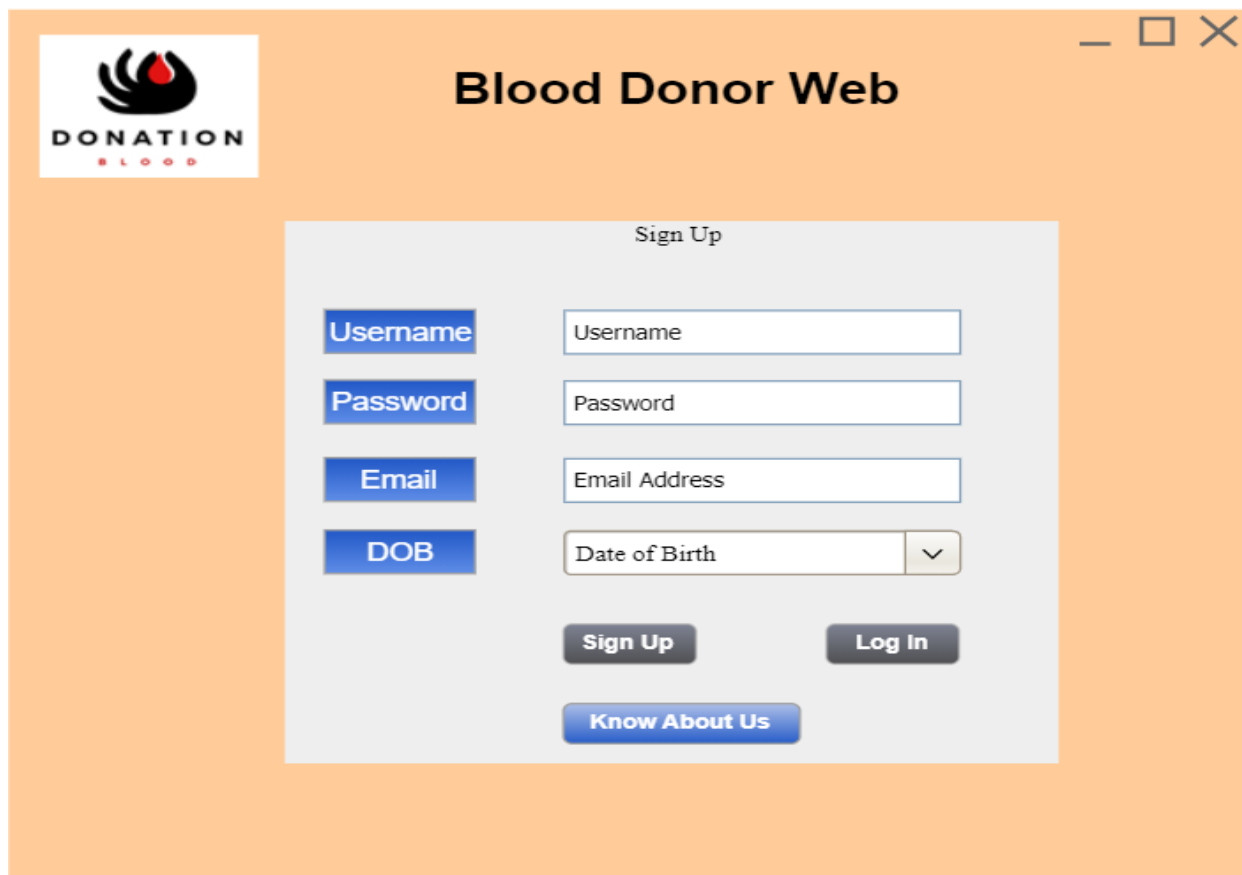
Log In

Username

Password

Log In

Sign Up Know About Us



The image shows a web browser window titled "Blood Donor Web". In the top left corner is a logo with a black flame-like shape above the text "DONATION" and "BLOOD" in red. The main heading is "Blood Donor Web". Below it is a "Sign Up" form. The form has four labels on the left: "Username", "Password", "Email", and "DOB", each in a blue box. To the right of each label is an input field: "Username" (text), "Password" (text), "Email Address" (text), and "Date of Birth" (calendar icon). Below the input fields are three buttons: "Sign Up" (dark grey), "Log In" (dark grey), and "Know About Us" (blue).

Blood Donor Web

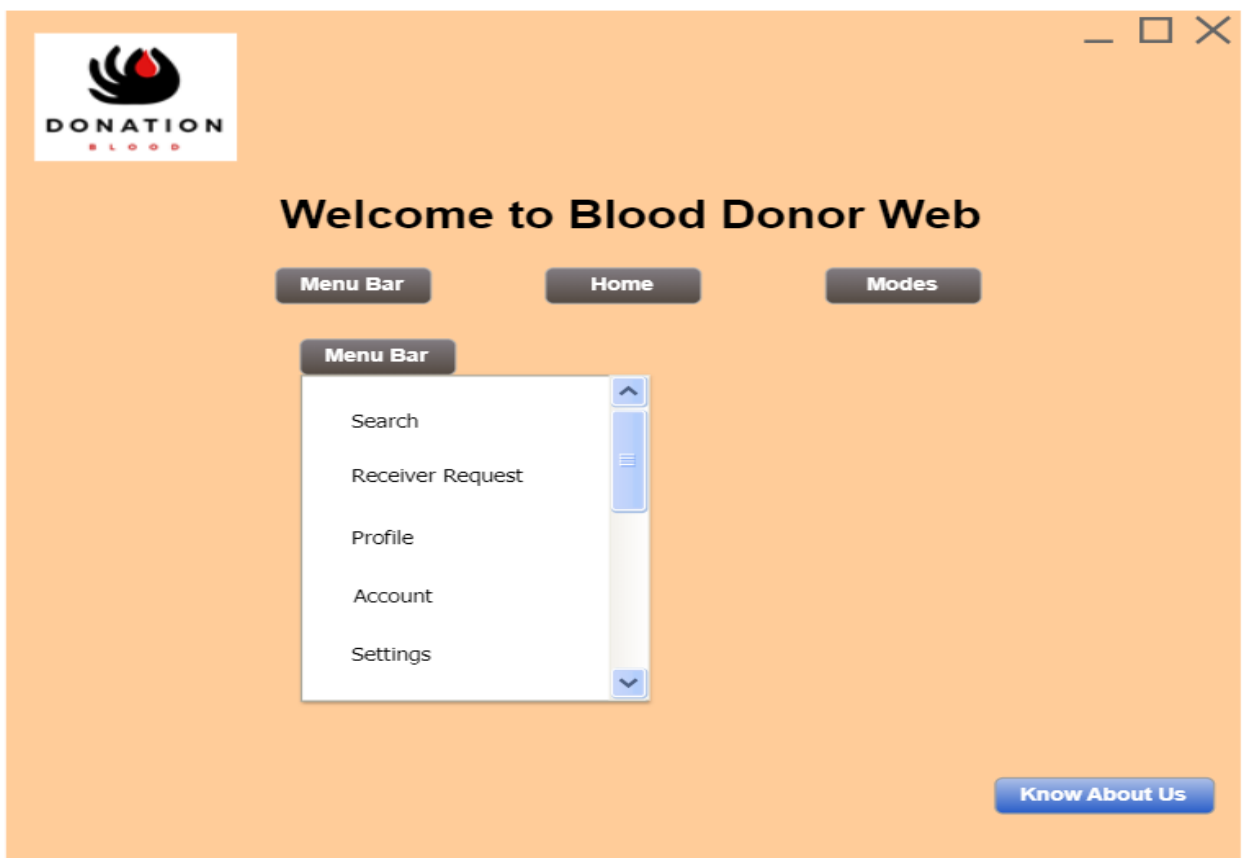
Sign Up

Username

Password

Email

DOB



The image shows a web browser window titled "Blood Donor Web". In the top left corner is a logo with a black flame-like shape above the text "DONATION" and "BLOOD" in red. The main heading is "Welcome to Blood Donor Web". Below it are three buttons: "Menu Bar", "Home", and "Modes". The "Menu Bar" button is highlighted, and a dropdown menu is open below it. The dropdown menu has a "Menu Bar" label and a list of items: "Search", "Receiver Request", "Profile", "Account", and "Settings". At the bottom right is a "Know About Us" button.

Welcome to Blood Donor Web

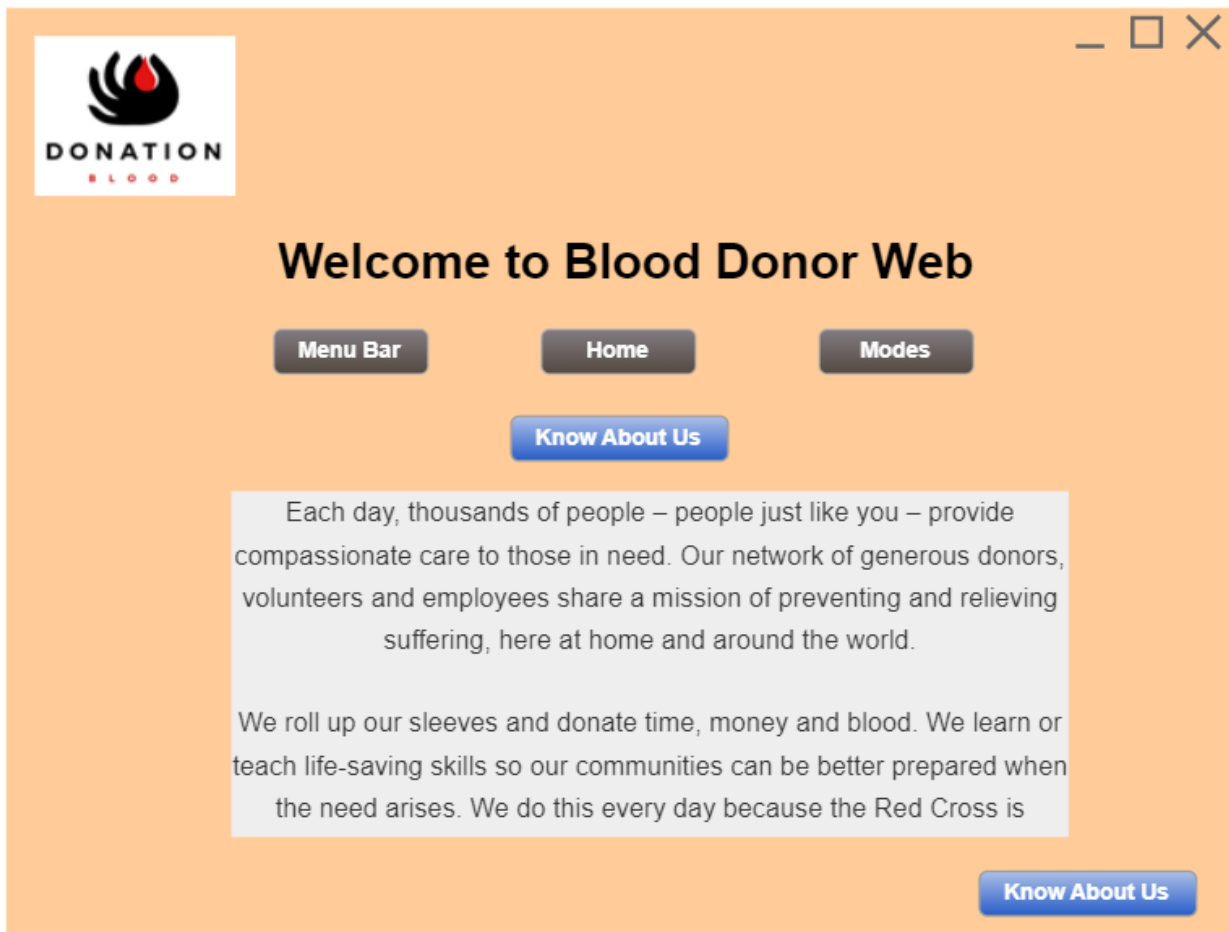
Search

Receiver Request

Profile

Account

Settings



Blood Donor Website

1. Log In

Test Case ID	CS_01	Test Case Description	Test the Login Functionality in the Application		
Created By	Rakin	Reviewed By	Rasel	Version	1.1

QA Tester's Log	Review comments from Rasel incorporate in version 1.1
------------------------	---

Tester's Name	Rasel	Date Tested	9-Nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	-------	--------------------	------------	---	------

S#	Prerequisites
1	Access to chrome Browser
2	
3	
4	

S#	Test Data
1	UserID: 20-42082-1
2	Password: 201665X
3	
4	

Test Scenario	Verify that by entering valid UserID and Password, the user can log in.
----------------------	---

Step#	Step Details	Expected Result	Actual Result	Pass/Fail/Not executed/Suspended
1	Navigate to https://donorblood.com	Site should open	As expected,	Pass
2	Enter UserID & Password	Credential can be entered	As expected,	Pass
3	Click Submit	User Is logged in	As expected,	Pass
4	Invalid UserID & Password	Return to previous page	As expected,	Pass

2. Sign Up

Test Case ID	CS_02	Test Case Description	Test the signup functionality in the Application		
Created By	Rakin	Reviewed By	Maruf	Version	1.1

QA Tester's Log	Review comments from Maruf incorporate in version 1.1
------------------------	---

Tester's Name	Maruf	Date Tested	9-Nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	-------	--------------------	------------	---	------

S#	Prerequisites
1	Access to chrome Browser
2	
3	
4	

S#	Test Data
1	UserID: 20-41991-1
2	Password: 111111X
3	Email: maruf@gmail.com
4	Date of Birth: October 2, 2000

Test Scenario	Verify that by entering valid UserID, Password, Email, and DOB, the user can sign up.
----------------------	---

Step#	Step Details	Expected Result	Actual Result	Pass/Fail/Not executed/Suspended
1	Navigate to https://donorblood.com	Site should open	As expected,	Pass
2	Click to sign up	Sign Up tap will open	As expected,	Pass
3	Enter UserID, DOB Password, and Email	User Signed up	As expected,	Pass

3. Menu

Test Case ID	CS_03	Test Case Description	Test the Menu Functionality in the Application		
Created By	Rakin	Reviewed By	Maruf	Version	1.1

QA Tester's Log	Review comments from Maruf incorporate in version 1.1
------------------------	---

Tester's Name	Maruf	Date Tested	9-Nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	-------	--------------------	------------	---	------

S#	Prerequisites
1	Access to chrome Browser
2	Should log in first
3	
4	

S#	Test Data
1	UserID: 20-41991-1
2	Password: 111111X
3	
4	

Test Scenario	Verify that by entering a valid UserID, Password, the user can log in.
----------------------	--

Step#	Step Details	Expected Result	Actual Result	Pass/Fail/Not executed/Suspended
1	Navigate to https://donorblood.com	Site should open	As expected,	Pass
2	Enter UserID & Password	Credential can be entered	As expected,	Pass
3	Go to the home page & click menu button.	Menu should open	As expected,	Pass

4. Modes

Test Case ID	CS_04	Test Case Description	Test the Modes Functionality in the Application		
Created By	Rakin	Reviewed By	Rasel	Version	1.1

<u>QA Tester's Log</u>	Review comments from Rasel incorporate in version 1.1
-------------------------------	---

Tester's Name	Maruf	Date Tested	9-Nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	-------	--------------------	------------	---	------

S#	Prerequisites
1	Access to chrome Browser
2	Should log in first
3	
4	

S#	Test Data
1	UserID: 20-41991-1
2	Password: 111111X
3	
4	

<u>Test Scenario</u>	Verify that by entering a valid UserID, Password, the user can log in.
-----------------------------	--

Step#	Step Details	Expected Result	Actual Result	Pass/Fail/Not executed/Suspended
1	Navigate to https://donorblood.com	Site should open	As expected,	Pass
2	Enter UserID & Password	Credential can be entered	As expected,	Pass
3	Go to the home page & click modes button.	Two-mode should open—donor and Receiver mode.	As expected,	Pass

5. Log Out

Test Case ID	CS_05	Test Case Description	Test the Menu Functionality in the Application		
Created By	Rakin	Reviewed By	Rasel	Version	1.1

QA Tester's Log	Review comments from Rasel incorporate in version 1.1
------------------------	---

Tester's Name	Rasel	Date Tested	9-Nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	-------	--------------------	------------	---	------

S#	Prerequisites
1	Access to chrome Browser
2	Should log in first
3	
4	

S#	Test Data
1	UserID: 20-42082-1
2	Password: 333333X
3	
4	

Test Scenario	Verify the logout functionality.
----------------------	----------------------------------

Step#	Step Details	Expected Result	Actual Result	Pass/Fail/Not executed/Suspended
1	Navigate to https://donorblood.com	Site should open	As expected,	Pass
2	Enter UserID & Password	Credential can be entered	As expected,	Pass
3	Click Log Out	Should be logged out.	As expected,	Pass

