Software Requirements Specification Document (CS360)

Bloodlink

Group Number: 21

Muhammad Farrukh - 23100240

Muhammad Hassnain - 23100250
Muhammad Ahmad Mahmood - 23100222
Sharjeel Ahmad Shaik - 23100214
Daim Armaghan - 23100221

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management Sciences (LUMS)

Version: 1.0

Date: (dd/02/2020)

Number of hours spent on this document: 78

Contents

1.0 Introduction

1.1 Document Purpose

The purpose of this document is to help us build and maintain an application for the Bloodlink. The app being developed will allow initially people from the LUMS community to request and donate blood and eventually all over Pakistan. This document will serve as a guideline for developers to develop the application. It is also a contract between the client (Bloodlink) and us the developers.

This document will help us better identify our requirements and reference to them later. It will even help us trace that from where we got this particular requirement. Another thing to be noted is that, it can also be equally helpful for people who will try to maintain or upgrade this application later.

1.2Product Scope

Scope:

The intended user group of this software is the people across Pakistan who are in need of blood or want to donate it. It is expected that initially, the users of this software will mainly comprise students, faculty, and staff members of LUMS. Depending on the marketing strategy applied by Bloodlink (the client) and the success of this software, the user group is eventually expected to expand to the general population of Pakistan.

Objectives:

1. Automate blood request management.
2. Decrease time delays in finding a donor.
3. Eliminate the potential of requests getting lost due to human error.
4. Expand scope of Bloodlink's services.
5. Find better donors based on location and availability.

Benefits:

1. Amount of time Bloodlink members serve in blood request management will be saved.
2. Blood requests will be managed more efficiently and reliably.
3. More people will be able to avail Bloodlink's services.
4. Time required for donors to travel to the hospitals where blood is required will be reduced.

1.3 Intended Audience and Document Overview

This project aims to automate the current manual process of the Bloodlink. It is also being developed as part of a course project therefore the intended audience not only includes our team, the bloodlink team, the people they will hire in future to maintain it but also the instructor and teaching assistants of the course titled "Software Enginerring." This document contains overal description, the functionality, target users, their characteristics and functional and non-funcitonal requirements.

Definitions, Acronyms and Abbreviations

Term	Definition
API	Application programming interface. Set of protocols for building and integrating application software.
BloodLink	Our webapp, named after BloodLink-LCSS, a department within LCSS bridging gaps between donors and requestors.
HIV	Human Immunodeficiency Virus
LCSS	Lums Community Service Society. An internal lums society working for human well-being and welfare.
LDF	Lums Discussion Forum. A Facebook group for LUMS student body.
LUMS	Lahore University of Management Sciences. A Not-for-profit university located in Lahore, Punjab, Pakistan
PBTA	Punjab Blood Transfusion Authority

1.4 References and Acknowledgments

Constantine, Larry, and Lucy Lockwood. Structure and Style in Use Cases for User Interface Design.

“Frequently Asked Questions | Punjab Blood Transfusion Authority.” Pbta.punjab.gov.pk, pbta.punjab.gov.pk/faqs.

“Use Case Diagram Guidelines for Better Use Cases.” Creately Blog, 16 Feb. 2015, creately.com/blog/diagrams/use-case-diagram-guidelines/.

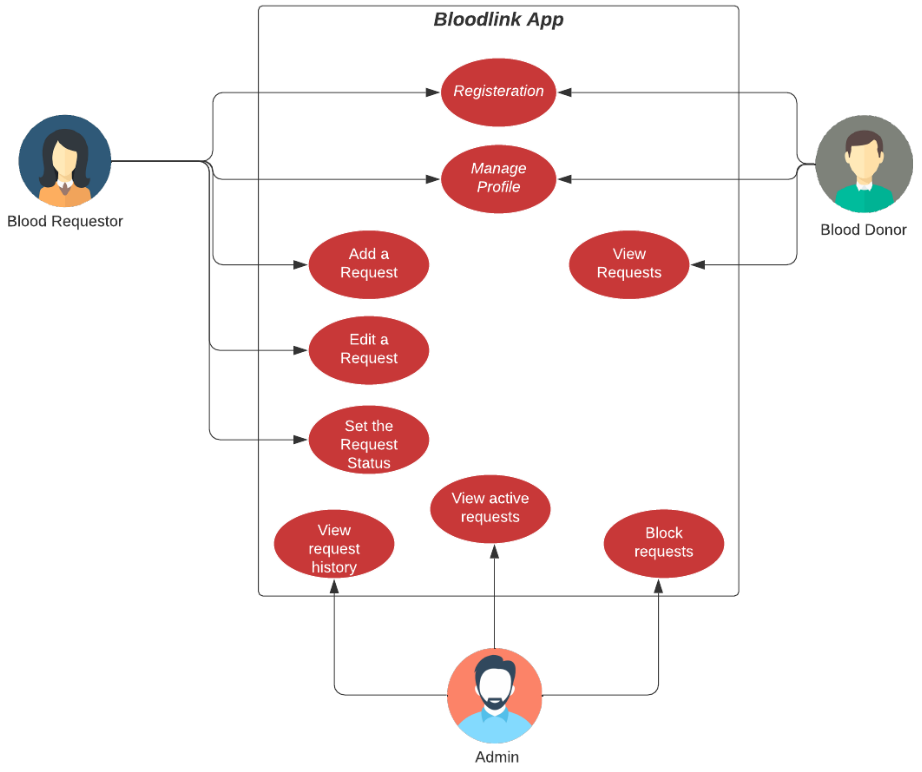
2 Overall Description

2.1 Product Perspective

The Bloodlink app is a new system that aims to automate the manual blood request management of Bloodlink and make it more efficient. Bloodlink is a department of LUMS Community Service Society (LCSS) that receives blood donation requests from people across Pakistan, mainly the students, staff, and faculty of LUMS. After receiving a request, members of Bloodlink make social media posts, contact registered donors that match the requirements, and email the student body of LUMS to arrange a donor.

The Bloodlink app will not only remove the manual hassle of managing blood requests for the Bloodlink team, but it will also streamline the process for requestors and donors by connecting them through a unified platform.

General Diagram to illustrate systems interaction with the environment



2.2 Product Functionality

Depending upon the user type, the major functions could be categorised into following three categories:

2.2.1 Requestor side functions

- Users can make blood requests

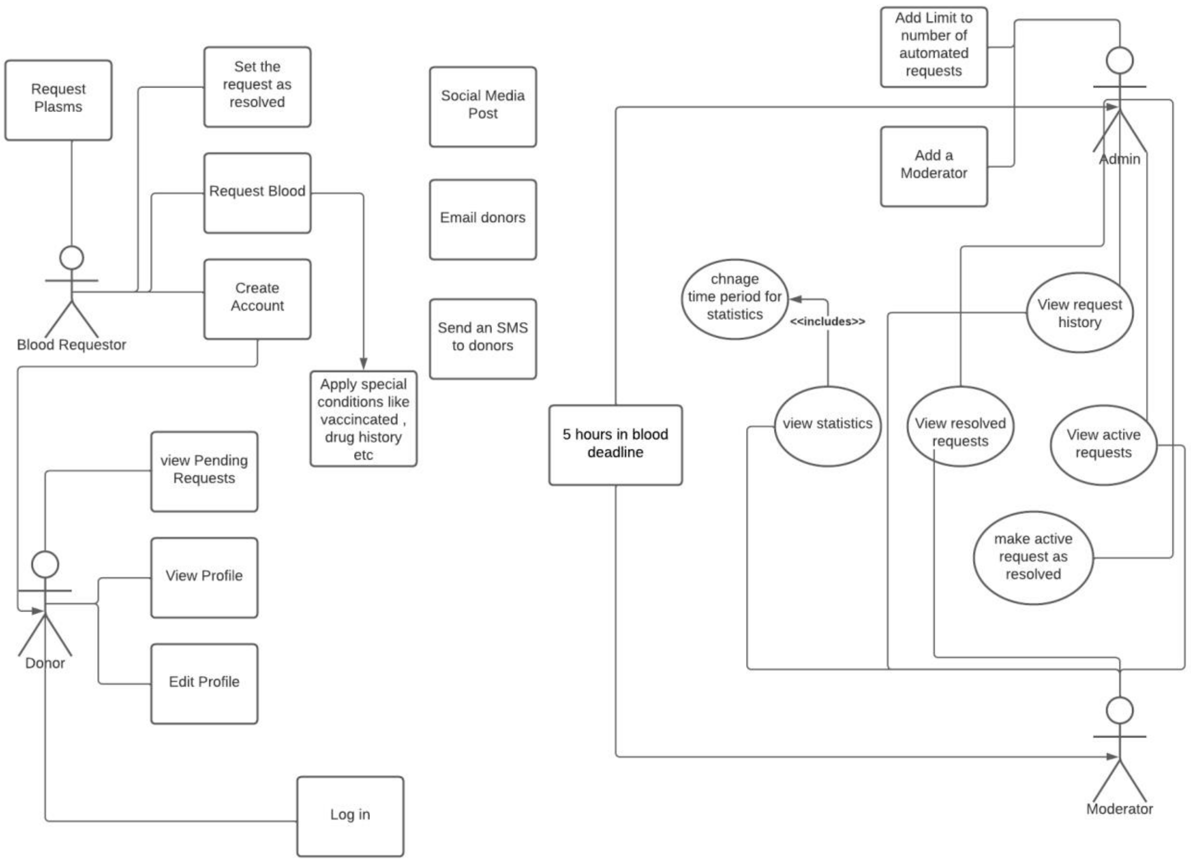
- Requestors can manage their profile
- Users will be able to add different details in a request like blood-type, urgency, attendant name, contact number, hospital name and location, number of bottles required, etc.
- Users can edit their requests
- The Bloodlink app will post request on facebook page
- The Bloodlink app will send an email upon receiving a request
- The Bloodlink app will send SMS to matching donors.
- Users can set the request status as Resolved or Unresolved

2.2.2 Donor side functions

- Users can view blood requests that matches their parameters
- Donors can manage their profile.
- Donors can mark themselves as inactive.

2.2.3 Admin side functions

- Admin will be able to track the progress of request
- Admin can view all the requests that have been generated on the app so far
- Admin can create moderators
- Set the number of automatic emails for urgent rquest



2.3 Users and Characteristics

2.3.1 Blood Requestors

Users who request blood are referred to as “requestors.” The blood requestors are the most significant users of the Bloodlink App. They are going to use the app more often as compared to donors because a donor will receive the request details through different means like SMS, email and social media.

2.3.2 Blood Donor

Blood Donors can view all blood requests that matches to their blood group and other parameters.The donors and requesters are both the most significant users, because without them the cycle of donation is not completed.

2.3.3 Admin

Admin can view all blood requests, block requests, approve requests and make moderators.

2.2.4 Moderator

Moderator will be able to remove, edit, resolve, and reactivate requests and can also view all requests history.

2.4 Assumptions and Dependencies

- Donors and recipients will have a valid email id and a phone number.
- The users of this product will be able to use browsers to use this web app and will have a basic understanding of the English language.
- The Third APIs used to send SMS, email and posting on facebook will work correctly.

3 Specific Requirements

3.1Functional Requirements

1: Sign In/ Sign Up

RQ 1.1 - Create Account/ Sign Up

- **Description**
  The system will allow the users to create their account
- **Input**
  User details like Name, password, email, country, city, area in that city, blood group, contact number.
- **Processing**
  Account creation (validation of entered details and entry in database).
- **Output**
  account is created and account successfully created message has been displayed
- **Alternate Output**
  If entered details are invalid (incomplete, incorrect format, already exists) then the account is not created and an error message is shown.

RQ 1.2 - Log In/ Sign In

- **Description**
  The system will allow the users to log into their account
- **Input**
  User login details email/username, password
- **Processing**
  authentication
- **Output**
  user is logged in and the home page is displayed
- **Alternate Output**
  Incorrect details are entered by the user and an error message is displayed.

RQ 1.3 - Log Out

- **Description**
  The system will allow the logged in users to log out of their account
- **Input**
  Just need to click on log out button
- **Processing**
  Log out
- **Output**
  the user logged out

RQ 1.4 - Phone Number Verification

- **Description**
  During account creation, the system will verify the user's phone number by sending a randomly generated code via SMS and asking them to enter the sent code.
- **Input**
  User’s phone number.
- **Processing**
  Generating a code, sending it to user’s phone number, verifying whether the user entered correct code or not.
- **Output**
  Phone number is verified.
- **Alternate Output:**
  If the wrong code is entered, the phone number is not verified and an error message is shown.

RQ 1.5 - Change Password

- **Description**
  Users can change their account’s password.
- **Input**
  Current password and new password.
- **Processing**
  Current password is verified and updated with the new password in the database.
- **Output**
  Password is changed.
- **Alternate Output**
  If the current password is wrong, an error message appears.

RQ 1.6 - Edit Account Details

- **Description**

  Users can change their account details such as name, phone number, etc.
- **Input**

  New details.
- **Processing**

  Validation of new details and updating database.
- **Output**

  User details are updated.
- **Alternate Output**

  Error message if user details are invalid.

2: Blood Request

RQ 2.1 - Create a Blood Request

- **Description**

  The user will be able to create a blood request by providing different related details
- **Input**

  Details required for a blood request (blood group, hospital name, etc.)
- **Processing**

  Requests details are validated.
- **Output**

  A blood request ready to submit.

RQ 2.2 - Create a Plasma Request

- **Description**

  The user will be able to create a plasma request by providing different related details
- **Input**

  Details required for a plasma request (blood group, hospital name, amount, etc.)
- **Processing**

  Requests details are validated.
- **Output**

  A plasma request ready to be submitted.

RQ 2.2 - Create a Plasma Request

- **Description**

The user will be able to create a plasma request by providing different related details

- **Input**
  Details required for a plasma request (blood group, hospital name, amount, etc.)
- **Processing**
  Requests details are validated.
- **Output**
  A plasma request ready to be submitted.

RQ 2.3 - Add Filters to Request

- **Description**
  The user will be able to add optional filters in their blood/plasma request such as non-smokers, vaccinated, etc.
- **Input**
  Filter selections from given options.
- **Processing**
  Filters are added to the request details.
- **Output**
  A blood/plasma request ready to be submitted with filters.

RQ 2.4 - Submit a Request

- **Description**
  The user will be able to submit their blood/plasma requests once they have entered all the relevant details.
- **Input**
  Click on the “submit” button.
- **Processing**
  System starts the process of finding a donor.
- **Output**
  Blood request becomes active.

RQ 2.5 - View My Requests

- **Description**
  The user will be able to all requests that they have submitted.
- **Input**
  Click on the “View My Requests” button.
- **Processing**
  System fetches relevant data from the database.

- **Output**

  A list of blood/plasma requests submitted by the specific user.

RQ 2.6 - Mark a Request as "Resolved"

- **Description**

  User who had requested blood/plasma, can mark their request as "resolved" to stop the system from contacting any more donors. Requestor is expected to use this option when they have found a donor.
- **Input**

  Click on the "Mark as Resolved" button in front of an active request in the list of requests shown in output of RQ 2.5.
- **Processing**

  System marks the request as resolved and stop the process of finding a donor.
- **Output**

  Request becomes inactive.

RQ 2.7 - Re-activate a Resolved Request

- **Description**

  User who had marked a request as "resolved", can reactivate that request. User might need this feature in case a donor backs out after accepting the donation request.
- **Input**

  Click on the "Reactivate" button in front of a resolved request in the list of requests shown in output of RQ 2.5.
- **Processing**

  System resumes the process of finding a donor.
- **Output**

  Request becomes active.

RQ 2.8 - Edit Details of an Active Request

- **Description**

  User who had submitted a request can edit the details of that request.
- **Input**

  Click on the "Edit" button in front of a resolved request in the list of requests shown in output of RQ 2.5.
- **Processing**

  System updates the request.
- **Output**

Request details are edited.

RQ 2.9 - Register as Donor

- **Description**

  Users can register as donors to be able to donate blood.
- **Input**

  Information required for donors (area of residence, smoking status, etc.)
- **Processing**

  System updates the database to mark the user as “donor.”
- **Output**

  User becomes a donor.

3: Blood Donation

RQ: 3.1 View Active Requests

- **Description**

  Donors will be able to see active blood/plasma requests.
- **Input**

  Just need to click on ‘View Active requests’ button
- **Processing**

  System fetches the requests marked as ‘unresolved/active’.
- **Output**

  List of active requests

RQ: 3.2 Filter requests

- **Description**

  Donors will be able to filter blood/plasma requests by their area/ hospital/ urgency etcetera.
- **Input**

  Just need to click on the ‘Filter’ button and provide relevant information.
- **Processing**

  System fetches the requests with the filter properties.
- **Output**

  List of filtered requests

RQ: 3.3 View Request Details

- **Description**

  Donors will be able to view details of any request.
- **Input**

Just need to click on the 'View' button of the request.

- **Processing**

  System fetches the request details.

- **Output**

  Details of request.

RQ: 3.4 Contact requestor

- **Description**

  Donors will be able to contact the blood requestor.

- **Input**

  Just need to click on 'Contact' button

- **Processing**

  System fetches the requestor contact number.

- **Output**

  Contact number of the requestor.

RQ: 3.5 Register as plasma donor

- **Description**

  Donors will be able to register themselves as plasma donors.

- **Input**

  Just need to click on the 'Register as Plasma Donor' button and enter information.

- **Processing**

  System registers the donor in the plasma donor directory.

- **Output**

  User is now registered as the plasma donor.

RQ: 3.6 Mark themselves as "unavailable"

- **Description**

  Donors will be able to mark themselves unavailable. So no notifications of blood request will be sent to them.

- **Input**

  Just need to click on the 'Mark Unavailable' button.

- **Processing**

  System updates the user details and marks them unavailable.

- **Output**

  User details updated.

4: Blood Request Management

RQ 4.1 - View Active Requests

- **Description**

  Admin and Moderators will be able to see active blood/plasma requests.
- **Input**

  Just need to click on ‘View Active requests’ button
- **Processing**

  System fetches the requests marked as ‘unresolved/active’.
- **Output**

  List of active requests

RQ 4.2 - View Resolved Requests

- **Description**

  Admin and Moderators will be able to see resolved blood/plasma requests.
- **Input**

  Just need to click on ‘View resolved requests’ button
- **Processing**

  System fetches the requests marked as ‘resolved’.
- **Output**

  List of resolved requests

RQ 4.3 - View Requests History

- **Description**

  Admin and Moderators will be able to see history of requests.
- **Input**

  Just need to click on ‘View History’ button
- **Processing**

  System searches for history of queried request.
- **Output**

  Detailed history of request get displayed.

RQ 4.4 - View Statistics

- **Description**

  Admin and Moderators can see statistics including number of active, resolved, reactivated, and unresolved requests.
- **Input**

Just need to click on ‘View Statistics’ button

- **Processing**

  System will process the statistical results of requests.

- **Output**

  Statistics get displayed.

RQ 4.5 - Change time period of statistics

- **Description**

  Admin and Moderators can change time period of statistics i.e., day, week, month

- **Input**

  Select time scale from ‘time period’ filter.

- **Processing**

  System will process statistical results for the specified time scale.

- **Output**

  Statistics across specified time period get displayed

RQ 4.6 - Mark request as ‘resolved’

- **Description**

  Admin and Moderators can change status of active request as ‘resolved’.

- **Input**

  Select ‘resolved’ under ‘current status’ bar.

- **Processing**

  System will update request status as resolved.

- **Output**

  Request marked as resolved.

RQ 4.7 - Edit an active request

- **Description**

  Admin and Moderators can edit details of an active request.

- **Input**

  Click ‘Edit details’ and change the entries as required.

- **Processing**

  System will process the changes and update details.

- **Output**

  Details get updated.

RQ 4.8 - Reactivate a resolved request

- **Description**

Admin and Moderators can reactive a resolved request if the allocatted donor gets unavailable.

- **Input**
  Select ‘re-activate’ under ‘current status’ bar.
- **Processing**
  System will update status as ‘active’, and store details under ‘re-activated’ requests for later statistical analysis.
- **Output**
  . Request gets activated.

RQ 4.9 - Approve student body emails for urgent requests

- **Description**
  Admin/Moderator can approve/disapprove whether to send an email to student body or not.
- **Input**
  Admin/Moderator will get a notification and they have to click on yes to approve and no to disapprove.
- **Processing**
  The response is processed at the server.
- **Output**
  In case of yes email is sent and for no, it is discarded.

5: Donor Catalog Management

RQ 5.1 - Add entry to donor catalog

- **Description**
  Admin/Moderator can add a donor to a catalog.
- **Input**
  Need to fill a form and click add.
- **Processing**
  The add information is verified.
- **Output**
  Donor information is added in the donor catalog.

RQ 5.2 - Remove entry from donor catalog

- **Description**

Admin/Moderator can delete an entry from the catalog.

- **Input**

  Admin/Moderator can select entry from catalog and click delete.
- **Processing**

  The selected entry is removed from catalog.
- **Output**

  An updated catalog is shown to the user.

RQ 5.3 - Edit an entry from donor catalog

- **Description**

  Admin/Moderator can edit an entry in the catalog.
- **Input**

  Admin/Moderator can select entry from catalog and click edit and enter the updated information..
- **Processing**

  The selected entry is updated in the catalog.
- **Output**

  An updated catalog is shown to the user.

6: Admin Control

RQ 6.1 - Create a moderator account

- **Description**

  Admin can create a moderator account on the application.
- **Input**

  Admin will enter the details and press create account.
- **Processing**

  The provided information is verified.
- **Output**

  An account with moderating access is created.

RQ 6.2 - Delete a moderator account

- **Description**

  Admin can delete a moderator account on the application.
- **Input**

  Admin can select an account from a list of moderator accounts and click delete.
- **Processing**

The selected account is deleted.

- **Output**

  An updated list of moderators is displayed.

RQ 6.3 - Set number of urgent requests per day for automatic emails

- **Description**

  An admin can select that how many emails for requests can be sent to student body automatically without admin approval.

- **Input**

  Admin can enter a number and press okay.

- **Processing**

  The limit for automatic emails is set to that number.

- **Output**

  Screen showing new limit for automatic emails is displayed.

3.2 External Interface Requirements

3.2.1 User Interfaces

The users will interact using GUI. The users for our applications include all sort of people. However, our users are not blind and they have access to internet via either a smart phone or a computer/laptop. With the access to these two things, we can reasonably assume that they are screen literate. We will be following the design of popular web apps so that the existing mental model can be used.

We won't just suffice on the transfer of mental model, we will start with low-fi prototyping which will be followed by proper testing. Learning from these low fi prototypes we will develop a high fi prototype on figma keeping in mind the material design guidelines. This will help us create an app whose user interface is welcoming for the audience.

3.2.2 Hardware Interfaces

Since Bloodlink app is a webapp, it can only be accessed through a browser. Although the user can use a smartphone and pc/laptop browsers, they will get the best experience if they use the website on pc/laptop. Another significant requirement is the internet connectivity.

3.2.3 Software Interfaces

The Web application will run on any operating system (Ex: Windows, Linux, MacOs, Android, iOS) which has an updated browser and can run web applications. BloodLink webapp with use Graphic user Interface (GUI) to interact with users. Webapp will require use of browsers (Google Chrome, Microsoft edge, Mozilla firefox, Safari). However, some functionalities might not work in older versions browsers like Microsoft explorer.

3.3Use Case View

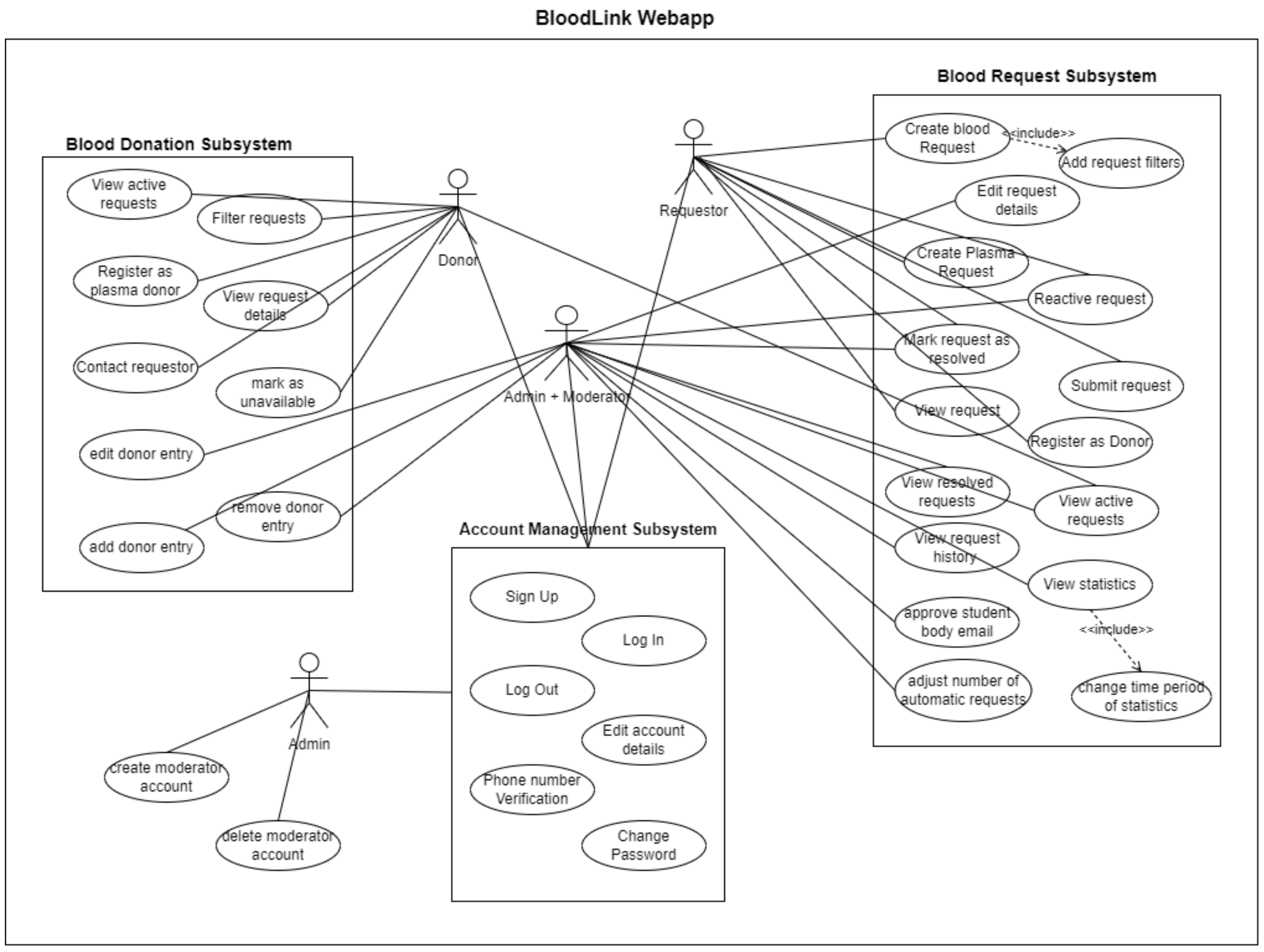
3.3.1 Use Case Table

1. Use Case table

Primary Actor(s)	Associated Use cases	
Guest User	1. Sign Up 2. Log In	
Guest User + Requestors (Non-Donor Registered User)	3.Create blood request 4.Create plasma request 5. Add request filters 6. Submit request 7. View my requests 8. Mark request as resolved 9. Re-activate request 10. Edit request details	
Requestors	11. Register as ‘donor’	
Donor	12. View Active Requests 13. Filter Requests 14. View Request Details	

	15. Contact Requestor 16. Register as 'plasma donor'	
Admin + Moderator	17. View active requests 18. View Resolved requests 19. View requests history 20. View statistics 21. Change time period of statistics 22. Mark as Resolved 23. Edit active request 24. Reactivate resolved request 25. Approve student body emails for urgent requests 26. Add entry to donor catalog 27. Remove entry from donor catalog 28. Edit an entry from donor catalog	
Admin	29. Create moderator account 30. Delete moderator account 31. Adjust email automation	
Admin + Moderator + Donor + Requestors	32. Log Out 33. Phone Number verification 34. Change Password 35. Edit account details	

3.3.2 Use Case Diagram



3.3.3 Use Case Description

Use Case ID	2.1
Use Case Name	Create Blood request

Created By:	Muhammad Farrukh	Last Updated By:	Muhammad Farrukh
Date Created	--	Date Last Updated:	----

Actors	Guest Users, All Registered Users
Description	The blood requestor will be able to add a blood request by mentioning different details like ● Name ● Hospital name ● Blood group ● Quantity of blood required ● Time before which blood is required
Trigger	Clicking the “Add Blood Request” button on the system’s homepage/dashboard.
Preconditions	User just need to go to homepage

Postconditions	User’s request will be added in the system and it will start the process of finding the donor
Normal Flow	User accesses the homepage by logging in or directly User clicks “Add Blood Request” on the homepage User enters the required information and clicks “Post” User’s request will be added in the system and it will start the process of finding the donor
Alternative Flows	1. User Enters wrong information, or leaves out a required field in the input. 2. The appropriate error message is shown to the user on screen, and the request will not be processed
Exceptions	None
Includes	None
Priority	High
Frequency of Use	Frequent: whenever a user has to add a blood request
Business Rules	None

Use Case ID	2.9		
Use Case Name	Registeration as Donor		
Created By:	Muhammad Farrukh	Last Updated By:	Muhammad Farrukh
Date Created	--	Date Last Updated:	----

Actors	Registered User
Description	A user can register themselves as a donor in order to donate blood by providing relevant details ● Name ● Contact Number ● City ● Region in a city ● Blood group

	● Details about medical/disease history
Trigger	Clicking the “Register as a Donor” button on the system’s homepage/dashboard.
Preconditions	User just need to go to homepage
Postconditions	User’s details will be added in the system and he/she will receive blood requests relevant to them
Normal Flow	1. User accesses the homepage by logging in or directly 2. User clicks “Register as a Donor” on the homepage 3. User enters the required information and clicks “Register” 4. User’s details will be added in the system and he/she will receive blood requests relevant to them

Alternative Flows	3. User Enters wrong information, or leaves out a required field in the input. 4. The appropriate error message is shown to the user on screen, and the user will not be registered
Exceptions	None
Includes	None
Priority	High
Frequency of Use	One time when the user has to donate blood very first ime using the bloodlink platform. For later donations, same account will be used
Business Rules	None

Use Case ID	4.9

Use Case Name	Emails Approval for urgent requests		
Created By:	Muhammad Farrukh	Last Updated By:	Muhammad Farrukh
Date Created	--	Date Last Updated:	----

Actors	Admin and Moderator
Description	To avoid the email spamming, automatic emails will only be sent for a certain number of urgent requests. Urgent Requests exceeding the limit will need admin and moderator approval for the email of that request
Trigger	Urgent Requests exceeding the limit set by admin
Preconditions	There should be unresolved urgent request equal to the limit set by the admin

Postconditions	Admin and Moderator will receive an email/sms notification about whether to approve the reques or not
Normal Flow	1. User adds an urgent request 2. The limit for email notoification for urgent requests has not reached. 3. SMS, emails and facebook posting will be done.
Alternative Flows	1. User adds an urgent request 2. User will have to wait for the approval of admin of whether email will be send or not because there are already enough unresolved requests 3. SMS and facebook posting will be done but emails will not be send.
Exceptions	None
Includes	None
Priority	High
Frequency of Use	Frequent: whenever limit for email notoification for urgent requests reach

Business Rules	None

Use Case ID	2.6		
Use Case Name	Marking Request as resolved		
Created By:	Muhammad Farrukh	Last Updated By:	Muhammad Farrukh
Date Created	--	Date Last Updated:	----

Actors	Blood Requestor
Description	The blood requestor will be able to set the status of its request as resolved in case the requestor finds a donor
Trigger	Clicking the "Mark as resolved" button on an auction

Preconditions	● Blood requestor should have added a request ● The requestor should have found a donor
Postconditions	The system will mark the request as resolved and will add that request in the resolved category
Normal Flow	1. Requestor adds a request 2. Requestor found a donor 3. User clicks “Mark as resolved” button 4. System will close the request and no further actions will take place
Alternative Flows	1. Requestor adds a request 2. Requestor does not find any donor and the deadline has passed. 3. System will discard the unresolved request after the deadline.
Exceptions	None
Includes	None
Priority	High

Frequency of Use	Frequent: whenever a request is resolved
Business Rules	None

Use Case ID	1.4		
Use Case Name	Phone Number Verification		
Created By:	Muhammad Farrukh	Last Updated By:	Muhammad Farrukh
Date Created	--	Date Last Updated:	----

Actors	Blood Requestor, Donor, Admin and Moderator

Description	Every user has to verify its contact number in order to register themselves
Trigger	Whenever users register themselves
Preconditions	User should add correct details including contact number in order to register themselves.
Postconditions	User's mobile number will be verified and he/she will registered in the system
Normal Flow	1. User clicks on "Sign Up". 2. User will add details 3. A unique code will be sent to user's mentioned mobile number. 4. user will enter that code 5. User will get registered
Alternative Flows	1. User clicks on "Sign Up". 2. User will add details 3. A unique code will be sent to user's mentioned mobile number. 4. User will not enter correct code that implies the entered mobile number is not correct 5. User will not be be able to register themselves.

Exceptions	None
Includes	None
Priority	Medium
Frequency of Use	Frequent: whenever users register themselves
Business Rules	None

4 Other Non-functional Requirements

4.1Performance Requirements

The performance requirements obtained through quantitative and qualitative data collected from clients and users is as follows:

4 The server should be operational 24/7.

5 The process of sending email, social media posts and SMS should not take more than 1 minute. The main purpose of the application is to significantly increase response and post time, this will help with that.

6 Entering a blood request should not take more than 1 minute. Application should be able to add the request quickly so the posting and donor looking can start. This is to increase response time.

7 Sign-up should not take more than 5 minute. This is for users/donors who want to enter/look at requests as soon as possible.

8 Log-in should not take more than 1 minute. This is an optional bonus feature.

9 Fetching filtered/unfiltered requests should not take more than 1 minute. This is to avoid lag between donors finding requests.

4.2 Safety and Security Requirements

4.2.1 Safety Requirements:

Following safety requirements are associated with use of BloodLink webapp:

1. **Privacy of Donor’s address:**

   Our user survey showed that, around 5 % people are not comfortable in sharing detailed addresses, due to privacy and safety concerns. To counter this, BloodLink app will allow requestors to only see the list of available donors in a certain region, and will hide address details of donors.
2. **Medical condition of donor:**

   Donors must meet certain medical conditions, as regulated by PTBA (Punjab Blood Transfusion Authority). The major checks include

   - Donor is at least 18 years old
   - Has not received recent surgery or long-term medications
   - Is free from blood transmittable diseases (HIV, syphilis, hepatitis).
   - Has not donated blood for the last three months.

   BloodLink webapp will require donors to enter these medical details while registering.

3. **Verification of User details:**

   BloodLink web app should verify user details such as contact numbers to prevent invalid entries.
4. **Donor Safety:**

   BloodLink app will require donors to mark their location status, once they go for blood donation at requestor’s location. This will prevent kidnapping incidents and ensure donor’s safety.
5. **Feedback Mechanism:**

BloodLink app will allow requestors and donors to actively give feedback under communication thread. This will mitigate incidents of misconduct and will help to improve user experience.

4.2.2 Security Requirements:

1. **Communication security:**
   Webapp should use end-to-end encrypted messages for communication.
2. **Data Security:**
   Webapp should use a secure hosting service to prevent data leakage.
3. **Database Backup:**
   Webapp should use a reliable hosting service that provides multiple data backups.

Expected Level of Security:

Users should expect that:

- Data shared on web app will stay secure and free from cyber attacks.
- Personal details including name, phone number, address, medical records will not be shared with some external party.
- Only necessary contact details of donors will be shared with requestors.
- Medical records and addiction details of donors will not be directly visible to requestor. Instead, requestors can add filters while searching for donors.

4.3 Software Quality Attributes

4.3.1 Availability:

The server will be available to all sorts of users with access to the internet on a latest browser. It will be available 24/7. It can be accessed using the internet.

.

4.3.2 Usability:

There is another important feature of our software that is usability, it is easy to understand for all sorts of users. The mental model is very easily developed. The UI is intuitive and easy to follow.

4.3.3 Correctness:

Correctness is another important feature of our application. It means that all of the requests are up to date. As soon as the requestor marks it complete it is updated on all clients, moreover, as soon as someone sends a request it is updated on all client ends.

4.3.4 Portability:

There is another important feature in our application that is portability. It can be accessed using a phone, it can also be accessed using a laptop. All you need is a device with the latest browser.

4.3.5 Testability:

Testability or modularity is another important feature of our application. The web application is easy to test both as a whole and module wise. The web application will go through a test run to ensure that all modules work.

4.3.6 Reusability:

Rusabilty will be another software quality attribute of our software. It can be integrated into other applications to use the same functionality there. Similarly, other software can also be integrated in our application.

4.3.7 Adaptability:

Adaptability is another important feature of our software. It can be adapted depending upon the screen size. Certain features can also be excluded / included based on priorities at that particular time.

4.3.8 Interoperability:

Interoperability is another important feature of our software. It can communicate with other devices. This feature is essential for adding/removing requests.

4.3.9 Maintainability:

Maintainability is another important feature of our application. It can easily be maintained and updated due to extensive documentation, not only by us but also by any other developers. Similarly, all of the requirements can be traced back to their roots, therefore allowing us to maintain it easily.

Appendix A – Top 10 User Stories

1. As a user, I shall be able to request blood, so I can get blood when I need it.
2. As a donor, I shall be able to see blood requests, so I can donate my blood.
3. As a donor, I shall be able to contact the requestors, so I can correspond with them to donate blood.
4. As a user who has requested blood, I shall be able to inform the system when my request has been resolved, so that no one contacts me afterwards.
5. System shall automatically contact donors registered in the Bloodlink webapp and donors present in catalog, so the procedure used by Bloodlink can be automated.
6. System shall only contact donors from the same city as requestor, so donors are not disturbed by
7. System should be able to automatically make posts on social media pages of Bloodlink, so that more donors can see the request.
8. System should be able to email the student body of LUMS for urgent requests, so a donor could be found efficiently.
9. System shall keep a track of blood requests, so the data can be used by the admin when required.
10. As an admin, I should be able to see the number of requests made and resolved per day, so I can track the success rate of this system.

Appendix B – Architectural Spike (One Story)

Proof of concept:

As CS undergrads, we used the MERN stack (MangoDB, Express, React, Node JS) in our Database webapp project. These technologies integrate three-tier architecture (database, frontend, and backend) and work well to develop a fully functional web app. BloodLink webapp will use MERN stack as architecture.

Complex Use case:

BloodLink web app will scan for donors in a region sphere, nearest to requestors' entered location. Implementation of this use case will employ certain shortest path algorithms to provide optimal results to the requestor. However, similar use cases are already implemented in other driver apps like careem, uber that work by integration with google map APIs.

Appendix C - Group Log

- We conducted focus meetings with clients to understand the process of how Bloodlink operates. The focus set meeting recording can be found in the link.
- We held interviews with Bloodlink to understand their requirements.
- Then we transcribed these client interviews. (Please find attached Google Drive link containing recordings and their transcripts, the transcripts can also be accessed here)
- We extracted requirements from these transcriptions and prepared a prototype traceability matrix. (that file can be accessed here)
- Then we conducted a quantitative data analysis on the requirements acquired from client interviews and collected over a hundred responses.
- We also held interviews with users to understand their perspective on a prospective Bloodlink app features and noted their stories. (Please find attached drive link to their interview recordings, it can be accessed here)
- There were some conflicts in the requirements of users and clients, we also did conflict resolution in a meeting with representatives from our client side.
- We held 4 group meetings to discuss the flow of the system, identify and categorize use cases, resolve conflicts in functional and non-functional requirements.

- We then assigned roles to each member for filling the SRS document.

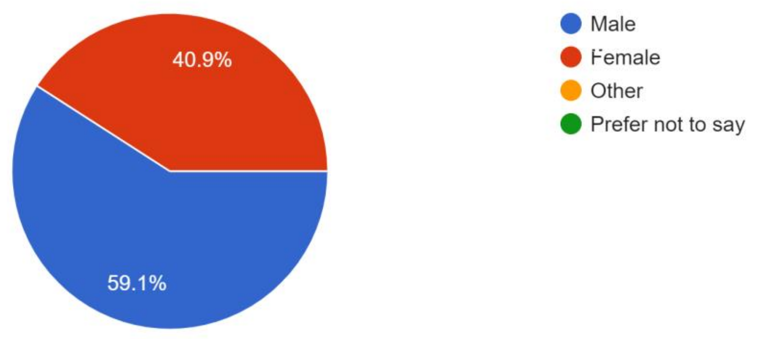
Appendix D – Contribution Statement

Name	*Contributions in this phase*	*Approx. Number of hours*	*Remarks*
Daim Armaghan	*Use Case diagram, Transcription, Srs*	*14*	
Mohammad Ahmad	*Planned and led first focus group meeting with client, translated client interview transcriptions into functional and non-functional requirements, identified and categorized use cases, conducted user interview, SRS: user stories, product scope, use cases (1.4-2.9), other minor contributions.*	*18*	
Mohammad Farrukh	*Interviews,Srs,Transcription,General diagram*	*17*	
Mohammad Hassnain	*Interviews,Srs,Transcription, , survey and interview design, data flow diagram*	*17*	
Sharjeel Ahmed	*Srs, Transcription,survey and interview design*	*12*	

Appendix E – User Survey

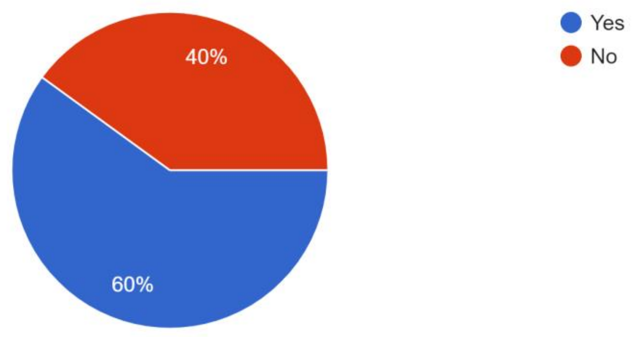
Gender

110 responses



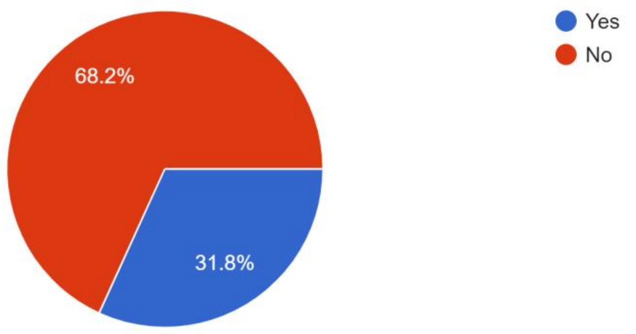
Do you know how to contact Bloodlink?

110 responses



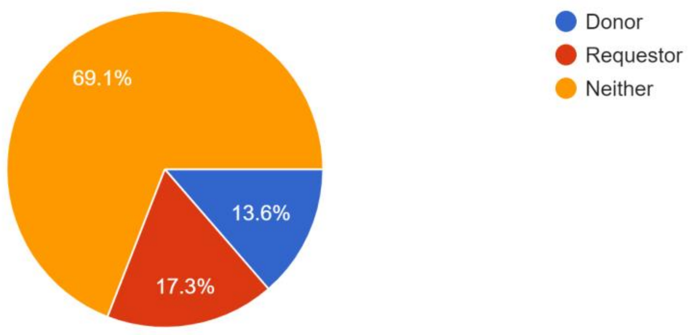
Have you ever used Bloodlink services?

110 responses



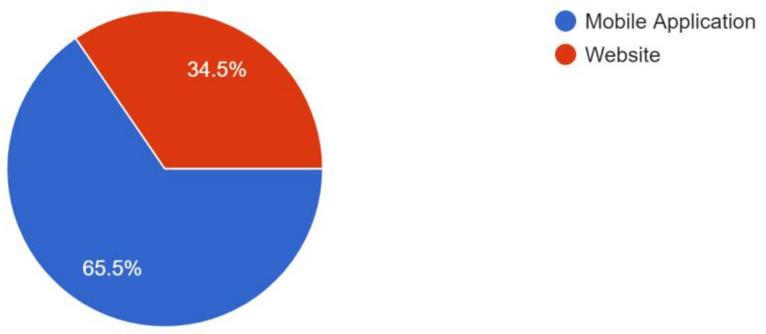
If Yes, were you the donor or the requestor?

110 responses

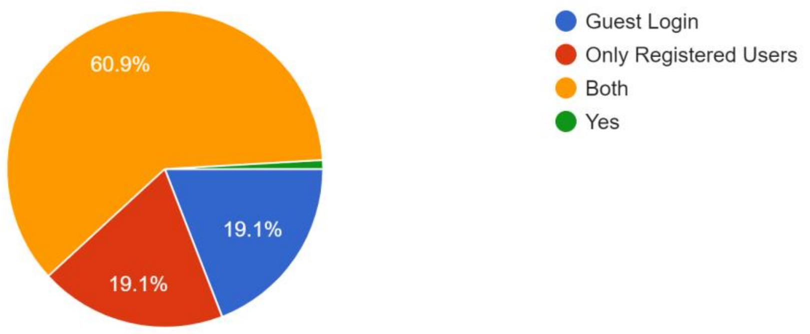


Will you prefer a mobile application or a website?

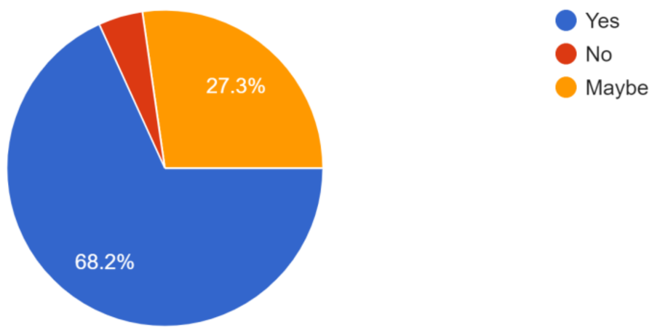
110 responses



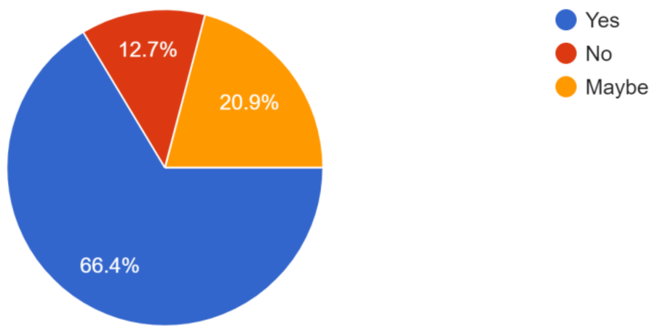
In your opinion should there be a guest login or only registered users should be allowed?
110 responses



Will you be comfortable sharing your location for a location based request system?
110 responses

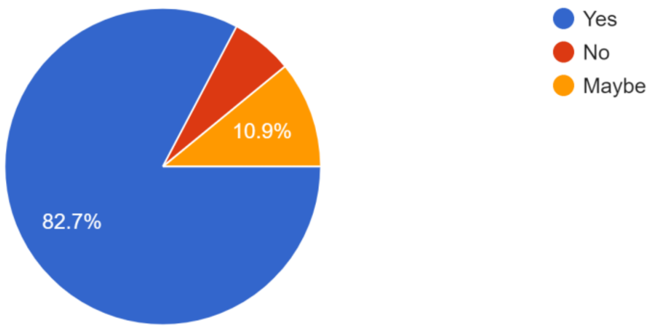


Will you be comfortable sharing your contact number? Where you can get messages in case of a blood request.
110 responses



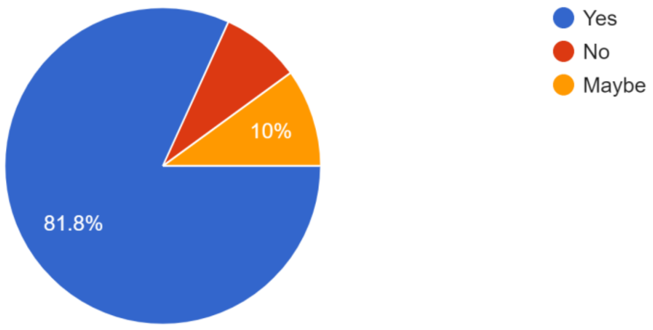
Will you be comfortable sharing your email? Where you can get email in case of a blood request.

110 responses



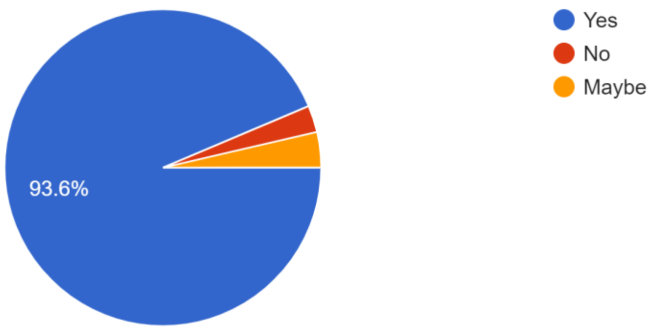
Are you in favor of the opinion that there should be a CNIC verification of the blood requestor to avoid fraudulent cases?

110 responses



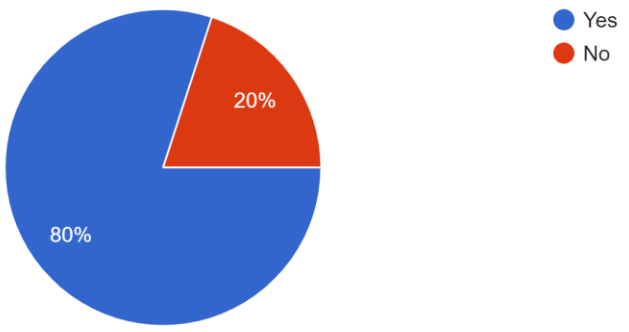
Requestors can add filters for the donors such as non-smokers, vaccinated etc. Will you be comfortable sharing those things with us during the registration process?

110 responses



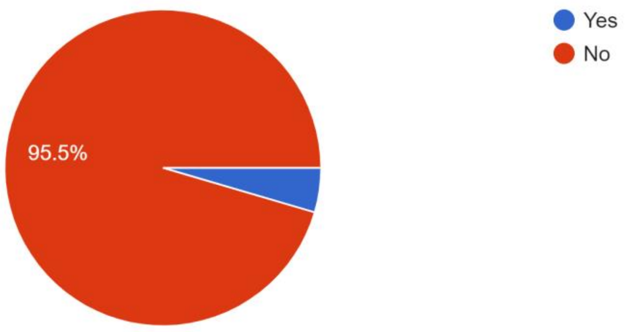
Are you comfortable with sharing your medical records/history (diseases, addictions)?

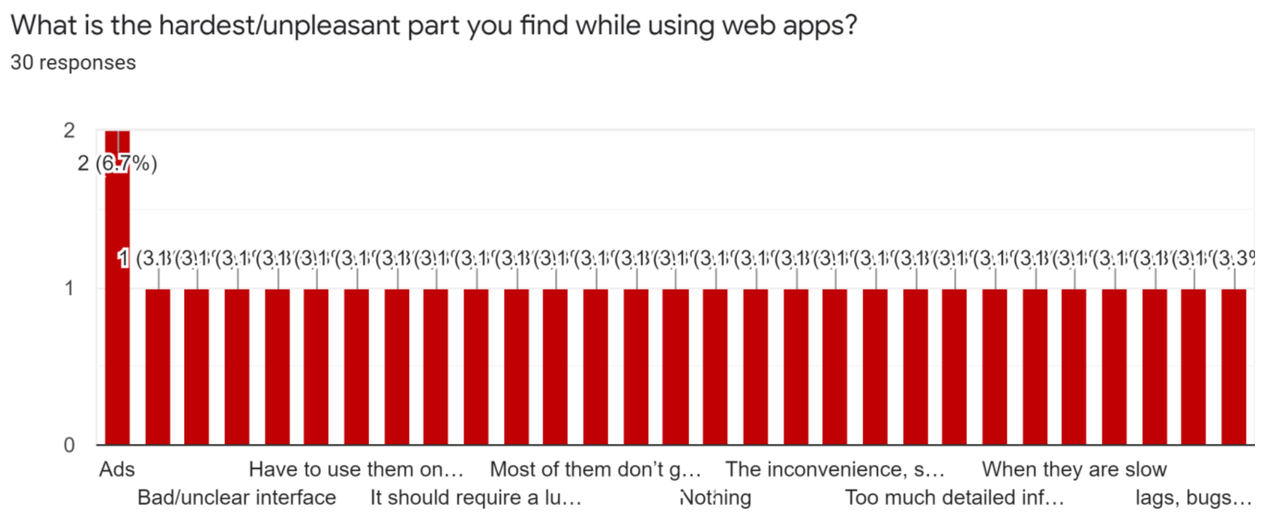
110 responses



Do you know about any other app/website for blood donation requests?

110 responses





Appendix F – User Interview

User Interview

The user interviews can be accessed here

Name:

Have you ever donated or received blood through Bloodlink?

For those who have donated blood before via Bloodlink:

- Any difficulty you faced?
- Any process that could have been done differently?
- How many often do you get calls requesting for blood?

For those who have received blood before via Bloodlink:

- Any issues that you have faced?
- Any process that could have been done in a better way?
- How much time did it take for you to get a donor (if you got one)?
- Did you receive a call asking if blood was arranged so we can close the request?

For Everyone:

- What features do you think Bloodlink App should have?
- Will you prefer a mobile application or a website? Please explain.
- Do you know how to contact Bloodlink for a request (ask only if they say they have never used Bloodlink services)
- Do you know about any other app/website for blood donation requests?
- What is the hardest/unpleasant part you find while using web apps?

As Recipient:

- Are you in favor of the opinion that there should be a CNIC verification of the blood requestor to avoid fraudulent cases? Will you be willing to share your information?

As Donor:

- Would you prefer SMS or Web Notification?
- Will you be comfortable sharing your contact number? Where you can get messages in case of a blood request.
- Will you be comfortable sharing your address to reach out to you if you are in close proximity to the requester?
- One feature in the app could be that the requestor can view nearby donors based on their location. Do you have any privacy concerns about sharing your location?

- Requestors can directly contact you if you are close to their location?
- Will you be comfortable sharing your contact number? Where you can get messages in case of a blood request.
- Will you be comfortable sharing your email? Where you can get messages in case of a blood request.
- Requestors can add filters for the donors such as non-smokers, vaccinated etc. Will you be comfortable sharing those things with us during the registration process?
- Are you comfortable with sharing your medical records/history (diseases, addictions)?

Appendix G – Client Interview

(The transcriptions can also be found in the same folder)

The client interviews can be accessed here

Classification

- Name
- For how long have you been working with Bloodlink?
- How many requests have you catered in total?

Process

- What does the user have to do to initiate a request?
- What actions do you take after receiving a request?
- How do you educate users about the bloodlink process?
- How much time does it take to process a request? (doing everything from your end counting from the time email received)

- Do you offer late night support for requests? If yes then till what time?
- What is the success rate from calling donors in a list?
- What is the success rate from social media posts and stories?
- What is the success rate from emails?
- How many average requests do you receive per day?

Requirements

- What are your requirements?
- Do you ever reject a request? If yes then in what case?
- Do you prioritize requests? If yes then on what basis?
- What if they get multiple responses or someone contacts after request closes
- What if the agreed donor backs up later?
- What do they do in case of multiple requests from the same client?
- When is a request considered closed?

Automation

- Which process needs automation?
- Do you want moderation or complete automation of above selected features?
- Do you want to provide frequent donors with some sort of incentives?
- Any additional features you would want?