PUNJAB
COLLEGES

Software Requirements Specification

Blood Donation System

Department of Computer Science
& Information Technology

Hajra Khan Lodhi	LEF19EBSIT083
Zarnub Khan Lodhi	LEF19EBSIT085
Ayesha Imtiaz	LEF19EBSIT084

Supervisor: Muhammad Shoaib

Punjab College Lahore (Main)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The aim of this project is to develop a mobile application that gathers volunteers who want to donate blood on one platform and alerts them whenever a hospital near them needs blood. The application will also have a feature that sends messages to people who do not know about the application to encourage them to install it and donate blood.

The application will have a user-friendly interface that will enable users to register as donors and specify their blood type and location. Hospitals, blood banks, and the people in need will also be able to register on the application and update their blood inventory levels in real-time. Whenever a hospital, blood bank and the people in need requires blood.

They login in this app as a needy person and fill the form then the application will send an alert to all registered donors with the required blood type who are located within a certain distance from the hospital. Donors can choose to accept or decline the request based on their availability and willingness to donate.

Another key feature of the application will be the ability to track blood donation history. Users will be able to see their previous blood donation records and the impact of their donations on the community.

1.2 Document Conventions

Main heading: Bold
Sub heading: Bold
Main heading font size is 12 and subheading font size is 12.The paragraph font size is 10.That is the font size which I use in my software requirement specification for Blood Donation App. The font is Times New Roman.

1.3 Intended Audience and Reading Suggestions

This document is intended for user, developers, admins, professors. we suggest that reader read the document very well to understand very clearly all the information including the goal of the system, the advantages of the system and how the system work.

The different types of reader's are:

- Customer.
- Developers.
- Management people.

1.4 Product Scope

The system should be able to maintain a database of blood donors, including their contact details, blood type, and other relevant information. Donor registration and approval process should be streamlined.

The system should be able to maintain a database of available blood inventory, including blood type, quantity, and expiration date.

The system should be able to manage blood requests and connect patients in need of blood with the nearest available donor.

2. Overall Description

2.1 Product Perspective

The Blood Donation App is a new, self-contained product that is being developed to enable people to donate blood easily and efficiently. The app is designed to be used by both blood donors and blood banks, and will provide a platform for donors to find nearby blood banks, schedule appointments to donate blood, and track their donation history. The app will also enable blood banks to manage their inventory, track donations, and communicate with donors.

The context of this product is the need for a more streamlined and efficient blood donation process. The traditional method of donating blood involves finding a blood bank, filling out forms, and waiting in line. This process can be time-consuming and inconvenient for donors, and can lead to a shortage of blood for those in need. The Blood Donation App aims to address these issues by providing a user-friendly platform that connects donors and blood banks in a seamless manner.

The app will interface with external systems, such as mapping and location services, to enable users to find nearby blood banks. It will also interface with blood bank inventory systems to enable donors to view the availability of blood types and schedule appointments accordingly. A diagram of the major components of the overall system may include the donor interface, the blood bank interface, the inventory management system, and external interfaces such as mapping and location services.

2.2 Product Functions

To provide a means for the blood bank to publicize and advertise blood donation programs.

To provide an efficient donor and blood stock management functions to the blood bank by recording the donor and blood details.

To improve the efficiency of blood stock management by alerting the blood bank staffs when the blood quantity is below it par level or when the blood stock has expired.

To provide pure blood with no wastages blood is been collected in different types of packs. They are double, triple, and triple (AS), quadruple pack.

To provide synchronized and centralized donor and blood stock database.

To provide immediate storage and retrieval of data and information

- User Interface:

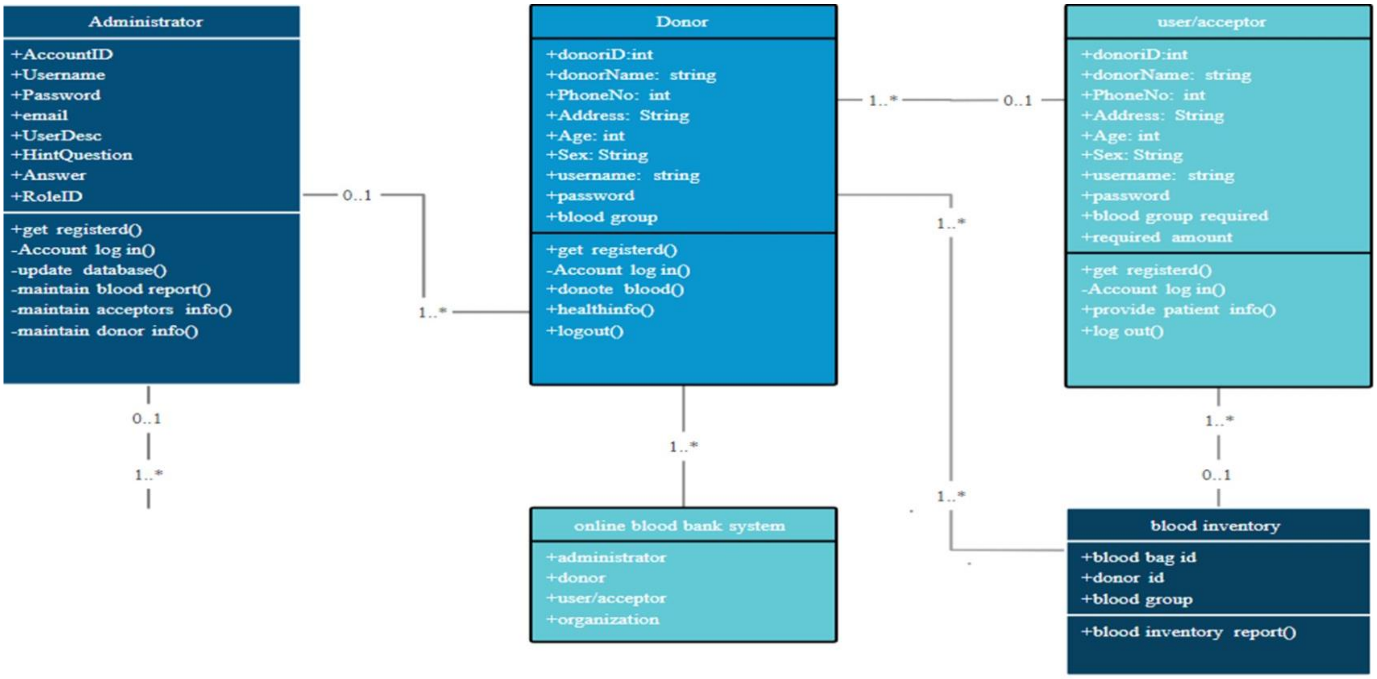
This component will be responsible for providing a user-friendly interface for the customers to interact with the application. It includes the mobile application interface for donors to register, search for nearby blood donation centers and receive alerts.

- Donor Management System: This component manages the donor database and includes features such as donor registration and donor profiles that store information such as blood type, location, and availability to donate blood.
- Analytics and Reporting System:

This component provides data analytics and reporting functionality to monitor blood inventory levels, donor registration and donor satisfaction. It includes features such as dashboard and reporting tools.

- Notification Module:

This module sends notifications to registered donors and volunteers when their blood type or availability is required by hospitals or blood banks. The system can also send reminders to donors and volunteers.



2.3 User Classes and Characteristics

System Owner: The Blood Bank

System Users: Administrators: has full privilege on the system's functions

Public: can view the blood donation events and donate or can make requests for donation (Donor and Recipients fall under this category)

2.4 Operating Environment

The operating system which are used as follow:

Android: Android is an open-source operating system developed by Google and is used to develop apps for mobile devices such as smartphones and tablets.

2.5 Design and Implementation Constraints

Each user must keep their password. More over the user must have individual ID for creating a login in the Blood Donation App. Only Administrator can control user addition and deletion in the system.

2.6 User Documentation

The product is under development stage and requires a complete implemented prototype to explain the user documentation. Once the prototype is designed and implemented online manuals, user manuals can be provided.

2.7 Assumptions and Dependencies

Assumed Factors:

Availability of reliable and stable internet connectivity for the app to function properly.

Users have a smartphone and basic knowledge of how to use mobile applications.

Availability of blood banks and donation centers in the target area where the app will be launched.

Users are willing to donate blood and participate in the blood donation process.

Availability of funding to support the development, maintenance, and marketing of the app.

Integration with third-party services such as payment gateways for donations and SMS gateways for communication with donors.

Compatibility with various smartphone platforms and operating systems.

The accuracy of the data provided by users, such as their blood type, health status, and contact information.

The accuracy and completeness of the data provided by the blood banks and donation centers, such as their availability and location.

Dependencies:

Integration with the database or information system of the blood banks or donation centers to obtain real-time information about blood availability.

Integration with a secure payment gateway to process donations.

Availability of SMS gateway for communication with donors and potential donors.

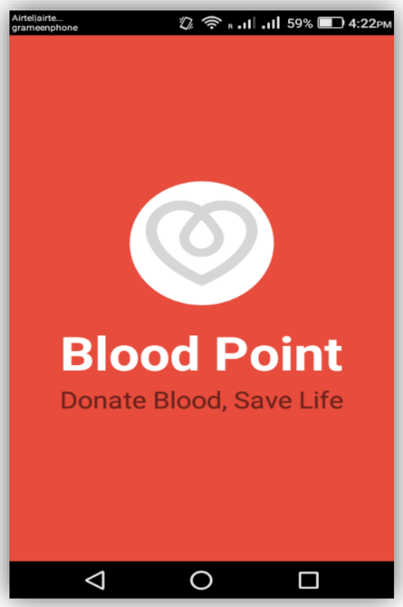
Compatibility with various mobile operating systems and devices.

3. External Interface Requirements

3.1 User Interfaces

A. Home screen:

On home screen we will provide Blood Donation Information, such as why it's important, the benefits of donating, and who can donate. This can help educate users and encourage them to become donors. This will also include the logo of the app. when the user will download the app the first pop up will be a little info of blood donation importance and a out logo of Blood unity app.



B. User registration:

- User registration will have login page and a sign-up option.
- Login page require information as Email and password.
- Sign up page for creating an account for user to use the app.

User registration will have following options on screen

1. Sign-up as donor
2. Sign-up as seeker

Sign up as donor will have following details:

- Enter name
- Gender
- Blood group
- Contact info (Email, Phone No.)
- Password
- Verify password
- Location
- Health info check box
- Available to donate check box

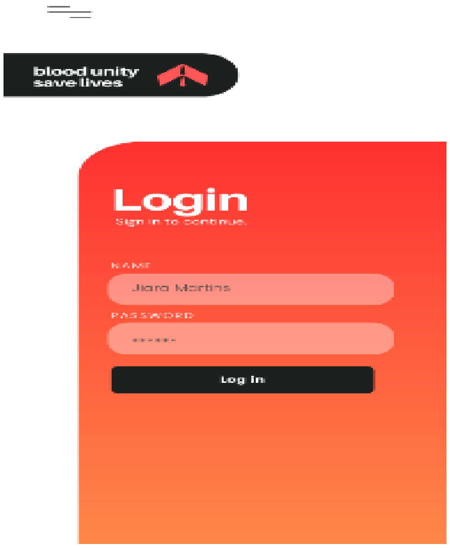
Health info will require details about user’s following things:

- Age
- Weight
- health issues (if any)

Sign up as seeker will have following details:

- Email
- Password
- Phone No.
- Location

After filling out these details’ user will have an account and now, they can login to use the app.



blood unity
save lives

Login
Sign in to continue.

NAME
Jiara Martins

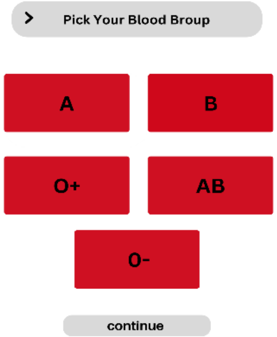
PASSWORD
******

Log in

C. Blood Bank Module:

This module will contain a blood group information like user can use search filter to search blood group they want And they will have a list of donors available for that blood group.
It will display info about donor’s profile through which user can view the donor and can select the most suitable one for them.
Right beside the donor’s information, a chat or call option will be given so the user can directly contact the donor and request for blood.

Blood group	User info	Contact here
A-	Lionel Messi 123 Main Street Apt 4B LA. 0317-2345671	☎ 💬

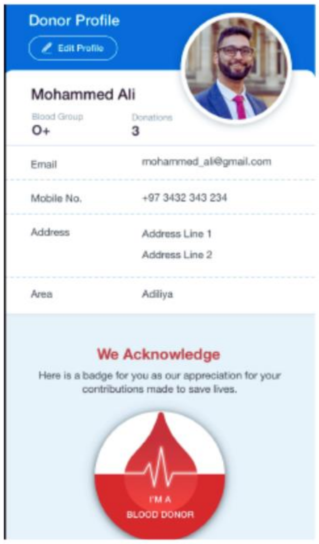


D. Profile module:

In this module user can edit their profile such as username, password or reset password in case they forgot or want to change in any case.

They can set reminder notification to ON for alerts to donate blood whenever needed.

It will also contain a calendar to set the date of their last donation



Mohammed Ali

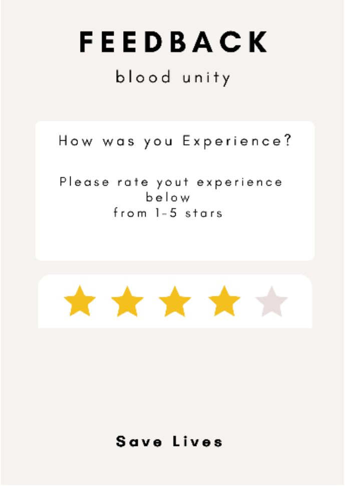
Blood Group	Donations
O+	3
Email	mohammed_ali@gmail.com
Mobile No.	+97 3432 343 234
Address	Address Line 1 Address Line 2
Area	Adiliya

We Acknowledge

Here is a badge for you as our appreciation for your contributions made to save lives.

E. Feedback Module:

Feedback module contains a form where user can enter their experience while using the app. It will contain a comment box and rating of the app.



FEEDBACK

blood unity

How was you Experience?

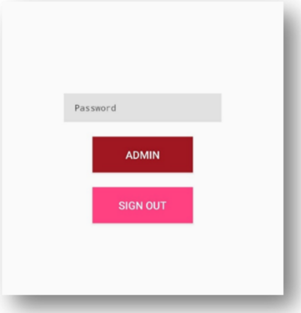
Please rate yout experience below from 1-5 stars

★★★★☆

Save Lives

F. Admin module:

It allows administrators to manage and monitor the app's operations. The admin module will allow administrators to manage blood donations, including tracking inventory levels, recording donations, and updating donor information.
The admin module will provide a way to manage notifications, including creating and sending notifications to users.
The admin module will provide a way to manage app content, including updating text, images, and other app assets. This can help keep the app up-to-date and relevant for users.
It will contain admin username and password and a sign-out button.



3.2 Hardware Interfaces

Mobile Devices: The primary hardware interface for a Blood Donation System app is the mobile device used by the user. This can include smartphones and tablets. These devices should have a compatible operating system and be able to connect to the internet.
Input Devices: The input devices used by the user to interact with the Blood Donation System app can include touchscreens, keyboards, and keypads. The input devices should be easy to use and accessible to all voters.
Display Devices: The display devices used by the user to view the Blood Donation System app can include the mobile device's screen. The display devices should be clear and easy to read
Servers: The servers used by the Blood Donation System app store and process the data and allow user to access the system. The servers should be secure, scalable, and reliable.

3.3 Software Interfaces

Blood Donation System is a multi-user, multi-tasking environment. It enables the user to interact with the server and attain access to the Internet and leaves a record in the inbuilt database. It uses React native as the programming tool and firebase as the backend application tool. It includes user interface that is the main software interface that allows the user to interact with the Blood Donation System app. It also includes APIs that will allow the app to retrieve and manipulate data from these sources and perform various functions such as Provide location access t both donor or seeker , to send alert messages to donor when seeker request for blood. It has the data storage interface which enables the Blood Donation System app to store and retrieve data from the database. It has a security interface that is a set of protocols and standards that ensure the Blood Donation System app is secure from unauthorized access and tampering. It has a reporting interface that enables the Blood Donation System app to generate reports.

3.4 Communications Interfaces

The Blood Donation System uses React and hence requires HTTP for transmission of data. The Blood Donation System uses HTTPS to protect sensitive data .Moreover, this allows easy interaction between the various clients and the server. It uses SMS that is a protocol used for sending text messages. The Blood Donation System may use SMS to send messages to people about the awareness of blood donation TCP is a protocol that ensures reliable and ordered delivery of data over the internet. The Blood Donation System uses TCP to establish and manage connections between different components of the system

4. System Feature

4.1 System Feature 1

4.1.1 Description and Priority

The system offers access to Internet at client level and access to server resources at server level only by validating the user with the unique username and password.

4.1.2 Stimulus/Response Sequences

The response for the different classes of users is:

1) Users: - Login.

2) Administrator: Adding new accounts, giving account status, getting & sending reports

4.1.3 Functional Requirements

User registration: Users should be able to create an account on the app by providing their basic information like name, contact details, blood group, and location.

Blood donation request: Users should be able to request blood donation by posting a notification on the app. The request should include the required blood group and the location where the donor is required.

Search for blood donors: Users should be able to search for donors based on their location and blood group.

Donation history: The app should maintain a record of blood donations made by the user.

Push notifications: Users should receive notifications when a blood donation request is posted in their vicinity.

Donor profile: The app should maintain a profile of donors with their contact details, blood group, and location.

Availability status: Donors should be able to update their availability status on the app.

Feedback system: Users should be able to provide feedback on the donation process and the donor.

1. Other Nonfunctional Requirements

5.1 Performance Requirements

It is available during all 24 hours.
The process of Donation is continous all the time
The seeker should find the required blood when it is needed

5.2 Safety Requirements

All the data must be backed up every 24 hours and the copies will be stored in the some other server in the other location

5.3 Security Requirements

Your database would be secure if another wants to change in it then your system should be restricted him/her

5.4 Software Quality Attributes

Usability: The app should be easy to use and navigate for both donors and recipients. The app should also have a user-friendly interface and be responsive to user actions. The target is to have a score of at least 90% in usability testing.

Reliability: The app should be reliable in terms of uptime and availability. Users should be able to access the app at all times and in all locations. The target is to have a 99.9% uptime over a period of 6 months.

Robustness: The app should be able to handle errors and unexpected situations. It should not crash or lose data during usage.

5.5 Business Rules

- Only registered and qualified donors can create a profile on the app.
- Donors must pass a health screening process to be eligible to donate blood.
- Donors must wait a certain period of time between donations, as determined by their blood type and other factors
- The app should send reminders to donors when they are eligible to donate again or when there is an urgent need for their blood type.

1. Other Requirements

There are no other requirements in this phase. If some extra requirements are wanted by the customer or acquirer, these are added in this part later

Use Cases:

UC Number: 1.1

UC Name: Sign- up

Functional Requirement No: FR1

Primary Actors/Stakeholders: Seeker, Donor

Description: This use case describes how a sign up to the system. System gives the access to the user/seeker, donor to create their account to use the System. A new user wishes to register for the blood bank app as donor or seeker.

Preconditions: The system must be in the logout state and has the Sign-up screen displayed.

Main Success Scenario (MSS):

1. User opens the app and clicks on the "Sign Up" button.
2. The app displays the Sign Up page, prompting the user to enter their personal details, including name, gender, blood group, contact info (email, phone number), password, and health information.
3. User fills out the required fields and clicks on the "Submit" button.
4. The app validates the user's input and checks if the email and phone number are not already associated with an existing account.
5. If the input is valid and the email and phone number are unique, the app creates a new user account and displays a success message.
6. The user can now log in with their newly created account details.

Alternative Scenario:

1. If any field mentioned is missing/blank that field will be highlighted and show message please fill/select this field
2. If email already exist than Message will be displayed This Account Already Exist try another

Basic flow:

1. User clicks on "Sign Up" button.
2. User fills out the required fields.
3. User clicks on "Submit" button.
4. App validates input and creates a new account.

Post conditions:

If the use case was successful, System will inform the user that his account is created successfully and go to the home page where he can login and proceed further.

Extensions:

1. Error Generate if all necessary information are not filled.

2. If corrected data entered then error generated.

3. If data is not submitted successfully then error is generated.

UC Number: 2

UC Name: Login

Functional Requirement No: FR1

Primary Actors/Stakeholders: Seeker, Donor

Description: This use case describes how a sign in to the system. System gives the access to them according to their roles.

Preconditions: The system is in the login state and has the login screen displayed.

Main Success Scenario (MSS):

1. User opens the app and is presented with the login screen.
2. User enters their email and password.
3. System verifies the email and password.
4. System logs the user into their account and redirects them to the home screen.

Basic flow:

1. The user opens the GUI.
2. System displays the "Log in" prompt.
3. The user enters his/her Email and password.
4. If donor clicks "Login" button app will provide him access donor panel and system plays same functionality with other users according to their role assigned.
5. The system validates the entered name and password ad logs the user into the system.

Alternative Scenario:

1. If user has no account.
   a) System gives option of creating account.
   b) If user enters invalid name and/or password.
   c) System displays an error message
2. If user is not authorized to login.
   a) System displays the unknown user message and terminates the use case.
3. Software should give the option to recover the password**.**

Post conditions:

If the use case was successful, the actor is now logged into the system, if not, the system state is unchanged.

Extensions:

1. If the email or password is incorrect, the system displays an error message to the user and prompts them to enter their login credentials again.

UC Number: 3

UC Name: Change password

Actors: seeker, donor

Description: This use case is about to give user rights to change her/his profile password.

Preconditions: login is required for change password.

Basic Flow:

1. Enter old password.
2. The user enters his/her username and password.
3. Enter new password.
4. Confirm new password.
5. Click 'change password' button.

Alternative Scenario:

1. User enters incorrect current password and is prompted to enter the correct one before proceeding with password

change.

2. User enters a new password that does not meet complexity requirements and is prompted to enter a new password that meets them.

Post conditions:

1. Go back to profile page.

Exceptions:

1. If password does not successfully updated then error message shown.

2. If old and new password is same then error message shown.

UC Number: 4

UC Name: Forgotten password

Actors: seeker, donor

Description: This case is about to a user of a system or application has forgotten their password and needs to reset it in order to gain access to their account. This scenario is a common occurrence in today's digital world, where people have numerous accounts with different websites and applications.

Preconditions:

The pre-condition of the Forgotten Password scenario is that the user must have previously created an account with the system and must have forgotten their password. The system should have a mechanism in place to enable password resets, such as a password reset tool

Alternative Scenario:

1. The user may try to reset the password themselves using a password reset tool provided by the system

2. The user may contact customer support to request a password reset or assistance with resetting their password.

3. The user may try to create a new account with the system if they cannot reset their password.

Post conditions:

1. The user should be able to reset their password successfully.

2. The user regain access to their account

3. The system should also store the new password securely and update their records accordingly.

Exceptions:

1. If the user is unable to verify their identity during the password reset process. In such cases, the system may require additional information or verification steps to confirm the user's identity before allowing them to reset their password.

2. If the system or application experiences technical issues or downtime, the user may not be able to reset their password until the issue is resolved.

UC Number: 5

UC Name: Invalid email

Actors: seeker and donor

Description: This use case is about a user of a system attempts to register or login using an email address that does not meet the required format or is not recognized by the system as a valid email address. This often happens when people mistype or misspell their email addresses.

Preconditions:

The user must have attempted to register or login using an email ID that does not meet the required format or is not recognized by the system as a valid email address. The system or application should have a mechanism in place to validate email IDs entered by users.

Alternative Scenario:

1. The user may be prompted to enter a valid email address and try again.
2. The user may be directed to a help or support page that provides information on how to format a valid email address.
3. The user may be given the option to contact customer support for assistance with their email ID.

Post conditions:

1. The user should receive an error message indicating that their email ID is invalid and should be prompted to enter a valid email ID.
2. The system should not allow the user to proceed until a valid email ID is entered.

Exceptions:

1. If the system or application incorrectly identifies a valid email ID as invalid, or if the system is unable to validate email IDs due to technical issues or downtime.
2. In such cases, the user may be directed to contact customer support for assistance with their email ID.

UC Number: 6

UC Name: view and edit profile

Actors: donor and seeker

Description: This use case allows a user to view & edit their profile, set their last donation date, and turn on/off the donation alert feature.

Preconditions: Login is required to see profile.

Main Success Scenario (MSS):

1. The user selects the "Edit Profile" option from the menu.
2. The app displays the user's current profile information, including name, contact info, and health details.
3. The user updates the desired fields, such as email or phone number.
4. The app validates the user's input and updates the profile with the new information.
5. The app displays a success message indicating that the profile has been updated

Basic Flow:

- → The donor / seeker can see their profile in detail and make necessary changes if required in profile as desire.
- → User selects "Profile" from the app menu.
- → App displays the user's current profile information.
- → User selects "Edit Profile".
- → App displays the "Edit Profile" screen.
- → User makes changes to their profile information and sets their last donation date.
- → User selects the "Save Changes" button.
- → App saves the new profile information and last donation date.

Alternate flow:

1. If the user selects "Cancel" instead of "Save Changes", the app will return to the previous screen without saving any changes.
2. If the user tries to set a last donation date that is too recent (within the last 8 weeks), the app will display an error

message and prevent the user from saving the changes until they enter a valid date.

3. If the user selects the donation alert OFF option, the app will turn off the donation alert feature and save the changes.

Post conditions:

1. User's profile information is updated with the new changes.
2. Last donation date is set and saved.
3. Donation alert feature is turned on/off according to the user's preference.

Exceptions:

None

UC Number: 7

UC Name: find donor

Functional Requirement No: FR1

Primary Actors/Stakeholders: Seeker

Description: A user wants to search for available blood donors in the blood bank app.

Preconditions:

1. The user has launched the blood bank app.
2. The user is registered and logged in to the app.
3. The app has information about available blood donors, including their blood groups and contact information.

Basic flow:

1. The user selects the "Find Donors" option on the app's home screen.
2. The app displays a list of available blood groups for the user to select.
3. The user selects the desired blood group from the list and selects "Search."
4. The app displays a list of available donors with the selected blood group, along with their contact information and other details.
5. The user selects a donor from the list and selects "Send Request."

Alternative Scenario:

If there are no donors available for the selected blood group, the app displays a message informing the user that there are no matches found.

Post conditions:

1. The user has successfully searched for and found a donor for their requested blood group.
2. The user has sent a request for blood donation to the selected donor**.**

Extensions:

Scenario: No matching donors found

1. The user enters the required information for the donor search.
2. The app searches the donor database for matching donors based on the search criteria.
3. If no matching donors are found, the app displays a message informing the user that no donors were found.

UC Number: 8

UC Name: submit feedback

Functional Requirement No: FR1

Primary Actors/Stakeholders: Seeker and donor

Description: This use case allows the user to submit feedback about the Blood unity.

Preconditions:

1. The user must be logged in to the app.
2. The user must have navigated to the feedback section of the app.

Basic flow:

1. The user selects the feedback option from the app's menu.
2. The app displays a form for the user to enter their feedback.
3. The user enters their feedback in the form.
4. The user selects the submit button.

Alternative Scenario:

If the user encounters an error while submitting feedback, the app will display an error message and prompt the user to try again.

Post conditions:

1. The feedback will be submitted to the Blood Bank App administrators.
2. The user will receive a confirmation message that their feedback has been submitted.

Extensions:

None

UC Number: 9

UC Name: logout

Actor: seeker and donor

Description: Actors firstly log-in in this system as per as their roles

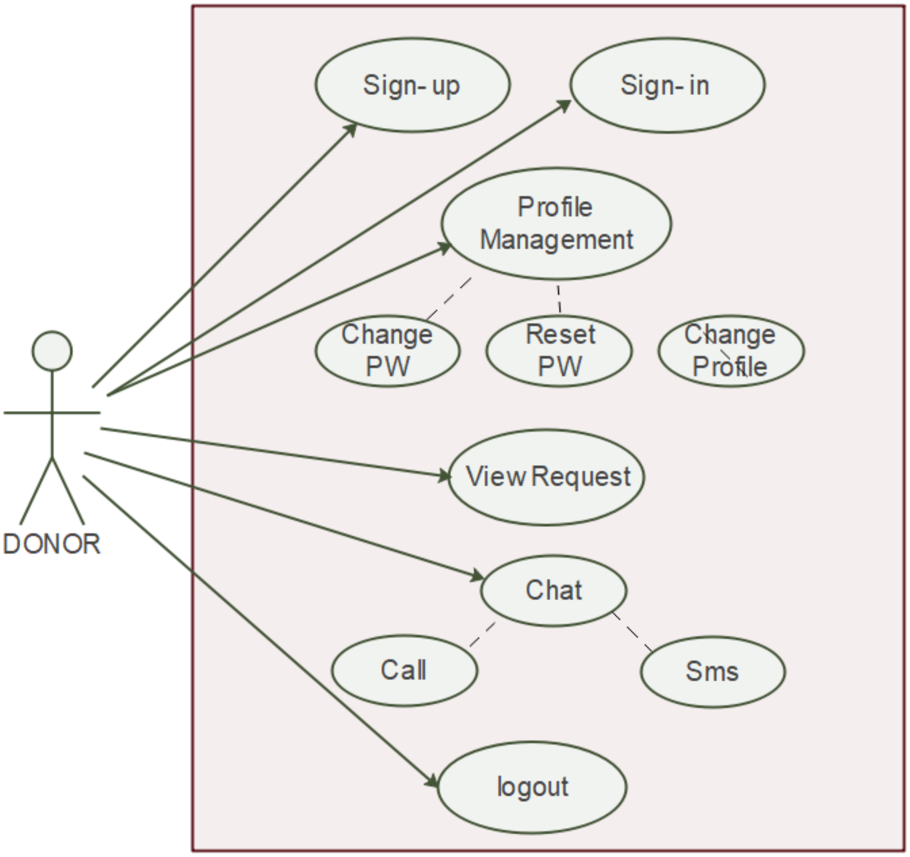
Preconditions: Must log in

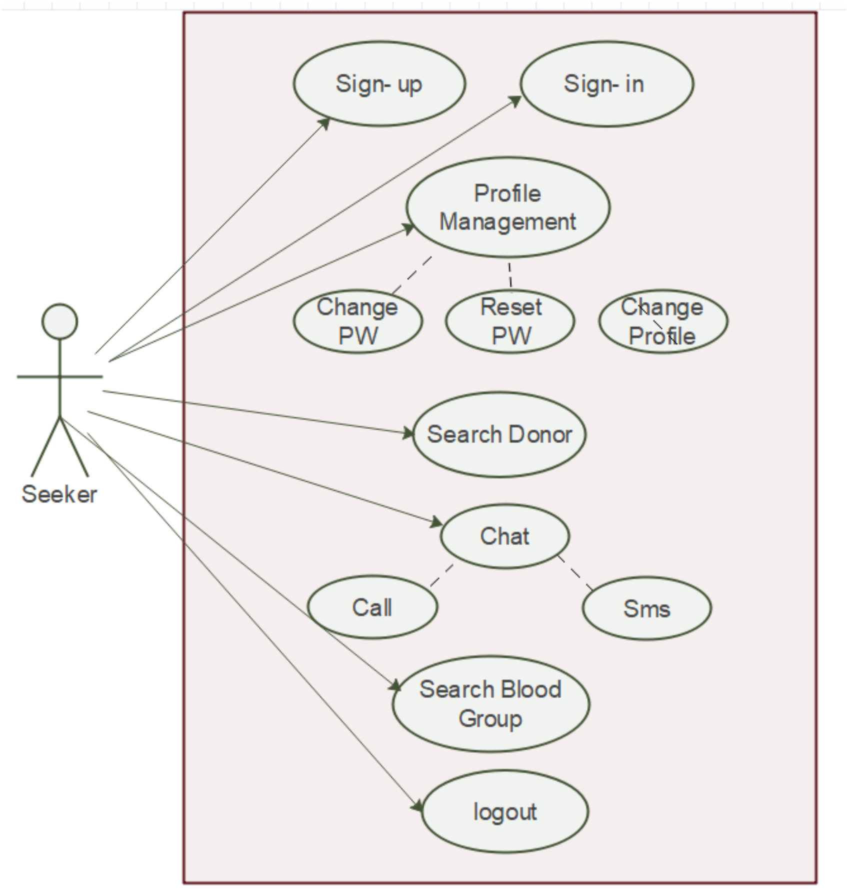
Basic flow: Sign up or log in required for users.

Alternate flow: Not Log out

Post condition: Redirect to main page of application.

Extension: None


Sign- up
Sign- in
Profile Management
Change PW
Reset PW
Change Profile
View Request
Chat
Call
Sms
logout
DONOR



TEST CASES:

Test Case ID: 01

Test Case Name: Login

Test Priority: High

Cross reference: Sign- up

Preconditions:

1. The user has a valid email and password.
2. The app is installed and opened on the user's device.
3. The user is not already logged in.

Post conditions:

1. The user is successfully logged in and can use the app's features.

Test Case ID: 02

Test Case Name: change password

Test Priority: High

Description: This test case is designed to verify that users can change their password successfully

Preconditions:

1. User must be logged into the blood donation app.

2. User must have access to the "change password" feature

Post conditions: The user is successfully logged in and can use the app's features.

Test Case ID: 03

Test Case Name: Search Donors by Blood Group

Test Priority: High

Cross reference: find donor

Description: This test case verifies that the user is able to find donors based on the required blood group

Preconditions:

1. The user has an account on the blood bank app.
2. The user is logged in to their account.

Test Case ID: 04

Test Case Name: Login with invalid email

Test Priority: High

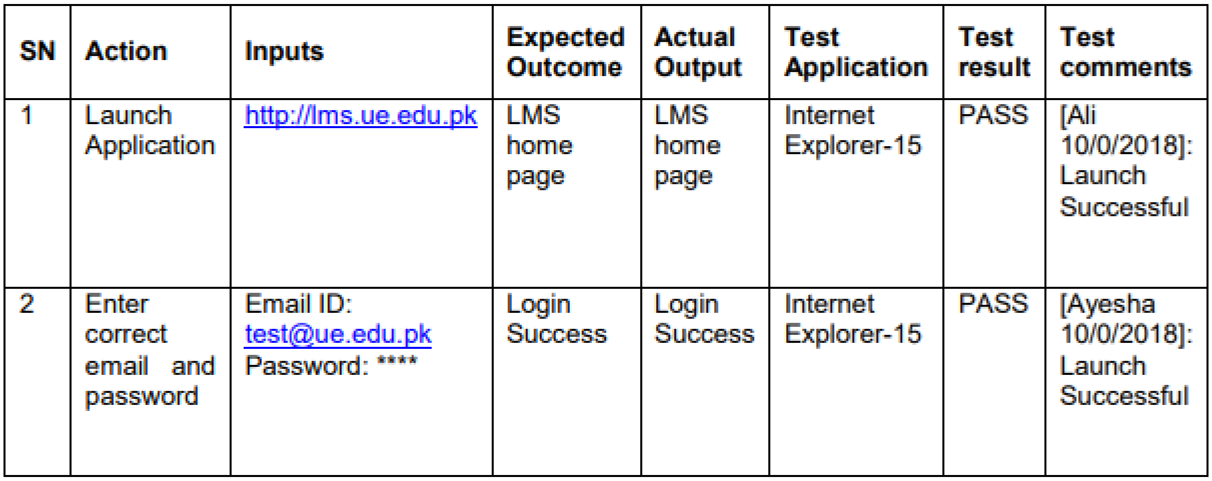
Cross reference: this test case is related to login use case.

Test case description: Test the system behavior when the user enters an invalid email during the login process.

Preconditions:

1. The app is installed and launched on the device.
2. The user is on the login page.

Post conditions: The user should be prompted to enter a valid email address in order to login.



SN	Action	Inputs	Expected Outcome	Actual Output	Test Application	Test result	Test comments
1	Launch Application	http://lms.ue.edu.pk	LMS home page	LMS home page	Internet Explorer-15	PASS	[Ali 10/0/2018]: Launch Successful
2	Enter correct email and password	Email ID: test@ue.edu.pk Password: ****	Login Success	Login Success	Internet Explorer-15	PASS	[Ayesha 10/0/2018]: Launch Successful